WEB TECHNOLOGIES NOTES FOR 6TH SEMESTER

Prepared by: Suraj B.S., DCS&E, B.E, M.Tech

COURSE CODE: P13CS61

UNIT 1: FUNDAMENTALS OF WEB, XHTML

A BRIEF INTRODUCTION ABOUT THE INTERNET

Origins:

1960s

- U.S. Department of Defence (DoD) became interested in developing a new large-scale computer network
- The purposes of this network were communications, program sharing, and remote computer access for researchers working on defence-related contracts.
- The DoD's Advanced Research Projects Agency (ARPA) funded the construction of the first such network. Hence it was named as ARPAnet.
- The primary early use of ARPAnet was simple text-based communications through e-mail.

late 1970s and early 1980s

- BITNET, which is an acronym for ***B**ecause **I**t's **T**ime **NET**work*, began at the City University of New York. It was built initially to provide electronic mail and file transfers.
- CSNET is an acronym for ***C**omputer **S**cience **NET**work*. Its initial purpose was to provide electronic mail.

1990s

- NSFnet which was created in 1986 replaced ARPAnet by 1990.
- It was sponsored by the National Science Foundation (NSF).
- By 1992 NSFnet, connected more than 1 million computers around the world.
- In 1995, a small part of NSFnet returned to being a research network. The rest became known as the ***Internet.***

What Is the Internet?

- The Internet is a huge collection of computers connected in a communications network.
- The Transmission Control Protocol/Internet Protocol (TCP/IP) became the standard for computer network connections in 1982.
- Rather than connecting every computer on the Internet directly to every other computer on the Internet, normally the individual computers in an organization are connected to each other in a local network. One node on this local network is physically connected to the Internet.
- So, the Internet is actually a ***network of networks***, rather than a network of computers.
- Obviously, all devices connected to the Internet must be uniquely identifiable.

Internet Protocol Addresses

- The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32-bit number.
- IP addresses usually are written (and thought of) as four 8-bit numbers, separated by periods.
- The four parts are separately used by Internet-routing computers to decide where a message must go next to get to its destination.
- Although people nearly always type domain names into their browsers, the IP works just as well.

- For example, the IP for United Airlines (`www.ual.com`) is `209.87.113.93`. So, if a browser is pointed at `http://209.87.113.93`, it will be connected to the United Airlines Web site.

Domain Names

The IP addresses are numbers. Hence, it would be difficult for the users to remember IP address. To solve this problem, text based names were introduced. These are technically known as ***domain name system (DNS).***

These names begin with the names of the host machine, followed by progressively larger enclosing collection of machines, called ***domains.*** There may be two, three or more domain names. DNS is of the form **hostname.domainName.domainName** . Example: **pesce.ac.in** The steps for conversion from DNS to IP:

- The DNS has to be converted to IP address before destination is reached.
- This conversion is needed because computer understands only numbers.
- The conversion is done with the help of *name server*.
- As soon as domain name is provided, it will be sent across the internet to contact name servers.
- This name server is responsible for converting domain name to IP
- If one of the *name servers* is not able to convert DNS to IP, it contacts other name server.
- This process continues until IP address is generated.
- Once the IP address is generated, the host can be accessed.
- The hostname and all domain names form what is known as FULLY QUALIFIED DOMAIN NAME.

This is as shown below:

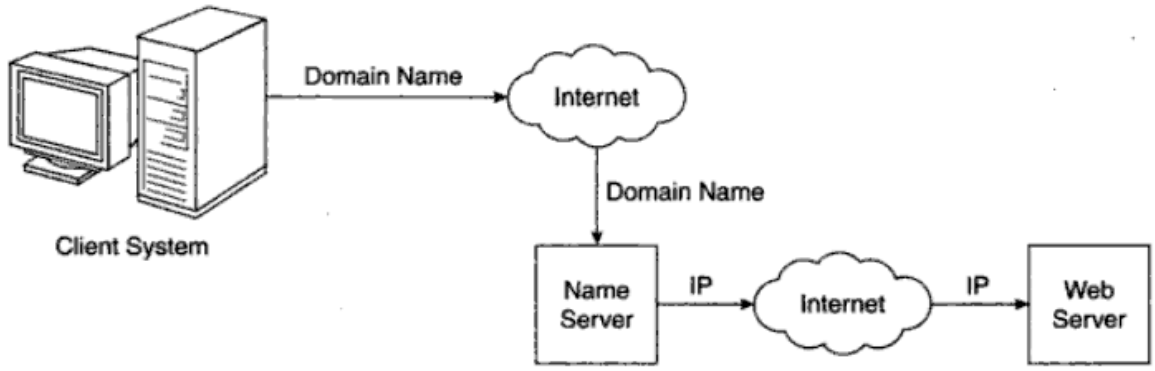


Figure 1.1 Domain name conversion

The World Wide Web

Origins

- Tim Berners Lee and his group proposed a new protocol for the Internet whose intention was to allow scientists around the world to use the Internet to exchange documents describing their work.

- The proposed new system was designed to allow a user anywhere on the Internet to search for and retrieve documents from the databases on any number of different document-serving computers.
- The system used *hypertext*, which is text with embedded links to text in other documents to allow non-sequential browsing of textual material.
- The units of web are referred as pages, documents and resources.
- Web is merely a vast collection of documents, some of which are connected by links.
- These documents can be accessed by web browsers and are provided by web servers.

Web or Internet?

It is important to understand that the Internet and the Web is not the same thing.

✓ The **Internet** is a collection of computers and other devices connected by equipment that allows them to communicate with each other.

✓ The **Web** is a collection of software and protocols that has been installed on most, if not all, of the computers on the Internet.

WEB BROWSERS

- They are called browsers because they allow the user to browse the resources available on servers.
- Mosaic was the first browser with a graphical user interface.
- A browser is a client on the Web because it initiates the communication with a server, which waits for a request from the client before doing anything.
- In the simplest case, a browser requests a static document from a server.
- The server locates the document among its servable documents and sends it to the browser, which displays it for the user.
- Sometimes a browser directly requests the execution of a program stored on the server. The output of the program is then returned to the browser.

• Examples: Internet Explorer, Mozilla Firefox, Netscape Navigator, Google Chrome, Opera etc.,

WEB SERVERS

Web servers are programs that provide documents to requesting browsers. Example: Apache

Web server operations:

- All the communications between a web client and a web server use the HTTP
- When a web server begins execution, it informs the OS under which it is running & it runs as a background process
- A web client or browser, opens a network connection to a web server, sends information requests and possibly data to the server, receives information from the server and closes the connection.
- The primary task of web server is to monitor a communication port on host machine, accept HTTP commands through that port and perform the operations specified by the commands.
- When the URL is received, it is translated into either a filename or a program name.

General characteristics of web server:

- The file structure of a web server has two separate directories
- The root of one of these is called **document root** which stores web documents
- The root of the other directory is called the **server root** which stores server and its support softwares

- The files stored directly in the document root are those available to clients through top level URLs
- The secondary areas from which documents can be served are called **virtual document trees.**
- Many servers can support more than one site on a computer, potentially reducing the cost of each site and making their maintenance more convenient. Such secondary hosts are called **virtual hosts.**
- Some servers can serve documents that are in the document root of other machines on the web; in this case they are called as **proxy servers**

Apache

- Apache is the most widely used Web server.
- The primary reasons are as follows: Apache is an excellent server because it is both fast and reliable.
- Furthermore, it is open-source software, which means that it is free and is managed by a large team of volunteers, a process that efficiently and effectively maintains the system.
- Finally, it is one of the best available servers for Unix-based systems, which are the most popular for Web servers.
- Apache is capable of providing a long list of services beyond the basic process of serving documents to clients.
- When Apache begins execution, it reads its configuration information from a file and sets its parameters to operate accordingly.

IIS

- Microsoft IIS server is supplied as part of Windows—and because it is a reasonably good server—most Windows-based Web servers use IIS.
- With IIS, server behaviour is modified by changes made through a window-based management program, named the IIS snap-in, which controls both IIS and `ftp`.
- This program allows the site manager to set parameters for the server.
- Under Windows XP and Vista, the IIS snap-in is accessed by going to *Control Panel, Administrative Tools, and IIS Admin.*

UNIFORM RESOURCE LOCATORS

- Uniform Resource Locators (URLs) are used to identify different kinds of resources on Internet.
- If the web browser wants some document from web server, just giving domain name is not sufficient because domain name can only be used for locating the server.
- It does not have information about which document client needs. Therefore, URL should be provided.
- The general format of URL is: **scheme: object-address**
- Example: **http: www.vtu.ac.in/results.php**
- The scheme indicates protocols being used. (http, ftp, telnet...)
- In case of http, the full form of the object address of a URL is as follows:

//fully-qualified-domain-name/path-to-document

- URLs can never have embedded spaces
- It cannot use special characters like semicolons, ampersands and colons

- The path to the document for http protocol is a sequence of directory names and a filename, all separated by whatever special character the OS uses. (forward or backward slashes)
- The path in a URL can differ from a path to a file because a URL need not include all directories on the path
- A path that includes all directories along the way is called a **complete path.**
- Example: http://www.pesce.ac.in/index.html
- In most cases, the path to the document is relative to some base path that is specified in the configuration files of the server. Such paths are called **partial paths.**
- Example: http://www.pesce.ac.in/

MULTIPURPOSE INTERNET MAIL EXTENSIONS

- MIME stands for **Multipurpose Internet Mail Extension.**
- The server system apart from sending the requested document, it will also send MIME information.
- The MIME information is used by web browser for rendering the document properly.
- The format of MIME is: **type/subtype**
- Example: text/html , text/doc , image/jpeg , video/mpeg
- When the type is either text or image, the browser renders the document without any problem
- However, if the type is video or audio, it cannot render the document
- It has to take the help of other software like media player, win amp etc.,
- These softwares are called as **helper applications or plugins**
- These non-textual information are known as **HYPER MEDIA**
- Experimental document types are used when user wants to create a customized information & make it available in the internet
- The format of experimental document type is: **type/x-subtype**
- Example: database/x-xbase , video/x-msvideo
- Along with creating customized information, the user should also create helper applications.
- This helper application will be used for rendering the document by browser.
- The list of MIME specifications is stored in configuration file of web server.

THE HYPERTEXT TRANSFER PROTOCOL

Request Phase:

The general form of an HTTP request is as follows

1. HTTP method Domain part of the URL HTTP version
2. Header fields
3. Blank line
4. Message body

The following is an example of the first line of an HTTP request: **`GET /storefront.html HTTP/1.1`**

Table 1.1 HTTP request methods

Method	Description
GET	Returns the contents of the specified document
HEAD	Returns the header information for the specified document
POST	Executes the specified document, using the enclosed data
PUT	Replaces the specified document with the enclosed data
DELETE	Deletes the specified document

The format of a header field is the field name followed by a colon and the value of the field. There are four categories of header fields:

1. ***General***: For general information, such as the date
2. ***Request***: Included in request headers
3. ***Response***: For response headers
4. ***Entity***: Used in both request and response headers A wildcard character, the asterisk (`*`), can be used to specify that part of a MIME type can be anything

The `Host:` *host name* request field gives the name of the host. The `Host` field is required for HTTP 1.1. The `If-Modified-Since:` *date* request field specifies that the requested file should be sent only if it has been modified since the given date. If the request has a body, the length of that body must be given with a `Content-length` field. The header of a request must be followed by a blank line, which is used to separate the header from the body of the request.

The Response Phase:

The general form of an HTTP response is as follows: 1. Status line 2. Response header fields 3. Blank line 4. Response body The status line includes the HTTP version used, a three-digit status code for the response, and a short textual explanation of the status code. For example, most responses begin with the following:

```
HTTP/1.1 200 OK
```

The status codes begin with `1`, `2`, `3`, `4`, or `5`. The general meanings of the five categories specified by these first digits are shown in Table 1.2

Table 1.2 First digits of HTTP status codes

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

- One of the more common status codes is one user never want to see: `404 Not Found`, which means the requested file could not be found.

SECURITY

Security is one of the major concerns in the Internet. The server system can be accessed easily with basic hardware support, internet connection & web browser. The client can retrieve very important information from the server. Similarly, the server system can introduce virus on the client system. These viruses can destroy the hardware and software in client. While programming the web, following requirements should be considered:

- **Privacy:** it means message should be readable only to communicating parties and not to intruder.
- **Integrity:** it means message should not be modified during transmission.
- **Authentication:** it means communicating parties must be able to know each other's identity
- **Non-repudiation:** it means that it should be possible to prove that message was sent and received properly

Security can be provided using cryptographic algorithm. Ex: private key, public key Protection against viruses and worms is provided by antivirus software, which must be updated frequently so that it can detect and protect against the continuous stream of new viruses and worms.

THE WEB PROGRAMMER'S TOOLBOX

Web programmers use several languages to create the documents that servers can provide to browsers.

The most basic of these is **XHTML**, the standard mark-up language for describing how Web documents should be presented by browsers. Tools that can be used without specific knowledge of XHTML are available to create XHTML documents.

A **plug-in** is a program that can be integrated with a word processor to make it possible to use the word processor to create XHTML. A **filter** converts a document written in some other format to XHTML.

- **XML** is a meta-mark-up language that provides a standard way to define new mark-up languages.
- **JavaScript** is a client-side scripting language that can be embedded in XHTML to describe simple computations. JavaScript code is interpreted by the browser on the client machine; it provides access to the elements of an XHTML document, as well as the ability to change those elements dynamically.
- **Flash** is a framework for building animation into XHTML documents. A browser must have a Flash player plug-in to be able to display the movies created with the Flash framework.
- **Ajax** is an approach to building Web applications in which partial document requests are handled asynchronously. Ajax can significantly increase the speed of user interactions, so it is most useful for building systems that have frequent interactions.
- **PHP** is the server-side equivalent of JavaScript. It is an interpreted language whose code is embedded in XHTML documents. PHP is used primarily for form processing and database access from browsers.

- **Servlets** are server-side Java programs that are used for form processing, database access, or building dynamic documents. JSP documents, which are translated into servlets, are an alternative approach to building these applications. JSF is a development framework for specifying forms and their processing in JSP documents.
- **ASP.NET** is a Web development framework. The code used in ASP.NET documents, which is executed on the server, can be written in any .NET programming language.
- **Ruby** is a relatively recent object-oriented scripting language that is introduced here primarily because of its use in Rails, a Web applications framework.
- **Rails** provides a significant part of the code required to build Web applications that access databases, allowing the developer to spend his or her time on the specifics of the application without the drudgery of dealing with all of the housekeeping details

ORIGINS AND EVOLUTION OF HTML AND XHTML

HTML → Hyper Text Mark-up Language
XHTML → eXtensible Hyper Text Mark-up Language

HTML	XHTML
HTML is much easier to write	XHTML requires a level of discipline many of us naturally resist
huge number of HTML documents available on the Web, browsers will continue to support HTML as far as one can see into the future	some older browsers have problems with some parts of XHTML.
HTML has few syntactic rules, and HTML processors (e.g., browsers) do not enforce the rules it does have. Therefore, HTML authors have a high degree of freedom to use their own syntactic preferences to create documents. Because of this freedom, HTML documents lack consistency, both in low-level syntax and in overall structure	XHTML has strict syntactic rules that impose a consistent structure on all XHTML documents. Another significant reason for using XHTML is that when you create an XHTML document, its syntactic correctness can be checked, either by an XML browser or by a validation tool
Used for displaying the data	Used for describing the data

BASIC SYNTAX

- The fundamental syntactic units of HTML are called ***tags***.
- In general, tags are used to specify categories of content.
- The syntax of a tag is the tag's name surrounded by *angle brackets* (< and >).

