UNIT-1

Lecture-1

Introduction

A Brief Introduction to the Internet

Internet has actually been in existence for quite a long time. Originally used as a means of keeping communication channels open between military bases in various parts of the United States of America, it was at that stage a small network of slow modems connected to mainframe computers. The network was extended by connecting computers from universities around America and it was not long before university staff were using the system to send messages to each other. The idea was used in other countries and it was not long before countries started connecting their networks to one another and. From these beginnings in the mid-1960s the Internet has grown to include a number of methods, or protocols, for communicating via computer.

Today, the Internet is best described as a network of computers spread across the world, making use of fibre optic cables, telephone lines and satellites to communicate with other computers in the network. The Internet makes use of vacant bandwidth in the telecommunications network to send messages from computer to computer, rather than relying on an entirely new infrastructure.

A standardized addressing system identifies specific computers, making it easy for other computers to hold information about what information other computers are storing and where they are. When we make use of the World Wide Web we are using this addressing system to go to a specific computer, either in Melbourne or possibly on the other side of the world, to read files stored on that computer. While any computer is connected to the network it is described as a "node" on the Internet, and with appropriate software we can use even a desktop computer to "serve" files to the rest of the world. It is the simplicity of this networking which has caused it to seize the imagination of users and to grow exponentially.

The Internet, and particularly the World Wide Web, has revolutionized the way we communicate. It is likely that fax machines will go the way of the telegraph and the telex, and while the Internet in ten years will probably look quite different from that which we see now, it is certain to have become even more pervasive.

The most commonly used parts of the Internet today include email, newsgroups, File Transfer Protocol, Internet Relay Chat, and of course the World Wide Web. Other areas which are rapidly growing include Internet telephony and video conferencing.

Components of the Internet

The various components of the internet are:

- World wide web(WWW)
- Web page
- Hypertext Markup Language(HTML)
- Web Browser
- Uniform Resource Locator(URL)
- Hypertext Transfer Protocol(HTTP)
- Transmission control protocol/Internet Protocol

World Wide Web(WWW):is a collection of Internet Servers that provide an easy to use point and click interface. The WWW or the Web (as it is commonly called) is the only multimedia service on the internet.

Web Page:A Web page is a document or single unit of information that belongs to a web site and consists of information on that site.

A Web page can contain text, video, graphics, audio files, and links to various other web pages.

To make a web page available to every one on the internet,you need to host the web page on a site that is placed on a web server.

WebSite: Collection of WebPages is called Web Site

Methods for Hosting Web Pages:

1. You can set up your own web server to host the web page.However setting up a web server is a costly affair.
2. If you have financial constraints,you can acquire services from an internet service provider(ISP) to host your web page.most ISP's offer space on their web server to its clients at a nominal price.
3. you can request a company that provides web hosting services to rent you web space at reasonable price.

Hypertext Markup Language (HTML):

- is a markup language used to create web pages
- HTML is a versatile language and can be used on any platform or desktop
- This language originated from another language which is called standard Generalized Markup Language(SGML)
- Being Platform independent,HTML indicates the manner in which the web page is to be read at a client computer

Web Browser:

- In order to access information you need a web browser.web browsers are client programs that fetch web pages for you
- Web browser act as an interface between the client computer and the web server.

Usage of Web Browser:

- Sending and receiving email messages
- Reading messages from news groups
- Browsing the web

Uniform Resource Locator (URL)

A URL contains the exact location of any document. It is an addressing scheme that provides the path to an Internet resource. When a user clicks a link, the URL provides information about that link to the Web browser, which in turn displays the linked Web page. Therefore, links are always implemented by using URLs. A URL may point to a document, image, video, or graphic.

A typical URL would be of the following format:

protocol://host.domain−name.toplevel−domain−name/path/dataname

Example:

http://cmrcet.org

http://cmrcet.org/aboutdepartment%28CSE%29.php

where:

protocol refers to the type of protocol to be used.

host refers to the server where the resource is stored.

domain−name and **toplevel−domain−name** are the name and the type of the domain, respectively.

The types of domains include *com* (used for commercial institutions), *edu*(educational institutions), *net* (network organizations), *org* (miscellaneous organizations),*gov* (government entities), and *mil* (U.S. military).

path/dataname refers to the location on the server where the data is stored.

Hypertext Transfer Protocol (HTTP):

- HTTP adopted the concept of hypertext but its protocol includes other methods

There are four messages with in this protocol

Connection: Establishes a connection between the client and the server

Request: Ask for a Resource

Response: Delivers the Resource

Close: Terminates the Connection

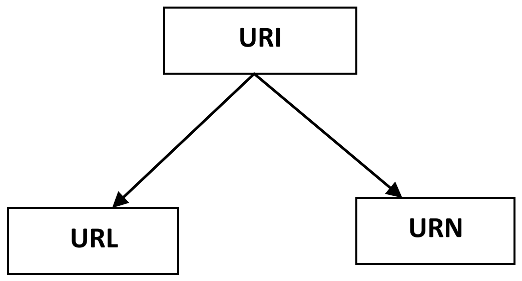
Lecture-2

Client Server Communication

- All web activity begins on the client side you could type a web address into the browser
- The browser first consult with the DNS to translate the home page name into an IP Address
- It then sends a request to the server using the HTTP Standard
- A server spends most of its time listening to the network waiting for a document request

URI (Uniform Resource Identifier):

URI is a string of characters in a particular Syntax that identifies a resource



URL: Uniform Resource Locator

URL is a pointer to a particular resource on the internet at a particular location

Syntax:

```
Protocol://username@hostname:port/path/filename
```

Example: http://localhost:8086/cmrsite/Register.html

URN: Uniform Resource Names:

URN is a name for a particular resource but without reference to a particular location.

Syntax:

```
Urn: namespace: resourcename
```

Example:urn:isbn:1565928709

UNIT-1

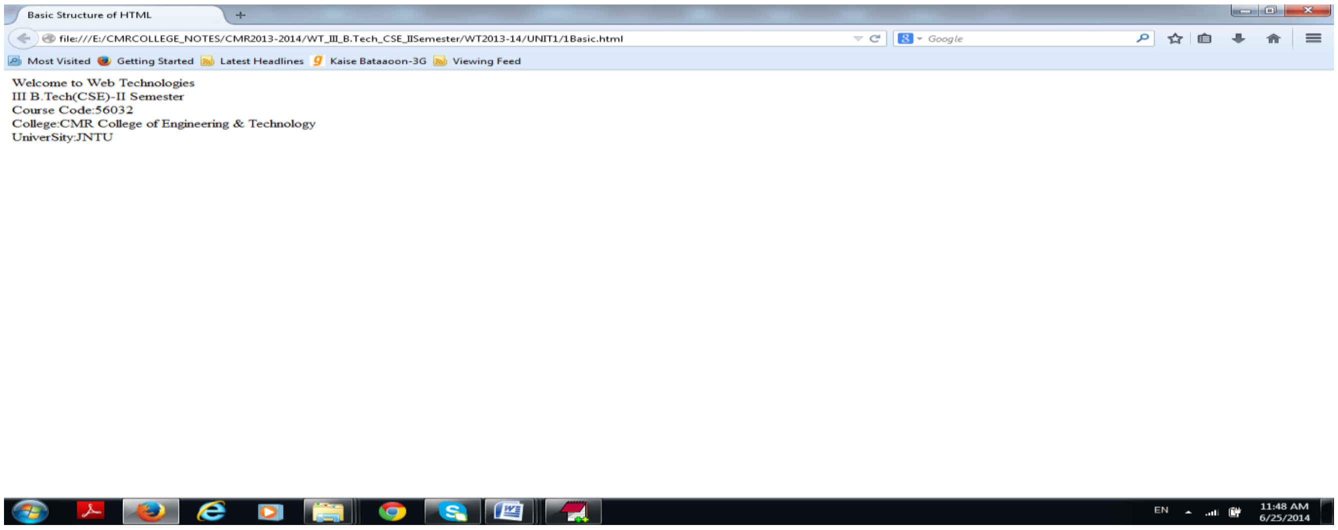
Lecture-3

HTML Common tags

Basic.html

```
<!DOCTYPE html>
<html>
<head>
	<title>Basic Structure of HTML</title>
</head>
<body>
Welcome to Web Technologies<br>
III B.Tech(CSE)-II Semester<br>
Course Code:56032<br>
College:CMR College of Engineering & Technology<br>
UniverSity:JNTU
</body>
</html>
```

Output:



```
Welcome to Web Technologies
III B.Tech(CSE)-II Semester
Course Code:56032
College:CMR College of Engineering & Technology
UniverSity:JNTU
```

- The DOCTYPE declaration defines the document type
- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between <p> and </p> is displayed as a paragraph

What is HTML?

HTML is a language for describing web pages.

- HTML stands for Hyper Text Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describes document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

HTML Tags

HTML markup tags are usually called HTML tags

HTML tags are keywords (tag names) surrounded by angle brackets like <html>

HTML tags normally come in pairs like <b> and </b>

The first tag in a pair is the start tag, the second tag is the end tag

The end tag is written like the start tag, with a forward slash before the tag name

Start and end tags are also called opening tags and closing tags

syntax:

<tagname>content</tagname>

HTML Elements

"HTML tags" and "HTML elements" are often used to describe the same thing.

But strictly speaking, an HTML element is everything between the start tag and the end tag, including the tags:

HTML Element:

<p>This is a paragraph.</p>

Web Browsers

The purpose of a web browser (such as Google Chrome, Internet Explorer, Firefox, Safari) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page

HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>

<body>

<h1>This a Heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>
```

HTML Versions

Since the early days of the web, there have been many versions of HTML:

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2012
XHTML5	2013

The <! DOCTYPE> Declaration

The <!DOCTYPE> declaration helps the browser to display a web page correctly.

There are many different documents on the web, and a browser can only display an HTML page 100% correctly if it knows the HTML type and version used.

Common Declarations

HTML5

<! DOCTYPE html>

HTML 4.01

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

XHTML 1.0

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

Text:

1. Displaying Header
2. HR Tag
3. Setting Font Style
4. Text Alignment
5. Setting the Font

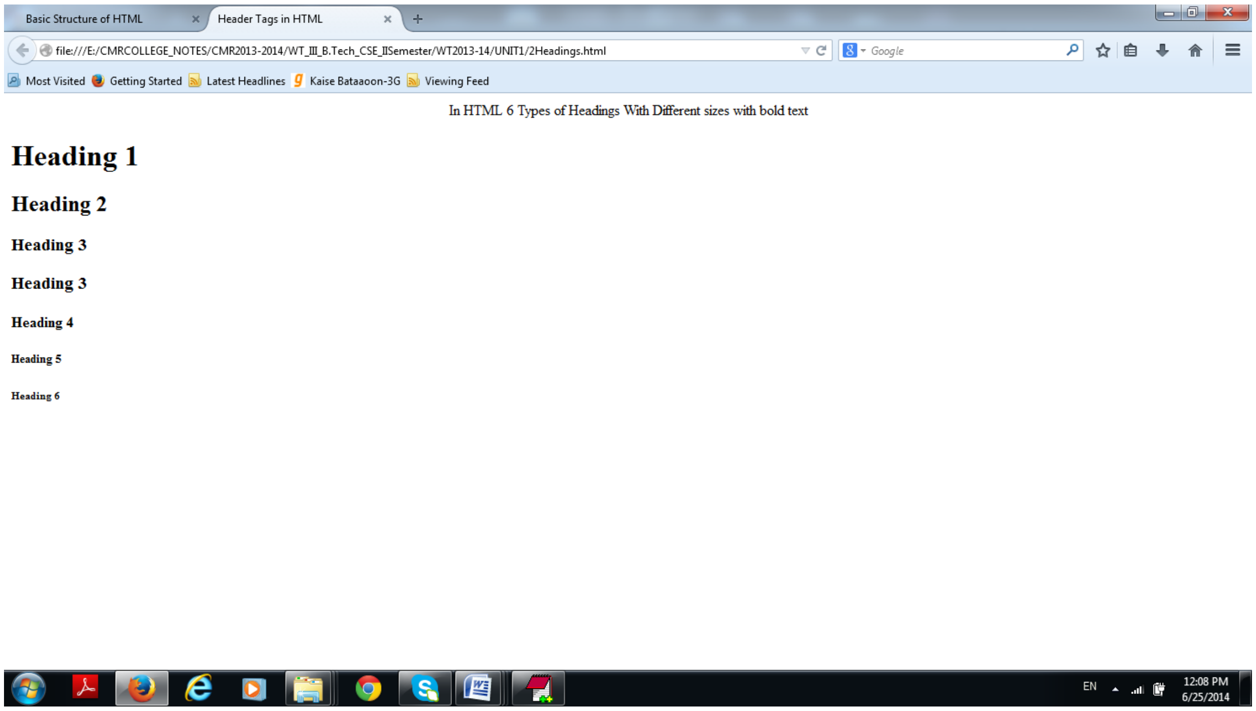
Displaying Header

Headings.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Header Tags in HTML </TITLE>
</HEAD>
<BODY>
<CENTER>In HTML 6 Types of Headings With Different sizes with bold text</CENTER>
<H1>Heading 1</H1>
<!-- <H1 [align="left"|"center"|"right"]>...</H1> -->

<H2>Heading 2</H2>
```

```
<H3>Heading 3</H3>
<H3>Heading 3</H3>
<H4>Heading 4</H4>
<H5>Heading 5</H5>
<H6>Heading 6</H6>
<!-- This is comment Tag in HTML-->
</BODY>
</HTML>
```

