

UNIT-2 - HTML

Introduction:

HTML stands for Hypertext Markup Language. It is used to display the document in the web browsers. HTML pages can be developed to be simple text or to be complex multimedia program containing sound, moving images and java applets. HTML is considered to be the global publishing format for Internet. It is not a programming language. HTML was developed by Tim Berners-Lee. HTML standards are created by a group of interested organizations called W3C (World Wide Web consortium). In HTML formatting is specified by using tags. A tag is a format name surrounded by angle brackets. End tags which switch a format off also contain a forward slash. Points to be remembered for HTML tags:

- ❖ Tags are delimited by angled brackets.
- ❖ They are not case sensitive i.e., <head>, <HEAD> and <Head> is equivalent.
- ❖ If a browser not understands a tag it will usually ignore it.
- ❖ Some characters have to be replaced in the text by escape sequences.
- ❖ White spaces, tabs and newlines are ignored by the browser.

Structure of an HTML document:

All HTML documents follow the same basic structure. They have the root tag as <html>, which contains <head> tag and <body> tag. The head tag is used for control information by the browser and the body tag contains the actual user information that is to be displayed on the screen. The basic document is shown below.

Sample.html

```
<html>
<head>
<title> Basic HTML document </title>
</head>
<body>
<h1> Welcome to the world of Web Technologies</h1>
<p> A sample HTML program </p>
</body>
</html>
```

Output:



Besides head and body tag, there are some other tags like title, which is a sub tag of head that displays the information in the title bar of the browser.

<h1> is used to display the line in its own format i.e., bold with some big font size.

<p> is used to write the content in the form of paragraph.

Comments in HTML documents start with **<!-- and end with -->**. Each comment can contain as many lines of text as you like. If comment is having more lines, then each line must start and end with -- and must not contain -- within its body.

<!-- this is a comment line **--**

-- which can have more lines **-->**

Basic HTML tags:

1. **Body tag** : Body tag contain some attributes such as bgcolor, background etc. bgcolor is used for background color, which takes background color name or hexadecimal number and #FFFFFF and background attribute will take the path of the image which you can place as the background image in the browser.

<body bgcolor="#F2F3F4" background= "c:\amer\imag1.gif">

2. **Paragraph tag**: Most text is part of a paragraph of information. Each paragraph is aligned to the left, right or center of the page by using an attribute called align.

<p align="left" | "right" | "center">

3. **Heading tag**: HTML is having six levels of heading that are commonly used. The largest heading tag is **<h1>**. The different levels of heading tag besides **<h1>** are **<h2>**, **<h3>**, **<h4>**, **<h5>** and **<h6>**. These heading tags also contain attribute called as align.

<h1 align="left" | "right" | "center"><h2>

4. **hr tag**: This tag places a horizontal line across the system. These lines are used to break the page. This tag also contains attribute i.e., width which draws the horizontal line with the screen size of the browser. This tag does not require an end tag.

<hr width="50%">.

5. **base font**: This specify format for the basic text but not the headings.

<basefont size="10">

6. **font tag**: This sets font size, color and relative values for a particular text.

7. **bold tag**: This tag is used for implement bold effect on the text

****

8. **Italic tag**: This implements italic effects on the text.

<i>..... </i>

9. **strong tag**: This tag is used to always emphasized the text

** **

10. **tt tag**: This tag is used to give typewriting effect on the text

<tt>..... </tt>

11. **sub and sup tag**: These tags are used for subscript and superscript effects on the text.

_{.....}

^{.....}

12. **Break tag:** This tag is used to break the line and start from the next line.

13. **& < > ":** These are character escape sequences which are required if you want to display characters that HTML uses as control sequences.

Example: < can be represented as <.

14. **Anchor tag:** This tag is used to link two HTML pages, this is represented by <a>

 some text

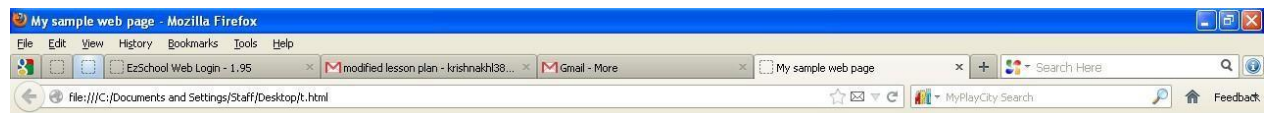
href is an attribute which is used for giving the path of a file which you want to link.

