Basic Computer Coding: HTML

2nd Edition

BIBLIOTEX
Digital Library

BASIC COMPUTER CODING: HTML

2nd Edition

BIBLIOTEX
Digital Library

www.bibliotex.com

BASIC COMPUTER CODING: HTML

2ND EDITION

BIBLIOTEX
Digital Library

www.bibliotex.com
email: info@bibliotex.com

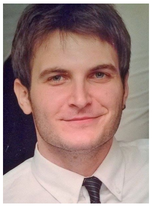
e-book Edition 2022
ISBN: 978-1-98467-601-6 (e-book)

This book contains information obtained from highly regarded resources. Reprinted material sources are indicated. Copyright for individual articles remains with the authors as indicated and published under Creative Commons License. A Wide variety of references are listed. Reasonable efforts have been made to publish reliable data and views articulated in the chapters are those of the individual contributors, and not necessarily those of the editors or publishers. Editors or publishers are not responsible for the accuracy of the information in the published chapters or consequences of their use. The publisher assumes no responsibility for any damage or grievance to the persons or property arising out of the use of any materials, instructions, methods or thoughts in the book. The editors and the publisher have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission has not been obtained. If any copyright holder has not been acknowledged, please write to us so we may rectify.

Notice: Registered trademark of products or corporate names are used only for explanation and identification without intent of infringement.

In Collaboration with 3G E-Learning LLC. Originally Published in printed book format by 3G E-Learning LLC with ISBN 978-1-98465-894-4

EDITORIAL BOARD



Aleksandar Mratinković was born on May 5, 1988 in Arandjelovac, Serbia. He has graduated on Economic high school (2007), The College of Tourism in Belgrade (2013), and also has a master degree of Psychology (Faculty of Philosophy, University of Novi Sad). He has been engaged in different fields of psychology (Developmental Psychology, Clinical Psychology, Educational Psychology and Industrial Psychology) and has published several scientific works.

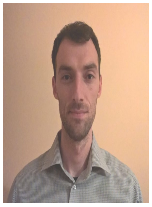
Dan Piestun (PhD) is currently a startup entrepreneur in Israel working on the interface of Agriculture and Biomedical Sciences and was formerly president-CEO of the National Institute of Agricultural Research (INIA) in Uruguay. Dan is a widely published scientist who has received many honours during his career including being a two-time recipient of the Amit Golda Meir Prize from the Hebrew University of Jerusalem, his areas of expertise includes stem cell molecular biology, plant and animal genetics and bioinformatics. Dan's passion for applied science and technological solutions did not stop him from pursuing a deep connection to the farmer, his family and nature. Among some of his interest and practices counts enjoying working as a beekeeper and onboard fishing.

Hazem Shawky Fouda has a PhD. in Agriculture Sciences, obtained his PhD. From the Faculty of Agriculture, Alexandria University in 2008, He is working in Cotton Arbitration & Testing General Organization (CATGO).

Felecia Killings is the Founder and CEO of LiyahAmore Publishing, a publishing company committed to providing technical and educational services and products to Christian Authors. She operates as the Senior Editor and Writer, the Senior Writing Coach, the Content Marketing Specialist, Editor-in-Chief to the company's quarterly magazine, the Executive and Host of an international virtual network, and the Executive Director of the company's online school for Authors. She is a former high-school English instructor and professional development professor. She possesses a Master of Arts degree in Education and a Bachelor's degree in English and African American studies.

Dr. Sandra El Hajj, Ph.D. in Health Sciences from Nova Southeastern University, Florida, USA is a health professional specialized in Preventive and Global Health. With her 12 years of education obtained from one of the most prominent universities in Beirut, in addition to two leading universities in the State of Florida (USA), Dr. Sandra made sure to incorporate interdisciplinary and multicultural approaches in her work. Her long years of studies helped her create her own miniature world of knowledge linking together the healthcare field with Medical Research, Statistics, Food Technology, Environmental & Occupational Health, Preventive Health and most noteworthy her precious last degree of Global Health. Till today, she is the first and only doctor specialized in Global Health in the Middle East area.

Fozia Parveen has a Dphil in Sustainable Water Engineering from the University of Oxford. Prior to this she has received MS in Environmental Sciences from National University of Science and Technology (NUST), Islamabad Pakistan and BS in Environmental Sciences from Fatima Jinnah Women University (FJWU), Rawalpindi.



Igor Krunic 2003-2007 in the School of Economics. After graduating in 2007, he went on to study at The College of Tourism, at the University of Belgrade where he got his bachelor degree in 2010. He was active as a third-year student representative in the student parliament.Then he went on the Faculty of science, at the University of Novi Sad where he successfully defended his master's thesis in 2013. The crown of his study was the work titled Opportunities for development of cultural tourism in Cacak". Later on, he became part of a multinational company where he got promoted to a deputy director of logistic. Nowadays he is a consultant and writer of academic subjects in the field of tourism.

Dr. Jovan Pehcevski obtained his PhD in Computer Science from RMIT University in Melbourne, Australia in 2007. His research interests include big data, business intelligence and predictive analytics, data and information science, information retrieval, XML, web services and service-oriented architectures, and relational and NoSQL database systems. He has published over 30 journal and conference papers and he also serves as a journal and conference reviewer. He is currently working as a Dean and Associate Professor at European University in Skopje, Macedonia.

Dr. Tanjina Nur finished her PhD in Civil and Environmental Engineering in 2014 from University of Technology Sydney (UTS). Now she is working as Post-Doctoral Researcher in the Centre for Technology in Water and Wastewater (CTWW) and published about eight International journal papers with 80 citations. Her research interest is wastewater treatment technology using adsorption process.

Stephen obtained his PhD from the University of North Carolina at Charlotte in 2013 where his graduate research focused on cancer immunology and the tumor microenvironment. He received postdoctoral training in regenerative and translational medicine, specifically gastrointestinal tissue engineering, at the Wake Forest Institute of Regenerative Medicine. Currently, Stephen is an instructor for anatomy and physiology and biology at Forsyth Technical Community College.

Michelle holds a Masters of Business Administration from the University of Phoenix, with a concentration in Human Resources Management. She is a professional author and has had numerous articles published in the Henry County Times and has written and revised several employee handbooks for various YMCA organizations throughout the United States.

HOW TO USE THE BOOK

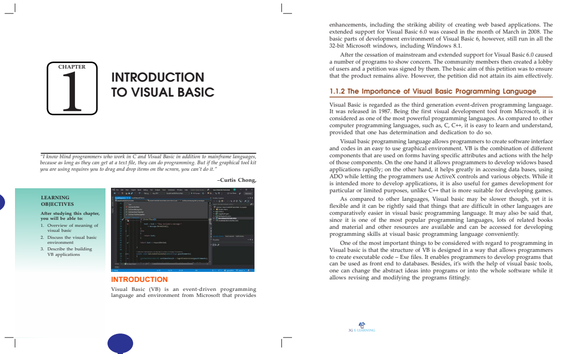
This book has been divided into many chapters. Chapter gives the motivation for this book and the use of templates. The text is presented in the simplest language. Each paragraph has been arranged under a suitable heading for easy retention of concept. Keywords are the words that academics use to reveal the internal structure of an author's reasoning. Review questions at the end of each chapter ask students to review or explain the concepts. References provides the reader an additional source through which he/she can obtain more information regarding the topic.

LEARNING OBJECTIVES

See what you are going to cover and what you should already know at the start of each chapter

ABOUT THIS CHAPTER

An introduction is a beginning of section which states the purpose and goals of the topics which are discussed in the chapter. It also starts the topics in brief.



CHAPTER 1

INTRODUCTION TO VISUAL BASIC

"I know blind programmers who work in C and Visual Basic in addition to mainframe languages, because as long as they can get at a text file, they can do programming. But if the graphical tool kit you are using requires you to drag and drop items on the screen, you can't do it."

–Curtis Chong,

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Overview of meaning of visual basic
2. Discuss the visual basic environment
3. Describe the building VB applications

INTRODUCTION

Visual Basic (VB) is an event-driven programming language and environment from Microsoft that provides

enhancements, including the striking ability of creating web based applications. The extended support for Visual Basic 6.0 was ceased in the month of March in 2008. The basic parts of development environment of Visual Basic 6, however, still run in all the 32-bit Microsoft windows, including Windows 8.1.

After the cessation of mainstream and extended support for Visual Basic 6.0 caused a number of programs to show concern. The community members then created a lobby of users and a petition was signed by them. The basic aim of this petition was to ensure that the product remains alive. However, the petition did not attain its aim effectively.

1.1.2 The Importance of Visual Basic Programming Language

Visual Basic is regarded as the third generation event-driven programming language. It was released in 1987. Being the first visual development tool from Microsoft, it is considered as one of the most powerful programming languages. As compared to other computer programming languages, such as, C, C++, it is easy to learn and understand, provided that one has determination and dedication to do so.

Visual basic programming language allows programmers to create software interface and codes in an easy to use graphical environment. VB is the combination of different components that are used on forms having specific attributes and actions with the help of those components. On the one hand it allows programmers to develop widows based applications rapidly; on the other hand, it helps greatly in accessing data bases, using ADO while letting the programmers use ActiveX controls and various objects. While it is intended more to develop applications, it is also useful for games development for particular or limited purposes, unlike C++ that is more suitable for developing games.

As compared to other languages, Visual basic may be slower though, yet it is flexible and it can be rightly said that things that are difficult in other languages are comparatively easier in visual basic programming language. It may also be said that, since it is one of the most popular programming languages, lots of related books and material and other resources are available and can be accessed for developing programming skills at visual basic programming language conveniently.

One of the most important things to be considered with regard to programming in Visual basic is that the structure of VB is designed in a way that allows programmers to create executable code – Exe files. It enables programmers to develop programs that can be used as front end to databases. Besides, it's with the help of visual basic tools, one can change the abstract ideas into programs or into the whole software while it allows revising and modifying the programs fittingly.

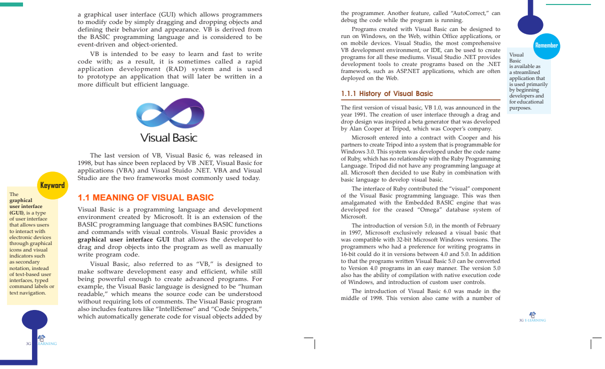
3G E-LEARNING

REMEMBER

This revitalizes a must read information of the topic.

KEYWORDS

This section contains some important definitions that are discussed in the chapter. A keyword is an index entry that identifies a specific record or document. It also gives the extra information to the reader and an easy way to remember the word definition.



a graphical user interface (GUI) which allows programmers to modify code by simply dragging and dropping objects and defining their behavior and appearance. VB is derived from the BASIC programming language and is considered to be event-driven and object-oriented.

VB is intended to be easy to learn and fast to write code with; as a result, it is sometimes called a rapid application development (RAD) system and is used to prototype an application that will later be written in a more difficult but efficient language.

The last version of VB, Visual Basic 6, was released in 1998, but has since been replaced by VB .NET, Visual Basic for applications (VBA) and Visual Stuido .NET. VBA and Visual Studio are the two frameworks most commonly used today.

Keyword

The **graphical user interface (GUI)**, is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

1.1 MEANING OF VISUAL BASIC

Visual Basic is a programming language and development environment created by Microsoft. It is an extension of the BASIC programming language that combines BASIC functions and commands with visual controls. Visual Basic provides a **graphical user interface GUI** that allows the developer to drag and drop objects into the program as well as manually write program code.

Visual Basic, also referred to as "VB," is designed to make software development easy and efficient, while still being powerful enough to create advanced programs. For example, the Visual Basic language is designed to be "human readable," which means the source code can be understood without requiring lots of comments. The Visual Basic program also includes features like "IntelliSense" and "Code Snippets," which automatically generate code for visual objects added by the programmer. Another feature, called "AutoCorrect," can debug the code while the program is running.

3G E-LEARNING

Programs created with Visual Basic can be designed to run on Windows, on the Web, within Office applications, or on mobile devices. Visual Studio, the most comprehensive VB development environment, or IDE, can be used to create programs for all these mediums. Visual Studio .NET provides development tools to create programs based on the .NET framework, such as ASP.NET applications, which are often deployed on the Web.

Remember

Visual Basic is available as a streamlined application that is used primarily by beginning developers and for educational purposes.

1.1.1 History of Visual Basic

The first version of visual basic, VB 1.0, was announced in the year 1991. The creation of user interface through a drag and drop design was inspired a beta generator that was developed by Alan Cooper at Tripod, which was Cooper's company.

Microsoft entered into a contract with Cooper and his partners to create Tripod into a system that is programmable for Windows 3.0. This system was developed under the code name of Ruby, which has no relationship with the Ruby Programming Language. Tripod did not have any programming language at all. Microsoft then decided to use Ruby in combination with basic language to develop visual basic.

The interface of Ruby contributed the "visual" component of the Visual Basic programming language. This was then amalgamated with the Embedded BASIC engine that was developed for the ceased "Omega" database system of Microsoft.

The introduction of version 5.0, in the month of February in 1997, Microsoft exclusively released a visual basic that was compatible with 32-bit Microsoft Windows versions. The programmers who had a preference for writing programs in 16-bit could do it in versions between 4.0 and 5.0. In addition to that the programs written Visual Basic 5.0 can be converted to Version 4.0 programs in an easy manner. The version 5.0 also has the ability of compilation with native execution code of Windows, and introduction of custom user controls.

The introduction of Visual Basic 6.0 was made in the middle of 1998. This version also came with a number of

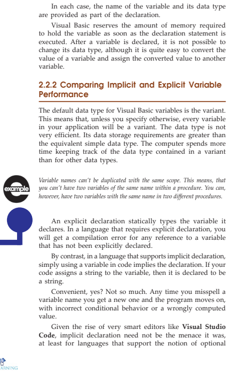
3G E-LEARNING

DID YOU KNOW?

This section equip readers the interesting facts and figures of the topic.

EXAMPLE

The book cabinets' examples to illustrate specific ideas in each chapter.



In each case, the name of the variable and its data type are provided as part of the declaration.

Visual Basic reserves the amount of memory required to hold the variable as soon as the declaration statement is executed. After a variable is declared, it is not possible to change its data type, although it is quite easy to convert the value of a variable and assign the converted value to another variable.

2.2.2 Comparing Implicit and Explicit Variable Performance

The default data type for Visual Basic variables is the variant. This means that, unless you specify otherwise, every variable in your application will be a variant. The data type is not very efficient. Its data storage requirements are greater than the equivalent simple data type. The computer spends more time keeping track of the data type contained in a variant than for other data types.

Variable names can't be duplicated with the same scope. This means, that you can't have two variables of the same name within a procedure. You can, however, have two variables with the same name in two different procedures.

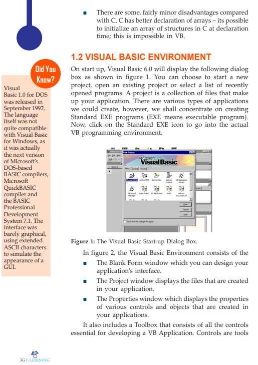
An explicit declaration statically types the variable it declares. In a language that requires explicit declaration, you will get a compilation error for any reference to a variable that has not been explicitly declared.

By contrast, in a language that supports implicit declaration, simply using a variable in code implies the declaration. If your code assigns a string to the variable, then it is declared to be a string.

Convenient, yes? Not so much. Any time you misspell a variable name you get a new one and the program moves on, with incorrect conditional behavior or a wrongly computed value.

Given the rise of very smart editors like **Visual Studio Code**, implicit declaration need not be the menace it was, at least for languages that support the notion of optional

3G E-LEARNING



- There are some, fairly minor disadvantages compared with C. C has better declaration of arrays – its possible to initialize an array of structures in C at declaration time; this is impossible in VB.

1.2 VISUAL BASIC ENVIRONMENT

On start up, Visual Basic 6.0 will display the following dialog box as shown in figure 1. You can choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that make up your application. There are various types of applications we could create, however, we shall concentrate on creating Standard EXE programs (EXE means executable program). Now, click on the Standard EXE icon to go into the actual VB programming environment.

Did You Know?

Visual Basic 1.0 for DOS was released in September 1992. The language itself was not quite compatible with Visual Basic for Windows, as it was actually the next version of Microsoft's DOS-based BASIC compilers, Microsoft QuickBASIC compiler and the BASIC Professional Development System 7.1. The interface was barely graphical, using extended ASCII characters to simulate the appearance of a GUI.

Figure 1: The Visual Basic Start-up Dialog Box.

In figure 2, the Visual Basic Environment consists of the

- The Blank Form window which you can design your application's interface.
- The Project window displays the files that are created in your application.
- The Properties window which displays the properties of various controls and objects that are created in your applications.

It also includes a Toolbox that consists of all the controls essential for developing a VB Application. Controls are tools

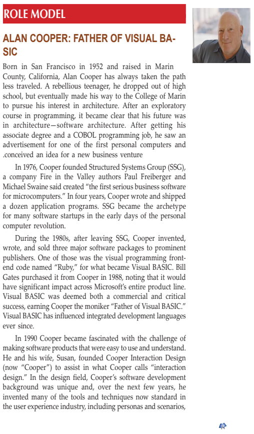
3G E-LEARNING

ROLE MODEL

A biography of someone who has/had acquired remarkable success in their respective field as Role Models are important because they give us the ability to imagine our future selves.

CASE STUDY

This reveals what students need to create and provide an opportunity for the development of key skills such as communication, group working and problem solving.



ROLE MODEL

ALAN COOPER: FATHER OF VISUAL BASIC

Born in San Francisco in 1952 and raised in Marin County, California, Alan Cooper has always taken the path less traveled. A rebellious teenager, he dropped out of high school, but eventually made his way to the College of Marin to pursue his interest in architecture. After an exploratory course in programming, it became clear that his future was in architecture—software architecture. After getting his associate degree and a COBOL programming job, he saw an advertisement for one of the first personal computers and .conceived an idea for a new business venture

In 1976, Cooper founded Structured Systems Group (SSG), a company Fire in the Valley authors Paul Freiberger and Michael Swaine said created "the first serious business software for microcomputers." In four years, Cooper wrote and shipped a dozen application programs. SSG became the archetype for many software startups in the early days of the personal computer revolution.

During the 1980s, after leaving SSG, Cooper invented, wrote, and sold three major software packages to prominent publishers. One of those was the visual programming front-end code named "Ruby," for what became Visual BASIC. Bill Gates purchased it from Cooper in 1988, noting that it would have significant impact across Microsoft's entire product line. Visual BASIC was deemed both a commercial and critical success, earning Cooper the moniker "Father of Visual BASIC." Visual BASIC has influenced integrated development languages ever since.

In 1990 Cooper became fascinated with the challenge of making software products that were easy to use and understand. He and his wife, Susan, founded Cooper Interaction Design (now "Cooper") to assist in what Cooper calls "interaction design." In the design field, Cooper's software development background was unique and, over the next few years, he invented many of the tools and techniques now standard in the user experience industry, including personas and scenarios,

CASE STUDY

FUJITSU FACILITATES SMOOTH MIGRATION TO VB.NET AT AN POST

Fujitsu has an excellent technical team, which works closely with our staff. We have had a good working relationship for many years and Fujitsu has an in-depth knowledge of our mission-critical application gained from several years' development and support work.

Challenge

A Post, one of Ireland's largest companies, is a major commercial organization providing a wide range of postal, communication, retail and financial services. With 9,600 employees throughout its national network of retail, processing and delivery points, the business also provides services to government departments, the National Treasury Management Agency and its own National Lottery Company. A decade ago, A Post implemented a new nationwide time and attendance system to calculate and record staff salary and wages functions. The Staff Remuneration and Administration Management System (STREAMS) is a bespoke, mission-critical application developed by Fujitsu as a reliable, scalable client server system using Microsoft technologies. The STREAMS front-end system gathers information and feeds the data to the company's HR, payroll and financial departments. It primarily creates more efficient processes for A Post to capture data for the weekly payroll run whilst simultaneously minimizing the number of payroll queries by employees. Following deployment, STREAMS improved cost center reporting, significantly lowered the time to record pay details and enhanced the processing of casual staff pay. During this period, Fujitsu provided quality support and maintenance services and application enhancements to increase functionality, ensuring the long-term reliability of STREAMS. For instance, as employee numbers steadily increased to exceed original expectations, Fujitsu boosted system performance by upgrading the infrastructure and optimizing the software. STREAMS originally employed Visual Basic (VB), a third-generation event-driven programming language and integrated development environment (IDE) from Microsoft. IDE provides programmers with comprehensive facilities for software development and comprises a source code editor, a compiler and/or an interpreter, build automation tools and a debugger. However, Microsoft no longer supports VB version 6.0, the edition employed by A Post. Syl Byrne, IT Manager Remuneration Services, A Post, explains: "To ensure that our business-critical application is future-proof, we needed to move to a platform that Microsoft will support for the foreseeable future." A Post therefore decided to migrate STREAMS to the VB.NET platform, an object-orientated programming language. This strategy would protect its investment for the next 10 years by creating a secure, scalable

3G E-LEARNING

KNOWLEDGE CHECK

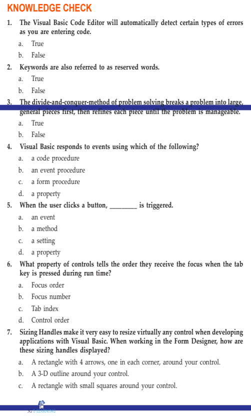
This is given to the students for progress check at the end of each chapter.

REVIEW QUESTIONS

This section is to analyze the knowledge and ability of the reader.

REFERENCES

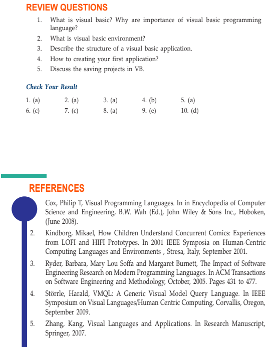
References refer those books which discuss the topics given in the chapters in almost same manner.



KNOWLEDGE CHECK

1. **The Visual Basic Code Editor will automatically detect certain types of errors as you are entering code.**
   a. True
   b. False
2. **Keywords are also referred to as reserved words.**
   a. True
   b. False
3. **The divide-and-conquer-method of problem solving breaks a problem into large, general pieces first, then refines each piece until the problem is manageable.**
   a. True
   b. False
4. **Visual Basic responds to events using which of the following?**
   a. a code procedure
   b. an event procedure
   c. a form procedure
   d. a property
5. **When the user clicks a button, ________ is triggered.**
   a. an event
   b. a method
   c. a setting
   d. a property
6. **What property of controls tells the order they receive the focus when the tab key is pressed during run time?**
   a. Focus order
   b. Focus number
   c. Tab index
   d. Control order
7. **Sizing Handles make it very easy to resize virtually any control when developing applications with Visual Basic. When working in the Form Designer, how are these sizing handles displayed?**
   a. A rectangle with 4 arrows, one in each corner, around your control.
   b. A 3-D outline around your control.
   c. A rectangle with small squares around your control.

3G E-LEARNING



REVIEW QUESTIONS

1. What is visual basic? Why are importance of visual basic programming language?
2. What is visual basic environment?
3. Describe the structure of a visual basic application.
4. How to creating your first application?
5. Discuss the saving projects in VB.

Check Your Result

1. (a)   2. (a)   3. (a)   4. (b)   5. (a)
6. (c)   7. (c)   8. (a)   9. (e)   10. (d)

REFERENCES

Cox, Philip T, Visual Programming Languages. In in Encyclopedia of Computer Science and Engineering, B.W. Wah (Ed.), John Wiley & Sons Inc., Hoboken, (June 2008).

2. Kindborg, Mikael, How Children Understand Concurrent Comics: Experiences from LOFI and HIFI Prototypes. In 2001 IEEE Symposia on Human-Centric Computing Languages and Environments , Stresa, Italy, September 2001.
3. Ryder, Barbara, Mary Lou Soffa and Margaret Burnett, The Impact of Software Engineering Research on Modern Programming Languages. In ACM Transactions on Software Engineering and Methodology, October, 2005. Pages 431 to 477.
4. Störrle, Harald, VMQL: A Generic Visual Model Query Language. In IEEE Symposium on Visual Languages/Human Centric Computing, Corvallis, Oregon, September 2009.
5. Zhang, Kang, Visual Languages and Applications. In Research Manuscript, Springer, 2007.

TABLE OF CONTENTS

PREFACE

HTML or Hypertext Markup Language is a programming language employed to describe the structure of information on a webpage. In a combined way, HTML, CSS, and JavaScript make up the vital building blocks of websites globally, with CSS controlling the appearance of a webpage and JavaScript programming and its functionality. This grooms a business standard in the market. A good design takes user by the hand and directs them where they need to go and find the information they need quickly and easily. HTML can be a little intimidating for those not accustomed to looking at code. That's why this book is designed as an introduction to web design basics. This book will break down the basics of how to get your HTML page created. This book will take you through what HTML is, how to create an HTML page, HTML basics, and what program you can use to edit the HTML.

Organization of the Book

This edition is comprised of eight chapters. In this book, you'll learn how to design and build beautiful websites by learning the basic principles of design like branding, color theory, and typography which are all instrumental in the design process of a website. You'll also learn HTML and CSS, which are the common code languages that all modern websites are built on. **Chapter 1** introduces web design principles that explores basic principles involved in developing a web site, planning process, five golden rules of web designing, designing navigation bar, page design, home page layout, and design concept.

Chapter 2 focuses on basics in web design that gives brief history of Internet, World Wide Web why create a Website, Web Standards, audience requirement of Web, etc.

Chapter 3 presents an introduction to HTML. In this chapter, you will find HTML Page Structure, Doctype, and Heading-Paragraphs, including HTML Version, Line Breaks, and HTML Tags.

Chapter 4 gives an exploration on the elements of HTML. Meta Tag and Div Tag are given in this chapter. You will know working with Lists, Tables, Hyperlinks, Images, and Forms.

Chapter 5 is aimed to discuss the elements of HTML5. In this chapter, you will know about Header Tag, Footer Tag, Section Tag, Nav Tag, Aside Tag, and Video Tag.

Chapter 6 begins with the concept of CSS, and further Creating Style Sheet, CSS Properties, and CSS Styling are also described. Working with block elements and objects, and Lists and Tables is also explored.

Chapter 7 is web publishing or hosting explores creating and saving the Web Site. Working on the web site and creating web site structure and titles for web pages is also discussed.

Chapter 8 focuses on JavaScript and external content. JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

CHAPTER
1

BASICS IN WEB DESIGN

"The success of every websites now depends on search engine optimization and digital marketing strategy. If you are on first page of all major search engines then you are ahead among your competitors in terms of online sales."

– Dr. Christopher Dayagdag

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Understand the overview of web design
2. Explain brief history of internet

INTRODUCTION

Web design is the planning and creation of websites. This includes a number of separate skills that all fall under the umbrella of web design. Web design has numerous components that work together to create the finished experience of a website, including graphic design, user experience design, interface design, search engine optimization (SEO) and content creation. These elements determine how a website looks, feels and works on various devices. Check out this in-depth guide on building a website, if you want to learn more about the whole process.

b

Web design is different from web development, which is the actual coding that makes a website work. When you're building a website, you need both web design and web development. Although you can find web designers who are also web and UX developers, these are distinct skill sets.

Keyword

Web accessibility refers to the inclusive practice of removing barriers that prevent interaction with, or access to websites, by people with disabilities.

Web designers take your ideas and turn them into a mockup that shows what your future website will look like. Web designers handle the creative part of designing a website.

Web developers—also sometimes called engineers or coders—take the mockup your web designer made and translate it into a coding language so it can be displayed on the web. They make websites functional, which often means custom-coding widgets and other tools.

1.1 OVERVIEW OF WEB DESIGN

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with **web accessibility** guidelines.

Everything about your website - including the content, the way it looks, and the way it works - is determined by the website design. Web design is a process of conceptualizing, planning, and building a collection of electronic files that determine the layout, colors, text styles, structure, graphics, images, and use of interactive features that deliver pages to your site visitors. Professional Web design helps to make your business appear credible online. When you are ready to build a website, you have several options. If you are not familiar with HTML programming language and Web design applications, there are a number of other options for creating your Web design:

- Hire a professional website design expert to create and build a unique Web site.
- Create your own website design using off-the-shelf software such as FrontPage or Dreamweaver. (These software packages do require technical expertise.)
- Use online website design tools with pre-set and/or customizable Web design templates to customize with your company colors, graphics and text.

The degree of customization included in your website design will determine the speed and cost of getting your site up and running.

1.1.1 The History of Website Design

The web today is a growing universe of interlinked web pages and web apps, teeming with videos, photos, and interactive content. What the average user does not see is the interplay of web technologies and browsers that makes all this possible.

The internet is a relatively new invention but boy have things changed in its short life! The internet has changed the way we live and it has been responsible for the creation of thousands upon thousands of jobs that simply would not exist without it. One of those categories of job is web design, something that we would sorely miss now if it disappeared. What would we do without the animations? The colorful backgrounds, the fancy writing and the music playing in the background?

When did it Begin?

In 1990, Tim Berners-Lee developed the very first web browser, and it was called WorldWdeWeb, although it was later renamed as Nexus. At that time, only text could be displayed on a web page. No fancy fonts, no pretty pictures or videos, just simple plain text, with links underlined in blue. In 1993 Mosaic was released, the first ever web browser that allowed developers to add images to their web pages. It was able to support .gif images and web forms, a massive leap forward for the time. Design was not brilliant because of the constraint of the browser and to a limit in bandwidth programmers rather than designers designed most websites.

Mid 1990's to 2000

By the mid-nineties, Netscape was the top web browser but it was soon knocked off its pedestal by Internet Explorer and so began the war of the browsers. Around this time, web design began to get a little more complex, using frames and tables as well as images. Animated .gif images could be used to create buttons on sites and JavaScript began to appear on some websites. From 1998, we began to see the introduction of web development tool kits. Remember Dreamweaver? GoLive? These began to be more popular as they gave a larger number of users access to web page creation. Jobs in web design began to grow as more designers were offered jobs to build sites. Flash technology also made its appearance during this era of web site design although it was not all that popular to start off with.

2000 -2004

In the year 2000, the bubble burst and hundreds of thousands of web businesses crashed. However, while this may have put the clamps on things for a while, it was not for long. Web design standards began to pick up again. Now we started to see a better class of design. We got designs that were not based on tables, we got transparency with .png images and CMS began to grow in popularity. Content management System was a program that allowed designers to publish content on the web. They could go back in and edit what they had published and modify it as they saw fit.

2007 – 2004

Web 2.0 was born in 2004. This was the era of bold websites, sites that were aimed at communities. There was bold typography and shiny gradients. Corners became rounded, edges softened and web design, once again, took off at the speed of light. Websites began to be more functional and needed more in the way of an interface to work properly. Widgets were introduced all over the place to help integrate one site with another. This was more often, where a social network site was involved, lining outside feeds to the site, or lining from the site to a blog. This era was also marked by an increase in accessibility of websites to common people. Developments such as WordPress and Blogger, along with user-friendly guides on how to make a website helped every day people make a website without having to learn HTML or CSS.

to the Present 2008

Web site design has evolved incredibly over the last few years and one thing that has given it a push, unbelievably, was the iPhone. Mobile website design was introduced, allowing people to view sites properly on their phones. Many of the bigger websites created mobile versions of their sites specifically for the smartphone and the tablets that were fast becoming popular devices. On the internet itself, the large and fast growing social network sites created more widgets for user to put on their blogs and other websites created widgets designed to go on social network sites. In design, typography increased tremendously and grid-based designs are fast becoming the norm. Today, website design is a huge business. Designs are more complex yet less cluttered. Early websites were difficult to navigate; today, a well-designed website is enough to ensure your business will succeed. In terms of design, where the internet goes from now is anyone’s guess. We have color, we have fonts and we have images. We can even embed videos into websites now so who knows where the next trend will take us.

Remember

Content management is an ongoing process and you should update your website frequently. You should gather a possible list of topics and make those into separate pages.

1.1.2 How to Learn Web Design

Web design is a very good skill to have, especially in today’s Internet-centered world. If you are really good, you can make money from clients, start membership websites, and pursue other money-making projects. If you want to know how to learn web design, you are in the right place.

Beginning

Note what web designs differs from. Web design is not book design, it is not poster design, it is not illustration, and the highest achievements of those disciplines are not what web design aims for. Although websites can be delivery systems for games and videos, and although those delivery systems can be lovely to look at, such sites are exemplars of game design and video storytelling, not of web design.

Understand what web design involves. Web design is the creation of digital environments that:

- facilitate and encourage human activity;
- reflect or adapt to individual voices and content; and
- change gracefully over time while always retaining their identity.

Determine what resource you want to use. The main resources are websites, videos, and books. Some popular websites are Codecademy and W3Schools. However, there are many more, so feel free to experiment!

- Sign up for an account if you have chosen a website. You will be able to save your progress this way.
- If you have chosen videos, bookmark all the videos you will need.
- Go to your local library or bookstore if you have chosen the book route. If you want something cheaper (or even free), download eBooks or PDF files.
- If you are willing to pay, you might be able to get individualized sections from a professional web designer.

Figure out how much time you will need. You will need to figure out how to fit this into your day if you are a busy person.

Download a web design program if you have not already. While Notepad will work, it is a good idea to have a program like Adobe Dreamweaver, Microsoft Expression Web, or KompoZer. There are many, many others, so browse around and get what you like.

Learning

Get started with HTML. HTML is an absolute requirement for any web designer. Master the concept of tags, classes, IDs, inputs, etc.

- The newest version of HTML is HTML5. HTML5 incorporates some new technologies, so that is probably the best one to learn.
- XHTML is also an option, but it is a little bit stricter in its rules.

Learn CSS. CSS stands for «Cascading Style Sheet». Without CSS, there would not be the pretty designs you see on some websites. HTML is merely the skeleton, but CSS is what makes a website colorful and pretty.

- CSS is a bit stricter in its rules. For instance, if you forgot to put a semicolon at the end of each line, you could be in for some headaches trying to figure out what went wrong.

Learn JavaScript and jQuery. This is optional for ones that plan to make simple website, but it is essential if you want interactive websites.

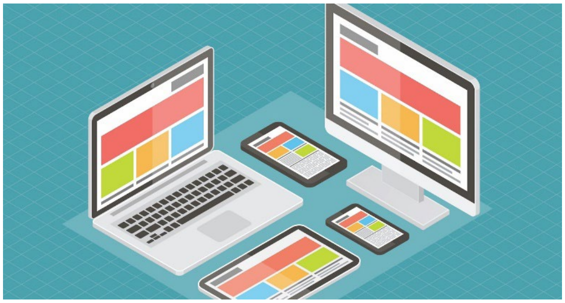
- If you are going to use jQuery, you really only need a basic knowledge of JavaScript, because jQuery makes things pretty simple.
- With jQuery, you can insert widgets like an accordion, a calendar, etc. There are different styles of it, too, so you can pick the style you want for your website.

Move on to the more advanced languages. There's PHP, MySQL, Perl, Ruby, and some others. Again, this is optional for simple website builders, but very useful for interactive and large .websites

- Some of these languages, like PHP, require a test server, so make sure to have an account with a web hosting company or install **server** software on your computer.

Keyword

Server is a computer program or a device that provides functionality for other programs or devices, called "clients".



Applying Your Knowledge

- *Make project websites.* Just experiment and make websites from scratch. This helps you put your knowledge to work, instead of letting it rust out in your brain.
- *Ask friends or family if they want a website***.** You could do it for free to get some experience
- *Start your own website***.** It can be about whatever you want, but it gives you an opportunity to show your work to the general public.
- *Do freelance web design, if you want to.* Charge reasonable rates and post a portfolio on your site. Your business may eventually grow into a full-fledged company.

1.1.3 Web site

Web site, Collection of files and related resources accessible through the World Wide Web and organized under a particular domain name. Typical files found at a Web site are HTML documents with their associated graphic image files (GIF, JPEG, etc.), scripted programs (in Perl, CGI, Java, etc.), and similar resources. The site's files are usually accessed through hypertext or hyperlinks embedded in other files. A Web site may consist of a single HTML file, or it may comprise hundreds or thousands of related files.

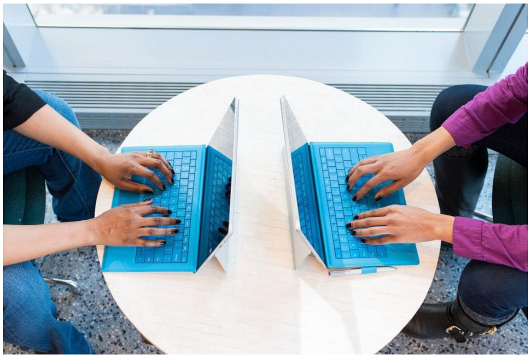
A Web site's usual starting point or opening page, called a home page, usually functions as a table of contents or index, with links to other sections of the site. Web sites are hosted on one or more Web servers, which transfer files to client computers or other servers that request them using the HTTP protocol. Although the term "site" implies a single physical location, the files and resources of a Web site may actually be spread among several servers in different geographic locations. The particular file desired by a client is specified by a URL that is either typed into a browser or accessed by selecting a hyperlink.

Remember

You would need someone to install and maintain the hardware and software for your server. It can get very costly to deal with bugs, viruses, network and other issues that could affect your server's performance.

Why Create A Website?

A website is a unique way to connect with the world. Whether you choose to create a website to share your passion for music, inform people of your business, sell products, or any other reason—there are no boundaries to what you can do! With a website, tons of possibilities are available to you.



Some of you might be thinking of creating a personal website. Outstanding! You cannot think of a better way of organizing, and sharing, family memorabilia than on a website. Put in everything you know about your family and preserve that knowledge for future generations. Publish it on the web to share it, and put a back-up copy of it onto a CD for safe keeping. Your family tree, your photos, your stories, everything you add to your site can be saved for the future. It will not fade or rot like paper and photo prints. It can be replicated in its entirety in a few minutes and it can be sent around the world in a matter of moments. Start now and in a generation or two your family will know their history like very few of us do today.

Many of you, however, would like to either obtain or at least supplement their income by creating web pages for others. There is not often too much money to be made in producing web sites for individuals to show off their family portraits, which means the targets will probably be businesses. Most large businesses either develop their websites in-house or outsource the job to one of the many businesses that have been providing web site development services for some time. This leaves the smaller businesses. For every small business that has its own site, there must be hundreds within a short distance that do not have one, but could definitely use one. With the idea of providing some food for sales thought, we take a look here at some of the reasons small business use websites. The first reason might seem to be the most obvious; on looking closer we might revise our thought. Businesses want to sell something. This results in first thought being that the business website should sell something. This would suggest E-commerce. Such a site might well be the answer for the business but is it really where they should start? The process of selling something has several stages to it. You have to have a product. You let people know about your product, persuading them of its virtues. Once they are considering purchasing, you have to close the deal. Finally you fulfill. E-commerce can be involved all of these steps, except the first.

In order to build an effective e-commerce site, you will need to be able to obtain a security certificate to set up and use a secure site. You will need scripts or software to provide shopping cart capability. You will need to provide methods of receiving payment by either incorporating credit card processing into your pages or by providing hooks into processing services like Paypal. Finally you will need to provide the product information, which might well be contained in a database of items. All of these things are very doable, and there are tutorials in our collection to provide you with assistance, should you need it. The question is, if you are going to sell a business on getting a new website, are you going to be able to persuade them of the virtues of all these things, as well as of your ability to effectively provide them? We hope so; but in case there is any doubt, you might consider starting with a simpler approach.

1.1.4 World Wide Web

World Wide Web (WWW), byname The Web, the leading information retrieval service of the Internet (q.v.; the worldwide computer network). The Web gives users access to a vast array of documents that are connected to each other by means of hypertext or hypermedia links—i.e., hyperlinks, electronic connections that link related pieces of information in order to allow a user easy access to them. Hypertext allows the user to select a word from text and thereby access other documents that contain additional information pertaining to that word; hypermedia documents feature links to images, sounds, animations, and movies. The Web operates within the Internet's basic client-server format; servers are computer programs that store and transmit documents to other computers on the network when asked to, while clients are programs that request documents from a server as the user asks for them. Browser software allows users to view the retrieved documents.

A hypertext document with its corresponding text and hyperlinks is written in Hypertext Markup Language (HTML) and is assigned an online address called a Uniform Resource .(Locator (URL

The development of the World Wide Web was begun in 1989 by Tim Berners-Lee and his colleagues at CERN, an international scientific organization based in Geneva, Switz. They created a protocol, Hypertext Transfer Protocol (HTTP), which standardized communication between servers and clients. Their text-based Web browser was made available for general release in January 1992.

Did You Know?

In 1989, Berners-Lee began work on the first World Wide Web server at CERN. He called the server "httpd" and dubbed the first client "WWW." Originally, WWW was just a WYSIWYG hypertext browser/editor that ran in the NeXTStep environment.

The World Wide Web gained rapid acceptance with the creation of a Web browser called Mosaic, which was developed in the United States by Marc Andreessen and others at the National Center for Supercomputing Applications at the University of Illinois and was released in September 1993.

Mosaic allowed people using the Web to use the same sort of "point-and-click" graphical manipulations that had been available in personal computers for some years. In April 1994 Andreessen cofounded Netscape Communications Corporation, whose Netscape Navigator became the dominant Web browser soon after its release in December 1994. By the mid-1990s the World Wide Web had millions of active users.

1.1.5 Web Standards

Web Standards is defined as a formal set of standards and technical specifications used to define aspects of the World Wide Web. These are best-practice standards used by organizations to build web sites and web applications.

Remember

You must secure space on a server where the files of your website will be accessible on the Internet. This server is referred to as a web host.

When Tim Berners-Lee first created ENQUIRE (the precursor to the web), he had a vision of an information super highway freely traveled by all. While still in its infancy, the World Wide Web promised an unfettered access to information, community and, yes, business opportunities. However, as the web was slowly taking shape, that freedom appeared to be threatened from within, and there was a good chance that the internet we enjoy today might collapse under its own potential.

W3C and the Web Standards Project

Not long after the web was introduced to the world, Netscape and Microsoft (two of the prime movers in software development) began a steadily escalating browser war. In an effort to outdo the other, each firm was introducing new elements to their underlying software and designing new ways to fetch and process online content. The result was two competing browsers that were growing less and less compatible. This had the potential to turn the dream of an unfettered internet into a convoluted nightmare, effectively denying users access to the full range of online content while driving up the costs of website design and development.

In response to the ongoing browser wars, and the potential fracturing of the internet project, Berners-Lee formed the World Wide Web Consortium (W3C). Its job was to craft a set of universally accepted protocols and guidelines for the internet. These would become the rulebook which would guide the work of programmers and software developers in the future, hopefully ensuring the continued growth and accessibility of the internet. In 1998, an independent group of web designers and developers formed the Web Standards Project (WaSP). Its mission was to refine and promote the W3C's list of best practices and end the browser wars once and for all. WaSP was successful, and the vision of a freely traveled information highway was largely realized. Today, WaSP continues to work toward keeping the internet free, open, and accessible to all.

The Importance of Web Standards

It would be difficult to overestimate the importance of the Web Standards Project and the guidelines that they promote. For the internet to realize its full potential, it must be guided by some common principles. These "web standards" provide a set of fundamental, and it is hoped universal, guidelines by which content for the internet is created and disseminated. The need for these web standards is clear, and they address some key aspects of programming and web design.

- Efficient Code — web standards brings efficient coding to web design. If all programming was limited to HTML, the result would be bloated and ungainly files that require longer than necessary download times. This not only affects the user's online experience, but it can actually increase the cost of accessing online content for those users who pay for downloads by the megabyte.

- Coding Compatibility — whether it is a web page, an add-on, or a software application, almost every piece of programming passes through the hands of many different designers. Having an accepted standard of coding ensures that all programmers are speaking the same "language," and that code remains streamlined and efficient regardless.
- Simplified Maintenance — this further supports the value of standardized coding practices. Lean and efficient code is easier, and less expensive, to update and maintain. With style and layout information specified in one place (i.e. CSS elements) updates do not need to be applied to every page of an active site.
- Device Compatibility — the adherence to web standards ensures that online content can be quickly and efficiently reformatted for different browsers and devices. This is particularly important now that mobile devices have become so popular with online users.
- Search Engine Compatibility — all web developers want their online content to be easily searchable by Google and other search engines. In order for any search engine to rank and return relevant content in a user search it must scan that content with a web crawler. If that web crawler cannot interpret the content, rankings will naturally suffer. By following the approved standards set down by the W3C, developers can more easily ensure that the content they produce can be quickly and accurately scanned by the search engine bots.
- Accessibility — finally, web standards also addresses the issue of accessibility for the visually impaired. Users with visual impairments often use screen readers to access content on the web. These readers literally read the **web page** aloud. Adherence to the accepted web standards governing semantic structure ensures that can easily decipher online content, keeping the web open and accessible to the visually impaired. This is one of the few aspects of the web standards project that has been codified into law.

Keyword

Web page is a document that is suitable for the World Wide Web and web browsers. A web browser displays a web page on a monitor or mobile device.

The W3C Standards

While WaSP addresses a diverse array of issues, the key guidelines originally set forth by the W3C have become the universally accepted core of the web standards rulebook. These form the foundation on which the continued growth and accessibility of the internet depend, and they can be broken down into a few basic topics.

- **Markup Languages** are systems for adding information to a document that is separate from the text itself. The web standards guidelines govern the use of HTML and its subsequent variations (including HTML5 and XHTML). As a markup language, the HTML variants are used to modify different types of content in an online document. The elements that make up these languages specify how each bit of content should be rendered in a web browser (i.e. headings, paragraphs, bullet lists, etc.). While the basic markup elements define the actual content, the use of additional attributes serve to provide more specific information about that content.
- **HTML** (Hypertext Markup Language) is the oldest web coding language. As such, it is the most common markup language used by web developers, and still forms a significant part of all online content. The use of HTML is governed by a set of syntax rules which, though fairly efficient, are also fairly lax. Elements are not case sensitive, closing tags are not strictly regulated, and shortcuts can be used for certain attributes. While basic HTML was the standard for many years, it has undergone several revisions, including HTML 2.0 and HTML 3.0. Its latest iteration is HTML5.
- **HTML5** is the latest iteration of the basic HTML markup language. It introduces some new rules and features, while discarding some of the elements from before versions. HTML5 introduced new semantic elements and control attributes as well as several Application Programming Interfaces (API) that support drag and drop features, geolocation, and local storage that eliminates the use of cookies.

