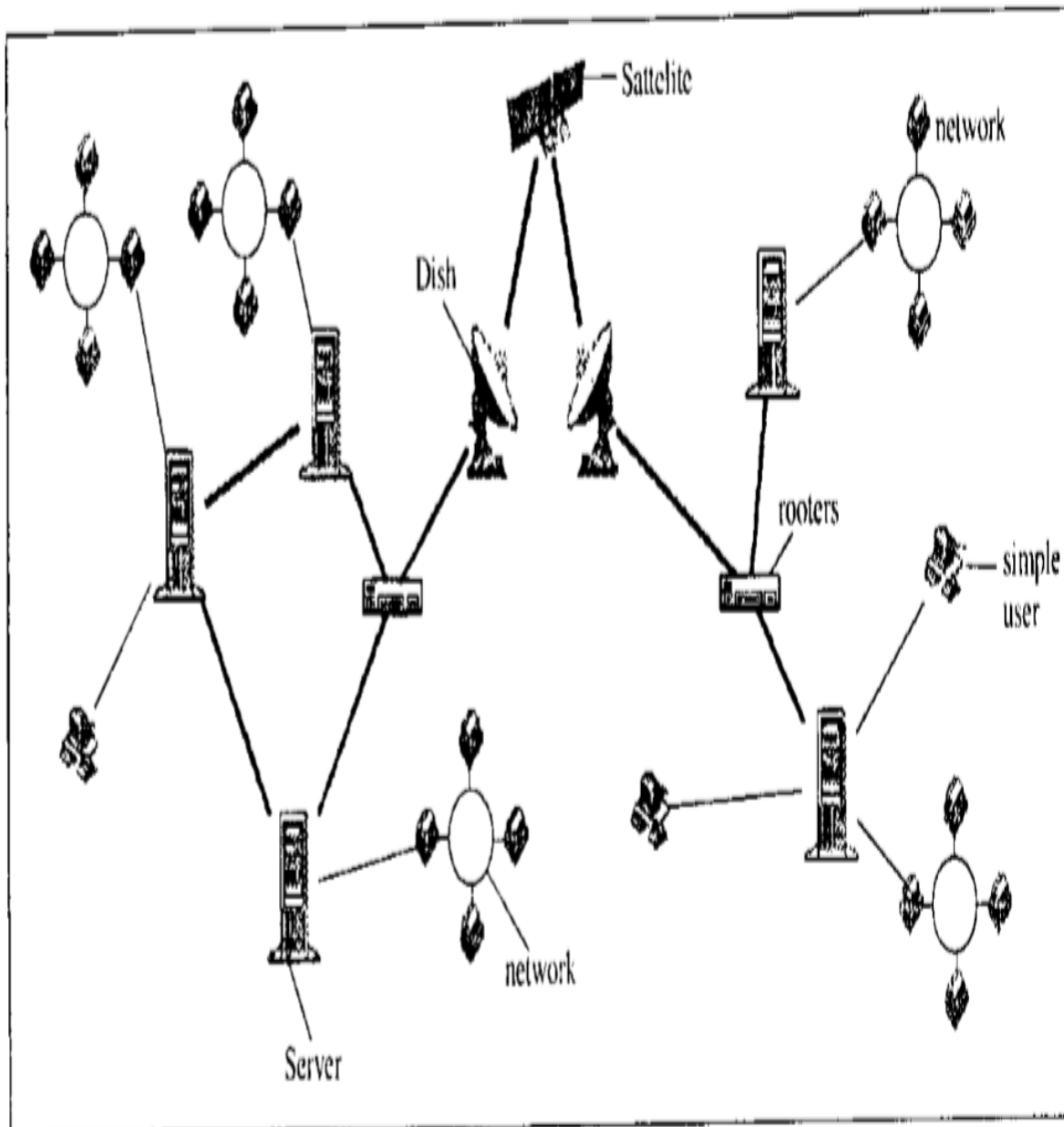


FEDERAL UNIVERSITY OF KASHERE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

CSC2308: INTERNET TECHNOLOGY
LECTURE NOTE

STATUS: CORE

CREDIT UNIT: 3



PREPARED BY

NUHU ABDUL'ALIM MUHAMMAD (EL-GURAMA)

COURSE OUTLINE

1. Internet: Definition, History, services, protocols, advantages and disadvantages.
2. WWW: Definition, History and fundamental concepts.
3. HTTP: TCP/IP, URLs server technology.
4. HTML: Document structure, images, maps, table, frames and forms.
5. Cascading Style Sheets (CSS):
6. JavaScript: Syntax, DOM, Form processing, common tasks.
7. Introduction to PHP:
8. DHTML:
9. Web Design and Usability:
10. Multimedia:
11. Web development tools:
12. Lab Exercises and Group Projects

LECTURE TIME

Monday

10am-12noon

New 350 CLT

Tuesday

10-12 noon

SCI 1

CHAPTER ONE

INTERNET

DEFINITION:

The internet can be defined in many ways, some of the definitions are as follows:

The Internet is a network of networks (Local Area Networks, Metropolitan Area Networks and Wide Area Networks) across the world.

Internet is defined as an Information Super Highway that is used to access information over the web.

Internet is a world-wide global system of interconnected computer networks.

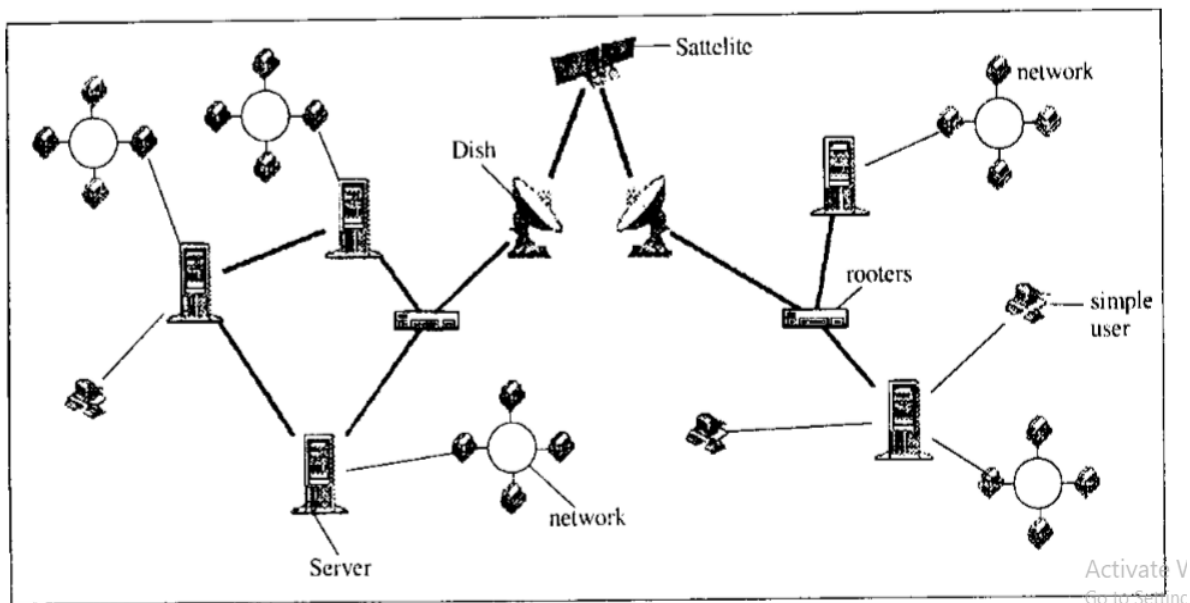


Fig.1.1: The Physical Components of the Internet

HISTORY

The Internet started as a project of the United States Government's Department of Defense (DOD), to create a non-centralized network. This project was called ARPANET (Advanced Research Projects Agency Network), created by the Pentagon's Advanced Research Projects Agency established in 1969 to provide a secure and survivable communications network for organizations engaged in defense related research.

ARPANET Initially started with four nodes (Hosts) They include:

- a. University of California, Los Angeles (UCLA).
- b. University of California, Santa Barbara (UCSB).
- c. University of Utah.
- d. STI (Stanford Research Institute).

In 1972, the ARPANET spread over the globe with 23 nodes located at different countries and thus became known as Internet.

In 1977, The standard protocol called TCP/IP (Transmission Control Protocol/Internet Protocol) was invented. TCP/IP allowed users to link various branches of other complex networks directly to the ARPANET, which soon came to be called the Internet.

By the time, with invention of new technologies such as TCP/IP protocols, DNS, WWW, browsers, scripting languages etc. Internet provided a medium to publish and access information over the web.

INTERNET SERVICES

The following are some of the services that can be accessed over the internet:

- a. **Communication Services:** these are services that offer exchange of information with individuals or groups. They include:

S.N.	Service Description
1	Electronic Mail Used to send electronic message over the internet.
2	Telnet Used to log on to a remote computer that is attached to internet.
3	Newsgroup Offers a forum for people to discuss topics of common interests.
4	Internet Relay Chat IRC Allows the people from all over the world to communicate in real time.
5	Mailing Lists Used to organize group of internet users to share common information through e-mail.
6	Internet Telephony VoIP Allows the internet users to talk across internet to any PC equipped to receive the call.
7	Instant Messaging Offers real time chat between individuals and group of people. Eg. Yahoo messenger, MSN messenger.

- b. **Information Retrieval Services:** these are services that offer easy access to information on the internet. The following table gives a brief introduction to these services:

S.N.	Service Description
1	File Transfer Protocol FTP Enable the users to transfer files.
2	Archie It's updated database of public FTP sites and their content. It helps to search a file by its name.
3	Gopher Used to search, retrieve, and display documents on remote sites.
4	Very Easy Rodent Oriented Netwide Index to Computer Achieved VERONICA VERONICA is gopher based resource. It allows access to the information resource stored on gopher's servers.

- c. **Web services:** these are services that allow exchange of information between applications on the web. Using web services, applications can easily interact with each other.
 - i. **World Wide Web (WWW):** It offers a way to access documents spread over the several servers over the internet. These documents may contain texts, graphics, audio, video, hyperlinks. The hyperlinks allow the users to navigate between the documents.

- ii. Video Conferencing (teleconferencing) is a method of communicating by two-way video and audio transmission with help of telecommunication technologies.

INTERNET PROTOCOLS

These are different set of rules that are used to send data, packets and files over the internet they include:

a. Transmission Control Protocol (TCP)

TCP is a reliable and connection oriented protocol.

It is a connection oriented protocol meaning that it requires a connection between nodes for data exchange to occur.

Reliable means there is a mechanism to ensure that data sent is received at the destination.

b. User Datagram Protocol

UDP is connectionless and unreliable protocol.

connectionless means that it doesn't require making a connection with the host to exchange data.

Unreliable means there is no mechanism for ensuring that data sent is received.

c. File Transfer Protocol FTP

FTP is an internet protocol that is used to copy files from one host to another.

FTP copies files by creating two processes:

FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.

FTP establishes two different connections: one is for data transfer and other is for control information.

Control connection is made between control processes while Data Connection is made between

FTP uses port 21 for the control connection and Port 20 for the data connection.

d. Trivial File Transfer Protocol TFTP

TFTP is an internet protocol that is used to transfer files without authentication. Unlike FTP, TFTP does not separate control and data information. Since there is no authentication exists, TFTP lacks in security features therefore it is not recommended to use TFTP.

TFTP makes use of UDP for data transport. Each TFTP message is carried in separate UDP datagram.

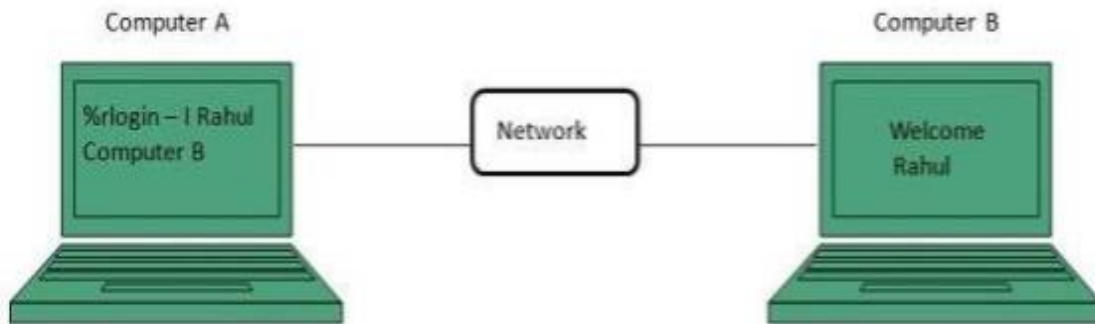
The first two bytes of a TFTP message specify the type of message.

The TFTP session is initiated when a TFTP client sends a request to upload or download a file.

The request is sent from an ephemeral UDP port to the UDP port 69 of an TFTP server. Telnet

e. Telnet

Telnet is an internet protocol that is used to log in to remote computer on the internet. There are a number of Telnet clients having user friendly user interface. The following diagram shows a person is logged in to computer A, and from there, he remote logged into computer B.



f. Hyper Text Transfer Protocol HTTP

HTTP is an internet protocol that is used for communication between a browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

ADVANTAGES OF THE INTERNET

Internet covers almost every aspect of life. It is useful in the following ways:

a. Communication

Internet allows us to communicate with the people in different locations. There are various apps available on the web that use Internet as a medium for communication e.g.

