

Unit - V - Advanced Website Designing

Frames: Frameset Definition, Frame Definition, Nested Framesets, Forms and Form Elements.

DHTML and Style Sheets: Defining Styles, elements of Styles, linking a style sheet to a HTML Document, Inline Styles, External Style Sheets, Internal Style Sheets & Multiple Style Sheets.

Q) Frames in HTML

Frames concept was introduced by Netscape communication. They allow us to divide a browser window into several independent parts. These are also called sub windows. We can insert web pages in these frames.

Frame is also called window. A group of frames in a single browser is called Frameset. The frameset is replaced by the HTML document **body** tag. We can divide browser window into sub forms using <Frameset> tag and closing tag is </frameset>. We can add the webpages (or) documents in frames using frame tag. No need to close this tag.

Syn: <Frameset

Cols="List" Rows="List"
Border="pixels"
Bordercolor="Color" or Hexadecimal value
Frameborder="Yes/No"
Framespacing ="Pixels">

</Frameset>

- ❖ **Cols:** It specifies a comma separated list of widths for division of window on columns.
 - **Ex:** cols="20%,80%" (or) cols="20% , *"
- ❖ **Rows:** It specifies the comma separated list of heights for division of window on rows.
 - **Ex:** rows = "20%,80%" (or) rows="20%,*"
- ❖ **Border:** It specifies the border width around frames.
- ❖ **Bordercolor:** It specifies the border color around frames. If omitted, the default color is gray.
- ❖ **Frameborder:** It specifies the frame border. If it sets "yes" the space between frames is filled with border color, if it sets "No" the space between frames is left blank(white color).
- ❖ **FrameSpacing:** It specifies the space between frames.

FRAME TAG:

This element defines a namable window region, known as a frame that can independently display its own content.

Syn:

```
<Frame Name="name" Marginheight="pixels"  
Marginwidth="pixels"  
Bordercolor="color" Frameborder="Yes" |  
"No" Noresize  
Scrolling ="Yes" | "No" | "Auto"  
Src = "URL">
```

- ❖ **Name:** It specifies the name of the frame for use with the target attribute of Aelement.
- ❖ **Marginheight:** It specifies the number of pixels to use as the top/bottom margins, within the frame.
- ❖ **Marginwidth:** It specifies the number of pixels to use as left/right margin
- ❖ **Bordercolor:** It specifies the border color around the frame.
- ❖ **Frameborder:** It sets the frame visible border
- ❖ **Noresize:** It prevents the user from resizing the frame

❖ **Scrolling:** It determines whether the frame has scroll bars.

❖ **Src:** It contains the URL of the contents to be displayed in the frame.

Example :

```
<Frameset rows= "20%, 80%">
<Frame src="controls.html" name="Controls" noresize scrolling="no">
<Frame src="contents.html" name="body">
<noframes>
<p> Error: No frame support </p>
</noframe>
</Frameset>
```

Iframe: The <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

```
<!DOCTYPE html>
<html>
<body>
<iframe src="https://www.w3schools.com">
  <p>Your browser does not support iframes.</p>
</iframe>
</body>
</html>
```

Q) NESTED FRAMES

Nesting of frames refers to a frame within another frame. A frame which contains other frames is called a container or outer frame. On the other hand, contained frames are called inner frames. Framesets may be nested at any level.

Example program:

```
<html>
<head>
<title> Nesting of frames</title>
</head>
<frameset rows="40%,*">
  <frameset cols="50%,50%">
    <frame name="f11">
    <frame name="f12">
  </frameset>
<frameset cols="33%,33%,*">
  <frame name="f21">
  <frame name="f22">
  <frame name="f23">
</frameset>
</frameset>
</html>
```

Q) Forms in HTML

The form can contain interface elements such as text fields, buttons, checkboxes, radio buttons, and selection lists that let users enter text and make choices, and then submit the form by clicking a “Submit” button. The form can be submitted the information to the web server.

Forms have many functions. They can be used for gathering information about a user, conducting a survey, selecting something of interest, placing an order online, submitting a query to a search engine and so on.

<form tag> :

In HTML, a form is defined with the <form> tag. The <form> element can contain one or more of the following elements:

- <input>
- <textarea>
- <button>
- <select>
- <option>

Syn: <Form action=“URL of Webpage” method=“GET” | “POST” name=“name”>
</Form>

Action: This attribute contains the URL of the server program that will process the contents of the form.