- **XHTML** stands for eXtensible Hypertext Markup Language. It combines the structural attributes of HTML with the features of an XML application. XHTML is typically used when online content needs to be reprocessed for compatibility with mobile devices or outlying browser designs. The addition of XML elements makes it quicker and easier for content to be reprocessed than standard HTML. XHTML also brings stricter semantic rules to the language. Elements and attributes in XHTML are case sensitive, all elements must be closed, and attribute values must be enclosed by quotes.
- **CSS** (Cascading Style Sheets) provide the mechanism by which changes in the appearance or position of online content can be assigned to a document's HTML or XHTML elements. The use of CSS defines the overall appearance of a website, and allows it to be remodeled quickly and efficiently. In short, the use of CSS allows web developers to more artfully control the formatting and layout of a document. This allows web developers to easily change backgrounds, alter fonts and text sizes, and reposition content on a web page. While much of this is possible with basic HTML, the use of CSS allows programmers and web developers to produce content that is underpinned by a more streamlined and efficient code.
- **JavaScript** is an object-oriented scripting language. It is primarily used to add behavior to a web page. It can be used to validate data on a form, provide drag-and-drop functions, enable button functions, and animate page elements. JavaScript is instrumental in making web pages more interactive, allowing them to more accurately respond to user events such as mouse clicks, page navigation, and inputted searches.

These are the basic fundamentals of programming and web design as laid out by the W3C and the Web Standards Project. More than simply a set of 'best practices', adherence to their use ensures the continued development and expansion of the World Wide Web.

1.2 BRIEF HISTORY OF INTERNET

Internet, a system architecture that has revolutionized communications and methods of commerce by allowing various computer networks around the world to interconnect. Sometimes referred to as a "network of networks," the Internet emerged in the United States in the 1970s but did not become visible to the general public until the early 1990s. By 2015, approximately 3.2 billion people, or nearly half of the world's population, were estimated to have access to the Internet.

The Internet provides a capability so powerful and general that it can be used for almost any purpose that depends on information, and it is accessible by every individual who connects to one of its constituent networks.

It supports human communication via electronic mail (e-mail), “chat rooms,” newsgroups, and audio and video transmission and allows people to work collaboratively at many different locations. It supports access to digital information by many applications, including the World Wide Web. The Internet has proved to be a spawning ground for a large and growing number of “e-businesses” (including subsidiaries of traditional “brick-and-mortar” companies) that carry out most of their sales and services over the Internet. Many experts believe that the Internet will dramatically transform business as well as society.

1.2.1 Origin and Development

The history of the Internet begins with the development of electronic computers in the 1950s. Initial concepts of wide area networking originated in several computer science laboratories in the United States, United Kingdom, and France.

Early Networks

The first computer networks were dedicated special-purpose systems such as SABRE (an airline reservation system) and AUTODIN I (a defense command-and-control system), both designed and implemented in the late 1950s and early 1960s. By the early 1960s computer manufacturers had begun to use semiconductor technology in commercial products, and both conventional batch-processing and time-sharing systems were in place in many large, technologically advanced companies. Time-sharing systems allowed a computer’s resources to be shared in rapid succession with multiple users, cycling through the queue of users so quickly that the computer appeared dedicated to each user’s tasks despite the existence of many others accessing the system “simultaneously.” This led to the notion of sharing computer resources (called host computers or simply hosts) over an entire network. Host-to-host interactions were envisioned, along with access to specialized resources (such as supercomputers and mass storage systems) and interactive access by remote users to the computational powers of time-sharing systems located elsewhere.

These ideas were first realized in ARPANET, which established the first host-to-host network connection on Oct. 29, 1969. It was created by the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense. ARPANET was one of the first general-purpose computer networks. It connected time-sharing computers at government-supported research sites, principally universities in the United States, and it soon became a critical piece of infrastructure for the computer science research community in the United States. Tools and applications—such as the simple mail transfer protocol (SMTP, commonly referred to as e-mail), for sending short messages, and the file transfer protocol (FTP), for longer transmissions—quickly emerged. In order to achieve cost-effective interactive communications between computers, which typically communicate in short bursts of data, ARPANET employed the new technology of packet switching. Packet switching takes large messages (or chunks of computer data) and breaks them into smaller, manageable pieces (known as packets) that can travel independently over any available circuit to the target destination, where the pieces are reassembled. Thus, unlike traditional voice communications, packet switching does not require a single dedicated circuit between each pair of users.

Commercial packet networks were introduced in the 1970s, but these were designed principally to provide efficient access to remote computers by dedicated terminals. Briefly, they replaced long-distance **modem** connections by less-expensive “virtual” circuits over packet networks. In the United States, Telenet and Tymnet were two such packet networks. Neither supported host-to-host communications; in the 1970s this was still the province of the research networks, and it would .remain so for many years

Keyword

Modem is a network hardware device that modulates one or more carrier wave signals to encode digital information for transmission and demodulates signals to decode the transmitted information.

DARPA (Defense Advanced Research Projects Agency; formerly ARPA) supported initiatives for ground-based and satellite-based packet networks. The ground-based packet radio system provided mobile access to computing resources, while the packet satellite network connected the United States with several European countries and enabled connections with widely dispersed and remote regions.

With the introduction of packet radio, connecting a mobile terminal to a computer network became feasible. However, time-sharing systems were then still too large, unwieldy, and costly to be mobile or even to exist outside a climate-controlled computing environment.

A strong motivation thus existed to connect the packet radio network to ARPANET in order to allow mobile users with simple terminals to access the time-sharing systems for which they had authorization. Similarly, the packet satellite network was used by DARPA to link the United States with satellite terminals serving the United Kingdom, Norway, Germany, and Italy.

These terminals, however, had to be connected to other networks in European countries in order to reach the end users. Thus arose the need to connect the packet satellite net, as well as the packet radio net, with other networks.

Foundation of the Internet

The Internet resulted from the effort to connect various research networks in the United States and Europe. First, DARPA established a program to investigate the interconnection of “heterogeneous networks.”

This program, called Internetting, was based on the newly introduced concept of open architecture networking, in which networks with defined standard interfaces would be interconnected by “gateways.” A working demonstration of the concept was planned. In order for the concept to work, a new protocol had to be designed and developed; indeed, a system architecture was also required.

In 1974 Vinton Cerf, then at Stanford University in California, then at DARPA, collaborated on a paper that first described such a protocol and system architecture—namely, the transmission control protocol (TCP), which enabled different types of machines on networks all over the world to route and assemble data packets. TCP, which originally included the Internet protocol (IP), a global addressing mechanism that allowed routers to get data packets to their ultimate destination, formed the TCP/IP standard, which was adopted by the U.S. Department of Defense in 1980. By the early 1980s the “open architecture” of the TCP/IP approach was adopted and endorsed by many other researchers and eventually by technologists and businessmen around the world.

By the 1980s other U.S. governmental bodies were heavily involved with networking, including the National Science Foundation (NSF), the Department of Energy, and the National Aeronautics and Space Administration (NASA). While DARPA had played a seminal role in creating a small-scale version of the Internet among its researchers, NSF worked with DARPA to expand access to the entire scientific and academic community and to make TCP/IP the standard in all federally supported research networks. In 1985–86 NSF funded the first five supercomputing centers—at Princeton University, the University of Pittsburgh, the University of California, San Diego, the University of Illinois, and Cornell University. In the 1980s NSF also funded the development and operation of the NSFNET, a national "backbone" network to connect these centers. By the late 1980s the network was operating at millions of bits per second. NSF also funded various nonprofit local and regional networks to connect other users to the NSFNET. A few commercial networks also began in the late 1980s; these were soon joined by others, and the Commercial Internet Exchange (CIX) was formed to allow transit traffic between commercial networks that otherwise would not have been allowed on the NSFNET backbone. In 1995, after extensive review of the situation, NSF decided that support of the NSFNET infrastructure was no longer required, since many commercial providers were now willing and able to meet the needs of the research community, and its support was withdrawn. Meanwhile, NSF had fostered a competitive collection of commercial Internet backbones connected to one another through so-called network access points (NAPs).

From the Internet's origin in the early 1970s, control of it steadily devolved from government stewardship to private-sector participation and finally to private custody with government oversight and forbearance. Today a loosely structured group of several thousand interested individuals known as the Internet Engineering Task Force participates in a grassroots development process for Internet standards. Internet standards are maintained by the nonprofit Internet Society, an international body with headquarters in Reston, Virginia. The Internet Corporation for Assigned Names and Numbers (ICANN), another nonprofit, private organization, oversees various aspects of policy regarding Internet domain names and numbers.

Commercial Expansion

The rise of commercial Internet services and applications helped to fuel a rapid commercialization of the Internet. This phenomenon was the result of several other factors as well. One important factor was the introduction of the personal computer and the workstation in the early 1980s—a development that in turn was fueled by unprecedented progress in integrated circuit technology and an attendant rapid decline in computer prices. Another factor, which took on increasing importance, was the emergence of Ethernet and other "local area networks" to link personal computers. But other forces were at work too. Following the restructuring of AT&T in 1984, NSF took advantage of various new options for national-level digital backbone services for the NSFNET. In 1988 the Corporation for National Research Initiatives received approval to conduct an experiment linking a commercial e-mail service (MCI Mail) to the Internet. This application was the first Internet connection to a commercial provider that was not also part of the research community. Approval quickly followed to allow other e-mail providers access, and the Internet began its first explosion in traffic.

In 1993 federal legislation allowed NSF to open the NSFNET backbone to commercial users. Prior to that time, use of the backbone was subject to an "acceptable use" policy, established and administered by NSF, under which commercial use was limited to those applications that served the research community. NSF recognized that commercially supplied network services, now that they were available, would ultimately be far less expensive than continued funding of special-purpose network services.

Also in 1993 the University of Illinois made widely available Mosaic, a new type of computer program, known as a browser, that ran on most types of computers and, through its "point-and-click" interface, simplified access, retrieval, and display of files through the Internet. Mosaic incorporated a set of access protocols and display standards originally developed at the European Organization for Nuclear Research (CERN) by Tim Berners-Lee for a new Internet application called the World Wide Web (WWW). In 1994 Netscape Communications Corporation (originally called Mosaic Communications Corporation) was formed to further develop the Mosaic browser and server software for commercial use. Shortly thereafter, the software giant Microsoft Corporation became interested in supporting Internet applications on personal computers (PCs) and developed its Internet Explorer Web browser (based initially on Mosaic) and other programs. These new commercial capabilities accelerated the growth of the Internet, which as early as 1988 had already been growing at the rate of 100 % per year.

By the late 1990s there were approximately 10,000 Internet service providers (ISPs) around the world, more than half located in the United States. However, most of these ISPs provided only local service and relied on access to regional and national ISPs for wider connectivity. Consolidation began at the end of the decade, with many small to medium-size providers merging or being acquired by larger ISPs. Among these larger providers were groups such as America Online, Inc. (AOL), which started as a dial-up information service with no Internet connectivity but made a transition in the late 1990s to become the leading provider of Internet services in the world—with more than 25 million subscribers by 2000 and with branches in Australia, Europe, South America, and Asia. Widely used Internet "portals" such as AOL, Yahoo!, Excite, and others were able to command advertising fees owing to the number of "eyeballs" that visited their sites. Indeed, during the late 1990s advertising revenue became the main quest of many Internet sites, some of which began to speculate by offering free or low-cost services of various kinds that were visually augmented with advertisements. By 2001 this speculative bubble had burst.

Future Directions

While the precise structure of the future Internet is not yet clear, many directions of growth seem apparent. One is the increased availability of wireless access. Wireless services enable applications not previously possible in any economical fashion. For example, global positioning systems (GPS) combined with wireless Internet access would help mobile users to locate alternate routes, generate precise accident reports and initiate recovery services, and improve traffic management and congestion control. In addition to wireless laptop computers and personal digital assistants (PDAs), wearable devices with voice input and special display glasses are under development.

Another future direction is toward higher backbone and network access speeds. Backbone data rates of 10 billion bits (10 gigabits) per second are readily available today, but data rates of 1 trillion bits (1 terabit) per second or higher will eventually become commercially feasible. If the development of computer hardware, software, applications, and local access keeps pace, it may be possible for users to access networks at speeds of 100 gigabits per second. At such data rates, high-resolution video—indeed, multiple video streams—would occupy only a small fraction of available bandwidth. Remaining bandwidth could be used to transmit auxiliary information about the data being sent, which in turn would enable rapid customization of displays and prompt resolution of certain local queries. Much research, both public and private, has gone into integrated broadband systems that can simultaneously carry multiple signals—data, voice, and video. In particular, the U.S. government has funded research to create new high-speed network capabilities dedicated to the scientific-research community. It is clear that communications connectivity will be an important function of a future Internet as more machines and devices are interconnected. In 1998, after four years of study, the Internet Engineering Task Force published a new 128-bit IP address standard intended to replace the conventional 32-bit standard. By allowing a vast increase in the number of available addresses (2^{128}, as opposed to 2^{32}), this standard will make it possible to assign unique addresses to almost every electronic device imaginable. Thus, the expressions "wired" office, home, and car may all take on new meanings, even if the access is really wireless.

The dissemination of digitized text, pictures, and audio and video recordings over the Internet, primarily available today through the World Wide Web, has resulted in an information explosion. Clearly, powerful tools are needed to manage network-based information. Information available on the Internet today may not be available tomorrow without careful attention's being paid to preservation and archiving techniques. The key to making information persistently available is infrastructure and the management of that infrastructure. Repositories of information, stored as digital objects, will soon populate the Internet. At first these repositories may be dominated by digital objects specifically created and formatted for the World Wide Web, but in time they will contain objects of all kinds in formats that will be dynamically resolvable by users' computers in real time. Movement of digital objects from one repository to another will still leave them available to users who are authorized to access them, while replicated instances of objects in multiple repositories will provide alternatives to users who are better able to interact with certain parts of the Internet than with others. Information will have its own identity and, indeed, become a "first-class citizen" on the Internet.

1.2.2 Society and the Internet

What began as a largely technical and limited universe of designers and users became one of the most important mediums of the late 20th and early 21st centuries. As the Pew Charitable Trust observed in 2004, it took 46 years to wire 30 % of the United

States for electricity; it took only 7 years for the Internet to reach that same level of connection to American homes. By 2005, 68 % of American adults and 90 % of American teenagers had used the Internet. Europe and Asia were at least as well connected as the United States. Nearly half of the citizens of the European Union are online, and even higher rates are found in the Scandinavian countries. There is a wide variance in Asian countries; for example, by 2005 Taiwan, Hong Kong, and Japan had at least half of their populations online, whereas India, Pakistan, and Vietnam had less than 10 %. South Korea was the world leader in connecting its population to the Internet through high-speed **broadband** connections.

Keyword

Broadband is wide bandwidth data transmission which transports multiple signals and traffic types. The medium can be coaxial cable, optical fiber, radio or twisted pair.

Such statistics can chart the Internet's growth, but they offer few insights into the changes wrought as users—individuals, groups, corporations, and governments—have embedded the technology into everyday life. The Internet is now as much a lived experience as a tool for performing particular tasks, offering the possibility of creating an environment or virtual reality in which individuals might work, socially interact with others, and perhaps even live out their lives.

History, community, and communications

The Internet was the result of some visionary thinking by people in the early 1960s who saw great potential value in allowing computers to share information on research and .development in scientific and military fields

Two agendas

The Internet has evolved from the integration of two very different technological agendas—the Cold War networking of the U.S. military and the personal computer (PC) revolution. The first agenda can be dated to 1973, when the Defense Advanced Research Projects Agency (DARPA) sought to create a communications network that would support the transfer of large data files between government and government-sponsored academic-research laboratories. The result was the ARPANET, a robust decentralized network that supported a vast array of computer hardware. Initially, ARPANET was the preserve of academics and corporate researchers with access to time-sharing mainframe computer systems. Computers were large and expensive; most computer professionals could not imagine anyone needing, let alone owning, his own "personal" computer. And yet Joseph Licklider, one of the driving forces at DARPA for computer networking, stated that online communication would "change the nature and value of communication even more profoundly than did the printing press and the picture tube." The second agenda began to emerge in 1977 with the introduction of the Apple II, the first affordable computer for individuals and small businesses. Created by Apple Computer, Inc. (now Apple Inc.), the Apple II was popular in schools by 1979, but in the corporate market it was stigmatized as a game machine. The task of cracking the business market fell to IBM. In 1981 the IBM PC was released and immediately standardized the PC's basic hardware and operating system—so much so that first *PC-compatible* and then simply *PC* came to mean any personal computer built along the lines of the IBM PC. A major center of the PC revolution was the San Francisco Bay area, where several major research institutions funded by DARPA—Stanford University, the University of California, Berkeley, and Xerox PARC—provided much of the technical foundation for Silicon Valley. It was no small coincidence that Apple's two young founders—Steven Jobs and Stephen Wozniak—worked as interns in the Stanford University Artificial Intelligence Laboratory and at the nearby Hewlett-Packard Company. The Bay Area's counterculture also figured prominently in the PC's history. Electronic hobbyists saw themselves in open revolt against the "priesthood" of the mainframe computer and worked together in computer-enthusiast groups to spread computing to the masses.

The WELL

Why does this matter? The military played an essential role in shaping the Internet's architecture, but it was through the counterculture that many of the practices of contemporary online life emerged. A telling example is the early electronic bulletin board system (BBS), such as the WELL (Whole Earth 'Lectronic Link). Established in 1985 by American publisher Stewart Brand, who viewed the BBS as an extension of his *Whole Earth Catalog*, the WELL was one of the first electronic communities organized around forums dedicated to particular subjects such as parenting and Grateful Dead concerts. The latter were an especially popular topic of online conversation, but it was in the parenting forum where a profound sense of community and belonging initially appeared. For example, when one participant's child was diagnosed with leukemia, members of the forum went out of their way either to find health resources or to comfort the distressed parents. In this one instance, several features still prevalent in the online world can be seen. First, geography was irrelevant. WELL members in California and New York could bring their knowledge together within the confines of a forum—and could do so collectively, often exceeding the experience available to any local physician or medical center. This marshaling of shared resources persists to this day as many individuals use the Internet to learn more about their ailments, find others who suffer from the same disease, and learn about drugs, physicians, and alternative therapies.

Another feature that distinguished the WELL forums was the use of moderators who could interrupt and focus discussion while also disciplining users who broke the rather loose rules. “Flame wars” (crass, offensive, or insulting exchanges) were possible, but anyone dissatisfied in one forum was free to organize another. In addition, the WELL was intensely democratic. WELL forums were the original chat rooms—online spaces where individuals possessing similar interests might congregate, converse, and even share their physical locations to facilitate meeting in person. Finally, the WELL served as a template for other online communities dedicated to subjects as diverse as Roman Catholicism, liberal politics, gardening, and automobile modification.

Instant broadcast communication

For the individual, the Internet opened up new communication possibilities. E-mail led to a substantial decline in traditional “snail mail.” Instant messaging (IM), or text messaging, expanded, especially among youth, with the convergence of the Internet and cellular telephone access to the Web. Indeed, IM became a particular problem in classrooms, with students often surreptitiously exchanging notes via wireless communication devices. More than 50 million American adults, including 11 million at work, use IM.

From mailing lists to “buddy lists,” e-mail and IM have been used to create “smart mobs” that converge in the physical world. Examples include protest organizing, spontaneous performance art, and shopping. Obviously, people congregated before the Internet existed, but the change wrought by mass e-mailings was in the speed of assembling such events.

For example, In February 1999 activists began planning protests against the November 1999 World Trade Organization (WTO) meetings in Seattle, Washington. Using the Internet, organizers mobilized more than 50,000 individuals from around the world to engage in demonstrations—at times violent—that effectively altered the WTO’s agenda.

More than a decade later, in June 2010 Egyptian computer engineer Wael Ghonim anonymously created a page titled “We Are All Khaled Said” on the social media site Facebook to publicize the death of a 28-year-old Egyptian man beaten to death by police. The page garnered hundreds of thousands of members, becoming an online forum for the discussion of police brutality in Egypt. After a popular uprising in Tunisia in January 2011, Ghonim and several other Internet democracy activists posted messages to their sites calling for similar action in Egypt. Their social media campaign helped spur mass demonstrations that forced Egyptian Pres. osnī Mubārak from power. (The convergence of mobs is not without some techno-silliness. "Flash mobs"—groups of strangers who are mobilized on short notice via Web sites, online discussion groups, or e-mail distribution lists—often take part in bizarre though usually harmless activities in public places, such as engaging in mass free-for-alls around the world on Pillow Fight Day.)

In the wake of catastrophic disasters, citizens have used the Internet to donate to charities in an unprecedented fashion. Others have used the Internet to reunite family members or to match lost pets with their owners. The role of the Internet in responding to disasters, both natural and deliberate, remains the topic of much discussion, as it is unclear whether the Internet actually can function in a disaster area when much of the infrastructure is destroyed. Certainly during the September 11, 2001, attacks, people found it easier to communicate with loved ones in New York City via e-mail than through the overwhelmed telephone network.

Keyword

Telephone network is a telecomm unications network used for telephone calls between two or more parties.

Following the earthquake that struck Haiti in January 2010, electronic media emerged as a useful mode for connecting those separated by the quake and for coordinating relief efforts. Survivors who were able to access the Internet—and friends and relatives abroad—took to social networking sites such as Facebook in search of information on those missing in the wake of the catastrophe. Feeds from those sites also assisted aid organizations in constructing maps of the areas affected and in determining where to channel resources. The many Haitians lacking Internet access were able to contribute updates via text messaging on mobile phones.

Social gaming and social networking

One-to-one or even one-to-many communication is only the most elementary form of Internet social life. The very nature of the Internet makes spatial distances largely irrelevant for social interactions. Online gaming moved from simply playing a game with friends to a rather complex form of social life in which the game's virtual reality spills over into the physical world. The case of *World of Warcraft*, a popular electronic game with several million players, is one example. Property acquired in the game can be sold online, although such secondary economies are discouraged by Blizzard Entertainment, the publisher of World of Warcraft, as a violation of the game's terms of service. In any case, what does it mean that one can own virtual property and that someone is willing to pay for this property with real money? Economists have begun studying such virtual economies, some of which now exceed the gross national product of countries in Africa and Asia. In fact, virtual economies have given economists a means of running controlled experiments.

Millions of people have created online game characters for entertainment purposes. Gaming creates an online community, but it also allows for a blurring of the boundaries between the real world and the virtual one. In Shanghai one gamer stabbed and killed another one in the real world over a virtual sword used in *Legend of Mir 3*. Although attempts were made to involve the authorities in the original dispute, the police found themselves at a loss prior to the murder because the law did not acknowledge the existence of virtual property. In South Korea violence surrounding online gaming happens often enough that police refer to such murders as "off-line PK," a reference to player killing (PK), or player-versus-player lethal contests, which are allowed or encouraged in some games. By 2001 crime related to *Lineage* had forced South Korean police to create special cybercrime units to patrol both within the game and off-line. Potential problems from such games are not limited to crime. Virtual life can be addictive. Reports of players neglecting family, school, work, and even their health to the point of death have become more common.

Social networking sites (SNSs) emerged as a significant online phenomenon since the bursting of the "Internet bubble" in the early 2000s. SNSs use software to facilitate online communities where members with shared interests swap files, photographs, videos, and music, send messages and chat, set up blogs (Web diaries) and discussion groups, and share opinions. Early social networking services included Classmates.

com, which connected former schoolmates, and Yahoo! 360° and Six Degrees, which built networks of connections via friends of friends. In the post bubble era the leading social networking services were Myspace, Facebook, Friendster, Orkut, and LinkedIn. LinkedIn became an effective tool for business staff recruiting. Businesses have begun to exploit these networks, drawing on social networking research and theory, which suggests that finding key "influential" members of existing networks of individuals can give those businesses access to and credibility with the whole network.

Advertising and e-commerce

Nichification allows for consumers to find what they want, but it also provides opportunities for advertisers to find consumers. For example, most search engines generate revenue by matching ads to an individual's particular search query. Among the greatest challenges facing the Internet's continued development is the task of reconciling advertising and commercial needs with the right of Internet users not to be bombarded by "pop-up" Web pages and spam (unsolicited e-mail).

Nichification also opens up important e-commerce opportunities. A bookstore can carry only so much inventory on its shelves, which thereby limits its collection to books with broad appeal. An online bookstore can "display" nearly everything ever published. Although traditional bookstores often have a special-order department, consumers have taken to searching and ordering from online stores from the convenience of their homes and offices.

Although books can be made into purely digital artifacts, "e-books" have not sold nearly as well as digital music. In part, this disparity is due to the need for an e-book reader to have a large, bright screen, which adds to the display's cost and weight and leads to more-frequent battery replacement. Also, it is difficult to match the handy design and low cost of an old-fashioned paperback book. Interestingly, it turns out that listeners download from online music vendors as many obscure songs as big record company hits. Just a few people interested in some obscure song are enough to make it worthwhile for a vendor to store it electronically for sale over the Internet. What makes the Internet special here is not only its ability to match buyers and sellers quickly and relatively inexpensively but also that the Internet and the digital economy in general allow for a flowering of multiple tastes—in games, people, and music.

Information and Copyright

Education

Commerce and industry are certainly arenas in which the Internet has had a profound effect, but what of the foundational institutions of any society—namely, those related to education and the production of knowledge? Here the Internet has had a variety of effects, some of which are quite disturbing. There are more computers in the classroom than ever before, but there is scant evidence that they enhance the learning of basic skills in reading, writing, and arithmetic. And while access to vast amounts of digital information is convenient, it has also become apparent that most students now see libraries as antiquated institutions better used for their computer terminals than for their book collections. As teachers at all education levels can attest, students typically prefer to research their papers by reading online rather than wandering through a library's stacks.

In a related effect the Internet has brought plagiarism into the computer era in two distinct senses. First, electronic texts have made it simple for students to "cut and paste" published sources into their own papers. Second, although students could always get someone to write their papers for them, it is now much easier to find and purchase anonymous papers at Web sites and to even commission original term papers for a fixed fee. Ironically, what the Internet gives, it also takes away. Teachers now have access to **databases** of electronically submitted papers and can easily compare their own students' papers against a vast archive of sources. Even a simple online search can sometimes find where one particularly well-turned phrase originally appeared.

Keyword

Database is an organized collection of data, stored and accessed electronically.

File sharing

College students have been at the leading edge of the growing awareness of the centrality of intellectual property in a digital age. When American college student Shawn Fanning invented Napster in 1999, he set in motion an ongoing legal battle over digital rights. Napster was a file-sharing system that allowed users to share electronic copies of music online. The problem was obvious: recording companies were losing revenues as one legal copy of a song was shared among many people. Although the record companies succeeded in shutting down Napster, they found themselves having to contend with a new form of file sharing, P2P ("person-to-person"). In P2P there is no central administrator to shut down as there had been with Napster. Initially, the recording industry sued the makers of P2P software and a few of the most prolific users—often students located on university campuses with access to high-speed connections for serving music and, later, movie files—in an attempt to discourage the millions of people who regularly used the software. Still, even while some P2P software makers have been held liable for losses that the copyright owners have incurred, more-devious schemes for circumventing apprehension have been invented.

The inability to prevent file sharing has led the recording and movie industries to devise sophisticated copy protection on their CDs and DVDs. In a particularly controversial incident, Sony Corporation introduced CDs into the market with copy protection that involved a special virus like 2005 in code that hid on a user's computer. This code, however, also was open to being exploited by virus writers to gain control .of users' machines

Electronic publishing

The Internet has become an invaluable and discipline-transforming environment for scientists and scholars. In

2004 Google began digitizing public-domain and out-of-print materials from several cooperating libraries in North America and Europe, such as the University of Michigan library, which made some seven million volumes available. Although some researchers and publishers challenged the project for fear of losing control of copyrighted material, similar digitization projects were launched by Microsoft Corporation and the online book vendor Amazon.com, although the latter company proposed that each electronic page would be retrieved for a small fee shared with the copyright holders.

The majority of academic journals are now online and searchable. This has created a revolution in scholarly publishing, especially in the sciences and engineering. For example, arXiv.org has transformed the rate at which scientists publish and react to new theories and experimental data. Begun in 1991, arXiv.org is an online archive in which physicists, mathematicians, computer scientists, and computational biologists upload research papers long before they will appear in a print journal. The articles are then open to the scrutiny of the entire scientific community, rather than to one or two referees selected by a journal editor. In this way scientists around the world can receive an abstract of a paper as soon as it has been uploaded into the depository. If the abstract piques a reader's interest, the entire paper can be downloaded for study. Cornell University in Ithaca, New York, and the U.S. National Science Foundation support arXiv.org as an international resource.

While arXiv.org deals with articles that might ultimately appear in print, it is also part of a larger shift in the nature of scientific publishing. In the print world a handful of companies control the publication of the most scientific journals, and the price of institutional subscriptions is frequently exorbitant. This has led to a growing movement to create online-only journals that are accessible for free to the entire public—a public that often supports the original research with its taxes. For example, the Public Library of Science publishes online journals of biology and medicine that compete with traditional print journals. There is no difference in how their articles are vetted for publication; the difference is that the material is made available for free. Unlike other creators of content, academics are not paid for what they publish in scholarly journals, nor are those who review the articles. Journal publishers, on the other hand, have long received subsidies from the scientific community, even while charging that community high prices for its own work. Although some commercial journals have reputations that can advance the careers of those who publish in them, the U.S. government has taken the side of the "open source" publishers and demanded that government-financed research be made available to taxpayers as soon as it has been published.

In addition to serving as a medium for the exchange of articles, the Internet can facilitate the discussion of scientific work long before it appears in print. Scientific blogs—online journals kept by individuals or groups of researchers—have flourished as a form of online salon for the discussion of ongoing research. There are pitfalls to such practices, though. Astronomers who in 2005 posted abstracts detailing the discovery of a potential 10th planet found that other researchers had used their abstracts to find the new astronomical body themselves. In order to claim priority of discovery, the original group rushed to hold a news conference rather than waiting to announce their work at an academic conference or in a peer-reviewed journal.

Privacy and the Internet

Concerns about privacy in cyberspace are an issue of international debate. As reading and writing, health care and shopping, and sex and gossip increasingly take place in cyberspace, citizens around the world are concerned that the most intimate details of their daily lives are being monitored, searched, recorded, stored, and often misinterpreted when taken out of context. For many, the greatest threats to privacy come not from state agents but from the architecture of e-commerce itself, which is based, in unprecedented ways, on the recording and exchange of intimate personal information.

The threats to privacy in the new Internet age were crystallized in 2000 by the case of DoubleClick, Inc. For a few years DoubleClick, the Internet's largest advertising company, had been compiling detailed information on the browsing habits of millions of World Wide Web users by placing "cookie" files on computer hard drives. Cookies are electronic footprints that allow Web sites and advertising networks to monitor people's online movements with telescopic precision—including the search terms people enter as well as the articles they skim and how long they spend skimming them. As long as users were confident that their virtual identities were not being linked to their actual identities, many were happy to accept DoubleClick cookies in exchange for the convenience of navigating the Web more efficiently. Then in November 1999 DoubleClick bought Abacus Direct, which held a database of names, addresses, and information about the off-line buying habits of 90 million households compiled from the largest direct-mail catalogs and retailers in the nation. Two months later DoubleClick began compiling profiles linking individuals' actual names and addresses to Abacus's detailed records of their online and off-line purchases. Suddenly, shopping that once seemed anonymous was being archived in personally identifiable dossiers.

Under pressure from privacy advocates and dot-com investors, DoubleClick announced in 2000 that it would postpone its profiling scheme until the U.S. government and the e-commerce industry had agreed on privacy standards. Two years later it settled consolidated class-action lawsuits from several states, agreeing to pay legal expenses of up to $1.8 million, to tell consumers about its **data-collection** activities in its online privacy policy, and to get permission before combining a consumer's personally identifiable data with his or her Web-surfing history. DoubleClick also agreed to pay hundreds of thousands of dollars to settle differences with attorneys general from 10 states who were investigating its information gathering.

The retreat of DoubleClick might have seemed like a victory for privacy, but it was only an early battle in a much larger war—one in which many observers still worry that privacy may be vanquished. “You already have zero privacy—get over it,” Scott McNealy, the CEO of Sun Microsystems, memorably remarked in 1999 in response to a question at a product show at which Sun introduced a new interactive technology called Jini. Sun’s cheerful Web site promised to usher in the “networked home” of the future, in which the company’s “gateway” software would operate “like a congenial party host inside the home to help consumer appliances communicate intelligently with each other and with outside networks.” In this chatty new world of electronic networking, a household’s refrigerator and coffeemaker could talk to a television, and all three could be monitored from the office computer. The incessant information exchanged by these gossiping appliances might, of course, generate detailed records of the most intimate details of their owners’ daily lives.

New evidence seemed to emerge every day to support McNealy’s grim verdict about the triumph of online surveillance technology over privacy. A survey of nearly a thousand large companies conducted by the American Management Association in 2000 found that more than half of the large American firms surveyed monitored the Internet connections of their employees. Two-thirds of the firms monitored e-mail messages, computer files, or telephone conversations, up from only one-third three years earlier. Some companies used Orwellian computer software with names like Spector, Assentor, or Investigator that could monitor and record every keystroke on the computer with video-like precision. These virtual snoops could also be programmed to screen all incoming and outgoing e-mail for forbidden words and phrases—such as those involving racism, body parts, or the name of the boss—and then forward suspicious messages to a supervisor for review.

Issues in new media

Changes in the delivery of books, music, and television extended the technologies of surveillance beyond the office, blurring the boundaries between work and home. The same technologies that make it possible to download digitally stored books, songs, and movies directly onto computer hard drives or mobile devices could make it possible for publishers and entertainment companies to record and monitor each individual’s browsing habits with unsettling specificity. Television too is being redesigned to create precise records of viewing habits. For instance, digital video recorders make it possible to store hours of television programs and enable viewers to skip commercials and to create their own program lineups. The data generated by such actions could create viewer profiles, which could then be used to make viewing suggestions and to record future shows.

Privacy of cell phone communication also has become an issue, as in 2010 when BlackBerry smartphone maker RIM reacted to demands from the United Arab Emirates(U.A.E.), Saudi Arabia, and India that security forces from those countries be given the ability to intercept communications such as e-mail and instant messages from BlackBerry users within their borders. The U.A.E. later canceled a planned ban on the BlackBerry service, saying that it had reached an agreement with RIM, which declined to reveal its discussions with the governments of other countries. The demands were part of a rising tide of security demands from national governments that cited the .need to monitor criminals and terrorists who used wireless communications

The United States is not immune to these controversies. In 2010 Pres. Barack Obama's administration said that in order to prevent terrorism and identify criminals, it wanted Congress to require that all Internet services be capable of complying with wiretap orders. The broad requirement would include Internet phone services, social networking services, and other types of Internet communication, and it would enable even encrypted messages to be decoded and read—something that required considerable time and effort. Critics complained that the monitoring proposal challenged the ideals of privacy and the lack of centralized authority for which the Internet had long been known.

Photos and videos also emerged as unexpected threats to personal privacy. "Geotags" are created when photos or videos are embedded with geographic location data from GPS chips inside cameras, including those in cell phones. When images are uploaded to the Internet, the geotags allow homes or other personal locations within the images to be precisely located by those who view the photos online. The security risk is not widely understood by the public, however, and in some cases disabling the geotag feature in certain models of digital cameras and camera-equipped smartphones is complicated.

Google's Street View photo-mapping service caused privacy concerns when the company disclosed that it had been recording locations and some data from unprotected household wireless networks as it took pictures. The company said that the data had been gathered inadvertently. German officials objected to Google's actions on the basis of Germany's strict privacy laws, and, although German courts decided against the objections, Google did not expand its Street View service in Germany beyond the handful of urban centers that it had already photo-mapped. The controversy led to other investigations of the Street View service by several U.S. states and the governments of several countries (including the Czech Republic, which eventually refused to grant Google permission to offer the Street View service there).

The social network Facebook has been a particular focus of privacy concerns on the Internet. Over the lifetime of the site, the default privacy settings for a Facebook user's information evolved from most content being accessible only to a user's friends or friends of friends to being accessible to everyone. In December 2009 Facebook rolled out a new privacy settings update that allowed users to exercise more "granular" control over what personal information was shared or displayed. However, the labyrinthine nature of the various privacy-control menus discouraged use of the new privacy settings. Users tended to fall back on Facebook's default settings, which, because of the expansion of Facebook's "opt-out" policy, were at the loosest level of security, forcing users to "opt-in" to make information private. Responding to criticism, Facebook revised its privacy policy again in May 2010, with a simplified system that consolidated privacy settings onto a single page.

Another privacy issue is cyberbullying—using the Internet to threaten or humiliate another person with words, photos, or videos. The problem received particular attention in 2010 when a male Rutgers University student committed suicide after two acquaintances reportedly streamed a video over the Internet of the student having a sexual encounter with a man. Also in 2010, Donna Witsell, the mother of a 13-year-old Florida girl who had committed suicide in 2009 after a cyberbullying incident, formed a group called Hope's Warriors to help curb abuse and to warn others of the threat. Most U.S. states have enacted laws against bullying, although very few of them include cyberbullying.

ROLE MODEL

SIR TIM BERNERS-LEE: BRITISH SCIENTIST

Sir Tim Berners-Lee is a British computer scientist who invented what is undoubtedly one of the most revolutionary inventions of the 20th century—the World Wide Web (WWW). A qualified software engineer who was working at CERN when he came up with the idea of a global network system, Sir Tim is also credited for creating the world's first web browser and editor. He founded the World Wide Web Foundation and directs the World Wide Web Consortium (W3C). Both of his parents worked on the Ferranti Mark I, the first commercial computer, and thus it is not surprising that he too chose the field of computers. But what is surprising is the phenomenal impact his idea of a global network has had on the world of information and technology. An alumnus of the University of Oxford, he realized the need for a global communication network while working at CERN as the researchers from all over the world needed to share their data with each other. By the late 1980s he had drawn up a proposal for creating a global hypertext document system using the internet. A few more years of pioneering work in the field led to the birth of the World Wide Web making Berners-Lee one of the most significant inventors of the modern era.

Childhood & Early Life

- He was born on June 8, 1955, as Timothy Berners-Lee to Mary Lee Woods and Conway Berners-Lee. He has three siblings. Both his parents worked on the first commercially-built computer, the Ferranti Mark I and thus Tim was fascinated by computers from a young age.
- He received his primary education from Sheen Mount Primary School before moving on to London's independent Emanuel School where he studied from 1969 to 1973.
- He enrolled at The Queen's College of the University of Oxford in 1973 and graduated in 1976 with a first-class degree in physics.

Career

- He was appointed as an engineer at the telecommunications company, Plessey in Poole after completing his studies. He remained there for two years, working on distributed transaction systems, message relays, and bar code technology.
- He left Plessey in 1978 and joined D. G. Nash Ltd. In this job he wrote typesetting software for intelligent printers and a multitasking operating system.
- In the late 1970s he began working as an independent consultant and worked for many companies, including CERN where he worked from June to December 1980 as a consultant software engineer.
- While at CERN he wrote a program called "Enquire" for his own personal use. It was a simple hypertext program which laid the conceptual foundation for the development of the World Wide Web in future.
- He started working at John Poole's Image Computer Systems, Ltd. in 1981. For the next three years he worked on the company's technical side which enabled him to gain experience in computer networking. His work included real time control firmware, graphics and communications software, and a generic macro language.
- He returned to CERN in 1984 after receiving a fellowship there. During the 1980s thousands of people were working at CERN and they needed to share information and data with each other. Much of the work was done by email and the scientists had to keep track of different things simultaneously. Tim realized that a simpler and more efficient method of data sharing had to be devised.
- In 1989, he wrote a proposal for a more effective communication system within the organization which eventually led to the conceptualization of the World Wide Web—an information sharing system that could be implemented throughout the world.
- The world's first ever website, Info.cern.ch, was built at CERN and put online on 6th August 1991, ushering in a new era in the field of communication and technology. The site provided information of what the World Wide Web was and how it could be used for information sharing.
- He established the World Wide Web Consortium (W3C) at the Massachusetts Institute of Technology's Laboratory for Computer Science in 1994. The W3C decided that its technologies should be royalty-free so that anyone could adopt them.
- He became a professor in the Computer Science Department at the University of Southampton, UK, in December 2004. There he worked on the Semantic Web.

- In 2006, he became the Co-Director of the Web Science Trust which was launched to analyze the World Wide Web and devise solutions to optimize its usage and design. He also serves as the Director of the World Wide Web Foundation, started in 2009.
- Along with Professor Nigel Shadbolt, he is one of the key figures behind data.gov.uk, a UK Government project to make non-personal UK government data more accessible to the public.

Major Works

- His invention, the World Wide Web, is counted among the most significant inventions of the 20th century. The web revolutionized the world of information and technology and has opened up several new avenues.

Awards & Achievements

- He was presented with The Software System Award from the Association for Computing Machinery (ACM) in 1995.
- He was named as one of the 100 Most Important People of the 20th century by the Time Magazine in 1999.
- He was made the Commander of the Order of the British Empire (KBE) in the New Year Honors "for services to the global development of the Internet" in 2004.
- In 2013, he became one of five Internet and Web pioneers awarded the inaugural Queen Elizabeth Prize for Engineering.

Personal Life & Legacy

- He met Jane while studying physics at Oxford and married her soon after graduation in 1976. This marriage, however, ended in a divorce.
- While working for CERN he became acquainted with Nancy, an American software engineer. They so fell in love and tied the knot in 1990. This marriage too ended after some years.
- Currently he is married to Rosemary Leith who he wed in June 2014.

SUMMARY

- Web design has numerous components that work together to create the finished experience of a website, including graphic design, user experience design, interface design, search engine optimization (SEO) and content creation.
- Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization.
- Web site, Collection of files and related resources accessible through the World Wide Web and organized under a particular domain name.
- A hypertext document with its corresponding text and hyperlinks is written in Hypertext Markup Language (HTML) and is assigned an online address called a Uniform Resource Locator (URL).
- Web Standards is defined as a formal set of standards and technical specifications used to define aspects of the World Wide Web. These are best-practice standards used by organizations to build web sites and web applications.
- HTML5 is the latest iteration of the basic HTML markup language. It introduces some new rules and features, while discarding some of the elements from before versions.
- CSS (Cascading Style Sheets) provide the mechanism by which changes in the appearance or position of online content can be assigned to a document's HTML or XHTML elements.
- Internet, a system architecture that has revolutionized communications and methods of commerce by allowing various computer networks around the world to interconnect.
- The Internet was the result of some visionary thinking by people in the early 1960s who saw great potential value in allowing computers to share information on research and development in scientific and military fields.

KNOWLEDGE CHECK

1. **What is a CMS in web design?**
   a. Content Management System
   b. Creative Management System
   c. Content Mixing System
   d. Creatives Managerial System
2. **To make your website mobile friendly, you can make your website**
   a. Responsive
   b. Reactive
   c. Fast Loading
   d. Light
3. **What does CSS stand for**
   a. Current Style Sheets
   b. Current Sheets Style
   c. Cascading Style Sheets
   d. Cascading Sheets Style
4. **Which of the following statements is false**
   a. You can make a website without using HTML
   b. You can make a website without using PHP
   c. You can make a website without using CSS
   d. You can make a website without using JavaScript
5. **What is WordPress**
   a. It is a software used to press text
   b. It is a text formatting software
   c. It is a CMS (Content Management System)
   d. It is mail service
6. **SQL stands for**
   a. Structured Query Language
   b. Statistical Query Language
   c. Superior Questions Lot
   d. Standard Query Lot

7. **Which of the following is true about JavaScript**
   a. It is a server side scripting language
   b. It is client side scripting language
   c. It is a Software
   d. It is a database
8. **Which of the following is true about PHP**
   a. It is a server side scripting language
   b. It is client side scripting language
   c. It is a Software
   d. It is a database
9. **Which of the following is true**
   a. You need a server to host your website files
   b. You do not need a server to host your website files
   c. You can create a website without using HTML
   d. You cannot create a website without a CMS
10. **Which of the following software could be used to build a website**
   a. Power Point
   b. Excel
   c. Dream Weaver
   d. ERP
11. **Web page editors works on a…………. principle.**
   a. WWW
   b. HTML
   c. WYSIWYG
   d. WYGWYSI
12. **What are shared on the Internet and are called as Web pages?**
   a. Programs
   b. Cables
   c. Hypertext documents
   d. None

13. What is the name of the location address of the hypertext documents?

a. Uniform Resource Locator

b. Web server

c. File

d. Web address

REVIEW QUESTIONS

1. What do you understand by web design?
2. How do you learn web design?
3. What do you mean by web site? Also define why create a website?
4. Write a short note on World Wide Web.
5. Discuss about web standards.
6. Explain the origin and development of internet.

Check Your Result

1. (a) 2. (a) 3. (c) 4. (a) 5. (c) 6. (a)
7. (b) 8. (a) 9. (a) 10. (c) 11. (c) 12. (c)
13. (a)

REFERENCES

1. Barrell, Dylan. Agile Accessibility Explained: A practical guide to sustainable accessible software development, Amazon Digital Services, 2019.
2. Blanck, Peter. eQuality: The Struggle for Web Accessibility by Persons with Cognitive Disabilities, Cambridge Disability Law and Policy Series, 2015.
3. Burgstahle, Sheryl. Universal Design in Higher Education: From Principles to Practice, Harvard Education Press, 2008.
4. Chisholm, and May. Universal Design for Web Applications: Web Applications That Reach Everyone, O'Reilly Media, 2008.
5. Feingold , Lainey.Structured Negotiation: A Winning Alternative to Lawsuits , American Bar Association, 2016.
6. Gilbert, Regine M. Inclusive Design for a Digital World: Designing with Accessibility in Mind (Design Thinking), Apress, 2020.
7. Hamraie, Aimi. Building Access: Universal Design and the Politics of Disability, University Of Minnesota Press, 2017.
8. Hassell, Jonathan. Including your missing 20% by embedding web and mobile accessibility, BSI British Standards Institution, 2014.
9. Horton, Sarah and Quesenbery, Whitney. Universal Design for Web Accessibility, Rosenfeld Media, 2014.
10. Lazar, Jonathan; Goldstein, Daniel; and Taylor, Anne . Ensuring Digital Accessibility through Process and Policy, Morgan Kaufmann, 2015.
11. McCall, Karen. Accessible and Usable PDF Documents: Techniques for Document Authors, Fourth Edition , pubcom.com, 2017.
12. McCall, Karen. Styles in Word: A Primer for Accessible Document Design , pubcom.com, 2017.
13. Meloncon, Lisa. Rhetorical AccessAbility: At the Intersection of Technical Communication and Disability Studies , Baywood Publishing Company Inc., 2013.
14. Connor, Joshue. Pro HTML5 Accessibility, Apress, 2012.
15. Paciello, Michael G. Web Accessibility for People With Disabilities, CMP Books, 2000.
16. Pickering, Heydon. Apps For All: Coding Accessible Web Applications, Smashing Magazine GmbH, 2014.
17. Pickering, Heydon. Inclusive Design Components: The Book, Smashing Magazine, 2019.

