

Unit 6: Document Object Model

Document Object Model (DOM)

The Document Object Model (DOM) is a programming API (Application Program Interface) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model.

The HTML DOM is a standard object model and programming interface for HTML. It defines:

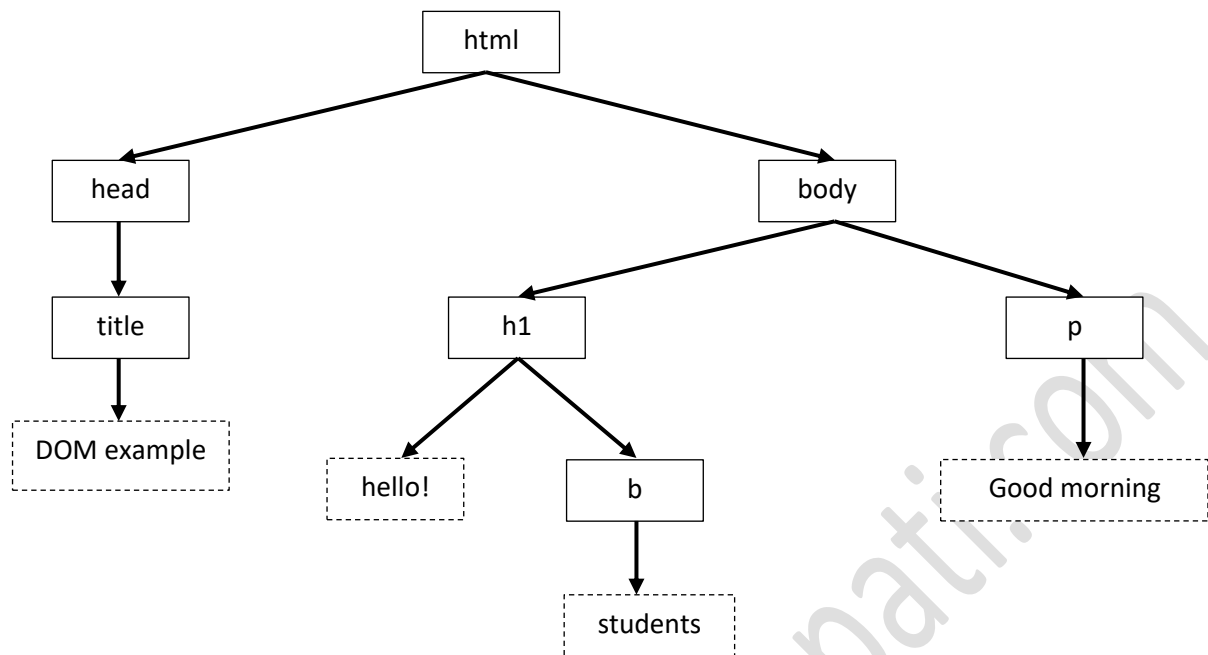
- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements. The object model itself closely resembles the structure of the documents it models.

Example:

```
<html>
<head>
  <title>DOM example</title>
</head>
<body>
  <h1>Hello! <b>Students</b></h1>
  <p>Good morning</p>
</body>
</html>
```

DOM structure:



Dynamic HTML (DHTML):

DHTML is a new and emerging technology that has evolved to meet the increasing demand for eye-catching and mind-catching web sites. DHTML combines HTML with cascading style sheets (CSS) and scripting language. HTML specifies a web page's element like table, frame, paragraph, bulleted list etc. Cascading style sheets can be used to determine an element's size, color, position and no. of other features. Scripting languages can be used to manipulate the web page's elements so that styles assigned to them can change in response to a user's input.

DHTML is a collection of technologies used together to create interactive and animated websites by using a combination of static markup language such as HTML, client-side scripting language like JavaScript, a presentation definition language such as CSS and Document Object Model (DOM).

Cascading Style Sheets:

Style sheets are used for adding styles (e.g. fonts, colors, spacing etc.) to web document. It maintains the standard and uniformity throughout a web sites and provide numerous attributes to create dynamic effects. With style sheets, text and image formatting properties can be predefined in a single list. The advantages of a style sheet includes the ability to make global changes to all documents for a single location.

