



**STUDY MATERIAL FOR BCA
WEB TECHNOLOGY
SEMESTER - V, ACADEMIC YEAR 2020 - 21**



| UNIT | CONTENT | PAGE Nr |
|-------------|--|----------------|
| I | INTRODUCTION TO THE WEB | 02 |
| II | HYPER TEXT MARKUP LANGUAGE (HTML) | 09 |
| III | JAVA SCRIPT | 22 |
| IV | EXTENSIBLE MARKUP LANGUAGE (XML) | 31 |
| V | COMMON GATEWAY INTERFACE (CGI) | 37 |



UNIT – I

INTRODUCTION TO THE WEB

Understanding the Internet and World Wide Web

- The Internet is a global system that consists of millions of public, private, academic, business and government networks of local global scope.
- It allows all the computers connected to it to exchange information with one another.
- The standard TCP/IP (Transmission Control Protocol/Internet Protocol) suite is used by the Internet to serve millions of users worldwide.
- The information contained within the Internet can be accessed through a number of standardized interfaces.
- It is believed that approximately 30 million people worldwide are somehow connected to the WWW.
- The information is typically accessed as 'webpages', or HTML (Hyper Text Markup Language) documents.
- Webpages contain links (called hyperlinks), which allows us to access other web pages by simply clicking on them.
- Web pages are viewed using a software called browser. The first successful browser was Mosaic.
- Commonly used browsers as Microsoft's Internet Explorer, Netscape's Navigator, Mozilla's Firefox etc.

Protocols covering the web

- A protocol is a set of rules that define the syntax and semantics of the connected communication, and data transfer between two computing endpoints.
- The WWW has a body on such protocols.
- The TCP/IP suite of protocols is actually the set of protocols which is used to govern the web and to communicate across the Internet.
- TCP/IP has a set of layers each of which implements a specific set of protocols.
- The commonly used protocols are HTTP, SMTP (Simple Mail Transfer Protocol) and DNS (Domain Name System)

Creating websites for individuals and the corporate world

- A website is a set of related web pages, that are addressed with a common domain name or address.
- A website is hosted on a machine called web server, which should be accessible via a network.
- The web pages of a website are accessed using URLs (Uniform Resource Locator).
- The web pages are organized into a hierarchy.
- They contain hyperlinks that guide the visitors navigating the website.

Websites are organized depending on their function into two groups:

- ✓ Personal websites
- ✓ Corporate websites
- Personal websites contain web pages that are created by an individual.
- Each personal website contains information about a person or about something the person is interested in.
- Internet Service Providers (ISPs) provide a few megabytes of space to their subscribers for hosting their own personal web pages.



- A corporate website is developed and maintained by a company or other private enterprises such as charities or non-profit organizations.

Web Applications

- Web applications are applications that are accessed via web browsers usually through networks.
- Web applications run on a server called web server.
- Each web application is identified by the Uniform Resource Locator.
- There are two types of applications: Service oriented and Presentation oriented.

Service-oriented web applications:

These applications implement web services and are typically coded by using server-side technologies such as CGI, JSP, ASP, etc.

Presentation-oriented web applications:

- These applications are usually for the clients of service-oriented applications.
- They are coded in browser supported languages such as HTML, XML, JavaScript etc.
- Presentation oriented applications generate interactive web pages.

Identification of Objects

- The first step is to determine the objectives of your website.
- To identify them, it is a good idea to dissect the entire idea into more manageable tasks.
- Then ask a question to yourself, why are you going to develop this website?

Some typical reasons are as follows:

- ✓ Improve quality of the existing website
- ✓ Sell goods or service online
- ✓ Educate people or disseminate information
- ✓ Build customer relationship
- ✓ Provide an interface to book airline tickets
- ✓ Help travelers to plan a trip
- ✓ Make available product information and price lists to distributors
- ✓ Offer customer service, technical support, or other online services

Web Team

- A typical web application consists of a large number of components.
- Usually a number of people are involved in developing an entire application.
- They form a group called the web team.
- The roles and responsibilities are divided among the team members.

User Research

Identify users' needs, how they think and how they react/ behave and incorporate the understanding into every aspect of the process.

Technology Strategy

Identify the relevant technologies needed for this site – platforms, technologies, standards, and how they all can interoperate as websites are becoming more complex day by day.



Content Strategy

- Before proceeding the site content, you should be able to answer the following questions:
- Will the content meet users expectations?
- Is the content appropriate? What form should it take?
- What tone or style should it have?

Abstract Strategy

Have a conceptual framework from information architecture and interaction design for the final user experience.

Technology Implementation

Implementing the idea requires specialized knowledge about languages, protocols, coding, debugging, testing and maintenance.

Concrete Design

Identify the details of interfaces, navigation, information and visual designs

Project Management

Integrate all the above steps as well as control and drive the entire team towards the completion of the project.

Planning and Process Development

- Websites are developed by a group of experts called a web team to fulfill requirements of other groups of people.
- Members are key to successful web projects.
- To develop a moderate site, you need various experts such as content experts, information architects, technical experts, graphic designer, writers and finally a supervisor.
- There are numerous steps to be followed in the website design and development process starting from information gathering to the creation of a website and finally maintain the website.

Information Gathering

This is the first step of any web project.

The following things should be taken into consideration:

- Purpose
- Goals
- Target audience
- Content

Planning

- In this phase, the information gathered in phase1 is put together to develop a site map.
- A site map is a list main topics of the website and sub-topics.
- It helps you to develop the rest of the system easily

Design

- In this phase, we determine the look and feel of the website.
- Target audience and type of the website are major factors to decide the look and feel.



Development

- In this phase, the website is actually created with the help of relevant technologies.
- Web designers may use prototypes before developing the actual content.
- Typically, the development starts from the home page followed by a skeleton for interior pages.

Testing and Delivery

- Before the website is delivered, it is tested extensively.
- It includes testing of complete functionality of scripts and forms and other elements.
- You should also deal with compatibility issues, optimization.

Maintenance

- Website development is a continuous process
- You can provide newer and newer content, update information or use different look and feel from time to time to attract your visitors

Web Architecture

The WWW is a two-tier architecture consisting of web servers, which produce and deliver information, and web clients, which retrieve and display information.

Three primary concepts are involved in this process:

- ✓ HyperText Markup Language (HTML)
- ✓ Uniform Resource Locator (URL)
- ✓ HyperText Transfer Protocol (HTTP)

HTML

- HTML is the primary language used to encode documents containing hyperlinks.
- Its first public version was released in 1991 by Tim Berners Le, the inventor of the WWW.
- HTML is an application of SGML, a standard that specifies a formal meta-languages for markup documents.
- HTML documents consist of case-insensitive.
- Document contents are embedding in starting and ending tags
- In additional documents are linked by special tags called anchor tags
- These anchor tags are also called hyperlinks.
- HTML allows us to embed virtually all kinds of data such as text, images, audio, video, etc.
- HTML documents are viewed by a software called browser.
- When a user selects a hyperlink the web browser retrieves the document and displays it on the screen.

URL

- The WWW is a repository of information called resources.
- URLs are used to address these resources
- URL is actually a special case of the general addressing protocol.
- Uniform Resource Identifier, URLs are a location independent addressing mechanism used on the WWW.

HTTP

- HTTP is an application layer protocol for the WWW
- HTTP is a request-response protocol.
- It basically consists of the following steps:
- HTTP client establishes a TCP connection to the HTTP Server



- Client sends an HTTP request to the server specifying the resource it wants to access.
- Server sends as HTTP response containing the desired information.