CHAPTER

2

WEB DESIGN PRINCIPLES

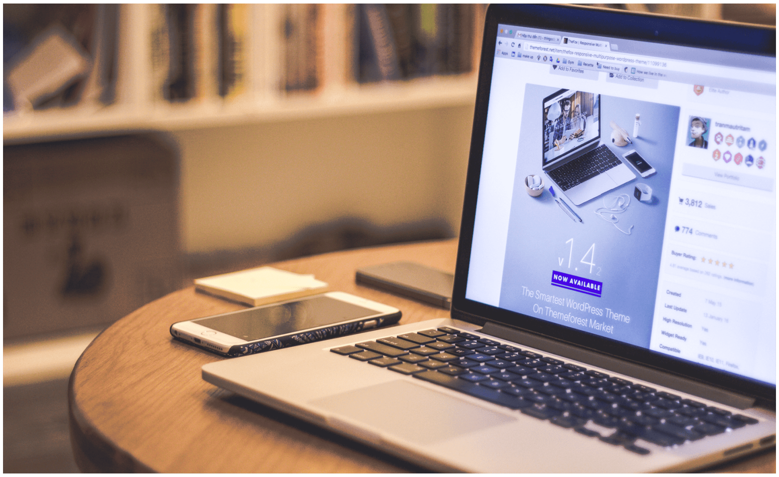
"A successful website does three things: It attracts the right kinds of visitors. Guides them to the main services or product you offer. Collect Contact details for future ongoing relation."

– Mohamed Saad

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Describe the effective web design principles
2. Explain the web site design and development process
3. Discuss the five golden rules of responsive web design
4. Discuss how to design a website template
5. Explain the prototyping a responsive web page



INTRODUCTION

Web design is a process of conceptualizing, planning, and building a collection of electronic files that determine the layout, colors, text styles, structure, graphics, images, and use of interactive features that deliver pages to ones site visitors. Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization. It also encompasses several different aspects, including webpage layout, content production, and graphic design. While the terms web design and web development are often used interchangeably, web design is technically a subset of the broader category of web development.

Keyword

Web page is a document that is suitable for the World Wide Web and web browsers. A web browser displays a web page on a monitor or mobile device.

2.1 EFFECTIVE WEB DESIGN PRINCIPLES

Web design is not only about how the website looks and feels but is also a lot about how it works and responds. When web designers work on a website, they incorporate not just those elements that add a visual appeal to it but also try to make it highly responsive, functional, quick and useful. In order to create a highly usable and effective website, designers follow certain principles that act as thumb rules or standard points to keep in mind. The following are the various principles of an effective web design:

Web design principle #1. Highly Intuitive Structure

The first law or principle of usability of a website says that a **web page** must have a highly intuitive structure and should be simple to understand so that users would not have to think which way to go. It must be self-explanatory in an obvious kind of way. Don't let any question marks or queries come up and make the navigation intuitive and simple. This helps to increase the usability of the website and also makes it much more engaging. The structure must be free from lots of cognitive load so that visitors don't have to wonder how to .move from point A to point B

Web design principle #2. Visual Hierarchy

The next principle that contributes to creating a successful and effective website is a visual hierarchy. Visual hierarchy is the order or sequence in which our eye moves and perceives the things it sees. When it comes to a web page, the visual hierarchy can be referred to the sequence in which our eye moves from one topic/content/block to another. When designing a web page, a designer first need to identify the order of importance of the various topics and then place them in such a way that the visitors first view what is most important and then move onto the others in a hierarchical manner.

There are two ways to create a visual hierarchy, and they are given as follows:

- **Size hierarchy** – As the name suggests, size hierarchy is the kind of hierarchy in which the most important content or image is of the largest size on a webpage, followed by the second most important content or image in the second largest size and so on. The distinction in sizes should be such that a visitor would view the items in the order of importance, and the pecking order of things is obvious.
- **Content hierarchy** – Besides the hierarchy of size, which is one of the best ways to create the order of importance, another way you can incorporate this principle is by creating a hierarchy of content. You can place content in such a way that the human eye first travels to the content that is most important, for example, the business's objective or purpose and then moves to the less important content blocks in a hierarchical order.

Web design principle #3. Accessibility

Another highly important principle that must not be ignored when designing a web page or website is the accessibility of it. When a visitor enters the website, he/she must be able to access each bit of information in the easiest manner. This means that the text must be legible, the colors must not be harsh on the eyes and the background must not overpower the content, etc. To make the website accessible to everyone, you can follow some of the following points:

- **Typefaces** – Make sure you select a font type and font size which is readable to all and is not too fancy for some to access or understand. For example, Fonts like Verdana, Times New Roman, Arial, etc. are simple fonts that almost everyone can easily read online. Similarly, the font size that works the best is 16 px but you can be a little flexible with it.
- **Colors** – As far as the user experience is concerned, your color scheme and contrast must be well thought of and should be able to create visual harmony and balance. It is always better to choose contrasting colors for the background and written content so that it can be easily read. Choose a darker text color and a lighter background shade so that the result is easy to the eyes. Extra bright colors must be used sparingly.

- **Images** – Do you know that the human mind perceives and processes images a lot faster than text? Well, it is thus a good idea to choose and place the right images on your web pages to communicate with the audiences in a better way. Make sure they are high-quality images and are suitable for your purpose.

Web design principle #4. Hick's law

Hick's law states that *'with every additional choice increases the time required to take a decision.* 'This law does not only hold true for web design but also in a number of other situations and settings. For example, if you visit a restaurant and are provided with too many food items to pick from, you will take a longer time to take a decision. As far as web designing is concerned, the more options you offer to your visitors, the more difficult will the website become to use and browse through. This means that we need to reduce the number of choices in order to provide a better user experience. Distracting options have to be eliminated to aid increased sales and better overall profit.

Hick's law can also be translated to 'More options mean less sales' In order to incorporate this law without having to sacrifice giving all product or service options that you have, you can organize the products in a hierarchy with the main categories shown in the sidebar and all the products of that category in a separate list.

Web design principle #5. Fitt's Law

Another law that acts as a major principle in web design is Fitt's law. According to this law, the time needed to move to a target is dependent upon the size of the target as well as the distance to the target. This means that the larger the object or target and the closer it is in the distance, the easier would it be to move to it or reach it. This law can effectively be incorporated in web design and is something that can enhance your web design by leaps and bounds. However, this does not mean that the bigger, the better but means that usability factor of a target runs as a curve and not as a straight line. When you apply this law to your web design, then users may be more motivated and encouraged to press the button that you want them to press.

If you want your website visitors to take actions like order a product, read about a service or click on something, then you must make sure that they can reach the 'click here' more easily and quickly. Thus, it is a good idea to consider this law and use it well.

Keyword

Visible Language is a peer-reviewed design journal that advocates the potential for the research and practice of visual communication to enhance the human experience.

Web design principle #6. Communication and Content

Everyone who visits your website is looking for some or the other kind of information or content, and thus it is very important for you to communicate with them clearly and in an engaging manner. Your information must be compelling, easy to read and easy to process. Communication is not just about providing written information but also about offering images, infographics and another form of media such as videos and audio pieces. Web design takes into consideration a concept called 'Visible language'. According to this concept, **visible language** is the content people see on the screen and involves three fundamental principles. They are given as follows:

- **Organize** – The first principle of visible language or effective communication is to provide the visitors of your website a clear and highly consistent layout or structure. Some important parts of the organization of concept include the layout involving the division of content, consistency, navigation, as well as visual appeal. In order to make your website communicate effectively, you need to organize and place your blocks of information in an easy to process manner.
- **Economize** – This principle involves achieving more by utilizing less. This means that you must try to deliver the most information by taking helping of less visual elements. It must throw emphasis on what is important and create distinctiveness between different elements.
- **Communicate** – The user interface of the web design must be such that users are able to understand everything in the most convenient way. There must be a balance between readability, legibility, color, texture, and views.

Remember

The design or layout must be free of clutter, should be simple and must have a high degree of clarity.

Besides this, the content given on the website must be appropriate, well written, and easy to understand and should be written keeping in mind the concept of search engine optimization. It is important that you adjust the style of writing according to the preference of the target audience and avoid promotional writing. The writing must pull visitors and must leave them intrigued and wanting for more. It must clearly spell out what you do and what you aim to achieve.

Web design principle #7. White space and Simple Design

A simple design is an effective design. Complexity is just not something that a visitor wants to see on your web page, and one of the most important aspects of a simple design is white space. White space is something that web designers must not be afraid of. White space not only helps to take the cognitive load off the website but also makes it easier for users to perceive the information provided on the website. White space helps to divide the web page into several distinct parts or areas which makes it simpler to process information. It is always better to have a whitespace solution to the problem of complex hierarchical structures. The following are some of the other things that :can be considered as a part of a simple design

- **Grid-based layout** – To avoid a messy structure or appearance of the website, you must opt for a grid-based layout in which content is divided into columns, boxes, and different sections.
- **F-pattern design** – It is a fact that the human eye scans screens in an ‘F’ pattern. Thus, it is a good idea to design a web page or website in a way that complements the natural reading behavior of the visitors.
- **Conventional designs** – Conventional designs are not always boring and rather work well as far as visitor response or likeability is concerned. They add a hint of trust, reliability as well as brand credibility.

Web design principle #8. Regular Testing

Test Early and Test Often or ‘TETO’ is another web design principle that all designers and website owners must consider. Conducting usability tests every now and then provide important results and insights into many kinds of problems and complications related to a website layout or aspects of design. What happens is that websites often tend to get into certain issues and by not testing them often, they may create issues that could be driving visitors away. Websites constantly need upgrades and updates so as to maintain the visitor footsteps and customer interest.

The following are some points you must keep in mind to test your webpage or website:

- Testing one user at the beginning of a project is better than testing 50 users towards the end as it makes way for improvements that could prove important for driving traffic and increasing sales. Also solving errors during process is least expensive rather than later.
- Another point to be kept in mind is that testing is an iterative process which means that designers must first design, then test, then fix and then test again. There is a strong chance that there may be some problems that were not solved the first time because of diversion of attention to other major issues.

2.2 THE WEB SITE DESIGN AND DEVELOPMENT PROCESS

There are numerous steps in the web site design and development process. From gathering initial information, to the creation of your web site, and finally to maintenance to keep your web site up to date and current.

The exact process will vary slightly from designer to designer, but the basics are the same.

- Information Gathering
- Planning
- Design
- Development
- Testing and Delivery
- Maintenance

Phase One: Information Gathering

The first step in designing a successful web site is to gather information. Many things need to be taken into consideration when the look and feel of your site is created.

This first step is actually the most important one, as it involves a solid understanding of the company it is created for. It involves a good understanding of *you* – what your business goals and dreams are, and how the web can be utilized to help you achieve those goals.

It is important that your web designer start off by asking a lot of questions to help them understand your business and your needs in a web site.

Certain things to consider are:

- **Purpose**: What is the purpose of the site? Do you want to provide information, promote a service, sell a product… ?
- **Goals**: What do you hope to accomplish by building this web site? Two of the more common goals are either to make money or share information.
- **Target Audience**: Is there a specific group of people that will help you reach your goals? It is helpful to picture the "ideal" person you want to visit your web site. Consider their age, sex or interests – this will later help determine the best design style for your site.
- **Content**: What kind of information will the target audience be looking for on your site? Are they looking for specific information, a particular product or service, online ordering…?

Keyword

Business goals describe what a company expects to accomplish over a specific period of time and are very important for the health and well-being of any company.

Phase Two: Planning

Using the information gathered from phase one, it is time to put together a plan for your web site. This is the point where a site map is developed.

The site map is a list of all main topic areas of the site, as well as sub-topics, if applicable. This serves as a guide as to what content will be on the site, and is essential to developing a consistent, easy to understand navigational system. The end-user of the web site – aka your customer – must be kept in mind when designing your site. These are, after all, the people who will be learning about your service or buying your product. A good user interface creates an easy to navigate web site, and is the basis for this.

During the planning phase, your web designer will also help you decide what technologies should be implemented. Elements such as what CMS (content management system) such as WordPress to incorporate, will any contact forms be needed, etc.

Phase Three: Design

Drawing from the information gathered up to this point, it's time to determine the look and feel of your site.

Target audience is one of the key factors taken into consideration. A site aimed at teenagers, for example, will look much different than one meant for a financial institution. As part of the design phase, it is also important to incorporate elements such as the company logo or colors to help strengthen the identity of your company on the web site.

Your web designer will create one or more prototype designs for your web site. This is typically a .jpg image of what the final design will look like. Often times you will be sent an email with the mock-ups for your web site, while other designers take it a step further by giving you access to a secure area of their web site meant for customers to view work in progress.

Either way, your designer should allow you to view your project throughout the design and development stages. The most important reason for this is that it gives you the opportunity to express your likes and dislikes on the site design.

In this phase, communication between both you and your designer is crucial to ensure that the final web site will match your needs and taste. It is important that you work closely with your designer, exchanging ideas, until you arrive at the final design for your web site.

Phase Four: Development

The developmental stage is the point where the web site itself is created. At this time, your web designer will take all of the individual graphic elements from the prototype and use them to create the actual, functional site. This is typically done by first developing the home page, followed by a "shell" for the interior pages. The shell serves as a template for the content pages of your site, as it contains the main navigational structure for the web site. Once the shell has been created, your designer will take your content .and distribute it throughout the site, in the appropriate areas

Elements such as the **CMS (content management system)** like WordPress, interactive contact forms, or ecommerce shopping carts are implemented and made functional during this phase, as well.

This entire time, your designer should continue to make your in-progress web site available to you for viewing, so that you can suggest any additional changes or corrections you would like to have done.

On the technical front, a successful web site requires an understanding of front-end web development. This involves writing valid HTML / CSS code that complies with current web standards, maximizing functionality, as well as accessibility for as large an audience as possible.

Keyword

Content management system (CMS) manages the creation and modification of digital content. It typically supports multiple users in a collaborative environment.

Phase Five: Testing and Delivery

At this point, your web designer will attend to the final details and test your web site. They will test things such as the complete functionality of forms or other scripts, as well last testing for last minute compatibility issues (viewing differences between different web browsers), ensuring that your web site is optimized to be viewed properly in the most recent browser versions.

A good web designer is one who is well versed in current standards for web site design and development. The basic technologies currently used are HTML and CSS (Cascading Style Sheets). As part of testing, your designer should check to be sure that all of the code written for your web site validates. Valid code means that your site meets the current web development standards – this is helpful when checking for issues such as cross-browser compatibility as mentioned above.

Once you give your web designer final approval, it is time to deliver the site. An FTP (File Transfer Protocol) program

Did You Know?

The World Wide Web (WWW) was created in 1990 by the British CERN physicist Tim Berners-Lee.

is used to upload the web site files to your server. Some web designers offer domain name registration and web hosting services as well, or have recommendations as to where you can host your site. Once these accounts have been setup, and your web site uploaded to the server, the site should be put through one last run-through. This is just precautionary, to confirm that all files have been uploaded correctly, and that the site continues to be fully functional.

Other final details include plugin installation (for WordPress or other CMS driven web sites and SEO (Search Engine Optimization). SEO is the optimization of your web site with elements such as title, description and keyword tags which help your web site achieve higher rankings in the search engines. The previously mentioned code validation is something that plays a vital role in SEO, as well. There are many WordPress plugins available that further enhance the default WordPress functionality – many of which directly relate to improving your SEO, as well.

There are a lot of details involved in optimizing your web site for the search engines – enough to warrant its own post. This is a very important step, because even though you now have a web site, you need to make sure that people can find it!

Phase Six: Maintenance

The development of your web site is not necessarily over, though. One way to bring repeat visitors to your site is to offer new content or products on a regular basis. Most web designers will be more than happy to continue working together with you, to update the information on your web site. Many designers offer maintenance packages at reduced rates, based on how often you anticipate making changes or additions to your web site.

If you prefer to be more hands on, and update your own content, there is something called a CMS (Content Management System) such as WordPress can be implemented to your web site. This is something that would be decided upon during the Planning stage. With a CMS, your designer will utilize online software to develop a database driven site for you.

A web site driven by a CMS gives you the ability to edit the content areas of the web site yourself. You are given access to a back-end administrative area, where you can use an online text editor (similar to a mini version of Microsoft Word). You'll be able to edit existing content this way, or if you are feeling more adventurous, you can even add new pages and content yourself. The possibilities are endless!

It's really up to you as far as how comfortable you feel with updating your own web site. Some people prefer to have all the control so that they can make updates to their own web site the minute they decide to do so. Others prefer to hand off the web site entirely, as they have enough tasks on-hand that are more important for them to handle directly.

That's where the help of your web designer comes in, once again, as they can take over the web site maintenance for you – one less thing for you to do is always a good thing in these busy times!

Other maintenance type items include regular site backups, WordPress upgrades, additional plugin installation, etc.

2.2.1 How to Design a Website

Designing Your Website

1. **Determine whether you want to use a website creator.** Websites created from scratch require a fairly detailed understanding of HTML coding, but you can easily create a website by using a free hosting service like Weebly, Wix, WordPress, or Google Sites. Website creators tend to be much easier for first-time designers to use than HTML.

- If you do decide to code your own website, you'll need to learn both HTML and CSS coding.
- If investing the time and energy to create your website doesn't sound appealing, you can also hire a website designer to create your site for you. Freelance designers can cost anywhere between $30 an hour and well over $100 per hour.

2. **Map out your site.** Before you even open a website creator, you should know approximately how many pages you want your website to have, what the content on each of those pages should be, and the general layout of important pages such as the home page and the «About» page.

- Your website's pages may be easier to visualize if you draw rough pictures of each of them rather than just determining what content should appear.

3. **Use intuitive design.** While there›s something to be said for fresh ideas, your website›s basic design should follow established guidelines such as the following:

- Navigation options (e.g., tabs for different webpages) should go at the top of the page.
- If you use a menu icon (□), it should be in the top-left corner of the page.
- If you use a search bar, it should be near the upper-right side of the page.
- Helpful links (e.g., links to the "About" page or the "Contact Us" page) should go at the very bottom of each page.

4. **Be consistent.** Whatever text font, color palette, image theme, and design options you choose, make sure that you use the same decision throughout your website. It can be incredibly jarring to see one font or color scheme used for the «About» page when a completely different one was used for the home page.

- For example, if you exclusively use cool-tone colors for your site's home page, don't implement bright, loud colors on the next page.
- Keep in mind that using loud or clashing colors, especially when the colors display in a dynamic (e.g., moving) fashion, can evoke epilepsy in a small number of web users. If you decide to use such colors on your site, make sure you add an epilepsy warning before any relevant pages.

5. **Add navigation options.** Placing direct links to important pages on your website at the top of the home page will help direct first-time visitors to the content that matters. Most site creators add these links by default.

- It’s important to make sure that every page on your website is accessible by clicking through options on your website rather than only being accessible via the page’s address.

6. **Use colors which complement each other.** Like any other kind of design, website design relies on visually pleasing combinations of color; because of this, picking theme colors which go together is crucial.

- Black, white, and grey is a good combination if you don’t know where to start.

7. **Consider using a minimalist design.** Minimalism encourages cool-tone colors, simple graphics, black-on-white text pages, and as little embellishment as possible. Because minimalism requires little in the way of fancy elements, it›s an easy way to make your website look professional and attractive without requiring much work.

- Many website creators will have a “minimalist” theme you can select when setting up your website.
- An alternative to minimalism is “brutalism”, which uses harsher lines, bright colors, bold text, and minimal imagery. Brutalism has less of a following than does minimalism, but depending on your website’s content, it may better suit your design needs.

8. **Make unique choices.** Straight lines and grid-locked web elements are safe bets, but making a few unique stylistic decisions will both lend personality to your site and help your site stand out.

- Don’t be afraid to buck trends by placing website elements asymmetrically or using overlapping elements to create a layered appearance.
- While elegant, sharp corners and squared-off elements (also known as a “card-based” presentation) are less favorable than are rounded, soft elements.

2.3 THE FIVE GOLDEN RULES OF RESPONSIVE WEB DESIGN

Designing a great website isn’t only about combining the components to create an attractive and user friendly interface. Website designing, in fact, requires a well-researched strategy. Only investing handsomely on web designing agencies isn’t going to help your business much, no matter how proficient resources you hire and how much time you allot. If you are planning to launch a website for your business, you got to do extensive research based on your needs and working style, to have one that drives your business towards success. The same also goes for the website designers who need to have a clear idea about how to fuse all the elements in an effective manner so that each one of them contributes towards a common goal. Web designing combines a lot of components including layout, topography, aesthetics and many more that ensure a website’s objective oriented and smooth functioning.

Here are five golden rules of web designing that you must follow to create a great website:

1. Define website goal

Define the purpose of your website and design an interface that serves the purpose efficiently. Focus on what you are trying to achieve for your organization through your website, like whether you want to entertain people or deliver information or sell products. It's important to combine the web designing elements in a fashion that helps you achieve your goal.

2. Highlight your brand image

Focus on your brand image and choose elements that best defines you. The look and feel of a website matters a lot. It should mirror the character of your brand and impress the visitors by eliciting the right emotions. Don't get driven by the latest trend and carefully analyses what color schemes and images would give the desired "*feel*" to the visitors.

3. Identify target audience

Analyze your customers and prospective clients on the basis of their gender, age, profession, behavior and technical competency. Identifying the target audience helps to decide upon the aesthetics and functionalities of websites. Determine whether your target audience are tech savvy or kids or women and choose a user-centric design.

4. Analyze competitor websites

Analyzing the competitor websites is important to get your website noticed among so many, offering similar services. Scrutinize what competitors are doing and incorporate something unique in your website to make a better emotional connection with your audience. Also, optimize your web design to get a good search engine ranking. Using relevant keywords in your content boosts your ranking and maintains the web visits.

5. Evaluate and improve

After designing and deploying your website, measure the results and find out if your website is actually working towards the specified goal in the specified manner. Ensure that your strategic planning and implementation has coincided and the website is efficient in fulfilling your goals. Continuous improvement is important for every website to increase visits and win clients. By measuring results and through feedback you can find out errors, areas that need improvements and elements that are required to be updated.

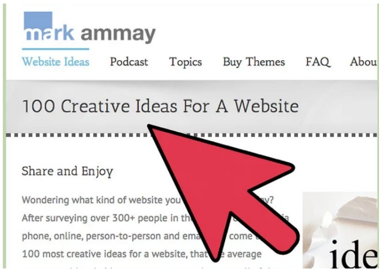
Web designing requires consistent focus on your goals to create a website that fulfils a specific purpose. But often the latest trends and alluring components divert the attention and people end up having a beautiful website that doesn't actually work towards the objective. Avoid falling in such traps and design a website that maintains a perfect balance between its aesthetics and functionality.

2.4 HOW TO DESIGN A WEBSITE TEMPLATE

This is an in-depth guide for those who know HTML and CSS but are clueless in how to make your layout.

Steps

1. Decide what your website topic is. There are millions of different kinds of websites that you can choose to your liking.

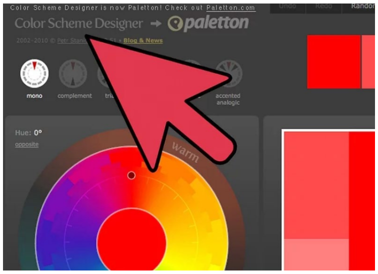


100 Creative Ideas For A Website

Share and Enjoy

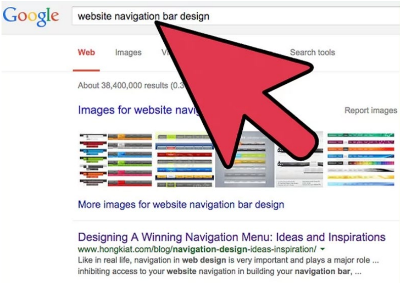
Wondering what kind of website you y?
After surveying over 300+ people in th ia
phone, online, person-to-person and ema come
100 most creative ideas for a website, that e average

2. Find the right color scheme. Try using colors that will pop out, like blue, purple or orange. You can also adjust the color to the topic of your website.



Color Scheme Designer is now Paletton! Check out Paletton.com

3. Decide what kind of navigation bar to use. Make sure it has useful resources, such as home, comments, and more.



website navigation bar design

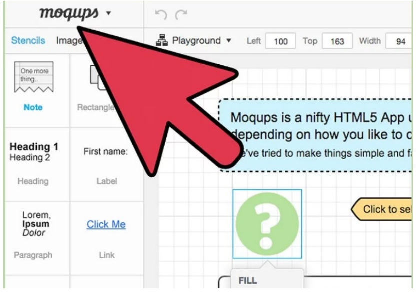
About 38,400,000 results (0.3

Images for website navi

More images for website navigation bar design

Designing A Winning Navigation Menu: Ideas and Inspirations
www.hongkiat.com/blog/navigation-design-ideas-inspiration/
Like in real life, navigation in web design is very important and plays a major role ...
inhibiting access to your website navigation in building your navigation bar, ...

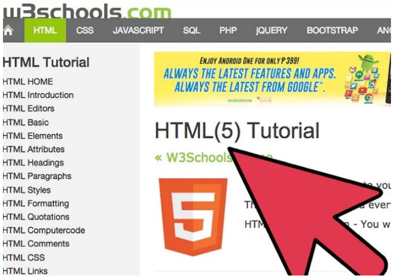
4. Mock up how your website will be laid out in a graphic editing program. Just like the topics, there are wide choices on how you would like the layout of each page to be.



Left 100 Top 163 Width 94

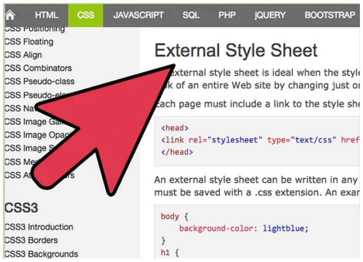
Moqups is a nifty HTML5 App u
depending on how you like to c
e've tried to make things simple and f

5. Create the page in HTML. You can also create it by going onto Google websites. Be sure to pick a creative name too!



HTML(5) Tutorial

6. Create a style sheet in CSS.



External Style Sheet

xternal style sheet is ideal when the styl
k of an entire Web site by changing just o

ach page must include a link to the style sh

```
<head>
<link rel="stylesheet" type="text/css" href
</head>
```

An external style sheet can be written in any
must be saved with a .css extension. An exar

```
body {
    background-color: lightblue;
}
h1 {
```

7. Duplicate the HTML page for the other pages and add content. This article is just an in-depth for layouts.

2.5 PROTOTYPING A RESPONSIVE WEB PAGE

The primary purpose of the study was to examine the principle behind the responsive web and use that principle and implement a simple prototype. To find the best practical way is to design a simple web page and make it adaptive and responsive. The prototype was carried out while following the **software life-cycle development** phases: requirement analysis, overall design, implementation and testing the result.

2.5.1 Requirement Analysis

The requirement analysis is the initial phase of software life-cycle development. In this phase the web page prototype requirements were analyzed and outlined. The requirements of the web page are listed below:

1. Responsive design approach

Keyword

Systems development life cycle (SDLC) is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system.

2. Uses the latest technologies HTML5 and CSS3
3. Following the design principles
4. Supporting cross browser / device testing

The First approach explains that the web page has to be responsive with mobile, tablet and desktop browsers.

In the web the basic web technologies are HTML, CSS and JavaScript. HTLM5 has more benefits compared to HTML4. Applying the new feature such as picture element, semantic tags and media element of HTML5 will enhance the design. Similarly, the web page has to use the CSS3 media query for the latest presentation and visual design of the web page. Rich user interface and better user experiences are valuable key points for web page success.

After analyzing the third requirement it was decided to follow the mobile first principle. This means the prototype was designed for a mobile initially. Combining JavaScript and CSS was done using a normal compression method, so that the script was minified for optimization. Finally, the fourth requirement is about testing the design in the latest version of browsers: Chrome, Mozilla, Internet explorer, Safari. It is also important that the design is appropriate for each device desktop, laptop, tablets, and mobiles. It does not mean the design should be the same, it is about coherent experience across all devices.

2.5.2 Overall Design

After analyzing the basic requirements, the user interface sketch layout was designed before the real implementation of web pages. The desktop, tablets and mobile layout of the content was designed separately. This helped to establish the system design of the whole web page. Sketching the webpage layout before implementing helped to identify the breakpoints of responsive design.

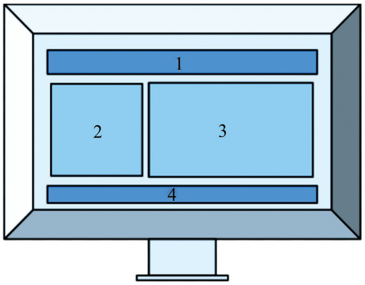


Figure 1: Sketch diagram for desktop layout.

The desktop browser's screen varies from 1024 to 1200 pixels in average. This range of screen resolution works for most laptops and notebooks. Designing the target resolution will be appropriate for all wide screen browsers. Figure 1 shows the layout primarily for a desktop browser with a higher than 1024 pixel screen resolution. Web page has well fitted web container with respective header, sidebar and container and footer which are represented with numbers 1, 2, 3 and 4 respectively. In the desktop design the web container has enough margin and padding on it.

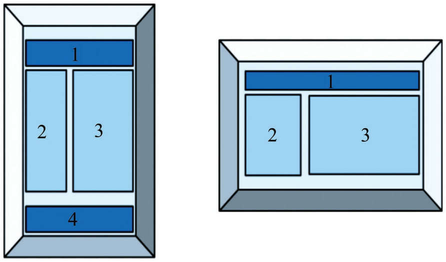


Figure 2: Sketch diagram for a tablets layout in the portrait and landscape modes.

Tablets have a screen resolution that varies from 600 to 1024 pixels in average. At present there are small tablets, 10-inch tablets from various vendors such as Apple, Google, and Microsoft. In figure 2, the sketch diagram for a tablet layout was designed. In the portrait mode the screen view ports support the entire layout by adjusting a dynamic web page container. Similarly this design is not supported in the landscape mode, so a separate design for the landscape mode was prepared which contained only container 1, 2 and 3 while the page loads. Upon scrolling down the user can see the fourth container.

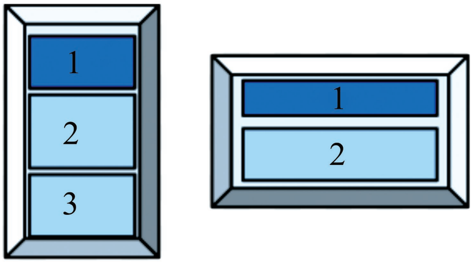


Figure 3: Sketch diagram for mobile layout in portrait and landscape modes.

In figure 3, the mobile sketch layout was designed. The screen widths of the mobile are nearly 80 percent smaller than the desktop width. The mobile screen resolution starts from 320 in the portrait mode to 480 in the landscape mode. The container of the mobile layout cannot fit the same layout as shown in the tablet and desktop layouts. So that the mobile design of the web page would fit in a single column. A single column layout that best fits the mobile screen was designed and containers

1, 2, 3 and 4 were aligning in a vertical order. By removing padding and margin and adjusting the entire container layout it is possible to have high user interface design for mobile users. Aligning these containers is based on the priority of a container. For example if a page has an important content in container 2 instead of container 1, then it is good practice to align in the top of the container stack.

2.5.3 Implementation

Once the layout and designed are finalized in the sketch which gives the basic idea of how the web page renders in different viewports, following the mobile first principle the first web page was designed for mobile using Dreamweaver CS5. Web development software, used for creating static and dynamic **web applications** and web pages. The created product was installed on a desktop local server and tested with latest browsers. The implementation procedures are briefly described below.

Keyword

Web application or web app is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser.

After the installation of Dreamweaver web platform, the new HTML5 project was created.

The Clientside is the name of the folder that represents the public_html folder where a website's index page is normally stored. It contains the entire required folder for storing CSS, JavaScript and HTML5 files, which is good practice because it makes it easy to navigate between files and locate the files when needed. The actual implementation of the requirement was started after creating this folder structure. Necessary files for each type of document were placed in their respective folders. The prototype was designed for adaptive and responsive web layout. For this reason flexible grid, flexible media, and media queries were implemented.

For the flexible grid, the width of the container was calculated from the context formula.

```
#wrapper aside {
margin:1%;
float:left;
width:40%;
box-shadow:0px 0px 3px 1px #888;
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px;
padding:2%;
min-height:10px;
}
#wrapper section{
margin:1%;
float:right;
width:56%;
box-shadow:0px 0px 3px 1px #888;
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px;
padding:2%;
min-height:100px;
}
```

The web page has main two containers excluding the header and footer. These two containers width was defined in a percentage value so that the containers were flexible when the width of browser narrowed down. The width of both containers is 96% of the viewports width. The 4% of width was used for the margin of the container.

Similarly, the page has two embedded videos in the aside container. For the demonstration purpose the video container was used for flexible media. For the flexible media, the video container defines as maximum width 100%. Finally, to achieve real adaptive and responsive design the CSS media queries were implemented inside the mediascreen.css file. The file has several breakpoint conditions to achieve different layouts for appropriate viewports. This is a simple prototype, so two breakpoint conditions were used. The breakpoint condition can vary according to the list of the supported viewport. If none of these conditions are matched then a default CSS would be applied for the web page. Those conditions that are not defined in this breakpoint are default.

```
@media only screen and (min-width :320px) and (max-width:600px){ // CSS style }
@media only screen and (min-width: 600px) and (max-width: 1024px){ // CSS style }
```

For example as shown in listing 4, if the media type is screen and the minimum width is 320 pixels to maximum width is 600 pixels, which is the range of a mobile and small tablet's viewport, the first condition will be true and all the CSS properties defined inside these conditions will be applied.

SUMMARY

- Web design is not only about how the website looks and feels but is also a lot about how it works and responds. When web designers work on a website, they incorporate not just those elements that add a visual appeal to it but also try to make it highly responsive, functional, quick and useful.
- Communication between both you and your designer is crucial to ensure that the final web site will match your needs and taste. It is important that you work closely with your designer, exchanging ideas, until you arrive at the final design for your web site.
- Elements such as the CMS (content management system) like WordPress, interactive contact forms, or ecommerce shopping carts are implemented and made functional during this phase, as well.
- A good web designer is one who is well versed in current standards for web site design and development. The basic technologies currently used are HTML and CSS (Cascading Style Sheets).
- Web development software, used for creating static and dynamic web applications and web pages. The created product was installed on a desktop local server and tested with latest browsers.
- The web page has main two containers excluding the header and footer. These two containers width was defined in a percentage value so that the containers were flexible when the width of browser narrowed down. The width of both containers is 96% of the viewports width. The 4% of width was used for the margin of the container.

KNOWLEDGE CHECK

1. **The first page of a website is called?**
   a. Design page
   b. Home page
   c. First page
   d. Main page
2. **Which of the following statements is true**
   a. The web designer shouldn't just be concerned about the looks but also about user interface
   b. Usability is very important in web design
   c. a and b
   d. None of the above
3. **Which of the following is not a web hosting company**
   a. Hostgator
   b. Blue Host
   c. WPX Hosting
   d. Facebook
4. **A web designer works on the appearance, layout, and, in some cases, content of a website.**
   a. True
   b. False
5. **Web design takes into consideration a concept called 'Visible language'.**
   a. True
   b. False
6. **Which statement best describes how a web user reads a website?**
   a. They don't they only ever look at pictures
   b. Scan the site looking for content
   c. Carefully read every word that is written
   d. None of these
7. **URL is an acronym for**
   a. Universal research locator
   b. Universal resource locator

c. Uniform research locator

d. Uniform resource locator

8. Www.lycos.com was developed by

a. Cambridge university

b. California university

c. Carnegie-Mellon university

d. None of the above

9. TCP is an acronym for

a. Transmission control protocol

b. Transit control protocol

c. Transmission calling protocol

d. None of the above

10. Which of the following statements is true

a. The web designer shouldn't just be concerned about the looks but also about user interface

b. Usability is very important in web design

c. a and b

d. None of the above

REVIEW QUESTIONS

What is principle of web design?

What is process of Website design?

How to design a website?

Discuss the prototyping a responsive web page.

Describe the five golden rules of responsive web design.

Check Your Result

1. (b)	2. (c)	3. (d)	4. (a)	5. (a)
6. (b)	7. (d)	8. (c)	9. (a)	10. (c)

REFERENCES

1. Krause, Jim. Color Index: Over 1100 Color Combinations, CMYK and RGB Formulas, for Print and Web Media, How Design Books, 2002.
2. Pickering, Heydon. Inclusive Design Patterns, Smashing Magazine, 2016.
3. Seale, Jane. E-learning and Disability in Higher Education, Routledge, 2014.
4. Slatin, John M. and Rush, Sharron. Maximum Accessibility: Making Your Web Site More Usable for Everyone, Addison Wesley Professional, 2002.
5. Sydik, Jeremy. Design Accessible Web Sites, Pragmatic Bookshelf, 2007.
6. Thatcher, Jim et al. Constructing Accessible Web Sites, Glasshaus, 2002.
7. Thatcher, Jim et al. Web Accessibility: Web Standards and Regulatory Compliance, friends of ED, 2006.
8. Watson, Leonie et al. Practical Approaches For Designing Accessible Websites, Smashing Magazine GmbH, 2015. (ISBN (EPUB: 978-3-945749-23-4).
9. Nielsen, Jakob (10 November 2003), The Ten Most Violated Homepage Design Guidelines, Nielsen Norman Group, archived from the original on 5 October 2013
10. Spool, Jared (29 September 2005), Is Home Page Design Relevant Anymore?, User Interface Engineering, archived from the original on 16 September 2013
11. Porter, Joshua (24 April 2006), Prioritizing Design Time: A Long Tail Approach, User Interface Engineering, archived from the original on 14 May 2013
12. Spool, Jared (6 August 2007), Usability Tools Podcast: Home Page Design, archived from the original on 29 April 2013
13. Knight, Kayla (20 August 2009), Essential Tips for Designing an Effective Homepage, Six Revisions, archived from the original on 21 August 2013
14. Chapman, Cameron (15 September 2010), 10 Usability Tips Based on Research Studies, Six Revisions, archived from the original on 2 September 2013
15. Bates, Chris (9 October 2012), Best practices in carousel design for effective web marketing, Smart Insights, archived from the original on 3 April 2013
16. Messner, Katie (22 April 2013), Image Carousels: Getting Control of the Merry-Go-Round, Usability.gov, archived from the original on 10 October 2013
17. Jones, Harrison (19 June 2013), Homepage Sliders: Bad For SEO, Bad For Usability, archived from the original on 22 November 2013

CHAPTER

3

INTRODUCTION TO HTML

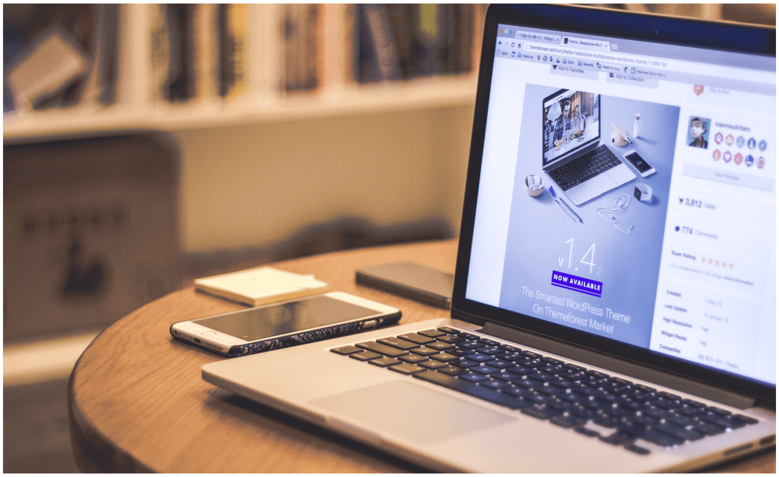
"If someone had protected the HTML language for making Web pages, then we wouldn't have the World Wide Web."

– Feng Zhang

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Learn about the hypertext markup language
2. Know about the HTML version



INTRODUCTION

HTML, or Hypertext Markup Language, is a markup language for documents designed to be displayed in a web browser. When used in conjunction with other technologies like CSS and JavaScript, it creates the vast majority of content seen on websites. HTML is used for a huge variety of things on the web, from building complex websites that offer email and calendar functions to constructing a simple course website or resume. We recommend at least a basic understanding of HTML for anyone who wants to publish on the web.

```
<html>
<head id="html_head">
   <title>website</title>
   <script type="text/javascript">
      pageNow = new Date();
   </script>
```

HTML is not considered a programming language as it can't create dynamic functionality. Instead, with HTML, web users can create and structure sections, paragraphs, and links using elements, tags, and attributes.

Here are some of the most common uses for HTML:

- **Web development.** Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
- **Internet navigation.** Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
- **Web documentation**. HTML makes it possible to organize and format documents, similarly to Microsoft Word.

3.1 HYPERTEXT MARKUP LANGUAGE

HyperText Markup Language (HTML) is a type of computer language that is used to create pages that can be posted on the Internet or sent via email. Although it might seem complex to many people, it is considered to be a relatively simple language. All text, graphics, and design elements a page designed with this language are "tagged" with codes that instruct the web browser or email program how to display those elements. The tags also provide layout and formatting information so that the web page or email will look as close as possible to the way its designer intended it to look. For the novice website designer or anyone else who needs to know a certain code or wants to learn how to create an entire website, there are many software utilities, programs and websites that can assist him or her in writing HTML code.

Types of Codes. There are many codes to allow for different text formatting, including italics, tables, paragraphs, and hyperlinks to web pages. Codes also can indicate to the browser or email program how to display or use other elements, such as pictures, graphics, video, and sound. Other types of codes without angle brackets can be used to create punctuation marks, diacritics, and other symbols that might appear in text. Although all web browsers and many email programs use HTML, each might interpret and display the code a little differently, and designers often must consider these variations when creating a web page.

An Evolving Language. Since the development of HTML in the early 1990s by British computer scientist Tim Berners-Lee, there have been many changes and versions. These versions have been maintained by the World Wide Web Consortium (W3C) since 1996. In January 2008, the First Public Working Draft of HTML 5 was published by the working group that was developing this specification. Still under development as of 2011, this revision was expected to dramatically change application development for the web. It introduces a number of new elements, including those for site structure, interactivity, and audio and video support, as well as new attributes.

Viewing a Web Page's Code. The code used to create any web page can be seen by navigating a browser to the page, then choosing the correct option from the browser's menu. In most browsers, the user can click on the "view" menu and select an option such as "source," "view source" or "page source." This will cause a pop-up window to appear, and it will show the code that was used to create that web page.

It is important to note that not all of the content found on all web pages is written in HTML. Extensible Markup Language (XML) and Extensible HyperText Markup Language (XHTML) are other types of markup languages used in web development. In addition, style sheets — like Cascading Style Sheets (CSS) — are used to attach style to HTML documents. Languages like Flash® and Java® are used to create interactive content. Many other programming languages can be used to add specific functionality to a website.

As Compared to HTM. Practically speaking, there's little difference between HTM and HTML extensions, since both are read as an HTML file by most machines. The reason there were two different extensions to begin with is that certain types of computers, like those that ran on 16 bit DOS or Windows® 3 systems, could not read four character extensions, and so needed the three character HTM extension. Most systems that can read four character extensions are automatically programmed to recognize HTM files as HTML files, though computer users may occasionally need a converter to change a file from HTM into a format that the system recognizes.

3.1.1 Basic HTML Document

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

- As its name suggests, HTML is a Markup Language which means you use HTML to simply «mark-up» a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

In its simplest form, following is an example of an HTML document −

```
<!DOCTYPE html>
<html>

   <head>
      <title>This is document title</title>
   </head>

   <body>
      <h1>This is a heading</h1>
      <p>Document content goes here.....</p>
   </body>

</html>
```

Keyword

Markup language is a system for annotating a document in a way that is syntactically distinguishable from the text.

HTML Tags

As told earlier, HTML is a **markup language** and makes use of various tags to format the content. These tags are enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their corresponding closing tags. For example, <html> has its closing tag </html> and <body> tag has its closing tag </body> tag etc.

Above example of HTML document uses the following tags –

Sr.No	Tag & Description
1	<!DOCTYPE...> This tag defines the document type and HTML version.
2	<html> This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
3	<head> This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
4	<title> The <title> tag is used inside the <head> tag to mention the document title.
5	<body> This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
6	<h1> This tag represents the heading.
7	<p> This tag represents a paragraph.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

HTML Document Structure

A typical HTML document will have the following structure –

```
<html>


   <head>
      Document header related tags
   </head>
```

```
<body>

Document body related tags

</body>


</html>
```

Remember

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

Let's see what is document declaration tag.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration –

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing <!DOCTYPE...> tag along with other HTML tags.

3.1.2 HTML - Basic Tags

The <html> tag tells the browser that this is an HTML document. The <html> tag represents the root of an HTML document. The <html> tag is the container for all other HTML elements (except for the <!DOCTYPE> tag).

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**. While displaying any heading, browser adds one line before and one line after that heading.

Example:

```
<!DOCTYPE html>
    <html>
```

```
   <head>
      <title>Heading Example</title>
   </head>

   <body>
      <h1>This is heading 1</h1>
      <h2>This is heading 2</h2>
      <h3>This is heading 3</h3>
      <h4>This is heading 4</h4>
      <h5>This is heading 5</h5>
      <h6>This is heading 6</h6>
   </body>

</html>
```

This will produce the following result –

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example –

Example:

```
<!DOCTYPE html>
<html>

   <head>
      <title>Paragraph Example</title>
   </head>

   <body>
      <p>Here is a first paragraph of text.</p>
      <p>Here is a second paragraph of text.</p>
      <p>Here is a third paragraph of text.</p>
   </body>

</html>
```

This will produce the following result −

```
Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.
```

Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <br /> tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML.