- Tag names must be written in all lowercase letters.
- Most tags appear in pairs: an *opening tag and a closing tag*.
- The name of a closing tag is the name of its corresponding opening tag with a slash attached to the beginning. For example, if the tag's name is p, the corresponding closing tag is named /p.
- Whatever appears between a tag and its closing tag is the ***content*** of the tag. Not all tags can have content.
- The opening tag and its closing tag together specify a container for the content they enclose.
- The container and its content together are called an ***element***.
- Example: ** This is PESCE Web Programming Notes. **
- The paragraph tag, , marks the beginning of the content; the tag marks the end of the content of the paragraph element.
- Attributes, which are used to specify alternative meanings of a tag, can appear between an opening tag's name and its right angle bracket.
- They are specified in keyword form, which means that the attribute's name is followed by an equal's sign and the attribute's value.
- Attribute names, like tag names, are written in lowercase letters.
- Attribute values must be delimited by double quotes.
- Comments in programs increase the readability of those programs. Comments in XHTML have the same purpose. They can appear in XHTML in the following form:

```
<!-- anything except two adjacent dashes -->
```

- Browsers ignore XHTML comments—they are for people only. Comments can be spread over as many lines as are needed. For example, you could have the following comment:

```
<!-- CopyRights.html This notes is prepared by Suraj B.S Asst
professor of computer Science Department PESCE, MANDYA-->
```

Standard XHTML Document Structure

- Every XHTML document must begin with an xml declaration element that simply identifies the document as being one based on XML. This element includes an attribute that specifies the version number 1.0.
- The xml declaration usually includes a second attribute, encoding, which specifies the encoding used for the document [utf-8].
- Following is the xml declaration element, which should be the first line of every XHTML document:

<?xml version = "1.0" encoding = "utf-8"?>

- Note that this declaration must begin in the first character position of the document file.
- The xml declaration element is followed immediately by an SGML DOCTYPE command, which specifies the particular SGML document-type definition (DTD) with which the document complies, among other things.
- The following command states that the document in which it is included complies with the XHTML 1.0 Strict standard:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml1-strict.dtd">

- An XHTML document must include the four tags <html>, <head>, <title>, and <body>.

- The <html> tag identifies the root element of the document. So, XHTML documents always have an <html> tag immediately following the DOCTYPE command, and they always end with the closing html tag, </html>.
- The html element includes an attribute, xmlns, that specifies the XHTML namespace, as shown in the following element:

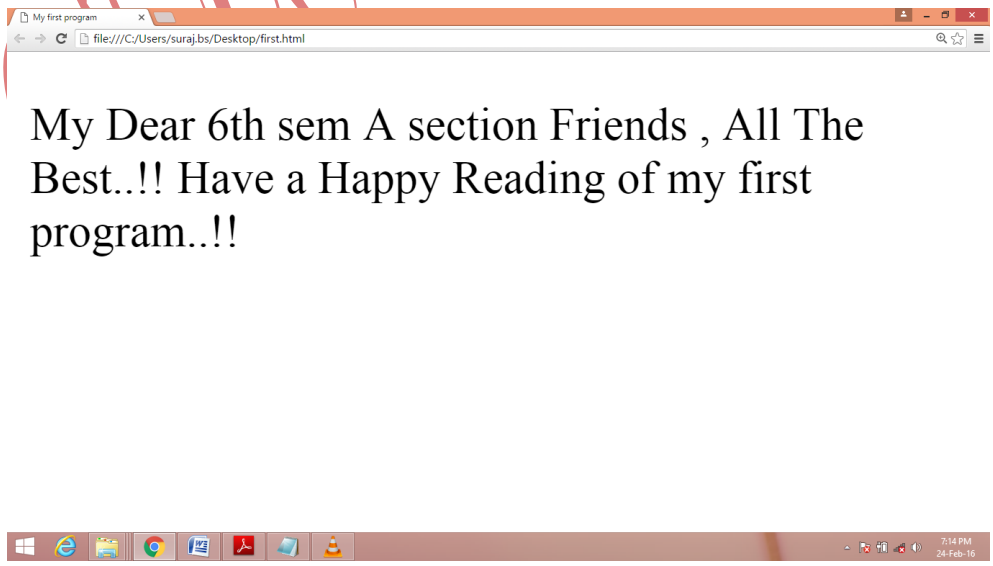
<html xmlns = "http://www.w3.org/1999/xhtml">

- Although the xmlns attribute's value looks like a URL, it does not specify a document. It is just a name that happens to have the form of a URL.
- An XHTML document consists of two parts, named the *head* and the *body*.
- The <head> element contains the head part of the document, which provides information about the document and does not provide the content of the document.
- The body of a document provides the content of the document.
- The content of the title element is displayed by the browser at the top of its display window, usually in the browser window's title bar.

BASIC TEXT MARKUP

We will have a look at a complete XHTML document

```
<?xml version = "1.0" encoding = "utf-8"?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml1-strict.dtd">
 <html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title> My first program </title>
</head>
<body>
 <p> My Dear 6th sem A section Friends , All The Best..!! Have a Happy Reading of my
first program..!! </p>
</body>
</html>
```



PLEASE NOTE: From here onwards programming in XHTML will begin. Please add the following compulsory document structure to all programs in the first 4 lines and skip the simple <html> tag of first line because I have begun the coding part directly .

```
<?xml version = "1.0" encoding = "utf-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml1-strict.dtd"> <html xmlns = "http://www.w3.org/1999/xhtml">
```

Paragraphs:

It begins with and ends with . Multiple paragraphs may appear in a single document.

```
<html>
<head>
 <title> Paragraph </title>
</head> <body>
 <p> Paragraph 1 </p>
<p> Paragraph 2 </p>
<p> Paragraph 3 </p>
</body>
</html>
```

Paragraph 1

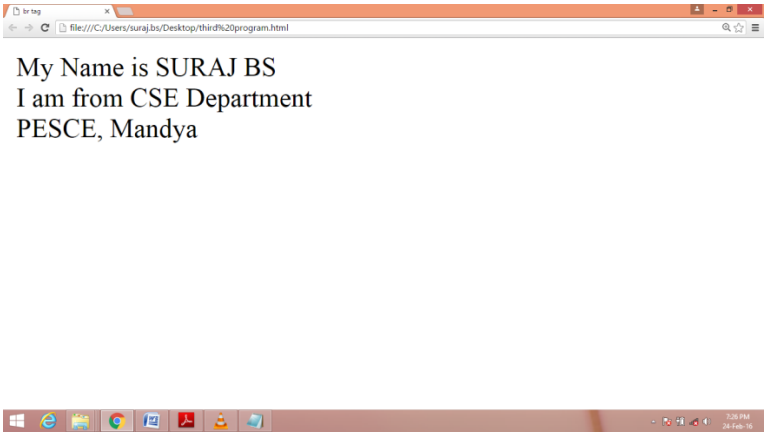
Paragraph 2

Paragraph 3

Line Breaks:

The break tag is specified as <br />. The slash indicates that the tag is both an opening and closing tag.

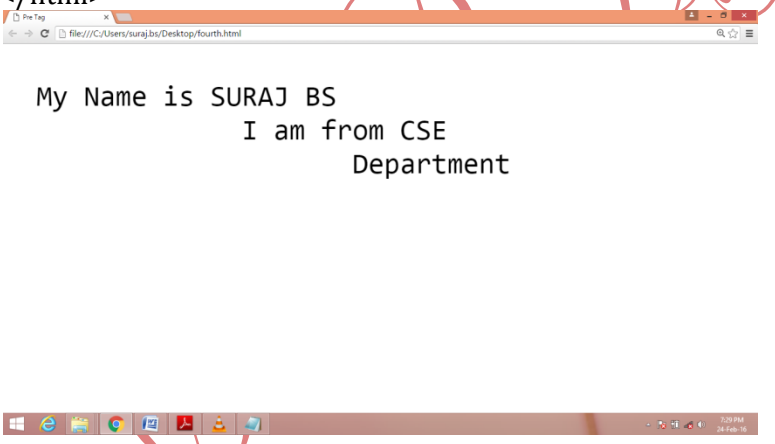
```
<html>
<head>
<title> br tag </title>
</head>
<body>
<p> My Name is SURAJ BS <br/> I am from CSE Department<br/>
 PESCE, Mandya</p>
</body>
</html>
```



Preserving White Space

Sometimes it is desirable to preserve the white space in text—that is, to prevent the browser from eliminating multiple spaces and ignoring embedded line breaks. This can be specified with the **<pre>** tag.

```
<html>
 <head>
<title> Pre Tag </title>
</head>
<body>
<p><pre> My Name is SURAJ BS
        I am from CSE
            Department </pre></p>
</body>
</html>
```

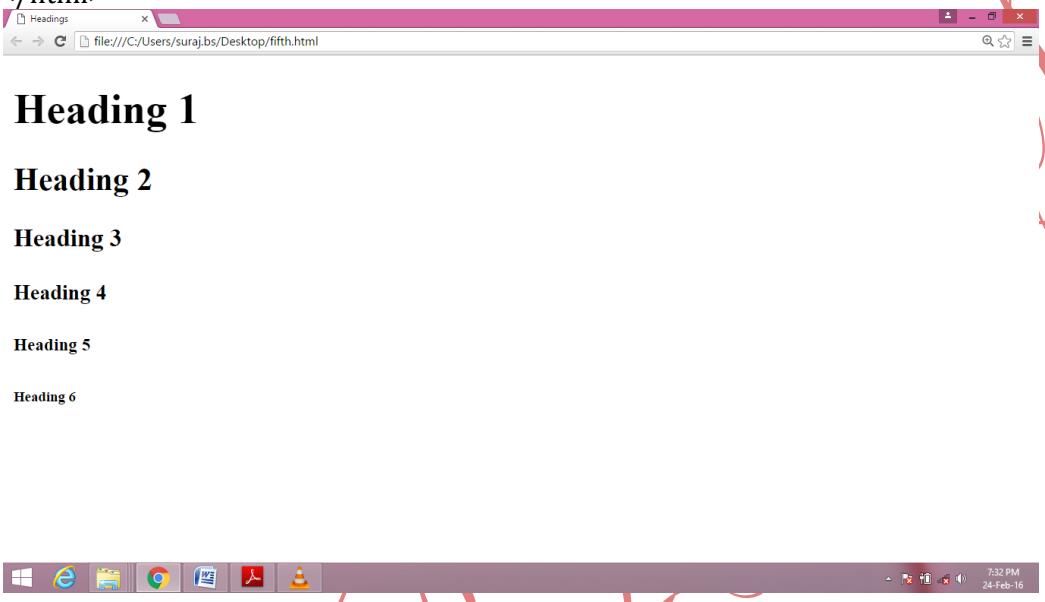


Headings:

In XHTML, there are six levels of headings, specified by the tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`, where `<h1>` specifies the highest-level heading. Headings are usually displayed in a boldface font whose default size depends on the number in the heading tag. On most browsers, `<h1>`, `<h2>`, and `<h3>` use font sizes that are larger than that of the default size of text, `<h4>` uses the default size, and `<h5>` and `<h6>` use smaller sizes. The heading tags always break the current line, so their content always appears on a new line. Browsers usually insert some vertical space before and after all headings.

```
<html>
```

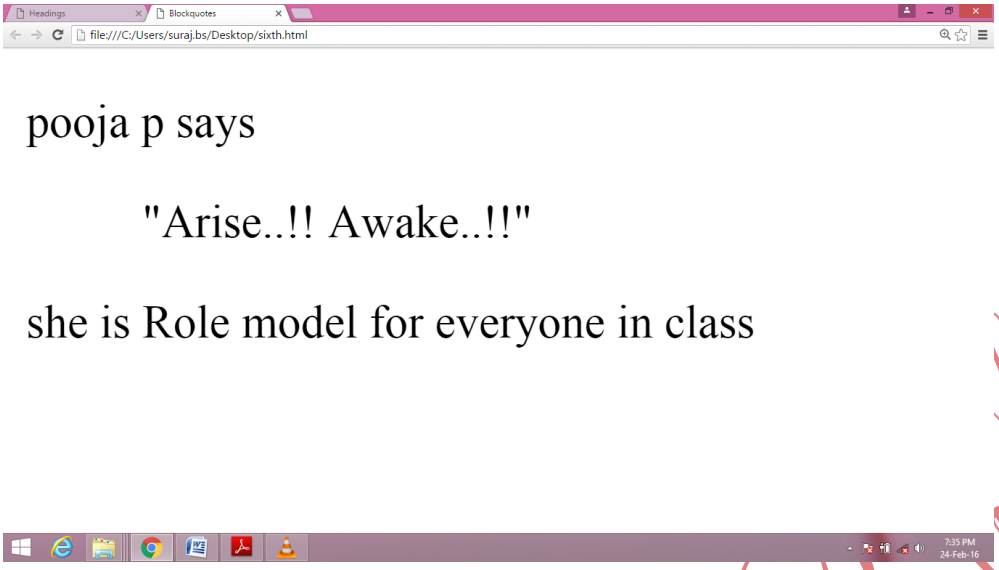
```
<head>
<title> Headings </title>
</head>
<body> <h1> Heading 1 </h1>
<h2> Heading 2 </h2>
<h3> Heading 3 </h3>
 <h4> Heading 4 </h4>
 <h5> Heading 5 </h5>
<h6> Heading 6 </h6>
</body>
</html>
```



Block Quotations:

The <blockquote> tag is used to make the contents look different from the surrounding text.

