

II MCA –III SEM WEBTECHNOLOGIES(MCA3103)

UNIT-I:Web Basics – Introduction, Concept of Internet-History of Internet, Protocols of Internet, World Wide Web, URL, Web Server, Web Browser. **HTML-** Introduction, History of HTML, Structure of HTML Document, Text Basics, Images and Multimedia, Links and Webs, Document Layout, Creating Forms, Frames and Tables, Cascading Style Sheets.

Internet: The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services

History of Internet:

This marvelous tool has quite a history that holds its roots in the cold war scenario. A need was realized to connect the top universities of the United States so that they can share all the research data without having too much of a time lag. This attempt was a result of Advanced Research Projects Agency (ARPA) which was formed at the end of 1950s just after the Russians had climbed the space era with the launch of Sputnik. After the ARPA got success in 1969, it didn't take the experts long to understand that how much potential can this interconnection tool have. In 1971 Ray Tomlinson made a system to send electronic mail. This was a big step in the making as this opened gateways for remote computer accessing i.e.telnet. During all this time, rigorous paper work was being done in all the elite research institutions. From giving every computer an address to setting out the rules, everything was getting penned down. 1973 saw the preparations for the vital TCP/IP and Ethernet services. At the end of 1970s, Usenet groups had surfaced up. By the time the 80s had started, IBM came up with its PC based on Intel 8088 processor which was widely used by students and universities for it solved the purpose of easy computing. By 1982, the Defense Agencies made the TCP/IP compulsory and the term —internetll was coined. The domain name services arrived in the year 1984 which is also the time around which various internet based marked their debut. A worm, or a rust the computers, attacked in 1988 and disabled over 10% of the computer systems all over the world. While most of the researchers regarded it as an opportunity to enhance computing as it was still in its juvenile phase, quite a number of computer companies became interested in dissecting the cores of the malware which resulted to the formation Computer Emergency Rescue Team (CERT). Soon after the world got

over with the computer worm, World Wide Web came into existence. Discovered by Tim Berners-Lee, World Wide Web was seen as a service to connect documents in websites using hyperlinks.

Internet/web Protocols: Web protocols are set of rules followed by everyone communicating over the web.

HTTP: The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

A web browser may be the client, and an application on a computer that hosts a web site may be the server.

Other Protocols:

TCP/IP Model: TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP is also used as a communications protocol in a private computer network (an intranet or extranet).

UDP: The User Datagram Protocol, or UDP, is a communication protocol used across the Internet for especially time-sensitive transmissions such as video playback or DNS lookups. It speeds up communications by not formally establishing a connection before data is transferred.

FTP: File Transfer Protocol(FTP) is an application layer protocol that moves files between local and remote file systems. It runs on top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.

SMTP: SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly.

SOAP: Simple Object Access Protocol (SOAP) is a lightweight XML-based protocol that is used for the exchange of information in decentralized, distributed application environments. You can

transmit SOAP messages in any way that the applications require, as long as both the client and the server use the same method.

World Wide Web :

The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland. The Web browser was released outside CERN in 1991, first to other research institutions starting in January 1991 and to the general public on the Internet in August 1991. The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet. Web pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML). In addition to formatted text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content. Embedded hyperlinks permit users to navigate between web pages. Multiple web pages with a common theme, a common domain name, or both, make up a website. Website content can largely be provided by the publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be mostly informative, primarily for entertainment, or largely for commercial, governmental, or non-governmental organizational purposes.



WWW is another example of client/server computing. Each time a link is followed, the client is requesting a document (or graphic or sound file) from a server (also called a Web server) that's part of the World Wide Web that "serves" up the document. The server uses a

protocol called HTTP or Hyper Text Transport Protocol. The standard for creating hypertext documents for the WWW is Hyper Text Markup Language or HTML. HTML essentially codes plain text documents so they can be viewed on the Web.

Web Browsers:

WWW Clients, or "Browser": The program you use to access the WWW is known as a browser because it "browses" the WWW and requests these hypertext documents. Browsers can be graphical, allows to see and hear the graphics and audio; text-only browsers (i.e., those with no sound or graphics capability) are also available. All of these programs understand http and other Internet protocols such as FTP, gopher, mail, and news, making the WWW a kind of "one stop shopping" for Internetusers.

Year	List of Web browsers
1991	World Wide Web (Nexus)
1992	Viola WWW, Erwise, MidasWWW, MacWWW (Samba)
1993	Mosaic,Cello,[2] Lynx 2.0, Arena, AMosaic 1.0
1994	IBM WebExplorer, Netscape Navigator, SlipKnot 1.0, MacWeb, IBrowse, Agora (Argo), Minuet
1995	Internet Explorer 1, Internet Explorer 2, Netscape Navigator 2.0, OmniWeb, UdiWWW, Grail
1996	Arachne 1.0, Internet Explorer 3.0, Netscape Navigator 3.0,Opera 2.0, PowerBrowser 1.5,[4] Cyberdog,Amaya 0.9,[5] AWeb,Voyager
1997	Internet Explorer 4.0, Netscape Navigator 4.0, Netscape Communicator 4.0, Opera3.0,[6] Amaya 1.0[5]
1998	iCab, Mozilla
1999	Amaya 2.0,[5] Mozilla M3, Internet Explorer 5.0
2000	Konqueror,Netscape 6, Opera 4,[7] Opera 5,[8] K-Meleon 0.2, Amaya 3.0,[5] Amaya 4.0[5]

