Introduction to cascading style sheet

css is a simple design language to make web pages more presentable

Cascading : Represents inherit.

Style : Represents attribute or property

Sheets: That represent .css file.

Features of css :

css has the following list of unic features

1] Easy manage
2] Global change
3] save lot of time
4] Easy maintainance
5] Inline style sheets
6] Internal style sheets
7] External style sheets
8] performance
9] superior styles
10] Multidevice compatibility
11] Global webstandards etc.

Why css?

css required because of the following reasons.

1] It is collection of style sheet mechanism
2] It is used to add more affects.
3] we can embed css in html, java script, PHP, ASP etc.
4] The extension of css file is .css.
5] use same style on multiple pages.
6] Reduce download size etc.

css versions History

css has the following list of versions

1] css 1.0 [1996]
2] css 2.0 [1998]
3] css 3.0 [2008]
4] css 4.0 [2014]

CSS Structure

→ As per w3C standard css has the following detailed structure.

```
<html>
<head>
        <style type="text/css">
        {
          _ _ _  _ _ _ _
          _ __  _____
        }
         </style>
<head>
 <body>
      _ _ _ _ _ _ _
      _  _ _ _ _
 </body>
 </html>
```

ex1:

```
<html>
<head>
 <style type="text/css">
 div
 {
  color: blue;
  background-color : #FFFFO;
  text-decoration: underline;
  font-family: Arial unicode ms;
 }
 </style>
 </head>
<body>
<div> welcome to cascading style sheet.....! </div>
<div> welcome to cascading style sheet.....! </div>
</body>
</html>
```

CSS comments

→ comments are non executable statements / non ignore statements. using these comment notations we can declare customized statements or user defined statements in the web environment or in the style sheet

→ In css every comment begins with "/*" and ends with "*/".

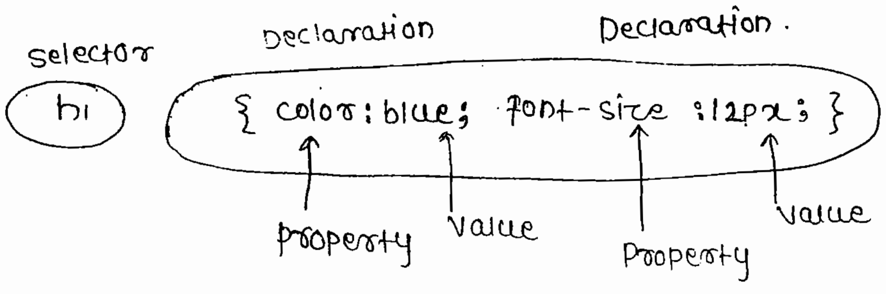
example :

```
<html>
<head>
<style type= "text/css">
  div
  {
   /* color: blue;
      background-color: pink; */
  }
 </style>
</head>
<body>
<div> welcome in cascading style sheet </div>
<div> welcome in cascading style sheet </div>
<l body>
 </html>
```

CSS Basic syntax :

It is made up of three parts

1] selector
2] property
3] value



syntax: made up of three parts
Selector { property: value }

→ The selector is normally the html tag

```
<head>
<style type="text/css">
 div  <——— selector
 {
  color: blue;   property
   }             value
  </style>
  </head>

 <body>
  <div> welcome to cascading style sheet </div>
  <Ibody>
```

Types of stylesheets

In css stylesheets are classified into the following three types.

1] Inline style sheets.
2] Internal / embeded style sheets.
3] External style sheets.

1] Inline style sheets : If we specify styles inside the tag in the body part. These styles able to apply only ~~that~~ for that perticular line

ex :

```
 <head>
 <title style='color:red'>   } style is not applicable in head section
  Hello                      } only applicable in body section.
 </title>
 </head>
 <body>
 <s style='color:red'> It is removed text from the web page </s>
<p style='color:blue; background-color: lightgreen'> It is removed
 text from the web page... </p>
<b style='text-decoration:underline'> It is removed text from the
 web page </b>
  </body>
```

2] Internal style sheets : ~~It~~ These are popularly known as embeded style sheets. If we specify the style in our html file itself then they are called as internal styles. These styles cannot be used in other files. but we should implement again and again in the same file.

Syntax :
```
<html>
<head>
    <style type=" text/css">
</style>
</head>
<body>
</body>
</html>
```

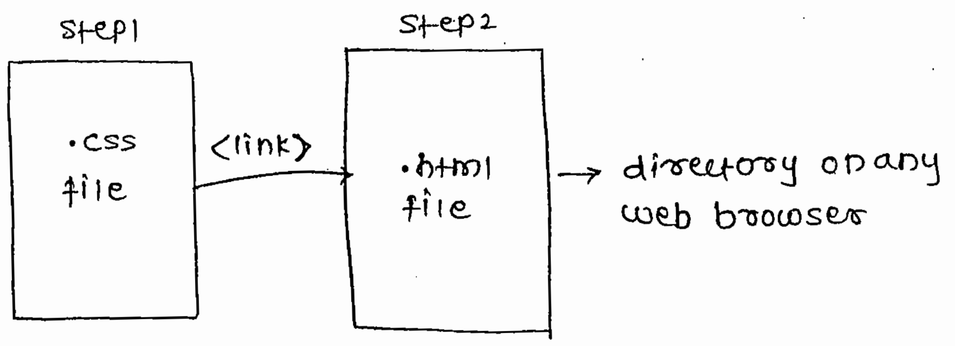
ex:
```
<head>
<style type= "text/css">
 p
 {
   Color: #FF00FF ;
   font-size: 20px;
   font-weight: bolder;
 }
</style>
</head>
<body>
 <p> welcome to Internal style sheets.... </p>
 <p> welcome to Internal style sheets.... </p>
 </body>
```

3] External style sheets:

If we declare the styles outside our html file then they are called external styles. These styles can be reusable on more than one file. These file extension is .css.

Syntax :
```
<head>
<link rel= "stylesheet" href="#" type= "text/css">
</head>
```

20-8-15 How to implement external style sheets.



Step1: creating .css file

```
div
{
  color: #FF00FF; font-size: 25px;
   font-family: Arial unicode MS;
  text-decoration: underline;
  font-weight: bold;
}
```

save with one.css extension @ any location.

step2: creating required html file

ex:

```
<head>
<link rel= 'stylesheet' href="one.css">
</head>
 <body>
 <div> welcome to external style sheets ....</div>
  </body>
  </html>
```

→ save with .html extension and run on any major web Browser.

ex2: with file path

```
<head>
  <link rel ="stylesheet" href= "file:///C:\Users\subbaraj\
                Desktop\nit.css">
    </head>
```

```
<body>
<div> welcome to External style sheet ..... </div>
 </body>
  </html>
```

Working with css selectors

→ selector means styles reusability. In css there are different types of selectors.

1] Tag selector
2] Type selector
3] ID selector
4] Class selector
5] Grouping selector
6] Universal selector
7] Decendent selector
8] Customized selector etc.

