

UNIT- I

What is Internet:

Internet or internetworking refers to a wide network through which computers are interconnected globally with one another and capable of sharing resources among themselves. This network is called 'internet'.

Internet services and accessibility:-

Internet access is the ability of individuals and organizations to connect to the Internet using computer terminals, computers, and other devices; and to access services such as email and the World Wide Web. Some of the services offered are as follows:

Electronic mail:- A services that helps to send and receive message and to attach files.

Electronic mailing list:- Everyone subscribed to the list gets a message sent to the list.

USENET newsgroup:- Electronic bulletin board service.

Real-time communication:- chat, messengers, video-conferencing , whiteboards, etc...

File Transfer Protocol (FTP):- A service that helps to send and receive files to and from a file server.

Telnet, ssh:- A remote login to other computers on the internet , possibly anonymously.

Gopher:- This is an older menu-driven document system , which is now mostly replaced by the web.

World Wide Web (WWW) :- It is an information system where documents and other web resources are identified by Uniform Resource Locators (URLs, such as <https://example.com/>), which may be interlinked by hypertext, and are accessible over the Internet.

The internet users gain access to these services through Internet Service Providers (ISP). There are different types of internet connections:-

Dial-Up Connections:- A dial-up connection is established when two or more communication devices use a public switched telephone network (PSTN) to connect to an Internet service provider (ISP).

Many remote areas depend on Internet dial-up connections because broadband and cable are rare in remote areas with low population numbers.

ISPs often provide free dial-up connections, a viable alternative for budget-conscious subscribers.

High Speed Connections:- High speed connection allow information to travel quickly. With this connection, a lot of internet services work better at higher speeds. There are four types of high speed connections:

Digital Subscriber Lines (DSL):- with DSL, the computer is always connected to the internet. This technology sends digital data through existing phone lines carry internet services.

Cable:- Although the same wires are used there is no interference to / from the telephone calls. The internet signals is carried on the TV cable network.

Satellite :- This connection needs a phone line to send information. It receives data at high speed from the satellite.

Integrated Services Digital Network (ISDN):- ISDN is an international communication standard for sending voice, video and over digital telephone lines (or) over normal telephone lines.

Uses of the Internet:

Internet service have led to its use in a number of different areas. Some of the services provided by the internet are:-

- Telecommunicating and online conferencing
- Business, advertising and online shopping
- News, jobs, softwares
- Online courses, virtual classrooms, coachings
- Government services, politics and national defence

- Electronic publishing
- Entertainment
- Teaching and learning
- Scholarly research
- General information about a subject, especially which is not easily available from other sources
- Correspondence (email, chatting, etc..)

Protocols:

Protocol is a set of rules (or) an agreement that specifies a common language that computers on a network use for communication with other computers. There are various protocol suites available.

Ethernet:- This is used to transfer information on a LAN. It specifies a number of wiring and signaling standards for the physical layer, two ways of network access (i) media access control ii) data link layer) and a common address format.

Internet Protocol (IP):- This protocol provides communicable global address of the computers. The computers identify each other by the IP address.

Transport Control Protocol (TCP):- This protocol guarantees reliable, proper delivery of data from the sender to receiver. It breaks large messages, transport them reliably and reassembles them.

File Transfer Protocol (FTP):- This is used to connect two computers over the internet so that the user of one computer can transfer files and perform file commands on the other computer. It exchanges files over any network that supports TCP/IP protocol.

Hypertext Transport Protocol (HTTP):- This protocol is used to retrieve web pages from a web server.

Simple Mail Transfer Protocol (SMTP):- This protocol is used for email transmissions.

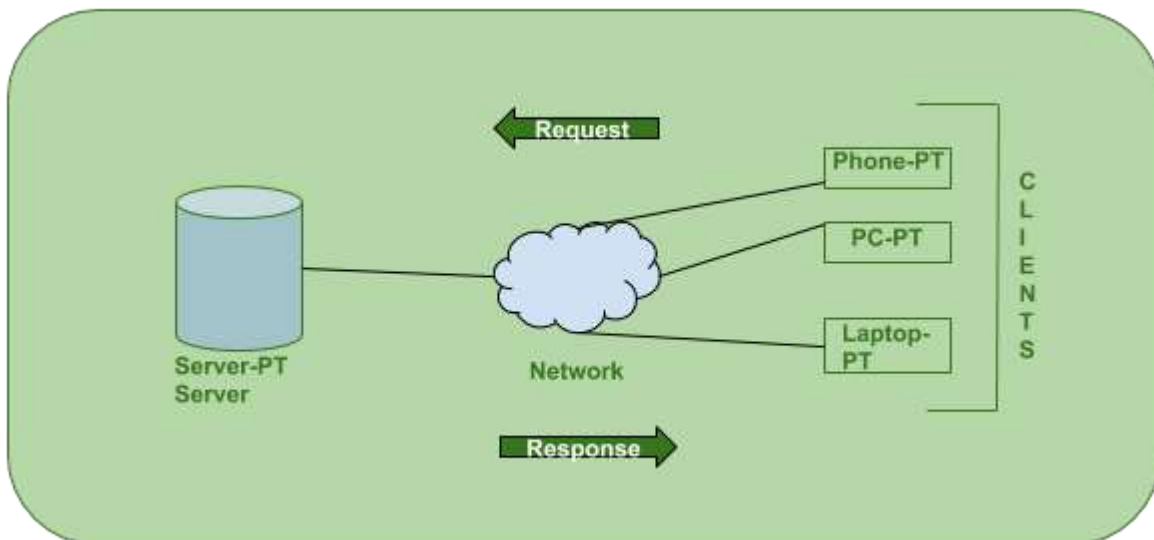
Web Concepts:-

- A World Wide Web is an international hypertext system that links together millions of documents.
- A hypertext link is a word or a picture, which requests a different file from the internet when you click on it.
- A website is a collection of related pages.
- A web browser is a program that displays the web pages it retrieves.

1) **The client / server model of the web:-**

Most internet services rely on the client/ server model. The internet user is the client and has the client software installed on his computer to access various internet services.

When a user wants to connect to a particular information tool , he uses his client software to connect to server programs which provide the service or the information needed.



The web browser is an example of the client software needed to access the world wide web servers.

2) Retrieving data from the web:-

We can retrieve data by using address of the file. This address is called a Uniform Resource Locator (URL). The address box of the browser shows the URL of the currently displayed document.

Structure of URLs:- In the same way as each host computer has a unique IP address, every data file or document on the internet also has a unique address called a URL. The URL consists of three parts:-

- i) **protocol:-** It is set of rules which the computer follows in order to communicate with other computers. It lets the computer know how to process the information it receives. If the protocol is http:// the computer knows it will be processing a world wide web document.
- ii) **domain name:-** domain name is the internet address of the computer that is hosting the site and storing the documents. This domain name may be expressed as an IP address.
- iii) **Path:-** It is directory of the file specification. It lets the computer know which directory and file to access after connecting to the server.

Example:- <http://www.linccweb.orb/eresources.asp>

http:// is the protocol. www.linccweb.orb/ is the domain name.
eresources.asp provides the path to the specific page you want.

INTERNET PROTOCOLS

- The internet is a collection of end-systems that communicate using the TCP/IP protocol suite.
- The TCP/IP protocol suite includes services at the network and transport layer.
- Network layer needs some services from data link layer to provide basic communication between end-systems on the same network.
- TCP/IP does not include a data link layer. It means TCP/IP can run on many different types of networks.