Facebook, Twitter, Yahoo, Google+, Flickr, Orkut e.t.c.

b. Research

One can browse (surf) any kind of information over the internet. Information regarding various topics such as Technology, Health & Science, Social Studies, Geographical Information, Information Technology, Products etc. can be surfed with help of a search engine.

c. Entertainment

The Internet is a tool for enjoying one's leisure time. There are the various modes for entertainment over internet. They include:

Online Television

Online Games

Music

Videos

- d. E-Commerce (electronic commerce) that allows the business deals to be conducted on electronic systems.
- e. Internet Banking.
- f. Matrimonial Services.
- g. Online Shopping.
- h. Online Ticket Booking.
- i. Online Bill Payment.
- j. Data Sharing.

DISADVANTAGES OF THE INTERNET

There are some disadvantages associated to the internet, they include the following:

a. Cybercrime

Personal information such as name, address, credit card number and even money can be stolen over the internet. Therefore, one should be very careful while sharing such information. One should use credit cards only through authenticated sites.

b. Spamming

Spamming refers to the unwanted e-mails in bulk. These e-mails serve no purpose and lead to obstruction of entire system.

c. Virus

Virus can easily be spread to the computers connected to internet. Such virus attacks may cause your system to crash or your important data may get deleted.

d. Pornography

There are many pornographic (explicit sexual content) sites that can be found, letting your children to use internet which indirectly affects the children healthy mental life.

e. Fake news and information

There are various websites that do not provide the authenticated information. This leads to misconception among many people.

EXERCISES

1. Define the internet.
2. Give a brief history of the internet.
3. Draw the diagram of the physical components of the internet
4. List all the nodes that ARPANET started with.
5. List and explain all the internet services.
6. List and explain all the internet protocols.
7. List and explain all the advantages of the internet.
8. List and explain all the disadvantages of the internet.

PROJECT

Every student should by his/herself open:

- a. A Gmail account (Compulsory).
- b. A WhatsApp account (Optional but highly recommended).
- c. A Facebook account (Optional).
- d. Buy an item on any E-commerce site (Optional). Jumia, Konga, Jiji e.t.c
- e. A bank account (Optional but highly recommended).

*The class rep should send the list of all students and their project

CHAPTER TWO

WORLD WIDE WEB

DEFINITION:

The World Wide Web (WWW) is a dynamic collection of documents that contain text, graphics, sounds, and movies etc. That can be accessed on the internet.

World Wide Web (WWW) is a collection of documents that are linked together and can be accessible via the internet.

The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the Internet.

HISTORY OF THE WEB

The WWW was developed at the European Particle Physics Lab (CERN) in Switzerland by Tim Berners-Lee in 1989 to allow information-sharing among internationally dispersed teams of high-energy physics researcher.

World Wide Web Consortium is formed in September 1994, with a base at MIT in the USA, INRIA in France, and now also at Keio University in Japan.

Factors Contributing to the Growth of World Wide Web

There are many Factors that contributed to the early growth of the web some of which includes:

- i. Open architecture. Open standards and freely published structure and code.
- ii. Minimal system requirements.
- iii. Administrators with limited funds could participate as information providers.
- iv. Interoperability between different computer platforms.
- v. Hypertext linking.
- vi. Established Internet protocols and tools File Transfer Protocol (FTP), Gopher, and Telnet
- vii. The needs for information storage and retrieval.

FUNDAMENTAL WEB CONCEPTS

- a. **WEB Servers:** A web server is the program that runs on a computer and is responsible for replying to web browser requests for files. To publish pages on the Web, you need a web server. You need a web server to publish documents on the Web. When you use a browser to request a page on a website, that browser makes a web connection to a server using the HTTP protocol. The browser then formats the information it got from the server. Server accepts the connection, sends the contents of the requested files and then closes.
- b. **Web Browsers:** A browser is a software program that allows users to access and navigate the World Wide Web via hyperlinks. To access the World Wide Web, you must use a Web browser.

There are two types of browsers:

- i. **Text:** these are text only web browsers. Navigation is accomplished by highlighting emphasized words in the screen with the arrow up and down keys, and then pressing the forward arrow (or Enter) key to follow the link. Lynx is a browser that provides access to the Web in text-only mode.
- ii. **Graphical:** these are web browsers that allows you to access all types of multimedia formats. Navigation is accomplished by pointing and clicking with a mouse on highlighted words. Examples are Netscape Navigator, Internet Explorer e.t.c. These browsers are available for both Windows-based and Macintosh computers.
- c. **Hyperlinks:** are links that are used to take a user to another page when you click on them. When you scroll over them with your mouse, you'll see that the pointer changes to indicate that a link is present. Pictures can also be hyperlinks. If a picture is a hyperlink, your mouse pointer will change when you move over it. Hyperlinks are a key element of how the web works. You don't have to type in the address of every page you visit: in fact, you rarely have to type in any addresses at all. Instead you follow the hyperlinks that lead you between sites and to new pages. That's why it's called the web.
- d. **HyperText:** is a word (text) or group of words that connect to other pages, documents, images, video, and sounds. These words are called links and are selectable by the user. A single hypertext document can contain links to many documents.
- e. **Internet Protocol Suite:** These are different set of rules that guides communication over the internet. They are used to send data, packets and files over the internet.

EXERCISE

1. Define world wide web.
2. List all the objectives of world wide web development.
3. Who developed the world wide web.
4. When was the world wide web developed.
5. List all the factors that contributed to the growth of world wide web.
6. List and explain all the fundamental web concepts.

CHAPTER THREE

URLS SERVER TECHNOLOGY, HTTP, TCP/IP

URL (Uniform Resource Locator). A URL is popularly known as web address.

The URL makes it possible for a computer to locate and open a web page on a different computer on the Internet.

DEFINITION: A URL is the fundamental network identification for any resource (e.g., pages, images, sound, files etc.) found on the web.

A URL is an address that shows where a particular page can be found on the World Wide Web.

HISTORY OF URL

In 1969, the Advanced Research Projects Agency Network (ARPANET) sent the first node-to-node transmission from UCLA to the Stanford Research Institute.

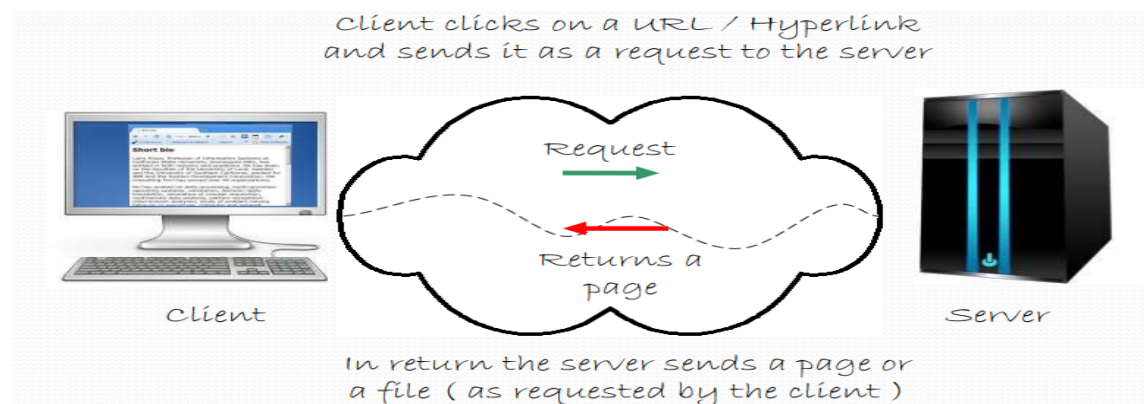
in 1970, Network Control Protocol (NCP) Consisting of the Physical, Data, and Network layers, a Transport layer was introduced.

In 1983, ARPANET switched from NCP to the Transmission Control Protocol/Internet Protocol (TCP/IP).

In 1989, Tim Berners-Lee created the World Wide Web (WWW), introducing the markup language, resource identifiers, and protocols for retrieving resources.

In 1992, Tim Berners-Lee created the URL as a tool to link to the location of any internet resource a user may require.

WORKING OF THE URL



COMPONENTS OF URL

The four main components of URLs are the protocol, domain, path, and query.

Let us have a closer look at the different URL components, using the following example:

`https://www.example.com/category-A/subcategory-A1/model-123.html`

i. Protocol: The protocol or scheme of a URL indicates the method that will be used for transmitting or exchanging data. The most familiar scheme is the Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) for the transmission of HTML files. Other Protocols can be:

i. `http://`

ii. `https://`

iii. `ftp://`

iv. `gopher://` etc.

In the example URL above, `https://` is the URL's secure protocol.

ii. Domain

The domain or hostname of a URL is a user-friendly expression of the Internet Protocol (IP) address of a website. It points to the location of the website's host server.

In the example above, the domain is `www.example.com`.

iii. Path

The path points to a specific file or other resource location.

In our example URL, `/category-A/subcategory-A1/model-123.html` shows the path of the URL, which in this example, ends in a product page.

iv. Query

The query string, also known as a fragment identifier, is frequently used for internal searches and is commonly preceded by a question mark (?).

BASIC COMPONENT OF URL

- i. Protocol
- ii. WWW
- iii. Domain name
- iv. Resource name
- v. Question mark signal “?”
- vi. Parameters

Protocols

Every URL starts with a name (Protocol) which personal computer and server should communicate.

Http is the mostly used protocol name.

Other specific protocols are ftp, smtp and https.

Protocols are followed by symbol “://”

WWW

stands for World Wide Web. It simply indicates that the resource that are seeking for resides on the internet. This portion of the URL is not required and many times can be left out.

Domain Name

The name of resource owner comes which are youtube and example in the above structures

Domain name is followed by the type of domain which the user is registered.

.com, .gov, .edu, .org,

Resource Name

In above first structure, resource name is the watch in the webpage that user is seeking.

If the structure is seen in totality “watch” is a webpage that resides on a website called YouTube.

Question Mark Signal “?”

It signals the end of entire domain name and beginning of the parameter list.

Some webpages accept certain parameters to be passed on them.

This is necessary to function properly

Parameters

Parameters are passed in a “key=value” format and parameter in the list are separated from each other by a character “&”

So in above first structure two parameters are passed (V=YR12&feature=relmfu)

STRUCTURE OF URL



CHARACTERS ALLOWED IN URLS

The characters that are allowed in URLs are specified by the World Wide Web Consortium. The specifications include lists of unreserved and reserved characters.

Unreserved characters: These are not reserved and can be used freely in URIs and URLs. They include all upper case and lower-case letters, all decimal numbers, hyphens, underscores and periods.

Reserved characters: These serve special purposes within a URL. They include / ; : ? @ & , + \$ = .

Reserved characters are used for delimiting or other special purposes inside a URL and cannot be used in any other way within a URL unless they are URL-encoded. This means that in order to display a "?" as a question mark or a "+" as a plus sign inside a URL string, these characters will require encoding.

NB: Space is not allowed in a URL.

USES OF URL

- i. The information contained in a URL allows users to move from one web page to another with just a click.
- ii. URL is used to uniquely identify specific file.

HTTP (Hypertext Transfer Protocol)

Hypertext Transfer Protocol provides a network protocol standard that web browsers and servers use to communicate. You see HTTP when you visit a website because the protocol appears in the URL (for example, <http://www.lifewire.com>).

Because browsers communicate using HTTP, you can drop the protocol from a URL when typing it in the browser's address bar.

Definition: HTTP (Hypertext Transfer Protocol) is an internet protocol that is part of the Internet protocol suite that defines or specifies commands and services used for transmitting webpage data.

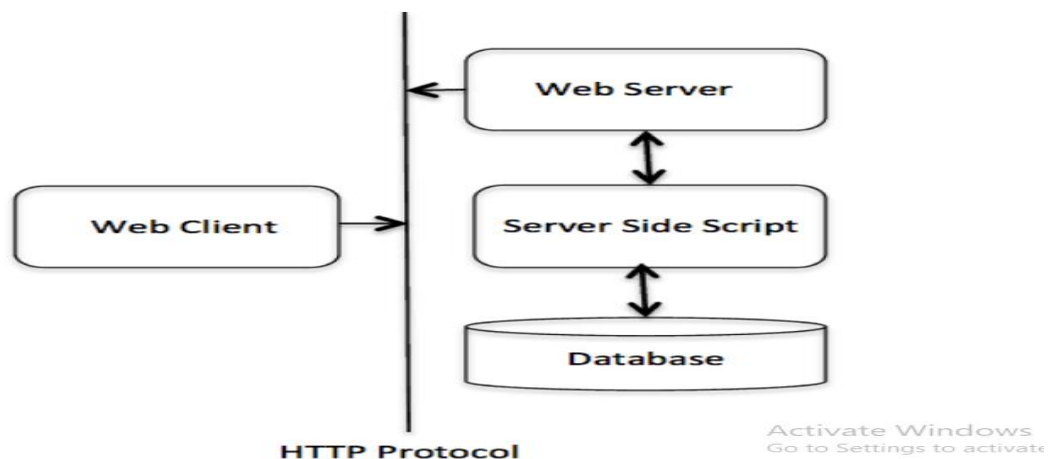
FEATURES OF HTTP

There are three basic features of HTTP they include:

- i. HTTP is connectionless: The client (browser) and server only get connected during current request and response. A connection is only established when there is communication between a client and server.
- ii. HTTP is media independent: meaning any type of media data (text, image, audio, video etc.) can be sent by HTTP.
- iii. HTTP is stateless: meaning neither the client nor the browser can retain information between different requests across the web pages. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other.

