Government Arts and Science College (Women), Sathankulam – 628704

Class : II B.Sc. (Computer Science)

Semester : IV

Subject : Web Technology

Topic : Unit V

Faculty : Mrs. A. Angeline Nancy Sophia M.Sc., M.Phil.,
Department of Computer Science

Dynamic HTML (DHTML)

DHTML is a technique of dynamically changing the content of an HTML code. It makes the web pages to react to the user actions or to the changes. It is a combination of three different technologies: client-side scripting, cascading style sheets and document object model.

The following list shows what comes under dynamic web page:

- Animation, audio and video files
- Page transition, automatic refresh and forwarding
- Changing text
- Visual change in response to user actions
- Automatically generated HTML
- Automatically generated data
- Forms processing

CASCADING STYLE SHEETS (CSS)

CSS are rules or styles for organizing the layout of an HTML document including its color, typefaces, margins, links and other formatting elements. It controls the style of a web document without compromising its structure. It also maintains its usability in multiple environments. It overrides the default settings of the browser for interpreting how tags should be displayed and how to use any HTML element indicated by an opening and closing tag to apply style attributes.

Style sheets contains rules, composed of selectors and declarations that define how style will be applied. The selector is the link between the HTML document and the style. There are two kinds of selectors, types and attributes.

Advantages of CSS

- CSS saves time: we have to specify the details once, CSS will automatically apply the specified styles whenever the element occurs.
- Pages load faster : less code means faster download times
- Easy maintenance: to change the style of an element, we have to make and edit in one place
- Superior styles to HTML : CSS has wider array of attributes than HTML

Disadvantage of CSS

- Browser compatibility : some style sheet features are supported and some are not by the main browsers

Coding CSS

A CSS declaration has two parts: a property and a value.

Syntax:

Selector { property1 : value1; property2 : value2}

```
<head>
<style type="text/CSS">
<!-
    tagname  { styleattibute:value;}
->
</style>
</head>
```

Example:

```
<html>
<head>
    <title> My first CSS page</title>
    <style>
        h1{color:red}
    </style>
</head
<body>
    <h1> My first CSS example</h1>
    <p> All the header elements of h1 will appear with font color as red </p>
<h1> Hai </h1>
</body>
</html>
```


Properties of Tags

Each HTML tag has some properties associated with it to determine how the browser will render something. For example a heading tag like h2 is associated with some text properties including font color, font family and font size. There is no attribute named color. But we can set this property by using in-line style.

```
<h2 style="color:red;"> This is a heading </h2>
```

Properties can be assigned multiple values.

Syntax: property-name: val1 val2 … valn;

A colon separates the name from the value and a space must separate each of the alternative values. Finally the value should end with a semicolon.

Property values

- Colors: We can specify a color by providing a hexadecimal number prefixed by the '#' character. The hexadecimal number indicates the amount of red, green and blue in the color.

  #808080

- Lengths: they are specified using a number followed by a 2-letter code that indicates the units.

  em – overall height of the current font

  px – pixels

  in – inches

  pt – point size

- URLs: specify the keyword url followed by the actual url inside parentheses.

  <body style="background-image url(http://www.cs.rpi.edu)">

Other Style Properties

Property	Values
font-family	[[<family-name> | <generic-family>],]* [<family-name> | <generic-family> default: UA-specific
font-style	normal | italic | oblique
font-variant	normal | small-caps
font-weight	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900
font-size	<absolute-size> | <relative-size> | <length> | <percentage> small, medium, large, larger default: medium
color	<color>
background-color	<color> | transparent
background-image	<url> | none
background-repeat	repeat | repeat-x | repeat-y | no-repeat
background-attachment	scroll | fixed
background-position	[<percentage> | <length>] {1,2} | [top |center | bottom] || [left | center | right]
text-decoration	none | [underline || overline || line-through || blink]
text-transform	none | capitalize | uppercase | lowercase

Adding styles

There are four ways to add styles to a web page.