2001	Internet Explorer 6, Galeon 1.0, Opera 6,[9] Amaya 5.0[5]
2002	Netscape 7, Mozilla 1.0, Phoenix 0.1, Links 2.0, Amaya 6.0,[5] Amaya 7.0[5]
2003	Opera 7,[10] Apple Safari 1.0, Epiphany 1.0, Amaya 8.0[5]
2004	Firefox 1.0, Netscape Browser, OmniWeb 5.0
2005	Opera8,[11]Apple Safari2.0, Netscape Browser 8.0, Epiphany 1.8, Amaya 9.0,[5] AOL Explorer 1.0, Maxthon 1.0,Shitira 1.0
2006	Mozilla Firefox 2.0, Internet Explorer 7,Opera 9,[12], SeaMonkey 1.0, K-Meleon 1.0, Galeon 2.0, Camino 1.0, Avant11, iCab 3
2007	Apple Safari 3.0, Maxthon 2.0, Netscape Navigator9,NetSurf 1.0, Flock 1.0, Conkeror
2008	Google Chrome 1, Mozilla Firefox 3, Opera 9.5,[13], Apple Safari 3.1, Konqueror 4, Amaya 10.0,[5] Flock 2, Amaya 11.0[5]
2009	Google Chrome 2–3, Mozilla Firefox 3.5, Internet Explorer 8,Opera 10,[14], Apple Safari 4, SeaMonkey 2, Camino 2,surf, Pale Moon 3.0[15]
2010	Google Chrome 4–8, Mozilla Firefox 3.6, Opera 10.50,[16], Opera 11, Apple Safari 5, K-Meleon 1.5.4,
2011	Google Chrome 9–16, Mozilla Firefox 4-9, Internet Explorer 9,Opera 11.50, Apple Safari 5.1, Maxthon 3.0, SeaMonkey 2.1–2.6
2012	Google Chrome 17–23, Mozilla Firefox 10–17, Internet Explorer 10, Opera 12, Apple Safari 6, Maxthon 4.0, SeaMonkey 2.7-2.14
2013	Google Chrome24–31,Mozilla Firefox 18–26,Internet Explorer 11, Opera 15–18, Apple Safari 7, SeaMonkey 2.15-2.23

2014	Google Chrome 32–39, Mozilla Firefox 27–34, Opera 19–26, Apple Safari 8
2015	Google Chrome 40–47, Microsoft Edge,Mozilla Firefox 35–43, Opera 27–34, Vivaldi
2016	Google Chrome 48–55,Mozilla Firefox 44–50,Microsoft Edge 14, Opera35–42, Apple Safari 10, SeaMonkey 2.24–2.30, Pale Moon 26.0.0[17], Pale Moon 27.0.0[18]
2017	Google Chrome56–60,Microsoft Edge 15,Mozilla Firefox 51–55.0.2, Opera43–45, Opera Neon

Uniform Resource Locators, or URLs:

A Uniform Resource Locator, or URL is the address of a document found on the WWW. Browser interprets the information in the URL in order to connect to the proper Internet server and to retrieve your desired document. Each time a click on a hyperlink in a WWW document instructs browser to find the URL that's embedded within the hyperlink. The elements in a URL:

Protocol://server's address/filename Hypertext protocol: http://www.aucegypt.edu File Transfer Protocol: ftp://ftp.dartmouth.edu Telnet Protocol: telnet://pac.carl.org News Protocol: news:alt.rock-n-roll.stones What are Domains? Domains divide World Wide Web sites into categories based on the nature of their owner, and they form part of a site's address, or uniform resource locator (URL).

Common top-level domains are:

.com—commercial enterprises	.mil—military site
org—organization site (non-profits, etc.)	int—organizations established by international treaty
.net—network	.biz—commercial and personal
.edu—educational site (universities, schools, etc.)	.info—commercial and personal
.gov—government organizations	.name—personal sites

Additional three-letter, four-letter, and longer top-level domains are frequently added. Each country linked to the Web has a two-letter top-level domain, for example .fr is France, .ie is Ireland. MIME (Multi-Purpose Internet Mail Extensions):- MIME is an extension of the original Internet e-mail protocol that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds, as well as the ASCII text handled in the original protocol, the Simple Mail Transport Protocol (SMTP). In 1991, Nathan Borenstein of Bellcore proposed to the IETF that SMTP be extended so that Internet clients and servers could recognize and handle other kinds of data than ASCII text. As a result, new file types were added to "mail" as a supported Internet Protocol file type. Servers insert the MIME header at the beginning of any Web transmission. Clients use this header to select an appropriate "player" application for the type of data the header indicates. Some of these players are built into the Web client or browser (for example, all browsers come with GIF and JPEG image players as well as the ability to handle HTML files); other players may need to be downloaded. New MIME data types are registered with the Internet Assigned Numbers Authority (IANA). MIME is specified in detail in Internet Request for Comments 1521 and 1522, which amend the original mail protocol specification, RFC 821 (the Simple Mail Transport Protocol) and the ASCII messaging header, RFC 822.

Hypertext Transport Protocol: HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed. HTTP is called a stateless protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason that it is difficult to implement Web sites that react intelligently to user input. HTTPS: A similar abbreviation, HTTPS means Hyper Text Transfer Protocol Secure. Basically, it is the secure version of HTTP. Communications between the browser and website are encrypted by Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL).

