

HTML AND CSS TUTORIAL

In html there are two types of tag

- Container tag
- Stand alone tag

Basic layout

```
<html>
  <head>
    //it store meta data
    <title>My first page</title>
  </head>
  <body>
    Hello World! this is my first program
  </body>
</html>
```

How to validate the html code

- Search w3c validator browser in google
- Click validate by upload file
- And check

Correct format is

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="Author" content="Sathish">
    <meta name="Description" content="My first page is super">
    <title>My first page</title>
  </head>
  <body>
    Hello World! this is my first program
```

```
</body>  
</html>
```

How to add icon in html

`<link rel="icon" href="green.jpg" type="image/x-icon">` in head tag

Heading tag

There are six types of heading in html

```
<h1>Heading Tag - 1</h1>  
<h2>Heading Tag - 2</h2>  
<h3>Heading Tag - 3</h3>  
<h4>Heading Tag - 4</h4>  
<h5>Heading Tag - 5</h5>  
<h6>Heading Tag - 6</h6>
```

Text Formatting Tag

To align paragraph left or right or center or justify

```
<p align="center"></p>  
<b>This is bold</b>  
<i>This is italic</i>  
<s>This is strike</s>  
<strong>This is strong same as bold</strong>  
<small>This is Small</small>  
<tt>Type Writing text format</tt>  
<u>Underline format</u>  
<sub>Subscript</sub>  
<sup>Superscript</sup>  
<pre>same output  
in editor</pre>  
<br> //Break tag
```

font tag

```
<font face="algerian" size = '5' color = "red">  
Any thing write as display in given font  
</font>
```

In HTML size is 1 to 10

Image Tag

```

```

Suppose image does not exit then work this
attribute

Table tag

```
<tr>Table Row </tr>  
<th>Table Heading</th>  
<td>Table data</td>
```

Some Example

Cellpadding cell around space

Cellspacing border around space

Colspan merge column

Border border in table

Rowspan merge row

```
<body>  
<h1>Table tag</h1>  
<table border="5px" cellpadding = "10px" cellspacing = "10px">  
  <tr>  
    <td colspan="2"> Student List</td>  
  </tr>  
  <tr>  
    <th>SNO</th>  
    <th>Name</th>
```

```

</tr>
<tr>
    <td>sathish</td>
    <td>17</td>
</tr>
</table>
<h1>OtherTable</h1>
<table border="2px" cellpadding = "10px" cellspacing = "10px">
    <tr>
        <td colspan="6"><h1>School Time Table</h1></td>

</tr>
<tr>
    <th>DAYS</th>
    <th>1</th>
    <th>2</th>
    <th rowspan="6">B<br>R<br>E<br>A<br>K</th>
    <th>3</th>
    <th>4</th>
</tr>
<tr>
    <td>MON</td>
    <td>ENG</td>
    <td>TAM</td>
    <td>MAT</td>
    <td>SCI</td>
</tr>
<tr>
    <td>TUE</td>
    <td>ENG</td>
    <td>TAM</td>
    <td>MAT</td>
    <td>SCI</td>
</tr>
<tr>
    <td>WED</td>
    <td>ENG</td>
    <td>TAM</td>

```

```
        <td>MAT</td>
        <td>SCI</td>
    </tr>
    <tr>
        <td>THU</td>
        <td>ENG</td>
        <td>TAM</td>
        <td>MAT</td>
        <td>SCI</td>
    </tr>
    <tr>
        <td>FRI</td>
        <td>ENG</td>
        <td>TAM</td>
        <td>MAT</td>
        <td>SCI</td>
    </tr>
</table>
</body>
```

Output:

Table tag

Student List	
SNO	Name
sathish	17

OtherTable

School Time Table					
DAYS	1	2	B R E A K	3	4
MON	ENG	TAM		MAT	SCI
TUE	ENG	TAM		MAT	SCI
WED	ENG	TAM		MAT	SCI
THU	ENG	TAM		MAT	SCI
FRI	ENG	TAM		MAT	SCI

List tag

- Order list
- Unorder list

<!--

unorder list type
disc
square
circle

-->

<ul type = "dsic">

```

        <li>Item-1</li>
        <li>Item-2</li>
        <li>Item-3</li>
        <li>Item-4</li>
        <li>Item-5</li>
    </ul>
    <!--
        order list type
        a
        A
        i
        I
    -->
    <ol type = "i">
        <li>Item-1</li>
        <li>Item-2</li>
        <li>Item-3</li>
        <li>Item-4</li>
        <li>Item-5</li>
    </ol>

```

Marquee tag

- Generally used in animation like
- That is photo is movement
- **direction**
 - left is default
 - right
 - up
 - down
- **behavior**
 - scroll is default
 - slide

- alternate

```
<marquee bgcolor = "orange" height = "50px" width = "300px" loop = "4"
  behavior="alternate" direction="left" scrollamount="10" scrolldelay =
"50">
```

```
     or any content
```

```
</marquee>
```

scrollamount content move by pixel by pixels, default move by 6 pixels (Means Speed)

scrolldelay means slow

Anchor tag

- href stands for hypertext reference
- it is used to navigate one page to another page
- self is default load in current page
- _blank open in other tab
- Also open in frame (inline fram iframe)

```
<a href="02_HeadingTag.html" >heading</a> self
```

```
<a href="08_MarqueeTag.html" target = "_blank">Marquee</a>
```

other tab

```
<a href="07_ListTag.html" target = "box">List Tag</a> open in
frame
```

```
<iframe src="" name="box" frameborder="0" height = "500px"
width="500px">
```

```
</iframe>
```

Open YouTube videos

Steps

- click share option
- click embed
- copy the code
- and paste


```

        <iframe width="560" height="315"
src="https://www.youtube.com/embed/0es56FoYABc" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay; clipboard-
write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>

```

Form elements

```

<fieldset>    design that is box
    <legend>Registration Form</legend>    Title
    <form >    any form element should be placed in form tag
        <label >Name</label>
        <input type="text">
        <br>
        <hr>    ruler
        <label >Password</label>
        <input type="password">
        <!--Combo Box-->
        <label >Select Course</label>
        <select >    denote combo box
            <option value="1">C</option>
            <option value="2">C++</option>
            <option value="3">Java</option>
        </select>
        <select multiple>
            <option value="1">C</option>
            <option value="2">C++</option>
            <option value="3">Java</option>
        </select>
        <br>
        <input type="date">
        <br>
        <textarea cols="30" rows="5"></textarea>
        <br>
        <!--name = "Gender" is common name -->
        <input type="radio" name = "Gender">Male
        <input type="radio" name = "Gender">Female
        <br>
        <input type="checkbox">I agree

```

CSS COMPLETE TUTORIAL

Intro

What is CSS ?(Cascading style sheets)

CSS – Cascading Style Sheet

Style sheets are a collection of formatting rules

How to link CSS rules in websites ?

Three types to link CSS rules in website

- **Inline style** (Inside the tag to style)
- **Internal style** (Inside the head tag)
- **External style** (External *.CSS file)

Inline style

Inline – CSS rules by using style attribute in HTML elements

