WEB PROGRAMMING - I
(Client Side Scripting)

THE COMMONWEALTH of LEARNING

Department of Computer Science
Allama Iqbal Open University, Pakistan

Web Programming - I
(Client Side Scripting)

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.

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

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

Mr. Sarfraz Haider (Directorate of ICT, AIOU, Pakistan)

Reviewer

Dr. Anwaar Manzar, TI (Hamdard University, Islamabad, Pakistan)

Copy Editor

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

Layout Design

Mr. Mushtaq Hussain (Print Production Unit, AIOU)

Video Production

Ms. Sana Nasim Karam (Department of Computer Science, AIOU, Pakistan)

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 Tayyab Shoaib (Department of Computer Science, AIOU, Pakistan)

Mr. Naveed Ullah (Department of Computer Science, AIOU, Pakistan)

Mr. Yasir Khan (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

Web Programming – I (Client Side Scripting) has been produced by Allama Iqbal Open University (AIOU) in collaboration with the 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 Web Programming – I (Client Side Scripting), 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 Web Programming – I (Client Side Scripting)

This course provides an introduction to web development and client-side scripting. After providing a review of HTML5 and CSS, the course provides exposer to the concepts of web programming using client side scripting. The course covers basic construction of web page, cascading style sheet, and java script. The course provides a solid foundation in computer programming in Javascript: syntax and data structures, AJAX, DOM, and JS libraries. The students will gain an understanding of the popular libraries that power rich web applications such as JQuery and AngularJS to build rich web applications.

Course Overview Video

https://youtu.be/KYPUE8xHHHI

Is this course for you?

This course is intended for the people who already have introductory knowledge about the Web Development. The course aims to:

- *Equip them with the understanding of the key concepts and demonstrations related to web programming.*
- *Enable students to write well-structured, easily maintained and standards-compliant web pages using HTML and CSS code.*
- *Make them expert of using JavaScript to add dynamic content to pages.*
- *Help to use JavaScript libraries (e.g. JQuery, AngularJS) to create dynamic pages.*
- *Learn to apply techniques of form validation using JavaScript.*
- *Be able to describe and appreciate emerging trends in client side web security.*

Course Outcomes

Outcomes

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

- *Demonstrate and understand the basic concepts of web programming*
- *Write well-structured, easily maintained, standards-compliant, web pages using HTML and CSS code.*
- *Use JavaScript to add dynamic content to pages that meet specific needs and interests.*
- *Use JavaScript libraries jQuery and AngularJS to create dynamic pages.*
- *Apply techniques of form validation using Java Script.*
- *Describe important concepts related to client side Web Security.*

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 and type "self-study basics", "self-study tips", "self-study skills" or similar.

Need Help?

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 Programming

Introduction

This is an introductory unit which covers the basic concepts about World Wide Web and its related terminologies. It provides an overview of Web programming and demonstrates how it is different from conventional computer programming. It highlights the process model for the development of websites. This unit also discusses different technologies related to web application development.

Unit Outcomes

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

1. Understand the basics of World Wide Web and its related concepts.
2. Differentiate between static and dynamic web pages.
3. Identify the characteristics of a website.
4. Explain web development process model.
5. Be familiar with important web technologies.

Terminologies

WWW:	World Wide Web is a collection of websites over the Internet.
Hypertext:	A document which contains text and multimedia objects.
Web Browser:	A client program that is used to retrieve and display web pages from Internet.
Domain Name:	A name which is used to identify a website.

1.1. Basic Concepts of World Wide Web (WWW)

The World Wide Web (WWW) commonly referred as the "Web" is a system of interlinked hypertext documents that can be accessed over Internet. Hypertext refers to interconnected documents that are linked together. Thus, the link available on a webpage is a Hypertext. A comprehensive definition comes from the World Wide Web Consortium (W3C) which states "*The WWW is the universe of network-accessible information, an embodiment of human knowledge*". The network of accessible information uses hypertext i.e. linked documents that may be located at different servers over Internet. WWW allows different text formats and multimedia components to organize information over Internet.

Internet is sometimes confused with WWW, but both of them are not same. Internet is a huge network which connects millions of computers around the world. It forms a wide area network in which any computer can communicate with another computer by following the specific rules called protocols. The information available over Internet is stored and communicated in the form of WWW. Internet connects computers and WWW connects people as shown in figure 1.1.

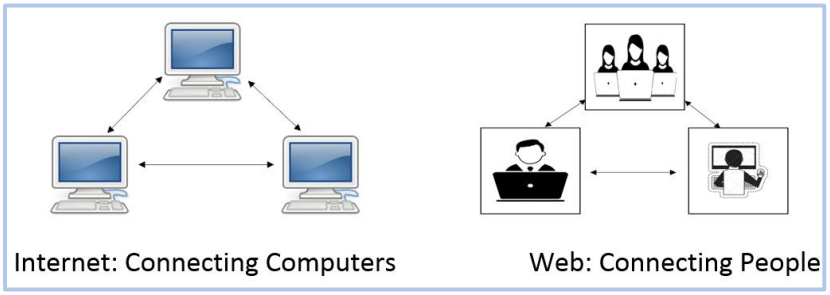

Figure 1.1: Internet connecting computers and people.

WWW connects contents and people while Internet connects computers.

Besides Internet, there are some other concepts that are related to WWW. These concepts are being described below:

1.1.1. Web page

A web page or webpage is a document that is accessible over Internet or other networks. A combination of web pages is called a website, which is a collection of pages, images, videos or other digital assets accessible via Internet, cell phone or a Local Area Network. Web pages are developed by using a special language called Hyper Text Markup Language (HTML).

1.1.2. Hyper Text Markup Language (HTML)

HTML is the standard markup language for the creation of web pages and web applications. HTML provides a way to place text and images and other media objects in a web page. We will study HTML in detail in chapter 2.

1.1.3. Hypertext and Hypermedia

Hypertext refers to the text and other kind of on-screen multimedia objects that are connected via hyperlinks displayed on our webpage. Hyperlinks give us choice to explore a large number of documents that are connected together, when we look for information, search for music, purchase products, and engage in similar activities on WWW. Hyperlinks appear in the form of underlined words in blue or some other color, buttons, and other "hot" areas on the screen. All the websites are entirely or largely hypertext documents.

Hypermedia refers to presentation of video, audio, animations, text and other media objects that are linked together on a website. It is a nonlinear medium and can be considered as an extension of hypertext. Hypertext and Hypermedia are used to access information on WWW and both organize and present contents on a website as shown in figure 1.2.

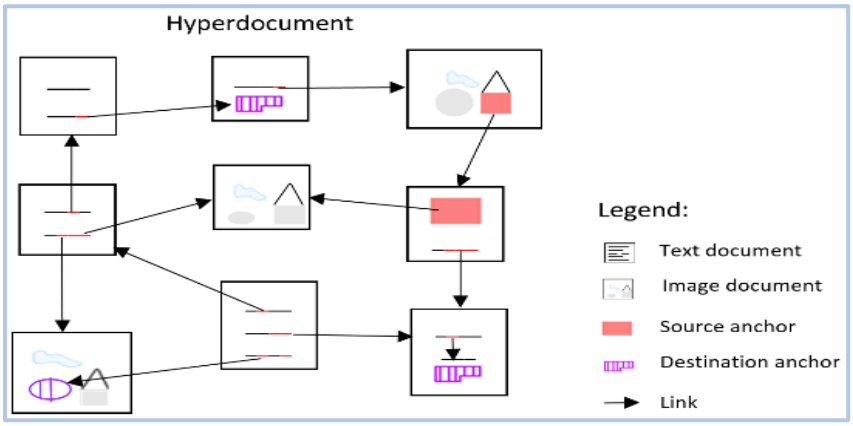

Figure 1.2: A simple structure of Hypertext document.

Hypertext is an electronic document where contents are connected using hyperlinks.

1.1.4. Hyper Text Transfer Protocol

Hyper Text Transfer Protocol (HTTP) is the foundation of data communication for the WWW. It is a stateless protocol, which is meant to transfer information on Internet and WWW. It defines how information is formatted before floating it over the network. For example, if we enter URL (address) of a web page in a web browser (client software), HTTP command is generated to the web server requesting it to fetch the web page. The server receives the request and processes it to generate the output. The response of the server is sent back to the client in the form of web page as shown in figure 1.3.

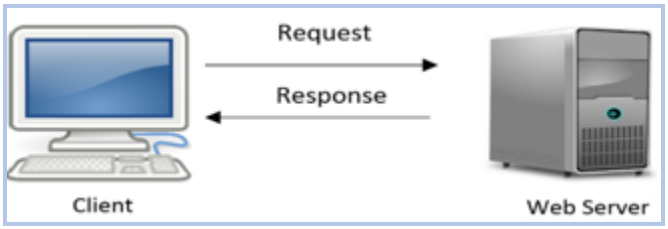

Figure 1.3: Client request and Server response model.

1.1.5. Client

Client refers to both software and hardware which retrieves information from a web server through a communication network including Internet as shown in figure 1.4. Our personal computer at home is client hardware and the browser running on it is the client software. Browser is a software application used to locate and display Web pages. Another example of client software is the email client which enables us to send and receive emails.

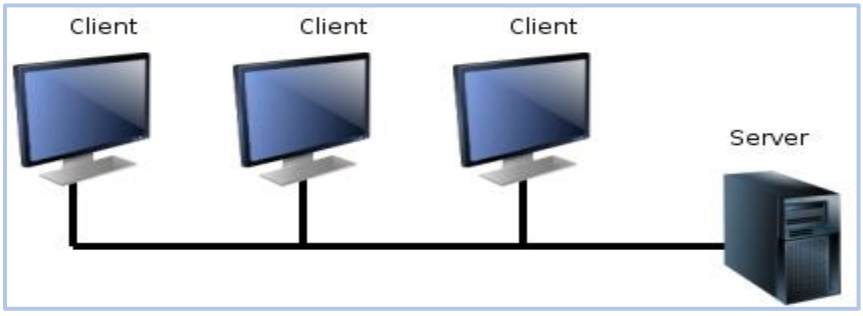

Figure 1.4: A communication network connecting clients with server.

1.1.6. Server

A server is a computer program or a device that provides services to other computer programs called clients. The device or machine that hosts a server program is also referred to as a server. That machine may be dedicated to provide services to different clients over a network. This arrangement is called client-server model where a single server can serve multiple clients, and a single client can use multiple servers. Examples of servers are database servers, file servers, mail servers, print servers, web servers, game servers and application servers.

Client is a program or system that sends request to the server. Server is a program or system that sends response back to the client.

1.1.7. Web Browsers

A browser is an application program that provides an interface to browse content on www. It is a client program that uses HTTP to send request to the web server. Few of the popular web browsers and their developer companies are shown in table 1.1.

Table 1.1: Commonly used web browsers

Browser Name	Developed By
Internet Explorer	Microsoft
Firefox	Mozilla
Flock	Capcom
Safari	Apple
Opera	Opera Software

1.1.8. Uniform Resource Locator (URL)

URL is an acronym that stands for Uniform Resource Locator and is a reference to a web resource that specifies its location on a computer network and a mechanism to retrieve it on the client's computer.

An example of a URL is http://www.aiou.edu.pk, which is the URL for the website of Allama Iqbal Open University, Pakistan. It has two important parts. The first part is "http" which is the protocol identifier and the second part "//aiou.edu.pk" is the resource name. The resource name specifies the IP address or the domain name where the resource is located.

1.1.9. Domain Name

Domain or domain name is used to identify a website. It is the location of a website on Internet. Normally websites are associated with IP addresses which are numerical values, difficult to remember by Internet users. In order to make things simple, domain names are used which are human readable format of IP addresses. For example, the domain name www.google.com locates the address for google website. The ".com" determines the purpose of the organization (commercial in this case) which is also called top level domain name. An organization called ICANN (Internet Corporation for Assigned Names and Numbers) authorizes reseller organizations to sell domain names.

1.1.10. IP Address or Number

An IP address (an abbreviation of Internet Protocol address) is an identifier assigned to each computer and other devices (e.g., printer, router, mobile device, etc.) connected to a TCP/IP network. Every computer that accesses Internet must have a unique identifying number. An IP address consists of four sets of numbers separated with a single dot(.). These numbers can range from 0 to 255 and each number contains one to three digits. For example, the IP address of your computer might be 192.168.0.4, which is shown in decimal form (readable by human) and in the binary form (readable by computer) used on Internet as given in table 1.2.

Table 1.2: Example of IP Address

Decimal	192.168.0 .4
Binary	11000000 10101000 00000000 00000100

Each number comprises of eight bits of storage and can represent any of the 256 numbers in the range between zero (binary 00000000) and 255 (binary 11111111).

Activity 1

1. Browse different websites and study their structure and layout.
2. Explore the concepts of domain and sub domain. What options are available to identify sub domain of a website?
3. Find IP address of your computer.
4. Download and configure different browsers on your system. Compare and contrast features.
5. Explore the concept of TCP/IP.

1.2. Characteristics of a Website

With the advancement in Information and Communication Technology (ICT), the WWW has become an important part of every organization. The availability of online information is expedited through

company's websites which contains necessary information for the potential users. The number of websites is growing exponentially as individuals, companies and organizations are floating their websites over Internet. The performance, quality and reliability of websites have become important. As a result, the development of website demands to follow a systematic approach.

It is important to notice that web development is different from traditional computer programming. Web applications have certain characteristics that need to be considered during the development process. The important characteristics include the following:

1. Web applications continue to evolve over time. They requires frequent changes in its structure and functionality.
2. Web applications contain multimedia components including text, audio, images and videos which are integrated with the procedural programming.
3. The design and visual presentation of a website is important to attract different users.
4. The community of website users is versatile including users from different fields of life.
5. There is a variety of delivery mediums for websites which includes PCs and mobile devices.
6. Security is also important to prevent websites from hacker's attack.

In order to cope with the above mentioned concerns, the development process should clearly define the steps to cater the real-time interaction, design and complexity of websites.

Activity 2

Surf Internet and explore the components of different educational websites.

1.3. Web Programming

The development of conventional software requires knowledge of traditional programming language in which the software is to be written. The well-known programming languages include C/C++ and Java which are high level computer languages. The syntax of these languages is similar to English like statements which require expertise of computer programmer's specific to syntax and semantics of a particular language. Using the syntax, a computer programmer writes a collection of instructions that performs a specific task. These instructions are called computer program. A computer program written in any high-level language is not suitable for web development because of the diverse nature of web applications.

Web development is associated with a different kind of programming paradigm called web programming. A person who programs a website is called a web programmer. It is the area which is concerned with writing of source code (computer program) to create a website. Web programming is sometimes called scripting. The source code is written in some scripting language. Web applications require multi-platform accessibility and employ a hypermedia paradigm. It may be static or dynamic depending upon the requirement of an organization.

Web Programming includes advance features and is different from the traditional style of coding.

There are two broad categories of scripting: client side and server Side. Client-side is the script which executes on the user's (the client's) computer and the server-side is the script which executes on a web server.

1.3.1. Client Side Scripting

The client side environment enables interaction within the web browser. The source code is downloaded from web server to client's computer and executed by the browser. An example of a client-side execution is a mouse over effect (typically triggered while choosing a navigation option), which runs on the user's machine and not on the server. The language JavaScript is used to script client side codes for web applications.

1.3.2. Server Side Scripting

The server side environment enables code to execute on the server machine. The requests of users are sent to server which executes a piece of code, generates output and sends the response back to the client. This response is utilized by the client computer to display the result for the user.

1.3.3. Static and Dynamic Websites

A static website is the simplest one website that does not allow frequent change in the contents. It is written in a plain HTML which displays the text on user computer using the web browser. A dynamic website is more complex where contents of website are changed frequently. It is written in HTML and some scripting language which dynamically updates the contents being requested by the user.

Activity 3

Browse some important websites available on Internet. Identify either they are static or dynamic.

1.4. Frontend and Backend Web Development

In website development domain, two terminologies are commonly used i.e. frontend and backend.

Frontend design

"Frontend" refers to the presentation layer that appears in the browser. It works on the client side and deals with the user interface of a website. The following tasks are commonly considered to be frontend tasks:

- Graphic design and image production.
- Interface design.
- HTML document and style sheet development.
- Client side scripting using client side programming languages.

Backend development

"Backend" refers to data access layer of a website. It works on the server side to make web pages dynamic and interactive. The following tasks take place on the backend:

- Information retrieval from server.
- Forms processing.
- Backend Database.
- Server side scripting using server side programming languages.

Activity 4

Explore set of web development tools for frontend and backend web development.

1.5. Web Application Development Process Model

Web applications have gradually evolved from static to dynamic nature keeping in view the needs of the potential users and advancement in technology. It is important to follow a systemic approach for the development of websites. An effective process model based on Software Engineering principles can help to follow this systematic approach especially for a website populated by framework activities. Roger Pressman has suggested a Web Engineering Process Model which is shown in figure 1.5.

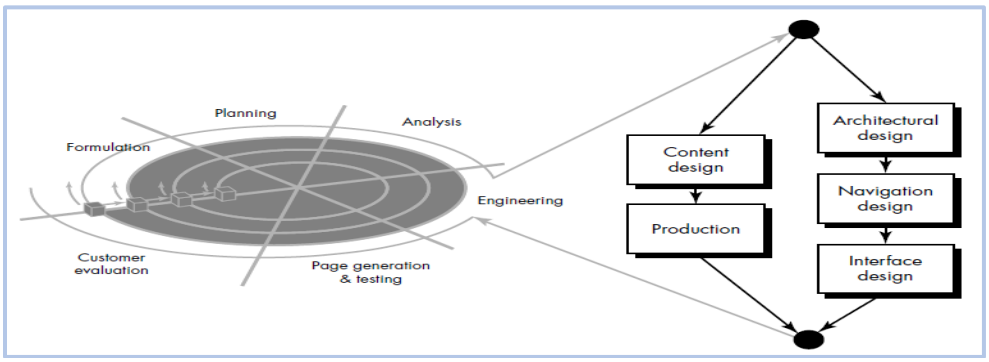

Figure 1.5: Web Engineering Process Model (Roger Pressman, 2014)

The process starts with the description of the objectives of web application and the introduction to its purpose. Then comes the planning stage which determines the development cost, resources required, risks associated and a schedule of development. Analysis determines the content items that will be displayed and also to work out the technical specifications of the web application.

The Web development process model is based on the basic principles of software engineering.

Both the activities of content and architectural design are carried out side by side. A team of technical and non-technical persons is involved in these parallel activities. The non-technical members acquire the text, graphics, audio and video contents. The technical team works on the aesthetic presentation with the blend of technology. The final content is merged with interface design and navigation controls are

defined at the same time. Page generation and testing is included in the incremental model. It helps to reveal the errors and ensure that application will execute correctly in different environments. The customer feedback is important which is carried out during the cyclic process. The changes suggested by the customer are incorporated in the next cycle through the incremental process.

1.6. Web Programming Technologies

As discussed in the previous section, Web programming Technologies are different from traditional programming languages. Let us see some important technologies related to web programming.

1.6.1. Programming Languages:

The important programming languages for web development are described below:

Preprocessor Hypertext (PHP)

PHP (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be easily embedded into HTML. The code of PHP is executed on the server side which distinguishes it from client side scripting. It is not only simple language but also has many advance features for the web development.

JavaScript

JavaScript is a popular programming language used in developing websites. This language was developed by Netscape. JavaScript is a powerful language which fully enables to add dynamic effects by incorporating client side scripting and control the display of content at run time by using the web browser. We will study more about JavaScript in unit 4, 5 and 6.

Swift

Swift is a powerful and intuitive programming language developed by Apple Inc. for iPhone Operating System (iOS) and Macintosh Operating System (OS X). It is a flexible programming language consist of modern programming features. Swift supports dynamic run time environment which makes the execution of programming code faster with the safety trade-offs.

Java

Java is a high level programming language released by Sun Microsystems in 1995. It is widely used to develop Internet applications and other software programs. Java can run on many different operating systems which make it platform independent. Java is a secure and reliable programming language.

Python

Python is a programming language that lets programmers do programming work more quickly and integrate applications efficiently. Python is an object-oriented programming language that has gained popularity because of its powerful syntax and readability. It is not only portable to multiple platforms but also easy to implement. It has a large collection of standard library function.

1.6.2. Frameworks:

A framework is a platform for the development of the software applications. It includes compilers, interpreters and code libraries used in software development. Some of the important frameworks are described below:

WordPress

WordPress is a Content Management System (CMS) to create, maintain and publish a website. It is an open-source CMS based on PHP and MySQL. It is installed on a web server which acts as a network host. WordPress allows the user to create and edit websites through a central administrative dashboard to modify the content and menus in the various design elements.

Bootstrap

Bootstrap is a front-end web application framework to develop websites. It has built-in design templates of HTML and CSS to create multiple layouts. Bootstrap supports responsive design which allows the dynamic change in layout of web pages for each device (PC, mobile or tablet) compatibility. It also consists of commonly used interface components that can be appended with HTML elements in a webpage.

Dot Net

".NET" Framework (pronounced dot net) is a software framework developed by Microsoft. It is designed to support the development of the next generation programming applications including the web services. Dot Net provides a large collection of tools and libraries to develop software applications in a much faster way.

Ruby on Rails

Ruby on Rails, sometimes known as "RoR" or just "Rails," is an open source framework to develop dynamic web applications. It facilitates the use of web standards such as XML for data transfer and also uses markup languages such as HTML and CSS to develop user interface.

Node. JS

Node.js is a cross platform and open source JavaScript runtime environment for building fast and scalable server side applications and tools. It is a lightweight and efficient tool which makes it suitable for data-intensive real-time applications.

AngularJS

AngularJS is a structural framework for the development of dynamic web applications. It uses HTML as template language and allows extending HTML's syntax to express application's components clearly and efficiently. We will study more about AngularJS in unit 8.

1.6.3. Libraries

Libraries are preprogrammed collection of code snippets that are used to develop software applications instead of writing it from the scratch. Libraries are designed to facilitate programming languages and software programmer to include a chunk of functional code in their software applications. jQuery is an important library for web application development.

jQuery

jQuery is a cross platform JavaScript library design to simplify the client side scripting of HTML. It is free open source and feature-rich JavaScript library. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. We will study more about jQuery in unit 7.

1.6.4. Databases

A database is a collection of data and information that is organized in systematic way so that it can be accessed easily. Data is organized in the form of special schemas called tables which are interlinked to form a relational database. Each table comprises of rows and columns where new data can be inserted, existing data can be updated in a systematic and easier way. Commonly used databases are MySQL, Oracle and SQL Server etc.

MySQL

MySQL is an open-source relational database management system. It can be used with all major operating systems including Linux, UNIX and Windows. MySQL is a popular backend database which is currently being used by Facebook, Twitter and YouTube.

Oracle

Oracle is a relational database management system (RDMS) developed and marketed by Oracle Corporation. This database application resides in a server and manages a large amount of data in multi user environment so that many users can concurrently access the same data. It also prevents unauthorized access and provides efficient solutions for failure recovery.

SQL Server

SQL stands for Structured Query Language. SQL Server is a Relational Database Management System (RDMS) developed by Microsoft. It is used to manage and store information. This database is used in Corporate IT environments for transaction, processing and business intelligence solutions. It also has some advance features like efficient buffer management and multi user support.

Activity 5

Study in detail the Web Programming Technologies and explore their potential usage in website development.

Unit Summary

This unit focused on an introduction to web programming. It described different concepts and configurations related to WWW. It further highlighted process model for both the development and improvement of a website. The latest technologies related to web development were also introduced.

Self Assessment Questions

Choose the correct answer.

1. What does WWW stand for?
   A. World Web Wide
   B. World Wide Web
   C. Web What Wide
   D. None

2. Are Internet and World Wide Web same?
   A. True
   B. False

3. HTTP is a stateless protocol, which defines _____
   A. client opens a program on user computer
   B. server runs a code and saves it on its drive
   C. information is formatted before floating it over the network
   D. URL is sent back to the client

4. In Uniform Resource Locator (URL), path is a pathname of the file where information is
   A. stored
   B. located
   C. programed
   D. transferred

5. Domain names are always read from the
   A. node up to root
   B. any node to root
   C. root to bottom
   D. All of them

6. An IP address consists of five numbers, each of which contains one to three digits, with a single dot
   A. True
   B. False

7. Web applications continue to_____ overtime. It require frequent changes in their structure and functionality.
   A. increase
   B. reduce
   C. evolve
   D. transform

8. The source code for web development is written in _____ language.
   A. scripting
   B. low level
   C. assembly
   D. programming

9. In web development, planning stage determines the _____
   A. development cost and URL
   B. domain name and IP address
   C. development cost and resources required
   D. hypertext and hypermedia

10. Graphic design and image production are part of _____
    A. frontend design
    B. backend design
    C. both ends design
    D. None of these

Answer Key:

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

Review Questions

Write short answers to the following questions

1. What is meant by World Wide Web?
2. What is the difference between hypertext and hypermedia?
3. Differentiate between client and server.
4. What are the characteristics of a website?
5. What is the difference between frontend and backend web development?

References and Further Reading

1. Pressman, R. S. (2014) 8th Edition. Software Engineering: a practitioner's approach. Palgrave Macmillan.
2. Hoffer, J. A. (2011). Modern Database Management, 10/e. Pearson Education India.
3. Sebesta, R. W. (2013). Programming the world wide web. Pearson Education India.
4. AngularJS. Available on: https://angularjs.org/
5. AngularJS. Available on: https://docs.mobiscroll.com/2-13-2/integration/angular
6. Bootstrap. Available on: getbootstrap.com/
7. Browser. Available on: http://searchwindevelopment.techtarget.com/definition/browser
8. Database. Available on: http://www.businessdictionary.com/definition/database.html
9. IP Address. Available on: http://www.livinginternet.com/i/iw_ip.htm
10. IP Address. Available on: https://en.wikipedia.org/wiki/IP_Address
11. jQuery. Available on: https://jquery.com/
12. Node.js. Available on: https://nodejs.org/
13. Oracle. Available on: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
14. PHP. Available on: php.net/manual/en/intro-whatis.php
15. PHP. Available on: http://www.dmelton.net/php/manual.html
16. Ruby on Rails. Available on: guides.rubyonrails.org/getting_started.html
17. SQL Server. Available on: http://www.databasejournal.com/features/mssql/article.php/3769211/What-is-SQL-Server.htm
18. WordPress. Available on: https://wordpress.org/


C·O·L

UNIT 2

HTML

Introduction

Hyper Text Markup Language (HTML) is an evolving language which was initially started with the small number of tags to develop static web pages. With the passage of time the requirement of information and business industry increased and the demand of more interactive web pages was felt. Keeping in view these needs, the later version of HTML had more tags and commands and the latest current version is the fifth version of HTML which has even more advance set of tags. These specifications and the international standards of the HTML are maintained by the Web Hypertext Application Technology Working Group (WHATWG) and World Wide Web Consortium (W3C). The specifications of HTML explain it as a sole standard language that is being followed universally. This unit will provide an overview of HTML. It covers basic syntax and some of the important tags of HTML which are necessary to learn Web Programming.

Unit Outcomes

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

1. Understand basic concepts of Hypertext Markup Language.
2. Use basic HTML tags to format your webpages.
3. Create hyperlinks to other documents.
4. Add images and multimedia to your pages.
5. Generate a basic form for taking user input.

Terminologies

HTML:	Hyper Text Markup Language: A standard language to create web pages.
Lists:	Group of items that are related to each other in a specific order.
Tag:	Tag specifies the format of a document, paragraph or text in a web page.
Plugins:	A component that adds specific features in a web browser.
DOM:	Document Object Model: It is a structured representation of an HTML document.

2.1. What is HTML?

HTML is the most frequently used language script to design web pages. The features which distinguishes it from other programming languages are its independence from any specific platform. HTML was created by Tim Berners -Lee in late 1991. The first standard HTML specification was HTML 2.0 which was published in 1995. Further, HTML 4.01 was a major version which was published in late 1999. The latest version of HTML is HTML5.

2.1.1. HTML Tags

HTML Tags are the basic building blocks of HTML. These are the markups used to identify content while designing web pages. The elements in HTML are written with a start tag and an end tag whereas the content of the block lies in between these tags. The start tag name is enclosed in angle brackets "<>". The end tag is also enclosed in angle brackets but has an additional slash "</>", to distinguish it from the start tag as shown in the following piece of code.

```
<tagname> web page content </tagname>
```

The names of HTML tags are used to denote labels such as bold, underline, italic, heading and tables, etc. Remember that HTML tags are not case sensitive.


HTML is a lanaguge used for the development of web pages.

2.1.2. Self-closing tags

It is not strict obligation to write closing tags in certain cases as a few HTML tags come with self-closing tags. For instance, line break tag is a self-closing tag which can be written as
. Both ways of writing self-closing tags either with </br> or without </br> are correct.

2.2. HTML DOM

DOM stands for the Document Object Model. HTML DOM depicts structured representation of the HTML page. It enables the programmers to edit, update, add and delete HTML elements. The HTML DOM comprises of two main parts: Head and Body. However, every HTML document starts and ends with html tags as shown by the following piece of code.

```
<html> Html web content </html>
```

The complete HTML DOM layout is shown in code 2.1.

2.2.1. Document Head

The document head is used to indicate the head of the HTML document. It is not dispalyed by the browser. The document head is written with the help of head tags and title tags are encapsulated within the head tags. The title tags display the desired title on the title bar of the web browser. For example we can create the title of a web page by using the following piece of code

```
<head> <title> Enter Web Page Title </title> </head>
```

</> Code 2.1

```
<!DOCTYPE html>
<html>
<head>
   <title>Page Title</title>
</head>
<body>
  <h1>This is Heading</h1>
  <p>This is a Paragraph</p>
  <p>This is another Paragraph</p>
</body>
</html>
```

Another important tag used under the document head is meta tags. Meta tags are used to specify keywords etc. for facilitating search engines optimization. You can learn more about it from the suggested reference materials.

2.2.2. Document Body

The document body refers to the entire section of an HTML document which is directly displayed by the browser on a web page. It starts with body tags and contains all other tags used to design a web page. The basic structure of HTML web document is shown in code 2.2.

</> Code 2.2

```
<!DOCTYPE html>
<html> <!-- Html tag  starts here -->
<head> <!-- head tag starts here -->
   <title>Enter Page Title Here</title>
</head><!-- head tag ends here -->
<body><!-- body tag starts here -->
   :
   :
</body><!-- body tag ends here -->
</html><!-- Html tag ends here -->
```

Note that <! DOCTYPE html> is always used at the start of the document. The comments can also be placed in the document enclosed within "<!" and ">".

2.3. Developing a Web Page

There are many editors available to design professional web pages. However, we will use a simpler approach of utilizing notepad (text editor) at initial level because hardcoding of tags will help to understand the functionality of HTML tags. Moreover, text files created in notepad are simple ASCII files that can be saved with ".htm" or ".html" file extension. For the view of html web page, you can use any web browser such as Internet Explorer, Google Chrome or Mozilla Firefox.

Any text file with .html extension can be used as a web page.

Follow the given steps in order to design your own web page.

Step 1: Open Notepad in Windows Operating System.
Step 2: Type your HTML code (given in the code 2.3) in Notepad. At this stage, you do not need to understand various HTML tags (written inside <>) used in the document.

</> Code 2.3

```
<! DOCTYPE html>
<html>
<head>
<title>Enter Page Title Here</title>
</head>
<body>
   <h1>This is my first web page </h1>
   <h2>Wish me best of luck for the future! </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). You will see the following screen as shown in figure 2.1.

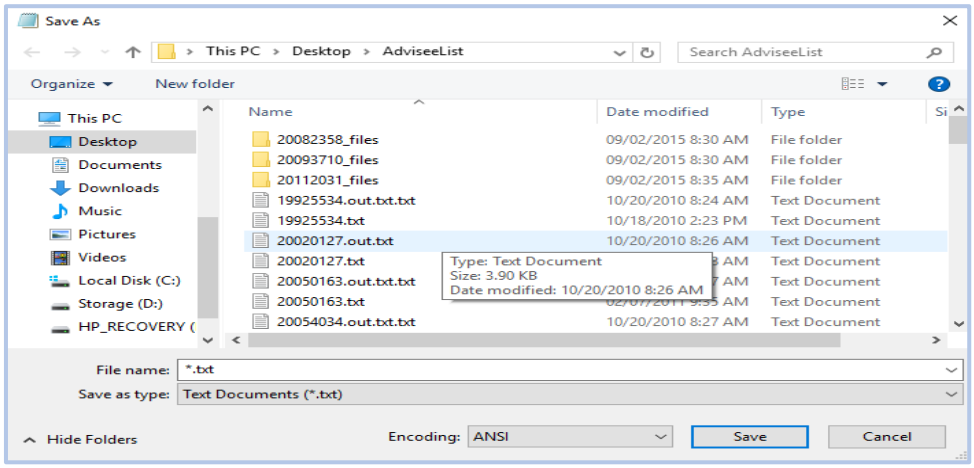

Figure 2.1: Save as screen shot

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. Your browser should display something similar to the following figure 2.2.

Figure 2.2: Output of code 2.3.

Let us now look at some more commonly used tags in HTML.

2.4. Commonly Used HTML Tags

HTML provides a wide variety of tags that can be used to format a web page. These tags can be utilized to make interactive and well-structured web pages. Some important tags are going to be discussed in the next sub section.

2.4.1. Header and Footer

The header tag is used to specify some important information at the top of the web page. This information explains what this web page is about. It can either be a heading or name of the author or logo of any company. The syntax is as follows:

```
<header> header information </header>
```

Similarly, the footer tag is used to specify some important information at the bottom of the web page. The syntax is as follows:

```
<footer> footer information </footer>
```

Both header and footer are defined inside the body tag.

2.4.2. Text Formatting

HTML provides a range of tags to format a text in a web page. Text formatting can be used to apply special effects to a piece of text such as font size, color, italic, bold, underline etc. You can use these tags to change the appearance of a piece of text in your web page. Some important tags are given in table 2.1.


HTML tags define how web browser format and display the content.

Table 2.1: Text formatting tags

Tags	Description
<b>	It is used to make the text appear bold
<i>	It is used to make the text look italic
<u>	It is used to underline the text
	It is used to make important tag look bigger than the rest of the text
	It is used to emphasize the text.
<h1>to<h6>	These tags are used for the headings according to their priority.
<mark>	It is used to mark the text
<p>	It is used to separate paragraphs.
<sub>	It is used to make the text subscript.
<sup>	It is used to make the text superscript.
 	It is used to insert a line break.

<!--. . .-->	It is used to add a comment.

Code 2.4 demonstrates the use of some important HTML tags.

</> Code 2.4

```
<!DOCTYPE html>
<html>
<head>
<title>Text Formatting</title>
</head>
<body>

   <p>Tags with their description are given below:</p>
   <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>
   <p><u>This text is underlined</u></p>

</body>
</html>
```

The output of code 2.4 is shown in figure 2.3, which displays the basic text formatting tags.

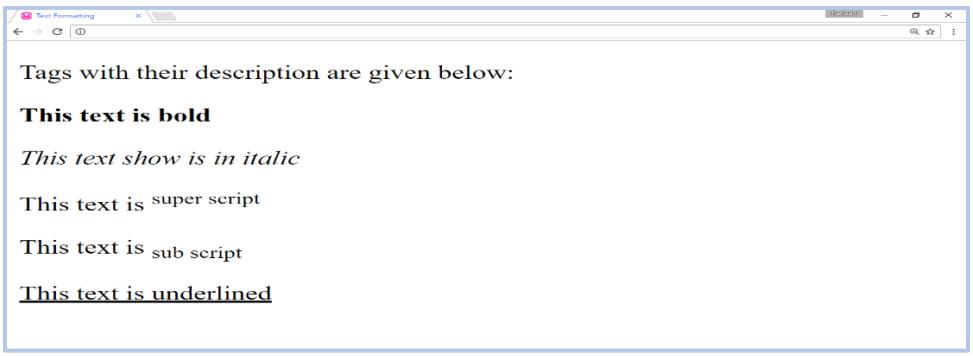

Figure 2.3: Output of code 2.4

Activity 1

Write HTML code to display basic information about your institute. Use different tags as described in table 2.1 to format the structure of your web page.

2.4.3. Paragraphs

Paragraph is a distinct portion of text indicated by new line. HTML allows arranging a piece of text in a paragraph format. The <p> tag is used to make paragraphing in an HTML web page. This tag formats the text from the new line as shown in code 2.5.

</> Code 2.5

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

```
<title>Paragraphs In HTML</title>
</head>
<body>
   <p>This text is written on first paragraph </p>
   <p>This is second paragraph</p>
   <p>This is third paragraph</p>
</body>
</html>
```

The output of code 2.5 is shown in figure 2.4.

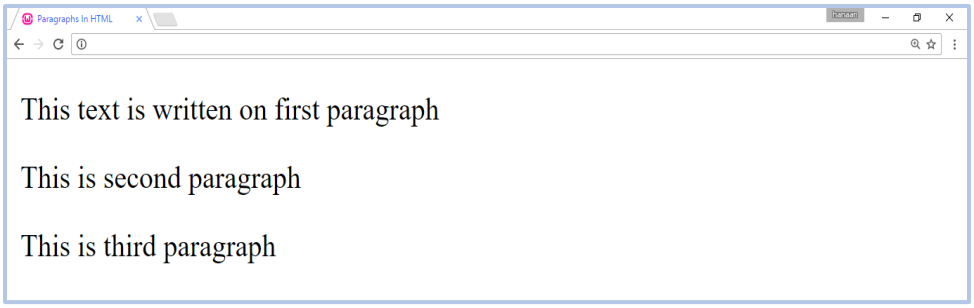

Figure 2.4: Output of code 2.5

Video Lecture

https://youtu.be/CHOoBInncc8

2.4.4. Text style

The text style is used to give attractive look to a piece of text in a web page. HTML tags use "style" attribute along with styling properties to give different styles of text. The syntax is as follows:

```
<tag_name style="property:value;"> web page content </tag_name>
```

Table 2.2 gives the summary of text styling properties.

Table 2.2: Text style attributes

Property	Explanation
background-color	It is used to change the background color of the text
color	It is used to change the text color
font-family	It is used to change the font writing style of the text
font-size	It is used to change the text size

text-align	It is used to change the alignment of the text to left, right and center.

Code 2.6 explains the use of all the above mentioned tags.

</> Code 2.6

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraphs In HTML</title>
</head>
<body style="background-color: lightblue;">
   <p>First Paragraph- normal text</p>
   <p style=" text-align: center;">Second paragraph - aligned
   center</p>
   <p style=" text-align: left;">Third paragraph - aligned
   left</p>
   <p style=" text-align: right;">Fourth paragraph - aligned
   right</p>
   <h1 style="color: red;"> Red color font is selected for
   heading </h1>
   < b style=" font-family: verdana; "> Verdana is used as font
   family under Bold tag </b>
</body></html>
```

The Output of code 2.6 is shown in figure 2.5.

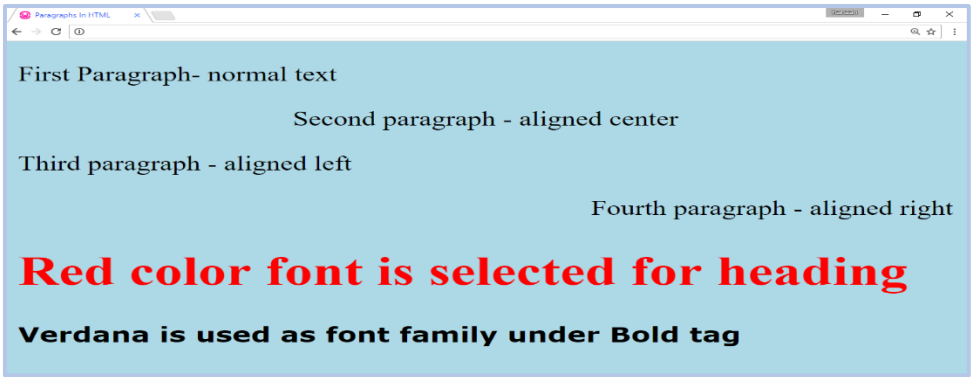

Figure 2.5: Output of code 2.6

Activity 2

Apply different text styles discussed in table 2.2 to the webpage created in Activity 1.

2.4.5. Lists and Bullets

Lists and Bullets are used to organize set of data in a categorical form. There are three types of lists available in HTML:

Unordered Lists (Bullets)

The unordered list is without any specific sequence or order. Bullets are used to indicate unordered lists of items.   tags are used to indicate the starting and ending of the unordered list while every item is enclosed within <li> tags as shown below:

```
<ul>
    <li> unordered list item 1 </li>
    <li> unordered list item 2 </li>
</ul>
```

Ordered Lists (Numbering)

The ordered list is a numbered list of items with a specific sequence or order. The <ol> tags are used to indicate the starting and ending of the ordered list while every item is enclosed in <li> tags as shown below:

```
<ol>
    <li> ordered list item 1 </li>
    <li> ordered list item 2 </li>
</ol>
```

Unordered list arrange set of related items without any sequence while ordered list keep the the sequence in order.

The code 2.7 demonstrates the use of ordered and unordered list items.

</> Code 2.7

```
<!DOCTYPE html>
<html>
<head>
<title>Ordered and Unordered list items </title>
</head>
<body>

   <ol>
     <li>Fruits:</li>
     <ul>
      <li>Mango</li>
      <li>Apple</li>
      <li>Banana</li>
      <li>Peach</li>
      <li>Grapes</li>
      <li>Water Melon</li>
     </ul>

      <li>Vegetables:</li>

     <ul>
      <li>Potato</li>
      <li>Onion</li>
     </ul>
    </ol>
```

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

The output of code 2.7 is shown in figure 2.6. The output clearly shows the difference between ordered and unordered list.

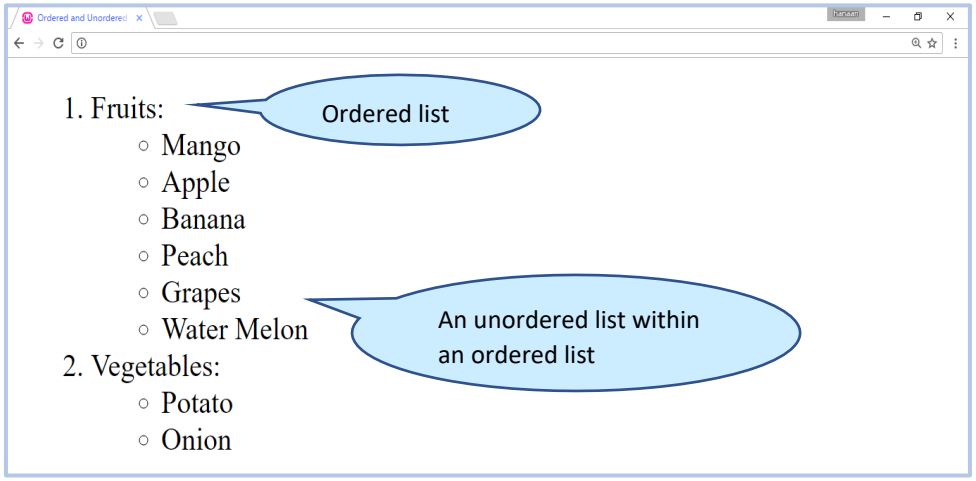

Figure 2.6: Output of code 2.7

Activity 3

Write HTML code to create the following lists:

1. Working days
   - Monday
   - Tuesday
   - Wednesday
   - Thursday
   - Friday
2. Off days
   - Saturday
   - Sunday

Definition list

The definition list is used to add description with every list item. The <dl> </dl> tags are used at the starting and ending of the definition list. <dt> </dt> tags are used to write definition/data term while <dd> </dd> tags are used to define description of the data terms. The syntax of definition list is as follows:

```
<dl>
 <dt>Term</dt>
 <dd>Description of the term</dd>
</dl>
```

The definition list is used to give description of every list item.

Code 2.8 describes the use of different tags in definition lists.

</> Code 2.8

```
<!DOCTYPE html>
<html>
<head>
<title>Ordered and Unordered list items </title>
</head>
<body>
  <b> Demonstration of description List tags</b>
  <dl>
    <dt>Coffee</dt>
    <dd>Brown hot drink</dd>
    <dt>Water </dt>
    <dd>Transparent cold drink</dd>
    <dt>Tea </dt>
    <dd>Milky hot drink</dd>
  </dl>
</body>
</html>
```

The output of code 2.8 is given in figure 2.7.

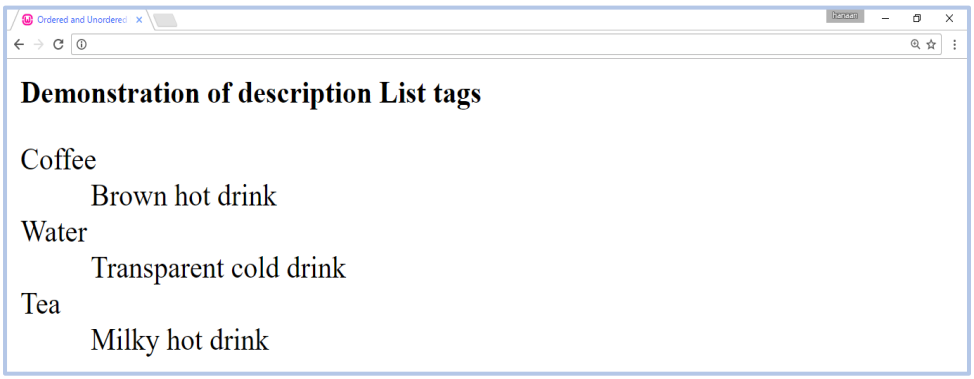

Figure 2.7: Output of code 2.8

2.5. Create Tables in HTML

A table is a data structure that arranges information in rows and columns. HTML enables the users to make web pages more structured by using tables. <table> </table> tags are used to create tables in HTML. Each table usually has a caption, heading and table data.

2.5.1. Components of table

The major components of the table are defined as under:

Table caption

The caption is the title of the table. Caption tag is used to display this title in a web page.

Table heading

The table heading is the heading of each column. <th> tag is used to incorporate this heading in the table.

Table Rows

The rows are the horizontal group of cells in a table. The <tr> </tr> tags are used to define the starting and ending of one row.

Table Data

The table data is used to enter content into a table cells. <td> </td> tags are used to indicate the starting and ending of these contents in a table cell. Code 2.9 shows an example of table creation in HTML.

</> Code 2.9

```
<!DOCTYPE html>
<html>
<head>
<title>Tables</title>
</head>
<body>
   <table>
   <tr>
     <th>First Name</th>
     <th>Last Name </th>
     <th>Age</th>
   </tr>
   <tr>
     <td>Ali</td>
     <td>Ahmad</td>
     <td>20</td>
   </tr>
   <tr>
     <td>Zahid</td>
     <td>Khan</td>
     <td>22</td>
   </tr>
  </table>
</body>
</html>
```

The output of code 2.9 is shown in figure 2.8.

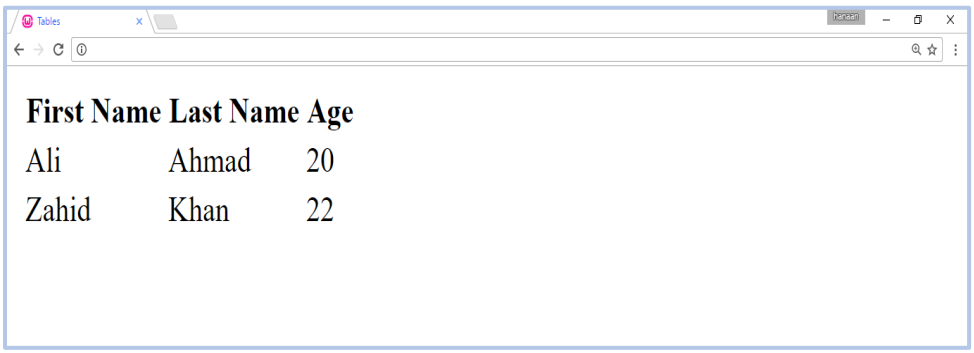

Figure 2.8: Output of code 2.9

2.5.2. Border Attribute

The default representation of tables in HTML is without borders. Therefore, border attribute is used to show a table border. It is defined within the tags of table. The syntax is as follows:

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

2.5.3. Width and Height Attribute

The width attribute is used to set width of a table or width of table cells. It can be specified in pixels or percentage. If width is not given the length of longest word in the cell is considered as its width by default. The syntax is as follows:

```
<table style= "width:100%;">
```

% sign is used to indicate the width in percentage while simple figure value indicates pixels. Similarly, the height attribute can be used to set the height of the table or individual data cells. However, the height attribute is not recognized by some of the browsers.

