Chapter: One
HTML & CSS

Creating Hyperlink

- ☞ Hyperlink means it create a link. Whenever you click on the link, then you will reach where it is linked.
- ☞ Generally there are two types of link:
  1. **Internal link**
  2. **External link**
- ☞ You can use any types of text, image, symbols or any part of an image for hyperlink.
- ☞ Internal links only work inside the single page and external links connect more than one individual page.

The <A NAME> Tag:

- ☞ This tag is used to define marking for hyperlinks within a single page.
- ☞ We can define any text within double quote ("....") for book marking.

Note: Don't forget to define bookmark with hash (#) key for hyperlinks.

The <A HREF> Tag:

- ☞ This tag (Anchor Hyperlink REFerence) is used to create required internal and external hyperlinks. In case of internal link, you have to define Anchor Name at first.

Uses of LINK, ALINK, VLINK attribute of <BODY> tag

LINK:

- ☞ This attribute of <BODY> tag defines the color of Hyperlink, which have not yet been visited.
  E.g. <BODY link="NAVY">

ALINK:

- ☞ This attribute of <BODY> tag defines the color of Hyperlink, when you click on it.
  E.g. <BODY ALINK="RED">

VLINK:

- ☞ This attribute of <BODY> tag defines the color of Hyperlink, which you have already visited.
  E.g. <BODY VLINK="CYAN">

Email And Web Link

Internal link Using an Image

Direct linking with an image

Inserting Special Character

- ☞ Apart from all alphanumeric and numeric characters, you can add special characters and symbols.

Description	Entry Number	Result
Non-breaking space		^s
Less than	&#60	<
Greater than	&#62	>
Ampersand	&#38	&
Cent	&#162	C
Pound	&#163	£
Yen	&#165	¥
Euro	&#8364	€
Copyright	&#169	©
Registered trademark	&#174	®
Trademark	&#153	™
	&#177	+ -

How to creating a Form

- ☞ A form is an area that can contain form elements.
- ☞ When you create a form, you must set up a form before adding in run to the form.
- ☞ To setup a form, you need to specify two important information: ACTION, METHOD.

METHOD:

- ☞ Method property tells the form how to transfer the data to the form processor.
- ☞ The value for method can be "POST" or "GET". POST means that you are going to post the information and GET means that you are just going to get another page.

  Note: GET is almost never used.

ACTION:

- Action property tells what action the form should take when the user precess the SUBMIT Button.
- Action is the URL (Uniform Resource Locator) to which you are posting the information.
- This normally has a CGI, ASP, Mailto: etc.

The <INPUT> Tag:

- It is the most used form tag. The type of INPUT is specified with the TYPE attribute.
- This tag doesn't have its ending tag.
- The most commonly used INPUT TYPES are as below:
  - TEXT(NAME, VALUE, SIZE, MAXLENGTH)
  - RADIO(NAME, VALUE, CHECKED)
  - CHECKBOX(Name, Value)
  - IMAGE(SRC)
  - RESET(Value)
  - SUBMIT(value)
  - PASSWORD/HIDDEN(Name, Value, Size, Maxlength)

Where NAME specify the name of input tag and value specify that its value.

<TEXTAREA> Tag:

- This tag is used to specify the area of the text. This tag makes two dimensional text area, in which the viewer can type from a short sentence to many paragraph. This tag has its ending tag.
- This tag has following attributes:
  - NAME(Name of the tag)
  - VALUE(Editable information of the tag)
  - COLS(width of the textarea)
  - ROWS(Height of the textarea)

The <SELECT> Tag

- This tag is used to create a menu that offers visitors a list of option to choose from.
- It has ending tag called </SELECT>.
- It has following attributes and values:
  - NAME (It specify name of the tag)
  - MULTIPLE(If more than one option shown at a time)
  - SIZE(Number options you want to display at a time)

The <OPTION> Tag

- This is the mini-tag of <SELECT> tag. Normally, it comes with <SELECT> tag. This tag is used to define the options to choose from the dropdown list. This tag mainly has one attribute:
    - SELECTED
- This attribute specifies the default option that appears first in the non-multiple SELECT tag. It is not really needed since the first <OPTION> tag is assumed to be default.

How to Using Frames

- Generally, a browser will display a single HTML document in its window. Using the frames extensions to HTML, you can divide the main browser window into a number of sub windows, which is referred to as "Frame".
- Each frame contains a different HTML documents and can be linked to other frames.
- A framed document is like any other HTML document but it has a <FRAMESET> tag instead of <BODY> tag.

The <FRAMESET> Tag

- This is the main container of a framed document. We use only <FRAME> tag and <NOFRAME> tag between the <FRAMESET> tags.
- This tag has only two types of attributes:
    - Rows
    - Cols

Rows:

- The values of the ROWS attribute determine how the screen is to be divided up between different frames.
- There are three ways to express the value:
  <FRAMESET ROWS="250,*">
  Where, 250 pixel is the height of the first row and the other take what is left out.
  <FRAMESET ROWS="25%,75%"> OR <FRAMESET ROWS="25%,*">

Cols:

- This attribute has the same syntax as the rows attribute, but divides the screen up horizontally.
- Frameset tag can have either the rows or the cols attribute but not both. There can be nested framesets (combination of rows and cols attributes).

The <FRAME> Tag:

- This tag describes the individual frames within a <FRAMESET> tag.
- It is not a container, so there is no its ending tag.
- The <Frame> tag has following attributes and values:

SRC:

- ☞ This attributes specifies the source of the HTML file to place in the same frame. If you have no SRC attribute, then the browser will display the blank frame.