- Internet Protocols are very popular open-system protocols because they are inter-communicable between different sets of inter-connected networks and are friendly to both LAN and WAN.

Ethernet-a data link layer

It is a popular data link layer used in many LANs. It provides unique addresses, but it is not globally communicable.

All Ethernet networks have some common mechanisms for transmission of data between computers on the same network. it is mentioned below:

- Multi access
- Every Ethernet interface has a unique 48-bit address called as hardware address or physical address.
- The broadcast is a message that is received by all end-systems on a network.

Internet protocols

- Internet protocols used to communicate across inter-connected networks.
- The TCP/IP along with the corresponding ISO/OSI layers.

1) Internet Protocols (IP)

- The network layer of TCP/IP is called Internet Protocol
- It is responsible for sending data between end-systems on different networks.
- IP networks have the following properties:
 - Each end-system have unique IP address. Some systems have multiple network interfaces and therefore multiple IP addresses.
 - The IP layer in each end-system uses a data link layer to transmit and receive IP packets.

IP addresses

Each IP address is a pair of network address (net id) and host address.

Network Address	Host Address
-----------------	--------------

An IP address is 32 bits long and its usually shown using **dotted decimal notation**

Example:-

128.213.1.1

The first two address 128.213 is the network address and last two digit 1.1 is the host address.

Types of internet addresses

Internet addresses are divided into 5 classes A,B,C,D,E. They are

Class A

7 bits 24 bits

0	net id	host id
---	--------	---------

Class B

14 bits 16 bits

1	0	net id	host id
---	---	--------	---------

Class C

21 bits 8 bits

1	1	0	net id	host id
---	---	---	--------	---------

Class D

28 bits

1	1	1	0	multicast group id
---	---	---	---	--------------------

Class E

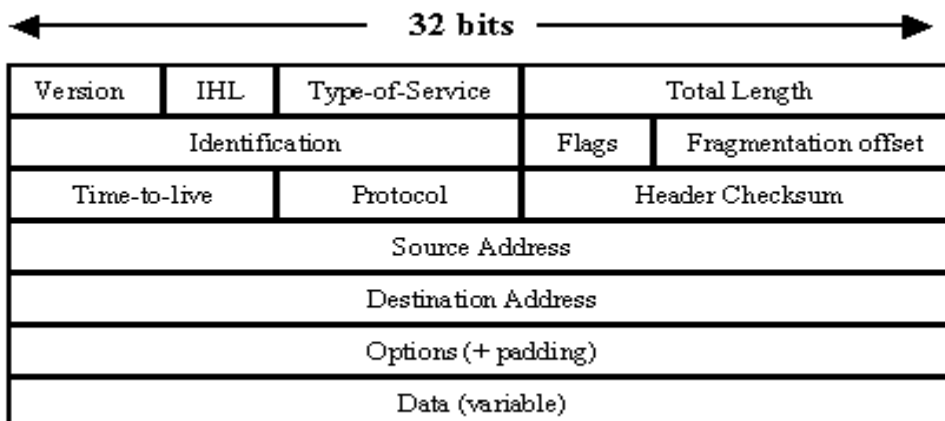
27 bits

1	1	1	1	0	reserved for future use
---	---	---	---	---	-------------------------

Address Range:

Class	Address Range
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 254.255.255.255

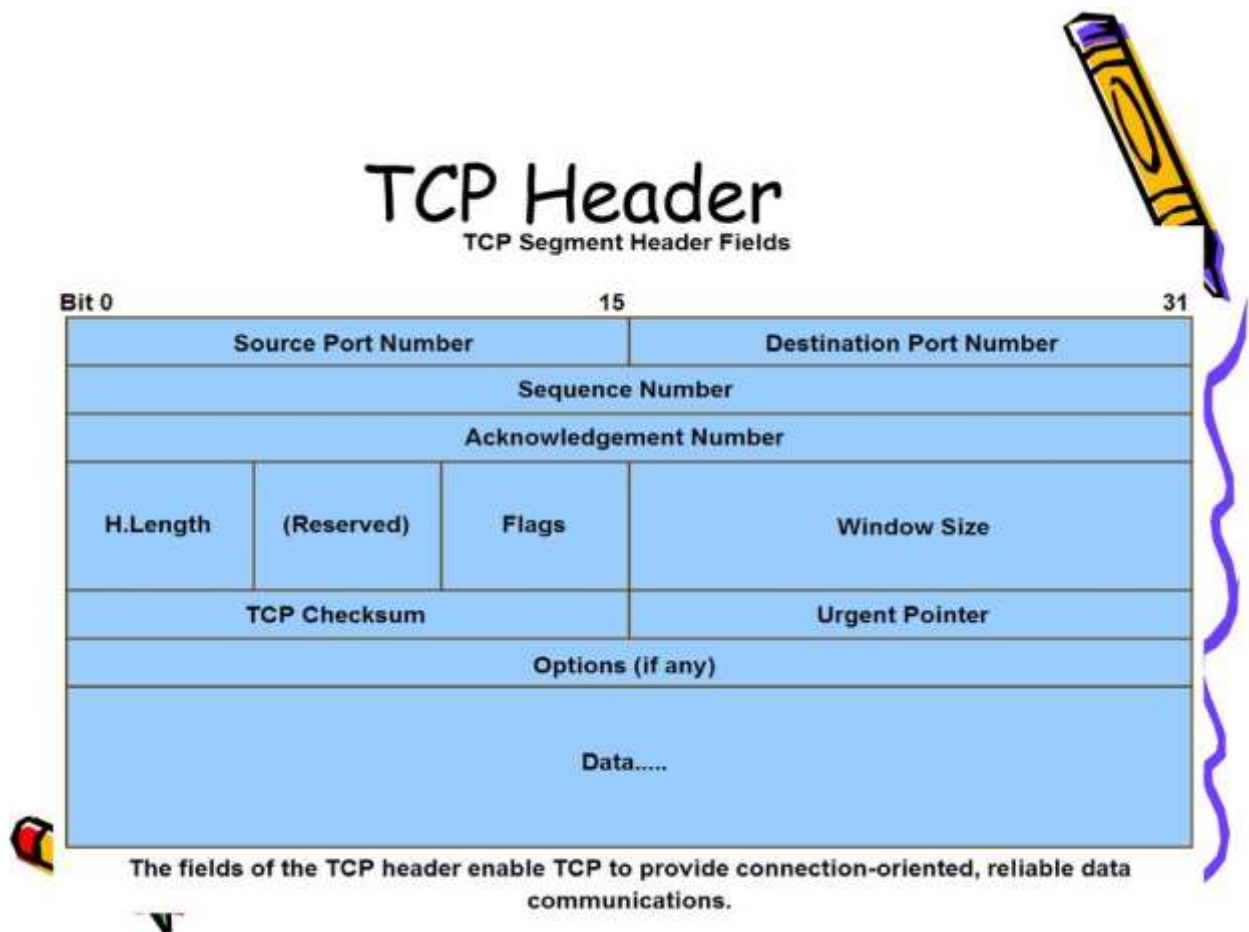
Packet Format:



- **Version** – Version number of Internet Protocol used (e.g. IPv4).
- **IHL** – Internet Header Length; Length of entire IP header.
- **DSCP** – Differentiated Services Code Point; this is Type of Service.
- **ECN** – Explicit Congestion Notification; It carries information about the congestion seen in the route.
- **Total Length** – Length of entire IP Packet (including IP header and IP Payload).
- **Identification** – If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to.
- **Flags** – As required by the network resources, if IP Packet is too large to handle, these ‘flags’ tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to ‘0’.
- **Fragment Offset** – This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live** – To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol** – Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum** – This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- **Source Address** – 32-bit address of the Sender (or source) of the packet.
- **Destination Address** – 32-bit address of the Receiver (or destination) of the packet.
- **Options** – This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

2) Transmission Control Protocol:-

- Transmission Control Protocol is the communication protocol for communication between computers on the Internet.
- It defines how electronic devices (like computers) should be connected to the Internet, and how data should be transmitted between them.