TCP/IP Protocol Suite:

- The TCP/IP protocol suite is a set of protocols used in the Internet and other communication networks.
- It creates the technical foundation of the Internet.
- There are two primary protocols
- Transmission Control Protocol (TCP)
- Internet Protocol (IP) but it consist of many other protocols
- The TCP/IP protocol suite consists of several layers each of which deals with a set of issues that arise while designing a communication network.

There are four layers in a TCP/IP protocol suite:

- ✓ Link layer
- ✓ Internet layer
- ✓ Transport layer
- ✓ Application layer

Link layer

- This is the lowermost layer in the TCP/IP protocol suite.
- Link layer deals with Local Area Network (LAN)
- It is responsible for transferring data from one machine to another within the same local network.

Following are the primary functions of the Link layer:

- ✓ Framing
- ✓ Error Detection
- ✓ Error Correction
- ✓ Flow Control
- ✓ Error Control
- ✓ Medium Access Control

Internet Layer

- The link layer cannot deliver packets across different Local Area Networks (LANs)
- The primary task of the Internet layer is to deliver packets from one network to another.
- It determines a suitable route from the source computer to the target and delivers packets through this route. This process is called routing

Transport Layer

- The task of transport layer is end-to-end message transfer.
- It provides two types of service to its upper layer, connection-oriented – where the segments of a message are delivered in order and connection-less where no such guarantee is given.
- The transmission control protocol (TCP) in a TCP/IP protocol sutie provides connection-oriented service, whereas User Datagram Protocol provides connection-less service.

It should also implement the following optional tasks:

- ✓ Segmentation
- ✓ Flow Control
- ✓ Error Control



- ✓ Congestion Control

Application Layer

- Application Layer is the top-most layer in the TCP/IP protocol suite.
- This is the layer that users actually interact with.
- It provides interfaces to users for network communication.
- Following is a list of some other protocols it implements:
- HTTP, SMTP, Telnet, SSH, DNS, POP3, rlogin, SSL

IP Addresses

- Internet Protocol in the TCP/IP protocol suite assigns a unique address called IP Address to each device participating in a computer network.
- An IP address servers basically two types: identifying a host/network interface and location addressing.
- According to Internet Protocol version 4 an IP address is a 32-bit binary number.
- IP address are usually written using 4 decimal numbers each representing 8 bits in the range 0-255, separated by three dots (.)
- This notation is known as dotted decimal notation 203.197.14.112

Every IP addresses are categorized into five class as follows:

- ✓ Class A – Address starts with 0, Possible values of octet are 1 to 127
- ✓ Class B – Address starts with 10, Possible values of octet are 128 to 191
- ✓ Class C – Address starts with 110, Possible values of octet are 192 to 223
- ✓ Class D – Address starts with 1110, Possible values of octet are 224 to 239
- ✓ Class E – Address starts with 1111, Possible values of octet are 240 to 255
- ✓ Class D and E are reserved for research purpose and are not used.

IPv4 Private Addresses

- In early network design, IP address assigned to particular computers or network devices were intended to be unique for global end-to-end communication
- Computers not connected to the Internet, such as factory machines that communicate only with each other via TCP/IP, need not have globally unique IP addresses.

MIME

- The Simple Mail Transfer Protocol (SMTP) is an important and frequently used protocol in the TCP/IP protocol suite.
- It is used to deliver Electronic mails (E-mail).
- SMTP was designed to transfer only text messages.
- Multipurpose Internet Mail Extension (MIME) extends the format of e-mail to support the following:
 - ✓ Text in character sets other than ASCII
 - ✓ Non-text attachment
 - ✓ Message bodies with multiple parts
 - ✓ Header information in non-ASCII characters sets.

Cyber laws

- The WWW evolved in a completely unplanned and unregulated manner.
- The internet is growing drastically with the number of users doubling roughly every 100 days.



- New and sensitive issues related to various legal aspects of cyberspace began cropping up.
- In this law, cyber crime is unlawful act where a computer is either a tool or a target or both.
- Cyber crimes involve traditional criminal activities such as theft, fraud, forgery, defamation, and mischief.
- Cyber laws regulate these crimes.

Cyber crimes can be categorized into two types:

Using computer as a tool

- In this type of crime, computers are used to attack other computers.
- It including hacking /hijacking, virus/worm attacks.
- DOS attacks, denial of service attacks, Trojan attacks, etc.

Using computers as a weapon

- In this type of crime, computers are used as a weapon.
- Examples include credit card frauds email spoofing, emails spamming, email bombing.
- Cyber terrorism, EFT frauds, pornography, IPR violations, etc.



UNIT – II

HYPER TEXT MARKUP LANGUAGE (HTML)

History of HTML and W3C

- The primary scripting language for developing webpages in HyperText Markup Language (HTML)
- It is used to describe how a web page will appear in a browser's window.
- HTML provides simple mechanisms for formatting text, creating links and lists, inserting images, embedding audio and video etc.
- It can also include Cascading Style Sheets (CSS) to specify the style and layout of txt and other components.
- Scripts written in various languages such as JavaScript and VBScript may also be used to change the appearance of web pages and make them interactive.
- HTML documents are written using HTML tags embedded in angular brackets.
- The set of tags that we can use to write HTML documents is defined by the World Wide Consortium (W3C).
- W3C has published many standards for HTML. Their current version is 4.01.

HTML and its Flavors:

- The first public version of HTML specification called HTML tags, was introduced in late 1991 by Berners-Lee. In this version, only 20 tags were incorporated to develop simple websites.
- In November 1995, HTML version 2.0 was introduced by Internet Engineering Task Force (IETF).
- The World Wide Web Consortium (W3C) published its first recommendation, HTML 3.2, in 1997.
- In the same year, W3C published HTML version 4.0, which consists of three flavors: Strict, Transitional and Frameset.
- In 1999, W3C published an enhanced version. HTML 4.01, which was modified in May 12, 2001.
- This current version is supported by most of the web browsers and is widely used.

HTML Basics:

- An HTML document itself is a text file that contains text and markups called tags.
- So, any simple text editor such as Notepad in Windows or Simple Text in Macintosh can be used to create and edit HTML files.
- Several HTML editors are also available in the market. Most of these editors provide a GUI, which the users use to design their web pages.
- All HTML tags in all about tags. Tags are similar to instructions or codes.
- Web browsers know their meaning and interpret them to render text, images, and other elements on the screen properly.
- The set of HTML tags is fixed and is standardized by W3C. Invalid tags are displayed on the screen.

Elements, Attributes, and Tags:

HTML Elements:



HTML elements are the fundamental building blocks of a web page. An element consists of a tag, its attributes, and content.

HTML tags:

Tags are codes each of which marks up a certain region in an HTML document. A tag is written within angular brackets (< and >). The general format of a tag is as follows:

`<tag> tag content </tag>`

Here `<tag>` is the opening tag and `</tag>` is the closing tag

`<i> italic </i>` -- This tag is called an embedded tag. Each embedded tag has both opening and closing tags.