Output:



In HTML 6 Types of Headings With Different sizes with bold text

Heading 1

Heading 2

Heading 3

Heading 3

Heading 4

Heading 5

Heading 6

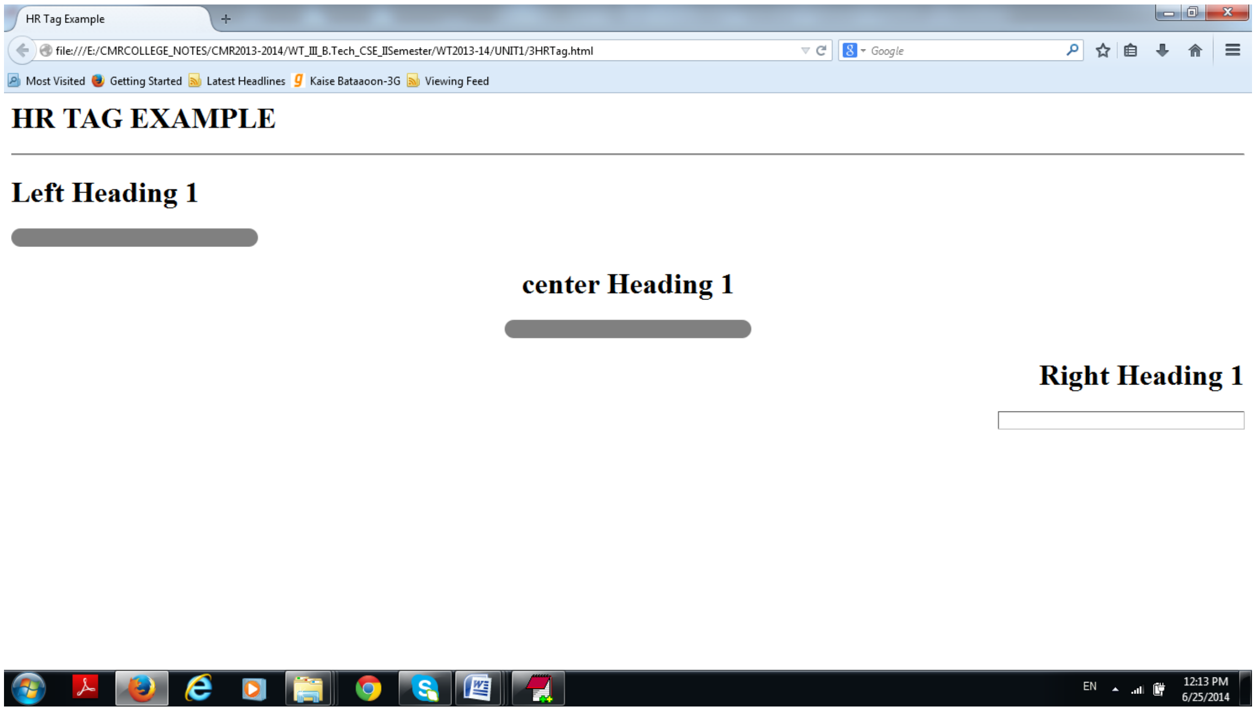
HR Tag:

HRTag.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>HR Tag Example </TITLE>
</HEAD>
<BODY>
<!-- Horizontal Rule <hr> Tag Example -->
<!-- <hr [align="left"|"center"|"right"][size="n"][noshade][width="nn%"] -->
<h1>HR TAG EXAMPLE</h1>
<hr>
```

```
<H1 align="left">Left Heading 1</H1>
<hr align="left" size="20" noshade width="20%">
<H1 align="center">center Heading 1</H1>
<hr align="center" size="20" noshade width="20%">
<H1 align="right">Right Heading 1</H1>
<hr align="right" size="20"  width="20%">
</BODY>
</HTML>
```

OUTPUT:



HR TAG EXAMPLE

Left Heading 1

center Heading 1

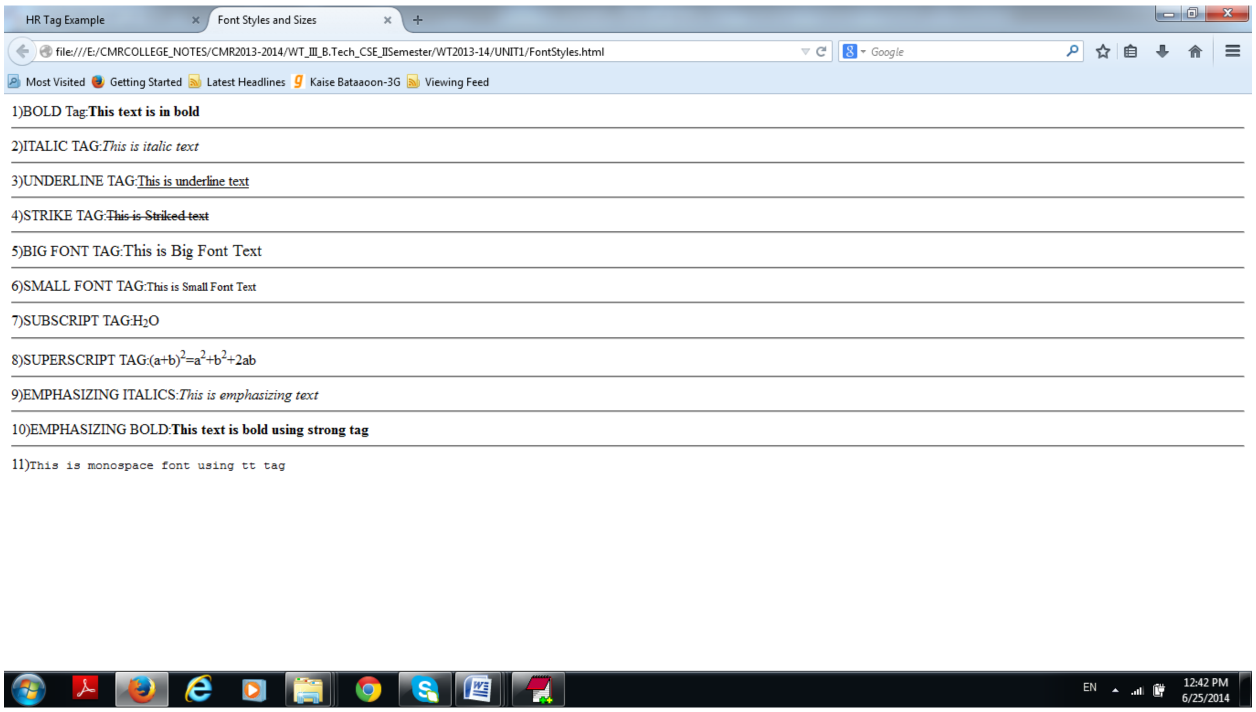
Right Heading 1

Setting Font Style:

FontStyles.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Font Styles and Sizes</TITLE>
</HEAD>
<BODY>
1)BOLD Tag:<b>This text is in bold</b>
<hr>
2)ITALIC TAG:<i>This is italic text</i>
<hr>
3)UNDERLINE TAG:<u>This is underline text</u>
<hr>
```

```
4)STRIKE TAG:<strike>This is Striked text</strike>
<hr>
5)BIG FONT TAG:<big>This is Big Font Text</big>
<hr>
6)SMALL FONT TAG:<small>This is Small Font Text</small>
<hr>
7)SUBSCRIPT TAG:H<sub>2</sub>O
<hr>
8)SUPERSCRIPT TAG:(a+b)<sup>2</sup>=a<sup>2</sup>+b<sup>2</sup>+2ab
<hr>
9)EMPHASIZING ITALICS:<em>This is emphasizing text</em>
<hr>
10)EMPHASIZING BOLD:<strong>This text is bold using strong tag</strong>
<hr>
11)<tt>This is monospace font using tt tag</tt>
</BODY>
</HTML>
```
OUTPUT:



1)BOLD Tag:**This text is in bold**

2)ITALIC TAG:*This is italic text*

3)UNDERLINE TAG:This is underline text

4)STRIKE TAG:~~This is Striked text~~

5)BIG FONT TAG:This is Big Font Text

6)SMALL FONT TAG:This is Small Font Text

7)SUBSCRIPT TAG:H_2O

8)SUPERSCRIPT TAG:$(a+b)^2=a^2+b^2+2ab$

9)EMPHASIZING ITALICS:*This is emphasizing text*

10)EMPHASIZING BOLD:**This text is bold using strong tag**

11)`This is monospace font using tt tag`

HTML Hyperlinks (Links)

The HTML <a> tag defines a hyperlink.

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to another document.

When you move the cursor over a link in a Web page, the arrow will turn into a little hand.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

HTML Link Syntax

The HTML code for a link is simple. It looks like this:

<a href="*url*">*Link text*</a>

Attribute for <A ...>

TARGET = "_blank" | "_parent" | "_self" | "_top"

- `"_blank"` opens the new document in a new window.

this code	produces this
`<A HREF="newwindow.html" TARGET="_blank">a new window</A>`	a new window

TARGET = "_parent"

`"_parent"` is used in the situation where a frameset file is nested inside another frameset file. A link in one of the inner frameset documents which uses `"_parent"` will load the new document where the inner frameset file had been.