```
<p style="color:orange;text-align: justify;">
    -----
    -----
</p>
<p style="color:orange;text-align: right;">
    -----
    -----

</p>
```

Internal style

```
<head>
    <style>
        h1{
            color:red;
            background: black;
        }
        p{
            color:blue;
        }
    </style>
</head>
```

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

External style

```
<head>
  <title>Tutor Joes</title>
  <!--rel - relative-->
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
```

```
style.css
h1{
  color:blue;
}
p{
  color:red;
}
```

Font Properties

Picture

```
<head>
  <link href="https://fonts.googleapis.com/css?family=Pacifico"
rel="stylesheet">
  <style>
    h1{
      font-family: 'Pacifico', cursive;
      font-weight:normal;
      font-variant: small-caps;
      color:rgb(0, 255, 17);
    }
    p{
      font-style:italic;
      font-size: 25px;
    }

    span{
```

```

        font: bold 12pt Arial;
        /* font-weight: bold;
        color: aqua; */
    }
</style>
</head>

```

Background Properties

picture

```

html{
    background:white;
}
body{
    /* url - uniform resource locator */
    background: teal;
    background-image: url(tom.png);
    color:black;
    background-repeat: no-repeat;
    /* background-repeat: repeat-x; */
    /* background-repeat: repeat-y; */
    background-size: 300px 300px;        width height
    /* bottom center, top center,top right,top left */
    background-position: bottom;
    /* to cover the image in case small image */
    background-size: cover;
    height: 800PX;
    width: 800px;
}

```

Background size properties

```

div{
    width: 900px;
    height: 500px;
    /* background-color: teal; */
    background: teal url(jerry.png) no-repeat;
    /* default property */
}

```

```
background-size: auto;

background-size: 200px;
/* width    height */
background-size: 400px 200px;
/* Some issue is created */
background-size: 100%;
/* solution */
background-size: cover;
background-size: contain;

padding: 10px;
}
```

Background attachments

```
div{
    color: khaki;
    font-size: 25px;
    font-weight: bold;
    padding: 10px;
    background: aliceblue url(natural.webp) no-repeat;
    background-size: cover;
    /* Scroll - default */
    background-attachment: scroll;
    background-attachment: fixed;
}
```

Background Origin

```
div{
    background: white;
    width: 800px;
    height: 600px;
    padding: 25px;
    border: 20px dashed black;
    background-image: url(natural.webp);
}
```

```

background-repeat: no-repeat;
/* padding-box is Default */
background-origin: padding-box;
background-origin: content-box;
background-origin: border-box;
}

```

Background Clip

```

div{
width: 800px;
height: 400px;
background-image: url(natural.webp);
background-repeat: no-repeat;
background-size: cover;
font-size: 100px;
font-family: impact;    bold
text-align: center;
color: transparent;
background-clip: text;
/* For chrome browser use webkit prefix use pannanum in
case */
-webkit-background-clip: text;
}

```

Background gradient

```

div{
width: 800px;
height: 400px;
/* gradient type
linear-gradient
repeating-linear-gradient
radial-gradient
repeating-radial-gradient
conic gradient
repeating-conic gradient */
height: 600px;
width: 800px;

```

```

background-color: red;
/* it used to show multi color
it also allow hexadecimal,rgb value
don't use background-color only use background*/
background:linear-gradient(red,blue) ;
background:linear-gradient(red,blue,black) ;

background:linear-gradient(to right,red,blue,black) ;
background:linear-gradient(to left,red,blue,black) ;
background:linear-gradient(to top,red,blue,black) ;
background:linear-gradient(to bottom,red,blue,black)
;

background:linear-gradient(to bottom
right,red,blue,black) ;
background:linear-gradient(to bottom
left,red,blue,black) ;
background:linear-gradient(to top
left,red,blue,black) ;
background:linear-gradient(to top
right,red,blue,black) ;

background:linear-gradient(45deg,red,blue,black) ;
background:linear-gradient(-45deg,red,blue,black) ;

background:linear-gradient(-45deg,red 50%,blue 20%) ;

background: repeating-linear-gradient(to top,red
20%,orange 30%);

/* In circle or ellipse form */
background: radial-gradient(red,yellow);
background: radial-gradient(red,yellow,green);
background: radial-gradient(red 5%,yellow 15%,green
60%);
background: radial-gradient(circle,red 5%,yellow
15%,green 60%);

```



```

        background: radial-gradient(closest-side at 50% 50%,red
5%,yellow 15%,green 60%);
        background: radial-gradient(closest-side at 80% 50%,red
5%,yellow 15%,green 60%);
        background: radial-gradient(closest-side at 80% 80%,red
5%,yellow 15%,green 60%);

        background: radial-gradient(farthest-side at 80%
80%,red 5%,yellow 15%,green 60%);

        background: radial-gradient(closest-corner at 80%
80%,red 5%,yellow 15%,green 60%);
        background: radial-gradient(farthest-corner at 80%
80%,red 5%,yellow 15%,green 60%);

        background: repeating-radial-gradient(red,orange,black);
        background: repeating-radial-gradient(red ,orange
10%,black 15%);

        background: conic-gradient(red,yellow,black);
        border-radius: 50%;
        background: conic-gradient(from 90deg,red,yellow,black);
        background: conic-gradient(red,yellow,black);
        background: conic-gradient(red 0deg,red 90deg,yellow
90deg,yellow 180deg, black 180deg,black 270deg);

        background: conic-gradient(at 40% 60%,red,yellow,black);
        background: repeating-conic-gradient(red,yellow 10%
,black 20%);
        border-radius: 60%;
    }

```

Background blend

```
div{
```

```

        width: 500px;
        height: 500px;
        border: 2px black;
        background-image: linear-
gradient(red,yellow),url(shoe.jpeg);
        background-size: cover;
        /* Default value normal */
        background-blend-mode: normal;
        background-blend-mode: screen;
        /* mostly used overlay */
        background-blend-mode: overlay;
        background-blend-mode: darken;
        background-blend-mode: hard-light;
        background-blend-mode: soft-light;
        background-blend-mode: multiply;
        background-blend-mode: difference;
        background-blend-mode: exclusion;
        background-blend-mode: hue;
        background-blend-mode: color-burn;
        background-blend-mode: color-dodge;
        background-blend-mode: color;
        background-blend-mode: lighten;
        background-blend-mode: luminosity;
        background-blend-mode: saturation;
    }

```

Opacity

```

div{
    width: 800px;

    border: 2px solid #333;
    padding: 10px;
    /* opacity value give from 0 to 1
    it means transparency */
    /* opacity: 0.8; */

}
img{
    opacity: 0.4;
}

```

```
        /* filter: alpha(opacity=90%); */
    }
}
```

Text properties

```
b{
    /* sub,super scribt */
    vertical-align: sub;
    vertical-align: super;
}
body{
    height:1000px;
    width:800px;
    background:white;
}
h1{
    letter-spacing: 5px;
    text-decoration: line-through;
    text-decoration: overline;
    text-decoration: underline;
    text-transform: uppercase;
    text-align: center;
}
p{
    text-align: justify;
    text-indent: 50px;
    line-height: 30px;
}
}
```

List properties