Example:

```
<!DOCTYPE html>
<html>

   <head>
      <title>Line Break  Example</title>
   </head>

   <body>
      <p>Hello<br />
         You delivered your assignment ontime.<br />
      Thanks<br />
      Mahnaz</p>
  </body>

</html>
```

This will produce the following result –

```
Hello
You delivered your assignment on time.
Thanks
Mahnaz
```

Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

Example:

```
<!DOCTYPE html>
<html>

  <head>
    <title>Centring Content Example</title>
  </head>

  <body>
    <p>This text is not in the center.</p>

    <center>
      <p>This text is in the center.</p>
    </center>
  </body>

</html>
```

This will produce following result –

```
This text is not in the center.
This text is in the center.
```

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below −

Example:

```
<!DOCTYPE html>
<html>

   <head>
      <title>Horizontal Line Example</title>
   </head>

   <body>
      <p>This is paragraph one and should be on top</p>
      <hr />
      <p>This is paragraph two and should be at bottom</p>
   </body>

</html>
```

This will produce the following result −

```
This is paragraph one and should be on top
This is paragraph two and should be at bottom
```

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The **<hr />** element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

Any text between the opening **<pre>** tag and the closing **</pre>** tag will preserve the formatting of the source document.

Example:

```
<!DOCTYPE html>
<html>

   <head>
      <title>Preserve Formatting Example</title>
   </head>

   <body>
      <pre>
         function testFunction( strText ){
            alert (strText)
         }
      </pre>
   </body>

</html>
```

This will produce the following result −

```
function testFunction( strText ){
   alert (strText)
}
```

Try using the same code without keeping it inside **<pre>...</pre>** tags

Nonbreaking Spaces

Suppose you want to use the phrase “12 Angry Men.” Here, you would not want a browser to split the “12, Angry” and “Men” across two lines −

An example of this technique appears in the movie “12 Angry Men.”

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;** instead of a normal space. For example, when coding the «12 Angry Men» in a paragraph, you should use something similar to the following code −

Example:

```
<!DOCTYPE html>
<html>

   <head>
      <title>Nonbreaking Spaces Example</title>
   </head>

   <body>
      <p>An example of this technique appears in the movie "12&nbsp;Angry&nbsp;Men."</p>
   </body>

</html>
```

This will produce the following result −

An example of this technique appears in the movie “12 Angry Men.”

3.1.3 HTML Page Structure

HTML coding is structured like a tree, with each different tag nested within it. In most cases, each formatting element requires a start tag and an end tag, and different tags should not overlap.

This is what is meant by “nested;” if tag 2 opens after tag 1, then tag 2 should be closed first so that the formatting element of tag 2 is entirely enclosed within tag 1. Elements are the individual components that make up the code, and include opening and closing tags and the content between them. Attributes provide more information about the element, and are made up of the attribute and its value, connected by an equal sign.

To create an HTML element, the user creates a tag that starts and finishes with angle brackets and places it before the text that needs to be formatted. The code — usually one or more letters, numbers, words, and/or symbols — inside the angle brackets indicates what the element is and the attributes that content should have, such as its size, font, or other characteristics.

To end the formatting, the user types the first angle bracket, then a backslash, then repeats the element code and closes the bracket. For example, <title><strong>What Is HTML?</strong></title> is the code used to format the title of this article; the “strong” element tag is nested within the “title” tag.

An HTML document has two* main parts:

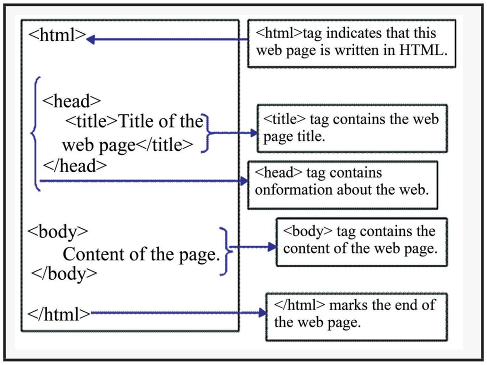
- *head*. The head element contains title and meta data of a web document.
- *body*. The body element contains the information that you want to display on a web page.

To make your web pages compatible with HTML 4, you * need to add a **document type declaration (DTD)** before the HTML element. Many web authoring software add DTD and .basic tags automatically when you create a new web page

Keyword

Document type declaration is an instruction that associates a particular SGML or XML document (for example, a webpage) with a document type definition (for example, the formal definition of a particular version of HTML1.0 - HTML 4.0).

In a web page, the first tag (specifically, <html>) indicates the markup language that is being used for the document. The <head> tag contains information about the web page. Lastly, the content appears in the <body> tag. The following illustration provides a summary.


<html>
<html>tag indicates that this web page is written in HTML.
<head>
<title>Title of the web page</title>
</head>
<title> tag contains the web page title.
<head> tag contains onformation about the web.
<body>
Content of the page.
</body>
<body> tag contains the content of the web page.
</html>
</html> marks the end of the web page.

3.1.4 HTML <!DOCTYPE> Declaration

The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.

The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.

HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document......
</body>


</html>
```

Common DOCTYPE Declarations

HTML 5

<!DOCTYPE html>

HTML 4.01 Strict

This DTD contains all HTML elements and attributes, but does NOT INCLUDE presentational or deprecated elements (like font). Framesets are not allowed.

<!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01//EN» «http://www.w3.org/TR/html4/strict.dtd»>

HTML 4.01 Transitional

This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font). Framesets are not allowed.

<!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01 Transitional//EN» «http://www.w3.org/TR/html4/loose.dtd»>

HTML 4.01 Frameset

This DTD is equal to HTML 4.01 Transitional, but allows the use of frameset content.

<!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01

Frameset//EN» «http://www.w3.org/TR/html4/frameset.dtd»>

XHTML 1.0 Strict

This DTD contains all HTML elements and attributes, but does NOT INCLUDE presentational or deprecated elements (like font). Framesets are not allowed. The markup must also be written as well-formed XML.

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Strict//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd»>

XHTML 1.0 Transitional

This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font). Framesets are not allowed. The markup must also be written as well-formed XML.

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>

XHTML 1.0 Frameset

This DTD is equal to XHTML 1.0 Transitional, but allows the use of frameset content.

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Frameset//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd»>

XHTML 1.1

This DTD is equal to XHTML 1.0 Strict, but allows you to add modules (for example to provide Ruby support for East-Asian languages).

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.1//EN» «http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd»>

3.2 HTML VERSION

HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with <HTML><HEAD>(enter here what document is about)<BODY> and ends with </BODY></HTML>. All the information you'd like to include in your Web page fits in between the <BODY> and </BODY> tags.

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup

- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

Since the early days of the web, there have been many versions of HTML:

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

3.2.1 HyperText Markup Language (1991)

The HyperText Markup Language (HTML) is a simple data format used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of domains.

As HTML is an application of SGML, this specification assumes a working knowledge of [SGML].

Scope: HTML has been in use by the **World-Wide Web (WWW)** global information initiative since 1990. Previously, informal documentation on HTML has been available from a number of sources on the Internet. This specification brings together, clarifies, and formalizes a set of features that roughly corresponds to the capabilities of HTML in common use prior to June 1994. A number of new features to HTML are being proposed and experimented in the Internet community.

This document thus defines a HTML 2.0 (to distinguish it from the previous informal specifications). Future (generally

Keyword

World Wide Web (WWW), is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.

upwardly compatible) versions of HTML with new features will be released with higher version numbers.

HTML is an application of ISO Standard 8879:1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML). The HTML Document Type Definition (DTD) is a formal definition of the HTML syntax in terms of SGML.

This specification also defines HTML as an Internet Media Type[IMEDIA] and MIME Content Type[MIME] called `text/html'. As such, it defines the semantics of the HTML syntax and how that syntax should be interpreted by user agents.

Conformance: This specification governs the syntax of HTML documents and aspects of the behavior of HTML user agents.

Documents: A document is a conforming HTML document if:

- It is a conforming SGML document, and it conforms to the HTML DTD.
- It conforms to the application conventions in this specification. For example, the value of the *HREF* attribute of the *A* element must conform to the URI syntax.
- Its document character set includes [ISO-8859-1] and agrees with [ISO-10646]; that is, each code position listed in section The HTML Coded Character Set is included, and each code position in the document character set is mapped to the same character as [ISO-10646] designates for that code position. (2)

Feature Test Entities: The HTML DTD defines a standard HTML document type and several variations, by way of feature test entities. Feature test entities are declarations in the HTML DTD that control the inclusion or exclusion of portions of the DTD.

HTML.Recommended. Certain features of the language are necessary for compatibility with widespread usage, but they may compromise the structural integrity of a document. This feature test entity selects a more prescriptive document type definition that eliminates those features. It is set to `IGNORE' by default. For example, in order to preserve the structure of a document, an editing user agent may translate HTML documents to the recommended subset, or it may require that the documents be in the recommended subset for import.

HTML.Deprecated. Certain features of the language are necessary for compatibility with earlier versions of the specification, but they tend to be used and implemented inconsistently, and their use is deprecated. This feature test entity enables a document type definition that allows these features. It is set to `INCLUDE' by default. Documents generated by translation software or editing software should not contain deprecated idioms.

User Agents: An HTML user agent conforms to this specification if:

- It parses the characters of an HTML document into data characters and markup according to [SGML]. (3)

- It supports the `ISO-8859-1' character encoding scheme and processes each character in the ISO Latin Alphabet No. 1 as specified in section The HTML Document Character Set. (4)
- It behaves identically for documents whose parsed token sequences are identical. For example, comments and the whitespace in tags disappear during tokenization, and hence they do not influence the behavior of conforming user agents.
- It allows the user to traverse (or at least attempt to traverse, resources permitting) all hyperlinks from *A* elements in an HTML document.

An HTML user agent is a level 2 user agent if, additionally:

- It allows the user to express all form field values specified in an HTML document and to (attempt to) submit the values as requests to information services.

3.2.2 HTML 2.0

HTML 2.0 was an expansion of the HTML language. Unlike the original version of HTML, HTML 2.0 was created to be a Web standard, "formalizing a set of features that roughly corresponds to the capabilities of HTML in common use".

Even though the original version of HTML was never referred to as HTML 1.0, HTML 2.0 was given the 2.0 suffix to "distinguish it from previous informal specifications.

The documentation related to HTML 2.0 can be found here through the W3C.

Information on HTML was originally only available from scattered, varying resources. With the Web growing in size, it was important to have a strict set of standards outlining exactly what HTML should be able to do. It was even more important to have this information available in one place, so that users could learn exactly how the language worked. The first official standards for HTML 2.0 were published in November of 1995. HTML 2.0 was created through the Internet Engineering Task Force (IETF), though Tim Berners-Lee still took an active role in its creation.

Did You Know?

After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based.

How was it created?

HTML 2.0 was standardized as a RFC, or "request for comment" project in a memo released by Tim Berners-Lee with the In's Network Working Group. The memo can be accessed here. It was clearly a work in progress; its status as a "request for comment" project meant that programmers could submit feedback for consideration in new updates.

HTML 2.0 was first created to standardize the features users were already utilizing HTML to perform. However, programmers wanted new features as well.

HTML 2.0 underwent several updates during 1995, prior the release of the memo that set the final standardization to refine it. If you're interested, you can view a list of the updates here.

Throughout the lifespan of the program, new features based on programmer feedback were added to the standards as RFC memos. These updates happened in November of 1995, May of 1996, August of 1996, and January of 1997.

3.2.3 HTML 3.2

HTML 3.2 was the first version of HTML to be released as a W3C Recommendation, and implemented most, but not all, of the features that were proposed in 3.0. Most notably, the mathematical equations were removed from 3.2. However, HTML 3.2 also included features that individual internet browsers were already using individually. HTML 3.2 was released on January 14th, 1997.

HTML 3.2 was the first set of standards for HTML to be released solely by the W3C. However, while the W3C codified the standards as official recommendations, the organization was not alone in developing said standards. Private companies, such as Netscape, IBM, Microsoft, Novell, among others, realized the importance of a unified standard, and took interest in the development of HTML 3.2 and contributed to the process.

Dave Raggett of Hewlett-Packard "Called together representatives of the browser companies" to form a development group for the language. The small number of people involved guaranteed that the group could make quick, efficient, educated decisions on the standards for 3.2. The representatives included "Lou Montulli from Netscape, Charlie Kindel from Microsoft, Eric Sink from Spyglass, [and] Wayne Gramlich from Sun Microsystems." Tim Berners-Lee and Dan Connolly served as representatives of the W3 Consortium. Together, the seven discussed and debated the standards, eventually coming to a consensus.

HTML as an SGML Application

HTML 3.2 is an SGML application conforming to International Standard ISO 8879 -- Standard Generalized Markup Language. As an SGML application, the syntax of conforming HTML 3.2 documents is defined by the combination of the SGML declaration and the document type definition (DTD). This specification defines the intended interpretation of HTML 3.2 elements, and places further constraints on the permitted syntax which are otherwise inexpressible in the DTD.

The SGML rules for record boundaries are tricky. In particular, a record end immediately following a start tag should be discarded. For example:

```
<P>
Text
```

is equivalent to:

```
<P>Text
```

Similarly, a record end immediately preceding an end tag should be discarded. For example:

```
Text
</P>
```

is equivalent to:

```
Text</P>
```

Except within literal text (e.g. the PRE element), HTML treats contiguous sequences of white space characters as being equivalent to a single space character (ASCII decimal 32). Note that future revisions to HTML may allow for the interpretation of the horizontal tab character (ASCII decimal 9) with respect to a tab rule defined by an associated style sheet.

SGML entities in PCDATA content or in CDATA attributes are expanded by the parser, e.g. &#233; is expanded to the ISO Latin-1 character decimal 233 (a lower case letter e with an acute accent). This could also have been written as a named character entity, e.g. &eacute;. The & character can be included in its own right using the named character entity &amp;.

HTML allows CDATA attributes to be unquoted provided the attribute value contains only letters (a to z and A to Z), digits (0 to 9), hyphens (ASCII decimal 45) or, periods (ASCII decimal 46). Attribute values can be quoted using double or single quote marks (ASCII decimal 34 and 39 respectively). Single quote marks can be included within the attribute value when the value is delimited by double quote marks, and vice versa.

Note that some user agents require attribute minimisation for the following attributes: COMPACT, ISMAP, CHECKED, NOWRAP, NOSHADE and NOHREF. These user agents don't accept syntax such as COMPACT=COMPACTor ISMAP=ISMAP although this is legitimate according to the HTML 3.2 DTD.

The SGML declaration and the DTD for use with HTML 3.2 are given in appendices. Further guidelines for parsing HTML are given in WD-html-lex.

The Structure of HTML documents

HTML 3.2 Documents start with a <!DOCTYPE> declaration followed by an HTML element containing a HEAD and then a BODY element:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 3.2 Final//EN”>
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
... other head elements
</HEAD>
<BODY>
... document body
</BODY>
</HTML>
```

In practice, the HTML, HEAD and BODY start and end tags can be omitted from the markup as these can be inferred in all cases by parsers conforming to the HTML 3.2 DTD.

Every conforming HTML 3.2 document **must** start with the <!DOCTYPE> declaration that is needed to distinguish HTML 3.2 documents from other versions of HTML. The HTML specification is not concerned with storage entities. As a result, it is not required that the document type declaration reside in the same storage entity (i.e. file). A Web site may choose to dynamically prepend HTML files with the document type declaration if it is known that all such HTML files conform to the HTML 3.2 specification.

Every HTML 3.2 document must also include the descriptive title element. A minimal HTML 3.2 document thus looks like:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 3.2 Final//EN”>
<TITLE>A study of population dynamics</TITLE>
```

The HEAD element

This contains the document head, but you can always omit both the start and end tags for HEAD. The contents of the document head is an unordered collection of the following elements:

- The TITLE element
- The STYLE element
- The SCRIPT element

- The ISINDEX element
- The BASE element
- The META element
- The LINK element

<!ENTITY % head.content “TITLE & ISINDEX? & BASE?”>

<!ENTITY % head.misc “SCRIPT|STYLE|META|LINK”>

<!ELEMENT HEAD O O (%head.content) +(%head.misc)>

The %head.misc entity is used to allow the associated elements to occur multiple times at arbitrary positions within the HEAD. The following elements can be part of the document head:

TITLE defines the document title, and is always needed.

ISINDEX for simple keyword searches, see PROMPT attribute.

BASE defines base URL for resolving relative URLs.

SCRIPT reserved for future use with scripting languages.

STYLE reserved for future use with style sheets.

META used to supply meta info as name/value pairs.

LINK used to define relationships with other documents.

TITLE, SCRIPT and STYLE are containers and require both start and end tags. The other elements are not containers so that end tags are forbidden. Note that conforming browsers won’t render the contents of SCRIPT and STYLE elements.

Remember

HTML 4.01 is the accepted standard, and the majority of web users do have browsers that support it fully. Some of the more peripheral new elements have yet to gain full support in the latest round of browsers, but they’re on the way. Modern browsers will generally have no problem with anything in these specs.

3.2.4 HTML 4.01

HTML 4.01 was a large shake-up to the HTML standards that arrived in April 1998. The HTML language you have learnt is constantly evolving to meet the needs of a growing Internet. Things get added, some things get taken away and still more elements are asked to fade out gracefully. These changes ensure that designers have the freedom and power available to create increasingly complex websites and are able to achieve this efficiently.

It only happens every few years, and the changes are made by the » World Wide Web Consortium (W3C), who are HTML’s governing body, as it were. They convene and design the specifications that we all work with when creating websites (CSS was designed by the W3C too). They look for weaknesses in HTML that are holding the web back, and sort them out, which makes creating compelling websites easier for everybody.

The standard we were all working with before this was HTML 3.2. That was used for a while before the W3C decided to step it up another notch a few years ago. They released HTML 4. Sometime later, when some minor errors in the specification were uncovered, they fixed these and called the final specification HTML 4.01.

Versions

If you have used any software you will have undoubtedly noticed how every few months it advances its number. Until they improved it and it became Firefox 2.1. Adding a decimal to the version number signifies a minor change to the original. When major changes are made to a software project, they will move up a whole number to version 3. This is the same way most dynamic things work. As you can see, the original HTML 3.0 spec was revised to version 3.2 before the big change to 4, and a minor change to 4.01.

There was some confusion when HTML 4 started being discussed, as at the time version 4 browsers like Internet Explorer 4 were making their appearance and people thought there was some connection. In reality, the two separate things had just reached those versions simultaneously, not because of each other. As you know, browser technology has advanced to version 7 stages and beyond by now, and HTML is still at level 4. So there's no real connection; though, that said, it was in the version 4 browsers that HTML 4 started being incorporated properly. Glad that's cleared up.

A few months after HTML 4.0 was released, its documentation was updated to correct some minor problems, and its version number was bumped up slightly. So the final version of this standard is HTML 4.01.

DOCTYPEs Ahoy

Nowadays the Document Type Declaration (DTD) at the top of your document is very important if you want the browsers to render your page correctly. Without it, browsers might interpret your code more loosely, and you may have display errors. The HTML 4.01 DTDs are below. Take your pick.

Use the strict DTD if you're using pure, structural code with no hacks:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

The transitional DOCTYPE, below, is the most commonly used, and still permits you to use certain old elements that we will eventually stop using altogether. It is probably the best choice until you've gotten to know HTML really well. Once you're ready, you can start using the stricter DOCTYPE above.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Finally, for **frameset** pages, use the frameset DTD:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

Simply add this line of code to the very top of your HTML pages (before the opening <html> tag), and you're away. You will also need to specify the character encoding of your page. The best encoding to use is called Unicode, and allows you to type almost any character you want (like punctuation, letters with accents etc.) directly into your content. Add this element in between your page's <head> and </head>:

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

Once you have added your DOCTYPE and encoding, run your page through the » **HTML validator** to see if you're obeying the rules.

The New Elements in HTML 4

There are 22 elements new to HTML in the 4 specifications, and they cover all the areas, from text-formatting to tables to frames and the rest. Most of the text formatting elements will make the text they mark up look a little different. You can see the effects of these elements in the text formatting list.

<abbr>

This is used to show an ABBReviated version of a word, and to offer the full version. When a reader leaves their mouse on the word, the full version pops up.

The code would be <abbr title="abbreviation">abbr.</abbr>

<acronym>

Similar to the one above, this is used for special abbreviations called acronyms (initialled abbreviations that can be spoken as a word themselves). It works the same way. <acronym title="North Atlantic Treaty Organisation">NATO</acronym>

<bdo>

Most text is read from left to right, but some languages are the opposite, like Hebrew for example. This new tag allows the browser to display your page correctly if you use one of these languages, and allows you to pull off cool effects like this: cool effects. That's typed in normally in the source code, but the tag changes its direction. RTL means 'Right To Left'. Code: <bdo dir="rtl">crazy text</bdo>

<button>

This is a simple way of adding form buttons to your pages. What's more, you can now format the text and put images and other elements on the button. <button><b>click</b> me</button> click me for nothing!

<colgroup>

A table tag that allows you to affect the attributes of an entire column with one line of code. More info in Tables Accessibility.

<del>

Wrapping this around some text creates a strike-through effect to signify DELeted text, so you can show readers what once existed without cutting it out altogether. <del>waffle</del>

<fieldset>

This allows you to group buttons and things together, giving you a framed container to hold them in. It works together with the <legend> tag below.

<frame>

Strangely, this has only been made part of the official specifications now, despite it having been in use for years. It has been given new style attributes but overall it's the same as the frame tag before.

<frameset>

This tag is in the same boat as its friend above. Nothing's really new.

<iframe>

Once a proprietary **Internet Explorer** tag, this was such a smart idea that it has been assimilated into HTML proper.

Keyword

Internet Explorer is a series of graphical web browsers developed by Microsoft and included in the Microsoft Windows line of operating systems, starting in 1995.

<ins>

This stands for INSerted text. It works in conjunction with the del tag, and the inserted text appears with an underline. <del>waffle</del><ins>quality literature</ins>

<label>

This allows you to give form elements a label. Clicking the label functions like clicking the element (a radio button, for example) itself. This improves Forms Accessibility.

```
<label for=”choice1”>Choice 1</label>
    <input type=”radio” id=”choice1”>
```

<legend>

When using a fieldset element, this element must come first before any other content inside the fieldset. It gives the title of the group.

```
    <fieldset><legend>Contact Info</legend>
    Email:<input type=”text”>
    Address:<input type=”text”></fieldset>
```

<noframes>

Another part of the frames umbrella that is being formally added to HTML 4. You can learn more about it at the basic frames page.

<noscript>

The same as above, this is for people who can’t do JavaScript.

<object>

This is set to become the do-it-all tag for inserting multimedia into your page, and is supposed to take over from img, ismap, applet, script and any others.

```
    <object data=”picture.gif” type=”image/gif”></object>
```

<optgroup>

With this tag you can group together many option elements which are part of a select field, and give the groups titles.

<param>

This tag is used to set PARAMeters for ActiveX, Applets and objects. It existed before HTML 4.01, but now is official code.

<span>

This tag was brought in specifically to work with stylesheets in applying classes and ids. It does nothing on its own, but it is great for applying your styles to text.

<tbody>

A new table tag that allows you to give attributes to a block of cells with this one tag.

<tfoot>

Allows you to add a footer to the tbody part of your tables.

<thead>

This allows you to add a header to the tbody part of a table. It comes before tbody, while tfoot comes after in the code. Both of these elements are also found in HTML4 Tables.

<q>

If you've ever used the blockquote tag, you'll know it's a big tag. How many letters are in that, ten?! This is much more like it, and is suitable for shorter quotations. Plus, it adds in the quotation marks for you. It will not add in the line breaks you get with blockquote.

The new Attributes

These new attributes are here to allow stylesheet implementation, with two more reflecting the new international concerns that the W3C have taken onboard in this new draught. They can all be applied to any element.

class

This is how you give your page elements and text their classes from your stylesheet. Read all about it in advanced CSS.

dir

This is the attribute that is used mostly with the new <bdo> tag above. Your possible values are rtl (right-to-left) or the default ltr (left-to-right).

id

ids are just like classes, but can be used with JavaScript and DHTML. More info.

lang

This attribute sets off a block of text as text typed in a foreign LANGuage, so that search engines and browsers know, and don't just take it as badly spelled English. It will not translate anything for you, it's just some behind-the-scenes help for things other than readers.

You can denote the text using the span tag, like

<span lang="fr">Bonjour!</span>.

If you're going to use it, have a look at the the common language codes.

title

This is one of my favorite things that came with HTML 4.01. It allows you to add in tooltip text, like the alt attribute; but now you can add it to absolutely anything. You can give table cells titles, add in extra information to your links, and even hide jokes in your code that will only appear when a reader is on a specific word or sentence in your text.

Deprecated Elements

A deprecated element is one that is on the way out, but one which has been given a few more months to live before its life fully ends. There are much better elements than these available now, so your usage of them should be downscaled as much as possible.

<applet>

Used to add Java applets. Use the new <object> element instead.

<basefont>

Used to affect text on the whole page. Use style sheets instead.

<center>

Used to center elements. Use <div align="center"> or stylesheets.

<dir>

Used to make lists. Use uls instead.

<font>

Ah, the classic font element. Still good for small things, but stylesheetshave taken over. This is one element you should really try to avoid using.

<isindex>

Just use the input tag.

<menu>

Another type of list that is redundant thanks to the ul element.

<s>

Creates strike-through effects. Use the stylesheets again, or the new delelement.

<strike>

Same as above, use style.

<u>

The underlining element, use stylesheets or ins instead.

Dead Elements

These are the elements that were so useless that they're out on their asses for good. Never use these, you can't guarantee the browsers will continue to support them. All three of these elements have been replaced by one new element — so you can see how they were useless.

<listing>, <plaintext>, and <xmp>. Use pre instead. This creates PRE formatted text (text which follows its layout in your code).

And that's the lot. By this stage, you are encouraged to use all of these elements. Most people have either upgraded their browsers or bought computers with the newest software on it, so the majority of the web audience are on the same level. That said, there are some elements here that still aren't supported in even the newest browsers, so test your pages out before you go using any of these elements.

3.2.5 XHTML

The extensible hypertext markup language (XHTML) is a quick way to refer to several language recommendations that are widely used on Internet-enabled devices for viewing web pages. Although named after its predecessor, the hypertext markup language (HTML), it's is actually based on the extensible markup language (XML), which is a very selective part of the **standard generalized markup language (SGML)**. In essence, they're all offspring of SGML. While HTML is a direct application of SGML, XHTML is what's referred to as a namespace, or a set of definitions for an XML document that helps to relieve ambiguity when more than one XML vocabulary is being used in any given situation.

The language came about because of a few limitations to HTML and the varied way HTML was being implemented. Around the time HTML made it to version four, it began to wane in proper usage by many HTML interpreters, the computer programs that parse HTML documents into a formatted, viewable web page. As mobile devices and other web-viewing platforms were also emerging, a better solution was needed. XML is a much more strict implementation of SGML over HTML, and different XML namespaces can be used in a single instance. So around the year 2000, the World Wide Web Consortium (W3C) drafted and made XHTML one of its recommendations to solve some of these emerging issues.

For all intents and purposes, XHTML mimics HTML in most ways, but since the former uses an XML namespace, it can be parsed by any XML interpreter, while HTML is limited to only HTML interpreters. XHTML is really HTML recreated under the more restrictive XML subset of SGML. In this way, the more recent language was immediately able to be interpreted by existing web browsers while also making itself available for other platforms. Living up to the extensible aspect of XHTML's moniker is also important to note. It not only offers the ability to be read by more programs and platforms, but it is also further extensible by allowing the use of other XML namespaces within its documents. With XHTML's ability to include other XML namespaces in a document, it can be extended in a number of ways to present more than just page formatting. The mathematical markup language (MathML), for example, can be included in these documents to display mathematical formulas and notation. Images can also be embedded using the scalable vector graphics (SVG) namespace within a document of this type. As such, XHTML can also be a included with another XML document. Since XHTML is really just HTML refined under XML's rules, it offers three document type definitions (DTD) that duplicate those of HTML version four. A DTD is a detailed description of the elements of a markup language, including when, where, and how it can be used, as well as any associated attributes. In later versions of XHTML, however, XML schemas, another, more robust way of describing an XML document, were established that further augmented XHTML. In turn, various stripped-down versions of XHTML were developed that can then be built upon for specific uses, many of which revolve around mobile computing platforms.

Convert HTML to XHTML

The eXtensible HyperText Markup Language (XHTML) has replaced the similar HyperText Markup Language (HTML) as the markup language of choice for creating webpages, but old webpages are relatively easy to convert from HTML to XHTML. First, choose the appropriate version of XHTML to convert to because different versions support different features. Then make changes to the code by hand to better understand the new language's syntax rules or use a software tool to convert pages for you. In either case, it is a good idea to check that the document was converted properly with an XHTML validator, which can detect any errors in the code.

Keep in mind that just like HTML, XHTML has several different versions, and not all of them support the same features. XHTML 1.0 Transitional allows certain presentational tags, elements, and attributes present in earlier HTML standards including the <font> tag and “center” attributes. If you’re moving from HTML to XHTML 1.0 Strict or XHTML 1.1, remove any any presentational tags or elements and recreate the desired look using Cascading Style Sheets because these versions of XHTML do not support any presentational elements.

For a simple web document, it’s possible to perform an HTML to XHTML conversion by hand. If you’re new to XHTML, this may also help you become more familiar with the stricter syntax rules of the language. To conform to these stricter requirements, make sure that all tags are in lower case, properly nested, and closed with a forward slash. In addition, all values must be inside quotation marks. The “find and replace all” option available in most text editors and web development programs can help you make these changes.

Software can also convert your code from HTML to XHTML with just a few clicks. Most professional web design programs have this option, often under the “File” or “Tools” menu. A popular tool called HTML Tidy, originally released by members of the World Wide Web Consortium (W3C), can be downloaded for free and will both clean up and convert between popular web formats. You might also be able to find a few tools online that can change documents from one markup language to another.

Whether you choose to edit your documents by hand or use software to perform the HTML to XHTML conversion, there are a few things you should check before publishing your pages to the web. To be valid XHTML, a page should have both a Document Type Definition (DOCTYPE or DTD) and an XML namespace declaration. Each of these will depend on what version of XHTML you use, so do a little research to figure out what these elements should look like. An XHTML validator, available within most web development software as well as on the W3C’s website, will inform you of any errors or omissions in your code.

XHTML Editor

HTML is HyperText Markup Language, a language derived from SGML (Standard Generalized Markup Language) and used on the World Wide Web to create structure for the presentation of documents and create links between them, using tags and a set of rules. XML is EXtensible Markup Language and simplified SGML. XHTML is EXtensible HTML, a markup language that combines features of HTML and XML, with stricter rules than those applied in HTML coding. An XHTML editor is an environment designed for authoring XHTML.

An XHTML editor is not usually found as a stand-alone product. It is quite common to find an XHTML editor incorporated in a product that provides a more comprehensive package of web development editors. For example, a software package might combine and XHTML editor with an HTML editor. A more complete package could include an HTML editor, XHTML editor, CSS (**Cascading Style Sheets**) editor, and JavaScript® editor.

XHTML editors often have both a text editor in which to input XHTML and a WYSIWYG (What You See Is What You Get) editor, and sometimes it's possible to show only the WYSIWYG window. The purpose is to allow a user who hasn't learned the technical aspects of web design, such as markup languages and CSS, to still be able to generate and edit content. However, one problem with web pages and websites being built with correct XHTML has been WYSIWYG editors that do not generate proper markup. XStandard points out the problem and makes XHTML WYSIWYG Editor™ to address this, assuring that only markup that meets best practice criteria is used.

Keyword

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. These pages contain information on how to learn and use CSS and on available software. They also contain news from the CSS working group.

One important feature you may find in an XHTML editor is the ability to choose the version of XHTML you are working on. Adobe Dreamweaver CS4, for example, offers a choice of XHTML 1.0 Transitional, XHTML 1.0 Strict, XHTML 1.1, or XHTML 1.0 Mobile. HTML 4.01 Transitional and HTML 4.01 Strict are also offered.

In any case, there are some other features that may be present in an XHTML editor. One valuable feature that is not universally offered is XHTML validation. Alternatively, dynamic correction may be offered along with an error log file. An HTML to XHTML conversion feature may also be included. Other desirable features include syntax highlighting, word wrap, and UTF-8 Unicode (8-bit UCS/Unicode Transformation Format) — UCS stands for Universal Character Set — to enable work on multilingual files.

Syntax

XHTML syntax is very similar to HTML syntax and almost all the valid HTML elements are valid in XHTML as well. But when you write an XHTML document, you need to pay a bit extra attention to make your HTML document compliant to XHTML.

Here are the important points to remember while writing a new XHTML document or converting existing HTML document into XHTML document −

- Write a DOCTYPE declaration at the start of the XHTML document.
- Write all XHTML tags and attributes in lower case only.
- Close all XHTML tags properly.
- Nest all the tags properly.
- Quote all the attribute values.
- Forbid Attribute minimization.
- Replace the name attribute with the id attribute.
- Deprecate the language attribute of the script tag.

Here is the detail explanation of the above XHTML rules –

DOCTYPE Declaration

All XHTML documents must have a DOCTYPE declaration at the start. There are three – types of DOCTYPE declarations. Here is an example of using DOCTYPE

```
"DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN!>
<"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
```

Case Sensitivity

XHTML is case sensitive markup language. All the XHTML tags and attributes need .to be written in lower case only

```
<-- This is invalid in XHTML --!>
<A Href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</A>

<-- Correct XHTML way of writing this is as follows --!>
<a href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</a>
```

.In the example, **Href** and anchor tag **A** are not in lower case, so it is incorrect

Closing the Tags

Each and every XHTML tag should have an equivalent closing tag, even empty elements should also have closing tags. Here is an example showing valid and invalid ways of using tags –

```
<!-- This is invalid in XHTML -->
<p>This paragraph is not written according to XHTML syntax.
<!-- This is also invalid in XHTML -->
<img src="/images/xhtml.gif" >
```

The following syntax shows the correct way of writing above tags in XHTML. Difference is that, here we have closed both the tags properly.

```
<!-- This is valid in XHTML -->
<p>This paragraph is not written according to XHTML syntax.</p>
<!-- This is also valid now -->
<img src=”/images/xhtml.gif” />
```

Attribute Quotes

All the values of XHTML attributes must be quoted. Otherwise, your XHTML document is assumed as an invalid document. Here is the example showing syntax –

```
<!-- This is invalid in XHTML -->
<img src=”/images/xhtml.gif” width=250 height=50 />

<!-- Correct XHTML way of writing this is as follows -->
<img src=”/images/xhtml.gif” width=”250” height=”50” />
```

Attribute Minimization

XHTML does not allow attribute minimization. It means you need to explicitly state the attribute and its value. The following example shows the difference –

```
    <!-- This is invalid in XHTML -->
<option selected>
    <!-- Correct XHTML way of writing this is as follows -->
    <option selected=”selected”>
```

Here is a list of the minimized attributes in HTML and the way you need to write them in XHTML –

HTML Style	**XHTML Style**
compact	compact=”compact”
checked	checked=”checked”
declare	declare=”declare”
readonly	readonly=”readonly”
disabled	disabled=”disabled”
selected	selected=”selected”
defer	defer=”defer”

ismap	ismap=”ismap”
nohref	nohref=”nohref”
noshade	noshade=”noshade”
nowrap	nowrap=”nowrap”
multiple	multiple=”multiple”
noresize	noresize=”noresize”

The *id* Attribute

The id attribute replaces the name attribute. Instead of using name = “name”, XHTML prefers to use id = “id”. The following example shows how –

```
<!-- This is invalid in XHTML -->
<img src=”/images/xhtml.gif” name=”xhtml_logo” />


<!-- Correct XHTML way of writing this is as follows -->
<img src=”/images/xhtml.gif” id=”xhtml_logo” />
```

The *language* Attribute

The language attribute of the script tag is deprecated. The following example shows this difference –

```
<!-- This is invalid in XHTML -->
<script language=”JavaScript” type=”text/JavaScript”>
   document.write(“Hello XHTML!”);
</script>
<!-- Correct XHTML way of writing this is as follows -->
<script type=”text/JavaScript”>
   document.write(“Hello XHTML!”);
</script>
```

Nested Tags

You must nest all the XHTML tags properly. Otherwise your document is assumed as an incorrect XHTML document. The following example shows the syntax –

```
<!-- This is invalid in XHTML -->
<b><i> This text is bold and italic</b></i>
<!-- Correct XHTML way of writing this is as follows -->
<b><i> This text is bold and italic</i></b>
```

Element Prohibitions

The following elements are not allowed to have any other element inside them. This prohibition applies to all depths of nesting. Means, it includes all the descending elements.

Element	**Prohibition**
<a>	Must not contain other <a> elements.
<pre>	Must not contain the <img>, <object>, <big>, <small>, <sub>, or <sup> elements.
<button>	Must not contain the <input>, <select>, <textarea>, <label>, <button>, <form>, <fieldset>, <iframe> or <isindex> elements.
<label>	Must not contain other <label> elements.
<form>	Must not contain other <form> elements.

A Minimal XHTML Document

The following example shows you a minimum content of an XHTML 1.0 document –

```
<?xml version=”1.0” encoding=”UTF-8”?>

<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//EN”
“http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd”>

<html xmlns=”http://www.w3.org/TR/xhtml1” xml:lang=”en” lang=”en”>
  <head>
    <title>Every document must have a title</title>
  </head>

  <body>
    ...your content goes here...
  </body>
</html>
```

3.2.6 HTML5

Hypertext Markup Language revision 5 (HTML5) is markup language for the structure and presentation of World Wide Web contents. HTML5 supports the traditional HTML and XHTML-style syntax and other new features in its markup, New APIs, XHTML and error handling.

There are three organizations that are currently in charge of the specification of HTML5:

Web Hypertext Application Technology Working Group (WHATWG) created the HTML5 specification and is in charge of the HTML5 development that provides open collaboration of browser vendors and other involved parties.

World Wide Web Consortium (W3C) is in charge with delivering the HTML5 specification.

Internet Engineering Task Force (IETF) is in charge of the development of HTML5 WebSocket API.

HTML5 is an effort is to bring order to web development chaos by organizing common practices, embracing implementations from various browsers. It is massive, with over 100 specifications as part of the HTML5 specs. Understanding this, you can simplify by thinking of HTML5 this way. HTML5 is simply just an umbrella term for the next generation of web apps an how functionality will be expanded with better markup (HTML), better style (CSS), and better interactivity (JavaScript).

The specification of HTML5 that has been published currently is not yet final. HTML5 is expected due for Candidate Recommendation (CR) by 2012, and is expected for Proposed Recommendation (PR) by 2022. However, this doesn't mean that HTML5 is not ready for use. The proposed recommendation does mean however that there will be two interoperable implementations. As of 2011, browser vendors are actively adding support for new features of HTML5.

New features of HTML5 include:

- New parsing rules that are not based on SGML but are oriented towards flexible parsing and compatibility.
- Support of use of inline Scalar Vector Graphics (SVG) and Mathematical Markup Language (MathML) in text/html.
- New available elements include article, aside, audio, bdi, canvas, command, datalist, details, embed, figcaption, figure, footer, header, hgroup, keygen, mark, meter, nav, output, progress, rp, rt, ruby, section, source, summary, time, video and wbr.

- New available types of form controls include dates and times, email, url, search, number, range, tel and color.
- New available attributes of charset on meta and async on script.
- Global attributes that can be applied for every element that include id, tabindex, hidden, data-* or customer data attributes.

SUMMARY

- HyperText Markup Language (HTML) is a type of computer language that is used to create pages that can be posted on the Internet or sent via email.
- HTML is a Markup Language which means you use HTML to simply «mark-up» a text document with tags that tell a Web browser how to structure it to display.
- The tag tells the browser that this is an HTML document. The tag represents the root of an HTML document. The tag is the container for all other HTML elements (except for the tag).
- Horizontal lines are used to visually break-up sections of a document. The <hr> tag creates a line from the current position in the document to the right margin and breaks the line accordingly.
- The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.
- HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with <HTML><HEAD>(enter here what document is about)<BODY> and ends with </BODY></HTML>.
- HTML 3.2 was the first version of HTML to be released as a W3C Recommendation, and implemented most, but not all, of the features that were proposed in 3.0. Most notably, the mathematical equations were removed from 3.2.
- HTML 4.01 was a large shake-up to the HTML standards that arrived in April 1998. The HTML language you have learnt is constantly evolving to meet the needs of a growing Internet.
- The extensible hypertext markup language (XHTML) is a quick way to refer to several language recommendations that are widely used on Internet-enabled devices for viewing web pages.
- The eXtensible HyperText Markup Language (XHTML) has replaced the similar HyperText Markup Language (HTML) as the markup language of choice for creating webpages, but old webpages are relatively easy to convert from HTML to XHTML.
- Hypertext Markup Language revision 5 (HTML5) is markup language for the structure and presentation of World Wide Web contents. HTML5 supports the traditional HTML and XHTML-style syntax and other new features in its markup, New APIs, XHTML and error handling.

KNOWLEDGE CHECK

1. **Recommendation of using attributes in HTML is to use it in**
   a. Uppercase
   b. Lower case
   c. Sentence case
   d. Proper case
2. **Language of document can be specified with tag**
   a. lang
   b. Ref
   c. language
   d. Href
3. **Style attribute of an element defines**
   a. Size
   b. Color
   c. Font type
   d. All of the Above
4. **In HTML, language of document is declared in tag**
   a. <!DOCTYPE html>
   b. <html>
   c. <body>
   d. <ref>
5. **The**
   a. Image is displayed
   b. Image is not displayed
   c. Image is shared
   d. Image is not shared
6. **HTML stands for -**
   a. HighText Machine Language
   b. HyperText and links Markup Language
   c. HyperText Markup Language
   d. None of these

7. **The correct sequence of HTML tags for starting a webpage is -**

a. Head, Title, HTML, body

b. HTML, Body, Title, Head

c. HTML, Head, Title, Body

d. HTML, Head, Title, Body

8. **HTML is a subset of ....**

a. SGMD

b. SGML

c. SGMH

d. None of the above

9. **Which of the following element is responsible for making the text bold in HTML?**

a. <pre>

b. <a>

c. <b>

d. <br>

10. **Which character is used to represent the closing of a tag in HTML?**

a. \

b. !

c. /

d. .

REVIEW QUESTIONS

1. What is the basics of HTML document?
2. Discuss about the various tags of HTML.
3. Explain the HTML page structure.
4. Define the HTML <!DOCTYPE> Declaration.
5. Write the features of HTML 2.0.
6. What are the different between HTML 3.2 and HTML 4.01?

Check Your Result

1. (b)	2. (a)	3. (d)	4. (b)	5. (b)
6. (c)	7. (d)	8. (b)	9. (c)	10. (c)

REFERENCES

1. Castro, Elizabeth and Hyslop . HTML5, and CSS, Eight Edition: (Visual QuickStart Guide), Peachpit Press, 2013.
2. Devlin, Ian. HTML5 Multimedia: Develop and Design, Peachpit Press, 2011.
3. Felke-Morris, Web Development & Design Foundations with HTML5, 10th Edition, Addison-Wesley, 2020.
4. Felke-Morris. Basics of Web Design: HTML5 & CSS3, 5th Edition, Pearson Education, 2019.
5. Keith, Jeremy and Andrew, Rachel. HTML5 For Web Designers , A Book Apart, 2016.
6. Lawson, Bruce and Sharp, Remy. Introducing HTML5, New Riders Press, 2010.
7. Meiert, Jens Oliver. The Little Book of HTML/CSS Coding Guidelines, O'Reilly, 2015.
8. Meiert, Jens Oliver. Upgrade Your HTML (the Booklet), Amazon Digital Services LLC, 2019.
9. Pfeiffer, Silvia, and Green, Tom. Beginning HTML5 Media: Make the most of the new video and audio standards for the Web, APress, 2015.
10. Pfeiffer, Silvia. The Definitive Guide to HTML5 Video, APress, 2010.

CHAPTER

4

ELEMENTS OF HTML

"If there's one thing you learn by working on a lot of different Web sites, it's that almost any design idea--no matter how appallingly bad--can be made usable in the right circumstances, with enough effort."

– Steve Krug

LEARNING OBJECTIVES

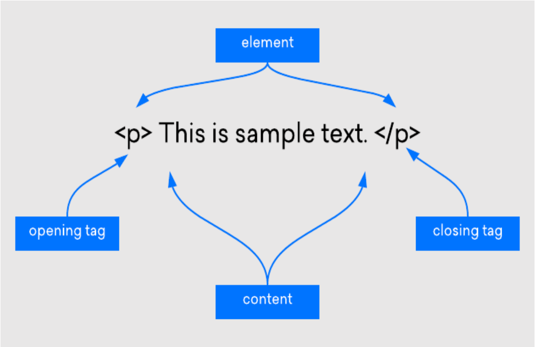
After studying this chapter, you will be able to:

1. Discuss the elements of html
2. Understand the meta tag
3. Understand the div tag
4. Explain the working with text
5. Define the working with lists, tables
6. Define the working with hyperlinks, images
7. Explain the working with forms

INTRODUCTION

An HTML element is a component of an HTML document that tells a web browser how to structure and interpret a part of the HTML document. HTML elements can contain formatting instructions, semantic meaning, and content. For example, HTML elements are used to denote document parts such as headers, paragraphs, and footers and to embed content such such as hyperlinks, text, and images. Although HTML can be used to provide formatting instructions, HTML standards strongly encourage using CSS for this purpose instead.

An HTML element is often — but not always — created by opening and closing HTML tags, which wrap around a piece of content. Below is an illustration that labels each of the parts of an HTML element:



4.1 ELEMENTS OF HTML

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

<tagname>Content goes here...</tagname>

:The HTML **element** is everything from the start tag to the end tag

<p>My first paragraph. </p>

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>		

HTML elements with no content are called empty elements. Empty elements do not .(have an end tag, such as the <br> element (which indicates a line break

4.1.1 NESTED HTML ELEMENTS

HTML elements can be nested (elements can contain elements).

.All HTML documents consist of nested HTML elements

:This example contains four HTML elements

Example

```
<! DOCTYPE html>
    <html>
    <body>
    <h1>My First Heading</h1>
    <p>My first paragraph. </p>
    </body>
    </html>
```

Example Explained

The <html> element defines the whole document.

It has a start tag <html> and an end tag </html>.

The element content is another HTML element (the <body> element).

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph. </p>
</body>
</html>
```

The <body> element defines the document body.

