

DHTML

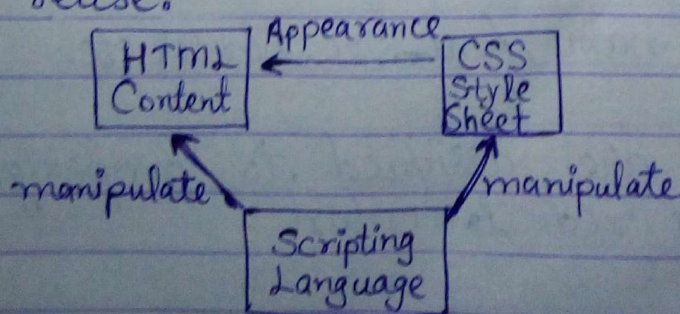
DHTML or Dynamic HTML is a term for a collection of technologies used together to create interactive and animated websites by using a combination of a static markup language (such as HTML), a client side scripting language (such as javascript), a presentation definition language (such as CSS) and the document object model (DOM).

Tuesday
28-2-17

DHTML combines HTML with CSS (Cascading style sheets) and scripting language.

→ Features of DHTML

- 1.) Simplest feature is making the page dynamic
- 2.) Can be used to create animations, games, applications providing new ways of navigating through websites.
- 3.) DHTML use low bandwidth effect which enhance web page functionality.
- 4.) Dynamic building of ~~the~~ a webpage is simple as no plug in is required.
- 5.) Facilitates the usage of events, methods and properties and code reuse.



Advantages of CSS

- 1.) CSS saves time.
- 2.) Pages load faster.
- 3.) Easy maintenance
- 4.) Superior Styles to HTML
- 5.) Multiple Device Compatibility
- 6.) Global web standard.
- 7.) Offline Browsing.
- 8.) Platform independence.

Syntax of CSS -

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in our document.

Style rule is made up of 3 parts:

- (1) Selector (2) Property (3) Value.

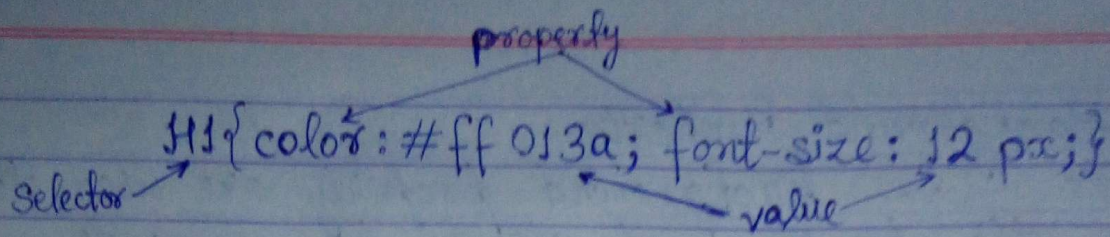
(1) Selector: A selector is a HTML tag at which a style will be applied.

e.g. → `<H1>` `<TABLE>` ``

(2) Property: A property is a type of attribute of a HTML tag.

e.g. → color, name, font-size, border etc.

(3) Value: Values are assigned to property.
e.g. → assigned red to color.



Multiple values of property:-

e.g. → TABLE { BORDER: 1px solid red; }

e.g. → <HTML>

<HEAD>

<STYLE>

H1 { color: red; font-size: 11px; }

P { color: blue; font-size: 8px; }

</STYLE>

</HEAD>

<BODY>

<H1> This is my heading </H1>

<H2> This is my subheading </H2>

<P> This is paragraph ----- </P>

</BODY>

</HTML>

CSS Background Property-

The CSS background property are used to define the background effects for elements.

