

# Introduction to Web Development

LEARN  
HTML , CSS  
& JS



C • O • L  
THE COMMONWEALTH *of* LEARNING



Department of Computer Science  
Allama Iqbal Open University, Pakistan



# **Introduction to Web Development**

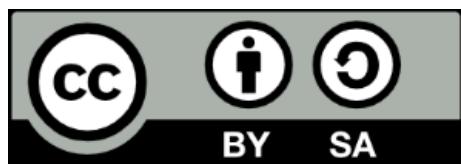
**Allama Iqbal Open University, Pakistan  
Department of Computer Science**



# Copyright

This course has been developed by Allama Iqbal Open University (AIOU) Pakistan as a part of collaborative advanced ICT course development project of the Commonwealth of Learning (COL). COL is an intergovernmental organization created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

The Allama Iqbal Open University (AIOU) is a fully fledged, autonomous and accredited public University. It offers its certificate, diploma, degree and postgraduate courses through the open and distance learning system which includes various means of communication such as face-to-face, broadcasting, telecasting, correspondence, seminars, e-learning as well as a blended mode.



© 2017 by the Commonwealth of Learning and Allama Iqbal Open University (AIOU). Except where otherwise noted, *Introduction to Web development* made available under Creative Commons Attribution-Share Alike 4.0 International (CC BY-SA 4.0) License: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>.

For the avoidance of doubt, by applying this licence the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of *Commonwealth of Learning* and do not commit the organisation.



Allama Iqbal Open University  
Sector H-8,  
Islamabad,  
Pakistan  
Phone: +92-51-9250091  
Fax: +92-51-9250092  
Email: colp@aiou.edu.pk  
Website: www.aiou.edu.pk



Commonwealth of Learning  
4710 Kingsway, Suite 2500, Burnaby  
V5H 4M2,  
British Columbia,  
Canada  
Phone: +1 604 775 8200  
Fax: +1 604 775 8210  
Email: info@col.org  
Website: www.col.org



## **Acknowledgements**

Department of Computer Science, Allama Iqbal Open University (AIOU) wishes to thank the following for their contribution to the production of this course material and video lectures.

### **Authors**

Dr. Amjad Mahmood (Department of Computer Science, University of Bahrain, Bahrain)

Dr. Moiz Uddin Ahmed (Department of Computer Science, AIOU, Pakistan)

Mr. Hanaan Sadeed Ahmad (Department of Computer Science, AIOU, Pakistan)

### **Reviewer**

Mr. Khalid Iqbal (Islamabad College for Boys, Islamabad, Pakistan)

### **Copy Editor**

Mr. Naveed Anjum (Islamabad College for Boys, Islamabad, Pakistan)

### **Layout Design**

Mr. Mushtaq Hussain (PPU, AIOU)

### **Video Production**

Dr. Amjad Mahmood (Department of Computer Science, University of Bahrain, Bahrain)

Dr. Moiz Uddin Ahmed (Department of Computer Science, AIOU, Pakistan)

Mr. Ghulam Umer Abbasi (Producer, Institute of Educational Technology, AIOU)

Mr. Muhammad Younis (Senior Cameraman, Institute of Educational Technology, AIOU)

Mr. Raheel Tahir (Associate Engineer, Institute of Educational Technology, AIOU)

Mr. Ahsan Farooq (Department of Computer Science, AIOU, Pakistan)

### **Research Associates**

Mr. Muhammad Aquil (Department of Computer Science, AIOU, Pakistan)

Ms. Reeba Raynah (Department of Computer Science, AIOU, Pakistan)

Mr. Hussnain Tariq (Department of Computer Science, AIOU, Pakistan)

### **Project Director**

Dr. Moiz Uddin Ahmed (Department of Computer Science, AIOU, Pakistan)

### **Co-Project Director**

Dr. Zahid Majeed (International Collaboration & Exchange Office, AIOU, Pakistan)

This course has been developed with the support of the **Commonwealth of Learning, Canada**.



## Contents

### About this course material

---

How this course material is structured .....	1
--	---

### Course overview

---

Welcome to Introduction to Web Development .....	3
Introduction to Web Development – is this course for you ?.....	3
Course outcomes.....	5
Timeframe .....	5
Study skills .....	6
Need help? .....	7
Assessments .....	7

### Getting around this course material

---

icons.....	8
------------	---

### Unit 1

---

Introduction to Web Development.....	9
Introduction .....	10
Unit outcomes.....	10
Terminologies.....	10
1.1. Internet .....	11
1.1.1. IP Addresses.....	11
1.2. TCP/IP Protocol Suits.....	12
1.3. ISP and its Working .....	13
1.4. World Wide Web (WWW).....	13
1.5. Website and Domain Names .....	14
1.5.1. Domain Names.....	14
1.6. Web Hosting .....	15
1.6.1. Types of Web Hosting .....	15
1.7. Website Development Process .....	16
1.8. Static vs Dynamic Web pages .....	18
1.9. Web Development Technologies.....	18
1.10. Client Side vs Server Side Scripting .....	19
1.10.1. Client Side Scripting .....	19
1.10.2. Server Side Scripting .....	20
1.10.3. Common Gateway Interface.....	20
1.11. Hypertext, Hypermedia Tools and IDEs .....	20

1.11.1. Hypermedia Tools .....	20
Unit Summary.....	21
Self Assessment Questions.....	22
References and Further Reading.....	24

---

**Unit 2**

HTML Basics.....	25
Introduction .....	26
Unit Outcomes .....	26
Terminologies.....	27
2.1. Developing Your First Web page.....	28
2.2. HTML Document Structure .....	29
2.3. HTML Tags.....	30
2.4. Basic HTML Tags.....	31
2.5. Comments .....	38
2.6. Formatting Tags .....	39
2.6.1. Content Based Tags .....	39
2.6.2. Physical Style Tags .....	41
2.7. Lists.....	43
2.7.1. Ordered List .....	44
2.7.2. Unordered List .....	45
2.7.3. Nesting of List .....	46
Unit Summary.....	48
Self Assessment Questions.....	49
References and Further Reading.....	51

---

**Unit 3**

HTML Links, Images and Tables .....	53
Introduction .....	54
Unit Outcomes .....	54
Terminologies.....	54
3.1. Images and Sounds .....	55
3.2. Adding Images in a Web page.....	56
3.2.1. Set Images Width and Height .....	57
3.3. HTML Links .....	58
3.3.1. Hyperlink .....	59
3.3.2. The Name Attribute .....	60
3.3.3. Images as Hyperlink .....	61
3.3.4. Link Tag .....	61
3.3.5. <nav></nav> .....	62
3.4. Tables .....	62
3.4.1. Adding Table Caption.....	65

3.4.2. The Rowspan and Colspan Attributes.....	66
3.4.3. Changing Alignment.....	68
Unit Summary .....	68
Self Assessment Questions .....	69
References and Further Reading .....	72

---

**Unit 4**

Marquee, Form and Input types .....	73
Introduction .....	74
Unit Outcomes .....	74
Terminologies.....	74
4.1. Marquee.....	75
4.2. Form .....	77
4.2.1. <form>.....	77
4.2.2. <input> .....	78
4.2.3. <text area> .....	85
4.2.4. <select> .....	86
4.2.5. <datalist>.....	87
4.2.6. Complete Example of Input types and forms .....	88
4.3. iframes .....	90
Unit Summary.....	91
Self Assessment Questions.....	92
References and Further Reading.....	93

---

**Unit 5**

Cascading Style Sheet .....	95
Introduction .....	96
Unit Outcomes .....	96
Terminologies.....	96
5.1. Advantages of CSS.....	97
5.2. Your First Experience with Style .....	97
5.3. Methods of Writing CSS .....	99
5.3.1. External Style Sheet .....	99
5.3.2. Internal Style Sheet.....	100
5.3.3. Inline Style.....	101
5.4. CSS Syntax .....	102
5.4.1. Element Selector.....	103
5.4.2. ID Selector.....	104
5.4.3. Class Selector .....	105
5.5. CSS Comments .....	106
5.6. Basic Properties of CSS .....	107
5.6.1. Background Properties in CSS.....	108

Unit Summary.....	114
Self Assessment Questions.....	115
References and Further Reading.....	117

---

## Unit 6

Text, Border and Table Properties in CSS .....	119
Introduction .....	120
Unit Outcomes .....	120
Terminologies.....	120
6.1. Text Properties in CSS .....	121
6.2. Border Properties.....	123
6.3. Margin Properties .....	125
6.4. Padding Properties in CSS .....	127
6.5. Table Properties .....	128
6.6. Position Properties in CSS .....	131
Unit Summary.....	136
Self Assessment Questions.....	137
References and Further Reading.....	140

---

## Unit 7

Client Side Scripting - I.....	141
Introduction .....	142
Unit Outcomes .....	142
Terminologies.....	142
7.1. JavaScript .....	143
7.1.1. Advantages and Limitations of JavaScript .....	144
7.2. First JavaScript Program.....	144
7.2.1. Where to put JavaScript code.....	146
7.2.2. JavaScript in Body Section .....	146
7.2.3. JavaScript in Head Section .....	148
7.3. External JavaScript .....	150
7.4. JavaScript Basics.....	152
7.5. JavaScript Variables .....	152
7.5.1. Declaring Variables .....	153
7.6. JavaScript Comments .....	154
7.6.1. Single line Comments .....	154
7.6.2. Multi line Comments .....	155
Unit Summary.....	156
Self Assessment Questions.....	157
References and Further Reading.....	159

**Unit 8**

Client Side Scripting - II.....	161
Introduction .....	162
Unit Outcomes .....	162
Terminologies.....	162
8.1. JavaScript Operators .....	163
8.1.1. Arithmetic Operators.....	163
8.1.2. Comparison Operators.....	164
8.1.3. Logical Operators.....	164
8.1.4. Assignment Operators .....	165
8.2. JavaScript Conditional Statements .....	166
8.2.1. if Statement .....	166
8.2.2. if-else Statement.....	168
8.2.3. if-else-if Statement .....	169
8.2.4. switch Statement.....	171
8.3. JavaScript Loops .....	173
8.3.1. for Loop.....	173
8.3.2. while Loop.....	175
8.3.3. do while Loop.....	176
8.3.4. break Statement .....	178
8.3.5. continue Statement .....	180
8.4. JavaScript Objects .....	181
8.4.1. Object Properties.....	181
8.4.2. Object Methods .....	181
8.4.3. Object Constructor.....	182
8.5. JavaScript and HTML Forms .....	183
8.6. JavaScript Events.....	185
Unit Summary.....	189
Self Assessment Questions.....	190
References and Further Reading.....	192

**Unit 9**

Website Hosting .....	193
Introduction .....	194
Unit Outcomes .....	194
Terminologies.....	194
9.1. Web Servers .....	195
9.1.1. Types of Web Server .....	195
9.2. Web Hosting.....	196
9.2.1. Types of Web Hosting .....	196
9.3. Domain Name .....	197
9.3.1. Sub Domain .....	197
9.4. File Transfer Protocol .....	198

9.4.1. FTP Server .....	198
9.4.2. How we can use FTP .....	198
9.5. Uploading a Website .....	198
9.5.1. Upload using cPanel File Manager.....	199
9.5.2. Upload using Filezilla FTP Client .....	201
Summary .....	203
Self Assessment Questions.....	204
References and Further Reading.....	205



## About this Course Material

Introduction to Web Development has been produced by Allama Iqbal Open University (AIOU) in collaboration with Commonwealth of Learning. This Course Material is structured as outlined below:

### How this Course Material is structured?

#### The Course Overview

The course overview gives you a general introduction to the course. Information contained in the course overview will help you determine

- If the course is suitable for you.
- What you will already need to know.
- What you can expect from the course.
- How much time you will need to invest to complete the course.

The overview also provides guidance on

- Study skills.
- Where to get help.
- Course assignments and assessments.
- Activity icons.
- Units (Chapters).



We strongly recommend that you read the overview *carefully* before starting.

#### The course content

The course is broken down into nine units. Each unit is comprising of:

- An introduction to the unit content.
- Unit outcomes.
- New terminology.
- Core content of the unit with a variety of learning activities.
- A unit summary.
- Self assessment questions
- Reference and further reading.

Links to video lectures for each unit are provided in the relevant section of the course material.

## Resources

For those interested in learning more on this subject, we provide you with a list of additional resources at the end of each unit. The additional resources include books, articles and websites.

## Your Comments

After completing Introduction to Web Development, we would appreciate it if you could take a few moments to give us your feedback on any aspect of this course. Your feedback might include comments on

- Course content and structure
- Course reading materials and resources
- Course activities
- Video lectures
- Course review questions
- Course duration
- Course support (assigned tutors, technical help, etc.)

Your constructive feedback will help us to improve and enhance this course in future.



## Course Overview

### Welcome to Introduction to Web Development

This course intends to provide basic knowledge and skills required for planning, designing and developing effective web pages with a focus on the practical application of the technologies used in the web development. It covers basic terminologies, tools, HTML5, Cascading Style Sheet (CSS) and the basics of javascript. It also describes how to host a website on a selected web server.



### Course Overview Video

<https://youtu.be/AZkSjNmGPbM>



---

### Is this course for you?

This course is intended for people who are already computer literate. The course aims to:

- *Introduce learners the basic terminologies and various phases of web application development process.*
- *Help learners to recognize and understand HTML web page elements and HTML tags.*
- *Enable learners to develop web pages using HTML.*
- *Make learners capable to apply CSS to format web page elements.*
- *Develop basic programming skills using javascript.*
- *Host a website using available hosting services.*





## Course Outcomes



### Outcomes

Upon successful completion of the course, students will be able to:

- *Describe basic terminologies and various phases of web application development process.*
- *Recognize and understand HTML web page elements and HTML tags.*
- *Develop web pages using HTML.*
- *Apply CSS to format web page elements.*
- *Enhance functionality of web pages using client side scripting with javascript.*
- *Plan, design and deploy a multi-page website using available hosting services.*

## Timeframe



This is a one-semester course.

This course requires timeframe that depends on individual institution's mode of delivery.

A minimum standard of delivery should be 18 weeks of blended learning mode which includes face-to-face and online lectures, supervised and unsupervised laboratory and tutorials workshops.

Self-study time is 10 hours per-week.

## Study Skills



As an adult learner, your approach to learning will be different to that from your school days: you will choose what you want to study, you will have professional and/or personal motivation for doing so and you will most likely be fitting your study activities around other professional or domestic responsibilities.

Essentially you will be taking control of your learning environment. As a consequence, you will need to consider performance issues related to time management, goal-setting, stress management, etc. Perhaps you will also need to reacquaint yourself in areas such as essay planning, coping with exams and using the web as a learning resource.

Your most significant considerations will be *time* and *space* i.e. the time you dedicate to your learning and the environment in which you engage in that learning.

We recommend that you take time now—before starting your self-study—to familiarize yourself with these issues. There are a number of excellent resources on the web which you can access to further guide you on this. The following are just a few suggested links:

- <http://www.how-to-study.com/>

The “How to study” website is dedicated to study skills resources. You will find links to study preparation (a list of nine essentials for a good study place), taking notes, strategies for reading text books, using reference sources, test anxiety.

- <http://www.ucc.vt.edu/stdysk/stdyhlp.html>

This is the website of the Virginia Tech, Division of Student Affairs. You will find links to time scheduling (including a “where does time go?” link), a study skill checklist, basic concentration techniques, control of the study environment, note taking, how to read essays for analysis, memory skills (“remembering”).

- <http://www.howtostudy.org/resources.php>

Another “How to study” website with useful links to time management, efficient reading, questioning/listening/observing skills, getting the most out of doing (“hands-on” learning), memory building, tips for staying motivated, developing a learning plan.

The above links are our suggestions to start you on your way. At the time of writing these web links were active. If you want to look for more go to [www.google.com](http://www.google.com) and type “self-study basics”, “self-study tips”, “self-study skills” or similar.



## Need Help?



This course is offered at Computer Science Department, Allama Iqbal Open University.

If you need help regarding this course, please contact:



Course Coordinator  
Department of Computer Science  
Allama Iqbal Open University  
Sector H-8,  
Islamabad,  
Pakistan  
Phone: +92-51-9250091  
Fax: +92-51-9250092  
Email: colp@aiou.edu.pk  
Website: www.aiou.edu.pk

## Assessment



The performance of the learners is accessed through assignments, practical and written examination.

## Getting around this Course Material

### Icons

While working through this course material you will notice the frequent use of certain icons. These icons serve to “signpost” a particular piece of text, a new task or change in activity; they have been included to help you to find your way around this course material.

A complete icon set is shown below. We suggest that you familiarize yourself with the icons and their meaning before starting your study.

			
Activity	Self Assessment	Video	Important Point
			
Section Heading	Outcomes	Help	Further Reading
			
Unit Summary	Terminology	Code	Study skills



## UNIT 1

# Introduction to Web Development



## Introduction

Though the focus of this course is on how to create websites using HTML, CSS, JavaScript and other technologies, but mastery of HTML, CSS and other technologies should not be confused with the understanding of the web development process. A web developer should also have basic knowledge about the working of the Internet, TCP/IP protocol suit and concepts related to them. Furthermore, developing large and complex websites require a careful planning and use of a suitable development process. This unit provides an introduction to the basics of Internet and various web development technologies. It provides web development process and various phases involved in website development. Website development process is followed through guidelines and documentation in the systematic way by the developer.



## Unit Outcomes

Upon completion of this unit you should be able to:

1. Describe the working of Internet and how services provided on it can be accessed.
2. Differentiate between static and dynamic web pages.
3. Explain web development process and activities carried out during each phase.
4. Explain various technologies used to develop a website.



## Terminologies

- IP:** Internet Protocol (IP) is a communication protocol that provides rules for exchanging data among computers.
- TCP:** Transmission Control Protocol (TCP) maintains data flow during end to end communication.
- ISP:** Internet Service Provider is a company that provides Internet services.
- WWW:** World Wide Web is a collection of resources over the Internet.
- Domain Name:** Domain name is used to identify a website on the Internet.



## 1.1. Internet

Internet, sometimes called simply "the Net," is collections of standalone computers (and computer networks in companies and organizations) that are linked together, using communication networks. It connects millions of computers around the world to exchange data. The connections between the computers are made using fiber-optic cables, copper cables, wireless radio connections and satellite links. Internet can handle different kinds of information enabling people to do different tasks such as sending/receiving emails, searching web, online chatting and many more. All the information is handled on the Internet exactly in the same way because of the flexible design of the Internet.



Internet is a worldwide computer network which consists of interconnected networks.



Internet provides a variety of information and communication facilities.

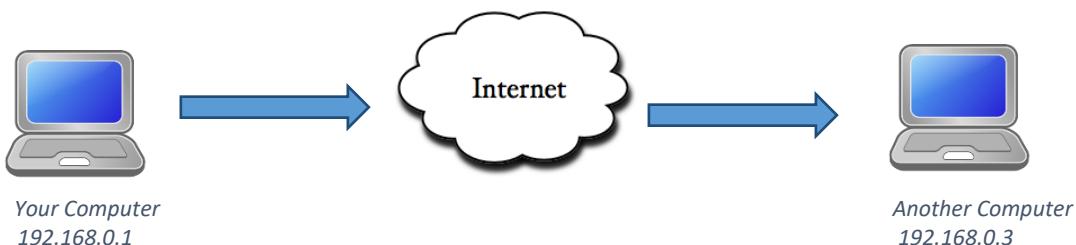
### 1.1.1. IP Addresses

As we mentioned that Internet is global network of computers, the question then arises how each computer is uniquely identified which is required for transmitting data from one computer (sources) to another one (destination). The answer is simple as every computer that connects to the Internet is given a unique address, called IP address (Internet Protocol address, to be more specific). The IP (IPv4) standards specify that each host is assigned a unique 32-bit IP address. However, because of the growth of Internet and the predicted depletion of available addresses, a new version of IP (IPv6), was developed that uses 128 bits address.



Each device connected to the Internet has a unique IP address.

IP addresses are usually written and displayed in human-readable notations, called the dotted-decimal notation such as 172.16.254.1. (IPv4), and 2001:db8:0:1234:0:567:8:1 (IPv6). Figure 1.1 shows two computers with their IP addresses connected to the Internet.



**Figure 1.1:** Computers connected to the Internet with their IP addresses

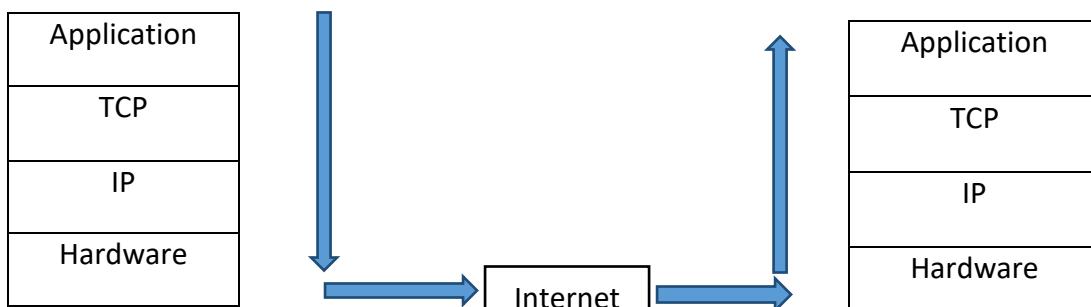
If you are connected to the Internet through an Internet Service Provider (ISP), ISP assigns a temporary IP address to your computer or device over a period of a session. In case of connection through Local Area Network (LAN), a temporary or permanent IP address obtained from DHCP (Dynamic Host Configuration Protocol) server is assigned to your computer.

## Activity 1

1. Find the IP address of your computer.
2. Disconnect your computer from Internet and reconnect it again after some time. Check its IP address. Is it the same or different from the previous IP address?
3. Explore the concepts of LAN and DHCP.
4. Explore the concept of 32 and 128 bit addresses?

## 1.2. TCP/IP Protocol Suits

TCP/IP is the protocol of the Internet that helps to provide data transfer mechanism based on packetizing, addressing, transmitting, routing and receiving. TCP/IP protocol is a layered protocol as shown in the figure 1.2.



**Figure 1.2:** TCP/IP protocol layers



TCP/IP is the basic communication protocol of the Internet.



TCP/IP protocol suite specifies how to transmit and receive data on a network.

Each layer of the TCP/IP has a particular function to perform as described in table 1.1.

**Table 1.1:** TCP/IP protocol layers and their functions

Protocol Layer	Description
Application Protocols Layer	This layer is concerned with providing network services to applications. Protocols specific to the application layer are http, FTP, etc.
Transmission Control Protocol Layer	This layer is concerned with the transmission of the data. Transmission Control Protocol(TCP) and User Datagram Protocol(UDP) are the two major protocols for this layer.
Internet Protocol Layer	This layer consists of packet construct which is to be transmitted. This serves the form of Internet Protocol(IP) which carries a packet that includes a source IP address, destination IP address and the actual data to be delivered. IP directs packets to a specific



	computer using an IP address.
Hardware Layer	This is the lowest level of the TCP/IP protocol stack. This hardware layer performs the function of encapsulation of IP packets into frames for transmission, mapping IP addresses to physical hardware addresses (MAC Addresses) and the use of protocols for the physical transmission of data.

## Activity 2

Explore TCP/IP protocol and functions performed at each layer.

### 1.3. ISP and its Working

Internet Service Provider (ISP) is a company that provides Internet access to individuals and other companies. ISP also provides many other services such as website building and hosting. ISP is sometimes used as an abbreviation for *Independent Service Provider* to distinguish a service provider that is an independent, separate company from a telephone company. The larger ISPs have their own high-speed lines to avoid complete dependence on the telecommunication providers and provide better service to their customers.



ISP provides access to Internet.

## Activity 3

List at least 4 Internet Service Providers in your country/city.

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_

### 1.4. World Wide Web (WWW)

The Web or World Wide Web (WWW) is a huge collection of electronic documents stored on many computers called servers around the world. The documents are formatted in a *markup language* called HTML that links documents so that you can jump from one document to another by simply clicking on the links or hot spots.



In WWW, documents are interlinked by hypertext links and can be accessed via Internet.

A computer, a modem (or some other connection device), a phone line, a software called a “browser” and an account with ISP are needed to connect with World Wide Web (WWW). A browser interprets HTML code to display text, images, sound and runs animation.



A browser is a software that can be used to access the Internet and view web pages on a computer.

The linked documents are available at different Internet sites. The World-Wide Web also provides access to many other tools which are widely used as the major means of accessing Internet resources.

Some important features of WWW are listed below:

1. WWW lets you retrieve information in a quick and easy way.
2. WWW is built on the technology called hypertext. Hypertext link are used to link web pages together.
3. WWW shares information, resources and services with widest possible community of users. The user can access the WWW on Apple, UNIX, Macintosh, Windows and other operating systems.



#### Activity 4

List three web browsers that you are familiar with.

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

### 1.5. Website and Domain Names

A website is a collection of related web pages (files) that has a beginning file called a *home page* that can be accessed using a browser. A company or an individual tells you how to get to their website by giving you the address of their home page. From the home page, you can access other pages on their site. For example, the website for AIOU has the home page address <http://www.aiou.edu.pk> from where you can access different pages related to the university.



A website has a collection of related web pages that you can access over Internet.

Any business, government, or person can create a website on the Internet. Today, the Internet consists of millions of websites created by different people and organizations.

#### 1.5.1. Domain Names

A domain name is a name used to identify a website on the Internet. Websites are associated with IP addresses. Domain name is the human readable version of the IP address. Domain names are used in



URLs to identify specific web pages. For example, in the URL <http://www.microsoft.com>, the domain name is microsoft.com.



A domain name locates an organization or an entity on the Internet.

## Activity 5

List domain names of some popular websites.

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_



## 1.6. Web Hosting

Web hosting service helps to publish or upload a website so that other people may see it. Web hosting service stores your web pages on powerful computers (Web Servers). When someone types your web address (such as [www.aiou.edu.pk](http://www.aiou.edu.pk)) in the browser, the browser connects to the web server holding the web pages to retrieve the requested web page.



A web hosting service provider provides you space to store your web pages on its server and charges you for the service.

### 1.6.1. Types of Web Hosting

With advancement in technology, different types of web hosting services have appeared to meet the needs of websites and customers. Some of the common web hosting are:

1. Shared Web Hosting
2. Dedicated Hosting
3. Cloud Hosting
4. Website Builder

These different types of hosting are discussed in unit 9.

## Activity 6

Name a few companies in your area that provide web hosting services.

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

## 1.7. Website Development Process

Website development process consists of various steps. It involves from gathering information to the creation of websites and finally to the maintenance to keep a website up to date and current.



The web developer use a development process to develop large and complex website in a systematic way.

A typical web development process has the following steps or phases.

1. Information gathering
2. Planning
3. Design
4. Development
5. Testing and delivery
6. Maintenance

We briefly discuss the above mentioned phases.

### Phase 1: Information Gathering

The first step in the development of a website is to gather information. During this phase, we need to understand the following:

- **Purpose:** What is the purpose of the site? For example, the purpose of a website may be to provide information, promote a business or sell a product?
- **Goals:** What do we want to accomplish by the website? Two of the more common goals are either to promote business or share information.
- **Target Audience:** Determine a group of people who are likely to visit and use the website. The interest of the target audience helps us to determine the best design for the website.
- **Content:** Determine what kind of information will the target audience be looking for on the site?



Information gathering or the requirements analysis is to know what we want to put on a website and what functionality we want to provide to the website visitors.

### Phase 2: Planning

After information gathering phase, it is time to put a plan for the website as a site map. A site map consists of all main topical as well as sub topical areas of the site. This helps to determine what content will be on the site and how they will be organized. During this phase, we also decide which technology is to be used for the development of this specific website.



Planning involves developing the sitemap of a website.



## Phase 3: Design

After the planning phase, it is time to determine the look and feel of the website to presents the content and services to the target audiences as determined in the information gathering phase. As a part of the design phase, it is important to develop basic elements of website such as the company logo. A prototype for the website may also be developed to see what the final design will look like.



Design is about determining the look and feel of a website.

## Phase 4: Development

Development stage is the point where a web designer takes all of the individual graphic element from the prototype and develops the actual functional website. This is typically done by first developing the home page. The home page serves as a template for the content pages of a website and it contains the main navigational structure of the website. Once the homepage is created, the contents can be distributed to a number of web pages as specified in the site map. Other elements such as contact form and shopping carts are also implemented and made functional.



Development is about implementing the design using web development technologies and tools.

## Phase 5: Testing and Delivery

During this phase, the website is tested to ensure that it confirms to the design and all the links and functionality provided by the website work as expected without error. We test the website for the functionality of the forms and scripts as well as compatibility issues. As a part of testing, we validate the code to ensure that the site meets the current web development standards. Once tested, the website is ready to go on-line.



Testing is about ensuring that a website works as expected.

## Phase 6: Maintenance

Once a website goes online, a number of errors and performance issues may be reported to the development team. During the maintenance phase, we resolve not only the reported issues but also improve the website by possibly, adding, updating contents and services.



Continuously update a website and resolve issues/errors.



## Activity 7

Your friend requested you to develop a personal website for him. Do the following:

1. Get his requirements and document them
2. Propose an initial design of the website.



## Video Lecture

<https://youtu.be/TjhPmC1h1cl>



### 1.8. Static vs. Dynamic Web Pages

A *static web page* is a page whose content are fixed and do not change unless changed by the developer himself. With static websites, requests for pages are handled by a web server delivering the content of these HTML files, "as is". Static websites are the cheapest to develop and host, and many smaller companies use static website to get a web presence.



Static website contains a collection of web pages that are delivered, on request, "as is".

On the other hand, the content of a dynamic web page are not fixed but they are generated dynamically on the fly when requested by user. That is, the content of a dynamic page may be different from different user depending upon his request. For each request, the web page is constructed from information stored in databases and programming logic.



A dynamic website builds a web page on the fly, when requested.

### Activity 8

Give an example of a static website and a dynamic website that you have visited.



### 1.9. Web Development Technologies

There are a number of technologies available for developing a website. A web developer should have the knowledge of the following basic technologies to develop a fully functional website.

1. HTML
2. CSS
3. JavaScript
4. Databases



A brief description on the above technologies is given below.

## HTML

HTML is a Hyper Text Markup Language used to develop a web page. It provides a set of tags that are used to structure the content of web pages and create links with other pages. We will learn more about HTML in Unit 2, 3 and 4.



HTML is a document-layout and hyperlink specification language.

## CSS

Cascading style sheets (CSS) is a language that is used to define styles for a web page or web pages. CSS describes how elements of a web page are to be displayed on screen. CSS handles the look and feel of a web page. Using CSS, we can control the color of the text, the style of fonts, the spacing between paragraphs, columns sized and layout, background images or colors, layout designs, and so on. We will learn more about HTML in Unit 5 and 6.



CSS is used to add style in your web pages.

## JavaScript

Javascript is a light weight programming language which is commonly used in developing dynamic web pages. We will learn more about HTML in Unit 7 and 8.

## Databases

A database is a collection of logical related data that can be easily accessed, managed, and updated. A database management system (DBMS) is a software for creating and managing databases.

## 1.10. Client Side vs. Server Side Scripting

### 1.10.1. Client Side Scripting

A script is a set of instructions written in some programming language. Client-side scripting refers to the computer programs that are executed on the client-side, by the web browser. The source code is transferred from the web server to the user's computer over Internet that runs in the browser.



A client side script runs inside a browser on the client computer.

### Activity 9

Give example of some programming languages which are used for client -side scripting.

### 1.10.2. Server Side Scripting

A server side script refers to a computer program that runs on the server before the data is passed to the user's browser. A server-side script/code never reaches the user, it is executed on the server and only the output is sent to the web browser.



A server-side script runs on the server.

#### Activity 10

Give examples of some programming languages which are used for server-side scripting.

### 1.10.3. Common Gateway interface

The web is a *client-server system*. A web browser residing on your computer acts as a client which can request web pages from different web servers. This can be shown as a diagram:

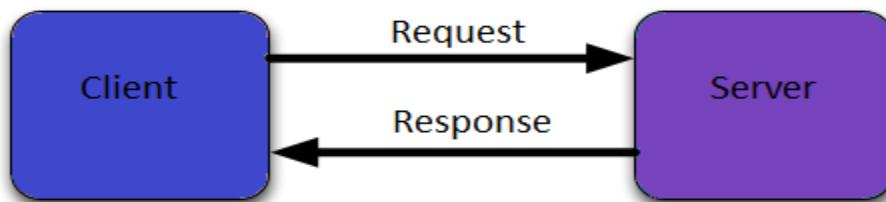


Figure 1.3: Client Server request response model

In order to deliver dynamic content, we need to extend the abilities of the web server so that it can do more than merely sending static web pages in response to the client requests. The **common gateway interface (CGI)** provides a mechanism to do this. The server is normally asked for a file that has a **.htm** or **.html** extension for static web pages. The extension would be different for dynamic web pages, for example **.cgi** or **.php**. If a request comes in with one of these extensions, the web server passes the request to the CGI which interprets it correctly and executes the server side script to generate a web page that is sent to the client's browser.

## 1.11. Hypertext, Hypermedia Tools and IDEs

The hypertext pages are inter-connected by hyperlinks typically activated when a user clicks on a link. Hypertext is not only used to describe text but sometimes used to describe table, images and other presentational content forms with hyperlinks. It enables user to share information over the Internet easily.



The hypertext pages are inter-connected or linked by hyperlinks.

### 1.11.1. Hypermedia Tools

Hypermedia can be developed by different tools. Multimedia development software such as Adobe Flash, Adobe Director etc. can be used to develop multimedia content such as images and video clips. Some database software's such as Visual FoxPro and FileMaker Developer may be used to



develop stand-alone hypermedia applications, with emphasis on educational and business content management.



A number of tools are available for developing multimedia contents.

Documentation software such as the Microsoft Office Suite and Office allow for hypertext links to other content within the same file, other external files, and URL links to files on external file servers. Any text editor may be used to build HTML files, accessible by any web browser. CD/DVD authoring tools such as DVD Studio Pro may be used to hyperlink the content of DVDs for DVD players or web links when the disc is played on a personal computer connected to the Internet. There are some HTML editors which are based on WYSIWYG (What You See Is What You Get). These editors provide rich text editing tools for developing web pages. Some of the important tools include Dreamweaver, Notepad ++, Sublime Text 3 etc.



## Unit Summary

In this unit, we learned about the concept and working of Internet and its services. We also learned the differences between static and dynamic web pages. We have explored web development process and its associated activities during each phase. We also learned about hypertext and hypermedia tools.



## Self Assessment Questions

Select the correct answer.

1. An Internet is -----
  - A. a single network
  - B. a vast collection of different networks
  - C. interconnection of local area networks
  - D. none of the mentioned
  
2. To join the Internet, the computer has to be connected to a/an
  - A. Internet architecture board
  - B. Internet society
  - C. Internet service provider
  - D. none of the mentioned
  
3. Which one of the following protocols is not used in Internet?
  - A. HTTP
  - B. DHCP
  - C. DNS
  - D. none of the mentioned
  
4. IPv6 address has a size of
  - A. 32 bits
  - B. 64 bits
  - C. 128 bits
  - D. 265 bits
  
5. Which protocol assigns IP address to the client connected in the Internet?
  - A. DHCP
  - B. IP
  - C. RPC
  - D. none of the mentioned
  
6. Computer that requests the resources or data from another computer is called as \_\_\_\_\_ computer.
  - A. Client
  - B. Server
  
7. Software which allows user to view the web page is called as \_\_\_\_\_.
  - A. Interpreter
  - B. Internet Browser
  - C. Website
  - D. Operating System
  
8. When a user opens a website, the first page that is displayed is called \_\_\_\_\_.
  - A. Backend Page
  - B. Dead End
  - C. Home Page
  - D. None of these



9. URL means \_\_\_\_\_.
- Term used to get online program
  - Term used to describe Website
  - None of these
  - Address of the resource on the web
10. Computer which stores the different web pages is called as \_\_\_\_\_.
- Web Server
  - Web Browser
  - Service Provider
  - None of these
11. WWW, e-mail and FTP are ----- layer protocols.
- Application
  - IP
  - Transmission
  - Physical
12. A company that provides Internet access to individuals and other companies is called
- Web server
  - Telephone
  - Internet Service Provider
  - None of the above
13. In which of the phase of the web development process, we determine the purpose of the site?
- Planning
  - Information gathering
  - Design
  - Development
14. Cascading Style Sheets (CSS) is a language that is used to define styles for a web page or web pages.
- True
  - False

**Answer Key:**

1. B	2. C	3. C	4. B	5. A	6. A	7. B	8. C	9. D	10. A
11.	12. C	13. B	14. A						



## Review Questions

Write short answers of the following questions.

1. What functions are performed by each layer of TCP/IP protocol?
2. Differentiate between static and dynamic web pages.
3. Briefly explain the concept of IP address?
4. How would you differentiate between a URL and domain name?
5. Discuss different activities that are performed during each phase of web development process.
6. Explain the basic concept of web hosting?
7. Explain the concept of Internet and WWW?



## References and Further Reading

1. MacDonald, M. (2013). HTML5: The Missing Manual. O'Reilly Media, Inc.
2. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. Sitepoint.
3. Stevens, L., & Owen, R. J. (2013). The Truth about HTML5. Apress.
4. Hogan, B. P., Warren, C., Weber, M., Johnson, C., & Godin, A. (2012). Web Development Recipes. Pragmatic Bookshelf.
5. Server Side Scripting. Available on: <http://www.pythonschool.net/serversidescripting/>
6. Website. Available on: <http://www.computerhope.com/jargon/w/website.htm>



## UNIT 2

# HTML Basics

## Introduction

When we look at web pages in a Web browser, most of the time we see they contain styled text (rather than plain text), images, video clips, and background music. They include drop-down menus, search boxes, or links that can be followed to access other pages. Usually, a web page designer may use several technologies (such as CSS, JavaScript, AJAX, JSON) to control the user interface but most fundamentally, developers write web pages in HTML, without it no web page can be developed.



Web pages may contain styled text, images, video clips and links to other pages.

The Hyper Text Markup Language (HTML) is a document-layout and hyperlink specification language. It is a simple, yet powerful, platform independent language for creating web pages. HTML defines a set of commands (called tags) that are placed around objects (i.e. text, pictures, sounds, table, forms etc.) to control their appearance in a browser. It also specifies how to create links to other documents.



HTML is a document-layout and hyperlink specification and platform independent language.

The World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG) define and maintain the HTML standards and specifications worldwide. The HTML specification defines a single language that can be written either with the relaxed HTML syntax or the stricter XML syntax (Extensible Markup Language). HTML only describes the meaning of the content, not style and formatting.



HTML is not a styling and formatting language.

This unit provides an Introduction to HTML 5. As mentioned before, HTML is a markup language. The word markup was used by editors who marked up manuscripts when giving instructions for revisions. "Markup" now means something slightly different: a language with specific syntax instructs a web browser how to display a page. HTML uses a pre-defined set of elements to define content types. Elements contain one or more "tags" to express content.



## Unit Outcomes

Upon completion of this unit, you should be able to:

1. Use a text editor (such as notepad) to type and save your HTML code and view its output in a browser.
2. Develop web pages using basic HTML tags to format text
3. Create ordered and unordered lists.



## Terminologies

- Web page:** A document that can be accessible on the Internet.
- Lists:** Group of items that are related to each other in a specified order.
- Tag:** Tag specifies the format of a document, paragraph or text in a web page.

## 2.1. Developing Your First Web page

Before we formally discuss and learn HTML itself, let us develop our first web page. A web page can be created by using professional HTML editors. However, to learn HTML, we will create HTML documents using a text editor (such as Notepad) as long as it can be stored on a disk in a text file format. That is because even though HTML documents include elaborated text layout and pictures, they are just plain ASCII text files themselves. HTML file are generally given .html extension.



Web pages may contain styled text, images, video clips and links to other pages.

While not needed to create an HTML document, having a latest version of a World Wide Web browser such as Netscape Navigator or Microsoft Internet Explorer on your computer to view your work is necessary. However, you do not need an Internet connection to view your own HTML documents.

To create our first web page with Notepad, follow these four steps.

**Step 1:** Open Notepad in Windows (Operating System).

**Step 2:** Type your HTML code (given in the figure below) in Notepad. At this stage, you do not need to understand various HTML tags (written inside <>) used in the document.