It has a start tag <body> and an end tag </body>.

The element content is two other HTML elements (<h1> and <p>).

```
<body>
<h1>My First Heading</h1>
<p>My first paragraph. </p>
</body>
```

The <h1> element defines a heading.

It has a start tag <h1> and an end tag </h1>.

The element content is: My First Heading.

<h1>My First Heading</h1>

The <p> element defines a paragraph.

It has a start tag <p> and an end tag </p>.

The element content is: My first paragraph.

<p>My first paragraph. </p>

4.1.2 Do Not Forget the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

Example

```
<html>
<body>

<p>This is a paragraph
<p>This is a paragraph

</body>
</html>
```

The example above works in all browsers, because the closing tag is considered optional.

Never rely on this. It might produce unexpected results and/or errors if you forget the end tag.

4.1.3 Empty HTML Elements

HTML elements with no content are called empty elements.

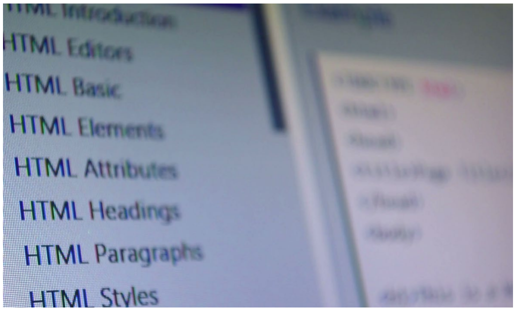
<br> is an empty element without a closing tag (the <br> tag defines a line break).

Empty elements can be "closed" in the opening tag like this: <br />.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

4.1.4 Use Lowercase Tags

HTML tags are not case sensitive: <P> means the same as <p>.



4.2 META TAG

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

4.2.1 Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes −

Sr.No	Attribute & Description
1	**Name** Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.

2	**content** Specifies the property's value.
3	**scheme** Specifies a scheme to interpret the property's value (as declared in the content attribute).
4	**http-equiv** Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

4.2.2 Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, and Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>

   <head>
      <title>Meta Tags Example</title>
      <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
   </head>

   <body>
      <p>Hello HTML5!</p>
   </body>

</html>
```

This will produce the following result –

```
Hello HTML5
```

4.2.3 Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

Example

```
 Live Demo
<!DOCTYPE html>
<html>

  <head>
    <title>Meta Tags Example</title>
    <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
    <meta name = “description” content = “Learning about Meta Tags.” />
  </head>
  <body>
    <p>Hello HTML5!</p>
  </body>

</html>
```

4.2.4 Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

Example

```
<! DOCTYPE html>
<html>
  <head>
    <title>Meta Tags Example</title>
    <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
    <meta name = “description” content = “Learning about Meta Tags.” />
    <meta name = “revised” content = “Tutorialspoint, 3/7/2014” />
  </head>
  <body>
    <p>Hello HTML5! </p>
  </body>
</html>
```

4.2.5 Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

Example

If you want your page keep refreshing after every 5 seconds, then use the following syntax.

```
<! DOCTYPE html>
<html>

  <head>
    <title>Meta Tags Example</title>
    <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
    <meta name = “description” content = “Learning about Meta Tags.” />
    <meta name = “revised” content = “Tutorialspoint, 3/7/2014” />
    <meta http-equiv = “refresh” content = “5” />
  </head>
  <body>
    <p>Hello HTML5!</p>
  </body>
</html>
```

4.2.6 Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

```
<! DOCTYPE html>
<html>
   <head>
      <title>Meta Tags Example</title>
      <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
      <meta name = “description” content = “Learning about Meta Tags.” />
      <meta name = “revised” content = “Tutorialspoint, 3/7/2014” />
       <meta http-equiv = “refresh” content = “5; url = http://www.tutorialspoint.com” />
   </head>
   <body>
      <p>Hello HTML5!</p>
   </body>
</html>
```

4.2.7 Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

```
<! DOCTYPE html>
<html>
   <head>
      <title>Meta Tags Example</title>
      <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
      <meta name = “description” content = “Learning about Meta Tags.” />
      <meta name = “revised” content = “Tutorials point, 3/7/2014” />
      <meta http-equiv = “cookie” content = “userid = xyz;
        expires = Wednesday, 08-Aug-15 23:59:59 GMT;” />

    </head>
    <body>
      <p>Hello HTML5!</p>
    </body>
  </html>
```

If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

4.2.8 Setting Author Name

You can set an author name in a web page using meta tag. See an example below –

Example

```
  <! DOCTYPE html>
  <html>

    <head>
      <title>Meta Tags Example</title>
      <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
      <meta name = “description” content = “Learning about Meta Tags.” />
      <meta name = “author” content = “Mahnaz Mohtashim” />
    </head>
    <body>
      <p>Hello HTML5!</p>
    </body>
  </html>
```

4.2.9 Specify Character Set

You can use <meta> tag to specify character set used within the webpage.

Example

By default, Web servers and Web browsers use ISO-8859-1 (Latin1) encoding to process Web pages. Following is an example to set UTF-8 encoding –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Meta Tags Example</title>
    <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
    <meta name = “description” content = “Learning about Meta Tags.” />
    <meta name = “author” content = “Mahnaz Mohtashim” />
    <meta http-equiv = “Content-Type” content = “text/html; charset = UTF-8” />
  </head>
  <body>
    <p>Hello HTML5!</p>
  </body>
</html>
```

To serve the static page with traditional Chinese characters, the webpage must contain a <meta> tag to set Big5 encoding –

```
<!DOCTYPE html>
<html>
  <head>
    <title>Meta Tags Example</title>
    <meta name = “keywords” content = “HTML, Meta Tags, Metadata” />
    <meta name = “description” content = “Learning about Meta Tags.” />
    <meta name = “author” content = “Mahnaz Mohtashim” />
    <meta http-equiv = “Content-Type” content = “text/html; charset = Big5” />
  </head>
  <body>
    <p>Hello HTML5!</p>
  </body>
</html>
```

4.3 DIV TAG

The div tag is known as Division tag. The Div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar etc). Div tag has both open(<) and closing (>) tag and it is mandatory to close the tag. The Div is the most usable tag in web development because it helps us to separate out data in the web page and we can create a particular section for particular data or function in the web pages.

- Div tag is Block level tag
- It is a generic container tag
- It is used to the group of various tags of HTML so that sections can be created and style can be Applied on them.

As we know Div tag is block level tag in this example div tag contain entire width. It will be displayed div tag each time on a new line, not on the same line.

```
<html>
   <head>
      <title>gfg</title>
<style type=text/css>

p{
  background-color:gray;
  margin: 10px;
}

div
{
  color: white;
  background-color: 009900;
  margin: 2px;
  font-size: 25px;
}
</style>

</head>
```

```
 <body>
   <div > div tag    </div>
   <div > div tag    </div>
   <div > div tag    </div>
   <div > div tag    </div>

   </body>
</html>
```

Output:

```
div tag
div tag
div tag
div tag
```

As we know div tag is used for grouping HTML elements together and is to apply CSS and web layout on them Lets see below example without using div tag. we need to applying CSS for each tag (in the example using H1 H2 and two paragraphs p tags)

```
<html>
   <head>
      <title>gfg</title>
      <style type=text/css>
         p{
                color: white;
                background-color: 009900;
                width: 400px;
                }
                h1
                {
                color: white;
                background-color: 009900;
                width: 400px;
                }
                h2
                {
                color: white;
                background-color: 009900;
                width: 400px;
                }
            </style>
        </head>
        <body>
            <h1>GeeksforGeeks</h1>
              <p>How many times were you frustrated while
looking out
              for a good collection of programming/algorithm/
interview
             questions? What did you expect and what did you
get?
            This portal has been created to provide well written,
           well thought and well-explained solutions for selected
questions.
            </p>
            <h2>GeeksforGeeks</h2>
          <p>GCET is an entrance test for the extensive classroom
programme
```

Keyword

CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

```
                  by GeeksforGeeks to build and enhance Data
Structures and Algorithm
                 concepts, mentored by Sandeep Jain (Founder &
CEO, GeeksforGeeks).
                He has 7 years of teaching experience and 6 years
of industry experience.
              </p>
          </body>
      </html>
```

Many dynamic websites generate webpages on-the-fly, using a server-side scripting language like PHP or ASP. However, even dynamic pages must be formatted using HTML. Therefore, scripting languages often generate the HTML that is sent to your web browser.

4.3.1 Creating Web Layout using Div Tag

Div tag is a container tag inside div tag we can put more than one HTML element and can group together and can apply CSS for them.

div tag can be used for creating layout of web pages in below examples shows creating a web layout

we can also create web layout using tables tag but table tags are very complex to modify layout

Div tag is very flexible to creating web **layouts** and easy to modify in below example will show grouping of HTML element using div tag and create block-wise web layout.

Keyword

Layout is the process of calculating the position of objects in space subject to various constraints.

```
      <html>
          <head>
              <title>gfg</title>
              <style type=text/css>
                  .leftdiv
                  {
                  float: left;
                  }
```

```
    .middlediv
    {
    float: left;
    background-color:gray
    }
    .rightdiv
    {
    float: left;
    }
    div{
    padding : 1%;
    color: white;
    background-color: 009900;
    width: 30%;
    border: solid black;
    }
  </style>
</head>
<body>
  <div class=”leftdiv”>
    <h1>GeeksforGeeks</h1>
    <p>How many times were you frustrated while looking out
   for a good collection of programming/algorithm/interview
   questions? What did you expect and what did you get?
   This portal has been created to provide well written,
   well thought and well-explained solutions for selected questions.
  </p>
    <h2>GeeksforGeeks</h2>
    <p>GCET is an entrance test for the extensive classroom programme
   by GeeksforGeeks to build and enhance Data Structures and Algorithm
   concepts, mentored by Sandeep Jain (Founder & CEO, GeeksforGeeks).
   He has 7 years of teaching experience and 6 years of industry experience.
  </p>
```

```
    </div>
    <div class=”middlediv”>
      <h1>GeeksforGeeks</h1>
      <p>How many times were you frustrated while looking out
     for a good collection of programming/algorithm/interview
     questions? What did you expect and what did you get?
     This portal has been created to provide well written,
     well thought and well-explained solutions for selected questions.
    </p>
      <h2>GeeksforGeeks</h2>
      <p>GCET is an entrance test for the extensive classroom programme
     by GeeksforGeeks to build and enhance Data Structures and Algorithm
     concepts, mentored by Sandeep Jain (Founder & CEO, GeeksforGeeks).
     He has 7 years of teaching experience and 6 years of industry experience.
    </p>
    </div>
    <div class=”rightdiv”>
      <h1>GeeksforGeeks</h1>
      <p>How many times were you frustrated while looking out
     for a good collection of programming/algorithm/interview
     questions? What did you expect and what did you get?
     This portal has been created to provide well written,
     well thought and well-explained solutions for selected questions.
    </p>
      <h2>GeeksforGeeks</h2>
      <p>GCET is an entrance test for the extensive classroom programme
     by GeeksforGeeks to build and enhance Data Structures and Algorithm
     concepts, mentored by Sandeep Jain (Founder & CEO, GeeksforGeeks).
     He has 7 years of teaching experience and 6 years of industry experience.
    </p>
    </div>
  </body>
</html>
```

Using class

We can use Class on that particular div either in internal CSS or external CSS

In case of internal CSS: we need to define Class in the <head> section of HTML within <style> element.

In case of External CSS: we need to create a separate .css file and include it in HTML code in <head> section using <link> element.

The class name should be different from other class name in other div otherwise the CSS used in one div can affect another division.

```
<html>
   <head>
      <link rel=”stylesheet” href=”color.css”>
      <title>
         gfg
      </title>
   </head>
   <body>
      <center>
         <div class=”color”>
            <!--open tag of Div!-->
            <caption>
               <h1>GEEKSFORGEEKS</h1>
            </caption>
            <h1>Inline CSS is not USED in THIS method.
            </h1>
         </div>
         <!--closing tag of Div!-->
      </center>
   </body>
</html>
```

CSS for color class: File name color.css

```
 .color
```

```
{
height:400px;
width:600px;
border:1px solid;
background-color: 009900;
}
```

Inline CSS

we can directly use CSS in div also this method does not require of CLASS. Div in HTML coding is used as a container tag also because it is the one that can contain all other tags.

```
<html>
     <head>
       <title>
         gfg
       </title>
     </head>
     <body>
       <center>
         <div style=”height:300px; width:500px; color:white;
           border:1px solid; background-color: 009900;”>
           <!--open tag of Div!-->
           <caption>
             <h1>GEEKSFORGEEKS</h1>
           </caption>
           <h1>Inline CSS is USED in THIS method.
             In this div no class is used.
           </h1>
         </div>
         <!--closing tag of Div!-->
       </center>
     </body>
   </html>
```

4.4 WORKING WITH TEXT

To display text, or anything for that matter, in your HTML document, it must be between the <BODY> and </BODY> tags.

Look at this example again:

```
<HTML>
<HEAD>
<TITLE>Howdy World</TITLE>
</HEAD>
<BODY>
<CENTER>HOWDY WORLD! </CENTER>
</BODY>
</HTML>
```

- First, the line, <TITLE>Howdy World</TITLE> displays the words, “Howdy World” on the title bar (upper-left corner) of your browser window.
- Second, the lines <BODY> ... <CENTER>HOWDY WORLD! </CENTER> ... </BODY> display “HOWDY WORLD!” in your browser window. Note that the tags,<CENTER> ... </CENTER> center the words, “HOWDY WORLD!”

Now, lets look at HTML codes for changing text.

Keyword

Web page is a document that is suitable for the World Wide Web and web browsers.

4.4.1 Text Styles

Look at this code:

```
<BODY>
Hello! Welcome to my personal web page. I hope you like it.
I am a High School Senior, who has bit off more than he can chew. I live near a little town called Benoit in the state of Wisconsin, land of cheese!
This is my first attempt at a web page. Enjoy!
</BODY>
```

Here is what the above code looks like:

Hello! Welcome to my personal **web page**. I hope you like it. I am a High School Senior, who has bit off more than he can chew. I live near a little town called Benoit in the state of Wisconsin, land of cheese! This is my first attempt at a web page. Enjoy!

Things didn’t work out quite right! To make the browser end lines and paragraphs correctly, you must put in special markers. Unlike the tags so far, these represent single actions that don’t have a “beginning” or “end”. They do not come in pairs. To simply end one line and jump to the next, use a line break, or <BR>. To end a line and also show a blank line before beginning anything else, use a paragraph marker <P>. To fix things up, you would edit the above text like this:

```
<BODY>
Hello! Welcome to my personal web page. I hope you like it.<P>
I am a High School Senior, who has bit off more than<BR>
he can chew. I live near a little town called Benoit in the state<BR>
of Wisconsin, land of cheese!
<P>
This is my first attempt at a web page. Enjoy!
</BODY>
```

Here is what that above code looks like:

Hello! Welcome to my personal web page. I hope you like it.

I am a High School Senior, who has bit off more than he can chew. I live near a little town called Benoit in the state of Wisconsin, land of cheese!

This is my first attempt at a web page. Enjoy!

Much better! Now let’s look at some different things that can be done with text.

4.4.2 Physical Text Styles

First, there are style tags. The most basic are for physical styles, such as boldface type. I only know three, all of which are demonstrated below. These have a beginning and end; you must say where the bold type, for example, begins and where it ends.

```
<B>Boldface type looks like this on your machine</B> <BR>
<I>Italics appear like this.</I> <BR>
<TT>Typewriter text uses a fixed-width font.</TT> <BR>
<B> <I>The styles may be combined as well.</I> </B> <BR>
```

Here is what the above code looks like:

Boldface type looks like this on your machine

Italics appear like this.

Typewriter text uses a fixed-width font

.The styles may be combined as well

4.4.3 Logical Text Styles and Headings

HTML also includes logical styles. These are more general descriptions that may be interpreted by each browser differently. They make your document easier to follow and edit. They will also allow your page to conform to local custom; if a journal citation (for example) is typically bold in one country and italicized in another, the use of the logical style <CITE> will cause it to appear correctly in each place. This flexibility also allows users to have text displayed according to their preferences; they could for example have text marked <STRONG> displayed in a different color rather than in boldface.

Take a look at this code and the example below:

```
<DFN>HTML is the language used for www pages.</DFN>
<CITE>Testing 1 2 3.</CITE>
<CODE>printf("The value of n is %d\n",n)</CODE>
<SAMP>(Abort, Retry, Fail)?</SAMP>
<VAR>username</VAR>
<EM>Is this thing on?</EM>
<KBD>username, host</KBD>
<STRONG>You cannot be serious!</STRONG>
<H1>Level 1 Heading</H1>
<H2>Level 2 Heading</H2>
<H3>Level 3 Heading</H3>
<H4>Level 4 Heading</H4>
<H5>Level 5 Heading</H5>
<H6>Level 6 Heading</H6>
```

Here is what the above code looks like:

HTML is the language used for www pages.

Testing 1 2 3

printf(“The value of n is %d\n”,n)

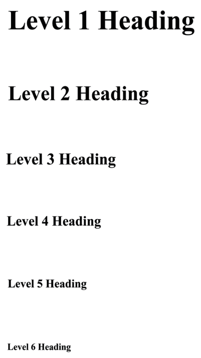
(Abort, Retry, Fail)?

username

Is this thing on?

username, host

You cannot be serious!



Level 1 Heading

Level 2 Heading

Level 3 Heading

Level 4 Heading

Level 5 Heading

Level 6 Heading

4.5 WORKING WITH LISTS, TABLES

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

- **<ul>** – An unordered list. This will list items using plain bullets.
- **<ol>** – An ordered list. This will use different schemes of numbers to list your items.
- **<dl>** – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

Example

```
<!DOCTYPE html>
    <html>
      <head>
        <title>HTML Unordered List</title>
      </head>
        <body>
        <ul>
          <li>Beetroot</li>
          <li>Ginger</li>
          <li>Potato</li>
          <li>Radish</li>
        </ul>
      </body>

    </html>
```

This will produce the following result –

The type Attribute

You can use **type** attribute for <ul> tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options –

```
<ul type = “square”>
<ul type = “disc”>
<ul type = “circle”>
```

Example

Following is an example where we used <ul type = “square”>

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
    <ul type = “square”>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
```

```
   </body>
</html>
```

This will produce the following result –

Example

Following is an example where we used <ul type = “disc”> –

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Unordered List</title>
   </head>

   <body>
      <ul type = “disc”>
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
      </ul>
   </body>

</html>
```

This will produce the following result –

Example

Following is an example where we used <ul type = “circle”> –

```
<!DOCTYPE html>
<html>
   <head>
      <title>HTML Unordered List</title>
   </head>
   <body>
```

```
      <ul type = “circle”>
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
      </ul>
    </body>
  </html>
```

This will produce the following result –

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

Example

```
<!DOCTYPE html>
  <html>

    <head>
      <title>HTML Ordered List</title>
    </head>

    <body>
      <ol>
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
      </ol>
    </body>

  </html>
```

This will produce the following result –

The type Attribute

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options –

<ol type = “1”> - Default-Case Numerals.

<ol type = “I”> - Upper-Case Numerals.

<ol type = “i”> - Lower-Case Numerals.

<ol type = “A”> - Upper-Case Letters.

<ol type = “a”> - Lower-Case Letters.

Example

Following is an example where we used <ol type = “1”>

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Ordered List</title>
    </head>
    <body>
        <ol type = “1”>
            <li>Beetroot</li>
            <li>Ginger</li>
            <li>Potato</li>
            <li>Radish</li>
        </ol>
    </body>

</html>
```

This will produce the following result –

Did You Know?

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. [6]

Example

Following is an example where we used <ol type = “I”>

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Ordered List</title>
   </head>
   <body>
      <ol type = “I”>
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
      </ol>
   </body>
</html>
```

This will produce the following result –

Example

Following is an example where we used <ol type = “i”>

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Ordered List</title>
   </head>
   <body>
      <ol type = “i”>
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
```

```
    </ol>
  </body>
</html>
```

This will produce the following result –

Example

Following is an example where we used <ol type = “A” >

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Ordered List</title>
  </head>
  <body>
    <ol type = “A”>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>
</html>
```

This will produce the following result –

Example

Following is an example where we used <ol type = “a”>

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>
  <body>
    <ol type = “a”>
```

```
            <li>Beetroot</li>
            <li>Ginger</li>
            <li>Potato</li>
            <li>Radish</li>
        </ol>
    </body>
</html>
```

This will produce the following result –

The start Attribute

You can use **start** attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options –

```
<ol type = "1" start = "4">    - Numerals starts with 4.
<ol type = "I" start = "4">    - Numerals starts with IV.
<ol type = "i" start = "4">    - Numerals starts with iv.
<ol type = "a" start = "4">    - Letters starts with d.
<ol type = "A" start = "4">    - Letters starts with D.
```

Example

Following is an example where we used <ol type = "i" start = "4" >

```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML Ordered List</title>
    </head>
    <body>
        <ol type = "i" start = "4">
            <li>Beetroot</li>
            <li>Ginger</li>
            <li>Potato</li>
            <li>Radish</li>
        </ol>
    </body>
    </html>
```

This will produce the following result –

HTML Definition Lists

HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> – Defines the start of the list
- <dt> – A term
- <dd> – Term definition
- </dl> – Defines the end of the list

Example

```
<!DOCTYPE html>
    <html>
      <head>
        <title>HTML Definition List</title>
      </head>
      <body>
        <dl>
          <dt><b>HTML</b></dt>
          <dd>This stands for Hyper Text Markup Language</dd>
          <dt><b>HTTP</b></dt>
          <dd>This stands for Hyper Text Transfer Protocol</dd>
        </dl>
      </body>
    </html>
```

This will produce the following result –

```
HTML
This stands for Hyper Text Markup Language
HTTP
This stands for Hyper Text Transfer Protocol
```

4.5.1 Table

An HTML table is defined with the <table> tag.

Each table row is defined with the <tr> tag. A table **header** is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

Keyword

Header refers to supplemental data placed at the beginning of a block of data being stored or transmitted.

Example

```
<table style=”width:100%”>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
    </tr>
    <tr>
      <td>Eve</td>
      <td>Jackson</td>
      <td>94</td>
    </tr>
  </table>
```

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS border property:

Example

```
table, th, td {
  border: 1px solid black;
  }
```

HTML Table - Collapsed Borders

If you want the borders to collapse into one border, add the CSS border-collapse property:

Example

```
table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
  }
```

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS padding property:

Example

```
  th, td {
      padding: 15px;
  }
```

HTML Table - Left-align Headings

By default, table headings are bold and centered.

To left-align the table headings, use the CSS text-align property:

Example

```
  th {
      text-align: left;
  }
```

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the CSS border-spacing property:

Example

```
table {
    border-spacing: 5px;
}
```

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the colspan attribute:

Example

```
<table style=”width:100%”>
  <tr>
    <th>Name</th>
    <th colspan=”2”>Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the rowspan attribute:

Example

```
<table style=”width:100%”>
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan=”2”>Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Adding a Caption

To add a caption to a table, use the <caption> tag:

Example

```
<table style=”width:100%”>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

A Special Style for One Table

To define a special style for a special table, add an id **attribute** to the table:

Example

```
<table id=”t01”>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Now you can define a special style for this table:

```
table#t01 {
    width: 100%;
    background-color: #f1f1c1;
}
```

And add more styles:

```
table#t01 tr:nth-child(even) {
    background-color: #eee;
}
table#t01 tr:nth-child(odd) {
    background-color: #fff;
}
table#t01 th {
    color: white;
    background-color: black;
}
```

Keyword

Attribute is a specification that defines a property of an object, element, or file

4.6WORKINGWITHHYPERLINKS,IMAGES

When you move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

In HTML, links are defined with the <a> tag:

<a href=”url”>link text</a>

Example

<a href=”https://www.w3schools.com/html/”>Visit our HTML tutorial</a>

The href attribute specifies the destination address (https://www.w3schools.com/html/) of the link.

The link text is the visible part (Visit our HTML tutorial).

Clicking on the link text will send you to the specified address.

Remember

Without a forward slash at the end of subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to the end of the address, and then create a new request.

Local Links

The example above used an absolute URL (a full web address).

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example

<a href=”html_images.asp”>HTML Images</a>

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using CSS:

Example

<style>

```
a:link {
    color: green;
    background-color: transparent;
    text-decoration: none;
}
a:visited {
    color: pink;
    background-color: transparent;
    text-decoration: none;
}

a:hover {
    color: red;
    background-color: transparent;
    text-decoration: underline;
}

a:active {
    color: yellow;
    background-color: transparent;
    text-decoration: underline;
}
</style>
```

HTML Links - The target attribute

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab as it was clicked (this is default)
- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- *framename* - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href=”https://www.w3schools.com/” target=”_blank”>Visit W3Schools!</a>
```

Tip: If your webpage is locked in a frame, you can use target=”_top” to break out of the frame:

Example

```
<a href=”https://www.w3schools.com/html/” target=”_top”>HTML5 tutorial!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href=”default.asp”>
  <img src=”smiley.gif” alt=”HTML tutorial” style=”width:42px;height:42px;border:0;”>
</a>
```

Note: border:0; is added to prevent IE9 (and earlier) from displaying a border around the image (when the image is a link).

Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

```
<a href=”https://www.w3schools.com/html/” title=”Go to W3Schools HTML section”>Visit our HTML Tutorial</a>
```

HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.

Example

First, create a bookmark with the id attribute:

<h2 id="C4">Chapter 4</h2>

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

<a href="#C4">Jump to Chapter 4</a>

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

Example

<a href="html_demo.html#C4">Jump to Chapter 4</a>

External Paths

External pages can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a web page:

Example

<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>

This example links to a page located in the html folder on the current web site:

Example

<a href="/html/default.asp">HTML tutorial</a>

This example links to a page located in the same folder as the current page:

Example

<a href="default.asp">HTML tutorial</a>

4.6.1 Images

Images can improve the design and the appearance of a web page.

Example

<img src="pic_trulli.jpg" alt="Italian Trulli">

Example

<img src=”img_girl.jpg” alt=”Girl in a jacket”>

Example

<img src=”img_chania.jpg” alt=”Flowers in Chania”>

HTML Images Syntax

In HTML, images are defined with the <img> tag.

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The src attribute specifies the URL (web address) of the image:

<img src=”url”>

The alt Attribute

The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

Example

<img src=”img_chania.jpg” alt=”Flowers in Chania”>

If a browser cannot find an image, it will display the value of the alt attribute:

Example

<img src=”wrongname.gif” alt=”Flowers in Chania”>

Note: The alt attribute is required. A web page will not validate correctly without it.

Image Size - Width and Height

You can use the style attribute to specify the width and height of an image.

Example

<img src=”img_girl.jpg” alt=”Girl in a jacket” style=”width:500px;height:600px;”>

Alternatively, you can use the width and height attributes:

Example

```
<img src=”img_girl.jpg” alt=”Girl in a jacket” width=”500” height=”600”>
```

The width and height attributes always defines the width and height of the image in pixels.

Width and Height, or Style?

The width, height, and style attributes are valid in HTML5.

However, we suggest using the style attribute. It prevents styles sheets from changing the size of images:

Example

```
<! DOCTYPE html>
    <html>
    <head>
    <style>
    img {
        width: 100%;
    }
    </style>
    </head>
    <body>

    <img src=”html5.gif” alt=”HTML5 Icon” width=”128” height=”128”>
    <img src=”html5.gif” alt=”HTML5 Icon” style=”width:128px;height:128px;”>

    </body>
    </html>
```

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

Example

```
<img src=”/images/html5.gif” alt=”HTML5 Icon” style=”width: 128px;height:128px;”>
```

Images on Another Server

Some web sites store their images on image servers.
Actually, you can access images from any web address in the world:

Example

```
<img src=”https://www.w3schools.com/images/w3schools_green.jpg” alt=”W3Schools.com”>
```

You can read more about file paths in the chapter HTML File Paths.

Animated Images

HTML allows animated GIFs:

Example

```
<img src=”programming.gif” alt=”Computer Man” style=”width:48px;height:48px;”>
```

Image as a Link

To use an image as a link, put the <img> tag inside the <a> tag:

Example

```
<a href=”default.asp”>
    <img src=”smiley.gif” alt=”HTML tutorial” style=”width:42px;height:42px;border:0;”>
  </a>
```

Note: border: 0; is added to prevent IE9 (and earlier) from displaying a border around the image (when the image is a link).

Image Floating

Use the CSS float property to let the image float to the right or to the left of a text:

Example

```
<p><img src=”smiley.gif” alt=”Smiley face” style=”float:right;width:42px;height:42px;”>
    The image will float to the right of the text.</p>
    <p><img src=”smiley.gif” alt=”Smiley face” style=”float:left;width:42px;height:42
px;”>
    The image will float to the left of the text.</p>
```

Image Maps

The <map> tag defines an image-map. An image-map is an image with clickable areas.

In the image below, click on the computer, the phone, or the cup of coffee:

Example

```
<img src=”workplace.jpg” alt=”Workplace” usemap=”#workmap”>
    <map name=”workmap”>
     <area shape=”rect” coords=”34, 44,270,350” alt=”Computer” href=”computer.htm”>
      <area shape=”rect” coords=”290,172,333,250” alt=”Phone” href=”phone.htm”>
      <area shape=”circle” coords=”337,300,44” alt=”Coffee” href=”coffee.htm”>
    </map>
```

The name attribute of the <map> tag is associated with the <img>’s usemap attribute and creates a relationship between the image and the map.

The <map> element contains a number of <area> tags, that define the clickable areas in the image-map.

Background Image

To add a background image on an HTML element, use the CSS property background-image:

Example

To add a background image on a web page, specify the background-image property on the BODY element:

```
    <body style=”background-image:url(‘clouds.jpg’)”>
    <h2>Background Image</h2>
    </body>
```

Example

To add a background image on a paragraph, specify the background-image property on the P element:

```
<body>
<p style=”background-image:url(‘clouds.jpg’)”>
...
</p>
</body>
```

4.7 WORKING WITH FORMS

Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax –

```
<form action = “Script URL” method = “GET|POST”>
   form elements like input, textarea etc.
</form>
```

4.7.1 Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes –

Sr.No	Attribute & Description
1	**action** .Backend script ready to process your passed data
2	**method** Method to be used to upload data. The most frequently used are GET and .POST methods

3	**target** Specify the target window or frame where the result of the script will be .displayed. It takes values like _blank, _self, _parent etc
4	**enctype** You can use the enctype attribute to specify how the browser encodes the − data before it sends it to the server. Possible values are **Application/x-www-form-urlencoded** − This is the standard method most .forms use in simple scenarios **mutlipart/form-data** − This is used when you want to upload binary data in .the form of files like image, word file etc

Note − You can refer to Perl & CGI for a detail on how form data upload works.

4.7.2 HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

4.7.3 Text Input Controls

There are three types of text input used on forms −

- **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

■ 4.7.4 Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name –

```
<!DOCTYPE html>
<html>
   <head>
      <title>Text Input Control</title>
   </head>
   <body>
      <form >
         First name: <input type = “text” name = “first_name” />
         <br>
         Last name: <input type = “text” name = “last_name” />
      </form>
   </body>
</html>
```

This will produce the following result –

First name:	
Last name:	

Attributes

Following is the list of attributes for <input> tag for creating text field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.

2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password –

```
<!DOCTYPE html>
<html>
   <head>
      <title>Password Input Control</title>
   </head>
   <body>
      <form >
         User ID : <input type = “text” name = “user_id” />
         <br>
         Password: <input type = “password” name = “password” />
      </form>
   </body>
</html>
```

This will produce the following result –

User ID :	
Password:	

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for password input control it will be set to **password**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

4.7.4 Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description –

```
<!DOCTYPE html>
<html>
   <head>
      <title>Multiple-Line Input Control</title>
   </head>
```

```
  <body>
    <form>
      Description : <br />
      <textarea rows = “5” cols = “50” name = “description”>
        Enter description here...
      </textarea>
    </form>
  </body>
</html>
```

This will produce the following result –

Description:

Enter description here...

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**rows** Indicates the number of rows of text area box.
3	**cols** Indicates the number of columns of text area box

4.7.5 Checkbox Control

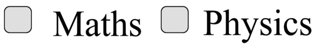
Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.

Example

Here is an example HTML code for a form with two checkboxes –

```
<!DOCTYPE html>
<html>
   <head>
      <title>Checkbox Control</title>
   </head>

   <body>
      <form>
         <input type = “checkbox” name = “maths” value = “on”> Maths
         <input type = “checkbox” name = “physics” value = “on”> Physics
      </form>
   </body>
</html>
```

This will produce the following result –



□ Maths □ Physics

Attributes

Following is the list of attributes for <checkbox> tag.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control ..it will be set to **checkbox**
2	**name** Used to give a name to the control which is sent to the server .to be recognized and get the value
3	**value** .The value that will be used if the checkbox is selected
4	**checked** .Set to *checked* if you want to select it by default

4.7.6 Radio Button Control

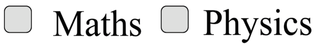
Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.

Example

Here is example HTML code for a form with two radio buttons –

```
<!DOCTYPE html>
<html>
   <head>
      <title>Radio Box Control</title>
   </head>
   <body>
      <form>
         <input type = “radio” name = “subject” value = “maths”> Maths
         <input type = “radio” name = “subject” value = “physics”> Physics
      </form>
   </body>

</html>
```

This will produce the following result –



□ Maths □ Physics

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to radio.

2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the radio box is selected.
4	**checked** Set to *checked* if you want to select it by default.

4.7.7 Select Box Control

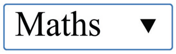
A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>

   <head>
      <title>Select Box Control</title>
   </head>
   <body>
      <form>
         <select name = “dropdown”>
            <option value = “Maths” selected>Maths</option>
            <option value = “Physics”>Physics</option>
         </select>
      </form>
   </body>
</html>
```

This will produce the following result –



Attributes

Following is the list of important attributes of <select> tag –

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**size** This can be used to present a scrolling list box.
3	**multiple** If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag –

Sr.No	Attribute & Description
1	**value** The value that will be used if an option in the select box box is selected.
2	**selected** Specifies that this option should be the initially selected value when the page loads.
3	**label** An alternative way of labeling options

4.7.8 File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

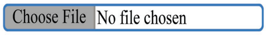
Example

Here is example HTML code for a form with one file upload box –

```
<!DOCTYPE html>
<html>
```

```
    <head>
       <title>File Upload Box</title>
    </head>
    <body>
       <form>
          <input type = "file" name = "fileupload" accept = "image/*" />
       </form>
    </body>
</html>
```

This will produce the following result –



Attributes

Following is the list of important attributes of file upload box –

Sr.No	Attribute & Description
1	name Used to give a name to the control which is sent to the server .to be recognized and get the value
2	accept .Specifies the types of files that the server accepts

4.7.9 Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values –

Sr.No	**Type & Description**
1	**submit** This creates a button that automatically submits a form.

2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**image** This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons –

```
<!DOCTYPE html>
<html>
   <head>
      <title>File Upload Box</title>
   </head>
   <body>
      <form>
         <input type = “submit” name = “submit” value = “Submit” />
         <input type = “reset” name = “reset” value = “Reset” />
         <input type = “button” name = “ok” value = “OK” />
          <input type = “image” name = “imagebutton” src = “/html/images/logo.png” />
      </form>
   </body>
</html>
```

This will produce the following result –

4.7.10 Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control –

```
<!DOCTYPE html>
<html>
   <head>
      <title>File Upload Box</title>
   </head>
   <body>
      <form>
         <p>This is page 10</p>
         <input type = “hidden” name = “pagename” value = “10” />
         <input type = “submit” name = “submit” value = “Submit” />
         <input type = “reset” name = “reset” value = “Reset” />
      </form>
   </body>
</html>
```

This will produce the following result –

```
This is page 10
```

SUMMARY

- An HTML element is a component of an HTML document that tells a web browser how to structure and interpret a part of the HTML document. HTML elements can contain formatting instructions, semantic meaning, and content.
- HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.
- The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.
- The tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.
- Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.
- The div tag is known as Division tag. The Div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar etc). Div tag has both open() tag and it is mandatory to close the tag.
- Div tag is a container tag inside div tag we can put more than one HTML element and can group together and can apply CSS for them.
- HTML also includes logical styles. These are more general descriptions that may be interpreted by each browser differently. They make your document easier to follow and edit. They will also allow your page to conform to local custom; if a journal citation (for example) is typically bold in one country and italicized in another, the use of the logical style *will cause it to appear correctly in each place.*
- HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.
- An HTML table is defined with the <table> tag. Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.
- Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.
- Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.

KNOWLEDGE CHECK

1. **Which one of the following is the most significant heading tag?**

   a. <h1>

   b. <h3>

   c. <h5>

   d. <h6>

2. **Choose the correct option.**

   a. Elements in HTML describes the way of presenting the content in the browser.

   b. All elements must have atleast one attribute associated with it.

   c. Both a and b.

   d. None of the above

3. **In HTML, <ul> means**

   a. Underline

   b. Unique list.

   c. Unordered list.

   d. None of the above

4. **Which one of the following options is correct?**

   a. Both block element and inline element start in the new line.

   b. Both block element and inline element start in the same line

   c. Only inline element starts in new line.

   d. Only block element starts in new line.

5. **Which of the following is an inline element?**

   a. <p>

   b. <h1>

   c. <div>

   d. <s>

6. **The BODY Tag Is Usually Used After**

   a. EM Tag

   b. TITLE Tag

   c. HEAD Tag

   d. HTML Tag

7. **Which of the following is the attribute of <form> tag in HTML?**
   a. Action
   b. Method
   c. Both (a) and (b)
   d. None of the above

8. **Which amongst the following are correct HTML code for creating a hyperlink?**
   a. <a>https://www.tutorialandexample.com/</a>
   b. <a url=”https://www.tutorialandexample.com/”>Tutorial and Example</a>
   c. <a”https://www.tutorialandexample.com/”>Tutorial and Example</a>
   d. <a href=”https://www.tutorialandexample.com/”> Tutorial and Example</a>

9. **A Simple Text File Which Tells The Browser What To Cache Is Known As A ....**
   a. Input Files
   b. Output Files
   c. A Manifest File
   d. HTML File

10. **Which tag is used to create a checkbox in HTML?**
   a. <checkbox>
   b. <Input type = “checkbox”>
   c. <cb>
   d. <Input type = “checkbox”>

REVIEW QUESTIONS

1. Explain the Nested HTML Elements.
2. Discuss the Empty HTML elements.
3. Define the adding Meta Tags to your documents.
4. Explain the document revision date.
5. Describe the creating web layout using Div Tag.
6. Explain the HTML form controls.

Check Your Result

1. (a)	2. (a)	3. (c)	4. (d)	5. (d)
6. (d)	7. (c)	8. (d)	9. (d)	10. (b)

REFERENCES

1. Berglund A., 2006. Extensible Stylesheet Language (XSL) Version 1.1. World Wide Web Consortium (W3C).
2. Bray, T. et al, 2008. Extensible Markup Language (XML) 1.0 (Fifth Edition). World Wide Web Consortium (W3C).
3. Costello, J. et al, 2002. XML: A New Web Site Architecture. SUNY Center for Technology in Government. Albany, NY, http://www.ctg.albany.edu/publications/reports/xml
4. Extensible Markup Language (XML) 1.0 (fifth edition) T. Bray, J. Paoli, C.M. Sperberg-McQueen, Eve Maler, François Yergeau, editors, 26 November 2008.
5. Extensible Markup Language (XML) 1.0 (fifth edition) T. Bray, J. Paoli, C.M. Sperberg-McQueen, Eve Maler, François Yergeau, editors, 26 November 2008.
6. Gabor Grothendieck and Thomas Petzoldt. R help desk: Date and time classes in R. R News, 4(1):29-32, 2004.
7. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 -- Proceedings in Computational Statistics*, pages 575-580. Physica Verlag, Heidelberg, 2002.
8. Pilgrim, Mark. HTML5: Up and Running, O'Reilly, 2010.
9. Powers, Shelley. HTML5 Media Integrating audio and video with the Web, O'Reilly, 2011.
10. Rhea, John. A Beginner's Guide to Learning HTML (and Smacking Zombies Upside the Web Development), Undead Institute, 2018.
11. Smith, M., 2013. HTML: The Markup Language (an HTML language reference). World Wide Web Consortium
12. Stevens, Luke. The Truth About HTML5 (For Web Designers), 2012.
13. Weyl, Estelle, et al. HTML5 & CSS3 For The Real World, Sitepoint, 2016.

CHAPTER 5

HTML5

"Betting completely on HTML5 is one of the, if not THE biggest strategic mistake we've made."

– Mark Zuckerberg

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Overview the HTML5
2. Understand the syntax of HTML5
3. Describe the elements of HTML5
4. Know about attributes of HTML5
5. Understand the web forms 2.0 and web storage of HTML5
6. Learn about HTML5 SVG

INTRODUCTION

HTML5 is a programming language whose acronym stands for Hyper Text Markup Language. It is a system that allows the modification of the appearance of web pages, as well as making adjustments to their appearance. It also used to structure and present content for the web.

With HTML5, browsers like Firefox, Chrome, Explorer, Safari and more, can know how to display a particular web page, know where the elements are, where to put the images and where to place the text.

Apart from HTML5, there are other languages that are necessary to give format and interactivity to a site, but the basic structure of any page is first defined in the HTML5 language.

The main virtues of HTML5 over its predecessor (HTML4) is that you can add multimedia content without using Flash or another media player. Thanks to HTML5, users can access websites without being connected to the Internet. Added to this is the drag and drop functionality, as well as the online editing of documents that has been popularized by Google Docs.

One of the design goals for HTML5 is to support for multimedia on mobile devices. New syntactic features were introduced to support this, such as video, audio and canvas tags. HTML5 also introduces new features which can really change the way users interact with documents including:

- New parsing rules for enhanced flexibility
- New attributes
- Elimination of outmoded or redundant attributes
- Drag and drop capabilities from one HTML5 document to another
- Offline editing
- Messaging enhancements
- Detailed rules for parsing
- MIME and protocol handler registration
- A common standard for storing data in SQL databases (Web SQL)

HTML5 official logo

HTML 5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML 5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav>, and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a />, <cite>, and <menu> have been changed, redefined, or standardized.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML 5 specification and HTML 5 also better defines the processing for any invalid documents.

5.1 OVERVIEW OF HTML5

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the **World Wide Web**.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

Keyword

World Wide Web is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.

Benefits of Html5 mobile responsive

5.1.1 Browser Support for HTML5

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

5.1.2 New Features of HTML5

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- **WebSocket** − A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.

- **Microdata** – This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** – Drag and drop the items from one location to another location on the same webpage.

5.1.3 HTML5 Backward Compatibility

HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. Its new features have been built on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

5.2 SYNTAX OF HTML5

The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.

HTML 5 does not have the same syntax rules as **XHTML** where we needed lower case tag names, quoting our attributes, an attribute had to have a value and to close all empty elements.

HTML5 comes with a lot of flexibility and it supports the following features –

- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.

Keyword

XHTML
is part
of the family of
XML markup
languages.
It mirrors or
extends versions
of the widely
used Hypertext
Markup
Language
(HTML), the
language in
which Web pages
are formulated.

5.2.1 The DOCTYPE

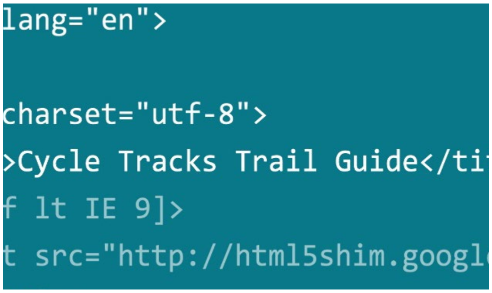
DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

HTML 5 authors would use simple syntax to specify DOCTYPE as follows –

```
<!DOCTYPE html>
```

The above syntax is case-insensitive.

HTML5: Structure, Syntax, and Semantics