```
ul li{
    /* disc
    square
    circle
    decimal
    decimal-leading-zero
    lower-roman
    upper-roman
```

```






lower-alpha
upper-alpha
lower-greek
lower-latin*/

/*list-style-type: decimal;
list-style-type: square;
list-style-image: url(natural.webp); */

list-style-type: none;
}

```

List properties

-  Home
-  Product
-  blog
-  download
-  contact

```

ul li::before{
content: " ";
display: inline-block;
margin-right: 10px;
height: 10px;
width: 10px;
background-image: url(natural.webp);
background-size: cover;
/* background-color: aqua; */
}

```

More about list

```

ul li{
list-style-type: disc;
list-style-type: square;
list-style-type: circle;
list-style-type: decimal;
list-style-type: decimal-leading-zero;
list-style-type: lower-roman;
list-style-type: upper-alpha;
list-style-type: lower-greek;
list-style-type: lower-latin;
}

```

```
}
```

List-1

- Home
- Product
- blog
- download
- contact

List-2

- Home
- Product
- blog
- download
- contact

More about list

```
#two li{  
    /* list-style-type: square; */  
    /* list-style: none;  
       it is use in multi-level list*/  
    list-style: outside; /*Default*/  
    list-style: inside;  
}
```

Margin properties

```
div{  
    /* Margin */  
    /* 1.Margin - Individual Sides  
       margin-top: value;  
       margin-right: value;  
       margin-bottom: value;  
       margin-left: value;  
       2.Margin - Shorthand property  
       margin: value (all side);  
       margin : top right bottom left;  
       margin: vertical (Top, Bottom) horizontal  
(Left, Right);  
       margin: top, horizontal, bottom;*/  
    margin: 75px;  
    /* specify each side */  
    margin-top: 40px;  
    margin-bottom: 30px;  
    margin-left: 10px;
```

```

margin-right: 70px;
/* specify side in single line
margin : top,right,bottom,left */
margin: 70px 70px 70px 70px;
margin: 10px 70px 30px 70px;
/* speccify horizontal and vertical */
margin: 50px 70px;
/* margin: top,horizontal,bottom; */
margin: 30px 50px 10px;

}

```

Padding properties

```

div{
    /* margin denoted as outer space
padding denoted as inner space ; */
margin: 30px 50px 10px;
/* Padding */
padding: 10px;
padding-left: 10px;
padding-right: 20px;
padding-bottom: 30px;
padding-top: 40px;
padding: 40px 20px;
padding: 40px 20px 30px;
/* 1.Padding - Individual Sides
    Padding-top: value;
    Padding-right: value;
    Padding-bottom: value;
    Padding-left: value;
2.Padding - Shorthand property
    Padding: value (all side);
    Padding : top right bottom left;
    Padding: vertical (Top, Bottom) horizontal
(Left, Right);
    Padding: top,horizontal,bottom;*/

```

```
}
```

Border properties

picture

```
div{  
    background: teal;  
    width: 300px;  
    height: 300px;  
    /* To specify the particular side */  
    border-top-width: 10px;  
    border-top-color: red;  
    /* border-top-style: double; */  
    /* border-top-style: solid; */  
    border-top-style: groove;  
  
    border-left-width: 10px;  
    border-left-color: green;  
    border-left-style: dotted;  
  
    /* in single line */  
    border: 10px solid red;  
}
```

Display properties

```
ul li{  
    display: inline;  
}  
b{  
    display: block;  
    display: inline-block;  
    /* display: none; */  
}
```

Table styling

```
<!DOCTYPE html>
<html lang="en">
<html>
  <head>
    <title>Tutorials</title>
    <style>
      /* in single line border */
      table{
        border-collapse: collapse;
      }
      table tr:nth-of-type(even){
        background-color: greenyellow;
      }
      table tr:nth-of-type(odd){
        background-color: rgb(100, 199, 238);
      }
      table td:nth-of-type(odd){
        background-color: aqua;
      }
    </style>
  </head>

  <body>

    <h1>Display table in HTML</h1>
    <table width="50%"
      border="1px"
      cellpadding="10px"
      cellspacing="10px"
      >
      <caption>Friends</caption>
      <colgroup>
        <col style="width: 150px;">
        <col style="width: 50px;">
        <col style="width: 100px;">
      </colgroup>
      <thead>
```



```

        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>City</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>sathish</td>
            <td>17</td>
            <td>salem</td>
        </tr>
        <tr>
            <td>reeghan</td>
            <td>16</td>
            <td rowspan="2">madurai</td>
        </tr>
        <tr>
            <td>raja</td>
            <td>18</td>
        </tr>
        <tr>
            <td>jaswin</td>
            <td>18</td>
            <td>coimbature</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="3">Noyes Friends</td>
        </tr>
    </tfoot>
</table>

```

Display table in HTML

Friends		
Name	Age	City
sathish	17	salem
reeghan	16	madurai
raja	18	
jaswin	18	coimbature
Noyes Friends		

```

        <td>raja</td>
        <td>18</td>
        <td>coimbature</td>
    </tr>
    <tr>
        <td>jaswin</td>
        <td>18</td>
        <td>coimbature</td>
    </tr>
    <tfoot>
        <tr>
            <td colspan="3">Noyes Friends</td>
        </tr>
    </tfoot>
</tbody>
</table>

```

```
    </table>
  </body>
</html>
```

Display table properties

```
<!DOCTYPE html>
<html lang="en">
<html>
  <head>
    <title>Tutorials</title>
    <style>
      .table{
        margin-top: 100px;
        margin-left: 50px;
        /* border: 1px solid black; */
        width: 500px;

        display: table;
      }

      .row{ /*tr*/
        display: table-row;
      }
      .col{ /*td*/
        border: 1px solid orangered;
        padding: 10px 30px;
        display: table-cell;
      }
      .header{ /*thead*/
        display: table-header-group;
        font-size: 25px;
        font-weight: bold;
        text-align: center;
        background-color: orangered;
        color: white;
      }
      .footer{ /*tfoot*/
        display: table-footer-group;
```

```

    }
    .caption{
        display: table-caption;
        font-size: 30px;
        font-weight: bold;
        text-align: center;
        color: orangered;
        padding: 10px;

    }
    .col-group{ /*colgroup*/
        display: table-column-group;
    }
    .tbl-col{ /*col*/
        display: table-column;
    }
    .col2{
        /* Only allow two property width and background-color
*/
        width: 30px;
        background-color: antiquewhite;

    }
    .row-group{
        display: table-row-group;
        color: orange;
    }
</style>
</head>

<body>
    <h1>Display Table in CSS</h1>
    <div class="table">
        <div class="caption">
            Student Details
        </div>
        <div class="col-group">
            <div class="tbl-col">col-1</div>

```