Example 1: HTML code to implement heading tags.

heads.html

```
<html>
<head> <!-- This page implements heading html tags -->
<title> My sample web page </title>
</head>
<body >
<h1>heading level 1</h1>
<h2>heading level 2</h2>
<h3 align="center">heading level 3</h3>
<h4 align="right">heading level 4</h4>
<h5 align="left">heading level 5</h5>
<h6>heading level 6</h6>
</body>
</html>
```

Output:



heading level 1

heading level 2

heading level 3

heading level 4

heading level 5

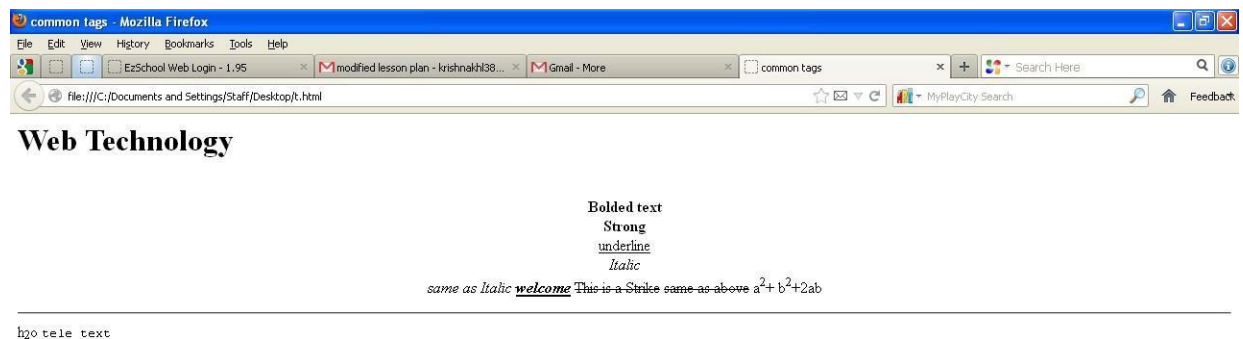
heading level 6

Example 2: HTML code to implement common tags.

common.html

```
<html>
<head><title> common tags </title></head>
<body>
<h1>Web Technology</h1>
<p align="center"><br>
<b>Bolded text</b><br>
<strong>Strong</strong><br>
<u>underline</u><br>
<i>Italic</i><br>
<em>same as Italic</em>
<b><i><u>welcome</u></i></b>
<strike>This is a Strike</strike>
<s>same as above</s>
a<sup>2</sup>+ b<sup>2</sup>+2ab<hr>
h<sub>2</sub>o
<tt>tele text</tt>
<font face="Arial" size="3" color="red">
</body>
</html>
```

Output:



Lists:

One of the most effective ways of structuring a web site is to use lists. Lists provides straight forward index in the web site. HTML provides three types of list i.e., bulleted list, numbered list and a definition list. Lists can be easily embedded easily in another list to provide a complex but readable structures. The different tags used in lists are explained below.

``

The **ordered (numbered)** and **unordered (bulleted)** lists are each made up of sets of list items. This tag is used to write list items

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

This tag is used for basic unordered list which uses a bullet in front of each tag, every thing between the tag is encapsulated within `` tags.

`<ol type="1" | "a" | "I" start="n">.....`

This tag is used for unordered list which uses a number in front of each list item or it uses any element which is mentioned in the type attribute of the `` tag, start attribute is used for indicating the starting number of the list.

`<dl>..... </dl>`

This tag is used for the third category i.e., definition list, where numbers or bullet is not used in front of the list item, instead it uses definition for the items.

`<dt>.....</dt>`

This is a sub tag of the `<dl>` tag called as definition term, which is used for marking the items whose definition is provided in the next data definition.

`<dd></dd>`

This is a sub tag of the `<dd>` tag, definition of the terms are enclosed within these tags. The definition may include any text or block.

Example 3: HTML code showing list tags. **List.html**

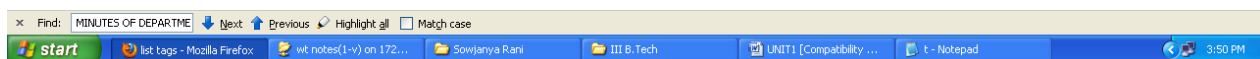
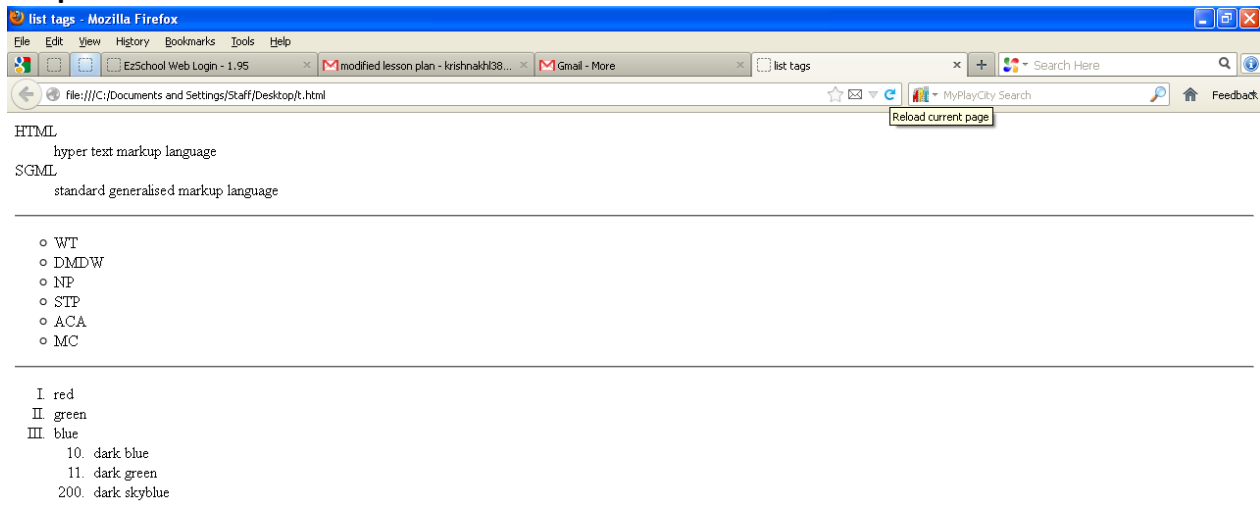
```
<html>
<head><title>list tags</title></head>
<body>
<dl>
<dt>HTML</dt>
<dd>hyper text markup language</dd>
<dt>SGML</dt>
<dd>standard generalised markup language</dd>
</dl><hr>
<ul type=circle>
<li>WT</li>
<li>DMDW</li>
<li>NP</li>
<li>STP</li>
<li>ACA</li>
<li>MC</li>
</ul>
```

```

<hr>
<ol type="I">
<li>red</li>
<li>green</li>
<li>blue</li>
    <ol start=10>
    <li>dark blue</li>
    <li>dark green</li>
    <li value=200>dark skyblue</li>
    </ol>
</ol>
<hr>
<h1></h1>
</body>
</html>