2.5.4. Align Attribute

The align attribute is used to set the position of table relative to the other elements on a web page. A table can be aligned left, right or centered. The syntax of the align attribute is as follows:

```
<table align = "center">
```

2.5.5. Cell Padding and Cell Spacing Attributes

The cell padding is used to set space between the cell borders and the content inside the cell. The cell spacing is used to set space among the cells. The syntax is as follows:

```
<table cellpadding = "pixels">

<table cellspacing = "pixels">
```

2.5.6. Column Span and Row Span Attributes

The column span attribute is used to merge two or more columns Whereas the rowspan does the same for two or more rows. The syntax of column span and row span is as follows:

```
<td colspan =" 3">

<td rowspan =" 3">
```

Code 2.10 demonstrates the use of table tag and its related attributes.

</> Code 2.10

```
<!DOCTYPE html>
<html>
<head>
<title>Tables 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>Zahid</td>
      <td>50</td>
      <td>75</td>
      <td>85</td>
     </tr>
     <tr>
      <td>Sana</td>
      <td>77</td>
      <td>82</td>
      <td>45</td>
     </tr>
    </table>
</body>
</html>
```

The output of code 2.10 is shown in figure 2.9.

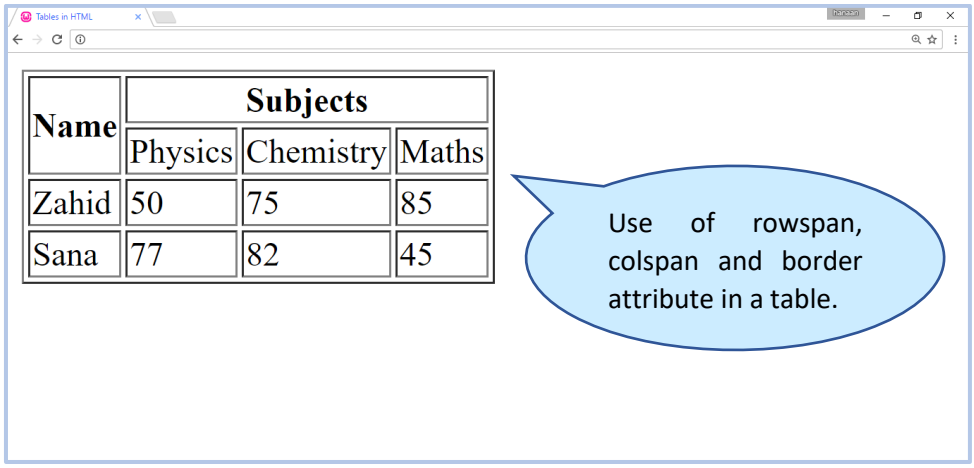

Figure 2.9: Output of code 2.10

Tables are frequently used to arrange and structure the contents in the web pages.

Activity 4

In the code 2.8 add some more rows and columns in your table and see the output. Also change the border properties to observe the difference in border width.

2.6. Inserting Images in HTML

HTML allows its users to insert images to their web page. The <img> tag is used for this purpose. The syntax as follows:

```
<img src = "logo.jpeg">
```

2.6.1. Using the ALT Attribute

The "alt" attribute is used to indicate alternative label for the image inserted in HTML web page. It is not displayed while the image has shown successfully and makes no significant changes. However, if the image is not displayed, alternative labels are shown in order to specify that something is missing or image is not loaded properly due to some reason. The syntax is:

```
<img src = "logo.jpej" alt = "Error in Downloading">
```

2.6.2. Image Height and Width

The attributes of height and width are used to adjust the dimensions or the size of the image. The syntax is shown below:

```
<img src = "logo.jpeg" height = "300px" width = "300px">
```

Code 2.11 demonstrates an example of image height and width.

</> Code 2.11

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

The output of code 2.11 is shown in figure 2.10.

2.7. Hyperlinks

A Hyperlink or simply a link is a location in a document on which user can click to access another file or object. Hyperlink can appear as text (usually underlined to make it stand out), as an image like a button or a picture representing the destination page. The syntax of hyperlink is as follows:

```
<a href = "index.html"> Home </a>
```

“index.html” is the name of the page that you want to load. “Home” is the caption of the hyperlink.

A hyperlink connects one web page to another document or different location of the same document.

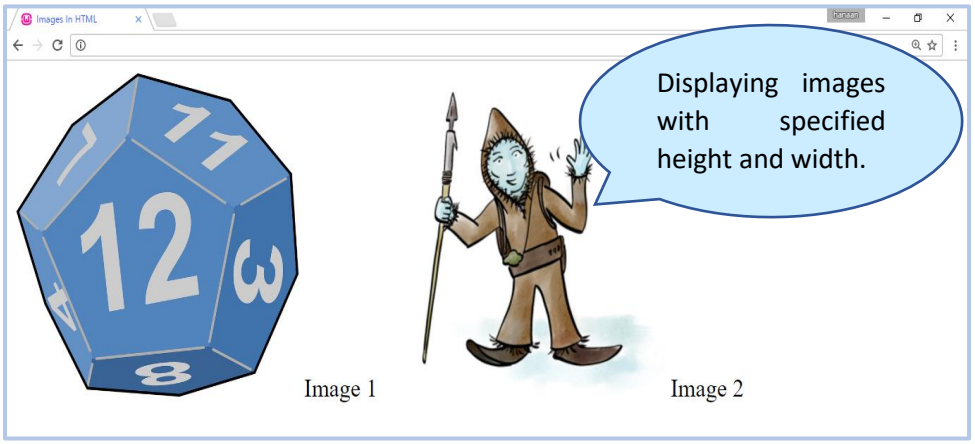

Figure 2.10: Output of code 2.11

2.7.1. Hyperlink of an email

Similar to the links of website, a hyperlink to any email allows sending electronic mail to an email address. It enables the user to open an email program and fills the "To:" field with any e-mail address. It provides a way to show any email address as a text link option so that the user feels no need to remember the email address. The example of email hyperlink is as follows:

```
<a href = "mailto:colp@aiou.edu.pk"> AIOU </a>
```

2.7.2. Hyperlink to another browser page

To open a link in a new browser window an extra attribute of “target” is added in href tag. Like:

```
<a href = "index.html" target = "_blank"> Home </a>
```

Activity 5

Create a web page that contains the links of your favorite travel places.

2.8. HTML Multimedia

Multimedia files which include audio, images, animations and videos can also be integrated effectively in an HTML file. “<audio>” tag is used to incorporate audio files in the HTML. The syntax is as follows:

```
<audio> … </audio>
```

The audio tags can further be enhanced with additional controls like increase/ decrease volume, pause and play options etc. The “Control” attribute can be added with audio tag as shown in the following example:

```
<audio controls>
   <source src = "1.mp3" type = "audio/mpeg">
</audio>
```

Audio controls can be incorporated as shown in code 2.12.

</> Code 2.12

```
<!DOCTYPE html>
<html>
<head>
<title>Audio in web page</title>
</head>
<body>
   <h2>Adding Audio in a web page</h2>
   <audio controls>
     <source src = "1.mp3" type = "audio/mpeg">
   </audio>
</body>
</html>
```

The output of code 2.12 is shown in figure 2.11.

Figure 2.11: Output of code 2.12

In the previous example, the user can play/pause the audio file and control the volume. If you want to let the audio play automatically at the startup, “autoplay” option of the attribute can be used as shown in code 2.13.

</> Code 2.13

```
<!DOCTYPE html>
<html>
<head>
<title>Auto play Audio in web page</title>
</head>
<body>
   <h2>Auto play Audio in a web page</h2>
   <audio autoplay>
     <source src = "1.mp3" type = "audio/mpeg">
```

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

Similarly, the video tags are used to display video as shown in the following syntax:

```
<video controls>…</video>
```

The "controls" attribute can configure the height and width of video display window as shown in the following code 2.14.

</> Code 2.14

```
<!DOCTYPE html>
<html>
<head>
<title>Video in web page</title>
</head>
<body>
   <h2>Add Video in a web page</h2>
   <video width = "320" height = "240" controls>
     <source src = "2.mp4" type = "video/mp4">
   </video>
</body>
</html>
```

The output of code 2.14 is shown in figure 2.12.

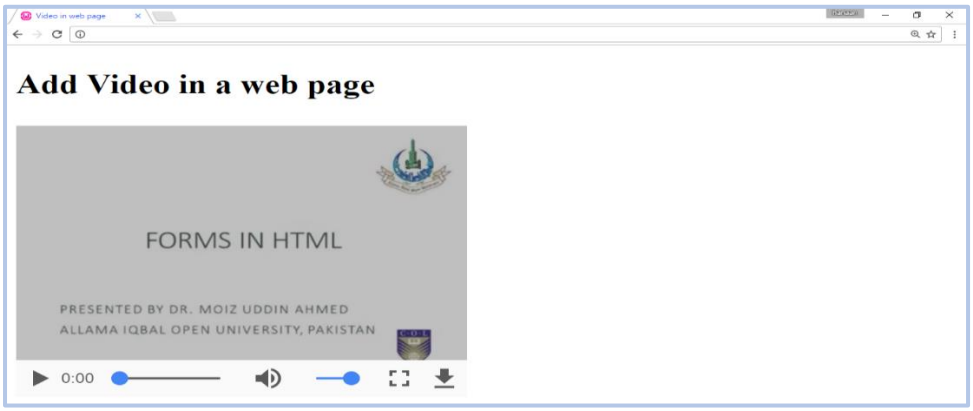

Figure 2.12: Output of code 2.14.

2.8.1. HTML Plugins

"Plugins" are the tools to enhance the functionality of any web browser. This addition can be implemented by using some advanced features which are not already existing for example displaying of maps and verification of bank accounts etc.

Object tag is now being used to add a new multimedia like "swf" in a web page. The complete syntax is as follows:

```
<object width = "500px" height = "100px" data = "sample.swf">
</object>
```

To show any webpage inside the main web page, “<embed>” tag is used as demonstrated in the following example:

```
<embed src="sample_img.html" width = "100%" height = "400px">
```

To incorporate YouTube in your web page, you can copy the link of that video from YouTube as shown in figure 2.13. Afterwards the same link can be pasted in your HTML code as shown in code 2.15. Note that the “<iframe>” tag is embedding a small window of given size inside your main HTML page.

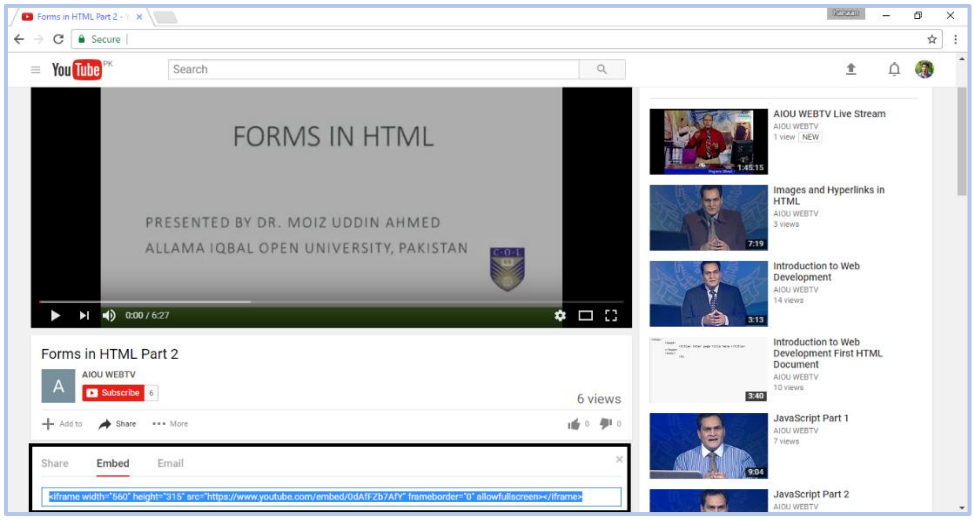

Figure 2.13: Embed video in your HTML code.

</> Code 2.15

```
<!DOCTYPE html>
<html>
<head>
<title>Video in web page</title>
</head>
<body>
   <h2>Embed Video in a code</h2>
   <iframe width = "560px" height = "315px" src =
    "https://www.youtube.com/embed/0dAfFZb7AfY" frameborder =
    "1">
   </iframe>
</body>
</html>
```

The output of code 2.15 is shown in figure 2.14.

You can embed different media objects in your web page.

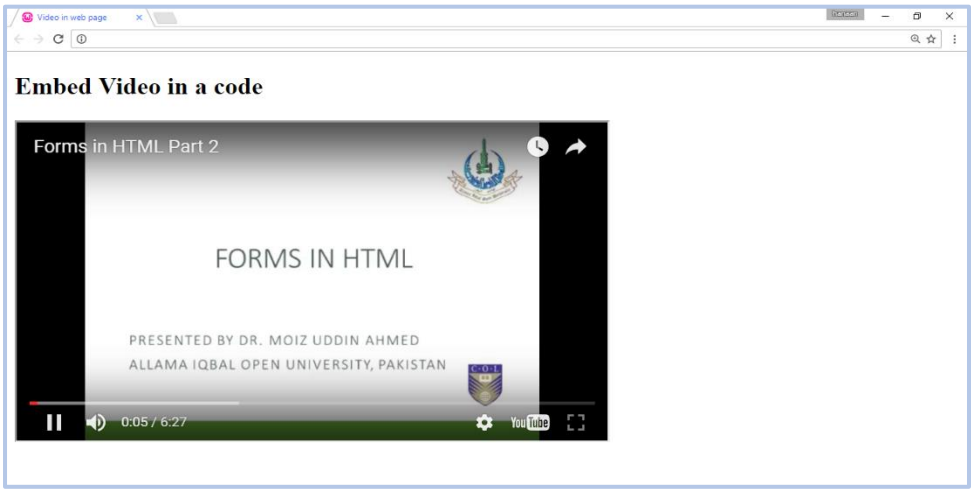

Figure 2.14: Output of code 2.15.

Activity 6

Create a simple web page and add plugins to read different multimedia objects (audio, videos etc.)

Video Lecture

https://youtu.be/nmh3vSmf1a4

2.9. HTML Form

HTML Form can be used to take input from the user. This form comprises of different types of input elements such as text fields, check boxes, radio buttons etc. A basic form is demonstrated in the code 2.16 in which some information is being taken from the user for further processing. The output of this code is shown in figure 2.15. The "submit" button will submit all the information entered by the user to the server.

</> Code 2.16

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Forms</title>
```

```
</head>
<body>
<h2>HTML Form </h2>
  <p> Please fill the form below: </p>
  <form name="myForm" action="" method="post">
    <table border = "1">
    <tr>
      <td>Name: </td>
      <td><input type = "text" name = "name" ></td>
    </tr>
    <tr>
      <td>Address: </td>
      <td><input type = "text" name = "address" ></td>
    </tr>
    <tr>
      <td>Email: </td>
      <td><input type = "email" name = "email" required></td>
    </tr>
    <tr>
      <td>Zip Code: </td>
      <td><input type = "text" name = "zip"></td>
    </tr>
    <tr>
      <td>Gender: </td>
      <td><input type = "radio" name = "male" checked >Male </td>
    </tr>
    <tr>
    <td></td><td><input type = "radio" name = "female"> Female
    </td>
   </tr>
   <tr>
     <td>Subscription</td>
     <td><input type = "checkbox" name = "news" id = "news" value
      = "news">Newsletter</td>
   </tr>
   <tr>
     <td></td>
     <td><input type = "checkbox" name = "mag" id = "mag" value =
     "mag"> Magzines</td>
   </tr>
   <tr>
     <td>City</td>
     <td>
       <select name = "city">
          <option value = "-1" selected>Select </option>
          <option value = "1" >Islamabad </option>
          <option value = "2" >Lahore </option>
          <option value = "3" >Karachi </option>
       </select>
     </td>
   </tr>
   <tr>
     <td></td>
     <td><input type = "button" name = "submit" id = "submit"
      value = "submit"></td>
   </tr>
</table>
```

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

The output of code 2.16 is shown in figure 2.15.

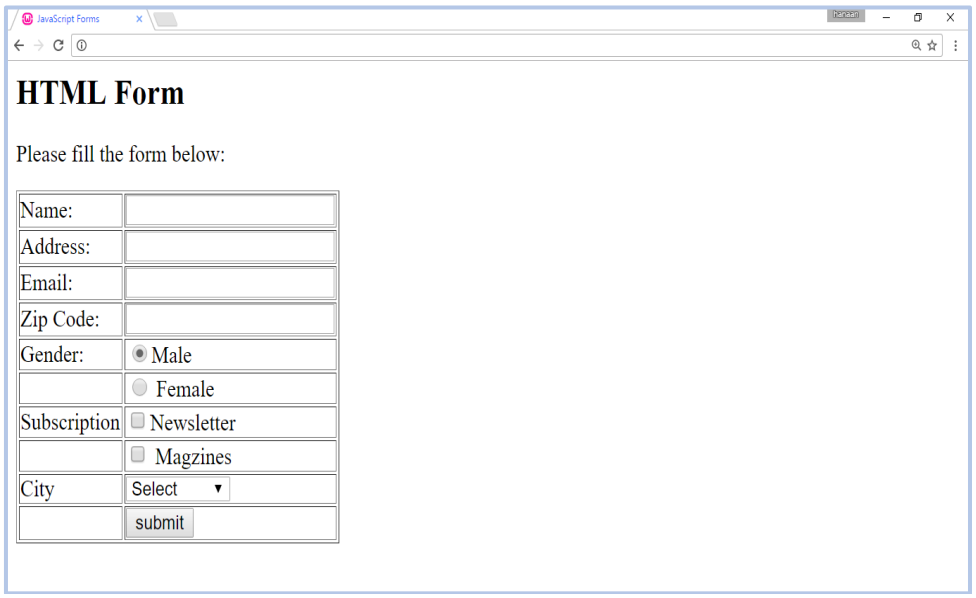

Figure 2.15: Output of code 2.16.

Activity 7

Create a web page that can take pizza order from the user. The form may comprise of name and address as text fields, pizza size (small, medium, large) as radio buttons, pizza toppings (cheese, mushrooms, pepperoni) as check boxes and phone number as number field.

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. Is <! DOCTYPE > declaration essential to start writing HTML document?
   A. No.
   B. Not matter.
   C. Yes, always.
   D. None of the above.

2. Which tag is used to start the content area of HTML document?
   A. <html>
   B. <body>
   C. <title>
   D. <p>

3. DOM stands for
   A. Document Object Model
   B. Demand Object Model
   C. Document Operation Model
   D. None of the above

4. If the image is not displayed in the browser due to some error, the _____ specifies an alternate text for image?
   A. text attribute
   B. caption attribute
   C. value attribute
   D. alt attribute

5. _____ is used as a default value of “target” attribute in HTML.
   A. _top
   B. _parent
   C. _blank
   D. _self

6. We can use _____ tag for showing headings in HTML.
   A. article
   B. strong
   C. heading
   D. paragraph

7. Which of the following is a style tag?
   A.
   B. <b>
   C. <sup>
   D. All of the above

8. Which of the following is used to insert a row in a table?
   A. <td> and </td>
   B. <tr> and </tr>
   C. <th> and </th>
   D. None of the above.

9. The object tag is used to _____
   A. display client information
   B. display images and text in a web page
   C. add a plugin in a web page
   D. None of the above

10. Which tag is used to insert a form in a web page?
    A. <select> and </select>
    B. <form> and </form>
    C. <table> and </table>
    D. None of the above

Answer Key:

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

Review Questions

Write short answers to the following questions.

1. Describe the basic concept of HTML.
2. Briefly explain the structure of HTML DOM?
3. Which tags are used to bold, italic and underline a piece of text in HTML?
4. Differentiate between ordered and unordered lists.
5. How can you add a space between cell border and the content inside the cell?
6. What are the functions of rowspan and colspan in HTML tables?
7. What is the use of plugin in HTML?
8. Briefly describe different input types available in HTML forms.

</> Coding Exercise

1. Create a web page that gives basic information about your hometown. The page should display some interesting facts and figures in a well-designed and attractive way. Also insert some images in your web page.
2. Create a web page about your favorite topic. Insert some interesting YouTube videos in a structured way.
3. Create a basic admission form in HTML which takes basic information like name, father name, postal address, gender (using radio buttons), city and program (using dropdown list), telephone number and email address.

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 Plugins. Available on: http://www.tutorialspoint.com/html/html_tutorial.pdf
7. HTML Tables. Available on: http://www.tutorialspoint.com/html/html_tables.htm
8. HTML Lists. Available on: http://www.c-sharpcorner.com/UploadFile/75a48f/html-for-beginners-part-3/
9. HTML Elements. Available on: https://en.wikipedia.org/wiki/HTML_element
10. Styling Tables Using CSS. Available on: http://teamtutorials.com/web-development-tutorials/styling-tables-using-css
11. HTML Tutorial. Available on: https://www.w3schools.com/html/

UNIT 3

Basics of CSS

Introduction

HTML was considered as a language to depict the logical structure rather than the appearance of documents. As the WWW turned out to be more conventional, end users began to exploit HTML labels to control the presentation. In early 1990s, HTML was still lacking the important structural components of a document to a great extent. These components included lists, heading, hyperlinks etc. To fulfill the needs of people and control the appearance of web pages and websites, Cascading Style Sheets (CSS) were introduced. CSS is used to portray styles for your website pages, including their arrangement, configuration and assortments in showcase for different devices and screen sizes. They are an approach to control the look and feel of your HTML reports in a composed and productive way. With CSS, you will have the capacity to add new look to your old HTML; totally restyle a site with just a couple of changes to your CSS code, utilize the "style" you make on any website page.

CSS is a style sheet language that deals with the presentation of the content.

CSS has improved the incorporation of many attractive features in HTML like control of text color, background color, borders, spacing among elements, text manipulation and decoration. Everything which was possible in HTML is possible in CSS with more features. To understand the basic idea of how to use CSS and how powerful it is, a simple example is required. This unit will provide an introduction to CSS and how we can use CSS to improve the structure and presentation of our web page.

Unit Outcomes

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

1. Use different methods of writing CSS.
2. Understand the basic properties of CSS.
3. Write comments to explain CSS code.
4. Use box model and control the opacity of an image.
5. Insert an image gallery using CSS.

Terminologies

CSS:	Cascading Style Sheet is a design language used to give different styles to a web page
Selectors:	Selectors are used to select different elements in CSS to further apply different properties.
Property:	An attribute or characteristics of an object to configure its appearance.
Navigation:	Different style of navigation menus can be used in CSS.
Image Sprite:	Collection of images that merge into a single image.

3.1. Methods of Writing CSS

There are three main methods of writing CSS: inline, internal and external CSS. The details are going to be discussed in the next sub section.

3.1.1. Inline CSS

Inline CSS is used to give style to a certain HTML tag directly. The "style" tag is used to control the style of each element in an HTML document. It needs extra effort as every new style has to be defined in the relevant tag to change the inline style. The following code segment highlights the example of using inline CSS by defining font size, color and text alignment:

```
<h1 style = "font-size: 20px; color: red; text-align: center;">
Introduction </h1>
```

3.1.2. Internal CSS

The internal style sheet is used to give a unique style to a single HTML document. The CSS code is defined in the head section of the HTML page where the required CSS property and the style attributes are defined once. Any change may affect the relevant HTML page in which the style is defined. The following code segment highlights the example of using internal CSS:

```
<html><head> <style type ="text/css"> ... </style></head></html>
```

3.1.3. External CSS

In external style sheet, the CSS code is written in a separate CSS file which is then linked with the main HTML page. This method is used when we want to give a style to the entire website. The required changes are incorporated in the CSS file. The link tag is used to specify the external file which is defined in the head section of the HTML page. The following code segment highlights the example of using link tag for external CSS:

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

Inline style is used to style a small piece of text and internal style sheet is used to style the whole document whereas external style sheet is used to style many web pages connected to a single website.

3.2. Selectors in CSS

Selector is the basic building block of CSS syntax, which defines rules to give styles to HTML elements. The rules set consists of combination of selectors and declarations. Each declaration consists of a property and

a value which configures the property. There are three types of selectors in CSS. They are being explained in detail below:

- Element Selector
- ID Selector
- Class Selector

Activity 1

Create a simple webpage and display the following information

Brief History of AIOU

The Allama Iqbal Open University was established in May 1974, with the main objectives of providing educational opportunities to masses and to those who cannot leave their homes and jobs. During all these years, the university has fulfilled this promise.

Change the font color and size of above text by using Inline CSS method. See the output in your browser and observe the effects of CSS.

3.2.1. Element Selector

Element selector specifies the style of an element with the given tag name. Using element selector, we can select every element of a paragraph by using <p> tag on a single page. For example, if we want to change the style of all paragraphs, we need to define a style for <p> as shown below:

```
p
{
    color: white;
    text-align: center;
 }
```

3.2.2. ID Selector

ID selector is used to select a specific HTML element with the given id attribute. The id is a unique identifier for an element which is specified by a (#) character. For example, if we want to change the style of a specific paragraph in a web page, we should define the style as shown below.

```
#para1
{
   text-align: center;
   color: red;
 }
```

"#para1" is the name of the style which we can apply on a paragraph as shown below:

```
<p id="para1">This line is written on second paragraph. </p>
```

3.2.3. Class Selector

Class selector selects all elements with the given class attribute. To select an element of a specific class a dot (.) character is used with the class name. For example, if we want to change the text alignment and color style of more than one elements in a web page, we should define the style as shown below.

```
.para1
{
     text-align: center;
     color: red;
  }
```

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

```
<p class="para1">This is a Paragraph </p>
<p class="para1">This is Second Paragraph </p>
```

Code 3.1 demonstrates the use of these selectors in CSS.

</> Code 3.1

```
<!DOCTYPE html>
<html>
<head>
<title>Selectors Example</title>
<style>
   p
   {
      font-size: 25px;
    }
   .p1
    {
      border: 2px solid blue;
      font-size: 25px;
      color: black;
      text-align: center;
    }
   #p2
    {
      border: 2px solid #673ab7;
      font-size: 25px;
      color: black;
      text-align: justify;
    }
   h2
    {
      color: blue;
    }
</style>
</head>
<body>
   <h2>Element Selector</h2>
```

```
    <p>We give font color "blue" to all heading tags</p>
    <h2>Id Selector</h2>
    <p id = "p2">With the help of id selector, we give background
    color "purple" and font color "black" to specific paragraph. </p>
    <h2>Class Selector</h2>
    <p class = "p1">With the help of class selector, we give border
    color "blue" and font color "black" to this specific paragraph.
    </p>
    <p class = "p1">This is another paragraph in class selector, you
    will notice that both paragraphs written in class selectors have
    same styles. </p>
</body>
</html>
```

The output of code 3.1 is shown in figure 3.1.

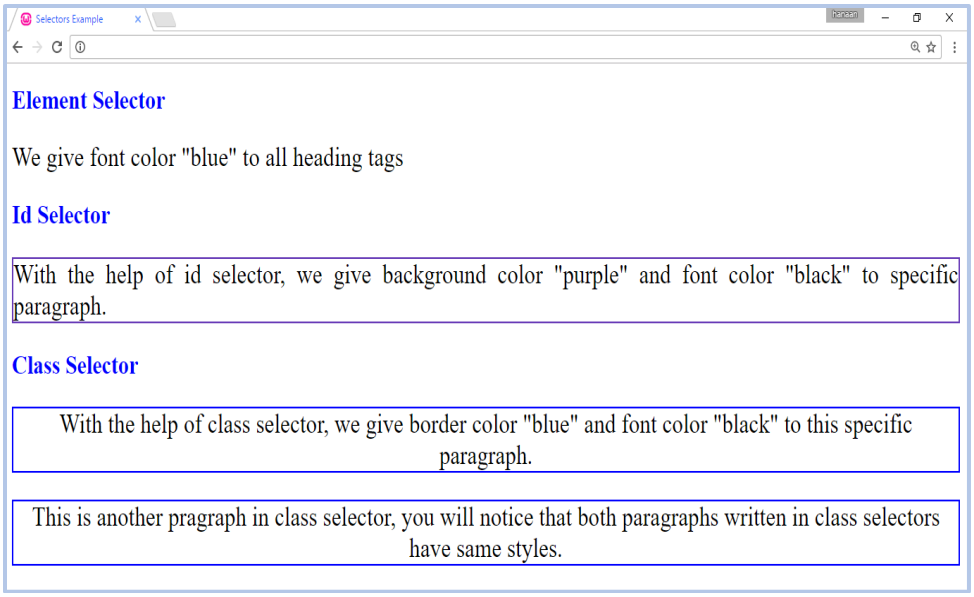

Figure 3.1: Output of code 3.1

Class selector is used to style multiple elements whereas ID selector is used to style a single element.

Activity 2

Modify the web page created in activity 1. Change the attributes (font size and color) of the paragraph by using ID selector.

3.3. Basic Properties of CSS

The basic process of forming the rules have been discussed in the previous section. However, CSS defines a large number of properties and their values. Some of the basic properties are:

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

We will discuss how to set the above properties in the following sub sections.

3.3.1. Background Properties

Background properties are used to define the background effects on elements. Some of the background properties are given in the table 3.1.

Table 3.1: background properties with description

Property	Description
background-color	It specifies the background color 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 scroll with the rest of the page
background-position	It specifies the position of the background images

Code 3.2 demonstrates the use of background properties.

</> Code 3.2

```
<!DOCTYPE html>
<html>
<head>
<title>Background Properties</title>
<style>
   body  /* Set background properties in body */
   {
      background-image: url("schemo.png");
      background-color: white;
      background-repeat: no-repeat;
      background-position: top right;
    }
   p
```

```
   {
      font-size: 25px;
     }
</style>
</head>
<body>
   <p>We give background image and set this image to top right side
    <br> of the page with the help of background properties. </p>
</body>
</html>
```

The output of code 3.2 is shown in figure 3.2

Figure 3.2: Output of code 3.2.

Activity 3

Modify the code 3.2 by changing the "background color" of the document to red and "background-repeat" to repeat property.

Video Lecture

https://youtu.be/RMDSnPwf-bM

3.3.2. Text Properties

Text properties are used to define text styles. Some of the text properties are given in table 3.2.

Table 3.2: text properties with description

Property	Description
color	This property is used to set the color of text
text-align	This property 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, over line and none.
text-transform	This property is used to specify uppercase and lowercase letters in a text. It can be used to turn all characters 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	This property is used to specify indentation of line.
word-pacing	This property is used to specify the space between the words in a text.
color	This property is used to set the color of text

Code 3.3 demonstrates the use of text properties.

</> Code 3.3

```
<! DOCTYPE html>
<html>
<head>
<title>Text Properties</title>
<style>
   body  /* Set style for body */
   {
      background-color: white;
      text-transform: capitalize;
      text-indent:50px;
   }
   h2  /* Set Text style for h2 heading */
   {
      text-decoration: underline;
      text-align: center;
      letter-spacing: 3px;
      word-spacing: 3px;
   }
</style>
</head>
```

```
<body>
   <img src = "dice.png" style ="height: 100px; width: 100px;"/>
   Image 1
   <h2>Text Properties in CSS</h2>
   <div> This div element contains all basic properties of text such
   as text indent, text decoration, word spacing etc. </div>
</body>
</html>
```

The output of code 3.3 is shown in figure 3.3

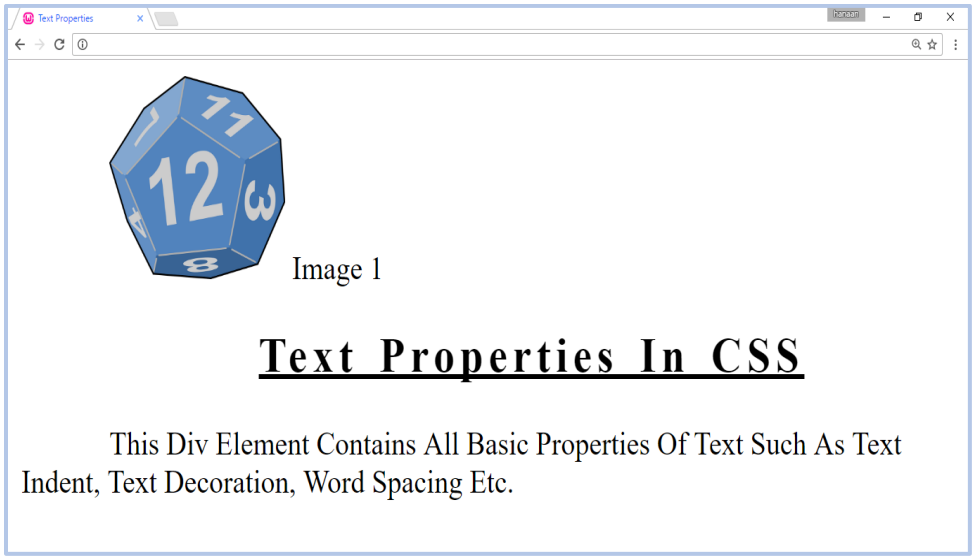

Figure 3.3: Output of code 3.3.

Activity 4

Specify the following properties in the text of above document.

I. Heading should be of 20 points and bold.
II. Paragraph text should be in red color, left justified and indented by 20 points.

3.3.3. Border Properties

Border properties are used to define different border orientations like dotted, solid, dashed etc. Table 3.3 defines some important properties of border.

Table 3.3: border properties with description

Property	Description
Dotted	It displays a dotted border.
Solid	This value displays a solid border.
Dashed	It displays a dashed border.
Groove	This defines a 3D grooved border and it depends on the border color.

Inset	It defines a 3D inset border and it depends on the border color.
Outset	It defines a 3D outset border and it depends on the border color.
None	This value defines no border.
Hidden	It defines a hidden border.
Double	This value defines a double border.

Activity 5

Create a simple web page with three small paragraphs enclosing each in different types of borders.

3.3.4. Table Properties

Table properties are used to define different borders and caption styles in the table. Some of the important table properties are given in table 3.4.

Table 3.4: table properties with description

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

The code 3.4 demonstrates different table properties.

</> Code 3.4

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

   table   /* Set style border collapse for table*/
   {
      border-collapse: collapse;
   }

   table, th, td   /* Set table header and table data border */
   {
      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>Salman</td>
   <td>Aslam</td>
   </tr>
   <table>
</body>
</html>
```

The output of the code 3.4 is shown in figure 3.4.

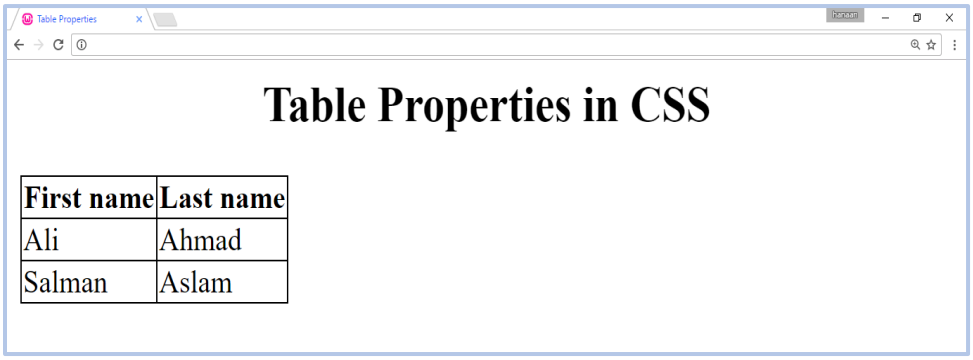

Figure 3.4: Output of code 3.4.

Table can be formed in multiple formats covering many contents and border styles.

Activity 6

Create a table with five English words in the first column and their meaning in the second column. Use table properties to adjust the border, text alignment and font size of the text. Also modify the code by using “border-collapse” property and see its effect on your table.

3.4. CSS Comments

Comments or remarks are used to explain the code to make it understandable. Comments will help the programmers when they edit or modify their own code after some time. Comment are not shown on the web page.


Comments in CSS starts with /* and ends with */.

The syntax of CSS comments is given below:

```
#para1
 {
    text-align: center; /* This property sets the text alignment */
    color: red;
  }
```

3.5. CSS Fonts

In CSS, font shows the style of writing text which is displayed on a webpage i.e. font size, font style etc. Font-style is used to set the font in italic or normal mode. Whereas font-size is used to set the size of the text from small, medium and large mode. The syntax of CSS Fonts is given below:

```
#h3
  {
    font-size: small;
   }
```

3.6. CSS Box Model

Box Model considers every element on a page as a rectangular box. This model has properties like width, height, padding, borders, and margins and also position of boxes. It surrounds every HTML element and allows adjusting the dimensions of boxes provided by their properties.


Box model is a rectangular box having dimensions and color properties.

CSS Box model has four parts: padding, margin, border and content.

- **Padding** - It is an area around the content which is appended in order to make it up to a desired length.

- **Margin** - Margin separates the elements from its neighbors. It clears an area outside the border.
- **Border** - It is an area inside the edge of border which extends the padding area.
- **Content** - It is an area where real text and images appear. It can have either a background or a color.

Code 3.5 demonstrates the use of CSS Box Model.

</> Code 3.5

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Box Model</title>
<style>
   div
   {
     margin: 15px;
     background-color: yellow;
     border: 5px solid blue;
     padding: 10px;
   }
</style>
</head>
<body>
   <h1>CSS Box Model Example</h1>
   <div>This CSS box model contains border, margin, color and
   padding properties of CSS. This box has yellow background, 5px
   solid blue border, 15px margin and 10px padding. </div>
</body>
</html>
```

Output of the code 3.5 is shown in figure 3.5.

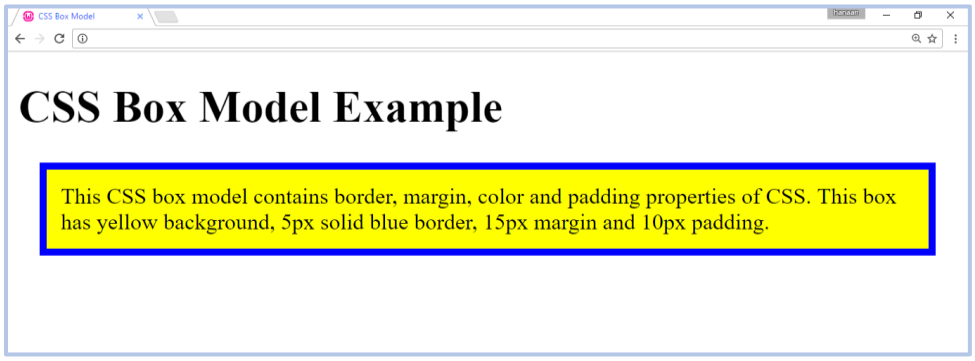

Figure 3.5: Output of code 3.5.

Activity 7

Create a simple web page with three paragraphs. Enclose each paragraph in different box models by giving different dimensions such as background-color, border-color and margin properties.

3.7. CSS Opacity

CSS Opacity is the quality of lacking transparency or translucence. CSS provides the features of opacity which enables adjusting the level of transparency for an image. The value of opacity lies between 0 and 1. The level 0 is completely transparent, 0.5 gives 50 % opacity and 1.0 is not transparent at all.

Opacity is opposite to transparency in an image.

Code 3.6 demonstrates the use of CSS opacity.

</> Code 3.6

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Opacity</title>
<style>
  #img1 {
    opacity: 0.5;
    filter: alpha(opacity=50); /* For IE8 and earlier */
    }
</style>
</head>
<body>
   <h1>CSS Opacity Example</h1>
   <div>The opacity specifies transparency of the object. </div>
   <p> See the difference between these two pictures</p>
   <img src= "dice.png" width= "100" height = "100" > Original Image
   <img id = "img1"src= "dice.png" width= "100" height = "100" >
   Image with 50% opacity.
</body>
</html>
```

Output of the code 3.6 is shown in figure 3.6.

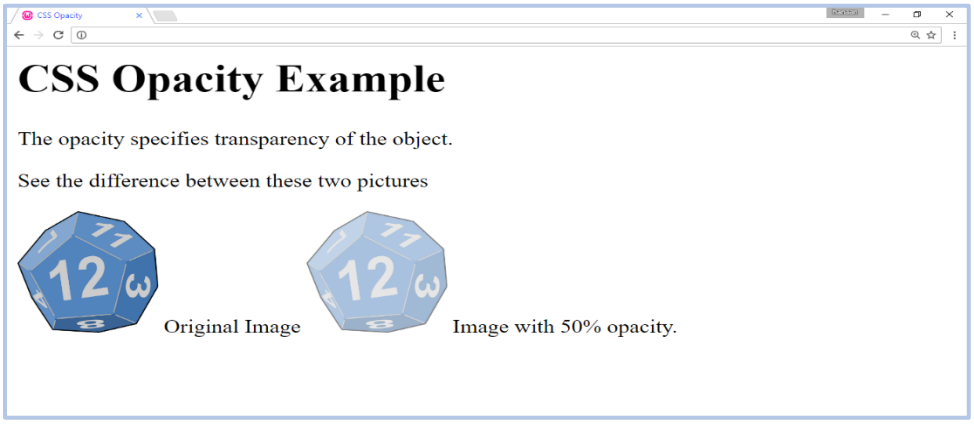

Figure 3.6: Output of code 3.6.

Activity 8

Download a scenic image of your choice and change the opacity of image by giving different value and see the effects.

3.8. CSS Navigation Bar

A navigation bar is a section located at a top of web page which acts as a control point to link different parts of the same web page or other web pages. By using CSS, we can create vertical and horizontal navigation bars.

3.8.1. Vertical Navigation Bar

Vertical navigation bar arranges links in a vertical line or a plane parallel to vertical direction. The code 3.7 demonstrates the use of vertical navigation bar.

</> Code 3.7

```
<!DOCTYPE html>
<html>
<head>
<title>Vertical Navigation bar</title>
<style>
  ul {
       list-style-type: none;
       margin:0;
       padding:0;
       border: 1px solid #555;
       width: 200px;
       background-color: #f1f1f1;
      }
  li {
       text-align: center;
       border-bottom: 1px solid #555;
     }
  li a{           /* Set style for list while hovering*/
        display: block;
        color: #000;
        padding: 8px 16px;
        text-decoration: none;
      }

  li:last-child {   /* Last list has no border */
       border-bottom: none;
      }

  li a.active {  /* Set style for active list*/
         background-color: #4CAF50;
```

```
            color: white;
        }
    li a:hover:not(.active) {
            background-color: #555;
            color: white;
        }
 </style>
</head>
<body>
   <h1>CSS Vertical Navigation Bar Example</h1>
   <p> This vertical navbar contains black border. </p>
   <ul>
     <li><a href="#home" class = "active">Home</a></li>
     <li><a href="#aboutus">About Us</a></li>
     <li><a href="#contactus">Contact Us</a></li>
     <li><a href="#news">News</a></li>
   </ul>
</body>
</html>
```

Output of code 3.7 is shown in figure 3.7.

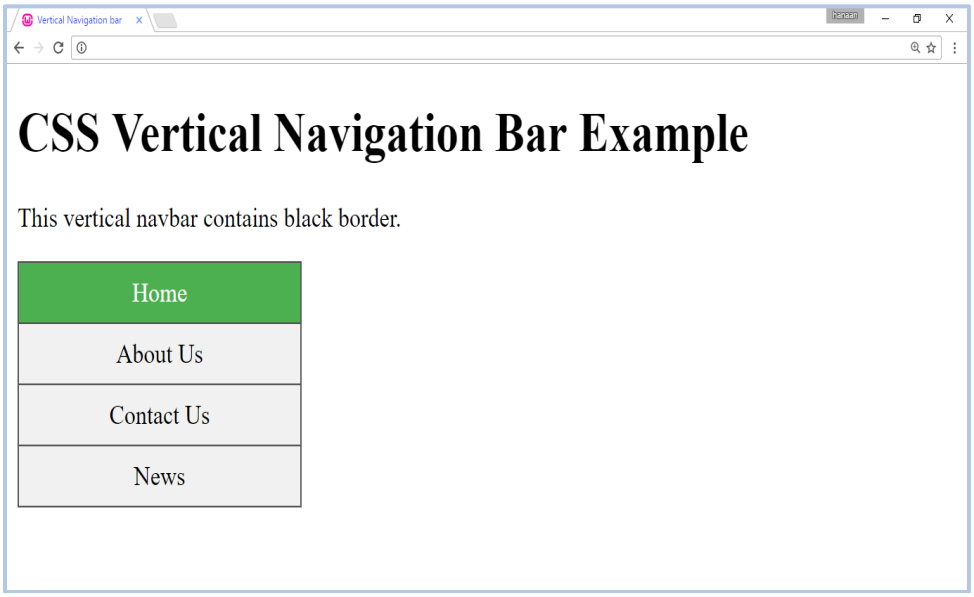

Figure 3.7: Output of code 3.7.

3.8.2. Horizontal Navigation Bar

Horizontal navigation bar arranges links in a horizontal line or plane parallel to horizontal direction. The code 3.8 demonstrates the use of horizontal navigation bar.

</> Code 3.8

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Horizantal Navigation bar</title>
<style>
  ul {
       list-style-type: none;
       margin:0;
       padding:0;
       overflow: hidden;
       background-color: #333;
      }
  li {
       float: left;
      }
  li a{   /* Set style for list while hovering */
        display: block;
        color: white;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
      }
    li a.active {  /* Set style for active list*/
         background-color: #4CAF50;
         color: white;
       }
/* Set style for list item which is not active*/
  li a:hover:not(.active) {
         background-color: #111;
     }
 </style>
</head>
<body>
   <h1>CSS Horizontal Navigation Bar Example</h1>
   <p> This Horizontal navbar contains black background. </p>
   <ul>
     <li><a href="#home" class = "active">Home</a></li>
     <li><a href="#aboutus">About Us</a></li>
     <li><a href="#contactus">Contact Us</a></li>
     <li><a href="#news">News</a></li>
   </ul>
</body>
</html>
```

Output of code 3.8 is shown in figure 3.8.

With the help of CSS, we can develop attractive navigation bars.

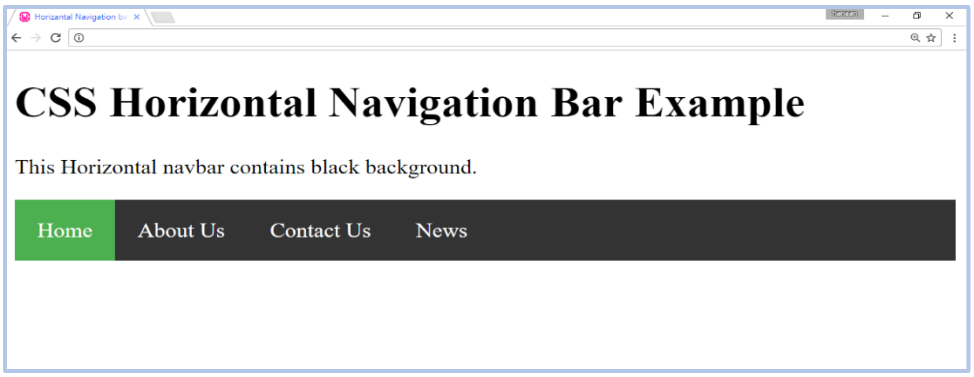

Figure 3.8: Output of code 3.8.

Activity 9

I. Modify code 3.7 and add more links to the vertical navigation bar.
II. Modify code 3.8 and add more links to the horizontal navigation bar.

3.9. CSS Dropdown

A drop down menu is a graphical control box which displays a list of items. It allows a user to select an item with the help of mouse. CSS provides a convenient way to arrange list of items using drop down menu by providing navigation between different tabs on a webpage. The navigation between links appears when the cursor moves on the link.

By using CSS, we can create dropdown menus with mouse over effects.

Code 3.9 demonstrates the use of drop down menu using CSS.

</> Code 3.9

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Dropdown</title>
<style>
.dropbtn {                    /* Style the Dropdown Button */
    background-color: #4CAF50;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;
}
.dropdown {                  /*  Positioning for dropdown content */
    position: relative;
```

```
    display: inline-block;
}
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}
.dropdown-content a {        /* Styling links inside the dropdown */
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}
/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content { /* Show dropdown menu on hover*/
    display: block;
}
.dropdown:hover .dropbtn {  /* Change background color of dropdown*/
    background-color: lightblue;
}
</style>
<body>
     <div class="dropdown">
     <button class="dropbtn">Menu</button>
     <div class="dropdown-content">
       <a href="#">Menu 1</a>
       <a href="#">Menu 2</a>
       <a href="#">Menu 3</a>
     </div>
   </div>
</body>
</html>
```

Output of the code 3.9 is shown in figure 3.9.