### </> Code 2.1

```
<!DOCTYPE html>
<html>
<head>
<title>Enter Page Title Here</title>
</head>
<body>
  <h1>First HTML Document</h1>
  <h2>I am very Excited</h2>
  <h3>Wish me best of luck</h3>
</body>
</html>
```

**Step 3:** Save your document on your computer by selecting File > Save As (go to the menu bar and click File and then click on Save As). You will see the following screen:

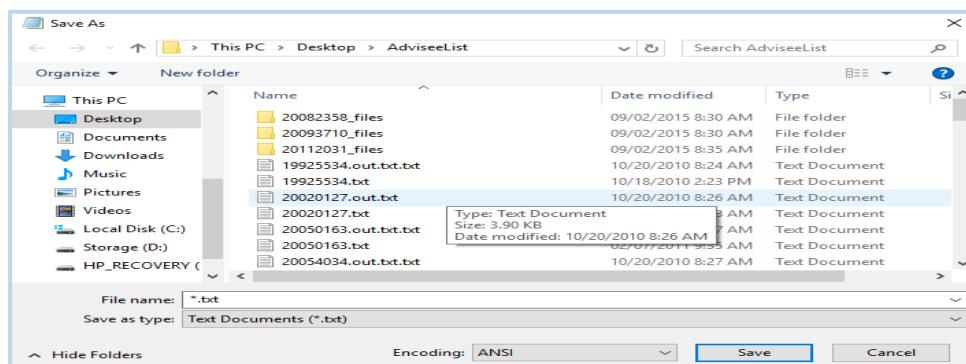


Figure 2.1: Save as screen shot



Select the folder in which you want to save the file, enter file name (such as new.html) and set the encoding to UTF-8 and click on Save button. Note that your file name should have extension .html.

**Step 4:** View the HTML Page in Your Browser. Open the saved HTML file by double clicking on the file, or right-click and choose "Open with" and select your browser.

If everything goes as planned, your browser should display something similar to the following figure 2.2.

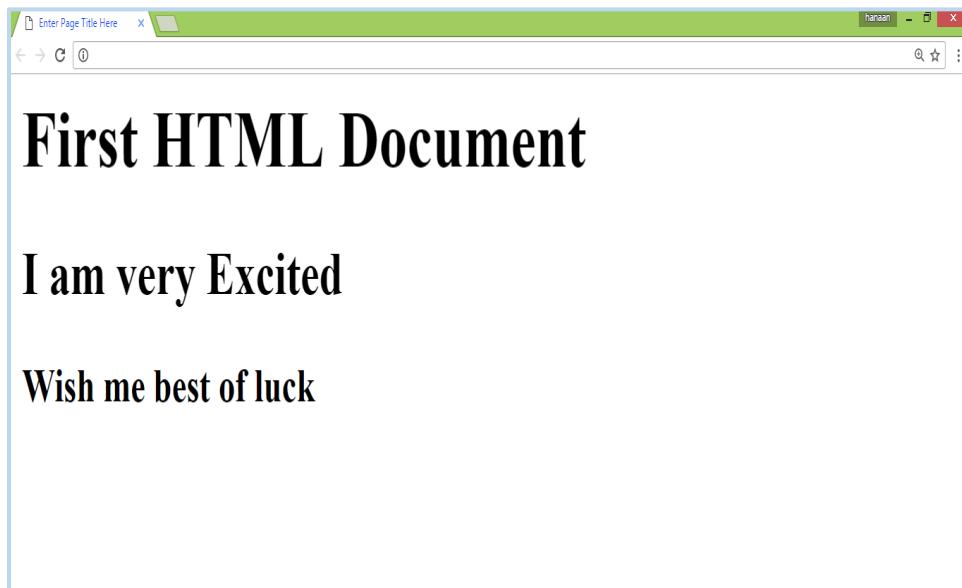


Figure 2.2: Output of code 2.1



## Video Lecture

<https://youtu.be/iBKj3UzOeng>



## 2.2. HTML Document Structure

HTML documents are composed of two main parts: a *head* and a *body*. Both head and body of a document are enclosed within `<html>` and `</html>` tag pair. The head of the document is enclosed between `<head>` and `</head>` tags. The body of a document contains all the information and can

contain images, links to other resources, lists, menus, entry fields, and plain text. The body of a document is enclosed between a `<body>` and a `</body>` tags.

The simple structure of an HTML document is given below.

**</> Code 2.2**

```
<!DOCTYPEhtml>
<html><!-- Html tag start here -->
<head><!-- head tag start here -->
    <title>Enter Page Title Here</title>
</head><!-- head tag end here -->
<body><!-- body tag start here -->

</body><!-- body tag end here -->
</html><!-- Html tag End here -->
```



- All HTML documents must start with a document type declaration:  
`<!DOCTYPE html>`.
- The HTML document itself begins with `<html>` and ends with `</html>`.
- The visible part of the HTML document is between `<body>` and `</body>`.

## 2.3. HTML Tags

The HTML language uses markup **tags** to identify the elements of a document. In creating your first web page, you have already used some of the most basic tags of HTML. Mostly, the tags are simple to understand and use, since they consist of common words, abbreviations, and notations. Every tag consists of a tag name, followed by an optional list of tag attributes. All tags are written inside a pair of angle brackets (`<>`). Tags and attribute names are not case-sensitive. For example, the tag `<html>` can be written as `<HTML>`, `<html>`, or `<Html>` without making any difference.



HTML provides a number of tags to organize and structure a web page. Tags are written inside `<>`.

Most of the tags are written in pairs: a start tag (e.g. `<HTML>`) and an end tag (e.g. `</HTML>`). There are only a few tags that do not have an end tag.



Most of the tags appear in pairs: an opening tag `<>` and a closing tag `</>`.

The HTML tags can be classified into the following categories for our convenience:

- Basic tags
- Formatting tags
- Lists



- Tables
- Images
- Links
- Forms and Input
- Frames

We will discuss basic tags, formatting tags and lists tags in this unit. The remaining tags will be discussed in the coming units.

## 2.4. Basic HTML Tags

The basic tags are HTML tags which exist in all HTML document. Some of the basic tags are considered to be core HTML tags, that is, without these tags, an HTML document cannot work. These tags are: `<!DOCTYPE html>`, `<html>`, and `<body>`. The rest of the basic tags are necessary for the basic information about the HTML document and for the very basic layout, such as the document title, search keywords, line breaks and paragraphs.



All HTML documents have a required structure that includes the following declaration and elements: `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>`

Following is a summary of the basic tags with a brief description.

**Table 2.1:** Basic tags

Tag	Description
<code>&lt;!DOCTYPE html&gt;</code>	Defines the document type
<code>&lt;html&gt;&lt;/html&gt;</code>	Defines an HTML5 document
<code>&lt;head&gt;&lt;/head&gt;</code>	Contains all the information about the HTML document, including the meta tags and title tag
<code>&lt;body&gt;&lt;/body&gt;</code>	Contains the content of the HTML document
<code>&lt;title&gt;&lt;/title&gt;</code>	Specifies the title of the HTML document
<code>&lt;meta&gt;</code>	Describes the metadata of the html document, such as the character set, keywords used for searching, and the author of the document
<code>&lt;h1&gt;&lt;/h1&gt;to &lt;h6&gt;&lt;/h6&gt;</code>	HTML headings
<code>&lt;p&gt;&lt;/p&gt;</code>	Paragraph
<code>&lt;br&gt;</code>	A single line break
<code>&lt;!-- . . . --&gt;</code>	A comment

We will now discuss each of the basic tag in a little more detail.

### **<! DOCTYPE html>**

This tag must be placed at the top of a web page, above everything else. This tag informs a web browser that this is an HTML5 page to ensure that all the properties of HTML5 would be recognized and applied in the output.



HTML5 document starts with <!DOCTYPE html>. It does not have an end tag.

### <html></html>

The <html> tag is placed right below the <!DOCTYPE html>. It tells the browser that this is an HTML document and represents the root of an HTML document. This tag is considered to be the container of all the other HTML elements. It has a closing tag, </html>, which is placed at the very bottom of the HTML code as shown in the following figure2.3.

```
1  <!DOCTYPE html>
2  <html>
3
4  html code goes here!
5
6  </html>
```

Figure 2.3: HTML starting and closing tag.



<html> tag comes immediately after <!DOCTYPE html>. Your HTML document ends with </html>.

### <head></head>

The <head> tag is the container that holds all the <head> elements. The <head> tag has no attributes and serves only to encapsulate other tags such as <title> of the HTML document, meta data that are used to provide information about a web page, keywords that are used for search engine and the tags that are used to include external documents such as JavaScript and CSS files.

### <title></title>

The <title> and </title> pair contains the title of the document. The title of a web page is displayed in the browser's title bar and the user's bookmark. It must be carefully chosen in order to give good estimation about the subject of the web page.

### <meta>

The <meta> tag provides metadata about the HTML document. We typically use metadata elements to specify page description, keywords, author of the document, last modified, and other metadata. The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.



<meta> tag is used to provide useful information about your documents.

We can include one or more <meta> tags in your document based on what information you want to keep in our document. Metadata is always passed as name/value pairs. Let us see how to specify some important information about a document using <meta> tag.



<meta> does not have a closing tag.



**Specifying keywords:** We can use `<meta name="keywords" content="list of keywords">` to specify important keywords related to the document. These keywords are used by the search engines while indexing a web page for searching purpose. This is how you specify important keywords

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

**Providing document description:** We can use `<meta>` tag to give a short description about the document as

```
<meta name="description" content="HTML Unit 2">
```

**Specifying the document author:** We can set an author name in a web page as shown below:

```
<meta name="author" content="Amjad Mahmood">
```

**Specifying document revision date:** We can use `<meta>` tag to give information about when last time the document was updated as the following code illustrates.

```
<meta name="revised" content=" HTML Unit 2, 3/7/2014">
```

**Specifying document refreshing duration:** We can also use `<meta>` tag to specify a duration after which a web page will keep refreshing automatically. The following code can be used to refresh a page after every 5 seconds.

```
<meta http-equiv="refresh" content="5">
```

Have a look at the following code to see various tags that are used in `<head>` section.

#### **</> Code 2.3**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
<meta charset="UTF-8">
<meta name="description" content="HTML Unit 2">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="Amjad Mahmood">
</head>
<body>
</html>
```

#### **<body></body>**

The `<body>` and `</body>` container contains the part of a page that will actually be displayed. This tag defines the HTML document's body, where it contains all the contents and tags of the HTML document, such as texts, hyperlinks, images, tables, lists, paragraphs and so on. Any HTML code that decides how the HTML file looks like, and how it behaves must be written in the opening and closing `<body>` tag.



Whatever you want to display in a browser, put in between `<body>` and `</body>`.

## Activity 1

Write a text “HTML Intro” in `<title></title>` tag and “Hi I am new to Html” in `<body></body>` tag and see how text is displayed in a web browser.

Let us now look at some of the tags that can be used between `<body></body>` tag pair.

### Headings: `<h1></h1>` to `<h6></h6>`

The text of documents is generally arranged in a number of sections. Each section may further be divided into subsections (like this book). This makes the document more readable and manageable. HTML provides tags that can be used to organize our web page into a number of sections and subsection with appropriate heading.

HTML defines six levels of headings. The six heading tags are written as `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. Each heading tag has a corresponding end tag (i.e. `</h1>`, `</h2>` ... `</h6>`) which should never be omitted. Tag `<h1>` is the biggest and used for the most important heading, whereas `<h6>` is the smallest and least important.



Documents can be divided into sections and subsection with appropriate headings. Heading tags help you just to do that.

The text written within a heading tag rendered by the browser depends on the available display technology. The browser may choose to center, embolden, enlarge, italicize, underline, or change the color of headings to make a heading stand out within the document. However, most browsers use a diminishing character size for the sequence of headers. Another important point to note is that headings always reside on their own line when displayed in a browser.



By default, it is your browser that determines how the heading should be displayed.

Most of the browsers display the heading left-aligned. We can alter this with ALIGN attribute with one of these values: left, center and right. For example, to display the heading of level 1 in the center, we write

```
<h1 align="center">Level 1 heading in center </h1>
```



You can change default alignment of heading with align attribute.

Here is an example of HTML code for all the different heading levels and the resulting output in a browser.

#### `</> Code 2.4`

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



```
<title>Paragraphs In HTML</title>
</head>
<body>
    <h1>First Heading </h1>
    <h2>Second Heading </h2>
    <h3>Third Heading </h3>
    <h4>Fourth Heading </h4>
    <h5>Fifth Heading </h5>
    <h6>Sixth Heading </h6>
</body>
</html>
```

The output of above code is shown in figure 2.4.

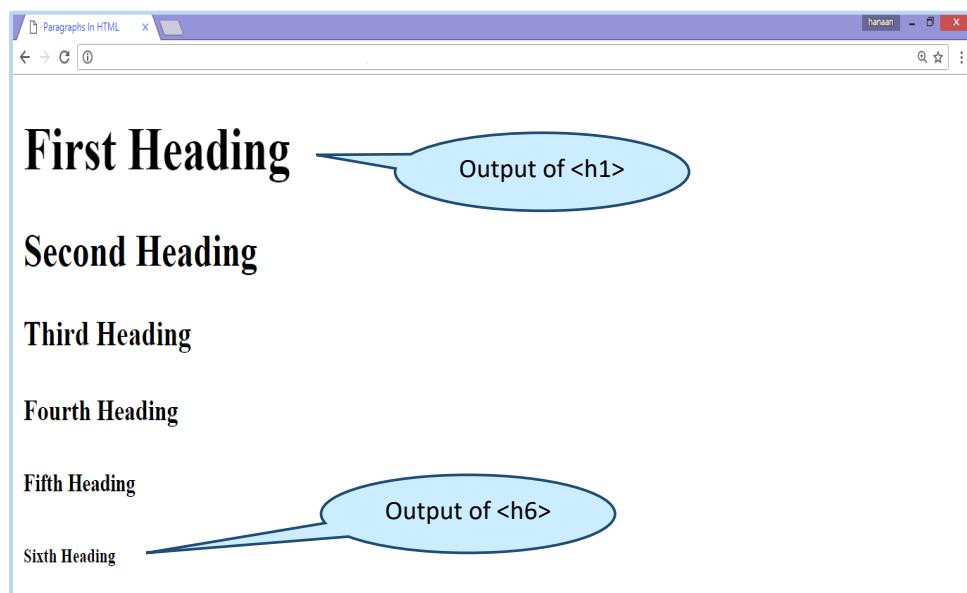


Figure 2.4: Output of code 2.4

#### Paragraphs: <p></p>

The paragraph tag, `<p>`, defines a paragraph. The paragraphs of a document are separated by inserting a blank line between them. When a browser encounters a new paragraph tag, `<p>`, it typically inserts one blank line plus some extra vertical spaces into the document before starting a new paragraph.



Paragraph tag `<p>` is used to insert paragraph. By default, paragraphs are left justified.

Carefully look at the output of the following HTML code (Figure: 2.4) and see how paragraphs are displayed in a browser.

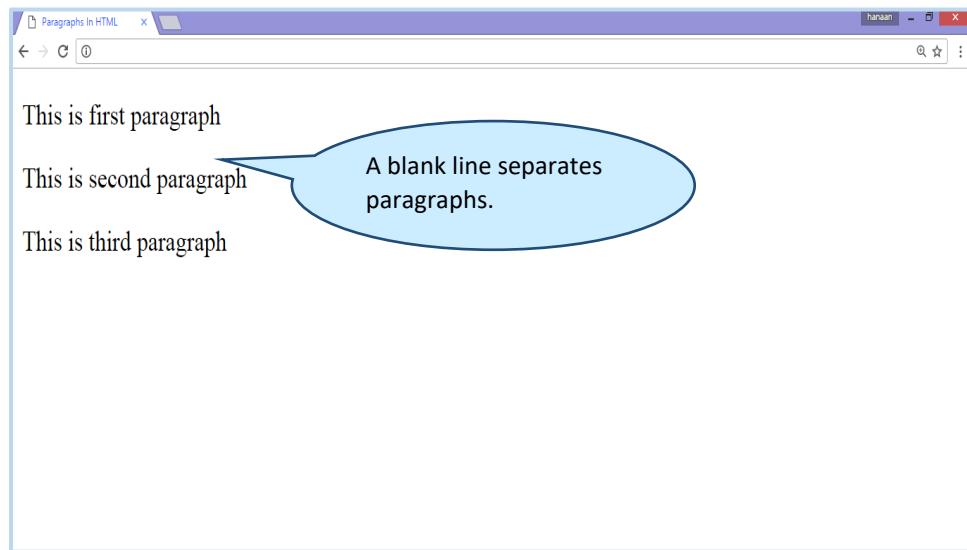
#### </> Code 2.5

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraphs In HTML</title>
</head>
```

```
<body>
  <p>This is first paragraph </p>
  <p>This is second paragraph </p>
  <p>This is third paragraph </p>

</body>
</html>
```

The output of above code is shown in figure 2.5.



**Figure 2.5:** Output of code 2.5

It can be observed in the above code that there are three paragraphs in the documents, each one starting with a `<p>` and ending with `</p>` tag. Paragraphs are, by default, left justified.



You can change the alignment of paragraph by using style attribute in `<p>` tag.

However, we can change paragraphs alignment by using style attribute of `<p>` tag as following example demonstrates (we will learn more about it in Unit 5).

**</> Code 2.6**

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraphs In HTML</title>
</head>
<body>

  <p>This paragraph is default</p>
  <p style = "text-align: right;">
    This paragraph is right justified</p>
  <p style = "text-align: center;">
    This paragraph is in the middle</p>
  <p style = "text-align: left;">
    This paragraph is left justified</p>
</body>
```



```
</html>
```

The output of above code is shown in figure 2.6.

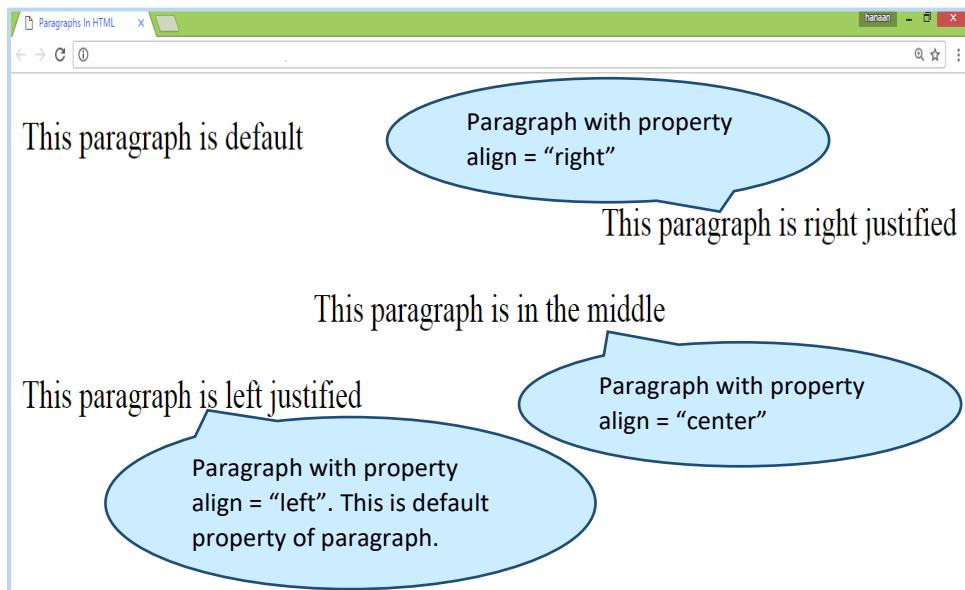


Figure 2.6: Output of code 2.6

## Activity 2

Write HTML code to display the following text. Use appropriate heading and paragraph tags.

### CREATING A WEB PAGE

Creating a web page is very simple. It does not need any prior knowledge of computing and the Internet.

What is important to learn is your commitment. The best way to learn is to start developing your pages starting with a simple one.

### TIPS

Learn by doing it.

HTML is not case sensitive. So don't worry how you type it. Learn at your own pace.

Practice, Practice and Practice.

### Line break: <br>

The <br> tag is used to force a line break, that is, it forces the text after it to be displayed on the next line. This tag is called an empty tag because it does not have a closing tag. We do not need to use this tag with paragraphs, because a line break automatically inserted by <p> tag.



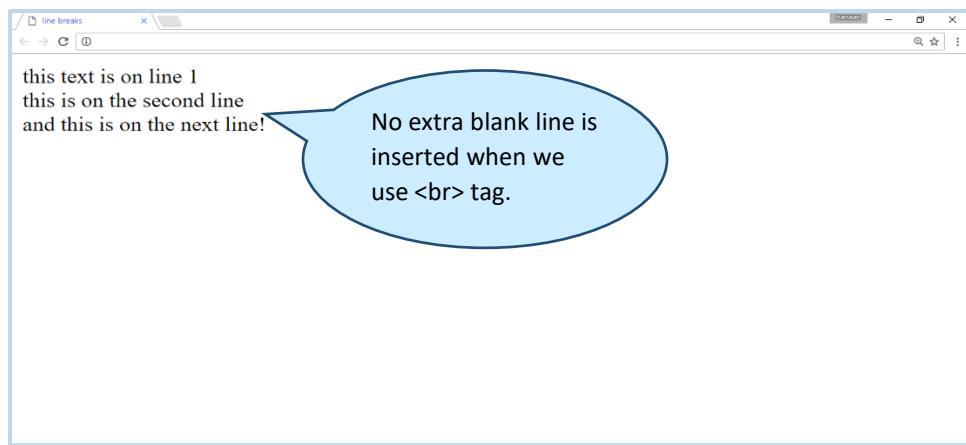
Break line tag <br> is used to give a line break. <br> does not have an end tag.

See the output of the following code to see how <br> tag forces a line break.

**</> Code 2.7**

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraphs In HTML</title>
</head>
<body> This text is on line 1<br>this is on second line
<br>and this is on next line
</body>
</html>
```

The output of above code is shown in figure 2.6.



**Figure 2.7:** Output of code 2.7

### **Activity 3**

Write HTML code to display the following text. Use appropriate tags.

This is my favorite poem

#### **Be Glad Your Nose is on Your Face**

By Jack Prelutsky

Be glad your nose is on your face,  
not pasted on some other place,  
for if it were where it is not,  
You might dislike your nose a lot.

### **2.5. Comments**

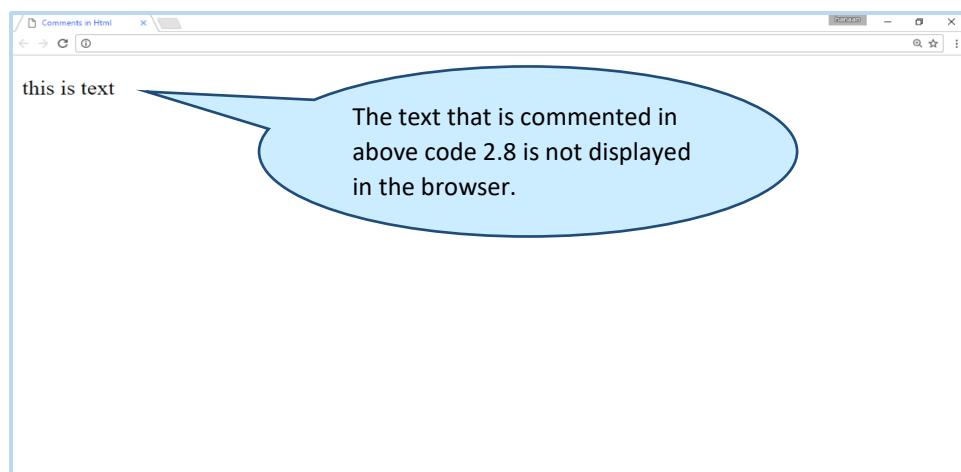
It is a good practice to use comments to organize your code in a way that is readable to others who check and modify your code. You can write a comment wherever you like within the HTML code. A comment is written between `<!-- -->`.

**</> Code 2.8**

```
<!DOCTYPE html>
<html>
<head>
<title>Comments In HTML</title>
</head>
<body>
    <!-- Start of Paragraph -->
    <p>this is text </p>
</body>
</html>
```

Here is the example that clearly shows how comments are used. Note that the lines of code that are commented are not displayed in the browser.

Output of code 2.8 is shown in figure 2.8.



**Figure 2.8:** Comment code 2.8



## 2.6. Formatting Tags

HTML offers a number of tags to change appearance of text. These tags can be grouped into two categories: *content-based* or *logical styles* and *physical styles*. The contents-based style tags inform the browser that the enclosed text has a specific meaning, context, or usage. The browser chooses an appropriate font style and formats the enclosed text in a manner consistent with that meaning, context, or usage. The physical styles are used when we intend to display text explicitly using a pre-determined font style, for example, italic or bold.



Formatting tags are used to give appropriate font style to the specific text.

### 2.6.1. Content Based Tags

As we mentioned in the previous section that content-based style tags inform the browser that the enclosed text has a specific meaning, context, or usage and a browser chooses appropriate font and formats the enclosed text properly.



Different content based tags are used to choose appropriate style that are consistent with that specific usage.

Some of the common content-based tags with a brief description are given the following table.

**Table 2.2:** Content based tags

Tag	Description
<code>&lt;em&gt;&lt;/em&gt;</code>	The <code>&lt;em&gt;</code> and <code>&lt;/em&gt;</code> tag pair tells the browser that the enclosed text be emphasized. Almost, all the browsers display emphasized text in italic.
<code>&lt;code&gt;&lt;/code&gt;</code>	The <code>&lt;code&gt;</code> tag is used to tell the browser that the enclosed text is a program code. The browser renders the enclosed text in a monospaced teletype-style font like courier.
<code>&lt;strong&gt;&lt;/strong&gt;</code>	Like <code>&lt;em&gt;</code> tag, <code>&lt;strong&gt;</code> tag is for emphasizing text. Browsers typically display the <code>&lt;strong&gt;</code> text differently from <code>&lt;EM&gt;</code> tag, usually by making the text bold so that the user can distinguish between the two. The code (2.9) illustrate the content-based tags.
<code>&lt;cite&gt;&lt;/cite&gt;</code>	The <code>&lt;cite&gt;</code> and <code>&lt;/cite&gt;</code> tag pair usually indicates that the enclosed text is a bibliographic citation like a book or a magazine. By convention, the citation is displayed in italic.

Carefully observe the following code and its output to see how these tags change the appearance of text when displayed by a browser.

#### **</> Code 2.9**

```

<!DOCTYPE html>
<html>
<head>
<title>Logical Styles</title>
</head>
<body>
    AS mentioned by Alister in <cite>The Book </cite>that a
    program code should be written as <p>
    </code>i,j:integer; <br>
    begin <br>
    i:=1; j:=2; write(i,j); <br>end; <br></code>
    <p> It is emphasized that variable
        must</strong>be<em>declared</em>
</body>
</html>

```

Output of code 2.9 is shown in figure 2.9.



AS mentioned by Alister in *The Book* that a program code should be written as

```
i,j:integer;  
begin  
i:=1; j:=2; write(i,j);  
end;
```

It is emphasized that variable **must** be *declared*

Output of code → i,j:integer;  
Output of cite → AS mentioned by Alister in *The Book* that a program code should be written as  
Output of <strong> → It is emphasized that variable **must** be *declared*  
Output of <em> → i,j:integer;

Figure 2.9: Output of code 2.9

## Activity 4

- Write a text "My name is Ali and I live in Lahore" in **<p>** tag, use font style **<strong>** to represent this text and show output on web browser.
- Write a text "I am 15 years old" in **<p>** tag, use font style **<em>** to represent this text and show output on web browser.

### 2.6.2. Physical Style Tags

HTML provides different physical styles, including bold, italic, underline etc. Some of the common physical style tags are given in the following table.

Table 2.3: Physical style tags

Tag	Description
<b>&lt;b&gt;&lt;/b&gt;</b>	Defines <b>bold</b> text
<b>&lt;blockquote&gt;&lt;/blockquote&gt;</b>	Defines a section that is quoted from another source
<b>&lt;i&gt;&lt;/i&gt;</b>	Defines <i>italic</i> text
<b>&lt;small&gt;&lt;/small&gt;</b>	Makes text smaller than others
<b>&lt;u&gt;&lt;/u&gt;</b>	Defines <u>underlined</u> text
<b>&lt;tt&gt;&lt;/tt&gt;</b>	Defined text in a mono spaced typeface.
<b>&lt;sub&gt;&lt;/sub&gt;</b>	Displays a text as subscript
<b>&lt;sup&gt;&lt;/sup&gt;</b>	Displays a text as superscript

Let us now discuss these tags in a little more detail.

**Bold: <b></b>**

The **<b>** tag is the physical equivalent of **<strong>** content-based tag. The **<b>** tag explicitly boldfaces a character, a word or a segment of text enclosed between **<b>** and **</b>**.

**Blockquote: <blockquote></blockquote>**

The **<blockquote>** and **</blockquote>** tag pair defines a section that is quoted from another source.

**Italic: <i></i>**

The **<i>** tag is like **<em>** content-based style tag. The text written between **<i>** and **</i>** is displayed in italic.

**Underline: <u></u>**

The **<u>** tag tells the browser to underline the text written between **<u>** and **</u>** tag pair.

**Typeface: <tt></tt>**

The **<tt></tt>** tag pair directs the browser to display the enclosed text in a monospaced typeface. The tags discussed above can also be embedded, that is, if we want to make a text both italic and bold, we need two tags as the following example shows.

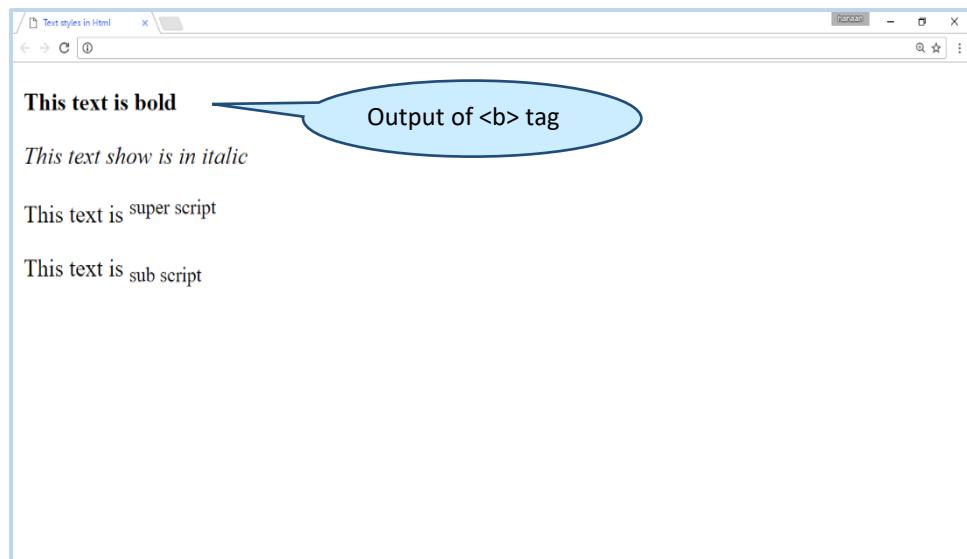
**<b><i>Bold and Italic text </i></b>**

Here is an example of HTML code and its output to change the appearance of text in a browser. Try it yourself as well.

**</> Code 2.10**

```
<!DOCTYPE html>
<html>
<head>
<title>Logical Styles</title>
</head>
<body>
<p><b> This text is bold </b></p>
<p><i> This text show is in italic </i></p>
<p>This text is <sup>super script</sup></p>
<p>This text is <sub>sub script</sub></p>
</body>
</html>
```

Output of code 2.10 is shown in figure 2.10.



**Figure 2.10:** Output of code 2.10



## Video Lecture

<https://youtu.be/8GLNpyonDAU>



### Activity 5

Write a text that uses all the physical style tags and show output on web browser.



## 2.7. Lists

A list is a collection of related items written in some special order or sequence. HTML provides a set of tags that helps us to organize contents into lists. There are two major types of HTML lists: unordered lists and ordered lists.



Lists are used to organize content in formatted list.

The HTML tags that are used to create lists are given in table 2.4.

**Table 2.4: List tags**

Tag	Description
<code>&lt;ul&gt;&lt;/ul&gt;</code>	Unordered list
<code>&lt;ol&gt;&lt;/ol&gt;</code>	Ordered list
<code>&lt;li&gt;&lt;/li&gt;</code>	Defines a list item

Let us see how these tags are used to create ordered and unordered lists.

### 2.7.1. Ordered Lists

The ordered lists are used when the sequence of list items is important. The items of an ordered list are numbered by the browsers. The numbering starts with 1 and is incremented by 1 for each successive list element.



We use `<ol>` tag for ordered list. Each list element starts with `<li>` tag.

The items of the ordered list are enclosed in `<ol>` and `</ol>` tags. Each element of the list is tagged with `<li>`, which does not have an ending tag. For example, we can create an ordered list of favorite drinks by using the following piece of code.

```
<ol>My Favorite Drinks:
    <li>Coffee
    <li>Lemon juice
</ol>
```



`<li>` tag may not have an ending tag.

The following HTML code illustrates how ordered lists are created using `<ol>` and `<li>` tags. Items in each ordered list are numbered automatically starting with 1.

#### **</> Code 2.11**

```
<!DOCTYPE html>
<html>
<head>
<title>Ordered list items in HTML documents</title>
</head>
<body>
    <!-- starting of the first ordered list -->
    <ol> My Favorite Drinks:
        <!--first item of the list-->
        <li>Coffee</li>
        <!--Second item of the list-->
        <li>Lemon juice </li>
    </ol>
    <!-- end of first list -->
    <!-- starting of the second ordered list -->
    <ol> My Favorite Foods:
        <li> Chicken burger </li>
    </ol>
</body>
</html>
```



```
<li>Chicken steaks </li>
</ol>
<!-- end of the second ordered list -- >
</body>
</html>
```

Output of above code is shown in figure 2.11.

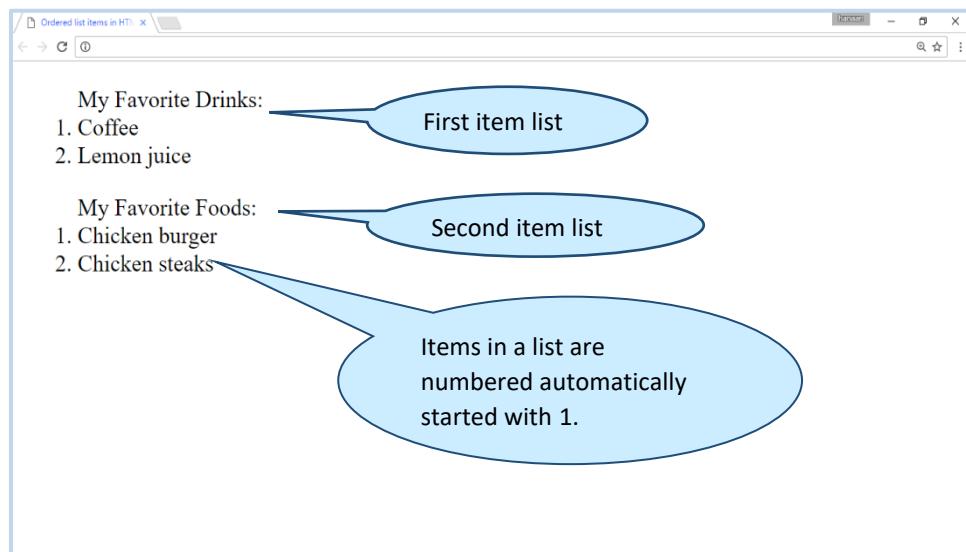


Figure 2.11: Output of code 2.11

### 2.7.2. Unordered Lists

The items of an unordered list are enclosed in `<ul>` and `</ul>` tags. Each item of the list is tagged with `<li>` which may not have a corresponding `</li>` tag. The browser adds a leading bullet character and formats every list item on a new line, indented from the left margin of the document.

For example, the following code creates an unordered list of my favorite drinks.

```
<ul>My Favorite Drinks:
  <li>Coffee</li>
  <li>Lemon juice </li>
</ul>
```



Unordered lists are created with `<ul>` tag. Each item in the list is preceded with a bullet symbol in a browser.

The following complete HTML code shows the use of `<ul>` tag.

#### `</> Code 2.12`

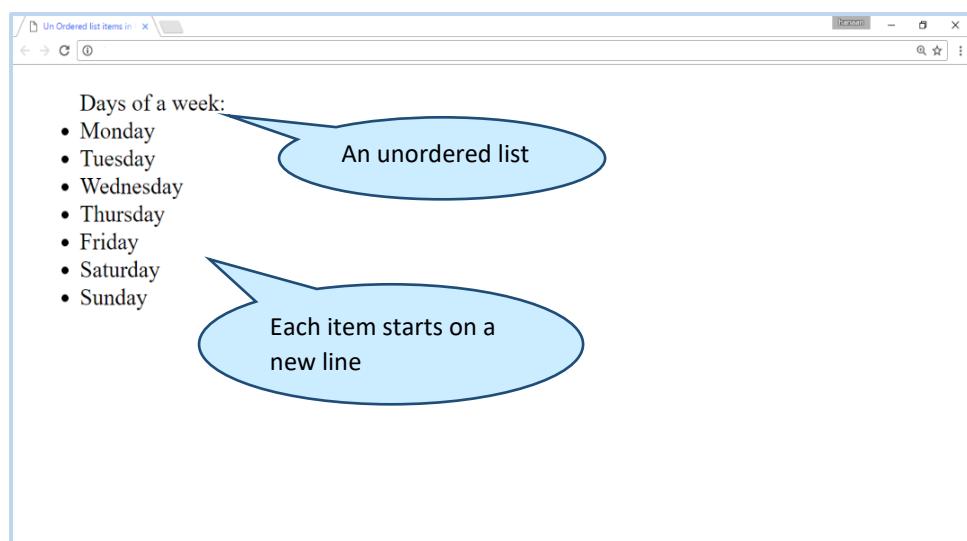
```
<!DOCTYPE html>
<html>
<head>
<title>UnOrdered list items in HTML documents</title>
</head>
<body>
<!-- starting of the unordered list -->
```

```

<ul> Days of a week:
<li>Monday</li>
<li>Tuesday</li>
<li>Wednesday</li>
<li>Thursday</li>
<li>Friday</li>
<li>Saturday</li>
<li>Sunday</li>
</ul>
</body>
</html>

```

Output of above code is shown in figure 2.12.



**Figure 2.12:** Output of code 2.12

## Activity 6

Create an unordered list of car manufacturers (Honda, Toyota, Mercedes, Ford, and Suzuki).

### 2.7.3. Nesting of Lists

HTML allows embedding or nesting of one type of list in another to produce more effective lists. The following HTML embeds unordered lists in an ordered list.



Nesting of lists allows embedding of one list into another.

#### </> Code 2.13

```

<!DOCTYPE html>
<html>
<head>
<title>Unordered list items in HTML documents</title>
</head>
<body>
  <ol>
    <li>Working days:</li>

```



```
<ul>
<li>Saturday</li>
<li>Sunday</li>
<li>Monday</li>
<li>Tuesday</li>
<li>Wednesday</li>
<li>Thursday</li>
</ul>
<li>Off day:</li>
<ul>
<li>Friday</li>
</ul>
</ol>
</body>
</html>
```

The output of above code is given in figure 2.13. The output clearly shows the difference between ordered and unordered list.

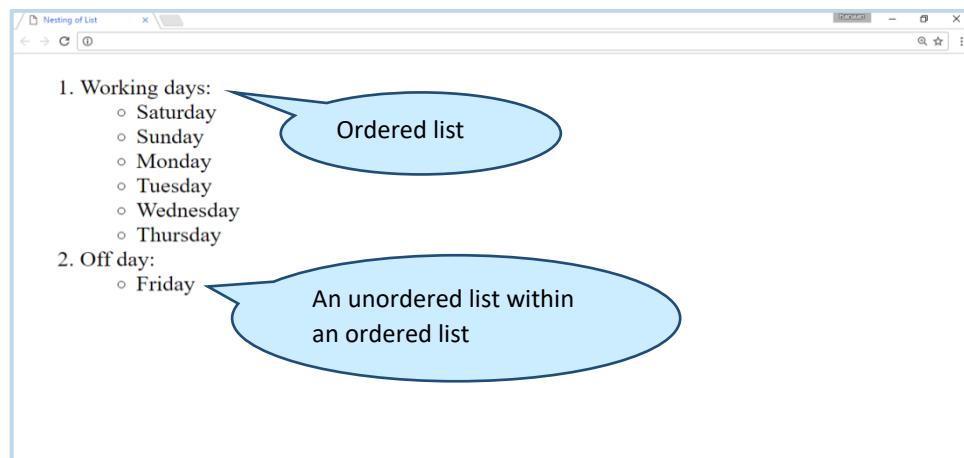


Figure 2.13: Output of code 2.13



## Video Lecture

<https://youtu.be/5yKteejDGZM>



 **Activity 7**

Write HTML code to create the following lists:

1. Subjects in Secondary School

- English
  - Mathematics
2. Subjects in Bachelor
- Mathematics
  - Physics
  - Chemistry
  - English
- 
- 

**Unit Summary**

In this unit, we learned how to use a text editor (such as notepad) to type and save HTML code and view its output in a browser. We have also learned how to use basic HTML tags with the help of examples.