HTML Basics

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML Document Structure

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPEhtml>
<html>
<head>
  <title>This is document title</title>
</head>
<body>
  All body related tags are represented here
</body>
</html>
```

Save it in an HTML file **Sample.htm/Sample.html** using your favorite text editor like Notepad or Notepad ++ etc. Finally open it using a web browser like Internet Explorer or Google Chrome, or Firefox etc.

HTML Basic Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>/ <Tag Element>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.

HTML Text Formatting Tags

1. Bold Text tag: ...

Anything that appears within ... element, is displayed in bold.

2. Strong Text tag: ...

Anything that appears within ... element is displayed as important text.

3. Italic Tag: <i>...</i>

<i> tag is used to display the content in italic.

4. Emphasized Text tag: ...

Anything that appears within ... element is displayed as emphasized text.

5. Underline tag: <u>...</u>:

The HTML <u> tag is used to underline text.

6. Marked Text tag: <mark>...</mark>

Anything that appears within <mark>...</mark> element, is displayed as marked with yellow ink.

7. Strike Text tag: <strike>...</strike> or ...

Anything that appears within <strike>...</strike> or ... element is displayed with strikethrough, which is a thin line through the text

8. Computer Code tag: <code>...</code>

Any programming code to appear on a Web page should be placed inside <code>...</code> tags. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

9 Short Quotation tag: <q>...</q>

The <q>...</q> element is used when you want to add a double quote within a sentence.

10. Blinktag: <blink> </blink>

The HTML <blink> tag is used to enclose text to make it blink. This tag was supported by Netscape and now this is obsolete.

11. Preformatted text tag: <pre> </pre>

The HTML <pre> tag is used for indicating preformatted text. The code tags surround the code being marked up. Browsers normally render pre-text in a fixed-pitched font, with whitespace intact, and without word wrap.

Examples:

1. Bold Text tag:

Anything that appears within element, is displayed in bold.

Example

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>BoldTextExample</title>
  </head>
  <body>
    The following word uses a <b>bold</b> typeface.
  </body>
</html>
```

This will produce the following result:

The following word uses a **bold** typeface.

2. Strong Text tag: ...

Anything that appears within ... element is displayed as important text.

Example

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>StrongText Example</title>
  </head>
  <body>
    The following word uses a <strong>strong</strong> typeface.
  </body>
</html>
```

This will produce the following result:

The following word uses a **strong** typeface.

3. ItalicTag: <i>...</i>

<i>tag is used to display the content in italic.

Example:

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>HTML iTag</title>
  </head>
  <body>
    The text displayed in<i>Italic</i> fashion.
  </body>
</html>
```

This will produce the following result:

The text displayed in *Italic* fashion.

4. Emphasized Text tag: ...

Anything that appears within ... element is displayed as emphasized text.

Example:

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>EmphasizedTextExample</title>
  </head>
  <body>
    The following word uses a<em>emphasized</em> typeface.
  </body>
</html>
```

This will produce the following result:

The following word uses an *emphasized* typeface.

5. Marked Text tag: <mark>...</mark>

Anything that appears within <mark>...</mark> element, is displayed as marked with yellow ink.

Example

```
<!DOCTYPEhtml>
<html>
  <head>
```

```
<title>MarkedText Example</title>
```

```
</head>
```

```
<body>
```

```
The following word has been <mark>marked</mark> with yellow
```

```
</body>
```

```
</html>
```

This will produce the following result:

The following word has been marked with yellow.

6. Underline tag: <u>...</u>:

The HTML <u> tag is used to underline text.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTMLuTag</title>
```

```
</head>
```

```
<body>
```

```
<u>This is Underlined text</u>
```

```
</body>
```

```
</html>
```

This will produce the following result: This

is Underlined text

7. StrikeText tag: <strike>...</strike> or ...

Anything that appears within <strike>...</strike> or ... element is displayed with strikethrough, which is a thin line through the text.

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>StrikeText Example</title>
```

```
</head>
```

```
<body>
```

```
<p>Thefollowingwordusesa<strike>strikethrough</strike>typeface.</p>
```

```
</body>
```

```
</html>
```

Thiswillproducethefollowing result:

Thefollowing wordusesa~~strikethrough~~typeface.

8. ComputerCode tag:<code>...</code>

Any programming code to appear on a Web page should be placed inside `<code>...</code>`tags. Usually the content of the `<code>` element is presented in a monospaced font, just like the code in most programming books.

Example

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<head>
```

```
<title>ComputerCodeExample</title>
```

```
</head>
```

```
<body>
```

```
<p>Regulartext.<code>Thisiscode.</code>Regulartext.</p>
```

```
</body>
```

```
</html>
```

Thiswillproducethefollowing result:

Regulartext.Thisiscode.Regulartext.

9Short Quotationstag:<q>...</q>

The`<q>...</q>`elementisusedwhenyouwanttoaddadoublequotewithina sentence.

Example

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<head>
```

```
<title>HTMLblockquoteTag</title>
```

```
</head>
```

```
<body>
```

```
<q>Welcometohtml</q>
```

```
</body>
```

```
</html>
```

Thiswillproducethefollowingresult:

“Welcome to html”

10. Blinktag:<blink>..... </blink>

The HTML <blink> tag is used to enclose text to make it blink. This tag was supported by Netscape and now this is obsolete.

Example

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>HTMLblinkTag</title>
  </head>
  <body>
    <blink>ThistextwillblinkinNetscapeVersion5.0</blink>
  </body>
</html>
```

This will produce the following result:

ThistextwillblinkinNetscapeVersion 5.0

11. Preformattedtext tag:<pre></pre>

The HTML <pre> tag is used for indicating preformatted text. The code tag surrounds the code being marked up. Browsers normally render pre text in a fixed-pitched font, with whitespace intact, and without word wrap.