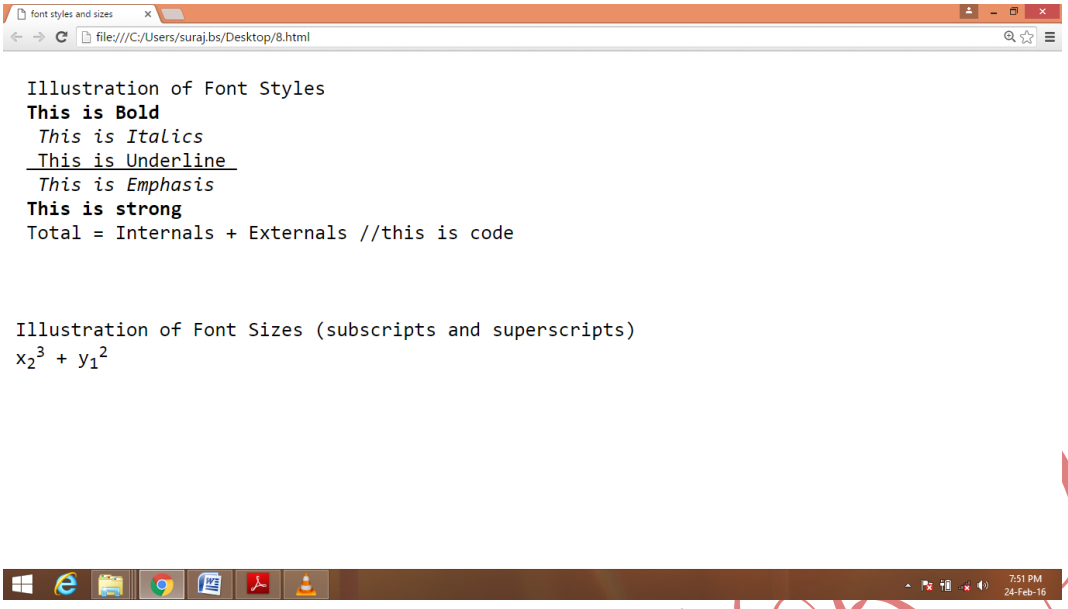
```
<html>
<head>
<title> Blockquotes </title> </head>
<body> <p> pooja p says </p> <blockquote> <p> "Arise..!! Awake..!!" </p> </blockquote>
 <p> she is  Role model for everyone in class</p>
</body>
</html>
```



Font Styles and Sizes:

- <b>, <i> and <u> specifies bold, italics and underline respectively.
- The emphasis tag, `<em>`, specifies that its textual content is special and should be displayed in some way that indicates this distinctiveness. Most browsers use italics for such content.
- The strong tag, `<strong>` is like the emphasis tag, but more so. Browsers often set the content of strong elements in bold.
- The code tag, `<code>`, is used to specify a monospace font, usually for program code.

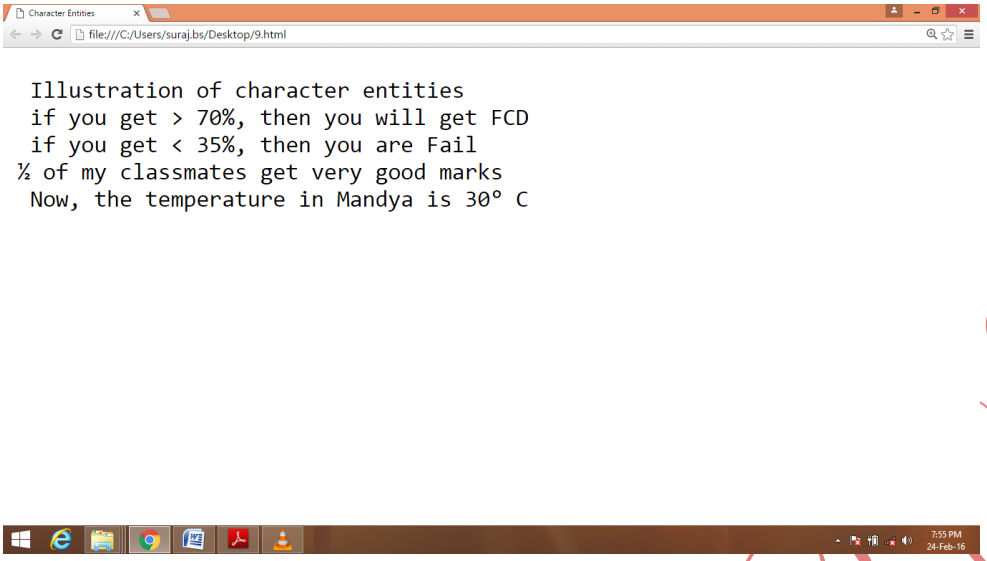
```
<html>
 <head>
<title> font styles and sizes </title>
 </head>
<body>
<p><pre>
 Illustration of Font Styles
<b> This is Bold </b>
 <i> This is Italics </i>
 <u> This is Underline </u>
 <em> This is Emphasis </em>
<strong> This is strong </strong>
<code> Total = Internals + Externals //this is code</code>
 </pre></p> <p><pre>
Illustration of Font Sizes (subscripts and superscripts)
x<sub>2</sub><sup>3</sup> + y<sub>1</sub><sup>2</sup>
</pre></p>
 </body>
</html>
```



Character Entities:

XHTML provides a collection of special characters that are sometimes needed in a document but cannot be typed as themselves. In some cases, these characters are used in XHTML in some special way—for example, >, <, and &. In other cases, the characters do not appear on keyboards, such as the small raised circle that represents "degrees" in a reference to temperature. These special characters are defined as entities, which are codes for the characters. An entity in a document is replaced by its associated character by the browser

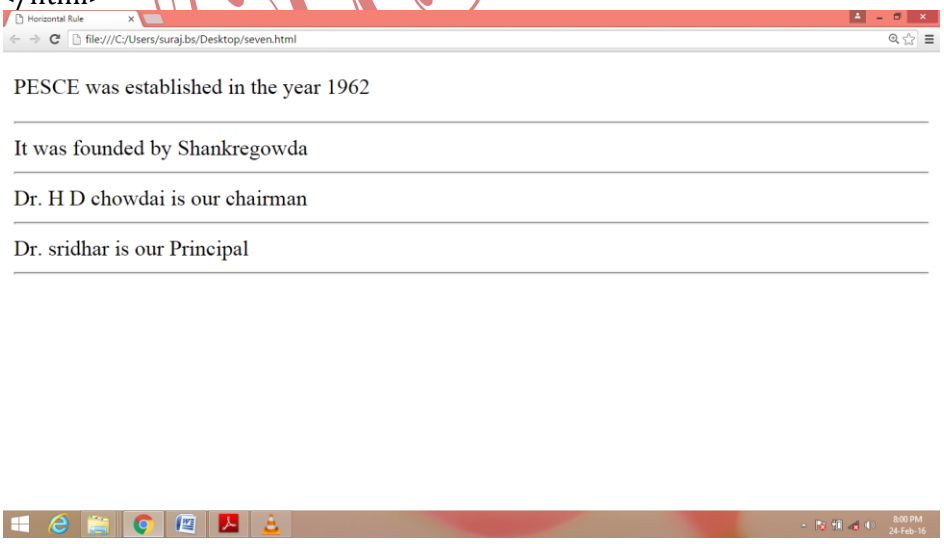
```
<html>
<head>
<title> Character Entities </title>
</head> <body>
<p><pre> Illustration of character entities
 if you get &gt 70%, then you will get FCD
 if you get &lt 35%, then you are Fail
&frac12 of my classmates get very good marks
 Now, the temperature in Mandya is 30&deg C
</pre></p>
</body>
</html>
```



Horizontal Rules:

The parts of a document can be separated from each other, making the document easier to read, by placing horizontal lines between them. Such lines are called horizontal rules. The block tag that creates them is <hr />. The <hr /> tag causes a line break (ending the current line) and places a line across the screen. Note again the slash in the <hr /> tag, indicating that this tag has no content and no closing tag.

```
<html>
<head>
<title> Horizontal Rule </title>
</head>
 <body> <p>
PESCE was established in the year 1962 <hr/>
It was founded by  Shankregowda <hr/>
 Dr. H D chowdai is our chairman <hr/>
Dr. sridhar is our Principal <hr/>
</p>
</body>
</html>
```



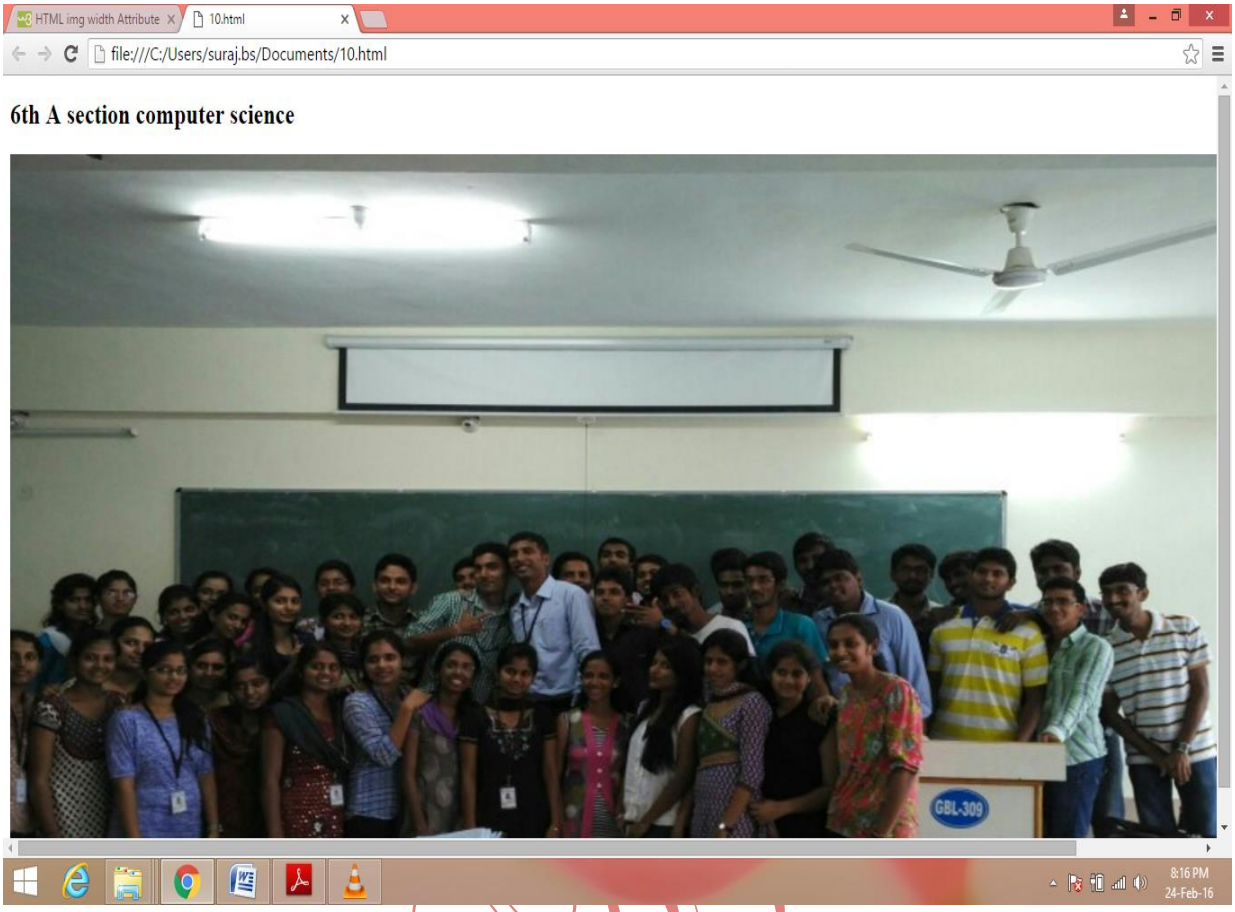
The meta Element:

The `meta` element is used to provide additional information about a document. The meta tag has no content; rather, all of the information provided is specified with attributes. The two attributes that are used to provide information are `name` and `content`. The user makes up a name as the value of the `name` attribute and specifies information through the `content` attribute. One commonly chosen name is `keywords`; the value of the `content` attribute associated with the keywords are those which the author of a document believes characterizes his or her document. An example is **<meta name = "Title" content = "Programming the Web" /> <meta name = "Author" content = "suraj" />**
Web search engines use the information provided with the `meta` element to categorize Web documents in their indices

IMAGES

- Image can be displayed on the web page using **<img>** tag.
- When the <img> tag is used, it should also be mentioned which image needs to be displayed. This is done using **src** attribute.
- Attribute means extra information given to the browser
- Whenever <img> tag is used, alt attribute is also used.
- Alt stands for alert.
- Some very old browsers would not be having the capacity to display the images.
- In this case, whatever is the message given to alt attribute, that would be displayed.
- Another use of alt is → when image display option has been disabled by user. The option is normally disabled when the size of the image is huge and takes time for downloading

```
<html>
<body>
<h2>6th A section computer science</h2>
<img src="6thsemAsection.jpg" alt="suraj" height="600" width="1500">
</body>
</html>
```



NOTE:

- JPEG → Joint Photographic Experts Group
- GIF → Graphic Interchange Format
- PNG → Portable Network Graphics

XHTML Document Validation:

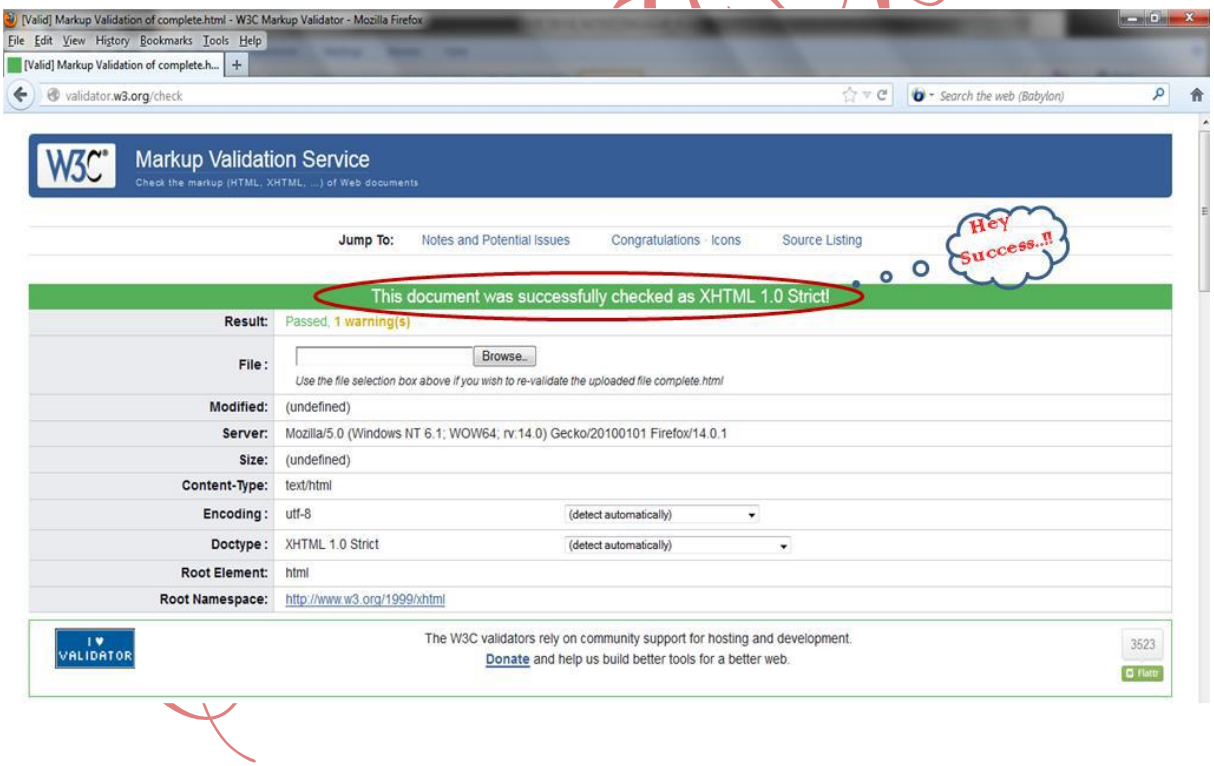
The W3C provides a convenient Web-based way to validate XHTML documents against its standards.