Activity 10

Modify the code 3.9 and change the color, font-size and other properties. Also add some more sub menus in the dropdown list.

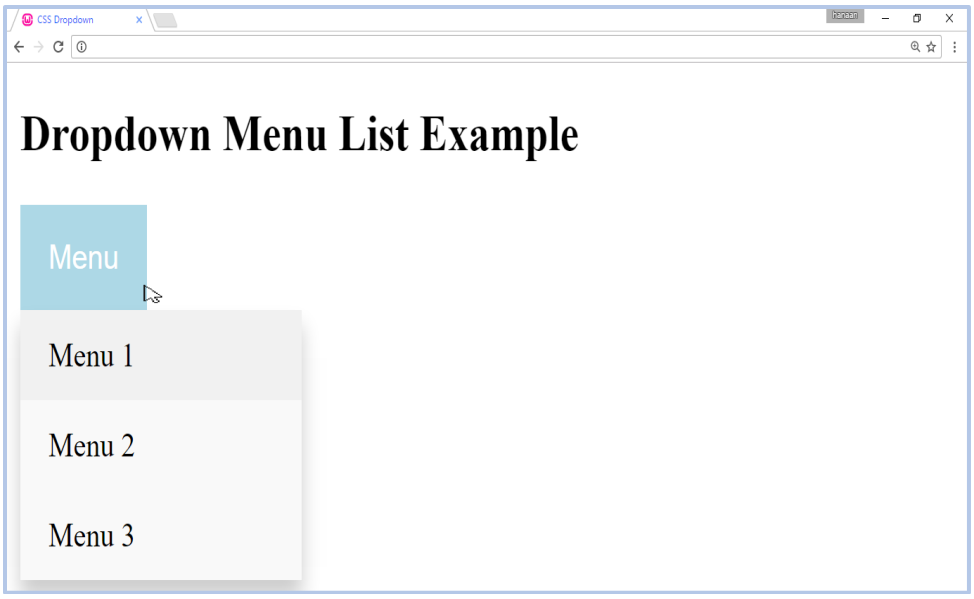

Figure 3.9: Output of code 3.9.

3.10. CSS Tooltip

Tooltip is a message which displays when a cursor is positioned over an object, icon, image or other element in a web page. It is used to indicate additional data about an object which is displayed only when the user moves the mouse pointer over a component. The code 3.10 demonstrates the use of tooltip in CSS.

</> Code 3.10

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Tooltip</title>
<style>
    /* Styling properties for Tooltip*/
    .tooltip {
      position: relative;
      display: inline-block;
      border-bottom: 1px dotted black;
     }
   /* Styling for text that are displayed in tooltip */
    .tooltip .tooltiptext {
      visibility: hidden;
      width: 120px;
      background-color: white;
      color: black;
      text-align: center;
      padding: 5px 0;
      border-radius: 6px;
```

```
      position: absolute;
      z-index: 1;
      border: 1px solid blue;
      text-align: justify;
     }
   /*  When mouse hovers over text, tooltip should visible */
    .tooltip:hover .tooltiptext {
      visibility: visible;
     }
</style>
<body>
   <h1>CSS Tooltip Example</h1>
   <div class="tooltip">Hover over me
   <span class="tooltiptext">When you move over "Hover over me"
   text, it will display this text in tooltip</span>
   </div>
</body>
</html>
```

Output of code 3.10 is shown in figure 3.10.

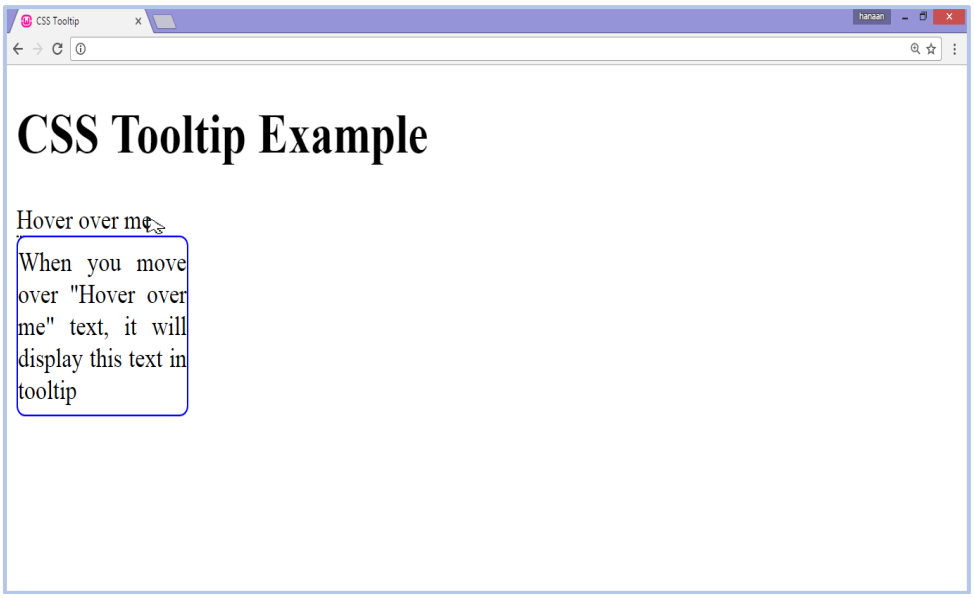

Figure 3.10: Output of code 3.10.

3.11. CSS Image Gallery

Image gallery allows users to place images on a website. CSS3 an advance version of CSS provides a convenient way to make gallery of images on a webpage. It also allows adjusting the dimension of images (width and height), color, captions and borders.

CSS can be effectively used to manage image gallery albums.

Code 3.11 demonstrates the use of image gallery using CSS.

</> Code 3.11

```
<html>
<head>
<title>CSS Image Gallery Example </title>
<style>
  /* Styling div that contains gallery */
   div.gallery {
    margin: 3px;
    border: 1px solid blue;
    float: left;
    width: 180px;
     }

  /* Border color changes while hovering over gallery */
   div.gallery:hover {
    border: 1px solid #777;
     }

 /* Set width and height of image that are placed inside gallery */
   div.gallery img {
    width: 100%;
    height: auto;
     }

   div.desc {  /* Set style for description of image */
    padding: 15px;
    text-align: center;
    }
</style>
</head>
<body>
   <h2> CSS Image Gallery</h2>

   <div class="gallery">
     <a target="_blank" href= "pc.png">
     <img src="pc.png" width="300" height="200">
     </a>
     <div class="desc">Computer image</div>
   </div>

   <div class="gallery">
     <a target="_blank" href="dice.png">
     <img src="dice.png" width="300" height="200">
     </a>
     <div class="desc">Dice Image</div>
   </div>

   <div class="gallery">
     <a target="_blank" href="puzzle.png">
     <img src="puzzle.png" width="300" height="200">
     </a>
```

```
    <div class="desc">Puzzle Image</div>
  </div>

</body>
</html>
```

Output of the code 3.11 is shown in figure 3.11.

Figure 3.11: Output of code 3.11.

Video Lecture

https://youtu.be/9D7gAlMbLQ4

Activity 11

Download your favorite images and create image gallery by using CSS. Also adjust the height and width of images to make your web page look attractive.

3.12. CSS Image Sprite

Image sprite is a collection of images that merge into a single image. It is used to reduce the download time by converting several requests for a number of images in to single request to a web server to send them as a solitary picture. The code 3.12 demonstrates the use of image sprite using CSS.

</> Code 3.12

```
<html>
<head>
<title>CSS Image Sprite Example </title>
<style>
   btn{
    background: url(myfile.png);
    display: inline-block;
    height: 20px;
    width: 20px;
     }
</style>
</head>
<body>
   <h2> CSS Image Sprite Example</h2>
   <div class="btn"> </div>
</body></html>
```

3.13. CSS Attribute Selector

The “attribute” selector makes use of specific attributes to select an element. The code 3.13 demonstrates the use of “target” attribute inside the <a> to select the element as desired.

</> Code 3.13

```
<html>
<head>
<title>CSS Attribute Selector Example </title>
<style>
   a[target] {
      background-color: yellow;
     }
</style>
</head>
<body>
   <h2> The links with target attribute gets yellow background</h2>
   <a href=http://www.google.com target="_blank">Google</a>
   <a href=http://www.yahoo.com target="_top"> Yahoo</a>
   <a href="http://www.aiou.edu.pk "> AIOU</a>
</body>
</html>
```

Output of code 3.13 is shown in figure 3.12.

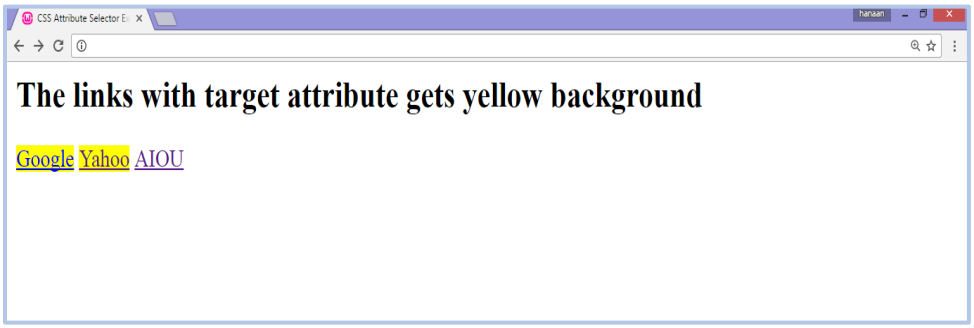

Figure 3.12: Output of code 3.13.

Unit Summary

In this unit, we learned about the different methods of writing CSS. We learned how to use basic properties of CSS to give style to different HTML elements. CSS allows us to control the opacity of image as desired. We have also learned how to create image gallery in CSS.

Self Assessment Questions

Choose the correct answer.

1. What are the advantages of using CSS in HTML?
   A. It makes the web page attractive.
   B. Maintenance is easy.
   C. Page loads faster.
   D. All of the above.

2. An inline style sheet is used when a single document has a unique style?
   A. True.
   B. False.

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. A CSS styling rule is composed of three parts which are :
   A. selector, attribute, value
   B. property, value, attribute
   C. selector, property, value
   D. selector, value, attribute

5. Which of the following property is used to set color of the text?
   A. color.
   B. text-color.
   C. background-color.
   D. None of the above.

6. Which of the following selector is used when applying a style to multiple elements?
   A. Multiple – id
   B. Class
   C. Id
   D. Multiple - class

7. The correct CSS syntax for using font property is :
   A.
   B.
   C.
   D. None of the above

8. “border-collapse” property defines whether the cell borders are connected or separate?
   A. True.
   B. False.

9. Which of the following property is used to insert an image in a web page?
   A. background-position.
   B. background-image.
   C. background-attachment.
   D. background-repeat.

10. 100% opacity means:
   A. original Image.
   B. transparent image.
   C. semitransparent image.
   D. None of the above.

Answer Key:

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

Review Questions

Write short answers of the following questions.

1. Briefly describe basic concept of CSS.
2. Write different ways to apply style to a web page.
3. What is the difference between attribute and property?
4. What is the difference between class and id selector?
5. How do you add comments in CSS?
6. What is the concept CSS box model and why it is used?
7. Describe the method of creating image gallery in CSS.
8. What is attribute selector and how it is used?

</> Coding Exercise

1. Create a web page that gives basic information about your hometown. Use CSS to display information in a well-designed and attractive way.
2. Create a web page that displays your mark sheet. Use table and text properties to define different borders and text styles in your web page.
3. Download your favorite pictures from your Facebook account. Create an album of image galleries using CSS.

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. Web Development Using CSS. Available on: http://www.slideshare.net/mahantaanjan/web-development-using-css3
7. Background Properties. Available on: http://www.w3schools.com/Css/pr_background.asp?output=print
8. CSS Image Gallery. Available on: https://www.w3schools.com/css/css_image_gallery.asp
9. CSS Tables. Available on: http://www.w3schools.com/Css/css_table.asp
10. CSS Style Sheets. Available on: http://wiki.answers.com/Q/FAQ/4628-9

UNIT 4

JavaScript – I

Introduction to JavaScript

In the previous units, you have learnt the basics of web, HTML and CSS. Now, we are moving forward to learn JavaScript which is a client side scripting language. This language is embedded inside a webpage and executed on the client computer.

JavaScript was invented by Brendan Eich in 1995 and later in 1997 it became European Computer Manufacturers Association (ECMA) standard, therefore, its official name is ECMA Script but is commonly known as JavaScript. It is usually used to make your web page interactive. It can be used to check or validate the contents of forms, change images, open new pop-up windows and write dynamic page contents. It can also be used with CSS to make static HTML into Dynamic Hyper Text Markup Language (DHTML). This allows you to divide your web page into different parts that appear, disappear or move around on your web page. JavaScript program is a client-side script so it runs only when a user or client loads a web page in any browser. In this unit of JavaScript, the basic topics like statements, variables, operators and function will be covered.

JavaScript and Java are two completely different languages. JavaScript is a client-side scripting language which resides inside the web page. Whereas Java is a general-purpose high level computer programming language.

Unit Outcomes

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

1. Understand JavaScript syntax.
2. Write JavaScript statements and functions.
3. Use a text editor (such as notepad) to type and save your JavaScript code and view its output in a browser.
4. Use of operators in JavaScript.
5. Understand the concept of arrays and its associated operations.

Terminologies

Statements:	Set of instructions/line of codes that are to be executed in browser.
Variables:	Containers in which we store values.
Function:	A named procedures which performs specific task.
String:	A combination of characters to form a word/sentence.
Arrays:	List of homogenous data items.
Push:	Inserting an element in to an array.
Pop:	Removing of element from an array.

4.1. JavaScript Syntax

JavaScript statements are placed within the **<script>**… **</script>** HTML tags in a web page. The **<script>** tags, can be placed anywhere within the web page. However, it is recommended that it should be kept within the **<head>** tag for faster execution. The **<script>** tag alerts the browser program to execute all the text between these tags line by line. A simple syntax of your JavaScript is given below:

```
<! DOCTYPE html>
<html>
 <head>
 <script>
       :
       :
---JavaScript Statements---
       :
       :
  </script>
  </head>
  <body>
 </body>
</html>
```

JavaScript is a case-sensitive language.

Activity 1

Browse any web page. Right click on it and click on "inspect element" to view the source code of web page and identify whether any scripts are running in that web page or not.

4.2. My First JavaScript Program

Let us write our first javascript program. Follow the given steps in order to design your own web page.

Step 1: Open Notepad in Window
Step 2: Type your HTML and JavaScript code (given in the code 4.1) in Notepad. At this stage, you do not need to understand various HTML tags (written inside <>) used in the document.

</> Code 4.1

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Statements</title>
</head>
<body>
    <p id="demo1">
    </p>
```

```
    <p id="demo2">
    </p>
    <p id="demo3">
    </p>
 <script>
     document.getElementById("demo1").innerHTML="Demonstration";
     document.getElementById("demo2").innerHTML="JavaScript
      Statements";
     document.getElementById("demo3").innerHTML="JavaScript
      Identifiers";
  </script>
  </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:

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. Your browser should display something similar to the following figure 4.1.

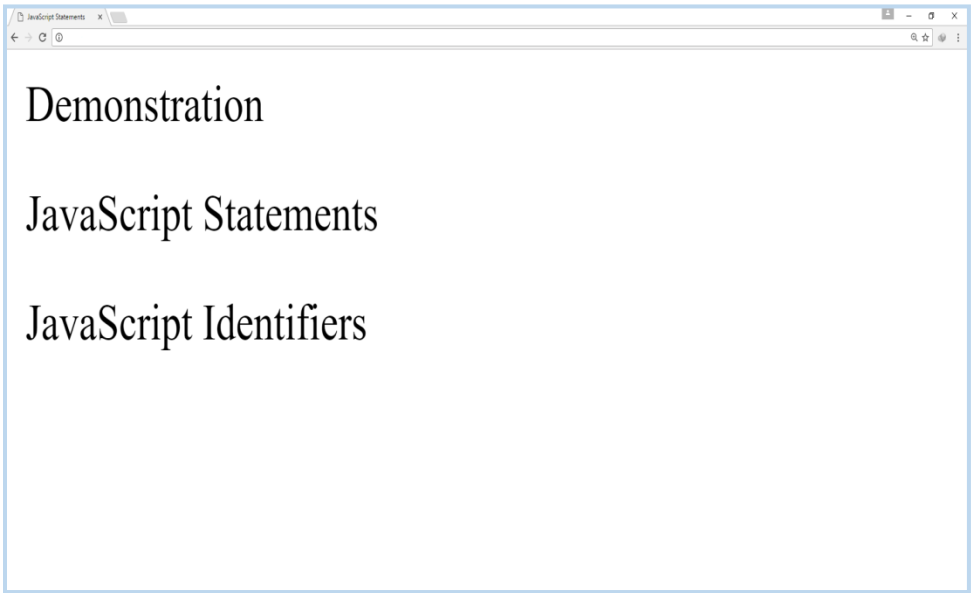

Figure 4.1: Output of code 4.1

In code 4.1, we have defined a paragraph named "demo". Then we have assigned a string "JavaScript Statements" to our paragraph demo using a JavaScript statement i.e. document.getElementbyId().innerHTML. This string is then printed using `<p id="demo"> </p>.` the output of code 4.1 is shown in figure 4.1.

Activity 2

Write a JavaScript statement to write the string "My first JavaScript Statement" in a paragraph having id= "first paragraph" and execute it in your web browser.

4.3. JavaScript Variables

JavaScript Variables are used to store information that can be referred later in the computer program. A variable is initialized by storing a value in it. We can initialize a variable at the time of declaration or later whenever we need that variable. We can also assign a new value to a variable at any time but this will be replaced by the previous stored value.

Variables are the containers to store data or values.

Code 4.2 demonstrates the use of variable declaration and assignment.

</> Code 4.2

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Variables</title>
</head>
<body>
 <p id="demo"> </p>
  <script>
   var a = 10;     // Variable Declaration.
   var b = 20;
   var c = a + b;
   document.getElementById("demo").innerHTML = " Value of
      variable c is: " + c;
   </script>
</body>
</html>
```

When the above JavaScript statements execute in your browser, you will see the output as shown in figure 4.2.

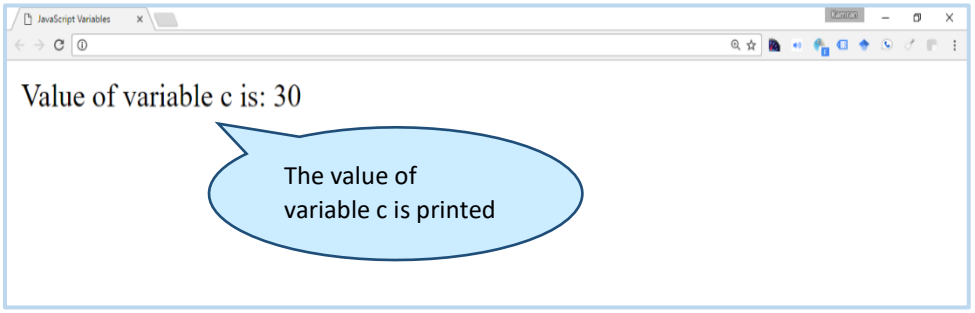

Figure 4.2: Output of code 4.2

Plus sign ("+") in JavaScript is used to add two variables and also to concatenate depending upon the type of variables.

Activity 3

Write JavaScript code in which you declare at least five different variables and display their sum.

4.4. JavaScript Operators

JavaScript operators are the special symbols used to perform a mathematical or logical operation etc. As an example, in A + B, the symbol "+" represents the operator and is used to add numbers or values stored in the variables. List of some operators that are used in JavaScript are given in table 4.1.

Table 4.1: JavaScript Operators

Operators	Description
+	Used for Addition.
-	Used for Subtraction.
*	Used for Multiplication.
/	Used for Division.
%	Used for Modulus (Remainder after division).
++	Used for Increment.
--	Used for Decrement.
=	Used to Assign values to variables.
== , !=	Used for checking equality (non equality).
< , <=	Used for checking less than (or equal to).
>=, >	Used for checking greater than (or equal to).

" += " operator updates and assigns a variable at the same time. For example A+=4 means A= A+4.

Code 4.3 shows the working of these operators.

</> Code 4.3

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Operators</title>
</head>
<body>
    <p id="add"></p>
    <p id="sub"></p>
```

```
    <p id="mul"></p>
    <p id="div"> </p>
      <script>
        var a = 20;     // Assignment operator used to assign value.
        var b = 10;
        var c = a + b;  // Adding two variables.
        document.getElementById("add").innerHTML = " Addition of
        <b>a</b> & <b>b</b> is: " + c; // + operator used for
        concatenation.
        var c = a - b;  // Subtracting two variables.
        document.getElementById("sub").innerHTML = " Subtraction of
        <b>a</b> & <b>b</b>is: " + c;

        var c = a * b;  // Multiplication of two variables.
        document.getElementById("mul").innerHTML = " Multiplication
        of <b>a</b> & <b>b</b>is: " + c;

        var c = a / b;  // Division of two variables.
        document.getElementById("div").innerHTML = " Division of
        <b>a</b> & <b>b</b>is: " + c;
      </script>
</body>
</html>
```

When the above JavaScript statements execute in your browser, you will see the output as shown in figure 4.3.

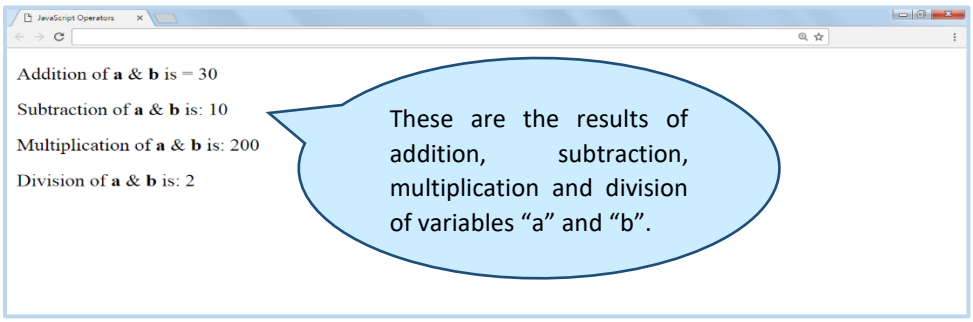

Figure 4.3: Output of code 4.3

Activity 4

Write a JavaScript code using % operator. You can find factors of a number and can verify whether a number is prime or not.

4.5. JavaScript Data Types

Data type is the classification of data which specifies the type of variable. While writing code, a programmer must declare each variable with its proper data type. JavaScript deals with all data types like string, boolean, decimal, integers etc. If we use the keyword “var” to assign a variable, JavaScript automatically determines its data type based on the nature of the value.


The variables in JavaScript can be defined at the time of the usage.

Code 4.4 shows how you can declare variables.

</> Code 4.4

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Data Types</title>
</head>
<body>
  <script>
    var a;      // "a" is not defined.
    var b = 0;  // "b" is a Number.
    var a = "We are learning JavaScript.";  //"a" is a now a String.
  </script>
</body>
 </html>
```

In JavaScript, variable data type is dynamic in nature which means that same variable can be given different data types at different times. For example, if the variable is given a numeric value, its type will become numeric and if the same variable at a later stage is assigned a string value, it will behave as a string variable.

4.6. JavaScript Functions

Functions are “self-contained” block of organized codes that are used to perform a specific task. Sometimes we write a code which is to be used again and again, then it is better to put this code inside a “function” and “call” that function whenever required. It helps in writing code in a modular form as each function can have the set of statements doing a separate and specific task. Functions are also used to divide a big program into many small patches of code (or modules). Functions should be defined before using or calling it. The keyword “function” is used to define a function whose name should be unique and list of parameters are optional. Parameters are those values which are sent to the function while calling it from outside code. The statement inside a function are written in a pair of curly braces as shown in code 4.5.

</> Code 4.5

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Functions</title>
  <script>
   function sayHello()
   {
    document.getElementById("result").innerHTML = "Hello Students";
```

```
    }
  </script>
 </head>
<body>
   <button onclick = "sayHello()">Click Me!</button>
   <p id="result"> </p>
</body>
</html>
```

In the above code a function is created which is assigning a string "Hello Students" to a variable "result". This function is called by invoking “onclick” event defined inside the <body> tag. When the user clicks the button, the function is called and the value is displayed on the screen as shown in figure 4.4 and 4.5.

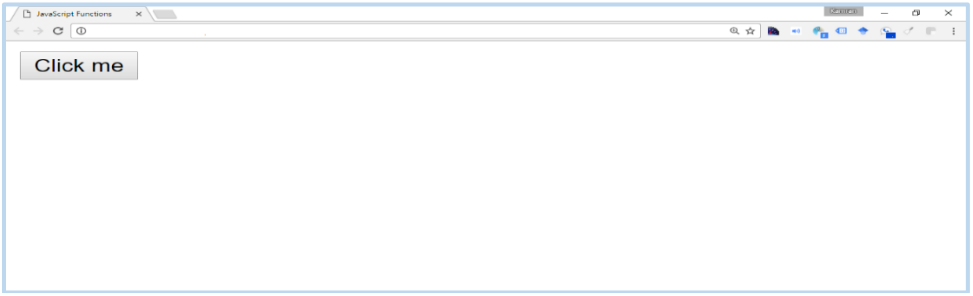

Figure 4.4: Output of code 4.5

After clicking the "Click Me!" button, the following output will be displayed as shown in figure 4.5.

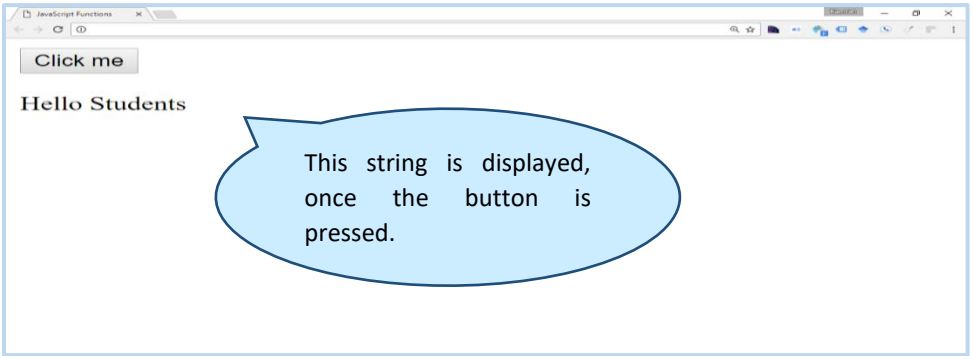

Figure 4.5: Output of code 4.5

The use of functions in a code makes it more modular and readable.

Activity 5

Write JavaScript code and define a function which receives two values as input and shows their sum.

Video Lecture

https://youtu.be/Hns2DtAmYGE

4.7. JavaScript Variable Scope

A scope of a variable is the region inside the computer program where that variable can be used. It defines whether that particular variable can be viewed or not from different portions of the code. The variables in javascript can be declared either with a local or global scope.

4.7.1 Global Variable

A global variable is a variable that is visible (accessible) throughout the program. All code statements and functions in a code can access the global variable. When we initialize a variable in JavaScript, it automatically becomes a global variable.

4.7.2 Local Variable

Contrary to the global variable, a local variable is accessible only inside a function where it is defined. It is also called function scope. If we declare a variable having both; local and a global scope with the same name, the local variable will take precedence when we use it inside a function. This type of behavior is called shadowing.

A function can use both global and local variables.

Code 4.6 shows the difference between local and global variable.

</> Code 4.6

```
<!DOCTYPE html>
<head>
<title>JavaScript Global and Local Variables</title>
</head>
<body>
<p>In this example we can explain how local variables differ from global
variables. </p>
```

```
<h2>Local variable when accessed globally</h2><p id = "demo"></p>
<h2>Local variable when accessed locally</h2><p id = "demo1"></p>
<h2>Global variable when accessed locally</h2><p id = "demo2"></p>
<h2>Global variable when accessed globally</h2><p id = "demo3"></p>
<script>
  var animal = "Lion";  // Declaring & Initializing a character
                        // variable.

  myFunction();
  document.getElementById("demo").innerHTML="The type of fruit is "
   + typeof fruit;
    document.getElementById("demo3").innerHTML="The type of animal is
    " + animal;
  function myFunction()
   {
     var fruit = "Apple";
     document.getElementById("demo1").innerHTML="The type of fruit
      is " + fruit;
     document.getElementById("demo2").innerHTML="The type of animal
      is " + animal;   }
    </script>
</body>
</html>
```

When the above JavaScript statements are executed, you will see the following output as shown in figure 4.6.

Figure 4.6: Output of code 4.6

Activity 6

Write a JavaScript code utilizing both the local and global variables.

4.8. JavaScript Strings

A string is a finite sequence of characters that can contain numbers, characters and space etc. JavaScript strings are commonly used for manipulating and storing text. A string can store any type of text inside quotation marks using single or double quotes e.g. var txt = “Pakistan, Canada, China”;

4.8.1 String Length

The length of a string stored in a variable can be found with the help of a built-in property “length”. Code 4.7 shows how you can use strings.

</> Code 4.7

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Strings</title>
<body>
<p>Length of variable “txt” is: </p>
<p id="demo"></p>
      <script>
      var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
      document.getElementById("demo").innerHTML = txt.length;
      </script>
      </head>
</body>
</html>
```

When the above JavaScript statements are executed in your browser, you will see the output as shown in figure 4.7.

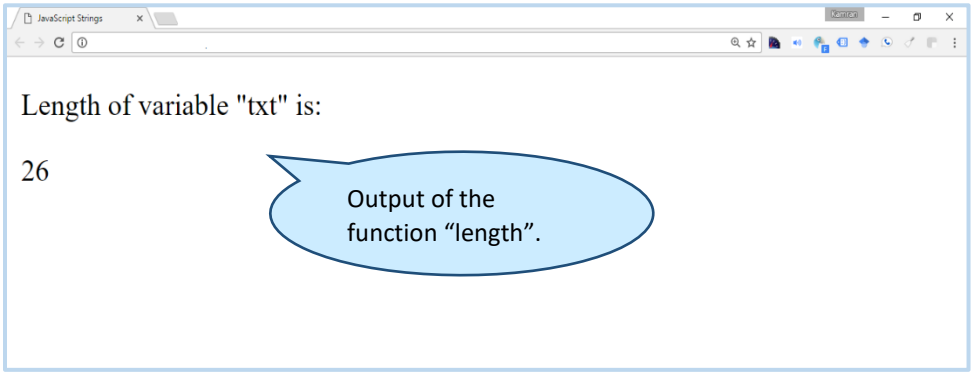

Figure 4.7: Output of code 4.7

4.8.2 Special Characters

Special characters like double quotes (“ ”), single quote (‘), back slash (\ ) etc. can also be the part of a string. However, we have to use a “\” escape character before these special characters that convert them into a string. Code 4.8 shows how we can use special characters.

</> Code 4.8

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Strings</title>
<body>
<p id="demo"></p>
    <script>
      var x = 'It\'s good.';
      var y = "We are checking \"Special Characters\" in our code.";
      document.getElementById("demo").innerHTML = x + "<br>" + y;
    </script>
</head>
</body>
</html>
```

When the above JavaScript statements are executed in your browser, you will see the output of special characters as a string, shown in figure 4.8.

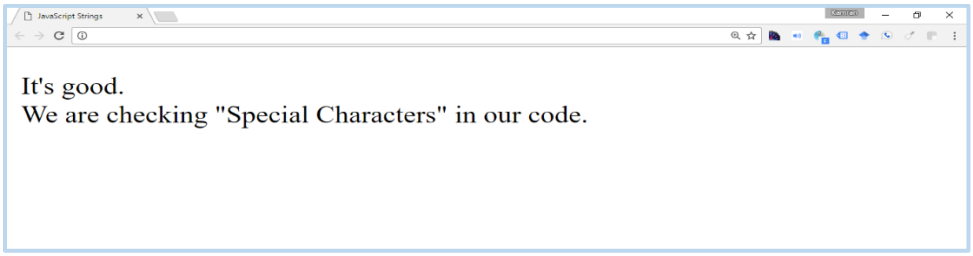

Figure 4.8: Output of code 4.8.

4.8.3 Breaking Long Code Lines

Programmers often like to avoid longer codes in their computer programs for more readability and understandability. We can easily break code line using a black slash "\" operator. Code 4.9 shows how we can use the special character to break a longer code line.

</> Code 4.9

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Strings</title>
<body>
<p>You can break a code line within a text string.</p>
   <p id="demo"></p>
    <script>
     document.getElementById("demo").innerHTML = "AIOU \
     Students";
    </script>
</head>
</body>
</html>
```

The output of above code is shown in figure 4.9.

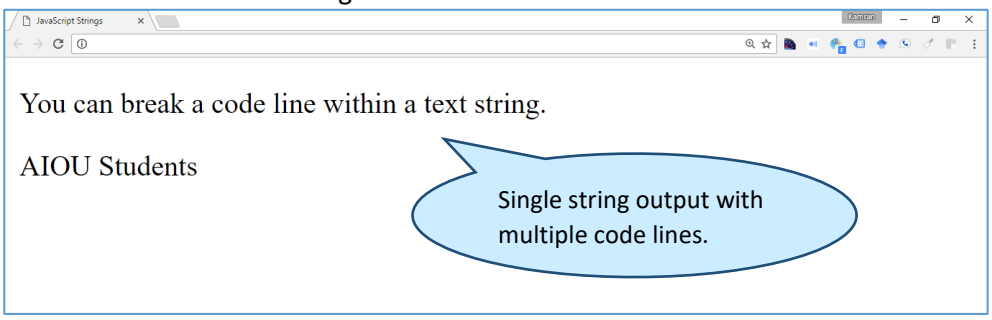

Figure 4.9: Output of code 4.9.

If you see the output, "\" joining the word "AIOU" and "Students" which are written in separate lines of code.

There are multiple other functions that can be used for string manipulation.

Activity 7

Write JavaScript code by inserting a space in a string and display it repeatedly in next lines with more spaces.

4.9. JavaScript Arrays

JavaScript Array is a list of homogenous data items that are grouped together usually in rows and columns. Once a program gets bigger with more functionality in it, we need to declare more variables. If these variables have the same functionality, you can declare them in the form of an array. JavaScript array is an object that allow us to store multiple values with a single variable name in the form of a list. An array can hold many values under a single name, and values can be accessed by referring to an "index number". Code 4.10 demonstrates the use of JavaScript Arrays.

Index of an array starts from 0. [0] is used for the first element in an array, [1] is for second element and so on.

</> Code 4.10

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Arrays</title>
</head>
<body>
<h2>All elements of the array</h2>
```

```
<p id="demo"></p>
<h2>Accessing third element of the array</h2>
<p id="demo1"></p>
  <script>
     // Declaring &  Initializing an array list.
    var animals = ["Cat", "Dog", "Elephant"];
    document.getElementById("demo").innerHTML= animals;
    document.getElementById("demo1").innerHTML= animals[2];
  </script>
</body>
</html>
```

The output of the code 4.10 is shown in figure 4.10.

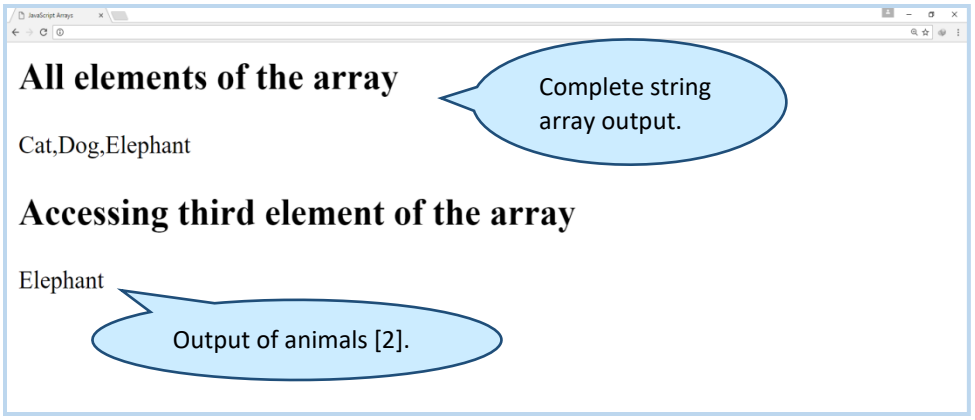

Figure 4.10: Output of code 4.10.

Activity 8

Write JavaScript code to define an array with at least 5 values and sort them alphabetically.

4.9.1 Convert array to String

An array can be converted into a string so that it may be displayed or inspected. The demonstration of this property is shown in code 4.11.

</> Code 4.11

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Array Methods</title>
</head>
<body>
   <p>JavaScript Method "toString()" returns an array as a comma
   separated  string. </p>
   <p id="demo"></p>
  <script>
    var animals = ["Cat", "Dog", "Elephant", "Horse"];
    document.getElementById("demo").innerHTML= animals.toString();
  </script>
</body>
```

```
</html>
```

This above example shows how we can convert an array into a single string. When you execute the above JavaScript statements in your browser, you will see the output as shown in figure 4.11.

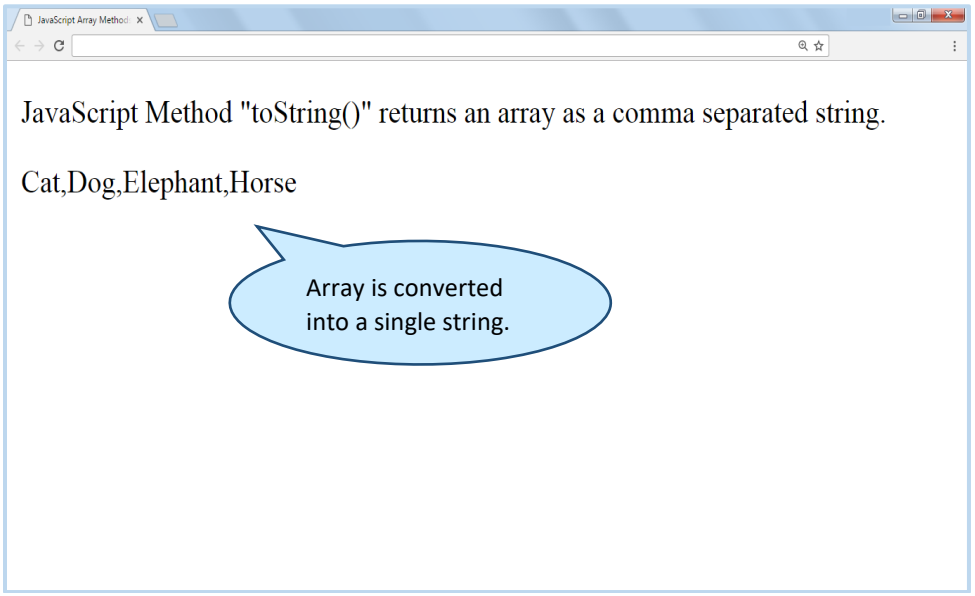

Figure 4.11: Output of code 4.11

Activity 9

Write JavaScript code that removes "Cat" and "Horse" from the animal's array using splice() function.

4.9.2 pop() and push() operations

pop() operation is used to remove an element from an array. When we say "pop" an element, we mean delete an element from an array. Code 4.12 demonstrates the use of pop() operation in an array.

</> Code 4.12

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Array Methods</title>
</head>
<body>
    <p>JavaScript Method "pop()" removes the last element from an
       array.</p>
    <button onclick = "popFunction()">Click to remove element!
    </button>
    <p id="demo"></p>
    <script>
       var animals = ["Cat", "Dog", "Elephant", "Horse"];
       document.getElementById("demo").innerHTML= animals;
       // Function to remove an element from a web page.
       function popFunction() {
          animals.pop(); // Calling of JavaScript built-in function.
```

```
        document.getElementById("demo").innerHTML= animals;
    }
</script>
</body>
</html>
```

When we execute the above JavaScript code in our browser, we will see the output as shown in figure 4.12 and 4.13.

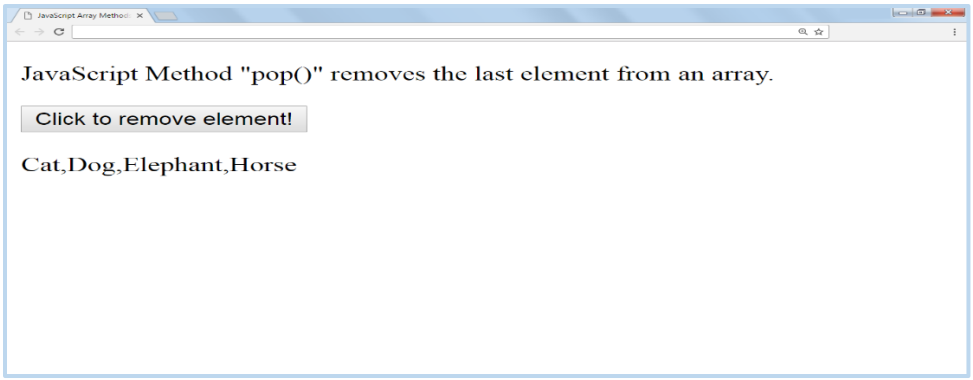

Figure 4.12: Output of code 4.12.

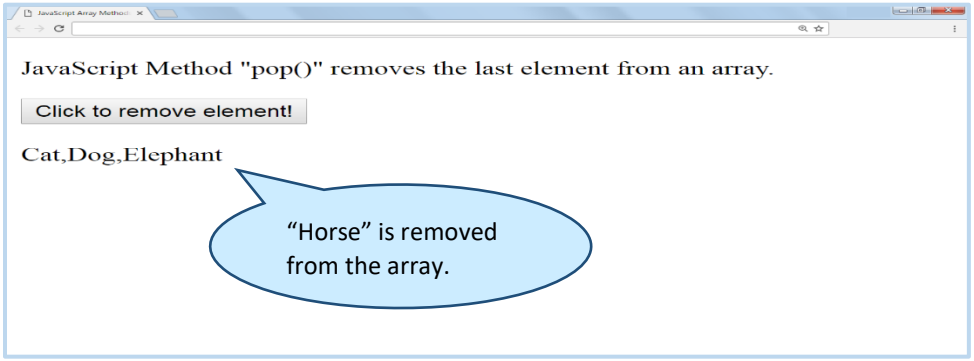

Figure 4.13: Output of code 4.12.

The push() operation is inverse of pop. push() operation is used to insert an element in an array. When we say "push" an element, we mean inserting an element in an array. Code 4.13 demonstrates the use of push method.

</> Code 4.13

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Array Methods</title>
</head>
<body>
   <p>JavaScript Method "push()"add element at the of an array.</p>
   <button onclick = "pushFunction()">Click to add another element!
   </button>
   <p id="demo"></p>
   <script>
```

```
        var animals = ["Cat", "Dog", "Elephant", "Horse"];
        document.getElementById("demo").innerHTML= animals;
        function pushFunction() {
         animals.push("Lion");
         document.getElementById("demo").innerHTML= animals;
         }
    </script>
</body>
</html>
```

Output of code 4.13 is shown in figure 4.14 and 4.15. Please note that the array was originally having four elements and after the click of the button a fifth element is added into "animal's" array.

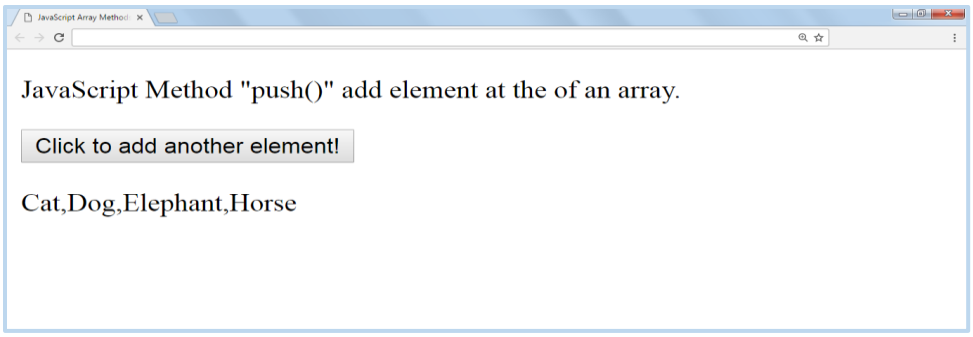

Figure 4.14: Output of code 4.13.

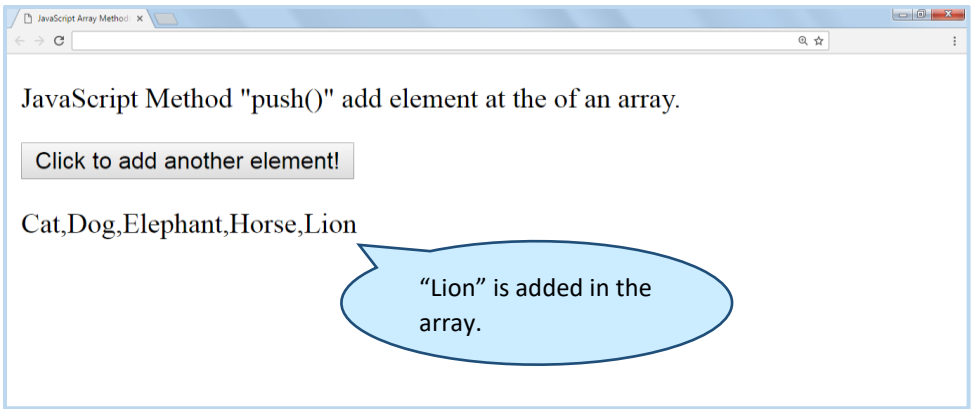

Figure 4.15: Output of code 4.13.

Activity 10

Modify code 4.13 and push more elements in the array of animals. Convert array elements in upper case and display them in alphabetical order.

Video Lecture

https://youtu.be/ctJAtvGAlDM

Unit Summary

This was an introductory unit to JavaScript. Basic JavaScript syntax is explained. Efficient code formation in the form of functions is also covered. Array declaration and manipulation was demonstrated at the end.

Self Assessment Questions

Select the correct answer.

1. Which is the correct place to insert a JavaScript code in a web page?
   A. The <body> section
   B. The <head> section
   C. Both the <head> section and the <body> section are correct.
   D. None of the above

2. JavaScript code is enclosed inside the following _____ section.
   A. <java>
   B. <java-code>
   C. <script>
   D. None of the above

3. Which operator is used to assign a value to a variable?
   A. -
   B. =
   C. *
   D. X

4. JavaScript is case-sensitive language?
   A. True.
   B. False.

5. How do you create a function in JavaScript?
   A. function = myFunction()
   B. function myFunction()
   C. function:myFunction()
   D. None of the above

6. How do you call a function named "myFunction"?
   A. call myFunction()
   B. call function myFunction()
   C. myFunction()
   D. None of the above

7. Integer or numeric variable in JavaScript can be defined as:
   A. int var marks;
   B. var marks = 100;
   C. marks interger;
   D. var marks = “100 int”;

8. How can you add a comments in JavaScript code?
   A. <!--This is a comment-->

B. 'This is a comment
C. //This is a comment
D. None of the above

9. How to insert a multiline comment in JavaScript?
   A. //This comment has more than one line//
   B. <!--This comment has more than one line-->
   C. /*This comment has more than one line*/
   D. None of the above

10. JavaScript and Java are same language.
   A. True
   B. False

11. How do you declare a variable in JavaScript code?
   A. v carName;
   B. variable carName;
   C. var carName;
   D. None of the above

12. Which of the following is NOT an operator in JavaScript?
   A. Arithmetic Operator
   B. Comparison Operator
   C. Conditional Operator
   D. Static Operator

Answer Key:

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

Review Questions

Write short answers of the following.