HTTP ARCHITECTURE OR MODEL

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

HISTORY OF HTTP

Tim Berners-Lee created the initial HTTP standard in the early 1990s as part of his work in defining the original World Wide Web. Three primary versions were deployed during the 1990s:

- i. HTTP 0.9: Support of basic hypertext documents.
- ii. HTTP 1.0: Extensions to support rich websites.
- iii. HTTP 1.1: Developed to address performance limitations of HTTP 1.0, specified in Internet RFC 2068.

The latest version, HTTP 2.0, became an approved standard in 2015. It maintains backward compatibility with HTTP 1.1 but offers additional performance enhancements.

While standard HTTP does not encrypt traffic sent over a network, the HTTPS standard adds encryption to HTTP through the use of Secure Sockets Layer or, later, Transport Layer Security.

HTTP MESSAGE TYPES

The three HTTP message types are:

- i. HTTP GET: Messages sent to a server contain only a URL. Zero or more optional data parameters may be appended to the end of the URL. The server processes the optional data portion of the URL, if present, and returns the result (a web page or element of a web page) to the browser.
- ii. HTTP POST: Messages place any optional data parameters in the body of the request message rather than adding them to the end of the URL.

iii. HTTP HEAD: Requests work the same as GET requests. Instead of replying with the full contents of the URL, the server sends back only the header information (contained inside the HTML section).

TROUBLESHOOTING HTTP

Messages transmitted over HTTP may fail for several reasons:

- i. User error.
- ii. Malfunction of the web browser or web server.
- iii. Errors in the creation of web pages.
- iv. Temporary network glitches.

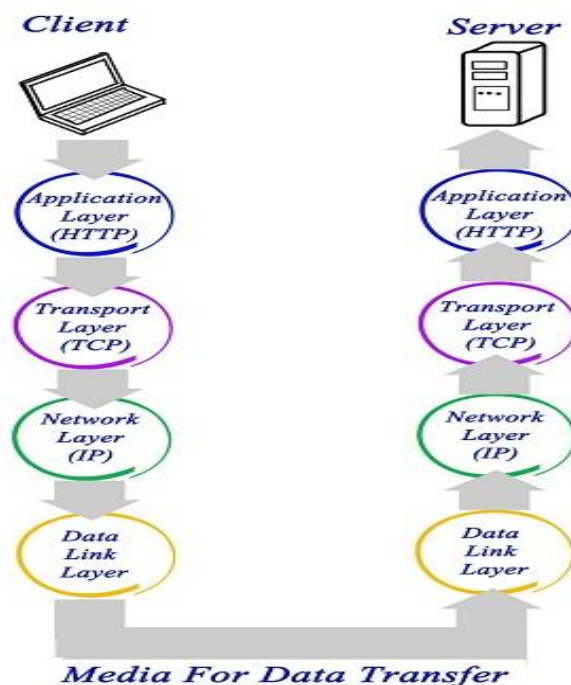
When these failures occur, the protocol captures the cause of the failure and reports an error code to the browser called an HTTP status line/code. Errors begin with a certain number to indicate what kind of error it is.

For example, errors with a failure code beginning with a four indicate that the request for the page cannot be completed properly, or that the request contains the incorrect syntax. As an example, 404 errors mean that a web page cannot be found.

TCP/IP

The key concept in TCP/IP is that every computer has to know or can figure out where all other computers are on the network, and can send data by the quickest route, even if part of the route is down.

DEFINITION: TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of layered protocols used for communication over the Internet. The communication model of this suite is client-server model. A computer that sends a request is the client and a computer to which the request is sent is the server.



TCP/IP LAYERS

TCP/IP has four layers namely:

- i. Application layer – Application layer protocols like HTTP and FTP are used.
- ii. Transport layer – Data is transmitted in form of datagrams using the Transmission Control Protocol (TCP). TCP is responsible for breaking up data at the client side and then reassembling it on the server side.
- iii. Network layer – Network layer connection is established using Internet Protocol (IP) at the network layer. Every machine connected to the Internet is assigned an address called IP address by the protocol to easily identify source and destination machines.
- iv. Data link layer – Actual data transmission in bits occurs at the data link layer using the destination address provided by network layer.

TCP/IP transfers information in small chunks called packets (small amount of data sent over a network).

Each packet includes the following information:

- i. The computer (or last few computers) the data came from
- ii. The computer to which it is headed
- iii. The data itself
- iv. The error-checking information (to ensure that the individual packet was accurately and completely sent and received).

EXERCISES

1. What is the full meaning of URL.
2. Define URL.
3. Explain the history of URL.
4. Draw and explain the working of the URL.
5. List and explain all the components of URL.
6. Identify the components of each of the following URL.

I. [Http://www.test101.com/docs/index.html?bookid=1234&author=tan+ah+tech](http://www.test101.com/docs/index.html?bookid=1234&author=tan+ah+tech)

Ii. [Https://www.seobility.net/en/wiki/url](https://www.seobility.net/en/wiki/url)

Iii. [Https://blog.cloudflare.com/the-history-of-the-url/](https://blog.cloudflare.com/the-history-of-the-url/)

Iv. [Https://www.example.com/category-a/subcategory-a1?searchterm=model+123](https://www.example.com/category-a/subcategory-a1?searchterm=model+123)

V. [Https://www.geeksforgeeks.org/computer-science-projects/](https://www.geeksforgeeks.org/computer-science-projects/)

Vi. [Https://www.computerhope.com/](https://www.computerhope.com/)

Vii. [Http://www.test101.com/index.html](http://www.test101.com/index.html),

Viii. [Http://www.abc.com/smith/home.html](http://www.abc.com/smith/home.html)

7. Explain and give 5 examples of characters allowed in URLs.
8. Explain and give an examples of characters not allowed in URLs.
9. List all the uses of URL.
10. What is the full meaning of HTTP.
11. Define HTTP.
12. List and explain all the features of HTTP.
13. Explain HTTP architecture or model.
14. Draw and explain HTTP architecture or model.
15. Explain the history of HTTP.
16. List and explain HTTP message types.
17. List 4 reason HTTP message may fail.
18. What is the full meaning of TCP/IP.
19. Define TCP/IP.
20. List and explain all the layers of TCP/IP.
21. List and explain 4 packet information in TCP/IP.

CHAPTER FOUR

HTML

INTRODUCTION TO HTML

HTML is a plain-text, human-readable language that is used for creating the structure of web pages. It is used to aid in the publication of web pages by providing a structure that defines elements like tables, forms, lists and headings, and identifies where different portions of our content begin and end. It can be used to embed other file formats like videos, audio files, documents like PDFs and spreadsheets, among others.

DEFINITION HTML (Hyper Text Markup Language) is a language that is used for creating the structure and content of web pages and web applications.

Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage.

Markup Language: a markup language is a computer language that is used to apply layout and formatting to a text document.

Web Page: A web page is a document that is written in HTML and translated by a web browser.

HTML VERSIONS

Since the time HTML was invented there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

HTML 1.0: The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in 1991.

HTML 2.0: This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

HTML 3.2: HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a

web page with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

HTML 4.01: HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

HTML5: HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG (Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

FEATURES OF HTML

- a. It is a very **easy and simple language**. It can be easily understood and modified.
- b. HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.
- c. It is a **markup language**, so it provides a flexible way to design web pages along with the text.
- d. It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.
- e. It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- f. It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.
- g. It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

HTML TEXT EDITORS

Text editors are the programs which allow editing in a written text, hence to create a web page we need to write our code in some text editor.

There are various types of text editors available which you can directly download, but for a beginner, the best text editor is Notepad (Windows) or TextEdit (Mac).

Example of text editors

- a. Notepad.**
- b. Notepad++**
- c. Sublime Text**

To save a HTML file, a file is saved with a .html or .htm extension.

HTML file name syntax: filename.html or filename.htm

Example

- 1. FirstFileDemo.html
- 2. HomePage.htm

WEB BROWSERS

A web browser is a program or software that is used to read HTML documents and display them correctly. A browser does not display the HTML tags, but uses them to determine how to display the document as seen in the above example.

Examples

- i. Internet Explorer.
- ii. Google Chrome.
- iii. Mozilla Firefox.
- iv. Edge.
- v. Safari etc.

HTML BASIC STRUCTURE

A simple html file has the following basic structure:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    ...HTML Code...
    ...HTML Code...
    .
    .
    .
    ...HTML Code...
  </body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>The basic building blocks of HTML</title>
  </head>
  <body>
    <h2>The building blocks</h2>
    <p>This is a paragraph tag</p>
    <p style="color: red">The style is attribute of paragraph tag</p>
    <span>The element contains tag, attribute and content</span>
  </body>
</html>
```

OUTPUT

The building blocks

This is a paragraph tag

The style is attribute of paragraph tag

The element contains tag, attribute and content

EXPLANATION

The `<!DOCTYPE html>` The `<!DOCTYPE>` NOT actually a tag, rather a declaration to the web browser of what version of html the following document uses For HTML5, doctype declaration is `<!DOCTYPE html>` This declaration must be the very first line of an html document

The `<html>` element is the root element of an HTML page because it can be considered the root of the tree-like structure of elements in an HTML page

The `<head>` Contains information about the document, not content. Common elements included within `<head></head>`:

- i. <title> - contains page title, displayed in browser's title bar
- ii. <link> used to add CSS stylesheets and icons to page
- iii. <meta> used to specify metadata like page descriptions and keywords
- iv. <script> used to add JavaScript code to the page

```
<!DOCTYPE>
<html>
<head>
  <title>My Web Page 2.0</title>
  <link rel="stylesheet" type="text/css" href="/style.css">
  <meta name="description" content="Learning about HTML.">
  <meta name="keywords" content="html, web development">
  <script src="code.js"></script>
</head>
</html>
```

The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc. Appears directly beneath the head element.

The <h2> used for next important sub header while h1 used for most important titles.

The <p> element defines a paragraph

Note: Some HTML elements have no content (like the
 element). These elements are called empty elements. Empty elements do not have an end tag!

HTML DOCUMENT

An HTML document consist of its basic building blocks which are:

i. Tags: An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.

ii. Attribute: An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax:

`<tag name attribute_name= " attribute_value"> content </ tag name>`

- i. name – usually specifies the property of the element for which additional information is being provided
- ii. value –this is selected from set of possible values for given property

iii. Elements: An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.

HTML HEADINGS

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the heading with the largest size.

<h6> defines heading with the smallest size.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Heading Demo</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>This is heading 1</h1>
```

```
    <h2>This is heading 2</h2>
```

```
    <h3>This is heading 3</h3>
```

```
    <h4>This is heading 4</h4>
```

```
    <h5>This is heading 5</h5>
```

```
    <h6>This is heading 6</h6>
```

```
  </body>
```

```
</html>
```


HTML PARAGRAPHS

HTML paragraphs are defined with the <p> tag: Appears anywhere within the body to represent a paragraph of text expressing a single thought. Usually displayed with vertical space before and after paragraph.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Paragraph Demo</title>
```

```
  </head>
```

```
  <body>
```

```
    <p> The Internet started as a project of the United States Government's Department of  
    Defense (DOD), to create a non-centralized network. This project was called ARPANET  
    (Advanced Research Projects Agency Network), created by the Pentagon's Advanced  
    Research Projects Agency established in 1969 to provide a secure and survivable  
    communications network for organizations engaged in defense related research.
```

```
  </p>
```

```
  <p> A web browser is a program or software that is used to read HTML  
  documents and display them correctly. A browser does not display the HTML  
  tags, but uses them to determine how to display the document as seen in the above  
  example.
```

```
  </p>
```

```
  </body>
```

```
</html>
```

HTML **** and **<i>** Tags

These are text formatting tags.

**** indicates that the text should be **bold**

<i> indicates that the text should be *italicized*

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Text Formatting Demo</title>
```

```
  </head>
```

```
    Once upon a time, there lived an <i>ogre</i>. His name was <b>Shrek</b>.
```

```
    He saved a princess named <b><i>Fiona</i></b>. They married and lived  
    happily ever after.
```

```
  <body>
```

```
  </body>
```

```
</html>
```

HTML <hr> and

These tags are used to mark a break in content.

<hr> Is used to insert a visible line between preceding and subsequent content.

 Is used to insert a blank invisible line between preceding and subsequent content.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Line Breaks Demo</title>
  </head>
  <body>
    <p> This is a paragraph about
    how your browser works. </p>
    <hr>
    <p> This is a paragraph about
    how to set up your environment
    to begin creating HTML
    documents. </p>
    <br>
    <p> This is a paragraph about
    important HTML tags that
    you should know. </p>
  </body>
</html>
```

OUTPUT

This is a paragraph about how your browser works.

This is a paragraph about how to set up your environment to begin creating HTML documents.

This is a paragraph about important HTML tags that you should know.

HTML LINKS

A link is a connection from one web resource (page) to another. HTML links are defined with the <a> tag.

Syntax

```
<a href="locationName.html">Link Text</a>
```

Example

LinkDemo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Link Demo</title>
  </head>
  <body bgcolor="pink">
    <a href="biodata.html">Bio</a>
  </body>
</html>
```

BioData.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Biodata</title>
  </head>
  <body>
    <p>
      Name: Musa Isa
      D.O.B: 23/04/1997
      Phone: 09034568219
      Email: coolmusa@yahoo.com
      Address: No. 7 Nijar Layi Jos
    </p>
  </body>
</html>
```

HTML IMAGE

This is a tag that is used to display image on the web page. It uses tag. HTML tag is an empty tag that contains attributes only, closing tags are not used in HTML image element. The tag creates a holding space for the referenced image.

Syntax

```

```

Attributes of HTML img tag

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below:

- i. **src:** It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server. The location of image may be on the same directory or another server that is relative or absolute path.
- ii. **alt:** The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describes the image in words.
- iii. **width:** It is an optional attribute which is used to specify the width in pixel (e.g. 50px) to display the image.
- iv. **height:** It is also an optional attribute which is used to specify the height to display the image. The HTML height attribute also supports iframe and object elements.
- v. **title:** tooltip text displayed when mouse hovers over image

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <h2>Image Example</h2>
    <img src = "flower.jpg" alt="Flowers in China" width="300" height="250">
  </body>
</html>
```

OUTPUT

Image Example



IMAGE PATHS

HTML image paths are used to define the image location to be used as an external resource within the HTML document. The location of the file is specified as per the structure of the web folder.