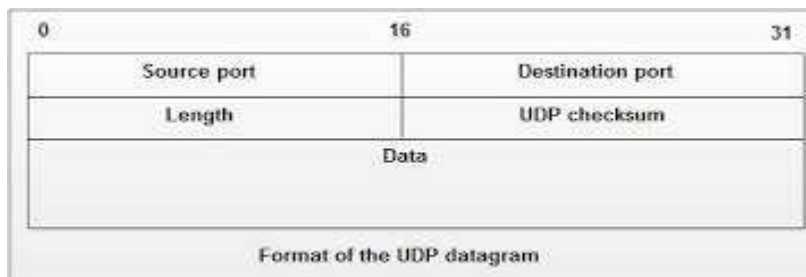
- **Source Port Address** –
Specify the port numbers at source host and destination host
- **Destination Port Address** –
Number assigned to first byte of data stream if it has been split.
- **Acknowledgement Number** –
Specifies acknowledgement from the other end.
- **Header Length (HLEN)** –
Length of the header in bytes.
- **Control flags** –
Defines 6 control bits for enabling flow control, connection establishment, and termination and the mode of transfer.
- **Window size** –
Defines the size of the window.
- **Checksum** –
Provided for error detection and correction.
- **Urgent pointer** –
Used if the some optional information in the header field.

User Datagram Protocol:

User Datagram Protocol connectionless transport layer protocol.

UDP ports identify the various applications running on a device.

Most applications on the internet are based on TCP. Both TCP and UDP must provide communication between process identified by protocol ports.



Source port :	Source number of the source host
Destination port:	Port number of destination host
Length:	Total length of the user datagram
Checksum:	Used to detect errors for the entire datagram

3) Host Names

- A host name is used when a web client makes an HTTP request to a host.
- The user making the request can specify the IP address of the server rather than the host name, but that is now unusual on the Internet.
- Host names are more convenient for users than numeric IP addresses.
- Companies, organizations, and individuals frequently choose host names for their web sites that can be easily remembered by users.

UNIT - II

HTML

Introduction

- HTML stands for Hyper Text Markup Language
- It is the standard markup language for creating Web pages
- It describes the structure of a Web page
- It consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

SGML (Standard Generalized Markup Language)

- SGML (Standard Generalized Markup Language) is a standard for how to specify a document markup language or tag set.
- Such a specification is itself a document type definition (DTD).
- SGML is not in itself a document language, but a description of how to specify one.
- It is metadata.

Advantages:

- They can be created by thinking in terms of document structure rather than appearance characteristics (which may change over time).
- They will be more portable because an SGML compiler can interpret any document by reference to its document type definition (DTD).
- Documents originally intended for the print medium can easily be re-adapted for other media, such as the computer display screen.

DTD (Document Type Definition):

- A DTD is a Document Type Definition.
- A DTD defines the structure and the legal elements and attributes of an XML document.
- With a DTD, independent groups of people can agree on a standard DTD for interchanging data.
- An application can use a DTD to verify that XML data is valid.

The DTD above is interpreted like this:

- **!DOCTYPE note** defines that the root element of this document is note
- **!ELEMENT note** defines that the note element must contain four elements: "to,from,heading,body"
- **!ELEMENT to** defines the to element to be of type "#PCDATA"
- **!ELEMENT from** defines the from element to be of type "#PCDATA"
- **!ELEMENT heading** defines the heading element to be of type "#PCDATA"
- **!ELEMENT body** defines the body element to be of type "#PCDATA"

DTD – Elements

- DTD element is declared with an ELEMENT declaration.
- When an XML file is validated by DTD, parser initially checks for the root element and then the child elements are validated.

Syntax

All DTD element declarations have this general form –

```
<!ELEMENT elementname (content)>
```

- **ELEMENT** declaration is used to indicate the parser that you are about to define an element.
- **elementname** is the element name (also called the **generic identifier**) that you are defining.
- **content** defines what content (if any) can go within the element.

Attributes:

- Attribute gives more information about an element or more precisely it defines a property of an element.
- An XML attribute is always in the form of a name-value pair. An element can have any number of unique attributes.

Syntax

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

The DTD attributes start with **<!ATTLIST** keyword if the element contains the attribute.

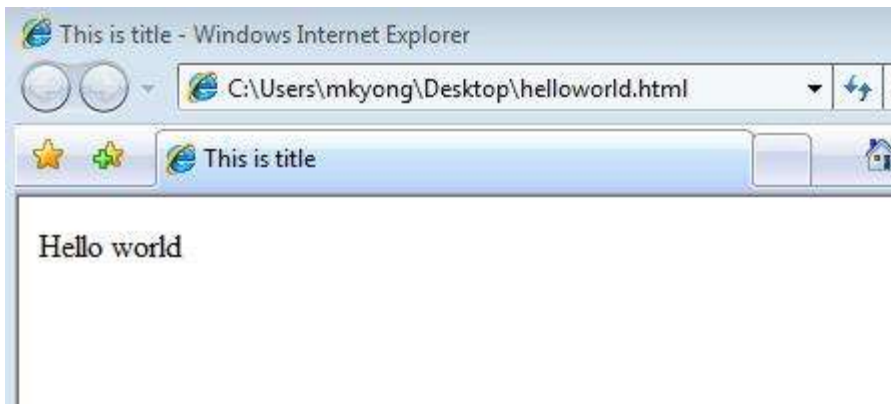
- **element-name** specifies the name of the element to which the attribute applies.
- **attribute-name** specifies the name of the attribute which is included with the element-name.
- **attribute-type** defines the type of attributes. We will discuss more on this in the following sections.
- **attribute-value** takes a fixed value that the attributes must define. We will discuss more on this in the following sections.

Outline of an html document:

Every tag should be enclosed with angled brackets and should have closing tag like `<html>...</html>`

Example:

```
<html>
  <head>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```



Every html document has **2** sections:

- head
- body

Head section

The HTML `<head>` element is a container for the following elements: `<prologue>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

Prologue

The prologue component is only a comment which tells about the HTML version

```
<!Doctype HTML 4.0>
```


Style

The `<style>` element is used to define style information for a single HTML page:

Example

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

Meta

The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

Define the character set used:

```
<meta charset="UTF-8">
```

Link

The `<link>` element defines the relationship between the current document and an external resource.

The `<link>` tag is most often used to link to external style sheets:

```
<link rel="stylesheet" href="mystyle.html">
```

Script

The `<script>` element is used to define client-side JavaScripts.

The following JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

Example

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

Base

The <base> element specifies the base URL and/or target for all relative URLs in a page.

The <base> tag must have either an href or a target attribute present, or both.

There can only be one single <base> element in a document!

syntax

```
<base href="https://www.w3schools.com/" target="_blank">
```

Body Section

The <body> tag defines the document's body.

The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Attributes

Bgcolor:- specifies the background color of the web page. it can be specified as bgcolor="red"

Link:- specifies the color of the unvisited link.

Alink:- specifies the color of the active link.

Vlink:- specifies the visited link color.

Text:- specifies the text color.

1) Headers:-

These are formatting tags to vary the text size.