1] Tag selectors : These are popularly known as type selectors it matches every instance of the element type in the document tree

Syntax:
```
div
{
 styles
 styles
 styles
}
```

ex:
```
<head>
<style type='text/css'>
 h1
 {
  color: #FF0000;
  background-color : #FFFF00;
  font-size : 20px;
  text-decoration: underline;
  font-family: comic sans ms';
 }
</style>
</head>
```

```
<body>
<h1> welcome to Type or Tag selector </h1>
 </body>
```

2) ID selectors:

It is used to specify style for a single, unique element. The id selector uses the id attribute of the HTML element, and is defined with a "#"

Syntax: 1)
```
#div
 {
   styles
   styles
   styles
 }
```

Syntax: 2)
```
div#div
 {
   styles
   styles
   styles
 }
```

ex: 1)
```
<body>
~~#div~~ <style type= "text/css>

#h1
{
  Color: #FF0000; font-size: 20px;
  text decoration: underline;
  font-family : Comic sans ms ;
}

</style>
</head>
<body>
<p id='h1' > welcome to ID Selector </p>
</body>
```

ex: with prefix

```
<body>
<style type="text/css">
b#h1
{
 ~~font~~-color: FF00000; font-size: 20px;
 text-decoration: underline;
 font-family: comic san cs;
}
</style>
</head>
<body>
<b id="h1"> welcome to ID selector </b>
</body>
```

ex:

```
<head>
<style type="text/css'>
#div
{
  color:blue; font-family: tahoma;
}
</style>
<script type='text/~~css~~ javascript' language="javascript">
function myval()
{
  document.getElementById("div1").innerHTML= "HTML5";
}

</script>
</head>
<body>
<p> click the button to change the value based on ID... </p>

~~</body>~~
```

```
<div id="div"> AngularJs </div>
 <div id='div1"> jquery </div>
 <button onclick=" myval()"> Click me </button>
  </body>
```

3] * Selectors Grouping

we can group selector using comma separator

1] Example1:

```
h1 { font-family : sans-serif }
h2 { font-family : sans-serif }
h3 { font-family : sans-serif }
 is equivalent to
h1, h2, h3 { font-family : sans-serif }
```

ex2:

```
  ~~<body>~~
  <head>
   <style type="text/css">
  #div, p   <——  Grouping selectors (Id, tag)
   {
      Color: blue ; font-size: 20px;
      font-family: tahoma;
   }
  </style>
  </head>
  <body>
 <div id="div"> Java script is client side or scripting language
         </div>
 <p> welcome to selector grouping </p>
  </body>
```

3) class selectors

It is used to specify a style for a group of elements. The class selector uses the HTML classes attribute, and is defined with a "." (period).

syntax :

```
    .div
    {
      styles
       styles
        styles
      }
```

ex1 :

```
   <head>
   <style type= 'text/css'>
   .div
   {
    Color: blue; font-size: 20px;
     font-family: tahoma;
    }
   </style>
   <Ihead>
   <body>
   <div Class = "div"> welcome to class selector </div>
   <div class = "div">
  </ body>
```

ID and class grouping

```
 <head>
  <style type='text/css'>
div.div, #h1
   {
     color: blue; font-size: 20px;
   }
   </style>
   </head>
    <body
    <div class="div"> welcome to class </div>
    <p id="H1"> welcome to ~~div~~ ID selector </p>
    </body>
```

Class selector with java script

```
<head>
 <style='text/css'>
  .div
   {
    Color: blue ; font -size: 20px;
   }
  </style>
 <script type='text/javascript'>
 function myval()
  {
   document.getElementById ("div"). innerHTML = "HTML5";
  }
 </script>
  </head>
  <~~/~~body>
 <p> click the button to display class related result </p>
 <div id= "div" class="div"> AngularJS </div>
 <div id= "div1" class= "div"> Java script </div>
<button onclick ="myvalue()"> click me </button>
 </body>
```

Working with css units or measurements

css supports a number of measurement including absolute and relative unix. These units are:

1] EM
2] EX
3] px
4] in
5] cm
6] mm etc.

1] EM : It is a unit of measurement in the field of typography
It is developed based on "m"

<u>syntax</u>: em;

<u>ex</u>:
```
<head>
<style type= 'text/css'>
div
{
  font-size:4em;
}
</style>
</head>
<body>
<div> working with css units ....!!</div>
</body>
```

2] ex: is another measurement developed based on lower case 'x'.

$\boxed{\text{syntax: ex;}}$

```
<head>
<style type= 'text/css'>
div
{
 font-size: 4ex;
}
</style>
</head>
<body>
<div> working with css units.... </div>
 </body>
```

3] px: It defines a measurement in screen pixels.

syntax: px;

```
<head>
<style type='text/css'>
~~<h~~ div
    {
      font-size: 4 px;
    }
<div> working
```

4] in: It defines a measurement in inches.

syntax : in;

```
<head>
<style type='text/css'>
 div
 {
   font-size! 1 in;
 }
  <body>
<div> working with css units!! </div>
</body>
```

5] cm: It defines a measurement in centimeters.

syntax: cm;

```
<head>
<style type='text/css'>
 div
  {
    font-size : 1cm ;
  }
 </style>
 </head>
<body>
<div> working with css units !! </div>
 </body>
```

6] ~~mm:~~ pt: It defines a measurement in points.

Syntax: pt;

```
<head>
<style type='text/css'>
  div
  {
    font-size=1pt;
  }
</style>
</head>
<body>
<div> working

<Ibody>
```

pc: It defines measurement in picas.

syntax: pc;

```
<head>
< style
 div
  {
    font-size= 1pc;
  {
  <      >
  <      >
  <body>
   <div> working
```

% It defines measurement in percentage

syntax: % ;

```
<
<
div
{
 font-size= 100% ;
}
<      >
<      >
<body>
 <div>
 </body>
```

CSS Background property

CSS supports the following list of background properties.

1] background-color
2] background-image
3] background-repeat
4] background-attachment
5] background-position

1] background-color

It is used to set the background color of an element

Syntax: background-color: colorName | colorcode | Hexacode ;

or

background: Color Name| Color code| Hexa code;

Ex:

```
<head>
 <style type='text/css'>
 div
  {
   background-color: #FF00FF ;
  }
 </style>
 </head>
 <body>
  <div> working with css background... </div>
 </body>
```

2) Background-image : It is used to set the background image of an element.

Syntax: background-image : url ('imgpath');
or
background: url ('imgpath');

```
<head>
 < head>
  <style
  body
  {
   background-image : url("html5.png");
   }
  </style>
  </head>
  <body>
  <div> working with css background.... </div>
  </body>
```

3) Background repeat : It is used to control the repeatition of an image in the background.

