**CSS**

Cascading Style Sheet is used to control the style of a web document in a simple and easy way. It is used to control the style and layout of multiple web pages all at once.

Three ways of inserting a style sheet:

* **External: Create a separate CSS file and embed into html file within <head> tag**

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

* **Internal: You have to create CSS within HTML document under <head> tag**

<style type=”text/css”>**h2 { color: red; }**</style>

* **Inline: You have to specify your style where you have used that element under <body> tag**

<h2 style=”color: red”>Welcome</h2>

**CSS ID AND CLASS**

The **CLASS selector(.)** is used to specify a style for a group of elements.

<h2 class =”firstid”>Welcome </h2>

<h2 class =” firstid”>World </h2>

Now, if we want to create an external css for both tags, then:

.firstid{

color: red;

}

**ID selector (#)** is used to specify a style for single, unique element.

<h2 id =”firstid”>Welcome </h2>

<h2 id =”secondid”>World </h2>

Now, if we want to create an external css for both tags, then:

#firstid{

color: red;

}

#secondid{

color: green;

}

**CSS BACKGROUND PROPERTY**

It will define the background property of an element.

<div class=”backImg”></div>

<p>Hello World</p>

<div class=”backImg”></div>

**External CSS**

**.** backImg{

width: 100%;

height: 100vh; **//vh represents view height which can you can see onto page**

background-color: green;

background-image: url(‘logo.png’);

background-repeat: no-repeat;  **//so that anything cannot be repeated horizontally or vertically (if you want your data to be repeated horizontally then use, repeat-x and if you want your data to be repeated vertically, then use, repeat-y)**

background-size: 100% 100%;  **//if you want to show your image to be shown at exactly in between of your web page and cover whole page. (This property is a part of CSS3)**

background-attachment: fixed; **//this property is used to control the scrolling of an image (parallex website in which we have image over another image).**

}

**NOTE: You always have to use some extra style in css file while creating your websites, otherwise some portion of your page(left, right, top and bottom) will not be covered.**

**\*{**

**margin: 0;**

**padding: 0;**

**}**

**FONTS CSS**

**font-style** is used to make a font italic or oblique

**font-size** is used to set the size of the text (px, em)

**font-weight** specifies the weight of a font (bold)

**font-family** is used to change the face of a font. To use this, we goto google fonts and choose our font family like calibri which we want to embed with our text. We have **CSS** also for those font family, just add that **CSS** into our html page.

<link href=”<https://fonts.googleapis.com/css?family=Notable>” rel=”stylesheet”>

**Example:**

<p id =”paraid”>Welcome</p>

**External CSS**

#paraid{

font-family: ‘Notable’, sans-serif; **//if Notable isnot found then use sans-serif font**

font-size: 2em;  **//1em = 16px**

font-style: italic;

font-weight: bold; **//you can also give value from 100 to 900 or use ‘lighter’**

**}**

**TEXT CSS**

**color** property is used to set the color of a text.

**direction** property is used to set the direction. Possible values are ltr and rtl.

**letter-spacing** is used to add or subtract space between the letters that make up a word.

**word-spacing** is used to add or subtract space between the word of a sentences.

**text-indent** is used to indent the text of a paragraph.

**text-align** is used to align the text of a document.

**Example:**

<p id =”textid”>Welcome</p>

**External CSS**

**#textid{**

font-family: ‘Notable’, sans-serif;

color: red;

letter-spacing: 2px;

word-spacing: 2px;

direction: ltr/rtl; **//rtl refers to right to left and ltr refers to left to right**

text-indent: 50px; **//it will create some space like first line of paragraph**

text-align: left/right/center;

text-transform: uppercase/lowercase; **//to convert your text into uppercase**

**}**

**LINKS CSS**

The **:link** signifies unvisited hyperlinks.

The **:visited** signifies visited hyperlinks.

The **:hover** signifies an element that currently has the user’s mouse pointer hovering over it.

The **:active** signifies an element on which the user is currently clicking.

**Example:**

<a href=”demo.php” target=”\_blank”>This is a link</a>

**External CSS**

a:link{

color: red;

}

a:visited{

color: green;

}

a:hover{

color: yellow;

}

a:active{

color: blue;

}

**NOTE: a:hover must come after a:link and a:visited in the CSS definition. a:active must come after a:hover in the CSS definition.**

**BORDER IN CSS**

Border properties allows you to specify how the border of the box representing an element should look.

**border-color** specifies the color of border.

**border-style** specifies whether a border should be solid, dashed line, double line, dotted or one of the other possible values.