```
<html>
```

```
<head>
```

```
<title> header examples </title>
```

```
</head>
```

```
<body>
```

```
<center>
```

```
<h1> Hello World </h1>
```

```
<h2> Hello World </h2>
```

```
<h3> Hello World </h3>
```

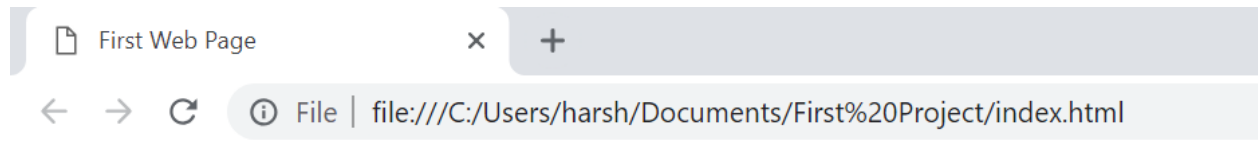
```
<h4> Hello World </h4>
```

```
<h5> Hello World </h5>
```

```
<h6> Hello World </h6>
```

```
</body>
```

```
</html>
```



Hello World!

Hello World!

Hello World!

Hello World!

Hello World!

Hello World!

2) Paragraph

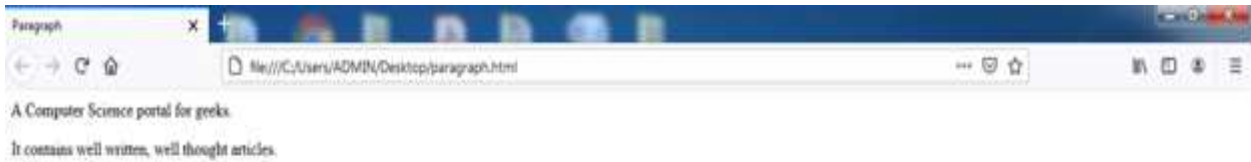
- The `<p>` tag in HTML defines a paragraph. These have both opening and closing tag. So anything mentioned within `<p>` and `</p>` is treated as a paragraph.
- Most browsers read a line as a paragraph even if we don't use the closing tag i.e, `</p>`, but this may raise unexpected results. So, it is both a good convention and we **must** use the closing tag.

Syntax:

```
<p> Content </p>
```

Example:

```
<html>
<head>
<title>Paragraph</title>
</head>
<body>
<p>A Computer Science portal for geeks.</p>
<p>It contains well written, well thought articles.</p>
</body>
</html>
```



3) Text Formatting

- The page content in the body of the web page can be presented in a neat way using different tags like ``, `` ``.
- The `` tag is used to make the text in **italics**.
- The `` tag is used to make text as **bold**.
- The `` tag is used for text styling. This tag contains various attributes like **face**, **color** and **size**.

```
<p><fontcolor="red"size="+1"face="Arial"></font></p>
```

Example:

```
<html>
<head>
<title>Paragraph</title>
</head>
<body>
<h1 align=center> hello world </h1>
<p> The world is so beautiful.</p>
<br>
<p> The most beautiful creation of the world is the <strong><em>
human kind </strong></em> </p>
</body>
</html>
```



4) Linking

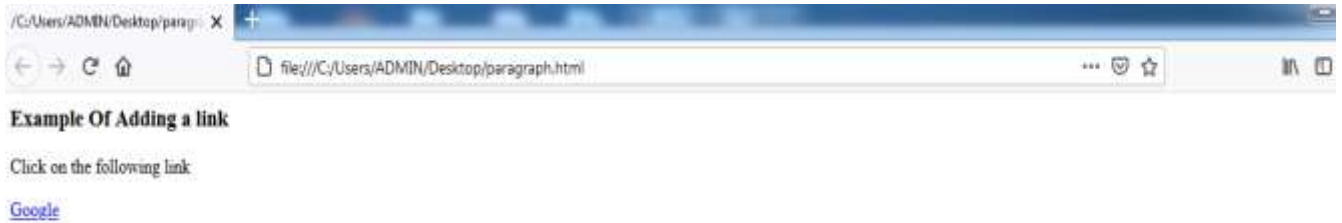
- The link starts at the “source” anchor and points to the “destination” anchor, which may be any Web resource such as an image, a video clip, a sound bite, a program, an HTML document or an element within an HTML document.

Syntax

```
<a href =”url”>text link </a>
```

Example

```
<html>  
  
<h3>Example Of Adding a link</h3>  
  
<body>  
  
<p>Click on the following link</p>  
<a href = "www.google.com">Google</a>  
</body>  
  
</html>
```



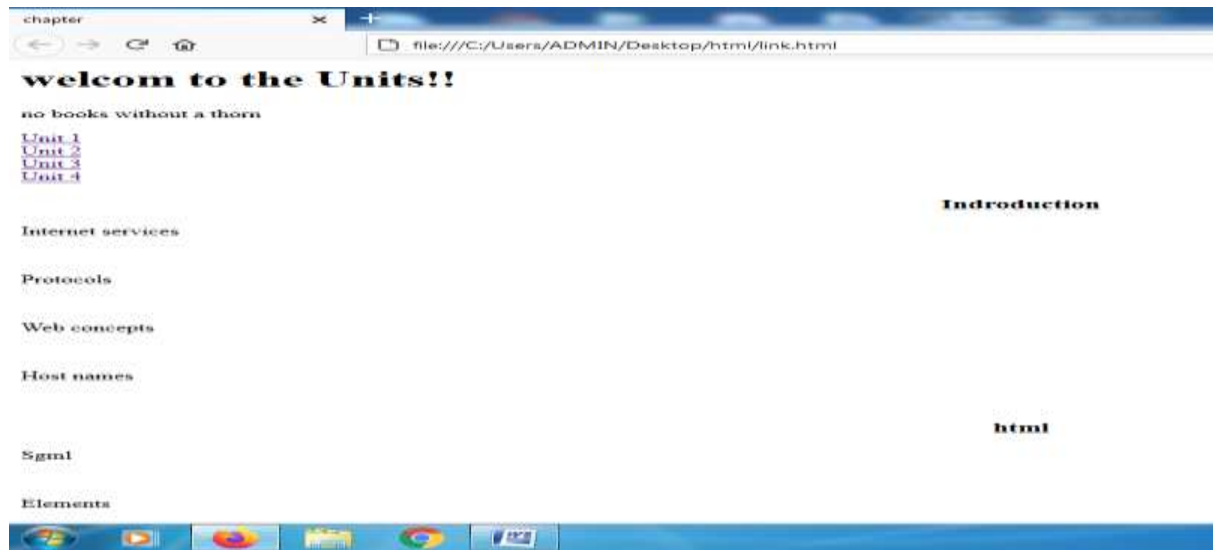
5) Internal linking

- HTML internal link is linked within the same web page. This link can be an absolute path or relative path.
- When you click on an internal anchor link, you will scroll automatically to the referred section and display it on your browser.

Example