Syntax: background-repeat : no-repeat;
or
background : no-repeat;

~~ex:~~ ~~<head>~~
~~<style~~

property value

Value	Description
repeat	The background image will be repeated both vertically and horizontally.
repeat-x	The background image will be repeated only horizontally.
repeat-y	The background image will be repeated only vertically
no-repeat	The background image will not be repeated.

ex:
```
<head>
 <style
 body
 {
  ~~f~~o background-image: url("html5.png");
    background-repeat: no-repeat;
  }
 </style>
 </head>
<body> working with css Backgrounds.....!! </body>
 </body>
```

Background attachment : It is used to controll scrolling of an image in the background.

syntax: background-attachment : fixed;
 or
 background : fixed;

Background - position : It is used to control position of an image in the background.

syntax: background - position: center;
 or
 background : center;

```
<head>
 <style>
 body
 {
  background-image: url ("html5.png");
  background-repeat: no repeat;
  background-attachment: fixed;
  background-position: center;
 }
</style>
</head>
 <body>
 <div> working with css Background </div>
  </body>
```

CSS Font Family

CSS supports following list of font properties:

1] The font-family property is used to change the face of a font.

2) The font-style property is used to make a font italic or oblique.

3] The font-varient property is used to create a small-caps effect.

4] The font-weight property is used to display bold or light a font appears.

5] The font-size property is used to increase or decrease the size of a font.

ex:
```
<head>
<style type="text/css">
div
 {
  font-family: tahoma;
  font-size: 10 px;
  font-style: oblique;
  font-weight: bolder;
  font-varient: small-caps;
 }
 </style>
 <body>
<div> working with css ~~background~~ ~~prop~~ font family </div>
 </body>
```

⊛ CSS Text properties

CSS supports the following list of text properties:

1] color:
2] Direction
3] Text-decoration
4] text-indent
5] text-align
6]

1) color: using this property we can change the text color.

syntax: color: colorName / color code

Example:
```
<head>
<style type='text/css'>
div
{
  color: blue;
}
</style>
</head>
<body>
<div> welcome to css Text properties...</div>
</body>
```

2) Direction: It is used to display the direction of the text. The Text default direction is left to right.

Syntax: direction: value(s).

Example:
```
<head>
<style type='text/css'>
div
{
  direction: ltr;
}
p
{
  direction: rtl;
}
</style>
<body>
<div> welcome to css direction property </div>
<p> welcome to css Text property</p>
</body>
```

3] Text-decoration : It supports different decoration, like, underline, overline, line through.....

Syntax: text-decoration : value ;

Ex:
```
<head>
<style type='text/css'>
 div
 {
   text-decoration : underline;
 }
 p
 {
   text-decoration : overline;
 }
b
 {
   text-decoration: line through;
 }
</style>
 </head>
 <body>
<div> It is underline format </div>
 <p> It is underline format</p>
  <b> It is underline format </b>
 </body>.
```

4] Text - Indent: This property is used to display indent at the starting of a paragraph.

Syntax : Text-indent : pixels ;

5] Text - align: This property is used to align the text in a specific direction.

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

ex:
```
<head>
<style type='text/css'>
~~div~~ p
 {
   text-indent: 50px;
   text-align: justify;
 }
</style>
</head>
<body>
<~~div~~p> welcome to indent property </~~div~~p>
</body>
```

6] Letter-spacing: This property is used to apply space between letters.
 Syntax: letter spacing pixels;

ex:
```
<head>
<style type='text/css'>
p
{
  letter-spacing: 10 ~~pixels~~ px;
}
</style>
</head>
<body>
 <p> welcome to css properties </p>
 </body>
```

7] word-spacing: This property is used to add or substract space between the words.
```
<head>
<style-type='text/css'>
 p
 {
   word-spacing: -5px;
 }
```

```
<Istyle>
<I head>
<body>
<P> welcome to css property <IP>
<Ibody>
```

8] Text - transform : This property is used to capitalize the text, convert text upper case or lower case format.

Syntax : Text - transform : value;

ex:
```
<head>
<Style = 'text/css'>
P
{
 text- transform : capitalize;
}
<Istyle>
<Ihead>
<body>
<P> welcome to css properties <Ip>
<Ibody>
```

9] white-space : This property is used to control the flow and formatting of text.

```
<head>
<style type= 'text/css'>
 P
  {
    White-space: nowrap;
  }
<Istyle>
 <Ihead>
 <body>
 <P> Line never break ; Line never break <IP>
 <Ibody>
```

10] Verticle Align: This property sets the verticle alignment of an element. It supports the following list of values:

1] baseline
2] sub
3] super
4] top
5] text-top
6] middle
7] bottom
8] text-bottom
9] 0px
10] 10px

```
<head>
   <style type= 'text/css'>
    p
     {
       verticle-align: sub;
      }
    </style>
     <body> verticle alignment of an <p> element </p>
     </body>
```

CSS Borders:

CSS supports the following list of border properties.

1] Border-color: The border color specifies the color of a border.

2] Border-style: It specifies whether a border should be solid, dashed line, double line, or one of the other possible values.

3] The border width specifies the width of a border.

ex:
```
<head>
  <style type='text/css'>
   div
    {
      border-color: green;
      border-style: dashed;
      border-width: 2px;
    }
   </style>
   <head>
   <body>
   <div> welcome </div>
   </body>
```

Working with Advanced style sheets (CSS3)

CSS3 is new style sheet, it is divided into several modules. Each module having new capabilities and extended features.

1] @font-face
2] opacity
3] RGBA (Red Green Blue Alpha or Amber)
4] Border-Radius
5] Box-Shadow
6] Text-Shadow
7] Gradient
8] multiple background images
9] Transform
10] Transition
11] multi Column layout
12] styling forms with Attribute selector.
13] Wrapping up
14] Box-sizing and Box-model
15] CSS3 Selector

*] CSS3 Browser support

→ In CSS3 we are using the following prefixes for advanced properties.

1] IE requires the prefix -ms- (microsoft seem)

2] Firefox requires the prefix -moz-

3] chrome and safari requires the prefix -webkit-

4] opera requires the prefix -o-.

Working with CSS3 multiple column properties

In CSS3 we can create multiple columns layout for E-news papers. The following list of properties frequently we are using.

1] column - count
2] column - gap
3] column - rule
4] column - fill
5] column - rule - color
6] column - rule - style
7] column - rule - width
8] column - span
9] column - width
10] columns

1] column - count : This property is used to specify number of columns and element divided ~~should~~ into:

Syntax: Column count: number/auto;

Value	Description
number	The optimal number of columns into which the Content of the element will be flowed.
auto	The number of columns will be determined by other properties.

ex:
```
<head>
 <style type='text/css'>
  div
  {
   Column-count: 3;
  -moz- column-count: 3; /* mozilla firefox */
  -webkit- Column-count: 3; /* chrome/safari */
  - ms- Column- Count: 3; /* IE */
  - o- opera- Column-count: 3; /* opera */
  }
  </style>
  </head>
  <body>
 <div>Some text.... !! </div>
  </body>
```

2) Column- gap

This property specifies the gap between the columns.

syntax: column- gap: length | normal;

Value	Description
length	a specified length that will set the gap between the column.
normal	specifies a normal gap between the column.

ex:
```
<head>
<style type= 'text/css'>
div
{ Text-align: justify;
  column-gap: 30px;
 -moz- column-gap: 30px;
  column- count: 3;
 -moz- column- Count : 3;
}
```

```
</style>
 </head>
<body>
 <div> some Text.... !! </div>
 </body>
```

3) <u>column-rule-property</u> : It is a shorthand property for setting all the column-rule-* properties. The column rule property sets the width, style, and color of the rule between columns.