There are some tags that do not have their corresponding closing tags. These tags are called standalone tag. For example of standalone tag is: `<hr>`

Attributes:

- Tags may have properties that can optionally be assigned values to change the default behavior of these tags. These properties are called attributes. Attributes are placed within the starting tag. For example: `<hr width="400">`
- This tag represents horizontal rule whose length is specified by the attribute width as 400 pixels. Multiple attribute can be specified separated by white space(s)

`<hr width="400" color="red">`

Basic Structure:

- Every HTML document starts with the `<html>` tag.
- The `<html>` tag does not have any properties. Everything within opening `<html>` and closing `</html>` tags is parsed by the browser.
- An HTML page has basically two distinct logical sections:
- head section specified by `<head>` and `</head>` tags
- body section specified by `<body>` and `</body>` tags. The skeleton of a HTML page looks like this:

`<html>`

`<head>`

`</head>`

`<body>`

`</body>`

`</html>`



Common HTML tags and their functionality

| Tag | Meaning |
|---------------------|---|
| <html>....</html> | Root tag of any webpage |
| <body>...</body> | Represents the body of an HTML document |
| <head>....</head> | Represents the header of a web page |
| <p>.....</p> | Starts a new paragraph |
| | Inserts a line break |
| | Makes the enclosed text bold |
| <i>.....</i> | Makes the enclosed text italic |
| <u>.....</u> | Makes the enclosed text underlined |
| <hr> | Inserts a horizontal rule |
| <table>....</table> | Inserts a table |

Common HTML tags placed in head tag

| Tag | Meaning |
|-----------------------|--|
| <title>.....</title> | Makes the enclosed text the title on the web page. |
| <link> | Makes a link between an external source and this html file |
| <script>....</script> | Contains scripts such as JavaScript and VBScript |
| <base> | Specifies base URL of the document |

Common attributes of body tag

| Tag | Meaning |
|------------|---|
| bgsColor | Specifies background color of the document |
| background | Specifies background image of the document |
| link | Specifies the color of a not yet visited link |
| alink | Represents the color of an active link |
| vlink | Represents the color of a visited link |
| Text | Specifies the color of the enclosed text |

Basic Tags:

HTML Comments:

An HTML comment is not really a tag. It starts with <!-- and ends with -->. Everything within these character sequences will be ignored by the browsers and will not be parsed. The general syntax : <!-- comment text -->

Adding a title:

The <title> and </title> tags represent page titles. Their content is displayed on the top of the browser window for example: <title> My Home Page </title>



First HTML page:

HTML document that displays a simple text “Hello World” on the screen

```
<html>
  <head>
    <title> My First html page</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```

HTML Background:

The background of an HTML page is specified in the <body> tag.

Using plain color:

- To set a plain color as the background use the bgcolor attribute of the <body> tag
- For example: <body bgcolor="yellow">

Using Image:

- To use an image as background, we use the background attribute of the <body> tag
- For example: <body background="Window_XP.jpg">

Paragraph:

- HTML provides a paragraph tag <p> which is used to start a fresh paragraph. This tag inserts a blank line immediately before the new paragraph
- <p>This is one sentence paragraph</p>

Inserting Line Breaks:

- The tag
 is used to insert a line break.
- For example: This is first line
 This is second line

Bold Text:

This tag makes the enclosed text bold.

For example : welcome

Underlined Text:

- This tag <u> makes the enclosed text underlined.
- For example : <u>welcome</u>

Italicized Text:

- This tag <i> makes the enclosed text italic.
- For example : <i>welcome</i>

Heading:

- There are total six levels of headings h1 through h6. The h1 heading has the largest and h6 has the smallest font size.

```
<h1>heading1</h1>
```



<h2>heading2</h2>

<h3>heading3</h3>

<h4>heading4</h4>

<h5>heading5</h5>

<h6>heading6</h6>

Lists:

A list is a collection of one or more items. HTML supports three types of lists for formatting text:

- ✓ Unordered (bulleted)
- ✓ Ordered (numbered)
- ✓ Definition lists

Unordered lists:

An unordered list is created using the tag. Items in the list are created using the tag and are displayed using bullets.

unordered list

ordered list

definition list

Bullet options

| Bullet options | Meaning |
|--------------------|--------------------------------|
| | Default bullet shape is disc |
| <ul type="disc"> | Disc shaped bullets are used |
| <ul type="circle"> | Circle shaped bullets are used |
| <ul type="square"> | Square shaped bullets are used |

Ordered Lists:

In this type of list, items are displayed using numbers. The list is created using the tag and items are created using the tag

unordered list

ordered list

definition list



Number options:

| Numbering options | Meaning |
|-------------------------|-------------------------------------|
| | Use default number type |
| <ol type="1"> | Use default number type |
| <ol type="A"> | Use capital letters for numbering |
| <ol start="3"> | Use default number starting from 3 |
| <ol type="A" start="4"> | Use capital letters starting from D |

Definition Lists:

- A definition list is one where list items consist of two parts: a term and its description.
- It is created using the <dl> tag. The term part and the definition part of each item are created using <dt> (definition term) and <dd> (definition description) tags.

<dl>

<dt> dl tag</dt>

<dd>It is the outermost tag of definition</dd>

<dt>dt tag</dt>

<dd>contains item to be described</dd>

<dt>dd tag</dt>

<dd>Contains description of the item</dd>

<dd>Each term have multiple description</dd>

</dl>

Advanced Tags:

Table:

- A table is created using three tags: <table> <tr> and <td>. The tag table is the container of the whole table.

<table>

</table>

HTML table tags

| Tag | Meaning |
|---------|----------------------------|
| table | Represents the whole table |
| tr | Represents a row |
| td | Represents a cell in a row |
| th | Column header |
| caption | Title of the table |



- A table consists of a number of rows each of which is made up of a number of cells. The tag `<tr>` creates a row.
- The opening tag `<tr>` and closing tag `</tr>` respectively

`<table>`

`<tr>.....</tr>`

`<tr>.....</tr>`

`</table>`

- A table or a row cannot contain data. To add data, cells are created within a row. Each cell in a row is defined by the tag `<td>`. Usual tags such as `<p>`, `
`, `<i>`, `` may be used within a cell

`<table>`

`<tr><td>....</td><td>....</td><tr>`

`<tr><td>....</td><td>....</td><tr>`

`</table>`

Border:

- Adding a table border is very simple. A value 0 (zero) indicates no border

`<table border="1">`

`<tr><td>33</td><td>55</td></tr>`

`<tr><td>66</td><td>77</td></tr> </table>`

Row and column header:

- The tag `<th>` is used to add row/column headers. They are also added using cells. Rows/column headers are center-aligned and in bold font, which distinguishes them from other cells.

`<table border="1">`

`<tr><th>Name</th><th>Place</th><tr>`

`<tr><td>Balan</td><td>Chennai</td></tr>`

`</table>`

Rowspan and colspan:

- To make a cell span multiple rows and columns, the tags `<rowspan>` and `<colspan>` are used. The tag `<rowspan>` indicates the number of rows a cell spans while `<colspan>` indicates the number of columns a cell spans.

`<table border="1">`

`<tr><th rowspan=2>Name</th>`

`<th colspan=2>Marks</th></tr>`

`<tr><th>OS</th><th>OOP</th></tr>`

`<tr><td>Sunjay</td><td>88</td><td>77</td></tr></table>`



Cellspacing and cellpadding

These attributes are used to adjust white spaces in your table. The attribute cellspacing specifies the amount of spaces, in pixels, between adjacent cells. Similarly, cellpadding specifies the amount of spaces in pixels between the cell border and its content.