```
lang="en">

charset="utf-8">
>Cycle Tracks Trail Guide</ti
f lt IE 9]>
t src="http://html5shim.googl
```

5.2.2 Character Encoding

HTML 5 authors can use simple syntax to specify Character Encoding as follows –

<meta charset = “UTF-8”>

The above syntax is case-insensitive.

The <script> tag

It’s common practice to add a type attribute with a value of “text/javascript” to script elements as follows –

<script type = “text/javascript” src = “scriptfile.js”></script>

HTML 5 removes extra information required and you can use simply following syntax –

<script src = “scriptfile.js”></script>

The <link> tag

So far you were writing <link> as follows –

<link rel = “stylesheet” type = “text/css” href = “stylefile.css”>

HTML 5 removes extra information required and you can simply use the following syntax –

<link rel = “stylesheet” href = “stylefile.css”>

5.2.3 HTML5 Document

The following tags have been introduced for better structure −

section − This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.

article − This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.

aside − This tag represents a piece of content that is only slightly related to the rest of the page.

header − This tag represents the header of a section.

footer − This tag represents a footer for a section and can contain information about the author, copyright information, et cetera.

nav − This tag represents a section of the document intended for **navigation**.

- **dialog** − This tag can be used to mark up a conversation.
- **figure** − This tag can be used to associate a caption together with some embedded content, such as a **graphic** or video.

Keyword

Navigation is a field of study that focuses on the process of monitoring and controlling the movement of a craft or vehicle from one place to another.

5.3 HTML5 ELEMENTS

HTML5 elements are marked up using start tags and end tags. Tags are delimited using angle brackets with the tag name in between.

The difference between start tags and end tags is that the latter includes a slash before the tag name.

Following is the example of an HTML5 element −

<p>...</p>

HTML5 tag names are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Most of the elements contain some content like <p>...</p> contains a paragraph. Some elements, however, are forbidden from containing any content at all and these are known as void elements. *For example*, **br, hr, link, meta**, etc.

5.3.1 List of Tag

A complete list of standard tags available in HTML5 is given below. All the tags are ordered alphabetically along with an indication if they have been introduced newly or they have been deprecated in HTML5.

Tag	Description
<!--...-->	Specifies a comment
<!DOCTYPE>	Specifies the document type
<a>	Specifies an anchor
<abbr>	Specifies an abbreviation
<acronym>	Deprecated: Specifies an acronym
<address>	Specifies an address element
<applet>	Deprecated: Specifies an applet
<area>	Specifies an area inside an image map
<article>	New Tag: Specifies an independent piece of content of a document, such as a blog entry or newspaper article
<aside>	New Tag: Specifies a piece of content that is only slightly related to the rest of the page.
<audio>	New Tag: Specifies an audio file.
<base>	Specifies a base URL for all the links in a page
<basefont>	Deprecated: Specifies a base font
<bdo>	Specifies the direction of text display
<bgsound>	Specifies the background music
<blink>	Specifies a text which blinks
<blockquote>	Specifies a long quotation
<body>	Specifies the body element
<br>	Inserts a single line break
<button>	Specifies a push button
<canvas>	New Tag: This is used for rendering dynamic bitmap graphics on the fly, such as graphs or games.
<caption>	Specifies a table caption
<center>	Deprecated: Specifies centered text
<col>	Specifies attributes for table columns

<colgroup>	Specifies groups of table columns
<command>	New Tag: Specifies a command the user can invoke.
<comment>	Puts a comment in the document
<datalist>	New Tag: Together with the a new list attribute for input can be used to make combo-boxes
<dd>	Specifies a definition description
<del>	Specifies deleted text
<details>	New Tag: Specifies additional information or controls which the user can obtain on demand.
<dir>	Deprecated: Specifies a directory list
<div>	Specifies a section in a document
<dl>	Specifies a definition list
<dt>	Specifies a definition term
<embed>	New Tag: Defines external interactive content or plugin.
<fieldset>	Specifies a field set
<figure>	New Tag: Specifies a piece of self-contained flow content, typically referenced as a single unit from the main flow of the document.
<b>	Specifies bold text
<big>	Deprecated: Specifies big text
<i>	Specifies italic text
<small>	Specifies small text
<tt>	Deprecated: Specifies teletype text
<font>	Deprecated: Specifies text font, size, and color
<footer>	New Tag: Specifies a footer for a section and can contain information about the author, copyright information, et cetera.
<form>	Specifies a form
<frame>	Deprecated: Specifies a sub window (a frame)
<frameset>	Deprecated: Specifies a set of frames
<head>	Specifies information about the document
<header>	New Tag: Specifies a group of introductory or navigational aids.

<hgroup>	New Tag: Specifies the header of a section.
<h1> to <h6>	Specifies header 1 to header 6
<hr>	Specifies a horizontal rule
<html>	Specifies an html document
<isindex>	Deprecated: Specifies a single-line input field
<iframe>	Specifies an inline sub window (frame)
<ilayer>	Specifies an inline layer
<img>	Specifies an image
<input>	Specifies an input field
<ins>	Specifies inserted text
<keygen>	New Tag: Specifies control for key pair generation.
<keygen>	Generate key information in a form
<label>	Specifies a label for a form control
<layer>	Specifies a layer
<legend>	Specifies a title in a field set
<li>	Specifies a list item
<link>	Specifies a resource reference
<map>	Specifies an image map
<mark>	New Tag: Specifies a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.
<marquee>	Create a scrolling-text marquee
<menu>	Deprecated: Specifies a menu list
<meta>	Specifies meta information
<meter>	New Tag: Specifies a measurement, such as disk usage.
<multicol>	Specifies a multicolumn text flow
<nav>	New Tag: Specifies a section of the document intended for navigation.
<nobr>	No breaks allowed in the enclosed text
<noembed>	Specifies content to be presented by browsers that do not support the <embed>tag
<noframes>	Deprecated: Specifies a noframe section
<noscript>	Specifies a noscript section

<object>	Specifies an embedded object
<ol>	Specifies an ordered list
<optgroup>	Specifies an option group
<option>	Specifies an option in a drop-down list
<output>	New Tag: Specifies some type of output, such as from a calculation done through scripting.
<p>	Specifies a paragraph
<param>	Specifies a parameter for an object
<cite>	Specifies a citation
<code>	Specifies computer code text
<dfn>	Specifies a definition term
<em>	Specifies emphasized text
<kbd>	Specifies keyboard text
<samp>	Specifies sample computer code
<strong>	Specifies strong text
<var>	Specifies a variable
<plaintext>	Deprecated: Render the raminder of the document as preformatted plain text
<pre>	Specifies preformatted text
<progress>	New Tag: Specifies a completion of a task, such as downloading or when performing a series of expensive operations.
<q>	Specifies a short quotation
<ruby>	New Tag: Together with <rt> and <rp> allow for marking up ruby annotations.
<script>	Specifies a script
<section>	New Tag: Represents a generic document or application section.
<select>	Specifies a selectable list
<spacer>	Specifies a white space
<span>	Specifies a section in a document
<s>	Deprecated: Specifies strikethrough text
<strike>	Deprecated: Specifies strikethrough text
<style>	Specifies a style definition
<sub>	Specifies subscripted text
<sup>	Specifies superscripted text
<table>	Specifies a table

<tbody>	Specifies a table body
<td>	Specifies a table cell
<textarea>	Specifies a text area
<tfoot>	Specifies a table footer
<th>	Specifies a table header
<thead>	Specifies a table header
<time>	New Tag: Specifies a date and/or time.
<title>	Specifies the document title
<tr>	Specifies a table row
<u>	Deprecated: Specifies underlined text
<ul>	Specifies an unordered list
<video>	New Tag: Specifies a video file.
<wbr>	New Tag: Specifies a line break opportunity.
<wbr>	Indicate a potential word break point within a <nobr> section
<xmp>	Deprecated: Specifies preformatted text

5.4 HTML5 ATTRIBUTES

Elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attribute which illustrates how to mark up a div element with an attribute named class using a value of “example” −

<div class = “example”>...</div>

Attributes may only be specified within start tags and must never be used in end tags.

5.4.1 Standard Attributes

The attributes listed below are supported by almost all the HTML 5 tags.

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.

align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true, false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.
id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.
spellcheck	true, false	Specifies if the element must have it's spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

5.4.2 Custom Attributes

A new feature being introduced in HTML 5 is the addition of custom data attributes.

A custom data attribute starts with **data-** and would be named based on your requirement. Here is a simple example –

```
<div class = “example” data-subject = “physics” data-level = “complex”>
  ...
</div>
```

The above code will be perfectly valid HTML5 with two custom attributes called *datasubject* and *data-level*. You would be able to get the values of these attributes using JavaScript APIs or CSS in similar way as you get for standard attributes.

5.5 WEB FORMS 2.0

Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.

5.5.1 The <input> element in HTML4

HTML4 input elements use the **type** attribute to specify the data type.HTML4 provides following types –

.Sr.No	Type & Description
1	**text** .A free-form text field, nominally free of line breaks
2	**password** A free-form text field for sensitive information, nominally free of .line breaks
3	**checkbox** .A set of zero or more values from a predefined list
4	**radio** .An enumerated value
5	**submit** .A free form of button initiates form submission
6	**file** .An arbitrary file with a MIME type and optionally a file name

7	**image** A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form .submission
8	**hidden** .An arbitrary string that is not normally displayed to the user
9	**select** .An enumerated value, much like the radio type
10	**textarea** .A free-form text field, nominally with no line break restrictions
11	**button** A free form of button which can initiates any event related to .button

Following is the simple example of using labels, radio buttons, and submit buttons –

```
  ...
  <form action = “http://example.com/cgiscript.pl” method = “post”>
     <p>
        <label for = “firstname”>first name: </label>
        <input type = “text” id = “firstname”><br />

        <label for = “lastname”>last name: </label>
        <input type = “text” id = “lastname”><br />

        <label for = “email”>email: </label>
        <input type = “text” id = “email”><br>

        <input type = “radio” name = “sex” value = “male”> Male<br>
        <input type = “radio” name = “sex” value = “female”> Female<br>
        <input type = “submit” value = “send”> <input type = “reset”>
     </p>
<form/>
...
```

5.5.2 The <input> element in HTML5

Apart from the above-mentioned attributes, HTML5 input elements introduced several new values for the **type** attribute. .These are listed below

Remember

Try all the following example using latest version of Opera browser.

Sr.No.	Type & Description
1	**datetime** A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.
2	**datetime-local** A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.
3	**date** A date (year, month, day) encoded according to ISO 8601.
4	**month** A date consisting of a year and a month encoded according to ISO 8601.
5	**week** A date consisting of a year and a week number encoded according to ISO 8601.
6	**time** A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.
7	**number** It accepts only numerical value. The step attribute specifies the precision, defaulting to 1.
8	**range** The range type is used for input fields that should contain a value from a range of numbers.

9	**email** It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.
10	**url** It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in http://www.example.com format or in http://example.com format.

5.5.3 The <output> element

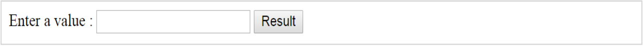
HTML5 introduced a new element <output> which is used to represent the result of different types of output, such as output written by a script.

You can use the **for** attribute to specify a relationship between the output element and other elements in the document that affected the calculation (for example, as inputs or parameters). The value of the for attribute is a space-separated list of IDs of other elements.

```
<!DOCTYPE HTML>

<html>
  <head>
    <script type = “text/javascript”>

      function showResult() {
        x = document.forms[“myform”][“newinput”].value;
        document.forms[“myform”][“result”].value = x;
      }
    </script>
  </head>

  <body>

    <form action = “/cgi-bin/html5.cgi” method = “get” name = “myform”>
```

```
            Enter a value : <input type = “text” name = “newinput” />
            <input type = “button” value = “Result”  onclick = “showResult();” />
            <output name = “result”></output>
        </form>

    </body>
</html>
```

It will produce the following result –



Enter a value : Result

5.5.4 The Placeholder Attribute

HTML5 introduced a new attribute called **placeholder**. This attribute on <input> and <textarea> elements provide a hint to the user of what can be entered in the field.

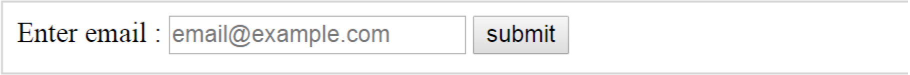
Here is the simple syntax for placeholder attribute –

```
<input type = “text” name = “search” placeholder = “search the web”/>
```

This attribute is supported by latest versions of Mozilla, Safari and Crome browsers only.

```
<!DOCTYPE HTML>

<html>
    <body>

        <form action = “/cgi-bin/html5.cgi” method = “get”>
            Enter email : <input type = “email” name = “newinput”
                placeholder = “email@example.com”/>
            <input type = “submit” value = “submit” />
        </form>

    </body>
</html>
```

This will produce the following result –



Enter email : email@example.com submit

5.5.5 The Autofocus Attribute

This is a simple one-step pattern, easily programmed in JavaScript at the time of document load, automatically focus one particular form field.

HTML5 introduced a new attribute called **autofocus** which would be used as follows −

```
<input type = “text” name = “search” autofocus/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```
<!DOCTYPE HTML>

<html>
  <body>

    <form action = “/cgi-bin/html5.cgi” method = “get”>
      Enter email : <input type = “text” name = “newinput” autofocus/>
      <p>Try to submit using Submit button</p>
      <input type = “submit” value = “submit” />
    </form>

  </body>
</html>
```

5.5.6 The Required Attribute

Now you do not need to have JavaScript for client-side validations like empty text box would never be submitted because HTML5 introduced a new attribute called **required** which would be used as follows and would insist to have a value −

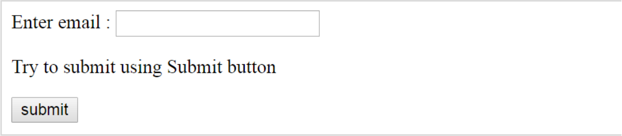
```
<input type = “text” name = “search” required/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```
<!DOCTYPE HTML>

<html>
  <body>
```

```
      <form action = "/cgi-bin/html5.cgi" method = "get">
         Enter email : <input type = "text" name = "newinput" required/>
         <p>Try to submit using Submit button</p>
         <input type = "submit" value = "submit" />
      </form>

   </body>
</html>
```

It will produce the following result –



Enter email :

Try to submit using Submit button

5.6 HTML5 SVG

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X,Y coordinate system etc.

SVG became a W3C Recommendation 14. January 2003 and you can check latest version of SVG specification at SVG Specification.

5.6.1 Viewing SVG Files

Most of the web browsers can display SVG just like they can display PNG, GIF, and JPG. Internet Explorer users may have to install the Adobe SVG Viewer to be able to view SVG in the browser.

5.6.2 Embedding SVG in HTML5

HTML5 allows embedding SVG directly using **<svg>...</svg>** tag which has following simple syntax –

```
<svg xmlns = "http://www.w3.org/2000/svg">
      ...
   </svg>
```

Firefox 3.7 has also introduced a configuration option (“about:config”) where you can enable HTML5 using the following steps –

- Type **about:config** in your Firefox address bar.
- Click the “I’ll be careful, I promise!” button on the warning message that appears (and make sure you adhere to it!).
- Type **html5.enable** into the filter bar at the top of the page.
- Currently it would be disabled, so click it to toggle the value to true.

Now your Firefox HTML5 parser should be enabled and you should be able to experiment with the following examples.

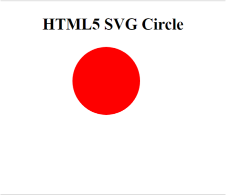
5.6.3 HTML5 SVG Circle

Following is the HTML5 version of an SVG example which would draw a circle using <circle> tag –

```
<!DOCTYPE html>

<html>
   <head>

      <style>
         #svgelem {
            position: relative;
            left: 50%;
            -webkit-transform: translateX(-20%);
            -ms-transform: translateX(-20%);
            transform: translateX(-20%);
         }
      </style>
      <title>SVG</title>
      <meta charset = “utf-8” />
   </head>

   <body>
      <h2 align = “center”>HTML5 SVG Circle</h2>
```

```
    <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
      <circle id = “redcircle” cx = “50” cy = “50” r = “50” fill = “red” />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Circle

HTML5 SVG Circle

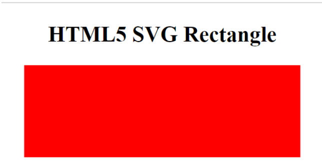
5.6.4 HTML5 SVG Rectangle

Following is the HTML5 version of an SVG example which would draw a rectangle using <rect> tag –

```
<!DOCTYPE html>
<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-50%);
        -ms-transform: translateX(-50%);
        transform: translateX(-50%);
      }
    </style>
    <title>SVG</title>
    <meta charset = “utf-8” />
  </head>
```

```
    <body>
      <h2 align = “center”>HTML5 SVG Rectangle</h2>

      <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
        <rect id = “redrect” width = “300” height = “100” fill = “red” />
      </svg>
    </body>
  </html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Rectangle

HTML5 SVG Rectangle

5.6.5 HTML5 SVG Line

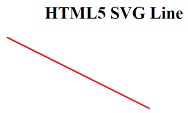
Following is the HTML5 version of an SVG example which would draw a line using <line> tag –

```
  <!DOCTYPE html>

  <html>
    <head>

      <style>
        #svgelem {
          position: relative;
          left: 50%;
          -webkit-transform: translateX(-50%);
          -ms-transform: translateX(-50%);
          transform: translateX(-50%);
```

```
            }
        </style>
        <title>SVG</title>
        <meta charset = “utf-8” />
    </head>

    <body>
        <h2 align = “center”>HTML5 SVG Line</h2>

        <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
            <line x1 = “0” y1 = “0” x2 = “200” y2 = “100”
                style = “stroke:red;stroke-width:2”/>
        </svg>
    </body>
</html>
```

You can use the **style** attribute which allows you to set additional style information like stroke and fill colors, width of the stroke, etc.

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Line

HTML5 SVG Line

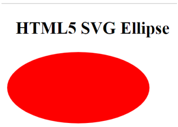
5.6.6 HTML5 SVG Ellipse

Following is the HTML5 version of an SVG example which would draw an ellipse using <ellipse> tag −

```
<!DOCTYPE html>

<html>
    <head>
                <style>
```

```
         #svgelem {
            position: relative;
            left: 50%;
            -webkit-transform: translateX(-40%);
            -ms-transform: translateX(-40%);
            transform: translateX(-40%);
         }
      </style>
      <title>SVG</title>
      <meta charset = "utf-8" />
   </head>

   <body>
      <h2 align = "center">HTML5 SVG Ellipse</h2>

      <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
         <ellipse cx = "100" cy = "50" rx = "100" ry = "50" fill = "red" />
      </svg>

   </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.


HTML5 SVG Ellipse

HTML5 SVG Ellipse

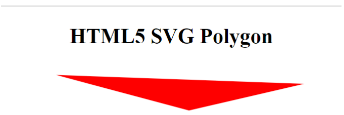
5.6.7 HTML5 SVG Polygon

Following is the HTML5 version of an SVG example which would draw a polygon using <polygon> tag –

```
<!DOCTYPE html>
```

```
<html>
   <head>

      <style>
         #svgelem {
            position: relative;
            left: 50%;
            -webkit-transform: translateX(-50%);
            -ms-transform: translateX(-50%);
            transform: translateX(-50%);
         }
      </style>
      <title>SVG</title>
      <meta charset = “utf-8” />
   </head>

   <body>
      <h2 align = “center”>HTML5 SVG Polygon</h2>

      <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
         <polygon  points = “20,10 300,20, 170,50” fill = “red” />
      </svg>
   </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Polygon

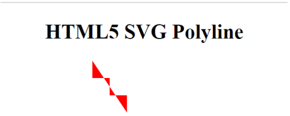
HTML5 SVG Polygon

5.6.8 HTML5 SVG Polyline

Following is the HTML5 version of an SVG example which would draw a polyline using <polyline> tag –

```
<!DOCTYPE html>


<html>
   <head>


      <style>
#svgelem {

            position: relative;
            left: 50%;
            -webkit-transform: translateX(-20%);
            -ms-transform: translateX(-20%);
            transform: translateX(-20%);
         }
</style>

<title>SVG</title>

<meta charset = “utf-8” />

</head>


<body>

<h2 align = “center”>HTML5 SVG Polyline</h2>

<svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>

<polyline points = “0,0 0,20 20,20 20,40 40,40 40,60” fill = “red” />

</svg>

</body>

</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Polyline

HTML5 – SVG Polyline

5.6.9 HTML5 SVG Gradients

Following is the HTML5 version of an SVG example which would draw an **ellipse** using <ellipse> tag and would use <radialGradient> tag to define an SVG radial gradient.

Similarly, you can use <linearGradient> tag to create SVG linear gradient.

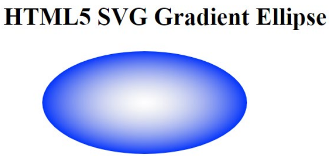
```
<!DOCTYPE html>

<html>
   <head>

      <style>
         #svgelem {
            position: relative;
            left: 50%;
            -webkit-transform: translateX(-40%);
            -ms-transform: translateX(-40%);
            transform: translateX(-40%);
         }
      </style>
      <title>SVG</title>
      <meta charset = “utf-8” />
   </head>
```

Keyword

Ellipse is a curve in a plane surrounding two focal points such that the sum of the distances to the two focal points is constant for every point on the curve.

```
      <body>
         <h2 align = “center”>HTML5 SVG Gradient Ellipse</h2>

         <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
            <defs>
                <radialGradient id=”gradient” cx = “50%” cy = “50%” r = “50%” fx =
“50%”
                  fy = “50%”>
               <stop offset = “0%” style = “stop-color:rgb(200,200,200); stop-opacity:0”/>
                <stop offset = “100%” style = “stop-color:rgb(0,0,255); stop-opacity:1”/>
               </radialGradient>
            </defs>
            <ellipse cx = “100” cy = “50” rx = “100” ry = “50”
               style = “fill:url(#gradient)” />
         </svg>

      </body>
   </html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Gradient Ellipse

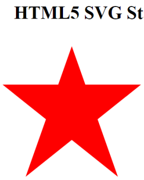
HTML5 SVG Gradients

5.6.10 HTML5 SVG Star

Following is the HTML5 version of an SVG example which would draw a star using <polygon> tag.

```
   <html>
      <head>

         <style>
            #svgelem {
               position: relative;
               left: 50%;
               -webkit-transform: translateX(-40%);
               -ms-transform: translateX(-40%);
               transform: translateX(-40%);
            }
         </style>
         <title>SVG</title>
         <meta charset = “utf-8” />
      </head>

      <body>
         <h2 align = “center”>HTML5 SVG Star</h2>

         <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
            <polygon points = “100,10 40,180 190,60 10,60 160,180” fill = “red”/>
         </svg>
       </body>
   </html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Star

5.7 WEB STORAGE OF HTML5

HTML5 introduces two mechanisms, similar to HTTP session cookies, for storing structured data on the client side and to overcome following drawbacks.

- Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.

- Cookies are included with every HTTP request, thereby sending data unencrypted over the internet.
- Cookies are limited to about 4 KB of data. Not enough to store required data.

The two storages are **session storage** and **local storage** and they would be used to handle different situations.

The latest versions of pretty much every browser supports HTML5 Storage including Internet Explorer.

5.7.1 Session Storage

The *Session Storage* is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.

If a user buying plane tickets in two different windows, using the same site. If the site used cookies to keep track of which ticket the user was buying, then as the user clicked from page to page in both windows, the ticket currently being purchased would "leak" from one window to the other, potentially causing the user to buy two tickets for the same flight without really noticing.

HTML5 introduces the *session Storage* **attribute** which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window, i.e., **session** and as soon as you close the window, the session would be lost.

Keyword

Attribute is a specification that defines a property of an object, element, or file.

Following is the code which would set a session variable and access that variable –

```
<!DOCTYPE HTML>

<html>
   <body>
      <script type = "text/javascript">

         if( sessionStorage.hits ) {
               sessionStorage.hits = Number(sessionStorage.
hits) +1;
```

```
         } else {
            sessionStorage.hits = 1;
         }
         document.write("Total Hits :" + sessionStorage.hits );
      </script>

      <p>Refresh the page to increase number of hits.</p>
      <p>Close the window and open it again and check the result.</p>

   </body>
</html>
```

This will produce the following result –

```
Total Hits :1

Refresh the page to increase number of hits.

Close the window and open it again and check the result.
```

5.7.2 Local Storage

The *Local Storage* is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons.

Again, cookies do not handle this case well, because they are transmitted with every request.

Example

HTML5 introduces the *localStorage* attribute which would be used to access a page's local storage area without no time limit and this local storage will be available whenever you would use that page.

Following is the code which would set a local storage variable and access that variable every time this page is accessed, even next time, when you open the window –

```
<!DOCTYPE HTML>
```

```
<html>
  <body>
    <script type = "text/javascript">

      if( localStorage.hits ) {
        localStorage.hits = Number(localStorage.hits) +1;
      } else {
        localStorage.hits = 1;
      }
      document.write("Total Hits :" + localStorage.hits );
    </script>

    <p>Refresh the page to increase number of hits.</p>
    <p>Close the window and open it again and check the result.</p>

  </body>
</html>
```

This will produce the following result –

```
Total Hits :1

Refresh the page to increase number of hits.

Close the window and open it again and check the result.
```

5.7.3 Delete Web Storage

Storing sensitive data on local machine could be dangerous and could leave a security hole.

The *Session Storage Data* would be deleted by the browsers immediately after the session gets terminated.

To clear a local storage setting you would need to call **localStorage.remove('key')**; where 'key' is the key of the value you want to remove. If you want to clear all settings, you need to call **localStorage.clear()** method.

Following is the code which would clear complete local storage –

```
<!DOCTYPE HTML>
```

```
<html>
   <body>

      <script type = “text/javascript”>
         localStorage.clear();

         // Reset number of hits.
         if( localStorage.hits ) {
            localStorage.hits = Number(localStorage.hits) +1;
         } else {
            localStorage.hits = 1;
         }
         document.write(“Total Hits :” + localStorage.hits );

      </script>

      <p>Refreshing the page would not to increase hit counter.</p>
      <p>Close the window and open it again and check the result.</p>

   </body>
</html>
```

This will produce following result −

```
Total Hits :1

Refreshing the page would not to increase hit counter.

Close the window and open it again and check the result.
```

CASE STUDY

HTML5: WEB APP DEVELOPMENT

The Client

The client is an industry leading market research firm.

They build cloud-based software that facilitate two-way communication that ranges from market research to civic engagement for their clients' specific platforms.

Their solutions have accessible UIs and feature modules for survey authoring, community management, discussion forums, and reporting services.

The Challenge

The client required their survey application to be accessible on mobile devices. They were looking for a web-based solution that would work across multiple devices and have a rich mobile user interface.

Key Challenges

- The client required a rich web GUI that is comparable to a native application in terms of control sophistication and user experience.
- The application had to be extensible so that modifications to elements of the survey and new ways to integrate with the back end would be easy to accommodate.
- The application was to be used on a variety of devices; consequently, the respective native browser's behavior and constraints had to be considered.

The Process

- Agile development process consisting of multiple sprints.
- Testing locally on multiple devices.
- Deliveries made in overlapping meetings with clients.
- Cloud-based staging environment for a shared test system.

How Optimus Helped

Optimus created a web-based survey tool that was based on the principle of determining the user agent and rendering compatible UI.

To create a rich GUI we used HTML5, CSS3, XML, and XSLT. We made sure that our framework had the necessary provisions so that it could easily include new elements within the surveys.

A factory based solution was created that can dynamically load handlers for different type of data fetching services. This allows the clients to add new data generation services when they are created.

We then conducted device and user agent detections by using open source 3rd party WURFL services. Device-specific code customizations were made to handle the device's limitations.

Finally, the application was vetted to see if it performed at an acceptable performance level. We made sure to verify an acceptable level for all question types, including those that required heavy UI side processing.

The end result is a web-based survey tool that works across virtually all modern mobile devices. The UI has a consistent look and feel and provides a positive user experience.

SUMMARY

- HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.
- HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. Its new features have been built on existing features and allow you to provide fallback content for older browsers.
- The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.
- DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.
- Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.
- HTML5 introduced a new attribute called placeholder. This attribute on <input> and <textarea> elements provide a hint to the user of what can be entered in the field.
- SVG stands for Scalable Vector Graphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer. SVG is mostly useful for vector type diagrams like Pie charts, Two dimensional graphs in an X,Y coordinate system etc.
- Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.
- The Session Storage is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.
- The Local Storage is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire userauthored documents or a user's mailbox, on the client side for performance reasons.

KNOWLEDGE CHECK

1. **Which of the following browser supports HTML5 in its latest version?**
   a. Apple Safari
   b. Google Chrome
   c. Both of the above.
   d. None of the above.

2. **Which of the following tag can be used to mark up a conversation in HTML5?**
   a. footer
   b. nav
   c. dialog
   d. figure

3. **Which of the following input control represents a date consisting of a year and a week number encoded according to ISO 8601 in Web Form 2.0?**
   a. week
   b. time
   c. number
   d. range

4. **Can you use MathML tags directly in HTML5 without any plugin?**
   a. True
   b. False

5. **When a session storage data gets deleted in HTML5?**
   a. The Session Storage Data would be deleted by the browsers immediately after the session gets terminated.
   b. If you want to clear all settings, you need to call localStorage.clear() method.
   c. Both of the above.
   d. None of the above.

6. **Which browsers does the HTML5 supports?**
   a. Firefox
   b. Safari
   c. Internet Explorer
   d. All the mentioned above

7. **………….is the replacement for cookies in HTML5.**
   a. Java scripts

b. Web beacons

c. Local Storage

d. All of the above

8. The new form elements are introduced in HTML5 is -

a. 2

b. 3

c. 4

d. 6

9. ………………is not a HTML5 tag.

a. <source>

b. <video>

c. <slider>

d. All of the above

10. …………..is not a HTML5 tag.

a. <source>

b. <video>

c. <slider>

d. All of the above

REVIEW QUESTIONS

1. What is HTML5? Name some of the new features of HTML5. Which browsers support HTML5?
2. What is web forms 2.0?
3. What are the purpose of ‘output’ tag and ‘placeholder’ attribute in HTML5?
4. What are the purpose of ‘autofocus’ attribute and ‘required’ attribute in HTML5?
5. Can you use SVG tags directly in HTML5 without any plugin?
6. What are the drawbacks of cookies?
7. What do you mean by session storage and local storage in HTML5?
8. What is purpose of getCurrentPosition() method of geolocation object of HTML5?
9. What is purpose of watchPosition() method of geolocation object of HTML5?
10. What is purpose of clearPosition() method of geolocation object of HTML5?

Check Your Result

1. (c)	2. (c)	3. (a)	4. (a)	5. (a)
6. (d)	7. (c)	8. (b)	9. (c)	10. (b)

REFERENCES

1. Baker, Mitchell (2014-05-14). “DRM and the Challenge of Serving Users”. Mozilla. Retrieved 2014-05-20.
2. Cimpanu, Catalin (29 May 2019). “Browser vendors Win War with W3C over HTML and DOM standards”. ZDNet. Archived from the original on 29 May 2019. Retrieved 29 May 2019.
3. Danny O’Brien (2013-10-02). “Lowering Your Standards: DRM and the Future of the W3C”. Electronic Frontier Foundation. Retrieved 2013-10-03.
4. Doctorow, Cory (2014-05-14). “Firefox’s adoption of closedsource DRM breaks my heart”. The Guardian. Retrieved 2014- 05-20.
5. Faulkner, Steve; Eicholz, Arron; Leithead, Travis; Danilo, Alex; Moon, Sangwhan; Doyle Navara, Erika; O’Connor, Theresa; Berjon, Robin, eds. (14 December 2017) [2016]. “HTML 5.2 W3C Recommendation”. Revised version. World Wide Web Consortium (W3C). Retrieved 26 July 2018.
6. Gal, Andreas (2014-05-14). “Reconciling Mozilla’s Mission and W3C EME”. Mozilla. Retrieved 2014-05-20.
7. Hickson, Ian, ed. (25 July 2018). “HTML Living Standard”. Multipage Version. WHATWG. Retrieved 26 July 2018.
8. HTML5 is a W3C recommendation. W3C Blog. World Wide Web Consortium (W3C). 28 October 2014. Retrieved 28 October 2014.
9. Manu Sporny (2013-01-26). “DRM in HTML5”. The Beautiful, Tormented Machine. Manu Sporny. Archived from the original on 2014-04-25. Retrieved 2014-05-16.
10. Scott Gilbertson (2013-02-12). “DRM for the Web? Say It Ain’t So”. Webmonkey. Condé Nast. Archived from the original on 2013-04-06. Retrieved 2013-03-21.
11. W3C Confirms May 2011 for HTML5 Last Call, Targets 2014 for HTML5 Standard. World Wide Web Consortium. 14 February 2011. Retrieved 18 February 2011.
12. W3C Workshop on Web Applications and Compound Documents (Day 1). W3.org. W3C. 1 June 2004. Retrieved 30 December 2011.

CHAPTER

6

CASCADING STYLE SHEETS

"The biggest mistake we made as a company was betting too much on HTML5."

– Mark Zuckerberg

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Discuss on the concept of CSS
2. Explain the various properties of CSS

INTRODUCTION

Cascading style sheets (CSS) are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML. CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

6.1 CONCEPT OF CSS

CSS is the standard and preferred mechanism for formatting HTML pages. Conforming with the separation of concerns design pattern and best practice, cascading style sheets provide a central location in which information about what various fonts, foreground colors, background colors, italicization and emphasization should be applied to various HTML elements within a webpage. Cascading style sheets can also control how various parts of a page, such as the header, footer, body, article content, sections and asides, are laid out on the page. This is extremely helpful when content must be laid out in a dramatically different fashion depending upon whether it is being viewed on a desktop, tablet or a **smartphone**.

Keyword

Smartphone is a handheld personal computer. It possesses extensive computing capabilities, including high-speed access to the Internet using both Wi-Fi and mobile broadband.

6.1.1 Proper use of CSS

In the early days of the World Wide Web (WWW), it was common for HTML files to include not only markup language and content, but formatting information and JavaScript as well.

This made webpages difficult to write, difficult to read, difficult to update and difficult to maintain.

As the web matured, it became a best practice to divide HTML, scripting content and style information into separate, easy-to-maintain files. As such, a modern webpage is typically made up of three separate entities: a cascading style sheet, a JavaScript file and the HTML file itself.

6.1.2 Implementing CSS formatting

The cascading nature of CSS files is attributed to the fact that style information for a webpage can be defined in any of three different places, also known as style levels.

The preferred practice is to put style information in a separate file with a .css extension. Using formatting information contained within an external cascading style sheet is accomplished via the HTML link tag. A webpage can link to zero, one or may different external CSS files by using multiple link tags.

```
<link rel=”stylesheet” type=”text/css” href=”what-is-css.css”>
```

However, on smaller projects or in cases where a given webpage is interested in overriding some of the style information in an external CSS file, style information can be written within a <style> tag inside the webpage. This is known as an internal style level. Internal style level information within a webpage will override any style information provided by an external cascading style sheet.

6.1.3 Cascading Style Rules

Furthermore, all HTML5 tags have a style property that one can use to override any style information defined at either the page style level or in an external style sheet. Using an HTML tag to define CSS information is referred to as an inline style. The fact that style rules dictate that parent-level styles are overridden by page-level styles and page-level styles are overridden by tag-level styles is what is meant by style sheets being cascading.

A Cascading Style sheet rule tells the browser what the HTML looks like, and what it should do. A rule can dictate what just one HTML tag should look like, or you can construct your own rule to be applied as and where you want it.

A rule can be set up that tells the browser to format every <P> tag so that its first line is indented. Or you could construct your own paragraph rule, and just apply the style to certain paragraphs, not all paragraphs.

6.1.4 Style Sheet Language

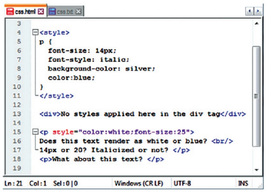
CSS syntax is relatively simple. The name of the element to style, referred to as the CSS selector, is followed by braces, within which various attributes, such as font-size and background-color are assigned values. The World Wide Web Consortium (W3C) standards organization defines the CSS attributes, although various browsers may offer supplemental support by defining their own custom fields. This is often done for proposed attributes that are expected to be included in future CSS releases.

6.1.5 CSS selectors

CSS selectors can be HTML tags, class attributes assigned to HTML tags and even states of a given element, such as the disabled state of an input field or the hover state of an anchor link. Making it possible to customize the style of components depending upon their state or how they are classified on a page provides the graphic designer a great deal of flexibility in determining how a webpage will be rendered by the browser.

CSS example

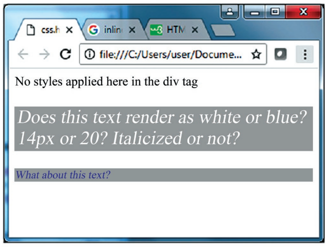
The following HTML page is using both an internal style and an inline style for defining cascading style sheet information:



```

<style>
p {
  font-size: 14px;
  font-style: italic;
  background-color: silver;
  color:blue;
}
</style>

<div>No styles applied here in the div tag</div>

<p style="color:white;font-size:25">
Does this text render as white or blue? <br/>
14px or 20? Italicized or not? </p>
<p>What about this text? </p>

```

Figure 1. Using inline styles vs. internal styles with cascading styles

Notice that, when the webpage renders, any text that is not contained within a style just renders as plain black text on a white background.



No styles applied here in the div tag

Does this text render as white or blue? 14px or 20? Italicized or not?

What about this text?

Figure 2. Rendering a CSS formatted HTML webpage that uses both inline styles and internal styles

Any text contained within a paragraph tag without an inline style pulls formatting information from the internal stylesheet, which means it is rendered as blue, 14-pixel, italicized text on a silver background. However, when the inline style is used, the internal style overrides and matches properties, while leaving nonconflicting attributes alone, causing the text to render as white, 20-pixel, italicized text on a silver background.

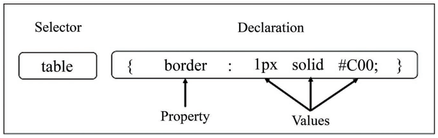
6.1.6 CSS - Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

- **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
- **Value** – Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

You can put CSS Style Rule Syntax as follows –

selector { property: value }



Example – You can define a table border as follows –

```
table{ border :1px solid #C00; }
```

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings –

```
h1 {
   color: #36CFFF;
}
```

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

```
* {
   color: #000000;
}
```

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

```
ul em {
   color: #000000;
}
```

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {
   color: #000000;
}
```

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example –

```
h1.black {
   color: #000000;
}
```

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

You can apply more than one class selectors to given element. Consider the following example –

```
<p class = "center bold">
   This para will be styled by the classes center and bold.
</p>
```

The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

```
#black {
   color: #000000;
}
```

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example –

```
h1#black {
   color: #000000;
}
```

This rule renders the content in black for only <h1> elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors, For example –

```
#black h2 {
   color: #000000;
}
```

In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having *id* attribute set to *black*.

The Child Selectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example −

```
body > p {
   color: #000000;
}
```

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text* −

```
input[type = “text”] {
   color: #000000;
}
```

The advantage to this method is that the <input type = “submit” /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

- **p[lang]** − Selects all paragraph elements with a *lang* attribute.
- **p[lang=”fr”]** − Selects all paragraph elements whose *lang* attribute has a value of exactly «fr”.
- **p[lang~=”fr”]** − Selects all paragraph elements whose *lang* attribute contains the word «fr”.
- **p[lang|=”en”]** − Selects all paragraph elements whose *lang* attribute contains values that are exactly «en», or begin with «en-».

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example −

```
h1 {
   color: #36C;
   font-weight: normal;
   letter-spacing: .4em;
```

```
    margin-bottom: 1em;
    text-transform: lowercase;
}
```

Here all the property and value pairs are separated by a semicolon (;). You can keep them in a single line or multiple lines. For better **readability**, we keep them in separate lines. For a while, don't bother about the properties mentioned in the above block.

Keyword

Readability is the ease with which a reader can understand a written text. In natural language, the readability of text depends on its content and its presentation.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example −

```
h1, h2, h3 {
    color: #36C;
    font-weight: normal;
    letter-spacing: .4em;
    margin-bottom: 1em;
    text-transform: lowercase;
}
```

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *id* selectors together as shown below −

```
#content, #footer, #supplement {
    position: absolute;
    left: 510px;
    width: 200px;
    }
```

Remember

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

6.2 CSS PROPERTIES

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

6.2.1 CSS - Colors

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element. They can also be used to affect the color of borders and other decorative effects.

You can specify your color values in various formats. Following table lists all the possible formats –

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

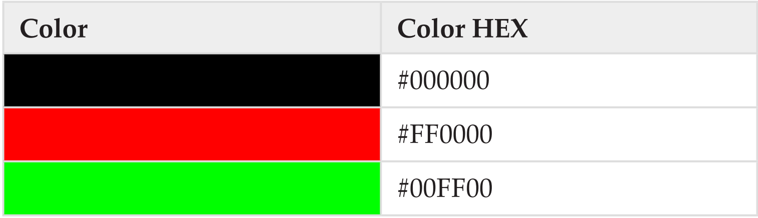
These formats are explained in more detail in the following sections –

CSS Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro, or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.



Color	Color HEX
⬛	#000000
🟥	#FF0000
🟩	#00FF00

🟦	#0000FF
🟨	#FFFF00
🟦	#00FFFF
🟪	#FF00FF
▨	#C0C0C0
⬜	#FFFFFF

CSS Colors - Short Hex Codes

This is a shorter form of the six-digit notation. In this format, each digit is replicated to arrive at an equivalent six-digit value. For example: #6A7 becomes #66AA77.

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro, or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.

Color	Color HEX
	#000
	#F00
	#0F0
	#0FF
	#FF0
	#0FF
	#F0F
	#FFF

CSS Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is the example to show few colors using RGB values.

Color	Color RGB
⬛	rgb(0,0,0)
🟥	rgb(255,0,0)
🟩	rgb(0,255,0)
🟦	rgb(0,0,255)
🟨	rgb(255,255,0)
🟦	rgb(0,255,255)
🟪	rgb(255,0,255)
▒	rgb(192,192,192)
⬜	rgb(255,255,255)

Keyword

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

Building Color Codes

You can build millions of color codes using our Color Code Builder. Check our HTML Color Code Builder. To use this tool, .you would need a **Java** Enabled Browser

Browser Safe Colors

Here is the list of 216 colors which are supposed to be most safe and computer independent colors. These colors vary from hexa code 000000 to FFFFFF. These colors are safe to use because they ensure that all computers would display the – colors correctly when running a 256 color palette

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF

009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF

FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

6.2.2 CSS - Fonts

This division teaches you how to set fonts of a content, available in an HTML element. You can set following font properties of an element −

- The font-family property is used to change the face of a font.
- The font-style property is used to make a font italic or oblique.
- The font-variant property is used to create a small-caps effect.
- The font-weight property is used to increase or decrease how bold or light a font appears.
- The font-size property is used to increase or decrease the size of a font.
- The font property is used as **shorthand** to specify a number of other font properties.

Keyword

Shorthand is an abbreviated symbolic writing method that increases speed and brevity of writing as compared to longhand, a more common method of writing a language.

Set the Font Family

Following is the example, which demonstrates how to set the font family of an element. Possible value could be any font family name.

```
<html>
   <head>
   </head>

   <body>
      <p style = “font-family:georgia,garamond,serif;”>
         This text is rendered in either georgia, garamond,
or the
          default serif font depending on which font  you
have at your system.
      </p>
```

```
  </body>
</html>
```

This will produce following result –

This text is rendered in either georgia, garamond, or the default serif font depending on which font you have at your system.

Set the Font Style

Following is the example, which demonstrates how to set the font style of an element. Possible values are normal, italic and oblique.

```
<html>
  <head>
  </head>
  <body>
    <p style = "font-style:italic;">
      This text will be rendered in italic style
    </p>
  </body>
</html>
```

This will produce following result –

This text will be rendered in italic style

Set the Font Variant

The following example demonstrates how to set the font variant of an element. Possible values are *normal and small-caps*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-variant:small-caps;">
      This text will be rendered as small caps
    </p>
  </body>
```

```
</html>
```

This will produce following result –

THIS TEXT WILL BE RENDERED AS SMALL CAPS

Set the Font Weight

The following example demonstrates how to set the font weight of an element. The font-weight property provides the functionality to specify how bold a font is. Possible values could be *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*.

```
<html>
   <head>
   </head>

   <body>
      <p style = “font-weight:bold;”>
         This font is bold.
      </p>

      <p style = “font-weight:bolder;”>
         This font is bolder.
      </p>

      <p style = “font-weight:500;”>
         This font is 500 weight.
      </p>
   </body>
</html>
```

This will produce following result –

This font is bold.

This font is bolder.

This font is 500 weight.

Set the Font Size

The following example demonstrates how to set the font size of an element. The font-size property is used to control the size of fonts. Possible values could be *xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %.*

```
<html>
   <head>
   </head>

   <body>
      <p style = “font-size:20px;”>
         This font size is 20 pixels
      </p>

      <p style = “font-size:small;”>
         This font size is small
      </p>

      <p style = “font-size:large;”>
         This font size is large
      </p>
   </body>
</html>
```

This will produce following result –

```
This font size is 20 pixels
This font size is small
This font size is large
```

Set the Font Size Adjust

The following example demonstrates how to set the font size adjust of an element. This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

```
<html>
   <head>
   </head>
   <body>
```

```
    <p style = “font-size-adjust:0.61;”>
      This text is using a font-size-adjust value.
    </p>
  </body>
</html>
```

This will produce following result –

This text is using a font-size-adjust value.

Set the Font Stretch

The following example demonstrates how to set the font stretch of an element. This property relies on the user’s computer to have an expanded or condensed version of the font being used.

Possible values could be *normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded*.

```
<html>
  <head>
  </head>

  <body>
    <p style = “font-stretch:ultra-expanded;”>
      If this doesn’t appear to work, it is likely that your computer
     doesn’t have a <br>condensed or expanded version of the font being used.
    </p>
  </body>
</html>
```

This will produce following result –

If this doesn’t appear to work, it is likely that your computer doesn’t have a condensed or expanded version of the font being used.

Shorthand Property

You can use the *font* property to set all the font properties at once. For example –

```
<html>
  <head>
  </head>
  <body>
   <p style = “font:italic small-caps bold 15px georgia;”>
       Applying all the properties on the text at once.
     </p>
  </body>
</html>
```

This will produce following result −

APPLYING ALL THE PROPERTIES ON THE TEXT AT ONCE.

6.2.3 CSS - Tables

This stage will teach you how to set different properties of an HTML table using CSS. You can set following properties of a table −

- The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The **border-spacing** specifies the width that should appear between table cells.
- The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
- The **empty-cells** specifies whether the border should be shown if a cell is empty.
- The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before **rendering** it.

Now, we will see how to use these properties with examples.

Keyword

Rendering is the automatic process of generating a photorealistic or non-photorealistic image from a 2D or 3D model by means of computer programs.

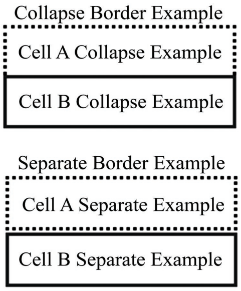
The border-collapse Property

This property can have two values *collapse* and *separate*. The following example uses both the values −

```
<html>
    <head>
        <style type = “text/css”>
            table.one {border-collapse:collapse;}
            table.two {border-collapse:separate;}

            td.a {
                border-style:dotted;
                border-width:3px;
                border-color:#000000;
                padding: 10px;
            }
            td.b {
                border-style:solid;
                border-width:3px;
                border-color:#333333;
                padding:10px;
            }
        </style>
    </head>

    <body>
        <table class = “one”>
            <caption>Collapse Border Example</caption>
            <tr><td class = “a”> Cell A Collapse Example</td></tr>
            <tr><td class = “b”> Cell B Collapse Example</td></tr>
        </table>
        <br />

        <table class = “two”>
```

```
      <caption>Separate Border Example</caption>
      <tr><td class = “a”> Cell A Separate Example</td></tr>
      <tr><td class = “b”> Cell B Separate Example</td></tr>
    </table>
  </body>
</html>
```

It will produce the following result –



Collapse Border Example

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example

Cell A Separate Example

Cell B Separate Example

The border-spacing Property

The border-spacing property specifies the distance that separates adjacent cells’. borders. It can take either one or two values; these should be units of length.