Step 1: The URL of the service is http://validator.w3.org/file-upload.html. Copy & paste this link.

Step 2: You will be driven to "**Validate by File Upload**" option automatically.

Step 3: Browse for a XHTML program file in your computer. (example: *F:/complete.html*)

Step 4: Click on "**More Options**" and select your criteria like *show source* Step 5: After all the settings, click on "**Check**" button Now you will be navigated to another page which shows success or failure. In our example, the file *complete.html* is a valid XHTML file. So the output shows success..!!



HYPERTEXT LINKS

Links:

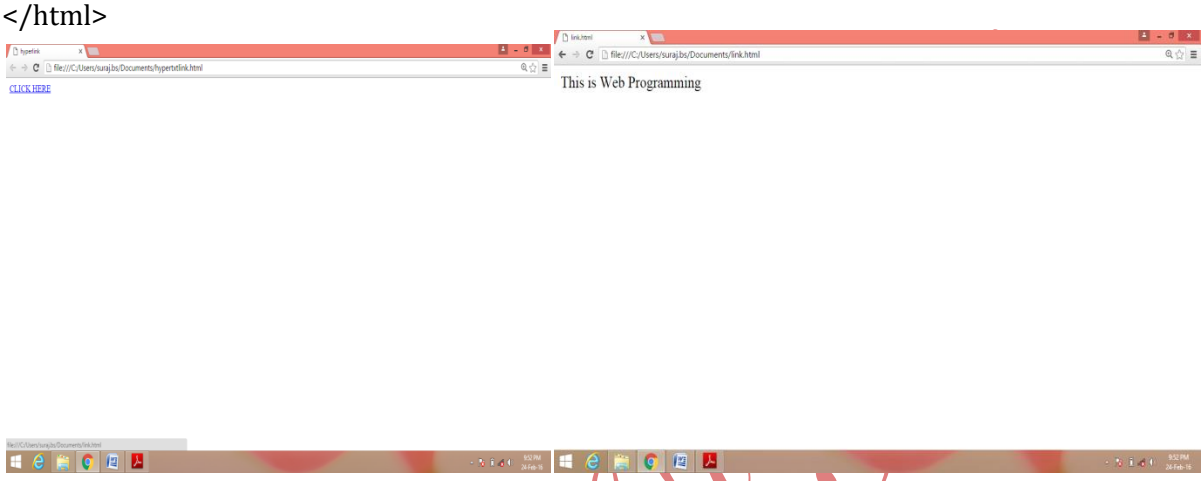
- Hyperlinks are the mechanism which allows the navigation from one page to another.
- The term "hyper" means beyond and "link" means connection
- Whichever text helps in navigation is called hypertext
- Hyperlinks cam be created using **<a>** (anchor tag)
- The attribute that should be used for <a> is **href**

<html>

```
<head>
<title> hyperlink </title>
</head>
<a href = "link.html"> CLICK HERE </a>
</html>
```

Link.html

```
<html>
<body> This is Web Programming </body>
</html>
```



Targets within Documents:

If the target of a link is not at the beginning of a document, it must be some element within the document, in which case there must be some means of specifying it. The target element can include an id attribute, which can then be used to identify it in an href attribute. (observe the scroll bar in the outputs given)

```
<html>
<head>
<title> target link</title>
</head>
<body>
<h1> Puneeth Rajkumar </h1>
<a href="#bottom"> Click Here For His Autobiography </a>
<p><pre>
Appu
Abhi
Veera Kannadiga
Maurya Akaash
Namma Basava
Ajay
Arasu
Milana
Bindaas
Vamshi
Raaj
Raam
Prithvi
Jackie
```

```
Hudugaru
Paramathma
Anna Bond
</pre></p>
<h2> AutoBiography </h2>
<p id = "#bottom"> <pre>
Puneeth Rajkumar was born on 17th of March, 1975.
 His father Dr. Rajkumar is the Legend of Kannada Film Industry.
His mother is Smt. Parvathamma Rajkumar who is a renowned producer in the industry.
 His brothers ShivaRajkumar and RaghavendraRajkumar are very popular heroes.
He is married to Smt. Ashwini Revnath He has two daughters namely Dhrithi and Vanditha..
At present, Puneeth is the greatest star of Kannada Film Industry.
Puneeth Rajkumar was born on 17th of March, 1975.
 His father Dr. Rajkumar is the Legend of Kannada Film Industry.
His mother is Smt. Parvathamma Rajkumar who is a renowned producer in the industry.
 His brothers ShivaRajkumar and RaghavendraRajkumar are very popular heroes.
He is married to Smt. Ashwini Revnath He has two daughters namely Dhrithi and Vanditha..
At present, Puneeth is the greatest star of Kannada Film Industry.
</pre></p>
</body>
</html>
```

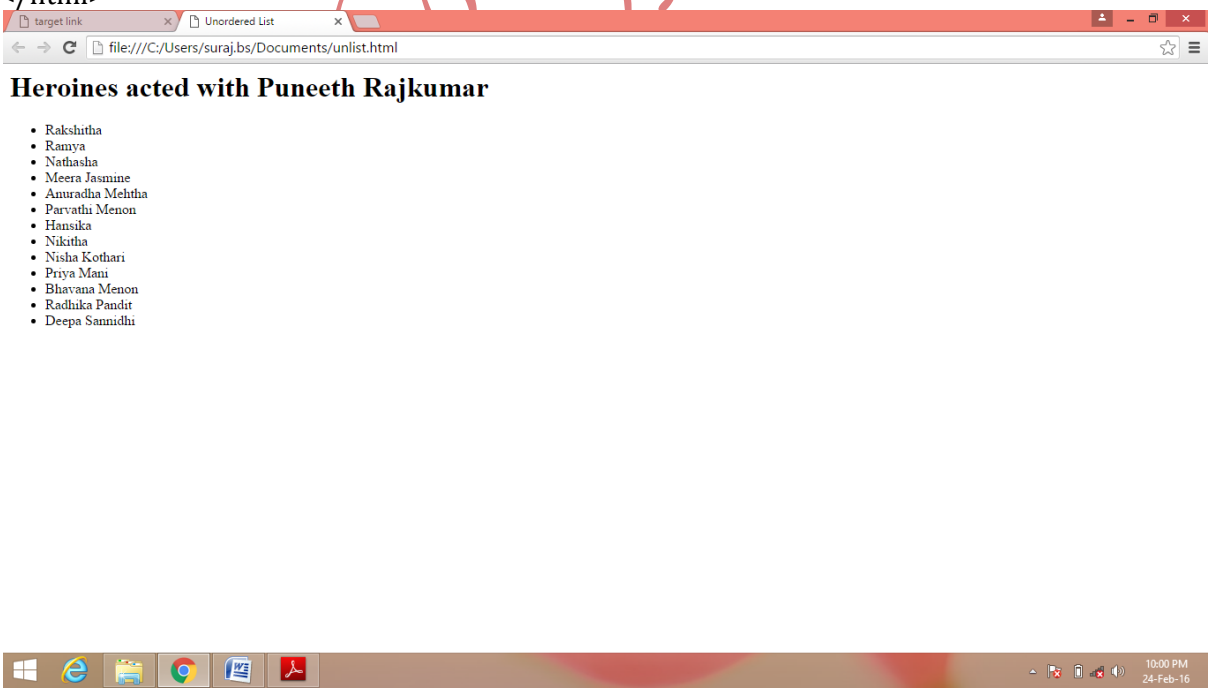
Actually, here we are not creating two separate files, but we are specifying a target within the same document itself. If you click on the above link, you will be redirected to the bottom of the page which contains Autobiography of Puneeth Rajkumar. This is useful for lengthy documents like e-newspaper, e-magazine etc

LISTS

Unordered Lists:

The <ul> tag, which is a block tag, creates an unordered list. Each item in a list is specified with an <li> tag (li is an acronym for *l*ist *i*tem). Any tags can appear in a list item, including nested lists. When displayed, each list item is implicitly preceded by a bullet.

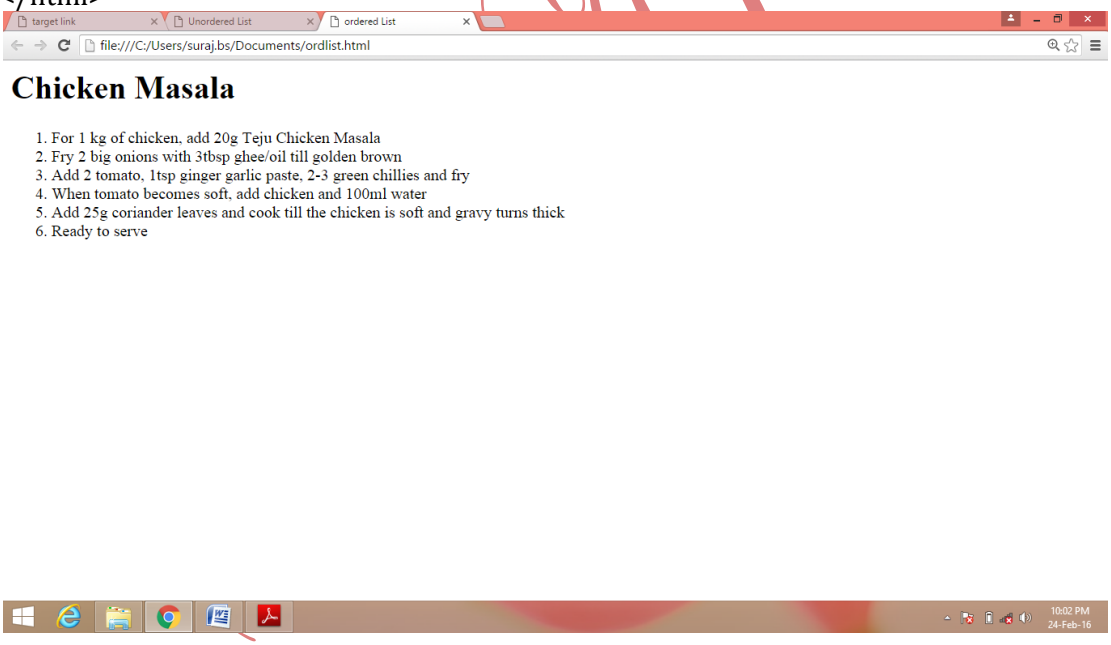
```
<html>
<head>
<title> Unordered List </title>
</head>
<body>
<h1> Heroines acted with Puneeth Rajkumar </h1>
<ul>
 <li>Rakshitha</li>
<li>Ramya</li>
 <li>Nathasha</li>
<li>Meera Jasmine</li>
<li>Anuradha Mehtha</li>
<li>Parvathi Menon</li>
<li>Hansika</li>
<li>Nikitha</li>
<li>Nisha Kothari</li>
 <li>Priya Mani</li>
<li>Bhavana Menon</li>
 <li>Radhika Pandit</li>
<li>Deepa Sannidhi</li>
</ul>
</body>
</html>
```



Ordered Lists:

Ordered lists are lists in which the order of items is important. This orderedness of a list is shown in the display of the list by the implicit attachment of a sequential value to the beginning of each item. The default sequential values are Arabic numerals, beginning with 1. An ordered list is created within the block tag <ol>. The items are specified and displayed just as are those in unordered lists, except that the items in an ordered list are preceded by sequential values instead of bullets.

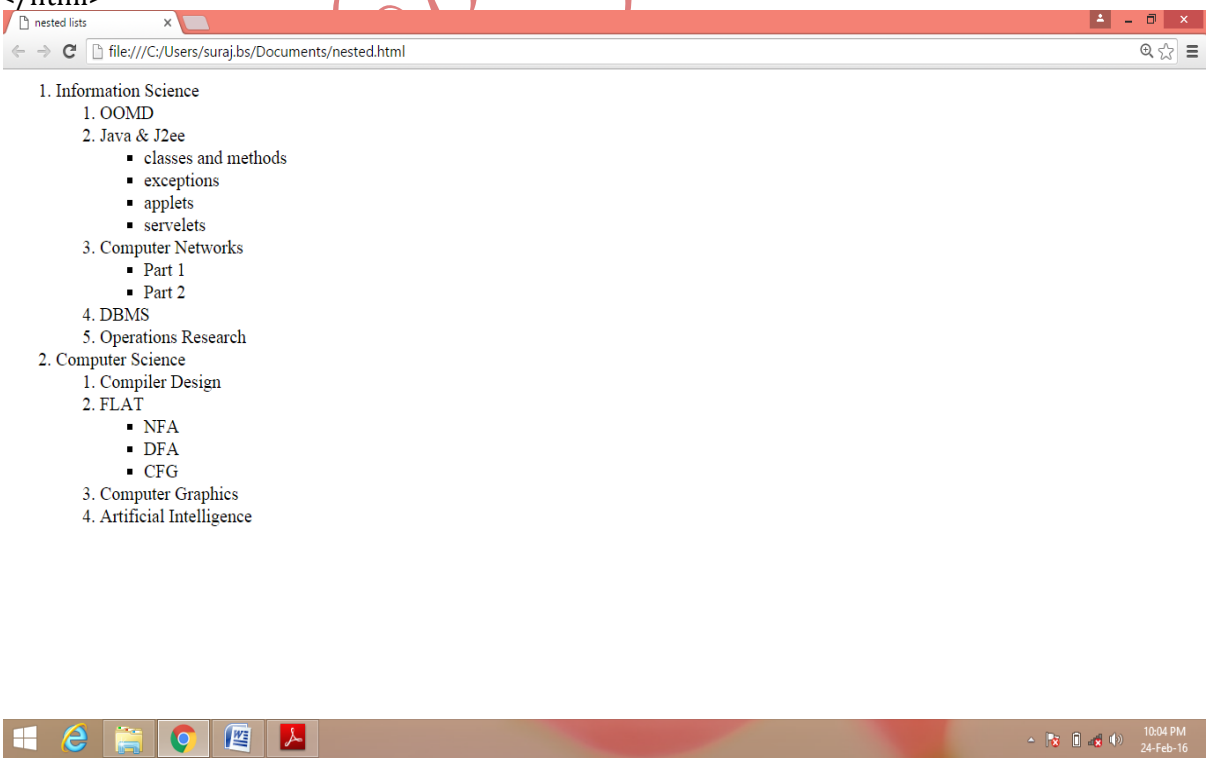
```
<html>
 <head>
<title> ordered List </title>
</head>
<body>
<h1>Chicken Masala</h1>
 <ol>
 <li>For 1 kg of chicken, add 20g Teju Chicken Masala</li>
<li>Fry 2 big onions with 3tbsp ghee/oil till golden brown</li>
<li>Add 2 tomato, 1tsp ginger garlic paste, 2-3 green chillies and fry</li>
<li>When tomato becomes soft, add chicken and 100ml water</li>
 <li>Add 25g coriander leaves and cook till the chicken is soft and gravy turns thick</li>
<li>Ready to serve</li>
 </ol>
 </body>
</html>
```