<u>syntax</u>: column-rule: column-rule-width column-rule-style column-rule-color ;

<u>Value</u>	<u>Description</u>
column-rule-width	sets the width of the rule between columns
column-rule-style	sets the style of the rule between columns.
column-rule-color	sets the color of the rule between columns.

<u>ex</u>:
```
<head>
<style type='text/css'>
div
{
  text align: justify ;
  column-count : 3 ;
-moz-column-count : 3 ;
  column-gap: 30px;
 -moz- column-gap : 30px;
 -column-rule-width: 2px;
-moz- column-rule-width: 2px;
 column-rule-style : solid ;
 -moz- column-rule-style: solid ;
 column-rule-color: #FF0000 ;
-moz- column-ruler color: #FF0000;
}
</style>
</head>
```

```
<body>
<div> some Text.... !! </div>
</body>
```

4] column-fill : It specifies how to fill columns, balanced or not.

Syntax: column-fill : balance| auto;

Value	Description
balance	Columns are balanced. Browsers should minimize the variation in column length.
auto	Columns are filled sequentially, and they will have different lengths.

~~<head>~~
~~<style type='text/css'>~~

Note: currently no major web browser support.

5] column-span : It specifies how many columns an element should span across.

Syntax: column-span : 1|all ;

Value	Description
1	The element should span across 1 column.
all	The element should span across all columns.

ex:
```
<head>
<style type= 'text/css'>
~~div newspaper~~ div
 {
   text-align: justify;
  -webkit - Column- count: 3;
   Column- count: 3
 }
 h2
 {
  - moz- column-span :1 ;
    column-span: 1 ;
 }
```

```
<|style>
<|head>
 <body>
  <div>
<div><h2> Heading </h2>
      Some Text </div>
    </body>
```

6] <u>column-width</u> : It specifies the width of the column.

<u>syntax</u> : column-width: auto | length;

<u>Value</u>	<u>Description</u>
auto	The column width will be determined by the browser.
length	A length that specifies the width of the column.

<u>Note</u>: If you are increasing the page size number of columns will desrease and vice versa.

<u>ex</u>:
```
<head>
 <style type='text/css'>
  div
   {
    -moz-column-width: 100px;
     Column-width: 100px;
   }
  </style>
  </head>
  <body>
  <div> Some Text ..... </div>
 </body>
```

7] <u>Columns</u> : This property is shorthand property for setting column width and column count.

syntax: columns: column-width column-count;

Value	Description
column-width	The width of the column
column-count	The number of column.

ex:
```
<head>
<style type='text/css'>
div
{
  columns: 100px 3;
 -moz- columns: 100px13;
}
</style>
</head>
<body>
<div> Some Text </div>
</body>
```

(*) Working with CSS3 background property

CSS3 supports the following list of advanced background properties

1] background size:

property	Description
background-size	specifies the size of the background images.
background-origin	specifies the positioning area of the background images.
background-clip	specifies the painting area of the background images.

1] Background size : This property specifies the size of the background images. Before CSS3 the background image size was determined by the actual size of an image.

syntax: background-size: length| percentage| cover| contain;

ex1:
```
<head>
<style type='text/css'>
body
{
 background: url(img_flwr.gif);
 background-size: 480px 480px;
-moz-background-size: 480px 480px;
 background-repeat: no repeat;
}
</style>
</head>
<body>
 <p>
 css-image
 </p>
<p> original image: <img src="water.gif" alt=
"Flowers" width="200" height="200" /> </p>
</body>
```

Background clip: It specifies the painting area of the background.

syntax: background-clip: border-box| padding-box| content-box;

ex:
```
<head>
<style='text/css'>
div
{
 width: 300px; height: 300px;
 padding: 50px; background-color: lightblue;
 background-clip: border-box;
 border: 4px dashed #FF0000;
 text-align: justify;
}
```

```
<s/style>
<lhead>
<body>
<div> some Text....!! </div>
<lbody>
```

3] <u>Background origin</u> : This property specifies what the background position property relative to

<u>Syntax</u> : background-origin : padding-box| border-box| Content box;

<u>Ex</u>:
```
<head>
<style>
div
{
  border: 2px solid red; padding: 30 px;
  background image: url('chrome.png');
  background-repeat: no repeat;
  background- position: left;
  background -origin: content- box;
}
<Istyle>
<Ihead>
<body>
   <div> some Text.....!! </div>
   <lbody>
```

Working with CSS3 borders

CSS3 supports the following list of advanced properties:

Property	Description
border-image	A shorthand property for setting all the border image - * properties.
border-radius	A shorthand property for setting all the four border - * - radius properties.
box-shadow	Attaches one or more drop-shadows to the box.

⊛ **Border image:** It is a shorthand property for setting a border image source, border image slice, border-image width, border-image outset and border image repeat properties.

Syntax: border-image: source slice $\overset{\text{width}}{\wedge}$outset repeat;

ex:
```
<head>
<style>
 div
 {
   border: 15px; solid transparent;
   width : 250px; padding: 10px 20px;
 }
#round
{
 border- image : url( border.png ) 30 30 round;
}
</style>
</head>
 <body>
        ~~<img src=~~
<div id='round'>   some Text.... </div>
  </body>
```

Box shadow : The box shadow property attaches one or more drop shadows to the box.

Syntax : box-shadow: h-shadow, v-shadow, blur shadow color inset;

ex :
```
<head>
<style>
div
{
 width: 300px ; height: 80 px;
 background-color: #FFFF00;
 box-shadow: 15px 15px 15 px #FF0000 ;
 }
</style>
</head>
<body>
<div> </div>
</body>
```

* Border- Radius : It ~~specifies~~ is a shorthand property for setting the four border-*- radius properties.

Syntax : border- radius : 1-4 length % / px;

ex :
```
<head>
<style>
div
{
 width: 300px ; height: 10px ;
 border : 2px solid #FF0000 ;
  Padding :5px ;
  border -radius : 5px;
}
</style>
</head>
<body>
```

```
<div> welcome to Borders.... </div>
  </body>
```

⊛ CSS3 Text properties

CSS3 supports the following list of text properties :

1] hanging- punctuation
2] punctuation- trim
3] Text - emphasis
4] Text - justify
5] Text - outline
6] Text - overflow
7] Text - shadow
8] Text - wrap
9] word - break
10] word - wrap

1] Word-wrap property :

This property allows long words to be able to be broken and wrap onto the next line.

java script syntax:

```
object.style.word wrap = "break - word"
```

Syntax :

```
word-wrap: normal | break- word ;
```

Value	Description
normal	Break words only at allowed break points.
break - word	Allows underbreakable words to be broken.

ex:
```
<head>
<style>
div
{
  width: 10em;
  border: 2px solid #FF0000;
  word-wrap : break-word;
}
```

```
</style>
</head>
<body>
<div> HTML5 is New Hypertext markup language for latest
web apps. It is new for mobile apps for responsive webapp </div>
</body>
```

2] word-break : It specifies the line breaking rules for non-CJK scripts.