ABSOLUTE IMAGE PATH

An absolute image path is the full URL to an image.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <h2>Image Example</h2>
    
  </body>
</html>
```

RELATIVE IMAGE PATHS

A relative image path points to an image relative to the current page.

EXAMPLE

a. file in the same folder as the html file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <h2>Image Example</h2>
    
  </body>
</html>
```

NB: The image file called *picture.jpg* is in the same folder as the html file.

b. file in a different folder as the html file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <h2>Image Example</h2>
    
  </body>
</html>
```

NB: the image *river.jpg* is in a folder *images*.

c. Folder at the root folder.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <h2>Image Example</h2>
    
  </body>
</html>
```

NB: The root folder (C:\ drive) has a folder *images* consisting of an image 'river.jpg'.

HTML TABLES

This is used to create a table and populated it with data. It is use to arrange data into rows and columns. The <table> tag is used.

HTML TABLE TAGS

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table

Examples

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Demo</title>
  </head>
  <body>
    <table>
      <tr>
        <th>Name:</th>
        <td>John Carter</td>
      </tr>
      <tr>
        <th rowspan="2">Phone:</th>
        <td>55577854</td>
      </tr>
      <tr>
        <td>55577855</td>
      </tr>
    </table>
  </body>
</html>
```

OUTPUT

Name	Age	Gender
Musa	20	Male
Rahama	19	Female

SPANNING MULTIPLE ROWS AND COLUMNS

Spanning allow you to extend table rows and columns across multiple other rows and columns.

Normally, a table cell cannot pass over into the space below or above another table cell. But, you can use the *rowspan* or *colspan* attributes to span multiple rows or columns in a table.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Table Row Demo</title>
  </head>
  <body>
    <table border="2">
      <tr>
        <th>Name:</th>
        <td>Khalid</td>
      </tr>
      <tr>
        <th rowspan="2">Phone:</th>
        <td>08012345687</td>
      </tr>
      <tr>
        <td>+2348062783681</td>
      </tr>
    </table>
  </body>
</html>
```

Name:	Khalid
Phone:	08012345687
	+2348062783681

```
<!DOCTYPE html>
<html>
  <head>
    <title>Table Row Demo</title>
  </head>
  <body>
    <table border="2">
      <tr>
        <th>Name:</th>
        <th colspan="2">Phone:</th>
      </tr>
      <tr>
        <td>Khalid</td>
        <td>08012345687</td>
        <td>+2348062783681</td>
      </tr>
    </table>
  </body>
</html>
```

OUTPUT

Name:	Phone:	
Khalid	08012345687	<u>+2348062783681</u>

HTML FORMS

Forms are used to send and collect different kinds of user data such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for further processing.

An HTML form facilitates the user to enter data that is to be sent to the server for processing.

The `<form>` tag is used to create an HTML form.

HTML Form Syntax

```
<form action="url" method="get|post">  
  //form contents  
</form>
```

The HTML `<form>` element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

THE ACTION ATTRIBUTE

The *action* attribute defines the action to be performed when the form is submitted. Usually, the form data is sent to a file on the server when the user clicks on the submit button.

THE METHOD ATTRIBUTE

The *method* attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with `method="get"`) or as HTTP post transaction (with `method="post"`).

The default HTTP method when submitting form data is GET.

DIFFERENCE BETWEEN POST AND GET

1. Post method is more secure.
2. Post method sent data to http response (body).
3. Post method sent more data.
4. Post method is not good for search but it is good for hidden data.

1. Get method is less secure.
2. Get method sent data to page address.
3. Get method sent less data.
4. Get method is good for search content in order to let users to used it again.

Tip: Always use POST if the form data contains sensitive or personal information!

HTML FORM TAGS

Some HTML 5 form tags.

Tags	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<select>	It defines a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML <input> element

The HTML <input> element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input field to gather different information from user.

Following is the example to show the simple text input.

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

Text Fields

The <input type="text"> defines a single-line input field for text input.

A form with input fields for text:

```
<form>
  <label>First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label>Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

First name:

Last name:

The <label> tag defines a label for many form elements.

The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

Therefore, attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

Example

```
<form>
  <input type="radio" id="male" name="gender" value="male">
  <label>Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label>Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label>Other</label>
</form>
```

☐ Male
☐ Female
☐ Other

Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label> I have a boat</label>
</form>
```

☐ I have a bike
☐ I have a car
☐ I have a boat

The <select> Element

The <select> element defines a drop-down list:

```
<label>Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Choose a car:

The <option> elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The cat was playing in the garden.

Input Type Password

<input type="password"> defines a **password field**:

Example

```
<form>
  <label>Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label>Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

OUTPUT

Username:

Password:

SUBMIT BUTTONS

A submit button is used to send the form data to a web server. When submit button is clicked the form data is sent to the file called *form handler* which is specified in the form's `action` attribute to process the submitted data.

Example

```
<form action="action_page.php" method="get" >
<label>First name:</label><br>
<input type="text" id="fname" name="fname" value="John"><br>
<label>Last name:</label><br>
<input type="text" id="lname" name="lname" value="Doe"><br><br>
<input type="submit" value="Submit">
</form>
```

OUTPUT

First name:

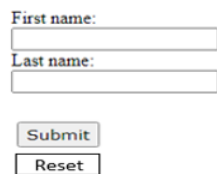
Last name:

THE RESET BUTTON

The reset button is used to delete (resets) all the forms data to default values.

```
<form action="action.php" method="post">
  <label>First Name:</label>
  <input type="text" name="first-name" id="first-name">
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
```

OUTPUT



First name:

Last name:

Example

Write the html code that displays the form below:

Username:

admin

Password:

.....

First name:

Last name:

Gender:

☐ Male

☐ Female

☐ Other

Properties:

☐ I have a bike

☐ I have a car

☐ I have a boat

Car Type: (Volvo, Audi, Mazda, Toyota)

Volvo ▾

Submit

Reset

Solution

```
<html>
  <head>
    <title>My Web Page</title>
  </head>
  <body>
    <h1>Internet Technology Practicals</h1>
    <form >
      <label>Username:</label><br>
      <input type="text" name="username"><br>
      <label>Password:</label><br>
      <input type="password" name="password" ><br>
      <label>First name:</label><br>
      <input type="text" name="fName"><br>
      <label>Last name:</label><br>
      <input type="text" name="lName"><br>
      <label> Gender:</label> <br>
      <input type="radio" name="gender" value="male">
      <label>Male</label><br>
      <input type="radio" name="gender" value="female">
      <label>Female</label><br>
      <input type="radio" name="gender" value="other">
      <label>Other</label><br>
      <label>Properties:</label> <br>
      <input type="checkbox" name="vehicle1" value="Bike">
      <label> I have a bike</label><br>
      <input type="checkbox" name="vehicle2" value="Car">
      <label> I have a car</label><br>
      <input type="checkbox" name="vehicle3" value="Boat">
      <label> I have a boat</label><br><br>
      <label>Car Type (Volvo, Audi, Mazda, Toyota): </label><br>
      <select name="cars">
        <option>Volvo</option>
        <option>Audi</option>
        <option>Mazda</option>
        <option>Toyota</option>
      </select><br>
      <input type="submit" name="submit" value="Submit">
      <input type="reset" name="reset" value="Reset">
    </form>
  </body>
</html>
```

HTML IFRAMES

An iframe or inline frame is used to display external objects including other web pages within a web page. An iframe pretty much acts like a mini web browser within a web browser. The content inside an iframe exists entirely independent from the surrounding elements.

HTML Iframe Syntax

The HTML `<iframe>` tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

Syntax

```
<iframe src="url" title="description">
```

```
</iframe>
```

The ***url*** specified in the ***src*** attribute points to the location of an external object or a web page.

The ***title*** attribute is used by screen readers to read out what the content of the iframe is.

IFRAME HEIGHT AND WIDTH

The ***height*** and ***width*** attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

Example

IFrameDemo.html

```
<html>
  <head>
    <title>My Web Page</title>
  </head>
  <body>
    <iframe src = "FirstIFrameDemo.html" width="300" height="200" title="First Frame">
    </iframe>
    <iframe src = " SecondIFrameDemo.html" width="300" height="200" title="Second
    Frame">
    </iframe>
  </body>
</html>
```

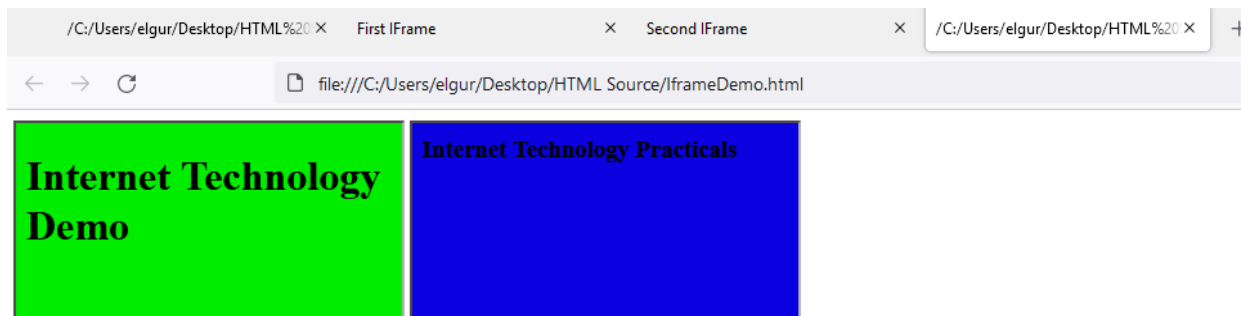
FirstIFrameDemo.html

```
<html>
  <head>
    <title>First IFrame</title>
  </head>
  <body bgcolor="green">
    <h1>Internet Technology Demo</h1>
  </body>
</html>
```

SecondIFrameDemo.html

```
<html>
  <head>
    <title>Second IFrame</title>
  </head>
  <body bgcolor="blue">
    <h1>Internet Technology Practicals</h1>
  </body>
</html>
```

OUTPUT



EXERCISES

1. Define html.
2. Write the full meaning of the acronym html.
3. List and explain all the html versions
4. List and explain all the features of html
5. Define html text editors.
6. List any two html text editors.
7. Define web browser.
8. List 5 examples of web browser.
9. Write the syntax of a basic Html file.
10. List and explain the basic blocks of a HTML document.
11. Write a HTML code demonstrating all the HTML headings.
12. Write a HTML code demonstrating HTML Paragraph.
13. Write a HTML code demonstrating HTML **** and *<i>* tags.
14. Write a HTML code demonstrating HTML **<hr>** and **
** tags.
15. Write the syntax of a HTML link.
16. Write a HTML code demonstrating HTML link.
17. Write the syntax of a HTML image.
18. Write a HTML code demonstrating HTML image
19. Explain image path.
20. Write HTML codes of the different image paths.
21. Explain the HTML table tags in a Tabular form.
22. Write a HTML code to display the following output:
 - i.

Name	Age	Gender
Musa	20	Male
Rahama	19	Female

- ii.

Name:	Khalid
Phone:	08012345687 +2348062783681

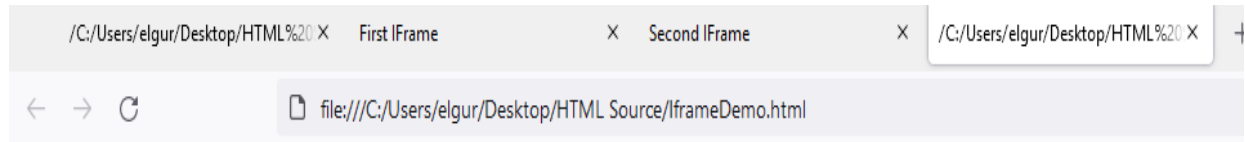
iii.

Name:	Phone:	
Khalid	08012345687	+2348062783681

23. Explain HTML forms.
24. Write and explain the HTML Form Syntax.
25. Write difference between post and get methods.
26. Explain the HTML form tags in a Tabular form.
27. Write the html code that displays the form below:

<p>Username: <input type="text" value="admin"/></p> <p>Password: <input type="password" value="....."/></p> <p>First name: <input type="text"/></p> <p>Last name: <input type="text"/></p> <p>Gender: <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other</p> <p>Properties: <input type="checkbox"/> I have a bike <input type="checkbox"/> I have a car <input type="checkbox"/> I have a boat</p> <p>Car Type: (Volvo, Audi, Mazda, Toyota) <input type="button" value="Volvo"/></p> <p><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>
--

28. Explain HTML Iframe.
29. Written the syntax of HTML Iframe.
30. Write a HTML code to display the following Iframe:



**Internet Technology
Demo**

Internet Technology Practicals

CHAPTER FIVE

CASCADING STYLE SHEET

INTRODUCTION TO CSS

CSS It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. Hence CSS describe the presentation of documents, define sizes, spacing, fonts, colors, layout, etc.