For example, this anchor:

```
<A HREF="bigframe.html" TARGET="_parent">bigframe</A>
```

TARGET = "_self"

"_self" puts the new document in the same window and frame as the current document. "_self" works the same as if you had not used TARGET at all.

this code	produces this
`go to <A HREF="selftarget.html" TARGET="_self">next</A> page`	The link on this page

TARGET = "_top"

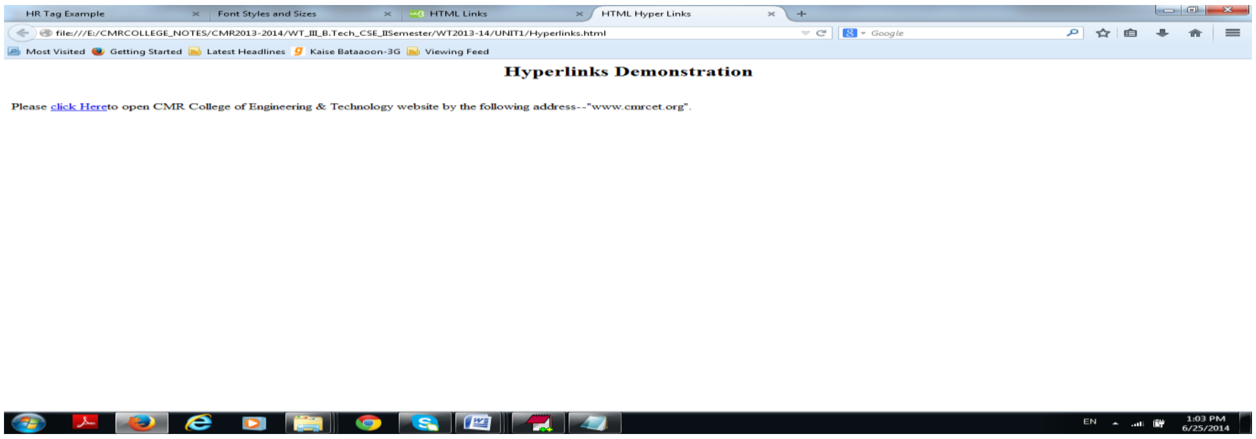
"_top" loads the linked document in the topmost frame... that is, the new page fills the entire window.

this code	produces this
`<A HREF="selftarget.html" TARGET="_top">top</A>`	the link in this page

Example:
Hyperlinks.html

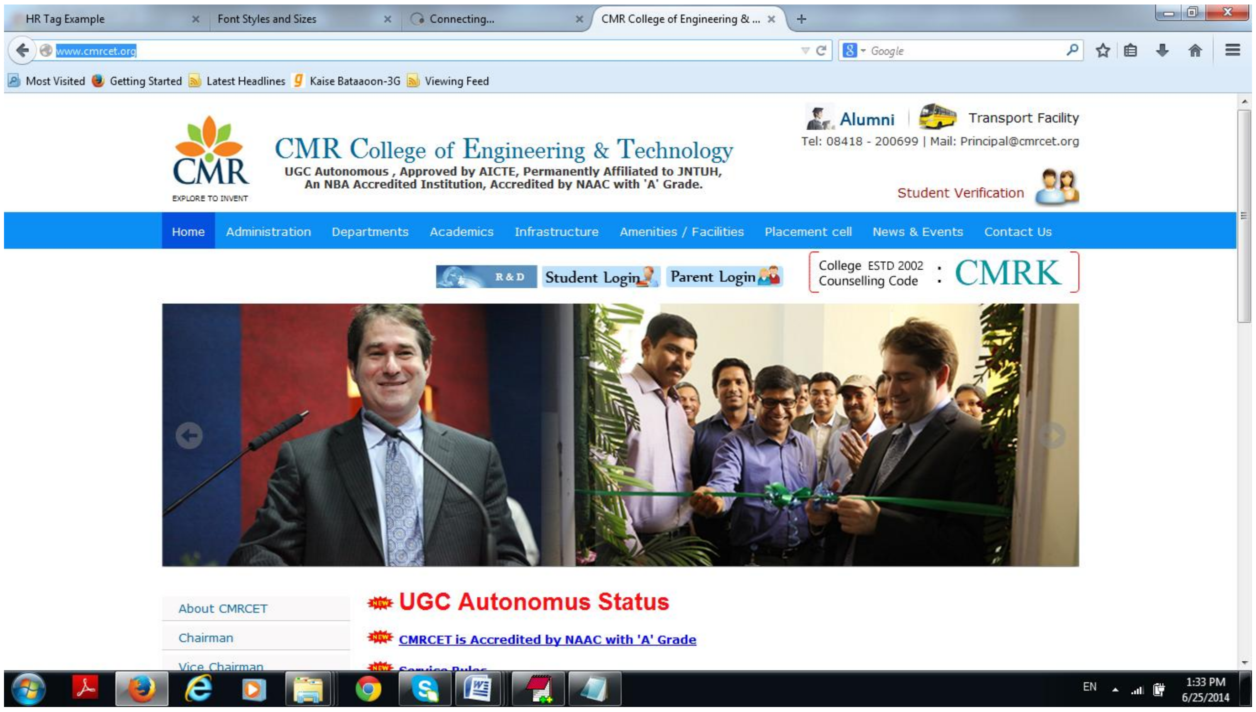
```
<html>
<head><title>HTML Hyper Links</title></head>
<body>
  <h2 align="center">Hyperlinks Demonstration</h2><br>
  Please <a href="http://www.cmrcet.org" title="CMRCET">click Here</a>to open CMR
College of Engineering & Technology website by the following address--"www.cmrcet.org".
</body>
</html>
```

Output:



Hyperlinks Demonstration

Please click Hereto open CMR College of Engineering & Technology website by the following address--"www.cmrcet.org".



HTML Entities

Reserved characters in HTML must be replaced with character entities.

HTML Entities

Some characters are reserved in HTML.

It is not possible to use the less than (<) or greater than (>) signs in your text, because the browser will mix them with tags.

To actually display reserved characters, we must use character entities in the HTML source code.

A character entity looks like this:

&entity_name;

OR

&#entity_number;

Non-breaking Space

A common character entity used in HTML is the non-breaking space (&nbsp;).

Browsers will always truncate spaces in HTML pages. If you write 10 spaces in your text, the browser will remove 9 of them, before displaying the page. To add spaces to your text, you can use the &nbsp; character entity.

HTML Useful Character Entities

Note: Entity names are case sensitive!

Result	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
§	section	&sect;	&#167;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;

HTML Images:

Displaying Images on Webpages

- the basic HTML code to insert images into your WebPages.

Syntax:

```
<img src="path/to/image“ >
Example:
<img src="images/1.jpg" width="1000" height="1000" border="0" alt="csegroupphoto">
```

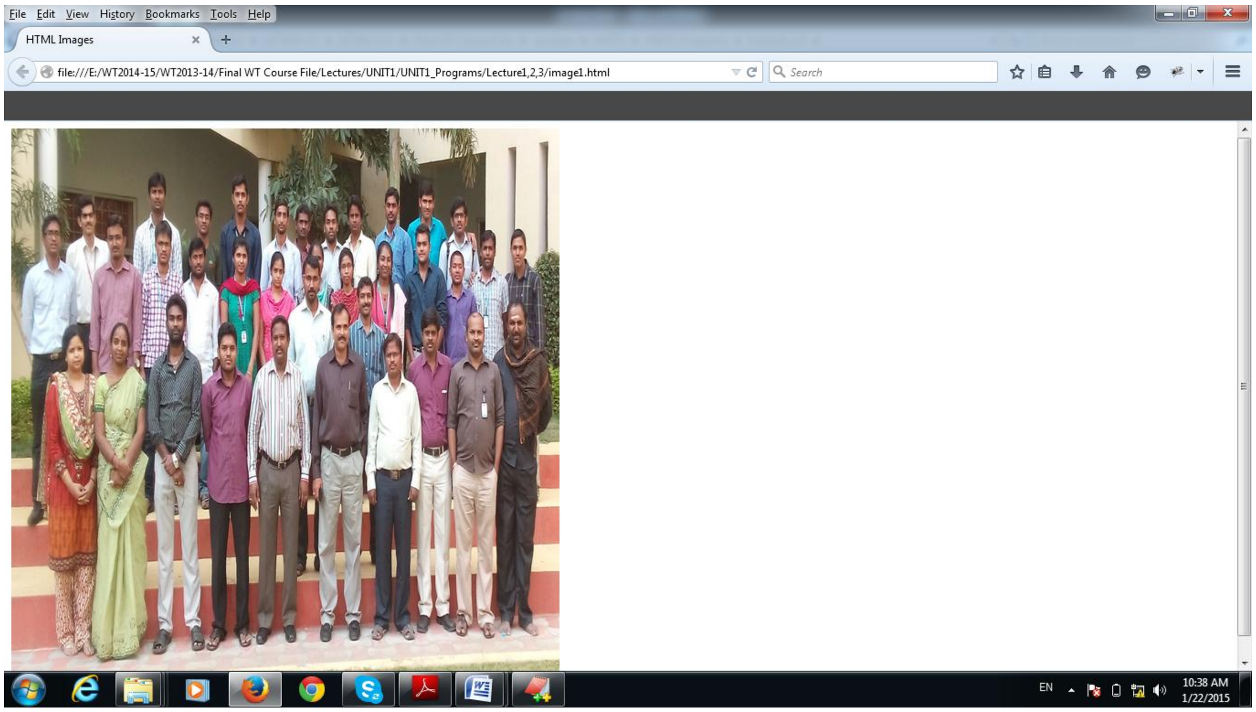
ATTRIBUTES OF <IMG ..>TAG

- SRC: where to get the picture
- ALT: text to show if you don't show the picture
- NAME
- WIDTH: how wide is the picture
- HEIGHT: how tall is the picture
- ALIGN: how text should flow around the picture
- BORDER: border around the picture
- HSPACE: horizontal distance between the picture and the text
- VSPACE: vertical distance between the picture and the text

Image1.html

```
<!doctype html>
<html lang="en">
 <head>

  <title>HTML Images</title>
 </head>
 <body>
 <img src="images/1.jpg" width="1000" height="1000" border="0" alt="csegroupphoto">

 </body>
</html>
```

Output:


File Edit View History Bookmarks Tools Help
HTML Images
file:///E:/WT2014-15/WT2013-14/Final WT Course File/Lectures/UNIT1/UNIT1_Programs/Lecture1,2,3/image1.html
Search
10:38 AM
1/22/2015

UNIT-1

Lecture-4

HTML-Lists

HTML Unordered Lists

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
The list items are marked with bullets (typically small black circles).

HTML Ordered Lists

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

HTML Definition Lists

A definition list is a list of items, with a description of each item.

The <dl> tag defines a definition list.