---



## Self Assessment Questions

**Choose the correct answer.**

1. What does HTML stand for?
  - A. Hyper Type Makeup Longing
  - B. Hyper Text Markup Language
  - C. Hyper Type Marking Logo
  - D. Hypermedia Type Markup Language
2. What are the two things you need to create and view web pages?
  - A. A text editor and a compiler
  - B. A text editor and a web browser
  - C. A compiler and a web browser
  - D. A compiler and interpreter
3. Should <!DOCTYPE> declaration be the first thing in an HTML document?
  - A. Yes, always
  - B. Yes, but only when the Strict version is used
  - C. No
  - D. Both A and B
4. Which tag is the root tag in HTML?
  - A. <body>
  - B. <html>
  - C. <title>
  - D. <head>
5. Where is the title text displayed?
  - A. In the body of the page
  - B. In the status bar
  - C. In the title bar
  - D. All of the above
6. What are the different levels of headings in HTML?
  - A. 1-6
  - B. 2-5
  - C. 1-4
  - D. 1-3
7. What happens if you forget a slash at the end of a header tag?
  - A. A Netscape errors
  - B. Everything following will be a heading format
  - C. Nothing
  - D. It hides the text
8. Which tag is used to create paragraphs?
  - A. <p>
  - B. <param>
  - C. <pre>

- D. <b>
9. Which tag is used to define items in an ordered or unordered list?
- list tag
  - ls tag
  - li tag
  - ol tag

**Answer Key:**

1. B	2. B	3. D	4. B	5. C	6. A	7. B	8. A	9. C
------	------	------	------	------	------	------	------	------



## Review Questions

**Write short answers of the following questions.**

1. What is a WYSIWYG editor? Give some examples of WYSIWYG editor.
  2. What should be the extension of HTML document?
  3. Which two tags should be written at the beginning of an HTML document?
  4. Discuss basic HTML tags for converting a text into bold, italic and underlined.
  5. Explain the purpose of content based tags? Discuss some of the common content based tags.
  6. Differentiate between ordered and unordered lists?
- 
- 

## </> Coding Exercise

1. Using ordered and unordered lists, create a web page that displays months of a year grouped according to the number of days each month has. That is, all the months having 31 days should be grouped together followed by months having 30 days and February should be placed in a separate group
  2. Create a web page that gives basic information about yourself. The page should display your personal information, academic qualification, professional qualification, job experience etc. Your page should be well-designed and should look attractive.
  3. Create a website that provides basic information about a number of historical places in your country. Information about each place should be properly divided into sections (with proper section headings) and paragraph.
-



## References and Further Reading

1. MacDonald, M. (2013). HTML5: The missing manual. O'Reilly Media, Inc.
2. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
3. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
4. Stevens, L., & Owen, R. J. (2013). The truth about HTML5. Apress.
5. HTML Tags. Available on: <http://socnedo.org/htlm/notes-of-html/>
6. HTML. Available on: <http://www.arconlab.com/dev/web/HTMLTheDefiniteGuide.pdf>
7. HTML MCQ. Available on: <http://www.c4learn.com/mcq/html-mcq/html-mcq-html-images-multiple-choice-question/>
8. HTML Tables. Available on: [http://www.tutorialspoint.com/html/html\\_tables.htm](http://www.tutorialspoint.com/html/html_tables.htm)





## UNIT 3

# HTML Links, Images and Tables

## Introduction

In the previous unit, we learnt how to create a web page and see its output in a browser. We also learnt how we can structure a web page and use a variety of tags to change the physical appearance of text. In addition to the basic tags that we discussed in Unit2, HTML provides a number of tags that can be used to make a web page more attractive by adding multimedia elements, such as images. As we know that a website, generally, has more than one pages which are linked together using hyperlinks. Sometimes, formatting the text and data in tabular form is the best way to structure a document and HTML provides a tag that allows us to format the text and data in tabular form. In this unit, we will learn the HTML tags which are used to add images, links and tables in a web page.



## Unit Outcomes

Upon completion of this unit, you should be able to:

1. Embed graphics with suitable size and location in a web page.
2. Create hyperlinks to link web pages together.
3. Use a set of HTML tags to structure data/information in tabular form.



## Terminologies

<b>Hyperlink:</b>	A hyperlink provides connection to another document or to a specific part of the same document on the Internet.
<b>Table:</b>	Tables are used to organize data/information in rows and columns.
<b>Rowspan:</b>	It spans the specified number of rows in a cell.
<b>Colspan:</b>	It spans the specified number of columns in a cell.



## 3.1. Images and Sounds

While the body of the most of HTML documents consists of plain text, an appropriate use of images and multimedia elements make the document more captivating and attractive. Multimedia elements bring HTML documents alive, providing a dimension of valuable information often unavailable in other media.



A web page may contain text, images and other multimedia elements.

HTML standards do not prescribe an official format for images and sounds. However, the popular browsers support many different types of image and sound formats. Some of the popular formats are:

1. *Graphics Interchange Format* (GIF) was adapted during 1980s because of its widespread use on CompuServe. It enables users to exchange images between different platforms. It is now widely used in applications supporting graphical capabilities.
2. *Joint Photographic Experts Group* (JPEG) is another popular image format. It has some advantages over GIF such as it supports millions of colours and is highly compressible. Therefore, it takes up less space and downloads faster than a GIF image with similar details.
3. *Tagged Image File Format* (TIFF) was designed by Microsoft and Aldus for use with scanners and desktop publishing programs.
4. *MPEG* is an animated video standard format based on the JPEG method. The format received its name from the group that defined the standard, i.e. *Motion Picture Experts Group*. It is the most commonly used movie format for the WWW.
5. *PNG* format holds special promise as a replacement for the aging *GIF* standard.
6. *AVI* is the movie format native to Microsoft Windows.
7. *QuickTime* is another popular format developed by Apple Computer.
8. *WAV* is the native sound-file format for Microsoft Windows.
9. *MIDI* is a popular electronic-music format and is supported by a variety of browsers and platforms. MIDI files are digitally synthesized directly from a computer rather than recorded.
10. *RealAudio* format is another popular sound format. Streaming audio (RealAudio) begins to play as soon as a minimal portion of the file downloads; while that portion is playing, the next is downloaded.

Multimedia files have standard file extensions depending on the image or sound format used. The following table gives the extensions for each of the formats.

**Table 3.1: Multimedia file extensions**

MIMI Type	Standard Extensions	3-Character/ DOS Extension
Image/GIF (and GIF89a)	.gif	.gif
Image/JPEG	.jpeg	.jpg .jpe

Image/TIFF	.tiff	.tif
Text/HTML	.html	.htm
PostScript	.eps .ps	.eps . ps
Adobe Acrobat (PDF)	.pdf	.pdf
Video/MPEG	.mpeg	.mpeg .mpe
Video/AVI	.avi	Avi
Video/Quick Time Movie	.mov	.mov
Sound/AU(Basic)	.au . snd	.au .and
Sound/WAV	.wav	.wav
Sound/MIDI	.mid .midi	.mid
Sound/RealAudio	.ra .ram	.ra .ram
VRML	.vrml	.vrm

## 3.2. Adding Images in a Web Page

Images are the most important elements of web design. Almost every single web page on the Internet contains images. To add an image in a web page, we use `<img>` tag which is written as:

```
<img src = "Image Link" ... attribute-list />
```

Various attributes of `<img>` tag are described in the following table.

**Table 3.2:** Image tag attributes

Attributes	Description
<code>src</code>	This attribute defines URL of the image to be displayed.
<code>alt</code>	This attribute defines an alternate text that will be displayed when there is problem in loading an image.
<code>height</code>	This attribute defines the height of the image
<code>width</code>	This attribute defines the width of the image



We use `<img>` tag to add an image in a web page. `<img>` tag does not have a separate end tag.

The following HTML code shows how we can add images in a web page using `<img>` tag.

### **</> Code 3.1**

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
    <img src = "tree.png" />Image 1
</body>
</html>
```



When the above HTML document is opened in a browser and we have "tree.png" image on our computer, the following output will be displayed.



Figure 3.1: Output of image tag

### 3.2.1. Set Image Width and Height

We can set width and height of an image according to our requirements by using *width* and *height* attributes. Based on our requirement using width and height attributes. The width and height of an image can be specified in terms of either pixels or percentage. For example, to display an image having a width 200 pixels and height 350 pixels, we write the following code

```
<img src = "tree.png" style = "width=200 px; height=350 px;"/>
```



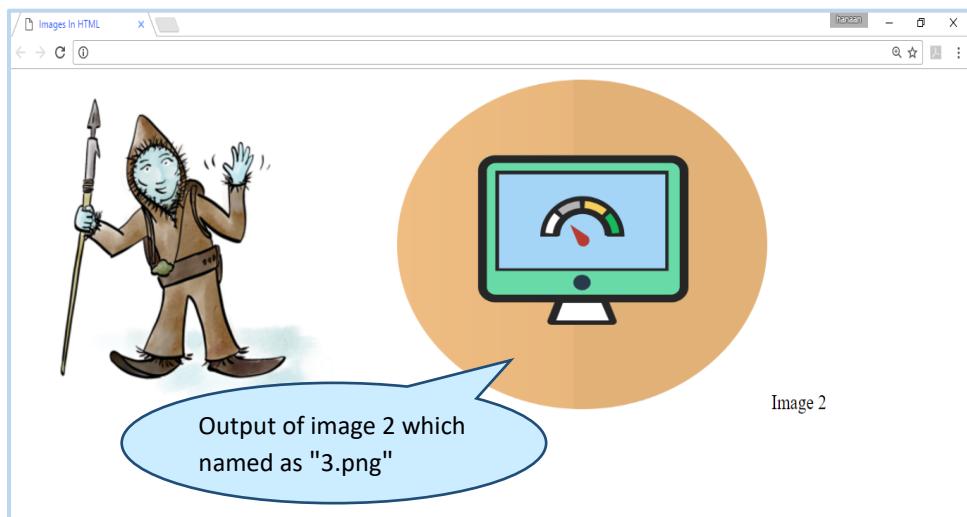
Use width and height attributes to set width and height of an image respectively.

The following HTML code displays two images with a specified width and height. Look at the output of the code to see how the browser adjusts the images according to the specified width and height.

#### </> Code 3.2

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
    <img src = "tree.png" style = "width=200px; height=300px;"/>
    Image 1
    <img src = "3.png" style = "width=200px; height=350 px;"/>
    Image 2
</body>
</html>
```

The output of above code is shown in figure 3.2.



**Figure 3.2:** Output of image tags in HTML

### Activity 1

Create a web page that provides some basic information about your city. Add some images of your city to make the page more attractive and informative.

## 3.3. HTML Links

Until this point, we have discussed documents as standalone entities, concentrating on the language elements for structuring and formatting documents. The real power of HTML and web technology rely on its ability to form a full library of information by linking collection of documents together around the world.



HTML documents can be linked together using hyperlinks. HTML provides a number of tags for creating links.

There are a number of tags that can be used to create link, as shown in the following table:

**Table 3.3:** Link tag attributes

Tags	Description
<code>&lt;a&gt;&lt;/a&gt;</code>	A anchor tag
<code>&lt;link&gt;</code>	A reference to an external resource or document, usually used to refer to a style sheet
<code>&lt;nav&gt;&lt;/nav&gt;</code>	Navigation links or a menu (HTML5 Specific)

Below we discuss how these tags can be used to link documents together.



### 3.3.1. Hyperlink (</a>)

The  tag is used to create a link to other document or to a section of the same document. The  tag requires its corresponding ending tag . The  tag is most commonly used with its HREF attribute to create a hypertext link or hyperlink to another place in the same document or to another document. In these cases, the current document is the source of the link and value of the HREF attribute, a URL, is the target. Making a mouse click on the link makes the browser to load the target document or jump to a named section of the same document.



</a> tag is used to create a link of a web page with another one.

For example, if we want to link a web page to another page (e.g. document2.html), the  tag is written as follows:

```
<a href="document2.html">Click here</a>
```

Clicking on the text written between  and  will make the browser load document2.html.

The following HTML code links our page to "document2.html" using  tag.

#### Code 3.3

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
    <p>Learning HTML is a fun but it need some practice. You have
       to put some efforts to become a master of HTML. To learn
       more about HTML,
        <a href="document2.html">Click here</a>
    </p>
</body>
</html>
```

The output of the above code is shown in figure 3.3. Notice that the browser, generally, displays the text which comes between  and  underlined or in a different color. Selecting the link by making a mouse click on "Click here" will make the browser to open document2.html.

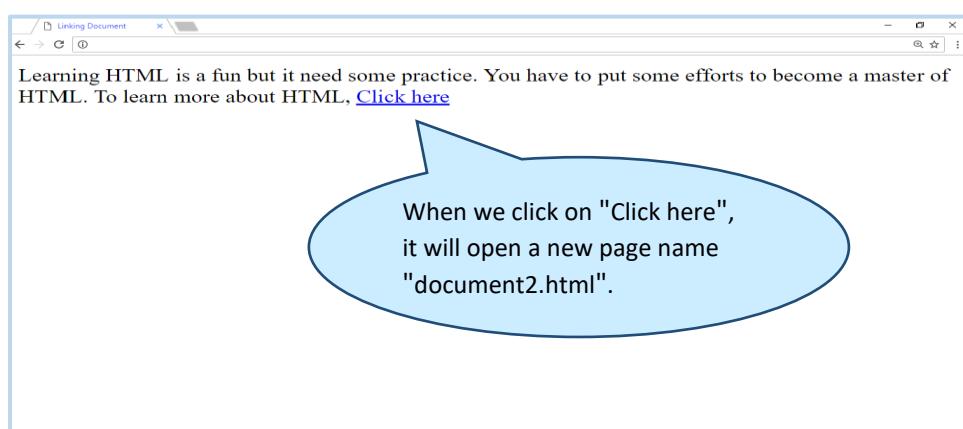


Figure 3.3: Output of hyperlink

## Activity 2

Create a web page that provides some basic information about you. Create a link on this page to open the page that you have already created in activity 1.

### 3.3.2 The Name Attribute

As we mentioned above that `<a>` tag can be used to create a link to another document or to a section of the same page. To create a link to a section of the same page, first we use the NAME attribute to place a fragment identifier within the HTML document. Once created, the fragment identifier becomes a potential target of a link. An easy way to think of a fragment identifier is analogous to the goto statement common in many programming languages.

For example, a part of an HTML document can be given a name using the `<a>` tag as shown below:

```
<a name="section1">Section One</a>
```

The value of the NAME attribute is any character string, enclosed in quotation marks. To jump a name section of a document, we can use `<a>` tag as given below

```
<a href="#section1">Click here </a>
```

When user clicks on the text “Click here”, he jumps directly to the section named as “section 1”.

Note that the section name is preceded by # symbol.



We can name part of a web page using `<a name="section name"></a>`

Use of NAME attribute is very useful when a web page is quite long and we want to allow the user to jump to various parts of the same document.



`href` tag is used to link to a section of the same page, the section name starts with #.

Have a look at the following HTML code and see how we first name a section of the document and then created a link to it using href.

#### `</> Code 3.4`

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
<a name="section1">Title Attribute</a>
<p>The TITLE attribute let us to specify a title for the document to which we are linking. The value of the attribute is any string, encoded in quotation marks.
```



A title attribute is especially useful for referencing an otherwise unlabeled resource, such as an image or a non-HTML document. For example, the browser might include the following title on otherwise title less image. </p>

```
<h3>The Name Attribute </h3>
<p> The name attribute is used to give name to that section so we can use then when user want to go to the specific section of the document. </p>
<a href="#section1">Click Here to go to Title attribute</a>
</body>
</html>
```

### Activity 3

Create a multi-section long web page that allows the users to jump to various part of the same document.

#### 3.3.3. Images as Hyperlinks

Images can also be used as hyperlinks to refer to other documents or locations within the same document just like plain text. The following HTML code is used to display an image as a hyperlink. When a user clicks on the image, document1.html will open.

```
<a href="document1.html"></a>
```



#### Video Lecture

<https://youtu.be/9BqlcMvRDGs>



### Activity 4

Create a web page, insert some of your favorite images, convert these images as hyperlinks.

#### 3.3.4. Link Tag

The link tag is used to link an HTML document with an external CSS or Javascript code. The <link> tag must be written in the <head> section of an HTML document. We will discuss the <link> tag in detail in Unit 5.



Link tag is used to link an HTML document with external CSS and JavaScript code.

### 3.3.5. <nav></nav>

The <nav> tag is an HTML5 specific tag that is not supported by older versions of HTML, therefore old browsers cannot recognize it. <nav> stands for navigation, which defines a menu of different links that point to other (internal or external) web or documents. Each menu item is a hyperlink (<a>) that can be clicked to navigate to other resources. The <nav> tag must be written within the <body> of the HTML document.

The following example illustrates how to use <nav> tag.

#### </> Code 3.5

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
    <p>Click on a link below to go to your preferred web site </p>
    <nav>
        <a href="www.google.com">Google</a>
        <a href="www.hotmail.com">Hotmail</a>
        <a href="www.yahoo.com">Yahoo</a>
        <a href="www.outlook.com">Outlook</a>
    </nav>
</body>
</html>
```

Output of above code is shown in figure 3.4.

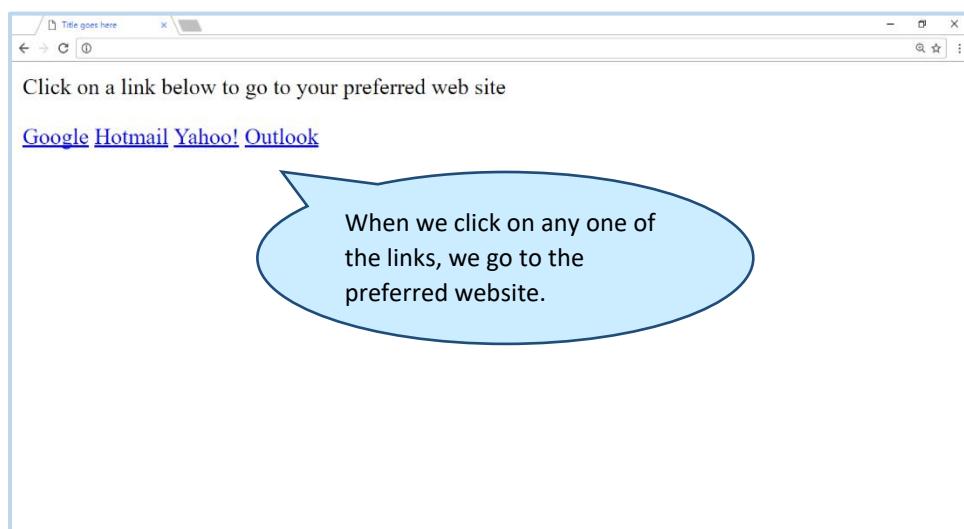


Figure 3.4: Output of code 3.5.



## 3.4. Tables

Tables are used to organize data/information in rows and columns. Table entries can contain images, paragraph, lists, and even other tables. Furthermore, the border around the table and between cells can be suppressed so that the reader may be unaware that a table is being used. Basic HTML tags used in creating tables are summarized below followed by a brief explanation for each of them.

**Table 3.4:** Table attributes

Tags	Description
<code>&lt;table&gt;&lt;/table&gt;</code>	Table tag defines an HTML table
<code>&lt;th&gt;&lt;/th&gt;</code>	<code>&lt;th&gt;</code> defines column headings
<code>&lt;tr&gt;&lt;/tr&gt;</code>	<code>&lt;tr&gt;</code> element defines row of a table
<code>&lt;td&gt;&lt;/td&gt;</code>	<code>&lt;td&gt;</code> defines data element of a column



Tables are a useful way of organizing data. Column heading defined with `<th>` are automatically displayed as bold.

Various tags used to create tables in HTML are discussed below.

### `<table></table>`

An HTML table can be created using the `<table>` tag. This tag must be located within the `<body>` section of an HTML document. By default, table does not have border. To give table border, we use `border` attribute of `<table>` tag as shown below.

```
<table border="1">
```



We use `<table>` tag to create a table.

### `<tr></tr>`

The `<tr>` tag with its ending `</tr>` tag is used to define a row of a table. A table row can either a column heading or row data. The `<th>` tag with its ending `</th>` tag is used to assign headers to the table columns. Data can be defined using the tags `<td>` and `</td>`.



`<tr>` tag is used to create a table row. A row can be column headings or data.

Assume we want to create the following table:

**Table 3.5:** Sample table

Name	Physics	Chemistry	Maths
Ali	50	70	65
Moen	77	82	45



`<th>` tag is used to define column headings and `<td>` tag is used to define data in each column.

Note that our table has three rows (including row with column headings). To create the column heading, we need to write the following code:

```
<tr>
  <th>Physics</th>
  <th>Chemistry</th>
  <th>Maths</th>
</tr>
```



To leave a column heading blank, use `<th></th>`.

Note that the column heading is specified as a row of the table. Each column heading is written using `<th></th>` tag pair. If a column does not have a heading, we still have to write `<th></th>` for that column. To create the second row, we need to write the following code:

```
<tr>
  <td></td>
  <td>50</td>
  <td>70</td>
  <td>65</td>
</tr>
```



To leave a data column blank, use `<td></td>` tag.

Note that data for each column is written between `<td>` and `</td>` tags.

Below is the complete code to create the above table. See how each row and data in each column has been given using `<tr>` and `<td>` tags.

#### `</> Code 3.6`

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
  <table>
    <tr>
      <th>Name</th>
      <th>Physics</th>
      <th>Chemistry</th>
      <th>Maths</th>
    </tr>
    <tr>
      <td>Ali</td>
      <td>50</td>
```



```
<td>70</td>
<td>65</td>
</tr>
<tr>
<td>Moen</td>
<td>77</td>
<td>82</td>
<td>45</td>
</tr>
</table>
</body>
</html>
```

The output of above code is shown in figure 3.5.

Name	Physics	Chemistry	Maths
Ali	50	70	65
Moen	77	82	45

Figure 3.5: Output of code 3.6

You might have noticed that the table, by default, does not have borders. A programmer has to define borders and lines in the table. To do that, we use BORDER attribute of `<table>` tag to specify the width (in pixels) of the border around the table. For example, `<table border="1">` gives a table border of WIDTH=1.

## Activity 5

In the above HTML code, change the `<table>` to `<table border="1">` and see the output. Change the border width by changing the number to 2, 3 and 4 and observe the difference.

### 3.4.1. Adding Table Caption

The `<caption>` tag can be used to add caption of a table. Caption, by default, appears on the top of the table. If we want to put the caption at the bottom of the table, we use ALIGN attribute as:

```
<caption align="bottom">Marks obtained by students</caption>.
```

The `<caption>` tag must be the first tag just inside the `<table>` tag.

## Activity 6

Add a caption in your table that you created in the previous activity.

### 3.4.2. The Rowspan and Colspan Attributes

By adding rowspan and colspan attributes to the table elements, it is possible to create data cells that span a given number of rows or columns. The basic idea of the rowspan and colspan attributes for `<td>` and `<tr>` tags is used to extend the size of the cells across two or more rows or columns, respectively. For example, to set a cell to span two rows, use `<td rowspan="2">` and to set a heading to span two columns, use `<th colspan="2">`



We can merge rows and columns of table using rowspan and colspan attributes of `<td>` and `<th>` tags.

Assume we want to create the following table

**Table 3.6:** Sample table

Name	Subjects		
	Physics	Chemistry	Maths
Ali	50	70	65
Moen	77	82	45

Note the following points in order to create the above table

- Row 1 and row 2 of the table are merged to create a single row column with column heading name.
- Column 2, 3 and 4 have been merge to a single column with heading “Subject”.

Let us create the first row of our table:

```
<tr>
<th rowspan = "2">Name</th>
<th colspan = "3">Subjects</th>
<tr>
```

Now we create the second row of the table that has the subject headings. This can be done as follows.

```
<tr>
<th>Physics </th>
<th>Chemistry </th>
<th>Math </th>
<tr>
```

Note that we do not specify the heading for column 1 (name) of row 2 since it is already merged with row 1. Complete HTML code for creating table 3.6 is given below:

**</> Code 3.7**

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
</head>
<body>
    <table border = "1">
        <tr>
            <th rowspan = "2">Name</th>
            <th colspan = "3">Subjects</th>
        </tr>
        <tr>
            <td>Physics</td>
            <td>Chemistry</td>
            <td>Maths</td>
        </tr>
        <tr>
            <td>Ali</td>
            <td>50</td>
            <td>70</td>
            <td>65</td>
        </tr>
        <tr>
            <td>Moen</td>
            <td>77</td>
            <td>82</td>
            <td>45</td>
        </tr>
    </table>
</body>
</html>
```

The output of above code is shown in figure 3.6.

Name	Subjects		
	Physics	Chemistry	Maths
Ali	50	70	65
Moen	77	82	45

We give border to the table by specifying the property border = "1".

**Figure 3.6:** Output of tables with border



## Video Lecture

<https://youtu.be/MEioWGewTkI>



### Activity 7

Write HTML code to create the following table.

First semester		Second Semester	
Subject	Marks	Subject	Marks
Physics	70	Java	85
Chemistry	80	HTML	92

### 3.4.3. Changing Alignment

From the output of our previous example, it can be seen that the column headings and data are, by default, displayed left-justified. We can change the alignment of column headings and data using the align attribute of `<th>` and `<td>`. Here is an example to show you how it is done

```
<th style = "align: center"> Physics</th>
<td style = "align: center"> Ali</td>
```



The alignment of column headings and data in a column can be changed by using align attribute.

### Activity 8

Explore other attributes that can be used with `<table>`, `<tr>`, `<th>` and `<td>` tags.



## Unit Summary

In this unit, we learned how to embed graphics with suitable size and location in a web page. We also learned how to create hyperlinks to link web pages together. We also discussed how to use a set of HTML tags to arrange data in tabular form.



## Self Assessment Questions

Select the correct answer.

1. The \_\_\_\_\_ tag defines an image in an HTML page.
  - A. <img>
  - B. <image>
  - C. <imge>
  - D. <pic>
  
2. <img> tag has no end tag.
  - A. True
  - B. False
  
3. If the image cannot be displayed in the browser, \_\_\_\_\_ specifies an alternate text for the image.
  - A. value attribute
  - B. caption attribute
  - C. alt attribute
  - D. text attribute
  
4. What is the correct HTML code for creating a hyperlink?
  - A. <a href="http://mcqsets.com">ICT Trends Quiz</a>
  - B. <a name="http://mcqsets.com">ICT Trends Quiz</a>
  - C. <http://mcqsets.com>ICT Trends Quiz
  - D. url="http://mcqsets.com">ICT Trends Quiz
  
5. The <img> tag automatically inserts a carriage return after displaying its graphics.
  - A. True
  - B. False
  
6. Which of the following tag is used to link the URL.
  - A. <a>
  - B. <hyperlink>
  - C. <style>
  - D. <link>
  
7. Default style and color of unvisited link in a browser is \_\_\_\_\_.
  - A. Underline and Pink
  - B. Underline and Magenta
  - C. Underlined and Blue
  - D. Normal and Blue
  
8. You can set image width and height based on your requirement using width and height attributes.
  - A. True
  - B. False



9. Which of the following is correct to jump to another section of a web page?
  - A. <a href="#section2">Click here
  - B. <a href="#section2">Click here </a>
  - C. <a href="section2">Click here </a>
  - D. <a href="section2#">Click here </a>
10. An image can be used as a link instead of text.
  - A. True
  - B. False
11. Which tag is used to link a style sheet to your web page?
  - A. <link>
  - B. <a>
  - C. <nav>
  - D. None of the above
12. Which tag is used to create a row of a table?
  - A. <tr>
  - B. <row>
  - C. <td>
  - D. <table>
13. Which tag is used to create a data element in column?
  - A. <tr>
  - B. <row>
  - C. <td>
  - D. <table>
14. Which of the following tag is used to give heading to a column?
  - A. <th>Physics</th>
  - B. <td>Physics</td>
  - C. <col>Physics</col>
  - D. <head>Physics</head>
15. Which of the following correctly merges two columns of a table?
  - A. <th rowspan=2> Subjects </th>
  - B. <th colspan=2>Subjects</th>
  - C. <mrg col=2>subjects</mrg>
  - D. <table col=2>

**Answer Key:**

1. A	2. A	3. C	4. A	5. B	6. A	7. C	8. A	9. B	10. A
11. A	12. A	13. C	14. A	15. B					



## Review Questions

**Write short answers of the following.**

1. What are the two most common graphic formats used on the web?
2. What do you understand by loosy graphic format?
3. What is the purpose of ALT attribute?
4. Is it possible to ALIGN all of the data cells in a particular row using the <tr> tag?
5. What is difference between <a>, <link> and <nav> tags?
6. How do you specify column headings of a table?
7. How can data be specified for a single row of a table?
8. How do we merge two columns of a table together?
9. How do we merge two rows of a table together?

---

## </> Coding Exercise

1. Create a website that provides basic information about a number of historical places in your country. You should prepare a separate web page for each historical site and a home that has the links to other pages. Add images to your pages to make them more attractive.
  2. Create a website that gives basic information about yourself. You should prepare a separate page for each of the following: personal information, academic qualification, professional qualification, job experience, hobbies, and photo gallery. Academic qualification page should display the information about your degrees and certificates along with the subjects you studied in each in a tabular form.
  3. Prepare multiple web pages with suitable links that provide basic information about a number of medicines. Add suitable images.
  4. Prepare a web page that displays marks obtained by a number of students in 5 different courses. You should also display student name and his photo alongside the marks obtained. Use tables.
-



## References and Further Reading

1. MacDonald, M. (2013). HTML5: The missing manual. O'Reilly Media, Inc.
2. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
3. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
4. Stevens, L., & Owen, R. J. (2013). The truth about HTML5. Apress.
5. Hogan, B. P., Warren, C., Weber, M., Johnson, C., & Godin, A. (2012). Web Development Recipes. Pragmatic Bookshelf.
6. Creating Hyperlinks. Available on:  
[http://www.sanych99.narod.ru/New\\_Site/arch/eng/html/001/ch07\\_03.htm](http://www.sanych99.narod.ru/New_Site/arch/eng/html/001/ch07_03.htm)
7. HTML Questions. Available on: <http://engineeringinterviewquestions.com/>
8. Images in HTML. Available on: <http://www.techieweb.in/images-in-html/>



## UNIT4

# HTML Marquee, Form and Input Types

## Introduction

We have so far learnt some important HTML tags to develop a good website. We have learnt how a document can be structured in sections and paragraphs, how we change physical appearance of our document and its elements, how we insert multimedia contents and table and links web pages together. HTML provides a few more tags to the web developers. One such tag is marquee tag which causes text to scroll up, down, left or right automatically which is, sometime, very useful. Another important tag is iframe or inline frame which is used to display external objects including other web pages within a web page.

Last but not the least is a form in an HTML document. HTML Form is one of the main points of interaction between a user and a website or application. The user uses it to send data to the website. An HTML Form consists of one or more widgets. These widgets can be text fields (single line or multiline), select boxes, buttons, checkboxes, or radio buttons. Mostly, these widgets are paired with a label that describes their purpose.



## Unit Outcomes

Upon completion of this unit, you should be able to:

1. Embed a document in your web page using marquee tag.
2. Create a web page with multiple iframes.
3. Create form for getting user data.



## Terminologies

<b>Marquee:</b>	A scrolling piece of text on a web page.
<b>Form:</b>	Allows a user to enter data on a web page.
<b>Textbox:</b>	It is a rectangular area for entering text in a web page.
<b>Dropdown list:</b>	It is a list of items for selecting a value from pull down menu.
<b>Radio button:</b>	It is used to select one option from set of many options
<b>Check box:</b>	It is used to select more than one options.
<b>Iframes:</b>	HTML document that is embedded inside another HTML document.



## 4.1. Marquee

A marquee is a scrolling piece of text displayed either horizontally across or vertically down a web page depending on the settings. It is created by using HTML <marquees> tag. The <marquee> tag may not be supported by all browsers so it is not recommended to rely on this tag, instead we can use javascript and CSS to create such effects.

A simple syntax of<marquee> tag is as follows:

```
<marquee attribute name = "attribute value"></marquee>
```



HTML Marquee is a scrolling piece of text either displayed horizontally or vertically in a web page.

The attributes that can be specified with the marquee tag are given in the following table.

**Table 4.1:** Marquee tag attributes

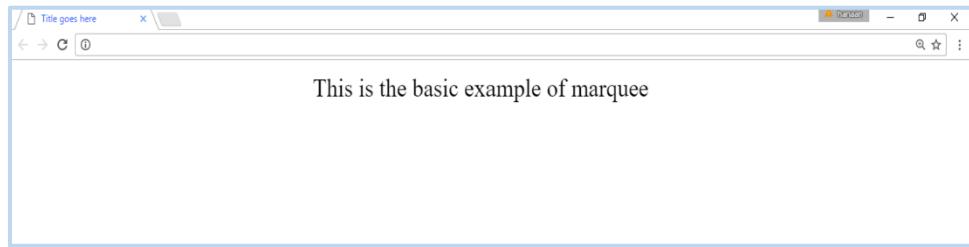
Attributes	Description
<b>width</b>	It specifies the width of the marquee. This can be a value like 10 or 20% etc. of browser window.
<b>height</b>	It specifies the height of the marquee. This can be a value like 10 or 20% etc. of browser window.
<b>direction</b>	It specifies the direction in which marquee should scroll. This can be a value like <i>up</i> , <i>down</i> , <i>left</i> or <i>right</i> .
<b>behavior</b>	It specifies the type of scrolling of the marquee. This can have a value like <i>scroll</i> , <i>slide</i> and <i>alternate</i> .
<b>scroll delay</b>	It specifies how long to delay between each jump. This will have a value like 10 etc.

Try the following code and see its effect for yourself.

### </> Code 4.1

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
    <marquee>This is the basic example of marquee </marquee>
</body>
</html>
```

The output of above code is shown in figure 4.1.



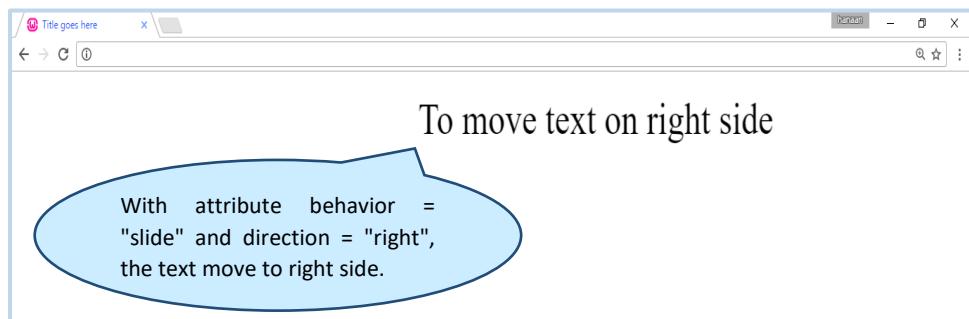
**Figure 4.1:** Output of code 4.1

By default, when we use marquee tag, the text written between the tag moves from left to right. To change the direction or behavior of text we use “direction” and “behavior” attributes. Below example will clarify how to change the direction and behavior of text which are written between marquee tags.

**</> Code 4.2**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
    <marquee behavior = "slide" direction = "right">To move text
        on right side </marquee>
</body>
</html>
```

The above code will move the text on right side while the behavior of text is slide. The following output (figure 4.2) will clearly show you that the text is moving on right side.



**Figure 4.2:** Output of code 4.2

Here is another example for you to try.

**</> Code 4.3**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
    <marquee behavior = "slide" direction = "left" width = "50%">
        To move text on left side </marquee>
</body>
</html>
```



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

The output of above code is shown in figure 4.3.

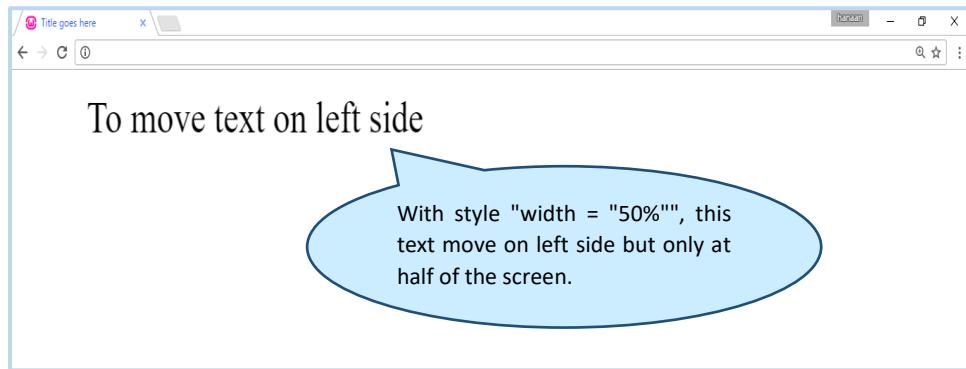


Figure 4.3: Output of code 4.3

## 4.2. Forms

A form is an interface between a user and a browser through which the user can enter the required information. One of the reasons for having the CGI (Common Gateway Interface) is that two-way communication is possible with HTML. The client can enter information by selecting buttons, pulling down or scrolling through menus, and entering text into text boxes. The interactive building blocks can be brought together and defined as a form. You can see forms when you create a new email, or send a comment in a forum on a website.

The following table shows the core components of an HTML form:

Table 4.2: Core components of an HTML form

Tags	Description
<code>&lt;form&gt;&lt;/form&gt;</code>	HTML tag for creation of HTML form
<code>&lt;input&gt;</code>	An input control, a single-line text field or a button in most cases
<code>&lt;textarea&gt;&lt;/textarea&gt;</code>	A multi-line text area
<code>&lt;select&gt;&lt;/select&gt;</code>	A drop-down list
<code>&lt;option&gt;&lt;/option&gt;</code>	A single option in a drop-down list
<code>&lt;datalist&gt;&lt;/datalist&gt;</code>	Drop down list (HTML5 specific)

Let us discuss various attributes and tags related to forms in more detail.

### 4.2.1. `<form></form>`

A form can be placed anywhere in the body of HTML document with its elements enclosed by the `<form>` tag and its ending `</form>` tag. The `<form>` tag has two important attributes: **action** and **method**. The **action** attribute points to the server side script (the ‘back end’) that handles the form submission. Usually, this will be a script (PHP, ASP, and Perl) or a CGI program. The **method** attribute has a value either GET or POST. When sending data, the GET method adds the data to the URL with

maximum URL length of 2048 characters. This method should not be used when sensitive data, such as user information and passwords are to be sent, because all the information will be available in clear text form in the URL. The **POST** submits data from an HTML form to an identified resource and the data is included in the body of the request. Information sent with post method will not be visible and therefore it is used to send sensitive information.



A form tag has two important attributes: action and method

Suppose our script to handle the form data is written in php named as formmail.php. The HTML form tag will be written as given below

```
<form action="cgi-bin/formmail.php" method="post">
    Your input items here.....
</form>
```

#### 4.2.2. <input>

We can have different types of input elements in a form such as check boxes, radio buttons, simple text boxes etc. The <input> elements are used within a <form> element to declare input controls that allow users to input data. The table given below describes different input types available in HTML.

**Table 4.3:** Input types in HTML

Attribute	Description
<input type = "text">	This input control is used to enter text.
<input type = "password">	This input control is used to enter password.
<input type = "submit">	This input control is used to submit data.
<input type = "reset">	This input control is used to reset form.
<input type = "file">	This input control is used to upload a file.
<input type = "radio">	This input control is used to select 1 option from many.
<input type = "checkbox">	This input control is used to select 1 or more than 1 options.
<input type = "button">	This input control is used to make button
<input type = "date">	This input control is used to enter date.
<input type = "email">	This input control is used to enter email.

Let us discuss different input type in a little more detail.

##### <input type = "text">

This element is used for only one line of user input items such as names. The code to create a simple text box is:

```
<input type = "text" name="FirstName" />
```

Where type="text" tells the browser that a single line of text input box should be created and name="First Name" gives a name to the input field. The field name is used to identify the field on the server and client side.



<input type = "text"> is used to create a single line text box in the form in which user can enter a line of text.

Below is a code that creates a form with two text boxes.