Name: It specifies a name for the form and was traditionally used by JavaScript or other client-side programming technologies.

Method: It indicates how form information should be transferred to the server using a particular HTTP method. A *Get* value in the attribute indicates that form data should be appended to the URL specified by the *action* attribute. This approach is quite simple but imposes a size limitation that is difficult to estimate.

A *Post* value in the attribute indicates that transfers the data of the form in the message body. It imposes no data size limitation.

< Input> Tag:

It specifies an input control for a form. The type of input is set by the *type* attribute and can be a variety of different types, including single-line text fields, password field, hidden, check box, radio button or push button.

Syn: <input name=“name” Size=“number”

Type = “text” | “password” | “Checkbox” | “radio” | “submit” | “Reset”

Value=“value” Checked Tabindex=“number”>

- ❖ **Name:** It specifies the name of the control. It is required, because data is sent as: name=“value”
- ❖ **Size:** It specifies the size of the text control.
- ❖ **Type:** It specifies the type of the control. Like Text, Checkbox, radio, submit, reset
- ❖ **Value:** It depends on the form control.
- ❖ **Checked:** It should be used only for the checkbox and radio of form controls.

<Textarea> tag:

Textarea is a multiline text input control. A text area can hold an unlimited number of characters.

The <textarea> tag is used within a form to define a textarea. The size of the textarea can be specified by the cols and rows attributes of <textarea> tag.

Syntax : <textarea name=“string” rows=“n” cols=“n”>.....</textarea>

<Button> tag:

The <button> tag is used for creating a button within forms. Buttons can be created using <input> tag. Inside a <button> element the context like text or images can be put.

Syntax : <button name="string" type="button | reset | submit ">.....</button>

<Select > and <Option> tags:

The <Select > tag is used to create a drop-down list.

The <Option> tags inside the <select > element define the available options in the list.

Syntax :

```
<select name="string">  
  <option value="string">....</option>  
  <option value="string">....</option>  
  <option value="string">....</option>  
</select>
```

Q) Defining styles

What is CSS?

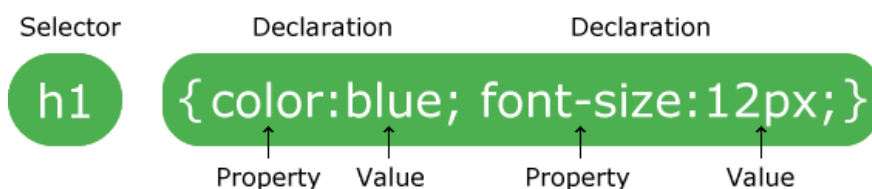
- ❖ CSS stands for Cascading Style Sheets
- ❖ CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- ❖ CSS saves a lot of work. It can control the layout of multiple web pages all at once
- ❖ External style sheets are stored in CSS files

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



- ❖ The selector points to the HTML element you want to style.
- ❖ The declaration block contains one or more declarations separated by semicolons.
- ❖ Each declaration includes a CSS property name and a value, separated by a colon.
- ❖ A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Advantages and disadvantages of CSS

Advantages:

- **Changes to the layout:** CSS makes it very easy to change the style of a document. The look and layout of a site can be changed by altering the CSS file. This makes CSS indispensable for large web sites.
- **File Size:** The most useful feature of CSS is that all of the styles and layout is removed from the html, so the html page size will be decreased so this can increase the downloading speed.
- **Accessibility:** Separating style from content makes very easy for visitors who prefer to view only the content of a web page, or to modify the content.
- **Consistency:** Layout and position of navigation can be completely consistent across a site. This was previously possible only using frames.

Disadvantages:

- **No Expressions:** There is currently no ability to specify property values as simple expressions (such as margin-left: 10% - 3em + 4px ;).
- **Lack of Variables:** CSS contains no variables. This makes it necessary to do a "replace-all" when one desires to change a fundamental constant, such as the color scheme or various heights and widths.
- **Inconsistent Browser Support:** Different browsers will render CSS layout differently as a result of browser bugs or lack of support for CSS features.
- **Vertical Control Limitation** - While horizontal placement of elements is generally easy to control, but vertical placement is very difficult to control, or impossible.
- **Control of Element Shapes** - CSS currently only offers rectangular shapes. Rounded corners or other shapes may require non-semantic markup.
- **No of Multiple Backgrounds per Element** - Highly graphical designs require several background images for every element, and CSS can support only one.