```
<html>
<head>
<title>chapter</title>
</head>
<body>
<h1>welcome to the Units!!</h1>
<p>no books without a thorn</p>
<a href="#unit1">Unit 1 </a><br>
<a href="#unit2">Unit 2 </a><br>
<a href="#unit3">Unit 3 </a><br>
<a href="#unit4">Unit 4</a><br>
<a name="unit1 "></a>
<h3 align="center"> Indroduction </h3>
<p> Internet services </p><br>
<p> Protocols </p><br>
<p> Web concepts </P><br>
<p> Host names </p><br>
<a name="unit2"></a>
<h3 align="center"> html </h3>
<p> Sgml </p><br>
<p> Elements </p><br>
```

```
<p> Body section </p><br>
<p> Headers </p><br>
</body>
</html>
```



6) Embedding images

The “img” tag is used to add images on a webpage. The “img” tag is an empty tag, which means it can contain only a list of attributes and it has no closing tag.

Syntax :

```

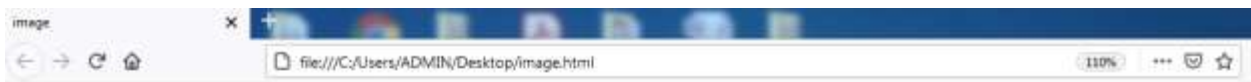
```

Attributes:

- Src
- Height
- Width
- Border
- Align
- Alt

Example

```
<html>  
  
<head>  
  
<title>image</title>  
  
</head>  
  
<body>  
  
<h1>welcom to the Units!!</h1>  
  
<center>  
  
  
  
</img>  
  
</center>  
  
</body>  
  
</html>
```



welcom to the Units!!



7) List

- A list is a record of short pieces of information, such as people's names, usually written or printed with a single thing on each line and ordered in a way that makes a particular thing easy to find.
- The types of lists that can be used in HTML are :

ul: An **unordered list**. This will list items using plain bullets.

ol : An **ordered list**. This will use different schemes of numbers to list your items.

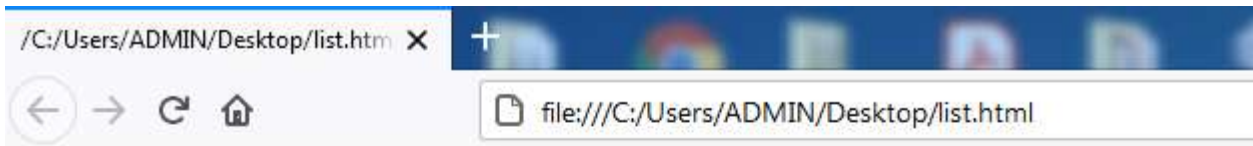
dl : An **definition list**. This arranges your items in the same way as they are arranged in a dictionary.

Unordered List

An unordered list starts with the “**ul**” tag. Each list item starts with the “**li**” tag. The list items are marked with bullets i.e small black circles by default.

Example

```
<html>
<body>
<h2>Unordered List with Circle Bullets</h2>
  <h2>Subject names</h2>
  <ul style="list-style-type:circle">
    <li>Computer Architecture</li>
    <li>Data Structure</li>
    <li>Java</li>
    <li>PHP</li>
  </ul>
</body>
</html>
```



Unordered List with Circle Bullets

Subject names

- Computer Architecture
- Data Structure
- Java
- PHP

Ordered List

An ordered list starts with the “ol” tag. Each list item starts with the “li” tag. The list items are marked with numbers by default.

Example

```
<html>
<body>
<h2>Unordered List with Numbers</h2>
  <h2>Subject names</h2>
  <ol type =”1”>
    <li>Computer Architecture</li>
    <li>Data Structure</li>
    <li>Java</li>
    <li>PHP</li>
  </ol>
</body>
</html>
```



Definition list

The definition list is created using `<dl>` element. The `<dl>` element is used in conjunction with the `<dt>` element which specify a term, and the `<dd>` element which specify the term's definition.

Example

```
<html>

<head>

<title>HTML Definition List</title>

</head>

<body>

<dl>

<dt><b>HTML</b></dt>

<dd>This stands for Hyper Text Markup Language</dd>

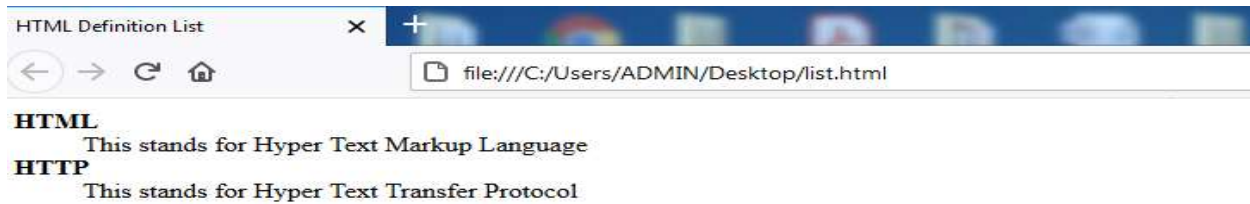
<dt><b>HTTP</b></dt>

<dd>This stands for Hyper Text Transfer Protocol</dd>

</dl>

</body>

</html>
```

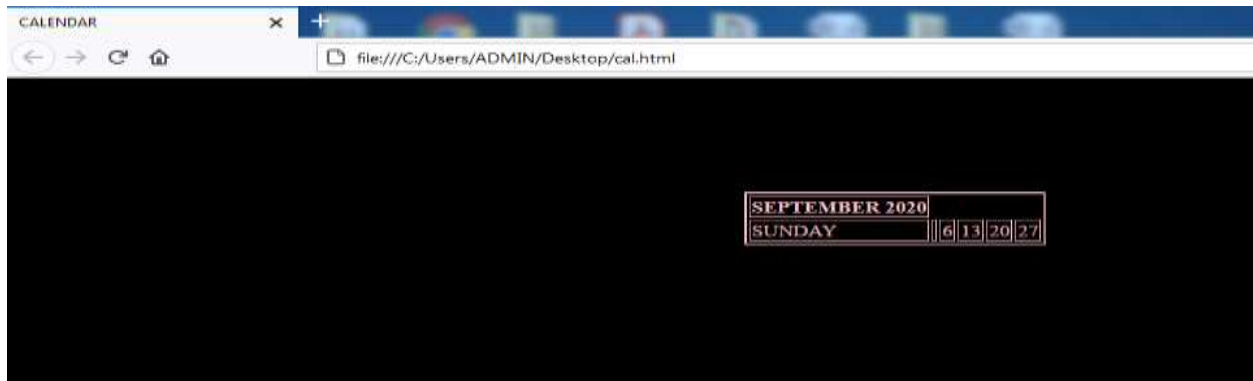


8) Tables

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

Example