The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list)

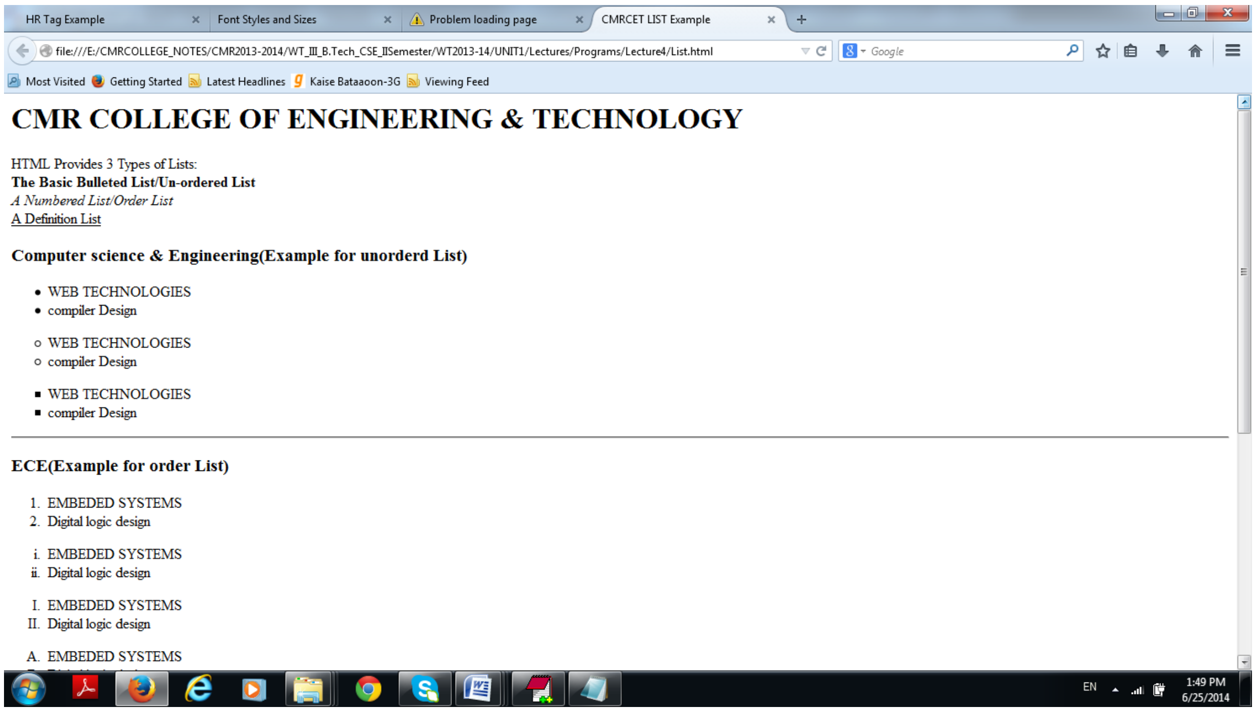
Example:

List.html

```
<html>
<head>
<title>CMRCET LIST Example</title>
</head>
<body>
<h1>
CMR COLLEGE OF ENGINEERING & TECHNOLOGY
</h1>
<p>HTML Provides 3 Types of Lists:<br>
<b>The Basic Bulleted List/Un-ordered List</b><br>
<i>A Numbered List/Order List</i><br>
<u>A Definition List</u><br>
</p>
<h3>Computer science & Engineering(Example for unorderd List)</h3>
<ul type="disc">
<li>WEB TECHNOLOGIES</li>
<li>compiler Design</li>
</ul>
<ul type="circle">
<li>WEB TECHNOLOGIES</li>
<li>compiler Design</li>
</ul>
<ul type="square">
<li>WEB TECHNOLOGIES</li>
<li>compiler Design</li>
</ul>

<hr>
<h3>ECE(Example for order List)</h3>
<ol>
<li>EMBEDED SYSTEMS</li>
<li>Digital logic design</li>
</ol>
<ol type="i">
<li>EMBEDED SYSTEMS</li>
<li>Digital logic design</li>
```

```
</ol>
<ol type="I">
<li>EMBEDED SYSTEMS</li>
<li>Digital logic design</li>
</ol>
<ol type="A">
<li>EMBEDED SYSTEMS</li>
<li>Digital logic design</li>
</ol>
<ol type="a">
<li>EMBEDED SYSTEMS</li>
<li>Digital logic design</li>
</ol>
<hr>
<h3>Definition List</h3>
<dl>
<dt>Widgets</dt>
<dd>Provided in three sizes<i>Small,Medium,large</i>
</dd>
<dt>CMR B.Tech CSE Students</dt>
<dt>TESTING</dt>
</dl>
<hr>
<dl>
<dt>Heading1
<dd>Subheading 1</dd>
</dt>
<dt>Heading2
<dd>Subheading 2</dd>
</dt>
<dt>Heading3
</dt>
<hr>
</dl>
</body>
</html>
```

Output:



HR Tag Example | Font Styles and Sizes | Problem loading page | CMRCET LIST Example

file:///E:/CMRCOLLEGE_NOTES/CMR2013-2014/WT_III_B.Tech_CSE_IISemester/WT2013-14/UNIT1/Lectures/Programs/Lecture4/List.html

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

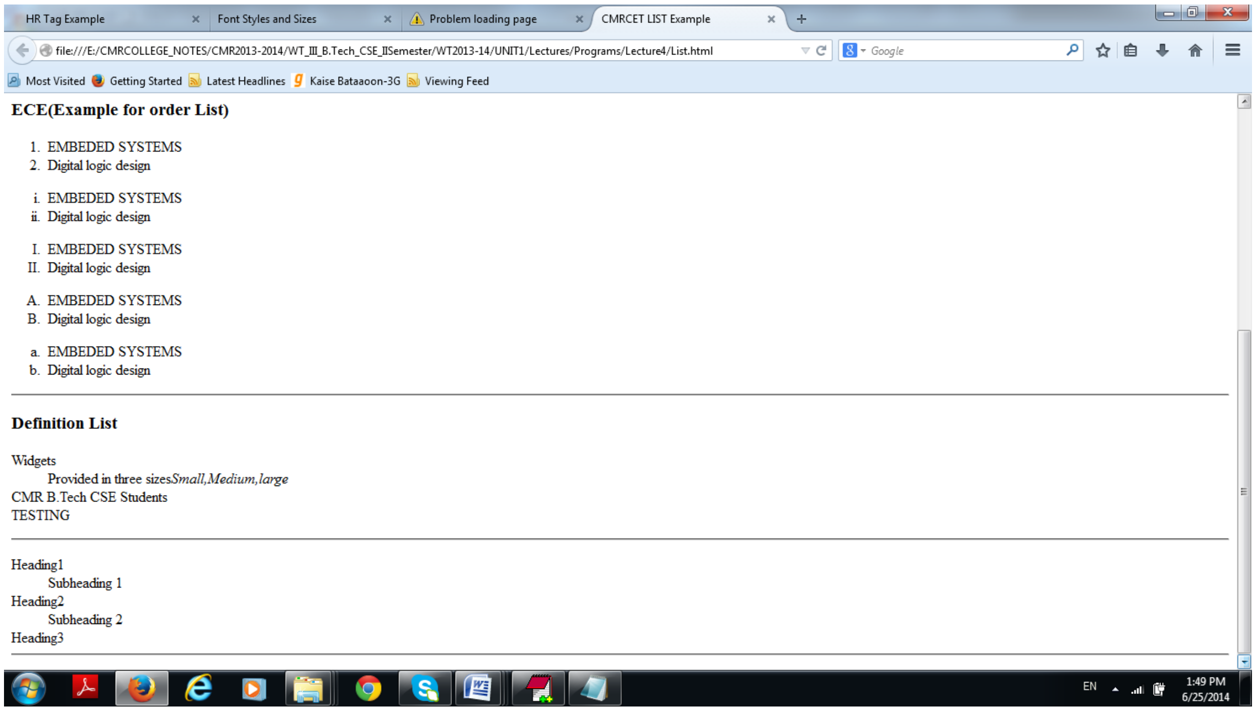
HTML Provides 3 Types of Lists:
The Basic Bulleted List/Un-ordered List
A Numbered List/Order List
A Definition List

Computer science & Engineering(Example for unorderd List)

```
• WEB TECHNOLOGIES
• compiler Design

○ WEB TECHNOLOGIES
○ compiler Design

▪ WEB TECHNOLOGIES
▪ compiler Design
```

ECE(Example for order List)

```
 1. EMBEDED SYSTEMS
 2. Digital logic design

 i. EMBEDED SYSTEMS
ii. Digital logic design

 I. EMBEDED SYSTEMS
II. Digital logic design

 A. EMBEDED SYSTEMS
```



HR Tag Example | Font Styles and Sizes | Problem loading page | CMRCET LIST Example

file:///E:/CMRCOLLEGE_NOTES/CMR2013-2014/WT_III_B.Tech_CSE_IISemester/WT2013-14/UNIT1/Lectures/Programs/Lecture4/List.html

ECE(Example for order List)

```
 1. EMBEDED SYSTEMS
 2. Digital logic design

 i. EMBEDED SYSTEMS
ii. Digital logic design

 I. EMBEDED SYSTEMS
II. Digital logic design

 A. EMBEDED SYSTEMS
 B. Digital logic design

 a. EMBEDED SYSTEMS
 b. Digital logic design
```

Definition List

```
Widgets
        Provided in three sizesSmall,Medium,large
CMR B.Tech CSE Students
TESTING
```

```
Heading1
        Subheading 1
Heading2
        Subheading 2
Heading3
```

Lecture-5

HTML Tables:
Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

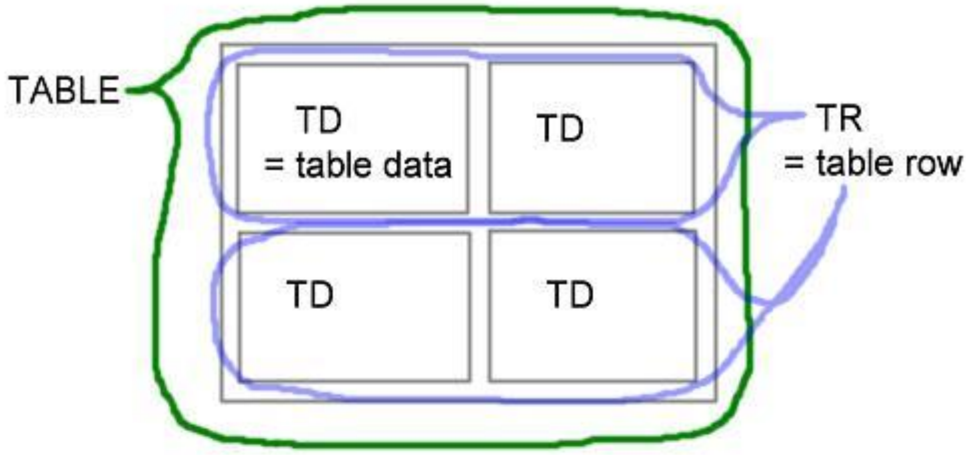
HTML Tables and the Border Attribute
If you do not specify a border attribute, the table will be displayed without borders. Sometimes this can be useful, but most of the time, we want the borders to show.

HTML Table Headers
Header information in a table are defined with the <th> tag.
All major browsers display the text in the <th> element as bold and centered.

The Basic Structure of an HTML Table

To form a table you basically need three HTML elements: **TABLE**, **TR**(Table Row) and **TD** (Table Data). How these elements are correctly nested you can see in the code example below, which creates a table of 2 rows and 2 columns rendering 4 table data cells. The graphic below the code illustrates the three elements.

```
<table>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
</table>
```



The most frequently used attributes for the **TABLE** element are: **ALIGN**,**BORDER**, **WIDTH**, **HEIGHT**, **CELLPADDING** and **CELLSPACING**. To illustrate each attribute´s effect on a table, I´ll apply them to the table in the previous example. And in order to save space here, I´ll only show you the relevant part of the HTML code.

The ALIGN Attribute

ALIGN: Possible values: **left / right / center**. By default, a table is aligned to the left margin, so you only need to include this attribute when you want to center the table or align the table to the right margin.

Example:

```
<table align="center">
```

The BORDER Attribute

BORDER: the thickness of the table border in pixels, default is 0.

Example:

```
<table align="center" border="5">
```

As you can see, the default for border color and style is a grey solid border with shadows and only the outside border takes the thickness specified with the **BORDER** attribute, while the borders around each single table cell are only 1 pixel thick.

The Attributes WIDTH and HEIGHT

WIDTH and **HEIGHT**: these two attributes define a table´s width and height, either with an absolute value in pixels or a relative value (percentage). So a value of 70 means an absolute width/height of 70 pixels and 70% means 70% of the available horizontal/vertical space.