- (1) Background-color
- (2) Background-image
- (3) Background-repeat
- (4) Background-position
- (5) Background-attachment

```
e.g. <!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE>
      H1,P{BACKGROUND-COLOR:#FF0000;}
    </STYLE>
  </HEAD>
  <H1>HELLO WORLD!!!</H1>
  <P>THIS IS A PARAGRAPH HAVING BACKGROUND
  _____ } (DIY)
  _____ }
  _____ }
  </BODY>
</HTML>
```

(Other background properties in body tag) →

<STYLE>

BODY { background-image: url ("pic.jpg"); }

BODY { background-repeat: no-repeat; }

(or) BODY { background-repeat: repeat-x; } (horizontally)

(or) BODY { background-repeat: repeat-y; } (vertically)

BODY { background-position: right top; } (right/bottom/left top or bottom)

or → : 150px 100px
(x-axis) (y-axis)

BODY { background-attachment: fixed; }

or → : scroll; }

BODY { margin-right: 200px }

</STYLE>

Short properties of CSS-

BODY { background-color: #ffff00 url ("pic.jpg") no-repeat right top; }

CSS Border properties-

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

Properties-

Border-style → what kind of border: dotted, dashed, solid, double, groove, ~~inset~~ inset, ridge, outset, none, hidden.

Border-width.

Border-color.

3D border

→ start from top to right to bottom to left.

```
<style>
```

```
{ border-style: solid dashed solid dashed;
  border-width: 5px 2px 1px 3px;
  border-color: red green blue yellow;
  border-radius: 8px; }
</style>
```

CSS margin properties-

CSS margin properties are used to generate space around elements. The margin property is set the size of the white space outside the border.

```
<style>
```

```
{ border: 1px solid red;
  margin-top: 100px;
  margin-right: 50px;
  margin-bottom: 150px;
  margin-bottom: 150px;
  margin-left: 100px;
  background-color: light blue; }
</style>
```

```
(or) { margin: 100px 50px 150px 100px; }
      (top) (right) (bottom) (left)
```

```
or { margin: auto; }
```

→ (browser automatically calculates margins.)

Monday
06-03-17

39

CSS Padding-

The CSS padding properties are used to generate space around content. The padding clears an area around the content of an element. (inside the border).

Properties-

Padding-top

Padding-right

Padding-bottom

Padding-left.

~~HTML~~

<HEAD>

<STYLE>

Pf BORDER: 2px solid green;

BACKGROUND-COLOR: yellow;

PADDING-TOP: 50px; (or) 50% etc)

PADDING-RIGHT: 75px;

PADDING-BOTTOM: 50px;

PADDING-LEFT: 70px;

}

</STYLE>

</HEAD>

~~HTML~~ Short hand properties of padding-

(or) PADDING: 50px 70px 50px 70px;

(or) PADDING: 50px;

(or) : 50 px; (T, B) (L, R)
 (or) : 50 px 70 px; (L) (2)
 (or) : 50 px 70 px 50 px;
 (or) : 50 px 70 px 50 px 70 px;
 (T) (L) (B) (2)

<div> tag - It defines a division or a section in a HTML document. The <div> tag is used to group the block elements to format them with CSS. <DIV> tag is just like a container unit which is used to encapsulate other HTML elements and divides the HTML documents in sections. <DIV> tag does not provide any visual change on the block but this has more meaning when it is used ~~used~~ with CSS. It leaves a blank line b/w divisions.

```
<body>
  <div>
    <h1> This is heading one </h1>
    <h2> This is heading two </h2>
    <p> - - - - - </p>
  </div>
</body>
```

CSS Height and Width -

```
<style>
  div { height: 200 px;
        width: 300 %;
        background-color: red;
      }
</style>
```


CSS Box Model-

All HTML elements can be considered as boxes. In CSS, the term Box Model is used when talking about design and layout. The CSS Box Model is essentially a box that wraps around every HTML elements. It consists of margins, borders, padding and the actual content.

Contents-

The content of the box where text and images appeared.

Padding-

Clears an area around the content.

Padding is transparent.

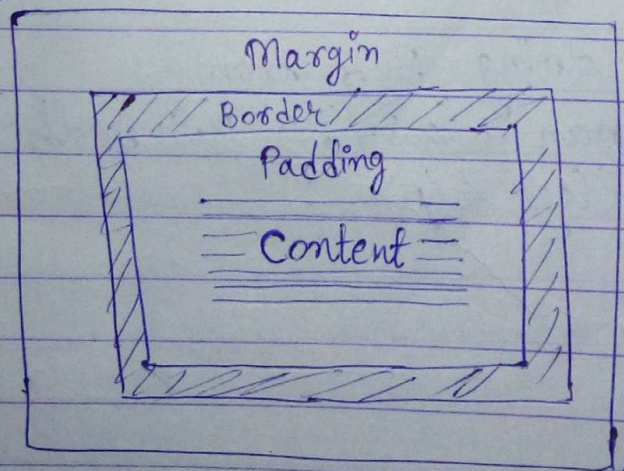
Border-

A Border that goes around padding and content.