CSS uses selectors to select certain parts of an HTML document in order to apply new styles to them. You can select all elements of the same type, or you can select specific elements by their attributes, even creating custom attributes to fine-tune which elements you'd like to apply styles to.

The browser reads CSS from top to bottom, line by line. That means that if you write a selector once, and then write the same selector later in the CSS document, the later styles will override the earlier ones. This is called the **cascade**. CSS specifications are maintained by the World Wide Web Consortium (W3C)

DEFINITION OF CSS

CSS is a formatting language used to describe the appearance of content in an HTML file.

CSS It is a style sheet language which is used to describe the looks and formatting of a document written in markup language.

VERSIONS OF CSS

i. CSS1

The first CSS specification to become an official W3C Recommendation is CSS level 1, published in December 1996.

ii. CSS2

CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998.

iii. CSS3

Work on CSS level 3 started around the time of publication of the original CSS 2 recommendation. The earliest CSS 3 drafts were published in June 1999. CSS3 is the latest version of the CSS specification. CSS3 adds several new styling features and improvements to enhance the web presentation capabilities.

ADVANTAGES OF USING CSS

1. Separation of style and layout: One of the advantages of CSS is that it allows the separation of style and layout from the content of the document.
2. CSS Save Lots of Time: CSS gives lots of flexibility to set the style properties of an element. You can write CSS once; and then the same code can be applied to the groups of HTML elements, and can also be reused in multiple HTML pages.
3. Easy Maintenance: CSS provides an easy means to update the formatting of the documents, and to maintain the consistency across multiple documents. Because the content of the entire set of web pages can be easily controlled using one or more style sheets.
4. Pages Load Faster: CSS enables multiple pages to share the formatting information, which reduces complexity and repetition in the structural contents of the documents. It significantly reduces the file transfer size, which results in a faster page loading.
5. Superior Styles to HTML: CSS has much wider presentation capabilities than HTML and provide much better control over the layout of your web pages. So you can give far better look to your web pages in comparison to the HTML presentational elements and attributes.
6. Multiple Device Compatibility: CSS also allows web pages to be optimized for more than one type of device or media. Using CSS the same HTML document can be presented in different viewing styles for different rendering devices such as desktop, cell phones, etc.

DISADVANTAGES OF CSS

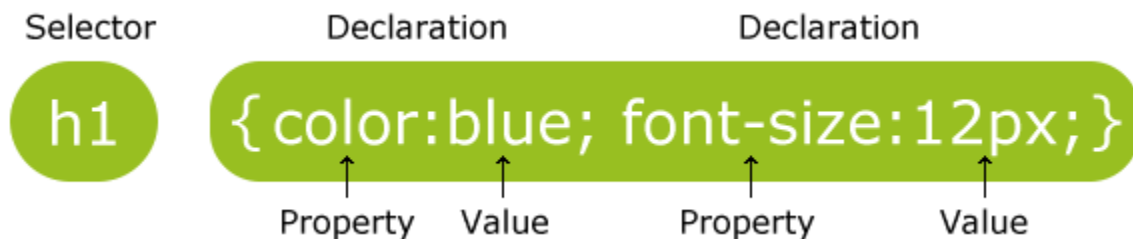
- a. Different browsers may interpret Style Sheets in different ways.
- b. Some styles may not be seen at all on some browsers. I.e. not uniformly supported by all browsers.

SYNTAX OF CSS

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

A style rule is made of three parts:

- a. Selector
- b. Property
- c. Value



i. **Selector:** is an HTML tag at which a style will be applied.

Example

- a. `table {border :1px;}`
- b. `h1 {color: red; font-size: 15px;}`

ii. **Property:** A property is a type of attribute of HTML tag.

- They could be color, border, etc.

iii. **Value:** these are the values assigned to properties. For e.g. color property can have the value either red, blue, green etc.

Declarations: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations: color set to blue and font-size set to 12px. Each declaration contains a property name and value, separated by a colon.

CSS INCLUSION

Including CSS in HTML documents. CSS can either be attached as a separate document or embedded in the HTML document itself. There are three methods of including CSS in an HTML document:

- i. Inline styles — Using the **style** attribute in the HTML start tag.
- ii. Embedded styles — Using the **<style>** element in the head section of a document.
- iii. External style sheets — Using the **<link>** element, pointing to an external CSS file.

1. INLINE STYLES

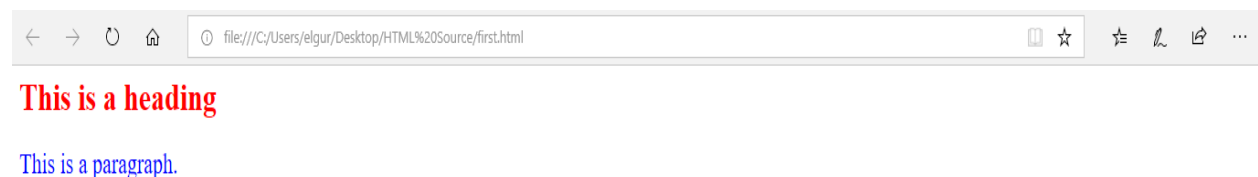
Inline styles are used to apply the unique style rules to an element by putting the CSS rules directly into the start tag. It can be attached to an element using the style attribute.

The style attribute includes a series of CSS property and value pairs. Each "property: value" pair is separated by a semicolon (;). But it needs to be all in one line i.e. no line break after the semicolon.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1 style="color : red; font-size:30px;">This is a heading</h1>
    <p style="color : blue; font-size:22px;">This is a paragraph.</p>
  </body>
</html>
```

OUTPUT



NB: Using the inline styles are generally considered as a bad practice. As style rules are embedded directly inside the HTML tag, it causes the presentation to become mixed with the content of the document; which makes the code hard to maintain and negates the purpose of using CSS.

ADVANTAGES OF INLINE CSS

- a. Easy to use.
- b. Good for quick, one-off situations.

DISADVANTAGES OF INLINE CSS

- a. Mixing content and presentation: this should be avoided.
- b. Hard to manage for large, complex pages.

2. EMBEDDED OR INTERNAL STYLE

Embedded or internal style sheets only affect the document they are embedded in.

Embedded style sheets are defined in the `<head>` section of an HTML document using the `<style>` element. You can define any number of `<style>` elements in an HTML document but they must appear between the `<head>` and `</head>` tags.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>My HTML Document</title>
    <style>
      body { background-color: YellowGreen; }
      p { color: blue; }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </body>
</html>
```

OUTPUT



ADVANTAGES OF INTERNAL CSS

- a. Separates content and presentation.
- b. Easy to use.

In particular, if you want to apply the same style to all tags of a certain type (say <p>) it's easier to do it with internal CSS than inline CSS

- c. Good if you only have a limited amount of CSS in the page

DISADVANTAGES OF INTERNAL CSS

- a. Hard to manage for large, complex pages
- b.** Cannot reuse across multiple pages

3. EXTERNAL CSS

An external style sheet is ideal when the style is applied to many pages of the website.

An external style sheet holds all the style rules in a separate document that you can link from any HTML file on your site. External style sheets are the most flexible because with an external style sheet, you can change the look of an entire website by changing just one file.

LINKING EXTERNAL STYLE SHEETS

An external style sheet can be linked to an HTML document using the <link> tag. The <link> tag goes inside the <head> section.

Before linking, we need to create a style sheet first.

SYNTAX

```
<link rel="stylesheet" href="filename.css">
```

EXAMPLE

style.css

```
body {  
  
    background: lightyellow;  
  
    font: 18px Arial, sans-serif;  
  
}  
  
h1 {  
  
    color: orange;  
  
}
```

cssDemo.html

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>My HTML Document</title>  
        <link rel="stylesheet" href="style.css">  
    </head>  
    <body>  
        <h1>This is a heading</h1>  
        <p>This is a paragraph of text.</p>  
    </body>  
</html>
```



ADVANTAGES OF EXTERNAL CSS

- a. Separates content and presentation
- b. Can include many different CSS pages with multiple tags

DISADVANTAGES OF EXTERNAL CSS

- a. Lots of different files to manage (which might be overkill for smaller pages)

CSS SELECTORS

Selectors is a pattern that is to target specific elements and apply some style on them. The style rules associated with that selector is applied to the elements that match the selector pattern.

Type of Selector	What's in the CSS File?	What does this selector apply to?	What does the HTML file contain?
Element Selector	<code>h1 { color:red; }</code>	All <code><h1></code> elements	<code><h1> ... </h1></code>
Class selector	<code>.address { ... }</code>	All elements in class "address"	<code><div class="address"> ... </div></code>
Id Selector	<code>#section1 { ... }</code>	Unique element with ID "section1"	<code><p id="section1"> ... </p></code>

Universal Selector

The style rules for Universal Selector is applied to every element in a document.

The universal selector may be omitted if other conditions exist on the element. This selector is often used to remove the default margins and paddings from the elements for quick testing purpose.

EXAMPLE

```
* {  
  margin: 0;  
  padding: 0;  
}
```

Element Selectors

An element type is applied on all instance of the element in the document with the corresponding element type name.

EXAMPLE

```
p {  
  color: blue;  
}
```

The style rules inside the `p` selector will be applied on every `<p>` element (or paragraph) in the document and color it blue, regardless of their position in the document tree.

Id Selectors

The id selector is used to define style rules for a *single* or *unique* element.

The id selector is defined with a hash sign (`#`) immediately followed by the id value.

EXAMPLE

```
#error {  
  color: red;  
}
```

This style rule renders the text of an element in red, whose `id` attribute is set to `error`.

Class Selectors

The class selectors can be used to select any HTML element that has a `class` attribute. All the elements having that class will be formatted according to the defined rule.

The class selector is defined with a period sign (`.`) immediately followed by the class value.

EXAMPLE

```
.blue {  
  color: blue;  
}
```

The above style rules renders the text in blue of every element in the document that has `class` attribute set to `blue`.

EXAMPLE

```
p.blue {  
  color: blue;  
}
```

The style rule inside the selector `p.blue` renders the text in blue of only those `<p>` elements that has `class` attribute set to `blue`, and has no effect on other paragraphs.

Descendant Selectors

You can use these selectors when you need to select an element that is the descendant of another element, for example, if you want to target only those anchors that are contained within an unordered list, rather than targeting all anchor elements.

EXAMPLE

```
ul.menu li a {  
    text-decoration: none;  
}  
h1 em {  
    color: green;  
}
```

The style rules inside the selector `ul.menu li a` applied to only those `<a>` elements that contained inside an `` element having the class `.menu`, and has no effect on other links inside the document.

Similarly, the style rules inside the `h1 em` selector will be applied to only those `` elements that contained inside the `<h1>` element and has not effect on other `` elements.

Child Selectors

A child selector is used to select only those elements that are the direct children of some element.

A child selector is made up of two or more selectors separated by a greater than symbol (`>`). You can use this selector, for instance, to select the first level of list elements inside a nested list that has more than one level. Let's check out an example to understand how it works:

EXAMPLE

```
ul > li {  
    list-style: square;  
}  
ul > li ol {  
    list-style: none;  
}
```

The style rule inside the selector `ul > li` applied to only those `` elements that are direct children of the `` elements, and has no effect on other list elements.

Grouping Selectors

Often several selectors in a style sheet share the same style rules declarations. You can group them into a comma-separated list to minimize the code in your style sheet. It also prevents you from repeating the same style rules over and over again.

EXAMPLE

```
h1 {  
    font-size: 36px;  
    font-weight: normal;
```

```
}  
h2 {  
    font-size: 28px;  
    font-weight: normal;  
}  
h3 {  
    font-size: 22px;  
    font-weight: normal;  
}
```

As you can see in the above example, the same style rule `font-weight: normal;` is shared by the selectors `h1`, `h2` and `h3`, so it can be grouped in a comma-separated list, like this:

```
h1, h2, h3 {  
    font-weight: normal;  
}  
h1 {  
    font-size: 36px;  
}  
h2 {  
    font-size: 28px;  
}  
h3 {  
    font-size: 22px;  
}
```


SELECTORS EXAMPLE

selector.css