Nested Lists:

```
<html>
 <head>
 <title> nested lists </title>
 </head>
 <ol>
 <li> Information Science </li>
```

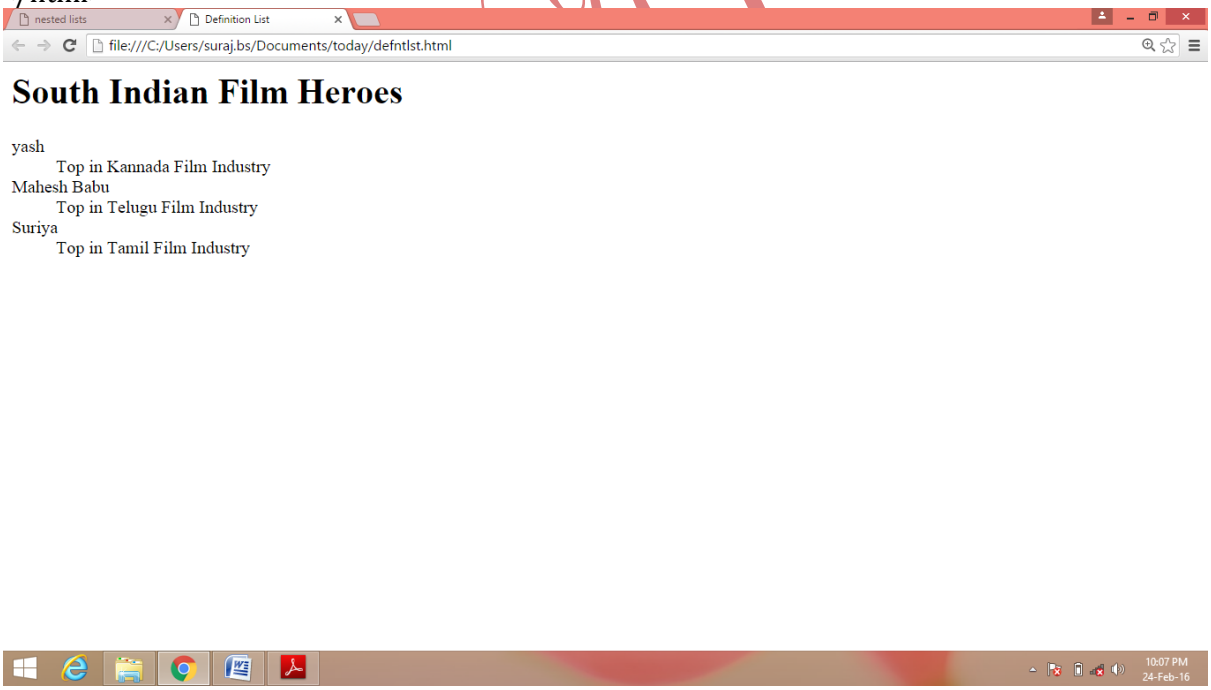
```
<ol> <li>OOMD</li>
 <li>Java & J2ee</li>
<ul>
<li>classes and methods</li>
<li>exceptions</li>
 <li>applets</li>
<li>servelets</li> </ul>
<li>Computer Networks</li>
 <ul>
<li>Part 1</li>
<li>Part 2</li>
 </ul>
<li>DBMS</li>
<li>Operations Research</li>
 </ol>
<li> Computer Science</li>
<ol>
 <li>Compiler Design</li>
<li>FLAT</li>
<ul>
<li>NFA</li>
<li>DFA</li>
<li>CFG</li>
</ul>
<li>Computer Graphics</li>
 <li>Artificial Intelligence</li>
 </ol>
</ol>
</html>
```



Definition Lists:

As the name implies, definition lists are used to specify lists of terms and their definitions, as in glossaries. A definition list is given as the content of a <dl> tag, which is a block tag. Each term to be defined in the definition list is given as the content of a <dt> tag. The definitions themselves are specified as the content of <dd> tags. The defined terms of a definition list are usually displayed in the left margin; the definitions are usually shown indented on the line or lines following the term.

```
<html>
<head>
<title> Definition List </title>
 </head>
<body>
 <h1> South Indian Film Heroes </h1>
 <dl>
<dt> yash </dt>
<dd>Top in Kannada Film Industry</dd>
<dt> Mahesh Babu </dt>
<dd>Top in Telugu Film Industry</dd>
<dt> Suriya </dt>
<dd>Top in Tamil Film Industry</dd>
</dl>
</body>
</html>
```



TABLES

A table is a matrix of cells. The cells in the top row often contain column labels, those in the leftmost column often contain row labels, and most of the rest of the cells contain the data of the table. The content of a cell can be almost any document element, including text, a heading, a horizontal rule, an image, and a nested table.

Basic Table Tags:

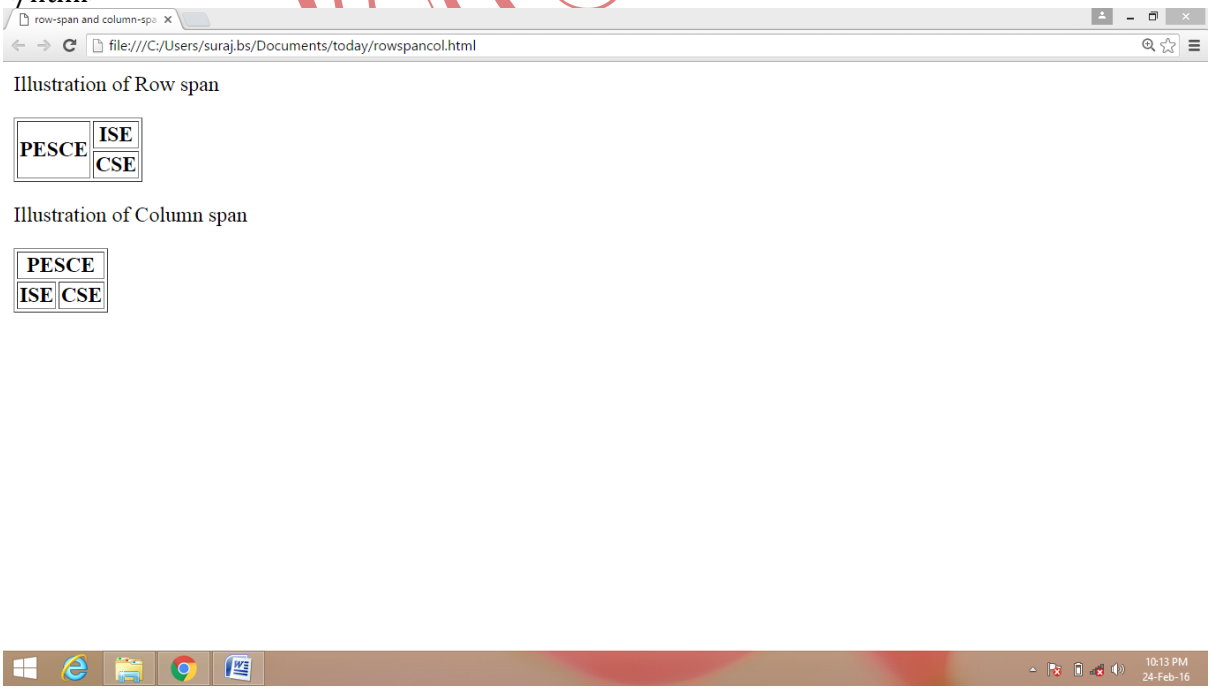
- A table is specified as the content of the block tag **<table>.**
- There are two kinds of lines in tables: the line around the outside of the whole table is called the *border*; the lines that separate the cells from each other are called *rules*.
- It can be obtained using **border** attribute. The possible values are "border" or any number.
- The table heading can be created using **<caption>** tag.
- The table row can be created using **<tr>** tag.
- The column can be created either by using **<th>** tag (stands for table header which is suitable for headings) or **<td>** tag (stands for table data which is suitable for other data).

```
<html>
 <head>
 <title> Table with text and image </title>
</head>
 <body>
<table border = "border">
<caption>PARAMATHMA Movie Details </caption>
<tr>
<th> Name</th>
<th> Image </th>
 </tr>
<tr>
<td> Puneeth Rajkumar </td>
<td> <img src = "puneeth.jpg" alt = "cant display"/></td>
</tr>
<tr>
<td> Deepa Sannidhi</td>
<td> <img src = "deepa.jpg" alt = "cant display"/></td>
 </tr>
</table>
</body> </html>
```

The rowspan and colspan Attributes:

Multiple-level labels can be specified with the rowspan and colspan attributes

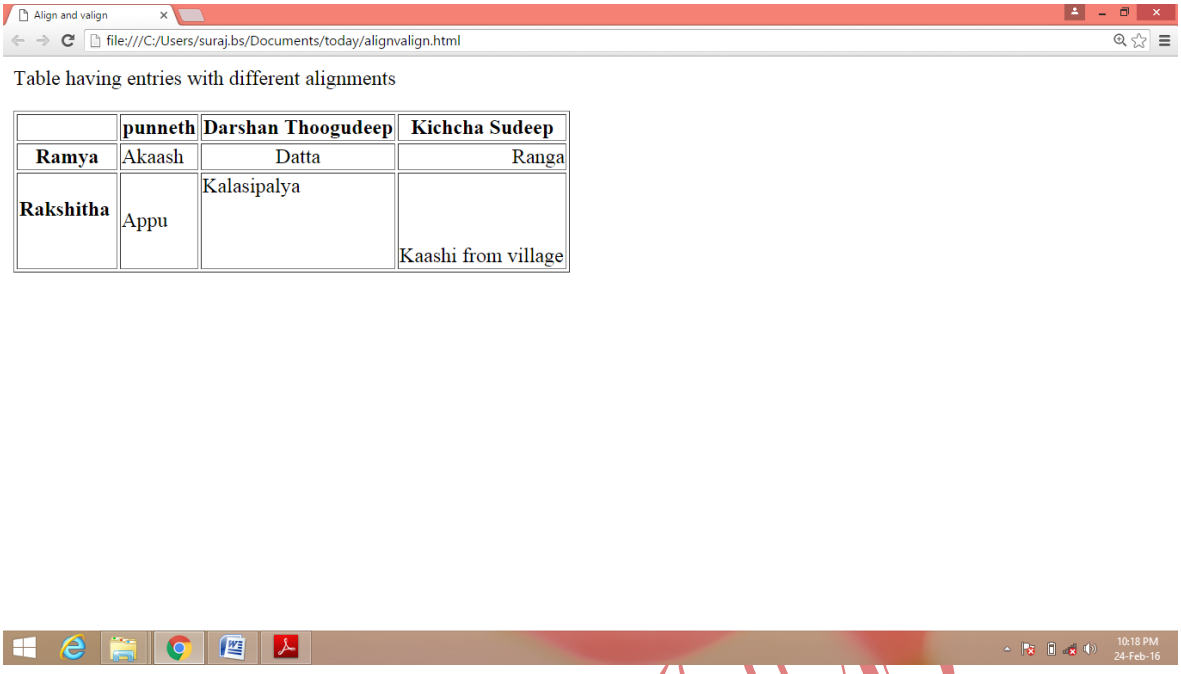
```
<html>
<head>
 <title>row-span and column-span</title>
</head>
<body>
<p> Illustration of Row span</p>
<table border="border">
<tr>
<th rowspan="2"> PESCE </th>
<th>ISE</th>
</tr>
<tr>
<th>CSE</th>
</tr>
</table>
<p> Illustration of Column span</p>
 <table border="border">
<tr>
<th colspan="2"> PESCE </th>
</tr>
<tr>
<th>ISE</th>
 <th>CSE</th>
</tr>
</table>
</body>
</html>
```



The align and valign Attributes:

The placement of the content within a table cell can be specified with the align and valign attributes in the <tr>, <th>, and <td> tags. The align attribute has the possible values left, right, and center, with the obvious meanings for horizontal placement of the content within a cell. The default alignment for th cells is center; for td cells, it is left. The valign attribute of the <th> and <td> tags has the possible values top and bottom. The default vertical alignment for both headings and data is center

```
<html>
 <head>
 <title> Align and valign </title>
</head>
<body>
<p>Table having entries with different alignments</p>
<table border="border">
<tr align = "center">
 <th> </th>
 <th> punneth</th>
 <th> Darshan Thoogudeep</th>
<th> Kichcha Sudeep </th>
</tr>
 <tr>
<th> Ramya </th>
 <td align = "left"> Akaash </td>
<td align = "center"> Datta </td>
<td align = "right"> Ranga </td>
</tr>
<tr>
<th>
<br/>Rakshitha <br/><br/><br/>
</th> <td> Appu </td>
<td valign = "top"> Kalasipalya </td>
 <td valign = "bottom"> Kaashi from village </td>
</tr>
</table>
</body>
 </html>
```



The cellpadding and cellspacing Attributes:

Cellspacing is the distance between cells. Cellpadding is the distance between the edges of the cell to its content

```
<html>
<head>
<title> cell spacing and cell padding </title>
</head> <body>
<h3>Table with space = 10, pad = 50</h3>
<table border = "7" cellspacing = "10" cellpadding = "50">
 <tr>
 <td> Suraj </td>
<td>Haripriya </td>
</tr> </table>
<h3>Table with space = 50, pad = 10</h3>
<table border = "7" cellspacing = "50" cellpadding = "10">
<tr>
 <td> suraj </td>
<td>Ramya</td>
</tr>
</table>
 </body>
</html>
```

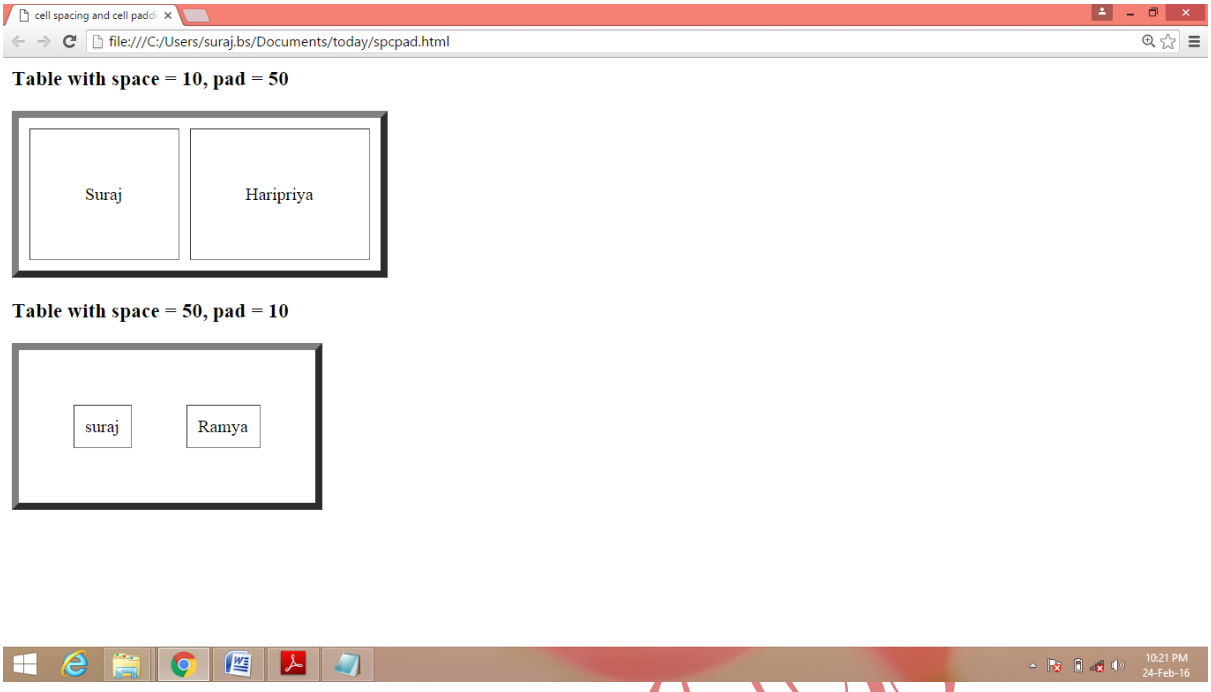


Table Sections:

Tables naturally occur in two and sometimes three parts: header, body, and footer. (Not all tables have a natural footer.) These three parts can be respectively denoted in XHTML with the thead, tbody, and tfoot elements. The header includes the column labels, regardless of the number of levels in those labels. The body includes the data of the table, including the row labels. The footer, when it appears, sometimes has the column labels repeated after the body. In some tables, the footer contains totals for the columns of data above. A table can have multiple body sections, in which case the browser may delimit them with horizontal lines that are thicker than the rule lines within a body section

FORMS

The most common way for a user to communicate information from a Web browser to the server is through a form. XHTML provides tags to generate the commonly used objects on a screen form. These objects are called *controls* or *widgets*. There are controls for single-line and multiple-line text collection, checkboxes, radio buttons, and menus, among others. All control tags are inline tags.