```
<table border=cellspacing="6" cellpadding="3" width=300>
<tr><th rowspan=2>Name</th>
<th colspan=2>Marks</th></tr>
<tr><th>OS</th><th>OOP</th></tr>
<tr><td>Sunjay</td><td>88</td><td>77</td></tr></table>
```

Attributes of the table tag

| Attribute | Meaning |
|-------------|---|
| align | Specifies table alignment, can have values: left, right and center |
| background | Specifies the URL of the background image |
| bgcolor | Specifies the background color |
| cellpadding | Specifies the amount of space in a number of pixels between the cell border |
| cellspacing | Specifies the amount of space in number pixels between table cells |
| height | Specifies the height of the table in pixels |
| width | Specifies the width of the table in pixels |

Attributes of the tr tag

| | |
|-------------|--|
| align | Specifies the horizontal alignment of the content of all cells in this row |
| bgcolor | Specifies the background color of all cells in this row |
| bordercolor | Specifies the bordercolor of all cells in this row |
| height | Specifies height, in pixels, of all cells in this row |
| width | Specifies width, in pixels, of all cells in this row |

Forms:

- The HTML <form> tag is a very important tag, which creates a section in your HTML document.
- It is used to collect information from visitors. In addition to usual contents and markups, it contains special elements called controls.
- HTML provides several control elements such as buttons, text boxes, password fields, check boxes, radio buttons, selection boxes hidden fields etc.
- The tag <form> is a container tag. <form> ---- </form>
- The tag <form> has another optional attribute method which specifies the HTTP method to be used. It can have values GET or POST.
- IF GET (the default) is used, the data are appended to the URL



- IF the POST method is used, data are sent as a part of the HTTP request message and are not appended to the URL.

```
<form action="handler.jsp" method="post">
```

```
</form>
```

Attributes of the input tag

| Attribute | Meaning |
|-----------|---|
| type | Specifies the type of a control element |
| name | Specifies the name of a control element |
| value | Specifies the initial value of the control element |
| checked | Used for checkbox and radio buttons, indicates that the controls is on. |
| readonly | Used for text, password, and file controls |
| disabled | It set disables controls from the user input |
| maxlength | Specifies the maximum number of characters a user can enter in a text or password |

Text field:

- The most commonly used input element is text field. It is used to get single-line textual data.
- Its common usage include name, date of birth, address, email address, login name etc.

Name <input type="text" name="name">

Password field:

Password fields are similar to text fields except that characters entered are displayed as dots or asterisks. This allows us to hide information from others. All the attributes of a text field are also valid for a password field.

Password: <input type="password" name="pass">

Label:

The content of the <label> tag is a piece of ordinary text. It is used to add a label to a form field.

```
<label for="married">
```

Married

```
<input type="checkbox" id="married">
```

```
</label>
```

Check box:

A checkbox is like a toggle switch i.e., it can be either of the two states checked or unchecked. Checkboxes allow visitors to select one (or more) options from a set of related alternatives. A checkbox is created using the <input> tag and specifying the type="checkbox"

Which of the following items do you have?


```
<input type="checkbox" name="c1" values="yes">Car<br>
```



`<input type="checkbox" name="c2" value="yes">Bike`

Radio button

- Radio buttons allow visitors to select any one among a set of related alternatives. When one radio button in a group is selected, the previously selected radio button (if any) is unselected.
- Radio buttons are unchecked by default. A checkbox is created using the `<input>` tag and by specifying `type="radio"`

Gender:`
`

`<input type="radio" name="gender" value="Male">Male
`

`<input type="radio" name="gender" value="Female">Female`

Selection list:

- A selection list (also called the drop-down list), like a group of radio buttons or checkboxes, allows visitors to choose one (or more) options among a set of options.
- A selection list is created using the `<select>` tag. Items of a selected list are created using the `<option>` tag. By default, the first option is displayed.

Gender:

`<select>`

`<option>Male`

`<option>Female`

`</select>`

Text area:

- A text area is an extension of a text field, where the text can span multiple lines. It is used when visitors want to enter a large amount of text.
- The number of rows and columns may be set using the attributes `rows` and `cols` respectively.

Comment:`
`

`<textarea rows="4" cols="20">`

Enter your comment here

`</textarea>`

Button:

Push Buttons can be created using the `<input>` tag, with the `type` attribute set to `"button"`. The caption is the value of the `value` attribute of the `input` tag.

`<input type="button" value="Logout">`

Image button:

- The submit button in a form has a gray colored simple appearance, unless you use styles for it.
- A simple way to create a good looking graphic submit button is to use the `<input>` tag with attribute set to `"image"`

`<input type="image" src="rightarrow.gif">`

Frames



HTML allows us to divide a web page into several blocks called frames. Each frame may display a separate web page. So, frames allow us to display multiple HTML documents in one browser window simultaneously. They are refreshed separately.

For example, the following code creates two frames separated vertically

```
<frameset cols="40% 60%">
<frame name="left" src="left.html">
<frame name="right" src="right.html">
</frameset>
```

Frameset Element:

- The layout of a document is specified using the <frameset> element.
- The <frameset> element is put in the parent HTML document without any <body> tag.
- It contains <frame> elements each of which creates a frame. For example, the frameset element vertically divides the page into two blocks (frames).
- The left frame is set to 40% of the total page width and the right frame is set to 60% of the total page width.

Frame Element:

The element <frame> specifies the HTML document to be loaded in each frame. It has several attributes as follows:

- ✓ name – refers to the name of the frame
- ✓ src – specifies the location of the initial web page to be loaded.
- ✓ frameborder – specifies the border information about the frame
- ✓ marginwidth – specifies the amount of space of pixels to be added between the frame's contents in its left and right margins.
- ✓ marginheight – specifies the amount of space in pixels to be added between the frame's contents in its top and bottom margins.

Layout:

Each frameset creates a set of rows or columns

```
<frameset rows="30%, 70%">
<frame name="top" src="top.html">
<frame name="bottom" src="bottom.html">
</frameset>
```

- This frameset element divides the page horizontally into two frames. The top frame is set to 30% of the total page height and the bottom frame is set to 70% of the total page height.
- The attribute of frameset element
- cols – This attribute is used to create frames vertically
- rows – This attribute is used to create frames horizontally

Nested Frame:

Nested frames are also possible



```
<frameset rows="20%, 80%">
```

```
  <frame name="top" src="top.html">
```

```
  <frameset cols="40%, 60%">
```

```
    <frame name="bottom-left" src="bottom-left.html">
```

```
    <frame name="bottom-right" src="bottom-right.html">
```

```
  </frameset>
```

```
</frameset>
```

- This HTML code first divides the window into two rows.
- The first row is given a name top and is set to 20% of the height of the browser window and the second row is set to 80% of the height of the browser window.
- In the first row, the HTML document top.html is loaded. The second (bottom) row is further divided into frames vertically.
- The first column of the bottom row is given a name bottom-left and is set to 40% of the width of the bottom frame and the second column is given a name bottom-right and is set to 60% of the width of the bottom frame.
- The HTML documents bottom-left.html and bottom-right.html are loaded in the bottom-left and bottom-right respectively.

Specifying Target:

The target attribute may be set for elements that create links such as <a>, <link>, <form> etc. The value of target attribute refers to the frame where the document is to be uploaded.

First we define a framset document main.html as follows:

```
<html>
```

```
<frameset cols="20%, 80%">
```

```
<frame name="left" src="left.html">
```

```
<frame name="right" src="right.html">
```

```
</html>
```

This divides the window into frames vertically. In the left frame HTML document left.html is loaded. It looks like this

```
<html>
```

```
  <head> <title>Target demo</title></head>
```

```
  <body>
```

```
    <a href="http://www.google.com" target="right">Google<br>
```

```
    <a href="http://www.yahoo.com" target="right">Yahoo
```

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

Inline Frame:

- HTML inline frames (also called iframes or floating frames) allow us to insert a frame even within a block of text.



- Iframes may contain external objects including other web pages. An iframe is created using the <iframe> tag.

The following code