For this example I´ve included two tables: one with an absolute width of 500 pixels and an absolute height of 200 pixels and another one with a relative width of 50 percent and a relative height of 20 percent.

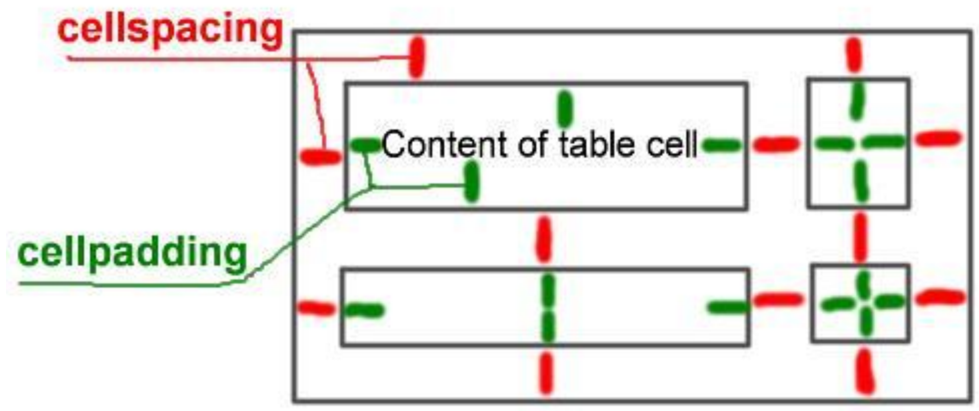
Example:

```
<table align="center" border="2" width="500" height="200">
...
...

<table align="center" border="2" width="50%" height="20%">
```

The Attributes CELLPADDING and CELLSPACING

CELLPADDING and **CELLSPACING**:

The attribute **CELLPADDING** describes the space in pixels between a table cell´s border and its content while the attribute**CELLSPACING** describes the space in pixels between the different table cells. These spaces are always equally applied to all four sides. The following graphic illustrates this. Much more flexibility in applying spaces to tables is given with CSS (Cascading Style Sheets),



Including cellspacing and cellpadding into our example table we get ...

Example:

```
<table align="center" border="5" width="500" height="200" cellpadding="10"
cellspacing="20">
```

Text Alignment inside the Table Cells

The default alignment for content (text, images) inside a table cell is left (horizontally) and middle (vertically). There are two attributes which you can use inside the **TD** element to align content: **ALIGN** for the horizontal alignment (left, center, right) and **VALIGN** for vertical alignment (top, middle, bottom).

However, a better option for aligning content within a table cell is given with CSS (Cascading Style Sheets - see lesson 10). For now, just take a look at the example below which shows you a few different options for the horizontal and vertical alignment.

Example:

Table.html

```
<html>

<head><title>HTML Table Example</title></head>

<body>

<h2 align="center">Table Demonstration</h2><br>

<table border="1" height="80%" width="80%" align="center">

<tr>

<th COLSPAN=2>Table demonstration </th>

</tr>

<tr>

  <td>First row first coloumn</td>

  <td>First row Second column</td>

</tr>

<tr>

  <td>Second row first column</td>

  <td>Second row Second column</td>

</tr>

</table>

</body>

</html>
```

Output:

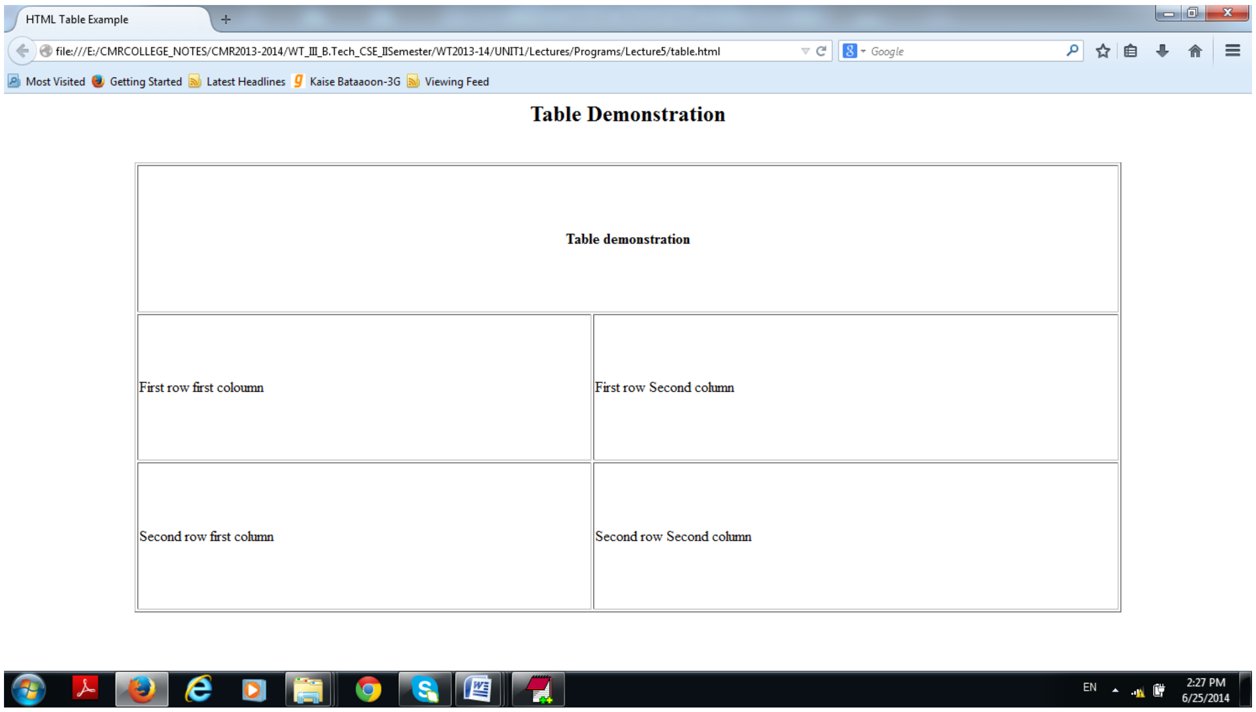

HTML Table Example
file:///E:/CMRCOLLEGE_NOTES/CMR2013-2014/WT_III_B.Tech_CSE_IISemester/WT2013-14/UNIT1/Lectures/Programs/Lecture5/table.html
Google
Most Visited
Getting Started
Latest Headlines
Kaise Bataaoon-3G
Viewing Feed
EN
2:27 PM
6/25/2014

Table Demonstration

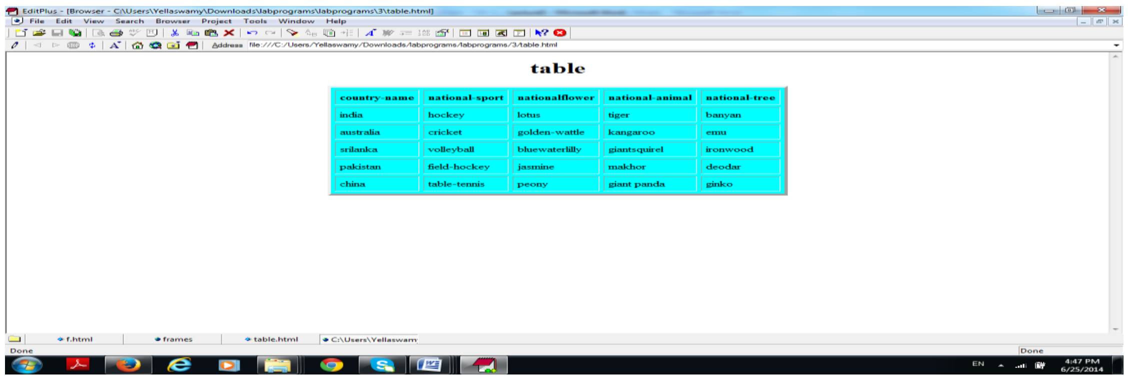
Table demonstration	
First row first coloumn	First row Second column
Second row first column	Second row Second column

Example2:

Table.html

```
<html>
<body ><h1 align="center"> table</h1>
<table  align="center" border="4" bgcolor="aqua" cellspacing="5" cellpadding="5">
<tr>
<th>country-name</th><th>national-sport</th><th>nationalflower</th><th>national-
animal</th><th>national-tree</th></tr>
<tr><td>india</td><td>hockey</td><td>lotus</td><td>tiger</td><td>banyan</td></tr>
<tr><td>australia</td><td>cricket</td><td>golden-wattle</td><td>kangaroo</td><td>emu</td></tr>
<tr><td>srilanka</td><td>volleyball</td><td>bluewaterlilly</td><td>giantsquirel</td><td>ironwood</t
d></tr>
<tr><td>pakistan</td><td>field-hockey</td><td>jasmine</td><td>makhor</td><td>deodar</td></tr>
<tr><td>china</td><td>table-tennis</td><td>peony</td><td>giant panda</td><td>ginko</td></tr>
</table>
</body>
</html>
```

OutPut:



EditPlus - [Browser - C:\Users\Yellaswamy\Downloads\labprograms\labprograms\3\table.html]

Address file:///C:/Users/Yellaswamy/Downloads/labprograms/labprograms/3/table.html

table

country-name	national-sport	nationalflower	national-animal	national-tree
india	hockey	lotus	tiger	banyan
australia	cricket	golden-wattle	kangaroo	emu
srilanka	volleyball	bluewaterlilly	giantsquirel	ironwood
pakistan	field-hockey	jasmine	makhor	deodar
china	table-tennis	peony	giant panda	ginko

UNIT-1

Lecture-6

HTML Images - The <img> Tag and the Src Attribute

In HTML, images are defined with the <img> tag.
The <img> tag is empty, which means that it contains attributes only, and has no closing tag.
To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

Syntax:

<img src="url" alt="some_text">

The URL points to the location where the image is stored. An image named "boat.gif", located in the "images" directory on "www.cmrcet.org" has the URL: http://www.cmrcet.org/images/logo.gif.