The <form> Tag:

All of the controls of a form appear in the content of a <form> tag. A block tag, <form>, can have several different attributes, only one of which, action, is required. The action attribute specifies the URL of the application on the Web server that is to be called when the user clicks the *Submit* button. Our examples of form elements will not have corresponding application programs, so the value of their action attributes will be the empty string ("").

The <input> Tag:

Many of the commonly used controls are specified with the inline tag <input>, including those for text, passwords, checkboxes, radio buttons, and the action buttons *Reset*, *Submit*, and *plain*.

❖ **Text Box**

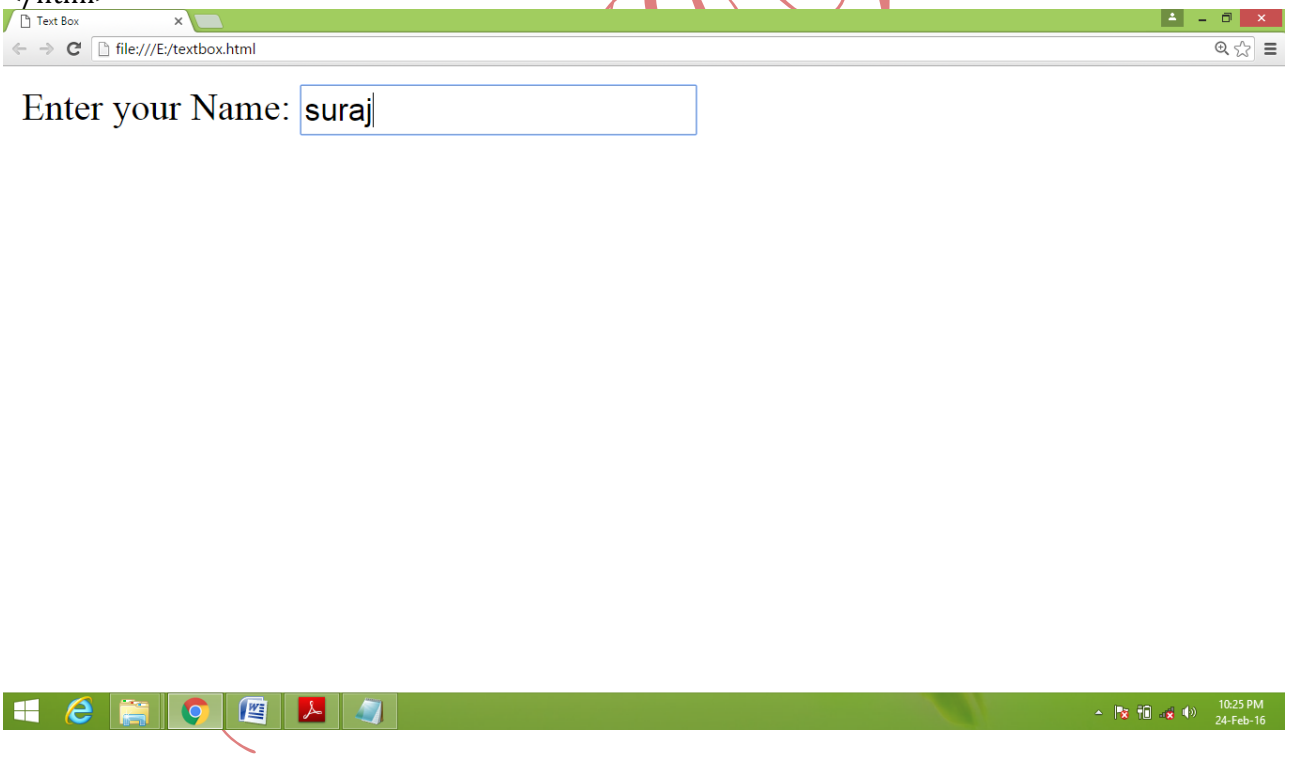
✓ It is a type of input which takes the text.

✓ Any type of input can be created using <input>

✓ The *type* attribute indicates what type of input is needed for the text box, the value should be given as text.

- ✓ For any type of input, a name has to be provided which is done using *name* attribute.
- ✓ The size of the text can be controlled using *size* attribute.
- ✓ Every browser has a limit on the number of characters it can collect. If this limit is exceeded, the extra characters are chopped off. To prevent this chopping, *maxlength* attribute can be used. When maxlength is used, users can enter only those many characters that is given as a value to the attribute.

```
<html>
<head>
<title>Text Box</title>
</head>
<body>
<form action = " ">
 <p>
<label>Enter your Name:
<input type = "text" name = "myname" size = "20"
maxlength = "20" />
</label>
</p>
 </form>
</body>
 </html>
```

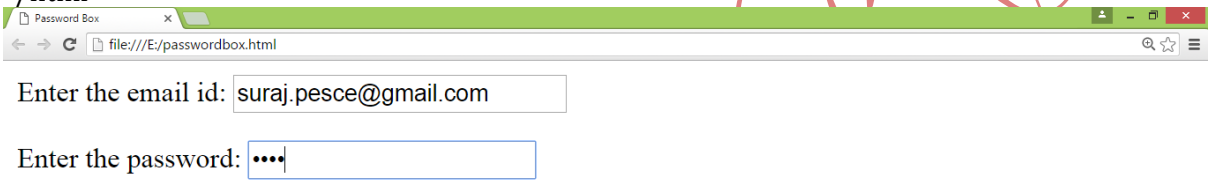


❖ **Password Box**

- ✓ If the contents of a text box should not be displayed when they are entered by the user, a password control can be used.
- ✓ In this case, regardless of what characters are typed into the password control, only bullets or asterisks are displayed by the browser.

```
<html>
<head>
<title>Password Box</title>
```

```
</head>
<body>
<form action = " ">
 <p>
<label>Enter the email id:
<input type = "text" name = "myname" size = "24"
 maxlength = "25" />
</label>
</p>
<p>
<label>Enter the password:
<input type = "password" name = "mypass" size = "20" maxlength = "4" />
 </label>
 </p>
</form>
</body>
</html>
```



❖ **Radio Button**

✓ Radio buttons are special type of buttons which allows the user to select only individual option

✓ Radio buttons are created using the input tag with the *type* attribute having the value **radio.**

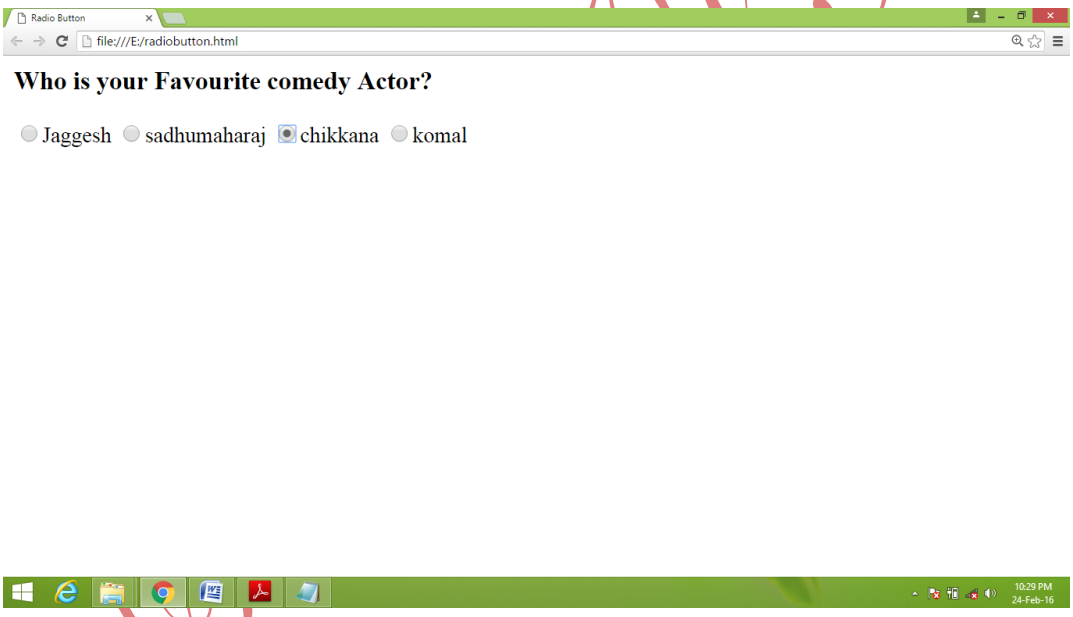
✓ When radio buttons are created, values must be provided with the help of *value* attribute.

✓ All the radio buttons which are created would have same name. This is because the radio buttons are group elements.

✓ If one of the radio buttons has to be selected as soon as the web page is loaded, checked attribute should be used. The value also would be checked.

```
<html>
 <head>
<title>Radio Button</title>
```

```
 </head>
<body>
<h3>Who is your Favourite comedy Actor?</h3>
<form action = " ">
<p>
<label>
<input type="radio" name="act" value="one"/>Jaggesh
</label>
 <label>
<input type="radio" name="act" value="two"/>sadhu maharaj
</label>
<label>
<input type="radio" name="act" value="three"/>chikkana
</label>
<label>
<input type="radio" name="act" value="four"/>komal
</label>
</p>
</form>
 </body>
</html>
```



❖ **Check Box**

✓ Check box is a type of input using which multiple options can be selected.

✓ Check box can also be created using the <input> tag with the *type* having the value "checkbox".

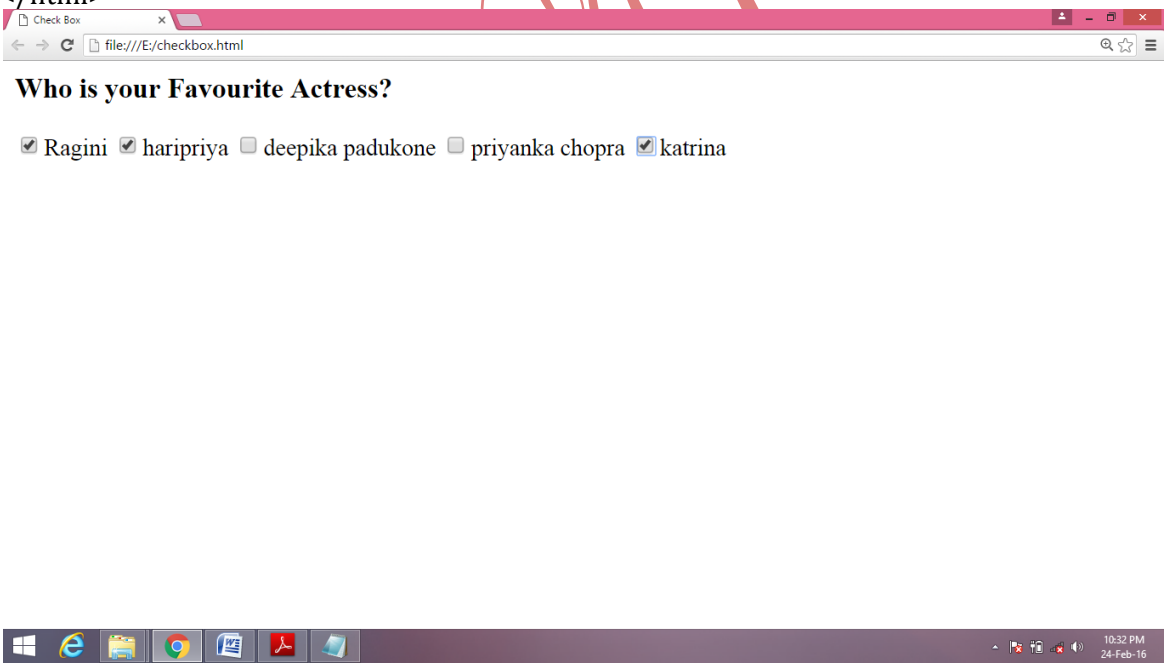
✓ During the creation of check box, the value should be provided using the *value* attribute.

✓ All the checkbox which are created would have the same name because they are group elements.

✓ If one of the check box have to be selected as soon as the page is loaded, checked attribute should be used with the value checked.