**border-width** specifies the width of border.

**border-radius** is included in CSS3

**Example:**

<div class=”bordercss”></div>

**External CSS**

.bordercss{

width: 500px;

height: 300px;

background-color: red;

border-width: 5px;

border-color: black;

border-style: solid;

border: 5px solid black; **//this can be used instead of three properties of width, color and style**

border-radius: 50px; **//property of CSS3 which will round its shape from corners**

}

**LISTS CSS**

**list-style-type** allows you to control the shape or appearances of the marker.

**list-style-position**

**Example:**

|  |  |
| --- | --- |
| <ul>  <li>Home</li>  <li>Home</li>  <li>Home</li>  </ul> | <ol>  <li>Home</li>  <li>Home</li>  <li>Home</li>  </ol> |

**External CSS**

ul{

list-style-type: circle/square; **//more options are there**

}

ol{

list-style-type: lower-alpha/upper-alpha/upper-roman/lower-roman; **//more options**

}

**BOX MODEL CSS**

**Content:** the content of box, where text and images appear

**Padding:** clears an area **around** the content. It is transparent (it increase or decrease the area between content and border)

**Border:** a border that goes around the padding and content

**Margin:** clears an area **outside** the border. It is transparent

**Example:**

<div class=”boxmodel”>

<label>Username</label>

<input type=”text” name=””>

</div>

**External CSS**

.boxmodel{

width: 400px;

height: 400px;

background-color: red;

}

input{

padding-left/padding-right/padding-top/padding-bottom: 20px;

padding: 5px; **//margin of 5px from all sides(Top, Right, Bottom and Left)**

padding: 5px 5px 5px 5px; //**Top Right Bottom Left**

margin-left/margin-right/margin-top/margin-bottom: 20px;

margin: 5px; **//margin of 5px from all sides(Top, Right, Bottom and Left)**

margin: 5px 5px 5px 5px; //**Top Right Bottom Left**

}

**SCROLLBARS (OVERFLOW)**

**Visible:** Allows the content to overflow the borders of its containing elements.

**Hidden:** The content of nested element is simply cut off at the border of the containing element and no scrollbars is visible.

**Scroll:** The size of the containing element doesnot change, but the scrollbars are added to allow the user to scroll to see the content.

**Auto:** The purpose is same as Scroll, but the scrollbar will be shown only if the content does overflow.

<p id =”textid”>Welcome</p> **//suppose we have more text to show than area**

**External CSS**

p{

width: 100px;

height: 100px;

border: 1px solid red;

overflow/overflow-x/overflow-y: hidden/auto/scroll; **//x and y represents x and y axis**

}

**POSITIONING CSS**

A **relative** positioned element is positioned relative to its normal position.

An **absolute** position element is positioned relative to the first parent element that has a position other than static.

<div class = “father”>

<div class = “pos1”>Aayush</div>

<div class = “pos2”>Kartik</div>

<div class = “pos3”>Agrim</div>

<div class = “pos4”>Shivam</div>

</div>

**External CSS**

.father{

width: 600px;

height: 300px;

background-color: red;

position: relative; **//as it is position where it is**

}

.child1{

width: 100px;

height: 60px;

background-color: green;

position: absolute; **// positioned relative to the first parent element that has a position other than static**

**left: 0; //with position, we have 4 properties i.e. left, right, top and bottom where we have to place our content in parent element**

}

.child2{

width: 100px;

height: 60px;

background-color: yellow;

position: absolute;

**right: 0;**

}

.child3{

width: 100px;

height: 60px;

background-color: orange;

position: absolute;

**bottom: 0;**

**right: 0;**

}

.child4{

width: 100px;

height: 60px;

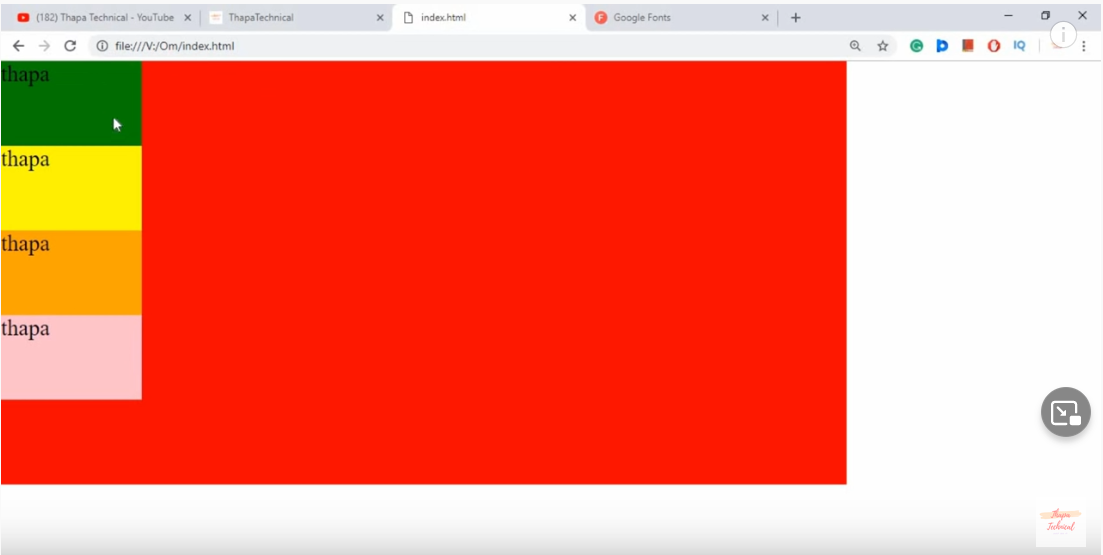
background-color: pink;