```
<iframe src=http://www.yahao0.com>
```

Your browser does not support iframe

```
</iframe>
```



UNIT - III **JAVA SCRIPT**

History of JavaScript

- The first scripting language was first introduced by Brendan Eich, a member of the Netscape 2 group, in early 1996 to provide dynamicity and interactivity to web pages.
- The first version capable of being embedded in web pages, could only process numbers and modify HTML form contents.
- In 1997, the ECMAScript standard was updated and Netscape 4 released a modified version, JavaScript 1.2.
- Netscape, in mid 1998, also released a new version, JavaScript 1.3, based on their layered DOM.
- In 1999, Microsoft introduced an updated version of Jscript (equivalent to JavaScript 1.5) with much greater functionality, as a part of Internet Explorer.

Inserting JavaScript Code:

- To insert JavaScript code is to embed it directly within the <script> and </script> tag pair
- The <script> tag notifies the browser that everything contained within is script and is to be interpreted by a suitable interpreter.

<script language="JavaScript">

JavaScript code

</script>

- The optional attribute language specifies the programming language used for scripting.
- There are many versions of each scripting language. For example. JavaScript has versions 1.0, 1.1, 1.2, 1.3 and so on

Advantages:

- This makes the HTML file neater and cleaner. Too much JavaScript code makes it large and unreadable.
- The same JavaScript source file can be reused in many HTML files. Common function can be placed in a source file, instead of inserting them in each HTML file that used them. Any changes to the source file reflect immediately in all HTML files that refer it.
- The JavaScript code remains hidden if the browser is not compatible with it.

First JavaScript Program

<html>

<head> <title> First JavaScript Program </title></head>

<body>

<script language="JavaScript">

document.writeln("Hello World");

</script>

</body>

</html>

Adding Comments:

- Adding comments to programs is considered to be a good practice.
- Comments are used for documentation. They help readers to understand the meaning of the code.
- Comments are of two types: line comment and block comment.



- A line comment starts with //
- Line comments are useful for comments that are smaller in size.
- A block comment starts with /* and ends with */ For example

/*

This is a block comment

This will also be ignored

*/

Placement of JavaScript Code:

- JavaScript code may be placed anywhere in an HTML file.
- It is interpreted in the order in which it appears. HTML tags are also rendered according to their document order. A function must be declared before it is called. This is true for variables as well.
- This suggests us to write JavaScript functions and declare global variables within the head section, which is rendered before the body section.
- Other JavaScript statements, including function calls, can be placed anywhere within the body tag.
- This ensures that a function is called after it is defined. This also ensures that a variable is used after it is created.

The following code:

```
<html>
<head>
<title> My first JavaScript Program</title>
<script language="JavaScript">
// Function declaration
function hello {
    document.writeln("Hello world");
}
</script>
</head></body>
<script language="JavaScript">
//Function call
Hello();
// other JavaScript statements
</script>
</body>
</html>
```

JavaScript Keywords

| | | | | | |
|----------|---------|--------|----------|----------|---------|
| abstract | boolean | break | byte | case | Catch |
| char | class | const | continue | debugger | Default |
| delete | do | double | else | enum | Export |



**STUDY MATERIAL FOR BCA
WEB TECHNOLOGY
SEMESTER - V, ACADEMIC YEAR 2020 - 21**



| | | | | | |
|------------|---------|-----------|------------|--------------|--------|
| extends | false | final | finally | float | For |
| function | goto | if | implements | import | In |
| instanceof | int | interface | long | native | New |
| null | package | private | protected | public | Return |
| short | static | super | switch | synchronized | This |
| throw | throws | transient | true | try | Typeof |
| var | void | volatile | while | with | |

Variables:

- Programming means manipulation of data. Data are stored in variables.
- JavaScript like other programming languages, allows us to create and manipulate variables.
- A variable can be a local (to a function) or global.
- A local variable is declared in a function using the keyword var.
- Variables declared without the var keyword is always global.
- Variables declared using the var keyword in the outermost scope are also global variables.

Naming convention:

- The name of a variable cannot be a keyword.
- It must start with a letter or an underscore (_) followed by zero or more letters, underscores, or digits.
- It can never contain white spaces, or any punctuation characters such as , (comma) . (full stop) etc.
- JavaScript, unlike HTML is case sensitive.

Local Variables:

Local variables are declared within a function or a block. The general syntax:

var variable = value;

for example

var sum = 0;

var month = "april";

Global Variables:

- A global variable is created without the keyword var.
- It is created by assigning a value to it. The general syntax:

variable = value;

for example

x = 1;

flag = true;



Functions and Global Variables:

- A global variable is created using the keyword var. Such a declaration can be placed anywhere within the script.
- It is also possible to make such declarations within a function.
- The lifetime of this global variable starts as soon as the function is called and the variable is created.

For example:

```
Function avg(a, b, c) {
```

```
Sum = a + b + c;
```

```
Average = sum / 3;
```

```
}
```

```
avg (2, 3, 1);
```

typeof Operator:

- The type of a variable is determined by its content

```
var x = 0;
```

```
var msg = "Hello world";
```

```
var found = false;
```

```
var month = "april";
```

- The type of a variable can be determined at any instant of time by using the type of operator.
- The type of operator returns a string describing the type of a value.

Table below shows the values of the typeof operator on applying different values.

| type | typeof | type | typeof |
|----------|------------|-----------|-------------|
| Object | 'object' | String | 'string' |
| Function | 'function' | Boolean | 'boolean' |
| Array | 'object' | Null | 'object' |
| Number | 'number' | Undefined | 'undefined' |

Literals:

JavaScript supports the following data types:

- String
- Number
- Boolean
- Function
- Object

String data type:

- A string value consists of a sequence of zero or more Unicode characters(e.g. alphabets, numeric, punctuation marks, etc).



- A string literal is a string value written in a matching pair of double quotation ("") or single quotation (') marks. For example

"Hello world"

'JavaScript Programming Language'

Numeric data type:

- A variable of type number may hold an integer or a real number.
- JavaScript does not differentiate between an integer and a real number.

Integers:

- Integers may be written either in decimal hexadecimal, or octal notation.
- Hexadecimal and octal numbers cannot have any fractional part.

Example of decimal integer literals are:

2, 100, 0, -1, -200

- A hexadecimal integer literal starts with "0x" or "0X". the base number is 16
- These numbers may contain digits as well as alphabetic characters A through F either in uppercase or in lowercase.
- The character A represents the decimal number 10, B represents the decimal number 11, and so on.

For example

0x4, 0xFF, 0xA2

- Octal integers start with 0 and may contain digits 0 through 7.
- For example

04, 010, -12

Floating point numbers:

Floating point numbers are written using base 10. They may have fractional parts separated by a decimal point. For example

4.5e-2, 2e-3

Boolean data type:

A Boolean type variable can have only two values true or false.

result = 2 < 3 // result is true

result = 2 > 3 // result is false

Operators:

- Arithmetic operators
- Assignment operators
- Relational operators
- Logical operators
- Bitwise operators
- Arithmetic operators

+Operator

It performs numeric addition

var result = 2 + 3;

-operator



It performs subtraction operation

Result = 3 - 1;

***operator**

➤ It performs multiplication of numeric operands.

result = 3 * 2;

/ operator

➤ It performs division of numeric operands.

Result = 5 / 2; // result is 2.5

% operator

➤ This is a binary modulus operator

Result = 5 % 2; // result is 1

++ operator

➤ The unary increment operator increments the value of the operand by one.

X = 1;

X++; // result x is 2

--operator

➤ The unary decrement operator decrements the value of the operand by one.

X = 5;

x--; // result x = 4;

-(unary minus) operator

➤ This operator is used to negate a number

Result = -1; // result is -1;

+ (unary plus) operator

➤ This operator does nothing except converting a non-number value (such as a string) to a number.

Result = +1; // result is 1

Operator precedence

JavaScript arithmetic operator precedence rule:

| Precedence | Operators |
|------------|--------------------|
| Lowest | *, - |
| | *, /, % |
| Highest | ++, -- |
| | -(unary), +(unary) |

Assignment Operators

The primary assignment operator (=) is a binary operator that assigns the value of the operand on its right hand to the operand on its left.



JavaScript assignment Operators

| Operator | Example | Meaning |
|----------|---------|----------|
| += | a += b | a=a+b |
| -= | a -=b | a=a-b |
| *= | A *= b | A = a*b |
| /= | A /= b | A=a/b |
| %= | A %=b | A=a%b |
| != | A!=b | A =a ! b |
| ^= | A^=b | A= a^b |
| <<= | A<<=b | A=a<<b |
| >>= | a>>=b | A=a>>b |

Relational operators

Relational operators are binary operators that test the relationship between two operands and return a Boolean value (i.e., either true or false)

JavaScript relational operators

| | |
|----|--------------------------|
| > | Greater than |
| >= | Greater than or equal to |
| < | Less than |
| <= | Less than or equal to |
| == | Equal to |
| != | Not equal to |

Logical operators

Logical operators typically operate on Boolean values, which are either true or false.

The logical operators are &&, ||, !

JavaScript logical operators

| a | b | Result | | |
|-------|-------|--------|--------|-------|
| | | A && b | A b | !a |
| false | false | false | False | True |
| false | true | false | True | |
| true | false | false | True | False |
| true | true | true | true | |



Bitwise Operators

JavaScript bitwise operators

| Operator | Meaning | Usage |
|----------|------------------------------|---------|
| & | Bitwise AND | A & b |
| | Bitwise OR | A b |
| ~ | Bitwise XOR | A ~ b |
| - | Bitwise NOT | - a |
| << | Left shift | A << b |
| >> | Sign-propagating right shift | A >> b |
| >>> | Zero-fill right shift | A >>> b |

Control Structure:

for

- The for loop is used to execute a set of instructions repeatedly.

for (initial expression; condition; update_expression)