```

Output:



Tables:

Table is one of the most useful HTML constructs. Tables are found all over the web application. The main use of table is that they are used to structure the pieces of information and to structure the whole web page. Below are some of the tags used in table.

```
<table align="center" | "left" | "right" border="n" width="n%"  
cellpadding="n" cellspacing="n"> </table>
```

Everything that we write between these two tags will be within a table. The attributes of the table will control in formatting of the table. Cell padding determines how much space there is between the contents of a cell and its border, cell spacing sets the amount of white space between cells. Width attributes sets the amount of screen that table will use.

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

This is the sub tag of <table> tag, each row of the table has to be delimited by these tags.

```
<th>.....</th>
```

This is again a sub tag of the <tr> tag. This tag is used to show the table heading .

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

This tag is used to give the content of the table.

Example 4: HTML code showing the use of table tag

Table.html

```
<html>  
<head>  
<title>time table</title>  
</head>  
<body>  
<table align="center" border="1" cellpadding="25" cellspacing="5">  
<tr bgcolor="yellow">  
<th rowspan=2>Name</th>  
<th rowspan=2>Sem</th>  
<th colspan=3> Marks</th></tr>  
<tr bgcolor="pink">  
<td>sub1</td>  
<td>sub2</td>  
<td>sub3</td></tr>  
<tr bgcolor="orange">  
<td rowspan=4>cse</td>  
<td rowspan=2> 1st</td>  
<td>A</td><td>B</td><td>C</td></tr>  
<tr><td>100</td><td>100</td><td>100</td></tr>  
<tr bgcolor="lightgreen"><td rowspan=2>  
2nd</td><td>D</td><td>E</td><td>F</td></tr>
```



```

<tr bgcolor="lightblue"><td>100</td><td>100</td><td>100</td></tr>
</table><br><br>
</body>
</html>

```

Output:

time table - Mozilla Firefox

File Edit View History Bookmarks Tools Help

EzSchool Web Login - 1.95 modified lesson plan - krishnaH38... Gmail - More time table

Search Here

file:///C:/Documents and Settings/Staff/Desktop/t.html

MyPlayCity Search

Feedback

Name	Sem	Marks		
		sub1	sub2	sub3
cse	1st	A	B	C
		100	100	100
	2nd	D	E	F
		100	100	100

Hyperlinks:

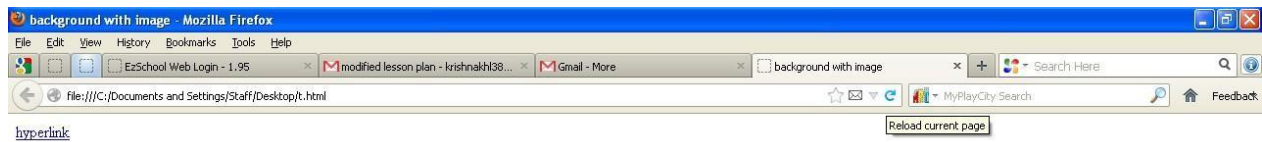
To create a link in an HTML page, you use the HTML link tag `<A>...`. The `<A>` tag is often called an anchor tag, as it can also be used to create anchors for links. The most common use of the link tag, however, is to create links to other pages.

Example 5: HTML code that implements color,image and background.

```

<html>
<head><!-- This page implements color,image and background -->
<title> background with image </title>
</head>
<body>
<a href="HTML/links.html" target="-blank">hyperlink</a>
</body>
</html>

```



Color and Image:

Color can be used for background, elements and links. To change the color of links or of the page background hexadecimal values are placed in the <body> tag.

```
<body bgcolor = "#nnnnnn" text = "#nnnnnn" link = "#nnnnnn" vlink = "#nnnnnn" alink = "#nnnnnn">
```

The vlink attribute sets the color of links visited recently, alink the color of a currently active link. The six figure hexadecimal values must be enclosed in double quotes and preceded by a hash(#).