For CSS `<style> ...</style>` tags are used. All CSS codes are written between these tags.

Syntax:

```
<style type="text/CSS">
```

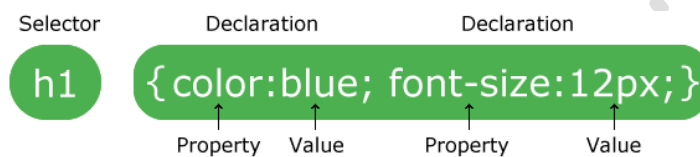
```

Tag/selectors {
Attribute: value;
Attribute: value;
}
...
</style>

```

CSS code 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.

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

Element Selector

The element selector selects elements based on the element name.

Example:

```

p {
    text-align: center;
    color: red;
}

```

Id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select

one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example:

```
#para1 {
    text-align: center;
    color: red;
}
```

Class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

Example:

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

Three ways of embedding CSS

1. Inline Style Sheets

Inline Style Sheets are included with HTML element i.e. they are placed inline with the element. To add inline CSS, we have to declare style attribute which can contain any CSS property.

Syntax:

```
<Tagname STYLE = "attribute1:value1; attribute2:value2;"> .... </Tagname>
```

Example

```
<p style="color: blue; text-align: left; font-size: 15pt">
```

This is an example of inline CSS embedding.

```
</p>
```

2. Embedded Style Sheets

Embedded Style Sheets are used to apply same appearance to all occurrence of a specific element. These are defined in <head> element by using the <style> element.

The <style> element must include type attribute. The value of type attribute specifies what type of syntax it includes when rendered by the browser.

Syntax

<head>

<title> </title>

<style type = "text/css">

Element/Selector {

Attribute1: value1;

Attribute2: value2;

}

</style>

</head>

Example

<style type="text/css">

p {color: green; text-align: left; font-size: 10pt;}

h1 {color: red; font-weight: bold;}

</style>

3. External Style Sheets

External Style Sheets are the separate .css files that contain the CSS rules. These files can be linked to any HTML documents using <link> tag with rel and href attribute.

Syntax:

<head>

<link rel= "stylesheet" type="text/css" href= "url of css file">

</head>

In order to create external css and link it to HTML document, follow the following steps:

First of all create a CSS file and define all CSS rules for several HTML elements. Let's name this file as external.css for example

Filename: external.css.

p{

Color: orange; text-align: left; font-size: 10pt;

}

h1{

```
Color: orange;    font-weight: bold;
}
```

Now create HTML document.

```
<html>
  <head>
    <title> External Style Sheets Demo </title>
    <link rel="stylesheet" type="text/css" href="external.css">
  </head>
  <body>
    <h1> External Style Sheets</h1>
    <p>External Style Sheets are the separate .css files that contain the CSS rules.</p>
  </body>
</html>
```

CSS attributes:

Background

Background-color:

We can set the background color for HTML elements. The background-color property specifies the background color of an element.

Values: colorname, #RRGGBB, rgb(R,G,B)

```
<h1 style="background-color: blue;>Hello world</h1>
```

or

```
h1 { background-color: blue; }
```



Hello World

Background-image

We can set the background image by using background-image property. By default, the image is repeated so it covers the entire element. HTML supports .jpg, .png, .gif image formats.

Values: url("filename")

```
body{  
background-image: url("photo.jpg");  
}
```

Background-repeat

By default, the background image is repeated horizontally and vertically.

Values: no-repeat, repeat-x, repeat-y.

```
body{  
background-image: url("photo.jpg");  
background-repeat: repeat-x;  
}
```

Text

Color:

The color property is used to set the color of the text. The color is specified by

- color name : like "red"
- Hex value: like "#ff0000"
- RGB value: like "rgb(255, 0, 0)"

```
h1{  
color: blue;  
}
```

Text-align:

It is used to set the horizontal alignment of a text.

Values: left, right, center, justified align.

```
h1{  
text-align: center;  
}
```

Text-decoration:

It is used to set or remove decorations from text. "text-decoration: none;" is used to remove underlines from links.