```
        <div class="tbl-col col2">col-1</div>
        <div class="tbl-col">col-1</div>
    </div>
    <div class="header">
        <div class="row">
            <div class="col">Name</div>
            <div class="col">Age</div>
            <div class="col">City</div>
        </div>
    </div>
    <div class="row-group">
    <div class="row">
        <div class="col">sam</div>
        <div class="col">17</div>
        <div class="col">salem</div>
    </div>
    <div class="row">
        <div class="col">ram</div>
        <div class="col">20</div>
        <div class="col">chennai</div>
    </div>
    <div class="row">
        <div class="col">ravi</div>
        <div class="col">25</div>
        <div class="col">salem</div>
    </div>
    <div class="row">
        <div class="col">sri</div>
        <div class="col">18</div>
        <div class="col">madurai</div>
    </div>
    <div class="row">
        <div class="col">mohan</div>
        <div class="col">24</div>
        <div class="col">bombay</div>
    </div>
</div>
    <div class="footer">
```

```

        <div class="row">
            <div class="col">Total student</div>
            <div class="col"></div>
            <div class="col">5</div>
        </div>
    </div>
</div>

</body>
</html>

```

Display Table in CSS

Student Details		
Name	Age	City
sam	17	salem
ram	20	chennai
ravi	25	salem
sri	18	madurai
mohan	24	bombay
Total student		5

Output:

Parent and child

```

<body> <!--Parent-->
    <h1>Sathish</h1> <!--Child-->
    <h3>Parent And Child in CSS</h3>
    <div> <!--Child but parent in own child-->
        <h4>Heading</h4> <!--child in div grand Child in body-->
        <p>paragraph</p>
    </div>

```

```
</div>
</body>
```

Style preferences

Default

- id selectors first preference
- class selectors second preference
- element selectors third preference

Suppose I change the rule(that is break the rule) --> element sectors is first preference

use !important

```
p{
    color:green !important;
}
```

Anchor tag properties

```
a[title]{
    color: blanchedalmond;
}
a[title="Link-1"]{
    color: orange;
}

a[href]{
    color:aqua
}
a[href="text.png"]{
    color:aqua
}
```

```

/* start with text change color(add symbol ^) */
a[href^="test"]{
    color: blueviolet;
}
/* end with text change color(add symbol ^) */
a[href$="#"]{
    color: red;
}

```

Access child class

```

ul li:first-child{
    color: red;
}

ul li:nth-child(2){
    color: yellowgreen;
}

ul li:nth-child(odd){
    color: yellowgreen;
}

ul li:last-child{
    color: orange;
}

```

Border radius

```

.box{
    height: 150px;
    width: 150px;
    background: -webkit-linear-gradient(#e67e22,#e74c3c);
    border-radius: 150px; /*Circle*/
    /* border-radius: */
    border-radius: 0px 20px 0px 10px;

    border-radius: 10px 20px 50px;
}

```

```
border-radius: 10px 20px ;
```

```
}
```

Colors

mostly used

1) color format(eg: red)

2) hexadecimal

3) rgb(x,y,z)

rgb stands for red green blue

x,y,z belong 0 to 255

4) rgba(x,y,z)

rgb stands for red green blue alpha(that mean opacity(transparency))

eg:rgba(255,0,0,0.5)

use website adobe color picker

Box shadow

```
.box{
  width: 150px;
  height: 150px;
  background: orangered;
}
.shadow{
  /* syntax
      boxshadow: horizontal offset vertical
offset blur spread radius color
  */
  /* -webkit-box-shadow: 10px 10px 10px 0px
  rgba(0,0,0,0.5);
  -moz-box-shadow: 10px 10px 10px 0px rgba(0,0,0,0.5);
  */
  box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.15);
}
```


Chess board in folder

CSS Layout Design

- Box Model in CSS
- Normal Document Flow
- Display Properties
- Float properties
- Position
 - **Static**
 - **Relative**
 - **Absolute**
 - **Fixed**
 - **Sticky**

Box model in CSS

Box Model

```
<html>
  <head>
    <title>Tutorials</title>
    <link rel="stylesheet" type="text/css" href="30(1)Style.css">
  </head>

  <body>
    <h1>Box Model</h1>
    <div class="box">para
  </div>
    <div class="box">para
  </div>
  </body>
</html>
```

Style.css

```
*{
  margin: 0px;
  padding: 0px;
}

body{
  background-color: #a1a1a1;
  color: white;
}

.box{
  height: 150px;
  width: 400px;
  text-align: justify;
  background: teal;
  margin: 50px;
  padding: 10px;
  border: 10px double white;
}
```

Normal document flow

```
<html>
  <head>
    <title>Tutorials</title>
    <link rel="stylesheet" type="text/css" href="31(1)Style.css">
  </head>

  <body>
    <h1>Normal Document Flow</h1>
    <div class="container">
      <p>para</p>
      <p>para</p>
      <p>para</p>
    </div>
  </body>
</html>
```

Style.css

```
*{
  margin: 0px;
  padding: 0px;
}
body{
  background-color: #a1a1a1;
}

.container{
  background: #fff;
  max-width: 800px;
  margin: 0px auto;
  padding: 20px;
}
p{
  /* Normal document flow refers sequence style */
  text-indent: 25px;
  background: teal;
  color: white;
```

```
margin-top: 10px;  
padding: 10px;
```

```
}
```

Position

Html document

```
<html>  
  <head>  
    <title>Tutorials</title>  
    <link rel="stylesheet" type="text/css" href="32(1)style.css">  
    <link rel="stylesheet" type="text/css"  
href="32(2)Position.css">  
  </head>  
  
  <body>  
    <h1>Position Properties in CSS</h1>  
    <div class="main">  
      Main Box  
      <div class="box1 box">Box-1</div>  
      <div class="box2 box">Box-2</div>  
      <div class="box3 box">Box-3</div>  
    </div>  
  </body>  
</html>
```

Style.css

```
.main{  
  border: 5px solid gray;  
  width: 500px;  
  min-height: 500px;  
  font-weight: bold;  
}  
  
.box{  
  width: 100px;  
  height: 100px;
```

```
    color: white;
    padding: 10px;
    font-weight: bold;
}
.box1{
    background-color: teal;
}
.box2{
    background-color: tomato;
}
.box3{
    background-color: yellowgreen;
}
```

Position.css

Sticky

- Its default
- it regular document flow

Relative

```
.main{

}

.box1{
    position: relative;
    left: 50px;
    top: 50px;
    bottom: 50px;
}

.box2{

}

.box3{
```

```
}
```

Absolute

```
.main{  
    position: relative;  
}  
  
.box1{  
    position: absolute;  
    /* display: none; */  
    /* top: 0; */  
    bottom: 0;  
    right: 0;  
    left: 100px;  
  
}  
  
.box2{  
  
}  
  
.box3{  
  
}
```

Fixed

```
.main{  
    position: relative;  
}  
  
.box1{  
    position: fixed;  
    top: 200px;  
    right: 200px;  
    /* left: 100px; */  
}  
.box2{
```