Images are one of the aspect of web pages. Loading of images is a slow process, and if too many images are used, then download time becomes intolerable. Browsers display a limited range of image types.

```
<body background = "URL">
```

This tag will set a background image present in the URL.

Another tag that displays the image in the web page, which appears in the body of the text rather than on the whole page is given below

```

```

Example 6: HTML code that implements color,image and background.

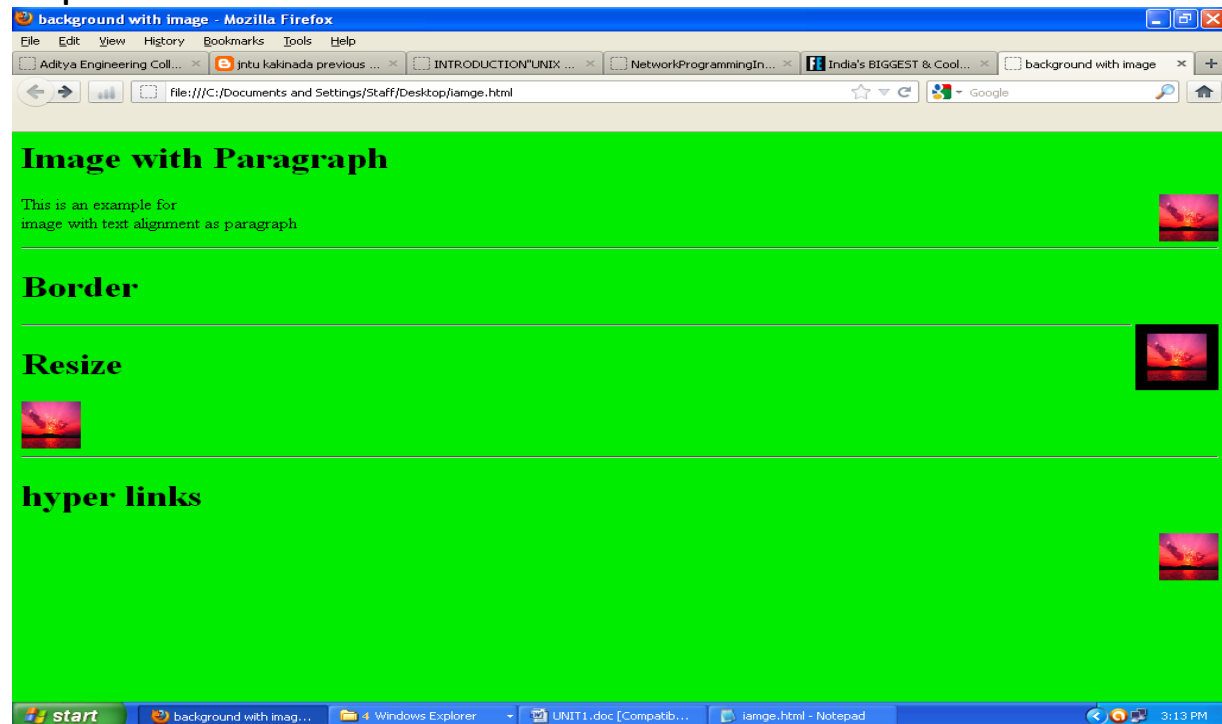
```
<html>
<head> <!-- This page implements color,image and background -->
<title> background with image </title>
</head>
<body bgcolor="red" background="flowers.jpg ">
<h1>Image with Paragraph</h1>
<p>
 This is
an example for<br>
image with text alignment as paragraph <br>
</p><hr>
<h1>Border</h1>

<hr>
<h1>Resize</h1>

<hr>
<h1>hyper links</h1>
<a href="HTML/links.html">

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

Output:



Frames:

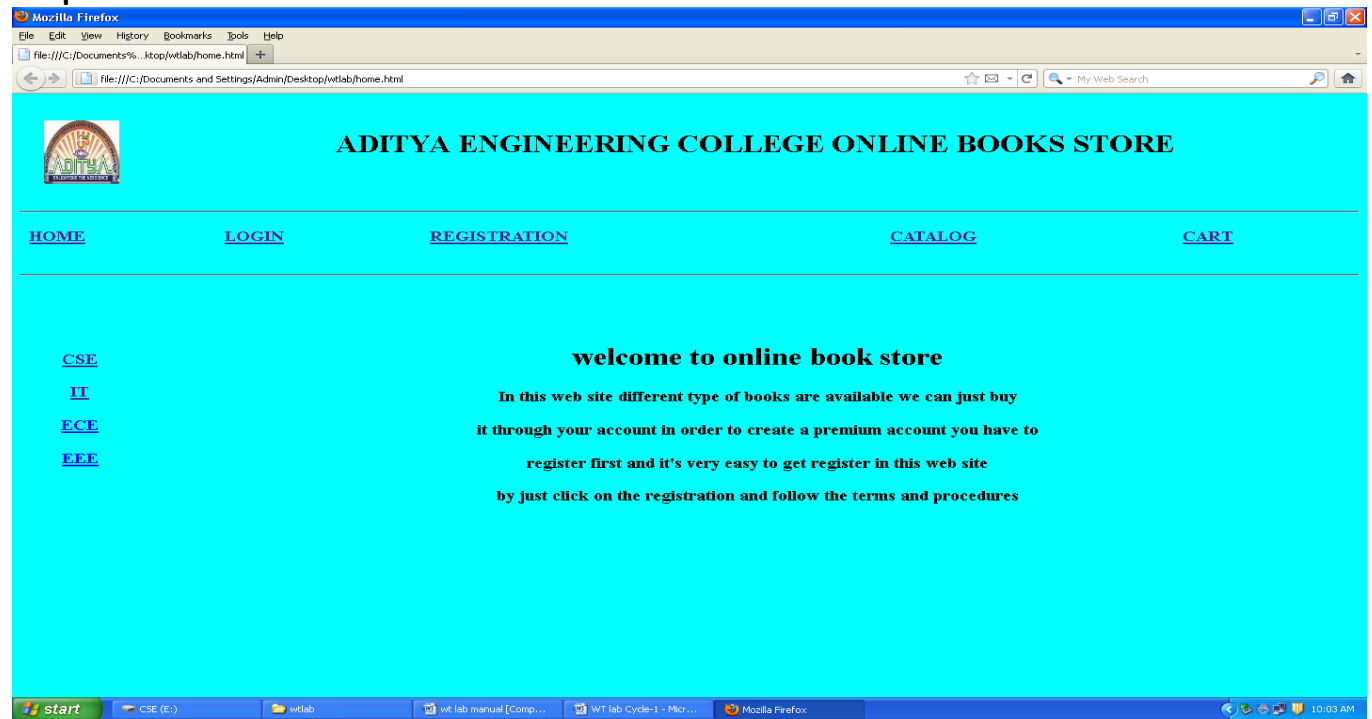
Frames provide a pleasing interface which makes your web site easy to navigate. When we talk about frames actually we are referring to frameset, which is a special type of web page. The frameset contains a set of references to HTML files, each of which is displayed inside a separate frame. There are two tags related to frames i.e., frameset and frame