```
<html>
<head>
<title>CALENDAR</title>
</head>
<body text="pink" bgcolor="black">
<table align="center" border="2">
<br><br><br><br><br><br>
<tr><th colspan="2">SEPTEMBER 2020</th></tr>
<tr><td>SUNDAY</td><td></td>
<td>6</td><td>13</td><td>20</td><td>27</td></tr>
</table>
</body>
</html>
```

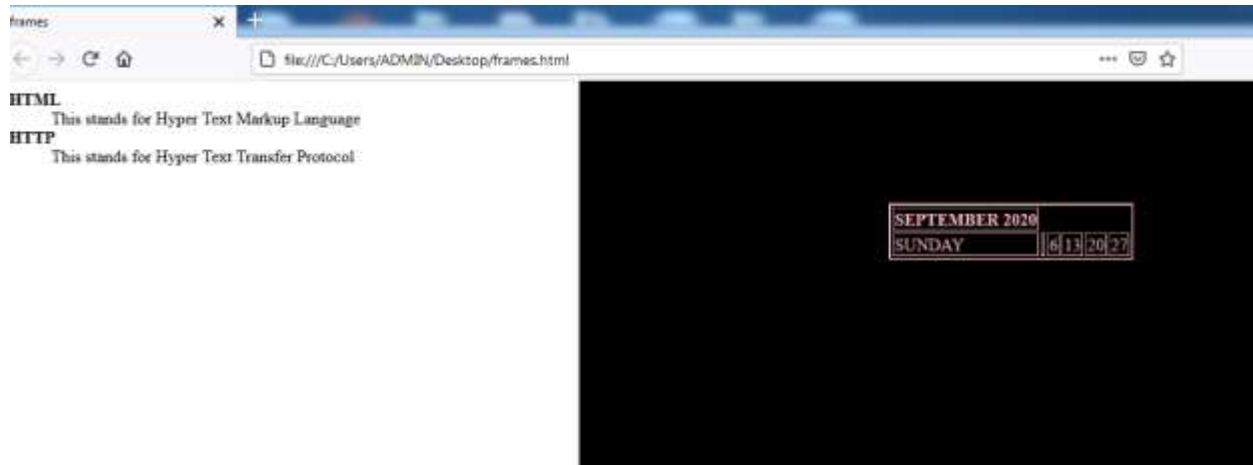


9) Frames

- HTML Frames are used to divide the web browser window into multiple sections where each section can be loaded separately.

Example

```
<html>
<head>
  <title>frames</title>
</head>
<frameset cols="40%,*">
  <frame src="list.html">
  <frame src="cal.html">
</frameset>
<noframes>
  <body>
    <p>Alternate content</p>
  </body>
</noframes>
</html>
```



10) Other special tags and characters

- Special characters are inserted in their code form.
- The tag `` is used to strike the text with a horizontal line.
- The tags `<sub>` and `<sup>` are used to the text into subscript and superscript respectively.
- The tag `<hr>` is used to draw a horizontal line in the web page. This `<hr>` tag has the following attributes:
 - 1) **Width**:- To specify the width of the line.
 - 2) **Size**:- To specify the height of the line.
 - 3) **No shade**:-Eliminates the default shading effect and instead displays the horizontal rule as a solid color bar.

Forms

- HTML Form is a document which stores information of a user on a web server using interactive controls.
- An HTML form contains different kind of information such as username, password, contact number, email id etc.
- The elements used in an HTML form are check box, input box, radio buttons, submit buttons etc. Using these elements the information of an user is submitted on a web server.

Example

```
<html>
<head>
<title> FORM </title>
</head>
<body>
<h1 align = center> form submission </h1>
<form action="/cgi-bin/submit.cgi" method="post">
<br>
  Enetr name:
  <input type="text" name="Enter"size=20>
  <br>
  Enter age:
  <input type="text"age="Enter"size=25>
  <br>
  sex: M
  <input type="radio" name="sex" value="M" CHECKED>
  F
  <input type="radio" name="sex" value="F">
  <br>
  Qualification:
  <select name="Qualification">
  <option> sslc </option>
```



```
<option> hsc </option>
```

```
</select>
```

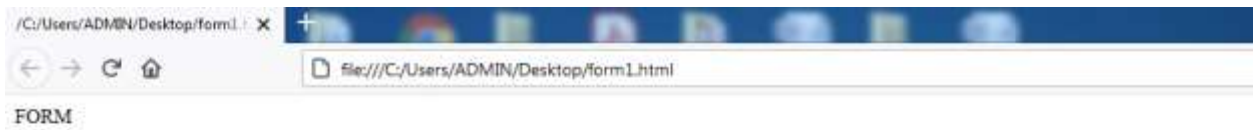
```
<br>
```

```
<input type="submit" value="login" onclick="submit()">
```

```
<input type="reset" value="clear" onclick="clear()">
```

```
</body>
```

```
</html>
```



form submission

Enter name:

Enter age:

sex: M F

Qualification: hsc

UNIT-III

Java Script

Introduction :

- JavaScript is designed to provide a quicker and simpler language for enhancing web pages and web servers.
- It was included to allow the client browser to generate HTML on the fly and execute JavaScript code resulting in a page that the user views or interacts with.

Need of a scripting language

- HTML provides more flexibility to web page authors, it is static.
- HTML documents have limited interactions with the users through HTML forms.
- A web page might use an HTML to get input and then set a parameter for java applet based on the input.
- Java script provides a fairly complete set of built-in functions and commands which help to perform mathematical calculations,manipulate strings,play sound,open new window and verify input to web forms.

Language Elements

1) Identifiers:-

- An identifier is a unique term that JavaScript uses to identify a variable, method, or an object
- The identifiers can take two forms
 - i) **Literals:-** literals are identifiers having fixed values.
 - ii) **Variable:-**variables have different values during execution.
- Literals and variables have several type supported in JavaScript. They are:

- i) Integers
- ii) Floating point numbers
- iii) Strings
- iv) Boolean

2) Expressions:-

- An expression is a statement that is evaluated to a value. The result can be of any type.
- Expression contains variables, literals, operators and even another expression.

Example:-

Y=67;

Str="Good Morning";

Val= x + y * b;

3) Operators:-

- Operators are commands which performs operations on variables to produce a result.
- There are two types of operators.
 - i) Unary:- it operate on only one operand
 - ii) Binary:- it operate on two operand

Types of operators:-

i) Assignment operators:-

- This operators takes the result of an expressions and assign it to a variable.

Shorthand	Expression	Description
$a += b$	$a = a + b$	Adds 2 numbers and assigns the result to the first.
$a -= b$	$a = a - b$	Subtracts 2 numbers and assigns the result to the first.
$a *= b$	$a = a * b$	Multiplies 2 numbers and assigns the result to the first.
$a /= b$	$a = a / b$	Divides 2 numbers and assigns the result to the first.
$a \% = b$	$a = a \% b$	Computes the modulus of 2 numbers and assigns the result to the first.

ii) Math operators:-

These operators include:

- + addition
- - subtraction
- * multiplication
- / division

These expressions are of the form $X = a + b$; $U = y * x / a$

iii) Comparison operators:-

- < less than
- > greater than
- <= less than and equal
- >= greater than and equal
- == equal
- != not equal
-

iv) **Logical operators:-**

There are three logical operators

- && - returns true only if both the operands or expressions are true
- || - returns true if one is true
- ! – used to negate the expression

v) **String operators:-**

The comparison operators can be used with the strings. for concatenating two strings ‘ + ‘ is used.

Str= “good” + “day”

4) Statements

i) **for loop** :- it consists of three expressions enclosed within brackets and separated by semicolon.

- Initial expressions:- initialize the loop variables. It is evaluated before the loop starts.
- Condition:- evaluated in each iteration and checks whether evaluated to true or false.
- Update expression:- changes the value of the loop variables.

Syntax:-

```
for ([initial];[condition];[update])  
{  
    Statements  
}
```

ii) **for in loop**:- this is a special form of loop which iterates the variable over all the properties in an object.

Syntax:-