2. What are the different ways to add style sheets to Web pages?

A style is simply a set of formatting instructions that can be applied to a piece of text. We can apply the styles to the HTML pages with the following ways.

1. Inline CSS.
 2. Internal / Embedded CSS.
 3. External CSS
- **Inline CSS**

For Inline CSS every style content is in HTML elements. It is used for a limited section. Whenever our requirements are very small we can use inline CSS.

It will affect only single elements. In HTML we require that various HTML tag's views are different so then we use inline Cascading Style Sheets.

There are disadvantage of inline Cascading Style Sheets. It must be specified on every HTML tag. There is very much time consumed by that and it is not the best practice for a good programmer and the code will be quite large and very complex.

Inline CSS examples are given below:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;">This is a Blue Heading
</body>
</html>
```

- **Internal CSS**

In internal CSS the style of CSS is specified in the <head> section. This is internal CSS, it affects all the elements in the body section. Internal CSS is used in the condition when we want a style to be used in the complete HTML body. For that we can use style in the head tag.

This style performs an action in the entire HTML body.

```
<!DOCTYPE html>
<html>
<head>
<style>
  body
  {
  background-color: powderblue;
  }
  h1
  {
  color: blue;
  }
  p
  {
  color: red;
  }
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

- **External CSS**

In External CSS we create a .css file and use it in our HTML page as per our requirements. Generally external Cascading Style Sheets are used whenever we have many of HTML attributes and we can use them as required; there is no need to rewrite the CSS style again and again in a complete body of HTML that inherits the property of the CSS file.

Example:

styles.css

```
body
{
  background-color: powderblue;
}
h1
{
  color: blue;
}
p
{
  color: red;
}
```

HTML Program

```
<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

4. Properties and Values

There are number of style properties, including properties that control lists and positing floating elements like images.

(i) Font properties:

font-family: family name;

Fonts are identified by giving the name of a specific

font.font-style: normal | italic | oblique;

This property provides the style to the

font.font-weight: normal | bold | bolder | lighter;

The font-weight property selects the weight or darkness of the font. Values of the property range from 100 to 900 in increments of 100.

font-size: small | medium | large | smaller | larger;

This property is used to set the relative or physical size of the font used.

(ii) Backgrounds and color properties

Color:<value>

Background-color:<value> | transparent

Background-image: URL | none

- ❖ The color of any attribute can be changed. Value should be given as hexadecimal values.
- ❖ Backgrounds for the whole page or individual element can have their color set from the style sheet. Elements can also have transparent backgrounds.
- ❖ Instead of a color an image can be used, identified by its URL.

(iii) Text Properties

Text properties are used to effect the presentation, spacing and layout of text. The basic properties includes such as decoration, indentation, word spacing, letter spacing, spacing between lines, horizontal and vertical text alignment.

text-decoration: none | underline | overline | line-through | blink;

text-transformation: none | uppercase | lowercase | capitalize;

text-align: left | right | center | justify;

text-indentation: length | percentage;

line-height : length | percentage;

letter-spacing: length | percentage;

word-spacing: length | percentage;

Q) Including style sheets

We can include style sheets in our HTML page in two way (i) by linking (ii) by importing

(i) By linking

We can link our .css file in the head section of a HTML page using the following syntax

```
<link rel="style sheet" href="URL" type="text/css" media="screen">
```

The *href* is hyperlink to your style sheet

Rel tells to the browser what type of link you're using

Type tells to the browser what type of document you're including, the 'type' statement gives the relevant MIME type.

Finally it is useful, although not compulsory, to tell the browser how the document will be used. HTML Specifies a Variety of ways of using a document a document, including screen viewing, printing and as a presentations. Use the 'media' attribute to describe the type of use.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<style>
```

```
h1
```

```
{
```

```
color: orange;
```

```
}
```



```
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet, and internal style</p>
</body>
</html>
```

(ii) By importing

```
[<style type="text/css">]
<!--@import url(url); --> </style>
```

These lines are both needed if you intended to use more than one style sheet. The first sheet is included as if it were the only one; any further style sheets have to be imported.

Example:

```
<link rel="stylesheet" type="text/css" href="mystyle.css" media="screen">
<style type="text/css">
<!--@import url("http://www.smiggins.co./style.css"); --> </style>
```