**</> Code 4.4**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<form>
First name:
<input type="text" name="firstname"><br>
Last name:
<input type="text" name="lastname"><br>
</form>
</body>
</html>
```

Output of above code is shown in figure 4.4.

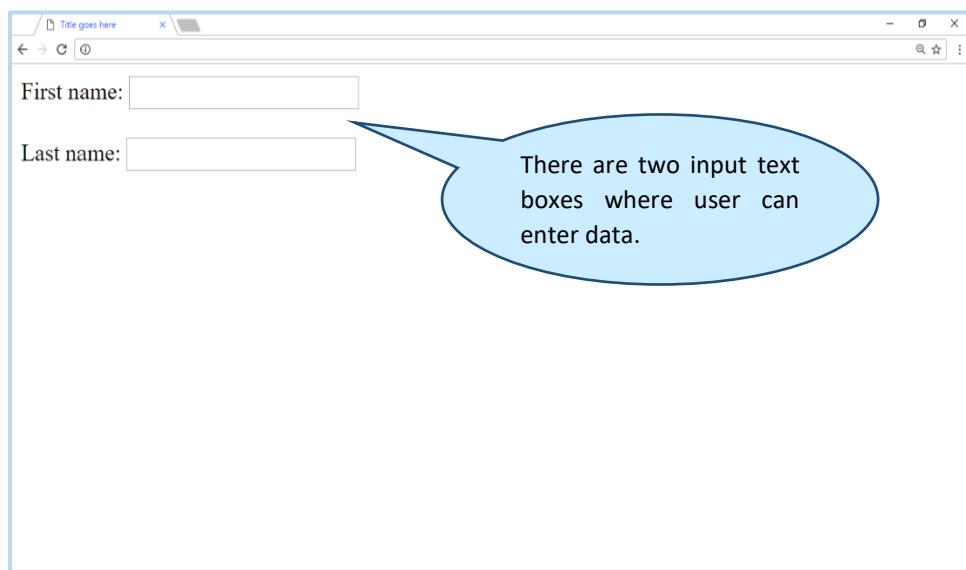


Figure 4.4: Output of code 4.4

**Activity 1**

Create a form that has the following two text boxes: Movie Title, Name of Hero.

There are some other attributes that can be used with the text boxes as discussed below:

A text element can be given a default value which will be displayed inside the text box when the form loads. A default value is specified by using *value* attribute as shown below

```
<input type="text" name="First Name" value="Your first name here please"/>
```

We can also specify maximum number of characters a user can enter in a text box by using *maxlength* attribute as:

```
<input type="text" name="First Name" value=" " size="25" maxlength="50" />
```

#### **<input type = "password">**

This is also a single-line text input that is used to mask the character as soon as a user enters it. For example, if we want the user to enter his password which should not be displayed in the browser, we will use the following code:

User password: <input type = "password" name = "pwd">



<input type = "password"> is a single line text box which hides the characters when entered by a user.

#### **<input type = "submit">**

This input type creates a button on a form. When the user clicks on the “submit” button, the form data is sent to the server. The code below shows how we can use this with a form.

#### **</> Code 4.5**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
  <form>
    First name:
    <input type="text" name="firstname"><br>
    <br> Last name:
    <input type="text" name="lastname">
    <br><input type="submit" value="Submit" >
  </form>
</body>
</html>
```

The output of the code is shown in figure 4.5.



The screenshot shows a simple HTML form within a browser window. The form contains two text input fields labeled "First name:" and "Last name:", each with a corresponding empty input box. Below the input fields is a grey "Submit" button. A blue callout bubble originates from the "Submit" button and contains the text: "When a user clicks on \"submit\" button, the data entered in text boxes are submitted to the server."

**Figure 4.5:** Output of code 4.5

#### <input type = "reset">

When a user clicks on reset button, all the data in the form are reset to the default values. The syntax below shows how we can use reset button.

```
<input type="reset">
```



When you click on reset button on form, it will reset to the default value.

## Activity 2

Add Reset and Submit buttons in the form that you created in activity 1.

#### <input type = "file">

If we want to allow a user to upload a file to a web server, we need to use "file" as input type. <input type = "file" > displays the browse button that can be used to select the file to be uploaded.

The following code shows how to upload a file.

```
<form>
    <input type="file" name = "upload" value="upload">
</form>
```

#### <input type = "radio">

This input type is used to create radio buttons a form. Radio buttons are used when out of many options, just one option is required to be selected. To create a radio button, input type should be specified as "radio" as shown in the following code

```
<input type="radio" name="value" value ="Value">
```

For example, if we want to create radio buttons that allows a user to select his/her gender, we have to specify each value separately with `<input>` tag and the name attribute must have the same value for each radio button as the following code shows.

```
<input type="radio" name = "gender" value = "male" checked>Male  
<input type="radio" name = "gender" value = "female">Female<br>
```

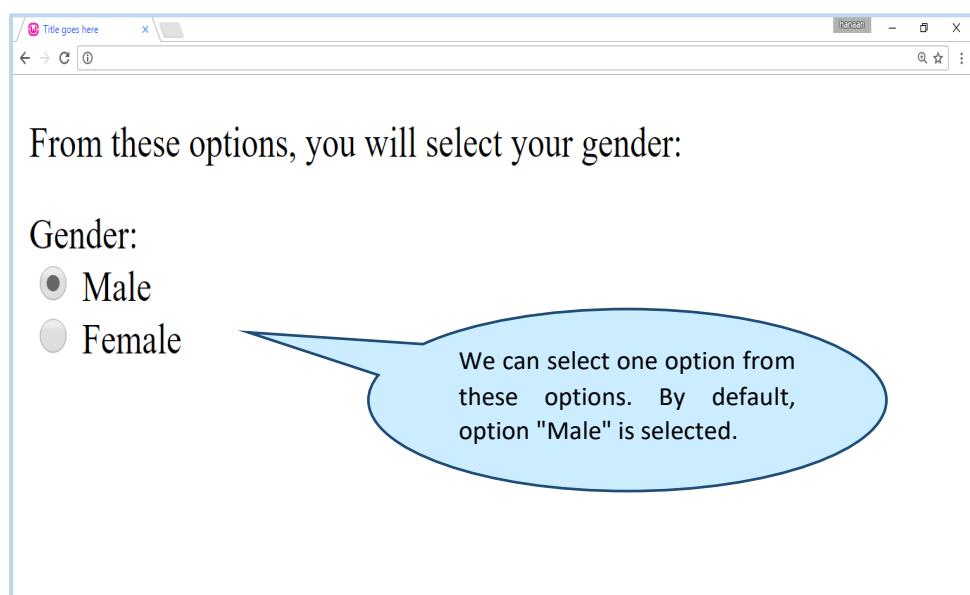
The text written after the input tag (i.e. after `>`) will be displayed after the radio button.

The following code creates a list of radio buttons.

**Code 4.6**

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Title goes here</title>  
</head>  
<body>  
<form>  
    <p>From these options, you will select your gender: </p>  
    Gender:<br>  
    <input type="radio" name="gender" value="male" checked>  
    Male<br>  
    <input type="radio" name="gender" value="female"> Female  
    <br>  
  
</form>  
</body>  
</html>
```

Output of above code is given below



**Figure 4.6:** Output of code 4.6



## Video Lecture

<https://youtu.be/aQM3PTV8r7g>



### Activity 3

Add a new field "Movie Type" in the form you create in activity 2. You should create the following three radio buttons from which a user can select an option: Comedy, Horror, Action. This new field should be added just before Reset and Submit buttons on your form.

**<input type = "checkbox">**

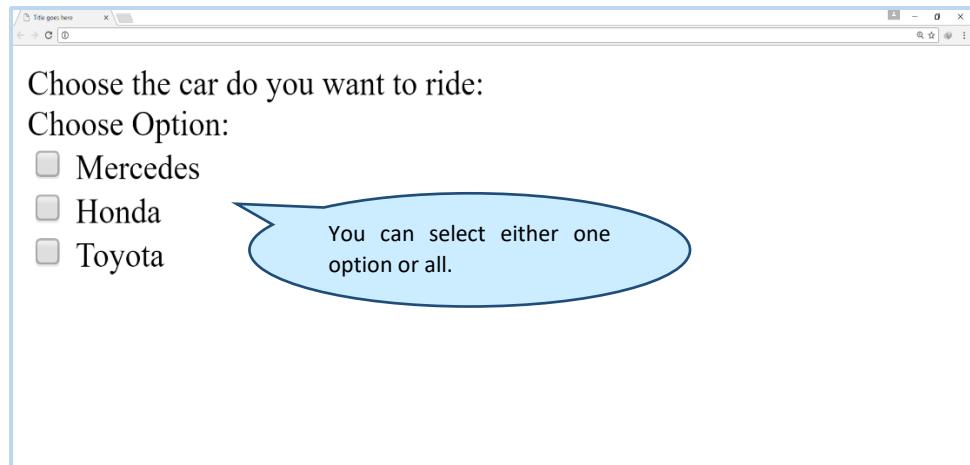
Checkboxes are used when more than one options may be selected by a user. They are also created using HTML <input> tag but type attribute is set to checkbox using the following code:

`<input type="checkbox" name="Fieldname" value="checkbox Value">`

**</> Code 4.7**

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<form>
Choose the car do you want to ride:<br>
Choose Option:
<input type="checkbox" name="vehicle1" value="Mercedes">
Mercedes<br>
<input type="checkbox" name="vehicle2" value="Honda"> Honda
<input type="checkbox" name="vehicle3" value="Toyota">
Toyota
</form>
</body>
</html>
```

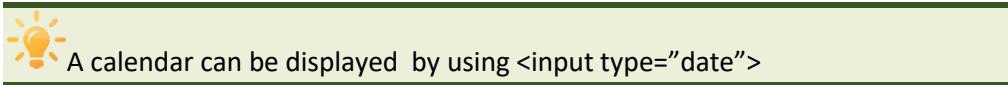
Output of above code is shown in figure 4.7



**Figure 4.7:** Output of code 4.7

#### <input type = "checkbox">

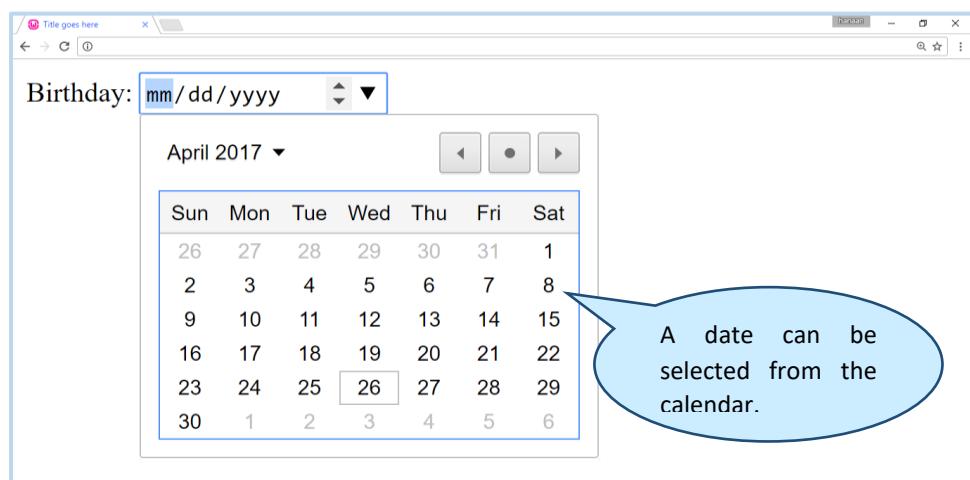
An input type Date allows a user to select a date from a calendar. The following code shows how to use date input type.



#### </> Code 4.8

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<form>
    Birthday:
    <input type="date" name="bday">
</form>
</body>
</html>
```

Output of above code is shown in figure 4.8.



**Figure 4.8:** Output of code 4.8



## Activity 4

Add a new field “Release Date” in your form that you created in activity 3. This new field should be created after the “Movie Type” field on the form. The user should be able to select a date from the calendar.

**<input type="email">**

Input type email is used to get email as input from a user. The code below shows how to get email as input from a user.

```
<form>
  <input type="email" name = "email" >
</form>
```

## Activity 5

Add a field email address in your form that you created in Activity 4 to read email address of a user.

### 4.2.3. <textarea> </textarea>

The <textarea> tag can be used to create a multi-line text field where a user can write a long text, such as a comment or a message. This tag has two important attributes:

- row:** it defines the number of lines the <textarea> contains.
- col:** it defines the number of characters per line.

The following example shows a text area with 4 lines (rows) and 50 characters per line:

#### </> Code 4.9

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
  <form action="mypage.html" method="post">
    First name:<input type="text" name="fname"><br>
    Last name:<input type="text" name="lname"><br>
    <textarea rows="4" cols="50">
      please write your comment!
    </textarea><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

The output of above code is shown in figure 4.9.

The screenshot shows a simple HTML form within a browser window. The form contains two text input fields labeled "First Name:" and "Last Name:", both with empty white boxes. Below them is a larger text area with the placeholder text "please write your comment!". At the bottom left is a "Submit" button. A blue callout bubble points from the right side of the text area towards the text "In textarea, user can enter multiple lines of text in textarea.".

**Figure 4.9:** Output of code 4.9

## Activity 6

Add a multi-line text area “Feedback” in the form that you created in activity 5.

### 4.2.4. <select> </select>

A select box, also called a drop down box, provides options as a dropdown list, from where a user can select an option. The `<select>` tag is used to create a dropdown list. Each value in the list is specified using the `<option>` tag inside `<select>` tag. For example, the following code creates a select box with two values to select from.

```
<select name="foods">
    <option value="pizza">Pizza</option>
    <option value="burger">Burger</option>
</select>
```

The name attribute in `<select>` tag gives a name to the drop-down list and value attribute of `<option>` tag gives a value to each option. The options that appear on the drop-down list are specified between `<option>` and `</option>`.



Select box also called dropdown list are used to select one option from dropdown list.

Here is a code that creates a form with different input types.

#### </> Code 4.10

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
```



```
<form action="mypage.html" method="post">
    Firstname: <input type="text" name="fname"><br>
    <br>Last name: <input type="text" name="lname"><br>
    <textarea rows="4" cols="50">
        please write your comment!
    </textarea><br>
    <br><select>
        <option value="item1">item1</option>
        <option value="item2">item2</option>
        <option value="item3">item3</option>
        <option value="item4">item4</option>
    </select>
    <br><br><input type="submit" value="Submit">
</form>
</body>
</html>
```

The output of above code is shown in figure 4.10.

The screenshot shows a web browser window with the following elements:

- A title bar with a placeholder "Title goes here".
- An address bar with a back arrow, forward arrow, and search icon.
- A main content area:
  - "First Name:" followed by an empty input field.
  - "Last Name:" followed by an empty input field.
  - A text area with the placeholder "please write your comment!".
  - A dropdown menu with five options: "item1", "item2", "item3", "item4", and "item1" (highlighted with a blue border).
  - A blue callout bubble pointing to the dropdown menu with the text: "You can select one option from list of options."
  - A "Submit" button at the bottom.

Figure 4.10: Output of code 4.10

#### 4.2.5. <datalist></datalist>

This tag is similar to <select> tag. However, there are two major differences between <datalist> and <select>. The first one is that <datalist> tag works only with HTML 5. The second difference is that the <datalist> tag has a small textbox where the user can type a text to search for an item in the list. This feature is very important when we have a large list of options to select from.

The following example shows how to create a list of browsers using <datalist> tag.

#### </> Code 4.11

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
```

```

<form action="mypage.html" method="post">
<p>Choose your favorite browser from the following browsers
</p>
<input list="browsers">
<datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
</datalist>
<input type="submit" value="Submit">
</form>
</body>
</html>

```

The Output of above code is shown in figure 4.11.

Choose your favorite browser from the following browsers

▼  
 Internet Explorer  
 Chrome  
 Opera  
 Safari

Submit

You can select one of the browser from the dropdown menu.

**Figure 4.11:** Output of code 4.11

## Activity 7

Create a datalist “Your favorite Movie type” in your form that you created in activity 6. The datalist should have the following options to choose from: Horror, Action, Comedy, Drama, and Musical.

### 4.2.6. A Complete Example of Input Types and Form

Once we know different elements of a form, we can create forms with different input types. The code given below demonstrate the use of different input types in HTML form.

#### </> Code 4.12

```

<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<form>
    First Name:
    <input type="text" name="firstname"><br>
    <br>Last Name:

```



```
<input type="text" name="lastname"><br>
<br>Date of Birth:<br>
<input type="date" name="bday"><br>
<br>Gender :<br>
<input type="radio" name="gender" value="male" checked> Male
<input type="radio" name="gender" value="female"> Female<br>
<br>Email:<br>
<br><br><input type="email" name="email">Qualification
<input type="checkbox" name="matric" value="matric">Matric
<br>
<input type="checkbox" name="fsc" value="intermediate">F.S.C
<br>
<input type="checkbox" name="bsc" value="bachelors">Bsc
<br><br><input type="submit" value="Submit">
<input type="reset">
</form>
</body>
</html>
```

The Output of above code is shown in figure 4.12.

First Name:

Last Name:

Date of Birth:  mm/dd/yyyy

Gender :

Male

Female

Email:

Qualifications

Matric

F.Sc.

B.Sc.

Here is the complete form where user can enter their information. After pressing "Submit" button, user data can be used for further purpose.

Figure 4.12: Output of code 4.12



## Video Lecture

<https://youtu.be/RtqFjhQsfPA>



## Activity 8

Create a web page that provides some basic information about your city. Add some images of your city to make the page more attractive and informative.

### 4.3. Iframes

Frames are areas (containers) where information from another web page can be displayed within a web page. For example, to open AIOU website in an iframe, we give link of AIOU website in a href attribute. The following code shows how to give link attribute in iframe.

```
<iframe><a href = "www.aiou.edu.pk"></iframe>
```

We can set width and height of iframe by giving its size in pixels or in percentage. For example, the following code sets **width** = "100%" and **height** = "300px" to iframe and gives it a name "iframe\_a".

```
<iframe height="300px" width = "100%" name="iframe_a">
```

Now we can open a web page or a different website in the frame using href as shown in the code below

```
<iframe height="300px" width = "100%" name="iframe_a">

<a href = "http://www.aiou.edu.pk" target = "iframe_a">AIOU Website
</a>
</iframe>
```

Note that the value of the target attribute is the name of the iframe in which we want to open a different page or a website. The website of AIOU will open in the iframe when the user clicks on the text "AIOU Website".



Iframes allows user to open another document within a document.

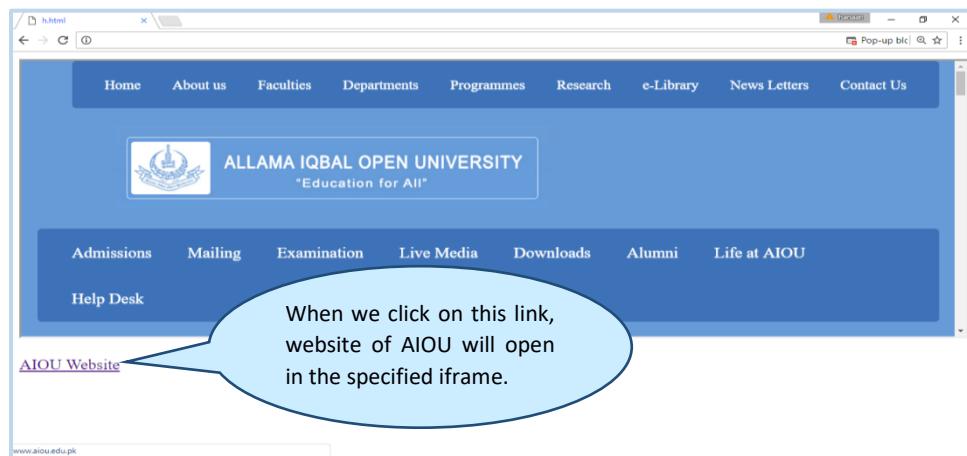
The following example shows the use of <iframe>.

#### Code 4.13

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<iframe height ="300px" width = "100%">
<p> <a href ="http://www.aiou.edu.pk" target ="iframe_a">AIOU
Website
</iframe>
<p>This is my page </p>
</body>
</html>
```



The output of above code is shown in figure 4.13.



**Figure 4.13:** Output of code 4.13

## **Activity 9**

Create two web pages and display them using iframes.



## **Unit Summary**

In this unit, we learned about how to embed a document in a web page using marquee tag. We also learned the development of a web page with multiple iframes. The last section of the unit is focused on the development of HTML forms and demonstrates, how to get user data using form elements.



## Self Assessment Questions

Select the correct answer.

1. Which of the following code is used to scroll the text from left to right?
  - A. <marquee behavior = "slide" direction = "right">
  - B. <scroll behavior = "slide" direction = "right">
  - C. <marquee behavior = "slide" direction = "left-to-right">
  - D. None of the above
2. Which of the following tag is used to specify a multi-line text area in a form?
  - A. <input type="Multi-text">
  - B. <textarea> <textarea>
  - C. <form type="textarea">
  - D. None of the above
3. A form can be placed anywhere in the head section of a HTML document.
  - A. True
  - B. False
4. Which of the following code is used to create a single-line text box in a form?
  - A. <input type="text" name="name">
  - B. <input type="text-box" name="name">
  - C. <form type="text" name="name">
  - D. <input type="box" name="name">
5. <input type = "submit"> creates a button on a form.
  - A. True
  - B. False
6. Which of the following code clears all the form elements?
  - A. <input type="clear">
  - B. <form type="reset">
  - C. <input type="reset">
  - D. <input type="reset-form">
7. Which of the following code is used to create a radio button?
  - A. <input type="radio-button" name = "gender" value = "male">
  - B. <input type="radio" name = "gender" value = "male">
  - C. <input type="button" name = "gender" value = "male" >
  - D. None of the above
8. <input type="checkbox" name="choice" value="Orange"> will create a check box in a form.
  - A. True
  - B. False
9. Which input type is used to create display a calendar?
  - A. Date
  - B. Calendar
  - C. Date-Time



- D. None of the above
10. Which of the following tag is used to specify items in a select-box?
- A. <choice>
  - B. <item>
  - C. <select>
  - D. Option

**Answer Key:**

1. A	2. B	3. B	4. A	5. A	6. C	7. B	8. A	9. A	10. C
------	------	------	------	------	------	------	------	------	-------



## Review Questions

**Write short answers for the following questions.**

1. Use marquee tag to display "Hello world" that moves to right.
2. Use marquee tag to display "Hello world" that moves from left to right and right to left continuously.
3. What is the purpose of iframes? How can you use them effectively?
4. What is difference between checkboxes and radio buttons?
5. Discuss various input types available for a form.

## </> Coding Exercise

1. Create a web page that gives some basic information about your city (add some nice photos as well). At the end of the page, create a simple form for more information form that allows the users to input their names, email addresses, and comments.
  - a. Add a checkbox on the web page asking the reader either he likes to join the group.
  - b. Add a drop down list that allows the user to select his/her country.
2. Create a new web page that uses an iframes to display the web page that you created in the above exercises.
3. Create a web page with a set of multiple choice questions. The reader should be given a list of options to select the answer for each question.



## References and Further Reading

1. MacDonald, M. (2013). HTML5: The missing manual. O'Reilly Media, Inc.
  2. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
  3. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
  4. Stevens, L., & Owen, R. J. (2013). The truth about HTML5. Apress.
  5. Hogan, B. P., Warren, C., Weber, M., Johnson, C., & Godin, A. (2012). Web Development Recipes. Pragmatic Bookshelf.
  6. HTML & CSS. Available on: <http://socnedo.org/html/notes-of-html/>
  7. HTML. Available on: <http://rajeevsharmamimt.blogspot.com/>
  8. Input Attributes. Available on: <http://www.html5andcss3.org/input.php>
-



## UNIT5

# Cascading Style Sheet

 **Introduction**

You might have noticed in the previous three units that HTML is basically a poor language for page formatting and styling. This is understandable since the scientific environment in which it was conceived, people were more concerned with the content rather than the presentation. To overcome stylistic limitations of HTML, the style sheets represent a major breakthrough for Web page designers, expanding their ability to improve the appearance of their pages. Cascading Style Sheets (CSS), as we know them, enable designers to add styles to the HTML Document. CSS describes how elements are to be displayed on screen. It handles the look and feel of a web page. With the help of CSS, we can control text color, column sizes and layout of tables, text fonts, spacing between paragraphs, columns size and layout, background images or colors, layout designs, and so on. Though very powerful, CSS is easy to learn and understand. This unit provides an introduction to CSS and how we can use them to improve the presentation of our web page.

**Unit Outcomes**

Upon completion of this unit, you should be able to:

1. Describe use and advantages of Cascading Style Sheets.
2. Describe external style sheet, internal style sheet and inline style.
3. Write suitable CSS code for different HTML elements.

**Terminologies**

- CSS:** Cascading Style Sheet is a design language used to give style to a web page
- External Style Sheet:** It contains CSS code in the head section of same web page which is linked to an HTML document.
- Internal Style Sheet:** It is defined within the same web page. It contains CSS code in the head section of same web page.
- Inline Style Sheet:** Inline style is used to give style to a particular tag.



## 5.1. Advantages of CSS

Before we discuss CSS itself, it is worth looking at the potential benefits one might get when we use CSS. Some of the advantages of using CSS are briefly discussed below:

- **CSS saves time** – We can define a style for each HTML element and apply it to as many web pages as we want.
- **Pages load faster** – If we use CSS, web pages can load faster.
- **Easy maintenance** – To make stylistic changes require only changing CSS file which will affect all the web pages using that CSS file.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so we can give a far better look to HTML pages as compared to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for different types of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Since certain HTML attributes are being deprecated therefore it is recommended to use CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** – CSS can store web applications locally with the help of an offline cache. It allows viewing of websites offline. The cache also ensures faster loading and better overall performance of a website.
- **Platform Independence** – The CSS script offers consistent platform independence and can support latest browsers as well.



Use of CSS provides a number of advantages.

## 5.2. Your First Experience with Style

Let us say, we want to develop a web page that suits our stylistic taste. We want that the page background should be light blue, heading (h1) should be centered and displayed in red color with a font size of 20 points. That is, we want our page to look like as follows:

My preferred web page style is here.

### Introduction

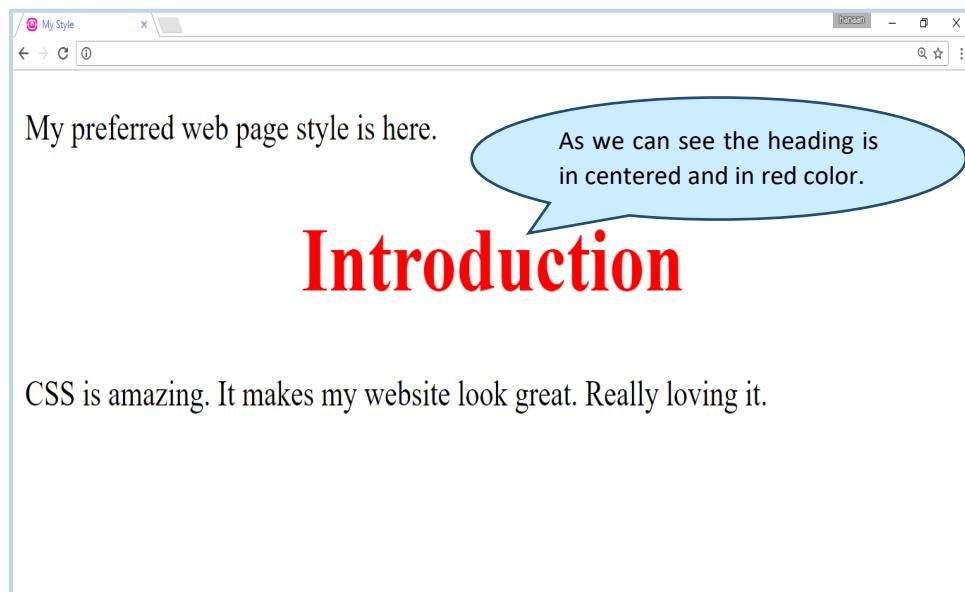
CSS is amazing. It makes my website look great. Really loving it.

Let us see, how we can do all of the above by adding a few more lines of code in our HTML document (don't worry if you do not understand the code at this time, just do it). Here is the code that you should type.

**</> Code 5.1**

```
<!DOCTYPE html>
<html>
<head>
<title>My Style</title>
<style>
h1
{
    font-size: 40px;
    color: red;
    text-align: center;
}
</style>
</head>
<body>
<p>My preferred web page style is here. </p>
<h1>Introduction </h1>
CSS is amazing. It makes my website look great. Really loving
it.
</body>
</html>
```

Output of above code is shown in figure 5.1.



**Figure 5.1:** Output of code 5.1.

What we have done is a simple thing. We simply told the browser how to format and style a web page and its various elements such as headings. Have a look at the code written between `<style>` and `</style>`. It is easy and quite self-explanatory. It tells the browser how to display and format two elements of HTML (body and h1). Now let us get into the details of CSS

## **Activity 1**

Try the above code by changing the font color to blue and font size to 24. See your output in the browser.



## 5.3. Methods of Writing CSS

There are basically three methods for writing CSS or inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

### 5.3.1. External Style Sheet

In external style sheets, the CSS code is stored in a separate file with .css extension (such as mystyle.css) which is then linked to an HTML document using <link> tag. The <link> tag must be in the <head> section of the HTML document as the following code shows.

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



In external style sheet, CSS code is stored in a separate file and linked with an HTML document using <link> tag. CSS files should have .css extension.

For example, to create the web page that we did in our previous example, we first write the following CSS code in a separate file.

```
h1
{
  font-size: 20px;
  color: red;
  text-align: center;
}
```

The above CSS file now can be linked with our main HTML code as shown in the following code.

#### </> Code 5.2

```
<!DOCTYPE html>
<html>
<head>
<title>Images In HTML</title>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
  <p>My preferred web page style is here. </p>
  <h1>Introduction </h1>
  CSS is amazing. It makes my website look great. Really loving
  it.
</body>
</html>
```



## Video Lecture

<https://youtu.be/Ueqbgb-6cZY>



### Activity 2

Change the style sheet and set the font color to blue and font size to 24 using external style sheet. See the output of the HTML file in the browser.

#### 5.3.2. Internal Style Sheet

Internal style sheet is used if a single page has a unique style. Internal styles are defined within `<style>` element, inside the `<head>` section of an HTML page. That is, we do not store the CSS in a separate file rather the CSS code is made a part of the HTML document. Here is the code that we would type when using internal style sheet to get the same look and feel that we got with external style sheet in the earlier example. Notice that we write the CSS code in the document between `<style>` and `</style>` tag in the head section.



We use internal style sheet to specify style for a specific web page.

#### Code 5.3

```
<!DOCTYPE html>
<html>
<head>
<title>My Style</title>
<style>
  h1
  {
    font-size: 20px;
    color: red;
    text-align: center;
  }
</style>
</head>
<body>
  <p>My preferred web page style is here. </p>
  <h1>Introduction </h1>
  CSS is amazing. It makes my website look great. Really loving
  it.

```



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



## Video Lecture

<https://youtu.be/GBITN83inGQ>



### 5.3.3 Inline Style

Inline style is specific to the HTML tags itself. It uses the (style) attributes to style a certain tag directly. The inline style is not recommended when developing a website with multiple pages.

The following example shows how to style `<h1>` directly:

#### `</> Code 5.4`

```
<!DOCTYPE html>  
<html>  
<head>  
<title>My Style</title>  
</head>  
<body>  
  
<p>My preferred web page style is here.</p>  
<h1 style = "font-size: 20px; color: red; text-align: center;"> Introduction </h1>  
CSS is amazing. It makes my website look great. Really loving it.  
  
</body>  
</html>
```



Inline style is used to define style for a single HTML tag in the document.



### Activity 3

In the above code, add `<h2>` and some text under it. Display `h2` with a font size of 16 and in blue color. See the output of the HTML file in the browser

### Which Style Sheet to Use

Generally, it is not very difficult to decide which style sheet to use (external, internal or inline). Simply keep the following advantages and disadvantages of each type of style sheet in your mind while making a choice.

**Table 5.1:** CSS style sheets

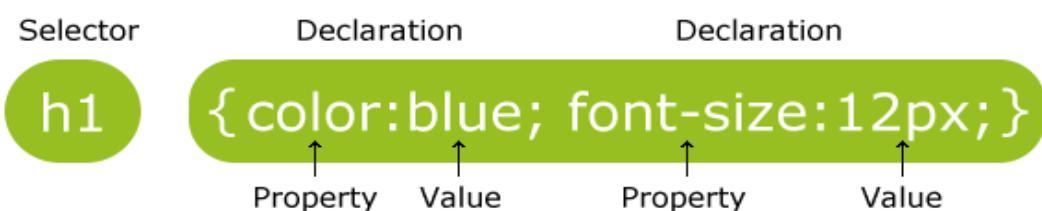
Style Sheet	Advantages	Disadvantages
External	It can be used to set the style of many documents.	It requires extra downtime to load the style sheet.
Internal	It can control the style of each document separately. No additional downtime is required to load the style sheet.	It needs to reapply style information for each document.
Inline	It can control the style of each element. Can be used to override any external or document style.	It needs to apply style information for each element. Need extra work. Documents are difficult to update.

### Activity 4

Which style sheet you will use if a website has a single page? Give reasons.

## 5.4. CSS Syntax

We have used CSS code without discussing how to write it. It is time to get into CSS. The purpose of a style sheet is to create a presentation for a particular HTML element or set of elements. Binding an element to a style specification is very simple. It consists of a rule-set. A rule-set has a selector and one or more declarations. Each declaration consists of a property and its value. The figure below sums it up:



The selector refers to the HTML tag or element that we want to apply the style on. A selector can be an HTML element name, id, class, or an attribute etc. For example, the above figure defines properties for `<h1>` tag. A declaration is a combination of a property name and a value separated by a colon. Each declaration must end with a semi-colon, even if there is only one declaration. The declaration set must be included between curly-braces. A CSS rule-set can be written in a single line, or multiple lines. That is we can define the style for `<h1>` as

```

h1
{
    font-size: 20px;
    color: red;
}
    
```



```
        text-align: center;  
    }
```

The above style definition can also be written on a single line as shown below

```
h1 {font-size: 20px; color: red; text-align: center;}
```



Each declaration consists of a property name and its value pair and ends with a ";".

We have mentioned above that a selector can be an HTML element name, id, class, or an attribute. Let us see how they are different from each other.

#### 5.4.1. Element Selector

An element selector specifies the style for an element by name. For example, if we want to change the style of `<p>` element in a web page, we need to define a style for `<p>` as shown below.

```
p  
{  
    font-size: 25px;  
    color: white;  
    text-align: center;  
}
```



Element selector can be used to define the style for HTML element name.

Once the style is defined, it will be applied to all paragraphs (tag `<p>`) automatically by the browser. A complete example of HTML code is given below. Note that we are using internal style sheets in our example. You can use external style sheets, if you like.

#### `</> Code 5.5`

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Element Selector</title>  
<style>  
    body  
    {  
        background-color: blue;  
    }  
    p  
    {  
        font-size: 25px;  
        color: white;  
        text-align: center;  
    }  
</style>  
</head>  
<body>  
    <p>This line is written in first Paragraph </p>  
    <p>This line is written in second Paragraph </p>  
</body>
```

```
</html>
```

The output of above code is shown in figure 5.2.

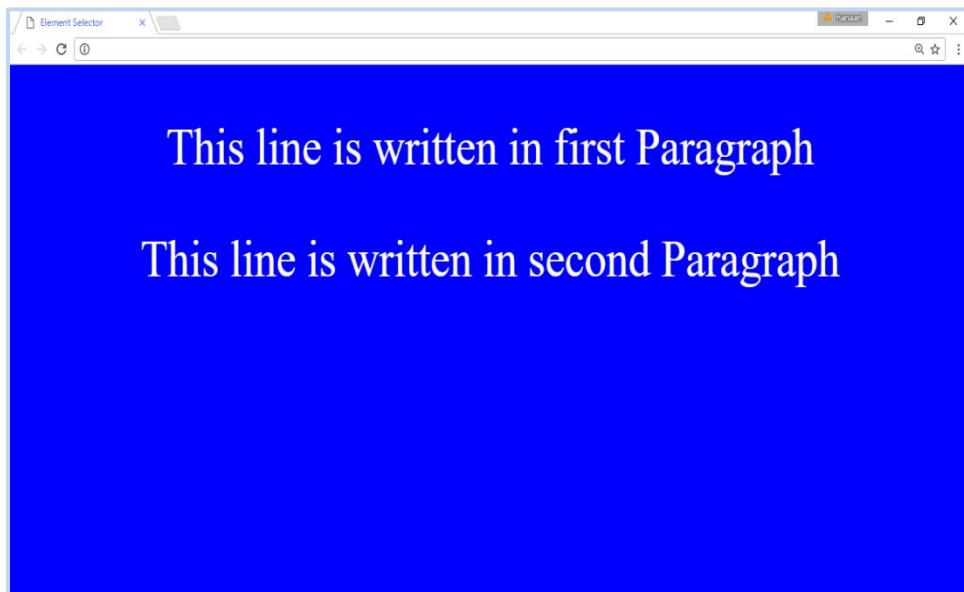


Figure 5.2: Output of code 5.5

## Activity 5

1. Create the above web page by removing the style from the HTML document and store it in a CSS file instead. Link your document with the style sheet.
2. Repeat the above activity by using inline style.

### 5.4.2. ID Selector

The ID selector is used to select a specific HTML element. The ID of an element must be unique. To select an element with a specific ID, (#) character is used with ID name. Suppose we want to apply a style to one or more specific paragraphs (but not to all), then we should define the style as:

```
#p1
{
    font-size: 25px;
    color: white;
    text-align: center;
}
```

Now, we can use the above style with a paragraph by specifying the above style as shown below:

```
<p id="p1">This line is written in second Paragraph </p>
```

Note that the element selector applies to all whereas an ID selector is applied to the selected elements. We can define multiple styles for a single element. Try the following example and observe the output. We notice that style is applied to the second paragraph only.

**</> Code 5.6**

```
<!DOCTYPE html>
<html>
<head>
<title>ID Selector</title>
<style>
body
{
    background-color: blue;
}
#p1
{
    font-size: 25px;
    color: white;
    text-align: center;
}
</style>
</head>
<body>
    <p>This line is written in first Paragraph </p>
    <p id="p1">This line is written in second Paragraph </p>
</body>
</html>
```

The output of above code is shown in figure 5.3.

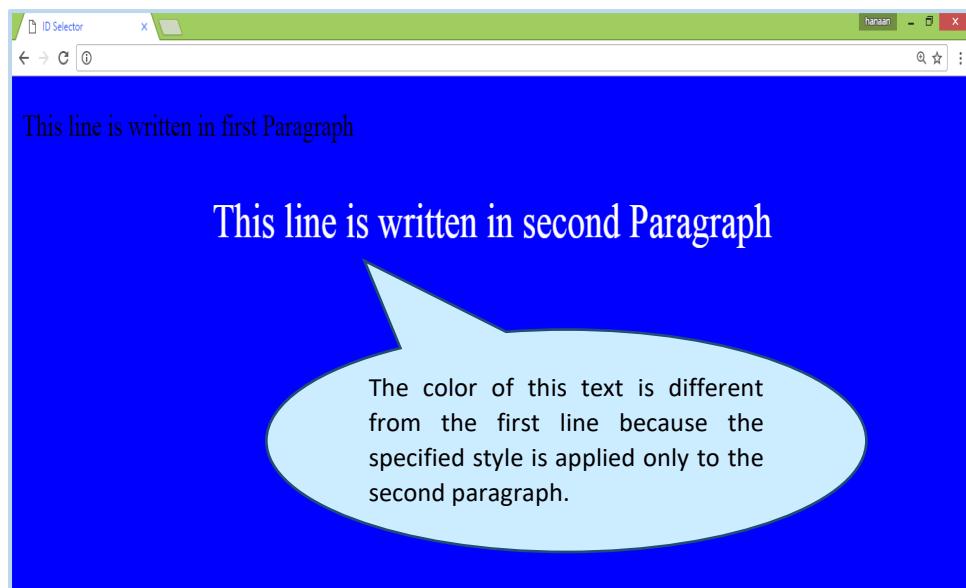


Figure 5.3: Output of code 5.6

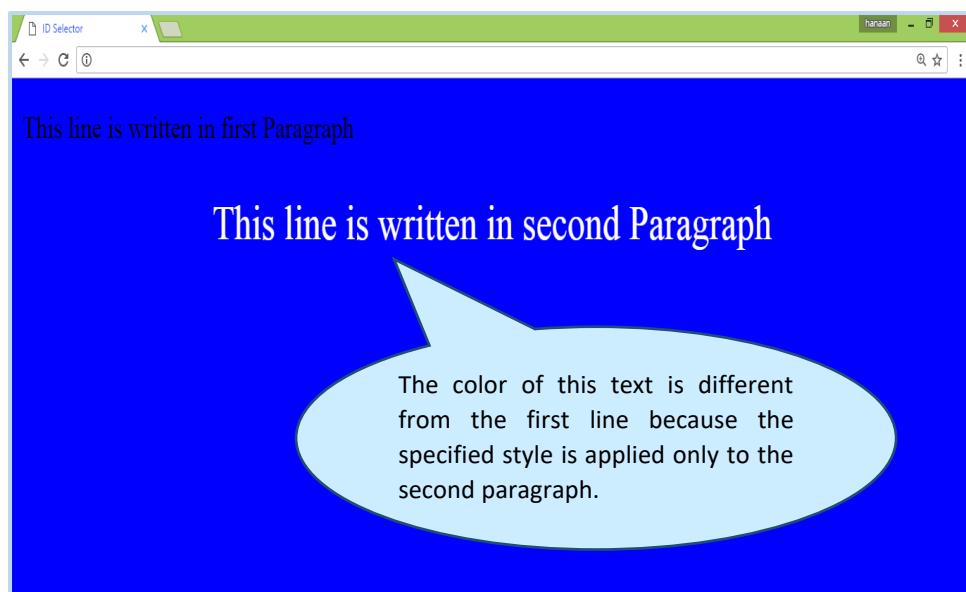
#### 5.4.3. Class Selector

Class selector selects the element with a specific class attribute. To select an element of a specific class (.) character is used with class name. Suppose we have two paragraphs and want to apply different style on the second paragraph. We give an ID to the second paragraph in order to apply different CSS properties. A class selector and an ID selector are somehow similar to each other, however, an ID selector identifies an element only, whereas a class selector can identify more than one elements.

**</> Code 5.7**

```
<!DOCTYPE html>
<html>
<head>
<title>Class Selector</title>
<style>
  body
  {
    background-color: blue;
  }
  .p1
  {
    font-size: 25px;
    color: white;
    text-align: center;
  }
</style>
</head>
<body>
  <p>This line is written in first Paragraph </p>
  <p class="p1">This line is written in second Paragraph </p>
</body></html>
```

The output of above code is shown in figure 5.4.



**Figure 5.4:** Output of code 5.7

## 5.5. CSS Comments

Writing a comment within a code is very useful because it helps to understand and modify the source code at later time. when you edit the source code at a later time, a comment is written as:

```
/* A comment */
```



Comments are written within /\* and \*/. Comments are not executed by the browser.

The following code shows how useful comments can be written in an HTML document. See the code below how we can write useful comments in an HTML document.

**</> Code 5.8**

```
<!DOCTYPE html>
<html>
<head>
<title>Element Selector</title>
<style>
body
{
    background-color: blue; /* Set background color to blue */
}
.p1 /* Specify the style for paragraph */
{
    font-size: 25px;
    color: white;
    text-align: center;
}
</style>
</head>
<body>
    <p>This line is written in first Paragraph </p>
    <p class="p1">This line is written in second Paragraph </p>
</body>
</html>
```



## 5.6. Basic Properties of CSS

We discussed how rules are formed in CSS style sheets but did not mention what are various properties that can be set. CSS defines a large number of properties and their values. Some of the basic properties are:

1. Background property
2. Border, margin & padding properties
3. Height / width property
4. Text
5. Fonts
6. links
7. Lists
8. Tables

We will discuss how to set the above properties in this and the next unit. Let us start with the background property.



### 5.6.1. Background Properties

Background properties are used to define the background effects for elements. Some of the background properties are given in the following table:

**Table 5.2:** Background properties with their description

Property	Description
background-color	It specifies which background color is to be used
background-image	It specifies one or more background images to be used
background-repeat	It specifies how to repeat the background images
background-attachment	It specifies whether the background images are fixed or scrolls with the rest of the page
background-position	It specifies the position of the background images

Let us see how these properties are defined.

#### **background-color**

This property defines the background color of an HTML element or a document. The background-color property is defined as:

```
background-color: color;
```

A color can be specified in one of the following three methods:

1. Using a valid color name, like red
2. Using a RGB value, like `rgb(255, 0, 0)`
3. Using a hexadecimal value like `#ff0000`.

For example, to set background color of an HTML document, we specify the `background-color` property as:

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

The following example shows how to set background color for a paragraph

```
p
{
    background-color: yellow;
}
```

Similarly, the following code can be used to set the `background-color` for `<h1>`.