```
}  
  
.box3{  
  
}
```

Sticky

```
<!DOCTYPE html>  
<html lang="en">  
<html>  
  <head>  
    <title>Tutorials</title>  
    <link rel="stylesheet" type="text/css" href="32(1)style.css">  
    <link rel="stylesheet" type="text/css"  
href="32(2)Position.css">  
  </head>  
  
  <body>  
    <h1>Position Properties in CSS</h1>  
    <h2>sticky</h2>  
    <div id="header">Header</div>  
    <div id="navbar">  
      <ul>  
        <li>Home</li>  
        <li>About</li>  
        <li>Contact</li>  
        <li>Downloads</li>  
        <li>Products</li>  
      </ul>  
    </div>  
    <div id="box">  
      <p>big paragraph</p>  
    </div>  
    <div id="footer">Footer</div>  
  </body>  
</html>
```

Style.css

```
*{
    margin: 0;
    padding: 0;
}

html{
    width: 100%;
    height: 100%;
}

#header, #footer{
    width: 100%;
    height: 200px;
    background-color: yellowgreen;
    padding: 5px;
}

#navbar{
    width: 100%;
    height: 50px;
    background-color: tomato;
    padding: 5px;
    position: sticky;
    top: 0;
}

#box{
    width: 100%;
    padding: 10px;
}

#navbar ul li{
    display: inline-block;
    list-style-type: none;
    padding: 10px;
    line-height: 30px;
    color: white;
```



```
}
```

Z-index

Z-index is high that is first

Z-index is 0 that is last

Z-index is -ve that is outside the parent

```
<html>
<head>
  <title>Z-index</title>
  <style>
    .box{
      width: 800px;
      height: 500px;
      background: #ccc;
      position: relative;
    }
    .box div{
      height: 200px;
      width: 200px;
      position: absolute;

    }
    .box-1{
      background: teal;
      color: white;
      top:100px;
      left: 50px;
      z-index: 2; /*top*/
    }
    .box-2{
      background: orangered;
      color: white;
      top:150px;
      left: 100px;
      z-index: -1;/*outside the parent*/
    }
  </style>
</head>
<body>
  <div class="box">
    <div class="box-1">
    </div>
    <div class="box-2">
    </div>
  </div>
</body>
</html>
```

```

    }
    .box-3{
        background: purple;
        color: white;
        top:200px;
        left: 150px;
        z-index: 0;/*last*/
    }
</style>
</head>
<body>
    <h1>Z-index in CSS</h1>
    <div class="box">
        <div class="box-1">Box-1</div>
        <div class="box-2">Box-2</div>
        <div class="box-3">Box-3</div>
    </div>
</body>
</html>

```

Outline

```

div{
    width: 300px;
    border: 3px solid brown;    border ku outside
    padding: 30px;
    outline: 2px solid blue;

    outline-width: 3px;
    outline-style: dotted;
    outline-color: red;

    outline-offset: 3px;
}

```

Outline in CSS

Lorem, ipsum dolor sit amet consectetur
adipiscing elit. Voluptas iure dignissimos
laudantium nostrum ratione labore earum velit
minima harum. Culpa blanditiis quaerat
mollitia odio laudantium iure iusto quis
assumenda excepturi.

```
<!--  
    outline-width  
    outline -style  
        * solid  
        * dotted  
        * dashed  
        * double  
        * groove  
        * ridge  
        * inset  
        * outset  
        * none  
  
    outline-color  
    outline-offset  
-->
```

min-height & max-height

min-height denoted as minimum height(that is atleast height, more content are there, its expand)

max-height denoted as maximum height(that is atmost height, more content are there, it cannot expand)

max-width & max- width

It is similar to height

Box-Sizing

- Border-box
- Content-box is default

Picture

Picture

```
div{  
    width: 300px;  
    height: 300px;  
    background-color: aliceblue;  
    padding: 10px;  
    border: 10px solid black;  
    box-sizing: border-box;  
  
}
```

Overflow

```
div{  
    width: 500px;  
    height: 200px;  
    border: 5px solid green;  
    overflow: visible; /*Default*/  
    overflow: hidden;  
    overflow: scroll;  
    overflow: auto;  
    overflow-y: scroll; /*Manual*/  
    overflow-x: scroll;  
  
}
```

Column

```
div{  
    text-align: justify;
```

```

border: 5px solid black;
padding: 10px;
column-count: 3;
column-width: 150px;
column-gap: 50px;
/* column-rule: 3px; */
column-rule-width: 5px;
column-rule-color: red;
column-rule-style: solid;

/* in single line */
column-rule: 2px dashed red;
}
h2{
column-span: none; /*default*/
column-span: all;
}

```

Column in CSS

heading

Lorem ipsum dolor sit amet consectetur adipisicing elit. Architecto ipsa voluptatum reprehenderit nihil, excepturi sint quam officia consequatur ipsam error similique dolorem ab ad illum distinctio natus vero magni blanditiis ullam ratione vel labore atque libero! Ipsum debitis doloremque quod, repudiandae veritatis laborum. Impedit, sequi totam sapiente libero magni eligendi

assumenda quas, velit, unde earum temporibus porro quia ipsam! Nostrum quidem aliquam repellat laborum ratione cum ullam reprehenderit sit rerum ipsum tempora corporis officia dicta quis debitis accusantium labore reiciendis consequatur quam voluptatibus nulla, repudiandae nihil? Voluptas sed dicta consequatur magnam ab autem iure ipsa corrupti aut, hic facilis

molestias illo molestiae quis eaque pariatur doloremque adipisci excepturi quae similique ad. Magni commodi suscipit itaque. Aut non repudiandae itaque inventore nisi cum? Ab, modi hic quod, veritatis vel molestias voluptate aperiam eligendi in temporibus eos quam iste soluta quaerat, accusantium est. Sed qui maiores sapiente praesentium deserunt molestiae dolorum a.

CSS Units

regular usage

px

%

em

rem

vw

vh

vmax

vmin

Absolute units

- cm Centimeters
- mm Millimetewrs
- in Inches
- px pixels point size
- pt Points
- pc Picas

Relative Units

- **em**

1em = 16px browser default size

1em = base value 16px * 1 => 16px

2em = base value 16px * 2 => 32px

suppose setting change

2em = base value 10px * 2 => 32px

Also used decimal value

Conclusion:

- if you give em in font or other, then get size based upon browser(default is 16px) if parent does not exit

- Suppose parent are given, then get size based upon parent size
- rem root em
html is root element for all element
conclusion:
 - if you give rem in font or other, then get size based upon browser if does not exist html size and this unit not affected by parent
 - suppose html size are given, then get size based upon html size
- ch
- ex
- vw viewport width and vh viewport height

```
.box{
    height: 100vh;
    width: 100vw;
}
```

It used for full screen website

- vmax viewport maximum
- vmin viewport minimum
- % Percentage depending upon parent

CSS Selector

Simple Selectors

Element

- Particular element can be select
- Element selectors that is normal tag use

```
<h1>Content</h1>
```

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

Class

- Can select select group of selectors
- class selector represent dot(.)