Example

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>HTMLpreTag</title>
  </head>
  <body>
    <pre>
      Thistextis
        inafixed-pitch font,
      and it preserves
        bothspacesandlinebreaks
    </pre>
  </body>
</html>
```

This will produce the following result:

This text is
in a fixed-pitch font,
and it preserves
both spaces and line breaks

HTML <marquee> Tag

Description

The HTML <marquee> tag is used for scrolling a piece of text or image displayed either horizontally across or vertically down your web site page depending on the settings.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee>This is basic example of marquee</marquee>
    <marquee direction="up" behavior="slide">This is basic example of marquee scrolling from bottom
to up with slide fashion</marquee>
  </body>
</html>
```

This will produce the following result:

This is basic example of marquee in scrolling manner.

This is basic example of marquee scrolling from bottom to top with slide fashion

Marquee tag attributes:

Attribute	Value	Description
behavior	scroll slide alternate	Defines the type of scrolling.
bgcolor	rgb(x,x,x) #xxxxxx colorname	<i>Deprecated</i> -Defines the direction of scrolling the content.
direction	up down left right	Defines the direction of scrolling the content.

height	pixels or %	Defines the height of the marquee.
hspace	pixels	Specifies horizontal space around the marquee.
loop	number	Specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
scrollDelay	seconds	Defines how long to delay between each jump.
scrollAmount	number	Defines how far to jump.
width	pixels or %	Defines the width of the marquee.
vspace	pixels	Specifies vertical space around the marquee.

Linking Documents: <a>..... (HyperLinks)

A link is specified using the HTML tag <a>. This tag is called an **anchor tag** and anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach the linked document. Following is the simple syntax to use <a> tag.

Syntax:

```
<a href="DocumentURL" ...attributes-list>LinkText</a>
```

Example

Let's try following example which links <http://www.google.com> at your page:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hyperlink Example</title>
  </head>
  <body>
    <p>Click following link</p>
    <a href="http://www.google.com" target="_self">Goto Google</a>
  </body>
</html>
```

This will produce the following result, where you can click on the link generated to reach the homepage of Google (in this example).

Click following link [Go to Google](#)

The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options:

Option	Description
_blank	Open the linked document in a new window or tab.
_self	Open the linked document in the same frame.
_parent	Open the linked document in the parent frame.
_top	Open the linked document in the full body of the window.
targetframe	Open the linked document in a named <i>targetframe</i> .

Example

Try following example to understand basic difference in few options given for target attribute.

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>HyperlinkExample</title>
  </head>
  <body>
    <p>Click any of the following links</p>
    <a href="sample.html" target="_blank">Opens in New</a>|
    <a href="sample.html" target="_self">Opens in Self</a>|
    <a href="sample.html" target="_parent">Opens in Parent</a>|
    <a href="sample.html" target="_top">Opens in Body</a>
  </body>
</html>
```

This will produce the following result, where you can click on different links to understand the difference between various options given for target attribute.

Click any of the following links

[Opens in New](#) | [Opens in Self](#) | [Opens in Parent](#) | [Opens in Body](#)

InsertImage:....

You can insert any image in your web page by using tag. Following is the simple syntax to use this tag.

```
<imgsrc="ImageURL"...attributes-list/>
```

The tag is an emptytag, which means that it can contain only list of attributes and it has no closing tag.

Example

Totryfollowingexample,let'skeepourHTMLfiletest.htmlandimagefiletest.jpginthesame directory:

```
<!DOCTYPEhtml>

<html>

  <head>

    <title>UsingImage inWebpage</title>

  </head>

  <body>

    <p>SimpleImageInsert</p>

    <imgsrc="/html/images/test.jpg"alt="TestImage"/>

  </body>

</html>
```

Thiswillproducethefollowing result:



You can use PNG, JPG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Width and Height</title>
</head>
<body>
<p>Setting image width and height</p>

</body>
</html>
```

Image `` attributes:

Attribute	Value	Description
align	top bottom middle left right	<i>Deprecated</i> -Specifies the alignment for the image.
alt	text	Specifies alternate text
border	pixels	<i>Deprecated</i> -Specifies the width of the image border.
height	pixels or %	Specifies the height of the image.

hspace	pixels	<i>Deprecated</i> -Amountofwhitespacetobeinsertedto the left and right of the object.
src	URL	theurl of an image
vspace	pixels	<i>Deprecated</i> -Amountofwhitespacetobeinsertedto the top and bottom of the object.
width	pixels or %	Setsthe width of an image in pixels or in %.

HTML Lists

HTML offers web author three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- ****- An unordered list. This will list items using plain bullets.
- ****- An ordered list. This will use different schemes of numbers to list your items.
- **<dl>**- A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **** tag. Each item in the list is marked with a bullet.

Example

```
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

The type Attribute

- You can use **type** attribute for `` tag to specify the type of bullet you like. By default it is a disc. Following are the possible options:

- `<ul type="square">`
- `<ul type="disc">`
- `<ul type="circle">`

Example

Following is an example where we used `<ul type="square">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
  <ul type="square">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ul>
</body>
</html>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

Example

Following is an example where we used `<ul type="disc">`:

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLUnordered List</title>
</head>
<body>
  <ul type="disc">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ul>
</body>
</html>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