Syntax: word-break : normal|break-all| hyphenate;

Java script syntax :

object.style.wordBreak= "hyphenate".

Value	
normal	Breaks non-CJK scripts according to their own rules.
break-all	Lines may break between any two characters for non-CJK script.
hyphenate	words may be broken at an appropriate hyphenation point.

3] Text wrap property

This property specifies line breaking rules for text

Syntax: text-wrap: ~~n~~ normal|none| unrestricted | suppress;

Java script syntax : object.style.textwrap = "none".

Note: This property currently major web browsers are not supported

4] Text shadow : This property applies shadow to text. you specify the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow.

Syntax : text-shadow : h-shadow v-shadow blur color;

Java script syntax :

object. style . textshadow = "2px 2px #ff0000"

ex:
```
 <head>
 <style type='text/css'>
div
 {
  font-size: 30 px ;
  text-shadow : 5px 5px 5px # FF0000;
 }
p
 { font-size: 30px;
  text shadow: 5px 5px #FF0000 ;
 }
 </style>
 </head>
<body>
 <div> welcome to text shadow property.... </div>
 <p> welcome to text shadow property.... </p>
</body>
```

Text overflow property :

This property specifies what should happen when text overflows the containing element.

syntax : text-overflow : clip| ellipses| string;

Java script syntax : object. style . textoverflow = "ellipses" ;

value	Description
clip	clips the text
ellipses	Render an ellipses ("....") to represent clipped text.
string	Render the given string to represent clipped text

```
<head>
 <style>
 div.test
 {
   white-space :nowrap;
   width :12em ; overflow: hidden;
   border : 1px solid #ff0000 ;
   text overflow :ellipsis ;
 }
 </style>
 </head>
  <body>
  <div . class ="test"> This will some long text that will not fit in
               box <|div>
   </body>
```

Text-justify : This property specifies the justification method to use when text-align is set to "justify". This property specifies how justified text should be aligned and spaced.

syntax : Text-justify : auto |inter-word| inter-ideograph| inter-cluster | distribute| kashida |trim;

Java script syntax : object.style.text justify ="inter-word";

ex:
```
<head>
<style
div
{
 text-align: justify;
 text-justify :inter-word;
}
</style>
</head>
<body>
 <div> some text.... Resize the Browser window..... </div>
  </body>
```

CSS3 font properties

CSS3 supports the following list of advanced font properties.

1) @ font-face
2) font-size-adjust
3) font-stretch

1) @ font-face : The CSS3 font-face Rule: your "own" fonts are defined in the CSS3 @font face. Before CSS3 webdesigners had to use fonts that are already installed in the users computers.

www.1001freefonts.com

1) .ttf - True Type Fonts
2) .otf - opene type fonts
3) EOT - Embeded open type
4) WOFF → web open font format
5) SVG → Scalable

Note : firefox, chrome, safari and opera support font of type .ttf and ~~.ttf~~ ,

Note : Enternet Explorer support the new @ font-face rule, but it only supports fonts of type .eot.

Ex1 :
```
<head>
<style
 @font-face
   {
     font-family : myFirst font;
     src: url ('sansation_Bold_Italic.ttf')
   }
 div
 {
   font- family : my first font ;
 }
</style>
</head>
 <body>
   <div> some text </div>
     </body>
```

2] Font_size_adjust

This property gives you better control of the font size.

Syntax: font-size-adjust : number| none| inherit;

Value	Description
number	Defines the aspect value to use.
none	Default value. No font size adjustment.
Inherit	Inherits the font size adjustment from parent elements.

ex:
```
<head>
 <style
 div
 {
   font-size-adjust : 20px;
 }
</style>
 </head>
<body>
<div> some text..... </div>
 </body>
```

3] font-stretch : This property make or allow text wider and narrower.

Syntax : font-stretch: wider| narrower| ultra-condensed| extra-condensed| condensed| semi-condensed| normal| semi-expanded| extra-expanded| ultra-expanded| inherit.

Note : none of the major supports the font- stretch property.

CSS3 Transforms:

Transform is a powerfull property to move, scale, turn, spin and stretch element. It supports the following list of properties.

1] transform
2] transform-origin
3] transform-style

Transform support the following five methods.

1] translate()
2] rotate()
3] scale()
4] skew()
5] matrix()

1] translate() : with the help of this method you can move your object depending on its parameter. Two type of parameter you can pass in this method one is from left (x-axis) and the another is from top (y-axis).

Syntax : transform : translate(x,y);

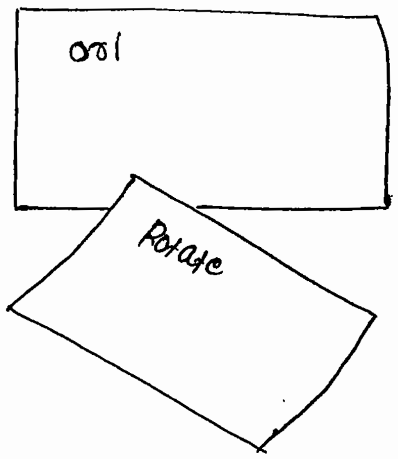
Example:

```
<head>
<style type='text/css'>
div
{
    width :90px ; height:60px;
    background-color: #FF9900;
    border: 2px solid #FF00FF;
}
div#div1
{
  transform : translate (20px, 30px);
}
div #div2
{
  transform : translate (40px, 60px);
}
</style>
</head>
<body>
<div>  </div>
<div id="div1">  </div>
<div id = "div2">  </div>
</body>
```

2] Rotate (): with the help of this method you can rotate your object depending on its value. Two types of value you can pass in this method one is positive and the another one is negative

Syntax: transform: rotate (x,y);

x → represent clock wise Rotation
y → represent counter clock wise rotation. (Anticlock).



ex:
```
<head>
<style>
 div
 {
  ~~transform:~~ width: 90px; height: 60px;
   background-color: #FF9900;
   border: 2px solid #FF00FF;
 }
 div #div1
 {
  transform: rotate (30deg);
 }
</style>
</head>
<body>
<div> ori </div>
<div id='div1'> rotate </div>
</body>
```

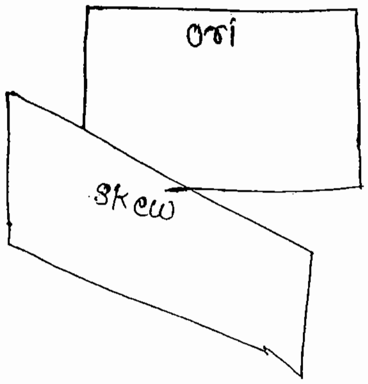
3] Scale : with the help of this method you can increase or decrease your object size depending on its value passes in the parameters. Two types of value you can pass in the parameter, one is for width (x-axis) and the another one is for height (y-axis).

Ex: scale (x,y);

ex:
```
<head>
<style>
 div
 {
   width: 90 px; height: 60 px;
    background color: #FF9900;
    border: 2px solid #FF00FF;
 }
div#div1
{
  margin: 100 px;
} transform : scale (2,3);
</style>
 </head>
<body>
<div> Ori </div>
<div id='div1' > scale </div>
 </body>
```

4] Skew() : with the help of this method you can change the angle of your object depending on its value passed in the parameter. Two types of value you can pass in the parameter, one is for horizontal (x-axis) and the another one for vertical (y-axis).