```
h1
{
    background-color: yellow;
}
```

See the HTML code below that sets the background color for the whole document to blue.

**</> Code 5.9**

```
<!DOCTYPE html>
<html>
<head>
<title>Background Property in CSS</title>
<style>
body
{
    background-color: blue;
}
</style>
</head>
<body>
    <h1>Playing with colors</h1>
    <p>This line is written in first Paragraph </p>
    <p>This line is written in second Paragraph </p>
</body>
</html>
```

**Activity 6**

1. Change the background color of the document to red and background color of heading h1 to white.
2. Repeat the above activity by storing the style in a CSS file and linking it to your HTML document.

**background-image**

We can use an image as a background of an HTML document to make it look like a wallpaper. The following example illustrates how to use background-image property:

```
body
{
    background-image: url("paper.gif");
```



By default, background-image property repeats the images both vertically and horizontally.

By default, the image is repeated on the screen to cover the whole document. Try the following code and see the output in the browser.

**</> Code 5.10**

```
<!DOCTYPE html>
<html>
<head>
<title>Element Selector</title>
<style>
body
{
    background-image: url("new.jpg");
}
```

```
p  
{  
    color: white;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
    <p>This is Paragraph 1 </p>  
    <p>This is Paragraph 2</p>  
</body>  
</html>
```

The output of above code is shown in figure 5.5.

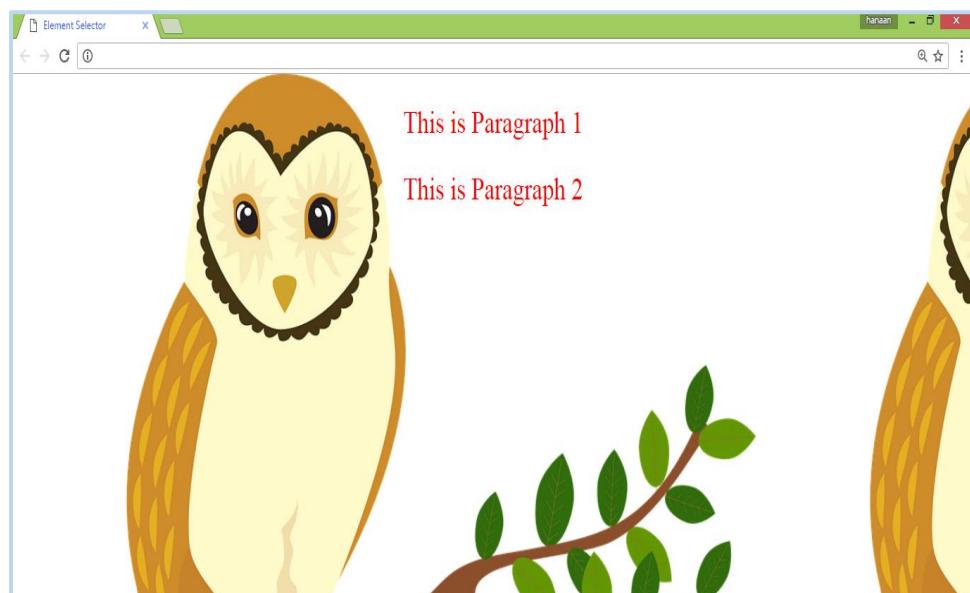


Figure 5.5: Output of code 5.10

## Activity 7

Change the background image in the above code to one of your favorite images.

### background-repeat

Some images are designed in a way that they can be attached to each other, either horizontally or vertically, to form a certain design. By default, background-image property repeats the images both vertically and horizontally. To set the repeat direction of a background image, we use background-repeat property that accepts one of two values: repeat-x or repeat-y. To prevent the image from being repeated, we use the value no-repeat.

The following example shows how to use background-repeat property. Keep in mind that background-repeat must be used along with background-image property and not by itself.

```
body  
{  
    background-image: url("tree.png");  
    background-repeat: repeat-x;
```



}



background-image can be repeated either horizontally or vertically by specifying background-repeat property.

The following code displays an image in the background with repeat property. The output is amazing.

**</> Code 5.11**

```
<!DOCTYPE html>
<html>
<head>
<title>Element Selector</title>
<style>
    body
    {
        background-image: url("tree.png");
        background-repeat: repeat;
    }
    p
    {
        color: red;
        text-align: center;
    }
</style>
</head>
<body>
</body>
</html>
```

The output of above code is shown figure 5.6.

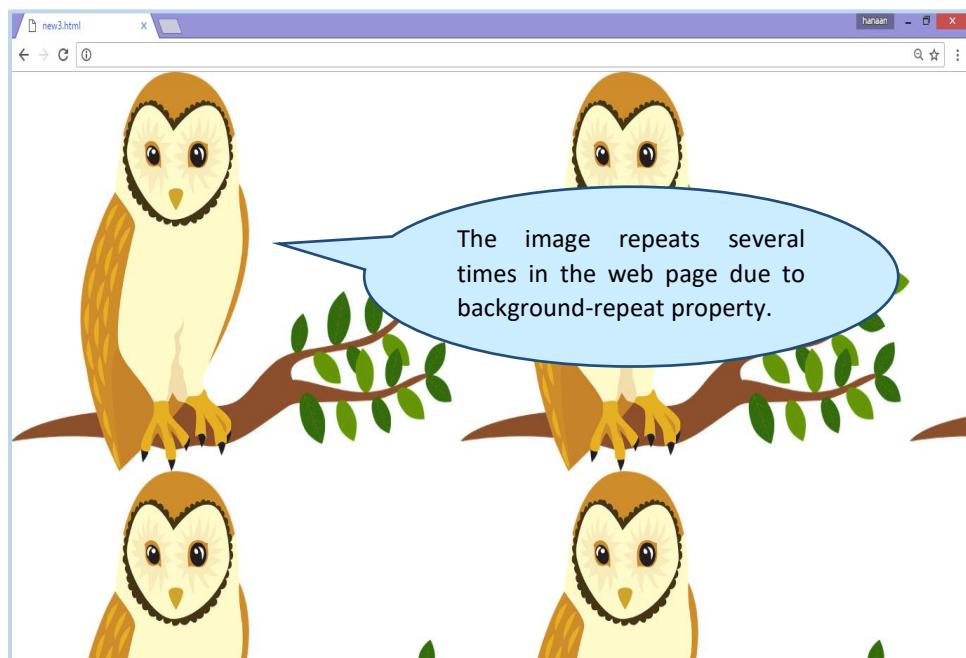


Figure 5.6: Output of code 5.11



## Video Lecture

<https://youtu.be/oKK9ljKweqs>



### Activity 8

1. Change the repeat property in the above code to repeat-x, repeat-y and no-repeat. What do you notice in the output?
2. Repeat the above activity by storing the style sheet in a CSS file and linking it to HTML document.

### background-attachment

This property determines whether that background image will scroll or not when the content with which it is associated scrolls. This can be specified by background-attachment property as shown in the following code.

```
body
{
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-attachment: fixed;
}
```

In the above code, we set the background-attachment to "fixed" which allow us to scroll up and down in the document without moving the background image.



The scrolling of an image can be controlled with background-attachment property.

### background-position

This property determines the position of the background image within the canvas space used by its element. The background position can be set to "left top", "left center", "left bottom", "right top", "right center", "right bottom", "center top", "center center". If we specify only one value only, the second value is taken as center.



We can also set background position where image is going to be placed.

The following example defines the position of the background image as top right side of the document.

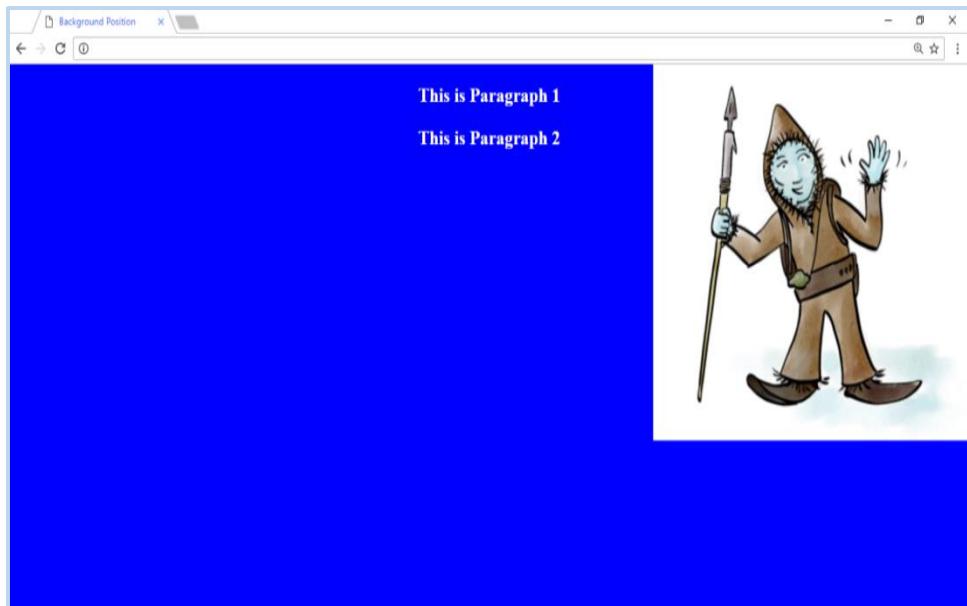
```
body
{
    background-image: url("tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
}
```

The following code displays an image on the top right side of the web page. We can see in the output that the image is displayed on the top right side of the web page, while “background-attachment” is used to fix the background image so that image will not scroll.

**</> Code 5.12**

```
<!DOCTYPE html>
<html>
<head>
<title>Element Selector</title>
<style>
body
{
    background-color: blue;
    background-image: url("tree.png");
    background-repeat: no-repeat;
    background-position: right top;
}
p
{
    color: white;
    text-align: center;
}
</style>
</head>
<body>
    <p>This is Paragraph 1 </p>
    <p>This is Paragraph 2</p>
</body>
</html>
```

The output of code 5.12 is shown in figure 5.7.



**Figure 5.7:** Output code 5.12.

### **Activity 9**

Use different values for background-position property in the above code and observe its effect in the output.



### **Unit Summary**

In this unit, we learned about advantages and the use of Cascading Style Sheet in web page development. We discussed how CSS code can be written to style different HTML elements. We have also discussed the differences and usage of external, internal and inline style sheets.



## Self Assessment Questions

Choose the correct answer.

1. Which of the following is correct about CSS?
  - A. CSS is used to control the style of a web document in a simple and easy way.
  - B. CSS is the acronym for "Cascading Style Sheet".
  - C. You can write CSS once and then reuse same sheet in multiple HTML pages.
  - D. All of the above.
  
2. Which of the following property is used to set the background image of an element?
  - A. background-color
  - B. background-image
  - C. background-repeat
  - D. background-position
  
3. The correct place to refer an external style sheet in an HTML document is
  - A. <stylesheet>mystyle.css</stylesheet />
  - B. <style src = "mystyle.css" />
  - C. <link rel = "stylesheet" type = "text/css" href = "mystyle.css">
  - D. None of the above
  
4. Where in an HTML document is the correct place to refer to an external style sheet?
  - A. At the end of the document
  - B. In the <head> section
  - C. At the top of the document
  - D. In the <body> section
  
5. Which HTML tag is used to define an internal style sheet?
  - A. <style>
  - B. <css>
  - C. <script>
  - D. All of the above
  
6. Which is the correct CSS syntax?
  - A. body {color: black}
  - B. {body;color:black}
  - C. {body:color=black(body)}
  - D. body:color=black
  
7. How do you add a background color for all <h1> elements?
  - A. all.h1 {background-color:#FFFFFF}
  - B. h1.all {background-color:#FFFFFF}
  - C. h1 {background-color:#FFFFFF}
  - D. None of the above



8. What is the correct CSS syntax for making all the <p> elements bold?
  - A. <p style = "text-size:bold">
  - B. p {font-weight:bold}
  - C. p {text-size:bold}
  - D. <p style = "font-size:bold">
9. What is the correct CSS syntax for displaying background image?
  - A. body { background-image: url("paper.gif"); }
  - B. background.body: "paper.gif"
  - C. body {background-image = "paper.gif"
  - D. None of the above
10. Which CSS property is used to specify position of the background image?
  - A. background-position
  - B. image-position
  - C. background-image-position
  - D. None of the above

**Answer Key:**

1. D	2. B	3. C	4. B	5. A	6. A	7. C	8. B	9. A	10. A
------	------	------	------	------	------	------	------	------	-------



### Review Questions

**Write short answers of the following questions.**

1. What are the benefits of using CSS?
2. Differentiate between external, internal and inline style sheets?
3. How an external style sheet is linked to a HTML document?
4. What is CSS syntax to define properties of an HTML element?
5. List background properties and their possible values.

### </> Coding Exercise

1. Create a CSS with suitable background properties for a web page that gives some basic information about your city (add some nice photos as well).
2. Create a website that gives basic information about yourself. You should prepare a separate page for each of the following: personal information, academic qualification, professional qualification, job experience, hobbies, and photo gallery. Academic qualification page should display the information about your degrees and certificates along with the subjects you studied in each in a tabular form. Create suitable CSS file and link it with all of your pages.
3. Write suitable background properties for a web page that provides basic information about a number of medicines.



## References and Further Reading

1. Duckett, J. (2011). HTML and CSS: design and build websites. John Wiley & Sons.
  2. Robbins, J. N. (2012). Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. O'Reilly Media, Inc.
  3. Nixon, R. (2012). Learning PHP, MySQL, JavaScript, and CSS: A step-by-step guide to creating dynamic websites. O'Reilly Media, Inc.
  4. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
  5. Casario, M., Wormser, N., Saltzman, D., Bradford, A., Reid, J., Improta, F., & Congleton, A. (2013). CSS3 Solutions: Essential Techniques for CSS3 Developers. Apress.
  6. Advantages of CSS. Available on: [http://www.tutorialspoint.com/css/css\\_quick\\_guide.htm](http://www.tutorialspoint.com/css/css_quick_guide.htm)
-





## UNIT 6

# Text, Border and Table Properties in CSS



## Introduction

In the previous unit, we discussed some basic concepts of CSS and how we can enhance the appearance of our web pages by specifying various properties of HTML document and its elements. We continue to explore CSS in this unit as well to see how we specify some other documents, texts and table properties.



## Unit Outcomes

Upon completion of this unit, you should be able to:

1. Specify and use various text properties.
2. Specify and use various table properties.
3. Specify and use various document properties.
4. Develop a web page that use CSS.



## Terminologies

**Text properties:** It is used to change the presentation and layout of text.

**Border properties:** It allows to customize border in an HTML document.

**Margin properties:** It is used to set spaces around the elements.

**Table properties:** It allows to set presentation and layout of table elements.



## 6.1. Text properties in CSS

Text properties are used to change the presentation, spacing and layout of text. The basic text properties enable the web developers to set text color, alignment, decoration, text indentation, line height, word spacing and others. Some of the important text properties are given in the following table.



Text properties can be used to change the appearance and layout of text.

**Table 6.1:** Text Properties

Property	Description
color	This property is used to set the color of text
text-align	It is used to set the horizontal alignment of a text. Possible values are right, center and justify.
text-decoration	This property is used to define an effect on text. The standard values for text-decoration include line-through, underline, overline and none.
text-transform	This property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word. Possible values for this property are uppercase, lowercase and capitalize.
text-indent	It is used to specify indentation of 1 <sup>st</sup> line.
word-spacing	This property is used to specify the space between the words in a text.

For example, the following CSS code sets properties of <h1> to red color, center and underline.

```
h1
{
    color: red;
    text-align: center;
    text-decoration: underline;
}
```

If we want that each sentence of a document starts with a capital letter, we will specify it as

```
body
{
    text-transform: capitalize;
}
```

The following code can be used to set the indent and text align properties of a paragraph to 12pt and justify.

```
p {text-indent: 12pt; text-align: justify;}
```

Below is a complete HTML and CSS code that specifies various properties for the document and `<h2>` tag.

#### `</> Code 6.1`

```

<!DOCTYPE html>
<html>
<head>
<title>Text Properties</title>
<style>
body
{
    background-color: blue;
    color: white;
    text-transform: capitalize;
    text-indent: 50px;
}
h2
{
    text-decoration: underline;
    text-align: center;
    letter-spacing: 3px;
    word-spacing: 3px;
}
</style>
</head>
<body>
    <img src = "tree.png" />Image 1
    <h2>Text CSS Properties </h2>
    <div> This div element contains all properties of text such as
        text-align, text-decoration, word spacing, letter spacing,
        background color, color, text transformation, text indent
    </div>
</body>
</html>

```

The output above code is shown in figure 6.1.

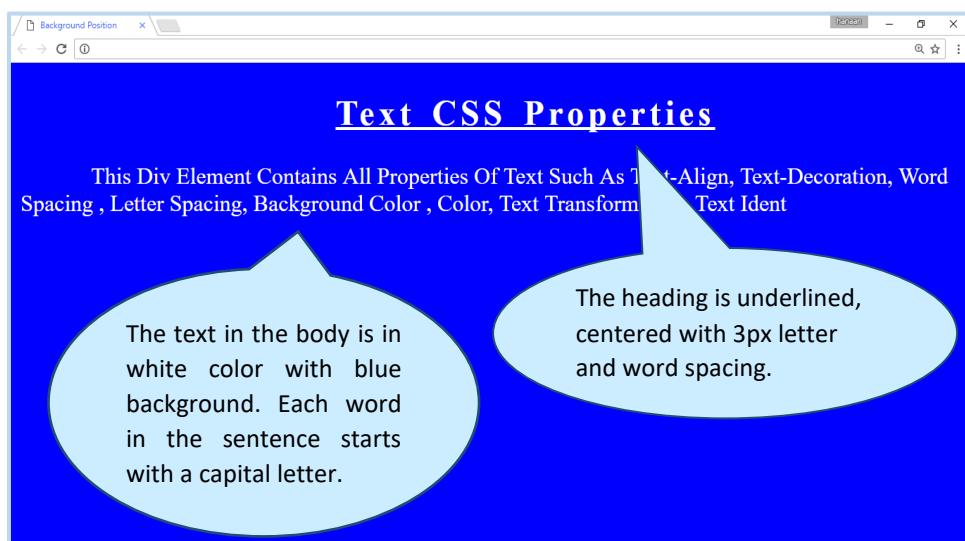


Figure 6.1: Output of code 6.1.



## Video Lecture

<https://youtu.be/IgsTOtqGOhA>



### Activity 1

Specify the following properties for the above document: Heading should be red, 20 points, centered and underlined. Paragraph text should be in blue, left justified and indented by 20 point.



## 6.2. Border Properties

Border properties allow us to specify the style, width and color of border of a document and its elements. The (“**border-style**”) property specifies what kind of border you want to display. The following values are allowed with “border-style” property:



Border properties can be used to specify border style, width and color.

**Table 6.2: Border Properties**

Values	Description
Dotted	The dotted value displays a dotted border.
Solid	The solid value displays a solid border.
Dashed	The dashed value displays a dashed border.
Groove	The groove value defines a 3D grooved border and it depends on the border color.
Inset	The inset value defines a 3D inset border and it depends on the border color.
Outset	The groove value defines a 3D outset border and it depends on the border color.
None	The none value defines no border.
Hidden	The hidden value defines a hidden border.
Double	The double value defines a double border.

We can define different border types for different elements of a web page. We can also set border width by using border-width property. For example, to specify a thick border for a paragraph, we use the following code

```
p {border-style: solid; border-width: thick;}
```

The following brief example shows how we can define different types of borders for different paragraphs of a document.



We can set border style as well as its thickness using border properties. Explore other possible values for the border-width property.

#### **</> Code 6.2**

```
<!DOCTYPE html>
<html>
<head>
<title>Border Properties</title>
<style>
body
{
    background-color: blue;
    color:white;
}
h2
{
    text-align: center;
    font-size: 25px;
}
#p1 {border-style: dotted;}
#p2 {border-style: solid;}
#p3 {border-style: dashed;}
#p4 {border-style: groove;}
#p5 {border-style: inset;}
#p6 {border-style: outset;}
#p7 {border-style: none;}
#p8 {border-style: hidden;}
#p9 {border-style: double;}
</style>

</head>
<body>

    <h2> Border Properties</h2>
    <p id = "p1">A dotted border</p>
    <p id = "p2">A solid border</p>
    <p id = "p3">A dashed border</p>
    <p id = "p4">A groove border</p>
    <p id = "p5">An inset border</p>
    <p id = "p6">An outset border</p>
    <p id = "p7">No border</p>
    <p id = "p8">A hidden border</p>
    <p id = "p9">A double border</p>
</body>
</html>
```



The output of the above code is shown in figure 6.2.

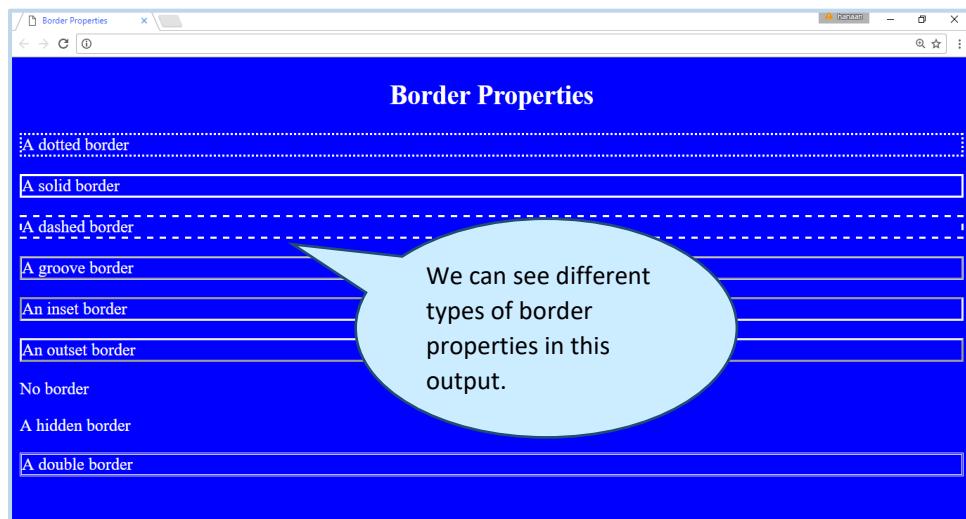


Figure 6.2: Output of code 6.2



## Video Lecture

<https://youtu.be/YMiyQxjQ-fk>



### Activity 2

Write a small document that gives some information about 3 different cities. The information for each city should be written in different types of borders having different thicknesses.



## 6.3. Margin Properties

Margin properties are used to give spaces around the elements. The CSS margin property covers four sides of an element i.e. top, right, bottom & left. Margin properties can have the following values:

- **auto:** The browser automatically generates the margin
- **length:** Specifies margin in px, pt, cm etc.
- **%:** Specifies margin in % of width of containing element.



Margin properties can be used to specify the margin around HTML document.

In the following example we give margin to the all four sides of <div> element. See the effect of it in the output.

**</> Code 6.3**

```
<!DOCTYPE html>
<html>
<head>
<title>Margin Properties</title>
<style>
body
{
    background-color: blue;
    color: white;
    font-size: 20px;
}
div
{
    border: 1px solid black;
    margin-top: 120px;
    margin-bottom: 120px;
    margin-right: 170px;
    margin-left: 100px;
}
h2
{
    text-align: center;
    font-size: 25px;
}
</style>
</head>
<body>
    <h2> Margin Properties in CSS</h2>
    <div>This div element has a top margin of 120px, a right
        margin of 170px, a bottom margin of 120px, and a left margin
        of 100px. </div>
</body>
</html>
```

The following output shows that how spaces are created through margin.

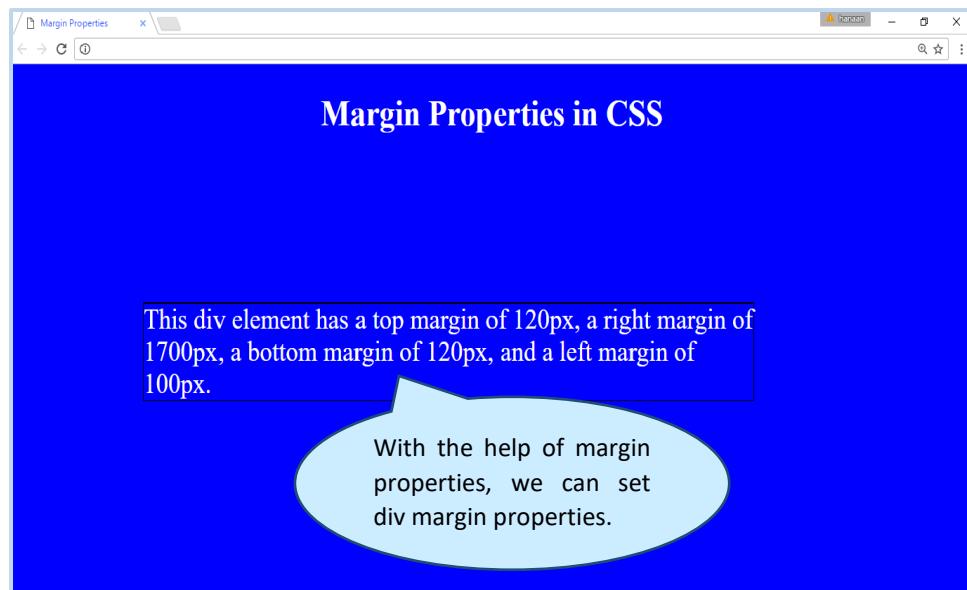


Figure 6.3: Output of code 6.3

## 6.4. Padding Properties in CSS

Padding property is used to give spaces around the HTML element. CSS property for setting the padding for each side of <div> element is shown in the following code.

### </> Code 6.4

```
<!DOCTYPE html>
<html>
<head>
<title>Padding Properties</title>
<style>
body
{
    background-color: blue;
    color: white;
    font-size: 20px;
}
div
{
    border: 1px solid black;
    padding-top: 50px;
    padding-bottom: 50px;
    padding-right: 50px;
    padding-left: 50px;
}
h2
{
    text-align: center;
    font-size: 25px;
}
</style>
</head>
<body>
    <h2> Padding Properties in CSS</h2>

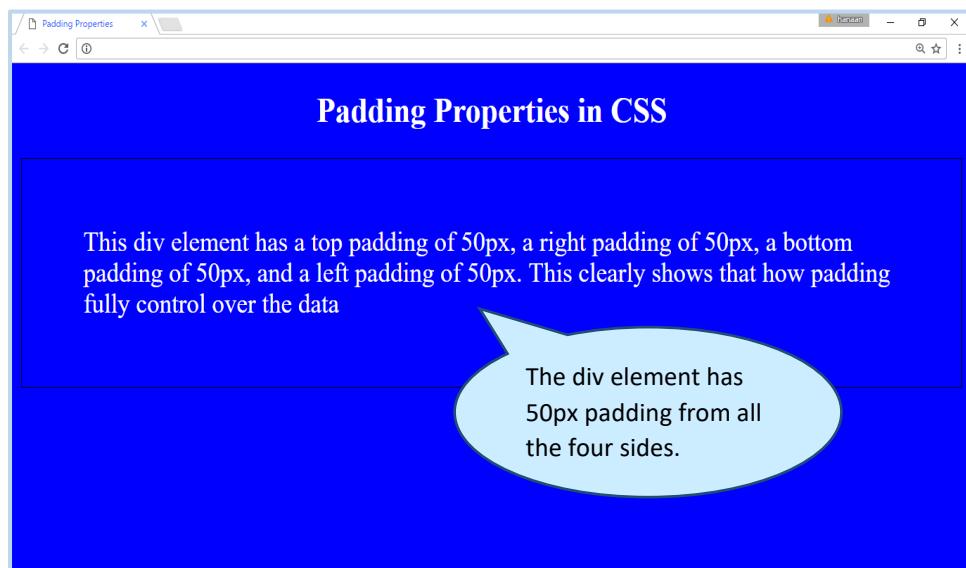
```

```

<div>This div element has a top padding of 50px, a right
padding of 50px, a bottom padding of 50px, and a left padding
of 50px. This clearly shows that how padding fully control
over the data</div>
</body>
</html>

```

The following output shows that how padding property adds spaces around the text.



**Figure 6.4:** Output of code 6.4



Padding properties are used to generate spaces around the content.

### Activity 3

Change the top and bottom heading to 100px and see its output.

## 6.5. Table Properties

If we want to give style to a table such as changing font color, border color, border width etc. we have to set various table properties. Some of the important table properties are given in the following table

**Table 6.3:** Table Properties

Properties	Description
border	It sets all the border properties in one declaration
border-collapse	It specifies whether or not table borders should be collapsed
caption-side	It specifies the placement of a table caption



We can add different styles to a table by using table properties.

### border property

The border property can be used to different properties in the following order. The general format of border property is

```
border: border-width border-style border-color
```

The border-width property sets the width of an element's four borders. The value of border-width property can be thin, medium or thick. We can also specify the thickness in number of pixels.

The border-style property can have one of these values: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset. The border-color property can be used to set the color of the four borders. For example, the following code sets table border having thick border width, double line style and red color

```
<style>
  table
  {
    border: thick double red;
  }
</style>
```

Similarly, we can set properties of <tr>, <th> and <td> tags as well. The following CSS code sets the background color of <th> to green and text color to white.

```
th
{
  background-color:green;
  color:white;
}
```

Below is complete HTML and CSS code that sets various properties of whole document body as well as of the table.

#### </> Code 6.5

```
<!DOCTYPE html>
<html>
<head>
<title>Table Properties</title>
<style>
  body
  {
    background-color: blue;
    color: white;
    font-size: 20px;
  }

  table
  {
    border-collapse: collapse;
  }
</style>

```

```
}

table, th, td
{
    border: 1px solid black;
}
h2
{
    text-align: center;
    font-size: 25px;
}
</style>
</head>
<body>
<h2> Table Properties in CSS</h2>
<table>

<tr>
<th>First name</th>
<th>Last name</th>
</tr>

<tr>
<td>Ali</td>
<td>Ahmad</td>
</tr>
<tr>
<td>Zahid</td>
<td>Ahmad</td>
</tr>
</table>
</body>
</html>
```

The output of the above code is shown in figure 6.5.

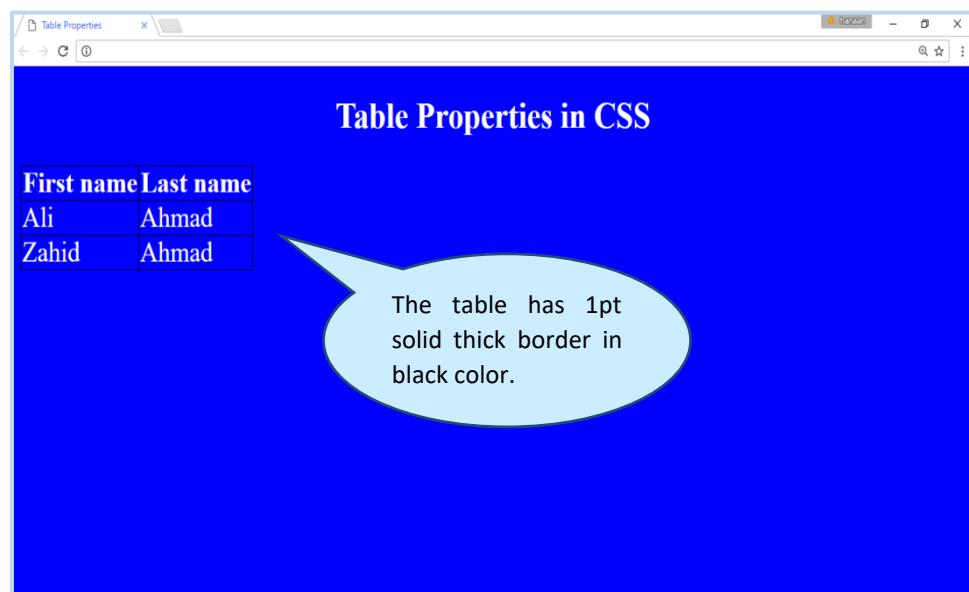


Figure 6.5: Output of code 6.5

### border collapse



By default, table as well as columns of a table have their separate borders. The border-collapse property can be used to have a single border for table as well as for the columns. This property is specified as

```
table
{
    border-collapse: collapse;
}
```

#### **caption side**

The caption-side property specifies the placement of caption of a table. Its default value is top which can be changed to bottom using the following code

```
caption
{
    caption-side: bottom;
}
```

#### **Activity 4**

Create a table given in figure 6.5 with suitable caption at the bottom. Also use the border-collapse property to collapse the border of a table.

## **6.6. Position Properties**

Position is the property that lets the browser to adjust the position of HTML elements. We can move elements on screen by giving properties top, down, left and right.



Position is the property that tells the browser how to position the element.

Position property specifies the type of position method used for an element. There are four different position values.

- Static
- Relative
- Fixed
- Absolute

#### **Static Positioning**

This is the default way HTML elements are positioned. Static position elements are not affected with top, bottom, left and right properties. The element with static position is not specifically positioned, they always positioned as normal flow on the page.

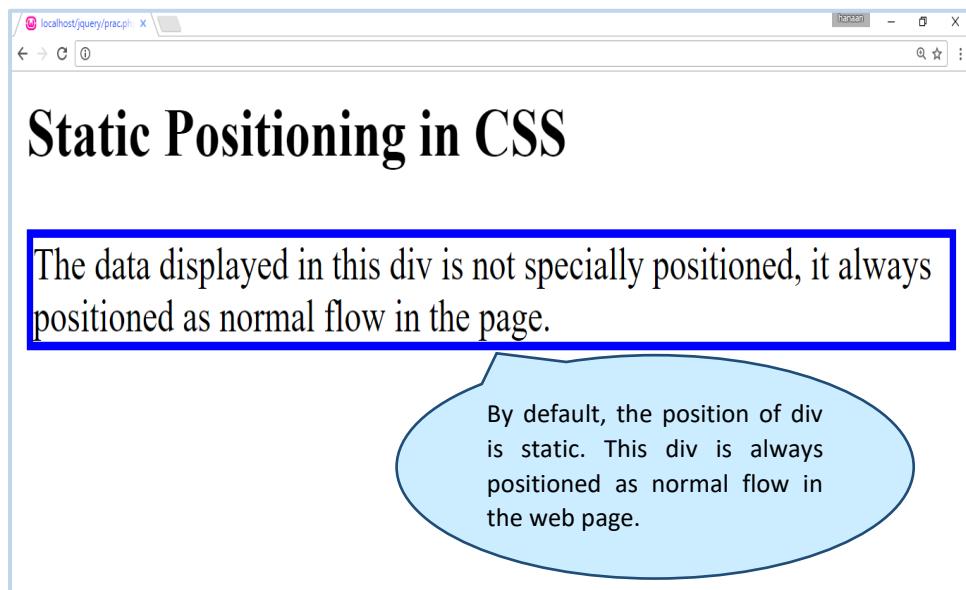


Static position is the default position of an HTML element.

**</> Code 6.6**

```
<!DOCTYPE html>
<html>
<head>
<style>
#static
{
position: static;
border: 3px solid blue;
}
</style>
</head>
<body>
    <h2> Static Positioning in CSS</h2>
    <div id = "static">
        The data displayed in this div is not specially positioned, it
        always positioned as normal flow in the page.
    </div>
</body>
</html>
```

Output of the code 6.6 is shown in figure 6.6.



**Figure 6.6:** Output of code 6.6

### Absolute Positioning

Absolute position places the HTML element on the page at precise position. We can place elements anywhere in the web page by simply giving values to the properties top, down, left and right. With the help of following example you can easily understand the working of an absolute position in a web page.

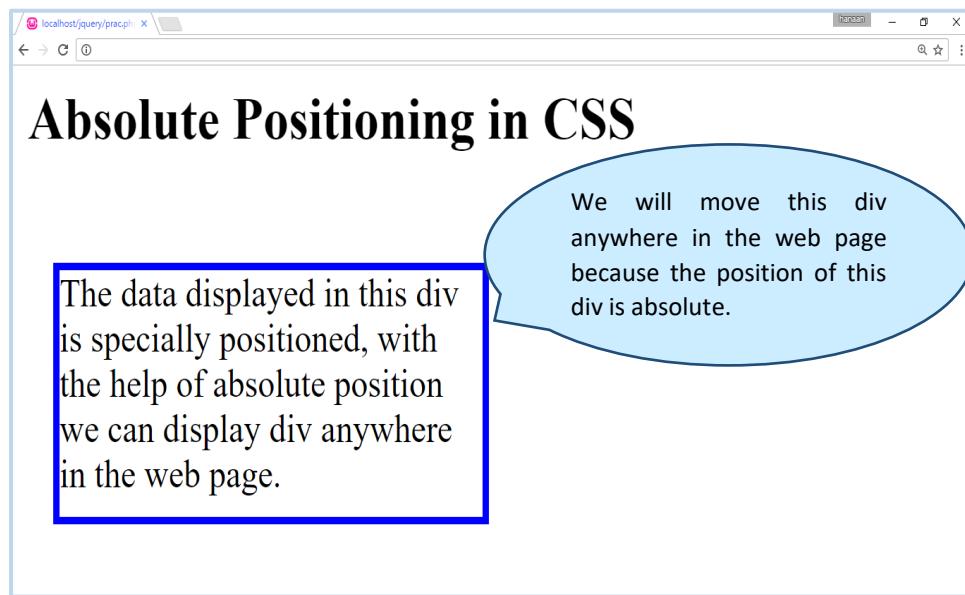


With the help of absolute positioning, you can place your element anywhere in your web page.

**</> Code 6.7**