- Embed style sheet within head tag
- Link to an external style sheet
- Import an external style sheet
- In-line style sheet

Embed Style Sheets

Global style sheet declarations, applicable to an entire document defined within the <style> and </style> tags and are usually placed in the head section.

```
<html>
<head>
<title>Title</title>
<style type=”text/css”>
h1{ color:maroon;}
</style>
</head>
<body>
        <h1> This is a heading </h1>
        <p> This is a paragraph</p>
</body>
</html>
```

Output

This is a heading

This is a paragraph.

External Style Sheet

Linked style sheet declarations use a single style sheet to define multiple pages. The <link> element can be used to include an external style sheet in our HTML document. An external style sheet is a separate text with .css extension.

```
<head>
<link type=”text/css” href =”style.css”>
```

```
</head>
```

Import an External Style Sheet

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

```
<head>
  @import "URL";
</head>
```

Here URL is the URL of the style sheet file having style rules.

Inline Style Sheet

We can use style attribute of any HTML element to define style rules.

<h2 style="color:red; font-size:7+pt"> This is a heading</h2>

```
<html>
   <head>
   </head>

   <body>
      <h1 style = "color:#36C;">
         This is inline CSS
      </h1>
   </body>
</html>
```

It will produce the following result −

This is inline CSS

Grouping

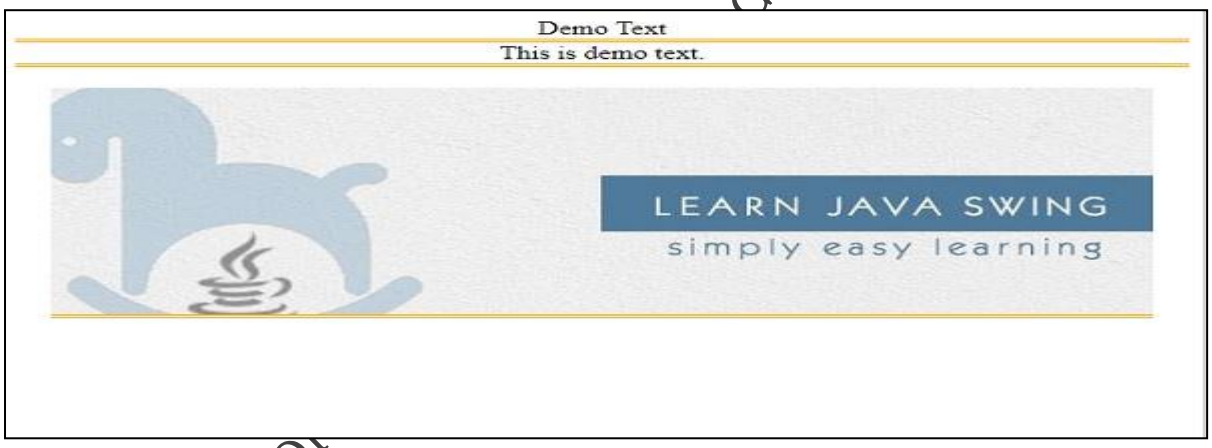
The size of the style sheets can be reduced using grouping. One can group selectors in comma-separated lists.

H1, H2, H3 { font-family: Helvetica}

```
<html>
<head>
<style>
article, p, img {
  display: block;
  margin: auto;
```

```
   text-align: center;
   border-bottom: double orange;
}
</style>
</head>
<body>
<article>Demo Text</article>
<p>This is demo text.</p>
<br/>
<img src="https://www.tutorialspoint.com/swing/images/swing.jpg">
</body>
</html>
```

This gives the following output −



Inheritance

HTML elements can inherit CSS styles from their parent elements. This is called *CSS inheritance*. Suppose there is an h1 element with an emphasized element inside as follows:

```
<h1>The headline <em>is</em> important! </h1>
```

If no color has been assigned to the *em* element the emphasized *is* will inherit the color of the parent element.

Class as Selector

Classes can be created to allow referring to a group of style settings by name. The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.