```
<frameset cols=" % , %" | rows=" % , %" > ..... </frameset>
<frame name="name" src="filename" scrolling =" yes" | "no" frameborder
="0"|"1">
```

Example 7: HTML code that implements frames

```
<html>
<head>
<title> frames </title>
</head>
<frameset rows="40%,60%" >
<frame src="frame.html" scrolling="no" >
<frameset cols="10%,90%" >
<frame src="branch.html">
<frame src="inf.html" name="f1">
</frame>
</frame>
</frameset>
</frame>
</frameset>
</html>
```

Output:



Forms:

Forms are the best way of adding interactivity of element in a web page. They are usually used to let the user to send information back to the server but can also be used to simplify navigation on complex web sites. The tags that use to implement forms are as follows.

<form action="URL" method = "post" | "get">.....</form>

When get is used, the data is included as part of the URL. The post method encodes the data within the body of the message. Post can be used to send large amount of data, and it is more secure than get. The tags used inside the form tag are:

<input type = "text" | "password" | "checkbox" | "radio" | "submit" name="string" value="string" size="n">

In the above tag, the attribute type is used to implement text, password, checkbox, radio and submit button.

Text: It is used to input the characters of the size n and if the value is given than it is used as a default value. It uses single line of text. Each component can be given a separate name using the name attribute.

Password: It works exactly as text, but the content is not displayed to the screen, instead an * is used.

Radio: This creates a radio button. They are always grouped together with a same name but different values.

Checkbox: It provides a simple checkbox, where all the values can be selected unlike radio button.

Submit: This creates a button which displays the value attribute as its text. It is used to send the data to the server.

<select name="string">.....</select>

This tag helps to have a list of item from which a user can choose. The name of the particular select tag and the name of the chosen option are returned.

<option value="string" selected>.....</option>

The select statement will have several options from which the user can choose. The values will be displayed as the user moves through the list and the chosen one returned to the server.

<textarea name="string" rows="n" cols="n">.....</textarea>

This creates a free format of plain text into which the user can enter anything they like. The area will be sized at rows by cols but supports automatic scrolling.

Example 8: HTML code that implements forms

```
<html>
<head>
<title>form</title>
</head>
<body>
<p align="left">Name:<input type="text" maxlength=30 size=15>
<p align="left">Password:<input type="password" maxlenght=10 size=15>
<p align="left">Qualification: <br>
<input type="checkbox" name="q" value="be">B.E
```