Syntax : transform: skew (x,y);



ex:
```
<head>
 <style>
  div
  {
    height: 60px ; width: 90px;
     background-color: #FF9900;
      border: 2px solid #FF00FF;
  }
  div#div1
  {
    transform: skew (20deg, 30deg);
  }
 </style>
 </head>
 <body>
   <div> ori </div>
   <div id='div1' > skew </div>
   </body>
```

5] Matrix() :

CSS3 Transitions

A Transition is such a property of css3 which is used to animate the object, without using flash or any other animation application. with this feature of css3 you can change the shape and size of your object with animated effects.

Transition properties : It supports the following list of properties

1] Transition	A short hand property for setting the four transition property into a single property.
2] transition-property	specifies the name of the css property to which the transition is applied
3] Transition-Duration	Defines the length of time that a transition takes.
4] transition-timing-function	Describes how the speed during the transition will be calculated.

```
  transition-duration : 5s ; float: right ;
 }
</style>
</head>
<body>
  <div>    </div>
 <div>    </div>
 </body>
```

<u>Transition Delay</u>: It specifies when the transition effect will start.
The transition delay value is ~~defied~~ defined in seconds(s) or
milliseconds (ms).

<u>syntax</u> : transition-delay: time;

<u>value</u>	<u>Description</u>
time	specifies the number of seconds for milliseconds to wait before the transition effect will start.

<u>ex</u>:
```
<head>
 <style>
  div
  {
   width: 90 px;  height: 60 px;
   background-color: #FF0000;
   transition: width;  border-radius: 20px;
   transition-duration: 2s ;  transition-delay : 1s;
  }
   div : hover
   {
    ø width: 350 px;
    }
  </style>
  </head>
   <body>
    <div> </div>
     </body>
```

Working with CSS3 Animation.

CSS3 supports the following list of Advanced Animations. we can create animations, which can replace animated images, flash animation and javascript in many webpages.

Animation is a property to change an object from one style to another style in a animated way.

Animation properties

CSS3 supports the following list of animation properties

1] @keyframes
2] animation
3] animation name
4] animation duration
5] animation delay
6] animation direction etc.

1] @keyframes rules : The @keyframe rule is where the animation is created. specify a css style inside the @keyframes rule and the animation will gradually change from the current style to the new style. we can create user defined animations.

How to Implement Animation :

An animation is an effect that lets an element gradually change from one style to another style : you can change as many styles you want, as many times you want.

specify when the change will happen in percent or the keywords "from" and "to" which is the same as 0% and 100% . 0% is the begining of the animation . 100% when the animation is complete.

CSS3 Animation property

The Animation property is a shorthand property for six of the animation properties: animation-name, animation-duration, animation timing-function, animation-delay, animation-iteration count, and animation direction.

ex:
```
<head>
 <style>
  div
  {
    width: 90px; height: 60px;
    background-color: #FF9900;
     transition: width; float: rigth;
     transition-duration: 5s;
   }
   div: hover
    {
      width: 350px;
     }
   </style>
   </head>
    <body>
      <div>   </div>
     </body>
```

Multiple Transitions

```
<head>
<style>
  div
   {
     width: 90px; height: 60px;
      background color : #FF0000;
      transition: width; border: ~~5px;~~ -radius: 20px;
        transition-duration: 2s;
     }
    div : hover
    {
      width: 350px;
    }
   div1
    {
       width: 90px; height: 60px;
          background-color: #0000FF;
        transition: width; border-radius : 5px;
```

Transition property : The transition property is a shorthand property for the four transition properties: transition property, transition duration, transition-timing-function, and transition-delay.

syntax : transition: property duration timing-function delay;

Ex1:
```
<head>
<style>
div
{
 width: 90px; height : 60px;
  background - color : #FF0000 ;
  transition: width 2s ;
 }
div hover
 {
  width: 350 px;
 }
</style>
</head>
<body>
 <div>   </div>
</body>
```

Transition Duration : It specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.

syntax : transition- duration: time;

Value	Description
time	specifies how many seconds or milliseconds a transition effect takes to complete.

Transition Timing function : It specifies the speed curve of the transition effect. This property allows a transition effect to change speed over its duration.

Syntax: Transition-Timing-function; linear| ease| ease-in| ease-out | ease-in-out | cubic - bezier (n,n,n,n) ;

syntax: animation: name duration timing-function delay -iteration- count direction;

ex1:
```
<head>
<style type='text/css'>
div
{
 width: 80 px ; height : 70 px;
 background -color: #FF0000;
 position: relative;
 animation: mymove 1S 4 ;
 border-radius: 20px;
}
-moz-@keyframes mymove  /* firefox */
{
 from {left : 0px;}
 to { left :300px;}
}
</style>
</head>
<body>
<div>   </div>
</body>
```

ex2:
```
- moz- @ keyframes mymove
{
```

CSS3 Animation duration property

The animation duration property defines how many seconds or milliseconds an animation takes to complete one cycle.

syntax: animation- duration: time ;

Value	Description
time	specifies the length an animation takes to finish. Default value is 0, meaning there will be no animation.

Animation iteration count property

The animation iteration count property defines how many times an animation should be played.

Syntax : animation-iteration-count: value;

Value	Description
n	A number that defines how many times an animation should be played.
infinite	specifies that the animation should be played infinite times.

ex:
```
<head>
 <style
 div
 {
   width: 80px; height: 90px;
   background-color : #FF0000;
    position: relative;
    animation: mymove;
    animation-duration: 5s;
    animation-iteration-count : 6;
    border-radius: 20px;
  }
  @keyframes mymove
  {
    from { left : 0px ; }
      to { left : 300px; }
  }
 </style>
 </head>
```

ex: with different direction

```
<head>
<style
div
{
  width: 60 px; height : 70 px;
   background color : #FF0000 ;
   position: relative ;
   animation: mymove ;
   animation-duration: 1s ;
   animation- iteration- count : infinite ;
   border-radius : 20 px ;
 }
 @keyframes mymove
 {
   from { top: 0px ;}
   to {top: 250 px ;}
 }
<lstyle>
<lhead>
<body>
 <div>  <div>
 <lbody>
```

Animation Direction

The animation direction property defines whether or not the animation should play in reverse on alternate cycles. If the animation direction value is "alternate" the animation will be played as normal every odd time (1,3,5 etc...) and backwards every even time (2,4,6. etc...).