```
<div class="box"></div>
```

```
.box{  
    height: 100px;  
    width: 100px;  
    background: teal;  
    margin-bottom: 10px;  
    /* important thing div is block level element but  
span is not block level element */  
    display: block;  
}
```

Id

- Id is unique that can be select particular unique element
- id selectors represent #

```
<h3 id="title">content </h3>
```

```
#title{  
    color: orange;  
    font-size: 25px;  
}
```

Universal

All the elements can be select(all tag)


```
*{  
  margin: 0;  
  padding: 0;  
}
```

Multiple selectors

```
h1,h3,.box{  
  color: green;  
}
```

Combinators Selectors

Descendant

```
.box p{  
  color: red;  
}
```

Direct child

child selectors only style in child(direct child)(Only select direct child)

```
.box > p{  
  color: red;  
}
```

Adjacent siblings

adjacent(nearest) or Nearby element can select

```
.box + p{  
  color: red;  
}
```

General siblings

that is brother

```
.box ~p{  
  color:red  
}
```

Attribute selectors

```
div[class]{
    background: gray;
}
div[id]{
    background: gray;
}
```

in particular class or id

```
div[class="inner"]{
    background: gray;
}

/* suppose use multiple */
/* For example -->inner and txt */
/* inner class property affected */
/* solution is symbol ~ */
div[class~="inner"]{
    background: gray;
}
.txt{
    color: white;
}
```

Pseudo Class selectors

:root pseudo selectors

- Must be add two hyphen –
- Eg: That is stored color in variable and used variable name

```
:root{
    --bgcolor: aliceblue;
    --txtcolor: teal;
}
h1{
```

```

        color: var(--txtcolor);
    }
    p{
        color: var(--txtcolor);
        background: var(--bgcolor);
    }

```

Pseudo class selectors | first-child | last-Child

- Child(Pseudo selectors) should have parent, then it is work

```

/* choosing first child */
li:first-child{
    color: orange;
}
/* choosing last child */
li:last-child{
    color: red;
}
/* Choosing specific selectors and first child or last child */
#list-1 li:first-child{
    color: teal;
}

```

Pseudo class selectors | nth-child() | nth-last-Child()

```

/* Choosing nth child */
li:nth-child(5){
    color: blue;
}
/* Choosing specific list nth child */
#list-1 li:nth-child(5){
    color: teal;
}
/* Choosing last child that is reverse order that is last element
is first element */

```

```

li:nth-last-child(1){
  color: red;
}
/* Also possible formula
2n
2n+1
3n-1 where n=0,1,2,3,4,5,6..... */
/* Choosing formula in nth-child()*/
li:nth-child(2n){
  color: blue;
}

```

Pseudo class selectors | nth-of-type() | nth-last-of-type ()

- Child need not have parent

```

p:nth-of-type(1){
  color:red
}
p:nth-last-of-type(1){
  color:blue
}
li:nth-of-type(1){
  color:red
}
li:nth-last-of-type(1){
  color:blue
}
/* also using odd, even, formula and useful in table styling */

```

Pseudo class selectors | :only-child()

- should have parent
- Parent should have only one child, then it works

```

li:only-child{

```

```
    color: teal;
}
```

Pseudo class selectors | :only-of-type()

- Need not have parent
- But only one element in document

```
li:only-of-type{
    color: teal;
}
p:only-of-type{
    color: red;
}
```

Pseudo class selectors | :first-of-type() | :last-of-type()

- It can select first or last element with parent or without parent

```
li:first-of-type{
    color:red;
}
li:last-of-type{
    color:green;
}
p:first-of-type{
    color:red;
}
p:last-of-type{
    color:green;
}
```

Pseudo class selectors | :empty

- Only select empty tag(eg : paragraph)
- Note that space is also character, it means have content

```
p:empty{
    padding: 10px;
}
```

```
border: 2px solid red;
}
```

Pseudo class selectors | :not

- It is used to select the tag based on not
- Eg select paragraph based on not class “para”

```
p.para{
    color: red;
}
p:not(.para){
    color: blue;
}
```

Pseudo class selectors | :lang

It is used to style language based

```
<head>
  <title>Element selectors</title>
  <style>
    p:lang(en){
      color: red;
    }
    p:lang(fr){
      color: blue;
    }
  </style>
</head>
<body>
  <h1>heading</h1>
  <p lang="fr">FRENCH PARAGRAPH</p>
  <p lang="en">ENGLISH PARAGRAPH</p>
</body>
</html>
```

Pseudo class selectors :link | :hover | :visited | :active

```
a{
```

```
        text-decoration: none;
    }
    a:hover{
        color: red;
    }
    a:active{
        color: orange;
    }
    a:visited{
        color: pink;
    }
}
```

Pseudo class selectors| :target

```
<html>
  <head>
    <title>Tutorials</title>
    <style>
      p{
        padding: 10px;
        background-color: aliceblue;
      }
      p:target{
        background-color: aquamarine;
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Target</h1>
    <a href="#para-1">para-1</a>
    <a href="#para-2">para-2</a>
    <a href="#para-3">para-3</a>
    <p id="para-1">paragraph</p>
    <p id="para-2">paragraph</p>
    <p id="para-3">paragraph</p>
  </body>
</html>
```

Output:

Target

[para-1](#) [para-2](#) [para-3](#)

paragraph

paragraph

paragraph

Form CSS Pseudo class selectors

```
<html>
  <head>
    <title>Tutorials</title>
    <style>
      /* Focus */
      #txt:focus{
        outline: none;
        border: 1px solid brown;
      }
      /* checkbox is tick then work */
      input[type="checkbox"]:checked{
        box-shadow: 0 0 0 3px red;
      }
      /* enabled */
      input[type="text"]:enabled{
        background-color: pink;
      }
      /* Disabled */
      input[type="text"]:disabled{
        background-color: red;
      }
      /* required */
      input[type="text"]:required{
        background-color: green;
      }
      /*Read write */
```



```

        input[type="email"]:read-only{
            background-color: gray;
        }
        /*Read only */
        input[type="email"]:read-write{
            background-color: plum;
        }
        /* invalid */
        input[type="text"]:invalid{
            border-color: red;
        }
        /* valid */
        input[type="text"]:valid{
            border-color: rgb(15, 225, 15);
        }
        /* Default in radiobutton */
        input[type="radio"]:default{
            box-shadow: 0 0 0 3px green;
        }
        /* Select default option */
        option:default{
            color: red;
        }
    }

</style>
</head>

<body>
    <h1>Form CSS Pseudo Class Selector</h1>

    <span>:focus</span>
    <input type="text" id="txt">
    <hr>

    <span>:checked</span>
    <input type="checkbox" id="cricket">
    <label for="cricket">Cricket</label>
    <input type="checkbox" id="football">

```

```

<label for="football">Football</label>
<input type="checkbox" id="tennis">
<label for="tennis">Tennis</label>
<!-- for attribute is important for attribute la entha
element target panirigalo atha kudukanum-->
<hr>