Example

Following is an example where we used `<ul type="circle">`:

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLUnordered List</title>
</head>
<body>
  <ul type="circle">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
```

```
<li>Radish</li>
</ul>
</body>
</html>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted then HTML ordered list will be used. This list is created by using `` tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with ``.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

The type Attribute

You can use the `type` attribute for the `` tag to specify the type of numbering you like. By default it is a number. Following are the possible options:

```
<ol type="1">-Default-Case Numerals.
```

```
<ol type="I">-Upper-Case Numerals.
```

```
<ol type="i">-Lower-Case Numerals.
```

```
<ol type="a">-Lower-Case Letters.
```

```
<ol type="A">-Upper-Case Letters.
```

Example

Following is an example where we used `<ol type="1">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
  <ol type="1">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

This will produce the following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

Example

Following is an example where we used `<ol type="I">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
  <ol type="I">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

This will produce following result:

- I. Beetroot
- II. Ginger
- III. Potato
- IV. Radish

Example

Following is an example where we used `<ol type="i">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
```

```
</head>
<body>
  <ol type="i">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

This will produce following result:

- i. Beetroot
- ii. Ginger
- iii. Potato
- iv. Radish

Example

Following is an example where we used `<ol type="A">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
  <ol type="A">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

This will produce following result:

- A. Beetroot
- B. Ginger
- C. Potato
- D. Radish

Example

Following is an example where we used `<ol type="a">`

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLOrderedList</title>
</head>
<body>
  <ol type="a">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

This will produce following result:

- a. Beetroot
- b. Ginger
- c. Potato
- d. Radish

The start Attribute

You can use **start** attribute for `` tag to specify the starting point of numbering you need. Following are the possible options:

```
<ol type="1" start="4"> -Numerals start with 4.
<ol type="I" start="4"> -Numerals start with IV.
<ol type="i" start="4"> -Numerals start with iv.
```

`<oltype="a"start="4">` -Lettersstartswith d.

`<oltype="A"start="4">` -Lettersstartswith D.

Example

Followingisanexamplewhereweused`<oltype="i"start="4">`

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLOrderedList</title>
</head>
<body>
  <oltype="i"start="4">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

Thiswillproducefollowingresult:

- iv. Beetroot
- v. Ginger
- vi. Potato
- vii. Radish

HTMLDefinitionLists

HTML and XHTML support a list style which is called **definition lists**where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

DefinitionListmakesuseoffollowingthreetags.

- `<dl>` -Definesthe startofthelist
- `<dt>`-Aterm
- `<dd>`-Termdefinition

- `</dl>`-Defines the end of the list

Example

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLDefinitionList</title>
</head>
<body>
<dl>
<dt><b>HTML</b></dt>
<dd>ThisstandsforHyperTextMarkup Language</dd>
<dt><b>HTTP</b></dt>
<dd>ThisstandsforHyperText TransferProtocol</dd>
</dl>
</body>
</html>
```

This will produce the following result:

HTML

This stands for HyperText Markup Language

HTTP

This stands for HyperText Transfer Protocol

HTML Tables:

TheHTMLtablesallowweb authorsto arrangedataliketext, images, links, other tables, etc. into rowsand columns of cells.

TheHTMLtablesarecreatedusingthe **<table>**taginwhichthe **<tr>**tagisusedtocreatetablerows and **<td>**tag is used to create data cells.

Example

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLTables</title>
</head>
<body>
<tableborder="1">
<tr>
<td>Row1,Column1</td>
<td>Row1,Column2</td>
</tr>
<tr>
<td>Row2,Column1</td>
<td>Row2,Column2</td>
</tr>
</table>
</body>
</html>
```

Thiswillproducefollowingresult:

Row1,Column1	Row1,Column2
Row2,Column1	Row2,Column2

Here**border**isanattributeof<table>tagandit isusedto putaborderacrossallthecells. Ifyoudonot need a border then you can use border="0".

TableHeading

Table heading can be defined using **<th>**tag. This tag will be put to replace **<td>** tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use **<th>**element in any row.

Example

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLTableHeader</title>
</head>
<body>
<tableborder="1">
<tr>
<th>Name</th>
<th>Salary</th>
```

```

</tr>
<tr>
<td>RameshRaman</td>
<td>5000</td>
</tr>
<tr>
<td>ShabbirHussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>

```

This will produce following result:

Name	Salary
RameshRaman	5000
ShabbirHussein	7000

Cellpadding and Cellspacing Attributes

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The *cellspacing* attribute defines the width of the border, while *cellpadding* represents the distance between cell borders and the content within a cell.

Example

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>RameshRaman</td>
<td>5000</td>
</tr>
<tr>
<td>ShabbirHussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>

```

This will produce following result:

Name	Salary
RameshRaman	5000
ShabbirHussein	7000

Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

Example

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLTableColspan/Rowspan</title>
</head>
<body>
<tableborder="1">
<tr>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
</tr>
<tr>
<tdrowspan="2">Row1 Cell1</td>
<td>Row1 Cell2</td>
<td>Row1 Cell3</td>
</tr>
<tr>
<td>Row2 Cell2</td>
<td>Row2 Cell3</td></tr>
<tr>
<tdcolspan="3">Row3Cell1</td>
</tr>
</table>
</body>
</html>
```

This will produce following result:

Column1	Column2	Column3
Row 1Cell1	Row 1Cell2	Row 1Cell3

	Row 2Cell2	Row 2Cell3
Row 3Cell1		

TableHeader,Body,andFooter

Tables can be divided into three portions: a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are:

- **<thead>**-to create a separate table header.
- **<tbody>**-to indicate the main body of the table.
- **<tfoot>**-to create a separate table footer.

A table may contain several **<tbody>** elements to indicate different pages or groups of data. But it is notable that **<thead>** and **<tfoot>** tags should appear before **<tbody>**

Example