```

<input type="checkbox" name="q" value="me">M.E
<p align="left">Gender:<br>
<input type="radio" name="g" value="m">Male
<input type="radio" name="g" value="f">Female
<p align="left">course:
<select name="course" size=1>
<option value=cse >CSE</option>
<option value=it>CSIT</option>
</select>
<p align="left">Address:<br>
<textarea name="addr" rows=4 cols=5 scrolling=yes></textarea>
<p align="center"><input type="submit" name="s" value="Accept">
<p align="center"><input type="reset" name="c" value="Ignore">
</body>
</html>

```

Output:

The screenshot shows a Mozilla Firefox browser window displaying the rendered HTML form. The form contains the following elements:

- Name:
- Password:
- Qualification: ☐ B.E ☐ M.E
- Gender: ☐ Male ☐ Female
- course:
- Address:
- Buttons: and



Cascading Style Sheets:

The anatomy of a style sheet includes some terminology that is likely new to you, such as declarations and selectors. Here are some examples of how these terms relate to style sheet programming.

Rules

At the very core of CSS are rules. The following is an example of a simple CSS rule: `h1 {color:green;}`

Rules consist of two parts - the selector and the declaration. In the example above, the H1 tag is the selector (the object being modified), and the color is the declaration

The following is an example of a CSS declaration:
`{color:green;}`

The declaration has two parts - the property and the value. In the example, above, property is the color of the H1 tag that is being modified. The value is set to green, which specifies how the property is being modified.

Adding CSS to XHTML

Defining the CSS rules is only a part of adding CSS to your pages. Once you have defined your rules, you need to add the CSS rules to your XHTML document so it is rendered properly by the browsers. This is done by simply adding a small bit of code within the `<head>` tag of your XHTML document.

```
<html>
<head>
<style type="text/css">
h1 {
    color: #FF0000;
}
</style>
</head>
<body>
<h1>How are you doing today?
</h1>
</body>
</html>
```

One of the most important aspects of HTML is the capability to separate presentation and content. A style is simply a set of formatting instructions that can be applied to a piece of text. There are three mechanisms by which we can apply styles to our HTML documents.

- Style can be defined within the basic HTML tag.
- Style can be defined in the `<head>` tag
- Styles can be defined in external files called stylesheets which can then be used in any document by including the stylesheet via a URL.

A style has two parts: a **selector** and a **set of declarations**.

- ❖ The **selector** is used to create a link between the rule and the HTML tag.
- ❖ The **declaration** has two parts: a property and a value. Declarations must be separated using colons and terminated using semicolons.

Selector{property: value; property: value}

Types of Style Sheets

There are three popular types of style sheets: Internal, External, and Inline. There is also another type of style sheet called as Imported.

1. Internal Style Sheets

Here is an example of “internal” style sheet. Notice that the <style> element is used inside the <head> portion of the page. All of the formatting information is contained within the HTML page. These types of style sheets will only affect the appearance of this single page. They can be effective for creating design efficiency within the single page.

Example:

```
<html>
<head>
<style type="text/css">
<!--
h1 {
    color: #FF0000;
} -->
</style>
</head>
<body>
<h1>How are we doing today? </h1>
</body>
</html>
```

2. External Style Sheets

Here's an example of an “external” style sheet. Notice the <link> element is using the <href> attribute to point to the “external.css” file. The external.css file is the document that contains all the formatting information for this XHTML page. Because the formatting information exists independent of the XHTML page, it can easily be applied to several XHTML pages and updated more efficiently.

```
<html>
<head>
<link href="external.css" rel="stylesheet" type="text/css" />
</head>
<body>
<h1> Check it Out! </h1>
<p> My style is really cool! </p>
</body>
</html>
```


The file this document references is called external.css. You can give your style sheet any name you want; just be sure to save it in text only mode and upload it to the same directory. Make sure to give it a .css extension. You don't have to put an external style sheet inside a comment tag because no style information is stored within the XHTML code.

Your **external.css** document could look like this:

```
h1 {color: red; font-family: sans-serif}  
p {background: black; color: yellow; font-family: verdana;}
```

3. Inline Style Sheets

An "inline" style sheet only applies to the parts of an HTML document that are specified and will override any style settings being applied by an external or internal style sheet.

```
<html>  
<head>  
</head>  
<body>  
<h1 style="font-family:sans-serif; color: gray"> Check it Out! </h1>  
<h1> My style is really cool! </h1>  
</body>  
</html>
```

In the above example, Inline style will override both external and internal style. Internal will override external.

Properties and values in styles:

Fonts:

font-family: family name;
font-style: normal | italic | oblique;
font-weight: normal | bold | bolder | lighter;
font-size: small | medium | large | smaller | larger;

Backgrounds and Colors:

color: value;
background-color: value;
background-image: URL;
background-repeat: repeat-X;
background-position: center | top | bottom | left | right | center | x y | ...;

Text:

text-decoration: none | underline | overline | line-through | blink;
text-transformation: none | uppercase | lowercase | capitalize; text-align: left | right | center | justify;
line-height : length | percentage;
letter-spacing: length | percentage;
word-spacing: length | percentage;

link:

```
a:link{color:green;font-size:18px;}  
a:hover{color:orange;} a:visited{color:blue;text-  
transform:uppercase;} a:active{color:black;}
```

list:

list-style-type: lower-greek | upper-latin | roman;

list-style-image: url("new.gif");

border:

border-style:dashed | solid | double | dotted;

border-color:color;

border-width:top | bottom | left | right;

table:

width:length | percentage;

border:solid | dotted...;

vertical-align:bottom | center;

Three Ways to Insert CSS:

There are three ways of inserting a style sheet:

1. External style sheet or Linking

```
<head>
```

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

```
</head>
```

2. Internal style sheet

```
hr
```

```
{color:sienna;}
```

```
p {margin-left:20px;}
```

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

3. Inline or Embedded style

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

Example : HTML code representing cascading style sheet