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

```
<title>Check Box</title>
 </head>
 <body>
<h3>Who is your Favourite Actress?</h3>
<form action = " ">
<p>
<label>
<input type="checkbox" name="act" value="one"/>Ragini
</label>
<label>
<input type="checkbox" name="act" value="two"/>haripriya
</label>
<label>
<input type="checkbox" name="act" value="three"/>deepika padukone
</label>
<label>
<input type="checkbox" name="act" value="four"/>priyanka chopra
</label>
<label><input type="checkbox" name="act" value="four"/>katrina
</label>
 </p>
</form>
</body>
</html>
```

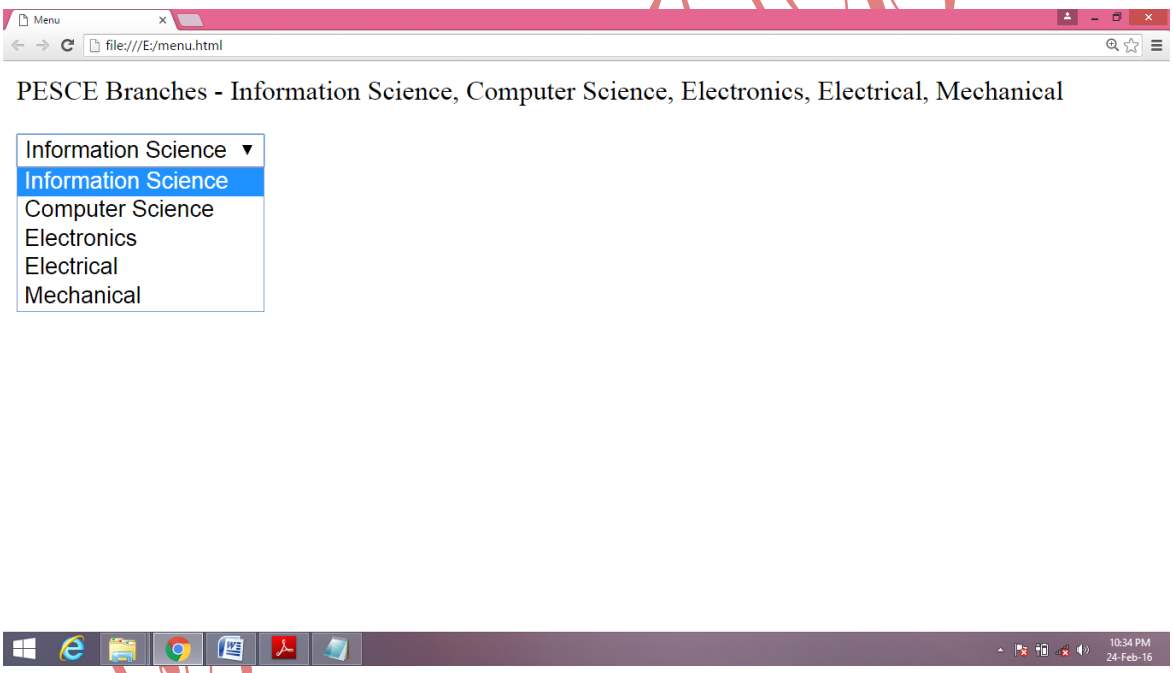


The <select> Tag:

- Menu items is another type of input that can be created on the page.
- To create the menu item, <select> tag is used.
- To insert the item in the menu, <option> tag is used

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

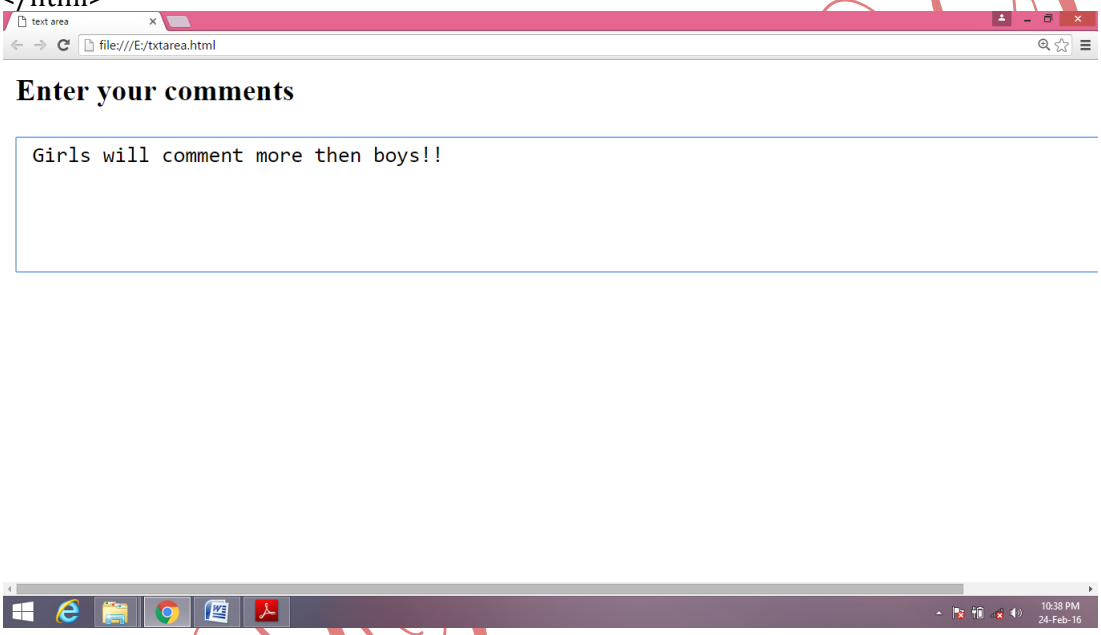
```
<title> Menu </title>
 </head>
<body>
<p> PESCE Branches - Information Science, Computer Science, Electronics,
Electrical, Mechanical
</p>
<form action = "">
<p>
 <select name = "branches" size="1">
<option> Information Science </option>
 <option> Computer Science </option>
<option> Electronics </option>
<option> Electrical </option>
<option> Mechanical </option>
 </select>
 </p>
</form>
</body>
</html>
```



The <textarea> Tag:

- Text area is a type of input using which multiple statements can be entered.
- Text area is created using <textarea> tag.
- Text area should have the name.
- During the creation of text area, it should be mentioned how many sentences can be entered. This is done using *rows* attribute.
- Similarly, it should also be mentioned how many characters can be entered in a line. This is done using *cols* attribute.
- If the value given to rows is exceeded i.e. if users enter sentences more than specified, the *scroll bar* automatically appears.

```
<html>
<head>
 <title> text area </title>
</head>
<body>
<form action=" ">
<h3> Enter your comments</h3>
 <p>
<textarea name="feedback" rows="5" cols="100">
</textarea>
</p>
 </form>
</body>
</html>
```



The Action Buttons:

The *Reset* button clears all of the controls in the form to their initial states. The *Submit* button has two actions: First, the form data is encoded and sent to the server; second, the server is requested to execute the server-resident program specified in the action attribute of the <form> tag. The purpose of such a server-resident program is to process the form data and return some response to the user. Every form requires a *Submit* button. The *Submit* and *Reset* buttons are created with the <input> tag.

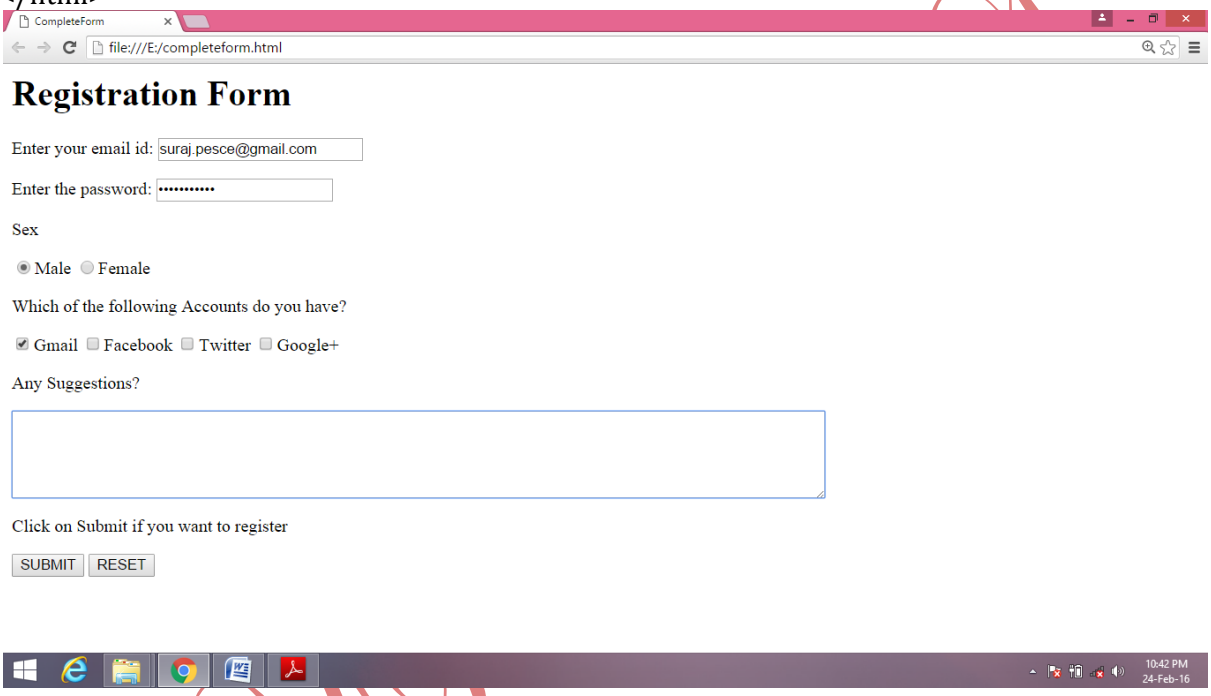
```
<html>
 <head>
 <title> action buttons </title>
</head>
<body>
<form action=" ">
<p> <input type="SUBMIT" value="SUBMIT"/>
 <input type="RESET" value="RESET"/>
</p>
</form>
 </body>
```

```
</html>
```

Example of a Complete Form:

```
<html>
<head>
<title> CompleteForm</title>
</head>
 <body>
<h1>Registration Form</h1>
<form action=" ">
<p>
<label>Enter your email id:
<input type = "text" name = "myname" size = "24" maxlength = "25" />
</label>
 </p>
<p>
 <label>Enter the password: <input type = "password" name = "mypass"
size = "20" maxlength = "20" />
</label>
</p>
<p>Sex</p>
<p>
<label><input type="radio" name="act" value="one"/>Male
</label>
<label>
<input type="radio" name="act" value="two"/>Female
</label>
</p>
<p>Which of the following Accounts do you have?</p>
<p>
<label>
<input type="checkbox" name="act" value="one"/>Gmail
</label>
<label>
<input type="checkbox" name="act" value="two"/>Facebook
</label>
<label>
<input type="checkbox" name="act" value="three"/>Twitter
</label>
<label><input type="checkbox" name="act" value="four"/>Google+
</label>
```

```
</p>
<p> Any Suggestions?</p>
<p>
<textarea name="feedback" rows="5" cols="100">
</textarea>
</p>
<p>Click on Submit if you want to register</p> <p>
<input type="SUBMIT" value="SUBMIT"/> <input type="RESET" value="RESET"/>
</p>
</form>
</body>
</html>
```



FRAMES

The browser window can be used to display more than one document at a time. The window can be divided into rectangular areas, each of which is a *frame.* Each frame is capable of displaying its own document.

Framesets:

- The number of frames and their layout in the browser window are specified with the <frameset> tag.
- A frameset element takes the place of the body element in a document. A document has either a body or a frameset but cannot have both.
- The <frameset> tag must have either a *rows* or a *cols* attribute. (or both)
- To create horizontal frames, *rows* attribute is used.
- To create vertical frames, *cols* attribute is used.
- The values for these attributes can be numbers, percentages and asterisks.

To Demonstrate Horizontal Frames using *rows* Attribute

```
<html>
<head>
<title>Frameset Rows</title>
```

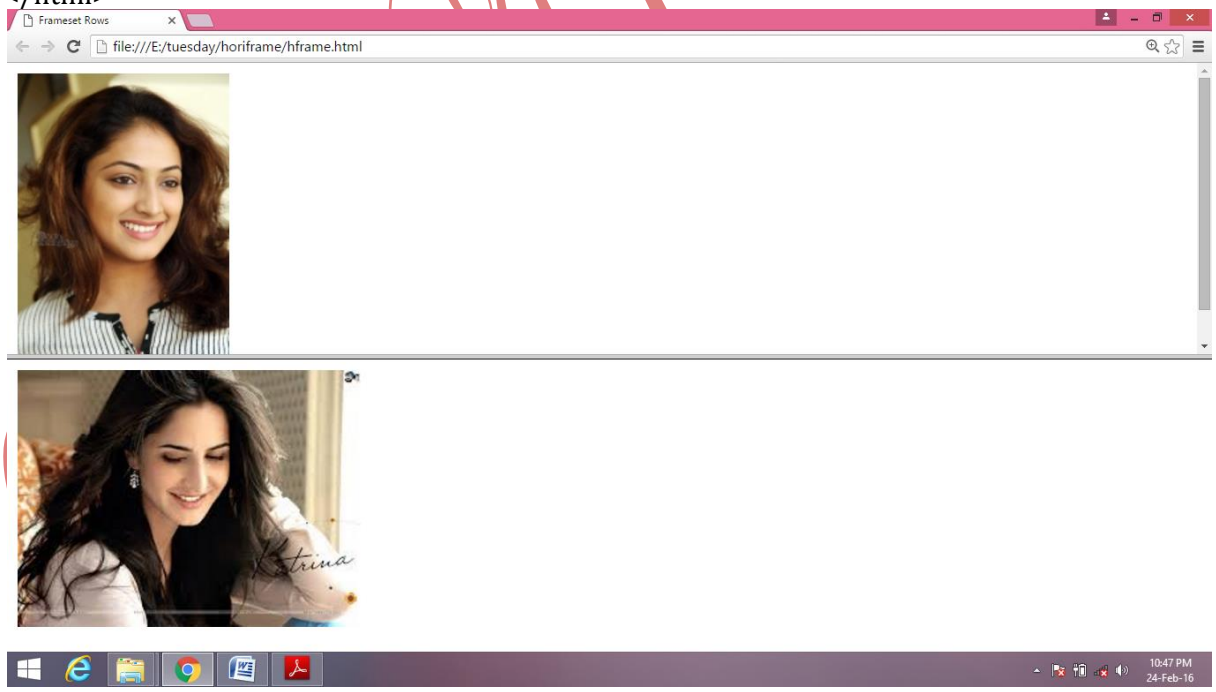
```
 </head>
 <frameset rows = "*,*">
<frame src = "img1.html"/>
<frame src = "img2.html"/>
</frameset>
</html>
```

img1.html

```
<html>
<head>
<title>frame row 1</title>
 </head>
<body>
 <img src="hr1.jpg" alt="cannot display"/>
 </body>
</html>
```

Img2.html

```
<html>
<head>
<title>frame row 2</title>
 </head>
<body>
<img src="kk1.jpg" alt="cannot display"/>
 </body>
</html>
```