```

<!DOCTYPEhtml>
<html>
<head>
<title>HTMLTable</title>
</head>
<body>
<tableborder="1"width="100%">
<thead>
<tr>
<tdcolspan="4">This istheheadofthetable</td>
</tr>
</thead>
<tbody>
<tr>
<td>Cell1</td>
<td>Cell2</td>
<td>Cell3</td>
<td>Cell4</td>
</tr>
</tbody>
<tfoot>
<tr>
<tdcolspan="4">Thisisthefootofthetable</td>
</tr>
</tfoot>
</table>
</body>
</html>

```

This will produce the following result:

This is the head of the table			
Cell1	Cell2	Cell3	Cell4
This is the foot of the table			

HTMLForms

HTML Forms are required when you want to collect some data from the site visitor. For example during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML `<form>` tag is used to create an HTML form and it has the following syntax:

```
<form action="ScriptURL" method="GET|POST">form
    elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like <code>_blank</code> , <code>_self</code> , <code>_parent</code> etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls

- FileSelectboxes
- HiddenControls
- ClickableButtons
- Submit andResetButton

TextInputControls

Therearethreetypesoftextinputusedonforms:

- **Single-linetextinputcontrols-** Thiscontrolisusedforitemsthatrequireonlyonelineofuser input, such as search boxes or names. They are created using HTML `<input>`tag.
- **Password input controls** -This is alsoa single-line textinputbutit masks the character as soon as a user enters it. They are also created using HTML `<input>` tag.
- **Multi-line text input controls** - Thisis usedwhen the useris requiredtogive details thatmay be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>`tag.

Single-linetextinputcontrols

This control is used for items that require only one line ofuser input, such as search boxes or names. They are created using HTML `<input>` tag.

Example

Hereisabasicexampleofasingle-linetextinputusedtotakefirstnameandlastname:

```
<!DOCTYPEhtml>
<html>
<head>
<title>TextInputControl</title>
</head>
<body>
<form>
Firstname:<inputtype="text"name="first_name"/>
<br>
Lastname:<inputtype="text"name="last_name"/>
</form>
</body>
</html>
```

Thiswillproducefollowingresult:

Firstname:

Last name:

Attributes

Followingisthelistofattributesfor<input>tagforcreatingtextfield.

Attribute	Description
type	Indicates the type of input control and for text input control it will be set to text .
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPEhtml>
<html>
<head>
<title>PasswordInputControl</title>
</head>
<body>
<form>
User ID: <input type="text" name="user_id" />
<br>
Password: <input type="password" name="password" />
</form>
</body>
</html>
```

This will produce following result:

User ID :

Password:

Attributes

Following is the list of attributes for `<input>` tag for creating password field.

Attribute	Description
Type	Indicatesthetypeofinput controlandforpasswordinputcontrolit will be set to password .
Name	Usedtogive anametothecontrolwhichissenttotheservertobe recognized and get the value.
Value	Thiscanbeusedtoprovideaninitialvalueinsidethecontrol.
Size	Allowstospecifythewidthofthe text-inputcontrolintermsof characters.
Maxlength	Allowstospecifythemaximumnumberofcharactersausercanenter into the text box.

Multiple-LineTextInput Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

Example

Here isabasicexampleofamulti-linetextinputusedtotakeitemdescription:

```
<!DOCTYPEhtml>
<html>
<head>
<title>Multiple-LineInputControl</title>
</head>
<body>
<form>
Description:<br/>
<textarearows="5"cols="50"name="description">Enter
description here...
</textarea>
</form>
</body>
</html>
```

Thiswillproducefollowingresult:

Description :

Attributes

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box.

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox**.

Example

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPEhtml>
<html>
<head>
<title>CheckboxControl</title>
</head>
<body>
<form>
<inputtype="checkbox"name="maths"value="on">Maths
<inputtype="checkbox"name="physics"value="on">Physics
</form>
</body>
</html>
```

This will produce the following result:

Maths Physics

Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	Description
type	Indicatesthe type of input control and for checkbox input control it will be set to checkbox .
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to <i>checked</i> if you want to select it by default.

RadioButtonControl

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPEhtml>
<html>
<head>
<title>RadioBoxControl</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths">Maths
<input type="radio" name="subject" value="physics">Physics
</form>
</body>
</html>
```

This will produce following result:

Maths Physics

Attributes

Following is the list of attributes for radio button.

Attribute	Description
Type	Indicatesthetypeofinput control andforcheckboxinputcontrolit will be set to radio .
Name	Usedtogive anametothecontrolwhichissenttotheservertobe recognized and get the value.
Value	Thevaluethatwill beusediftheradioboxisselected.
Checked	Setto <i>checked</i> ifyouwanttoselect itbydefault.

SelectBoxControl

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

HereisexampleHTML codeforaformwithone drop downbox