```
/*Universal selector*/
```

```
* {  
    margin: 0;  
}
```

```
/*Element selector*/
```

```
h1 {  
    color:red;  
}
```

```
/*class selector*/
```

```
.address {  
    font-size:18px;  
    font-style:italic;  
}
```

```
/*id selector*/
```

```
#section1 {  
    Color:blue;  
}
```

```
/*grouping selector*/
```

```
h4, h5, h6 {  
    font-family:algerian;  
}
```

SelectorDemo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selector Demo</title>
    <link rel="stylesheet" href="selector.css">
  </head>
  <body>
    <h1>This is a heading</h1><br><hr>

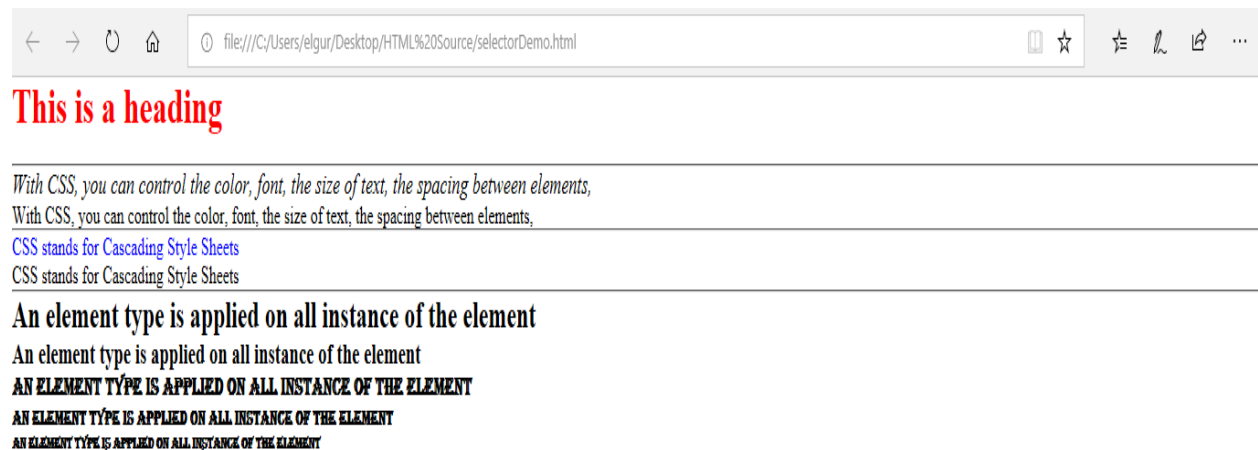
    <div class="address">  With CSS, you can control the color,
                          font, the size of text, the spacing between elements,<br>
    </div>
    <div>  With CSS, you can control the color,
          font, the size of text, the spacing between elements,<br><hr>
    </div>

    <p id="section1">CSS stands for Cascading Style Sheets</p>
    <p>CSS stands for Cascading Style Sheets<br><hr></p>

    <h2>An element type is applied on all instance of the element</h2>
    <h3>An element type is applied on all instance of the element</h3>
    <h4>An element type is applied on all instance of the element</h4>
    <h5>An element type is applied on all instance of the element</h5>
    <h6>An element type is applied on all instance of the element</h6>

  </body>
</html>
```

OUTPUT



CASCADING OR OVERRIDING ORDER

What style will be used when there is more than one style specified for an HTML element? Generally, all the styles will "cascade" into a new "virtual" style sheet.

Here are the rules for cascading or overriding Style Sheet.

- a. Any inline style sheet takes the highest priority. So, it will override any rule defined in `<style>...</style>` tags or the rules defined in any external style sheet file.
- b. Any rule defined in `<style>...</style>` tags will override the rules defined in any external style sheet file.
- c. Any rule defined in the external style sheet file takes the lowest priority, and the rules defined in this file will be applied only when the above two rules are not applicable.

CSS FONT

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more.

These are some important font attributes:

- i. **CSS Font color:** This property is used to change the color of the text. (Standalone attribute). There are three different format to define a color: By a color name (`h1 {color: red;}`), By a hexadecimal value (`h2{color: #9000A1;}`) and RGB (`p{color: rgb(0, 220, 98);}`)
- ii. **CSS Font family:** This property is used to change the face of the font. And is divided into two types: Generic family it includes Serif, Sans-serif, and Monospace. Font family it specifies the font family name like Arial, Verdana etc. Example `p {font-family: Arial;}`
- iii. **CSS Font size:** This property is used to increase or decrease the size of the font. Example `p {font-size: 18px;}`.
- iv. **CSS Font style:** This property is used to make the font bold, italic or oblique. Example `h2 {font-style: italic;}`.
- v. **CSS Font variant:** This property creates a small-caps effect. Example `p {font-variant: small-caps;}`.

- vi. **CSS Font weight:** This property is used to increase or decrease the boldness and lightness of the font. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200... up to 900). Example `p{font-weight: bold;}`

TEXT DECORATION

The text-decoration property is used to add or remove an underline. Usually you do not want to add an underline to text unless it is a link. But you may want to remove the underline from a link, which is a popular style. Example: `{text-decoration: none;}`. To underline an element after you define the selector, put the property as text-decoration and set the value to underline.

CSS BACKGROUND

You can set the following background properties of an element.

- i. **The background-color:** The background-color property specifies the background color of an element. Example: `body{background-color: lightblue;}`
- ii. **The background-image:** specifies an image to use as the background of an element. Example: `body{background-image: url("paper.gif");}`
- iii. **The background repeat:** repeat an image both horizontally and vertically. Example: `body{background-image: url("paper.gif"); background-repeat: repeat-x;}`
- iv. **The background position:** The background position is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage. It has other values such as right, left, center, bottom etc.
- v. **The background attachment:** Is used to control the scrolling of an image in the browser window. You can set it as fixed or scroll.

CSS – MEASUREMENT UNITS

CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. You need these values while specifying various measurements in your Style rules e.g. border="1px solid red".

We have listed out all the CSS Measurement Units along with proper Examples:

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 16pt; line-height: 125%;}
cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}
em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 7em;}
ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {font-size: 24pt; line-height: 3ex;}
in	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 20pc;}

pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

CSS COLORS

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or for the background of the element.

DEFINING COLOR VALUES

Colors in CSS are specified in the following formats:

- i. **Color keyword** e.g. "red", "green", "blue", "transparent" etc.

```
h1 {  
  color: red;  
}  
p {  
  color: purple;  
}
```

- ii. **HEX value** e.g. "#ff0000", "#00ff00" etc.

```
h1 {  
  color: #ffa500;  
}  
p {  
  color: #00ff00;  
}
```

- iii. **RGB value** e.g. "rgb(255, 0, 0)" etc.

```
h1 {  
  color: rgb(255, 165, 0);  
}  
p {  
  color: rgb(0, 255, 0);  
}
```

COLOR

HEX

	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

EXERCISES

1. What is the full meaning of CSS.
2. Define CSS.
3. List all the versions of CSS and their date of release.
4. List and explain the advantages of using CSS.
5. List the disadvantages of CSS.
6. List and explain the syntax of CSS.
7. List and explain CSS inclusion.
8. Write a code to demonstrate inline CSS.
9. Write a code to demonstrate internal CSS.
10. Write a code to demonstrate external CSS.
11. Write the syntax for linking external style sheets.
12. Write a code to demonstrate linking external style sheets.
13. Explain CSS selectors.
14. Explain CSS selectors with example.
15. Write a code to demonstrate the output below using selectors:



This is a heading

With CSS, you can control the color, font, the size of text, the spacing between elements,

With CSS, you can control the color, font, the size of text, the spacing between elements,

CSS stands for Cascading Style Sheets

CSS stands for Cascading Style Sheets

An element type is applied on all instance of the element

An element type is applied on all instance of the element

AN ELEMENT TYPE IS APPLIED ON ALL INSTANCE OF THE ELEMENT

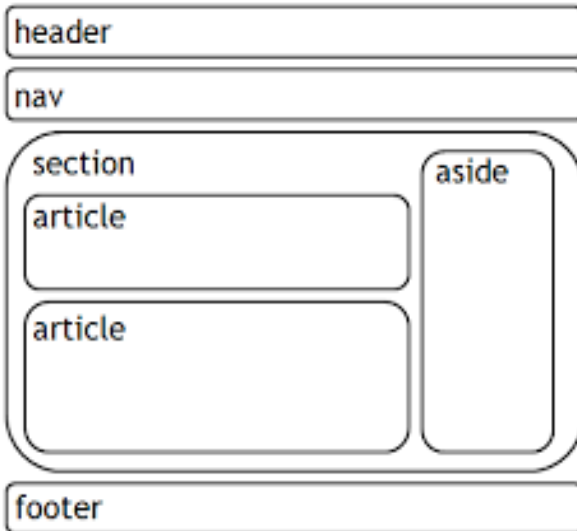
AN ELEMENT TYPE IS APPLIED ON ALL INSTANCE OF THE ELEMENT

AN ELEMENT TYPE IS APPLIED ON ALL INSTANCE OF THE ELEMENT

16. Write the cascading order.
17. Demonstrate cascading order with an example.
18. Explain CSS font with a code example.
19. Explain text decoration with a code example.
20. Explain CSS background with a code example.
21. Explain CSS measurement units with a code example.
22. List and explain ways of defining CSS color values with a code example.

HTML5 STRUCTURAL ELEMENT

HTML 5 has introduced a number of sectioning elements that can be used to structure a given web page.



They include:

1. Header

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

The header is a place to put your site title and navigation:

2. Nav

```
<nav></nav>
```

The nav is the element that contains your site navigation (all of the links that you put in your header, sidebar, or footer to help people get around your site.)

3. Main

```
<main></main>
```

The main is the big container for your main content. You should only use one per HTML document.

4. Article

`<article></article>`

The article element contains an article, or the main information, of your web page. It may be an article, a blog post, an interactive form, or any other type of content. One `<main>` element may contain multiple articles.

5. Section

`<section></section>`

Articles are divided into sections. Each section has a heading `<h1...h6>` etc., which we can defined in the section called Adding Text.

6. Aside

`<aside></aside>`

An aside is a place to put extra information that doesn't go into the main article, but is not header or footer content. This may be a place for an author's bio, extra links, ads, and more. This is placed outside of the `<main>` element.

7. Footer

`<footer></footer>`

The footer is used for a web page's footer content. It may include copyright information, your site navigation, other related links, and more.

EXAMPLE

newhtml.css

```
header{
    width: 100%;
    text-align: center;
    background-color: lightblue;
}
section{
    margin: 0;
    padding: 5px;
    background-color: pink;
}
article{
    margin: 0;
    padding: 5px;
    background-color: yellow;
}
aside {
    width: 30%;
    padding-left: 15px;
    margin-left: 15px;
    float: right;
    font-style: italic;
    background-color: lightgray;
}
footer{
    margin: 0;
    width: 100%;
    text-align: center;
    background-color: lightgreen;
}
```

html5Demo.html

```
<!DOCTYPE html>

<html>

  <head>

    <title>HTML 5 Demo</title>
    <link rel="stylesheet" href="newhtml.css">
  </head>
  <body>
    <header>
      <nav>
        <a href="/html/">HTML</a> | <a href="/css/">CSS</a> |
        <a href="/js/">PHP</a> | <a href="/jquery/">MySQL</a>
      </nav>
    </header>
    <section>
      <h1>Microsoft Edge</h1>

      <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>

      <aside>

        <h4>HTML5 </h4>

        <p>HTML5 has new structural elements that define specific web page features with much more accurate semantics than were available in HTML 4.</p>

      </aside>
    </section>
    <article>
      <h2>Google Chrome</h2>

      <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>

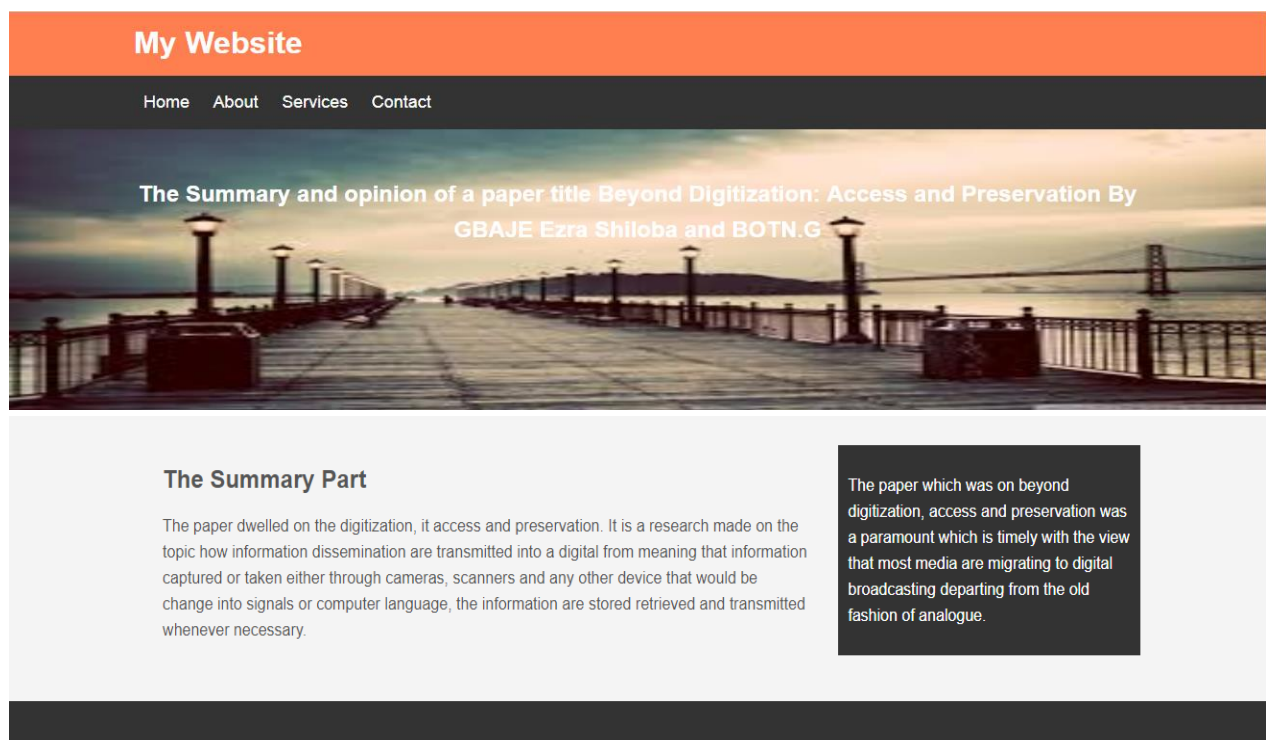
      <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>
    </article>
    <footer>
      <p>Author: Elgurama<br><a href="mailto:hege@example.com">elgee@gmail.com</a>
    </p>
    </footer>
  </body>
</html>
```