```
{  
    Statements;  
}
```

for (i = 1; i <= 5; i++)

```
    document.writeln(i);
```

while:

- The while loop is used to execute a block of statements as long as a given condition is true.

while (condition)

```
{  
    Statements;  
}
```

i = 1;

while(i<=5)

```
{  
    document.write(i+ ' ');  
    i++;  
}
```

do- while

- The do-while loop is exactly the same as the while loop except that the condition is specified at the end of the body.

do

```
{
```



Statements;

} while (conditions);

i = 1;

do

{

document.write(i + ' ');

} while (i<=5);

Conditional Statements

➤ If the condition is true execute one part otherwise execute another part.

If (condition) {

Some javascript code;

}

else {

Some javascript code;

}

var x = 2;

If(x % 2==0) {

alert("even");

}

else {

alert ("odd");

}



UNIT – IV

EXTENSIBLE MARKUP LANGUAGE (XML)

Extensible Markup Language:

Common Usage:

XML is the acronym of extensible markup language. The xml standard was developed by the world wide web consortium (W3C) in the late 1990.

Role of XML:

- XML also promotes easy data sharing among applications.
- Since applications hold data in different structures, mapping data from one to another is a difficult task.
- An xml document consists of the following parts
 - ✓ Prolog
 - ✓ Body

Prolog

The part of the xml document may contain the following parts

XML declaration

Optional processing instruction

Comments

Document type declaration

XML Declaration

- Every XML document should start with a one-line XML declaration
- For XML declaration : `<? xml version="1.0"?>`
- The current version is 1.0. The declaration may use two optional attributes: encoding and standalone

Encoding:

- It specifies the type of encoding scheme used in the document.
- `<? xml version="1.0" encoding="UTF-8"?>`

UTF stands for Unicode Transformation Format is used which has the same character set as ASCII.

Standalone:

This optional attribute indicates whether the document can be processed as a standalone document or is dependent on other documents.

`<? xml version="1.0" encoding="UTF-8" standalone="no"?>`

Processing Instruction:

Processing Instruction start with `<?` and end with `?>`.



They allow xml documents to contains special instructions that are used to pass parameters to the application.

```
<?xml=stylesheet href="simple.xml" type="text/xsl"?>
```

This processing instruction states that the xml document should be transformed using the style sheet simple.xml.

Comments:

- Like HTML, comments may appear anywhere in the XML documents.
- An XML comment starts with <!-- and ends with -->

The following points using comments:

- ✓ Never place a comment inside an entity declaration
- ✓ Never place a comment within a tag.
- ✓ Never place a comment before the xml declaration.

Document Type Declaration:

- The document type declaration (or DOCTYPE declaration) is used to specify the logical structure of the xml document
- The structure is specified by imposing constraint on what tags can be used and where.
- A parser read this section and checks whether the XML document has been written according to the rules specified
- A Document Type Declaration may contain the following
 - Name of the root element
 - Reference to an external DTD
 - Element declaration
 - Entity declaration

Body

- This portion of the xml document contains textual data marked up by tags.
- It must have one elements, called the document element or root elements which defines the content in the xml documents.
- The root elements contains other elements which, in turn, contain other elements and so on.

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

```
<contact>
```

```
<person>
```

```
    <name> Roy</name>
```

```
    <number>1001</number>
```

```
</person>
```

```
<person>
```

```
    <name> Ram</name>
```




<number>1002</number>

</person>

</contact>

- The xml document has the root element <contact> which the two <person> elements.
- Each <person> element has two elements <name> and <number>

Elements:

- An xml element consists of a starting tag, and ending tag, and its contents and attributes.
- The content may be simple text, or other elements, or both.
- Each element contains different types of data that are stored in the xml document.
- The xml tags are very must similar to that of html tags.
- A tag begins with the less than character ('<') and ends with the greater than ('>') character.
- It takes the form <tag-name>.
- Every tag must have its corresponding ending tag.

Naming rules:

- Names can only contain letters, digits, and some other special characters
- Names cannot start with a number or punctuation marks
- Names must not contain the string "xml", "XML", or "Xml"
- Names cannot contain white space(s).

Empty elements:

- Empty elements are those that do not have any content. However, they can have attributes.
- Every xml element must have a closing tags.

<line width="100"></line>

Attributes

- Attributes are used to describe elements or to provide more information about elements.

<element-name attribute-name=attribute-value></element>

<employee gender="male"></employee>

Well-formed XML

- An XML document must have one and exactly one root element.
- All tags must be closed.
- All tags must be properly nested
- Xml tags are case –sensitive
- Attributes must always be quoted
- Certain characters are reserved for processing
- Valid xml
- Well formed xml documents obey only basic well-formedness constraints. So, valid xml documents are those that
- Are well-formed



- Comply with rules specified in the DTD or schema.

Validation

It is a method of checking whether an xml document is well-formed and conforms to the rules specified by the DTD or schema.

Elements or attributes

Attributes are usually used to provide information that is not an integral part of the xml document.

<employee>

<gender>male</gender>

</employee>

Displaying XML

There are many ways to display data stored in an xml document.

The two common methods are

- CSS (Cascading Style Sheet)
- XSL (eXtensible Stylesheet Language)

XML schema languages

- In computer science, a schema is an abstract representation of an object's characteristics and its relationship to other objects.
- An xml schema represents the interrelationship between the elements and attributes in an xml document
- An xml schema language is a formal language to express xml schemas.
- Several such xml schema languages are proposed in below table.
- The document type definition (DTD) is on such xml schema language that is native to the xml standardized by the world wide web consortium (W3C)
- Two primary xml schema languages, namely, DTD, which is very popular and is widely used and W3C xml schema, which is known to be an extremely powerful schema language.

Major XML language

- DTD – Document Type Definition
- W3C XML Schema
 - ✓ Xml – data
 - ✓ Xml – data reduced (xdr)
 - ✓ Document content description for xml (dcd)
 - ✓ Document definition markup language or xschema (DDML)

Relax NG Family

- ✓ Relax
- ✓ Xduce
- ✓ Tree Regular expressions for xml (trex)
- ✓ Relax new generation (relax ng)

Validation:

- Xml validation is a procedure used to verify whether an xml document conforms to the rules specified by a given xml schema.



- This task is accomplished by a special programs called xml parser (also called xml processor).
- This parser typically takes two arguments an xml document to be verified and an xml schema (e.g. W3C XML Schema or a DTD) where the rules to be followed by an xml instance document are specified.
- If the xml document follows the rules specified in the xml schema, it is said to be valid.
- Otherwise the xml processor reports a set of diagnostic error messages.
- These messages are used to filter the errors from the xml documents
- Moreover, an xml processor may optimize the xml document.
- Optimization includes removal of white space, insertion of default elements and attributes, an so on.
- Validating an xml instance document against a schema is language-specific.

Introduction of DTD

- Document type definition (DTD) is one of several xml schema languages and was introduced as a part of the xml 1.0 recommendation

Purpose of DTD

- The purpose of a DTD like other schema languages, is to define the legal building blocks of an xml documents.
- It specifies the document structure with a list of legal elements.
- It is also used to specify a content model for each element and attribute used in an xml document.
- The content description is a part of the element declaration in the DTD and specifies the order and quantity of elements that can be contained within the element being declared.
- A DTD cannot specify all the constraints that may arise in a XML document.
- There are two types of xml documents 1. Well defined and ii. Valid
- An xml document is said to valid with respect to a DTD, if ti conforms with the rules specifies by a given xml dtd.
- Independent developers can agree to use a common DTD for exchanging xml documents.

Using a DTD in an xml document

- A document type declaration is used to make such a link and the keyword DOCTYPE is used for this purpose
- There are three ways to make this link
- The DTD can be embedded directly in the XML document as a part of it called internal DTD.
- Form an xml document, a reference to an external file containing the dtd can be made called external DTD.
- Both internal and external DTD can be used.

Internal DTD

- When we embed a DTD is an xml document, the DTD information is included within the xml document itself.

Syntax:

```
<!DOCTYPE root-element {  
    <!-- doctype declaration -->  
}>
```



Ex:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE greeting
```

```
  < ! element greeting ( %PCDATA) >
```

```
}>
```

```
<greeting>Hello world</greeting>
```

- The advantage of an internal DTD is that you only have to handle a single xml document instead of many.
- It is useful particularly when you are debugging and editing both DTD and xml documents, while you are experimenting.

External DTD:

- Another way of connecting a DTD to the XML document is to reference it within the xml document i.e, create a separate document, put DTD information there, and point to it from the xml document.

Syntax:

```
<! DOCTYPE root-element [SYSTEM OR PUBLIC FPI ] "URL">
```

The following example depicts DTD declaration using a external reference and the SYSTEM keyword

```
<?xml version="1.0"?>
```

```
< ! DOCTYPE greeting SYSTEM 'hello.dtd'>
```

```
<greeting>hello world</greeting>
```

- The advantages of External DTDs are useful for creating a common DTD that can be shared among multiple documents.
- You can create a DTD once and reference it as many times as you need.

Combining Internal and External DTDs

- XML specifications allow document authors to combine both internal and external DTD subsets for a complete collection of rules for a given document

Syntax:

```
< ! DOCTYPE root-element [SYSTEM OR PUBLIC P[] 'URL' [DTD]]
```

Ex

```
<?xml version="1.0"?>
```

```
< ! DOCTYPE greeting SYSTEM 'hello.dtd'> {
```

```
  <! ENTITY ex11 '&ex12'>
```

```
}>
```

```
<greeting>hello world</greeting>
```



UNIT – V

COMMON GATEWAY INTERFACE (CGI)

Internet Programming paradigm

- Internet programming can be classified into two categories: client-side programming and server side programming.
- In the client-side programming paradigm, programs/scripts are downloaded, interpreted, and executed by the browser.
- Applet are client-side programming technology, where special Java programs are embedded directly into web pages with the help of the <applet> tag.

Server-Side Programming

- The Common Gateway Interface (CGI) is one of the important server-side programming techniques.
- The CGI connects a web server to an external applications.
- When a CGI-enabled web server receives a request for a CGI program.
- It does not send the file as it is. Instead the web server executes the program at the server end and sends the output back to the client, which is then displayed in the browser's window.
- This simple and elegant ideas can be used to develop many powerful applications.
- Most of the web servers have built-in-support for CGI
- Moreover, the CGI specification is independent of any programming language.

Languages for CGI

C/C++:

- C is one of the popular programming languages.
- It is well known for its extremely good performance.
- It is widely used on many different software platforms and is a primary language for CGI

Perl:

- This interpreted language provides powerful text processing and file manipulation facilities.
- Perl borrows features from other programming languages such as C, shell script (sh) AWK and sed.

Tel:

- Tel is a scripting language that was originated from "Tool Command Language", but is conventionally rendered as "Tel".
- It is commonly used for CGI scripting as well as rapid prototyping, and other scripted applications, Guls, and testing.

Python:

- It is an interpreted, interactive, portable, object oriented programming (OOP) language.
- Its significant power and clear syntax make Python an excellent instructional tool and ideal for CGI programming.
- Languages features include modulus, classes, very high level dynamic data types, and dynamic typing.

Unix/Linux Shell:

- A shell is a command line interpreter that provides an interface to the user, to execute their commands.
- Unix/Linux shell is extremely powerful for manipulating files pattern searching, and matching and is ideal for CGI scripting in the Unix/Linux platform.



Applications:

- There are numerous applications of the CGI.
- It is usually used to build database applications in conjunction with HTML forms.

Server Environment

- Most the web servers standardize the CGI mechanism.
- Usually, the directory “cgi-bin” exists under the web servers’s installation directory.
- The files in this directory are treated differently.
- Any file requested from this special “cgi-bin” directory is not simply read and sent.
- Instead, it is executed in the computer where the web server is installed.
- The output of this program is actually sent to the browser that request this file.
- The program is an executable file of typically a Perl or Python script.
- Suppose that you have entered the following URL in the address bar of your browser
- `http:// 192.168.1.2/cgi-bin/hello.pl`
- The web server recognizes the file `hello.pl`, which is nothing but a Perl script in the `cgi-bin` directory.
- It then executes `hello.pl` and sends the output of this script to the browser, which then displays it on the screen.

Environment Variables

| Variable Name | Description |
|-------------------|---|
| SERVER_NAME | The server’s DNS name or IP Address |
| SERVER_SOFTWARE | The name and version of the web server software answering the request |
| GATEWAY_INTERFACE | The version of the CGI specification that the server complies with |
| SERVER_PORT | The port number used by the web browser |
| SERVER_PROTOCOL | The HTTP version used by the server |
| CONTENT_LENGTH | The length of the query information in case of the POST method |
| CONTENT_TYPE | The type of the data of the content |
| HTTP_ACCEPT | The list of MIME types that the client can accept |
| PATH | The value of PATH environment variable |
| QUERY_STRING | The URL encoded information which follows the ? in the URL |
| REMOTE_ADDR | The IP address of the remote host that made the request |
| REQUEST_METHOD | The HTTP method used in the request. The most common methods are GET, HEAD and POST |
| SCRIPT_FILENAME | The full path of the CGI script being executed |
| SCRIPT_NAME | The relative path of the script being executed |

CGI Scripting using C, shell Script

- Writing CGI programs using C/C++ language is somehow different from writing them in shell script.
- Here, shell refers to the Unix/Linux shell.
- The windows shell is not so powerful and is hardly used.
- Note that shell scripts are interpreted by the underlying shell.
- C/C++ programs are, compiled programs. Writing CGI programs using C/C++ basically consists of two steps:



- Compile the program to generate the executable file
- Put this executable file in the cgi directory

Writing CGI Programs

- We develop simple programs to display “Hello World” using different languages such as Perl, C, and Python.