1. What is the difference between Java and JavaScript?
2. Why is JavaScript called the Client-Side Scripting language?
3. How are variables defined in JavaScript? Give few examples.
4. How are comments added in JavaScript code?
5. What would be the result of 3+2+"7"?
6. What is the difference between local and global variables?
7. Write a short note on JavaScript Functions.
8. Explain push and pop methods in JavaScript.

</> Coding Exercise

1. Write a JavaScript program to find the area of a triangle when the lengths of its sides are 5, 6, 7.
2. Write a JavaScript function that converts an array element into a string.
3. Write the JavaScript program to demonstrate push and pop operations in a single program.

References and Further Reading

1. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
2. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH.
3. Rauschmayer, A. (2014). Speaking JavaScript: An In-Depth Guide for Programmers. O'Reilly Media, Inc.
4. Elliott, E. (2014). Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. O'Reilly Media, Inc.
5. JavaScript Scope. Available on: https://www.w3schools.com/js/default.asp
6. JavaScript. Available on: https://www.tutorialspoint.com/javascript/index.htm
7. JavaScript Online Examination. Available on: http://www.sreejobs.com/onlineexam/exam.php?exm=JS
8. JavaScript Variable. Available on: https://www.sitepoint.com/demystifying-javascript-variable-scope-hoisting/
9. JavaScript Functions. Available on: http://www.w3resource.com/javascript-exercises/javascript-functions-exercises.php
10. JavaScript Basics. Available on: http://www.w3resource.com/javascript-exercises/javascript-basic-exercises.php

UNIT 5

JavaScript – II

Introduction

In the previous unit, we have learnt some basic concepts of JavaScript. Now, it's time to move ahead and learn some advance topics related to JavaScript. So, this unit will focus on JavaScript conditions, loops and events. Main functionality and usage of JavaScript is to validate the data input by a user. This helps a lot to manage data efficiently without sending it to the server. So, form validation will also be covered in this unit.

Unit Outcomes

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

1. Implement JavaScript conditions.
2. Understand JavaScript Events.
3. Use loops for different programming constructs.
4. Validate different input types in a form.

Terminologies

Condition Statement:	Conditions statement are based on "if" and "else if" conditions.
Loops:	Sequence of instructions which can be invoked repeatedly.
Events:	Events are user actions and can be linked with specific block of code.

5.1. JavaScript Conditions

The conditional statement performs specific actions based on a certain condition. If a condition is true a block of code is executed and if it is false, some other block of code is executed. Table 5.1 shows JavaScript conditional statements.

Table 5.1: JavaScript Conditional Statements

Conditions	Description
if	Allows statements to be executed as long as condition is true.
if…else	Allow the statements (after "if") to be executed as long as condition is true otherwise the second block of statement will be executed (after else).
else…if	Allows statements to be executed as long as the first condition is false.
switch	Allows statements of many alternative blocks to be executed on their respective conditions.

Conditional statements are keywords and are written in lowercase letters.

Code 5.1 demonstrates the use of JavaScript conditions.

</> Code 5.1

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Conditions</title>
</head>
<body>
    <h3>Result:</h3>
    <p id="demo"></p>
 <script>
     var marks = 40;
      if (marks > 50)
       {
         document.getElementById("demo").innerHTML=" Student got
         passing marks.";
       }
      else
       {
         document.getElementById("demo").innerHTML=" Student didn't
         got passing marks.";
       }
  </script>
</body>
</html>
```

As the variable “marks” has the value 40, the condition “marks > 50” is false. Therefore, the commands after “if” statements will be skipped and “else” block will be executed as shown in figure 5.1.

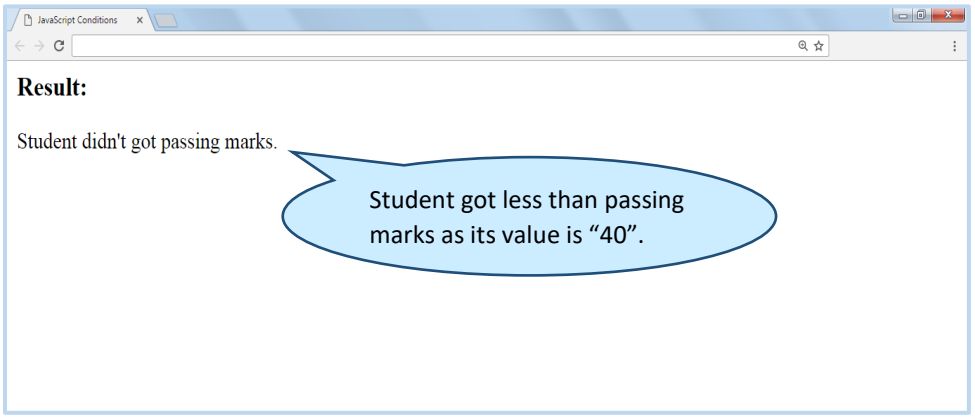

Figure 5.1: Output of code 5.1

Activity 1

Incorporate multiple conditions for the variable “marks” in code 5.1.

5.2. JavaScript Switch

A switch statement is a selection control mechanism that can have a number of possible conditions with their respective block of code. It is combination of many if then else statements. In this way, code will become more readable, simple to understand and efficient. Code 5.2 describes the use of switch statement.

"break" (a keyword) is used to jump outside the “switch” block.

</> Code 5.2

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Switch</title>
</head>
<body>
   <h3>Result:</h3>
   <p id="demo"></p>
   <script>
      var number = 60;
      switch (number)
      {

       case 50: document.write("Student got passing marks.");
       break;
```

```
        case 60: document.write("Student got Good marks.");
        break;

        case 70: document.write("Student got Very Good marks.");
        break;

        case 80: document.write("Student got Excellent marks.");
        break;

        default: document.write("Student result not shown.");
}
   </script>
</body>
</html>
```

When the above JavaScript statements executes in your browser, you will see the output shown in figure 5.2. As the variable "number" has the value "60", the code associated with "case 60" will be executed.

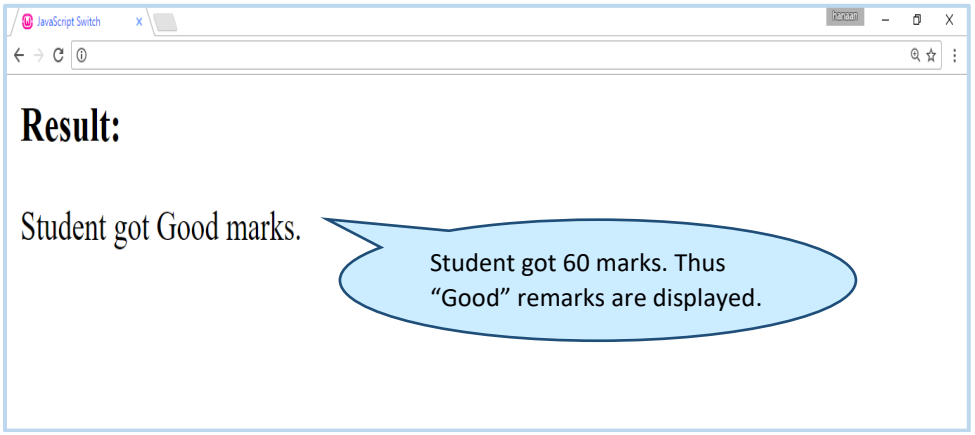

Figure 5.2: Output of code 5.2

Activity 2

Modify the code 5.2 and change the variable values to demonstrate the switch statement.

5.3. JavaScript Loops

Sometimes we need to execute a command or a group of instructions repeatedly in our program. So in order to write them again and again, we can write them once in a block and execute the block repeatedly. A loop is a set of statements that are executed until a condition is true. It is a method of control flow in a computer program. Table 5.2 has a list of identifiers used for loops implementation.

Table 5.2: JavaScript Loops

Loops Identifiers	Description
for	Allows statements to be executed for a given number of times.
while	Allows statements to be executed while some specific condition is true.
do-while	Allows statements to be executed while a specified condition remains true (first execution will be executed in any case whereas the conditions are checked at the end of the loop).

for loop:

The syntax of the "for" statement is given in the following sample code. Instruction 1 initializes a loop variable which will be used for the execution of the loop. Instruction 2 declares the loop execution conditions and the loop will be executed as long as the given condition is true. Instruction 3 determines the incremental change in the loop variable after each loop cycle execution.

</> Syntax Code

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Loops</title>
</head>
<body>
    for (instruction 1; instruction 2; instruction 3)
       {
            :
          Block of code to be executed after satisfy each condition.
            :
       }
</body>
</html>
```

Code 5.3 give a complete example of for loop.

</> Code 5.3

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Loops</title>
</head>
<body>
    <h3>List of animals:</h3>
    <p id="demo">
    </p>
      <script>
        var animals = ["Cat", "Dog", "Elephant", "Horse", "Mouse"];
        var text = ""; // Initializing a character variable with
                          null value.
        for (var i = 0; i < animals.length; i++)
```

```
        {
          text += animals[i] + "<br>";
         // Name of each animal is stored in "text" sequentially
            using for-loop.

        }
        document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

The output of code 5.3 is shown in figure 5.3. Names of the animals are displayed sequentially inside a loop with loop variable "i". The variable "i" keeps the record of which animal is to be displayed and when to terminate the loop.

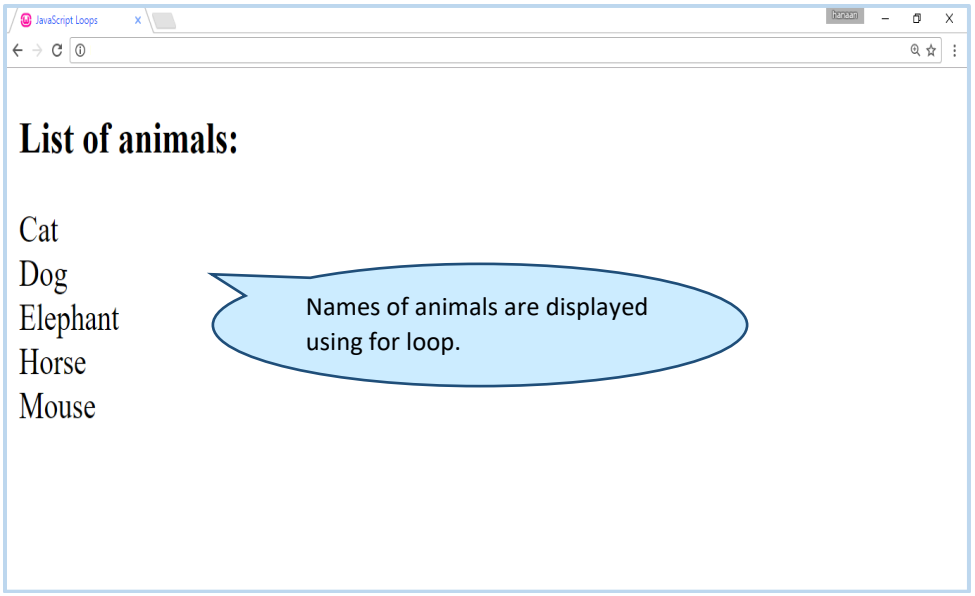

Figure 5.3: Output of code 5.3

while loop:

While loop executes as long as the set of conditions remains true. These conditions are checked at start of each loop and if found true the execution is proceeded.

The loop variable must be incremented during each loop execution to meet the loop termination criteria.

</> Syntax Code

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Loops</title>
</head>
<body>
       while (condition)
```

```
        {
            :
           Block of code to be executed after satisfy each condition.
            :
        }
</body>
</html>
```

The same code (5.3) is now re-written for the while conditions as shown in code 5.4

</> Code 5.4

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Loops</title>
</head>
<body>
    <h3>List of animals:</h3>
    <p id="demo">
    </p>
      <script>
        var animals = ["Cat", "Dog", "Elephant", "Horse", "Mouse"];
        var text = "";
        var i = 0;
        while (i < animals.length)
        {
          text += animals[i] + "<br>";
          i++;
        }
        document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

The output of code 5.4 is shown in figure 5.4.

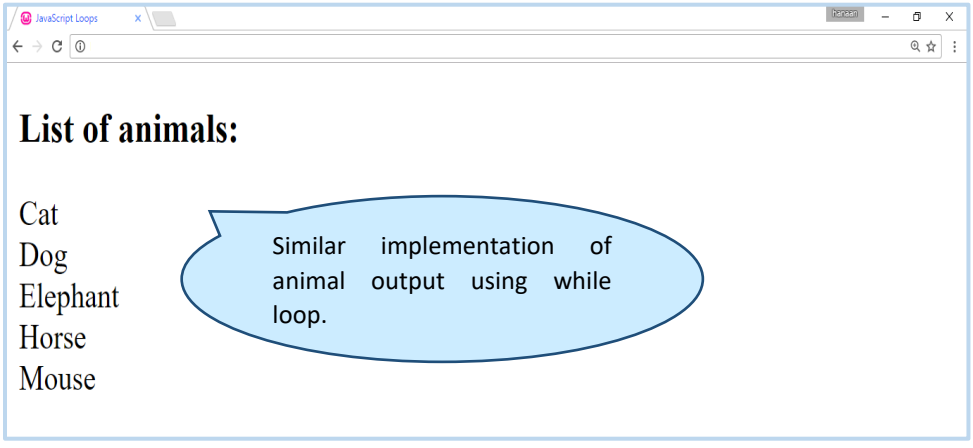

Figure 5.4: Output of code 5.4

do-while loop:

In do-while statements, the loop is executed once before checking the condition of the loop. Even if the conditions are not true, the first execution will be made in any case. The syntax of do while loop is shown below.

do while loops executes at least once.

</> Syntax Code

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Loops</title>
</head>
<body>
       do
        {
              :
          Block of code to be executed after satisfy each condition.
              :
        }
       while (condition);
</body>
</html>
```

Activity 3

Write a JavaScript code to display first ten even numbers starting from 2.

5.4. JavaScript Events

Windows is based on graphical user interface theme. The user has the provision to enter data through a keyboard, enter his choice through a mouse or any other peripheral interface with the computer. An event is an action that takes place as a result of these user tasks. The programmers have to take care of these inputs from the user. Most of the events include the mouse movements, clicks and key press etc.

Event takes place as a result of user action.

Some commonly used and important JavaScript events are given in table 5.3.

Table 5.3: JavaScript Events

Events	Description
onclick	It is invoked when a user clicks an element.
onmouseover	It is invoked when a user moves the mouse over an element.
onmouseout	It is invoked when a user moves the mouse away from an

	element.
onchange	It is invoked when an element has been changed.
onkeydown	It is invoked when a user pushes a key on keyboard.
onload	It is invoked when the browser has finished loading the web page.

onclick event:

The code 5.5 demonstrates the use of this event while displaying the current time and date when user clicks on the button.

</> Code 5.5

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Events</title>
</head>
<body>
        <p>Click the button below to show Time:</p>
        <button onclick = "document.getElementById('demo').innerHTML
         = Date()">Time</button>
          // Built-in function of “Date” to show system Date & Time.
        <p id="demo">
</body>
</html>
```

When the above code is executed in the browser, it will show a button named “Time”. When user click on this button, the current date and time are displayed as shown in figure 5.5 and 5.6.

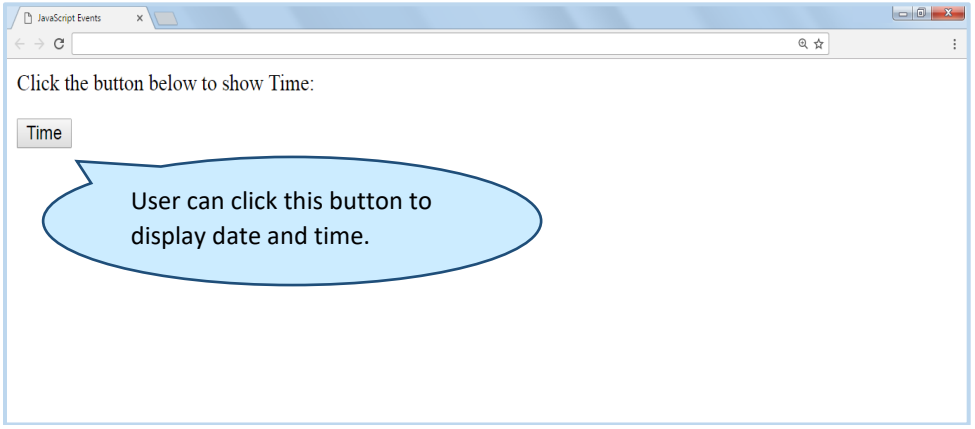

Figure 5.5: Output of code 5.5

Activity 4

Write a JavaScript code which displays your name after clicking the button.

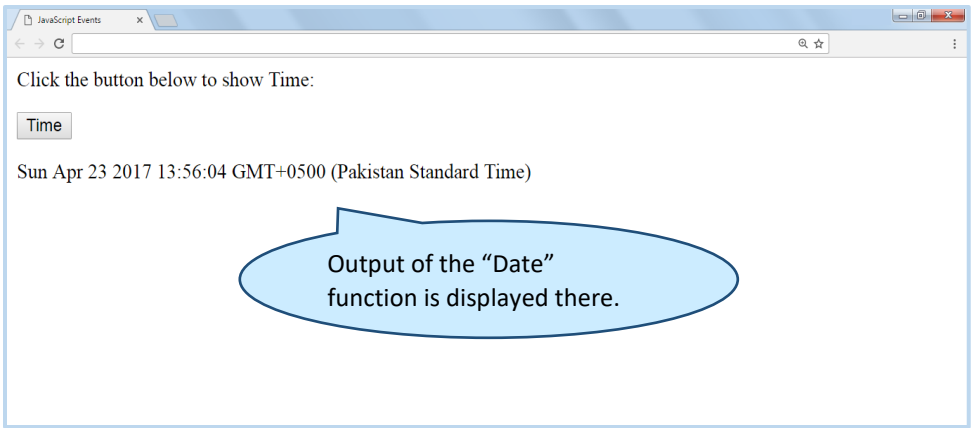

Figure 5.6: Output of code 5.5

Video Lecture

https://youtu.be/xKPx1bhCUqs

mouseover & mouseout event:

Mouseover and mouse events are the events based on mouse movements. When the user brings or removes mouse cursor on any specified word/sentence, these events will occur. The code 5.6 describes the event as a practical example.

</> Code 5.6

```
<! DOCTYPE html>
<html>
<head><title>JavaScript Events</title>
</head>
<body>
     <h3 onmouseover="style.color='red'"   onmouseout="style.color =
      'blue'"> Please! Bring mouse cursor on this sentence.
     </h3>
</body>
</html>
```

When the above code 5.6 is executed in the browser, it will initially show sentence written in black color. When a user places the mouse on the sentence, its color will become red and when the user

moves the mouse away from the sentence, it will turn to blue. The output of this program is shown in figure 5.7 and 5.8.

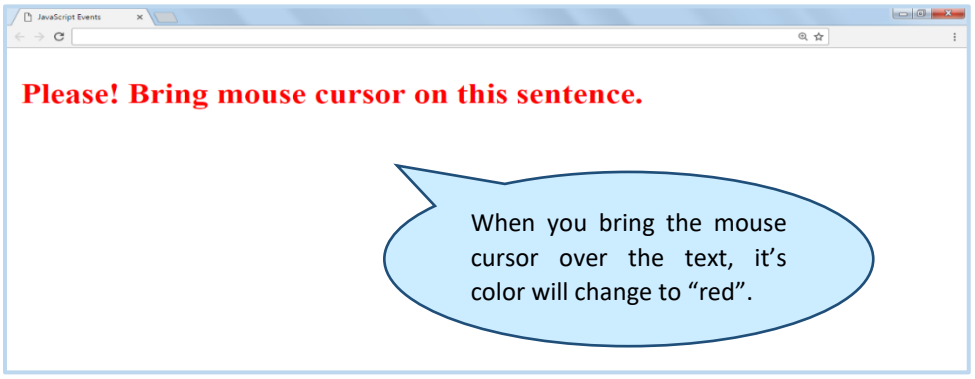

Figure 5.7: Output of code 5.6

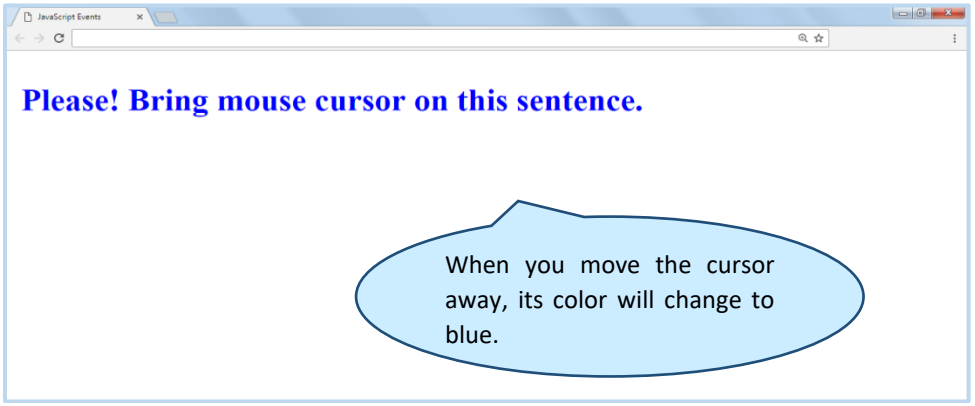

Figure 5.8: Output of code 5.6

Video Lecture

https://youtu.be/yQTOk0Zlhbs

Activity 5

Write a JavaScript code that displays multiple color names in black. When you bring mouse cursor on that name, it changes the color as written.

onchange event:

"onchange" event occurs when the value of an element is changed. The code 5.7 demonstrates the use of this event with the help of drop down menu.

</> Code 5.7

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Events</title>
</head>
<body>
        <h4>Select Web Development course:</h4>
     <select id="mySelect" onchange="myFunction()">
        <option value="Html5">Html5
        <option value="CSS3">CSS3
        <option value="JavaScript">JavaScript
        <option value="JQuery">JQuery
     </select>
        <p>When you select a course, a function is called and output
            will shown below:</p>

     <p id="demo"></p>
<script>
   function myFunction() {
    var x = document.getElementById("mySelect").value;
    document.getElementById("demo").innerHTML = "You selected: "+x;
}
</script>
</body>
</html>
```

When you execute the above JavaScript statements in your browser, you will see the output as shown in figure 5.9. When the user changes the option from the dropdown menu, the description will change accordingly as shown in figure 5.10 and 5.11.

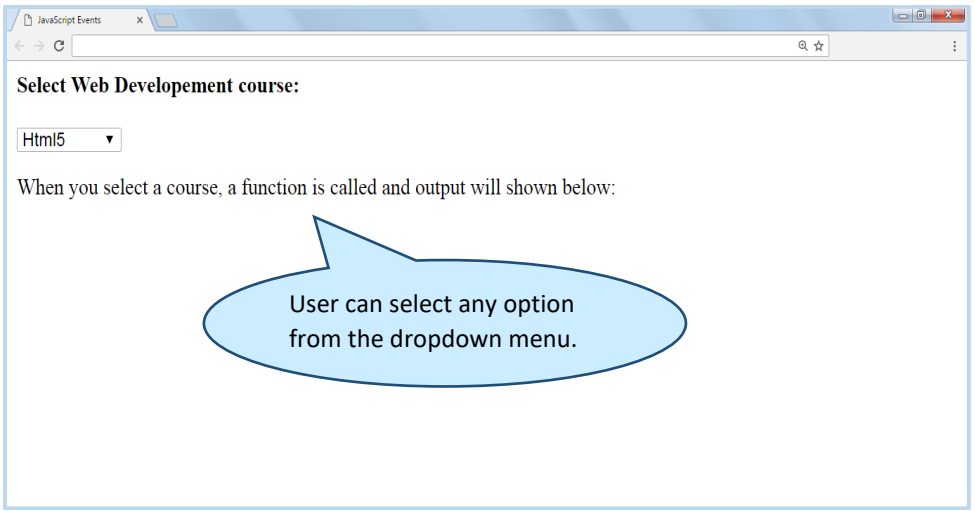

Figure 5.9: Output of code 5.7

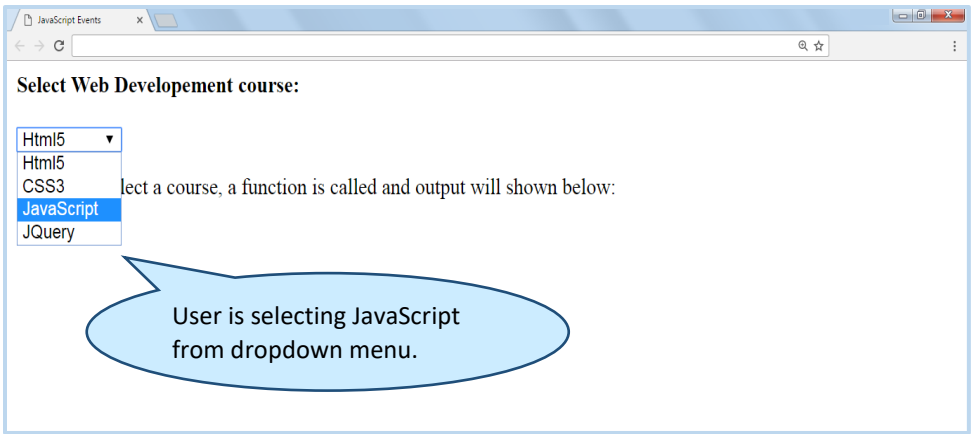

Figure 5.10: Output of code 5.7

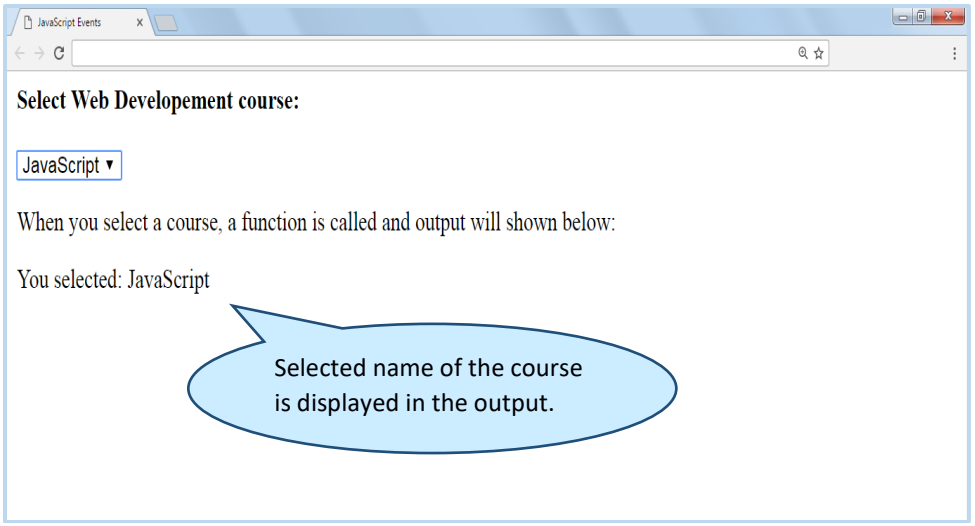

Figure 5.11: Output of code 5.7

Activity 6

Modify code 5.8 to display five major cities of your country. When user selects a city, a small description should be displayed on the screen.

onkeydown event:

onkeydown event occurs when the user presses any key from the keyboard. Code 5.8 demonstrates the use of this event.

</> Code 5.8

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Events</title>
</head>
<body>
      <h3>onkeydown</h3>
```

```
    <p>When you pressed any button from keyboard, a notification
       will appear:</p>
    <input type="text" onkeydown="myFunction()">
<script>
   function myFunction() {
      alert("You pressed a key from keyboard.");
   }
</script>
</body>
</html>
```

When you will execute the code 5.8, you will see the output as shown in figure 5.12 and 5.13. Here the user presses a key from the keyboard, to show this event.

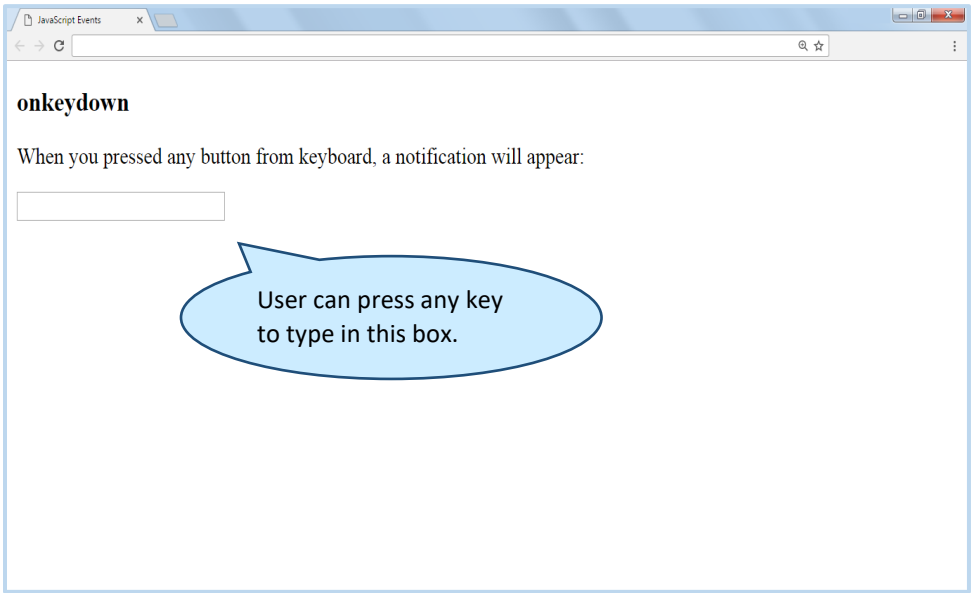

Figure 5.12: Output of code 5.8

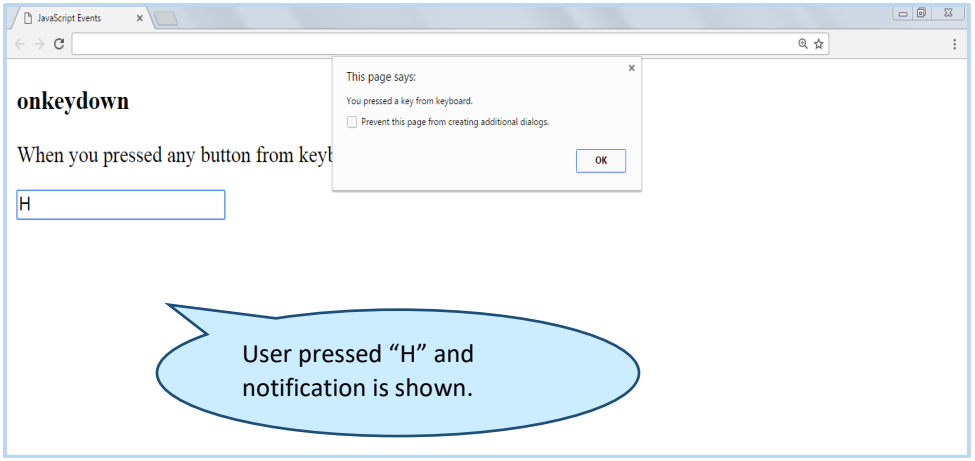

Figure 5.13: Output of code 5.8

Activity 7

Modify code 5.9 for more understanding and practice.

onload event:

onload event occurs when an object is loaded in the browser. It is used within the <body> tag to execute a script when all contents are completely loaded by a web page (including CSS files, images, script files, etc.). It can also be used to check the visitor's browser type, its version, and load the content of the web page which matches with the proper version of the browser. Its demonstration program is shown in code 5.9.

</> Code 5.9

```
<! DOCTYPE html>
<html>
<head>
 <title>JavaScript Events</title>
</head>
<body onload="myFunction()">
        <h3> Example of onload event.</h3>
<script>
function myFunction() {
    alert("Your web page is loaded.");
}
</script>
</body>
</html>
```

When you execute the above JavaScript statements, the output is shown in figure 5.14. The message box will appear instantaneously with the output.

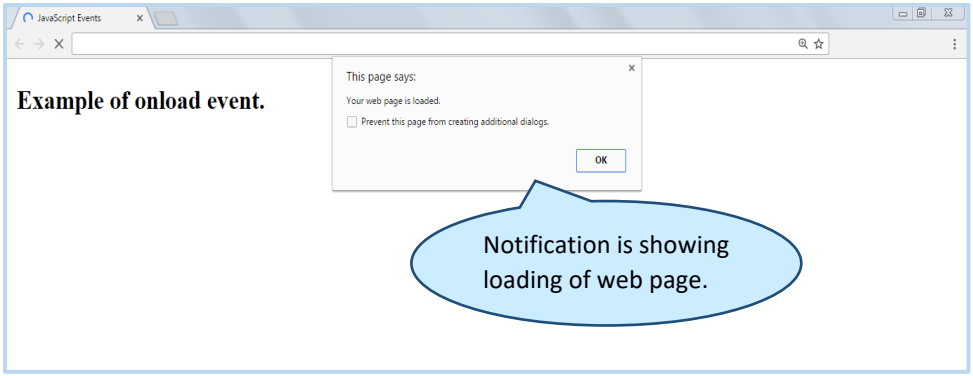

Figure 5.14: Output of code 5.9

5.5. JavaScript Forms

You have already learnt about HTML forms in unit 3. Now, we are about to incorporate form validation using JavaScript. Form validation is to check the accuracy of user input entered during form filling action.

Form validation checks the accuracy of data.

JavaScript from validation helps the system to get input from user properly according to the type of input element. Code 5.10 illustrates the working of form validation.

</> Code 5.10

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Forms</title>
<script type="text/javascript">

 // Form validation code will come here.
    function validate()
      {
         if(document.myForm.Name.value == "")
         {
            alert("Please provide your Name !" );
            document.myForm.Name.focus();
            return false;
         }
         if (document.myForm.Address.value == "")
         {
            alert("Please provide your Address !");
            document.myForm.Address.focus();
            return false;
         }
         if(document.myForm.Email.value == "")
         {
            alert("Please provide your Email !");
            document.myForm.Email.focus();
            return false;
         }
         if (document.myForm.Email.value != "")
         {
            var emailID = document.myForm.Email.value;
             atpos = emailID.indexOf("@");
             dotpos = emailID.lastIndexOf(".");

           if (atpos < 1 || (dotpos - atpos < 2))
            {
               alert("Please enter correct Email ID ! ");
               document.myForm.Email.focus();
               return false;
            }

         }
        if (document.myForm.Zip.value == "")
            {
               alert("Please provide your five digit city Zip code
                       in Number format !" );
```

```
                document.myForm.Zip.focus();
                return false;
             }
       // "NaN" is used to check whether a user give a number or not.
        if (isNaN (document.myForm.Zip.value ) ||
            document.myForm.Zip.value.length != 5 )
            {
                alert("Zip code should contain atleast 5 characters
                       (Numbers) !" );
                document.myForm.Zip.focus();
                return false;
            }
        if( document.myForm.City.value == "-1" )
          {
                alert( "Please provide your City !" );
                return false;
          }

       }
</script>
</head>
<body>
<h1> JavaScript Can Validate Input </h1>
<h3> Please fill the form below:</h3>
<form action="/cgi-bin/test.cgi" name="myForm" onsubmit="return
  validate()">
          <table cellspacing="3" cellpadding="1" border="3">
             <tr>
                <td align="left">Name</td>
                <td><input type="text" name="Name" /></td>
             </tr>
             <tr>
                <td align="left">Address</td>
                <td><input type="text" name="Address" /></td>
             </tr>
             <tr>
                <td align="left">E-mail</td>
                <td><input type="text" name="Email" /></td>
             </tr>
             <tr>
                <td align="left">Zip Code</td>
                <td><input type="text" name="Zip" /></td>
             </tr>
             <tr>
                <td align="left">Gender</td>
                <td><input type="radio" name="Male" checked />
                     Male</td>
             </tr>
             <tr>
               <td></td>
               <td><input type="radio" name="Female"/> Female </td>
             </tr>
             <tr>
                <td align="left">City</td>
                <td>
```

```
                <select name="City">
                    <option value="-1" selected>Select</option>
                    <option value="1">Islamabad</option>
                    <option value="2">Lahore</option>
                    <option value="3">Karachi</option>
                </select>
            </td>
        </tr>
        <tr>
            <td align="left">Subscription</td>
            <td><input type="checkbox" name="checkbox">
             Newsletter</td>
        </tr>
        <tr>
             <td></td>
             <td><input type="checkbox" name="checkbox"> Magazine
              </td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Submit"/></td>
        </tr>
    </table>
</form>
</body>
</html>
```

In this code, some different input elements are obtained from the user and are being validated. When the user presses the "submit" button, the code checks the validity of different inputs. Whenever a discrepancy is found, it is reported in a message box as shown in figure 5.15 - 5.22.

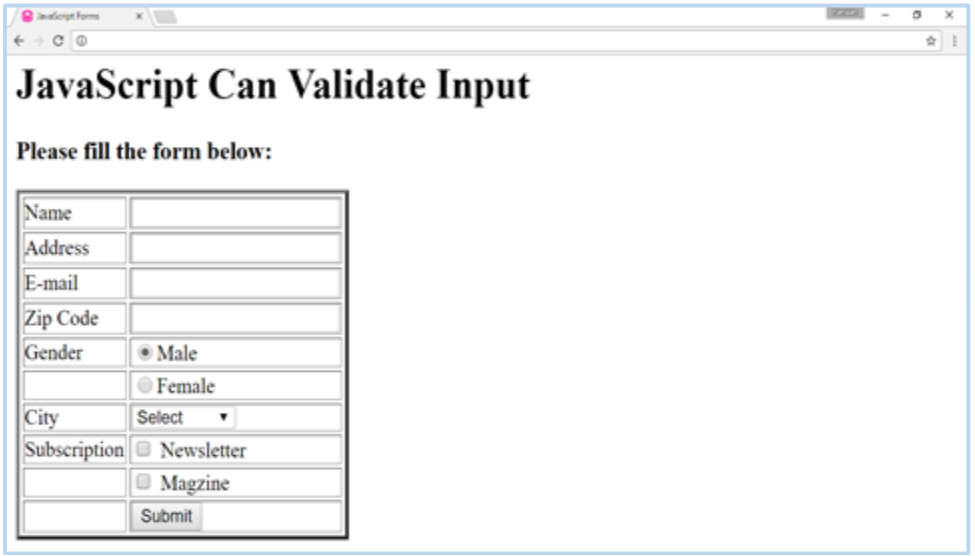

Figure 5.15: Output of code 5.10

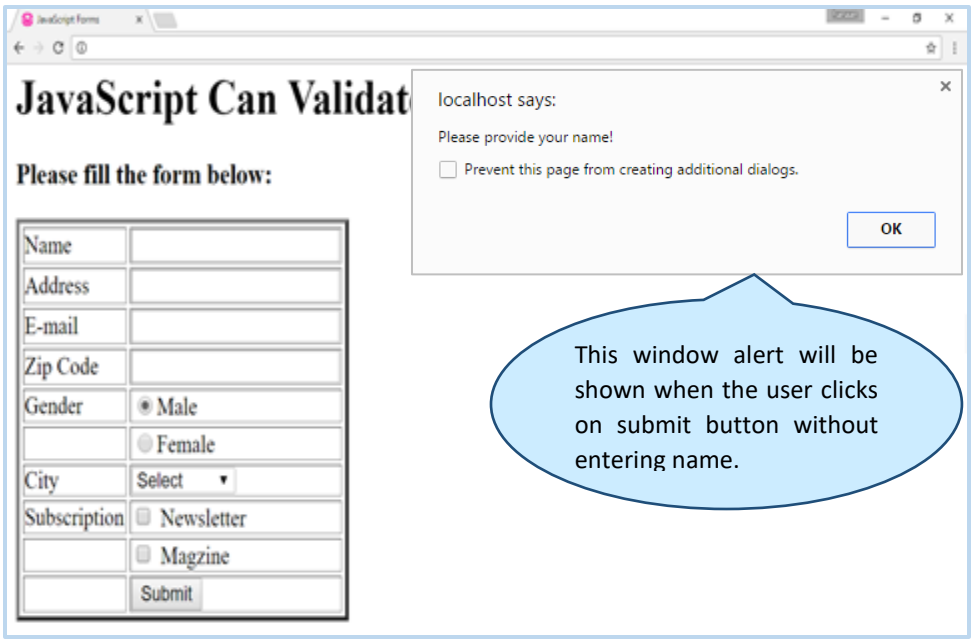

Figure 5.16: Output of code 5.10

If user doesn't enter address, the following output is shown.

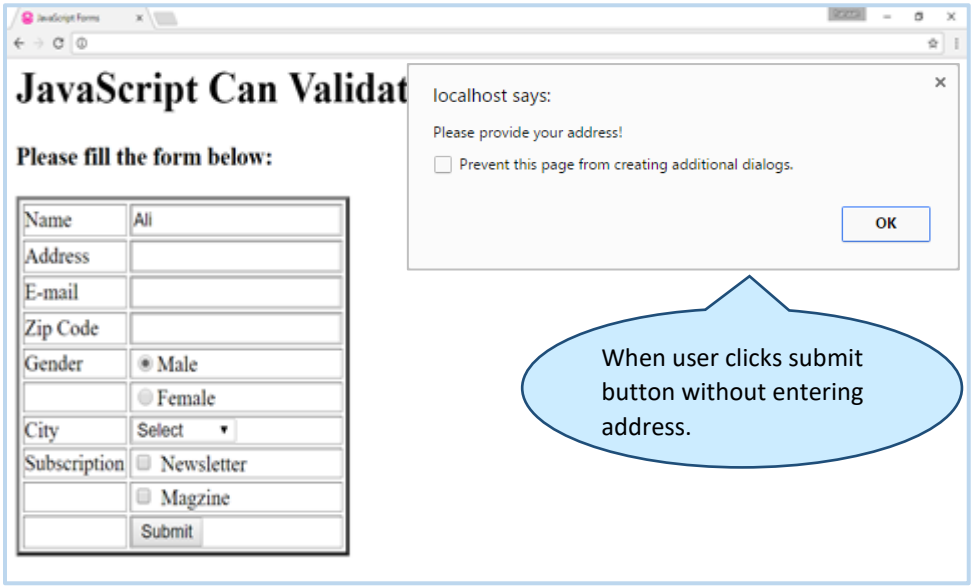

Figure 5.17: Output of code 5.10

If user doesn't enter email address, the following output is shown.

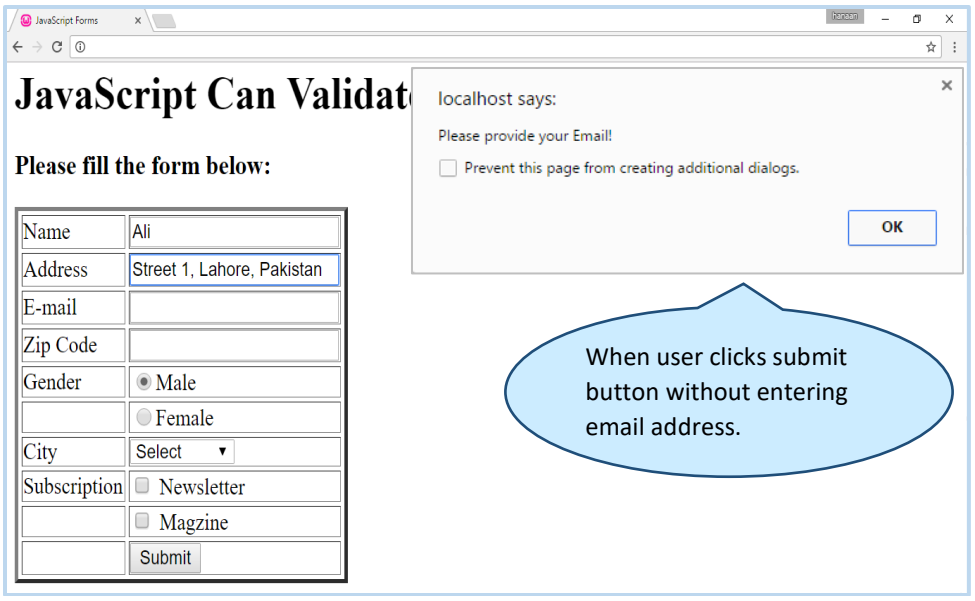

Figure 5.18: Output of code 5.10

If user doesn't valid email address, the following output is shown.

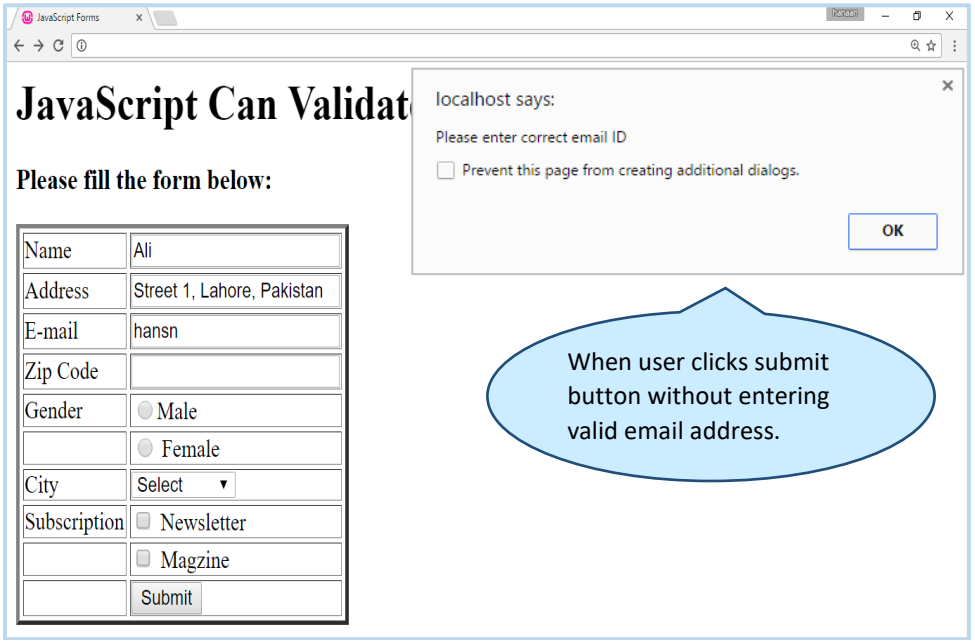

Figure 5.19: Output of code 5.10

If user doesn't enter zip code, the following output is shown.

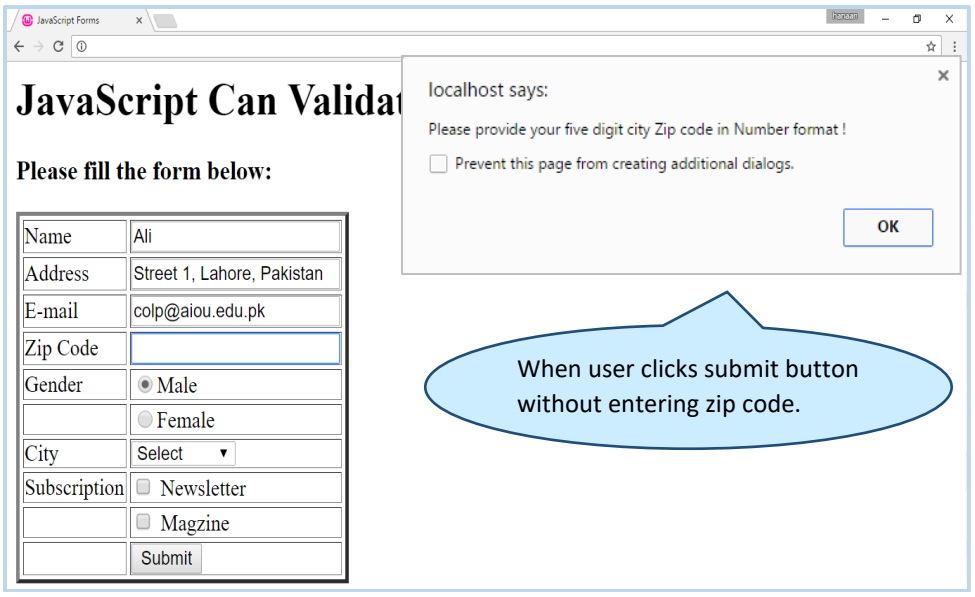

Figure 5.20: Output of code 5.10

If user doesn't enter correct zip code, the following output is shown.

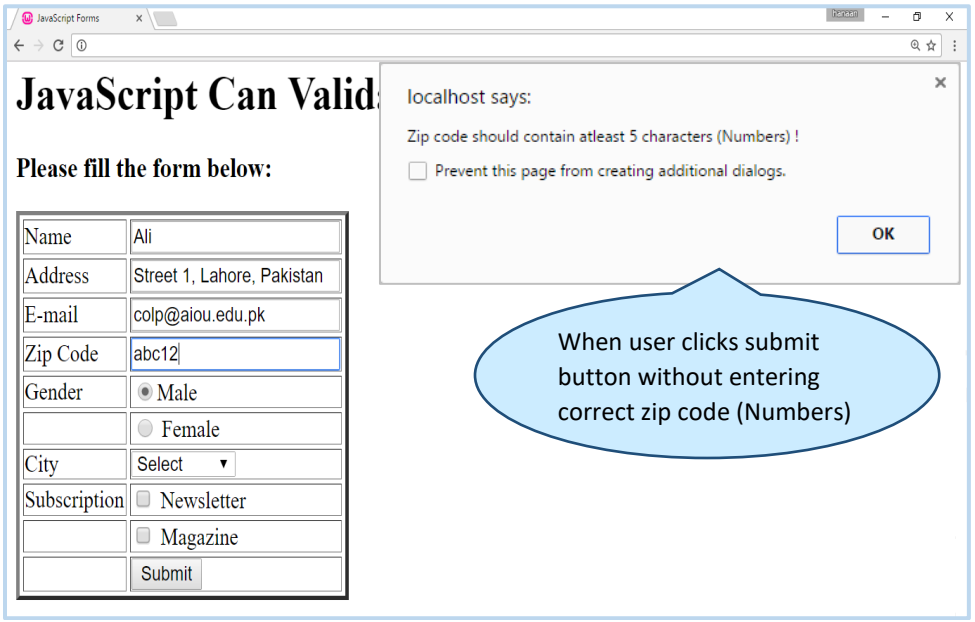

Figure 5.21: Output of code 5.10

If user doesn't select city, the following output is shown.

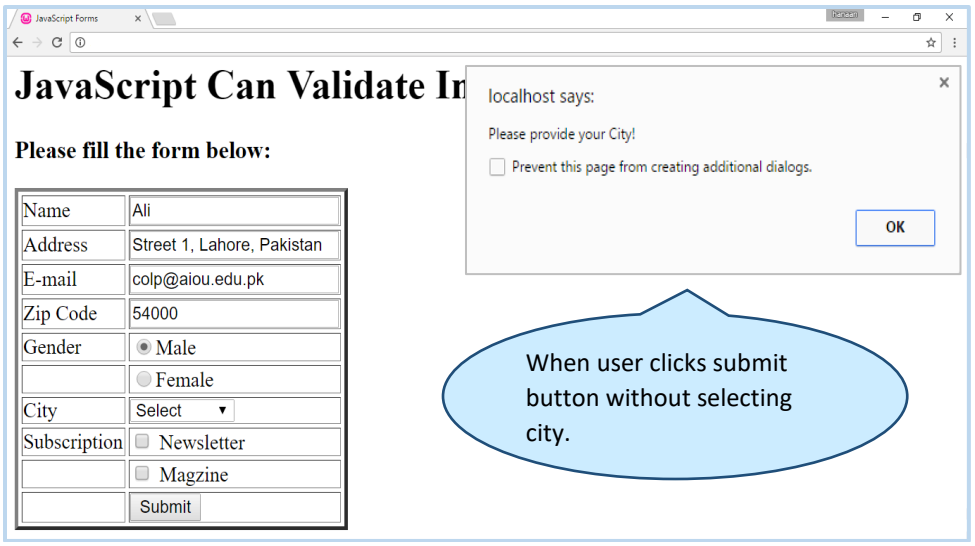

Figure 5.22: Output of code 5.10

Activity 8

Develop a pizza ordering form and apply JavaScript validation techniques.

Unit Summary

In this unit, you have learnt to incorporate different conditions in the code. You can also handle different events in your program. Now you can use loops and functions to make your code more effective and efficient. The form validation technique can be used effectively to validate the user input.

Self Assessment Questions

Choose the correct answer.

1. JavaScript is a _____ language.
   A. programming
   B. system
   C. scripting
   D. All of the above.

2. Where is the correct place to insert a JavaScript code in HTML document?
   A. The <body> section
   B. Both the <head> section and the <body> section
   C. The <head> section
   D. None of the above

3. How can you write an “if” statement in JavaScript?
   A. if i = 5 then
   B. if i == 5 then
   C. if (i == 5)
   D. if i = 5

4. How does a “while” loop start?
   A. while i = 1 to 10
   B. while (i <= 10)
   C. while (i <= 10; i++)
   D. None of the above

5. What is the syntax of “for” loop in JavaScript?
   A. for i = 1 to 5
   B. for (i <= 5; i++)
   C. for (i = 0; i <= 5; i++)
   D. for (i = 0; i <= 5)

6. Which event occurs when the user clicks on an HTML element?
   A. onmouseover
   B. onclick
   C. onmouseclick
   D. onchange

7. Is JavaScript a case-sensitive language?
   A. Yes
   B. No

8. Can JavaScript variables be declared at run time?
   A. Yes
   B. No

9. Which event is used to display a keyboard input?
   A. click
   B. mouseover
   C. on keydown
   D. None of the above

10. _____ event is used to change the color of text when mouse is moved over the text?
   A. on keydown
   B. mouseover
   C. click
   D. None of the above

Answer Key:

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

Review Questions

Write short answers of the following questions

1. Name some of the JavaScript features.
2. What is a named function in JavaScript? How can we define a named function?
3. Describe few of the events in JavaScript which are not explained in this unit.
4. What is the difference between while loop and do while loop?
5. How are different input types validated in JavaScript?

</> Coding Exercise

1. Write the output of the following code:

```
(myFunction()
 {
     var x = y = 5;
  }) ();
document.getElementById("demo").innerHTML = y;
```

2. Write a code in Javascript to check if the temperature of last five days goes below zero.
3. Write a code in JavaScript using for loop to print odd numbers from 50-100.
4. Write a code using while loop in Javascript that prints prime numbers from 100-150.
5. Modify the admission form created in coding exercise of unit 2 and apply validation techniques to different form elements .

References and Further Reading

1. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
2. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH.
3. Rauschmayer, A. (2014). Speaking JavaScript: An In-Depth Guide for Programmers. O'Reilly Media, Inc.
4. Elliott, E. (2014). Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. O'Reilly Media, Inc.
5. JavaScript Scope. Available on: https://www.w3schools.com/js/default.asp
6. JavaScript. Available on: https://www.tutorialspoint.com/javascript/index.htm
7. JavaScript Tutorial. Available on: https://www.w3schools.com/js/default.asp
8. JavaScript Tutorial. Available on: https://www.tutorialspoint.com/javascript/index.htm
9. JavaScript Loops. Available on: https://www.eduonix.com/blog/web-programming-tutorials/javascript-loops/
10. JavaScript Online Examination. Available on: http://www.sreejobs.com/onlineexam/exam.php?exm=JS
11. JSON. Available on: https://www.slideshare.net/RafaelMontesinosMuoz/json-tutorial-a-beguiner-guide

UNIT 6

JavaScript – III

Introduction

JavaScript can be used to do a lot of interaction with browser. Until now you have learned the basics of JavaScript, its syntax, elements, events and core methods. You are well aware of using JavaScript and you can use it to show any popup or validate any web form. In this unit, we will learn how to use Document Object Model (DOM) to access and modify the properties of any web document. This unit will also cover methods and events associated with DOM. You will learn about JavaScript screen, location, navigator and popup boxes. This unit will also cover how to read the location of web pages, the operating system being used in order to use the right set of JavaScript code.

Unit Outcomes

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

1. Understand DOM and its working.
2. Use DOM Methods, Document and events in JavaScript.
3. Determine values of screen width and height.
4. Get location of a web page in JavaScript.
5. Understand the working of the JavaScript navigators.
6. Use popup boxes in JavaScript code.

Terminologies

DOM:	Document Object Model, an application programming interface which is used to define logical structure of the documents.
Screen:	A method which returns current screen settings.
Location:	A method which returns location of the web page.
Navigator:	A method which provides information about visitor's browser.
Popup Boxes:	A small window that displays information in the browser and take user's response.

6.1. DOM Introduction

As we know that the Web pages have different form of the representation and can either be shown in the web browser or viewed as HTML source file. But both these forms are two different ways of representing the same document. The DOM is just another efficient representation of these web pages. It provides a unique way to display, store and modify the same document. It utilizes objects which contains some methods and properties. The methods can be used to perform a specific task whereas the properties are used to configure the object. The hierarchal representation of DOM is shown in figure 6.1. All these objects can be accessed to modify the window or document structure, style and contents. Window object represents one open window in a browser. The document object resides under the HTML tag. Location object stores the URL details of the web page and History object store the URL's that are visited earlier.

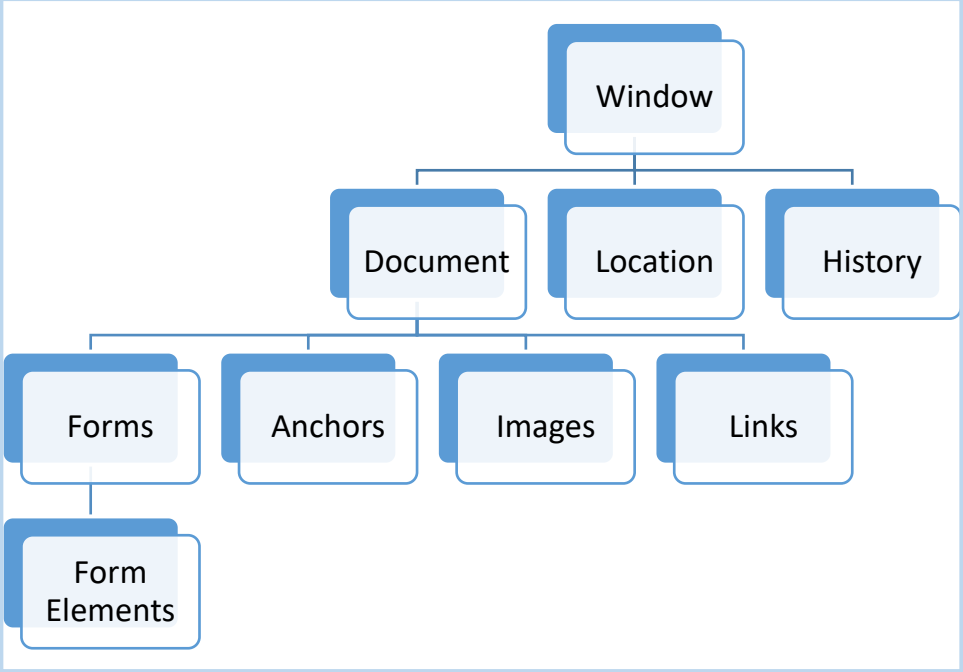

Figure 6.1: Hierarchal representation of DOM.

DOM defines a standard for managing and accessing documents.

6.2. DOM Methods

Each DOM object contains two vital elements inside them. One is DOM methods and the other is configuration setup values. The methods are used to perform actions on required objects and properties are values that can be set to perform these actions. The code 6.1 demonstrates the use of the DOM values and methods.

</> Code 6.1

```
<! DOCTYPE html>
<html>
<head>
<title>DOM Methods</title>
</head>
<body>
<h1>DOM Methods</h1>
<p id="demo"> </p>
<script>
    document.getElementById("demo").innerHTML = "Accessing and
    Changing HTML Elements Value with JavaScript is Fun!";
</script>
</body>
</html>
```

When the above JavaScript statements execute in your browser, you will see the output as shown in figure 6.2. Please note that a method “getElementById” is used in this object which returns any of the elements in the html code having same id. Secondly, the property “innerHTML” is used to get or set the value of HTML element.

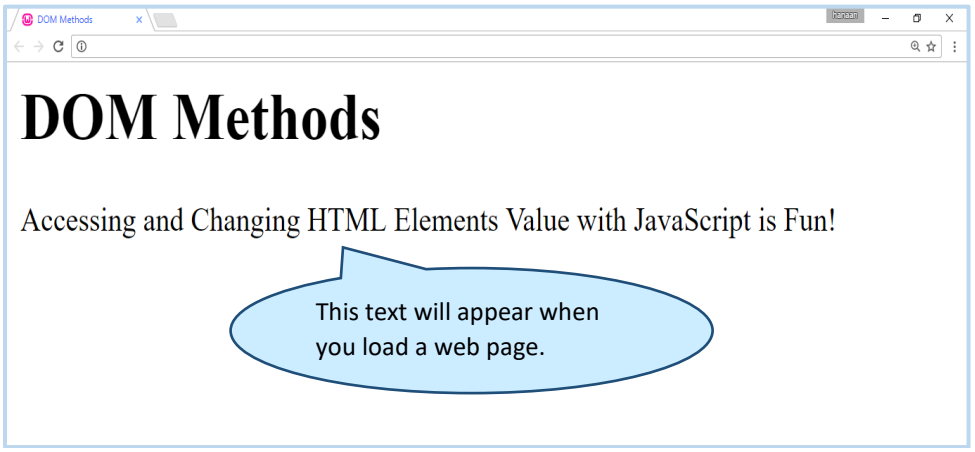

Figure 6.2: Output of code 6.1

Each HTML element has “innerHTML” property.

6.3. DOM Document

As we know that DOM has a hierarchal representation structure of webpage and its elements can be identified as the child of the document object as shown in figure 6.1. We can access or manipulate any HTML element in webpage by using the document object and id of the element as shown in the following syntax.

```
document.getElementById(id)
```

The HTML DOM document object is the parent of all other objects in web page hierarchy.

Code 6.2 demonstrates the use of this property. The sentence "This is all about DOM Document !" is displayed in two stages. The second portion is displayed through "innerHTML" property.

</> Code 6.2

```
<! DOCTYPE html>
<html>
<head>
<title>DOM Document</title>
</head>
<body>
<p id="dom">DOM Document!</p>
<p>This example demonstrates the <b>getElementById</b> method!</p>
<p id="demo"></p>
<script>
    var domDocument = document.getElementById("dom");
        document.getElementById("demo").innerHTML = "This is all
        about " + domDocument.innerHTML;
</script>
</body>
</html>
```

When you execute the above JavaScript statements in your browser, you will see the output shown in figure 6.3.

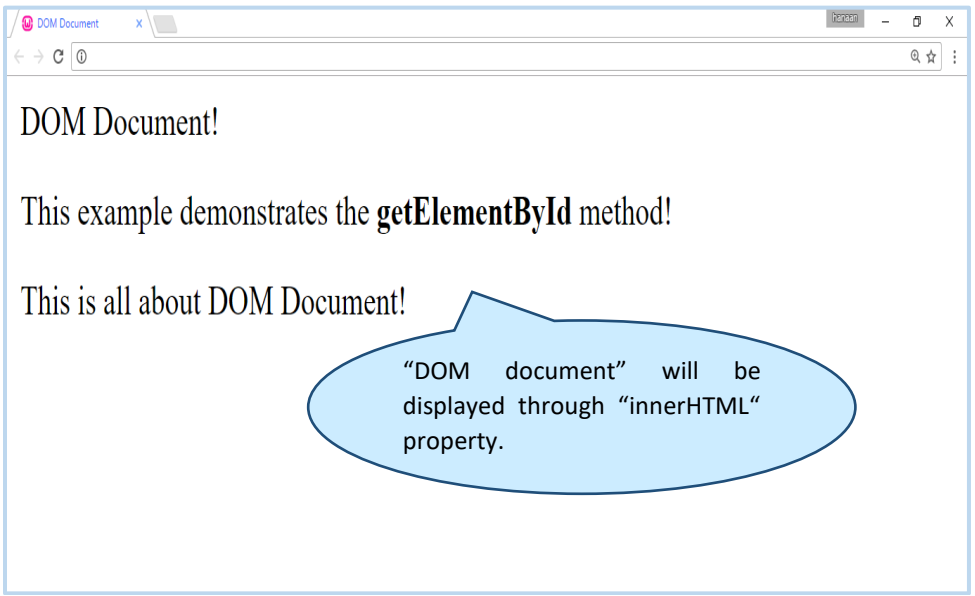

Figure 6.3: Output of code 6.2

6.4. DOM Events

As we know that Interaction of JavaScript with HTML is handled through events. These events are used to execute JavaScript code in response like buttons to close windows, messages to be displayed to users, data to be validated etc.

Now let us do a practical example as demonstrated in code 6.3 in which the response to the event "onclick" is used to change a piece of text.

</> Code 6.3

```
<! DOCTYPE html>
<html>
<head>
<title>DOM Events</title>
</head>
<body>
<h1 onclick="changeText(this)">Click this text to see change!</h1>
// "onclick" & "changeText" results in the change of text when user
   clicks on the text.
<script>
    function changeText(id) {
    id.innerHTML = "Text has been changed!";
}
</script>
</body>
</html>
```

The output of the code 6.3 is shown in figure 6.4 and 6.5.

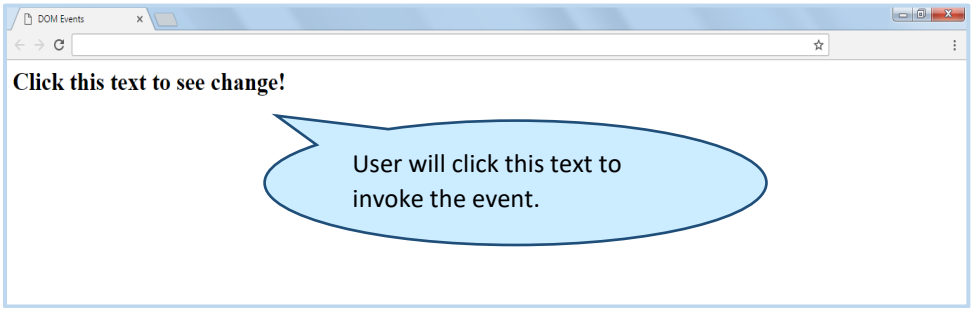

Figure 6.4: Output of code 6.3

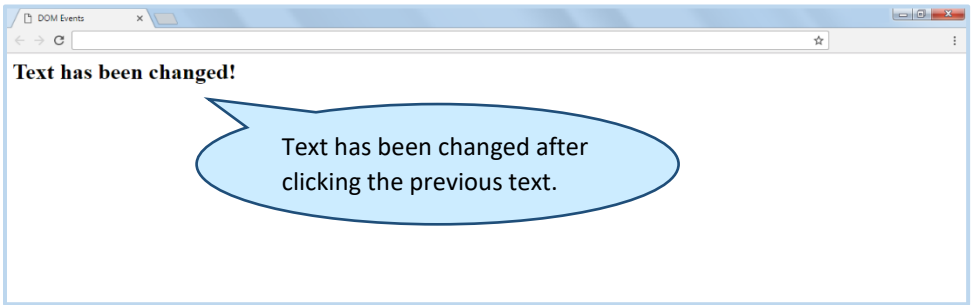

Figure 6.5: Output of code 6.3

Activity 1

Modify the code 6.3 so that the original text may reappear on second click.

Video Lecture

https://youtu.be/yxZgvoESBTE

6.5. JavaScript Window Screen

The "window.screen" object contains information about the user screen. It can also be written without the window prefix. This object contains some informative property about the size of the window. Table 6.1 shows some important properties of JavaScript Window Screen.

Table 6.1: JavaScript Window Objects

Conditions	Description
`screen.width`	Returns the width of a screen.
`screen.height`	Returns the height of a screen.
`screen.colorDepth`	Returns the value of color depth of the application.
`screen.pixelDepth`	Returns the value of color depth of the monitor.

Screen object is used to inspect the properties of the screen.

screen.width:

The screen.width property is used to show width of the visitor's screen in pixels. It depends upon the current resolution setting of the screen. Now let us read an example to demonstrate the parameter related to screen as shown in code 6.4.

</> Code 6.4

```
<! DOCTYPE html>
```

```
<html>
<head>
<title> Javascript Window Screen </title>
</head>
<body>
<h3>Screen Width</h3>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The width of the
    screen is = </b>" + screen.width + "pixels";
   // Calculates the width of screen & shown using id=demo.

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

Output of the code 6.4 is shown in figure 6.6.

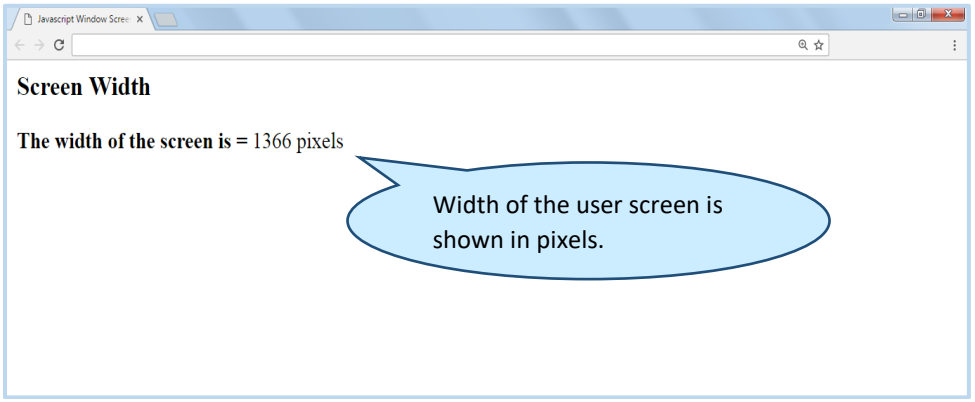

Figure 6.6: Output of code 6.4

screen.height:

The screen.height property is used to show the height of the visitor's screen in pixels. Code 6.5 shown an example of “screen.height” property.

</> Code 6.5

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Screen </title>
</head>
<body>
<h3>Screen Height</h3>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The height of
    the screen is = </b>" + screen.height + "pixels";
    // Calculates the height of screen & shown using id=demo.
</script>
</body>
</html>
```

Output of code 6.5 is shown in figure in 6.7.

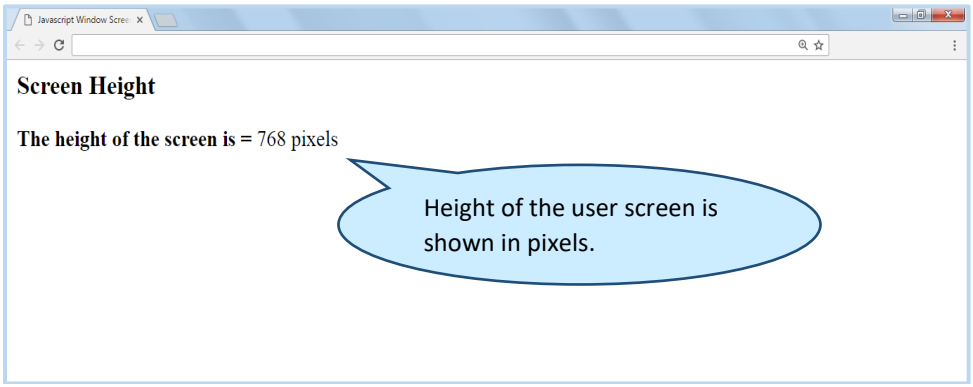

Figure 6.7: Output of code 6.5

screen.colorDepth:

The screen.colorDepth property shows the number of bits used to display one color. Both 24 and 32 bit color resolution are used to describe color range. Code 6.6 shows the color resolution settings of the user display.

</> Code 6.6

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Screen </title>
</head>
<body>
<h3>Screen ColorDepth</h3>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The ColorDepth
    of the screen is = </b>" + screen.colorDepth + "bits";
    // Calculates the colorDepth of screen & shown using id=demo.
</script>
</body>
</html>
```

The output of the code 6.6 is shown in figure 6.8.

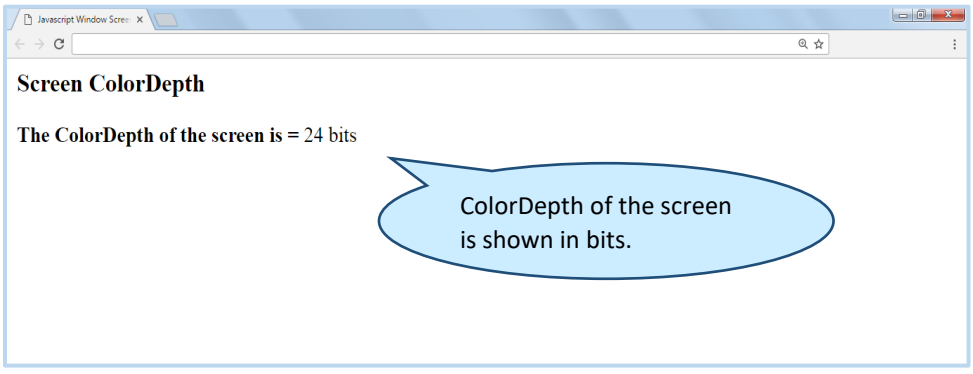

Figure 6.8: Output of code 6.6

Activity 2

Try the same code with different screen resolutions and color settings.

6.6. JavaScript Window Location

The window.location object can be used to get information about the current location of the document. It can also be written without the window prefix. Table 6.2 shows some important properties of Window Location.

Window location is used to get the current location of the document.

Table 6.2: JavaScript Window Objects

Conditions	Description
window.location.href	It returns the URL of current page.
window.location.hostname	It returns the domain name of the web host.
window.location.pathname	It returns the path and file name of the current page.
window.location.protocol	It returns the web protocol used i.e. (http: or https :)

window.location.href:

The window.location.href property returns the URL of the current page as shown in code 6.7.

</> Code 6.7

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Location</title>
</head>
<body>
<h2> Window Location</h2>
<h3>This is the explanation of window.location object</h3>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The complete URL
    of this page is: </b>" + window.location.href;
    // Returns the location of a web page.
</script>
</body>
</html>
```

Output of code 6.7 is shown in figure 6.9. Please note that the output shows the current location of HTML file in "C" drive.

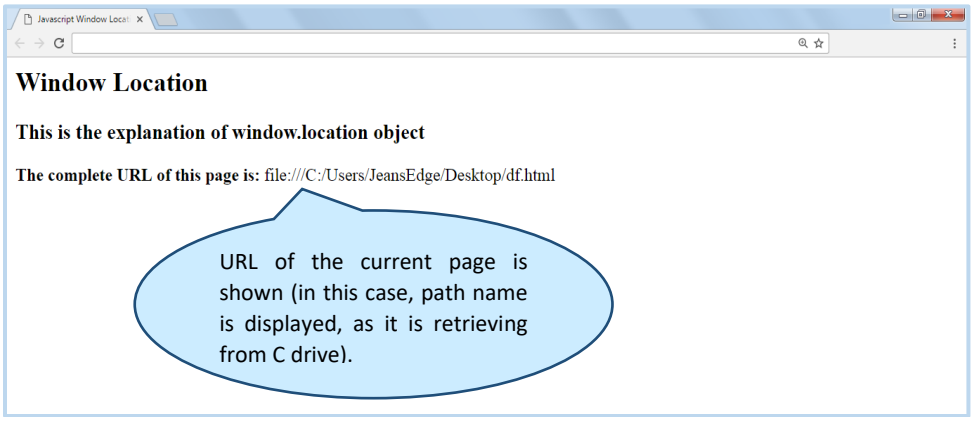

Figure 6.9: Output of code 6.7

window.location.hostname:

The window.location.hostname property returns the name of the internet host (of the current page). Code 6.8 demonstrates the use of "hostname" property.

</> Code 6.8

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Location Hostname</title>
</head>
<body>
   <h2> Window Location</h2>
   <h3>This is the explanation of window.location object</h3>
   <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The hostname of
    this page is: </b>" + window.location.hostname;
    // Returns the name of internet host.
</script>
</body>
</html>
```

Output of the code 6.8 is shown in figure 6.10. Currently there is no definite host therefore the host name is not being displayed in the web page.

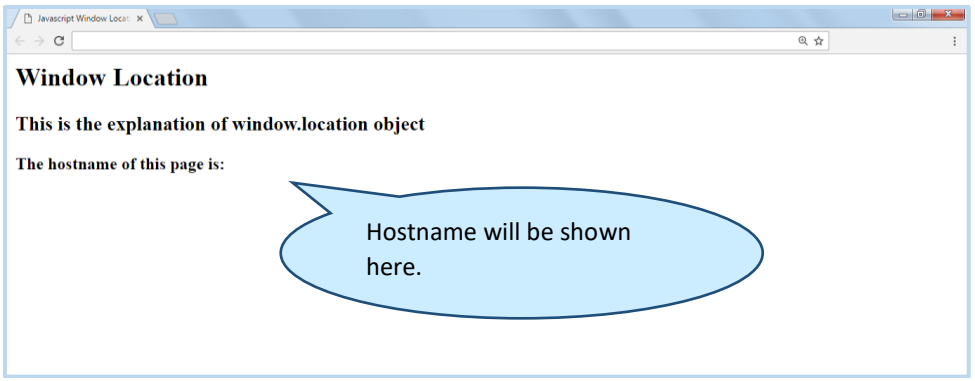

Figure 6.10: Output of code 6.8

window.location.pathname:

The window.location.pathname property returns the pathname of the current page. Code 6.9 demonstrates the use of "pathname" property.

</> Code 6.9

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Location Pathname</title>
</head>
<body>
<h2> Window Location</h2>
<h3>This is the explanation of window.location object</h3>
  <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The pathname of
    this page is: </b>" + window.location.pathname;
    // Returns the pathname of a current web page.
</script>
</body>
</html>
```

Output of code 6.9 is shown in figure 6.11. it is also used to keep track of the history.

Figure 6.11: Output of code 6.9

window.location.protocol:

The window.location.protocol property returns the web protocol of the page. Code 6.10 demonstrates the use of “protocol” property.

</> Code 6.10

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Location Protocol</title>
</head>
<body>
  <h2> Window Location</h2>
  <h3>This is the explanation of window.location object</h3>
  <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>The protocol of
   this page is: </b>" + window.location.protocol;
    // Returns the protocol of a current web page.
</script>
</body>
</html>
```

Output of code 6.10 is shown in figure 6.12. Either “http:” or “https:” will be shown in the output whatever is applicable.

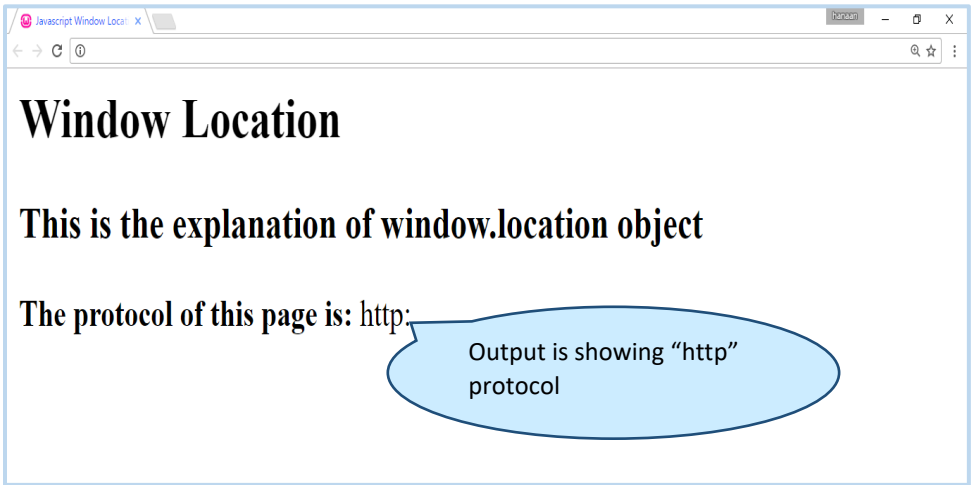

Figure 6.12: Output of code 6.10

Activity 3

Implement code 6.7 to 6.10 by using JavaScript functions in a single web page.

6.7. JavaScript Window Navigator

The “window.navigator” object is used to take information about the visitor's browser. It can also be written without the window prefix. As all the browsers are of different nature, sometimes it is required

to show the information differently on each. The navigator gives useful information about the browser to enable the compatible JavaScript code. Some properties of Windows Navigator are shown in table 6.3.

Table 6.3: JavaScript Window Objects

Conditions	Description
navigator.appName	It returns the application name of the browser.
navigator.appCodeName	It returns the application code name of the browser.
navigator.platform	It returns the information about browser's operating system.

Window Navigator is used to get information about browser and its operating system.

navigator.appName:

The appName property returns the application name of the browser. Code 6.11 demonstrates the use of "appName" property.

</> Code 6.11

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Navigator</title>
</head>
<body>
   <h2> Window Navigator</h2>
   <h3>This is the explanation of navigator object</h3>
   <p id="demo"></p>
   <p>"Netscape" is the application name for all these browsers;
      <b>IE11, Chrome, Firefox, and Safari. </b></p>
<script>
    document.getElementById("demo").innerHTML ="<b>navigator.appName
    is:  </b>" + navigator.appName;
    // Returns the name of a browser in which user loads a web page
</script>
</body>
</html>
```

When we execute the above JavaScript statements in our browser, we will see the output as shown in figure 6.13. Here the application name will be displayed in the output window.

Figure 6.13: Output of code 6.11

navigator.platform:

The navigator.platform property returns the information about browser's Operating System. Code 6.12 demonstrates the use of "platform" property.

</> Code 6.12

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Window Navigator</title>
</head>
<body>
   <h2> Window Navigator</h2>
   <h3>This is the explanation of navigator object</h3>
   <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "<b>Your browser is
    installed in  </b>" + navigator.platform + " Operating System";
    // "navigator.platform" returns the type of an operating system
</script>
</body>
</html>
```

When you execute the above JavaScript statements in your browser, you will see the output shown in figure 6.14. Here the platform information of the operating system is being displayed in the window.

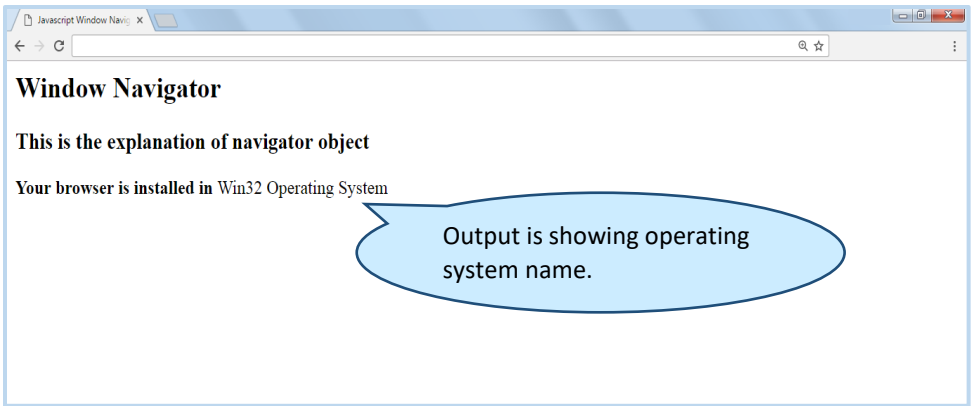

Figure 6.14: Output of code 6.12

Activity 4

Run the code from 6.11 to 6.13 with different operating systems and browsers.

6.8. JavaScript Popup Boxes

Popup boxes are small alert windows that are shown in the browser to display some information. JavaScript has three kinds of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box:

An alert box is a special dialogue box that is displayed when something occurs that requires the user's attention. When an alert box pops up, the user will have to click "OK" to proceed further. Code 6.13 demonstrates the use of an alert box.

</> Code 6.13

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Popup Boxes</title>
</head>
<body>
   <h2> JavaScript Popup Boxes</h2>
   <h5>This is the explanation of Alert Box.</h5>
   <button onclick="myFunction()">Click Me!</button>
    // “onclick” calls a function when button is clicked.
<script>
    function myFunction()
      {
         alert("Yes! This is an Alert Box.");
      }
</script>
</body>
</html>
```

When we execute the above JavaScript statements, we will see the output shown in figure 6.15. The alert statement generates a message box on which the user has to press ok button for the acknowledgement of the message.

Figure 6.15: Output of code 6.13

Confirm Box:

A confirm box is the dialogue box that asks user to approve or disapprove the requested operation. This dialogue box is used when some critical operation is to be performed. When a confirm box pops up, the user will have two options either user can click on "OK" or "Cancel" button to proceed. If "OK" button is clicked by the user, the box will return true and if the "Cancel" is clicked, the box will return false. Code 6.14 demonstrates the use of confirm box.

</> Code 6.14

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Popup Boxes</title>
</head>
<body>
   <h2> JavaScript Popup Boxes</h2>
   <h5>This is the explanation of Confirm Box.</h5>
   <button onclick="myFunction()">Click Me!</button>
<script>
    function myFunction()
      {
         confirm("This is an example of Confirm Box.");
      // Built-in JavaScript function to show pop-up box & take user
         input in form of “ok” or “cancel”.
       }
</script>
</body>
</html>
```

When we execute the above JavaScript statements, we will see the output shown in figure 6.16. This confirm box is used to display some message to the user where you want the user to acknowledge by pressing either ok or cancel button.

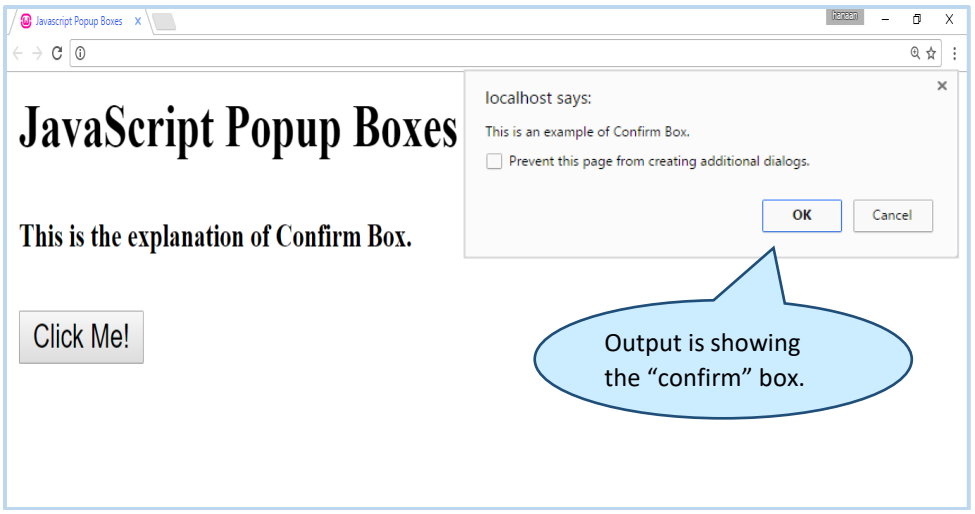

Figure 6.16: Output of code 6.14

Video Lecture

https://youtu.be/QNqtAY2B1mA

Prompt Box:

A prompt box is used when you want the user to input a value. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box will return the input value. If the user clicks "Cancel", the box will return null value (Nothing). Code 6.15 demonstrates the use of prompt box.

</> Code 6.15

```
<! DOCTYPE html>
<html>
<head>
<title> Javascript Popup Boxes</title>
</head>
<body>
  <h2> JavaScript Popup Boxes</h2>
  <h5>This is the explanation of Confirm Box.</h5>
  <button onclick="myFunction()">Click Me!</button>
  <p id="demo"></p>
```

```
<script>
    function myFunction()
      {
        var txt;
        var person = prompt("Please enter your name:", "Your Name");

            // Built-in JavaScript function to show pop-up box & take
            user input in alphabetical form.

        if (person == null || person == "") {
           txt = "You clicked on Cancel button.";
           } else {
           txt = "Very Nice " + person + "! We hope you understand
                   about prompt box.";
           }
           document.getElementById("demo").innerHTML = txt;
      }
</script>
</body>
</html>
```

When we execute the above JavaScript statements, we will see the output as shown in figure 6.17. Once the user clicks on the button, a prompt dialog box will appear as shown in figure 6.18 asking the name of the user. The user can then enter the name and press "OK" button. Afterwards, another text sentence with the user name will appear on the screen as shown in figure 6.19. If the user clicks on the cancel button, the same message will be displayed in the window as shown in figure 6.20.

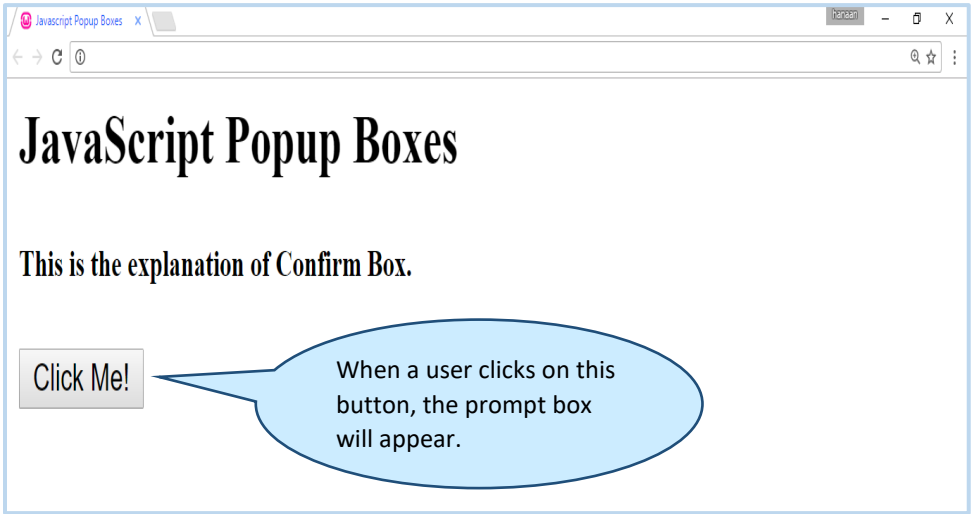

Figure 6.17: Output of code 6.15

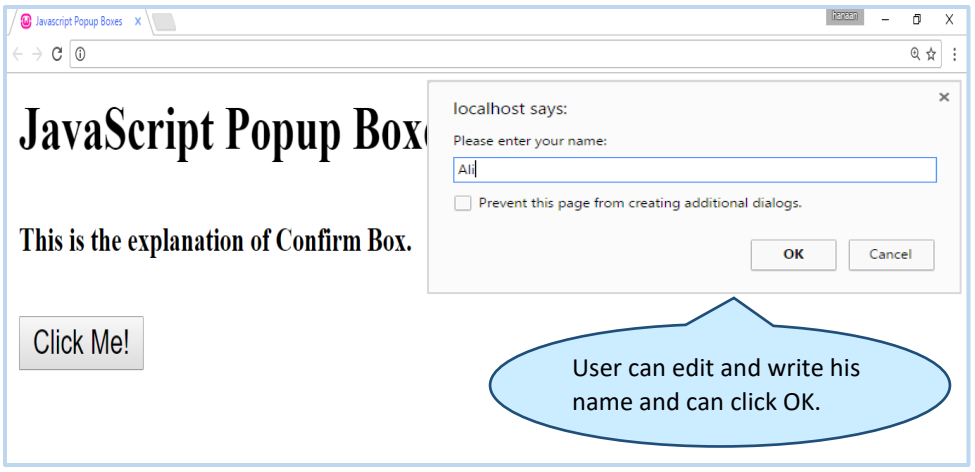

Figure 6.18: Output of code 6.15

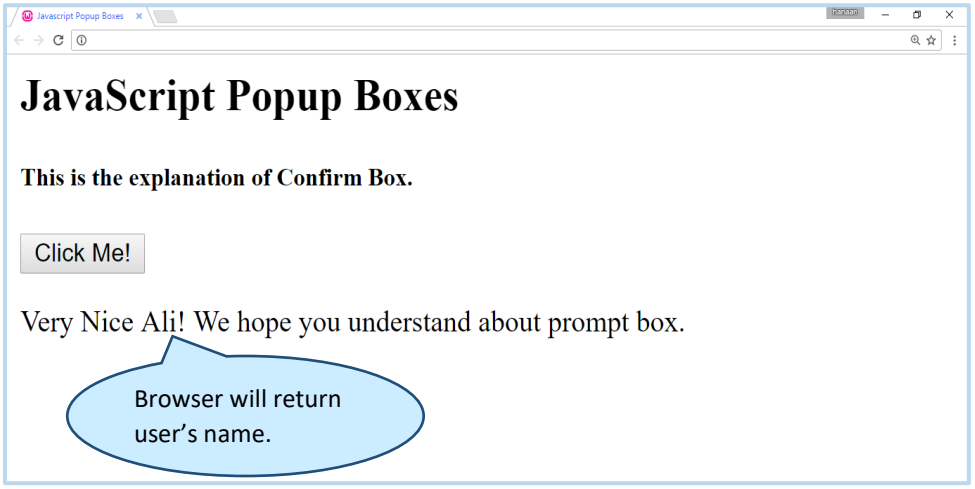

Figure 6.19: Output of code 6.15

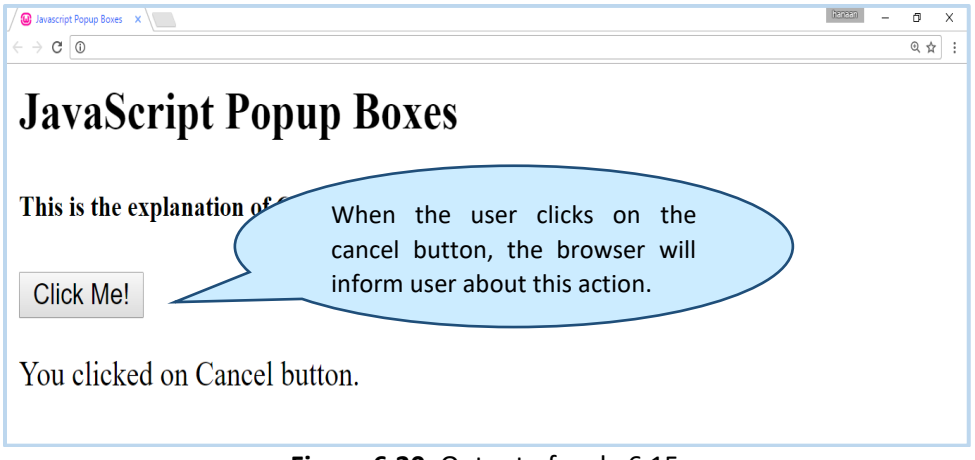

Figure 6.20: Output of code 6.15

Video Lecture

https://youtu.be/W_fbDVX_jNI

Activity 5

Write a JavaScript code that takes two inputs from the user by using prompt box.

Unit Summary

In this unit, we have learnt about DOM along with its associated methods and events. We have also learnt about window screen, window location, window navigator and popup boxes. We can also read the current screen configurations and check our browser details now.

Self Assessment Questions

Choose the correct answer.