Vertical frame

```
<html>
 <head>
<title>Frameset Cols</title>
```

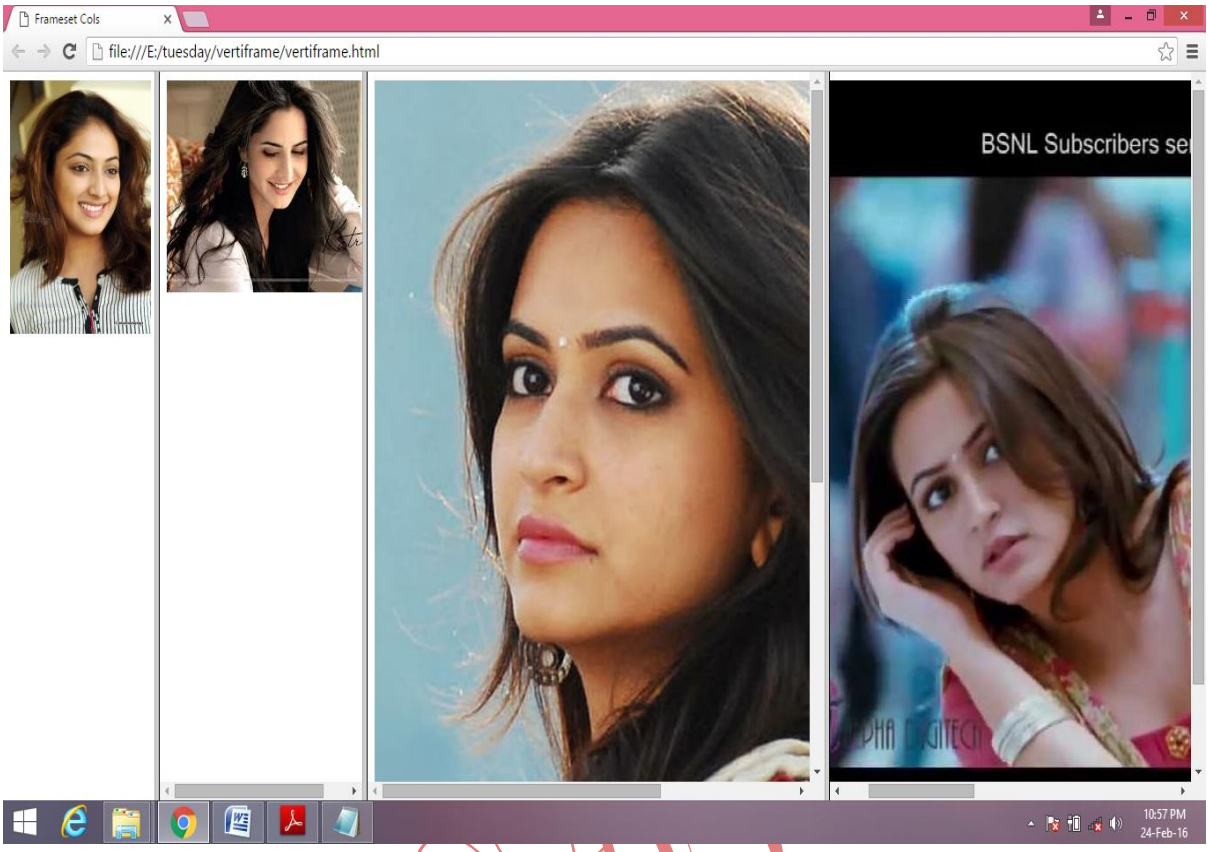
```
</head>
<frameset cols = "50%,50%,25%,25%">
<frame src = "img1.html"/>
 <frame src = "img2.html"/>
 <frame src = "img3.html"/>
<frame src = "img4.html"/>
</frameset>
</html>
```

```
//img1.html
<html>
<head>
<title>frame row 1</title>
 </head>
<body>
 <img src="hr1.jpg" alt="cannot display"/>
 </body>
</html>
```

```
//img2.html
<html>
<head>
<title>frame row 2</title>
 </head>
<body>
<img src="kk1.jpg" alt="cannot display"/>
 </body></html>
```

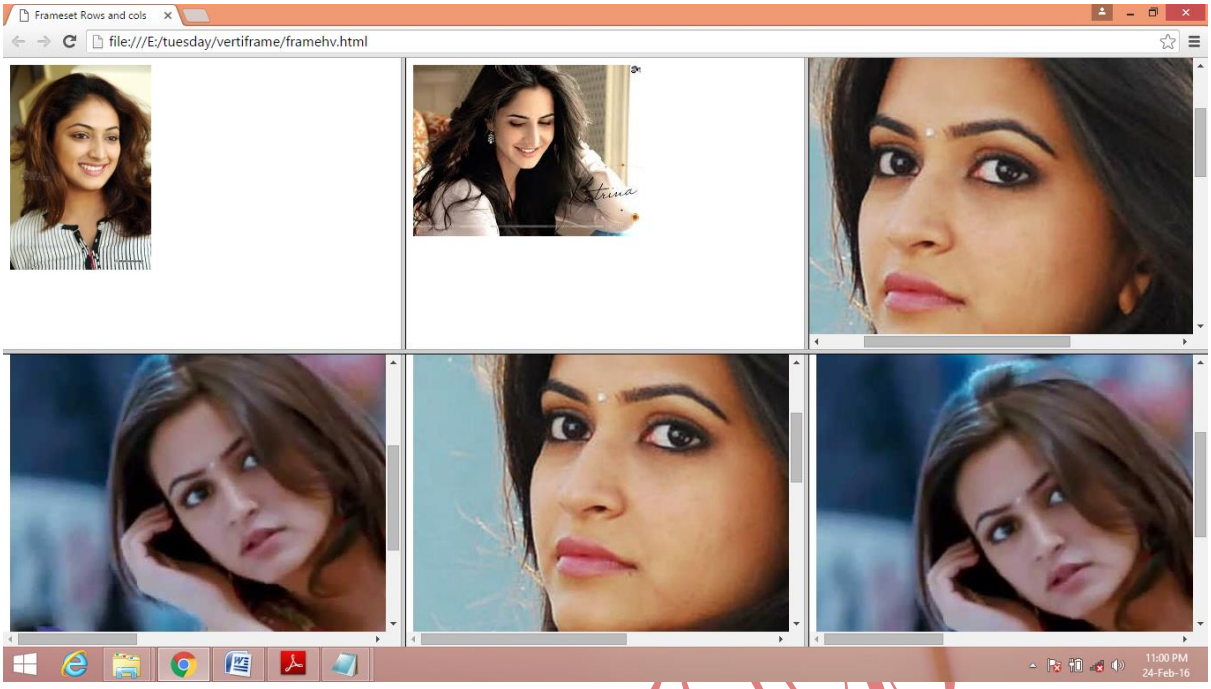
```
//img3.html
<html>
<head>
<title>frame row 2</title>
 </head>
<body>
<img src="kr1.jpg" alt="cannot display"/>
 </body></html>
```

```
//img4.html
<html>
<head>
<title>frame row 2</title>
 </head>
<body>
<img src="kr2.jpg" alt="cannot display"/>
 </body></html>
```



Frames rows and columns

```
<html>
<head>
<title>Frameset Rows and cols</title>
</head>
<frameset rows = "50,50" cols = "*,*,*">
<frame src = "img1.html"/>
<frame src = "img2.html"/>
<frame src = "img3.html"/>
<frame src = "img4.html"/>
<frame src = "img3.html"/>
<frame src = "img4.html"/>
 </frameset>
 </html>
```



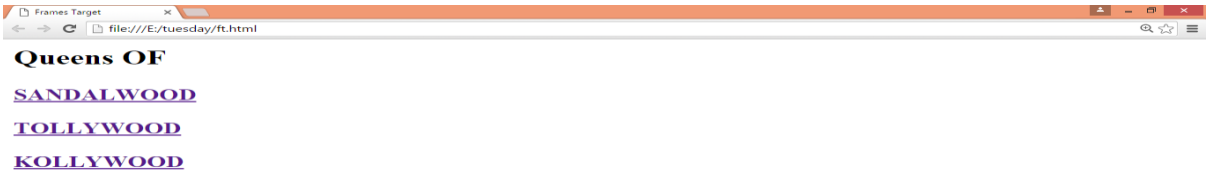
Frame target

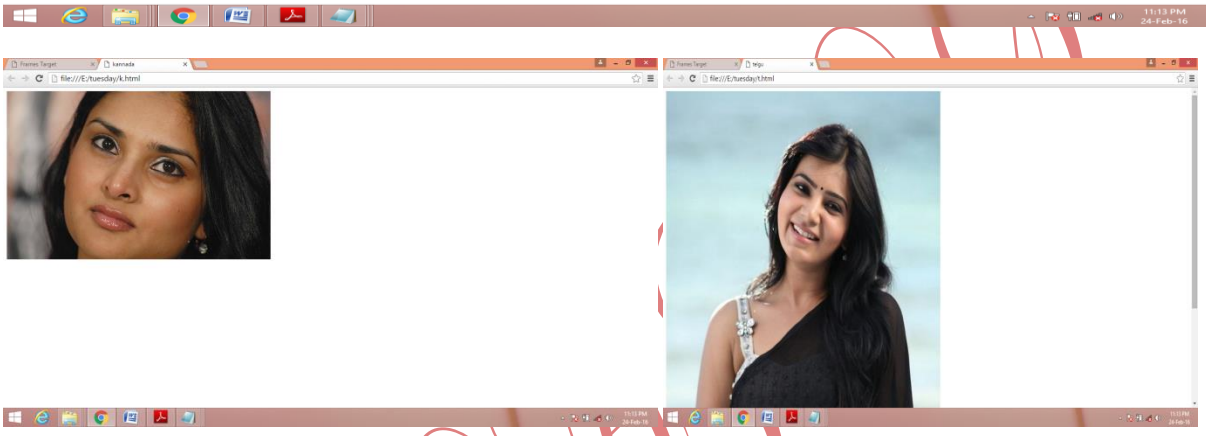
```
<html>
 <head>
 <title>Frames Target</title>
 </head>
<body>
<h2>Queens OF</h2>
<h3> <a href="k.html" target = "description"> SANDALWOOD</a>
 </h3> <h3> <a href="t.html" target = "description"> TOLLYWOOD</a>
 </h3> <h3> <a href="m.html" target = "description"> KOLLYWOOD</a>
 </h3>
</body>
</html>
```

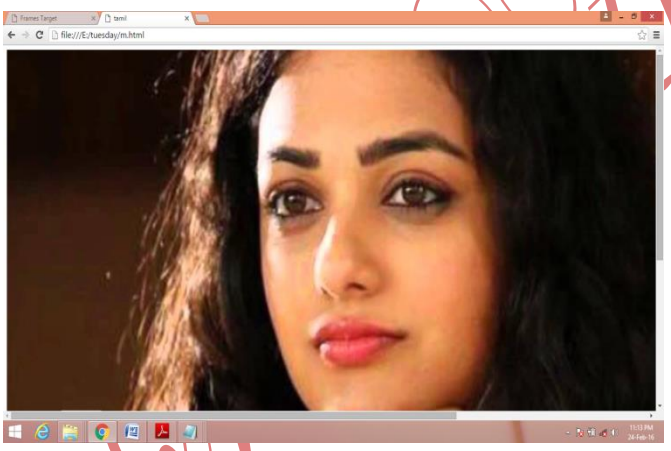
```
<html>
<head>
<title>kannada</title>
 </head>
<body>
<img src="ramya.jpg" alt="cannot
display"/>
 </body>
</html>
```

```
<html>
<head>
<title>tamil</title>
 </head>
<body>
<img src="nitya.jpg" alt="cannot
display"/>
 </body>
</html>
```

```
<html>
<head>
<title>telgu</title>
 </head>
<body>
<img src="samantha.jpg"
alt="cannot display"/>
 </body>
</html>
```


Frames Target
file:///E:/tuesday/ft.html
Queens OF
SANDALWOOD
TOLLYWOOD
KOLLYWOOD





SYNTACTIC DIFFERENCES BETWEEN HTML AND XHTML

PARAMETERS	HTML	XHTML
Case Sensitivity	Tags and attributes names are case insensitive	Tags and attributes names must be in lowercase
Closing tags	Closing tags may be omitted	All elements must have closing tag
Quoted attribute values	Special characters are quoted. Numeric values are rarely quoted.	All attribute values must be quoted including numbers
Explicit attribute values	Some attribute values are implicit. For example: <table border>. A default value for border is assumed	All attribute values must be explicitly stated
***id* and *name* attributes**	Both *id* and *name* attributes are encouraged	Use of *id* is encouraged and use of *name* is discouraged
Element nesting	Rules against improper nesting of elements (for example: a form element cannot contain another form element) are not enforced.	All nesting rules are strictly enforced

QUESTION BANK OF UNIT 1

1. Explain the tasks of DNS name server
2. Explain response phase of http
3. Give syntax and an example for each of the following tags. 1.<pre> 2.<a> 3.<img> 4.<sub> 5.<p>
4. Explain at least two uses of the following. 1. Perl 2.php 3.java script 4.xml 5.MIME type specification
5. Explain with an example the following tags 1.select 2.frames 3.colspan 4.radio button 5.style class selector
6. How does domain name conversion happen on web? Describe the concept with a figure by taking a suitable example.
7. Give and explain response and request phases of hyper text transfer protocol
8. What is the purpose of MIME type specification in request or response transaction between browser and server
9. Give and explain syntax of following tags 1.<blockquote> 2.meta
10. Explain web server operations and general server characteristics
11. Explain two web programmers tool used in web programming
12. Describe a fully qualified domain name and explain how fully qualified domain names are translated into IP
13. What is HTTP? Explain its phases in detai
14. Explain the following tags with examples: <img> and <a>
15. Write an XHTML document to describe an ordered list of your five favourite movies. Each element of the list must have a nested list of atleast two actors in your favourite movies.
16. With examples, explain a style class selector.
17. Write an XHTML document that has six short paragraphs of text. Define three different paragraph styles p1, p2 and p3. The p1 style must use left and right margins of 20 pixels, a background colour of yellow, and a foreground color of blue . The p2 style must use font size of 18 points, font name 'Arial' and a font style in italic form. The p3 style must use a text indent of 1 centimeter , a background color of green, and a foreground color of white. The 1st and the 4th paragraph must use p1, the 2nd and 5th must use p2 and the 3rd and 6th must use p3.
18. Explain the following with respect to table creation in XHTML documents: Align and valign attributes tr, th and td attributes Rowspan and Colspan attributes Cell padding and Cell spacing attributes
19. Create XHTML document to describe a table with the following contents: The columns of the table must have the headings pine, maple, Oak and fir. The rown must have the labels average height, average width, typical lifespan, and leaf type. Fill the data cells with some values.
20. Create , test and validate a XHTML document that has a form with Three text boxes to collect user name and address. Tables with the headings product name , price and quantity and the values are 100—watts light bulb, $2.39 , 4 200—watts light bulb, $4.29 , 8 100—watts long life light bulbs, $3.95 , 4 200—watts long life light bulbs, $7.49 , 8 A collection of 4 radio buttons that are labeled as Visa Master card Discover Check A submit and a reset button
21. Explain the syntactic differences between HTML and XHTML
22. Create XHTML document that has two frames. The left frame displays contents.html and the right frame displays cars.html where the second frame is a target of link from the first frame. [Note: contents.html is a list of links to the cars description.]

23. What tag and attribute are used to describe a link? Discuss about it.
24. Explain all controls that are created with the <input> tag with examples, which are used for text collection
25. Explain the XHTML tags used for lists in documents.
26. Write an XHTML document to describe an ordered list of four states. Each element of the list must have an unordered list of atleast two cities in the state
27. Explain the following , with respect to table creation in XHTML documents. <table> tr, th and td attributes rowspan and colspan attributes align and valign attributes cell padding and cell spacing
28. Write an XHTML program to create a link within a document
29. Create XHTML document that defines a table with 5 rows and 5 columns. The first row should contain country name, gold, silver, bronze (all three indicating the type of medals) and total in each column respectively. Fill in the information details in the table with appropriate values. After filling the details, set red color to the background for the first row, blue for the second, yellow for the third, purple for the fourth and green for the fifth row. Use of align and valign attributes for this table has to be made at the appropriate places