OUTPUT

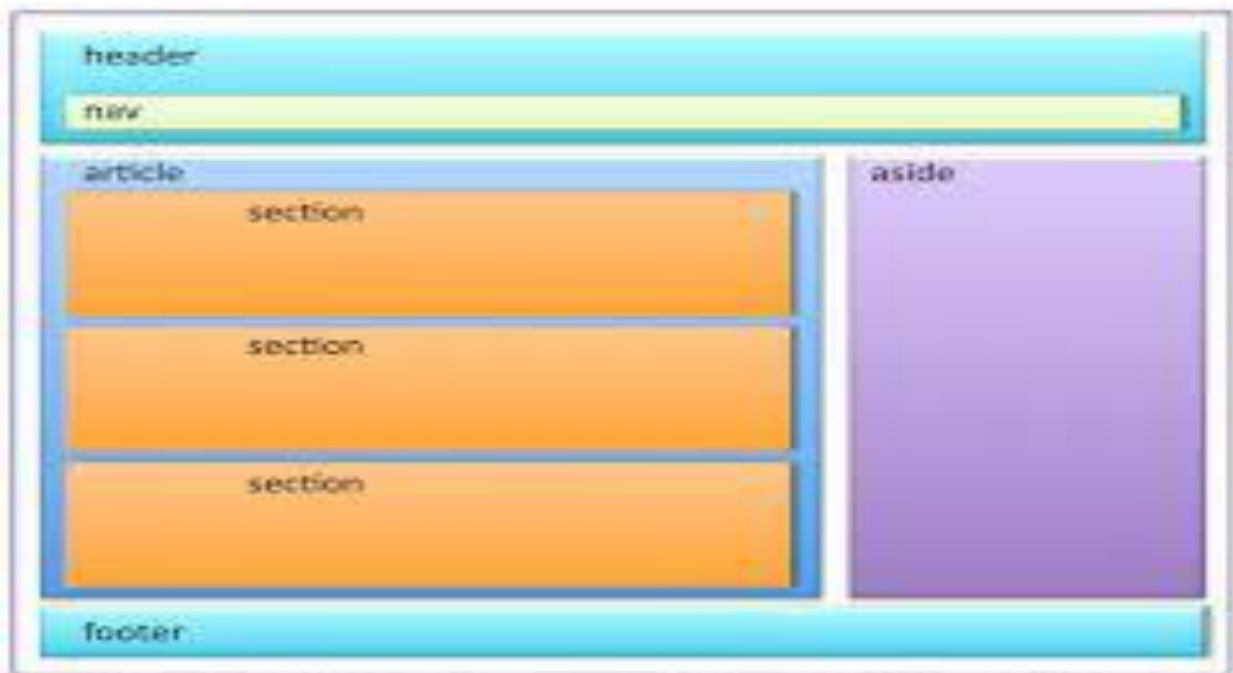


EXERCISE

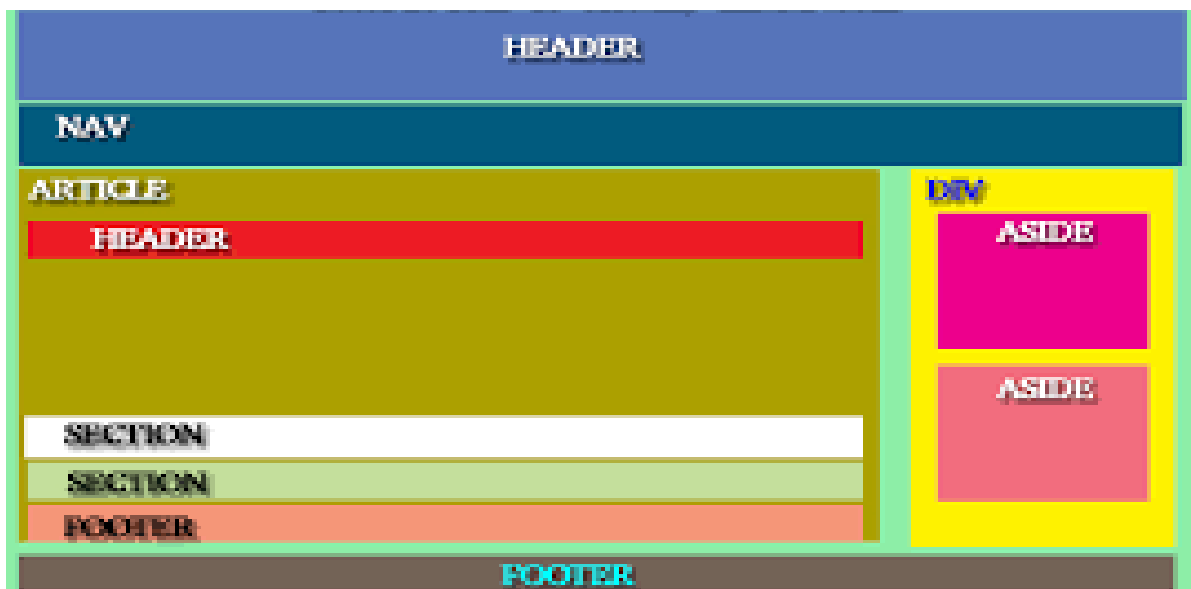
1. Explain html5 structural element with a code example.
2. Draw the diagram the basic html5 structural element.
3. Use HTML 5 and CSS to design the websites below:
 - i.



ii.



iii.



CHAPTER SIX

JAVASCRIPT

INTRODUCTION TO JAVASCRIPT

JavaScript is the most popular and widely used client-side scripting language. Client-side scripting refers to scripts that run within your web browser. JavaScript is designed to add interactivity and dynamic effects to the web pages by manipulating the content returned from a web server.

JavaScript is an object-oriented language, and it also has some similarities in syntax to Java programming language. But JavaScript is not related to Java.

Definition: JavaScript is a client side (front-end) scripting language that is used for creating dynamic web pages.

HISTORY OF JAVASCRIPT

JavaScript was originally developed at Netscape in 1995 by Brandon Eich. It was originally called Mocha, later renamed as LiveScript and lastly renamed to JavaScript. It became an ECMA (European Computer Manufacturers Association) standard in 1997. Now JavaScript is the standard client-side scripting language for web-based applications, and it is supported by virtually all web browsers available today, such as Google Chrome, Mozilla Firefox, Apple Safari, etc.

APPLICATION OF JAVASCRIPT

JavaScript is used to create interactive websites. It is mainly used for:

- i. Client-side validation.
- ii. Dynamic drop-down menus
- iii. Displaying date and time
- iv. Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box)
- v. Displaying clocks etc

ADVANTAGES OF JAVASCRIPT

JavaScript allows interactivity such as:

- i. Implementing form validation
- ii. React to user actions, e.g. handle keys
- iii. Changing an image on moving mouse over it
- iv. Sections of a page appearing and disappearing
- v. Content loading and changing dynamically
- vi. Performing complex calculations
- vii. Custom HTML controls, e.g. scrollable table

DISADVANTAGES OF JAVASCRIPT

- i. Client – side security: since the code runs on user computer, it can be exploited for malicious purposes.
- ii. JavaScript is sometimes interpreted differently by different browsers

Syntax

```
<script type="text/javascript" src="filename.extension"></script>
```

The script tag takes two important attributes:

- i. **Type:** This attribute is what is to indicate the scripting language in use and its value should be set to “text/javascript”.
- ii. **Src (Source):** This attribute is use to point where the .js file is located.

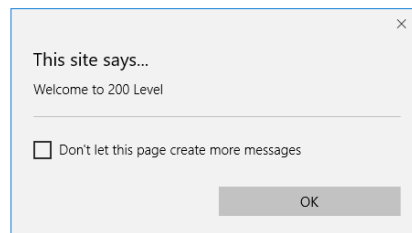
EMBEDDING JAVASCRIPT IN HTML

- i. **INTERNAL:** Placing the JavaScript code directly inside body.
- ii. **EMBEDDED:** Placing the JavaScript code in the head of the HTML file.
- iii. **EXTERNAL:** Creating an external JavaScript file with the .js extension and then load it within the page through the src attribute of the <script> tag.

INTERNAL EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline JavaScript</title>
  </head>
  <body>
    <button onclick="alert('Hello World!')">Click Me</button>
  </body>
</html>
```

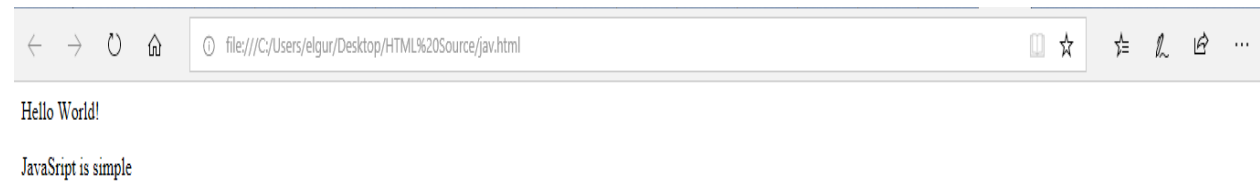
OUTPUT



EMBEDED EXAMPLE

```
<!DOCTYPE html>
<html>
<head>
  <title>Embedding JavaScript</title>
  <script>
    var greet = "Hello World!";
    document.write(greet);
  </script>
</head>
<body>
  <p>JavaSript is simple</p>
</body>
</html>
```

OUTPUT



EXTERNAL EXAMPLE

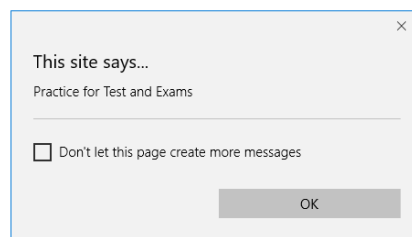
hello.js

```
function msg() {  
    alert("Practice for Test and Exams");  
}
```

javascriptDemo.html

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Including External JavaScript File</title>  
        <script type="text/javascript" src="hello.js"></script>  
    </head>  
    <body>  
        <form>  
            <input type="button" value="CLICK" onClick="msg()">  
        </form>  
    </body>  
</html>
```

OUTPUT



GENERATING OUTPUT IN JAVASCRIPT

There are certain situations in which you may need to generate output from your JavaScript code. For example, you might want to see the value of variable, or write a message to browser console to help you debug an issue in your running JavaScript code, and so on.

In JavaScript there are several different ways of generating output including writing output to the browser window or browser console, displaying output in dialog boxes, writing output into an HTML element, etc.

1. WRITING OUTPUT TO BROWSER CONSOLE

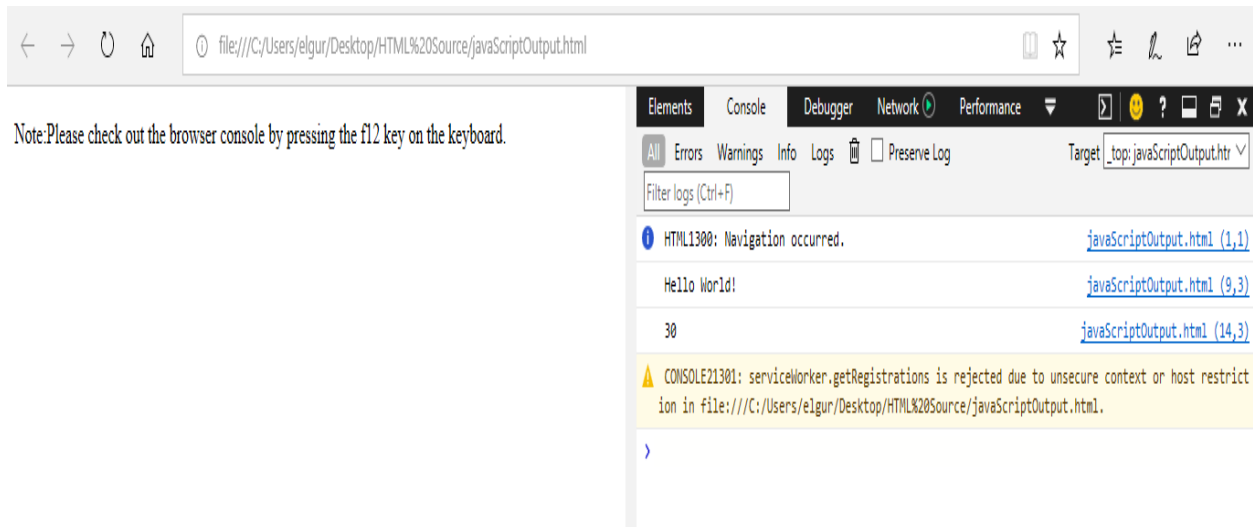
You can easily outputs a message or writes data to the browser console using the `console.log()` method. This is a simple, but very powerful method for generating detailed output.