1. The element which resides on the root of DOM hierarchy is _____
   A. window
   B. document
   C. location
   D. history

2. “innerHTML” can be associated with HTML tags and content between the tags.
   A. True
   B. False

3. Which is the correct syntax to use screen width property?
   A. document.getElementById("demo").outerHTML = "Screen Width: " + screen.width;
   B. document.getElementById("demo").innerHTML = "Screen Width: " + screen.height;
   C. document.getElementById("demo").innerHTML = "Screen Width: " + screen.width;
   D. document.getElementById("demo").innerJavaScript = "Screen Width: " + screen.width;

4. Which was the first browser to support JavaScript?
   A. Mozilla
   B. Google Chrome
   C. IE
   D. Netscape

5. If para1 is the DOM object for a paragraph, what is the correct syntax to change the text within this paragraph?
   A. "New Text"?
   B. para1.value="New Text";
   C. para1.firstChild.nodeValue= "New Text";
   D. para1.nodeValue="New Text";

6. The code: <script>document.write (navigator.appCodeName); </script>
   A. set code name of the browser of a visitor
   B. get code name of the browser of a visitor
   C. Both A and B
   D. None of the above

7. To which object does the location property belong?
   A. window
   B. position
   C. location
   D. element

8. _____ scripting embedded in an HTML page can respond to user events such as mouse-clicks, form input, and page navigation.
   A. Server-side
   B. Client-side
   C. Local
   D. Native

9. _____ returns the URL of the current page.
   A. window.location.href
   B. window.location.hostname
   C. window.location.pathname
   D. window.location.protocol

10. _____ returns the application name of the browser.
    A. navigator.appCodeName
    B. navigator.appName
    C. navigator.platform
    D. None of the above.

Answer Key:

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

Review Questions

Write short answers of the following questions

1. Explain the working and concept behind DOM?
2. How are DOM methods used in JavaScript?
3. How different DOM events can be associated with our JavaScript code?
4. Which property returns the complete URL of the current document?
5. What value is returned by the confirm() method if the viewers clicks on OK button?
6. Briefly explain the working and purpose of popup boxes in JavaScript.
7. How can we get the size (width and height) of the monitor screen?

</> Coding Exercise

1. Here is a sample html file with a submit button. Modify the style of the paragraph text using “innerHTML” property .

```
<!DOCTYPE html>
<html>
<head>
<title>JS DOM paragraph style</title>
</head>
<body>
<p id ='text'> </p>
<div>
<button id="jsstyle" onclick="js_style()">Style</button>
</div>
</body>
</html>
```

What will be printed in the browser? On clicking the “style” button the font, its size, and color of the paragraph text should change.

2. Write a JavaScript program to set the background color of a paragraph.
3. Write a JavaScript program to count and display the items of a dropdown list, in an alert window.
4. Write a JavaScript program to get the width and height of the window.
5. Write a JavaScript program to calculate the volume of a sphere (User have to input the value of Radius) by using prompt box.

References and Further Reading

1. Simpson, K. (2012). JavaScript and HTML5 Now. O'Reilly Media, Inc.
2. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH.
3. Rauschmayer, A. (2014). Speaking JavaScript: An In-Depth Guide for Programmers. O'Reilly Media, Inc.
4. Elliott, E. (2014). Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. O'Reilly Media, Inc.
5. JavaScript Scope. Available on: https://www.w3schools.com/js/default.asp
6. JavaScript. Available on: https://www.tutorialspoint.com/javascript/index.htm
7. JavaScript Tutorial. Available on: https://www.w3schools.com/js/default.asp
8. JavaScript Tutorial. Available on: https://www.tutorialspoint.com/javascript/index.htm
9. JavaScript Window Object. Available on: http://www.w3resource.com/javascript/client-object-property-method/window.php
10. JavaScript Basics. Available on: http://www.tasheva.info/en/lectures/FPI_lecture6.pdf
11. JavaScript DOM. Available on: http://www.w3resource.com/javascript-exercises/javascript-dom-exercises.php

UNIT 7

jQuery

Introduction

JQuery is a JavaScript library used to develop efficient JavaScript code by utilizing its rich collection of functions. JQuery was first released by John Resig in 2006 at BarCamp NYC. By using jQuery, we are, in fact calling javascript functions which are designed to achieve many common functionalities. It can be considered as the compact version of JavaScript, because many tasks that require a number of lines in JavaScript can be accomplished by a single line of code through JQuery.

Unit Outcomes

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

1. Install jQuery in your computer.
2. Understand the syntax of jQuery and use jQuery Selectors.
3. Use events and effects to make your web pages interactive.
4. Use jQuery to validate different input fields of form.

Terminologies

CDN:	Content Delivery Network, a method of using jQuery library online, without installing it.
Effects:	jQuery effects can be used to animate elements in a web page.
AJAX:	Asynchronous JavaScript and XML, is a client side scripting technique.

7.1. How to Use jQuery

We can use the following two styles to use jQuery library to write and execute code in HTML document.

Local Installation

For local installation, we can download the jQuery library from its official website https**://jquery.com/download**. After downloading, copy the jQuery file "jquery.min.js" in the directory of your project and reference it as shown in code 7.1.

</> Code 7.1

```
<!DOCTYPE html>
<html>
<head>
<title>Local Installation of jQuery</title>
<script type = "text/javascript" src = "jquery.min.js">
    :
</script>
<script>
$(document).ready(function(){
document.write("Example of Local Installation of jQuery !");
});
</script>
</head>
 <body>

</html>
```

The output of code 7.1 is shown in figure 7.1.

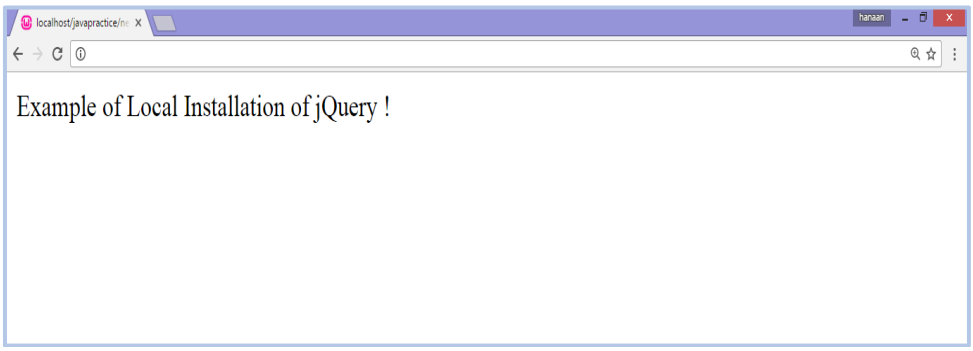

Figure 7.1: Output of code 7.1

CDN Based Version

CDN stands for Content Delivery Network. If you are using CDN version of JQuery, you need to provide the link of that jQuery file instead of downloading it. You can simply include the jQuery library in to your HTML document as shown in code 7.2.

By using Content Delivery Network, you can use jQuery without installing it on your computer.

</> Code 7.2

```
<!DOCTYPE html>
<html>
<head>
<title>Local Installation of jQuery</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
   :
</script>
<script>
  $(document).ready(function(){
     document.write("Example of CDN Based Version !");
   });
</script>
</head>
 <body>
</html>
```

The output of code 7.2 is shown in figure 7.2.

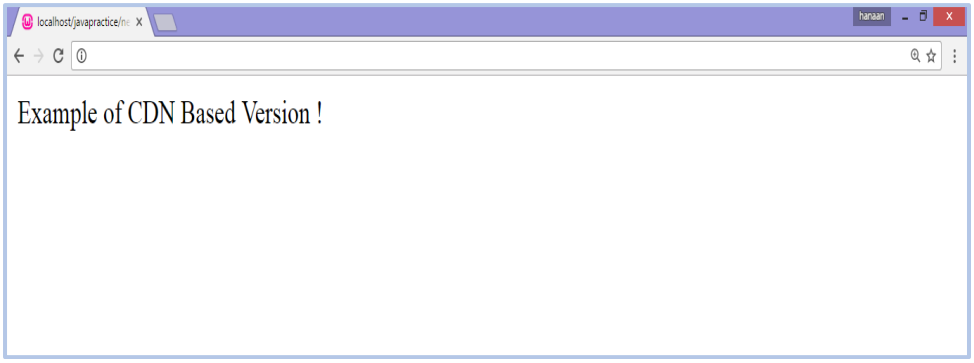

Figure 7.2: Output of code 7.2

Activity 1

Install jQuery library in your computer and explore few of its functions.

Syntax:

jQuery has its own syntax which is used to select the HTML element and perform some actions on it. Between the ($) and (;), following two parameters must be placed.

- A “selector” which is used to "query (or find)" HTML elements.
- A jQuery “action ()” to be performed on selection.

jQuery statement starts with ($) sign and ends with a semicolon (;).

7.2. jQuery Selectors

jQuery Selectors are used to select one or more HTML elements. Once an element is selected, we can perform various operations on it. Please note that selectors are placed inside parenthesis () after the dollar sign $. Table 7.1 shows the list of jQuery selectors and their examples.

Selectors are generally accessed by their name and their attributes.

Table 7.1: jQuery Selectors

Selector	Example	Selects
*	$("*")	All elements
#id	$("#name")	The element with id = "name"
.class	$(".intro")	All elements with class="intro"
element	$("p")	All <p> elements
:even	$("tr:even")	All even <tr> elements
:odd	$("tr:odd")	All odd <tr> elements
:header	$(":header")	All header elements <h1>, <h2> ...
:focus	$(":focus")	The element that currently has focus
:hidden	$("p:hidden")	All hidden <p> elements
[attribute]	$("[href]")	All elements with a href attribute
[attribute!=value]	$("[href!='default.htm']")	All elements with a href attribute value not equal to "default.htm"
:radio	$(":radio")	All input elements with type="radio"
:submit	$(":submit")	All input elements with type="submit"
:button	$(":button")	All input elements with type="button"
:image	$(":image")	All input elements with type="image"
:file	$(":file")	All input elements with type="file"

7.2.1. Element Selector

The jQuery Element Selector selects different elements based on their name. If we want to select all paragraphs on a webpage, we can use: $("p"). The code 7.3 demonstrates the use of element selector. When a user clicks on the button, all paragraphs will become invisible.

</> Code 7.3

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
```

```
</script>
</head>
 <body>
   <h2>This is a Heading</h2>
   <p>This is Paragraph </p>
   <p>This is Second Paragraph</p>
   <p>when you click on ‘Hide’ button, it will hide all
    paragraphs</p>
   <button>Hide</button>
</html>
```

The output of code 7.3 is shown in figure 7.3.

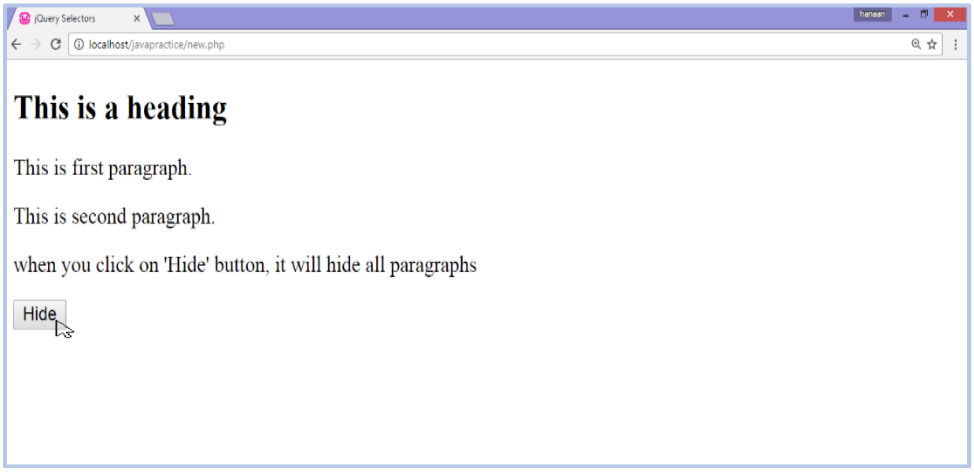

Figure 7.3: Output of code 7.3

When we click on ‘Hide’ button, it hides all text that are written in <p> tag as shown in figure 7.4.

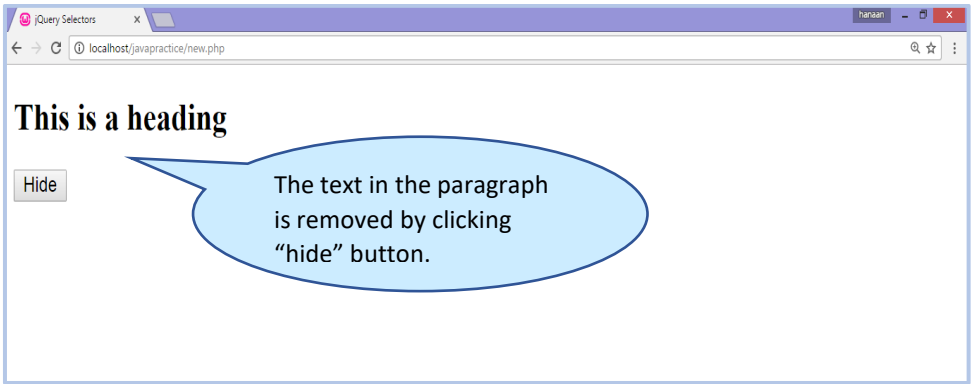

Figure 7.4: Output of code 7.3

Activity 2

Modify code 7.3 to incorporate “Show” in order to display hidden paragraph.

7.2.2. ID Selector

ID selector is used to find a specific element of an HTML tag. If one “id” is assigned to more than one elements, only the first matched id is used.

To find a specific element in a webpage, id selector is used.

A hash character is used to find the element with specific id like: `$("#test").` Here "test" is the "id" of the paragraph. Code 7.4 demonstrates the use of ID selector.

</> Code 7.4

```
<!DOCTYPE html>
<html>
<head>
<title>ID Selectors</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
  $(document).ready(function(){
      $("button").click(function(){
         $("#test").hide();
      // It hides a paragraph which has id "test".
    });
  });
</script>
</head>
 <body>
    <h2>This is a Heading</h2>
    <p>This is a Paragraph </p>
    <p id = "test">This is Second Paragraph</p>
    <p>Click on 'Hide' button to hide second Paragraph. </p>
    <button>Hide</button>
 </body>
</html>
```

When a user clicks on the button, the element with id="test" will be hidden from the webpage as shown in figure 7.5.

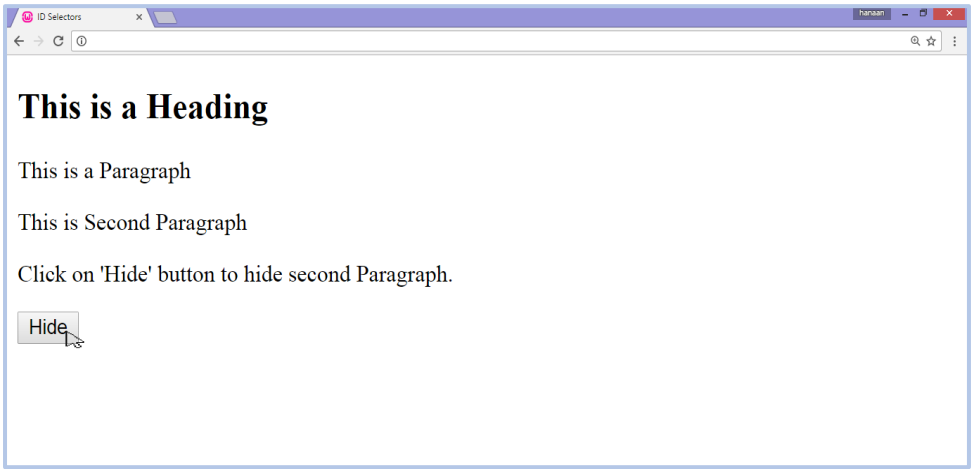

Figure 7.5: Output of code 7.4

When we click on 'Hide' button, it will hide all text that are written in `<p id = "hide">` as shown in figure 7.6.

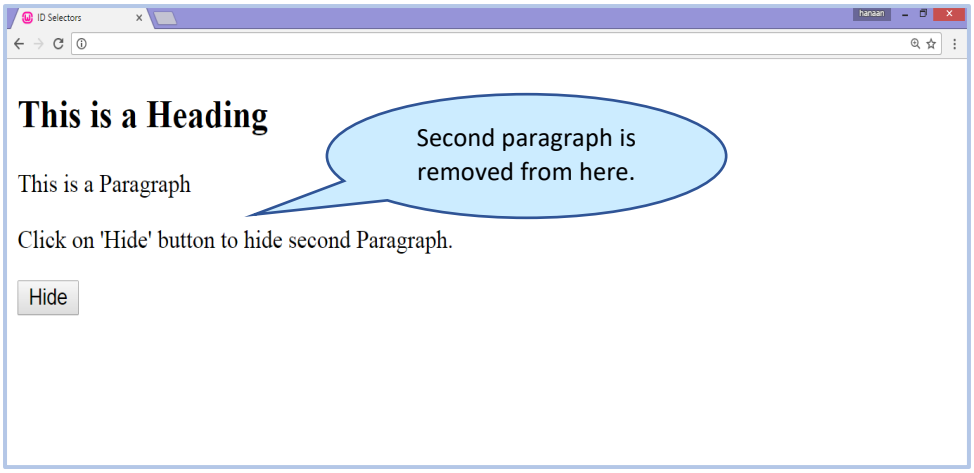

Figure 7.6: Output of code 7.4

Activity 3

- Write three paragraphs "Hello World", "I am new to jQuery" and "jQuery is a library of javascript" with three different ids "1", "2" and "3" respectively. You have to hide and show all paragraphs with the help of id selectors.
- Create a web page that has few lists of items in it. Use selectors to find all elements in the list. Change color of elements to “red” using element and id selector.

Video Lecture

https://youtu.be/TmwB1j1-2iU

7.3. jQuery Events

Web page can respond to all events as desired or programmed. An event represents the precise moment when something happens. There are many actions which are performed by user such as clicking on a link, checking or unchecking radio button, pressing a key from keyboard etc. By using jQuery, we can perform the desired action by using “event listener”. Table 7.2 lists some jQuery events with their descriptions.

Events are actions that occur as a result of the user interactions.

Here is the list of some jQuery Events with description.

Table 7.2: jQuery events

Event	Description
click()	Triggers when the user presses the mouse button.
dblclick()	Triggers when the user presses the mouse button twice.
event.pageX	Returns the mouse position relative to the left edge of the document
focus()	Triggers when the element is selected.
hover()	Triggers when the user moves the mouse over an element.
keydown()	Attaches/Triggers the keydown event
keypress()	Attaches/Triggers the keypress event
mouseup()	Attaches/Triggers the mouseup event
on()	Attaches event handlers to elements
submit()	Triggers when the user presses the submit button.

click event

Click event occurs when a user clicks on an element in a web page. The code 7.5 demonstrates the usage of click event.

</> Code 7.5

```
<!DOCTYPE html>
<html>
<head>
<title>Click Event Example</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
  $(document).ready(function(){
    $("button").click(function(){
    $("#test").html("Web Engineering <br> Operating System <br>
    Database Management System <br>");
    // Add data on paragraph which has id “test”
    $("#test").show();
    // It shows data that are contained in paragraph which has id
       “test”
        });
     });
</script>
</head>
 <body>
   <h2>Click Event Example</h2>
   <p>Courses offered in BS(CS)</p>
```

```
    <p>Click on 'Show' button to show different courses offered in
      BS(CS). </p>
    <p id = "test"> </p>
    <button>Show</button>
</body>
</html>
```

The output of code 7.5 is shown in figure 7.7 and 7.8.

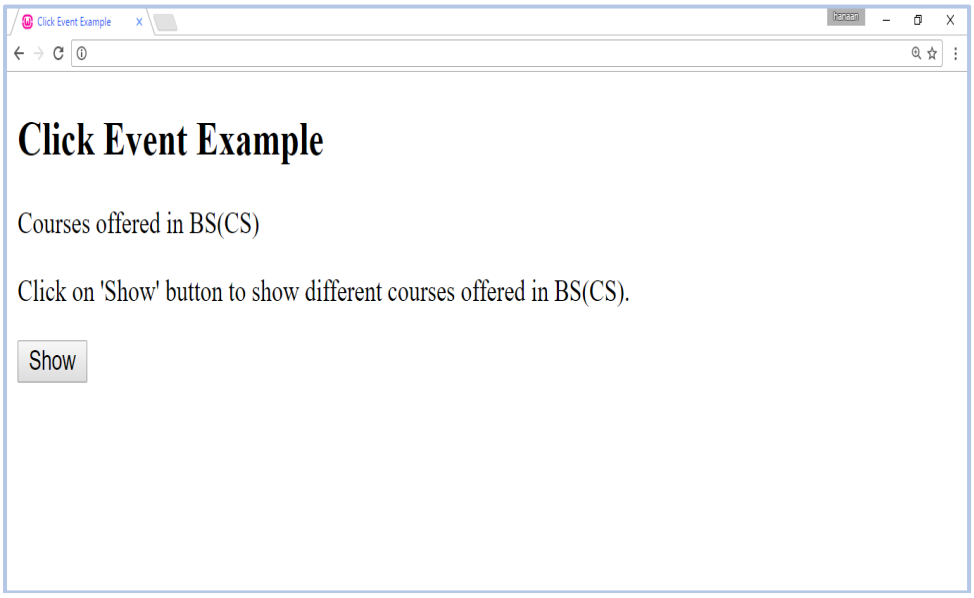

Figure 7.7: Output of code 7.5

When we click on 'Show' button, it will show different courses that are written in `<p id = "show">` and the output will be shown in figure 7.8.

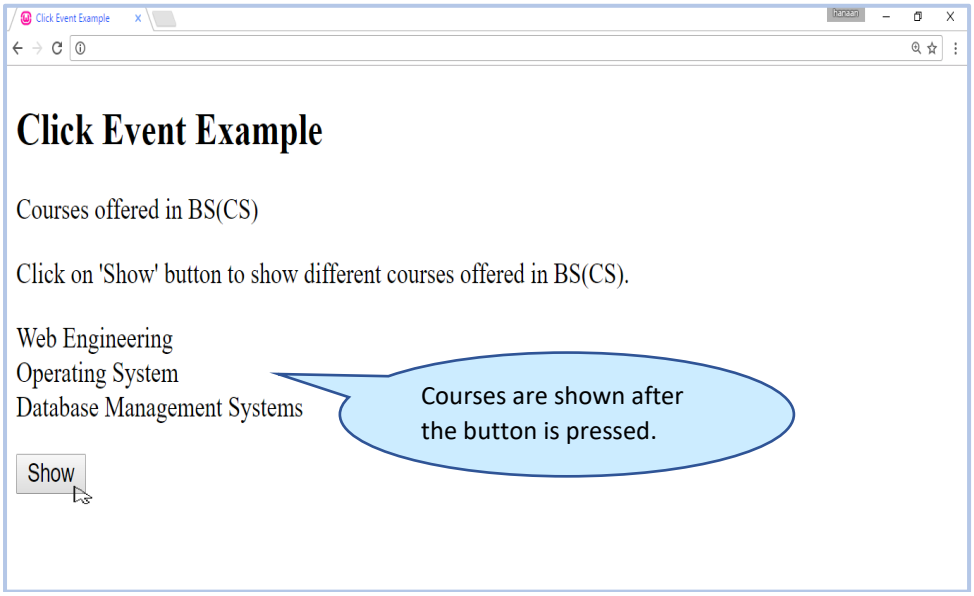

Figure 7.8: Output of code 7.5

Video Lecture

https://youtu.be/-T3uqUIOWPM

hover event

Hover event occurs when the mouse pointer moves over any element. Hover method contains two functions. First function contains the code which is to be executed while the mouse is hovering whereas the second function contains the code to be executed when the mouse leaves the hovering area. The code 7.6 demonstrates the use of hover event.

</> Code 7.6

```
<!DOCTYPE html>
<html>
<head>
<title>Selectors</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
   $(document).ready(function(){
   $("p").hover(function(){
   $(this).css("background-color", "#c8aefd");
   <! - - Background color of paragraph changes to “purple” when
          we move our mouse cursor over the paragraph -->
  },
  function(){
   $(this).css("background-color", "yellow");
   <! - - Initially the background color of paragraph is “yellow”
      -->

  }); });
</script>
</head>
 <body>
   <p>We give background colour "Yellow" to this paragraph. When
     mouse hovers on this paragraph, background color of this
     paragraph changes. </p>
 </body>
</html>
```

The output of code 7.6 is shown in figure 7.9 and 7.10.

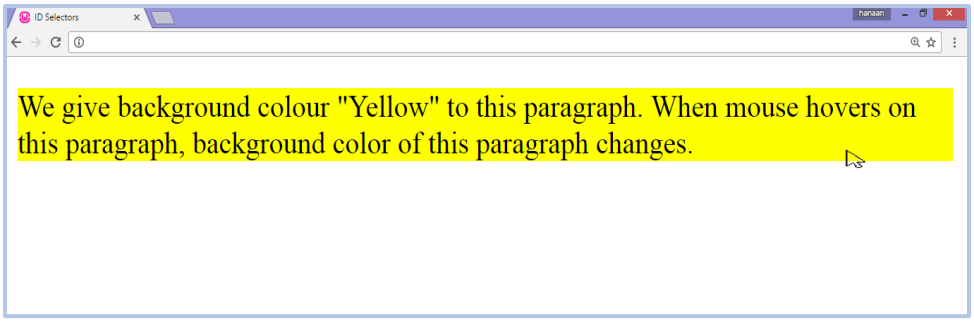

Figure 7.9: Output of code 7.6.

Initially the “color” of paragraph is “yellow” as shown in figure 7.9. When we move our mouse over this paragraph, background color of this paragraph turns to "Purple" as shown in figure 7.10. Whereas when the mouse pointer leaves the text area, it will become “yellow” again.

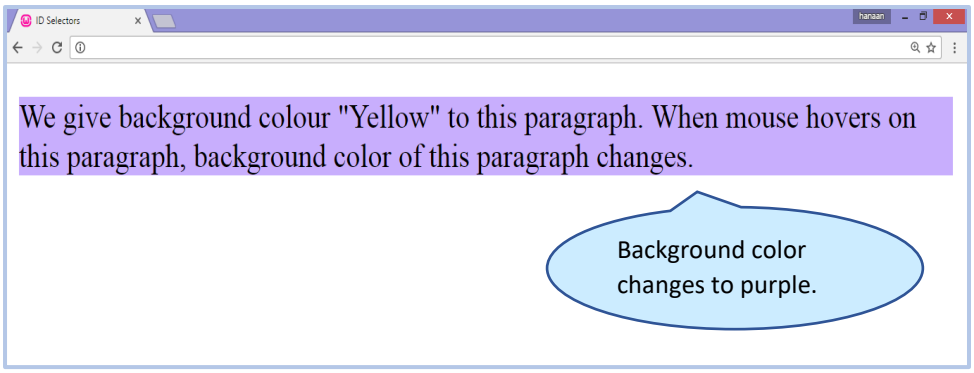

Figure 7.10: Output of code 7.6

mouseenter and mouseleave

The mouseenter method invokes when the mouse pointer enters the location of a specific HTML element. It attaches an event handler function with an HTML element. An event handler is a block of code which controls the response of an event. Similarly, mouseleave method invokes when the mouse pointer leaves the location of a specific HTML element. Code 7.7 demonstrates the use of mouseenter and mouseleave function.

</> Code 7.7

```
<!DOCTYPE html>
<html>
<head>
<title>Events</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
    $(document).ready(function(){
    $("#p2").mouseenter(function(){
     $("#p3").show();
      // It shows data when mouse enter on paragraph “p2”.
     $("#p3").html("<b>Note: </b> Four semesters in each
       program");
     $("#p3").css("color", "red");
    });
     $("#p2").mouseleave(function(){
     $("#p3").hide();
```

```
    // It hides data when mouse leaves paragraph “p2”.
   });
});
</script>
</head>
 <body>
   <h2>mouseenter and mouseleave() events in jQuery</h2>
   <p id = "p2">Programs offered in Computer Science Department
    </p> <span id = "p3"></span>
   <p>MS(Computer Science) </p>
   <p>MS(Software Engineering) </p>
   <p>MS(Information Security) </p>
 </body>
</html>
```

The output of code 7.7 is shown in figure 7.11.

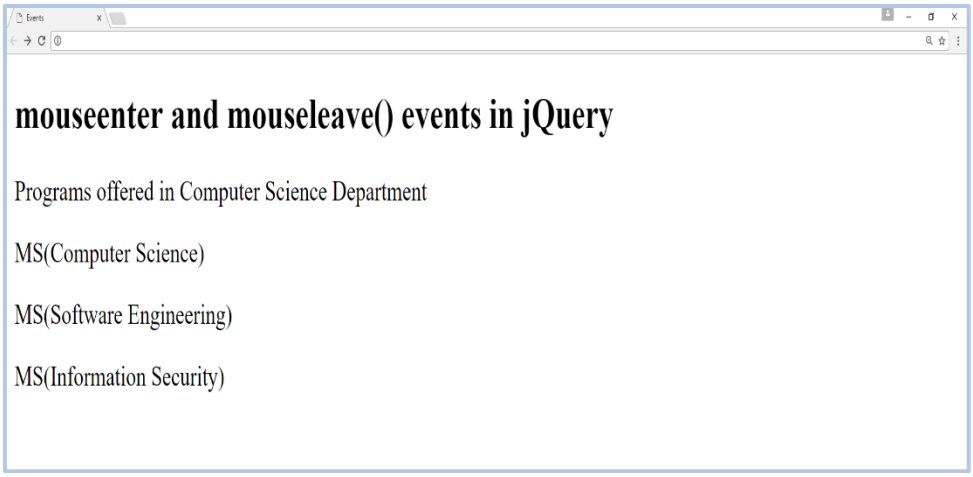

Figure 7.11: Output of code 7.7.

When we move mouse cursor over “Programs offered in Computer Science Department” paragraph, an alert message will be displayed as shown in figure 7.12.

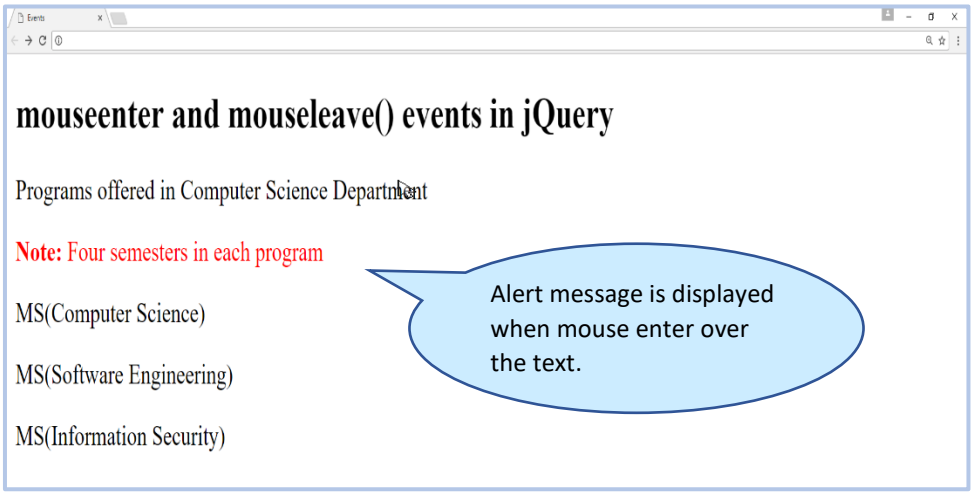

Figure 7.12: Output of code 7.7.

You notice that when we move our mouse cursor away from this paragraph, the alert message will get hidden as shown in figure 7.13.

Figure 7.13: Output of code 7.7.

focus and blur event

The focus event is invoked when any element receives attention. It attaches an event handler function to an element like HTML form field. The function is executed when the user brings cursor and clicks on the form field. The blur event is opposite to focus which is invoked when the user leaves the focus on the current element and clicks on any other element. The codes 7.8 gives the example of focus and blur events.

</> Code 7.8

```
<!DOCTYPE html>
<html>
<head>
<title>Events</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
    $(document).ready(function(){
     $("input").focus(function(){
        $(this).css("background-color", "#cccccc");
     // It changes the background color of input element when
       focused.
    });
    $("input").blur(function(){
        $(this).css("background-color", "#ffffff");
     // It changes the background color of input element to
        “white” when user moves to another input element
      });
    });
</script>
</head>
 <body>
   <table>
     <tr>
         <td>Name: </td><td><input type="text" name="fname"><br>
    </td>
     </tr>
```

```
    <tr>
      <td>ID: </td><td><input type="text" name="cnic"></td>
    </tr>
  </table>
  <p>When we click on any one of these field, then both
  background and border color of that specific field changes
  </p>
</body>
</html>
```

The output of code 7.8 is shown in figure 7.14 and 7.15.

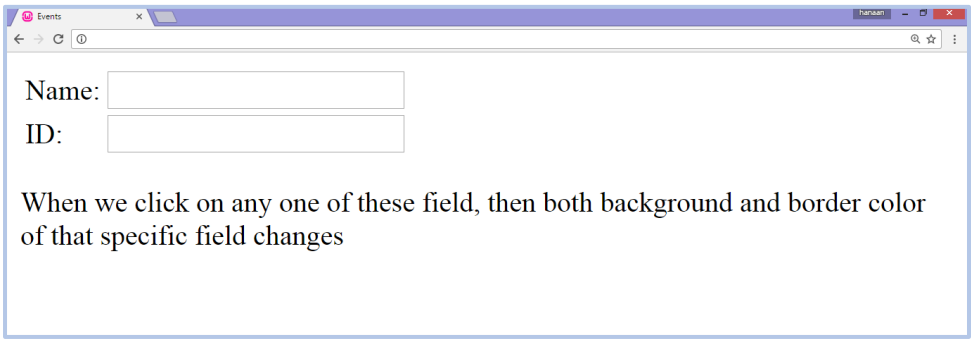

Figure 7.14: Output of code 7.8.

When we click on "Name" input field, the background color of this field is changed to “grey” as shown in figure 7.15 and when we leave this element and click on next input field, the background color of the previous “Name” field returns to "white".

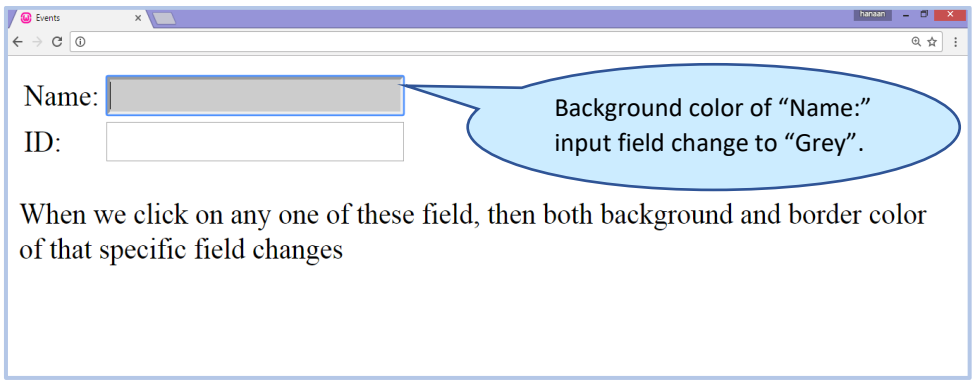

Figure 7.15: Output of code 7.8.

on() method

The on() method attaches one or more event handlers for the selected elements. Attach a click event to a <p> element by using “on” method is demonstrated in the code 7.9.

</> Code 7.9

```
<!DOCTYPE html>
<html>
<head>
<title>Events</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
```

```
</script>
<script>
    $(document).ready(function(){
    $("p").on("click", function(){
    $(this).hide();
   // it hides data when we click on any paragraph.
     });
    });
</script>
</head>
 <body>
   <h2>On Method Example</h2>
    Click on your favorite brand of car. When you click on any
    one from these options, it will hide from the screen !
   <p>1. Honda</p>
   <p>2. Toyota</p>
   <p>3. Suzuki</p>
   <p>4. Mercedes</p>
 </body>
</html>
```

The output of code 7.9 is shown in figure 7.16 and 7.17.

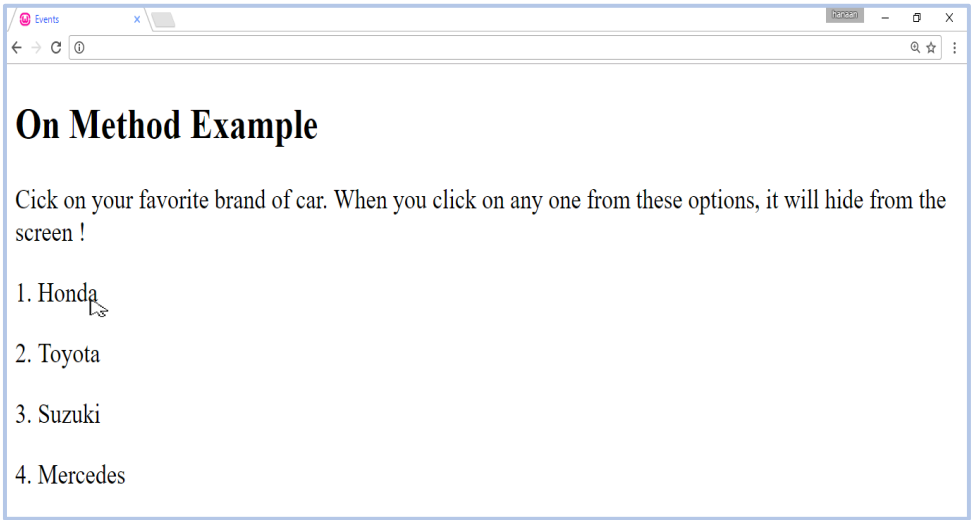

Figure 7.16: Output of code 7.9.

When a user clicks on first option, it will hide from the screen as shown in figure 7.17.

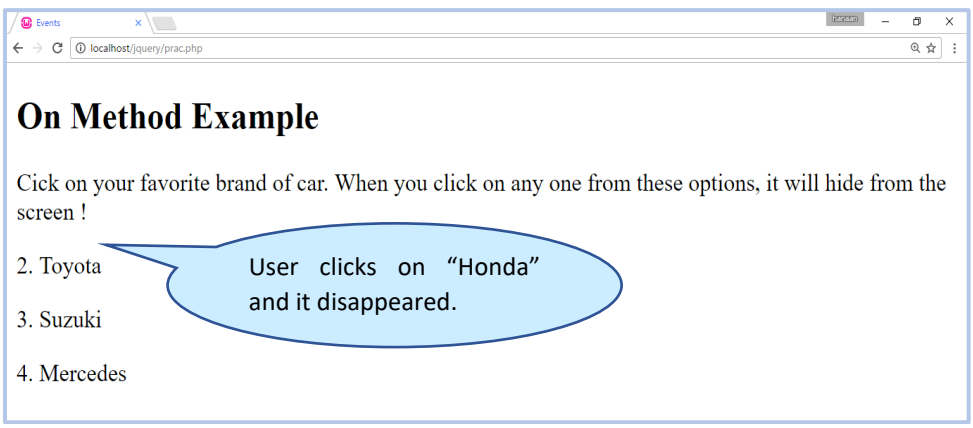

Figure 7.17: Output of code 7.9.

Activity 4

Create a paragraph and practice jQuery Events given in table 7.2.

7.4. jQuery Effects

jQuery library provides different techniques to add animations to make web page appear both more attractive and interactive. There are wide ranges of effects that can be applied to enable text and other elements to move in different fashion. Table 7.3 shows a list of jQuery effects that can be used to create animations.

jQuery Effects can be used to make web pages appear both more attractive and interactive.

Table 7.3: jQuery effects

Property	Description
hide() and show()	These effects are used to hide and show elements.
fadeIn()	It is used to show a hidden element slowly.
fadeOut()	It is used to hide a visible element slowly.
slideUp()	It is used to move the element upward.
slideDown()	It is used to move the element downward.
slideToggle()	It toggles between "slideUp " and "slideDown" element.

7.4.1. jQuery hide() and show()

With jQuery, we can hide and show HTML elements with the hide() and show() methods as shown in code 7.10.

</> Code 7.10

```
<!DOCTYPE html>
<html>
<head>
<title>Effects</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
    $(document).ready(function(){
    $("#h1").click(function(){
      $("p").hide();
```

```
        // This function will hide all the text that are written in
              paragraph
    });
    $("#h2").click(function(){
        $("p").show();
      // This function will show text that are written in
            paragraph.
    });
  });
</script>
</head>
<body>
   <p>When you click on "Hide" button, then the paragraph will
    disappear from the screen. Similarly, when you click on
    "Show" button, then the paragraph will be shown again. </p>
   <button id="h1">Hide</button>
   <button id="h2">Show</button>
</body>
</html>
```

The output of code 7.10 is shown in figure 7.18 and 7.19.

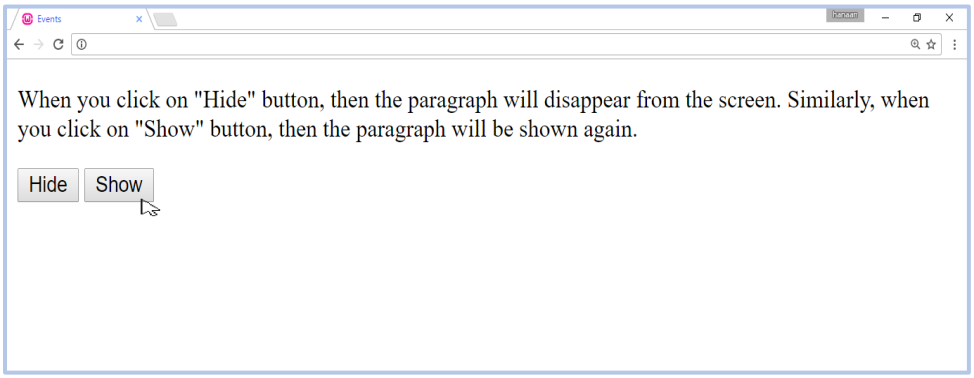

Figure 7.18: Output of code 7.10.

In the above output, we can observe that a paragraph is displayed in the output. When we click on "Hide" button, the paragraph will disappear from the screen as shown in figure 7.19.

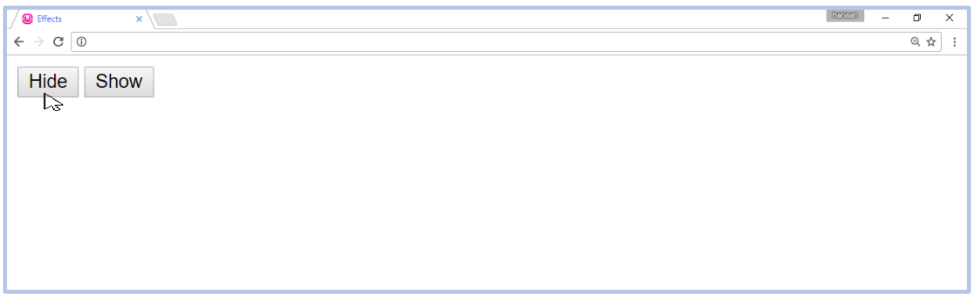

Figure 7.19: Output of code 7.10.

The speed of fadeIn() and fadeOut() can be controlled with an optional speed parameters with exact time in milliseconds.

7.4.2. jQuery fadeIn and fadeOut methods

fadeIn() method

The fadeIn() method is used to change the opacity of the selected elements, from hidden to visible.
The code 7.11 demonstrates the fadeIn() method with different parameters:

</> Code 7.11

```
<!DOCTYPE html>
<html>
<head>
<title>Effects</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
    $(document).ready(function(){
    $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    // This function will display second div slowly.
    $("#div3").fadeIn(5000);
    // This function will display third div with delay time of
            5000 seconds.
            });
  });
</script>
</head>
 <body>
    <button>Click on this button to fadeIn() with three different
    parameters</button><br>
    <div id="div1" style= "width:100px; height:80px;
    display:none; background-color: yellow;"></div><br>
    <div id="div2" style= "width:100px; height:80px;
    display:none; background-color: blue;"></div><br>
    <div id="div3" style="width:100px;height:80px;display:none;
    background-color: green; "></div>
 </body>
</html>
```

The output of code 7.11 is shown in figure 7.20 and 7.21.

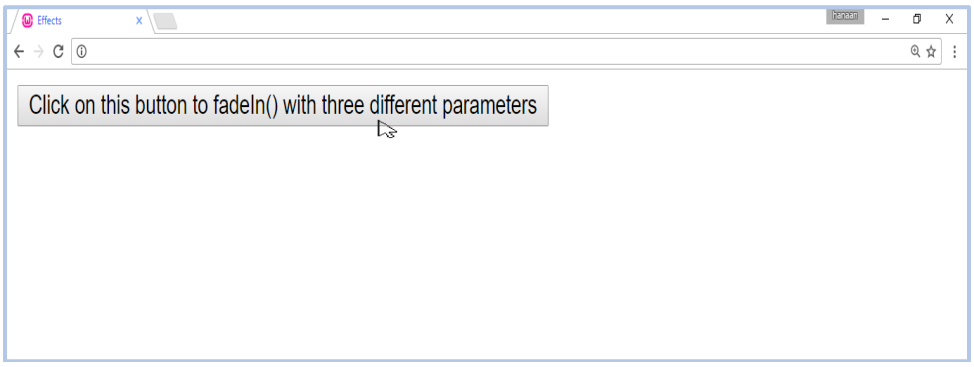

Figure 7.20: Output of code 7.11.

When we click on this "button", three different colored boxes will appear on the screen in a sequence. First the "yellow" color box will appear, after which the “blue” box will be shown and the “green” will be displayed in the last as shown in the figure 7.21. The optional speed parameter specifies the duration of the effect. It can take "slow", "fast", or milliseconds as values.

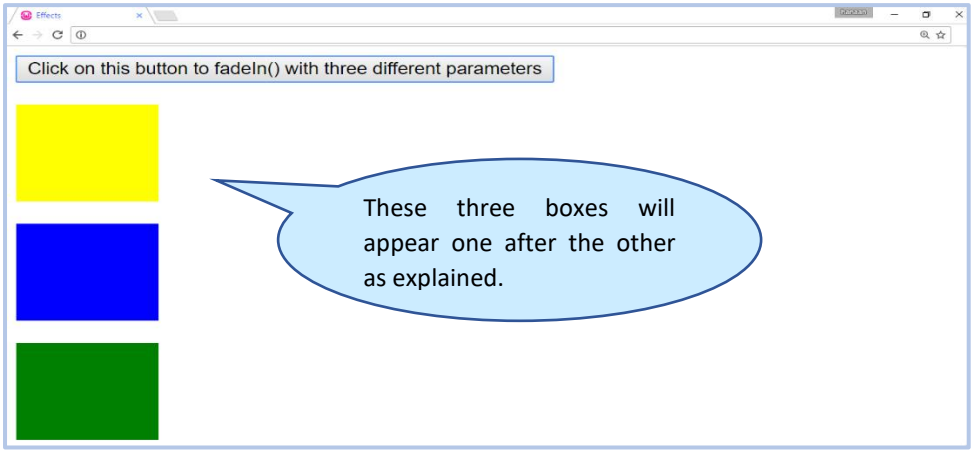

Figure 7.21: Output of code 7.11.

fadeOut() method

The fadeIn() method is used to change the opacity of selected elements, from visible to hidden.
The syntax of jQuery fadeIn() method is given below:

```
$(selector).fadeOut(speed,callback);
```

7.4.3. jQuery slideToggle method

The toggle method switches from display to hide or vice versa for the selected element. The functionality of “hide” and “show” will be invoked depending upon the current state of the object. If it is already shown, it will be hidden and vice versa.

The slideToggle() method toggles between the slideDown () and slideUp (). If the elements are slided down, slideToggle() will slide them up and vice versa. The code 7.12 shows the example of slideToggle() method in jQuery.

The slideToggle() method toggles between the slideDown() and slideUp().

</> Code 7.12

```
<!DOCTYPE html>
<html>
<head>
<title>Effects</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
 $(document).ready(function(){
    $("#div1").click(function(){
     $("#div2").slideToggle("slow");
```

```
    // it will toggle slideUp and slideDown with slow speed.
  });
 });
</script>
<style>
#div1, #div2 {
    padding: 5px;
    text-align: center;
    background-color: #00a1ff;
    border: solid 1px black;
}
#div2 {
    padding: 20px;
    display: none;
}
</style>
</head>
<body>
    <div id="div1">Click to slide up or down</div><br>
    <div id="div2">When user click on "button", panel will open.
    Similarly when user again click on this "button", panel will
    slide up! </div><br>
 </body>
</html>
```

The output of code 7.12 is shown in figure 7.22, 7.23 and in 7.24.

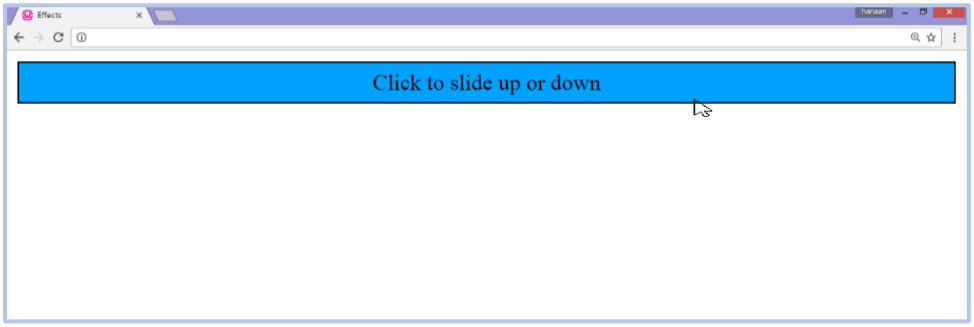

Figure 7.22: Output of code 7.12.

When the user clicks on "Click to slide up or down" button, panel will open as shown in figure 7.23. When you again click on this button, panel will slide up as shown in figure 7.24.

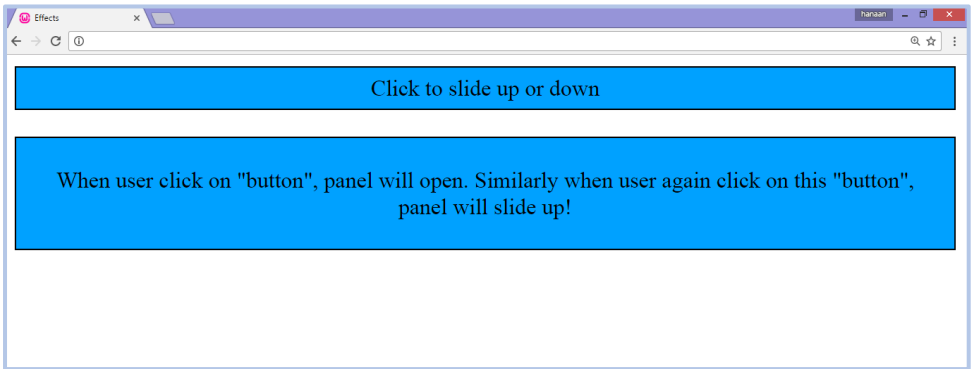

Figure 7.23: Output of code 7.12.

Figure 7.24: Output of code 7.12.

Activity 5

Create a paragraph and practice jQuery Effects given in table 7.3.

7.5. jQuery Get / Set

JQuery contains a lot of methods to manipulate and modify HTML elements. It is also capable of manipulating DOM. JQuery Get/Set has a major role to play in this manipulation.

7.5.1. jQuery get method

Get methods reads the text from an object in different forms like simple text, HTML, or equivalent values from the text field. These methods are shown below:

- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields

Code 7.13 shows how to get content with the jQuery text() and html() methods:

</> Code 7.13

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery Get Method</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
  $(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#p1").text());
     // It will show text that are written in paragraph which has
       id = “p1”.
     });
         $("#btn2").click(function(){
           alert("HTML: " + $("#p1").html());
     // It will show HTML in that are written in paragraph which
       has id = “p1”.
     });
  });
</script>
</head>
```

```
<body>
    <p id="p1">When we <b>click</b>on "show text" button, it will
    show you text written in this paragraph. When we click on
    "show HTML" button, it will show HTML of this paragraph. <p>
    <button id = "btn1"> show text </button>
    <button id = "btn2"> show HTML </button>
 </body>
</html>
```

The output of code 7.13 is shown in figure 7.25 and 7.26. when the user clicks on “show text” button, it will display the text in alert box whereas “show HTML” button will show the similar text with HTML tags in the alert box.