The browser displays the image where the <img> tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

HTML Images - The Alt Attribute

The required alt attribute specifies an alternate text for an image, if the image cannot be displayed.
The value of the alt attribute is an author-defined text: <img src="logo.gif" alt="CMRCET LOGO">
The alt attribute provides alternative information for an image if a user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

HTML Images - Set Height and Width of an Image

The height and width attributes are used to specify the height and width of an image.
The attribute values are specified in pixels by default:

Example:
<img src="image.jpg" alt="image text" width="304" height="228">

ATTRIBUTES OF <IMG ..>TAG

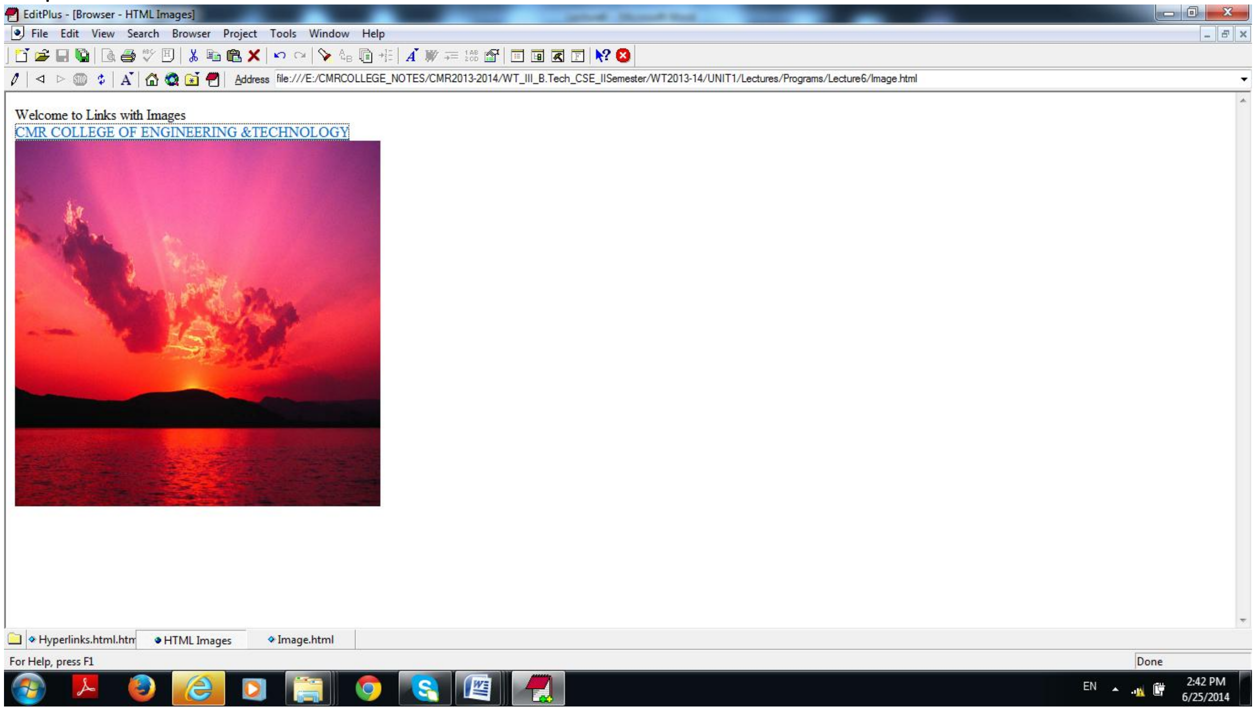
- SRC: where to get the picture
- ALT: text to show if you don't show the picture
- NAME
- WIDTH: how wide is the picture

- HEIGHT: how tall is the picture
- ALIGN: how text should flow around the picture
- BORDER: border around the picture
- HSPACE: horizontal distance between the picture and the text
- VSPACE: vertical distance between the picture and the text

Example Program:

```
<html>
<head>
<title>HTML Images</title>
</head>
<body>

Welcome to Links with Images
<br>
<a href="Hyperlink.html" target="_blank">CMR COLLEGE OF ENGINEERING &TECHNOLOGY</a>
<br>

<img src="images\Sunset.jpg" alt="addimage" width="400" height="400" />
</body>

</html>
```

Output:


EditPlus - [Browser - HTML Images]
File Edit View Search Browser Project Tools Window Help
Address file:///E:/CMRCOLLEGE_NOTES/CMR2013-2014/WT_III_B.Tech_CSE_IISemester/WT2013-14/UNIT1/Lectures/Programs/Lecture6/Image.html
Hyperlinks.html.htm
HTML Images
Image.html
For Help, press F1
Done
EN
2:42 PM
6/25/2014

```
Welcome to Links with Images
CMR COLLEGE OF ENGINEERING &TECHNOLOGY
```

UNIT-1

Lecture-7,8&9

HTML FRAMES

1. Introduction to frame
2. Layout of frames
    a. The FRAMESET element
    b. Rows and columns
    c. Nested frame sets
    d. Sharing data among frame
    e. The FRAME element
        i. Setting the initial contents of a frame
    f. Visual rendering of a frame
3. Specifying target frame information
    a. Setting the default target for links
    b. Target semantics
4. Alternate content
    a. The NOFRAMES element
    b. Long descriptions of frame
5. Inline frames: the IFRAME element

Introduction to frames

HTML frames allow authors to present documents in multiple views, which may be independent windows or subwindows. Multiple views offer designers a way to keep certain information visible, while other views are scrolled or replaced.

For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled through or replaced by navigating in the second frame.

Example:

```
<HTML>

<HEAD>

<TITLE>A simple frameset document</TITLE>

</HEAD>

<FRAMESET cols="20%, 80%">

  <FRAMESET rows="100, 200">
```

```
    <FRAME src="contents_of_frame1.html">

    <FRAME src="contents_of_frame2.gif">

  </FRAMESET>

  <FRAME src="contents_of_frame3.html">

  </FRAMESET>

</HTML>
```

Layout of frames

An HTML document that describes frame layout (called a frameset document) has a different makeup than an HTML document without frames.
A standard document has one HEAD section and one BODY.
A frameset document has a HEAD, and a FRAMESET in place of the BODY.
The FRAMESET section of a document specifies the layout of views in the main user agent window.
In addition, the FRAMESET section can contain a NOFRAMES element to provide alternate content for user agents that do not support frames or are configured not to display frames.

Elements that might normally be placed in the BODY element must not appear before the first FRAMESET element or the FRAMESET will be ignored.

The FRAMESET element

```
<![ %HTML.Frameset; [

<!ELEMENT FRAMESET - - ((FRAMESET|FRAME)+ & NOFRAMES?) -- window subdivision-->

<!ATTLIST FRAMESET

  %coreattrs;                    -- id, class, style, title --

  rows       %MultiLengths; #IMPLIED  -- list of lengths,

                              default: 100% (1 row) --

  cols       %MultiLengths; #IMPLIED  -- list of lengths,

                              default: 100% (1 col) --

  onload     %Script;       #IMPLIED  -- all the frames have been loaded  --

  onunload   %Script;       #IMPLIED  -- all the frames have been removed --

  >]]>
```

Attribute definitions

rows = multi-length-list [CN]

This attribute specifies the layout of horizontal frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%, meaning one row.

cols = multi-length-list [CN]

This attribute specifies the layout of vertical frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%, meaning one column.

Attributes defined elsewhere

id, class (document-wide identifiers)

title (element title)

style (inline style information)

onload, onunload (intrinsic events)

The FRAMESET element specifies the layout of the main user window in terms of rectangular subspaces.

Rows and columns

Setting the rows attribute defines the number of horizontal subspaces in a frameset. Setting the cols attribute defines the number of vertical subspaces. Both attributes may be set simultaneously to create a grid.

If the rows attribute is not set, each column extends the entire length of the page. If the cols attribute is not set, each row extends the entire width of the page. If neither attribute is set, the frame takes up exactly the size of the page.

Frames are created left-to-right for columns and top-to-bottom for rows. When both attributes are specified, views are created left-to-right in the top row, left-to-right in the second row, etc.

The first example divides the screen vertically in two (i.e., creates a top half and a bottom half).

```
<FRAMESET rows="50%, 50%">

...the rest of the definition...

</FRAMESET>
```

The next example creates three columns: the second has a fixed width of 250 pixels (useful, for example, to hold an image with a known size). The first receives 25% of the remaining space and the third 75% of the remaining space.

```
<FRAMESET cols="1*,250,3*">

...the rest of the definition...

</FRAMESET>
```

The next example creates a 2x3 grid of subspaces.

```
<FRAMESET rows="30%,70%" cols="33%,34%,33%">

...the rest of the definition...

</FRAMESET>
```

For the next example, suppose the browser window is currently 1000 pixels high. The first view is allotted 30% of the total height (300 pixels). The second view is specified to be exactly 400 pixels high. This leaves 300 pixels to be divided between the other two frames. The fourth frame's height is specified as "2*", so it is twice as high as the third frame, whose height is only "*" (equivalent to 1*). Therefore the third frame will be 100 pixels high and the fourth will be 200 pixels high.

```
<FRAMESET rows="30%,400,*,2*">

...the rest of the definition...

</FRAMESET>
```

Nested frame sets

Framesets may be nested to any level.

In the following example, the outer FRAMESET divides the available space into three equal columns. The inner FRAMESET then divides the second area into two rows of unequal height.

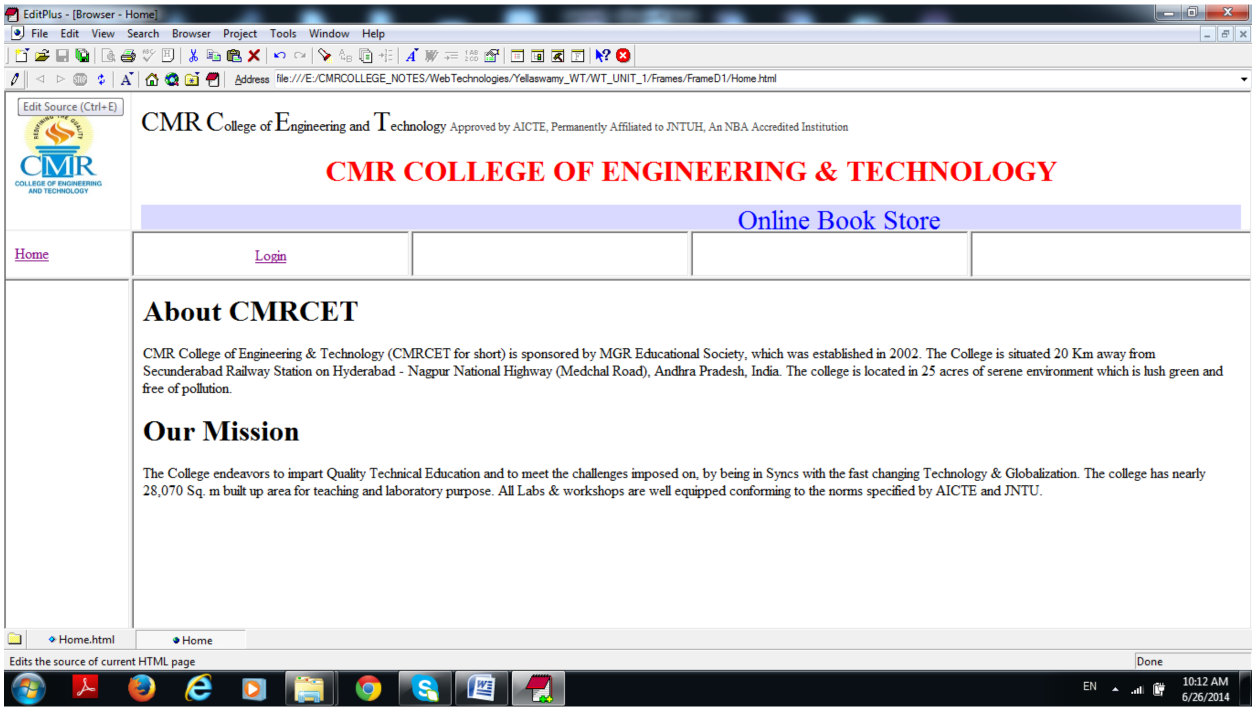
```
<FRAMESET cols="33%, 33%, 34%">

    ...contents of first frame...

    <FRAMESET rows="40%, 50%">

      ...contents of second frame, first row...

      ...contents of second frame, second row...

    </FRAMESET>

    ...contents of third frame...

</FRAMESET>
```

Examples:

Home.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> Home</TITLE>

</HEAD>
<FRAMESET ROWS="30%,10%,80%" COLS="10%,*">


        <FRAME SRC="logo.html" NAME="logo" scrolling=no frameborder="0" noresize>


        <FRAME SRC="sitename.html" NAME="sitename" scrolling=no frameborder="0" noresize>
        <FRAME SRC="main.html" NAME="main" scrolling=no frameborder="0" noresize>

<FRAMESET COLS="50%,50%,50%,50%">
                <FRAME SRC="login.html" NAME="login" scrolling=no  noresize>
                <FRAME SRC="" NAME="reg" scrolling=no  noresize>
                <FRAME SRC="" NAME="profile" scrolling=no  noresize>
                <FRAME SRC="l" NAME="catalog" scrolling=no  noresize>