```
for (var in obj)  
{  
    Statements  
}
```

iii) while and do....while loop

The while statement contains a condition and block of statements. if the condition is evaluated to true the block statements are executed. Otherwise the control goes to the statement following the while loop.

Syntax:-

```
While(condition)
{
    Statements;
}
do
{

    Statements;
} while (condition);
```

Break statement:-

The statement is used to terminate the current iteration in the while or for loop. If the break statement is executed the control break the iteration comes out the loop and start executing the statements following loop statement.

Syntax:-

```
break;
```

Continue statement:-

This statement is used to continue the loop for the next iteration without executing the statements following the continue statement.

Syntax:-

```
continue;
```

if...else statement:-

The if...else is a conditional statement. It executes the block1 statements if the condition is true. The else statement is optional whose block is executed if the condition is false.

Syntax:-

```
if(condition)
{
    block1
}
[ else
{
    block2;
}]
```

var statement:-

This statement is used to declare the variables.

Syntax:-

```
var name1 [=value], var name2 [=value],...
var x=0 , y=1;
```

Functions:-

- functions are used where the same operation has to be performed many times within a program or for understandability of the program by other users.
- Functions allows to have parameters or arguments and return the value to the statement that invoked the function.

Syntax:-

```
function name ( )  
{  
statement;  
statement;  
statement}
```


UNIT- IV

Objects of JavaScript

Introduction

- JavaScript is an object-oriented or object based language.
- An object is a set of variables, functions, etc., that encapsulate data and methods.
- Javascript also provides objects that encapsulate various attributes and methods in a script. Objects are grouped together and are given names.
- The objects are arranged into a hierarchy known as Document Object Model (DOM).

1) The Window Object:-

Window is the fundamental object in the browser. It represents the browser window in which the document appears. The properties include:-

Status:- The contents of the status bar. For example,

```
Window.status="hi";
```

Location:- The location and URL of the document currently loaded into the window.

`alert(window.location)`- will display an alert containing the location and URL of this document.

Length:- The number of frames into which the current window is divided.

`alert(window.length)`-will display an alert indicating the number of frames in the current window.

Parent:- The parent window, if the current window is a sub-window in a frameset.

```
var parentwindow=window.parent;
```

```
alert(parentwindow.length);
```

will place a string representing the parent window into the variable parentwindow, and all use it to report the number of frames in the parent window.

Top:- The top level window, of which all other windows are sub-windows.

```
var topWindow = window.top ;
```

```
alert (topwindow.length);
```

will place a string representing the top-level window into the variable topwindow, and will use it to report the number of frames in the top-level window.

Window Object Methods

Method	Description
alert()	displays an alert box with a message and an OK button
Confirm()	Displays confirm dialog box, containing text entered by the user, an OK button and a CANCEL button
close()	closes the current window
open()	opens a new browser window
prompt()	prompts for input

2) The Document Object:

The document object represents the HTML document displayed in a browser window.document properties include:-

bgColor- The colour of the background.

```
document.bgcolour="green";
```

background colour of the document to change to green.

fgColor- The color of the text.

```
Document.fgcolor="blue";
```

Colour of the text in the document to change to blue.

linkColor - standard link - to a page the visitor hasn't been to yet.

```
Document.linkcolor="brown";
```

Will change the colour of all the un-visited links in a document to brown.

alinkColor - active link - the color of the link when the mouse is on it.

```
Document.alinkcolor="lightred";
```

Will change the colour of active links in a document to lightred.

vlinkColor - visited link - to a page the visitor has been to before.

```
Document.vlinkcolor="darkred";
```

Will change the colour of all the visited links in a document to darkred.

Title- The title of the document, as displayed at the top of the browser window.

```
Document.title= "this title is displayed now";
```

Will replace the existing page title with . the text “ this title is displayed now”

Forms- An array containing all the forms in the document.

```
Forms(index-number)
```

Where index-number is the number of a particular form.

3) Form Object

When a form is created in an HTML document using <form> and </form> tags, a form object is created automatically with properties, methods, and events that relate to the form itself and to the individual elements within the form. Form properties include:

Name- name of the form ,as defined in the HTML <form> tag when a form is created.

```
<form name= "myform">
```

Method- The method used to submit the information in the form, as defined in the HTML <form> tag when the form is created.

```
<form method= "post">
```

Action- The action to be taken when the form is submitted, as defined in the HTML <form> tag when the form is created.

```
<form action= mailto:sales@bigco.com>
```

Length- The number of elements in the form.

```
Alert (document, forms[2].action);
```

Elements- An array of all elements in the form. Individual elements are referenced by index number.

```
Alert ( document.forms [2] .elements[0].name);
```

Form Methods

Submit ()- submits the form data to the destination specified in the action attribute using the method specified in the method attributes.

Form **events** include:-

Onsubmit () – message sent each time a form is submitted.

Text – Boxes and Text- areas

Each element within a form is an object in its own rights, and each has properties, methods and events that can be accessed using JavaScript. Text-box and text-areas **properties** include:-

Name – the name of the text-box or text-areas ,as defined in the HTML <input> tag when form is created. Example:-

```
<input type=text name="textbox1">
```

Value – the text typed into a text-box by the user. The text box is named textbox1.

```
Alert ( document.forms[2].textbox1.value )
```

Text-box and text-areas **events** include:-

Onfocus- Event signal generated when a user clicks in a text-box or text-area.

```
<input type = text name = "textbox2" onfocus = "alertonfocus()">
```

Onblur- Event signal generated when a user clicks outside a text-box or text-area having previously clicked inside it.

```
<input type = text name = "textbox3" onblur =  
"alertonblur()">
```

Buttons, Radio-buttons and Checkboxes:-

Buttons, Radio-buttons and Checkboxes **properties** include:-

Name- The name of the button , radio button or checkbox , as defined in the HTML <input > tag when the form is created.

```
<input type=button name = "button1">
```

Value- the value given to the button when it is created. On standard buttons the value is displayed as a label. On radio buttons and checkboxes the value is not displayed.

```
Document.forms[2].button1.value="new value, new label";
```

Checked- this property-which is used with a radio buttons and check-boxes but not standard buttons- indicates whether or not the button has been selected by the user.

```
if (document.forms[2].checkbox1.checked==true)
{
    Alert (“checked”)
}
Else
{
    Alert (“not checked”)
};
```

Buttons, Radio-buttons and Checkboxes **methods** include:-

Focus() – gives the button focus. This button is named button2 and has the value hello.

```
Document.forms[2].button2.focus();
```

Blur () – removes focus from a button.

```
Document.forms[2].button2.blur();
```

Click ()- simulates the effect of clicking the button.

```
Document.forms[2].button2.click();
```

Buttons, Radio-buttons and Checkboxes **events** include:-

OnClick – signal sent when the button is clicked. This can be used to call a function, and is probably the most frequently used of all the button function.

```
<input type= button name = “button3” value= “click here” onclick=”alert
(‘onclick event received’)”>
```

Onfocus- signal sent when the button receives the focus.

```
<input type= button name = "button4" value= "click here" onfocus="alert  
(‘this button is now the default’)">
```

Onblur- signal sent when the button loses focus.

```
<input type= button name = "button5" value= "click here" onblur="alert  
(‘this button is no longer the default’)">
```

The select object:-

Selection-boxes behave in a very similar fashion to radio-buttons – they present several options, of which only one can be selected at a time. they also have a similar set of properties methods and events.

Selectedindex- returns an integer indicating which one from a group of options has been selected by the user.