Following is the program (hello.pl) written in Perl

```
#!/D:/perl/bin/perl.exe
```

```
Print "content/type: text/html\n";
```

```
Print "<html>\n";
```

```
Print "<head><title>first perl script</title></head>\n";
```

```
Print "<body>\n"
```

```
Print "<h2>Hello World</h2>\n";
```

```
Printf "</body>";
```

```
Print "</html>";
```

- The script hello.pl is a simple script that prints a simple HTML document on the standard out i.e, screen.
- The first line of any Perl script should look like this:

```
#!/D:/perl/bin/perl.exe
```

Getting Arguments:

- We discuss how CGI programs can retrieve URL encoded parameters passed to them. The parameters passed have the following characteristics:
- They are sent in the form of value pairs
- Each name-value pair starts with an & sign
- The name and value are separated by the equal (=) sign.

URL encoding changes some special characters to placeholders and replaces them with hexadecimal values. To retrieve the information, the CGI program should implement the following basic steps:

- ✓ Get the information from the proper environment variable, depending upon the method type. For example, we can retrieve information for the QUERY_STRING environment variable for the GET method
- ✓ Change all placeholders to their correct values
- ✓ Split each name-value pair
- ✓ Convert hexadecimal values back to their original characters
- ✓ Find the respective name and value.

CGI Security

- The CGI technology allows users to run programs in a system remotely using their URLs.
- This may become vulnerable.
- Therefore, some security precautions need to be taken when CGI technology is used.
- Note that CGI programs are kept in a special directory, so that the webserver can execute the program rather than just send them to the browser.
- The most important concept that CGI writers follow:
- The external users may try to use this concept to perform malicious events.



- The CGI directory is usually controlled by the webmaster, who should prohibit the average user from creating and running CGI programs.

The following points should be taken into consideration while writing the security policy:

- Users who can access the system
- When they are allowed to use the system
- What operations they are allowed to perform
- A way to grant access to the system
- Acceptable permission to use the file system
- Procedure for monitoring the system
- Actions to be taken against suspected security breaches.

Server-Side Java

- Servlet is a Java technology for server-side programming.
- It is a Java module that runs inside a Java-enabled web server.
- Servlets are not tied to a specific client-server protocol, but are most commonly used with HTTP. The word “servlet” is often used to mean “HTTP servlet”
- Execution of a servlet consists of four steps:
 - ✓ The client sends a request to the web server
 - ✓ The web server interprets it and forwards it to the corresponding servlet.
 - ✓ The servlet process the request, generates the and sends it back to the web server
 - ✓ The web server sends the response back to the client. The browser then displays it on the screen

Advantages over Applets

- Applets, upon download from the server, run in the client’s browser.
- They will function property provide a proper Java Runtime Environment (JRE) is installed in the client’s browser.

Servlet Alternatives

- There are several alternatives to servlets
- Common Gateway Interface (CGI)
- CGI is one of the most common server side solutions.
- Although widely used, CGI scripting technology has a number of shortcomings, including platform dependence and lack of scalability.
- A CGI application is an independent program that receives requests from the web server and sends it back to the web server.
- Some of the common problems of this technology are as follows:
 - ✓ A new process is created every time the web server receives a CGI request.
 - ✓ Since a new process is created and initialized, it results in the delay of response time.
 - ✓ Servers may also run-out of memory if not configured properly.

Proprietary APIs

- Many proprietary web servers have built-in support for server-side programming
- Examples include netscape’s NSAPI, Microsoft’s ISAPI and O’Reilly’s WSAPI.
- Most of these are developed in C/C++ languages and hence can contain memory leaks and core dumps that can crash the web server.

Active Server Page (ASP)

- Microsoft Active Server Pages (ASP) is another technology that supports server-side programming.



- Unfortunately, the only web server that supports this technology is Microsoft Internet Information Server (IIS), which is not free.
- Although some third-party products support ASP, they are not free either.

Server-Side JavaScript

- Server-Side JavaScript is another alternative to servlets.
- However, the only known servers, that supports it are Netscape's Enterprise and FastTrack servers.

Servlet Strengths

- Java servlet technology has some distinct advantages over other competent technologies.
- Truly speaking, except JSP no technology can compete with servlet technology.
- Following are some advantages of Java servlet technology

Efficient:

- When a servlet gets loaded in the server, it remains in the server's memory as a single object instance.
- Each new request is then served by a lightweight Java thread spawned from the servlet.

Persistent:

- Servlets can maintain the session by using the session tracking/cookies, a mechanism that helps them track information from request to request.
- Critical information can also be saved to a persistent storage and can be retrieved from it when the servlet is loaded next time.

Portable:

- Servlets are written in Java and so they are portable across operating systems and server implementations.
- This allows servlets to be moved across new operating systems.
- We can develop a servlet on a Windows machine running the Tomcat or any other server and later we deploy that servlet effortlessly on any other operating system such as a Unix server running as Iplanet Netscape application server.
- So servlets are Write Once Run Anywhere (WORA) programs

Robust:

- Servlets can use the entire JDK, they can provide extremely robust solutions.
- Java has a very powerful exception handling mechanism and provides a garbage collector to handle memory leaks.
- It also has a very large and rich class library with network and file support, database access, DOM supports, security, utility, packages, etc.

Extensible:

- Servlets are nothing but Java technology, which is popular for its well-known features such as platform independence, garbage collection, exception handling and extremely powerful utility packages.

Secure:

- Servlets run in a Java-enabled web server.
- So they can use security mechanisms from both the web server and the Java security manager.

Cost-effective:

- There are number of free web servers available for personal and commercial use.



- Java Development Kit (JDK) is also free and can be download from <http://java/jav.sun.co>.
- So, applications can be developed practically without any cost.

Servlet Life Cycle:

- The container in which the servlet has been deployed supervises and controls the life cycle of a servlet.
- Typically, this container is nothing but a web browser.

init()

- A servlet's life begins here.
 - This method is called only_once by the web container, just after it instantiates and loads the servlet.
 - One the servlet is initialized, it becomes ready to handle the client's request.
- The prototype of the init() method is as follows:

`void init(ServletConfig)`

service()

- This is the most important method, whose signature is as follows:
`void service(ServletRequest request, ServletResponse response)`
- The web container calls the service() method on a servlet with two objects: a ServletRequest_type object request and a ServletResponse_type object response.
- The request object encapsulates the communication from the client to the server, while response_object encapsulated the communication form the servlet back to the client
- The ServletRequest interface provides methods to retrieve information sent by the clients, Similary, the ServletResponse interface provides methods to send data to the clients.

destroy()

- The method signature is as follows:
`void destroy()`
- This method gets called when the web container uninstalls the servlet.
- This method is override to clean up the resources that were allocated to this servlet.

Other Methods

- The javax.servlet.servlet interface also provides the following methods to inspect the properties of a servlet at runtime

`ServletConfig getServletConfig()`

- Returns a servletConfig object, which contains the initialization parameters and startup configuration used for this servlet.

`String.getServletInfo();`

- Returns a string object containing information about the servlet, such as its version, author, copyright, etc.