```
<html>
<head>
<title>My Web Page</title>
<link href="CSS\MyStyle.css" rel="stylesheet" type="text/css"/>
<style type="text/css"> li{
list-style-image:url('CSS//new.gif');
}
.newstyle{
text-decoration:line-through;
font-size:20px;
background-color:orange;
}
p{
border-style:dotted;
font-family:courier;
font-size:20px;
color:blue;
font-weight:normal;
font-style:italic;
}
a:visited{color:black;}
a:hover{visibility:hidden;}
a:active{text-transform:uppercase;color:blue;}
a:link{color:green;text-decoration:blink;}
td{
color:red;
border-top-style:dotted;
border-bottom-style:solid;
border-width:2px;
border-left-width:10px;
border-top-color:blue;
border-bottom-color:green;
}
th{
color:blue;
letter-spacing:-3px;
text-align:right;
background-color:red;
text-decoration:blink;
text-transform:capitalize;
}
</style>
</head>
<body>
<br><ul>
```

```

<li>Red</li>
<li>Blue</li>
<li>Green</li></ul>
<p>This is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading
Cascading
<span class="newstyle">StyleSheetThis is Cascading StyleSheetThis is Cascading
StyleSheetThis is</span> Cascading StyleSheetThis is Cascading StyleSheetThis is
Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheet
</p><br>
<table width=70% align=center border>
<tr>
<th>S.No</th>
<th>Name</th>
<th>Marks</th>
<th>Profile</th>
</tr>
<tr>
<td>1</td>
<td class="newstyle">Kiran</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr>
<tr>
<td>2</td>
<td>Aravind</td>
<td>99</td>
<td><a href="Profile1.html" class="newstyle">View Profile</a></td>
</tr><tr>
<td>3</td>
<td>Praveen</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr><tr>
<td class="newstyle">4</td>
<td>Arun</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr><tr>
<td>5</td>
<td class="newstyle">Raju</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr><tr>
<td>6</td>
<td>Ramu</td>
<td class="newstyle">99</td>

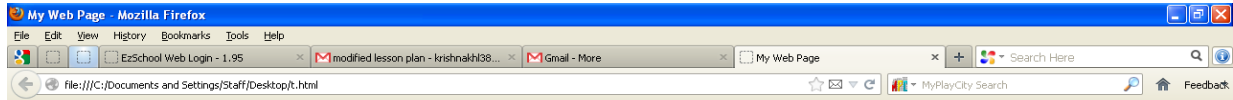
```

```

<td><a href="Profile.html">View Profile</a></td>
</tr>
</table>
</body>
</html>

```

Output:



- Red
- Blue
- Green

This is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheetThis is Cascading StyleSheet

Sl	Emp	Age	Job
1	Karan	99	View Profile
2	Aravind	99	View Profile
3	Praveen	99	View Profile
4	Arun	99	View Profile
5	Raju	99	View Profile
6	Ramu	99	View Profile

Grouping Selectors

As you add more and more styles to your pages, you might find yourself making the same stylistic change to multiple XHTML elements. In these cases, you might consider grouping your CSS selects. This can shorten the amount of code and make for a quicker download.

```
h1 {color: blue; font-family: verdana}  
h2 {color: blue; font-family: verdana}  
h3 {color: blue; font-family: verdana}
```

They could be optimized and grouped like this:

```
h1, h2, h3 {color: blue; font-family: verdana}
```

This rule specifies that all the text within h1, h2, h3 tags will display in the Verdana font with a blue color.

Class Selectors

In the preceding examples, you learned how to create styles based on XHTML elements using selectors. Selectors are related to XHTML elements. If you wanted to apply a style to something that wasn't a tag (let's say there was a certain sentence in your document that you wanted to be bold, but it wasn't anything other than text from a structural definition), this kind of situation is where Class selectors are of value. Another situation in which a Class selector comes in handy is when you want to style the appearance of XHTML tags differently. Earlier you learned how to redefine the appearance of an XHTML element (the h1 tag). Suppose you don't want to format every h1 tag the same way? This is a perfect place to consider creating a Class selector. Consider the following code:

```
<html>  
<head>  
<style type="text/css">  
<!--  
.mytext {  
    color: #FF6600;  
}  
-->  
</style>  
</head>  
<body>  
<p class="mytext">This is how you use class selectors.  
</p>  
<p>&nbsp;</p>  
</body>  
</html>
```

- Class selectors are written in this syntax: always beginning with a dot (.) and a unique name.
- The Class selector is applied to the XHTML element using the class attribute and the name you assigned to the class. In this example, we gave the class the name of “mytext”. Notice that the dot (.) is not included in the class attribute.

By creating a Class selector, you could now apply this formatting anywhere in your document independent of the XHTML element. It will apply this format only where it encounters this class, not across every instance of an XHTML element like previously shown.

Here is an example of Class selector being applied only to a specific portion of a paragraph:

```
<html>
<head>
<style type="text/css">
<!--
.mytext {color: #FF6600; font-family:verdana; font-weight:bold;}
-->
</style>
</head>

<body>
<p>This is how you use <span class="mytext"> class</span> selectors. </p>
<p>&nbsp;</p>
</body>
</html>
```

- The Class selector syntax is displayed on this line with the font-family and the font-weight properties applied.
- The tag is used to designate the area of text that is to be formatted using the Class selector. Notice that there is a closing tag to end the formatting.

Block-Level and Inline-Level Elements

Block-level elements acts like boxes that start at the margin of one line of text and end so that the content after the closing element is forced to start on a new line of text. The content of a block-level element can be, and typically is, several lines long. Basically, block level elements start and end a line of text. For example, the paragraph <p> element is a block-level element. It starts at one margin and anything that comes after the closing </p> element is forced to appear on a new line. Any formatting applied to a block-level element will affect everything within it.