</FRAMESET>
<FRAMESET  ROWS="100%">
                <FRAME SRC="" NAME="">
                <FRAME SRC="right.html" NAME="right">

</FRAMESET>
</FRAMESET>
</HTML>
```

Output:



CMR College of Engineering and Technology Approved by AICTE, Permanently Affiliated to JNTUH, An NBA Accredited Institution

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

Online Book Store

Home

Login

About CMRCET

CMR College of Engineering & Technology (CMRCET for short) is sponsored by MGR Educational Society, which was established in 2002. The College is situated 20 Km away from Secunderabad Railway Station on Hyderabad - Nagpur National Highway (Medchal Road), Andhra Pradesh, India. The college is located in 25 acres of serene environment which is lush green and free of pollution.

Our Mission

The College endeavors to impart Quality Technical Education and to meet the challenges imposed on, by being in Syncs with the fast changing Technology & Globalization. The college has nearly 28,070 Sq. m built up area for teaching and laboratory purpose. All Labs & workshops are well equipped conforming to the norms specified by AICTE and JNTU.

UNIT-1

Lecture-10

HTML Forms

HTML Forms

HTML forms are used to pass data to a server.An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

The <form> tag is used to create an HTML form:

```
<form>
.
input elements
.
</form>
```

HTML Forms - The Input Element

The most important form element is the <input> element.

The <input> element is used to select user information.

An <input> element can vary in many ways, depending on the type attribute. An <input> element can be of type text field, checkbox, password, radio button, submit button, and more.

The most common input types are described below.

Text Fields

<input type="text"> defines a one-line input field that a user can enter text into:

Example;

```
<form>

First name: <input type="text" name="firstname"><br>

Last name: <input type="text" name="lastname">

</form>
```

Password Field

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

Example:

```
<form>
Password: <input type="password" name="pwd">
</form>
```

Radio Buttons

<input type="radio"> defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

Example:

```
<form>
<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female
</form>
```

Checkboxes

<input type="checkbox"> defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example:

```
<form>
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br>
<input type="checkbox" name="vehicle" value="Car">I have a car
</form>
```

Submit Button

<input type="submit"> defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

Example:

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

Example: Registration Form

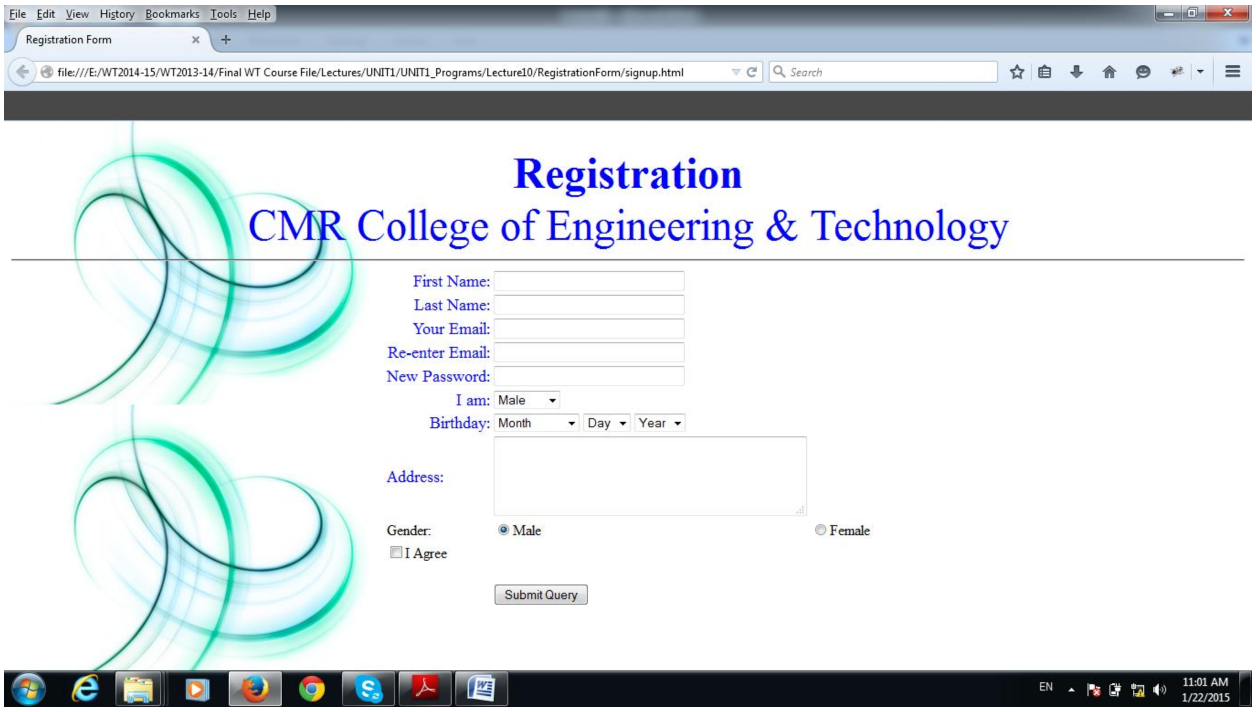
```
<html>
<head>
<title>Registration Form</title>

<style>
body
{

 background-image: url("login.jpg");

  background-repeat: repeat-X;
  background-repeat: repeat-y;
}
</style>
</head>
<body>
<br><font size="12" color="Blue"><center><b>Registration</b>
<br>CMR College of Engineering & Technology</font></center>
<hr>
```

```
<table align="center">
<form>

<tr><td align="right"><font size="4" color="Blue">First Name:</td><td><input type="text" name ="fn"
size="30"></td></tr>
<tr><td align="right"><font size="4" color="Blue">Last Name:</td><td><input type="text" name="ln"
size="30"></td></tr>
<tr><td align="right"><font size="4" color="Blue">Your Email:</td><td><input type="text"
name="email" size="30"></td></tr>
<tr><td align="right"><font size="4" color="Blue">Re-enter Email:</td><td><input type="text"
name="remail" size="30"></td></tr>
<tr><td align="right"><font size="4" color="Blue">New Password:</td><td><input type="password"
name="pw" size="30"></td></tr>
<tr><td align="right"><font size="4" color="Blue">I am:</td><td><select name="Select sex:">
<option value="male">Male</option>
<option value="female">Female</option>
</select></td></tr>
<tr><td align="right"><font size="4" color="Blue">Birthday:</td><td>
<select name="month">
<option value="na">Month</option>
<option value="1">January</option>
<option value="2">February</option>
<option value="3">March</option>
<option value="4">April</option>
<option value="5">May</option>
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select>

<select name="day">
<option value="na">Day</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
```

```
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>

<select name="year">
<option value="na">Year</option>
<option value="2012">2012</option>
<option value="2011">2011</option>
<option value="2010">2010</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option>
<option value="2006">2006</option>
<option value="2005">2005</option>
<option value="2004">2004</option>
<option value="2003">2003</option>
<option value="2002">2002</option>
<option value="2001">2001</option>
<option value="2000">2000</option>
<option value="1999">1999</option>
<option value="1998">1998</option>
<option value="1997">1997</option>
<option value="1996">1996</option>
<option value="1995">1995</option>
<option value="1994">1994</option>
<option value="1993">1993</option>
<option value="1992">1992</option>
<option value="1991">1991</option>
<option value="1990">1990</option>
<option value="1989">1989</option>
<option value="1988">1988</option>
```

```
<option value="1987">1987</option>
<option value="1986">1986</option>
<option value="1985">1985</option>
<option value="1984">1984</option>
<option value="1983">1983</option>
<option value="1982">1982</option>
<option value="1981">1981</option>
</select></td></tr>
<tr>
<td><font size="4" color="Blue">Address:</font></td>
<td><textarea rows="4" cols="40"></textarea>
</tr>
<tr>
<td>Gender:</td>
<td><input type="radio" name="gender" value="male" checked="checked">Male</input>
<td><input type="radio" name="gender" value="female">Female</input>
</tr>


<tr>
<td><input type="checkbox" name="agree" value="accept">I Agree</input></td>
</tr>
<tr><td></td><td><br><input type="submit" size="10"></td>



</form></table>
</body>
</html>
```

Output


File Edit View History Bookmarks Tools Help
Registration Form
file:///E:/WT2014-15/WT2013-14/Final WT Course File/Lectures/UNIT1/UNIT1_Programs/Lecture10/RegistrationForm/signup.html
Search

Registration

CMR College of Engineering & Technology

First Name:
Last Name:
Your Email:
Re-enter Email:
New Password:
I am: Male
Birthday: Month Day Year
Address:
Gender: (● Male) ○ Female
☐ I Agree
Submit Query

UNIT-1

Lecture-11&12

Cascading Style Sheets

What is CSS?

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save a lot of work
- External Style Sheets are stored in CSS files

Styles Solved a Big Problem

HTML was never intended to contain tags for formatting a document.

HTML was intended to define the content of a document, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

In HTML 4.0, all formatting could be removed from the HTML document, and stored in a separate CSS file.

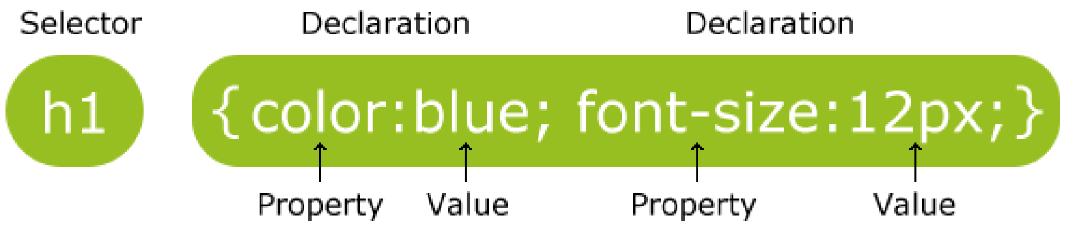
All browsers support CSS today.CSS Saves a Lot of Work!

CSS defines HOW HTML elements are to be displayed.

Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

CSS Example

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
color:red;
text-align:center;
}
</style>
</head>

<body>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
</body>
</html>
```

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment begins with "/*", and ends with "*/", like this:

```
/*This is a comment*/

p
{
text-align:center;
/*This is another comment*/
color:black;
font-family:arial;
}
```

The id and class Selectors

In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

The id Selector

- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
- The style rule below will be applied to the element with id="para1":

Example:

```
#para1

{

text-align:center;

color:red;

}
```

```
<html>
<head>
<style>
#para1
{
text-align:center;
color:red;
}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

The class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

This allows you to set a particular style for many HTML elements with the same class.

The class selector uses the HTML class attribute, and is defined with a "."
In the example below, all HTML elements with class="center" will be center-aligned:
Example:

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

```
<html>
<head>
<style>
.center
{
text-align:center;
}
</style>
</head>
<body>
<h1 class="center">Center-aligned heading</h1>
<p class="center">Center-aligned paragraph.</p>
</body>
</html>
```

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet/embedded style sheet
3. Inline style

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css"/>
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

```
hr {color:sienna;}

p {margin-left:20px;}

body {background-image:url("images/back40.gif");};}

</style>

</head>
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:

```
<head>

<style>

hr {color:sienna;}

p {margin-left:20px;}

body {background-image:url("images/back40.gif");}

</style>

</head>
```

Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the h3 selector:

```
h3

{

color:red;

text-align:left;

font-size:8pt;

}
```

And an internal style sheet has these properties for the h3 selector:

```
h3

{

text-align:right;

font-size:20pt;

}
```

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

```
color:red;

text-align:right;

font-size:20pt;
```

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

Multiple Styles Will Cascade into One

Styles can be specified:

- inside an HTML element
- inside the head section of an HTML page
- in an external CSS file

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

Cascading order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).