Tip: To access your web browser's console, first press F12 key on the keyboard to open the developer tools then click on the console tab.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Output</title>
  </head>
  <body>
    <script>
      // Printing a simple text message
      console.log("Hello World!");
      // Printing a variable value
      var x = 10;
      var y = 20;
      var sum = x + y;
      console.log(sum);
    </script>
    <p>Note:Please check out the browser console by pressing the f12 key on the
keyboard.</p>
  </body>
</html>
```

OUTPUT



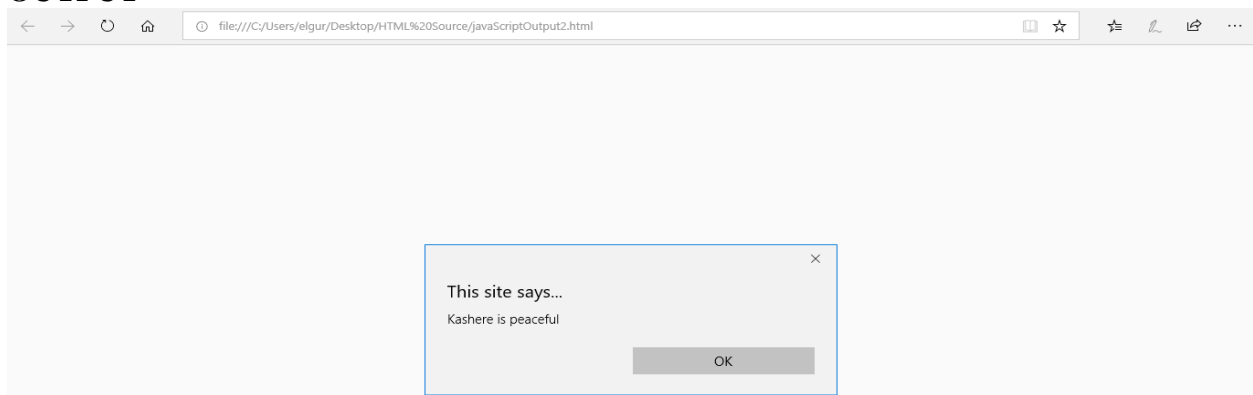
2. DISPLAYING OUTPUT IN ALERT DIALOG BOXES

You can also use alert dialog boxes to display the message or output data to the user. An alert dialog box is created using the `alert()` method.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Output</title>
  </head>
  <body>
    <script>
      // Printing a simple text message
      alert("Kashere is peaceful");
    </script>
  </body>
</html>
```

OUTPUT



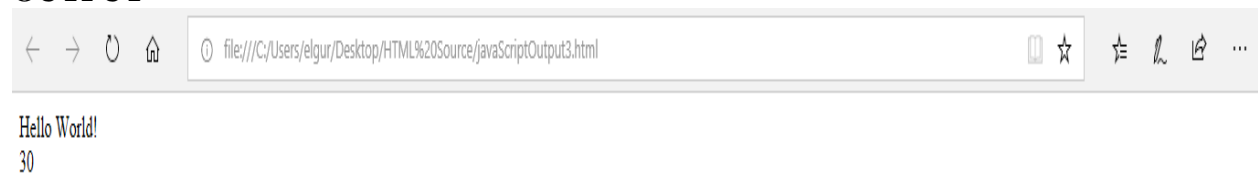
3. WRITING OUTPUT TO THE BROWSER WINDOW

You can use the `document.write()` method to write the content to the current document only while that document is being parsed.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Output</title>
  </head>
  <body>
    <script>
      // Printing a simple text message
      document.write("Hello World!");
    </script>
    <br>
    <script>
      // Printing a variable value
      var x = 10;
      var y = 20;
      var sum = x + y;
      document.write(sum);
    </script>
  </body>
</html>
```

OUTPUT



4. INSERTING OUTPUT INSIDE AN HTML ELEMENT

You can also write or insert output inside an HTML element using the element's `innerHTML` property. However, before writing the output first we need to select the element using a method such as `getElementById()`.

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Output</title>
  </head>
  <body>
    <p id="greet"></p>
    <p id="result"></p>

    <script>
      // Writing text string inside an element
      document.getElementById("greet").innerHTML = "Welcome to Gombe State";

      // Writing a variable value inside an element
      var x = 10;
      var y = 2;
      var sum = x / y;
      document.getElementById("result").innerHTML = sum;
    </script>
  </body>
</html>
```

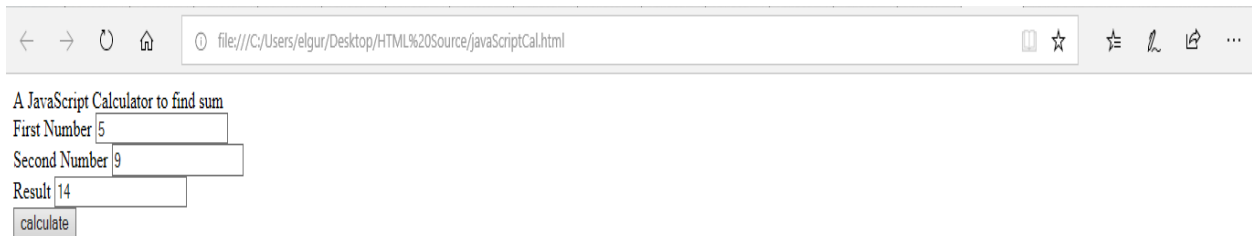
OUTPUT



Welcome to Gombe State

JAVASCRIPT SIMPLE CALCULATOR

Write a code that displays the following calculator that can calculate the sum of First and Second Number:



A JavaScript Calculator to find sum
First Number 5
Second Number 9
Result 14
calculate

SOLUTION

```
<DOCTYPE html>
<html>
  <head>
    <title>JavaScript sum Calculator</title>
  </head>
  <body>
    <form name="cal">
      <label>A JavaScript Calculator to find sum</label><br>
      <label>First Number</label>
      <input type="text" name="num1"><br>
      <label>Second Number</label>
      <input type="text" name="num2"><br>
      <label>Result</label>
      <input type="text" name="res"><br>
      <input type="button" name="cal" value="calculate"
        onClick="sumNum()">
    </form>
    <script type="text/javascript">
      function sumNum()
      {
        var num1, num2, res;
        num1=Number(document.cal.num1.value);
        num2=Number(document.cal.num2.value);
        res=num1+num2;
        document.cal.res.value=res;
      }
    </script>
  </body>
</html>
```

JAVASCRIPT EVENT

HTML events are "**things**" that happen to HTML elements. When JavaScript is used in HTML pages, JavaScript can "**react**" on these events. An HTML event can be something the browser does, or something a user does. Here are some examples of HTML events:

- i. An HTML web page has finished loading
- ii. An HTML input field was changed
- iii. An HTML button was clicked

Often, when events happen, you may want to do something. JavaScript lets you execute code when events are detected.

MOUSE EVENT

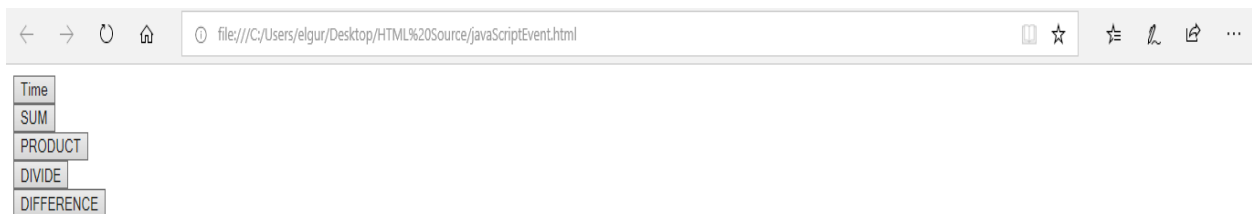
Events	Description
OnClick	occurs when element is clicked.
Ondblclick	occurs when element is double-clicked.
Onmouseover	occurs when mouse is moved over an element.
Onmouseout	occurs when mouse is moved out from an element (after moved over).
Onmousedown	occurs when mouse button is pressed over an element.
Onmousemove	occurs when mouse pointer moves within an element.
Onmouseup	occurs when mouse is released from an element (after mouse is pressed).

EXAMPLE

```
<DOCTYPE html>

<html>
  <head>
    <title>JavaScript Events</title>
  </head>
  <body>
    <script>
      var x=5;
      var y=10;
      var sum=x+y;
      var diff=x-y;
      var mul=x*y;
      var div=x/y;
    </script>
    <button onclick="document.write(Date())">Time</button><br>
    <button ondblclick="document.write(sum)">SUM</button><br>
    <button onmouseout="document.write(mul)">PRODUCT</button><br>
    <button onmousedown="document.write(div)">DIVIDE</button><br>
    <button onmouseover="document.write(diff)">DIFFERENCE</button><br>
  </body>
</html>
```

OUTPUT



JAVASCRIPT FORM VALIDATION

Form validation is the process of making sure appropriate values are inserted and submitted through a form. JavaScript provides facilities to validate forms on the client side so processing will be fast than server-side validation. So, most of the web developers prefer JavaScript form validation. Through JavaScript, we can validate name, password, email, date, mobile number etc fields.

EXAMPLE

Validate a form to have the following:

A form with name, password and email elements.

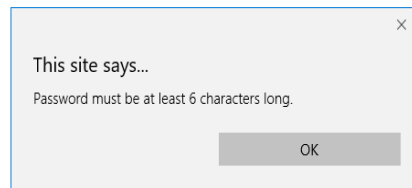
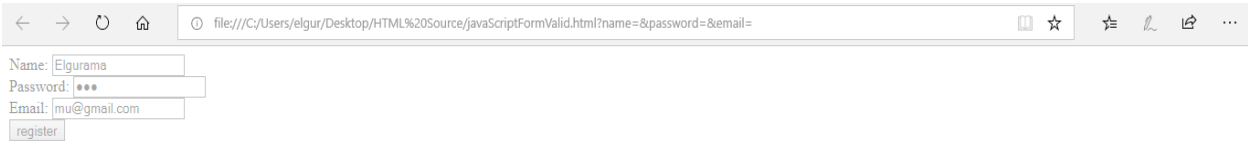
Name can't be empty, **Password** can't be less than 6 characters long, **Email** can't be empty.

NB: The user will not be forwarded to the next page until all the values are correct.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function validateform(){
        var name=document.myform.name.value;
        var password=document.myform.password.value;
        var email=document.myform.email.value;
        if (name==null || name==""){
          alert("Name can't be blank");
        }
        else if(password.length<6){
          alert("Password must be at least 6 characters long.");
        }
        else if (email==null || email==""){
          alert("Email can't be blank ");
        }
      }
    </script>
  </head>
```

```
<body>
    <form name="myform" method="" action="" onsubmit="return validateform()" >
        Name: <input type="text" name="name"><br/>
        Password: <input type="password" name="password"><br/>
        Email: <input type="text" name="email"><br/>
        <input type="submit" value="register">
    </form>
</body>
</html>
```

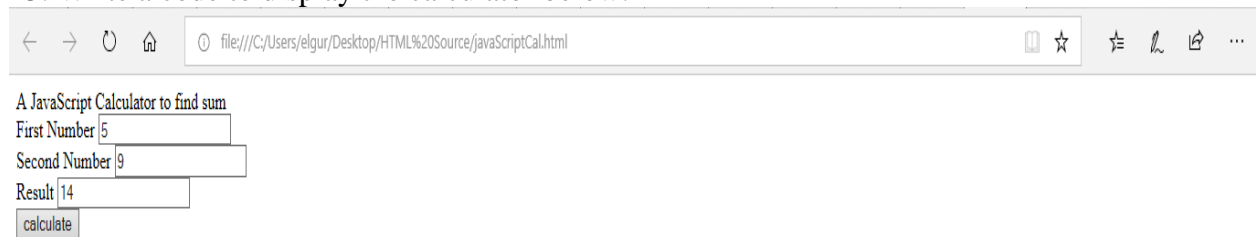
OUTPUT



EXERCISES

1. Define JavaScript.
2. Write the brief history of JavaScript.
3. List 5 application of JavaScript.
4. List 5 advantages of JavaScript.
5. List 5 disadvantages of JavaScript.
6. List and explain the ways of embedding JavaScript in html.
7. Write a code to demonstrate inline JavaScript in html.
8. Write a code to demonstrate embedded JavaScript in html.
9. Write a code to demonstrate external JavaScript in html.
10. Write and explain JavaScript script tag.
11. List all the ways of generating output in JavaScript.
12. Write a code to demonstrate each ways of generating output in JavaScript.

13. Write a code to display the calculator below:



14. Explain JavaScript mouse events.

15. List and explain 5 JavaScript events.

16. Write a code to demonstrate all the JavaScript mouse events.

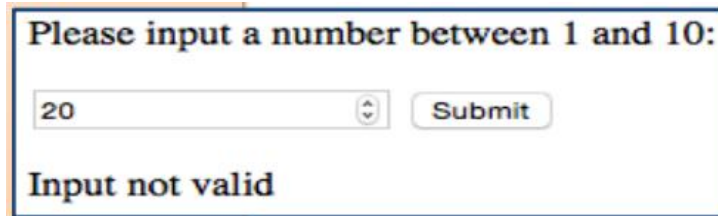
17. Write a code to demonstrate form validation with the following:

- i. **Name** can't be empty, **password** can't be less than 6 characters long, **email** can't be empty.

18. Write a code to demonstrate form validation with the following:

- i. **Name** can't be empty, **password** and **confirm password** must be the same characters

19. Write a code to demonstrate the following form validation:



20. Write a code to demonstrate form validation with the following:

- i. **Name** can't be empty, **phone number** must be 11 characters, **email** must have @ symbol.

21. Write a code to checks if an alphabet is a **vowel** or **consonant**.

22. Write a code to checks if a number is **even** or **odd**.