<span>:enabled :disabled :required :optional</span>
<br>
<input type="text">
<br>
<input type="text" disabled>
<br>
<input type="text" required>
<br>
<hr>
<span>:read-only :read-write</span>
<br>
<input type="email" > <!--read-write is default-->
<br>
<input type="email" read-only>
<br>
<span>:valid :invalid</span>
<br>
<input type="text" placeholder="Enter Username" pattern="[a-
z]*"> <!--Regular expression (pattern) -->
<hr>

<span>Default</span>
<br>
<input type="radio" name="gender" id="male" checked>
<label for="male">Male</label>
<input type="radio" name="gender" id="female">
<label for="female">Female</label>

<hr>
<select >
    <option value="">select</option>

```

Output:

```
:focus
```

```
:enabled :disabled :required :optional
```

```
:valid :invalid
```

 Male Female

<html>

<title>Tutorials</title>

<style>

#d1{

background-color: aliceblue;

}

#d2{

```
background-color: antiquewhite;
```

}

```

/* Now task */
/* d1 la irruka h2,h3,h4 select panuga d2 */

/* Normal Method */
#d1 h2,
#d1 h3,
#d1 h4{
    color: red;
}

/* Easy method */
#d1 :is(h2,h3,h4){
    color: blue;
}
/* Other Task */
/* d1 la irruka h2 and d2 la irruka h2 select panuga */

/* Normal Method */
#d1 h2,
#d2 h2{
    color: green;
}
/* Easy method */
:is(#d1,#d2) h2{
    color: brown;
}

/* Other new task */

/* d1, d2 la irruka h2 and paragraph select panuga */
:is(#d1,#d2) :is(h2,p){
    color: red;
}
/* Hover */
:is(#d1,#d2) h2: hover{
    color: chocolate;
}
</style>

```

```

</head>
<body>
  <h1>is() Pseudo Class Selector</h1>
  <div id="d1">
    <h2>H2 - Heading One</h2>
    <h3>H3 - Sub Heading - 1</h3>
    <p>paragraph</p>
    <h4>H4 - Sub Heading - 2</h4>
    <p>paragraph</p>
  </div>
  <hr>
  <div id="d2">
    <h2>H2 - Heading Two</h2>
    <p>paragraph</p>
  </div>
  <!-- This para is outside therefore cannot do any thing -->
  <p>
    <b>Outside</b> paragraph
  </p>
</body>
</html>

```

Output:

is() Pseudo Class Selector

H2 - Heading One

H3 - Sub Heading - 1

paragraph

H4 - Sub Heading - 2

paragraph

H2 - Heading Two

paragraph

Outside paragraph

Pseudo element selectors

::before

::after

::first-letter

::selection

::placeholder

```
<html>
  <head>
    <title>Tutorials</title>
    <style>
      /* Placeholder */
      input::placeholder{
        color: pink;
      }
      /* Selection */
      p::selection{
        background-color: #222f3e;
        color: white;
      }
      /* First letter pseudo selectors */
      P::first-letter{
        font-size: 40px;
        font-weight: bold;
        color: red;
      }
      p::first-line{
        color: blue;
      }

      .box{
        background-color: #333;
        color: white;
        width: 300px;
        height: 100px;
        position: relative;
```

```

    }
    /* Before and after acts a element
    speciality, both before and after selectors styling
individual
    before and after selectors inside block la irrukum(eg
box)
    position property do and style individual */
    .box::before{
        /* complusory content irrukanum may empty or some
information */
        content: '';
        width: 100%;
        height: 20px;
        background-color: #ff4757;
        position: absolute;
        top: 0;
        left: 0;
    }
    .box::after{
        /* compulsory content irrukanum may empty or some
information */
        content: '';
        width: 100%;
        height: 20px;
        background-color: #ff4757;
        position: absolute;
        bottom: 0;
        left: 0;
    }
</style>
</head>

<body>
    <h1>Pseudo element selectors</h1>
    <div class="box">Pseudo elements</div>
    <p>Paragraph <br> 2-line</p>
    <p>Paragraph <br> 2-line</p>
    <input type="text" placeholder="Enter Name">

```

```
        </body>
</html>
```

Output

Pseudo element selectors



Paragraph
2-line

Paragraph
2-line

TRANSITION IN CSS

- transition-property
- transition-duration
- transition-delay
- transition-timing-function
- transition

units	Description
ease	Start slow Fast End slowly(default)
linear	Maintain same speed from start to end
ease-in	Slow start and speed
ease-out	Slow end
ease-in-out	Slow start and end
step-start	
step-end	
step(4,end)	
cubic-bezier	We can create with own values Google search

Transitional properties
Background
Background-color
Background-position
Background-size
Backdrop-filter
border
border-block-end
border-block-end-color
border-block-end-width
border-block-start
border-block-start-color

border-block-start-width
border-bottom
border-bottom-color
border-bottom-left-radius
border-bottom-right-radius
border-bottom
border-bottom-color
border-bottom-width
border-color
border-end-end-radius
border-end-start-radius
border-image-outset
border-image-slice
border-image-width
border-inline-end
border-inline-end-color
border-inline-end-width
border-inline-start
border-inline-start-color
border-inline-start-width
border-left
border-left-color
border-left-width
border-radius
border-right
border-right-color
border-right-width
border-start-end-radius
border-start-start-radius
border-top
border-top-color
border-top-left-radius
border-top-right-radius
border-top-width
border-width
bottom
box-shadow
caret-color
clip

clip-path
color
column-count
column-gap
column-rule
column-rule-color
column-rule-width
column-width
columns
filter
flex
flex-basis
flex-grow
flex-shrink
font
font-size
font-size-adjust
font-stretch
font-variation-settings
font-weight
gap
grid-column-gap
grid-gap
grid-row-gap
grid-template-columns
grid-template-rows
height
inline-size
input-security
inset
inset-block
inset-block-end
inset-block-start
inset-inline
inset-inline-end
inset-inline-start
left
letter-spacing
line-clamp

line-height
margin
margin-block-end
margin-block-start
margin-bottom
margin-inline-end
margin-left
margin-right
margin-top
mask
mask-border
mask-position
mask-size
max-block-size
max-height
min-inline-size
max-lines
max-width
min-block-size
min-height
min-inline-size
min-width
object-position
offset
offset-anchor
offset-distance
offset-path
offset-position
offset-rotate
opacity
order
outline
outline-color
outline-offset
outline-width
padding
padding-block-end
padding-block-start
padding-bottom

padding-inline-end
padding-inline-start
padding-left
padding-right
padding-top
perspective
perspective-origin
right
rotate
row-gap
scale
shape-margin
scroll-margin-block
scroll-margin-block-end
scroll-margin-block-start
scroll-margin-inline
scroll-margin-inline-end
scroll-margin-inline-start
scroll-margin-left
scroll-margin-right
scroll-margin-top
scroll-padding
scroll-padding-block
scroll-padding-block-end
scroll-padding-block-start
scroll-padding-bottom
scroll-padding-inline
scroll-padding-inline-end
scroll-padding-inline-start
scroll-padding-left
scroll-padding-right
scroll-padding-top
scroll-snap-coordinate
scroll-snap-destination
scrollbar-color
shape-image-threshold
shape-margin
shape-outside
tab-size

text-decoration
text-decoration-color
text-decoration-thickness
text-emphasis
text-emphasis-color
text-indent
text-shadow
text-underline-offset
top
transform
transform-origin
translate
vertical-align
visibility
width
word-spacing
z-index
zoom