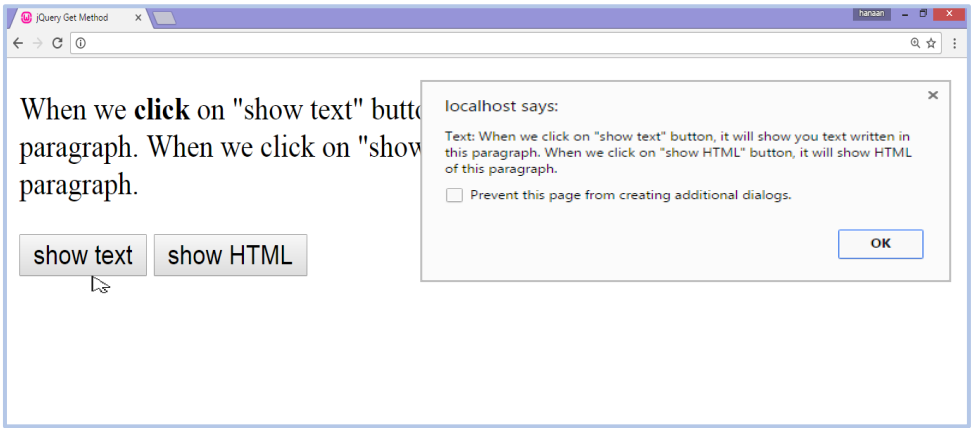

Figure 7.25: Output of code 7.13.

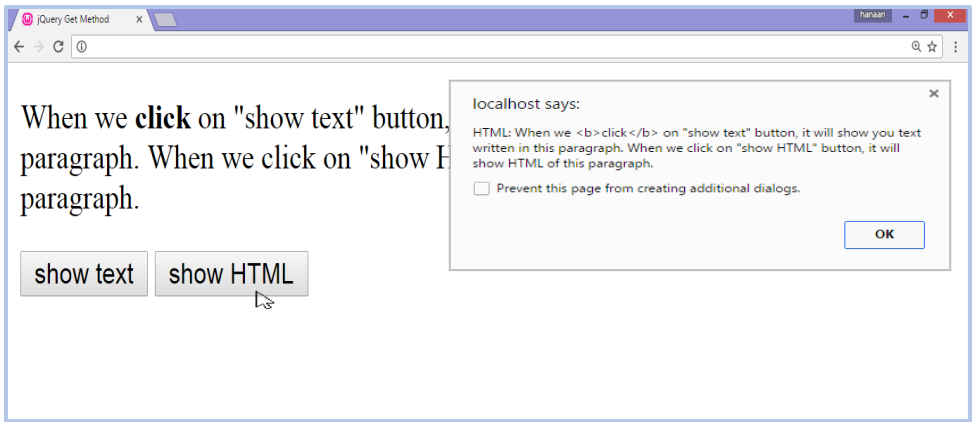

Figure 7.26: Output of code 7.13.

Code 7.14 shows example how to get text from an input field with value method and display it on alert box.

</> Code 7.14

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery Get Method</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
```

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert("Value: " + $("#p1").val());
    // It will show the text that is entered by user in input
      field.
   });
 });
</script>
</head>
<body>
    <p> Name: <input type="text" id="p1" value="Allama Iqbal Open
     University" style = "width:200px"></p>
     <button>Show Value</button>
 </body>
</html>
```

The output of code 7.14 is shown in figure 7.27. When we click on "Show Value" button, the value written in text box will be displayed in the popup box as shown in figure 7.28.

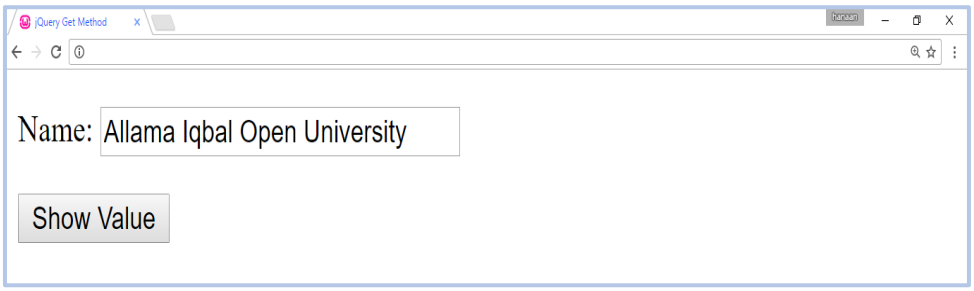

Figure 7.27: Output of code 7.14.

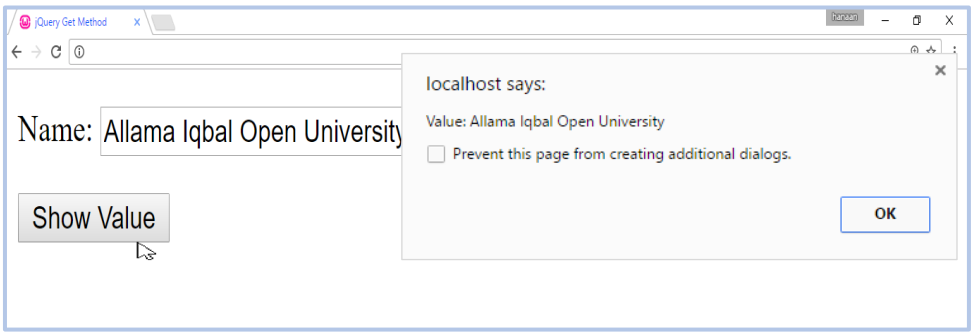

Figure 7.28: Output of code 7.14.

7.5.2. jQuery Set Method

The jQuery set method is used to write some specific text in an input field. We will use the same three methods from the previous to set content:

- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields

The code 7.15 demonstrates how to set content with the jQuery text(), html(), and val() methods:

</> Code 7.15

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery Set Method</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
$(document).ready(function(){
  $("#b1").click(function(){
    $("#p1").text("Hello I am new to jQuery!");
     // This function will replaced the text that are written in
        paragraph which has id “p1”.
     });
       $("#b2").click(function(){
         $("#p2").html("<b> Hello I am new to jQuery! </b>");
     // This function will replaced both text and HTML that are
        written in  paragraph which has id “p2”.
        });
          $("#b3").click(function(){
          $("#p3").val("AIOU");
           });
});
</script>
</head>
<body>

    <p id="p1">This is a paragraph. </p>
    <p id="p2">This is another paragraph. </p>
    <p>Input field: <input type="text" id="p3" value="University
     Name"></p>
     <button id="b1">Set Text</button>
     <button id="b2">Set HTML</button>
     <button id="b3">Set Value</button>
 </body>
</html>
```

The output of code 7.15 is shown in figure 7.29 and 7.30.

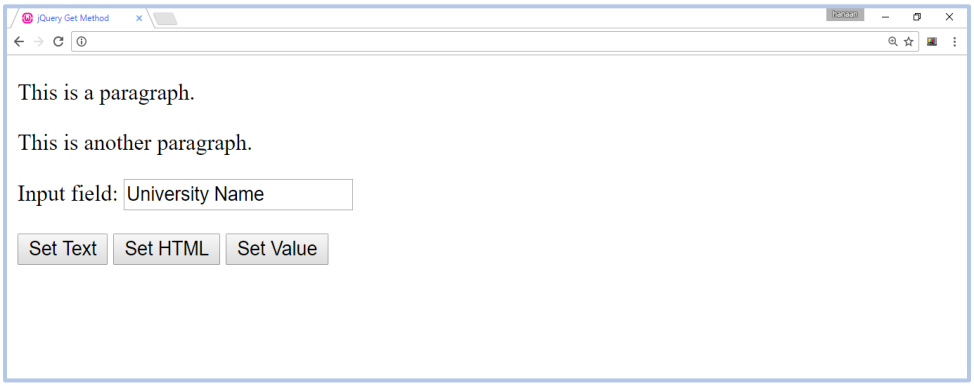

Figure 7.29: Output of code 7.15.

When we click on "Set Text" button, it will replace the text of first paragraph from “This is a paragraph” to “Hello I am new to jQuery!” with the help of jQuery text() method. Similarly, when the user clicks on "Set Html" button, it will replace the text of second paragraph from “This is another paragraph” to a bold form of “Hello I am new to jQuery!”. When the users click on “Set Value” button, the text of input field will change from “University Name” to “AIOU” with the help of val() method. The following output will show you how to set content with jQuery text(), html() and val() method.

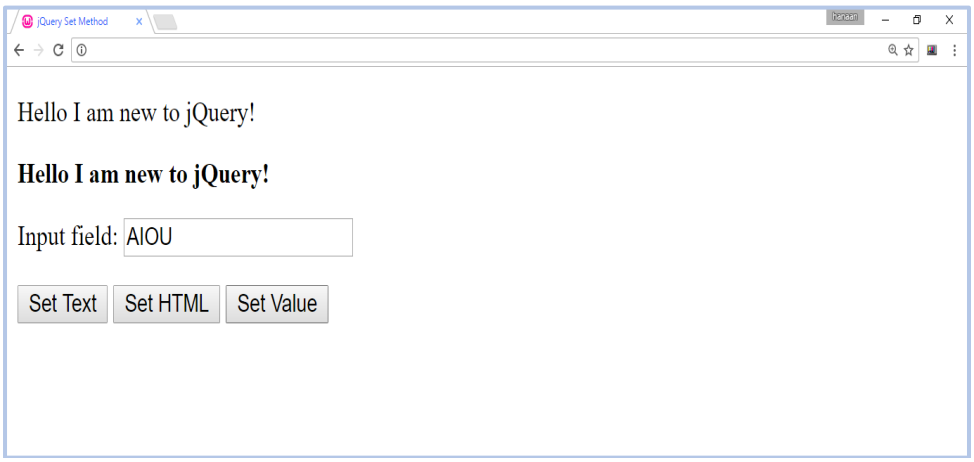

Figure 7.30: Output of code 7.15.

Activity 6

Demonstrate the use of GET and SET methods on the basic form input fields.

7.6. jQuery AJAX

jQuery AJAX stands for Asynchronous JavaScript and XML. It is a technique to create interactive web applications. Ajax uses client side scripting to control the display of dynamic content without the need of a complete page refresh. It periodically updates the contents of a web page based on the user action. AJAX completes a quick round trip to the server and fetch the complete data without displaying it on a web page until it is desired by the user. In this way, a user doesn’t have to wait for a whole new page to be loaded. For example, an online weather forecasting application has the weather information of all major cities. When the user types a city name or its zip code, the AJAX code can load the complete available weather data and display whatever is desired efficiently without again sending the request to the server for the complete data. Code 7.16 demonstrates the use of jQuery AJAX method.

jQuery Ajax can help us to load data without full page refresh.

</> Code 7.16

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery Ajax Method</title>
```

```
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js
">
</script>
<script>
   $(document).ready(function(){
     $(document).ready(function (){
     $("button").click(function (){
       $.ajax({url: "test.txt",
         // Get data from external file named “test.txt”.
        success:
        function(result){
         $("#p1").html(result);
       // Shows the loaded data on paragraph which has id “p1”.
     }});
   });
  });
});
</script>
</head>
<body>
    <h2>jQuery AJAX Example</h2>
     <p id="p1"></p>
     <button id="button">Get External Content</button>
 </body>
</html>
```

The output of code 7.16 is shown in figure 7.31. When the user clicks on “Get External Content” button, data from the file “test.txt” which is already fetched by using AJAX code is loaded on the web page without reloading the request as shown in figure 7.32.

Figure 7.31: Output of code 7.16.

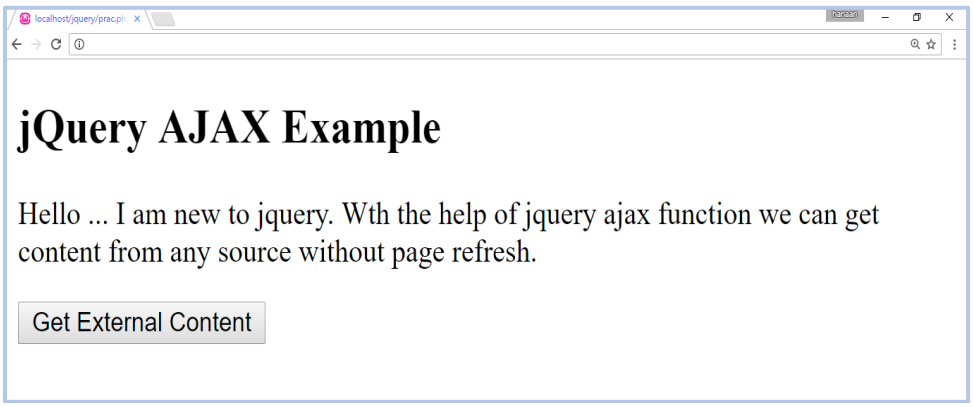

Figure 7.32: Output of code 7.16.

7.7. Form Validation

Finally, a previous example code of form validation is being represented here with the new application of jQuery. Note that the updated functionality is shown here in code 7.17.

</> Code 7.17

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Forms</title>
<script type="text/javascript"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.mi
n.js"></script>
<script>
$(document).ready(function(){ // take values from input fields.
  $("#submit").click(function(){
      var name = document.forms["myForm"]["name"].value;
      var address = document.forms["myForm"]["address"].value;
      var email = document.forms["myForm"]["email"].value;
      var zip = document.forms["myForm"]["zip"].value;
     // Check whether the input fields are empty or not.
        if (name == "")
        $("#name").html("Name must be filled out !");
        if (address == "")
        $("#address").html("Please enter your address !");
        if (email == "")
        $("#email").html("Please provide your email address!");
        if (zip == "")
        $("#zip").html("Please provide your state zip code !");

        if(document.myForm.city.value == "-1" )
         {
         $("#option").html("Please select your city !");
         }
    });
});
</script>
</head>
<body>
    <h2> jQuery Form Validation </h2>
    <p> Please fill the form below: </p>
    <form name="myForm" action="" method="post">
    <table border = "1">
      <tr>
         <td>Name: </td>
         <td><input type = "text" name = "name" ></td>
         <td style = "border:0"><span id = "name" style =
          "color:red"></span></td>
     </tr>
     <tr>
        <td>Address: </td>
        <td><input type = "text" name = "address" ></td>
        <td style = "border:0"><span id = "address" style =
          "color: red"> </span></td>
     </tr>
     <tr>
```

```
        <td>Email: </td>
        <td><input type = "email" name = "email" required></td>
        <td style = "border:0"><span id = "email" style = "color:
         red"> </span></td>
      </tr>
      <tr>
        <td>Zip Code: </td>
        <td><input type = "text" name = "zip"></td>
        <td style = "border:0"><span id = "zip" style = "color:
         red"> </span></td>
      </tr>
      <tr>
        <td>Gender: </td>
        <td><input type = "radio" name = "male" checked > Male
         </td>
      </tr>
      <tr>
        <td></td>
        <td><input type = "radio" name = "female">Female</td>
      </tr>
      <tr>
        <td>Subscription</td>
        <td><input type = "checkbox" name = "news" id = "news"
        value = "news">Newsletter</td>
        <td style = "border:0"><span id = "news" style =
        "color:red"> </span></td>
    </tr>
    <tr>
       <td></td>
       <td><input type = "checkbox" name = "mag" id = "mag" value
        = "mag"> Magzines</td>
       <td style = "border:0"><span id = "mag" style =
        "color:red"> </span></td>
    </tr>
    <tr>
        <td>City</td>
        <td>
            <select name = "city">
              <option value = "-1" selected>Select </option>
              <option value = "1" >Islamabad </option>
              <option value = "2" >Lahore </option>
              <option value = "3" >Karachi </option>
            </select>
        </td>
        <td style = "border:0"><span id = "option" style =
         "color: red"> </span></td>
    </tr>
    <tr>
       <td></td>
       <td><input type = "button" name = "submit" id = "submit"
        value = "submit"></td>
    </tr>
  </table>
  </form>
</body>
</html>
```

Output of code 7.17 is shown in figure 7.33.

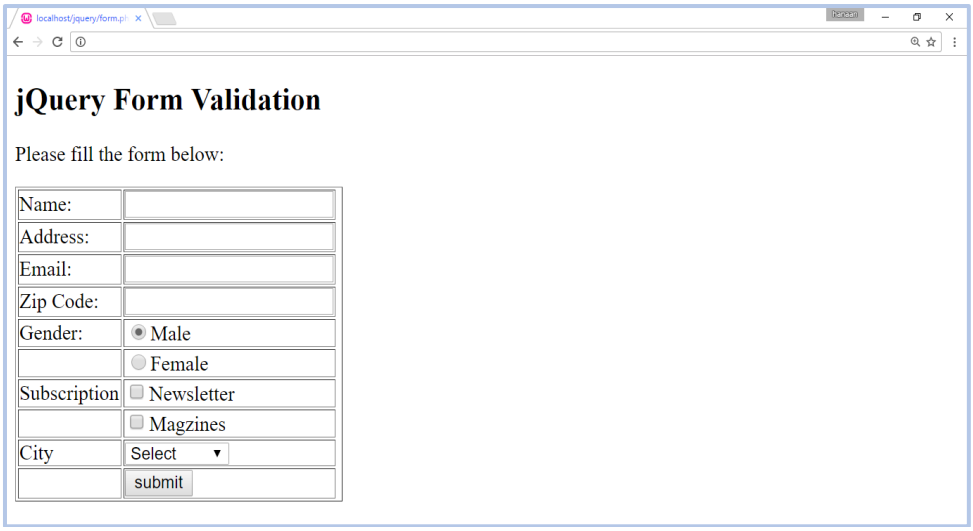

Figure 7.33: Output of code 7.17.

A form will be displayed for the user to enter the data in all given fields. Once the user enters the data, it is being validated for blank data input as shown in figures 7.34.

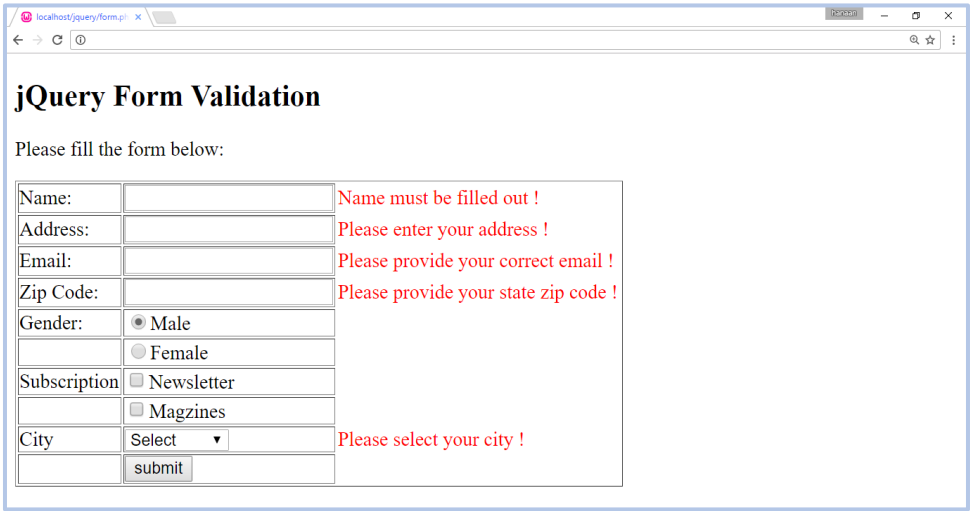

Figure 7.34: Errors in the user inputs

Activity 7

- Explore the code of form validation by using jQuery. Understand the syntax and semantic of jQuery statements.
- Modify the form validation code to verify the “Name:” field as non-numeric input, “Address:” field as alphanumeric, “Email:” field with standard input format including ”@” sign and “Zip Code:” as numeric data.

Video Lecture

https://youtu.be/IYTvLvLPtEQ

Unit Summary

In this unit, we learned about the basic concept of jQuery and its different methods. We used different events and effects to make our web pages look more attractive and interactive. We also learned how to validate a form using jQuery.

Self Assessment Questions

Choose the correct answer.

1. There are _____ methods to use jQuery library.
   A. one.
   B. two.
   C. three.
   D. four.

2. _____ method is used to get attribute of an element.
   A. getAttr()
   B. getAttributes()
   C. attr()
   D. None of the above.

3. Which of the following method is used to apply a style on an element?
   A. addClass(classes)
   B. addStyle(classes)
   C. addClassCSS(classes);
   D. None of the above.

4. If names are same, _____ variable take precedence over other.
   A. global variable
   B. local variable
   C. Both A and B.
   D. None of the above.

5. In jQuery $("div") selector selects
   A. first div element.
   B. All div elements.
   C. last div element.
   D. None of the above.

6. _____ is used to expand (if already collapsed) or collapse (if already expanded) page elements.
   A. hover()
   B. trigger()
   C. slide()
   D. toggle()

7. _____ method is used for the asynchronous HTTP request?
   A. ajax()
   B. hover()
   C. trigger()
   D. find()

8. Which of the following method is used to hide the selected elements?
   A. display(none)
   B. visible()
   C. hidden()

D. hide()

Answer Key:

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

Review Questions

Write short answers to the following questions

1. How can you install jQuery in your computer?
2. Differentiate between ID and Element selectors.
3. Briefly explain the concept of jQuery Events.
4. What is meant by jQuery Effect?
5. What is the difference between jQuery GET and SET methods?
6. What are the advantages of using jQuery Ajax?

</> Coding Exercise

1. Create a paragraph that comprises of ordered and unordered lists. Create a button that changes the color of list items to “Red” on pressing.
2. Create a paragraph that comprises of few sentences. Create a button, when pressed, slowly hides all the elements. Create another button that makes them reappear.
3. Attach two buttons with a paragraph. One button should turn the text into “bold” and the other should turn the color into “blue”. Count the number of milliseconds between the two clicked events.
4. Display a message and give style by changing “width”, “height”, “text-color” and “background-color” by using different click buttons.
5. Create a basic form input field that display a help sentence when the input field gets focused.
6. Create a text and apply fadeIn and fadeOut effects.
7. Create a paragraph and fade it to the given specified opacity.
8. Write jQuery code to toggle between fadeIn and fadeOut on different selected elements.
9. Modify the admission form created in coding exercise of unit 2 and apply validation techniques by using jQuery to different form elements.

References and Further Reading

1. Chaudhary, M., & Kumar, A. (2015). Ajax with jQuery. In *Practical jQuery* (pp. 195-210). Apress.
2. Overson, J., & Strimpel, J. (2015). *Developing Web Components: UI from JQuery to Polymer*. " O'Reilly Media, Inc.".
3. Duckett, J. (2015). JavaScript & jQuery. Wiley VCH
4. Otero, C., & Larsen, R. (2012). *Professional JQuery*. John Wiley & Sons.
5. Osmani, A. (2012). *Learning JavaScript Design Patterns: A JavaScript and jQuery Developer's Guide*. " O'Reilly Media, Inc.".
6. jQuery Basics. Available on: https://www.slideshare.net/MdSyefulIslam/jquery-besic
7. jQuery Date Time Pickers. Available on: http://quabr.com/37133263/add-number-of-days-to-date-datetimepicker
8. jQuery Date Time Pickers. Available on: http://quabr.com/37133263/add-number-of-days-to-date-datetimepicker
9. jQuery Selectors. Available on: https://www.w3schools.com/jquery/jquery_ref_selectors.asp
10. jQuery Event Handling. Available on: https://www.tutorialspoint.com/jquery/jquery-events.htm

UNIT 8

AngularJS

Introduction

This unit covers the elementary form of AngularJS. AngularJS was developed by Misko Hevery and Adam Abrons in 2009. It is a popular form of web application development framework which is now managed and upgraded by Google. AngularJS deals in the development of dynamic web applications. We can extend your HTML syntax to use the components of our application effectively. This framework provides a special form of binding which allows its use without writing the code from scratch.

AngularJS has the following major features:

- It can be used to create RICH Internet Application (RIA) in an effective form.
- It supports options to write the client side applications in the form of Model View Controller (MVC).
- It provides the cross-browser compliance by automatically handling the JavaScript code for all browsers.
- It is in an open source and free to use library.

Contrary to the strength of the AngularJS, it has some drawbacks as it is not secure due to the lack of server side authentication and if the scripts are blocked by the user, the functionality can not be utilized.

Unit Outcomes

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

1. Install AngularJS in your computer or use it through CDN.
2. Understand the syntax of AngularJS.
3. Use directive to enhance the functionality of your web page.
4. Add filters to format data.
5. Understand the concept of data binding.

Terminologies

Data Binding:	Synchronization of information between model and view.
Directive:	Extend HTML with new attributes.
Filter:	AngularJS filters are used to format the data.
Expression:	AngularJS expressions are written inside double braces.

8.1. AngularJS Development Environment

We can use the following two styles to use jQuery library to write and execute code in HTML document.

Local Installation

For local installation, we can download the AngularJS library from its official website **https://angularjs.org/**. When you click "download" button on the home page, the following window will be appeared as shown in figure 8.1.

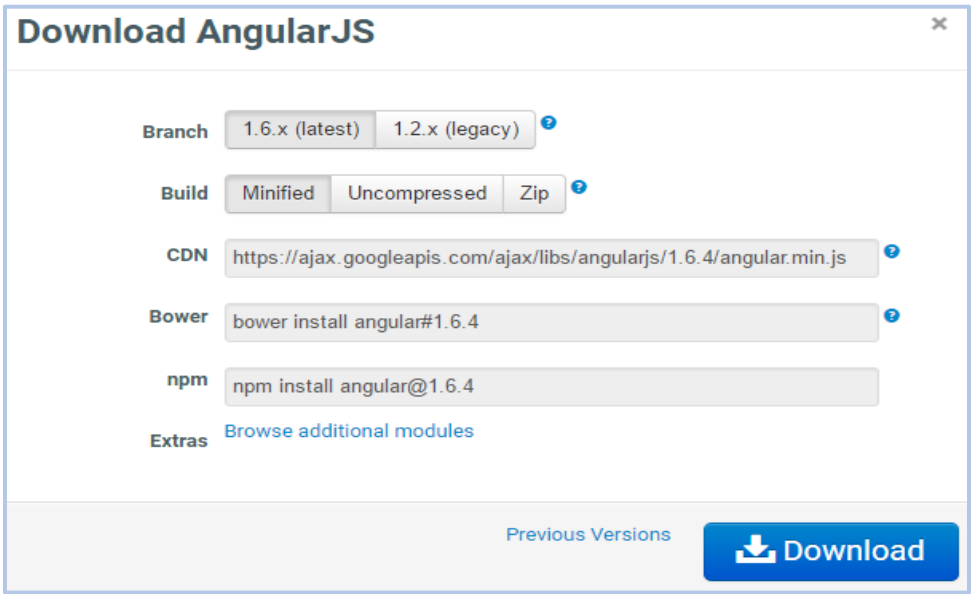

Figure 8.1: One way data binding

You can select the required version from the popup window and download it on your computer. After downloading, copy the AngularJS file "angular.min.js" in the directory of your project and then reference it.

CDN Based Version

If you are using CDN version of AngularJS, you need to provide the link of that AngularJS file instead of downloading it. You can simply include the AngularJS library in to your HTML document as shown in code 8.1.

</> Code 8.1

```
<!DOCTYPE html>
<html>
<head>
<title>Angular repeat directives</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js" ></script>
</head>
<body>
   <div ng-app="">
     <p>University Name: <input type="text" ng-model="uniname"
       style = "width:200px"> </p>
     <p ng-bind="uniname"></p>
```

```
      // it will bind with text field and shows entered text here.
   </div>
</body>
</html>
```

Output of code 8.1 is shown in figure 8.2.

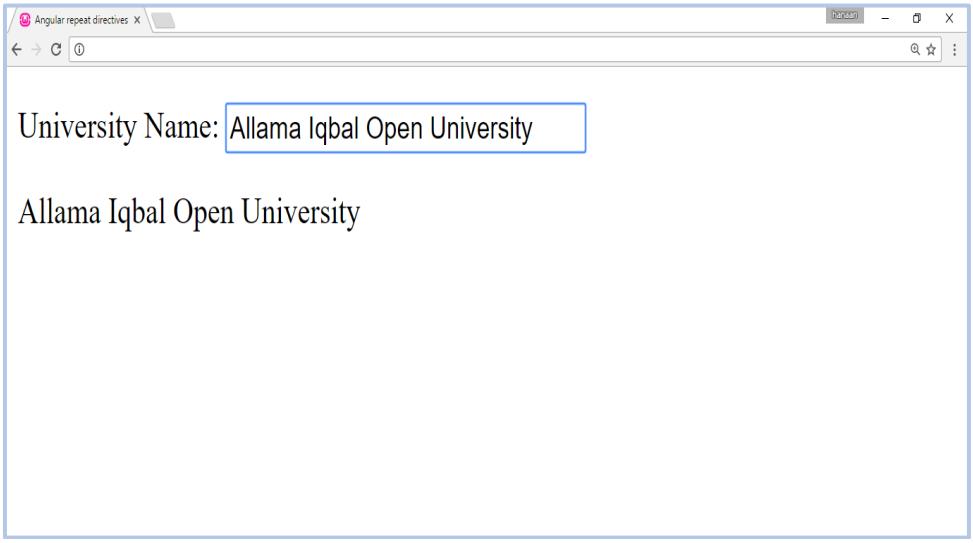

Figure 8.2: Output of code 8.1.

8.2. Expressions in AngularJS

An expression is a combination of one or more constants, variables, operators and functions that represent a value. In AngularJS, an expression is surrounded with double braces "{{expression}}". They are used to bind application data to HTML. Like JavaScript, the AngularJS expression can contain operators and variables.

</> Code 8.2

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Expressions</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js" ></script>
</head>
<body>
   <div ng-app="">
     6+3 = {{6+3}} <br>
     6-3 = {{6-3}} <br>
     6*3 = {{6*3}} <br>
     6/3 = {{6/3}} <br>
   </div>
</body>
</html>
```

Output of code 8.2 is shown in figure 8.3.

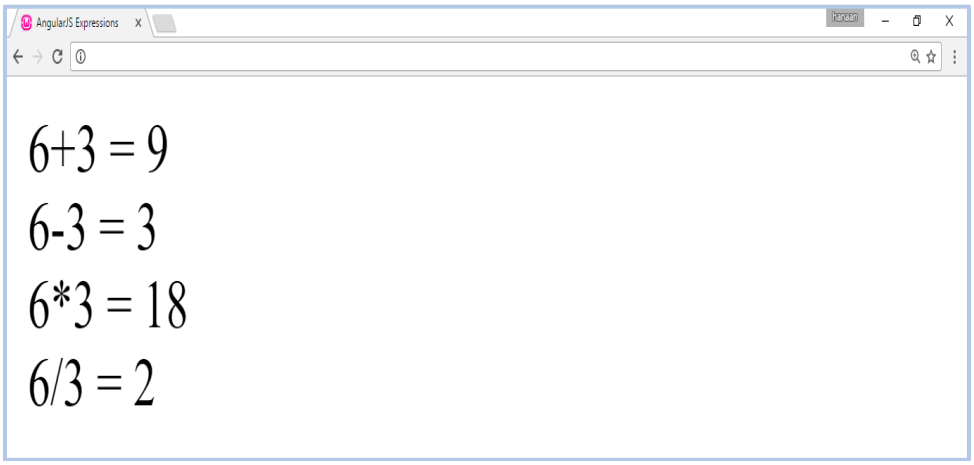

Figure 8.3: Output of code 8.2.

AngularJS expressions are like JavaScript applications. However, they can not contain conditional statements and loops.

Activity 1

Create a simple AngularJS expression and display "Hello World" in your output.

8.3. AngularJS Directives

AngularJS extends HTML functionality by providing new attributes called directives. These directives are special functions that attach specific behavior with the DOM elements. These special attributes start with "ng-" prefix where "ng-" stand for angular. AngularJS has a library of built in directives which are used to extend HTML functionality. For example, the static HTML code does not have the feature to update the date in a web page continuously. To enable an automatic updating of date in HTML, AngularJS can provide a directive to display an advance form of date picker widget (application code). These directives start with the prefix "ng-". We will study the following directives in this unit.

- **ng-app** – This directive starts an AngularJS Application.
- **ng-init** – This directive initializes application data.
- **ng-model** – This directive defines the model i.e. variable to be used in AngularJS.
- **ng-repeat** – This directive repeats html elements for each item in a collection.

ng-app:

"ng-app" directive starts an AngularJS application. It automatically initializes or bootstraps the application by defining the root element when web page containing AngularJS application is loaded. It is also used to load various AngularJS modules. The following example shows the syntax of "ng-app" attribute of a div element.

```
<div ng-app="">
  :
</div>
```

ng-init:

"ng-init" initializes an AngularJS application data. It is used to define a local variable which can be used in the application when required. The following example, initializes name of the person by using "ng-init".

```
<div ng-init="firstName='Ali'">
  :
</div>
```

ng-model:

"ng-model" binds an element with the property to be used in AngularJS application. The following example shows the binding of input type "text", using "ng-model" name.

```
<div ng-app="" ng-init="">
    Name: <input type="text" ng-model="name">
</div>
```

ng-repeat directive:

"ng-repeat" directive repeats html elements for each item in a collection. In following example, we have iterated over array of countries.

```
<div ng-repeat="">
    Name: <input type="text" ng-model="name">
</div>
```

The code 8.3 demonstrates the use "ng-init" and "ng-repeat" example.

</> Code 8.3

```
<!DOCTYPE html>
<html>
<head>
<title>Angular repeat directives</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js" ></script>
</head>
<body>
   <h2>Angular repeat directives example</h2>
   <div ng-app="" ng-init="names=['Yasir','Ahmad', 'Ali']">
    <ul>
      <li ng-repeat="x in names">
       {{ x }}
    // Get all elements from the array with the help of "ng-repeat"
       directive.
      </li>
    </ul>
</div>
</body>
</html>
```

Output of code 8.3 is shown in figure 8.4. In this code “ng-init” initializes names and “ng-repeat” is repeating HTML code to display each item of the array.

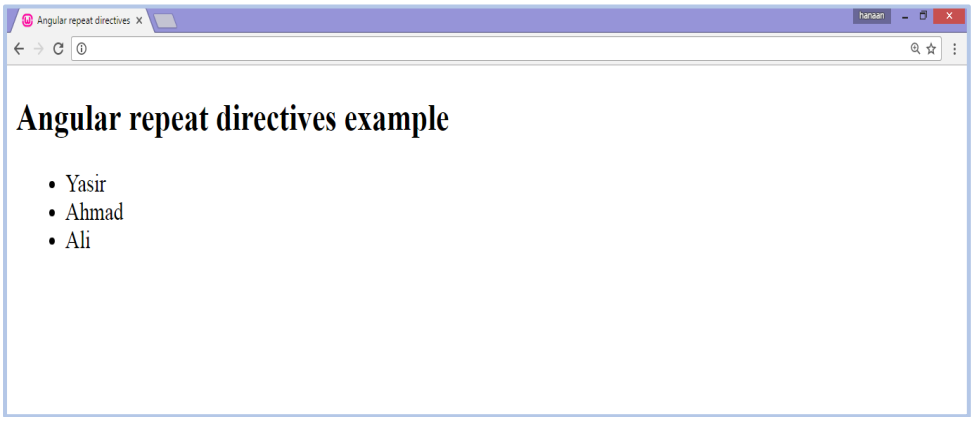

Figure 8.4: Output of code 8.3.

Video Lecture

https://youtu.be/0sFSUY-5DYc

8.4. Data Binding

Data Binding is a process through which we can establish a link between the data and the consumer frontend in a way that it gets synchronized automatically. AngularJS provides this feature in an effective way. This kind of binding links the data with the frontend at all times. Once the data changes, the display gets updated at the same time and vice versa. Thus, this update is being processed on all change events of the data.

Prior to AngularJS, the data binding was expedited in one direction only. The model and the template (data source) were merged to form a view (consumer frontend) only once. So, the view was generated after taking a snapshot of the data at any one time. The change in the data was not constantly being tracked for the view to get update as shown in figure 8.5.

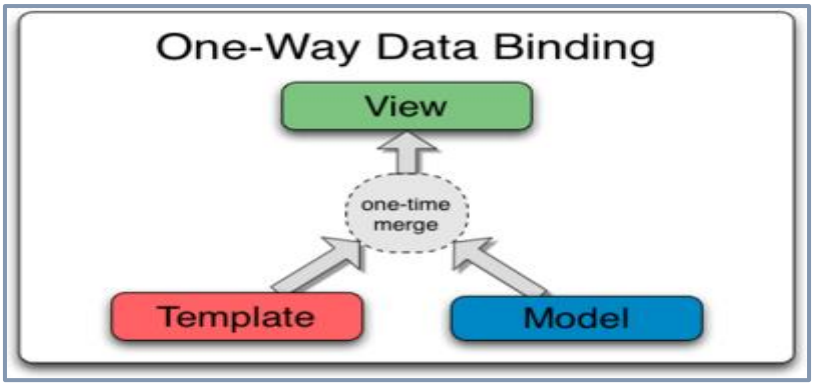

Figure 8.5: One way data binding ("Data Binding", n.d.)

Two way data binding maintains and continues update relationship between the model and the view by using AngularJS. This is achieved by placing the view at a central position on the theme as shown in figure 8.6.

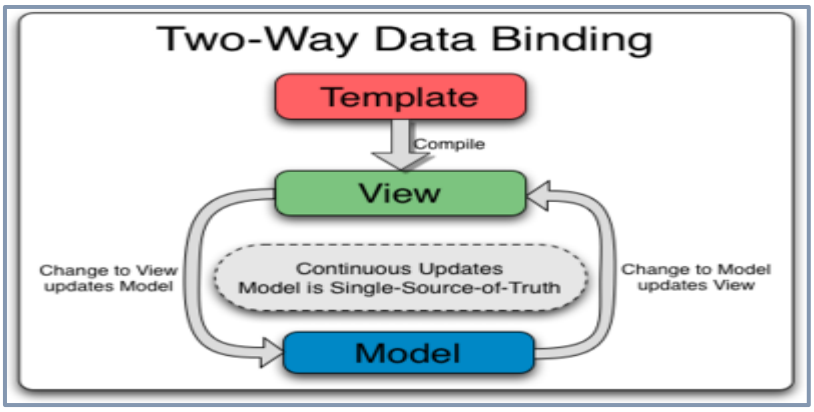

Figure 8.6: Two way data binding ("Data Binding", n.d.)

8.5. AngularJS Model Modes

8.5.1. One way binding

"ng-model" directive can be used in its basic form for one way data binding. Its application is shown in code 8.4. Code 8.4 describes the application of "ng-model" in form.

</> Code 8.4

```
<!DOCTYPE html>
<html>
<head>
<title>Angular Model directives</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>Angular Model example</h2>
   <div ng-app="App" ng-controller="name">
         Name: <input ng-model="name">
    </div>
```

```
    <p>With the help of ng-model directives, we can bind the
    value of input field. </p>
    <script>
     // bind value with input field.
        var app = angular.module('App', []);
        app.controller('name', function($scope) {
        $scope.name = "AIOU";
    });
    </script>
</body>
</html>
```

Output of code 8.4 is shown in figure 8.7. The phrase "AIOU" is displayed in the text field as it is binded from "scope.name".

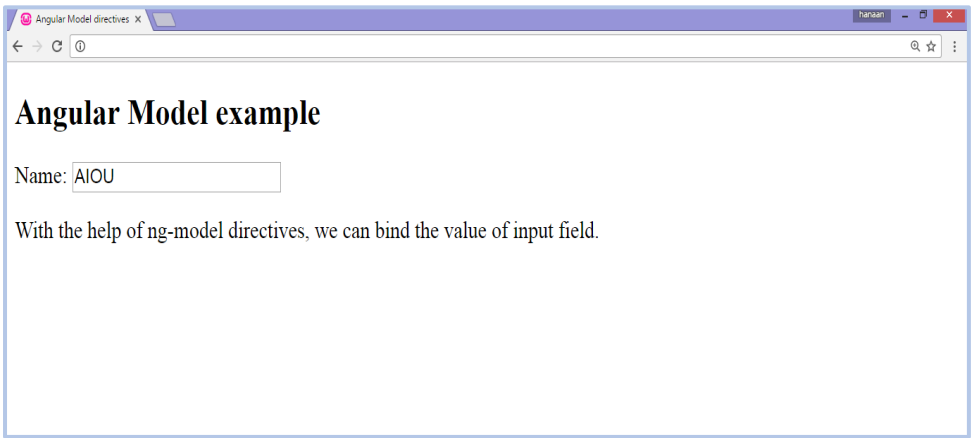

Figure 8.7: Output of code 8.4

8.5.2. Two Way Binding

"ng-model" directive can also be used for two way data binding. Its application is shown in code 8.5.

</> Code 8.5

```
<!DOCTYPE html>
<html>
<head>
<title>Angular Model</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>Two way data binding example</h2>
   <div ng-app="App" ng-controller="name">
    Name: <input ng-model="name">
    <h1>You entered: {{name}}</h1>
   </div>
   <p>The data that we are going to put in the textbox will be
   appeared with "You entered:" heading. </p>
<script>
        var app = angular.module('App', []);
        app.controller('name', function($scope) {
        $scope.name = "AIOU";
    });
```

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

The output of code 8.5 is shown in figure 8.8 and 8.9. Two way binding is achieved in this code as the phrase entered in the text field will appear instantaneously in the display. We are going to change the text "AIOU" with "Hello. I am new to angular!".

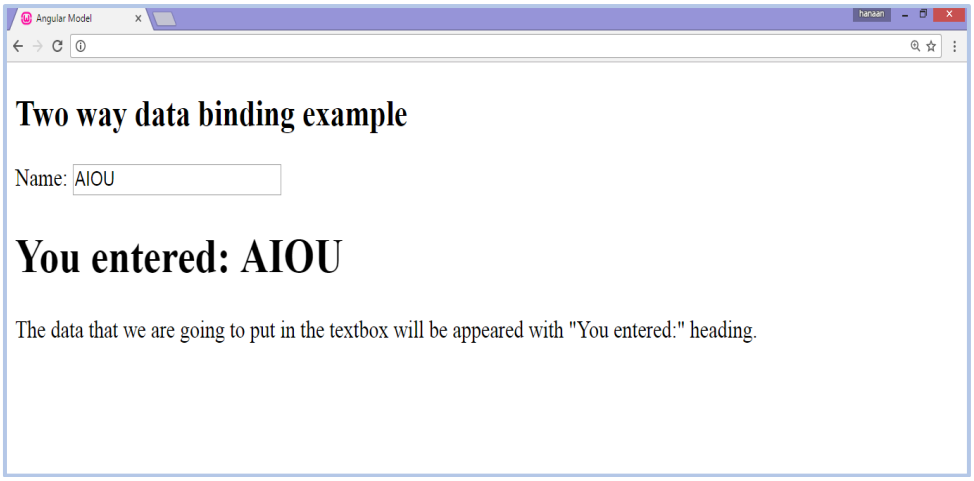

Figure 8.8: Output of code 8.5.