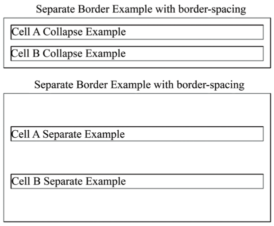
If you provide one value, it will applies to both vertical and horizontal borders. Or you can specify two values, in which case, the first refers to the horizontal spacing and the second to the vertical spacing –

```
<style type=”text/css”>
  /* If you provide one value */
  table.example {border-spacing:10px;}
  /* This is how you can provide two values */
  table.example {border-spacing:10px; 15px;}
</style>
```

Now let’s modify the example and see the effect –

```
<html>
  <head>
    <style type = “text/css”>
      table.one {
        border-collapse:separate;
        width:400px;
        border-spacing:10px;
      }
      table.two {
        border-collapse:separate;
        width:400px;
        border-spacing:10px 50px;
      }
    </style>
  </head>
  <body>
      <table class = “one” border = “1”>
      <caption>Separate Border Example with border-spacing</caption>
      <tr><td> Cell A Collapse Example</td></tr>
      <tr><td> Cell B Collapse Example</td></tr>
    </table>
    <br />

    <table class = “two” border = “1”>
      <caption>Separate Border Example with border-spacing</caption>
      <tr><td> Cell A Separate Example</td></tr>
      <tr><td> Cell B Separate Example</td></tr>
    </table>

  </body>
</html>
```

It will produce the following result –



Separate Border Example with border-spacing

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example with border-spacing

Cell A Separate Example

Cell B Separate Example

The caption-side Property

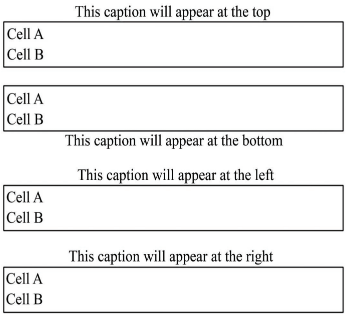
The caption-side property allows you to specify where the content of a <caption> element should be placed in relationship to the table. The table that follows lists the possible values.

This property can have one of the four values *top, bottom, left* or *right*. The following example uses each value.

```
<html>
   <head>
      <style type = “text/css”>
         caption.top {caption-side:top}
         caption.bottom {caption-side:bottom}
         caption.left {caption-side:left}
         caption.right {caption-side:right}
      </style>
   </head>

   <body>

      <table style = “width:400px; border:1px solid black;”>
         <caption class = “top”>
            This caption will appear at the top
         </caption>
         <tr><td > Cell A</td></tr>
         <tr><td > Cell B</td></tr>
      </table>
      <br />

      <table style = “width:400px; border:1px solid black;”>
         <caption class = “bottom”>
            This caption will appear at the bottom
         </caption>
         <tr><td > Cell A</td></tr>
         <tr><td > Cell B</td></tr>
```

```
    </table>
    <br />

    <table style = “width:400px; border:1px solid black;”>
      <caption class = “left”>
        This caption will appear at the left
      </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr>
    </table>
    <br />

    <table style = “width:400px; border:1px solid black;”>
      <caption class = “right”>
        This caption will appear at the right
      </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr>
    </table>

  </body>
</html>
```

It will produce the following result –



```
This caption will appear at the top
Cell A
Cell B

Cell A
Cell B
This caption will appear at the bottom

This caption will appear at the left
Cell A
Cell B

This caption will appear at the right
Cell A
Cell B
```

The empty-cells Property

The empty-cells property indicates whether a cell without any content should have a border displayed.

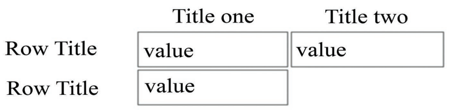
This property can have one of the three values - *show, hide* or *inherit*.

Here is the empty-cells property used to hide borders of empty cells in the <table> element.

```
<html>
   <head>
      <style type = "text/css">
         table.empty {
            width:350px;
            border-collapse:separate;
            empty-cells:hide;
         }
         td.empty {
            padding:5px;
            border-style:solid;
            border-width:1px;
            border-color:#999999;
         }
      </style>
   </head>

   <body>

      <table class = "empty">
         <tr>
            <th></th>
            <th>Title one</th>
            <th>Title two</th>
         </tr>

         <tr>
```

```
        <th>Row Title</th>
        <td class = “empty”>value</td>
        <td class = “empty”>value</td>
      </tr>

      <tr>
        <th>Row Title</th>
        <td class = “empty”>value</td>
        <td class = “empty”></td>
      </tr>
    </table>

  </body>
</html>
```

It will produce the following result –



	Title one	Title two
Row Title	value	value
Row Title	value	

The table-layout Property

The table-layout property is supposed to help you control how a browser should render or lay out a table.

This property can have one of the three values: *fixed*, .*auto* or *inherit*

The following example shows the difference between these .properties

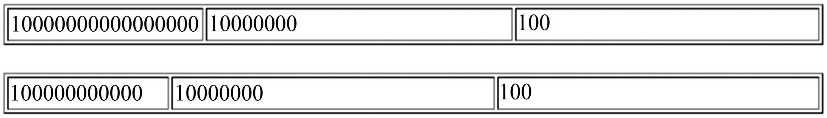
```
<html>
<head>
<“style type = “text/css>
} table.auto
table-layout: auto
{
} table.fixed
```

Did You Know?

CSS was first proposed by Håkon Wium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996.

```
            table-layout: fixed
         }
      </style>
   </head>

   <body>

      <table class = “auto” border = “1” width = “100%”>
         <tr>
            <td width = “20%”>1000000000000000000000000000</td>
            <td width = “40%”>10000000</td>
            <td width = “40%”>100</td>
         </tr>
      </table>
      <br />

      <table class = “fixed” border = “1” width = “100%”>
         <tr>
            <td width = “20%”>1000000000000000000000000000</td>
            <td width = “40%”>10000000</td>
            <td width = “40%”>100</td>
         </tr>
      </table>

   </body>
</html>
```

It will produce the following result –



10000000000000000	10000000	100

100000000000	10000000	100

SUMMARY

- Cascading style sheets (CSS) are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.
- A Cascading Style sheet rule tells the browser what the HTML looks like, and what it should do. A rule can dictate what just one HTML tag should look like, or you can construct your own rule to be applied as and where you want it.
- CSS syntax is relatively simple. The name of the element to style, referred to as the CSS selector, is followed by braces, within which various attributes, such as font-size and backgroundcolor are assigned values.
- CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.
- CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element. They can also be used to affect the color of borders and other decorative effects.
- A hexadecimal is a 6 digit representation of a color. The first two digits (RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).
- The border-collapse specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The border-spacing property specifies the distance that separates adjacent cells'. borders. It can take either one or two values; these should be units of length.

KNOWLEDGE CHECK

1. **Which of the following selector selects all elements of E that have the attribute attr that end with the given value?**
   a. E[attr^=value]
   b. E[attr$=value]
   c. E[attr*=value]
   d. none of the mentioned
2. **Which of the following selector selects the elements that are checked?**
   a. E ~ F
   b. ::after
   c. :checked
   d. none of the mentioned
3. **Which of the following selector selects the elements that are the default among a set of similar elements?**
   a. :default
   b. :%
   c. :disabled
   d. none of the mentioned
4. **Which of the following selector selects an element that has no children?**
   a. :empty
   b. :nochild
   c. :inheritance
   d. :no-child
5. **Which of the following selector selects the elements that are currently enabled?**
   a. :element
   b. :empty
   c. :enabled
   d. none of the mentioned
6. **CSS stands for -**
   a. Cascade style sheets
   b. Color and style sheets
   c. Cascading style sheets
   d. None of the above

7. Which of the following is the correct syntax for referring the external style sheet?

a. <style src = example.css>

b. <style src = “example.css” >

c. <stylesheet> example.css </stylesheet>

d. <link rel=”stylesheet” type=”text/css” href=”example.css”>

8. The property in CSS used to change the background color of an element is -

a. bgcolor

b. color

c. background-color

d. All of the above

9. The CSS property used to control the element’s font-size is -

a. text-style

b. text-size

c. font-size

d. None of the above

10. External stylesheets are stored in CSS files

a. TRUE

b. FALSE

c. Can be true or false

d. Can not say

REVIEW QUESTIONS

1. What is the cascading style rules?
2. Discuss on the style sheet language.
3. Give a suitable CSS example.
4. Write the syntax of CSS.
5. Explain the following CSS properties:
   a. CSS - Colors
   b. CSS - Tables

Check Your Result

1. (b) 2. (c) 3. (a) 4. (a) 5. (c)
6. (c) 7. (d) 8. (c) 9. (c) 10. (a)

REFERENCES

1. Andrew, Rachel. The New CSS Layout , A Book Apart, 2017.
2. Bartlett, Kynn. Sams Teach Yourself Cascading Style Sheets in 24 Hours, Second Edition, Sams, 2006.
3. Briggs, Owen et al. Cascading Style Sheets: Separating Content from Presentation, Second Edition, APress, 2004.
4. Brown, Tiffany. CSS Master, Sitepoint, 2021.
5. Budd, Andy, et al. CSS Mastery: Advanced Web Standards Solutions, Friends of ED, 2006.
6. Casciano, Chris. The CSS Pocket Guide, Peachpit Press, 2010.
7. Cederholm, Dan. CSS3 For Web Designers, A Book Apart, 2015.
8. Clarke, Andy. Hardboiled Web Design, Five Simple Steps, 2010.
9. Clarke, Andy. Transcending CSS: The Fine Art of Web Design, New Riders, 2006.
10. Collison, Simon. Professional CSS for Web Development: From Novice to Professional, APress, 2006.
11. Croft, Jeff, , et al. ProCSS Techniques, APress, 2006.
12. Debolt, Virginia. Integrated HTML and CSS, Sybex Inc, 2005.
13. Debolt, Virginia. Mastering Integrated HTML and CSS, Sybex Inc, 2007.
14. Freeman, Elisabeth and Freeman, Eric. Head First HTML with CSS & XHTML, O'Reilly & Associates, 2005.
15. Gasston, Peter. Book of CSS3, No Starch Press, 2011.
16. Gillenwater, Zoe, Mickley. Flexible Web Design: Creating Liquid and Elastic Layouts with CSS, New Riders Press, 2008.
17. Gillenwater, Zoe, Mickley. Stunning CSS3: A Project-based Guide to the Latest in CSS, New Riders Press, 2010.
18. Grannell, Craig. The Essential Guide to CSS and HTML Web Design, friends of ED, 2007.
19. Grannell, Craig. Web Designer's Reference, friends of ED, 2004.
20. Griffiths, Patrick. XHTML & CSS: A Web Standards Approach, New Riders, 2005.
21. Holzschlag, Molly and Clarke, Andy. Transcending CSS : The Fine Art of Web Design, New Riders Press, 2006.
22. Holzschlag, Molly. Spring Into HTML and CSS, Addison-Wesley Professional, 2005.

CHAPTER

7

WEB HOSTING

"Web Hosting Server is the lifeline of every website. It should be up and running 24/7."

– Dr. Chris Dayagdag

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Understand the overview of web hosting
2. Explain how to create website structures

INTRODUCTION

Web hosting is an online service that enables you to publish your website or web application on the Internet. When you sign up for a web hosting service, you basically rent some space on a physical server where you can store all the files and data necessary for your website to work properly.

A server is a physical computer that runs without any interruption so that your website is available all the time for anyone who wants to see it. Your web host is responsible for keeping that server up and running, protecting it from malicious attacks, and transferring your content — such as text, images, files, etc. — from the server to your visitors' browsers.

Website
(House)
Domain Name
(Address)
Web Hosting
(Land)
Internet
(Street)

When you decide to start a new website, you need to find a web hosting provider that will supply you with that server space. Your web host stores all your files, assets, and databases on the server. Whenever someone types your domain name into the address bar of their browser, your host transfers all the files necessary to serve that request.

Therefore, you need to choose a hosting plan that best fits your needs. In fact, it works similarly to housing rentals — you have to pay the rent regularly in order to keep the server running continuously.

To reduce risks, each Hostinger plan comes with a 30-day money-back guarantee so that you can see if our service really meets your expectations. When your website grows in traffic and you need more server space, you can move on to one of our more advanced plans without any delay.

In fact, you don't even need any programming knowledge to perform regular site management tasks. For instance, you can upload HTML and other files to the server, install content management systems such as WordPress or Drupal, access your database, and create backups for your site with ease.

Even though the cPanel hosting platform is used by most web hosting providers, it is a powerful tool, it can be intimidating to beginners who just want to get a site up and running quickly. Therefore, we decided to build a custom control panel for our users — hPanel.

Besides providing server space for your website, Hostinger also provides other services related to website management, such as:

- SSL certificates

- Email hosting and page builders
- Developer tools
- 24/7 customer support service
- Automated website backups
- One-click software installs for WordPress or Drupal and much more

7.1 OVERVIEW OF WEB HOSTING

Web hosting is a method by which a person or company rents a server to store data used to display a website, which is accessible through the **Internet**. Every website that can be found and accessed online is hosted on some type of server or similar machine, and various companies offer a number of different methods for this hosting. These can include free hosting for individuals who do not need many options, as well as dedicated systems that are more expensive but provide greater control.

Web hosting is an essential service for anyone that wants to set up a website. Without a web host, you cannot share your website with any users. It basically provides all of the services and technology that is required to get your website onto the internet including storage space/servers. There are many hosting companies that will offer to take care of your technology requirements and you can pretty much pick and choose how much support you require from such businesses depending on how much you want to spend, your own knowledge or experience and how much control you want to have over the hosting.

Did You Know?

Until 1991, the Internet was restricted to use only for research and education in the sciences and engineering and was used for email, telnet, FTP and USENET traffic - but only a tiny number of web pages. The World Wide Web protocols had only just been written and not until the end of 1993 would there be a graphical web browser for Mac or Windows computers. Even after there was some opening up of internet access, the situation was confused until 1995.

An Introduction to Web Hosting

Keyword

Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide.

Web hosting is when a person or company rents space a server to store data used to display a website.

Of course, the expense is a key driver in any decision around your web hosting solution and you may opt to go for one of the free web hosting options that are available. What you should be mindful of, however, is that free web hosting can leave you in a position where you are at the mercy of the supplier.

example

If sites have downtime, or they decide to add cost further down the line. There are a number of different types of web hosts and depending on the size of your business and your business priorities, you can choose the .most relevant type of web host

7.1.1 Basic Functionality of Web Hosting

Servers are powerful computers that have extremely large hard drives, or an array of hard drives, which can be rented to those who want a website on the Internet. Every server on the Internet has a unique numerical Internet Protocol (IP) address.

If the servers act as metaphorical apartment buildings with unique addresses, then each apartment unit within these buildings provides rented space for individual websites. Much like real apartment buildings, each unit also has an address based on "the building" in which it is located, which is the IP address for a particular website.

The best web hosting providers for 2018

When a person or company rents a space on a server through web hosting, then it is basically "setting up house" on the Internet. That website can be reached by a unique address, which is based on the server's address. Most people type in a domain name to find a site, but the actual address is a number that corresponds to that particular site.

Types of Web Hosting Services

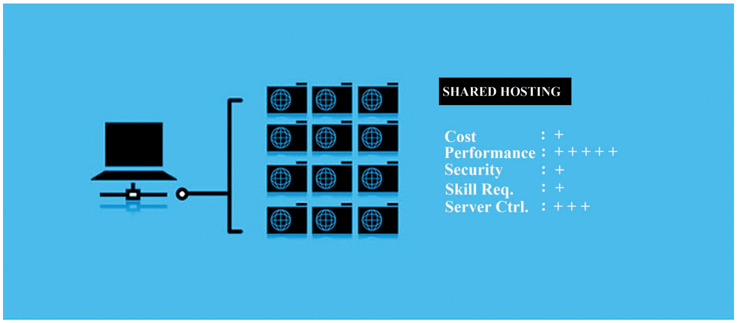
There are many different types of web hosting available, depending on the needs and budget of a person. Most commercial packages come with certain capabilities for users, such as scripts that allow interactive functions, forums, or email addresses. Shared hosting services provide multiple users with data storage on a single system. Dedicated systems, which are usually more expensive, give each user a drive or server dedicated solely to their site, providing greater speeds and more powerful options. There are also "virtual" dedicated systems, which use software to mimic the power of dedicated methods in a more practical, shared system.

Web Hosting

Depending on your level of expertise, web hosting, which is the task of making websites and web pages available to many users, can be either very simple of daunting. In this section we are going to explore the different types of hosting.

Shared Hosting

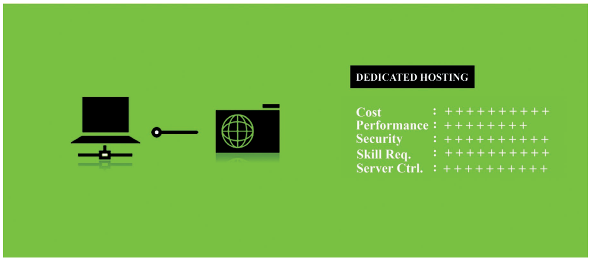
Shared hosting is one of the more common methods for website hosting. It is also the cheapest, and least reliable. This is when several website owners are sharing a single server including all resources such as CPU, memory and drive space. This is the most cost efficient as all of the different website owners share in the operational expenses of the single server. However, you can suffer from “bad neighbor syndrome”: if another website is sharing the same webhost as your site, and if it is very busy, then it will hog the CPU and RAM and network bandwidth of the web host, resulting in poor performance for all the other websites on the host. The result of this will be that all sites will load very slowly.



There are also security risks with this kind of hosting. If one of the other websites gets hacked, or infected with malware, then it is very likely that the other websites on the same webhost will also be infected since the all run on the same computer. And if there is a DDOS attack on one of the sites, then, again, they are all affected because they share the same resource. This type of hosting should only be used for small sites that are not expecting much traffic.

Dedicated Hosting

For those that rely on maximum performance and an ultimately secured platform, dedicated hosting provides that functionality. In this environment, a single person leases an entire server eliminating resource use by other website owners. This means that 100 % of the system is operating for your purposes alone. This also gives you full control of the machine, so that it can be configured any way you like, with whatever software, or modules and as little or as much security as you want. Dedicated servers are also one of the most expensive hosting options.

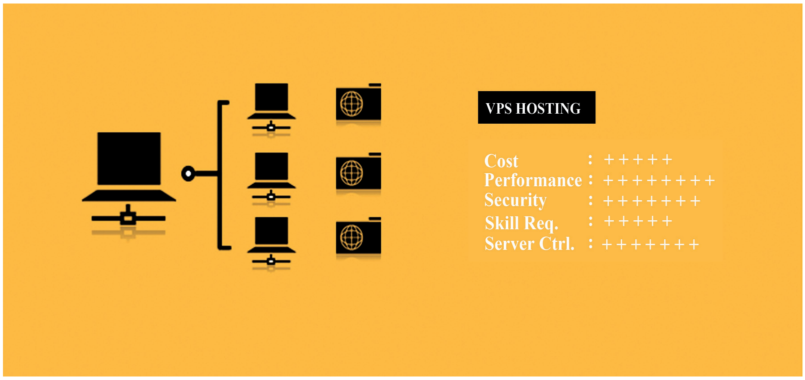

DEDICATED HOSTING
Cost : + + + + + + + + + +
Performance : + + + + + + + +
Security : + + + + + + + + + +
Skill Req. : + + + + + + + + + +
Server Ctrl. : + + + + + + + + + +

VPS Hosting

A Virtual Private Servers, or VPS, is a virtual machine running on top of a real machine. Several VPS servers may run on a single real server machine. A VPS runs its own separate copy of the operating system on its own separate space on the host server, and so can be treated as if it is really a separate computer. This means it has the advantages of a dedicated server in terms of having full control over the machine, but not the cost. The drawback is that performance will not be as good as a dedicated server, but it is far better than shared hosting. The main difference between VPS and shared hosting is that each VPS runs an entirely separate **operating system** instance inside its own virtual machine, whereas on a shared host, only one single operating system runs which is shared .by all the websites

Keyword

Operating system is system software that manages computer hardware and software resources and provides common services for computer programs.

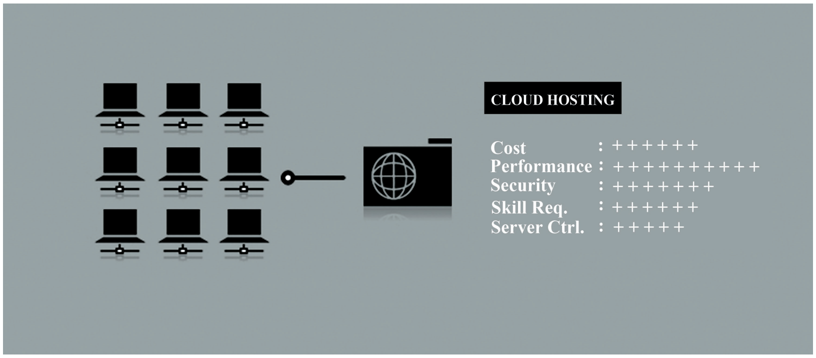


VPS HOSTING

Cost	:	+ + + + +
Performance	:	+ + + + + + + +
Security	:	+ + + + + + +
Skill Req.	:	+ + + + +
Server Ctrl.	:	+ + + + + + +

Cloud Hosting

Instead of relying on a single server to host a website, the Cloud alternative utilizes a network of machines to provide the content. A single account could be spread out across many different servers which could include those in other cities or even countries. This provides reliability, scaling of size and .faster recovery times in the event of a disaster



Using as much as they need without having to build and maintain their own computing infrastructure in-house. The resources required for maintaining your website are spread across more than one web server, reducing the chance of any downtime due to server malfunction. Cloud-based hosting is scalable, meaning your site can grow over time, using as much resource as it requires and you only pay for what you need.

Managed hosting

Most hosting packages you are likely to find online will be managed. Hosting companies provide technical services such as hardware and software setup and configuration, maintenance, hardware replacement, technical support, patching, updating and monitoring. Unlike standard dedicated hosting, the hosting provider looks after the day-to-day management of the hardware, operating systems and standardized applications. A popular sub-category is WordPress managed hosting. For many businesses, the beauty of Wordpress is in its simplicity. Its benefits are clear, it is easy to setup and it is straightforward to manage. Choosing a hosting provider for a Wordpress site can be more of a task, however, especially for those without expert knowledge or experience of the market. Many web-hosting companies now specialize solely in offering managed services for WordPress.

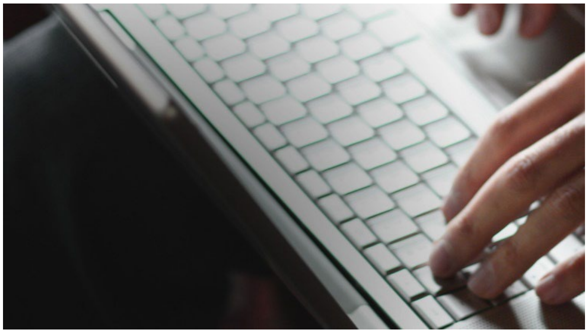
Colocation

Instead of keeping servers in-house or at a private data center, you may choose to "'co-locate" your equipment by renting space in a colocation center. The center will provide the power, bandwidth, IP address and cooling systems that your server requires. Space is rented out in "racks' and 'cabinets". Colocation gives access to higher levels of bandwidth than a normal office server room at a much lower cost. You are left to your own devices (literally) and will be expected to take care of everything including the hardware, software and services.

7.1.2 How Website Hosting Works?

Web hosts are companies that rent out their services and technologies to host websites on the internet. Once the hosting company hosts your website, users can access it by typing in your web address (domain name) in their web browser. When they do this, their computer connects to the server your website is hosted on. The server in turn serves (sends the files you have stored on the storage to display) the website to your web visitor in their **web browser**.

You all must have heard that to make a website visible on World Wide Web it is necessary to host it. Thus hosting helps an organization or business to make their website reachable via the World Wide Web. It basically provides the space, technology, and services that would be needed for your website to be reachable. There are three main types of web hosting viz. Shared hosting, VPS hosting, and Dedicated hosting. All three have different characteristics and have their own advantages and disadvantages.



How web hosting works?

Remember

Formerly, many colocation providers would accept any system configuration for hosting, even ones housed in desktop-style minitower cases, but most hosts now require rack mount enclosures and standard system configurations.

Keyword

Web browser is a software application for accessing information on the World Wide Web.

There are terms related to hosting a website like a web server, HTML, CSS, domain, IP address etc. your browser sends a request for the page to the server by connecting through an IP address. When the domain name is translated, you obtain an IP address. The web pages are made up of a computer language called HTML (Hypertext Markup Language) that narrates the format, content, and layout of the page with the help of CSS that is a stylesheet. The web servers then deliver the page through the internet by reacting to the browser's request. Therefore, in order to host your website on the web server, you need a hosting provider. It is a very simple procedure, but it might get confusing for people who are doing it or the first time. Have you ever thought of how this process of hosting works? Let us go through the steps to host your website.

IP address

An IP address is a numerical identity with numbers ranging from 1 to 255 and divided by periods. For e.g. the IP address of our website is 103.247.99.254. Each website has its own unique identity and right now there are around 4.2 billion numerical identities. The IP address is essential because there are many computers on the same location that is called LAN (Local Area Network) and many LANs are connected together that is called WAN (Wide Area Network), therefore, each should have a unique identity.

When you sign up for a hosting service, you get an IP address. Sometimes one IP address is shared by many and it is also cheaper in price but this can prove to be harmful to the security of your site and it could be penalized or blocked. Having your own dedicated IP address is safe but expensive. Making IP address the identity of your website is not very convenient, this is where domain name comes in the picture.

Domain names

Domain names are considered as the second name for IP address. It is difficult to remember these number and domain names are simpler to remember and understand. It is important to get a domain name for the website and acts as an address. For e.g. hostingrecipe.com. You are sure it is easy for you also to remember hostingrecipe.com rather than 103.247.99.254. One needs to purchase a domain name and it ranges from being cheap to expensive depending on your need. You can book your domain name from 1 to 10 years or so. Before its expiry, make sure, you restore it again or else your domain name will be publicly available.

DNS

DNS stands for Domain Name Server or Domain Name System. This step is known to be the hardest of all. It basically keeps a record of the information and is in charge of it i.e. Your IP address and domain name. Many times the hosting company itself maintains the information but there are many companies who solely provide DNS services. It gives an affirmation that the domain name is related to the IP address of the computer that has the website hosted on. It is a way for the domain names to guide us to the right computer. Now that you know what IP address, domain names and DNS plays a role, let us look at the last step that is hosting.

Hosting

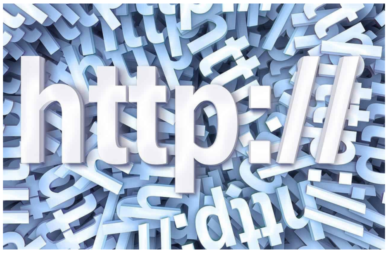
Now that you have your IP address, the domain name in place it is time to host it and make it reachable. In simple words, hosting is when you pay a company to put your website files on their web server. Your website's files consist of HTML files, images, videos, photos, etc. To put it all in place, when your **domain name** is put on the web browser, the DNS is connect it to the right computer/server and then that particular computer has your website hosted on it.

Keyword

Domain name is an identification string that defines a realm of administrative autonomy, authority or control within the Internet.

7.1.3 How Domain Name Works

A domain is the name of your website. Before you can create a website, you will need a domain. A domain name is not something physical that you can touch or see; it is merely a string of characters that give your website an identity (yes, a name, like human and businesses). Now, here are some quick examples: Google.com is a domain name; so are Alexa.com, Linux.org, eLearningEuropa.info, as well as Yahoo.co.uk. To have your own domain, you will need to register your domain with a domain registrar.



How Domain Name Works?

Top Level Domains (TLDs)

Alexa.com, Linux.org, eLearningEuropa.info, and Yahoo.co.uk – these domains end with a different 'extension', namely: .com, .org, .info, .co.uk. We call this "extension" as top level domain (shortform: TLD). Examples of other TLD include .uk, .ws, .co.jp, .com.sg, .tv, .edu, .co, .com.my, and .mobi. While most of these TLDs are open for public's registration, there are strict regulations on certain domain registration. For example the registration of country code top level domains (like .co.uk for United Kingdom) are restricted for the citizens of the corresponding country; and the activities with such domains website are ruled by local regulations and cyber laws.

Certain extensions of these TLDs are used to describe the 'characteristics' of the website – like .biz for businesses, .edu for education (schools, universities, colleagues, etc.), .org for public organization, and country code top level domain names are for locations.

Country code top-level domain

The full list of country code top-level domain (ccTLD) extensions are (in alphabet order):

.ac .ad .ae .af .ag .ai .al .am .an .ao .aq .ar .as .at .au .aw .ax .az .ba .bb .bd .be .bf .bg .bh .bi .bj .bm .bn .bo .br .bs .bt .bw .by .bz .ca .cc .cd .cf .cg .ch .ci .ck .cl .cm .cn .co .cr .cu .cv .cx .cy .cz .de .dj .dk .dm .do .dz .ec .ee .eg .er .es .et .eu .fi .fj .fk .fm .fo .fr .ga .gd .ge .gf .gg .gh .gi .gl .gm .gn .gp .gq .gr .gs .gt .gu .gw .gy .hk .hm .hn .hr .ht .hu .id .ie .il .im .in .io .iq .ir .is .it .je .jm .jo .jp .ke .kg .kh .ki .km .kn .kp .kr .kw .ky .kz .la .lb .lc .li .lk .lr .ls .lt .lu .lv .ly .ma .mc .md .me .mg .mh .mk .ml .mm .mn .mo .mp .mq .mr .ms .mt .mu .mv .mw .mx .my .mz .na .nc .ne .nf .ng .ni .nl .no .np .nr .nu .nz . om .pa .pe .pf .pg .ph .pk .pl .pn .pr .ps .pt .pw .py .qa .re .ro .rs .ru .rw .sa .sb .sc .sd .se .sg .sh .si .sk .sl .sm .sn .sr .st .sv .sy .sz .tc .td .tf .tg .th .tj .tk .tl .tm .tn .to .tr .tt .tv .tw .tz .ua .ug .uk .us .uy .uz .va .vc .ve .vg .vi .vn .vu .wf .ws .ye .za .zm .zw

Top 3 Benefits of Country Code Top-Level Domains

And that's not all. We now have more than 1,000+ generic TLDs (gTLD) opened to public, including .BAR, .BARCELONA, .BUILD, .FOREX, .CLUB, .COLLEGE, .REST, .WEBSITE, .WIEN, .XYZ, and so on. You can find the full list of top-level domains in the Root Zone Database.

Domain vs sub-domain

Take mail.yahoo.com for example – yahoo.com is the domain, mail.yahoo.com in this case, is the sub domain. A domain must be unique (for example there can only be one single Yahoo.com) and must be registered with a domain registrar (i.e. NameCheap); while for sub domains, users can freely add it on top of the existing domain as long as their web host provide the service. Some would say sub-domains are the 'third level' domains in the sense that they are simply "sub folders" under the domain root directory, normally used to organize your website content in different languages or different categories. However, this is not the case to many including the search engines – it is known fact that the search engines (namely, Google) treat sub domain .as a different domain independent from the primary domain

Terms of domain name

To quickly recap on what we have just learned: –

Website Domain	Name	Subdomain	TLD	ccTLD
yahoo.com	Yahoo	–	.com	–
mail.yahoo.com	Yahoo	mail	.com	–
finance.yahoo.com	Yahoo	finance	.com	–
yahoo.co.jp	Yahoo	–	–	.co.jp

7.1.4 The Difference between Web Hosting, Website and Domain

When you have a site visitor, they use your domain name to view your website. As simple as it seems, there is a process that occurs from the time that the site visitor types in your domain and presses enter. When a site visitor enters your domain name into a browser, the domain is then translated into your server IP address, then the server sends that user your site files, which their browser represents to them as a typical web-page. You can see that, without each of these 3 elements, you will not really have a web-site.

Web Hosting and Email Services

Remember

Dreamweaver would ask before updating all the pages. And, any changed pages would then have to be uploaded to the server.

The three basic parts that make up any current day website are:

- Domain Names
- Web-Hosting Servers
- Site Files

Domain Names

Computers communicate by using numbers, called IP addresses, to contact each other, much like you use a phone number to dial a specific person's phone. Domain names on the internet are much like entries in a phone book. The phone book tells people looking for a business what the entries are just as a domain tells people (i.e. their computers) that a domain is hosted on the server. Without a domain you would not be able to use a domain name such as mysite.com, making your site appear unprofessional and impractical if your visitors are able to load it at all.

Web-Hosting Servers

The web-hosting or server is much like the space that you rent out to have your business in. It is merely the space itself. It does not include furnishings like shelves for your products, just as the web-hosting account does not include a site for you to sell your products. Luckily, in the web-hosting world, it is very easy to furnish the space provided by your host, because you can install many framework applications through the QuickInstall icon within your cPanel. Without the hosting services, you will not have a place for your files to reside, so your domain would then become like a disconnected phone number in the phone directory, and your site files would have nowhere to stay.

Site Files

The site files are what your visitors and potential customers actually see when going to site such as your products and services. The site files are the same as any other file you normally use, like a .jpg photograph, or .mp3 music file. Though, website files are also .php files or .html files, which are **PHP** scripts or html pages respectively. The web-hosting server knows how to read these files, which explain how the webpage looks or instruct the server to do a series of computations. These computations are things like figuring out what blog article it is supposed to send back to the viewer, or what forum post it is supposed to send back.

Keyword

PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

7.1.5 Domain Name Registration Process

In order to reserve a domain name in a gTLD, a domain name registrant must register it with an ICANN-accredited registrar. The registrar will check if the domain name is available and create a WHOIS record with the domain name registrant's information. It is also possible to register domain names through a registrar's resellers. The diagram below illustrates the main functions of the parties that are usually involved in the process.

A domain name registrant is the person or organization who has registered the domain name. In order to do so, the domain name registrant will usually apply online to a domain registrar or one of their resellers. The domain name registrant is bound by the terms and conditions of the registrar with which it registers its domain name, for instance adhering to a certain code of conduct or indemnifying the registrar and registry against any legal or civil action taken as a result of use of the domain name. Domain name registrants have certain responsibilities that are incorporated into these terms and conditions like payment of registration fees and submission and timely update of accurate data.

In addition to registering the domain name, domain name registrants also need to have their domain names listed on name servers in order for that domain name to be reachable on the Internet. A domain name registrant is responsible for procuring or hosting his or her own name server if the registrar does not offer this service or he or she has opted out of the registrar's service. In some cases, a person or organization who does not wish to have their information listed in WHOIS may contract with a proxy service provider to register domain names on their behalf. In this case, the service provider is the domain name registrant, not the end customer.

Registrars are organizations accredited by ICANN and certified by the registries to sell domain names. They are bound by the Registrar Accreditation Agreement (RAA) with ICANN, and by their agreements with the registries. The RAA sets out responsibilities for the registrar including maintenance of WHOIS data, submission of data to registries, facilitating public WHOIS queries, ensuring domain name registrants details are escrowed, and complying with RAA conditions relating to the conclusion of the domain name registration period.

Some domain name registrants may opt to register through a reseller. These organizations are affiliated or under contract with registrars, and usually offer other services such as web hosting, email mailboxes etc. Resellers are bound by their agreements with the registrar(s) whose services they sell; they are not accredited by ICANN. However, the registrar for whom they are re-selling will still be the sponsor for the domain name registration and accountable for the domain names sold by the reseller.

While registrars are contracted to conduct the day-to-day business of selling domain name registrations, registries are responsible for maintaining the registry for each TLD. The responsibilities of the registries include accepting registration requests (whether from registrars or directly from domain name registrants), maintaining a database of the necessary domain name registration data and providing name servers to publish the zone file data (i.e. information about the location of a domain name) throughout the Internet. The Internet Corporation for Assigned Names and Numbers (ICANN) is the non-profit organization that oversees the assignment of both IP addresses and domain names. It has responsibility for managing root server and TLD name system management and has contractual agreements with both registries and registrars that provide the foundation for the WHOIS system.

7.2 CREATING WEBSITE STRUCTURES

A website's structure refers to how the website is set up, i.e. how the individual subpages are linked to one another. It is particularly important that crawlers can find all subpages quickly and easily when websites have a large number of subpages. Are you starting a new website project? Well, the first thing you should do is choose your website structures. If you do not already have a basic idea of how you will present the information on your site, it will be difficult, if not impossible, to organize it. Website structures are the first step to building incredible information architecture, something that separates good websites from awesome experiences.

You will want to invest time in selecting the right website structure now because it can help you in the future. Trends and customer needs can quickly change, and websites often need updating. If you want to avoid costly do-overs later down the road, you will want to choose a structure that will last over time. Choosing the right one from the start is the best way to do this; however, you should leave room for change over time. This section will not only tell you what **website** structures are and how they benefit you, but it will also explain how to choose the best one for your project.

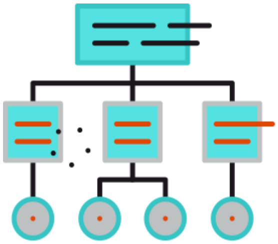
7.2.1 The Four Types of Website Structures

If you have ever been tasked with building a website from scratch, you probably already know how difficult it can be to know where to start. Many newbie designers might opt for beginning with a theme, however when you are building custom websites, and larger sites, a prebuilt theme may not cut it. All websites have a basic organizational structure that falls into one of four types. These website structures (or a combination of them) can help you begin to organize a site of any size.

Keyword

Website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server.

Hierarchical Model



These are possibly the most common types of website structures. They start with a broad set of information (parent pages) that filters down into more detailed information (child pages). Sometimes these structures are called trees, and they are very similar to organizational charts in corporations.

Sequential Model

These types of structures are exactly like they sound – they lead site visitors through a sequence. Whereas a hierarchical structure might lead site visitors down or across to another parent page, sequential structures only lead visitor either backward or forward from one step to another.

Matrix Model

Although this structure may be nontraditional, in the early years of the internet, it was quite popular. A matrix type structure lets site visitors choose where they would like to go next. Instead of building a sequence, or limiting navigation with parent/child relationships, this structure provides many links under topic groups with those who land on the page choosing where they go next.

Database Model

This dynamic approach to website structuring integrates a database with search. To build a site like this, you will need to think from the bottom up – carefully tagging your content's metadata based on information architecture principles. When done correctly, this structure produces a site where visitors can create experiences based on what they are looking for.

7.2.2 Why you should start with the Site Structure

Now that you know a little about the basic types of website structures, it is time to discuss how using them can help you. Besides making it easier to organize any website project, utilizing site structures can also improve usability. The process of creating a site structure forces a designer to think about how visitors will navigate through a website. This is a stark difference from merely thinking about what will be placed on the site. Whether you use a straightforward sequential model or build a complicated matrix, you will start by thinking about the how rather than simply the what.

Travel Blogging Guide for Beginners + Printable Checklist

Website structures explain how users navigate through a site. This impacts usability because much of usability is based on how easily website users can find their way through a site. A highly usable site is easy to navigate (in addition to other factors). Website structures do not guarantee high usability, but they certainly help with it.

You can also use your site structure to create themes, which can come in handy if you build the same types of website projects over and over again. While you could work from your last site to quickly make something similar, it is much easier to create a general theme that you can customize as needed. Perhaps you can even sell your themes online for newbie designers that need help structuring their sites. Choosing your site structure, however, is the only way to start.

Site structures also benefit larger websites, since they often need special attention when it comes to navigation. When there is a lot of content to be shared, it is incredibly easy to become overwhelming. By choosing a website structure that is most appropriate for your organization, you can reduce website visitor fatigue, and help keep them on the site longer.

7.2.3 How to Choose your Best Website Structure

To put it simply, website structures help you build better sites. So, the next logical question for most people is, "how do you know which structure to use?" Here is when it gets a bit more complicated. You start by understanding who you are building the website for.

How to make a website. Step-by-step guide

Site Structure and Audience

As we mentioned above, certain sites might benefit from a mix of website structures based merely on their size. However, you may also want to consider who will be reading the site (in addition to how it will be used) when choosing your site structure.

Hierarchical

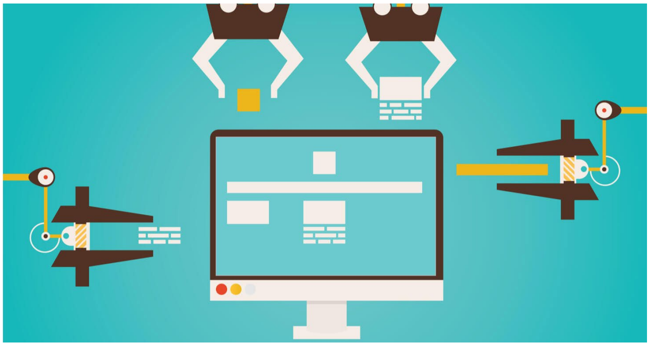
For most website users the hierarchical model works just fine. For this structure, information is ranked and presented in logical order. One reason most people appreciate this type of organization is because it is currently the most popular model online, making it easy to navigate simply because it is familiar. Hierarchical structures are also easy to adjust to fit many scenarios. You can go for a simple **hub** and spoke style or build more complex hierarchies with subheadings based on the intended uses and user. Simple hierarchies also work great on mobile sites, where the options are limited to a single page and a few links. The trick to choosing a structure that will stand the test of time is creating one that is not too shallow or deep. Too many (or too few) .subheadings is a bad thing

Keyword

Hub is the central part of a wheel that connects the axle to the wheel itself.

Sequential

Sequential sites are often associated with educational sites, and this is no surprise since this structure is based on the style used by print publications. If you are presenting content that teaches, naturally occurs in a logical order (such as alphabetical), or is intended to be given chronologically, a sequential structure is probably best. It should also go without saying, that the intended audience is those looking to learn a significant amount .of information that is presented over multiple pages



Creating and monetizing a travel website

Although sequential structures follow a clearly defined order and work best on smaller sites, there is room to customize these to fit the needs of larger sites. Utilizing some of the principles of hierarchical structures, programmers can add digressions to this site structure. These “subpages” leave room to add supporting pages of information without letting the site user go off-track from their goal. No one likes clicking a link only to get lost and not know how to get back (even if intentional). Digressions let site visitors wander, but not too far.

Matrix and Database

If there were one phrase to summarize this type of website structure, it would be: "Enter the matrix, if you dare." Not only are matrix and database structures more difficult to organize effectively, but they are also tricky to navigate if you do not already know what you are looking for. The matrix and database structure build upon large amounts of information and presents it either as a web of data or a simplified input-based interface. Those who enjoy utilizing associative thought processes or do not mind being given a large amount of information at one time, might enjoy these types of setups.

In web design, a matrix is often seen as a collection of links to ideas, or a word cloud of topics that takes visitors to where they need to be. Database designs rely on a search parameter and pull related information for the user. These types of sites can deliver mind-blowing amounts of data to users; so you must be careful to consider how much is too much. When there are a large number of topics to choose from with little to no organization, or no related information can be found from a search, many people become overwhelmed and leave. This likelihood to cause confusion and miss connections is the main reason why the matrix and database structure is reserved for smaller sites, those with readers that have a high amount of education around the topic already or organizations that can invest in advanced filtering and search.

7.2.4 Difference between Web Hosting and Web Publishing

Web hosting refers to a service that is associated with internet hosting. The companies that offer this service, provide their server computers to store the web sites. Web publishing refers to the process in which the content is published on the internet. It simply refers to the placement of webpages in a location that is accessible to users. It generally includes creation of websites which are later uploaded using a web server.

Many people do not understand the differences between web hosting and publishing as both seem to be connected and interrelated. A web hosting service is a kind of service through which one gets space on a server for his/her website. This space can be owned by some other party. The facility for a leased connection may also be available for providing the space. Thus, we can also state that the space is generally owned or leased to be used by the clients. The facility of web hosting may also include the service of internet connectivity.

Web publishing is just like any other normal publishing. In context to web, it deploys the websites using a web host. One can also publish a website from the folder of his/her computer. It simply refers to the placement of web pages in a location that is easily accessible to users. However, publishing from the home computer can be troublesome as your computer needs 24X7 connectivity to the internet. Therefore, the websites are generally published using a web server. The web site pages are transferred to the web server. If the company has its own web server than it publishes its own website using its own host. However, if the company does not have a **web server**, then it needs the services of a web hosting provider. Web hosting providers are companies that provide the space in order to publish the web pages on the internet. We can relate both the terms by stating that websites are published using the infrastructure provided by the web hosting companies.

ROLE MODEL

VINTON CERF: AMERICAN COMPUTER SCIENTIST

Vinton Cerf, in full Vinton Gray Cerf, (born June 23, 1943, New Haven, Connecticut, U.S.), American computer scientist who is considered one of the founders, along with Robert Kahn, of the Internet. In 2004 both Cerf and Kahn won the A.M. Turing Award, the highest honor in computer science, for their "pioneering work on internetworking, including the design and implementation of the Internet's basic communications protocols, TCP/IP, and for inspired leadership in networking."

In 1965 Cerf received a bachelor's degree in mathematics from Stanford University in California. He then worked for IBM as a systems engineer before attending the University of California at Los Angeles (UCLA), where he earned a master's degree and then a doctorate in computer science in 1970 and 1972, respectively. He then returned to Stanford, where he joined the faculty in computer science and electrical engineering.

While at UCLA, Cerf worked under fellow student Stephen Crocker in the laboratory of Leonard Kleinrock on the project to write the communication protocol (Network Control Program [or Protocol]; NCP) for the ARPANET (Advanced Research Projects Agency Network; *see* DARPA), the first computer network based on packet switching, a heretofore untested technology. (In contrast to ordinary telephone communications, in which a specific circuit must be dedicated to the transmission, packet switching splits a message into "packets" that travel independently over many different circuits.) UCLA was among the four original ARPANET nodes. Cerf also worked on the software that measured and tested the performance of the ARPANET. While working on the protocol, Cerf met Kahn, an electrical engineer who was then a senior scientist at Bolt Beranek & Newman. Cerf's professional relationship with .Kahn was among the most important of his career

In 1972 Kahn moved to DARPA as a program manager in the Information Processing Techniques Office (IPTO), where he began to envision a network of packet-switching networks—essentially, what would become the Internet. In 1973 Kahn approached Cerf, then a professor at Stanford, to assist him in designing this new network. Cerf and Kahn soon worked out a preliminary version of what they called the ARPA Internet, the details of which they published as a joint paper in 1974. Cerf joined Kahn at IPTO in 1976 to manage the office's networking projects. Together, with many contributing colleagues sponsored by DARPA, they produced TCP/IP (Transmission Control Protocol/Internet Protocol), an electronic transmission protocol that separated packet error checking (TCP) from issues related to domains and destinations (IP).