```
div{
  width: 300px;
  height: 300px;
  background-color: orangered;
  /* transition-property: width,height;background;
  transition-duration: 0.5s,0.5s,1s;
  transition-timing-function: ease-in-out; /*ease is default*/
  transition-timing-function: linear; */

  /* transition-property:all;
  transition-duration: 0.5s;
  transition-timing-function: ease-in-out;
  transition-delay: 0.5s; */
  border:5px solid black;

  /* transition:all 0.5s ease-in 1s; */
```

```
    transition-property:all;
    transition-duration: 2s;
    transition-timing-function:cubic-bezier(.96,.29,.48,1.13);
    transition-delay: 0.1s;
}
div:hover{
    width: 600px;
    height:600px;
    background:rebeccapurple;
    border:5px dashed red;
}
```

Flex Box

- Flex box is not a framework or anything like that. Its is a new box model and no extra dependencies required.
- Flex stands for flexible, adaptive

Flex box is the alternative for

display: block;

display: inline;

display: inline-block;

Advantages of flex box

- Easy to align element in vertical centering
- Flex box can have display order reversed or rearranged at the style layer

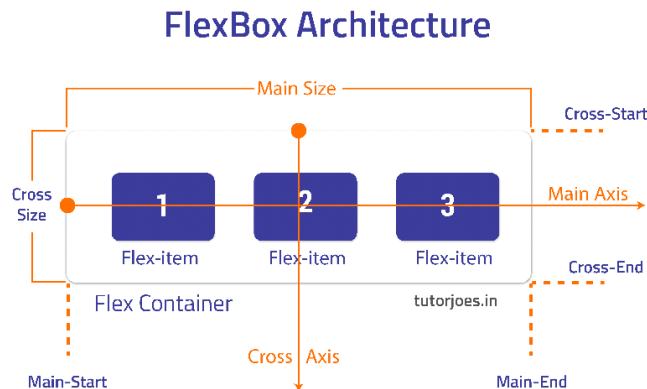
- One of the main advantages of flexbox is the ability to fill extra space without the need to use JavaScript

Main Axis and Cross axis

- Main axis represents x axis
- Cross axis represents y axis

Vertical centering

Flex Box Architecture



Flex Box Properties

Flexbox Container Properties

- display
- flex-direction
- flex-wrap
- justify-content
- align-items
- align-content

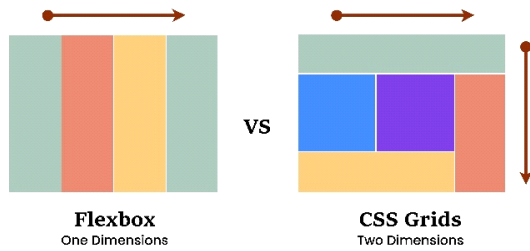
Flexbox Item Properties

- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self
- order
- **display:** This property specifies the display behavior of an element. To use Flexbox, you need to set the display property to flex or inline-flex.
- **flex-direction:** This property specifies the direction of the Flexbox layout. You can set it to row (horizontal), column (vertical), row-reverse, or column-reverse.
- **flex-wrap:** This property specifies whether the Flexbox items should wrap or not. You can set it to nowrap (default), wrap, or wrap-reverse.
- **justify-content:** This property specifies how the Flexbox items are distributed along the main axis. You can set it to flex-start, flex-end, center, space-between, space-around, or space-evenly.
- **align-items:** This property specifies how the Flexbox items are aligned along the cross axis. You can set it to flex-start, flex-end, center, stretch, or baseline.
- **align-content:** This property specifies how the Flexbox lines are aligned when there is extra space in the container. You can set it to flex-start, flex-end, center, space-between, space-around, or stretch.
- **flex-grow:** This property specifies how much an item should grow relative to the other items in the Flexbox container.
- **flex-shrink:** This property specifies how much an item should shrink relative to the other items in the Flexbox container.
- **flex-basis:** This property specifies the initial size of an item before any remaining space is distributed.

CSS grid layout

- CSS grid is a powerful tool allows for **two dimensional layouts** to be created on the web
- By using **rows and columns**, grid makes it easier to design **web page layouts, align and distribute space** among **items** in a grid
- Flexbox is for **one dimensional layout**

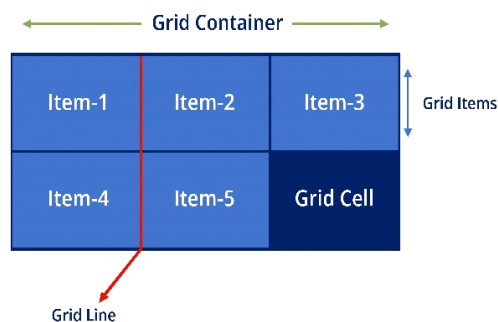
Flexbox VS CSS Grids



CSS Grids Terminology

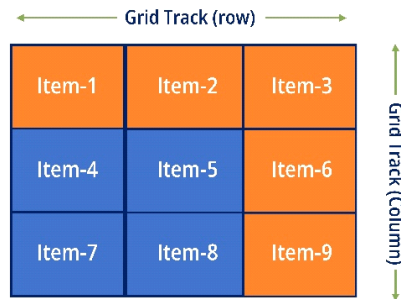
CSS Grid is a powerful layout system that allows web developers to create complex and responsive layouts with ease. Here are some of the key terms you need to know when working with CSS Grid:

- **Grid Container:** The element that contains all the items being placed in the grid. It's typically defined by setting the display property to grid or inline-grid.
- **Grid Cell:** The intersection of a row and a column, where a grid item is placed.



- **Grid Item:** The individual element that is placed within a grid cell.
- **Grid Line:** The horizontal or vertical line that separates the grid cells. Grid lines can be referred to by their number, starting from 1 or by their name.

- Grid Track: The space between two adjacent grid lines, either horizontal or vertical.



- Grid Template: Defines the structure of the grid by specifying the number of rows and columns, the size of the grid tracks, and the placement of grid items.
- Grid Template Rows/Columns: Specifies the number and size of rows or columns in the grid. You can use fixed sizes like px or relative sizes like fr to create flexible layouts.
- Grid Gap: The space between two adjacent grid tracks. It can be set using the grid-gap property.
- Grid Area: The total space covered by a grid item, spanning one or more cells.



- Grid Auto Placement: Specifies how items are placed on the grid when their location is not explicitly defined. It can be set to dense or sparse.

These are just a few of the key terms used in CSS Grid. Learning these terms will help you understand and use CSS Grid more effectively in your web development projects.