```
<html>
<head>
<style type="text/css">
<!--
.mytext {
    color: #CC9900;
}
-->
</style>
</head>
<body>
<div class="mytext">
<p>Greenwaves, Inc </p>
<p>Company Information</p>
</div>
</body>
</html>
```

A <div> element is used to create a range, or invisible box if you will, around the two <p> elements. The body Class is attached to the <div> element, which causes both paragraphs to be formatted with the body Class. So, instead of applying the body Class to both paragraph elements, it was applied once to the <div> element. This results in less code, which is a good thing.

ID versus Class Selectors

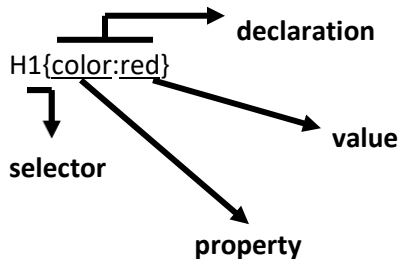
At this point, you might be wondering about the difference between a Class and ID selector. Functionally, it doesn't really matter because they both accomplish the same thing. However, there are a few differences.

Ids start with a # symbol instead of a dot (.). Also, it is considered an error to use the same ID selector more than once. In fact, Ids should be used only once within a given document, and they should always have a unique name different from other Ids in the document. This makes them great for absolute positioning, where they would only want to use them once. Classes are great for formatting text because they can be used multiple times on a web page. Because of this, you will find yourself probably using classes most of the time.

CASCADING STYLE SHEETS (CSS)

CSS style sheet

- made up of one or more rules.
- each style rule in a style sheet has 2 main parts:



SELECTOR -- indicates which elements will be formatted.

DECLARATION -- describes the formatting that should be executed.

EXAMPLE:

1. `h1{color:red;}`
2. `h1{color:red;`
`background:yellow;`
`font-family:georgia;`
`text-align:center;}`
`p{color:green;}`
3. `div#gaudi{color:red;}`
4. `div#gaudi p{color:red}`
6. `div.works{color:red}`

Breaking up a Page into Divisions

- allows you to apply styles to an entire chunk of your page at once.
- useful in designing layouts with CSS.

To break up a page into divisions:

1. Type `<div id="name">content</div>`, *where name uniquely identifies the division.*

OR

2. Type `<div class="name">content</div>`, *where name is the identifying name of the class that the division belongs to.*

FORMATTING WITH STYLES

1. Choosing a Font Family

Font-family property has a special characteristic: you can specify more than one font in the style rule.

Example:

1. `H1{font-family:"verdana"}`
2. `H1,P{font-family:"verdana", "tahoma"}`

2. Creating Italics

Italics are often used to set off quotations, emphasized text, foreign words, magazine names and much more.

Example:

1. `H1{font-style:italic}`
2. `H1{font-family:"verdana", "tahoma";`
`font-style:italic}`

3. Applying Bold Formatting

Bold formatting is the most common and effective way to make text stand out.

`font-weight:bold`

Example:

1. `H1{font-weight:bold}`
2. `H1{font-family:"verdana", "tahoma";`
`font-style:italic;`
`font-weight:bold}`

4. Setting the Font Size

font-size:xx-small	or	16px
x-small		18px
small		24px
medium		
large		
x-large		
xx-large		

Example:

1. P{font-size:medium}
2. H1{font-size:24px}

5. Setting the Text Color

Example:

1. P{color:blue;
font-size:small}

6. Changing the Text's Background

(background of the specified element)

Example:

1. P{background:red}
2. H1{font-family:"verdana", "tahoma";
font-style:italic;
font-weight:bold;
font-size:large;
color:blue
background:red}

7. Adding Indents

text-indent:length (18px, 24px....)

Example:

1. P{text-indent:24px}
2. P{text-indent:18px}

8. Aligning Text

text-align:left
right
center
justify

Changing the Text Case

text-transform:capitalize
uppercase
lowercase

Example:

H1{text-align:center;
text-transform:uppercase}

9. Decorating Text

text-decoration:underline
overline
line-through blink

Example:

H1,h2{text-
transform:uppercase;
text-decoration:blink}

10. To use a background image:

background-image:url

11. Setting the border

a. To define the border-style:

border-style:dotted
dashed
solid
double

b. To set the width of the border:

border-width:n (where n is the desired width)

c. To set the color of the border:

border-color:color (where color is the color name)

Example:

1. P{text-align:justify; text-indent:24px}
2. H1{text-align:center}