Values: overline, line-through, underline, none.

```
h1{  
text-decoration: underline;  
}
```

Text-transform:

It is used to specify uppercase and lowercase letters in text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Values: uppercase, lowercase, capitalize.

```
p{  
text-transform: uppercase;  
}
```

Text-indent:

It is used to specify the indentation of the first line of text. It takes pixel as value.

```
p{  
text-indent: 50px;  
}
```

Letter-spacing:

It is used to specify the space between the characters in a text. To decrease spacing, we use negative values.

```
p{  
letter-spacing: 3px;  
}
```

Line-height:

It is used to specify the space between lines. It takes float values or pixels.

```
p{  
line-height: 0.8;  
}
```

Text-direction:

It is used to change the text direction of an element.

Values: rtl, ltr.


```
p{
text-direction: rtl;
}
```

Word-spacing:

It is used to specify the space between the words in a text. It takes pixel values.

```
p{
word-spacing: 10px;
}
```

Text-shadow

It adds shadow to text. The value supplied contains horizontal shadow, vertical shadow, spread and color of shadow.

```
h1{
text-shadow: 3px 2px 5px black;
}
```

Font

The CSS font properties define the font family, boldness, size and style of a text. The difference between Serif and Sans-serif fonts is as below.



Font-family

It specifies the font to be used in html. It hold several font names as a fallback system. If the browser does not support the first font, it tries the next font and so on.

```
p{
font-family: "times new roman, times, serif";
}
```

Font-style

It specifies the style of the font. i.e. italic, normal etc.

Values: normal, italic, oblique (similar to italic)

```
p{  
font-style: italic;  
}
```

Font-size

It sets the size of the text. The default font size is 16px or 1em.

```
p{  
font-size: 40px;  
font-size: 1.6em;  
}
```

Font-weight

It specifies the weight of the font.

Values: normal, bold, strong

```
p{  
font-weight: bold;  
}
```

Font-variant

It specifies whether or not a text should be displayed in a small caps font. All the lowercase letters are converted to uppercase letters but they appear in a smaller font size than original uppercase letters.

```
p{  
font-variant: small-caps;  
}
```

Border

The CSS border properties allow you to specify the style, width, and color of an element's border.

I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

Border-style:

The border-style property specifies what kind of border to display.

Values: dotted, dashed, solid, double, hidden, none.

```
p{
border-style: dotted;
}
```

A dotted border.

A dashed border.

A solid border.

A double border.

Border-width:

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p{
border-width: 5px;
}
```

5px border-width

Border-color:

The border-color property is used to set the color of the four borders.

The color can be set by:


name - specify a color name, like "red"

Hex - specify a hex value, like "#ff0000"

RGB - specify a RGB value, like "rgb(255,0,0)"

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p{  
border-color: red;  
}
```

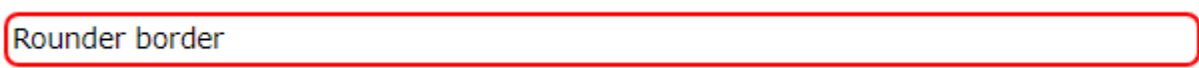


Red border

Border-radius:

The border-radius property is used to add rounded borders to an element.

```
p{  
border-radius: 5px;  
}
```



Rounded border

Margins

The CSS margin properties are used to create space around elements, outside of any defined borders. With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the margin for each side of an element:

margin-top

margin-right

margin-bottom

margin-left

```
p {
```

```
margin-top: 100px;  
margin-bottom: 100px;  
margin-right: 150px;  
margin-left: 80px;  
}
```

To shorten the code, it is possible to specify all the margin properties in one property.

```
p {  
    margin: 25px 50px 75px 100px;  
}
```

Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders. With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

```
div {  
    padding-top: 50px;  
    padding-right: 30px;  
    padding-bottom: 50px;  
    padding-left: 80px;  
}
```

To shorten the code, it is possible to specify all the padding properties in one property.

```
div {  
    padding: 25px 50px 75px 100px;  
}
```