Remark Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

Example: External Style sheet

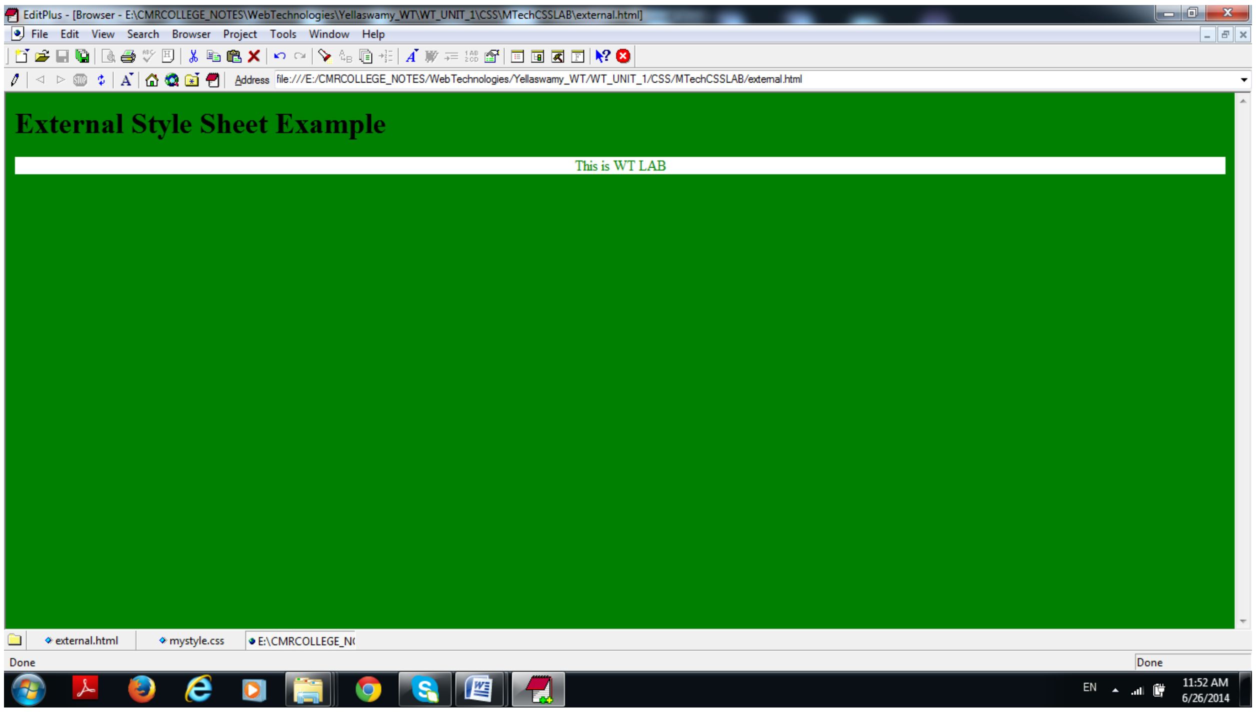
External.html

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css" type="text/css">
</head>
<body>
<h1>External Style Sheet Example</h1>
<p>This is  WT LAB</p>
</body>
</html>
```

mystyle.css

```
p
{
color:green;
text-align:center;
background-color: white;
}
body
{
background-color: green;
}
```

OUTPUT



EditPlus - [Browser - E:\CMRCOLLEGE_NOTES\WebTechnologies\Yellaswamy_WT\WT_UNIT_1\CSS\MTechCSSLAB\external.html]
File Edit View Search Browser Project Tools Window Help
Address file:///E:/CMRCOLLEGE_NOTES/WebTechnologies/Yellaswamy_WT/WT_UNIT_1/CSS/MTechCSSLAB/external.html
external.html
mystyle.css
Done

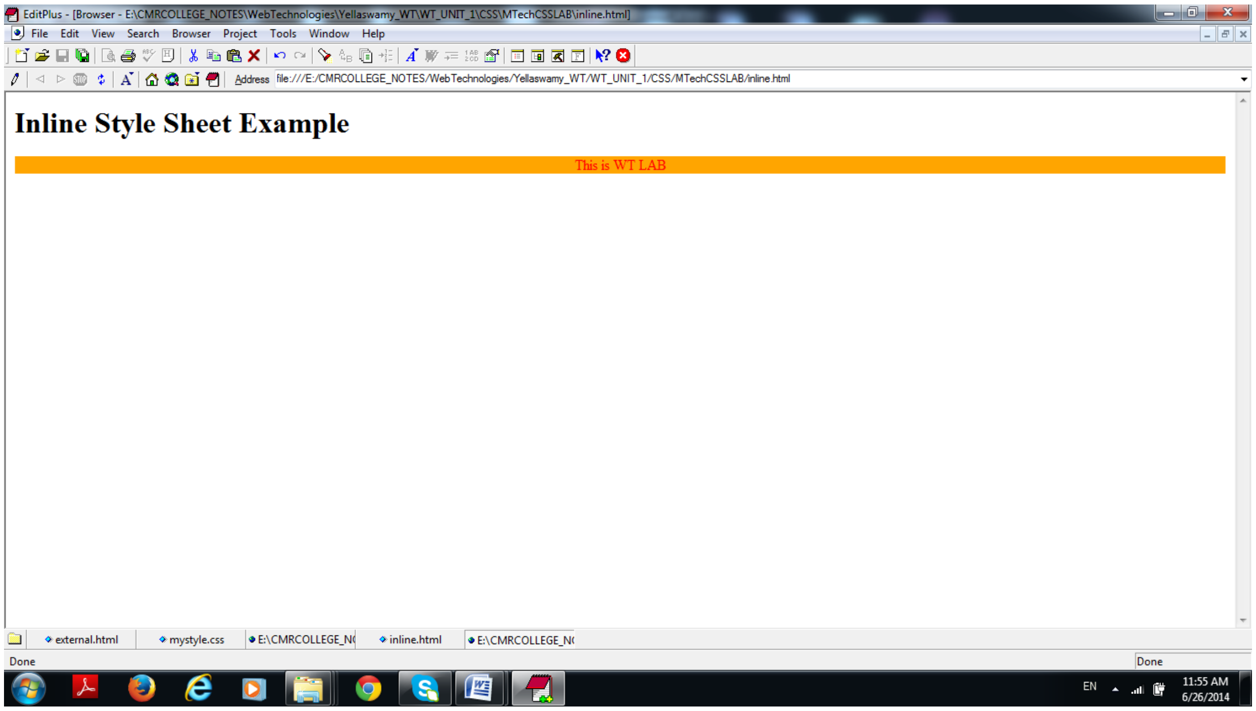
External Style Sheet Example

This is WT LAB

Example:Inline Style sheet

```
<html>
<head>
</head>
<body>
<h1>Inline Style Sheet Example</h1>
<p style="color:red;text-align:center;background-color: orange;">This is  WT LAB</p>
</body>
</html>
```

Output:


EditPlus - [Browser - E:\CMRCOLLEGE_NOTES\WebTechnologies\Yellaswamy_WT\WT_UNIT_1\CSS\MTechCSSLAB\inline.html]
File Edit View Search Browser Project Tools Window Help
Address file:///E:/CMRCOLLEGE_NOTES/WebTechnologies/Yellaswamy_WT/WT_UNIT_1/CSS/MTechCSSLAB/inline.html
external.html
mystyle.css
inline.html
Done

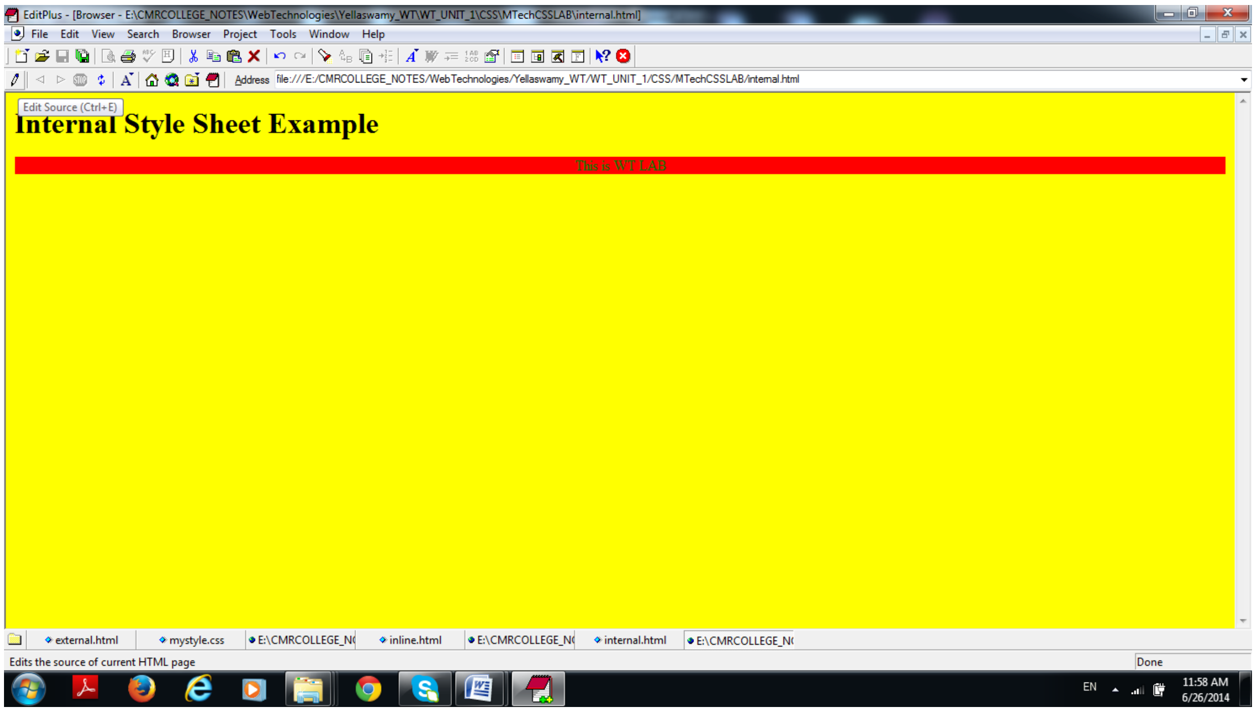
Inline Style Sheet Example

This is WT LAB

Example: Internal Style Sheet

```
<html>
<head>
<style>
p
{
color:green;
text-align:center;
background-color: red;
}
body
{
background-color: yellow;
}
</style>
</head>
<body>
<h1>Internal Style Sheet Example</h1>
<p>This is  WT LAB</p>
</body>
</html>
```

OUTPUT:


EditPlus - [Browser - E:\CMRCOLLEGE_NOTES\WebTechnologies\Yellaswamy_WT\WT_UNIT_1\CSS\MTechCSSLAB\internal.html]
File Edit View Search Browser Project Tools Window Help
Address file:///E:/CMRCOLLEGE_NOTES/WebTechnologies/Yellaswamy_WT/WT_UNIT_1/CSS/MTechCSSLAB/internal.html
Edit Source (Ctrl+E)
external.html
mystyle.css
E:\CMRCOLLEGE_N
inline.html
E:\CMRCOLLEGE_N
internal.html
E:\CMRCOLLEGE_N
Edits the source of current HTML page
Done
EN
11:58 AM
6/26/2014

Internal Style Sheet Example

This is WT LAB

All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated

All CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
unicode-bidi	
vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in a text

All CSS Font Properties

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text

font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

All CSS List Properties

Property	**Description**
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow
list-style-type	Specifies the type of list-item marker