Margin-

Clears an area outside the border.

The margin is transparent.



CSS Outline -

An outline is a line that is drawn around elements outside the border to make the elements stand out.

Properties -

1. Outline - style
2. Outline - color
3. Outline - width

Tues

7-3-17

 Tag - (inline element)

<HTML>

<HEAD>

<STYLE>

{color: red;}

.big {color: blue; text-decoration: underline; font-weight: bold;
font-size: 15px;}

</STYLE>

</HEAD>

<BODY>

<p>

I am doing BCA from

Subodh
college </p>

CSS Text properties

- (1) `color: red;`
`: #ff0000`
`: rgb(255, 0, 0)`
- (2) `text-align: left;`
`: right;`
`: center;`
`: justify;`
- (3) `text-decoration: underline;`
`: overline;`
`: line-through;`
- (4) `text-transform: uppercase;`
`: lowercase;`
`: capitalize;`
- (5) `text-indent: 50 px;`
- (6) `letter-spacing: 3 px;`
`: -3 px;`
- (7) `line-height: 0.8 px;`
- (8) `word-spacing: 5 px;`
`: -5 px;`
- (9) `direction: rtl;`
- (10) `text-shadow: 3 px 2 px red;`

CSS Font properties-

- (1) font-family: Times New Roman;
- (2) font-style: Normal;
: Italic;
: Oblique
- (3) font-size: 15 px;
: 100 %;
: 2.5 em;
: 10 pt
- (4) font-weight: bold;
: lighter;
: normal;
- (5) font-variant: small-caps

CSS Links -

Color
font-family
background
text-decoration

} Properties.

The four link states are

1. a: link - a normal, unvisited link
2. a: visited - a link the user has visited
3. a: hover - a link when the user moves ~~mouse~~ mouse over it.
4. a: active - a link the moment it is clicked.

Friday
10-03-17

eg.

```

<HTML>
  <HEAD>
    <STYLE>
      a: link { color: red;
                text-decoration: none;
                background-color: yellow;
              }
      a: visited { color: blue;
                   text-decoration: underline;
                   background-color: pink;
                 }
      a: hover {
                _____
                _____
              }
      a: active {
                _____
                _____
              }
    </STYLE>
  </HEAD>

```

```

<HTML>
  <BODY>
    <a href = "first.html" > Click here </a>
  </BODY>
</HTML>

```

- List -
-
 -
 -

e.g. → `<STYLE>`
`ul.a { list-item-type: circle; }`
`ul.b { list-item-type: square; }`
`ol.c { list-item-type: upper-roman; }`
`ol.d { list-item-type: lower-alpha; }`

`</STYLE>`

`<HEAD>`

`<BODY>`

`<ul class="a">`

` Pen `

o Pen

` Pencil `

o Pencil

` Eraser `

o Eraser

``

`<ul class="a">`

``

`<ol class="c">`

``

`<ol class="d">`

``

`</BODY>`

```
e.g. -> <STYLE>
ul a { list-style-image: url("pic.gif"); }
ul b { list-style-position: inside; }
                                     -> or outside.
```

```
e.g. -> ul { list-style: square inside url("pic.gif"); }
ul { background-color: blue; }
ul { color: yellow;
padding: 20 px;
margin-left: 30 px; }
```

```
ul li { background-color: red;
color: yellow;
padding: 20 px;
margin-left: 35 px; }
```

Saturday
11-03-17

CSS Table -

properties -

1.) Borders

```
table { border-collapse: collapse; }
width: 100%;
height: 50 px;
text-align: left;
th, td { padding: 15 px;
border-bottom: 7px solid
# ddbbddd; }
tr: hover { background: # f5f5f5; }
```

```
<Style>
th { text-align: left;
color: red; }
td { text-align: center;
color: blue;
vertical-align: top; }
</Style>
```

```
tr:nth-child(even){background-color:#f2f2f2;}
```

CSS Positioning

1. **Relative Positioning** - Relative positioning changes the position of the HTML element relative to where it normally appears. Setting the top, bottom, right and left property of relatively position element will cause it to be adjusted away from its normal position.