NAME:

- ☞ This attribute is used to give a frame name, so that it can be addressed by links in other documents.

SCROLLING:

- ☞ This attribute determines whether the frame should have scrollbars or not.
- ☞ This attribute has the following values:

  Scrolling = Yes
  Scrolling = No
  Scrolling = Auto (By default, but it holds vertical scrollbar)

NORESIZE:

- ☞ Normally, the user can change the sizes of individual frames by dragging them with the mouse, if NORESIZE attribute is used then the user can't change the sizes of the frames.

MARGINWITDTH:

- ☞ This attribute specifies the horizontal distance in pixel between the contents and left/right edges of the frame.

MARGINHEIGHT:

- ☞ This attribute specifies the vertical distance in pixels between the contents and top/bottom edges of the frame.

TARGET:

- ☞ This attributes specifies the target location of the URL of the HTML files.
  Eg. <FRAME NAME="contents" TARGET="main" SRC="content.html">

Banner	
Contents Hyperlink Here	Main Target of contents

The <NOFRAME>

- ☞ Some browser cannot handle or support frames. So, this tag is used to display message to inform use about it.
- ☞ It ends with </NOFRAME> tag. A browser that can handle frames will ignore anything contained inside these tags.

Cascade Style Sheet (CSS)

What is CSS?

CSS stands for Cascade Style Sheet is a simple design language intended to simplify the process of making web page presentable. CSS handles the look and feel part of web page. Using CSS, you can control the color of text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used as well as variety of other effects. CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantage of CSS

1. **CSS save time**

You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style sheet for each HTML element and apply it to as many web pages as you want.

2. **Page load faster**
   If you are using CSS, you do not need to write HTML tag attributes every time just write one CSS rules of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
3. **Easy maintenance**
   To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.
4. **Superior styles to HTML**
   CSS has a much wider array of attributes than HTML, So you can give a far better look to your HTML page in comparison to HTML attributes.
5. **Multiple Devices Compatibility**
   Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cellphones or for printing.
6. **Global web standards**
   Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

CSS Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts:

i. **Selector**

A selector is an HTML tag at which a style will be applied. This could be any tag like <H1> or<TABLE> etc.

ii. **Property**
A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border, etc.

iii. **Value**
Values are assigned to properties. For example, color property can have the value either red or #F1F1F1 etc.

Syntax:

Selector { property: value;}

Eg. you can define a table border as follws

Table{border:1px solid #C00;}

Where , table is a selector and border is a property and the given value 1px solid #C00 is the value of that property.

Type of Selector

you can define selectors in various simple ways base on your comfort.

1. **The Type Selectors**

The type selector is used to apply the specify type of HTML element or tag.

Eg

H1 {

Color:#CCC;

}

2. **The Universal Selector**
Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type. It is denoted by '*' operator.
Syntax:
    - * { propeeryt:values;}

```
Eg. *
{
        margin:0px;
        padding:0px;
}
```

Note: This rule renders the content of every element in our document margin and padding zero.

The Class Selector

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule. It is denoted by '. 'operator.

Syntax

. clsaa_nam { property:values;}

Eg. black { color:#000000;}

This class can also use in particular element

Eg. h1.black

{ color:#000000;}

Note: This rule renders the content in black for only <h1> elements with class attribute set to black.

The ID Selectors

You can define style rules based on the id attribute of the elements. All the elements having that id will be formatted according to the defined rule.

Syntax:

```
#id_name
{
    property:value;
}
Eg. #black
{
```

```
    Color:#000000;
}
```

This rule renders the content in black for every element with id attribute set to black in our document. You can also use id as well

```
H1#black
{
Color:#000000;
}
```

The Child Selector

Eg

```
Body>p{
color:#000000;
}
```

The attribute selector

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text.

Eg.

```
Input[type=”text”]
{
    Color:#000000;
}
```

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example

```
H1{
color:#36C;
font-weight:normal;
letter-spacing: .4em;
```

```
margin-bottom:1em;
text-transform:lowercase;
}
```

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example

Eg.

```
h1,h2,h3,p,div
{
color:#36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

Note:

You can combine the various class selectors together as shown below:

Eg.

```
#content, #footer, #supplement {
Position: absolute;
Left:510px;
Width:200px;
}
```

How to use CSS (CSS - Inclusion)

There are four ways to associate style with HTML document. Most commonly used methods are inline CSS and External CSS.

1. Internal CSS (Embedded CSS)
2. Inline CSS
3. External CSS
4. Imported CSS

Internal CSS (Embedded CSS)

Internal CSS can be use into an HTML document using the <style> element. This tag is placed inside the <head>……</head> tags.

Syntax:

<head>

<style type="text/css" media="…">

Style Rules

……

</style>

</head>

Attributes & Values of Style element

Attributes	Values	Description
Type	Text/css	Specifies the style sheet language as a content-type. This is required attribute.
Media	Screen	Specifies the devise, the document will be displayed on. Default value is all. This is an optional attribute.
	Tty	
	Tv	
	Projection	
	Handheld	
	Print	
	Braille	
	Aural	
	all	

Inline CSS

Inline style rule is only use with HTML element. This rule will be applied to that element only

Syntax:

<element style=”properties and values;”

Eg. <h1 style=”color:#36c;”>This is inline CSS</h1>

External CSS

The< link > element is used to include an external style sheet . It has following attributes and values

Rel=”stylesheet”

Type=”txt/css”

Href=”stylesheet.css”

Note: It is always included between <head></head> tag.

Eg

<head>

<link rel=”stylesheet” type=”txt/css” href=”stylesheet.css”>

</head>