Syntax: animation direction : value;

Value	Description
normal	Default value. The animation should be played as normal value.
alternate	The animation should be played in reverse on alternate cycle.

ex: <head>
```
<style
div
{
  height : 70 px; width : 60 px;
  background-color : #FF0000;
   position: relative;
   animation: mymove;
    animation-duration: 5s;
   animation- iteration- count: infinite;
    border-radius: 20px;
    animation- direction: alternate;
  }
-moz- @keyframe mymove /* firefox */
  {
  0% { background: Red ; left : 0px ; top : 0px ; }
  25% { background: yellow ; left : 200px ; top: 0px; }
  50% { background: Green ; left : 0px ; top : 200px; }
  75% { background: bule ; left : 0px ; top : 200px; }
   100% { background: Red ; left : 0px ; top : 0px; }
  }
 </style>
 </head>
 <body>
  <div> </div>
   </body>
```

Working with Advanced selectors :

In CSS3 selector means styles reusability. There are the following list of advanced selectors:

1. [attribute ^= values] selector

The [attribute ^= value] selector matches every element whose attribute value begins with a specified value.

Note: The [attribute^=value] selector is supported in all major browser.

Ex:
```
<head>
 <style
 div [class^= "test"]
 {
  background: #ff0000; font family: tahoma;
 }
<lstyle>
 <lhead>
<body>
 <div class= "first-test" > The first div element </div>
 <div class = "test"> The second div element </div>
 <~~div~~ p class = "test"> This is some text in a </p>
 </body>
```

ex 2:
```
<head>
 <style
  ~~di~~ [class ^ = "test"]
   {
    background : #ff0000 ; font family: tahoma ;
   }
  </style>
  </head>
 <body>
 <div class = "first -test" >
```

2] CSS3 [attribute $= value] selector

The [attribute $=value] selector matches every element whose attribute value ends with a specified value.

Note: The [attribute $= value] selector is supported in all major browsers.

ex:
```
<head>
<style>
div[class $= 'test']
{
  background: #ffff11;
}
</style>
</head>
<body>
<div class= "first_test"> The first div element </div>
<div class= "second"> The second div element </div>
<div class="test"> The Third div element </div>
<p class ="test"> The fourth div element </div>
</body>
```

3] css3 [attribute * = value] selector

The [attribute * = value] selector matches every element whose attribute value containing a specified value.

Note: The [attribute *= value] selector is supported in all major web browsers.

ex:
```
<head>
<style>
[class* ="test"]
{
  background : #ffff0 ;
}
</style>
</head>
```

```
<body>
<div class= "first_ test"> The first div element </div>
 <div class= "second "> The second div element </div>
 <div class= "test"> The Third div element </div>
  < p class= "test"> The fourth div element </div>
    </body>
```

CSS3 - first of type selector

The first of type selector matches every element that is the first child, of a particular type of its parent.

Note : The first of type selector is supported in all major web browser except IE8 and earlier.

ex:
```
<head>
 <style>
 p: first-of type
   {
     background : #00ffff;
   }
  </style>
  </head>
 <body>
 <h1> This is heading </h1>
  < p> This is first paragraph </p>
  <p> The second paragraph </p>
  <p> The Third paragraph </p>
   </body>
```

CSS 3- only-of-type selector

The: only of type selector is matches every element that is the only child of its type of its parent.

ex:
```
<head>
  <style>
  {
  p: only-of-type
  {
    background: #FF00FF;
    font-family: tahoma;
  }
  </style>
  </head>
  <body>
   <p> This is paragraph </p>
 <div><p> This is paragraph </p>
    <p> This is paragraph </p> </div>
  </body>
```

CSS3 - nth-child selector

The nth child selector matches every element that is the nth child regardless of type of its parent.

Note: The nth child selector is supported in all major web browsers. except IE8 and earlier.

ex:
```
<head>
  <style>
  p: nth-child ~~selector~~ (4)
  {
   background: #FF CC00 ; font family: tahoma;
  }
  </style>
 </head>
 <body>
 <h1> This is heading </h1>
  <p> The first paragraph </p>
  <p> The second paragraph </p>
 <p> The Third paragraph </p>
 <p> The fourth paragraph </p>
  </body>
```

CSS3 User Interface

User interface is powerfull property in css3 it is used to impleme nt user interface without modifiging a code section. you can resize div element and set the user interface property as follows:

CSS3 Resizing

It specifies whether or not an element should be resizable by the user.

resize : is a such property of user interface, by which you can resize your div layout on your browser. Three features of resize you use.

a) resize: both
b) resize: vertical
c) resize: horizontal

syntax: resize: none| both| horizontal | vertical

value	Description
none	User cannot resize the element. User can adjust both the height and the width.
horizontal	User can adjust the width of the element.
vertical	User can adjust the height of the element.

ex:
```
<head>
 <style>
 div
  {
   border: 2px solid;
   padding: 10px 40px;
   width: 300 px;
   resize: vertical;
    overflow: auto;
  }
 </style>
 </head>
  <body>
   <div> Some Text.... </div>
   </body>
```

HTML5 Web Storage

It has the following aliase name:

1] client side storage
2] web storage
3] off line storage
4] Local storage

It is very close to cookies concept. The main intesion of web storage is, store the data at client side without disturbing the performance of the website.

In HTML5 web storage is classiefied into the following two types:

1] session storage
2] Local storage

1] session storage: It is also use browsers data locally but it is for limited period when we close the browser it will automaticaly delete the stored data and we cant see the browsers stored data again. HTML5 introduces the session storage attribute which would be used by the sites to add data to the session storage.

ex:
```
<head>
 <script type='text/javascript'>
   var sno=100;
   alert(" The Entered Number is : "+sno);
 </script>
 </head>
 <body>
 <a href = "page2.html"> page2 </a>
 </body>
```

o/p is 100, page2 (link).

ex: o/p: hi, No msg
```
<head>
<script>
alert("hi");
aler (sno);
</script>
</head>
```

ex3 : with session storage

```
<head>
<script>
sessionStorage.sno=100;
alert(sessionStorage.sno).
</script>
</head>
<body>
<a href="page2.html"> page2 </a>
 </body>
```

O/P : 100 , page2.

ex:
```
<head>
<script type= 'text/javascript'>
 alert("hi");
 alert( session storage.sno);
</script>
</head>
```
o/p is Hi, 100.

Local storage : which is used to store the data locally. It stores the data with no expiration date, means if browser is closed then it will not delete the data and we can view the data any time and even after year.

ex1 :
```
<head>
<script>
localstorage.sno =100
alert (local storage.sno);
</script>
</head>
<body>
<a href ="page2.html" > page2 </a>
</body>
```

O/P : 100, Page2 (link)

ex2:
```
<head>
<script type ="text/javascript>
   alert ("hi");
   alert (local storage .sno)
   </script>
   </head>
o/p is hi, 100 .
```

Advanced Example for session Storage :

```
<body>
<script type= 'text/javascript'>
 if(session storage.hits)
  {
   sessionstorage.hits = Number(session storage.hits)+1 ;
  }
else
 {
   sessionStorage.hits = 1 ;
 }
 document.write (" Total Hits " + session Storage.hits);
   </script>
   <p> Refresh the page to increase the number of Hits </p>
    </body>
```

ex2:
```
<body>
   <script >
   if (~~session~~ local storage .hits)
   {
    ~~Session~~ local Storage .hits = Number (~~session~~ local Storage .hits)+1 ;
   }
  else
   {
    ~~session~~ local storage .hits =1;
   }
   document. write ("Total hits" + local storage .hits );
    </script>
    <P> Refresh the page to increase the number of Hits <P>
     </body>
```

In web storage storing sensitive data on a local machine could be dangerous and could leave a security whole. A session storage data deleted automatically when you close the browser where as local storage unable to that time we should clear the local storage with the help of local storage.clear() method.