```
<!DOCTYPE html>
<html>
<head>
<style>
    .absolute
    {
        position: absolute;
        top: 80px;
        left: 20px;
        width: 200px;
        height: 100px;
        border: 3px solid blue;
    }
</style>
</head>
<body>
    <h2> Absolute Positioning in CSS</h2>
    <div class = "absolute">
        The data displayed in this div is specially positioned, with
        the help of absolute position we can display div anywhere in
        the web page.
    </div>
</body>
</html>
```

The output of the code 6.7 is shown in figure 6.7.



**Figure 6.7:** Output of code 6.7

### Relative Positioning

Relative position means "relative to itself". For example, if we set position relative to an element but not giving any attributes (top, left, bottom, right), the position of an element will be considered as static. But if we give some position attributes, let us say top = "10px", it will shift position 10 pixels down where it would normally be. With the help of following code, you can easily understand the working of relative position.

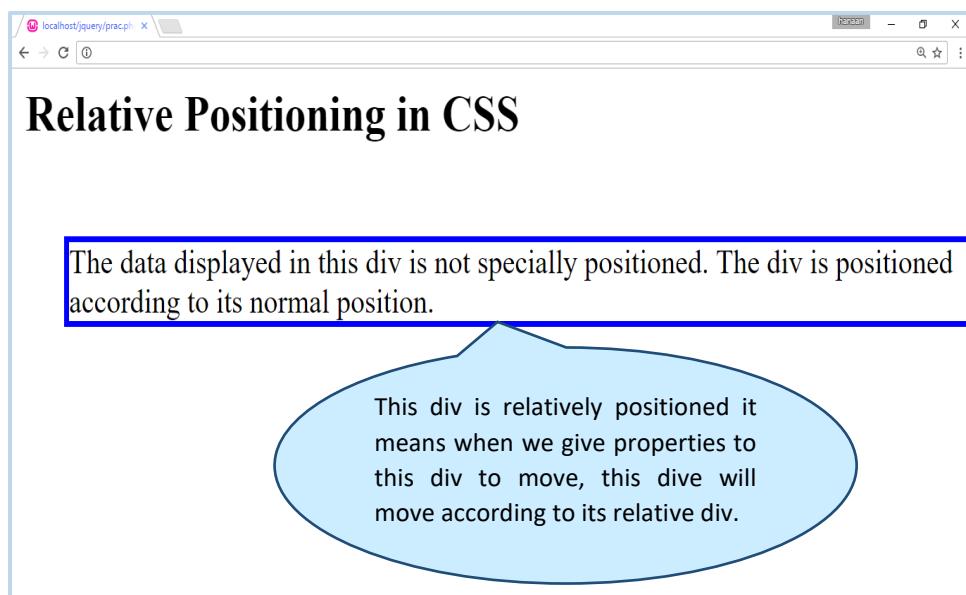


Relative position is the position with respect to other elements.

#### </> Code 6.8

```
<!DOCTYPE html>
<html>
<head>
<style>
.relative
{
    position: absolute;
    top: 80px;
    left: 30px;
    border: 3px solid blue;
}
</style>
</head>
<body>
<h2> Relative Positioning in CSS</h2>
<div class = "relative">The data displayed in this div is not
specially positioned. The div is positioned according to its
normal position.
</div>
</body>
</html>
```

The output of the code 6.8 is shown in figure 6.8.



The data displayed in this div is not specially positioned. The div is positioned according to its normal position.

This div is relatively positioned it means when we give properties to this div to move, this dive will move according to its relative div.

Figure 6.8: Output of code 6.8

#### Fixed Positioning

Fixed position is the **positioned** relative to viewport which means it always stays in the same place even if page is scrolled. You can use properties top, left, right and bottom to position the element.



In fixed position, your element position is fixed whether you can scroll down your page.

#### </> Code 6.9

```
<!DOCTYPE html>
<html>
<head>
<style>
.fixed
{
    position: fixed;
    bottom: 10px;
    right: 80px;
    left: 30px;
    width: 300px;
    border: 3px solid blue;
}
</style>
</head>
<body>
    <h2> fixed Positioning in CSS</h2>
    <p> You can see that the div that appears below has fixed positioned. </p>
    <div class = "fixed">
        The data displayed in this div is not specially positioned. B
        The div is positioned according to its normal position.
    </div>
</body>
</html>
```

The output of code 6.9 is shown in figure 6.9.

fixed Positioning in CSS

You can see that the div that appears below has fixed positioned.

The data displayed in this box has fixed position. It can't move down even if page scrolls down.

The data displayed in this div is not specially positioned. B The div is positioned according to its normal position.

Figure 6.9: Output of code 6.9



## Unit Summary

In this unit, we learned about the use of various text, table and document properties. We have also discussed how to make attractive web pages with the help of CSS.



## Self Assessment Questions

Choose the correct answer

1. What does CSS stand for?
  - A. Computer Style Sheet
  - B. Cascading Style Sheet
  - C. Colorful style Sheet
  - D. Creative Style Sheet
  
2. Which of the following is a component of CSS style rule?
  - A. Selector
  - B. Property
  - C. Value
  - D. All of Above
  
3. Which of the following defines a measurement in centimeters?
  - A. %
  - B. cm
  - C. em
  - D. ex
  
4. Which of the following property is used to create a small-caps effect?
  - A. font-family
  - B. font-style
  - C. font-variant
  - D. font-weight
  
5. The property that specifies the color of a border is :
  - A. border-color
  - B. border-style
  - C. border-width
  - D. border-bottom-color
  
6. The property changes the width of left border is
  - A. border-bottom-width
  - B. border-top-width
  - C. border-left-width
  - D. border-right-width
  
7. \_\_\_\_\_ property is used to set the background image of an element.
  - A. background-color
  - B. background-image
  - C. background-repeat
  - D. background-position



8. Which of the following property is used to increase or decrease the size of a font?
  - A. font-size
  - B. font
  - C. font-variant
  - D. font-weight
  
9. \_\_\_\_\_ property can be used to increase or decrease the space between the words?
  - A. word-spacing
  - B. word-space
  - C. space
  - D. both b and c above
  
10. The \_\_\_\_\_ property is mostly used to remove underline from links?
  - A. text- trans
  - B. text-deco
  - C. text-transformation
  - D. text-decoration
  
11. Which of the following statement is true for CSS?
  - A. An external style sheet can be written in HTML
  - B. An external style sheet is ideal when the style is applied to many web pages.
  - C. An inline style sheet should be used when a single document has a unique style.
  - D. Both b and c
  
12. \_\_\_\_\_ property is used to set opacity of the image?
  - A. height
  - B. border
  - C. opacity
  - D. width
  
13. The font size can be absolute or relative in size.
  - A. True
  - B. False
  
14. Which of the following property specifies the bottom padding of an element?
  - A. padding-bottom
  - B. padding-left
  - C. padding-right
  - D. padding-top

**Answer Key:**

1. B	2. D	3. B	4. C	5. A	6. C	7. B	8. B	9. A	10. D
11. D	12. C	13. B	14. A						



## Review Questions

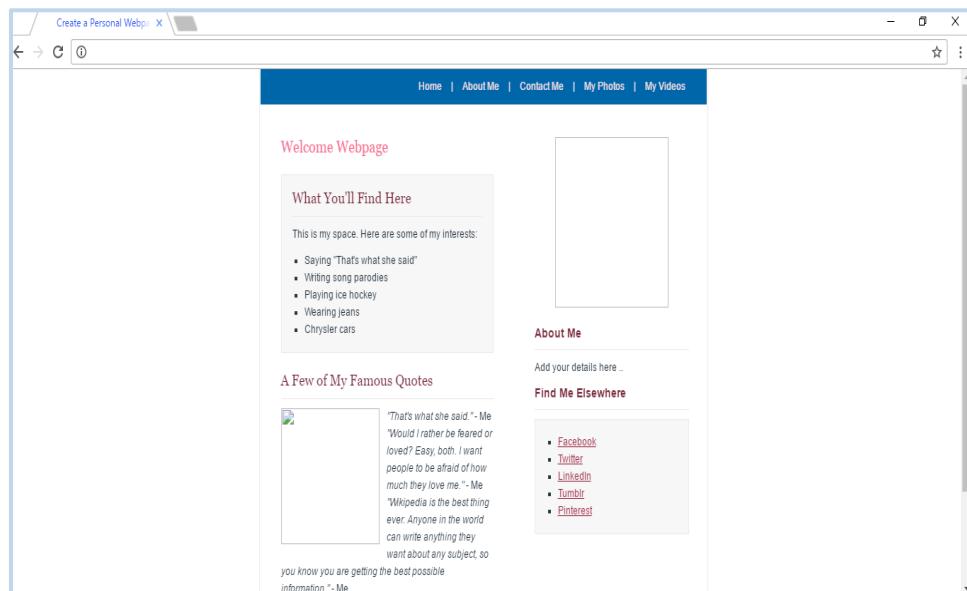
Write short answers of the following questions.

1. Explain types of positioning used in CSS?
2. How border properties are used in CSS?
3. Explain the function of padding property in CSS?

## </> Coding Exercise

1. Build the following resume design with the help of following CSS properties:

- Background
- Margin
- Border
- Headings
- Links
- Lists
- Positions



2. Repeat coding exercise by improving your CSS code by including background, margin, table, border, headings, link and position properties, as appropriate.



## References and Further Reading

1. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
2. Casario, M., Wormser, N., Saltzman, D., Bradford, A., Reid, J., Improta, F., & Congleton, A. (2013). CSS3 Solutions: Essential Techniques for CSS3 Developers. Apress.
3. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World. sitepoint.
4. Casario, M., Wormser, N., Saltzman, D., Bradford, A., Reid, J., Improta, F., & Congleton, A. (2013). CSS3 Solutions: Essential Techniques for CSS3 Developers. Apress.
5. CSS Online Quiz. Available on: [http://www.tutorialspoint.com/css/css\\_online\\_quiz.htm](http://www.tutorialspoint.com/css/css_online_quiz.htm)
6. CSS Syntax. Available on: <http://criss-ryo.blogspot.com/2012/04/css-script.html>
7. CSS Position. Available on: <http://sinahu-coding.blogspot.com/2015/11/mengatur-position-elemen-html.html>
8. CSS Layout. Available on: <http://www.sharptemplates.com/tag/more-examples/>



## UNIT 7

# Client Side Scripting - I

## Introduction

By now, we have learnt how to develop web pages using HTML. We added style into our web pages using CSS to make them more attractive and lively. We noticed that HTML and CSS do not provide any programming capabilities and therefore, the web pages were kind of static in nature. It is time to go a step further and explore how we can make our web pages more interactive and responsive to user input. Javascript is there to help us in this process. JavaScript is one of the programming languages that provide us to embed our programming logic into HTML code to make a web page more integrative and responsive. The javascript code that we embed in HTML document is called a script. This unit and the next unit will equip you with the basics of javascript. The purpose is to familiarize you with what you can establish with it and how you can embed javascript in your HTML code.



## Unit Outcomes

At the completion of this unit, you will be able to:

1. To describe difference between client side and server side scripting.
2. Discuss limitations of javascript.
3. Writing java script in body and head section in an HTML document.
4. Linking external java script with HTML document.
5. Understand basic javascript concepts such as variables and comments.



## Terminologies

**Javascript:** It is a scripting language that adds interactivity to the websites.

**Client side script:** It is a script that runs on the user computer using web browser.

**Server side script:** It is a script that runs on web server.

**Internal javascript:** It allows to place javascript code within the same HTML document.

**External javascript:** It allows to place javascript code in a separate file and link with HTML document.

**Variable:** It is used to store information.



## 7.1. JavaScript

The program code that we embed in an HTML page is called a script. The phrase ‘script’ generally refers to small programs. The set of instructions are either to the (client-side scripting) or to the (server server-side scripting). Traditionally, client-side scripting is used for page navigation, data validation and formatting.



A language in which series of commands are executed without being compiled known as scripting language.

JavaScript is a “platform independent, interpreted, object-oriented scripting language”. JavaScript code is embedded in HTML pages which can be interpreted by the web browser.

The client-side script runs in a browser. The code is transferred to the user’s computer from server over the Internet and runs directly in the browser. A server side script, on either hand, runs on the server which hosts web pages. Script directly runs on the web server to generate dynamic HTML web pages to fulfill the requirement of end user. These dynamically generated HTML pages are sent to the client browser.

As we mentioned before that a client-side script runs on the user computer and not on the web server. What really happens is as follows:

- A user requests a Web page from the server by either entering its URL or clicking on a link.
- After receiving the page request, web server sends the requested page to the user on the Internet.
- The page is displayed in the browser. If the web page has some script in it, the browser runs it.



Client-side script is a program embedded in a web page that executes on client-side by the user's web browser.

On the other hand, if a page has a server-side script, what really happens is as follows

- The user requests a web page from the server by either entering its URL or clicking on a link.
- If the web page has some script in it, it is interpreted by the server to create or change the content of the page.
- Finally, the web page in its final form is sent to the end user which is then displayed by a browser.



Server-side script is a program that executes on the server to generate dynamic web pages.

### Activity 1

Do a web search and prepare a list of languages that are used for client-side and server-side scripting.

### 7.1.1. Advantages and limitations of JavaScript

Let us have a quick look at some of advantages and disadvantages of using JavaScript. Some of the advantages of JavaScript are:

- JavaScript is executed on the client side to save bandwidth and avoid load on the web server.
- JavaScript is relatively easy to learn and comprises of syntax that is close to C/C++ and Java.
- It provides plenty of prewritten functionality.
- JavaScript plays nicely with other languages and can be used in a huge variety of applications.



JavaScript is executed on the client side, easy to learn, faster, works well with other languages.

On the negative side, JavaScript

- has some security problem as the code execute on the users' computer, and exploited for some malicious activities.
- does not have multithreading abilities.
- cannot access web pages hosted on different areas.
- cannot directly write to a file on the server side without sending an AJAX request that runs a server side script.



JavaScript has security issues, does not support multithreading, cannot access files and databases.

## 7.2. First JavaScript Program

It is time to see JavaScript in action. Follow the steps given below to write our first JavaScript code in an HTML document.

**Step 1: Open Notepad in Window**

**Step 2: Type your HTML and JavaScript code** (given in the figure below) in Notepad.

**</> Code 7.1**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - Client Side Scripting</title>
</head>
<body>
<p>My First JavaScript Program</p><hr>
<script type="text/JavaScript">
    document.write("JavaScript is a simple language ");
</script>
<h2>IT WORKED! </h2>
```



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

**Step 3: Save your document** on your computer by selecting File > Save As (go to the menu bar and click File and then click on Save As). Select the folder in which you want to save the file, enter file name (such as new.html) and set the encoding to UTF-8 and click on Save button. Note that your file name should have extension .html.

**Step 4: View the HTML document.** By double clicking on the HTML file or right-click and choose "Open with" and select browser to see the output of HTML page.

The output of above code is shown below.



**Figure 7.1:** Output of first code 7.1.

Understanding the code is quite easy as far as the HTML tags are concerned. Let us go through the code that is new for us.

1. `<script type="text/javascript">` tag is used to specify that we are going to write JavaScript. It has an ending tag `</script>`. In HTML5, Type attribute is not necessary for javascript, default value is "text/javascript".
2. The function `document.write()` is used to display dynamic content.

## Activity 2

Write a web page that displays your name and address (use JavaScript method `document.write`). Display "MY BIO" on the top and at the bottom of the page using HTML tags.

### 7.2.1. Where to put JavaScript Code?

In our first example of JavaScript, the script was in the body section of the HTML document. It is however not necessary to always put JavaScript code in the body of HTML document. HTML allows us to put our script in head section or even in a separate file (having extension .js). Before, we see how we can add script in the body and head section, we need to know a bit about **events**.



JavaScript can be written in head and body section of a document. It can also be in a separate .js file.

Events basically happen as a result of a user action (e.g. clicking a button, pressing a key etc.) or an external action such as loading a page. HTML provides a way to bind a script to an event. That is, the script will execute in response to that event. Binding of script to an event is done through an event handler attributes. It is the name of the event which is prefixed by "on", for example **onclick**.



JavaScript can be bound to events. Javascript code executes when a specific event happens.

Now let us see how to add JavaScript in an HTML document.

### 7.2.2. JavaScript in Body Section

We can write JavaScript code anywhere in the body section. A document can have one or more scripts depending on what and when we want to do something through JavaScript. If a JavaScript is not bound to any event, it will always execute when we load the page. The script that is bound to an event executes only when the specified event happens.



JavaScript is written inside <script> and </script> tags.

Let us say we first want to display “Script demo” when a page is loaded. This can be done by writing the following code

```
<script type="text/javascript">
    document.write ("Script demo");
</script>
```

Then we want to display a button (a form element). When the user clicks on the button, we want to display the message ‘You clicked on the button’ as an alert (an alert is displayed in a pop-up window). To do this, we have to bind our script with the event ‘onclick’. We can use the following code to do it

```
<form>
    <input type="button" value="click" onclick="alert ('You clicked
    on the button');">
</form>
```

Let us now put the things together and write our complete code.

**</> Code 7.2**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - Client Side Scripting</title>
```



```
</head>
<body>
    <p>Example: Our first JavaScript</p>
    <hr>
    <script type="text/javascript">
        document.write("Script demo");
    </script>
    <form>
        <input type="button" value="click" onclick="alert('You
            clicked on the button');">
    </form>
</body>
</html>
```

The output of above code is shown below:

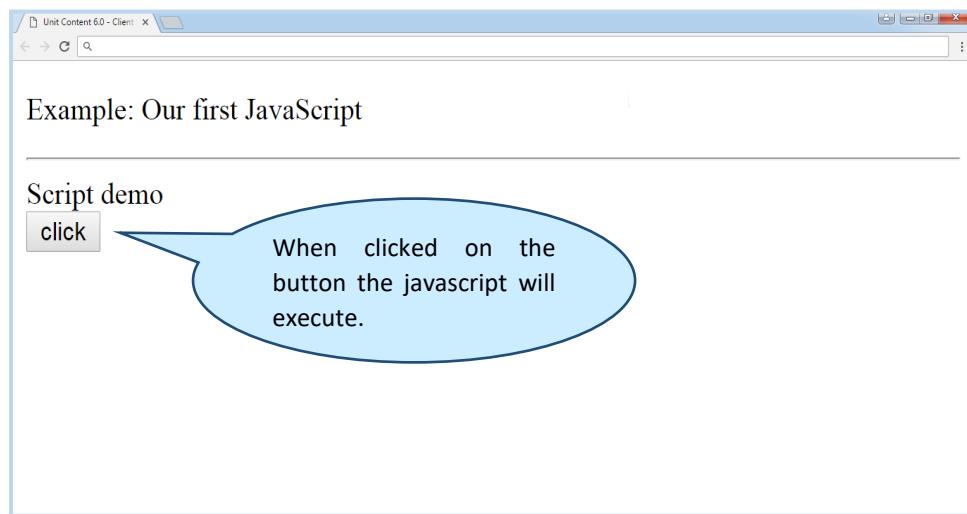


Figure 7.2: Output of code 7.2

Upon clicking on the button '*click*' an alert will be popup as shown in the figure 7.3.

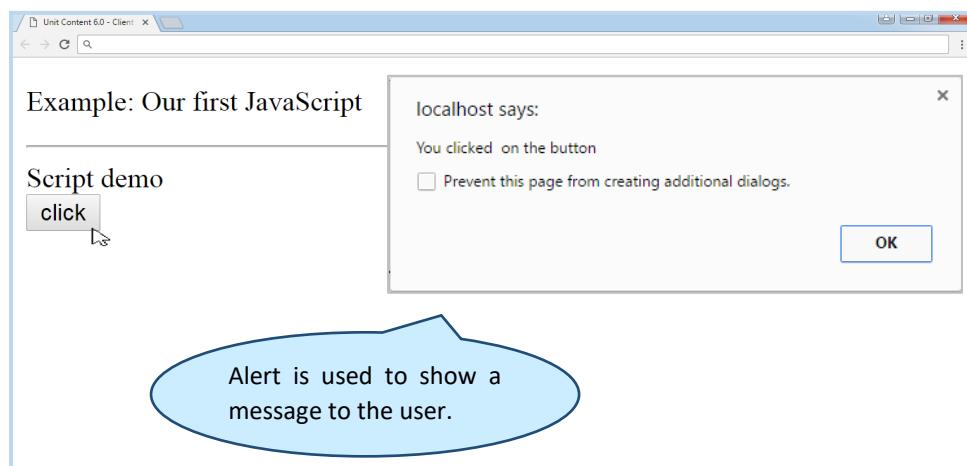


Figure 7.3: Output of javascript popup



## Video Lecture

<https://youtu.be/BFONuBSiHcM>



### Activity 3

Change the alert message to "Hello User" in code 7.2.

#### 7.2.3. JavaScript in Head Section

A special place for the `<script>` element is within the head section of a document. Because of the sequential nature of the web documents, the `<head>` is always executed first. However, we want to execute the script somewhere in the body of the document. Therefore, script written in the head section is generally used to declare variable or functions that may be used later on in the document.

Let us now write the same HTML document that we wrote in our previous example, but this time putting our script in the head section that will execute when the users click on the button. We will write the code as a function (more about function later)

```
<script type="text/javascript">
    function msg ()
    {
        alert ('You clicked on the button');
    }
</script>
```

The function has been given the name `msg`. Now we can call this function in the body of the document when needed. In our case, we want this code to execute when the user clicks on the button. This can be specified using the following code;

```
<form>
    <input type="button" value ="click" onclick="msg();">
</form>
```

Putting the pieces together gives us the following code. Note that we still have some JavaScript in the body as well and it will execute as usual. Try it and see how it works.

**</> Code 7.3**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - Client Side Scripting</title>
<script type="text/javascript">
    function msg()
    {
        alert('You clicked on the button');
    }
</script>
</head>
<body>
    <p>Example: Our first JavaScript</p>
    <hr>
    <script type="text/javascript">
        document.write("Script demo");
    </script>
    <form>
        <br><input type="button" value="click" onclick="msg();">
    </form>
</body>
</html>
```

The output of above code is shown below:

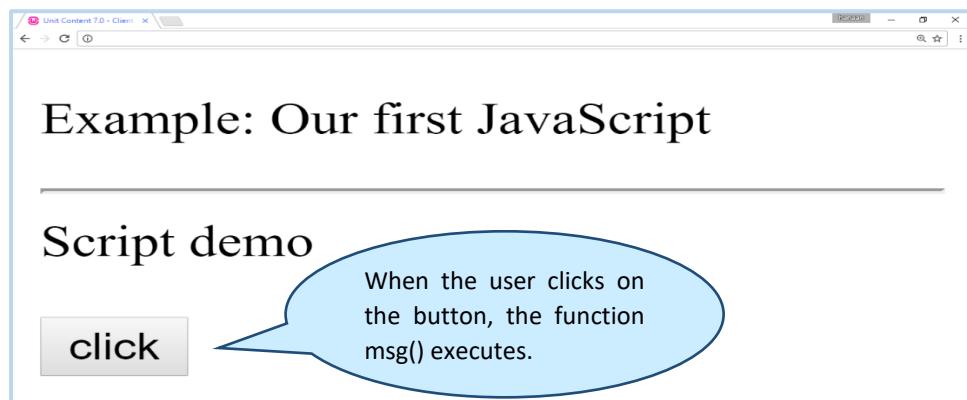


Figure 7.4: Output of code 7.3

Upon clicking on the above button 'click' an alert will be popup as shown in the following figure.

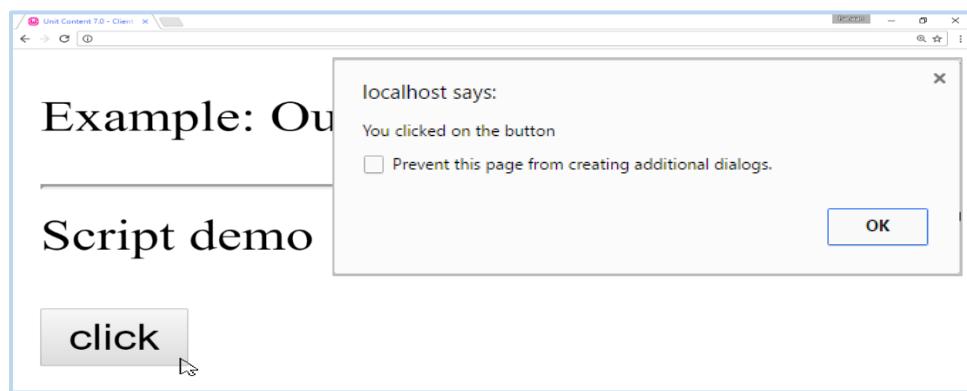


Figure 7.5: Output of javascript button popup



## Activity 4

Repeat the activity 3 but write the event handler in the head section of the document.

### 7.3. External JavaScript

In external javascript, the code is saved in a separate file which is then linked to an HTML document using **src** attribute of <script> tag. Let us see how we develop our web page using this approach.

- First of all, we should write the JavaScript code using a text editor, save it in a file that has the extension .js. We type and save the following code in *message.js* file

```
function msg()
{
    alert ('You clicked on the button');
}
```

- Now add a link to this code in the HTML document using the following code

```
<script src="message.js"></script>
```

Below is complete HTML code that we should now type.

#### </> Code 7.4

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - Client Side Scripting</title>
<script src="message.js"></script>
</head>
<body>
<p>Example: Our first JavaScript</p>
<hr>
<script type="text/javascript">
    document.write("Script demo");
</script>
<form>
    <input type="button" value="click" onclick="msg();">
</form>
</body>
</html>
```

The output of above code is given below



Figure 7.6: Output of code 7.4

When a user clicks on the button, the javascript executes and displays the alert as shown in the following figure.

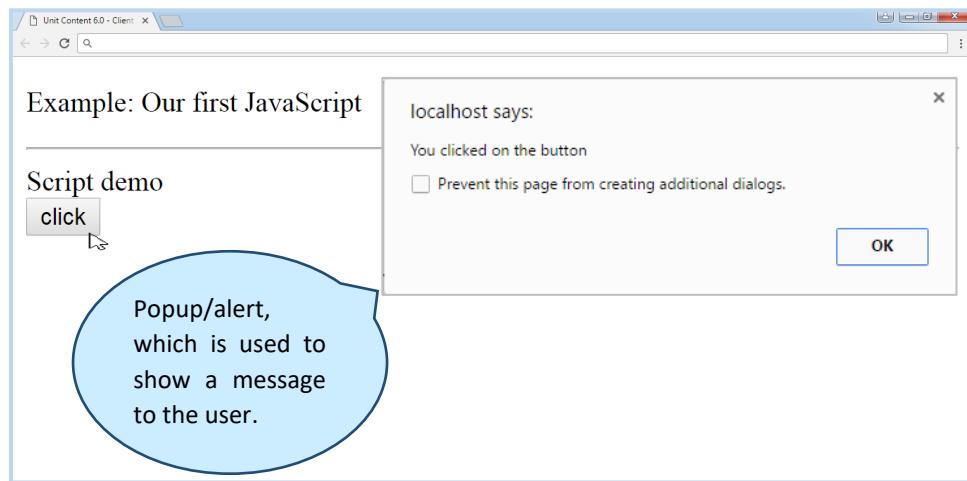


Figure 7.7: The Output when user clicks on button.



## Video Lecture

[https://youtu.be/\\_aaPbUXpSnY](https://youtu.be/_aaPbUXpSnY)



## Activity 5

Repeat activity 4 by writing a code in a separate file and linking it to HTML document.

### 7.4. JavaScript Basics

The core features of JavaScript are the syntax rules to which our script should adhere to and the basic constructs to store, process and manipulate flow control. We will discuss only the very basics features leaving the advanced features and constructs for an advanced course on JavaScript. Before we start with writing JavaScript code, keep the following important points about JavaScript in your mind.

- Java script is case-sensitive. This means that capital letters are distinct from their lowercase counterparts. For JavaScript, **this**, **This** and **THIS** are three distinct identifiers.
- JavaScript code found in HTML document is interpreted line by line and it is read from top to bottom.
- Any use of excessive whitespace characters (e.g. spaces, tabs, line breaks) are ignored by JavaScript. For example,

`x=5;`

Is same as

`x = 5;`

- JavaScript is written as a sequence of *statements*. A statement is an instruction to the interpreter to carry out a specific action.
- A semicolon (;) indicates end of a statement. This allows us to write one or more statements on the same line separated by ‘;’.

For example,

`x=5; y=4;`

Is same as

`x=5;`

`y=4;`



JavaScript is a case-sensitive language.

### 7.5. JavaScript Variables

A variable is used to store data. Every variable has a name, called its *identifier*. Variables are containers that hold information and then refer to the data simply by using the name of the container. Their sole purpose is to label and store data in memory that can be used throughout your program.

The general rules for constructing names for variables (unique identifiers) are:

1. Names can contain letters, digits, underscores, and dollar signs.
2. Names must begin with a letter



3. Reserved words (like JavaScript keywords) cannot be used as names.

For example, the following are valid variable names

area, go2there all\_done

The following variable names are not valid

9abc p&tp hello-there

### 7.5.1. Declaring variables

We must declare a variable in a JavaScript program before its use. Variables are declared with the *var* keyword as follows.

```
var x;
```

We can assign a value to a variable when declaring it as

```
var x=5;
```

In addition, multiple variables can be declared with one *var* statement but we have to separate each variable by a comma as shown below.

```
var x, y, z=15;
```

Try the following code and see what it does.

#### </> Code 7.5

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - JavaScript Detail</title>
</head>
<body>
<p>Example: JavaScript Variables</p>
<script type="text/javascript">
var math, eng = 200;
var computer = 100;
var math = 300;
var total = math + eng + computer;
document.write(total);
</script>
</body>
</html>
```

The output of above code is shown in figure 7.4.

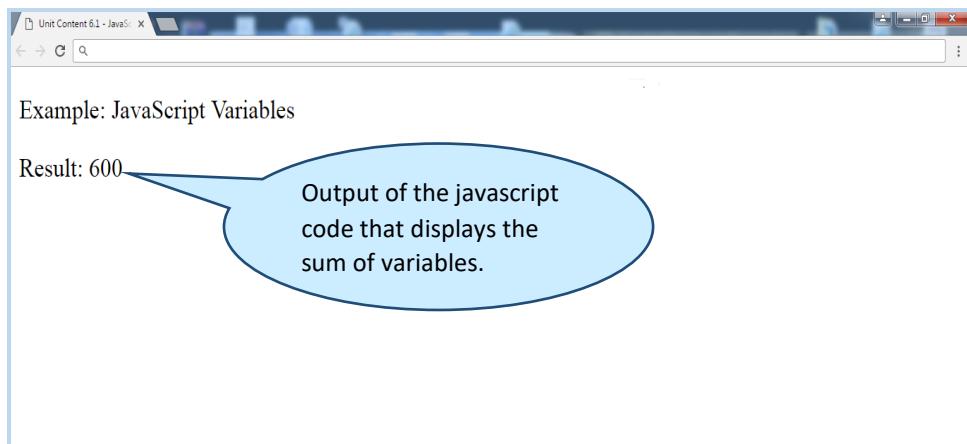


Figure 7.8: Output of code 7.5

## Activity 6

Change the javascript code 7.5 to display average of all the variables.

## 7.6. JavaScript Comments

Writing comments is an important way to document a program. Its purpose is to add information about the code so that the user can easily know the functionality and can easily interpret the code.

Types of comments used in JavaScript are:

- Single line comment
- Multi line comment



Comments are helpful in understanding the code.

### 7.6.1. Single line Comment

A single line comment starts with double slash like (//). It can either be used before or after a statement or statements as the following code demonstrates.

#### </> Code 7.6

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - JavaScript </title>
</head>
<body>
<p>Example: JavaScript Comments</p>

<script type="text/javascript">
 // Declaring variables
 var math = 300;
 var eng = 400;
 var total = math + eng; // it adds up two numbers
 document.write(total);
</script>
```



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

The output of above code is shown in figure 7.9.

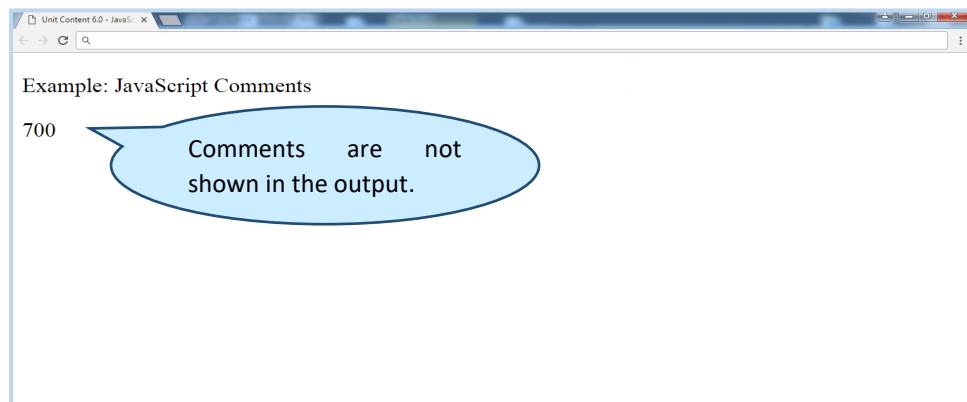


Figure 7.9: Output of code 7.6

### 7.6.2. Multi-line Comment

If we have to put a comment on more than one line, we can use multi-line comments. A multi-line comment starts with a forward slash and an asterisk and ends with an asterisk forwarded by a forward slash as shown below

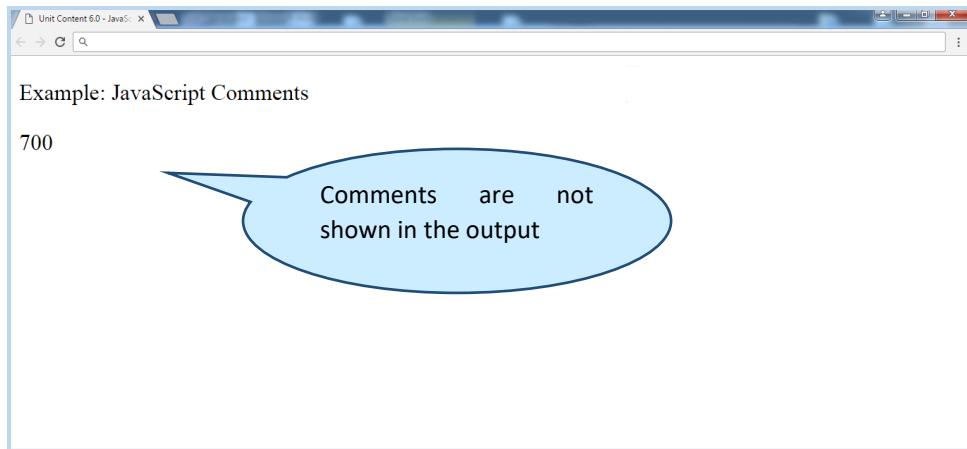
```
/* it is a multi line comment and
it will not be displayed */
```

Here is a sample code with multi-line comment.

#### </> Code 7.7

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 7.0 - JavaScript </title>
</head>
<body>
<p>Example: JavaScript Comments</p>
<script type="text/javascript">
    var math = 300;
    var eng = 400;
    var total = math + eng;
    /* it is a multi-line comment and
    It will not be displayed */
    document.write(total);
</script>
</body>
</html>
```

The output of above code is shown in figure 7.10.



**Figure 7.10:** Output of code 7.7.

## **Activity 7**

Add some suitable single and multiline comments in code 7.7.



## **Unit Summary**

In this unit, we learned about the concept of client side and server side scripting and basics of javascript as client side scripting language. We also learned how to write basic code of javascript in body and head section in an HTML document. We have also discussed the syntax of variables and comments in javascript.



## Self Assessment Questions

Select the correct answer.

1. \_\_\_\_\_ tag is an extension to HTML that can enclose any number of JavaScript statements.
  - A. <SCRIPT>
  - B. <BODY>
  - C. <HEAD>
  - D. <TITLE>
  
2. Select the correct way of writing comments in javascript
  - A. //, /\* ..... \*/
  - B. /, /\*\* ...../, \*/
  - C. /\*.....\*/ , //
  - D. \\*.....\*\, //
  
3. What is the correct javascript code to display "Hello World"?
  - A. system.out.println("Hello World")
  - B. println ("Hello World")
  - C. document.write("Hello World")
  - D. response.write("Hello World")
  
4. \_\_\_\_\_ is the correct code for referring to an external script called " abc.js".
  - A. <script href=" abc.js">
  - B. <script name=" abc.js">
  - C. <script src=" abc.js">
  - D. None of the above
  
5. What are variables used for in javascript programs?
  - A. Storing numbers, dates, or other values
  - B. Varying randomly
  - C. Causing high-school algebra flashbacks
  - D. None of the above
  
6. Which of the following tag is used to indicate the end of javascript code?
  - A. </script>
  - B. <script>
  - C. <END>
  - D. None of the above
  
7. A client side javascript code is interpreted by \_\_\_\_\_
  - A. Client
  - B. Server
  - C. Object
  - D. None of the above
  
8. Each javascript variable must be declared in a separate line?
  - A. True
  - B. False

9. What will be the output of following javascript code  
`<script language="javascript"> document.write (2+5+"8"); </script>`
- A. 258
  - B. Error
  - C. 7
  - D. 78
10. Javascript is \_\_\_\_\_
- A. compiled language
  - B. Interpreted language
  - C. compiled and interpreted language
  - D. none

**Answer Key:**

1. A	2. C	3. C	4. C	5. A	6. A	7. A	8. B	9. D	10. B
11. D	12. C	13. B	14. A						

**Review Questions**

**Write short answers of the following.**

1. What is a script?
2. Differentiate between client-side and server-side scripting?
3. Inlist the advantages and disadvantages of using javascript in your web pages?
4. How is javascript case sensitive?
5. Name some of the javascript features.
6. What are the valid scopes of a variable in javascript?
7. What data types are available in javascript?
8. How do we write single line and multi-line comments?
9. How do you link a JavaScript to a HTML document?
10. In which parts of HTML document can a javascript code be written?

**</> Coding Exercise**

1. Write an HTML code with required javascript that displays sum, product and average of two numbers.
2. Write an HTML code with required javascript with three buttons: Sum, Product and Avg. Your code should display sum, product or average depending on which button is pressed by the user.
3. Repeat the above exercise by writing all your javascript code in a separate file and linking it to the HTML document.



## References and Further Reading

1. Haverbeke, M. (2014). Eloquent javascript: A modern introduction to programming. No Starch Press.
  2. Elliott, E. (2014). Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. O'Reilly Media, Inc.
  3. Resig, J., & Bibeault, B. (2013). Secrets of the JavaScript Ninja. Manning.
  4. Rauschmayer, A. (2014). Speaking JavaScript: An In-Depth Guide for Programmers. O'Reilly Media, Inc.
  5. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH.
  6. Javascript Questions. Available on: <http://www.slideshare.net/stuartsandeep/top-javascript>
  7. Javascript Tutorial. Available on:  
[http://www.tutorialspoint.com/javascript/javascript\\_tutorial.pdf](http://www.tutorialspoint.com/javascript/javascript_tutorial.pdf)
-





## UNIT 8

# Client Side Scripting - II

## Introduction

We introduced in the previous unit that how javascript can be embedded in HTML code to make a web page more interactive and responsive. We also discussed some basic features of JavaScript and some of its basic constructs. In this unit, we continue our discussion on JavaScript and presents some of its core constructs such as expressions, assignment statements, selection statements, loops and manipulation of form elements.



## Unit Outcomes

Upon completion of this unit, you will be able to:

1. Write JavaScript programs with selection statements.
2. Write JavaScript programs with loops.
3. Write JavaScript programs to manipulate form elements.
4. Develop interactive and responsive web pages by using JavaScript



## Terminologies

**Operator:** It is a character symbol or sign which is used to perform some mathematical calculations or carry out logical functions.