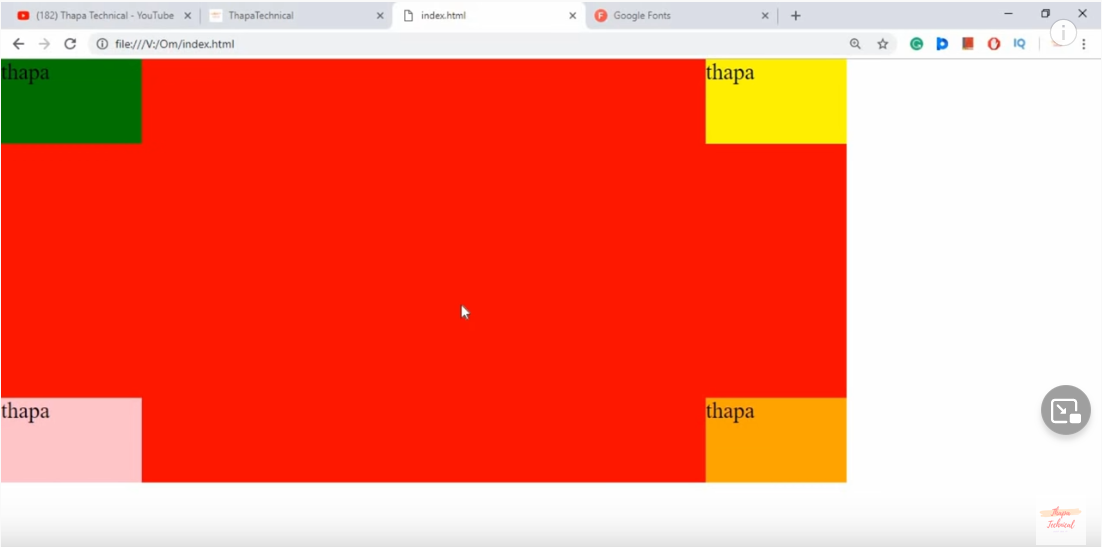
position: absolute;

**bottom: 0;**

**left: 0;**

}





**Z-INDEX**

z-index property specifies the stack order of an element.

An element with greater stack order is always infront of an element with lower stack order.

z-index only works on positioned element(position: relative, position: absolute, position: fixed)

**External CSS**

**z-index: 1/2/3…;**

**CSS FLOAT**

**left**: the element floats to the left of its container

**right**: the element floats to the right of its container

<div class = “header”>

<div class = “logo”>Aayush </div>

<div class = “menu”>

<ul>

<li>Home</li><li>Services</li><li>Contact</li>

</ul>

</div>

</div>

**External CSS**

.header{

width: 100%;

height: 80px;

background-color: black;

}

.logo{

width: 30%;

line-height: 80px; **//this will display your content in middle from top and bottom**

text-align: center;

background-color: orange;

float: left; **//this property will be used when we want to place one div section in left or right of another div (by default, one div is displayed in bottom of another)**

}

.menu{

width: 70%;

height: 80px;

background-color: green;

float: right;  **//this we have to do otherwise this menu will also take its width from left hand side not after logo div**

}

**CSS DISPLAY**

The display property is most important property for controlling layout.

Every HTML element has a default display value depending on what type of element is.

The default display value for most element is block or inline.

<div class = “menu”>

<ul>

<li>Home</li><li>Services</li><li>Contact</li>

</ul>

</div>

**External CSS**

ul{

list-style: none; **//this will remove list type like circle or square**

text-align: right;

}

li{

display: inline/inline-block; **//this will display your menu list one after another in right (we have display: block; also which will show list elements vertically one below another)**

padding: 0 10px;

}

**CURSORS**

The cursor property of CSS allows you to specify the type of cursor that should be displayed to the user.