Absolute Position - An element with position: absolute; is position: relative to the nearest position ancestor.

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

```
<style>
```

```
div.relative { position: relative;
                width: 400 px;
                height: 200 px;
                border: 3 px solid red;
            }
```

```
div.absolute { position: absolute;
                top: 80 px;
                right: 0;
                width: 200 px;
                height: 100 px;
                border: 3 px solid green;
            }
```

```
</style></head>
```



```
<body>
  <div class="relative"> - - - -
    <div class="absolute"> - - -
  </div>
</div> </body> </html>
```

Overlapping Elements -

```
<style>
  img { position: absolute;
        left: 50px;
        top: 50px;
        z-index: -1;
      }
</style>
```

```
<body>
  
  <p> - - - -
  </p>
</body>
```

Overflow -

e.g. >

```
<html>
  <head>
    <style>
      div { background-color: lightblue;
            width: 200 px;
            height: 50 px;
            border: 1 px dotted red;
            overflow: scroll;
              : visible;
              : hidden;
            }
    </style>
  </head>
  <body>
    <div> - - - - -
    </div>
  </body>
</html>
```

Pseudo class-

A Pseudo class is used to define a special state of an element. For example- :hover will apply a style when the user hovers over the element specified by the selector.

```
a:link{-----}  
a:hover{-----}
```

CSS layer tag-

Pieces of HTML that are placed on the top of regular page with pixel precision to create a layer all we need to do is assign the position attribute to our style. The position can be either absolute or relative. The position itself is defined with the top and left properties. Finally which layer is on the top is defined with Z-index attribute.

```
<html>  
<head>  
</head>  
<body>  
  <div style="background-color:red;  
    width:300 px;  
    height:100 px;  
    position:relative;  
    top:10 px;  
    left:80 px;  
    z-index:2;">  
</div>
```

```

<div style="background-color: yellow;
width: 300 px;
height: 100 px;
position: relative;
top: 10 px;
left: 80 px;
z-index: 1;">

```

```

</div>

```

```

<div style="background-color: green;
width: 300 px;
height: 100 px;
position: relative;
top: 10 px;
left: 80 px;
z-index: 3;">

```

```

</div>

```

```

</body>

```

```

</html>

```

The CSS layers refer to apply the z-index property to elements that overlap with each other.

CSS Filter Effects-

CSS filters are powerful tools that authors can use to achieve varying visual effects.

```

filter: brightness(0.4);
contrast(200%);
greyscale(50%);
hue rotate(90 deg);
invert(75%);
opacity(25%);
saturate(30%);
sepia(60%);

```

```

<html>
  <head>
    <style>
      img { -webkit-filter: blur(20px); }
    </style>
  </head>
  <body>
    <p>Apply Blur on image</p>
    
  </body>
</html>

```

Iframes -

An iframe is used to display a webpage within a webpage. The HTML inline frame `<iframe>` element represents a nested browsing context, effectively embed another HTML page into the current page. The `<iframe>` tag defines a rectangular region within the document in which the browser can display a separate document including scroll bars and borders.

- * -webkit-filter: -It is for chrome, safari, opera.
- * -moz-filter: -It is for mozilla firefox.

Syntax:-

```
<iframe src="URL"></iframe>
```

eg. → `<iframe src="demo.htm"></iframe>`

Attributes of `<iframe>` tag

1. `src` - The `src` attribute specifies the url (the web address) of the inline frame page.
2. `name` - This attribute allows us to give a name to a frame.
3. `height` - It specifies the height of window.
4. `width` - It specifies the width of window.
5. remove the border -
e.g. - `style="border:none;"`
6. `border: 5px solid red;`
7. Target for a link - An `iframe` can be used as the target frame for a link. The `target` attribute of the link must refer to the name attribute of the link must refer to the name attribute of the `iframe`.

```
<iframe src="demo.htm" name="iframe-a"
height="200" width="250"
```

```
<p><a href="books.htm" target="firstframe">
click here</a></p>
```