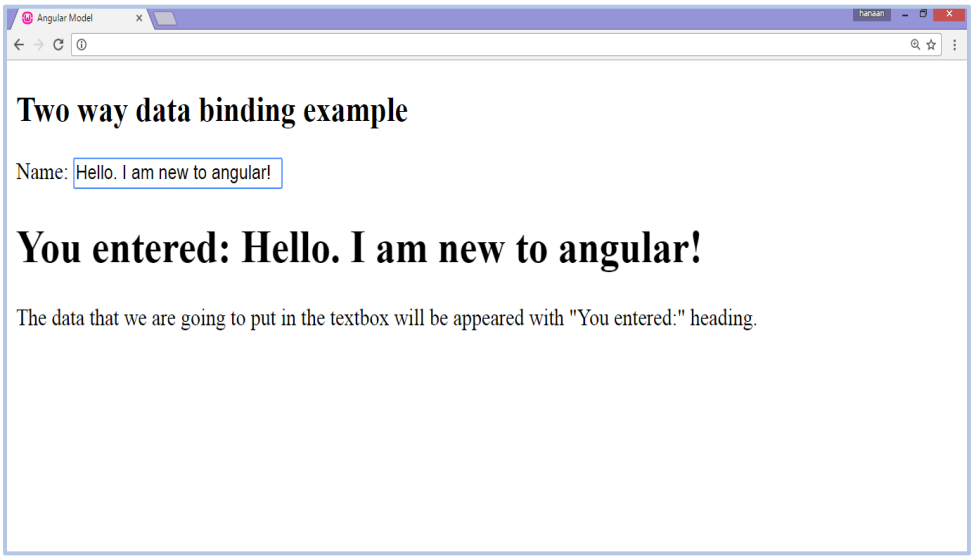

Figure 8.9: Output of code 8.5.

One way data binding binds the data from model to view whereas two way data binding binds the data from model to view and view to model.

Video Lecture

https://youtu.be/COCda6RPhaw

8.6. AngularJS Controller

The flow of data can be controlled by using AngularJS controller. It is a function that can change attributes and properties of data by using $scope object. This object maintains data and behavior by adding and updating the properties. These controllers can be linked by the model only whereas the view may be separated completely and independently. Code 8.6 describes the application of AngularJS controller.

</> Code 8.6

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Controller </title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>AngularJS Controller example</h2>
   <div ng-app="App" ng-controller="name">
       <h1 ng-click="changeName()">{{firstname}}</h1>
   </div>
<script>

   var app = angular.module('App', []);
   app.controller('name', function($scope) {
       $scope.firstname = "Hello";
   $scope.changeName = function() {
           $scope.firstname = "AIOU";
    // It will replace the text from “Hello” to “AIOU”.
       }
   });
</script>
   <p>When we click on "Hello" text, the text will be replaced
   with "AIOU". </p>
</body>
```

```
</html>
```

Output of code 8.6 is shown in figure 8.10.

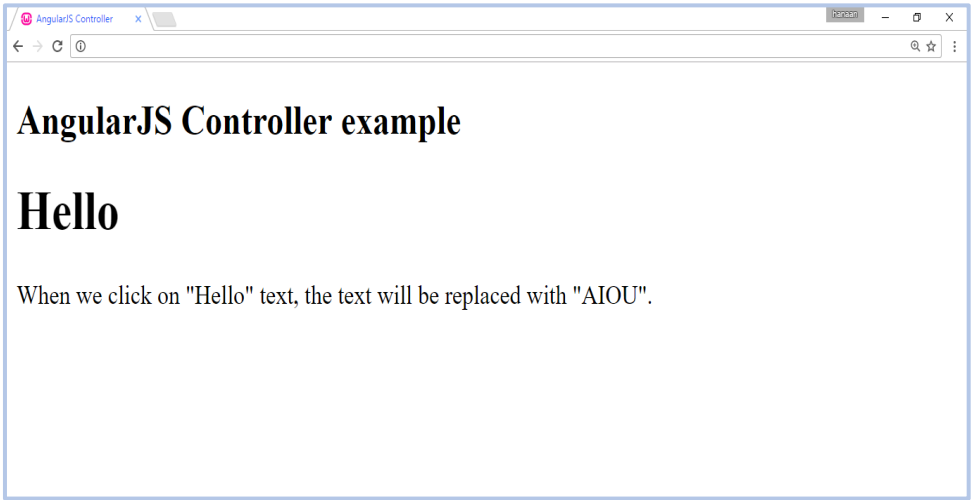

Figure 8.10: Output of code 8.6.

When we click on “Hello”, this text will be replaced by “AIOU” as shown in figure 8.11.

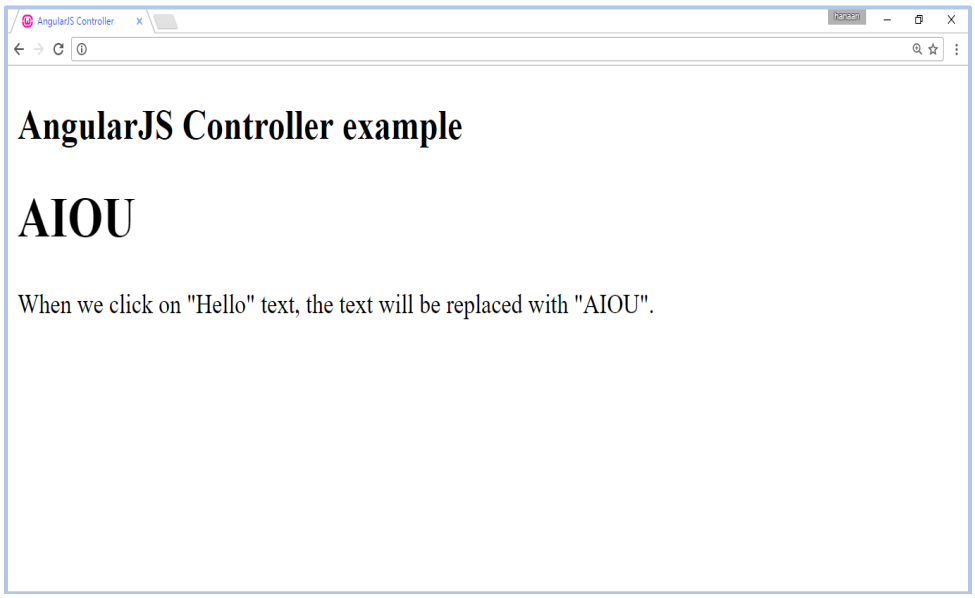

Figure 8.11: Output of code 8.6.

AngularJS controller controls the flow of data through its properties.

Activity 2

Write a small code to attach a property of “text-color” inside a controller and display the effect of this property in the output.

8.7. AngularJS Scope

A scope is an object which is shared by the controller and the view. This object joins the functionality of the controller with the views. As the controller defines the properties on the scope and the views= uses these properties. AngularJS keeps the controller and the view is synchronized by implementing two way data binding. Code 8.7 demonstrates the use of scope in AngularJS.

</> Code 8.7

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Scope</title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>AngularJS Scopes example</h2>
  <div ng-app="name" ng-controller="myCtrl">
    //University name will display here.
   <h2>University Name: {{uniname}}</h2>
   </div>
  <script>
   var app = angular.module('name', []);
   app.controller('myCtrl', function($scope) {
       $scope.uniname = "AIOU";
   });
  </script>
   <p>The property " uniname" was made in the controller and can
   be displayed in the output with the help of {{ }}
   brackets.</p>
 </body>
</html>
```

In the above example “scope. uniname” is being assigned a value “AIOU” and is being transferred to “{{uniname}}” before it is displayed in the heading. Output of code 8.7 is shown in figure 8.12.

Figure 8.12: Output of code 8.7.

8.8. AngularJS Filters

AngularJS Filters is an AngularJS code that reads a certain amount of data and transforms into an output pattern by adding or removing certain elements. It can change or update the data to display it= on user screen without changing its original format. Table 8.1 list down some important filters.

Table 8.1: Filter types

Attributes	Description
uppercase	Format a string to upper case.
lowercase	Format a string to lower case.
filter	Select a subset of items from an array.
orderBy	Order an array by an expression.
currency	Format a number to a currency format.
date	Format a date to a specified format.
limitTo	Limits an array/string, into a specified number of elements/characters.
number	Format a number to a string.

Filters can be used to change the format of the data.

Filters can be added to expressions by using the pipe character |, followed by a filter. Some of the filters are discussed below.

uppercase filter

The uppercase filter format strings to upper case. Code 8.8 demonstrates the use of uppercase filter.

</> Code 8.8

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Filters </title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>AngularJS Uppercase Filter example</h2>
   <div ng-app="App" ng-controller="name">
   <p>Last name: {{ lastname | uppercase }}</p>
   </div>
   <script>
   var app = angular.module('App', []);
   app.controller('name', function($scope) {
       $scope.firstname ="hello";
       $scope.lastname = "world";
    // it will convert the text “world” in to upper case.
   });
</script>
 <p>The first and last name are converted into uppercase </p>
</body>
```

```
</html>
```

Output of code 8.8 is shown in figure 8.13. The phrase “world” is being displayed after converting it into upper case.

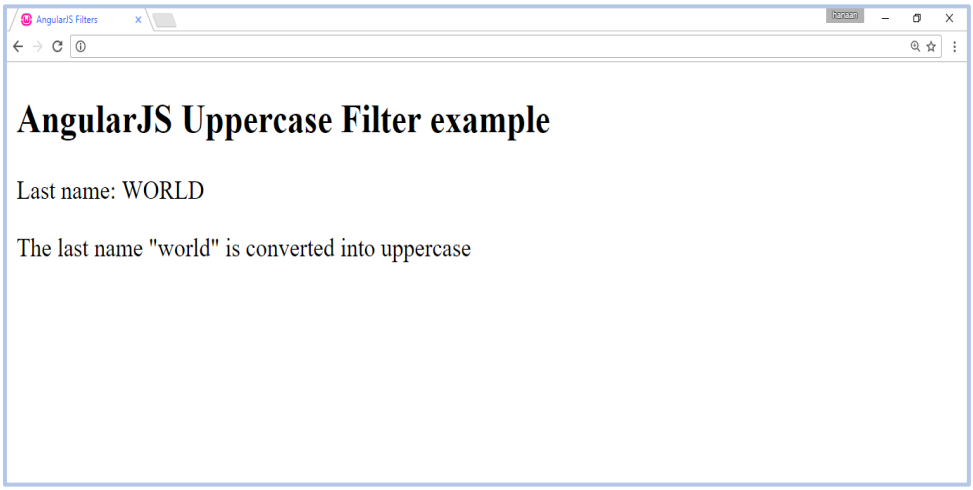

Figure 8.13: Output of code 8.8.

lowercase Filter

The lowercase filter format strings to lower case. Here we have added lowercase filter to print first name in all lowercase letters as shown in example.

```
Enter First name: <input type = "text" ng-model="firstname">
<p>Name in lower case: {{ firstname() | lowercase }}</p>
```

filter

To display only required subjects, we use subjectname as filter. This can be explained with the help of the following example.

```
Enter subject: <input type ="text" ng-model = "subjectname">
Subject:
<ul>
  <li ng-repeat ="subject in student.subjects | filter:
    subjectname" >  <li>
      {{ subject.name + ', marks:' + subject.marks }}
  </li>
</ul>
```

orderBy filter

To order subjects by marks, we use orderBy marks.

```
<ul>
  <li ng-repeat ="subject in student.subjects | orderBy: 'marks'" >
<li>
      {{ subject.name + ', marks:' + subject.marks }}
  </li>
</ul>
```

currency filter

The filter for the currency is demonstrated in code 8.9.

</> Code 8.9

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Filters </title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>AngularJS Currency Filter example</h2>
   <div ng-app="App" ng-controller="name">
   <h1>Price: {{ price | currency }}</h1>
   </div>

  <script>
   var app = angular.module('App', []);
   app.controller('name', function($scope) {
       $scope.price = 100;
     });
</script>
 <p>The currency filter format a number into proper currency.
   </p>
</body>
</html>
```

The output of code 8.9 is shown in figure 8.14. Please note that the figure of 100 is automatically formatted with the currency form.

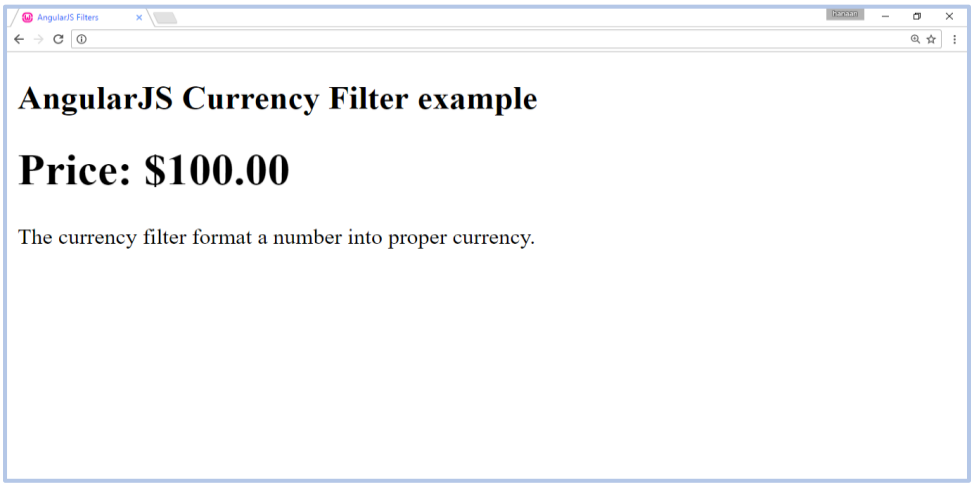

Figure 8.14: Output of code 8.9.

filter an array based on user input

The ng-model once linked with an input field, the current value given by the user in the field can be used as an expression filter to reorganize the array. The list will be displayed as per letter written in the input field as shown in the code 8.10.

</> Code 8.10

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Array </title>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js"></script>
</head>
<body>
   <h2>Array Based User Input example</h2>
   <div ng-app="App" ng-controller="name">
    <p><input type="text" ng-model="test"></p>
    <ul>
       <li ng-repeat="x in names | filter : test">
        {{ x }}
       </li>
    </ul>
  </div>
  <script>
   var app = angular.module('App', []);
   app.controller('name', function($scope) {
     $scope.names = [
       'Ali',
       'Hussnain',
       'Ahmad',
       'Zahid',
       'Danial',
       'David'
      ];
     });
</script>
 <p>Names can be filtered and displaying only matching name.</p>
</body>
</html>
```

Output of code 8.10 is shown in figure 8.15 and 8.16.

Figure 8.15: Output of code 8.10.

Now if we want to search a name "Danial" in the textbox, only the searched name will be shown in the list as shown in figure 8.16.

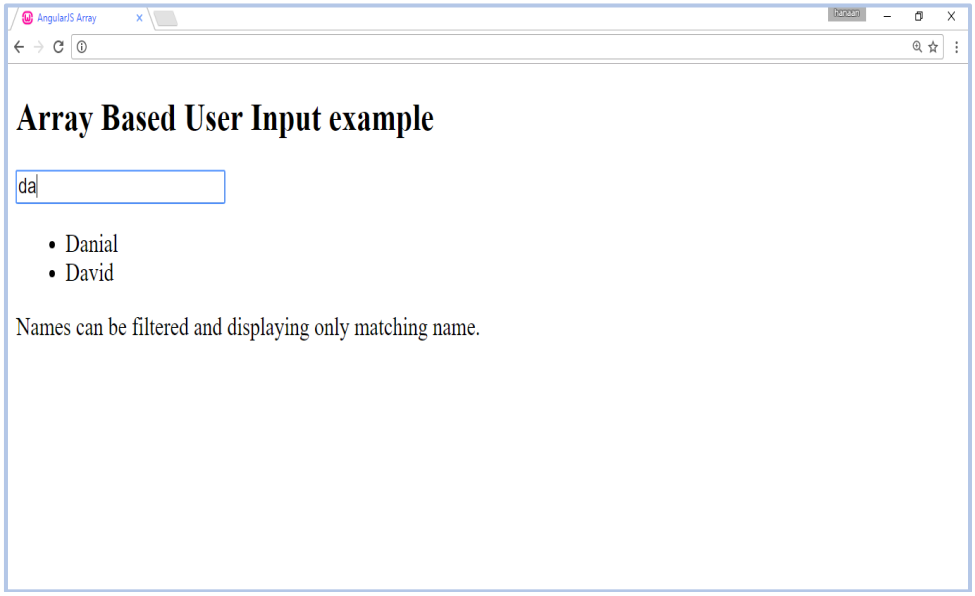

Figure 8.16: Output of code 8.10

Activity 3

Create a web page and practice AngularJS filters as given in table 8.1.

8.9. AngularJS Forms

Finally, a previous example code of form validation is being represented here with the new application of AngularJS. Note that the same functionality is shown here in code 8.11.

</> Code 8.11

```
<! DOCTYPE html>
<html>
<head>
<title>AngularJS Expressions</title>
<script type="text/javascript"></script>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.mi
n.js" ></script>
</head>
<body>
   <h2>AngularJS Form Validation</h2>
   <p> Please fill the form below: </p>
   <form ng-app="myApp" ng-controller="myForm" name="myForm"
    novalidate>
   <table border="1">
     <tr>
         <td>Name: </td>
         <td><input type="text" name="name" ng-model="user"
              required>
```

```
            <span style="color:red" ng-show="myForm.name.$dirty
             && myForm.name.$invalid">
            <span ng-show="myForm.name.$error.required">Username
             is required.</span>
            </span>
        </td>
    </tr>
    <tr>
        <td>Address: </td>
        <td><input type="text" name="address" ng-model="address"
             required>
          <span style="color:red" ng-show="myForm.address.$dirty
          && myForm.address.$invalid">

       // it will check address of user.

          <span ng-show="myForm.address.$error.required">Address
          is required.</span>
          </span>
        </td>
    </tr>
    <tr>
        <td>Email: </td>
        <td><input type="email" name="email" ng-model="email"
              required>

          // it will check email format.

            <span style="color:red" ng-show="myForm.email.$dirty
             && myForm.email.$invalid">
            <span ng-show="myForm.email.$error.required">Email
             is required.</span>
            <span ng-show="myForm.email.$error.email">Invalid
             email Address.</span>
            </span>
         </td>
    </tr>
    <tr>
        <td>Zip Code: </td>
        <td><input type="number" name="zip" ng-model="zip"
             required>
            <span style="color:red" ng-show="myForm.zip.$dirty
              && myForm.zip.$invalid">
            <span ng-show="myForm.zip.$error.required">Zip Code
             is required.</span>

          // it will check zip code whether it's a number or not.

            <span ng-show="myForm.zip.$error.number">Invalid Zip
             Code.</span>
            </span>
        </td>
    </tr>
    <tr>
        <td>Gender: </td>
        <td><input type = "radio" name = "male" checked> Male
        </td>
    </tr>
```

```
    <tr>
         <td></td>
         <td><input type = "radio" name = "female">Female</td>
    </tr>
    <tr>
         <td>Subscription</td>
         <td><input type = "checkbox" name = "news" id = "news"
              value = "news">Newsletter</td>
         <td style = "border:0"><span id = "news" style =
              "color:red"> </span></td>
    </tr>
    <tr>
         <td></td>
         <td><input type = "checkbox" name = "mag" id = "mag"
              value = "mag"> Magzines</td>
         <td style = "border:0"><span id = "mag" style =
             "color:red"> </span></td>
    </tr>
    <tr>
         <td>City</td>
         <td>
         <select name = "city" ng-model="city" required
           style="width: 170px;">
           <option value = "-1" selected>Select </option>
           <option value = "1" >Islamabad </option>
           <option value = "2" >Lahore </option>
           <option value = "3" >Karachi </option>
         </select>
        // it will check whether option is selected from drop down
           menu or not.
           <span id = "option" style = "color: red" ng-
           show="myForm.city.$error.required">Please select
           your city !</span>
         </td>
     </tr>
     <tr>
         <td></td>
        // it will disable the submit button until user name and
           password is not true.
         <td><input type="submit" ng-disabled="myForm.user.$dirty
              && myForm.user.$invalid ||  myForm.email.$dirty &&
              myForm.email.$invalid"></td>
     </tr>
   </form>
  </table>
<script>
   var app = angular.module('myApp', []);
   app.controller('myForm', function($scope) {
      $scope.name = '';
      $scope.email= '';
     });
</script>
</body>
</html>
```

Output of code 8.11 is shown in figure 8.17 and 8.18.

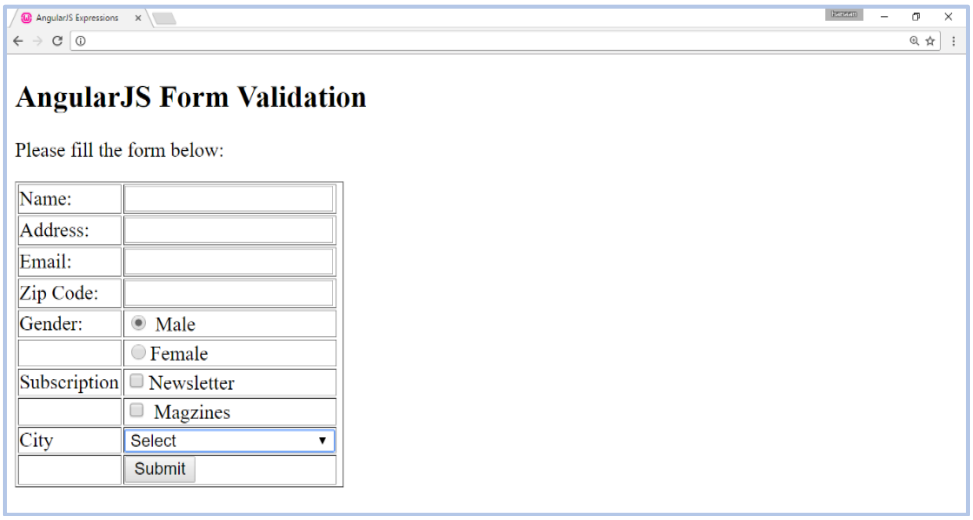

Figure 8.17: Output of code 8.11

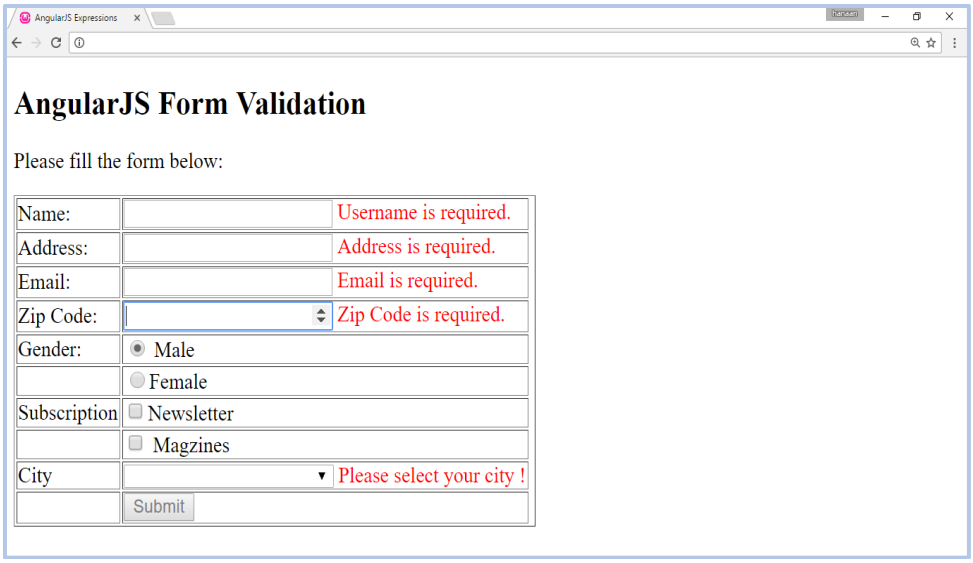

Figure 8.18: Output of code 8.11

Video Lecture

https://youtu.be/ZFCPlpIS3nU

Unit Summary

In this unit, we learned about the basic concepts of AngularJS and how to use AngularJS expressions in HTML page. We used different directives to enhance the functionality of HTML page. We also used AngularJS filters to format different data elements. The unit also covered how to validate form elements using AngularJS.

Self Assessment Questions

Choose the correct answer.

1. Data binding in AngularJS is based on _____ pattern
   A. MCV.
   B. MVC.
   C. CMV.
   D. None of the above.

2. Conditional statements can be incorporated in AngularJS.
   A. True
   B. False

3. _____ directives start an AngularJS application.
   A. ng-app
   B. ng-controller
   C. ng-init
   D. ng-show

4. AngularJS controller maintains the data and behavior by using _____ object.
   A. $object
   B. $parameter
   C. $app
   D. $scope

5. Which of the following is true about currency filter?
   A. Currency filter changes the original value of a number.
   B. Currency filter adds some currency to the existing value.
   C. It formats a number to a currency format.
   D. It formats a string to uppercase.

6. AngularJS expression are written using:
   A. double braces {{expression}}
   B. single braces {expression}.
   C. parenthesis (expression)
   D. capital bracket [expression]

7. Which of the following is true about “ng-init” directive?
   A. It starts an AngularJS application.
   B. It repeats html elements for each item in a collection.
   C. It defines the model that is variable to be used in AngularJS.
   D. It initializes application data.

8. _____ directive repeats html elements for each item in a collection
   A. ng-app
   B. ng-init
   C. ng-repeat
   D. ng-show

9. AngularJS supports two way data binding?
   A. True
   B. False

10. Filters can be applied in AngularJS application by using _____ character.
    A. string
    B. pipe
    C. concatenation
    D. bracket

Answer Key:

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

Review Questions

Write short answers of the following questions

1. What is meant by AngularJS?
2. Briefly explain the concept of directives in AngularJS.
3. How valid expressions are written in AngularJS?
4. Describe important filters with example.
5. What is meant by $scope?
6. Differentiate between one way and two way data binding.
7. How are form validations implemented in AngularJS?

</> Coding Exercise

1. Create a salary slip that manipulates basic pay, allowances and income tax. Display the report by using currency filter.
2. Create a list of your favorite cars and display it in your web page by using “ng-repeat” directive.
3. Consider any three numbers and calculate their sum, difference and product by using AngularJS expressions.
4. Create a web page that takes first name, last name as inputs and display the name in upper and lowercase as output.
5. Create a web page that takes the following input from the user: first name, last name, semester fee and three subjects. Change first name into uppercase, last name into lower case, semester fee in your local currency and display subject name in alphabetical order.
6. Modify the admission form created in coding exercise of unit 2 and apply validation techniques using AngularJS to different form elements.

References and Further Reading

1. Williamson, K. (2015). *Learning AngularJS: A Guide to AngularJS Development*. " O'Reilly Media, Inc.".
2. Freeman, A. (2014). *Pro AngularJS*. Apress.
3. Frisbie, M. (2014). *AngularJS Web Application Development Cookbook*. Packt Publishing Ltd.
4. Green, B., & Seshadri, S. (2013). *AngularJS*. " O'Reilly Media, Inc.".
5. AngularJS Quick Guide. Available on: http://www.tutorialspoint.com/angularjs/angularjs_quick_guide.htm
6. AngularJS Filters. Available on: https://www.w3schools.com/angular/angular_filters.asp
7. AngularJS Events Available on: http://followtutorials.com/pdf/angularjs-tutorial.html
8. AngularJS Scope. Available on: https://www.w3schools.com/angular/angular_scopes.asp
9. AngularJS Directives. Available on: https://www.w3schools.com/angular/angular_model.asp
10. AngularJS Introduction. Available on: https://www.w3schools.com/angular/angular_intro.asp
11. Data Binding. Retrieved from https://docs.angularjs.org/guide/databinding

UNIT 9

Web Security

Introduction

When we discuss the protection of information over the world wide Web, a sudden thought comes in mind regarding the hacking of websites, defacing information of credit cards numbers and the misuse of confidential information. Further, we think about the viruses and other mischievious codes which threaten the security of websites. However, the mother of all these problems is unawareness and ignorance to protect sensitive data over the world wide web.

This chapter will enable the students to get a detailed introduction to the basic concepts of web security and major terminologies used in the web security. Furthermore, this chapter is helpful in understanding the basic principles of web security and the process of securing webs with the help of form validation, human validation, SQL injections and CAPTCHTA implementation in order to overcome the web applications vulnerabilities.

Unit Outcomes

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

1. Understand basic concept of web security.
2. Have knowledge about the principles of web security.
3. Be familiar with common client side attacks.
4. Know about CAPTCHA role and its implementation.
5. Protect web applications from vulnerabilities of security threats.

Terminologies

CAPTCHA:	Completely Automated Public Turing test to tell Computers and Humans Apart.
SQL Injection:	SQL Injection is used to attack data driven applications.
Attack:	Implementation to exploit vulnerabilities of the web security.
Thread:	A presence of a potential danger to the integrity of web resources.

9.1. Web Security

When you start designing a website, you should adopt a pro-active approach towards managing the security risks linked to the web resources and user information. Therefore, it is very necessary to apply adequate risk handling techniques to protect the web applications from illegal access and misusage. The concept of web security refers to the protection of websites in order to save information from illegal use. Web security deals with the security threats and major principles of securing websites. Further, the solution of security threats is not merly technology related but the effective practices of the people who use websites.

According to Microsoft, the methods used for the assessment of web security risks are not as important as the actually models of performing structured security management. It has indentified the following model of risk handling as shown in figure 9.1.

1. Identification of security objectives.
2. Survey the web application.
3. Decomposition of the application.
4. Identification of potential threats.
5. Identification of potential vulnerabilities.

Web security deals with the security threats and major principles of securing websites.

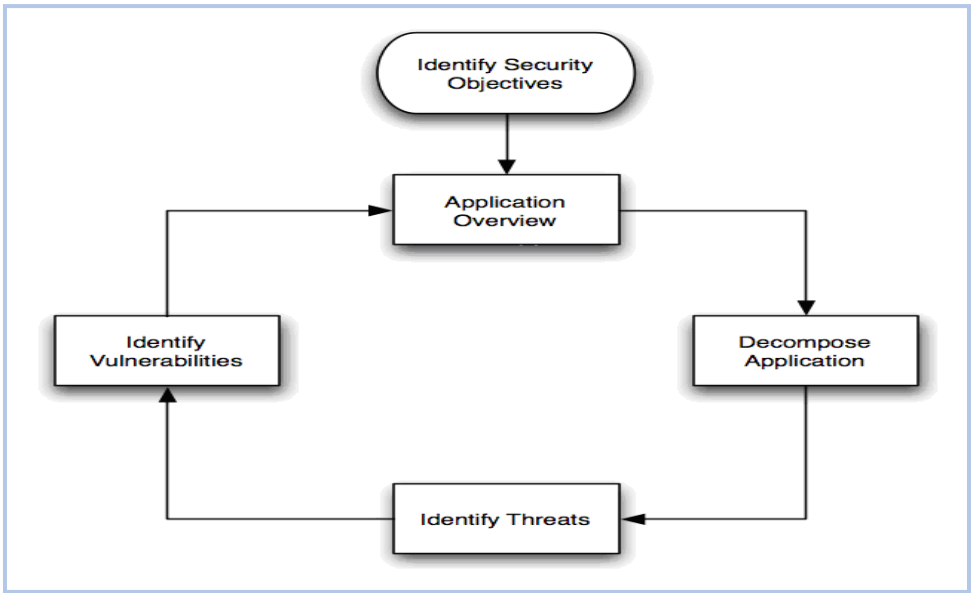

Figure 9.1: Model of risk handling ("Threat Risk Modeling", n.d.)

9.2. The Principles of Web Security

Generally, the security of web applications is related to the protection of assets such as information given on web and the user's information in the shape of contact numbers and credit cards detail. Therefore, web security is actually the management of risks related to the misuse of information available on World Wide Web. The foundation of web security is based on the elements given below:

9.2.1. Availability

The element of availability ensures that web resources remain accessible to the users without any error or denial of the service. The first and foremost objective of the web security is to provide the availability of web sources to all the users.

9.2.2. Authentication

Authentication is a process to identify the users as the authentic person to use the web. It basically gets the answer to the question that "who are you?". Authentication is not limited to the users only. It also identifies that the accessed data is being used by the persons or any other services module.

9.2.3. Authorization

Authorization is a process of supervising operations and web resources so that only authentic users can access the allowed functionality. It basically deals with the question "what are you permitted to do?" The operations on the web application define the access rights of the user on the basis of their category as if the user were an administrator, an editor or a guest.

9.2.4. Confidentiality

The element of confidentiality guarantees the privacy of the web resources and the users' information. The process ensures that data is only accessible to the authorized users and it is not possible to be accessed by the unauthorized users.

9.2.5. Auditing

The process of auditing ensures that the users perform the authorized activity in an effective manner without leaving any operation undone. For example while making a transaction online through an online shopping website; the user is required to perform all the key activities to accomplish the task successfully.

9.2.6. Integrity

The elements of integrity guarantees that the web resources and users' data are secured from the unintentional or deliberate modification. It is as important as the privacy is vital for the security of the web applications. The integrity of data is ensured by using various authentication techniques.

Availibility, authentication, authorization, confidentiality and integrity are the basic priciples of web security.

Activity 1

Consider a web based system and explain how principles of web security can be implemented.

9.3. Common Client-Side Attacks

As the communication has increased on the web, the unauthorized accesses to accounts and assests has also increased. This section explains few of the categories of client-side attacks. The methodology is being discussed along with the techniques for the counter-measures for each category.

9.3.1. Eavesdropping Attacks

This kind of attack is expedited after listenning to the network traffic of the other user. This listening is through DNS queries, HTTP requests and their consective responses. These kind of attacks are very common nowadays as the use of wirless networks are growing rapidly. Special security protocols can be deployed to avoid these kinds of attacks such as WPA2 and EAP. These protocols are built over different kinds of encryptions with special keys for the implementation.

9.3.2. Man-in-the-Middle Attacks

In this type of attack, the attacker places himself in the network, inpecting all the traffic manually and alter or insert the packets of interests in them. Attackers from all over the world have the tendency of using this kind of attack. There are softwares avalaible for this kind of attack which facilitate the attacker for the exececution of the attack. Mostly, these attacks are for the illegal bank transactions. These attacks are mitigated through HTTPS secure browser methodology.

9.3.3. Cross-Site Request Forgery

In this type of attack, the attacker forges a request to the target application from a ligimate user browser to modify the account setting or stealing the money etc. The mitigation is implemented on the server side which enforces the cross checks of the referrer. This enables the avoidance of such kind of attacks.

9.3.4. UI Redressing

This type of attack is also known as clip-jacking or tapjacking. In this attack the target application is confused with the interaction like a potential client. The attacker can avail any privilage as clicking on a button and reconfiguring any controls etc. A special “framebusting” code is incorporated inside the client side block for the detection of illegal framing of an application.

9.3.5. Session Hijacking

This type of attack let the attacker to gain control of an authenticated session from the victim’s browser to his browser. After this session tranferance the attacker can act as a user and can perform all the actions available for the user. To avoid this, we required “IP address binding” where the server binds the specific client IP with the specific session.

9.3.6. Cross-Site Scripting

In this attack the attacker gain control through his own JavaScript code to control the target application. This is marked as a serious problem in the current world of hacking and security. Advanced multiple solutions for the defence of such attacks are also available.

Client side attack requires user interaction suchas clicking a link or button and requesting a document.

Activity 2

Explore common client side attacks. Identify practical problems and find their solutions.

9.4. Security Threats

The enormous use of web has augmented the sharing of information as a great mean of the business adoption and socialization. It has multiplieds the threads of information theft and its misuse. Website applications are usually prone to risk from the user side while compromising user authentication and data confidentiality. The cross-site scripting (XSS) and SQL injection are the common ways of web application attacks which are used to create ambiguous and flawed coding results.

A threat is referred to any malicious or harmful incidence that can potentially damage an asset. The security threat with respect to web application is the malicious code that can either change or destroy the actual functionality of the web application and the misuse of users' data. Further, a defect or weakness in web application makes it more vulnerable to security threats. Moreover, the probability of security threats becomes high in case of mistakes in configuration, poor layout design and inappropriate techniques of coding.

Attacking of website from user side violates user authentication and data confidentiality.

9.4.1. SQL Injection

SQL Injection which is also known as an attack vector is the most commonly used method to attack web applications. It is usually used to inject SQL code in data driven web applications for the execution. It allows the attacker to adopt any identity, interfere in the privacy of others' data and change the existing information and to modify the transactions or balances of any person. The SQL injection code makes a dump copy of the database and sends it to the attacker. A potential vulnerable web application allows SQL injection work. For example, while entering data into web form, the fields of the form are not much secure to filter incorrect input; that time the attacker can escape security and can embed SQL injection into the database.

SQL injection code makes a dump copy of the database and sends it to the attacker.

The given below code segment is helpful in understanding the execution of SQL injection. Look at the following example which creates a SELECT statement by adding a variable (studentid) to a select string. The string is fetched from the text box having an id ("userid") using the function ("getRequestString") :

```
studentid = getRequestString("userid");
query = "SELECT * FROM Users WHERE UserId = " + studentid;
```

The above statements will create an SQL query to select a student with the given user id only. The user can however, give additional conditions with the input to be a part of the query. In the following statement, the user has "deliberately" given "OR 1=1" in addition to his id number.

```
UserId: 231 OR 1=1
```

This additional string will become the part of the main query also as shown below:

```
SELECT * FROM Users WHERE UserId = 231 OR 1=1;
```

The above SQL statement above is valid and will return "ALL" rows from the "Student" table, since OR 1=1 is always TRUE instead of only returning the data for the user id 231. Hence this provision can lead to access the data which was not meant to be given to that's specific student. The situation is aggravated further when the data contains classified information and passwords etc.

Video Lecture

https://youtu.be/y0zKvb1J_QM

Activity 3

- Suppose you have received an email alerting that a successful SQL injection has been detected against your website. How you will handle such attack?
- Give an example of SQL injection vulnerability.

9.5. Form Validation and Security

Forms validation is a useful way to secure web application from the potential attack. It secures both server and client sides. However, it is very valuable for client side because it reduces time and enhances bandwidth. It helps to keep the users on right direction while filling out data into the form. Therefore, while form filling, if any wrong entry or left blank is submitted, the error message will occur and direct the user to refill all the required fields properly according to the predefined pattern before submission. Form validation have already been covered in unit 5 section 5.5.

Form validation helps to keep the users on right direction while filling out data into the form.

Activity 4

Create a simple login page with “username” and “password” options. Make your login page secure by using form validation and security options.

9.6. CAPTCHA Role and Implementation

A CAPTCHA stands for Completely Automated Public Turing Test to tell Computers and Human Apart. The first CAPTCHA was used in 1997 by the two groups of people working together. It is a challenge or guess type test to reveal if the person who is accessing web application is a human or not. As per routine, a CAPTCHA requires the user to enter a set of data including letters and numbers from a disarranged pattern. The chief objective of the CAPTCHA is to restrict the response for non human requests to secure the web server. Numerous web applications are used to manipulate CAPTCHA for the validation of human beings. CAPTCHA can be generated by using different ways, however, JavaScript is used to perform the complete the verification of CAPTCHA code. Code 9.3 demonstrates the use of CAPTCHA code.

CAPTCHA is a type of challenge response test to determine whether or not the user is human.

</> Code 9.1

```
<! DOCTYPE html>
<html>
<head>
<title>JavaScript Forms</title>
<script type="text/javascript">
function Captcha(){
  var alpha = new
       Array('A','B','C','D','E','F','G','H','I','J','K','L','M',
       'N','O','P','Q','R','S','T','U','V','W','X','Y','Z','a','b
       ','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
       'q','r','s','t','u','v','w','x','y','z');
  var i;
     for (i=0;i<6;i++){
       // Declaring variables and add math random functions in these
            variables.

         var a = alpha[Math.floor(Math.random() * alpha.length)];
         var b = alpha[Math.floor(Math.random() * alpha.length)];
         var c = alpha[Math.floor(Math.random() * alpha.length)];
         var d = alpha[Math.floor(Math.random() * alpha.length)];
         var e = alpha[Math.floor(Math.random() * alpha.length)];
         var f = alpha[Math.floor(Math.random() * alpha.length)];
         var g = alpha[Math.floor(Math.random() * alpha.length)];
     }

     var code = a + ' ' + b + ' ' + ' ' + c + ' ' + d + ' ' + e +
     ' '+ f + ' ' + g;
     document.getElementById("mainCaptcha").value = code;
  }
     function ValidCaptcha(){
```

```
      var string1
      removeSpaces(document.getElementById('mainCaptcha').value);

     // Remove spaces from the “mainCaptcha” value.
      var string2 =
      removeSpaces(document.getElementById('txtInput').value);
      if (string1 == string2){
        return true;
       }
      else{
         return false;
     }
        }
     function removeSpaces(string){
     return string.split(' ').join('');
    }
</script>
</head>
<body onload="Captcha();">
   <table>
     <tr>
         <td> Text Captcha <br> </td>
     </tr>
     <tr>
        <td> <input type="text" id="mainCaptcha"/>
        <input type="button" id="refresh" value="Refresh"
          onclick="Captcha();" /> </td>
        // “onclick” event will refresh new captcha .
     </tr>
     <tr>
         <td><input type="text" id="txtInput"/> </td>
     </tr>
     <tr>
         <td><input id="Button1" type="button" value="Check"
               onclick="alert(ValidCaptcha());"/> </td>
       // call function “ValidCaptcha()” on “onclick” event.
     </tr>
   </table>
</body>
</html>
```

The ValidCaptcha() is called when the Onclick button event is triggered and a Boolean (yes, no) value is calculated. Where the method ValidCaptcha() compares the code entered by the user after displaying CAPTCHA code in the code box. The method RemoveSpaces(string) eliminates the probability of blank spaces in order to make comparison successful. The method ValidCaptcha indicates the result after comparison that if the entered code is true or false. The output of code 9.1 is shown in figure 9.2 and 9.3.

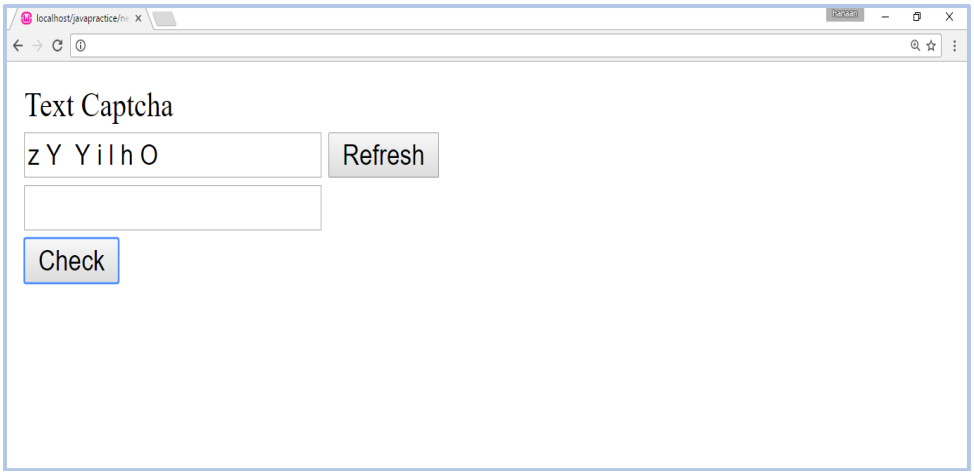

Figure 9.2: Output of code 9.3

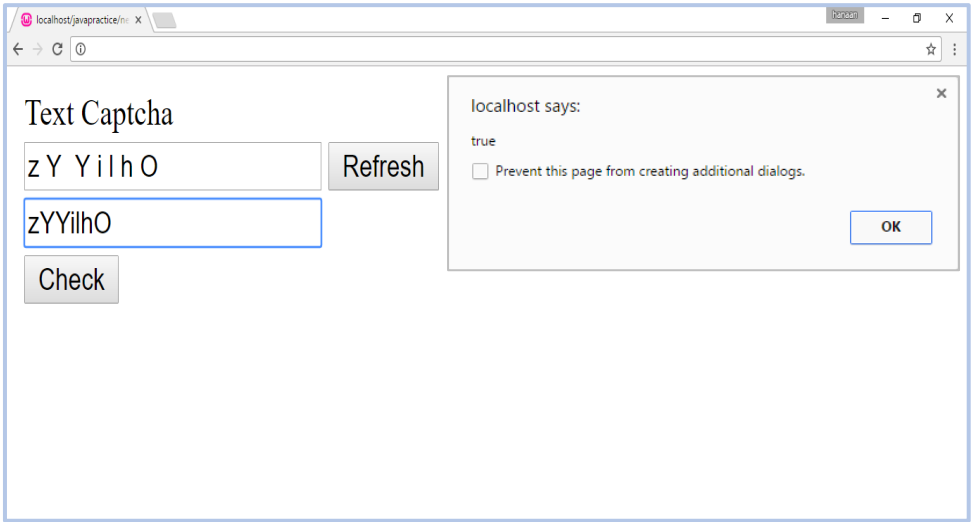

Figure 9.3: Output of code 9.3

Activity 5

Modify code 9.1 and give different format of CAPTCHA inputs.

Unit Summary

In this unit, we learned about the basic concept of web security and how to implement web security in client side scripting. We came to know about common client side attacks and related security threats. We also learned about CAPTCHA role and its implementation in web security.

Self Assessment Questions

Select the correct answer.

1. Web security deals with _____
   A. security of MS Office
   B. protecting web resources and user data
   C. computer protection from heat and dust.
   D. None of the above

2. According to Microsoft, the number of steps in applications' security model are:
   A. 4
   B. 3
   C. 5
   D. None of the above.

3. SQL Injection is used to
   A. dumb database illegally
   B. add injection of data
   C. add language
   D. Both A and B

4. CAPTCHA stands for?
   A. Completely Automated Public Turing test to tell Computers and Humans Apart
   B. Come At Public Turing Temperature to Talk and Hang out Apart
   C. Camp At Pakistan Public Turing test to tell Computers
   D. None of the above

5. Forms validation can be performed at :
   A. client side
   B. server side
   C. Both, client and server side
   D. None of the above

6. _____ means that computer system can be used only by authorized users.
   A. Confidentiality
   B. Authentication
   C. Auditing
   D. Availability

7. The elements of _____ guarantees that web resources are secured from unintentional modification
   A. Integrity
   B. Availability
   C. Authentication
   D. Confidentiality

8. In _____ type of attack, the attacker inspect all trafic by placing himself in the network.
   A. Cross-Site Request Forgery
   B. UI Redressing
   C. Session Hijacking
   D. Man-in-the-Middle Attacks

9. Security threat is the _____ that can change or destroy the data.
   A. trigger
   B. popup box
   C. malicious code
   D. output

10. SQL injection also known as _____ .
   A. attack scaler
   B. attack direction
   C. attack vector
   D. scaler vector

Answer Key:

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

Review Questions

Write short answers of the following.

1. What is web security and why is it important?
2. Explain CAPTCHA and its working technique.
3. Write at least five principles of web security.
4. What are common client side attacks?
5. Identify the potential threats posed to web applications.
6. Define SQL Injection and how it is a threat for web applications.

</> Coding Exercise

1. Use the given code of SQL Injection, perform the practical implementation.
2. Design a web page and implement CAPTCHA validation through it.
3. Modify the admission form created in coding exercise of unit 2 and apply validation techniques to different form elements using CAPTCHA and SQL injection.

References and Further Reading

1. Wu, H., & Zhao, L. (2015). *Web Security: A WhiteHat Perspective*. CRC Press.
2. De Ryck, P., Desmet, L., Piessens, F., & Johns, M. (2014). *Primer on client-side web security*. Springer.
3. Security Model for the Client-Side Web Application Environments Available on: http://www.w2spconf.com/2007/slides/20070524-w2sp-yoshihama-slides.pdf
4. The web application hacker's handbook. Available on: https://leaksource.files.wordpress.com/2014/08/the-web-application-hackers-handbook.pdf
5. SQL injection. Available on: https://www.w3schools.com/sql/sql_injection.asp
6. Web Application Security for Dummies. Available on: http://www.bradreese.com/qualys-web-application-security-for-dummies.pdf
7. Primer on Client-Side Web Security. Available on: https://link.springer.com/book/10.1007%2F978-3-319-12226-7
8. An Efficient Protective Layer Against SQL Injection Attacks Available on: http://ijcsi.org/papers/IJCSI-11-4-1-38-44.pdf
9. Threat Risk Modeling. Retrieved from https://www.owasp.org/index.php/Threat_Risk_Modeling

THE COMMONWEALTH of LEARNING

Department of Computer Science
Allama Iqbal Open University, Pakistan