Example

```
<html>
<head>
<title> CSS Example</title>
<style type="text/css">
.bluemoon {
        color:blue;
        background:white;
}
.bluemoon EM {
        font-style:normal;
        font-weight=bold;
        color:white;
        background:black
}
</style>
</head>
<body>
<p class='bluemoon'> This is an <em>example</em> of 'bluemoon' style.</p>
</body>
</html>
```

Output

This is an example of 'bluemoon' style.

ID as Selector

We can use ID in a similar way as Class. The IDs are unique and they can only be applied once per document. The ID selector is used with # character.

```
<style type=”text/css”>
#foo {color:green}
</style>
<p ID=’foo’> This is a gree paragraph by virtue of the ID foo. </p>
```

Contextual Selector

Contextual selectors allow you to define the appearance for HTML tags in parent-child relationships with other HTML tags. Instead of setting all style properties, one can create defaults and then list the exceptions.

To give 'EM' elements within 'H1' a different color specify as:

H1 {color:blue}

EM {color:red}

All the emphasized sections within or outside H1 will turn red. But only EM elements within H1 should turn red we can specify as:

H1 EM {color:red}

The selector is now a search pattern on the stack of open elements and this type is referred to as contextual selector.

Contextual selectors can look for element types, Class attributes, ID attributes or combinations of these.

DIV P {font: small sans-serif}

.red H1 {color:red}

#w60z CODE {background: blue}

DIV .note H1 {font-size: large}

The first selector matches all P elements that have a DIV among the ancestors. The second selector matches all H1 elements that have an element of class red. The third selector matches all CODE elements that are descendants of the element with ID #w60z. The fourth selector matches all H1 elements that have a DIV element with class note.

Pseudo Classes and Pseudo-elements

Pseudo classes are like regular classes, but they are attached to the tag name with a colon instead of a period. Pseudo classes are available with anchor tags.

A:link {color:red}

A:visited {color:blue}

A:active {color:lime}

Paragraph elements also have pseudo elements like :first-line or :first-letter.

<style type="text/css">

P:first-line {font-variant: small-caps}

</style>

<p> The first line of an article in News Letter.
The output is
THE FIRST LINE OF AN
article in News Letter.

Positioning

CSS has position property. There are two ways of defining the position of elements. These are:

- Absolute positioning
- Relative positioning

Absolute positioning

An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

```
<html>
<head>
<title> Absolute Positioning </title>
</head>
<body>
<img src="C:\Users\USER\Downloads\100-red-roses-bunch-rs-2100.jpg" style="position:absolute; top:50px; left:25px; z-index:1">
<h2 style="position:absolute; top:120px; left:130px; z-index:3"> Positioning Element</h2>
</body>
</html>
```

Relative positioning

Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position.

```
<html>
<head>
```

```
<title> Relative positioning </title>
<style type=”text/css”>
.super {position:relative; top:-1ex}
.sub {position:relative; bottom:-1ex}
</style>
</head>
<body>
        <h1>Demonstrating the use of relative positioning </h1>
<p> The running text is here. <span class=”super”>superscript</span><br>
Again there is a running text in the normal flow <span class=”sub”>subscript</span>
</p>
</body>
</html>
```

Backgrounds

CSS also gives more over backgrounds than simple HTML attributes.

```
<html>
<head>
<title>Backgrounds</title>
<style type="text/css">
p {font-size:14pt}
body  {background-image: url("C:\Users\USER\Downloads\100-red-roses-bunch-rs-2100.jpg");
        background-position:bottom-right;
        background-repeat:no-repeat;
        background-attachment:fixed;
        position:relative;}
</style>
</head>
<body>
<h1 align=center>Demonstrating the use of background properties</h1>
<p align=center> The sample text is here.<br>
The sample text is here.<br>
The sample text is here.<br>
The sample text is here.<br>
```

```
The sample text is here.<br>
</p>
</body>
</html>
```

The background-image property is used to set the image at a proper place in the document. The background-position property can have values like top, bottom, left and right individually or in combination for vertical and horizontal positioning. The background-repeat property controls the tiling of the background image. The background-attachment property fixes the image in the position specified.