Cerf's work on making the Internet a publicly accessible medium continued after he left DARPA in 1982 to become a vice president at MCI Communications Corporation (WorldCom, Inc., from 1998 to 2003). While at MCI he led the effort to develop and deploy MCI Mail, the first commercial e-mail service that was connected to the Internet. In 1986 Cerf became a vice president at the Corporation for National Research Initiatives, a not-for-profit corporation located in Reston, Virginia, that Kahn, as president, had formed to develop network-based information technologies for the public good. Cerf also served as founding president of the Internet Society from 1992 to 1995. In 1994 Cerf returned to MCI as a senior vice president, and from 2000 to 2007 he served as chairman of the Internet Corporation for Assigned Names and Numbers (ICANN), the group that oversees the Internet's growth and expansion. In 2005 he left MCI to become vice president and "chief Internet evangelist" at the search engine company Google Inc. In addition to his work on the Internet, Cerf served on many government panels related to cybersecurity and the national information infrastructure. A fan of science fiction, he was a technical consultant to one of author Gene Roddenberry's posthumous television projects, *Earth: Final Conflict*. Among his many honors were the U.S. National Academy of Engineering's Charles Stark Draper Prize (2001), the Prince of Asturias Award for Technical and Scientific Research (2002), the Presidential Medal of Freedom (2005), the Queen Elizabeth Prize for Engineering (2013), and the French Legion of Honor (2014).

SUMMARY

- Web hosting is a method by which a person or company rents a server to store data used to display a website, which is accessible through the Internet.
- Web hosting is an essential service for anyone that wants to set up a website. Without a web host, you cannot share your website with any users. It basically provides all of the services and technology that is required to get your website onto the internet including storage space/servers.
- Servers are powerful computers that have extremely large hard drives, or an array of hard drives, which can be rented to those who want a website on the Internet.
- Shared hosting is one of the more common methods for website hosting. It is also the cheapest, and least reliable. This is when several website owners are sharing a single server including all resources such as CPU, memory and drive space.
- A Virtual Private Servers, or VPS, is a virtual machine running on top of a real machine. Several VPS servers may run on a single real server machine.
- The IP address is essential because there are many computers on the same location that is called LAN (Local Area Network) and many LANs are connected together that is called WAN (Wide Area Network), therefore, each should have a unique identity.
- Domain names are considered as the second name for IP address. It is difficult to remember these number and domain names are simpler to remember and understand. It is important to get a domain name for the website and acts as an address.
- DNS stands for Domain Name Server or Domain Name System. This step is known to be the hardest of all. It basically keeps a record of the information and is in charge of it i.e.
- Computers communicate by using numbers, called IP addresses, to contact each other, much like you use a phone number to dial a specific person's phone. Domain names on the internet are much like entries in a phone book.
- A website's structure refers to how the website is set up, i.e. how the individual subpages are linked to one another. It is particularly important that crawlers can find all subpages quickly and easily when websites have a large number of subpages.

KNOWLEDGE CHECK

1. **A service provider that manages the server used to host an organization website and its connections to the Internet backbones is known as:**
   a. Client server
   b. Internet service provider
   c. Hosting provider
   d. All of the above
2. **The website for a company is hosted on a:**
   a. Web infrastructure
   b. Web page
   c. Web server
   d. Web client
3. **The website content of a company is accessed by an end-user through using:**
   a. Web client/server
   b. Web server
   c. Web e-mail
   d. Web browser
4. **A provider enabling home or business users access and web hosting services for the Internet is referred to as:**
   a. Application service providers
   b. Internet solutions providers
   c. Internet service providers
   d. Online solutions providers
5. **The likelihood of a poor response from a website can be reduced through:**
   a. Selecting a dedicated server
   b. Signing a service level agreement
   c. Selecting high bandwidth connection
   d Both the second and third answer above
6. **The data storage space accessed via the Internet, usually used to host websites and data files is called......................**
   a. Web server
   b. Web page
   c. Webmaster
   d. Web space

7. **The process of moving data from one storage device to another is ....**
   a. Bandwidth
   b. Migration
   c. Server
   d. Router
8. **Web hosting based on type of servers are.**
   a. UNIX WINDOWS DOS
   b. Dedicated Hosting Shared Hosting VPS Hosting
   c. Web page Web site Web browser
   d. Server Web server Webmaster
   e. Web Hosting Email Hosting Script hosting
9. **Web hosting service where more than one web site is hosted on the same server.**
   a. VPS Hosting
   b. Windows Hosting
   c. Dedicated hosting
   d. Shared hosting
10. **An HTML/XHTML document that is available through the World Wide Web and can be accessed through a web browser is called as ....**
    a. Webmaster
    b. Web server
    c. Web page
    d. Computer
    e. Home page

REVIEW QUESTIONS

1. What do you understand by web hosting?
2. Define the different types of web hosting services.
3. What is the difference between domains vs hosting vs website?
4. Describe the domain name registration process.
5. Write the four types of website structures.
6. Differentiate between web hosting and web publishing.

Check Your Result

1. (c)	2. (c)	3. (d)	4. (c)	5. (d)
6. (d)	7. (b)	8. (b)	9. (d)	10. (c)

REFERENCES

1. A History of Web Hosting [Infographic]. BizTech. 2012-02-24. Retrieved 2016-11-04.
2. Buyya, Rajkumar; Yeo, Chee Shin; Venugopal, Srikumar (2008). "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities". 2008 10th IEEE International Conference on High Performance Computing and Communications. pp. 5–13.
3. Dawson, Christian. "Why Uptime Guarantees are Ridiculous". Servint. Retrieved 7 October 2014. a good SLA will clearly state how uptime is defined and what you'll receive if the "uptime promise" is not met.
4. Dawson, Christian. "Why Uptime Guarantees are Ridiculous". Servint. Retrieved 7 October 2014. a good SLA will clearly state how uptime is defined and what you'll receive if the "uptime promise" is not met.
5. InstantShift. "A Guide to Web Hosting Security Issues and Prevention". InstantShift - Web Designers and Developers Daily Resource. Retrieved 2016-10-31.
6. InstantShift. "A Guide to Web Hosting Security Issues and Prevention". InstantShift - Web Designers and Developers Daily Resource. Retrieved 2016-10-31.
7. Intark Han; Hong-Shik Park; Youn-Kwae Jeong; Kwang-Roh Park (2006). "An integrated home server for communication, broadcast reception, and home automation". IEEE Transactions on Consumer Electronics. 52: 104–109.
8. Schultz, Eugene (2003). "Attackers hit Web hosting servers". Computers & Security. 22 (4): 273–283.
9. Vanvleet, Derek. "Google Apps vs. Office 365: What Does 99.9% Uptime Really Mean?". Cloud Sherpas. Retrieved 7 October 2014.
10. Ward, Mark (3 August 2006). "How the web went world wide". BBC News. Retrieved 24 January 2011.

CHAPTER

8

JAVASCRIPT AND EXTERNAL CONTENT

"JavaScript is the only language that I'm aware of that people feel they don't need to learn before they start using it."

– Douglas Crockford

LEARNING OBJECTIVES

After studying this chapter, you will be able to:

1. Understand the canvas
2. Disuses the briefly of javascript
3. Define javascript events and jQuery
4. Explain the HTML5 <canvas> tag

INTRODUCTION

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers. This chapter also discusses the new HTML5 tag and other HTML tags for including external content within your Web page.

8.1 THE CANVAS

One of the most exciting elements added in HTML5 (and there are many) is the canvas element. The canvas element enables advanced **graphics** and interactions in ways that previously you could only achieve by using slow, plug-in–laden pages. The canvas element is a free-form area where you can use images, drawn graphics, animation, and text to enhance the user experience on a Web page.

Keyword

Graphics are visual images or designs on some surface, such as a wall, canvas, screen, paper, or stone to inform, illustrate, or entertain.

You add a canvas element to a Web page with the aptly titled tag. Like other HTML tags, the tag accepts attributes, the most basic of which are the pixel height and width of the canvas on the page. The code to create a 200 × 200 pixel canvas looks like this:

```
<canvas width=”200” height=”200”></canvas>
```

Here’s the syntax for an entire page with the 200 × 200 canvas inside:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html;
charset=utf-8”>
<title>Canvas</title>
</head>
<body>
<canvas width=”200” height=”200”></canvas>
```

```
</body>
</html>
```

This HTML creates a canvas on the page that is ready for content. But wait! What advantage does a canvas element provide on a page? How do you make a cool animated clock or menu or other widget? Alas, therein lies the problem. When simply included in a page like the one shown, a tag doesn't do anything. In fact, it's just blank; go ahead and load that code into a canvas-compatible browser.

So, how might you get the canvas to do something? The answer lies in the programming language that all modern browsers understand: JavaScript. You bring the canvas element to life with JavaScript; everything within the canvas is drawn, animated, and destroyed with JavaScript. You might not know much, if anything, about JavaScript. After learning a bit about JavaScript you'll be able to make the canvas element do some fun and interesting things!

8.2 BRIEFLY OF JAVASCRIPT

JavaScript is not Java; it bears no relation to the Java programming language, the drink, or the country. JavaScript is a programming language that is used primarily to provide additional functionality to Web pages and applications, and it's used heavily in the "Web 2.0" paradigm of highly interactive Web sites.

Just as HTML is codified through standards, JavaScript is also based on a standard called the ECMA-262 specification. The latest version of the specification, version 5, was released in 2009. Unfortunately, different Web browsers implement the ECMA-262 specification differently—which means that JavaScript programmers must take care to work around the quirks and differences in the various browser implementations. One approach to provide JavaScript that works the same way across all popular browsers is to use a library or framework. One popular framework, jQuery, enables rapid development and use of JavaScript without the need for you to learn the intricacies and nuances involved for the different Web browsers. jQuery also simplifies many common JavaScript tasks for Web developers.

Did You Know?

JavaScript is the dominant client-side scripting language of the Web, with 97% of websites using it for this purpose.

8.2.1 Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

8.2.2 Syntax

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.

You can place the <script> tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the <head> tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

```
<script ...>
 JavaScript code
</script>
```

The script tag takes two important attributes:

- Language: This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.

- Type: This attribute is what is now recommended to indicate the scripting language in use and its value should be set to “text/javascript”.

So your JavaScript syntax will look as follows.

```
<script language=”javascript” type=”text/javascript”>
 JavaScript code
</script>
```

Your First JavaScriptCode

Let us take a sample example to print out “Hello World”. We added an optional HTML comment that surrounds our JavaScript code. This is to save our code from a browser that does not support JavaScript. The comment ends with a “//-->”. Here “//” signifies a comment in JavaScript, so we add that to prevent a browser from reading the end of the HTML comment as a piece of JavaScript code. Next, we call a function document.write which writes a string into our HTML document.

This function can be used to write text, HTML, or both. Take a look at the following code.

```
<html>
<body>
<script language=”javascript” type=”text/javascript”>
<!--
 document.write (“Hello World!”)
//-->
</script>
</body>
</html>
```

Remember

Care should be taken while writing variable and function names in JavaScript.

This code will produce the following result:

```
Hello World!
```

8.2.3 Including JavaScript on Your Web Page

To use JavaScript on your page, you include a <script> tag. Specifically, the opening tag you use is as follows:

```
<script type=”text/javascript”>
```

And the closing tag is this:

```
</script>
```

The magic happens between the opening and closing tags, which is where you place the JavaScript code. The <script> tag also frequently includes a src attribute that specifies that the page should include an external JavaScript file. For example, if you had a file containing JavaScript code called “myjavascript.js”, you could include it like this:

```
<script type=”text/javascript” src=”myjavascript.js”></script>
```

Note that you still need to include the closing </script> tag when you’re including an external JavaScript file, as in the example just shown.

8.2.4 Your First JavaScript Web Page

You’ll see how to use JavaScript within your Web page.

SET UP

1. In Notepad, type the following:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=utf-8”>
<title>JavaScript 101</title>
</head>
<body>
<div id=”contentDiv”>Your first JavaScript page.</div>
</body>
</html>
```

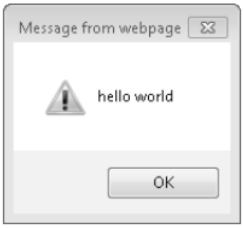
2. Add some JavaScript to the page, placing it just before the closing

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=utf-8”>
<title>JavaScript 101</title>
</head>
```

```
<body>
<div id="contentDiv">Your first JavaScript page.</div>
<script type="text/javascript">
 alert("hello world");
</script>
</body>
</html>
```

3. Save the file in Notepad as javascript01.html.

4. View the page within a Web browser. You should receive an alert such as this:



hello world

SET UP

1. Delete the following line from the code:

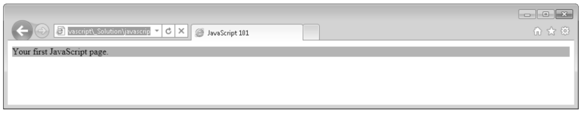
```
alert("hello world");
```

2. Between the <script> and </script> tags, place the following:

```
<script type="text/javascript">
 var divContent = document.getElementById("contentDiv");
 divContent.style.backgroundColor = "#abacab";
</script>
```

3. Save your work in Notepad.

4. Open the file in Internet Explorer to test it. Instead of a Hello World **dialog box**, the text on the page appears with a gray background:



Your first JavaScript page.

Keyword

Dialog box (also spelled dialogue box, also called a dialog) is a common type of window in the GUI of an operating system.

8.3 JAVASCRIPT EVENTS AND JQUERY

Most modern Web sites use JavaScript to dynamically respond to mouse actions on a Web page and change elements of that page based on those movements or on other user input. To introduce event handling at this very early stage in your JavaScript adventure, called jQuery. jQuery is an open source JavaScript file that not only removes the need for developers to handle many of the cross-browser incompatibilities but also simplifies much more advanced programming than would normally be available to novice JavaScript programmers. The jQuery JavaScript framework is an excellent tool for both working with JavaScript events and for all-around general JavaScript programming.

8.3.1 Obtaining jQuery

jQuery is a single file, and you should place it in the document root, or main folder, of your Web site (or wherever you place JavaScript files in your environment). jQuery comes in two forms, a development version and a production version. The production version is “minified,” meaning that it’s been optimized for speed (the download size is smaller), but that makes it very hard to use. The development version is not minified, so you can read the code more easily.

As of this writing, jQuery was at version 1.4.4. That version’s downloaded file is called “jquery-1.4.4.min.js.” You include this file in your Web page just like any other external JavaScript file, using the <script> tag.

```
<script type=”text/javascript” src=”jquery-1.4.4.min.js”></script>
```

Here’s a completed example page (although it doesn’t do anything) that includes jQuery:

```
<!DOCTYPE html>
<head>
<script type=”text/javascript” src=”jquery-1.4.4.min.js”></script>
<title>Including jQuery</title>
</head>
<body>
</body>
</html>
```

It’s also possible to use jQuery hosted on a Content Delivery Network (CDN). With the CDN-based jQuery, you can simply point the <script> tag toward the URL of the library on the external CDN. Doing so looks like this:

```
<script type=”text/javascript”
 src=”http://code.jquery.com/jquery-1.4.4.min.js”></script>
```

However, as just noted, it’s recommended to host the file yourself for production purposes on live sites rather than relying on the CDN. There’s nothing worse than having your Web site up and operational but reliant on a CDN-based library Web site that is down. It’s perfectly acceptable to use the CDN based version rather than downloading jQuery yourself.

8.3.2 Getting Ready for jQuery

Imagine this problem: You’ve written some JavaScript and included it just before the closing < /body> tag. One of the things that your JavaScript does is change an image on the page. The problem is that your image hasn’t loaded, so now the JavaScript runtime can’t find it, and the page loads incorrectly.

The root cause of this all-too-common problem is that browsers execute JavaScript as they encounter scripts on the page. So even though the JavaScript is at the bottom of the page, the browser may not have loaded the entire page prior to running the JavaScript. In other words, the document isn’t ready by the time the JavaScript runs, so chaos ensues.

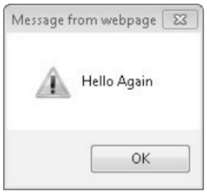
A workaround for this problem is to use an onload or load event on the page, but even that is fraught with danger, not to mention it’s bad practice. Luckily, jQuery includes a function called .ready() which executes only after the page has been loaded by the browser and is ready for JavaScript code to be run. The .ready() function is a simple way to execute JavaScript while safely knowing that all elements of the page have been loaded and are ready to use. jQuery code begins with a dollar sign and parentheses, as you’ll see in the upcoming example.

Using the .ready() function is easy. The following code shows an example of the .ready() function in action.

```
<!DOCTYPE html>
<html>
<head>
<title>Document Ready</title>
<script type=”text/javascript” src=”jquery-1.4.4.min.js”></script>
</head>
<body>
<script type=”text/javascript”>
$(document).ready(alert(‘Hello Again’));
```

```
</script>
</body>
</html>
```

When viewed in a browser, you'll receive an alert like this:



Message from webpage

Hello Again

The code shown above includes the jQuery library, and then it uses both the .ready() function and some other JavaScript to show an alert. This is an important point about jQuery: you use it to help write JavaScript. jQuery is not JavaScript; instead, it's a tool that you use to help perform common JavaScript tasks and sometimes to simplify tasks that are difficult to perform using JavaScript alone.

You've now seen how to get jQuery, how to include it in a Web page, and how to run JavaScript code with the help of jQuery's .ready() function. The next item on the agenda is selecting elements.

8.3.3 Selecting Elements with jQuery

jQuery has its own syntax for selecting elements such as <p>, <img>, <div>, and so on. Recall that the example HTML code included a <div> element with an id attribute of contentDiv, as shown here:

```
<div id=”contentDiv”>Your first JavaScript page.</div>
```

jQuery makes it easy to select that element using jQuery, using this syntax:

```
$(“#contentDiv”)
```

Alternatively, you could select all the <div> elements with this syntax:

```
$(“div”)
```

You can also select elements by their cascading style sheet (CSS) class with a dot prefix, similar to that used in the CSS file itself:

```
$(“.className”)
```

jQuery provides several other ways to select elements, including hierarchical functions by which you can select the children of an element; a way to select every other element except a specified element; an element's parent elements; and many other selectors.

What can you do with a selected element? The answers are virtually limitless. Here's how you would do the same thing using jQuery:

```
$("#contentDiv").css("backgroundColor","#abacab");
```

Here's a bonus example using the jQuery fadeOut() function:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>jQuery 101</title>
<script type="text/javascript" src="jquery-1.4.4.min.js"></script>
</head>
<body>
<div id="contentDiv">Your second JavaScript page.</div>
<script type="text/javascript">
 $("#contentDiv").css("backgroundColor", "#abacab");
 $("#contentDiv").fadeOut();
</script>
</body>
</html>
```

Viewing this page in a browser will result in the contentDiv fading out after a certain default (and short) time period. If you find that time period too short, you can specify the duration, as well. Time is measured in milliseconds for this (and most other) functions in jQuery, so every second is 1000 milliseconds. Therefore, to set the fade-out duration to 5 seconds, you would write:

```
$("#contentDiv").fadeOut(5000);
```

You've now seen how to download jQuery, connect it to your page, and use it to select elements. And you just looked at a bonus example of a built-in function in jQuery called .fadeOut(). This leads to a more generalized discussion of functions in jQuery and Java Script.

8.3.4 Calling Functions with JavaScript

Functions are groupings of code that perform a task. Here's a function:

```
function doSomething() {
 alert("Hello World");
}
```

That's it, that's all there is to functions. Well, almost. But there's no reason to clutter the discussion of functions when a simple example will suffice. The .fadeOut() example gave you a glimpse at another important part of a function: a function argument. A function argument is a value that is passed to the function that determines how or what the function should do as it carries out its designed task. The .fadeOut() function uses the duration argument (passed as 5000 in the example at the end of the preceding section) to set the length of time that the function waits before it fades the element out.

For example, here's a showAlert() function that accepts a single argument called alertText, and then shows it in an alert dialog box:

```
function showAlert(alertText) {
 alert(alertText);
}
```

Calling or invoking the function looks like this:

```
showAlert("Showing an alert is fun and easy.");
```

One other important aspect of functions is that they can return a value. Typically, the return value would be the result of whatever the function accomplishes, although the return value can be whatever you'd like it to be. For now, you'll work on a typical example, where the return value is the logical result of the function. In this next example, the function adds two numbers and returns the result.

```
function addTwo(num1, num2) {
 var result = num1 + num2;
 return result;
}
```

You call the function the same way as in the previous example, but this time, the function returns a value that you want to capture so that you can use it later, as shown here:

```
var getSum = addTwo(2,5);
```

With this code, the sum of the two numbers, 2 and 5, would be placed into the variable getSum, which you could then use later in the program. You'll frequently use functions and pass arguments into functions, especially when working with event handling in JavaScript and jQuery. A JavaScript program of minimal complexity and size will typically use functions, as well. With all this background knowledge now complete, it's time to look at working with events in JavaScript and jQuery. Events are actions like mouse clicks, keystrokes, and entering text into a form.

8.3.5 Responding to Events with jQuery and JavaScript

Responding to events with JavaScript is a complex process that involves working with multiple event models exposed by different Web browsers and multiple versions of the **Document Object Model (DOM)** to try to get the correct code to execute at the correct time.

jQuery includes several event-related functions such as .click() for responding to mouse clicks, .submit() for responding to a form submission, .hover() to respond when a mouse cursor hovers over an element, and several others. In fact, the .ready() function that you saw earlier is an event handler.

You saw the .fadeOut() function in use. That function ran when the page loaded (or more accurately, when the browser encountered the JavaScript). To make the

element fade out when a user clicks it with the mouse, you need to attach a click event handler to that

element. In jQuery, you do this with the help of the .click() function, which looks like the following example (I've highlighted the relevant code in bold):

Keyword

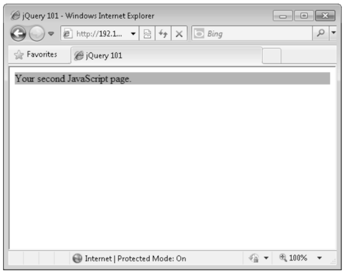
Document Object Model (DOM) is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document.

```
<!DOCTYPE HTML>
<html>
 <head>
  <meta http-equiv=”Content-Type” content=”text/html; charset=utf-8”>
 <title>jQuery 101</title>
 <script type=”text/javascript” src=”jquery-1.4.4.min.js”></script>
 </head>
<body>
 <div id=”contentDiv”>Your second JavaScript page.</div>
 <script type=”text/javascript”>
 $(“#contentDiv”).css(“backgroundColor”, “#abacab”);
 $(“#contentDiv”).click(function() {
 $(this).fadeOut(5000);
 });
 </script>
 </body>
</html>
```

Note that the .click() function is attached directly to the <div> that has the id of contentDiv. The .click() function itself calls another function (an anonymous function, enclosed in curly brackets) which calls the .fadeOut() function. You'll notice that there's a new part here, the $(this) identifier. The $(this) identifier refers to the item that raised the event, so in the example shown, $(this) refers to the contentDiv element. You could also write it like this:

```
$("#contentDiv").click(function() {
 $("#contentDiv").fadeOut(5000);
 });
```

When you load this page in a Web browser you'll see a screen like the one shown below. When you click within the <div> element, the entire <div> will slowly fade out.


jQuery 101 - Windows Internet Explorer
http://192.1...
Bing
Favorites
jQuery 101
Internet | Protected Mode: On
100%

Your second JavaScript page.

8.3.6 Validating a Web Form with jQuery and JavaScript

One typical use of JavaScript is to validate a Web form, or more appropriately, prevalidate a Web form. When a Web form is submitted, it is sent to a server-based program which can then do something useful with the data, such as complete an order, or store the data in a database. However, using JavaScript for validation provides no security for the server-based program.

With that in mind, here's a simple form and some JavaScript/jQuery code to validate the text box on the form:

```
<!DOCTYPE HTML>
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>jQuery 101</title>
<script type="text/javascript" src="jquery-1.4.4.min.js"></script>
</head>
<body>
<form id="myForm" name="myForm" action="#" method="post">
<p>Answer: <input id="firstName" type="text" name="firstname"></p>
<p><input type="submit" name="formSubmit" value="Check Form"></p>
</form>
<script type="text/javascript">
 $("#myForm").submit(function() {
 if ($("#firstName").val() == "Yes") {
 alert("Form entered correctly");
 } else {
 alert("Form not correct");
 return false;
 }
});
</script>
</body>
</html>
```

The code introduces a few new concepts, namely the if conditional. The code uses the if conditional to test whether the value entered by the user matches what you're expecting from the text field on the form. The jQuery .val() function in the preceding code is also new here. The .val() function obtains the value of whatever has been entered into the text box (or whatever element has been selected). Finally, when the text box is not filled in correctly, there's a return false; statement. In this context, return false; indicates that processing of the Web form should not continue, and the form should not be submitted.

When submitted with a value of Yes in the text box, the code displays an alert and continues with form submission. If the user enters anything else and submits the form, the validation code displays an alert indicating that the form wasn't filled in correctly, and halts form submission to the server by returning false. A more complex yet more user-friendly approach for handling errors is to change the background color on the form field that was filled in incorrectly. Obviously, for the one field form in this example, it's clear which field is incorrect. But on a more complex form, it may not be as obvious which field contains an incorrect value. Here's the code to change the background color:

```
$("#myForm").submit(function() {
 if ($("#firstName").val() == "Yes") {
 alert("Form entered correctly");
 } else {
 $("#firstName").css("backgroundColor","red");
 return false;
 }
 });
```

This code changes the backgroundColor of the form element identified by the id first Name to red when filled in incorrectly. However, best practice dictates not changing CSS style information within JavaScript code. It's much better to add and remove CSS styles from elements. Doing so makes troubleshooting easier and results in cleaner code all around.

The CSS style might look like this:

```
.errorField {
 background-color: red;
}
```

Then within the code, rather than changing the actual CSS background color, I'll apply the error class with the help of the .addClass() function, as shown in the following:

```
$("#firstName").addClass("errorClass");
```

The full page, including JavaScript code and CSS style information, is shown in the following code:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>jQuery 101</title>
<script type="text/javascript" src="jquery-1.4.4.min.js"></script>
<style type="text/css">
 .errorClass {
```

```
 background-color: red;
 }
</style>
</head>
<body>
<form id="myForm" name="myForm" action="#" method="post">
<p>Answer: <input id="firstName" type="text" name="firstname"></p>
<p><input type="submit" name="formSubmit" value="Check Form"></p>
</form>
<script type="text/javascript">
 $(document).ready(function() {
 $("#myForm").submit(function() {
 if ($("#firstName").val() == "Yes") {
 $("#firstName").removeClass("errorClass");
 alert("Form entered correctly");
 } else {
 $("#firstName").addClass("errorClass");
 return false;
 }
 });
 }); //end document ready function
</script>
</body>
</html>
```

Notice that the JavaScript code is wrapped within the .ready() function. While this is not strictly necessary here, I wanted to illustrate how you can use .ready(). Also, you'll note that upon form submission, the code uses the .removeClass() function to remove the errorClass class from the form element. In practice, you could move that .removeClass()call to any number of other places in the code, including within the main .ready() function call, or within the .submit() function.

There is much, much more to error handling, jQuery, but with this basic knowledge in hand, you can now explore why the HTML5 <canvas> tag is so important.

8.4 USING THE HTML5 <CANVAS> TAG

New in HTML5 is a tag called <canvas>, which provides a space on which you can draw graphics, run animations, and much more within a Web page. However, the tag, by itself, doesn’t do anything. Instead, the tag relies on JavaScript to draw the graphics it can contain.

The <canvas> tag is one of the easiest tags in HTML5. You simply tell the browser how large you want the canvas area to be, and it creates the container accordingly. For example, The following example shows a 250 × 250 pixel <canvas>:

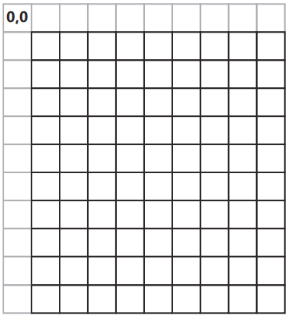
```
<canvas width=”250” height=”250” id=”myCanvas”></canvas>
```

Unfortunately, as of this writing, many browsers don’t support the <canvas> element; therefore, you’ll need to provide alternative content for browsers that aren’t ready for canvas-based graphics yet. You can provide this alternative content by placing it between the opening and closing <canvas> tags, like this:

```
<canvas width=”250” height=”250” id=”myCanvas”>
 <p>Alternate content goes here</p>
</canvas>
```

Browsers that don’t understand the <canvas> tag will ignore it and display the contents of the HTML found within it; in this case, a <p> tag.

To draw on the canvas you need to specify the coordinates at which you’d like to begin. Canvas coordinates use a grid system, like the one shown in the following image:



In the grid shown above, the coordinates 0,0 represent the top-most and left-most cell in the grid and the numbers increase as you move both to the right and down. The cells themselves represent pixels on the screen. The points along the horizontal axis on the grid are called x coordinates while points along the vertical axis are called y coordinates. You draw using combinations of lines and primitive shapes. For example, you can create rectangles of various forms using functions such as fillRect to draw a filled-in rectangle and strokeRect to draw an outlined rectangle. Both of these functions accept x and y coordinates to determine where they should begin drawing, along with a width and height specification. The code to draw a 50 × 100-pixel rectangle beginning at x coordinate 10 and y coordinate 20 looks like this:

```
fillRect(10,20,50,100)
```

Before you start drawing though, you need a canvas. Additionally, you need to call the getContext() function for the canvas as well. You'll see examples of how to do this in the following code example, which shows an entire page that uses the tag to draw this rectangle.

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=utf-8”>
<title>Canvas</title>
<script type=”text/javascript”
 src=”http://code.jquery.com/jquery-1.4.4.min.js”></script>
</head>
<body>
<canvas width=”250” height=”250” id=”myCanvas”>
 <p>Alternate content goes here</p>
</canvas>
<script type=”text/javascript”>
 $(document).ready(function() {
 var canvas = document.getElementById(“myCanvas”);
 if (canvas.getContext) {
 var canvasContext = canvas.getContext(“2d”);
 canvasContext.fillStyle = “blue”;
 canvasContext.fillRect(10,20,50,100);
 } else {
 // You could do something else here
 // because the browser doesn’t support
 // the canvas element.
 }
 });
```

```
</script>
</body>
</html>
```

This code creates a element (and provides alternative content for non-HTML5 browsers). The bulk of the page consists of the JavaScript code to draw the rectangle, as shown here:

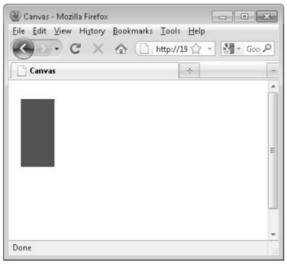
```
var canvas = document.getElementById("myCanvas");
 if (canvas.getContext) {
 var canvasContext = canvas.getContext("2d");
 canvasContext.fillStyle = "blue";
 canvasContext.fillRect(10,20,50,100);
 } else {
 // You could do something else here
 // because the browser doesn't support
 // the canvas element.
 }
```

This code retrieves the element with the id myCanvas, and places that into a JavaScript variable called canvas. Next, the code tests to see if the getContext() function is available. If this function isn't available, then the browser almost certainly doesn't support the <canvas> element. Therefore, attempting to use getContext() would result in a JavaScript error. If, however, the getContext() function is available, then the code continues by calling getContext(), setting the fillStyle to blue, and drawing the rectangle.

Keyword

Mozilla Firefox or simply Firefox, is a cross-platform, free and open-source web browser developed by the Mozilla Foundation and its subsidiary, the Mozilla Corporation.

Viewing this page in a -compatible browser such as **Mozilla Firefox** results in a page with a blue rectangle, like the one shown here:



Viewing the page in a browser that doesn’t support <canvas> results in the alternative content being displayed.

Here’s another example. This example resizes the rectangle dynamically using JavaScript, based on where you clicked within the canvas area:

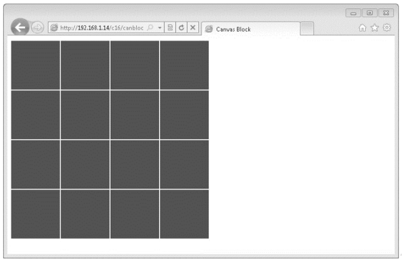
```
var canvas = document.getElementById(“myCanvas”);
 if (canvas.getContext) {
 var canvasContext = canvas.getContext(“2d”);
 canvasContext.fillStyle = “blue”;
 canvasContext.fillRect(0,0,50,100);
 $(“#myCanvas”).click(function(f) {
 var x = f.pageX - this.offsetLeft;
 var y = f.pageY - this.offsetTop;
 canvasContext.clearRect(0,0,250,250);
 canvasContext.fillRect(0,0,x,y)
 });
 } else {
 // You could do something else here
 // because the browser doesn’t support
 // the canvas element.
 }
```

This example adds a .click() function thanks to jQuery. The .click() function examines where the mouse click occurred within the canvas element. It then clears the canvas and draws a new rectangle at the point where the mouse was clicked. This example begins to show the interactivity that’s possible with the canvas element.

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=utf-8”>
<title>Canvas Block</title>
<script type=”text/javascript”
 src=”http://code.jquery.com/jquery-1.4.4.min.js”></script>
```

```
</head>
<body>
<canvas width=”400” height=”400” id=”myCanvas”>
 <p>Alternate content goes here</p>
</canvas>
<script type=”text/javascript”>
 $(document).ready(function() {
 var canvas = document.getElementById(“myCanvas”);
 if (canvas.getContext) {
 var canvasContext = canvas.getContext(“2d”);
 canvasContext.fillStyle = “blue”;
 var numBlocks = 4;
 var canWidth = $(“#myCanvas”).attr(“width”);
 var canHeight = $(“#myCanvas”).attr(“height”);
 var blockWidth = (canWidth/numBlocks) - 2;
 var blockHeight = (canHeight/numBlocks) - 2;
 var offsetX = 0;
 var offsetY = 0;
 var colCount = 0;
 var numTotal = numBlocks * numBlocks;
for (i = 0; i < numTotal; i++) {
 canvasContext.fillRect(offsetX,offsetY,
 blockWidth,blockHeight);
 offsetX = offsetX + blockWidth + 2;
 colCount++;
 if (colCount == numBlocks) {
 colCount = 0;
 offsetY = offsetY + blockHeight + 2;
 offsetX = 0;
 }
 }
 $(“#myCanvas”).click(function(f) {
 var x = f.pageX - this.offsetLeft;
 var y = f.pageY - this.offsetTop;
 var xBlock = Math.floor((x / blockWidth));
 var yBlock = Math.floor((y / blockHeight));
 var xSpan = 0, ySpan = 0;
 if (xBlock > 0) {
 xSpan = xBlock * 2;
 }
 if (yBlock > 0) {
 ySpan = yBlock * 2;
 }
 var xPos = (blockWidth * xBlock) + xSpan;
 var yPos = (blockHeight * yBlock) + ySpan;
 canvasContext.clearRect(xPos,yPos,blockWidth,
 blockHeight);
 });
 } else {
 // You could do something else here
 // because the browser doesn’t support
 // the canvas element.
 }
 });
</script>
</body>
</html>
```

Here’s what this application initially looks like:



If you become bored with a 4 by 4 grid, change the number of blocks by changing this line in the code, as follows:

```
var numBlocks = 4;
```

The code in the example will dynamically change the grid to match the number of blocks you specify by setting the numBlocks variable.

ROLE MODEL

BRENDAN EICH

Brendan Eich (/ˈaɪk/; born July 4, 1961) is an American computer programmer and technology executive. He created the JavaScript programming language and co-founded the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation. He served as the Mozilla Corporation's chief technical officer before he was appointed chief executive officer, but resigned shortly after his appointment due to controversy over his opposition to same-sex marriage. He subsequently became the CEO of Brave Software.

Early Life

Eich grew up in Pittsburgh; Gaithersburg, Maryland; and Palo Alto, and he attended Ellwood P. Cubberley High School, graduating in the class of 1979. He received his bachelor's degree in mathematics and computer science at Santa Clara University, and he received his master's degree in 1985 from the University of Illinois at Urbana–Champaign. Eich is Roman Catholic.

He began his career at Silicon Graphics, working for seven years on operating system and network code. He then worked for three years at MicroUnity Systems Engineering writing microkernel and DSP code.

Netscape and JavaScript

Eich started work at Netscape Communications Corporation in April 1995. Eich originally joined intending to put Scheme "in the browser", but his Netscape superiors insisted that the language's syntax resemble that of Java. As a result, Eich devised a language that had much of the functionality of Scheme, the object-orientation of Self, and the syntax of Java. He completed the first version in ten days in order to accommodate the Navigator 2.0 Beta release schedule, and was called Mocha, but renamed LiveScript in September 1995 and finally in a joint announcement with Sun Microsystem it was named JavaScript in December. Simultaneously, he designed the first SpiderMonkey engine for the Netscape Navigator browser at Netscape Communications. When Mozilla inherited the Netscape base code in 1998, it included this engine, which was written in the C programming language. It was then changed in JavaScript 1. 5 to comply with the ECMA-262 standard. Eich continued to oversee the development of SpiderMonkey, the specific implementation of JavaScript in Navigator.

Mozilla

In early 1998, Eich co-founded the free and open source software project Mozilla with Jamie Zawinski and others, creating the mozilla.org website, which was meant to manage open-source contributions to the Netscape source code. He served as Mozilla›s chief architect. AOL bought Netscape in 1999. After AOL shut down the Netscape browser unit in July 2003, Eich helped spin out the Mozilla Foundation.

In August 2005, after serving as a lead technologist and as a member of the board of directors of the Mozilla Foundation, Eich became chief technical officer (CTO) of the newly founded Mozilla Corporation, meant to be the Mozilla Foundation's for-profit arm. Eich continued to "own" the Mozilla SpiderMonkey module, its JavaScript engine, until he passed on the ownership of it to Dave Mandolin in 2011.

Appointment to CEO and resignation

On March 24, 2014, Mozilla made the decision to appoint Eich as CEO of Mozilla Corporation. The appointment triggered widespread criticism due to Eich's past political donations – specifically, a 2008 donation of $1,000 to California Proposition 8, which called for the banning of same-sex marriage in California, and donations in the amount of $2,100 to Proposition 8 supporter Tom McClintock between 2008 and 2010. *The Wall Street Journal* initially reported that, in protest against his coming appointment, half of Mozilla›s board (Gary Kovacs, John Lilly, and Ellen Siminoff) stepped down, leaving Mitchell Baker, Reid Hoffman, and Katharina Borchert. CNET later reported that of the three board members who had left, only Lilly left due to Eich's appointment. Lilly told *The New York Times*, "I left rather than appoint him", and declined to elaborate further.

On March 26, 2014, Eich expressed "sorrow for causing pain" and pledged to "work with LGBT communities and allies" at Mozilla. Some of the activists created an online campaign against Eich, with online dating site OkCupid automatically displaying a message to Firefox users with information about Eich's donation, and suggesting that users switch to a different browser (although giving them a link to continue with Firefox). CREDO Mobile collected more than 50,000 signatures demanding that Eich resign.

After 11 days as CEO, Eich resigned on April 3, 2014, and left Mozilla over his opposition to same-sex marriage. In his personal blog, he posted, "under the present circumstances, I cannot be an effective leader". Mozilla made a press release saying that board members tried to get Eich to stay in the company in a different role, but that he had chosen to sever ties for the time being.

Brave Software

Eich is the CEO of Brave Software, an Internet browser platform company that raised $2.5 million in early funding from angel investors. In January 2016, the company released developer versions of its open-source, Chromium-based Brave web browser, which blocks ads and trackers.

At Brave Software, Eich co-created the Basic Attention Token (BAT), a cryptocurrency designed for use in the Brave browser. BAT launched its ICO on May 31, 2017, and raised $35 million.

In 2020, during the COVID-19 pandemic, Eich ardently denied the effectiveness of face masks and lockdowns in combating the transmission of COVID-19 and stated that United States' top immunologist Anthony Fauci "lies a lot", sparking calls for his removal from Brave on social media.

SUMMARY

- JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.
- The canvas element enables advanced graphics and interactions in ways that previously you could only achieve by using slow, plug-in–laden pages. The canvas element is a free-form area where you can use images, drawn graphics, animation, and text to enhance the user experience on a Web page.
- JavaScript is not Java; it bears no relation to the Java programming language, the drink, or the country. JavaScript is a programming language that is used primarily to provide additional functionality to Web pages and applications, and it's used heavily in the "Web 2.0" paradigm of highly interactive Web sites.
- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.
- JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.
- jQuery is an open source JavaScript file that not only removes the need for developers to handle many of the cross-browser incompatibilities but also simplifies much more advanced programming than would normally be available to novice JavaScript programmers.
- jQuery provides several other ways to select elements, including hierarchical functions by which you can select the children of an element; a way to select every other element except a specified element; an element's parent elements; and many other selectors.

KNOWLEDGE CHECK

1. **Which type of JavaScript language is ……………..**
   a. Object-Oriented
   b. Object-Based
   c. Assembly-language
   d. High-level

2. **Which of the following is the correct output for the following JavaScript code:**

```
varx=5,y=1
var obj ={ x:10}
with(obj)
{
      alert(y)
}
```

   a. 1
   b. Error
   c. 10
   d. 5

3. **Which one of the following also known as Conditional Expression:**
   a. Alternative to if-else
   b. Switch statement
   c. If-then-else statement
   d. immediate if

4. **In JavaScript, what is a block of statement?**
   a. Conditional block
   b. block that combines a number of statements into a single compound statement
   c. both conditional block and a single statement
   d. block that contains a single statement

5. **What will happen, if the following JavaScript code is executed?**

```
var count =0;
while (count <10)
{
     console.log(count);
```

```
    count++;
}
```

a. An error is displayed

b. An exception is thrown

c. The values of count variable are logged or stored in a particular location or storage

d. The value of count from 0 to 9 is displayed in the console

6. Which one of the following is the correct way for calling the JavaScript code?

a. Preprocessor

b. Triggering Event

c. RMI

d. Function/Method

7. Which of the following is not JavaScript Data Types?

a. Undefined

b. Number

c. Boolean

d. Float

8. Inside which HTML element do we put the JavaScript?

a. <script>

b. <head>

c. <meta>

d. <style>

9. Which of the following is correct about features of JavaScript?

a. It can not Handling dates and time.

b. JavaScript is a object-based scripting language.

c. JavaScript is not interpreter based scripting language.

d. All of the above

10. Choose the correct JavaScript syntax to change the content of the following HTML code.

a. document.getElement (“letsfindcourse”).innerHTML = “I am a letsfindcourse”;

b. document.getElementById (“letsfindcourse”).innerHTML = “I am a letsfindcourse”;

c. document.getId (“letsfindcourse”) = “I am a letsfindcourse”;

d. document.getElementById (“letsfindcourse”).innerHTML = I am a letsfindcourse;

REVIEW QUESTIONS

1. What is JavaScript?
2. What are the different data types present in javascript?
3. What is DOM?
4. What are the jQuery functions used to provide effects?
5. What is the difference between javascript and jquery?

Check Your Result

1. (b)	2. (a)	3. (d)	4. (b)	5. (c)
6. (d)	7. (d)	8. (a)	9. (b)	10. (b)

REFERENCES

1. Baker, Loren (November 24, 2004). "Mozilla Firefox Internet Browser Market Share Gains to 7.4%". Search Engine Journal. Archived from the original on May 7, 2021. Retrieved May 8, 2021.
2. Brendan Eich: An Introduction to JavaScript, JSConf 2010. p. 22m. Archived from the original on August 29, 2020. Retrieved November 25, 2019. Eich: "function", eight letters, I was influenced by AWK.
3. Champeon, Steve (April 6, 2001). "JavaScript, How Did We Get Here?". oreilly.com. Archived from the original on July 19, 2016. Retrieved July 16, 2016.
4. Eich, Brendan (April 3, 2008). "Popularity". Archived from the original on July 3, 2011. Retrieved January 19, 2012.
5. Enzer, Larry (August 31, 2018). "The Evolution of the Web Browsers". Monmouth Web Developers. Archived from the original on August 31, 2018. Retrieved August 31, 2018.
6. Flanagan, David. JavaScript: The Definitive Guide. 7th edition. Sebastopol, California: O'Reilly, 2020.
7. Jean-Pharuns, Alix (July 30, 2015). "Rowhammer.js Is the Most Ingenious Hack I've Ever Seen". Motherboard. Vice. Archived from the original on January 27, 2018. Retrieved January 26, 2018.
8. McCracken, Harry (September 16, 2010). "The Unwelcome Return of "Best Viewed with Internet Explorer"". technologizer.com. Archived from the original on June 23, 2018. Retrieved July 16, 2016.
9. Mozilla Foundation, Mozilla Foundation Security Advisory 2005–41: Privilege escalation via DOM property overrides Archived 2014-06-04 at the Wayback Machine
10. Purdy, Kevin (June 11, 2009). "Lifehacker Speed Tests: Safari 4, Chrome 2". Lifehacker. Archived from the original on April 14, 2021. Retrieved May 8, 2021.
11. Server-Side JavaScript Guide. Oracle Corporation. December 11, 1998. Archived from the original on March 11, 2021. Retrieved May 8, 2021.
12. Usage statistics of JavaScript as client-side programming language on websites. w3techs.com. Archived from the original on 2021-08-13. Retrieved 2021-04-09.

Index

Level: Beginner to Advanced
Subject: Computer and Information Science

Basic Computer Coding: HTML

2nd Edition

HTML or Hypertext Markup Language is a programming language employed to describe the structure of information on a webpage. In a combined way, HTML, CSS, and JavaScript make up the vital building blocks of websites globally, with CSS controlling the appearance of a webpage and JavaScript programming and its functionality. This grooms a business standard in the market. A good design takes user by the hand and directs them where they need to go and find the information they need quickly and easily. HTML can be a little intimidating for those not accustomed to looking at code. That's why this book is designed as an introduction to web design basics. This book will break down the basics of how to get your HTML page created. This book will take you through what HTML is, how to create an HTML page, HTML basics, and what program you can use to edit the HTML.

This edition is comprised of eight chapters. In this book, you'll learn how to design and build beautiful websites by learning the basic principles of design like branding, color theory, and typography which are all instrumental in the design process of a website. You'll also learn HTML and CSS, which are the common code languages that all modern websites are built on.

ISBN 978-1-98467-601-6

BIBLIOTEX
Digital Library

www.3ge-learning.com
email: info@3ge-learning.com