Define web SQL ?

web SQL Database is web page API for storing data in databases that can be queried ~~as~~ ~~a~~ using a variant of SQL. The API is supported by Google chrome, opera, safari and the Android Browser.

Indexed DB : It is basically a persistant data store in the browser - a database on the client side. Like a regular relational databases it maintains indexes over the records it stores and developers use the IndexedDB Java script API to locate records by key or by looking up an index.

HTML5 Geolocation (Deeper Integration with OS)

HTML5 Geolocation API allows users to share their location with web applications by capturing the approximate longitude and latitude Coordinates of the user with their permission.

Define latitude : latitude is a geographic coordinate that specifies the north- ~~south~~ position of a point on the Earths surface.

Longitude : It is a geographic ~~to~~ Co-ordinate that specifies the east west position of a point on the Earths surface.

Types of maps

maps are classified into following basic types

1) ROADMAP (normal , default 2D map)

2) SATELITE (photographic map)

3) HYBRID (Photographic map + Roads & city names)

4) TRRRAIN (map with mountains, rivers etc.)

Google maps controls

Google map supports the following list of controls.

1] zoom
2] pan
3] map Type
4] Street View
5] Scale
6] Rotate

Q] write a script to verify the Browser support.

```
<head>
  <script>
   function  my support()
    {
     if(navigator.geolocation)
      {
        alert("your Browser supports");
      }
    else
      {
        alert("your Browser unable to support");
      }
     }
  </script>
    </head>
   <body>
    <p> click the button to display the Browser </p>
  <button onclick =" my support ()"> support_Browser </button>
     </body>
```

getbootstrap.com

Geolocation methods

Method	Description
get current position()	It retrieves the current geographic location of the user.
watch position()	It retrieves periodic updates about the current geographic location of the device.
clearwatch()	This method cancels an ongoing watch position call.

7-9-15

Working with HTML5 Drag and Drop

Define Drag and Drop

It is powerful user interface concept which makes it easy to copy, reorder and deletion of items with the help of mouse clicks. It allows the user to click and hold the mouse button down over an element, drag it to another location, release the mouse button.

Creating draggable content

Making an object draggable is simple. set draggable = true attribute on the element you want to make moveable.

Example :
```
<body>
    <img src="html5.png" width="100px" height="100px"
        draggable="true">
    </body>
```

Drag and drop Events

Drag and Drop supports the following list of events.

Events
dragstar
dragenter
dragover
dragleave
drag
drop
dragend

allow Drop : By default all elements are not dragged or dropped. To allow a drop we must prevent the default handling of the element

Ex: function allowDrop (ev)

```
{
  ev - preventDefault ();
}
```

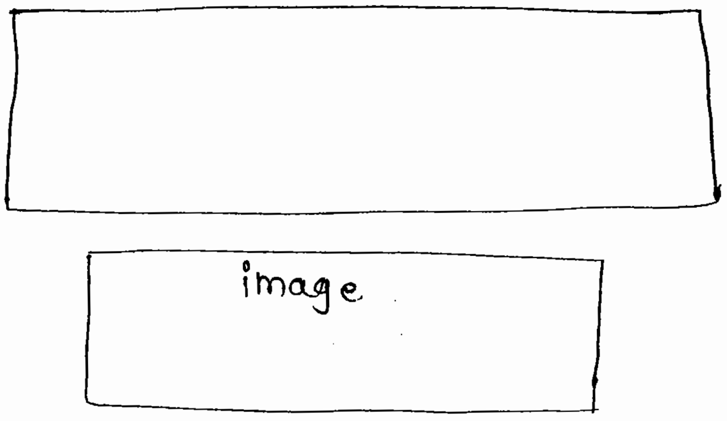
What to Drag : The dataTransfer. setData() method sets the data type and the value of the dragged data.

Ex:

```
function drag (ev)
{
  ev.data dataTransfer .setData(" text", ev. target .id);
}
```

DO the Drop : when the drag data is dropped the drop event occures. This event fires as follows:

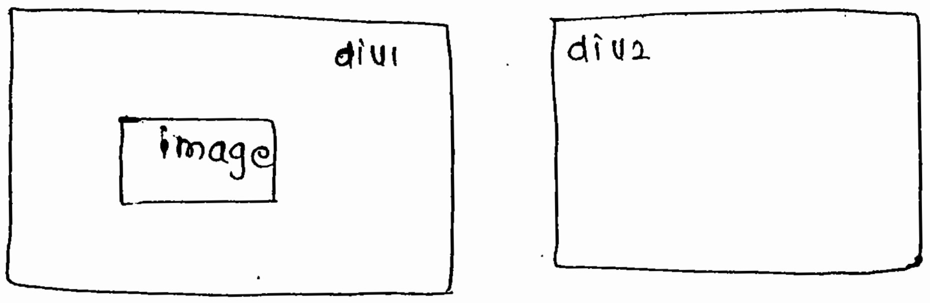
```
function drop (ev)
{
 ev_ preventDefault ();
 var data = ev.data Transfer. get Data("text");
 ev. target . appendChild
 (document . getElementById (data));
 }
```



```
<head>
<style>
 #div1
  {
   width: 350px; height: 70px;
    border: 1px solid #FF00FF;
  }
 </style>
<script type="text/javascript">
 function allowDrop(ev)
 {
   ev.preventDefault();
  }
 function drag(ev)
 {
  ev.dataTransfer.setData("text/html", ev.target.id);
 }
 function drop(ev)
 {
   ev.preventDefault();
   var data= ev.dataTransfer.getData("text/html");
   ev.target.appendChild (document.getElementById(data));
  }
 </script>
  </head>
   <body>
    <div id = "div1" ondrop =" drop(event)" ondragover= "allowDrop
         (event)"> </div>
     <br>
    <img id= "drag1" src= "water.gif" draggable ="true"
         ondragstart = "drag(event)" width ="336" height ="69" >

   </body>
```

drag and drop between divisions



ex:
```
<head>
<style>
#div1,#div2
{
  width : 100px; height : 95 px;
  border: 1px solid #FF00FF ;
}
</style>
<script type= "text/javascript">
function allow Drop (ev)
{
  ev. prevent Default ( );
}
function Drag (ev)
{
 ev. dataTransfer. setData ( "text/html" , ev.target.id);
}
function drop (ev)
{
```

```
<body>
<div id="div1" ondrop="drop(event)" ondragover=" allowdrop
        (event)">
 <img id= " drag1" src="html5.png" draggable="true"
    ondragstart = "drag (event)"  width="100 px" height="50px">
   </div>
<div id= "div2" ondrop ="drop (event)" ondragover="allowdrop
        (event)"> </div>
  </body>
```

Task 1 → Gmail login

Task 2 → Facebook Registration

Task 3 → Naukri registration

Task 4 → Gmail Registration page

~~Tas 5 → Google & Bing search page~~

ksubbaraju.tm@gmail.com