```
<!DOCTYPEhtml>
<html>
<head>
<title>SelectBoxControl</title>
</head>
<body>
<form>
<selectname="dropdown">
<optionvalue="Maths"selected>Maths</option>
<optionvalue="Physics">Physics</option>
</select>
</form>
</body>
</html>
```

Thiswillproducefollowing result:

Attributes

Followingisthelist ofimportant attributesof<select> tag:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
size	This can be used to present as a scrolling list box.
multiple	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag:

Attribute	Description
value	The value that will be used if an option in the select box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options

FileUploadBox

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but the type attribute is set to **file**.

Example

Here is an example HTML code for a form with one file upload box:

```
<!DOCTYPEhtml>
<html>
<head>
<title>FileUploadBox</title>
</head>
<body>
<form>
```

```

<input type="file" name="fileupload" accept="image/*"/>
</form>
</body>
</html>

```

This will produce following result:

Attributes

Following is the list of important attributes of fileupload box:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
accept	Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input> tag by setting its type attribute to **button**. The type attribute can take the following values:

Type	Description
submit	This creates a button that automatically submits a form.
reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons:

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>FileUploadBox</title>
</head>
<body>
<form>
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
</body>
</html>

```

This will produce following result:



HiddenFormControls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page has to be displayed next based on the passed current page.

Example

Here is an example HTML code to show the usage of hidden control:

```

<!DOCTYPE html>
<html>
<head>
<title>FileUploadBox</title>
</head>
<body>
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
</body>

```

```
</html>
```

This will produce following result:

This is page 10

Example Student Registration Form in table format

```
<html>
<head>
  <title>Forms</title>
</head>
<body>
  <form>
<table border="1" align="center">
  <tr>
    <th colspan="2"><h3>Student Registration Form</h3></th>
  </tr>
  <tr>
    <td>Enter Your Name:</td>
    <td><input type="text"></td>
  </tr>
  <tr>
    <td>Enter your password:</td>
    <td><input type="password"></td>
  </tr>
  <tr>
    <td>Gender:</td>
    <td><input type="radio" name="gender" value="Male">
      <input type="radio" name="gender" value="Female"></td>
  </tr>
  <tr>
    <td>Languages Known:</td>
    <td><input type="checkbox" name="Languages" value="Telugu">Telugu
      <input type="checkbox" name="languages" value="English">English
      <input type="checkbox" name="hobbies" value="Hindi">Hindi</td>
  </tr>
  <tr>
    <td>Choose your country</td>
    <td><select>
      <option>India</option>
      <option>Australia</option>
      <option>England</option>
    </select>
  </td>
  </tr>
  <tr>
    <td>Give your feedback</td>
    <td><textarea rows="4">please enter your feedback</textarea>
  </td>
  </tr>
</tr>
```

```

<td>EnteryourDOB:</td>
<td><inputtype="date"name="dob"value="DOB"></td>
</tr>
<tr>
<td>Chooseurbest color:</td>
<td><inputtype="color"name="color"value="color"></td>
</tr>
<tr>
<td>Chooseurfile:</td>
<td><inputtype="file"name="file"></td>
</tr>
<tr>
<td></td>
<td><inputtype="submit"value="Submit">
<inputtype="Reset"value="Reset"></td>
</tr>
</table>
</form>
</body>
</html>

```

Output:

Student Registration Form	
Enter Your Name:	<input type="text"/>
Enter your password:	<input type="password"/>
Gender:	<input type="radio"/> <input type="radio"/>
Languages Known:	<input type="checkbox"/> Telugu <input type="checkbox"/> English <input type="checkbox"/> Hindi
Choose ur country	India ▼
Give your feedback	<input type="text" value="please enter your feedback"/>
Enter your DOB:	mm/dd/yyyy
Choose ur best color:	<input type="color" value="black"/>
Choose ur file:	<input type="button" value="Choose File"/> No file chosen
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>

HTMLFrames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your Web Pages:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

To use frames on a page we use `<frameset>` tag instead of `<body>` tag. The `<frameset>` tag defines how to divide the window into frames. The **rows** attribute of `<frameset>` tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by `<frame>` tag and it defines which HTML document shall open into the frame.

Following is the example to create three horizontal frames:

```
<!DOCTYPEhtml>
<html>
<head>
<title>HTMLFrames</title>
</head>
<framesetrows="10%,80%,10%">
  <frame name="top" src="top.html"/>
  <frame name="main" src="middle.html"/>
  <frame name="bottom" src="bottom.html"/>
  <noframes>
  <body>
    Your browser does not support frames.
  </body>
</noframes>
</frameset>
</html>
```

This will produce following result:



Example

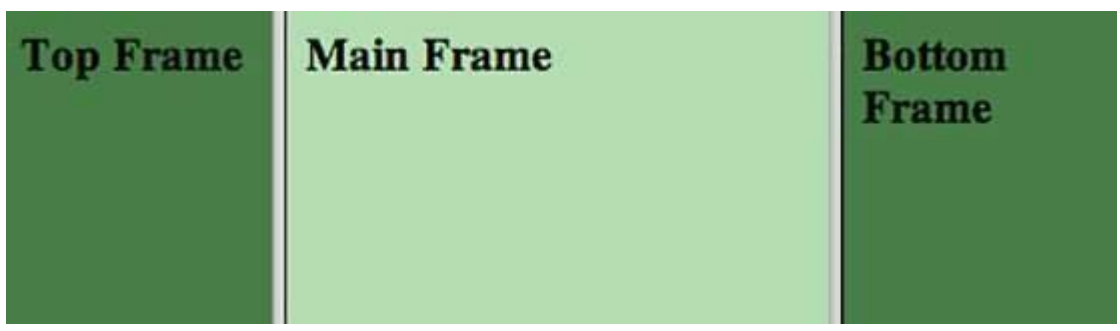
Let's put above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically:

```

<!DOCTYPEhtml>
<html>
<head>
<title>HTMLFrames</title>
</head>
<framesetcols="25%,50%,25%">
  <frame="left"src="left.html"/>
  <frame="center"src="middle.html"/>
  <frame="right"src="right.html"/>
  <noframes>
  <body>
    Yourbrowserdoesnotsupportframes.
  </body>
  </noframes>
</frameset>
</html>