**Logical operators:** They are used to combine and compare Boolean expressions

**Comparison operators:** They are used to compare two or more expressions.

**Conditional Statement:** It is a set of statements executed when the condition is true.

**Loop:** It is a set of instructions that are continuously repeated until condition is true.

**Object:** An object is a collection of variables and related methods.

**Event:** An event is a result of user action.



## 8.1. JavaScript Operators

In a programming language, an operator can be a mathematical character, symbol or a sign which is used to do mathematical calculation or to carry out some logical functions. JavaScript operators are used to perform different operations on the operands. Operands can be constant values, variables and constant values. Let us look at some of the basic JavaScript operators.



Operators are used to perform different operations.

### 8.1.1. Arithmetic Operators and Expressions

Arithmetic operators are used to perform arithmetic operation on the operands. The following table highlights arithmetic operators and presents examples of each assuming the following declarations:

```
var x=3, y=2;  
var sum, diff, prod, rem;
```

**Table 8.1:** Arithmetic operators

Operators	Meaning	Example
+	Addition	sum=x+y; //sum= 12
-	Subtraction	diff=x-y; // diff=1
*	Multiplication	prod=x*y; // prod=6
/		div1=x/y ; //div1=1.5
++	Increment	x++; //x=4
--	Decrement	y--; //y=1
%	Modulus (remainder)	rem=x/y; //rem=1



The operators which perform some arithmetic operations like addition, subtraction etc. are known as arithmetic operators.

Arithmetic expressions can be constructed by combining arithmetic operators and operands. Some examples of arithmetic are:

```
2 + 4 * 5  
2 + x * y - 6/2
```

When an expression has more than one operator in an expression, JavaScript uses operator precedence rules to evaluate the expression. According to the precedence rules, \*, / and % have higher precedence than + and - (that is \*, / % are evaluated before + and -). If an expression has operators at the same level of precedence, evaluation is done from left-to-right. For example, the expression  $5+6/3$  will be evaluated as follows. First, we will divide 6 by 3 (since / has higher precedence than +) which give 2; adding 2 to 5 will give us the result 7. Note that the precedence can be changed by using ( ) which is evaluated first.



If an expression has several operators, precedence rules are used to evaluate the expression. Operators with higher precedence are evaluated first.



## Activity 1

- Write JavaScript code that prints sum, difference, product and average of two numbers.
- Write java script that calculates the area of a triangle.
- What will be the value of expression  $6 + 2 * 6 / 3 - 4$

### 8.1.2. Comparison Operators

Comparison operators are used to compare two operands and give a Boolean (True, False) result. These operators are summarized, with examples, in the following table.

**Table 8.2:** Comparison operators

Comparison Operator	Meaning	Example	Result
>	Greater than	$40 > 10$	true
$\geq$	Greater than or Equal	$40 \geq 50$	false
<	Less than	$60 < 50$	false
$\leq$	Less than or equal to	$10 \leq 20$	true
$=$	Equal to	$40 == 10$	false
$==$	Strictly equal to (and have the same type)	$5 === '5'$ $5 === 5$	False false



Comparison operator gives Boolean value either true or false

### 8.1.3. Logical Operators

As discussed in the previous section, comparison operators give Boolean (true/false) results. The logical operators are used for combining such values together in order to express complicated conditions/logic. The following table gives explanations and examples of JavaScript logical operators.

**Table 8.3:**Logical operators

Operator	Meaning	Explanation	Example
$\&\&$	Logical AND	If both expressions are true, it will return the true answer	$(10 == 20 \&\& 20 == 33) = \text{false}$
$\ $	Logical OR	If any or both expressions are true, it will return the true answer	$(10 == 20 \  20 == 33) = \text{false}$
!	Logical NOT	Performs logical negation on an expression.	$! (10 == 20) = \text{true}$



Logical operators are used to make complex expressions.



### 8.1.4. Assignment Operator and Combining Assignment with Arithmetic Operators

An assignment operator ( = ) can be used to assign a value or value of an expression to a variable. An assignment statement has the following form:

Variable=expression;

For example, to assign 5 to variable x, we write

x = 5;

We can assign value of an expression to a variable x, as

x = 4 \* 5 / x -7;

JavaScript also has shorthand assignment operators that combine an arithmetic operator with assignment operator. These shorthand forms let us express common assignment statements precisely but are otherwise equivalent to their expanded form. The following table summarizes these operators.

**Table 8.4:** Assignment with arithmetic operators

Shorthand Assignment	Description	Example
+ =	It adds the value of a variable to an expression and then assigns the result to the variable.	var p=10; p+=20; // p = 30
- =	It subtracts the value of an expression from the value of a variable and then assigns the result to the variable.	var p=20; p-=10; // p = 10
* =	First, it will multiply the value of an expression with the value of a variable and then assigns the result to a variable.	var p=10; p*=20; //p = 200
/ =	First, it will divide the value of an expression with the value of a variable and then assigns the result to a variable.	var p=10; p /=5; //p = 2



The operator which is used to assign a value to variable is known as assignment operator.



## Video Lecture

<https://youtu.be/D9zr7kNsx4>



### Activity 2

Write the following using shorthand assignment.

1.  $i = i + 7;$
2.  $p = p / x$
3.  $g = g * (x + y)$



## 8.2. JavaScript Conditional Statements

In computer programming, when we have to take any decision, we use conditional statements. Let us look at JavaScript conditional statements one by one.



Statements are executed one by one in an orderly manner.

### 8.2.1. if Statement

The syntax of basic *if* statement is

```
if (Condition)  
    statement;
```

In the *if* statement, if the condition evaluates to true then the statement is executed. For example, the following code displays “Congratulation...” message if marks are 90 or more. Otherwise, the message will not be displayed.

```
marks = 95;  
if (marks >= 90)  
    alert ("Congratulation..");
```



If statement is used to perform some action when a condition is true.



If we want to execute more than one statements when a condition is true, then we have to put all those statements in a block (inside { }) as shown in the following code.

```
marks = 95;
if (marks >= 90)
{
    alert ("Congratulation..");
    alert ("You got A Grade");
}
```

Below is a complete HTML document with a small JavaScript code in it.

#### </> Code 8.1

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result:
    <script type="text/javascript">
        var marks = 70;
        if (marks>60)
            document.write("Pass");
    </script>
</body>
</html>
```

Output of above code is shown in figure 8.1.

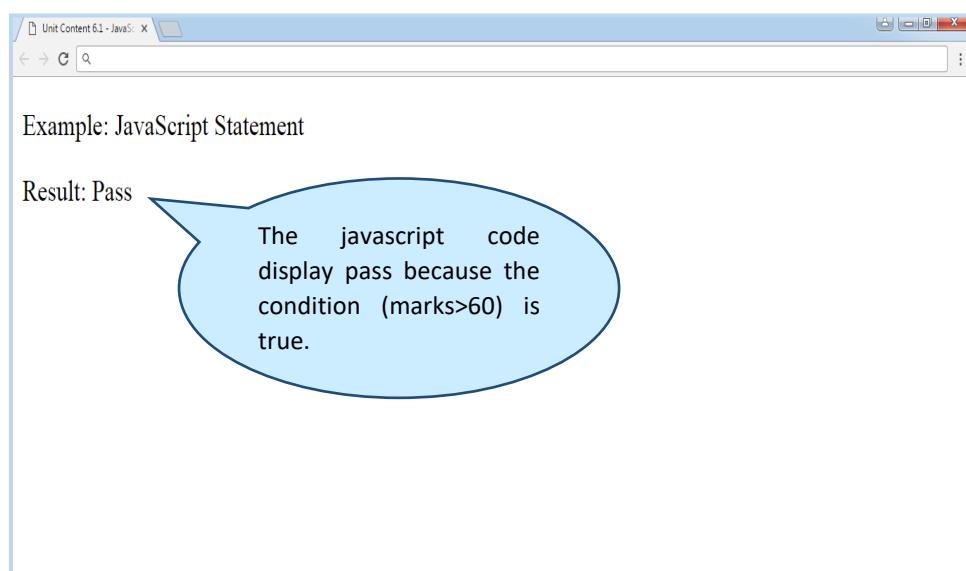


Figure 8.1: Output of code 8.1

## Activity 3

Write JavaScript that prints a message if you fail a course. That is "you obtained less than 50 marks"

### 8.2.2. If-else Statement

There are many programming situations in which we choose between two alternatives. To choose between alternatives, JavaScript provides if-else statement which has the following syntax:

```
if(Condition)
    statement1;
else
    statement2;
```



If else statement can be used to select between two alternatives.

If the condition is true, then the statement1 is executed otherwise statement2 is executed. If we want to execute multiple statements when given condition is true or false, we will make a block of the statement by writing them in { }. The following code segment prints whether a number is even or odd using if-else statement.

```
var n = 10;
if (n%2==0)
    alert("It is even");
else
    alert("It is odd");
```

Below is a complete HTML document with a small JavaScript code in it.

#### </> Code 8.2

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result:
    <script type="text/javascript">
        var marks = 40;
        if (marks>60)
            document.write("Pass");
        else
            document.write("Fail");
    </script>
</body>
</html>
```

Output of above code is shown in figure 8.2.

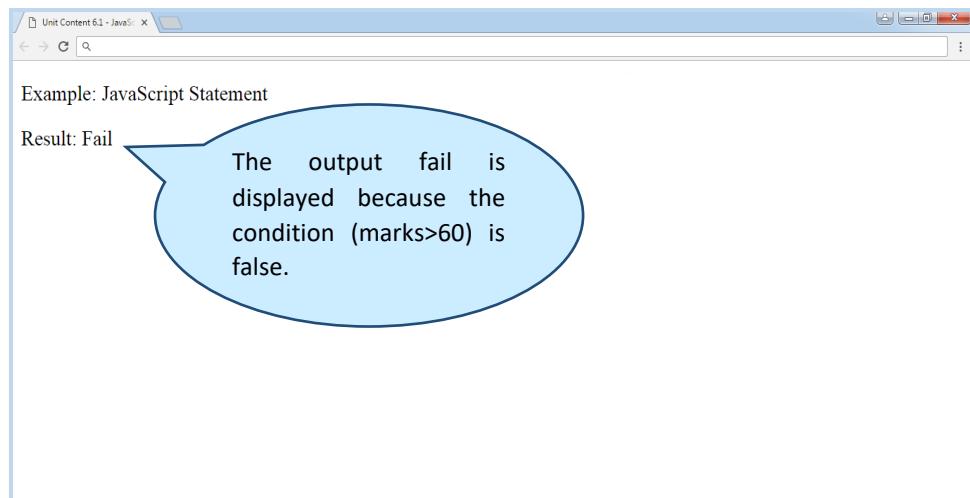


Figure 8.2: Output of code 8.2

## Activity 4

Write JavaScript code that checks whether a number is positive or negative.

### 8.2.3. if-else-if Statement

We can build more complex logic by using if-else-if statement. The general syntax is

```
if (Condition1)
    statement1

else if (Condition2)
    statement2

else if (Condition3)
    statement3
    ...
else
    //statement to be evaluated if no condition is true
```



If-else-if is a series of if conditional statements. If the condition is true then statements are executed otherwise else block is executed.

This following example shows how if statements might be chained together.

```
if (n < 0 )
    alert ("-ve number");

else if (x==0)
    alert("It is zero");

else
    alert("+ve number");
```

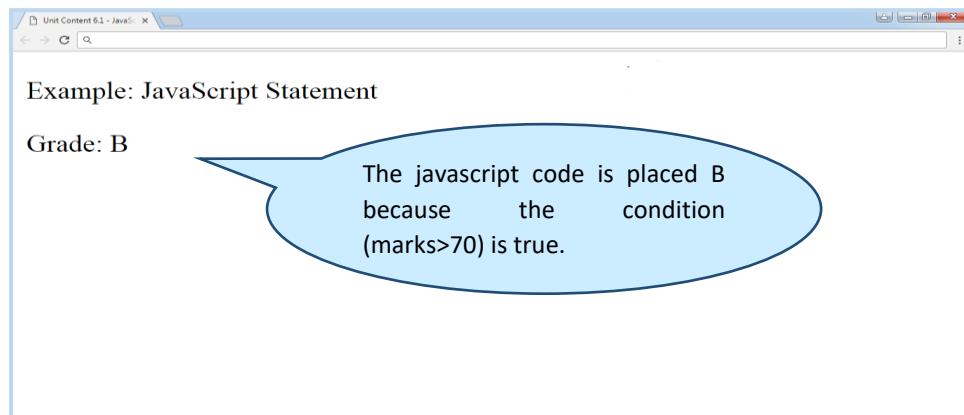
Following is a complete HTML code with javascript that prints grade of a student based on the marks he obtained.

**</> Code 8.3**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8- JavaScript Detail</title>
</head>
<body>
<p>Example: JavaScript Statement</p>
Result:
<script type="text/javascript">
var marks = 75;
if (marks >90)
    document.write("A+");

else if (marks > 80)
    document.write("A");
else if (marks > 70)
    document.write("B");
else if (marks > 60)
    document.write("C");
else
    document.write("F");
</script>
</body>
</html>
```

Output of above code is shown in figure 8.3.



**Figure 8.3:** Output of code 8.3



## Video Lecture

<https://youtu.be/tI-OXBbLKGg>



### Activity 5

To find position of an employee based on his salary. Write a code in which you have to find a senior programmer's post through his salary. Manager's salary is 100,000, assistant manager's salary is 80,000, senior programmer's salary is 60,000 and junior programmer salary is 40,000.

#### 8.2.4. switch Statement

There is another conditional statement which is called 'switch statement'. The syntax of switch statement is shown below:

```
switch (expression)
{
    case value1: statements_1
        break;

    case value2: statements_2
        break;
    ...
    default: statement_def
}
```

The switch statements execute as follow:

1. The expression is evaluated first.
2. The value of the expression is matched with each case value (i.e. with value1, value 2 etc.). If a match is found, then the statements written after the matching value are executed until a break statement is found or the end of switch structure is reached.
3. If there is no matching value found, then the statements after the default label execute.

The break statement is a reserved word that indicates the end of that particular case.



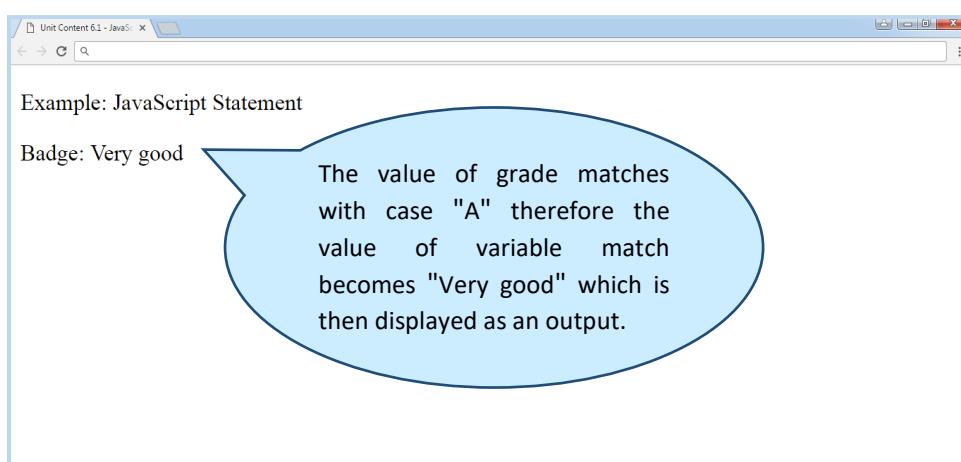
When break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.

The following code shows how we use a switch statement.

**</> Code 8.4**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8- JavaScript Detail</title>
</head>
<body>
<p>Example: JavaScript Statement</p>
Result:
<script type="text/javascript">
    var badge = "Not decided";
    var grade = "A";
    switch (grade)
    {
        case "A+": badge = "Excellent"; break;
        case "A": badge = "Very good"; break;
        case "B": badge = "Good"; break;
        case "C": badge = "Fair"; break;
        case "D": badge = "Poor"; break;
        case "F": badge = "Failed"; break;
        default: badge = "Unknown grade";
    }
    document.write(badge);
</script>
</body>
</html>
```

Output of above code is shown in figure 8.4.



**Figure 8.4:** Output of code 8.4



## Video Lecture

<https://youtu.be/hcuRnVGRrl0>



### Activity 6

Write a switch statement that displays the name of the month given the month's number i.e. If the month number is one then January is displayed, if the month number is two February is displayed and so on. For an incorrect value of month number, the program should display "Incorrect month number".



## 8.3. JavaScript Loops

Loops are used to repeat some tasks for a number of times or until a certain condition remains true/false. JavaScript provide a number of loop construct that we now look at.



Loops are used to repeat a statement or a block of statements.

### 8.3.1. for loop

A *for* loop is the simplest loop construct that is used to repeat a statement or a block of statements for number of times. The syntax of a *for* statement is as follows:

```
for (initial expression; condition; increment expression)
    statement or statement-block
```

A for loop executes as follows: First the initial expression is evaluated and then the condition is checked. If the condition is true, then the statement or block of statements (called the body of the loop) is executed. After completing the body of the loop, the increment expression is evaluated and the loop continues executing until condition becomes false.



for loop statement executes statement or block of statements for number of times.

Let us look at an example

```
for (var i = 1; i<= 5; i++)
    document.write("Iteration# " +i+ "<br>");
```

The above loop executes as follows:

1. Initial expression (var i=0), declares variable i and assigns 0 as its value.
2. Condition ( i<= 5) is checked. If the condition evaluate to true,
  - i. Statement (document.write("Iteration# "+i+"br")) is executed
  - ii. The increment expression (i++) is executed.
3. Repeat step 2 until the condition becomes false.

The javascript in the following HTML document prints first 5 positive integers as can be seen in the output given after the code.

#### </> Code 8.5

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result:
    <script>
        for (var i = 1; i<= 5; i++)
            document.write("i = " + i + "<br>");
    </script>
</body>
</html>
```

Output of above code is shown in figure 8.5.

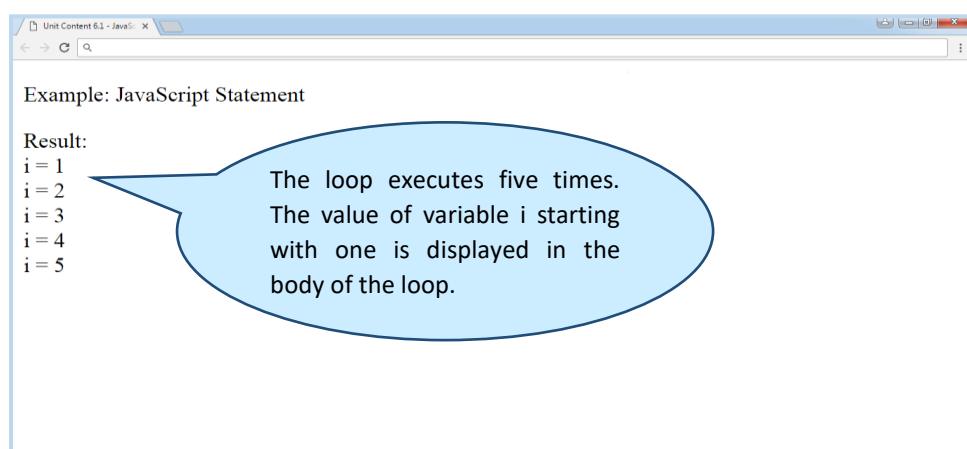


Figure 8.5: Output of code 8.5



## Video Lecture

<https://youtu.be/2cYexKjbbFU>



### Activity 7

1. Modify the code 8.5 to print even numbers from 1 to 20.
2. Modify the code 8.5 to print odd numbers from 1 to 20.

#### 8.3.2. while loop

A *while* loop, similar to *for* loop, executes a statement or block of statements until a condition remains true. If the condition is false, then the control will pass to the statement following the loop construct. The general form of a while loop is:

```
while (condition)
    statement or statement-block
```

Let us see how a while loop executes with the help of following code

```
var i= 0;
while (i<5)
{
    document.write ("Iteration# "+i+"br");
    i++;
}
```

The above loop executes as following:

1. Before the while loop, variable *i* is declared and given a value 0 (*i*=0).
2. The condition (*i*<5) is evaluated. If it evaluate to true then
  - a. First statement in the loop (`document.write ("Iteration# "+i+"br");`) prints the value of *i*.
  - b. The second statement (*i++*) increases the value of *i* by 1.
3. Repeat step 2 until the condition remains true.

To avoid infinite loops, make sure the condition in a loop eventually becomes false; otherwise, the loop will never get terminated.

The while loop written in following code iterates as long as *i* is less than or equal to fifteen.

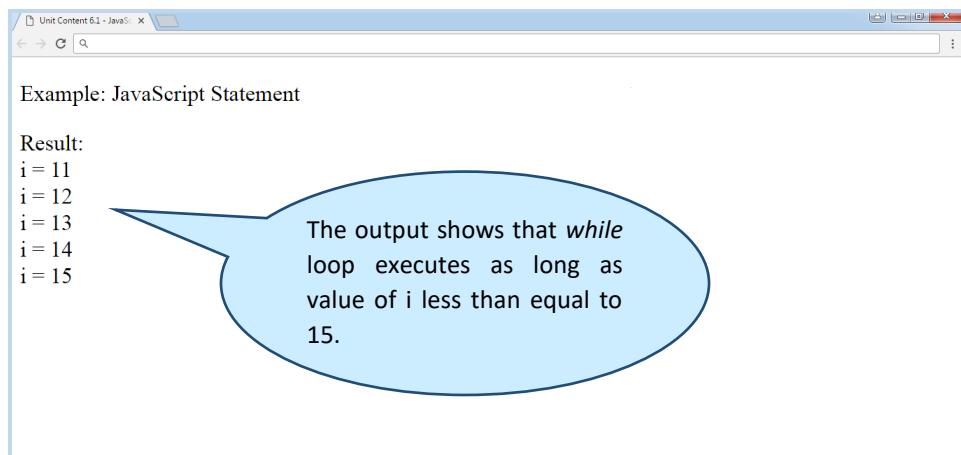
**</> Code 8.6**

```

<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result:
    <script>
        var i = 11;
        while (i<= 15)
        {
            document.write ("i = " + i + "<br>");
            i++;
        }
    </script>
</body>
</html>

```

Output of above code is shown in figure 8.6.



**Figure 8.6:** Output of code 8.6

### **Activity 8**

1. Modify the code 8.6 to execute the loop for 10 times.
2. Write javascript code to print even numbers from 1 to 20.

#### **8.3.3. do while loop**

The *do...while* loop is similar to while loop except that the condition is checked at the end of the loop. Therefore, in any case, the loop will execute at least once. The general form of a do-while loop is:

```

do
{
    Statement(s)

```



```
} while (condition);
```

In a do-while loop, the statement or statements are executed first and then the condition is evaluated. If condition is true, the statement(s) executes again. When the condition becomes false, execution of the loop statement stops and control passes to the statement following the *do.... while* statement.



do while loop executes atleast once.

The following code prints integers from 0 to 9.

```
var i = 0;
do

{
    document.write ("Iteration# "+i+"<br>");
    i++;
} while (i<10);
```

Below is a complete example program that prints integers from 21 to 25.

**</> Code 8.7**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result:
    <br>
    <script>
        var i = 21;
        do
        {
            document.write ("i = " + i + "<br>");
            i++;
        }while (i<= 25);
    </script>
</body>
</html>
```

The output of above code is shown in figure 8.7.

Example: JavaScript Statement

Result:

```
i = 21
i = 22
i = 23
i = 24
i = 25
```

The output shows that *do while* loop executes as long as value of i which initializes from 21 and less than or equals to 25.

Figure 8.7: Output of code 8.7



## Video Lecture

<https://youtu.be/8dube0Veurs>



### Activity 9

1. Modify the code 8.7 to execute the loop for 10 times using do while loop.
2. Modify the code 8.7 to print even numbers from 1 to 20 using do while loop.

#### 8.3.4. break Statement

The break statement is used to exit from a loop early and the program continues executing with the first statement after the loop. The example below illustrates the use of break statement in a while loop. Note that the loop breaks early when the sum become 20 or more.

```
var x=1;
var sum=0;
while (x<=10)
{
    Sum+=x;
    document.write ("The Sum= "+sum+"<br>");
```



```
if (sum>=20) break;  
    x=x+1;  
}  
document.write ("Loop Done.");
```



The break statement ends execution of the loop or the conditional statement in which it appears.

The following code illustrates that the inner while loop will end when the value of z becomes equal to 3 and the outer while loop will end when the value of x becomes equal to 3.

#### </> Code 8.8

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Unit Content 8 - JavaScript Detail</title>  
</head>  
<body>  
    <p>Example: JavaScript Statement</p>  
    Result: <br>  
    <script>  
        var x = 0;var z = 0;  
        while (true)  
        {  
            document.write ("Outer loops: " + x + "<br>");  
            x += 1;  
            z = 1;  
            while (true)  
            {  
                document.write ("Inner loops: " + z + "<br>");  
                z += 1;  
                if (z === 3)  
                    break;  
            }  
            if (x === 3)  
                break;  
        }  
    </script>  
    </body>  
</html>
```

The output of above code is shown in figure 8.8.

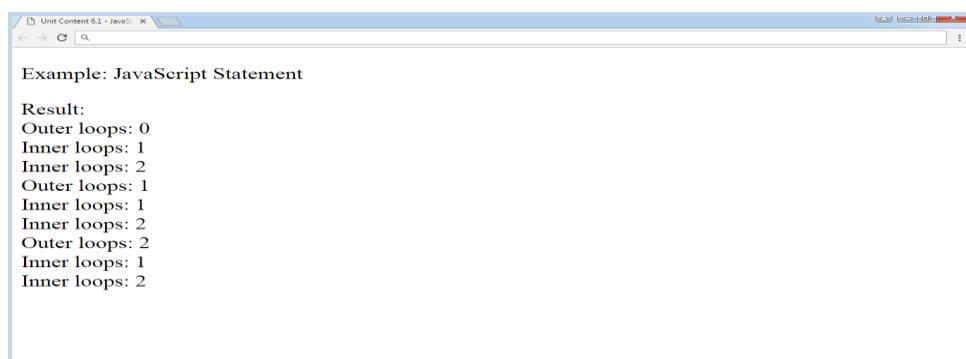


Figure 8.8: Output of code 8.8

## Activity 10

Write a program that calculates the sum of first 20 integers starting from 1. The loop should terminate early if the sum becomes greater than 40.

### 8.3.5. continue statement

The purpose of continue statement is to start the next iteration of a loop while bypassing the remaining statements in the body of the loop. The following code illustrates the use of continue statement. In the code, when the value of variable i is equal to 3 then the while loop starts again from the beginning (i++) without executing next two lines (`n+=i; document.write ("n=" + n + ", i=" + i + "<br>");`).

#### </> Code 8.9

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
</head>
<body>
    <p>Example: JavaScript Statement</p>
    Result: <br>
    <script>
        var i = 0;
        var n = 0;
        while (i<5)
        {
            i++;
            if (i == 3)
                continue;

            n += i;
            document.write ("n=" + n + ", i=" + i + "<br>");
        }
    </script>
</body>
</html>
```

The output of above code is shown in figure 8.9.

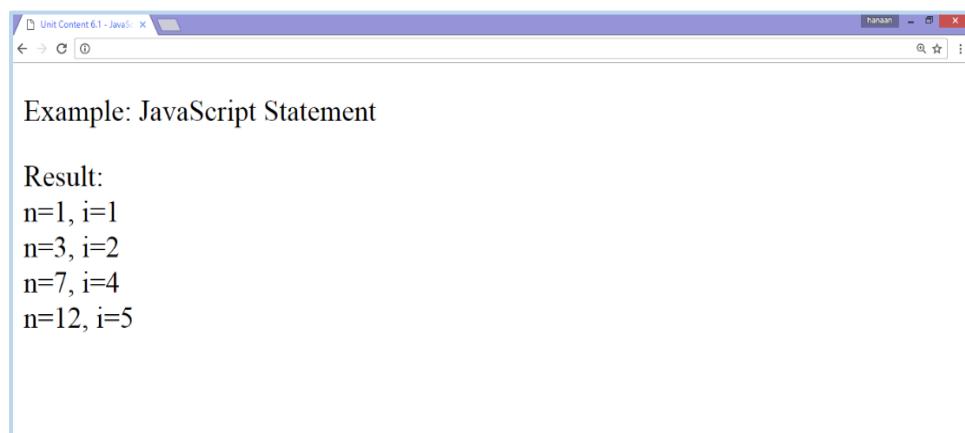


Figure 8.9: Output of code 8.9



## Activity 11

Write a program that calculates the sum of first 10 even numbers, negative numbers are skipped from calculation.

### 8.4. JavaScript Objects

JavaScript is an Object-Oriented Programming (OOP) language. In JavaScript, everything is an object except language constructs, keyword and operators. Objects in JavaScript fall into four groups: user-defined, native, host and document. User-defined objects are those which are created by the programmer. Native objects are objects that are specified as a part of JavaScript language (e.g. Array, Boolean, Date). Host objects are those objects that are not specified as part of JavaScript language but that are supported by the host environment, typically by browser (e.g. Window, Navigator). Document objects are those objects that are specified only for Document Object Model (DOM) (e.g. Image, HTMLInputElement).



Objects are also called variables but it contains many values

#### 8.4.1. Object Properties

A property of an object is a piece of named data (i.e. variables). The properties of an object can be accessed with dot ( . ) operator applied to an object. For example, length property of object str can be accessed by str.length as the following code shows

```
var str= new String("JavaScript Objects");
alert(str.length);
```



The object properties define the characteristics of an object.

#### 8.4.2. Object Methods

The object methods are functions that allow the object to do something or let something can be done to it. The following example shows how we can invokeUpperCase() method of the String object:

```
var str = new String ("good luck");
alert(str.toUpperCase());
```

Similarly, the following line of code invokes write() method of Document object.

```
document.write ("some text");
```



Object methods are the actions that are performed on objects.

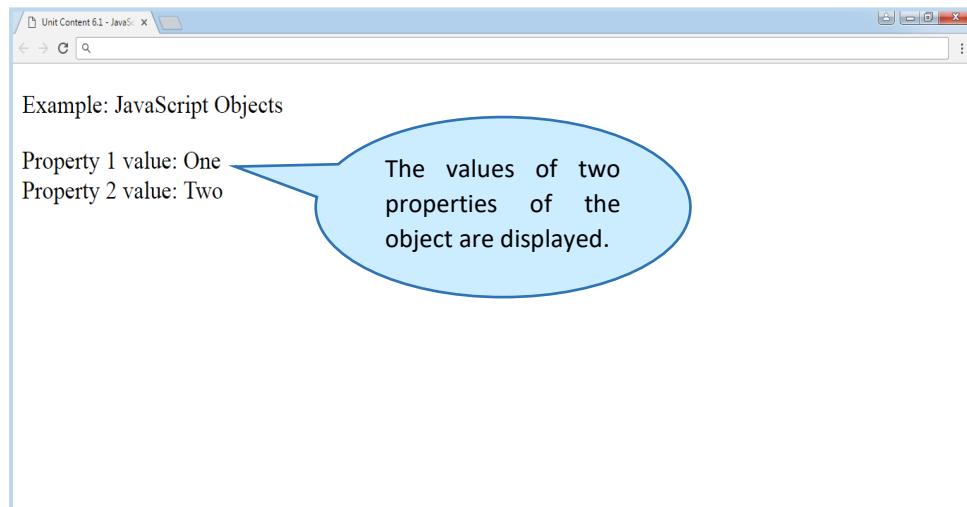
### 8.4.3. Object Constructor

The function which creates and initializes an object is known as object constructor. JavaScript has a special constructor called **Object()** to create an object. The following example shows how a constructor assigns and accesses properties.

**</> Code 8.10**

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
    // Create the object
    var obj = new Object();
    // Assign properties to the object
    obj.prop1 = "One";
    obj.prop2 = "Two";
</script>
</head>
<body>
    <p>Example: JavaScript Objects</p>
    <script>
        document.write("Property 1 value: " + obj.prop1 + "<br>");
        document.write("Property 2 value: " + obj.prop2 + "<br>");
    </script>
</body>
</html>
```

The output of above code is shown in figure 8.10.



**Figure 8.10:** Output of code 8.10

### Activity 12

Modify the code 8.10 to create two objects and assign different properties to each object and display the object properties.



## 8.5. JavaScript and HTML Form

JavaScript code can be used to manipulate and validate input data entered by a user in HTML forms. Assume we have the following form element in our HTML document

```
<input type="text" name="name" id="inputName" value="" />
```

Then we can access the value from the text box using the text box id (i.e. inputName) using the following JavaScript code

```
iName = document.getElementById("inputName").value
```

The value stored in a variable (iName) can be manipulated. For example, if we want to display an error message if the user does not enter the name, we can use the following code

```
if (iName == "")  
    alert('Please enter name.');
```

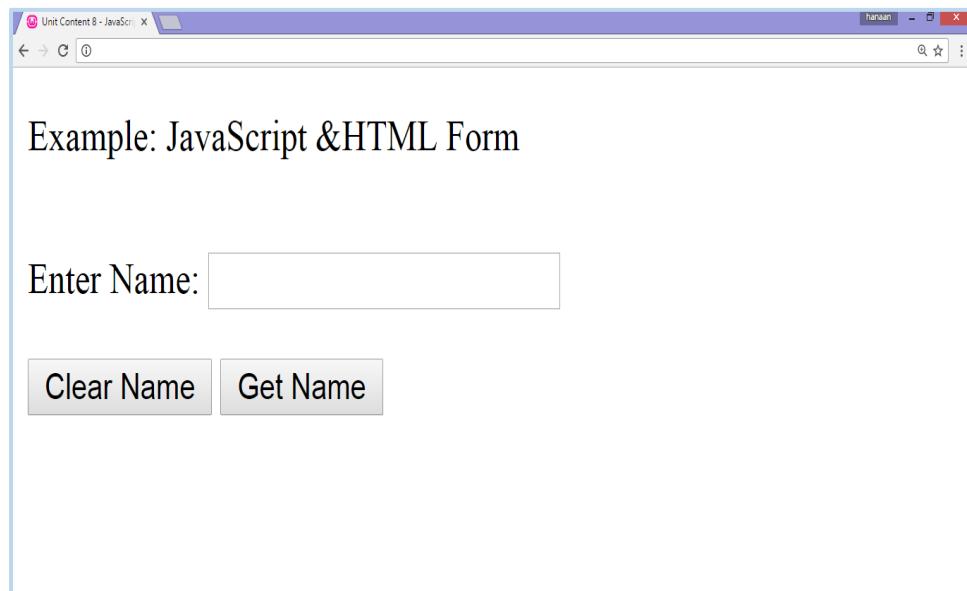
The JavaScript code can be bound to events such as onClick. The following code shows how we can validate data entered in a text box.

### Code 8.11

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Unit Content 8 - JavaScript Detail</title>  
<script type="text/javascript">  
    function clearName()  
    {  
        document.getElementById("inputName").value = "";  
    }  
    function getName()  
    {  
        var inputName = document.getElementById("inputName").value;  
        if (inputName != "")  
            alert(inputName);  
  
        else  
            alert('Please enter name.');//  
    }  
</script>  
</head>  
<body>  
    <p>Example: JavaScript &HTML Form</p><br>  
    <form name="htmlForm" action="" method="post">  
        Enter Name:  
        <input type="text" name="name" id="inputName" value="" />  
        <br>  
        <br><input type="button" name="btnClear" value="Clear Name"  
              onclick="clearName()" />  
        <input type="button" name="btnGet" value="Get Name"  
              onclick="getName()" />
```

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

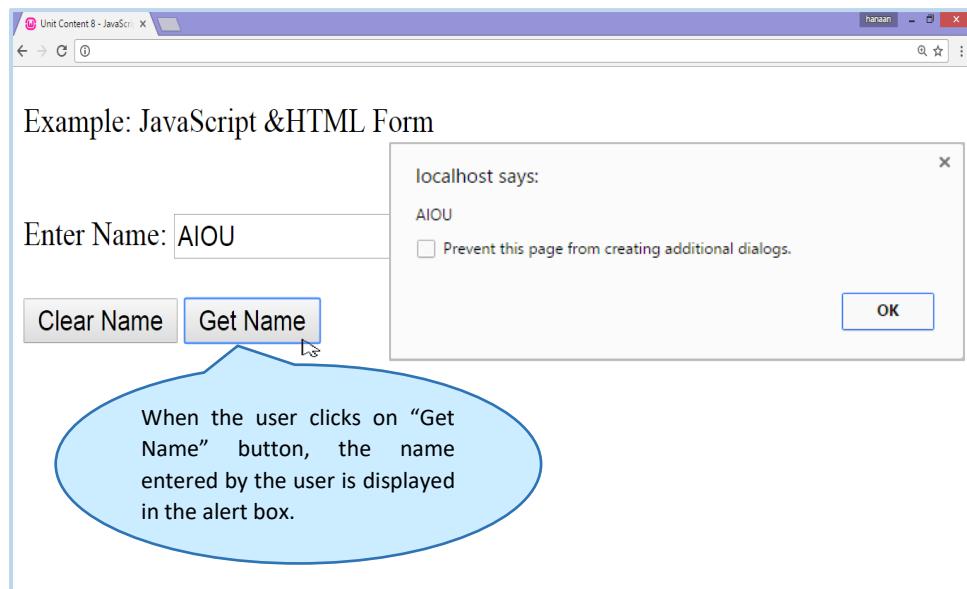
The output of above code is shown in figure 8.11 and 8.12.



Example: JavaScript & HTML Form

Enter Name:

**Figure 8.11:** Output of code 8.11



Example: JavaScript & HTML Form

Enter Name:

localhost says:  
AIOU  
 Prevent this page from creating additional dialogs.

When the user clicks on “Get Name” button, the name entered by the user is displayed in the alert box.

**Figure 8.12:** Output of code 8.11

### Activity 13

Create student registration form and perform input validation using JavaScript before submit, also place reset button to clear all input fields of registration form.



## 8.6. JavaScript Events

The actions to which JavaScript can respond to are called events. The browser can trigger some events such as page loading, but most of the events are initiated by the user interaction such as clicking a button, pressing a key, closing a window, or moving the mouse over some page element. An event handler is a javascript code associated with a particular event. An event handler is executed when a given event occurs. Let us look at some of the events which are very important in developing interactive and responsive websites.



An event handler is a code which executes when a particular event occurs.

### onclick event

This event occurs when a user clicks left mouse button. The following code shows how we can respond to this event. Note that what all we have to do is to write a function (event handler) in the head section of the HTML document that will be called and executed when a user click mouse button on an element of a form to which it is bound.