```
Alert ( document.forms[2].selectbox1.selectedindex);
```

Other objects:-

There are a number of other objects that form part of the javascript language

The Date object-

The date object allows us to obtain the current date and time , and to perform various timing operations.

```
var mydateobject= new date;
```

Methods	Description
getFullYear ()	Returns the current year as a four digit number.
getMonth ()	Returns the current month as an integer from 0-11
getDate ()	Returns the day of the month as an integer between 1 and 31
getDay()	Returns the day of a week as an integer between 0 and 6 starting from sunday
getHours()	Returns the hour of the day as an integer between 0 and 23
getMinutes ()	Returns the number of minutes since the beginning of the hour as an integer
getSeconds()	Returns the number of seconds since the start of the minute as an integer

The Math object:-

The math object allows us to perform various mathematical operations that are not provided by the basic operators we have already looked at.

Methods	Descriptions
sqrt (x)	Returns the square root of x.
log (x)	Returns the natural logarithm of x
max(x,y)	Returns whichever is the larger of x and y
min(x,y)	Returns whichever is the smaller of x and y
round (x)	Returns the value of x rounded to the nearest integer
ceil (x)	Returns the absolute value of x rounded up to the next integer value
floor (x)	Returns the absolute value of x rounded down to the next integer value
abs(x)	Returns the absolute value of x
pow (x,y)	Returns the value of x raised to the power y

The String object:-

Methods	Description
charAt()	Returns the character at a specified position in a string
indexOf()	Searches a string to see if it contains a specified character, and if it does, returns the position of the character.
lastIndexOf()	If the specified character occurs more than one in the string, it returns the position of the last occurrence of that character rather than the first
Substring()	Returns the portion of the string between two specified position
Substr()	Returns the portion of the string, starting from a specified position and continuing for a specified number of character.
Charcodeat()	Returns the numerical value of the character at the specified position
Fromcharcode()	Returns the characters represented by a sequence of numerical values
Tostring()	Converts a number into a string

Regular expressions

Methods	Description
\w	Represents any alphanumerical character
\W	Represents any non-alphanumerical character
\d	Represents any numerical character
\D	Represents any non- numerical character
\a	Represents any 'whitespace' character
\g	Represents any 'non-whitespace' character
[, ,]	Match any one of the characters within the brackets
[^, ,]	Match any one of the character other than those with in the brackets

UNIT- V

Dynamic HTML

Introduction:

- **DHTML** stands for **Dynamic Hypertext Markup language** i.e., **Dynamic HTML**.
- Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.

Components of Dynamic HTML

DHTML consists of the following four components or languages:

- CSS
- JavaScript
- DOM.

CSS

- CSS stands for Cascading Style Sheet, which allows the web users or developers for controlling the style and layout of the HTML elements on the web pages.

JavaScript

- JavaScript is a scripting language which is done on a client-side.
- DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements.
- The statements in JavaScript are the commands which tell the browser for performing an action.

DOM

- DOM is the document object model. It is mainly used for defining the objects and properties of all elements in HTML.

Uses of DHTML

- It is used for designing the animated and interactive web pages that are developed in real-time.
- DHTML helps users by animating the text and images in their documents.
- It allows the authors for adding the effects on their pages.
- It also allows the page authors for including the drop-down menus or rollover buttons.
- This term is also used to create various browser-based action games.
- It is also used to add the ticker on various websites, which needs to refresh their content automatically.

Cascading Style Sheets:

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External style sheets are stored in CSS files

Advantages of CSS

- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - Less code means faster download times.
- **Easy maintenance** - To change the style of an elements you only have to make an edit in one place.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML.

Disadvantages of CSS

Browser Compatibility: The two main browsers Netscape and Internet Explorer have varying levels of compliance with style sheets.

Coding CSS

A CSS declaration has two parts: 1) color 2) value

Syntax:

Selector { property 1: value1; property 2: value: 2 }

<head>

<style type= "text/css">

tag name {styleattribute:value;}

</style>

</head>

Properties of tags

- Different tags support different properties. All the text tags support the same set of text related properties.
- Properties can be assigned multiple values, sometimes the browser cannot comply with the first choice specified, so it uses the second choice. The general syntax for specifying a property value is

property-name:val1 val2.....valn;

Property Values

There are a number of different kind of values that are specified using style settings, including colors, length, numbers, percentage and URLs.

Other Style Property

Property	Description
background-attachment	Specify whether the background image is fixed in the viewport or scrolls.
background-color	Defines an element's background color.
background-image	Defines an element's background image.
background-position	Defines the origin of a background image.
background-repeat	Specify whether/how the background image is tiled.
color	Specify the color of the text of an element.
font-family	Defines a list of fonts for element.
font-size	Defines the font size for the text.
font-variant	Specify the font variant.
font-weight	Specify the font weight of the text.
text-transform	Transforms the case of the text.

Inline Style Sheet

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example

```
<html>
<body>
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
</html>
```

Embedded Style Sheet

Embedded style sheets refer to when you embed style sheet information into an HTML document using the <style> element. Embedding the style sheet information within <style> </style> tags in the head of your document.

Example

```
<style>

  p {

    font-family: georgia, serif;

    font-size: x-small;

  }

  hr {

    color: #ff9900;

    height: 1px;

  }

</style>
```

Grouping

The size of the style sheet can be reduced using grouping. One can group selectors in comma-separated lists.

Syntax

```
H1 { Font : bold 12pt/14pt Helvetica }
```

Inheritance

Inheritance is a process of receiving values of properties by a child element from its parent element.

Class as Selector

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

Example

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

ID as Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

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

Pseudo Classes and Pseudo elements

- Pseudo classes are like regular classes, but they are attached to the tag name with a colon, instead of a period. Pseudo classes are available with anchor tag.

- Pseudo code elements are used to address sub-parts of elements, while pseudo classes allow style sheets to differentiate between different element types.

A: {color: red} - unvisited link

A: {color: blue} - visited links

A: {color: lime} - active links

Positioning

There are two ways of defining the position of elements. These are

- Absolute positioning
- Relative positioning

Absolute Positioning

- An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

Example

```
<html>
  <head>
  </head>

  <body>
    <div style = "position: absolute; left: 80px; top: 20px; background-
color: yellow;">
      This div has absolute positioning.
    </div>
  </body>
</html>
```


Relative Positioning

An element with `position: relative` is positioned relatively with the other elements which are sitting at top of it. If we set its `top`, `right`, `bottom` or `left`, other elements will not fill up the gap left by this element.

Example

```
<html>
<head>
</head>

<body>
  <div style = "position:relative; left:80px; top:2px; background-
color:yellow;">
    This div has relative positioning.
  </div>
</body>
</html>
```

Backgrounds

- The **background-color** property is used to set the background color of an element.
- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The **background** property is used as a shorthand to specify a number of other background properties.

Example

```
<html>
<head>
</head>

<body>
  <p style = "background-color:yellow;">
    This text has a yellow background color.
  </p>
</body>
</html>
```

Element Dimensions

The dimensions of an element can be set using width attribute in the inline style.

Example

```
<html>
<head>
</head>
<body>
  <p style = "width:400px; height:100px; border:1px solid red; padding:5px;
margin:10px;">
    This paragraph is 400pixels wide and 100 pixels high
  </p>
</body>
</html>
```