```

Thiswillproducefollowingresult:



The<frameset>TagAttributes

Followingareimportantattributesofthe<frameset>tag:

Attribute	Description
-----------	-------------

Cols	<p>specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of four ways:</p> <ul style="list-style-type: none"> • Absolute values in pixels. For example to create three vertical frames, use <code>cols="100, 500, 100"</code>. • A percentage of the browser window. For example to create three vertical frames, use <code>cols="10%, 80%, 10%"</code>. • Using a wildcard symbol. For example to create three vertical frames, use <code>cols="10%, *, 10%"</code>. In this case wildcard takes remainder of the window. • As relative widths of the browser window. For example to create three vertical frames, use <code>cols="3*, 2*, 1*"</code>. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
Rows	<p>This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use <code>rows="10%, 90%"</code>. You can specify the height of each row in the same way as explained above for columns.</p>
border	<p>This attribute specifies the width of the border of each frame in pixels. For example <code>border="5"</code>. A value of zero means no border.</p>
frameborder	<p>This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example <code>frameborder="0"</code> specifies no border.</p>
framespacing	<p>This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example <code>framespacing="10"</code> means there should be 10 pixels spacing between each frames.</p>

The <frame> Tag Attributes

Following are important attributes of <frame> tag:

Attribute	Description
-----------	-------------

Src	This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, <code>src="/html/top_frame.htm"</code> will load an HTML file available in html directory.
Name	This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into another frame, in which case the second frame needs a name to identify itself as the target of the link.
frameborder	This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the <code>frameborder</code> attribute on the <code><frameset></code> tag if one is given, and this can take values either 1 (yes) or 0 (no).
marginwidth	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example <code>marginwidth="10"</code> .
marginheight	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example <code>marginheight="10"</code> .
noresize	By default you can resize any frame by clicking and dragging on the borders of a frame. The <code>noresize</code> attribute prevents a user from being able to resize the frame. For example <code>noresize="noresize"</code> .
scrolling	This attribute controls the appearance of the scrollbar that appears on the frame. This takes values either "yes", "no" or "auto". For example <code>scrolling="no"</code> means it should not have scroll bars.

Browser Support for Frames

If a user is using any old browser or any browser which does not support frames then `<noframes>` element should be displayed to the user.

So you must place a `<body>` element inside the `<noframes>` element because the `<frameset>` element is supposed to replace the `<body>` element, but if a browser does not understand `<frameset>` element then it should understand what is inside the `<body>` element which is contained in a `<noframes>` element.

You can put some nice message for your user having old browsers. For example *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where `test.html` file has following code:

```
<html>
<head>
<title>HTMLTargetFrames</title>
</head>
<framesetcols="20%,*">
  <framesrc="menu.html" name="menu_page"/>
  <framesrc="main.html" name="main_page"/>
</frames>
<body>
  Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

Here we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by `menu.html` file. The second column fills in remaining space and will contain the main part of the page and it is implemented by `main.html` file. For all the three links available in menu bar, we have mentioned target frame as `main_page`, so whenever you click any of the links in menu bar, available link will open in `main_page`.

Following is the content of `menu.html` file

```
<!DOCTYPEhtml>
<html>
<body bgcolor="#4a7d49">
<a href="https://www.google.com" target="main_page">Google</a>
<br /><br />
<a href="https://www.microsoft.com" target="main_page">Microsoft</a>
<br /><br />
<a href="https://news.bbc.co.uk" target="main_page">BBCNews</a>
</body>
</html>
```

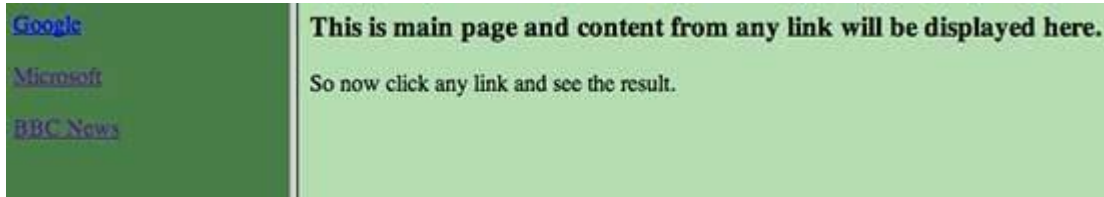
Following is the content of `main.html` file:

```

<!DOCTYPEhtml>
<html>
<body bgcolor="#b5dcb3">
<h3>This is main page and content from any link will be displayed here.</h3>
<p>So now click any link and see the result.</p>
</body>
</html>

```

When we load **test.html** file, it produces following result:



Now you can try to click links available in the left pane and see the result. The *target* attribute can also take one of the following values:

Option	Description
_self	Load the page into the current frame.
_blank	Load a page into a new browser window, opening a new window.
_parent	Load the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Load the page into the browser window, replacing any current frames.
targetframe	Load the page into a named target frame.