#### </> Code 8.12

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
<script type="text/javascript">
    function sayHello()
    {
        alert("Hello World!");
    }
</script>
</head>
<body>
    <p>Example: JavaScript Events</p>
    <p>Click the following button and see result</p>
    <input type="button" onClick ="sayHello()" value="Say Hello"/>
</body>
</html>
```

The output of above code is shown in figure 8.13 and 8.14.

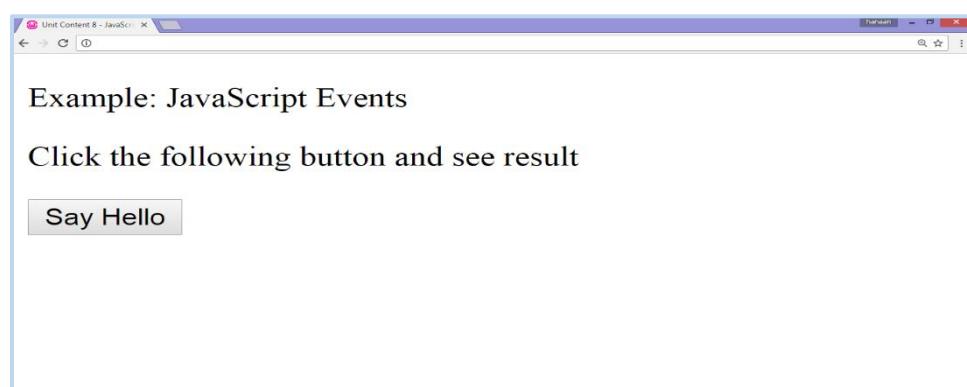
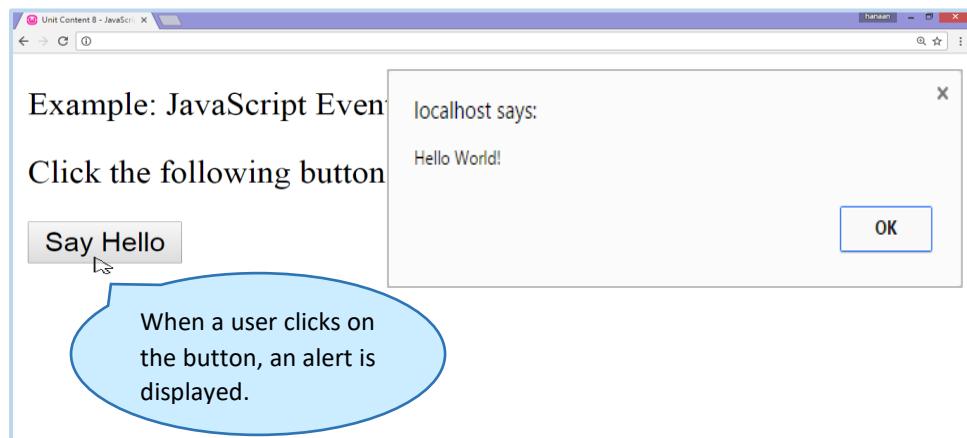


Figure 8.13: Output of code 8.12

When a user clicks on the button ("Say Hello"), then the alert will be displayed as shown in the following output.



**Figure 8.14:** Output of code 8.12

## Activity 14

We want to update the “to-do” list, but not before the user clicks on the “Click me!” button. Change the JavaScript code in such a way that it will execute when the user clicks the button. Following are some tips to do this activity.

- Put all of the given code into a function with a name of your own choice.
- Set the “onclick” event handler for the button which refers to the created method.

## onsubmit Event

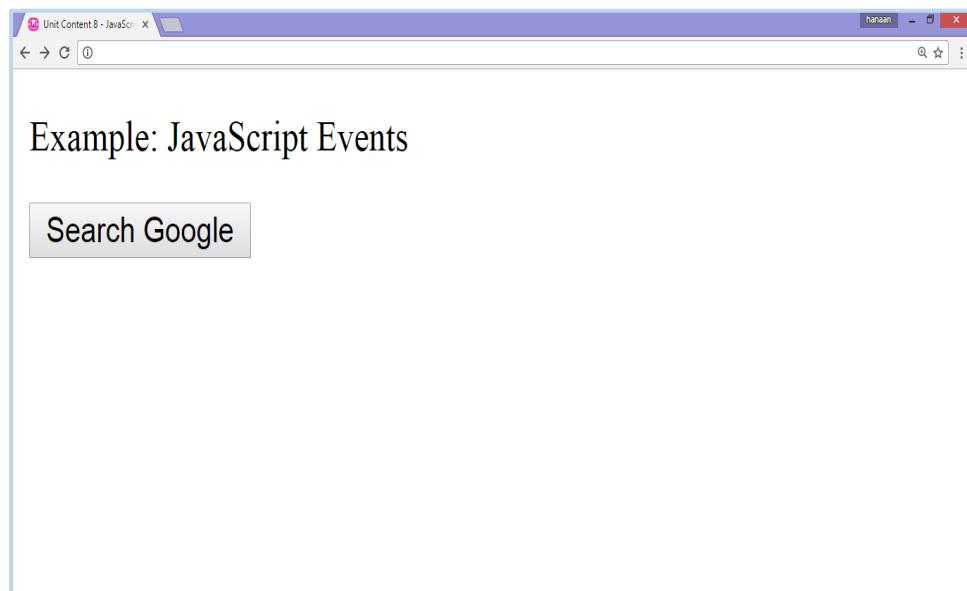
This event occurs when a user clicks on the submit button on a form. In the following code, doConfirm() function is executed before submitting a form data. The result for using the doConfirm() function is that the form will be submitted if the function returns true otherwise it will not submit the data.

### Code 8.13

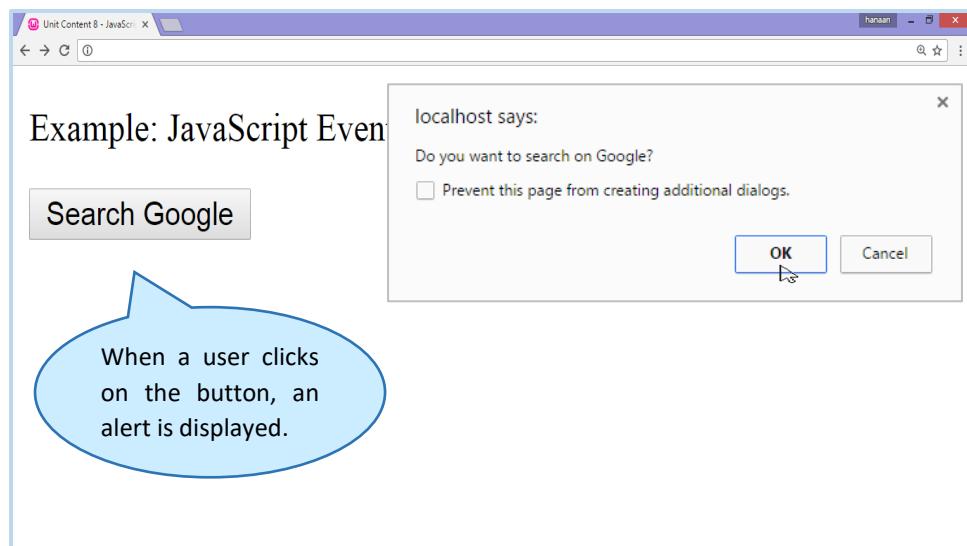
```
</> Code 8.13
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
<script type="text/javascript">
    function doConfirm()
    {
        return confirm("Do you want to search on Google?");
    }
</script>
</head>
<body>
    <p>Example: JavaScript Events</p>
    <form method="GET" action="https://www.google.com.pk/">
        <input type="submit" value="Search Google"/>
    </form>
</body>
</html>
```



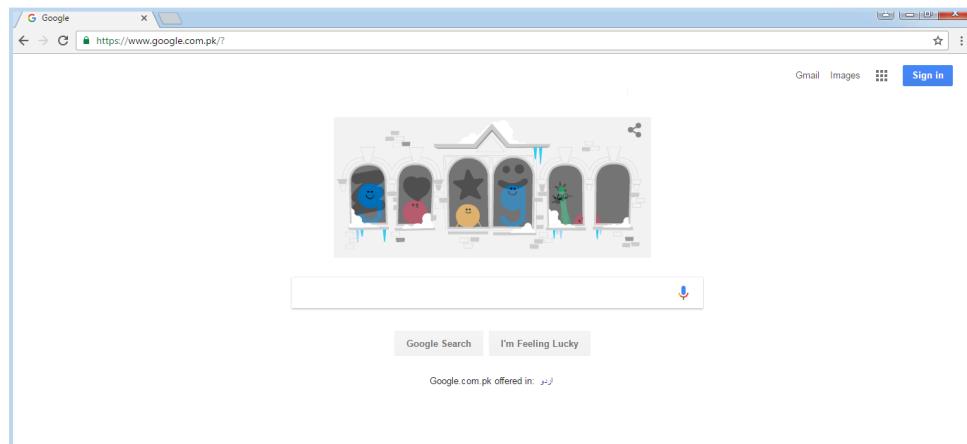
The output of above code is shown in figure 8.15, 8.16 and 8.17.



**Figure 8.15:** Output of code 8.13



**Figure 8.16:** Output of code 8.13



**Figure 8.17:** Output of code 8.13

## Activity 15

Create a web page that has a form element with an input button. When the user clicks on the input button, the background color should be changed to green using `document.bgColor` property and `onClick` event handler.

### onMouseOver and onMouseOut

`onMouseOver` and `onMouseOut` events trigger when user brings the mouse on the element and moves the mouse out of the element respectively. The following example code illustrates how we catch these events and trigger the event handlers

#### Code 8.14

```
<!DOCTYPE html>
<html>
<head>
<title>Unit Content 8 - JavaScript Detail</title>
<script type="text/javascript">
    function doMouseIn(e)
    {
        e.style.background = "yellow";
    }
    function doMouseOut(e)
    {
        e.style.background = "white";
    }
</script>
</head>
<body>
    <p>Example: JavaScript Events</p>
    <div>
        onMouseOver="doMouseIn(this);"
        onMouseOut =
            "doMouseOut(this);"
        Move your mouse pointer into and out of this element!
    </div>
</body>
</html>
```

The output of above code is shown in figure 8.14 and 8.15.

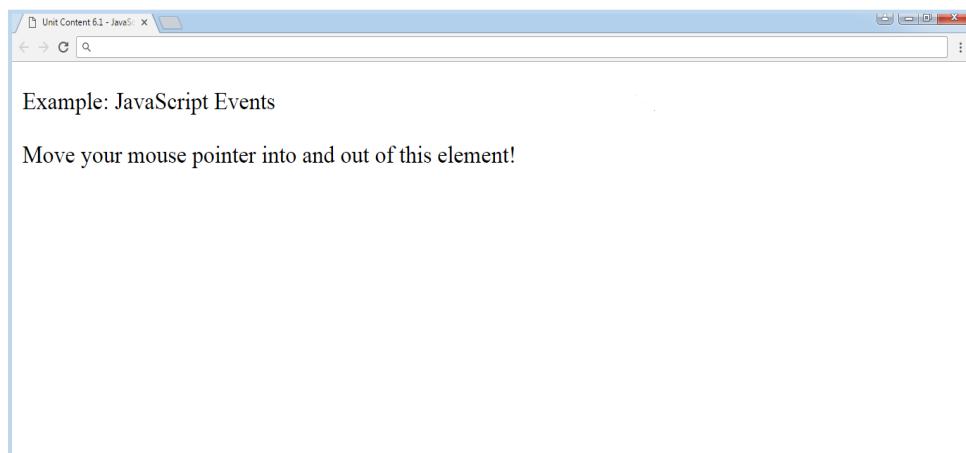
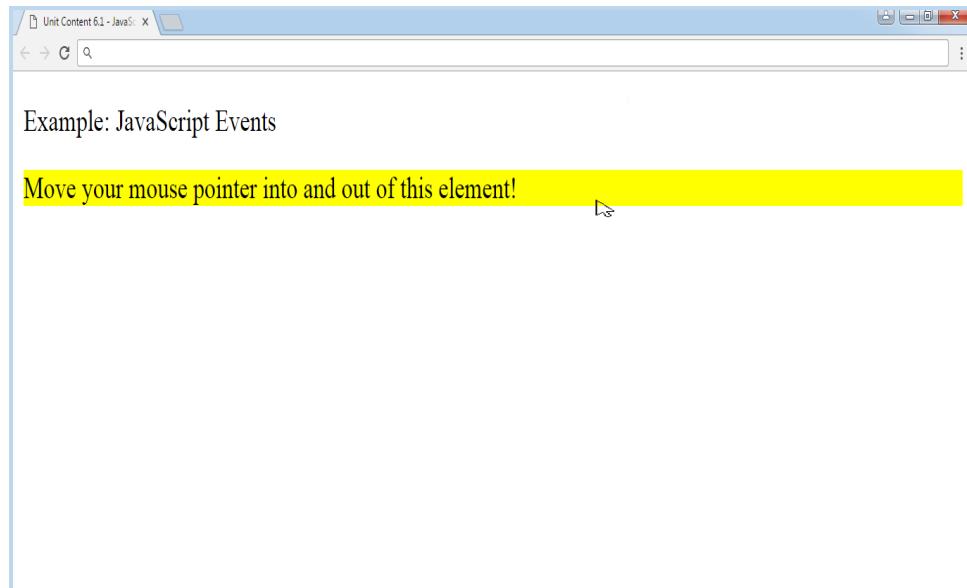


Figure 8.18: Output of code 8.14



**Figure 8.19:** Output of code 8.14

## **Activity 16**

Modify the code 8.14 and change the event handlers by changing the background colors.



## **Unit Summary**

In this unit, we learned how to use selection statements in javascript programs. We also learned the use of loops to control the program statements. This unit also covers important concept of HTML form and its associated elements in detail. The last section of the unit is focused on developing interactive and responsive web pages using javascript.



## Self Assessment Questions

**Choose the correct answer.**

1. The javascript code written in an HTML page is executed on the \_\_\_\_\_
  - A. Client-side
  - B. Server-side
  - C. Local
  - D. Native
  
2. Which of the following statement is used to perform a specific action when a condition is true or false?
  - A. While
  - B. If
  - C. For
  - D. All of the above.
  
3. Which of the following if statement is written correctly?
  - A. if i <5 i++;
  - B. if (i <5) then i++;
  - C. if (i <5) i++;
  - D. None of the above.
  
4. \_\_\_\_\_ loop executes at least once.
  - A. for loop
  - B. while loop
  - C. do while loop
  - D. All of the above.
  
5. Every switch statement can also be written as an if statement?
  - A. True
  - B. False
  
6. A while loop executes as long as the loop condition remains true?
  - A. True
  - B. False
  
7. \_\_\_\_\_ statement can be used for early termination of a loop?
  - A. continue
  - B. next
  - C. i++
  - D. break
  
8. Which of the following statement is wrong?
  - A. document.write("hello");
  - B. alert("hello")
  - C. Both A and B
  - D. None



9. \_\_\_\_\_ event occurs when a form element loses the focus.

- A. Onfocus
- B. Onblur
- C. Onclick
- D. Ondblclick

10. \_\_\_\_\_ event occurs when a user clicks on submit button.

- A. Onclick
- B. Submit
- C. Onsubmit
- D. Onmouse over

### Answer Key:

- |      |      |      |      |      |      |      |      |      |       |
|------|------|------|------|------|------|------|------|------|-------|
| 1. A | 2. B | 3. D | 4. C | 5. A | 6. A | 7. D | 8. B | 9. A | 10. C |
|------|------|------|------|------|------|------|------|------|-------|



## Review Questions

Write short answers of the following questions

1. What are different types of operators?
2. Differentiate between comparison operator and logical operators with examples.
3. Which operator is used to compare two values, = or ==?
4. What is the difference between "while-do" and "do-while" loops?
5. Discuss how if else statement is executed.
6. How can we assign properties to an Object in javascript?
7. How are object created in JavaScript?
8. What do you mean by events? What is difference between Onmouse over and Onmouse out event?
9. How we can retrieve a value from a form element? Explain with example
10. What is the purpose of on error event handler in JavaScript?

## </> Coding Exercise

1. Create a variable x, and set it to 10. Create another variable y, and set it to 12. Create another variable z, and set it to the sum of x and y. Display the value of z.
2. Create a variable age and set it to 15. Now write a conditional statement that displays the message “You are too young” if the value of age is less than 16. If the value of age is 16 or more, display the message “You may enter”.
3. Write javascript code using a while-loop that calculates the sum of the numbers from 1 to 30 and displays the results. The message to the user should look like: “The sum of all numbers from 1 to 30 is...”
4. Write javascript code that calculate and displays the factorial of x. You should declare variable x and give it a value.

5. Write a javascript code that displays minimum and maximum values out of four given numbers. The number should be stored in four different variables.
  6. Develop a web page with javascript that displays all the even numbers from 1 to 100.,
  7. Write a javascript program which prints "PASS" if the int variable "mark" is more than or equal to 50; prints "FAIL" otherwise.
  
  8. Write a program which allows entering the age of user and shows that either he is eligible to vote. The basic criteria of eligibility of voting age are above and equal to 18 years.
  9. Write a JavaScript program to calculate the area and perimeter of a circle.
  10. Create a web page with a form that allows a user to enter two numbers and performs the addition, subtraction, multiplication, division depending on the user's choice. For each operation, create a button on a form and write appropriate event handlers for each button.
- 



## References and Further Reading

1. Haverbeke, M. (2014). Eloquent javascript: A modern introduction to programming. No Starch Press.
  2. Elliott, E. (2014). Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. O'Reilly Media, Inc.
  3. Resig, J., & Bibeault, B. (2013). Secrets of the JavaScript Ninja. Manning.
  4. Rauschmayer, A. (2014). Speaking JavaScript: An In-Depth Guide for Programmers. O'Reilly Media, Inc.
  5. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH.
  6. Javascript Questions. Available on: <http://www.slideshare.net/stuartsandeep/top-javascript>
-



## UNIT 9

# Website Hosting



## Introduction

We have discussed how we can develop a website using HTML, CSS and JavaScript. Once we have our website ready and tested for its content and functionality, we need to upload the website (web page) to a web server to make it accessible to the users. To make a website up and running, we have to find a suitable web hosting server. There are number of ISPs that provide web hosting services on payment. We might find some free hosting services as well. This unit discusses about different web hosting options available to us. We then discuss a step by step procedure to upload our web pages to a selected server.

---

## Unit Outcomes

At the completion of this unit, you should be able to:

1. Explain different types of web servers.
  2. In list different types of web hosting services.
  3. Describe domain names.
  4. Upload a website using FTP.
- 

---

## Terminologies

**Web Server:** A web server is a computer system that deliver services to the end users

**Web Hosting:** It is a service that host websites by providing space on the Internet.

**Domain Name:** It is a unique identifier for a website.

**Internet Service Provider:** A company that provide Internet services.

**File transfer protocol:** It is a protocol which is used to transfer files on a computer network.

---



## 9.1. Web Servers

A web server is software running on a computer that responds to HTTP requests to deliver content and services. Generally, when we use the word “server”, we mean server software and the computer on which it is running. Every web server has a unique IP address that can be used to access the content stored on it.



Any device/computer that delivers an HTML document to another device can be a web server.

### 9.1.1. Type of Web Servers

There are different types of web servers available in the market. Some commonly used web servers are:

- Apache Web Server
- IIS Web Server
- Nginx Web Server
- Light Speed Web Server

#### Apache Web Server

Apache is widely web server developed by Apache Software Foundation. It is an open source software that can be installed on almost all operating systems including i.e. Linux, UNIX, Windows, and Mac OS X etc. Its modular structure helps to customize it easily. It is very stable, easy to install and manageable.



Apache is an open source software and can be installed on almost all operating systems.

#### IIS Web Server

IIS is a Microsoft web server which has almost all the features that are available in Apache. However, it is not an open source and hence new modules and features cannot be added easily. It works on Windows platform.

#### Nginx Web Server

Nginx is a free open source web server. It is known for its high performance, stability, simple configuration and low resource usage.



Nginx is known for its high performance, stability, simple configuration and low resource usage.

#### Lite Speed Web Server

Lite Speed (LSWS) - the 4th most popular commercial web server is a high-performance Apache drop-in replacement. Use of Lite Speed Web Server improves the performance and reduces the operating cost.



This server is compatible with most common Apache features, including mod rewrite, .ht access, and mod security.



## Activity 1

Surf the Internet and explore detailed features of different web servers.



## 9.2. Web Hosting

Web hosting is a service that helps organizations and individuals to upload a website or web pages on Internet. A web host, or a web hosting service provider, is a business/company that provides the technologies and services needed for the website or web page to be stored and viewed on Internet. Websites are hosted or stored on special computers called servers. Most of the hosting companies require that we own a domain name in order to host with them. If we do not have a domain name, the hosting companies helps to purchase one.



Web hosting is a service that allows organizations and individuals to place or host a website to make it accessible on the Internet.

### 9.2.1. Type of Web Hosting

There are various types of web hosting services available to host a website. Before signing up for web hosting services, it is important to understand what kind of services our website needs, the kind of server our business needs, our budget, and what type of services the web host offers. There are different types of web hosting services available such as:

- Free hosting
- Shared hosting
- Dedicated hosting
- Collocated hosting

#### Free hosting

Free web hosting provides free of cost hosting facility for a website. Due to free of cost hosting the problems like slow speed connection, frequent linking down of the website and automatic addition of advertising banners are usually faced. Few companies compel to purchase a domain name to avail free hosting services, while others offer a free subdomain under them, such as [yourname.webhost.com](http://yourname.webhost.com).



Free hosting services may be down frequently. Advertising banners are automatically added to the website.

#### Shared Hosting

In a shared hosting environment, a number of websites share one server. This includes sharing the physical server as well as software applications running on the server. Due to shared hosting, the performance of a website may goes down.



Shared hosting services are affordable because the cost to operate the server is shared among the clients.

### Dedicated hosting

In a dedicated hosting environment, a dedicated web server is reserved for a particular website only which results into faster performance, as we have all the server's resources due to no sharing with anybody else. However, the user has to bear the entire cost solely. This is a good choice for websites that requires a lot of system resources or need a higher level of security.



Dedicated hosting allows for faster performance.

### Collocated hosting

In this type of hosting, we purchase our own server which is housed in a rental space along with server of many other companies and individuals. It allows full control of the server and the installation of any script or application we need.



Collocated hosting provide full control of the web server.

## Activity 2

Surf the Internet and find the organizations which provide free hosting services. Find the local companies which provide web hosting services.



## 9.3. Domain Name

A domain name is a unique name that identifies a website. Each website has a domain name that serves as an address, which is used to access the website. Whenever we visit a website, the domain name appears in the address bar of the web browser. Some domain names are preceded by "www" (which is not part of the domain name), while others omit the "www" prefix. All domain names have a domain suffix, such as .com, .net, or .org. The domain suffix helps identify the type of website. For example, ".com" domain names are typically used by commercial website, while ".org" websites are often used by non-profit organizations. Some domain names end with a country code, such as ".pk" (Pakistan).



Each website has a domain name that serves as an address, which is used to access the website.

### 9.3.1. Sub Domain

It is not always necessary to register a new domain name. Rather than registering a new domain name, we can always create a **subdomain** using a domain we already own. The subdomain name looks like forums.domain.com, help.domain.com and help2.domain.com, assuming we already have domain.com.



A subdomain is a second website, with its own unique content, but there is no new domain name.

### Activity 3

Find the domain name of your university website.

## 9.4. File Transfer Protocol

FTP stands for "file transfer protocol". It powers one of the fundamental Internet functions and is the prescribed method for the transfer of files between computers. An FTP address looks like a website address except that it uses the prefix `ftp://` instead of `http://`. The most common use of FTP is to download and upload files.



FTP is the easiest and most secure way to exchange files over the Internet.

### 9.4.1. FTP Server

Typically, a computer with an FTP address is dedicated to receive an FTP connection. A computer dedicated to receiving an FTP connection is referred to as an FTP server or FTP site.

### 9.4.2. How we can use FTP?

1. Most web hosting services allow their customers to upload the contents of websites through FTP.
2. Companies usually have their own FTP servers that allow users to send and receive files.
3. Most universities have FTP servers to facilitate their students to download course materials and upload assignments for submission.

### Activity 4

Download the FTP user guide from the Internet and prepare the Unit Summary of FTP commands.

## 9.5. Uploading a Website

There are two methods for uploading web pages on a server but before uploading web pages on a server, we must have domain name and web hosting account on which we can upload files. The following two utilities can be used to upload a website on a server:

- Upload using cPanel File Manager
- Upload using Filezilla FTP client



### 9.5.1 Upload using cPanel File Manager

cPanel is the most well-known software for uploading a website on a server. Using cPanel's inbuilt File Manager option, we can upload a website's files using the following steps:

1. Login to cPanel.
2. In cPanel's **Files** section, click on the **File Manager** icon as shown the following figure 9.1.

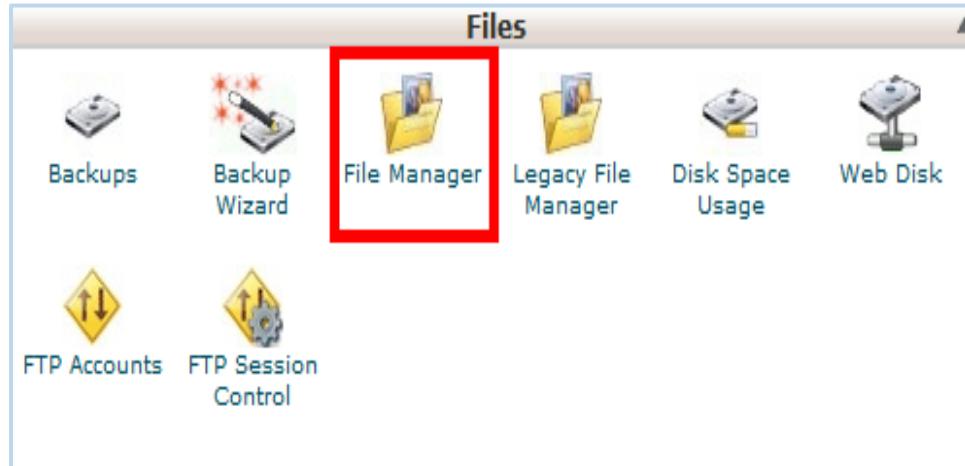


Figure 9.1: File section of cPanel

3. Now select the directory as **Web Root (public\_html/www)** and choose the domain name from drop down arrow in case we have multiple domains under a hosting plan.
4. Check on **Show Hidden Files (dotfiles)** and click on **Go** button as shown in the figure 9.2.

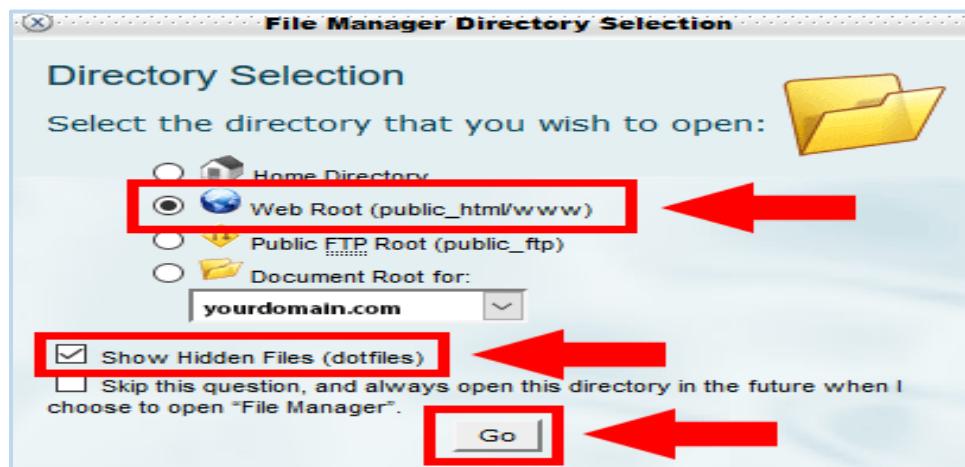
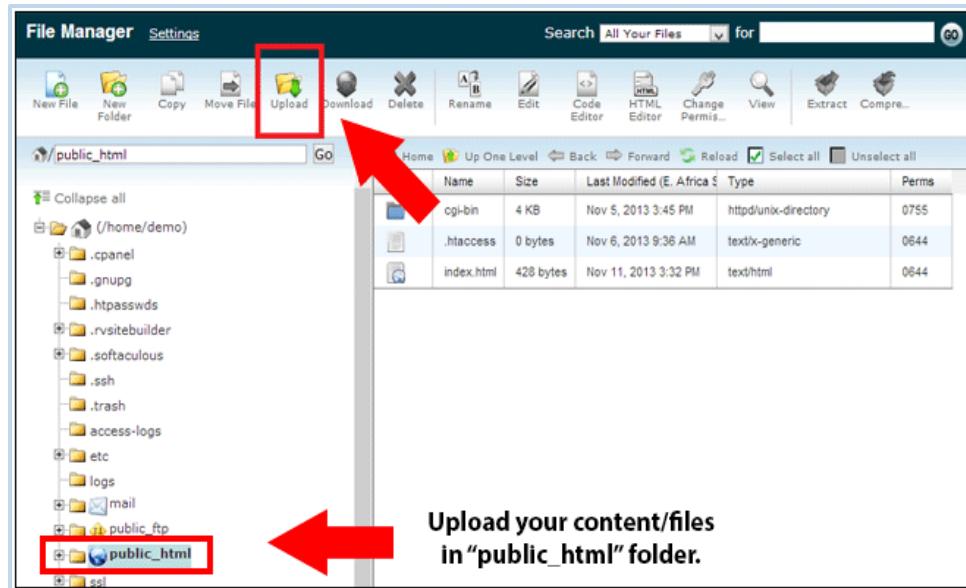


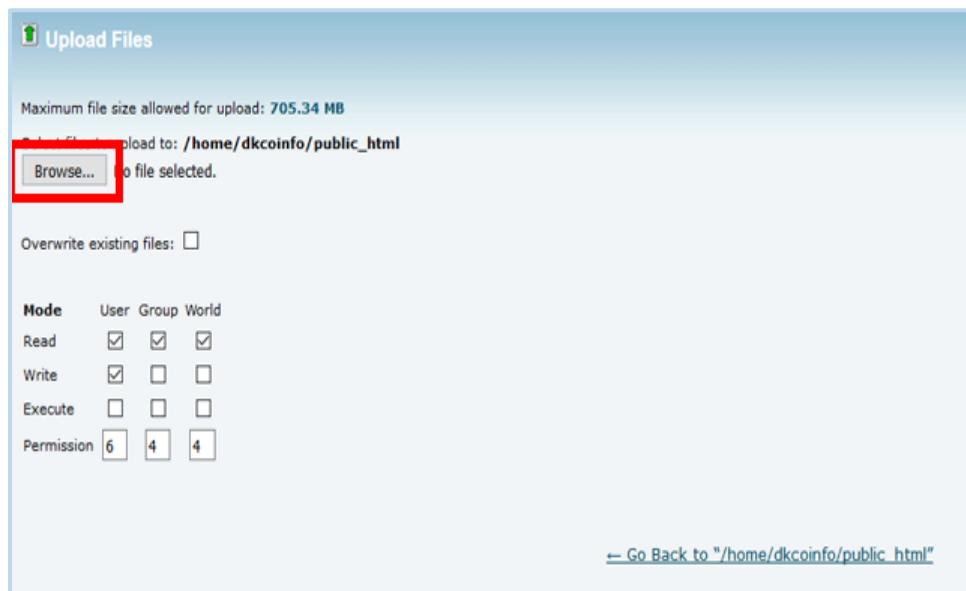
Figure 9.2: File manager directory selection.

5. Click on **Public\_html** folder shown at the left side on the screen under the folder directory. This will show list of files in the public\_html folder. All files must be upload in **public\_html** folder.
6. Click on **Upload** in the navigation bar on the top of the page as shown in the figure 9.3.



**Figure 9.3:** File manager setting.

- Click on **Browse** button and select the website's files from your PC/mobile which you want to upload as shown in figure 9.4.



**Figure 9.4:** File selection menu

- After selecting the files we want to upload, click on **Open** button and wait for the uploading process to get completed.
- Once the uploading process is complete, click **Go Back** button. It will take us back to **public\_html** folder in the cPanel as shown in figure 9.5.



Figure 9.5: Link of Go Back button

10. Now the website has been uploaded. We can check the website by opening it in the browser.

### 9.5.2 Upload using Filezilla FTP client

We can easily upload web pages from our PC using FTP software like Filezilla. For uploading web pages using Filezilla, follow these steps:

1. Download and install Filezilla software from the internet.
2. Open FTP software (Filezilla).
3. Login to Filezilla. In the top of the navigation bar there are 3 fields as shown in figure 9.6. You need to enter the following:

**Host Name:** Enter the host as a domain name or the IP address of the hosting account.

**User Name:** Enter the User name of your hosting account.

**Password:** Enter the password of your hosting account.

**Port:** Leave the port box blank.

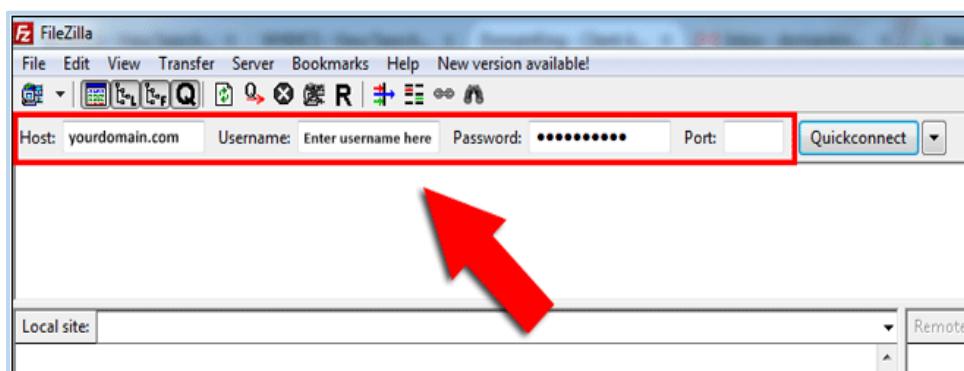
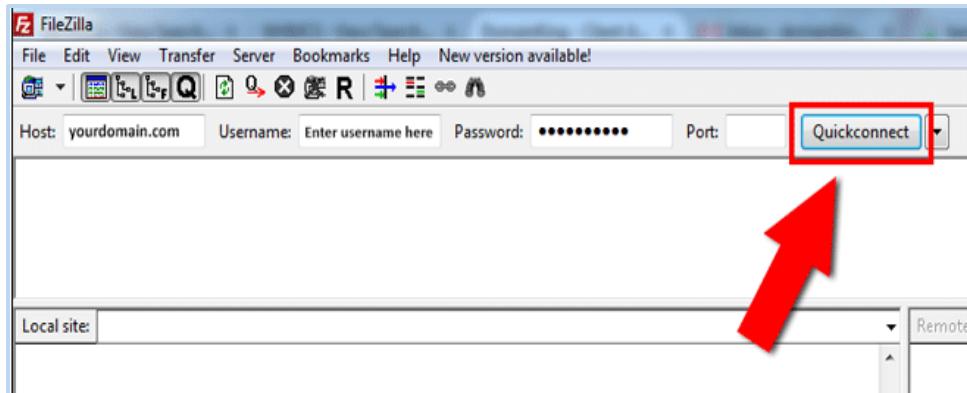


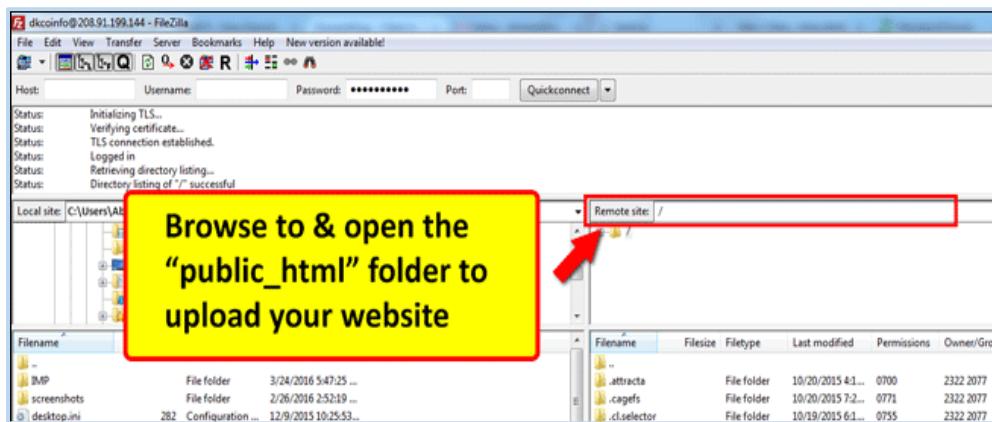
Figure 9.6: Login screen of FileZilla

4. Click on **Quickconnect** button to connect Filezilla with the hosting account as shown in figure 9.7.



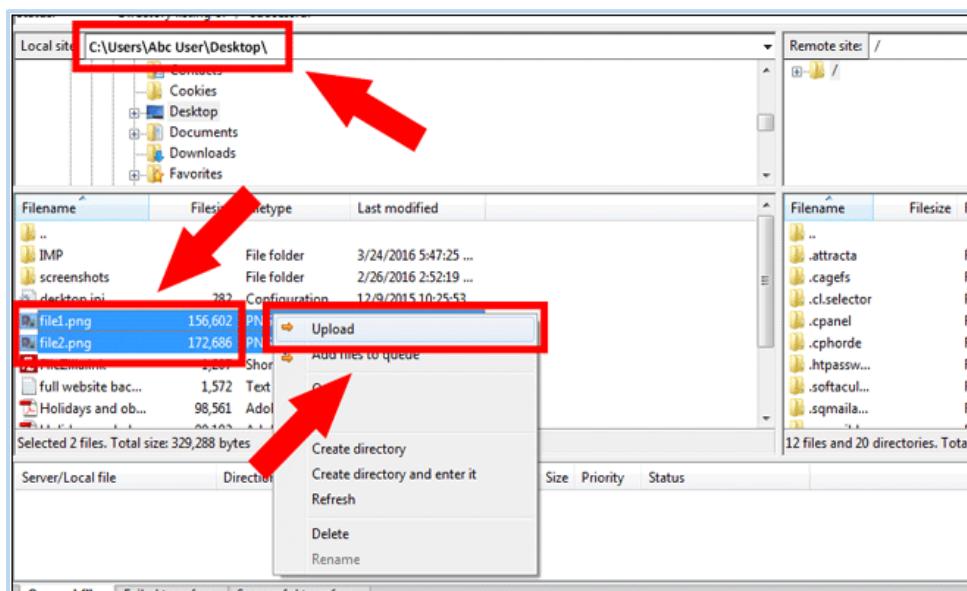
**Figure 9.7:** Connection of FileZilla with hosting account.

5. On the right side under the **Remote Site**, a list of folders is displayed. Select the folder on server in which to upload website as shown in the figure 9.8.



**Figure 9.8:** Folder / File selection menu.

6. Now on the left side under **Local Site** section, we will see a list of folders on our PC/Desktop. Navigate to the files that we want to upload to the server. Select the files and then **Right Click** on the files and click **Upload** as shown in figure 9.9.



**Figure 9.9:** Uploading of file.



The website's files are uploading to server. We can check the uploading process showing at the bottom of your Filezilla FTP client. Once the website has been successfully uploaded, we can check our website by opening it in the browser.

---

### **Activity 5**

Upload your personal website on a free web hosting server.

---



### **Unit Summary**

In this unit, we learned about different types of web servers, web hosting services and domain names. We have also discussed how to upload a website on a server using File Transfer Protocol.

---



## Self Assessment Questions

Select the correct answer.

1. \_\_\_\_\_ is an open source software and can be installed on almost all operating systems.
  - A. Apache Web Server
  - B. IIS Web Server
  - C. Nginx Web Server
  - D. Light Speed Web Server
  
2. \_\_\_\_\_ is known for its high performance, stability, simple configuration and low resource usage.
  - A. Apache Web Server
  - B. IIS Web Server
  - C. Nginx Web Server
  - D. Light Speed Web Server
  
3. \_\_\_\_\_ Server is compatible with most common Apache features, including mod rewrite, .htaccess, and mod security.
  - A. Apache Web Server
  - B. IIS Web Server
  - C. Nginx Web Server
  - D. Light Speed Web Server
  
4. A web hosting service provider allows individuals to post a website or web page on to the Internet?
  - A. True
  - B. False
  
5. \_\_\_\_\_ hosting services are affordable because the cost to operate the server is shared among the clients.
  - A. Shared Hosting
  - B. Dedicated Hosting
  - C. Collocated Hosting
  - D. Free Hosting
  
6. \_\_\_\_\_ hosting service allows for faster performance.
  - A. Shared Hosting
  - B. Dedicated Hosting
  - C. Collocated Hosting
  - D. Free Hosting
  
7. \_\_\_\_\_ hosting service provides faster performance, as you have all the server's resources entirely.
  - A. Shared Hosting
  - B. Dedicated Hosting
  - C. Collocated Hosting
  - D. Free Hosting



8. \_\_\_\_\_ is a second website, with its own unique content, but there is no new domain name.
  - A. Sub Domain
  - B. Domain
  - C. Web Server
  - D. None of these
  
9. \_\_\_\_\_ delivers an HTML document to a client on the Internet.
  - A. Web Server
  - B. Web Hosting
  - C. All of the above
  - D. None of these
  
10. \_\_\_\_\_ is compatible with most common Apache features.
  - A. Web Server
  - B. Web Hosting
  - C. All of the above
  - D. None of these

**Answer Key:**

1. A	2. B	3. D	4. A	5. A	6. B	7. B	8. A	9. B	10. C
------	------	------	------	------	------	------	------	------	-------

**Review Questions**

**Write short answers of the following.**

1. What are web servers? Explain the types of web servers.
2. Explain web hosting and its types.
3. What is domain name? How domain name differs from sub domain?
4. Briefly explain the concept of FTP?
5. How can we upload a website on a server with cPanel file manager?

**References and Further Reading**

1. Web Servers. Available on: <http://www.fastwebhost.in/blog/what-is-web-servers-different-types-of-web-servers/>
2. Website Hosting. Available on: <http://www.itmanaged.net/services/website-hosting/>
3. Web Hosting. Available on: <http://www.clickwebhosting.com/>
4. Web Hosting. Available on: <http://www.labwebdesigns.com/>



Website Hosting



THE COMMONWEALTH of LEARNING



Department of Computer Science  
Allama Iqbal Open University, Pakistan