



Learning CSS (Cascading Style Sheet)

PRAVEEN NAIR

Cascading Style Sheet (CSS-3)

CSS describes how an HTML file should be display.

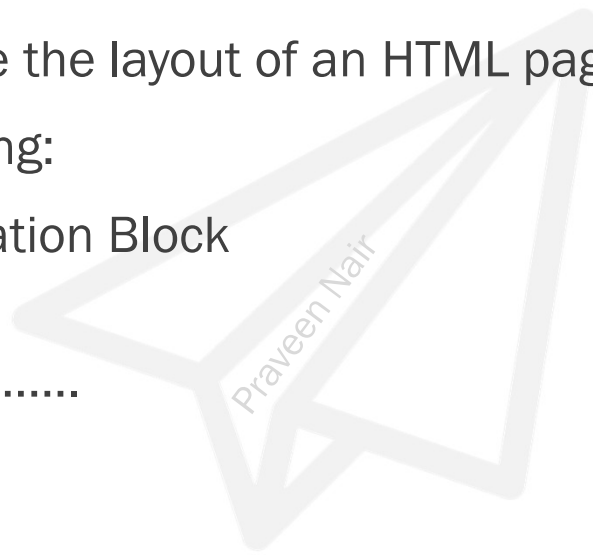
CSS is used to change the layout of an HTML page.

CSS consist of following:

1. Selector & Declaration Block
2. Property & Value

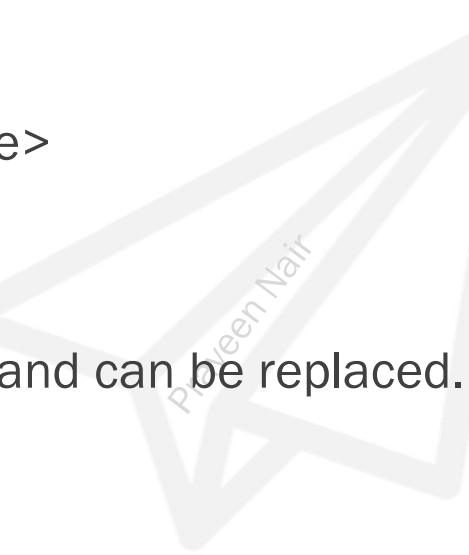
.....

H1 = {color:blue}



Inserting inline CSS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <p style="color:blue">
      This is a dummy text and can be replaced.
    </p>
  </body>
</html>
```



Inserting internal CSS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      p {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>Paragraph 1 - This is a dummy text and can be replaced</p>
    <p>Paragraph 2 - This is a dummy text and can be replaced</p>
  </body>
</html>
```

Inserting external CSS

```
<html lang="en">
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <p>This is a dummy text and can be replaced.</p>
  </body>
</html>
```

.....

styles.css

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



Order of precedence

1. Inline style
2. Internal style
3. External style
4. Browser default

Note: If external stylesheet is referred after internal block then external style will take precedence



Select based on tag, id or class

`p {color: red;}`

`#p1{color: blue;}`

`.c1 {color: green;}`

`.c2 {color: yellow;}`

`p.c1 {color: maroon;}`



Refer more than one class

```
.c1 {color: green;}
```

```
.c2 {text-align: center;}
```

```
#p1{color: blue;}
```

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

```
.....
```

```
<p class="c1 c2" id=p1>This is dummy text</p>
```

```
.....
```

Id selector takes the first precedence and tag is last precedence

Universal – affects every element

```
. * {  
  text-align: center;  
  color: blue;  
}
```



Grouping selectors

```
p, h1, div {  
  color: red;  
}
```



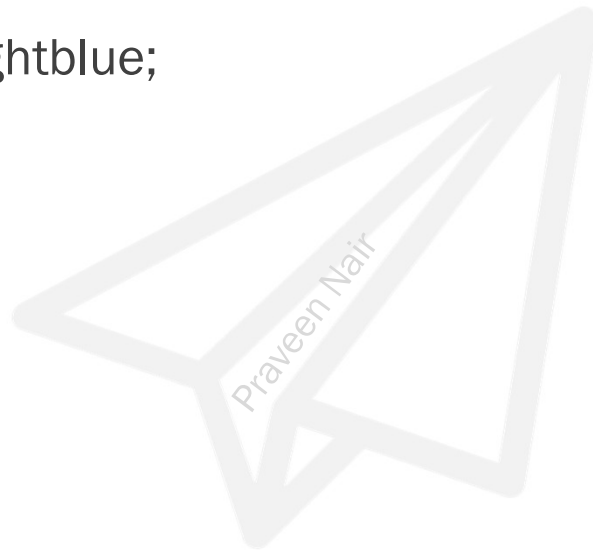
CSS Comments

```
/* This is a single-line comment */  
H1 {  
  color: blue;  
}
```



Colors and Backgrounds

```
div {  
  background-color: lightblue;  
  color: green  
}
```



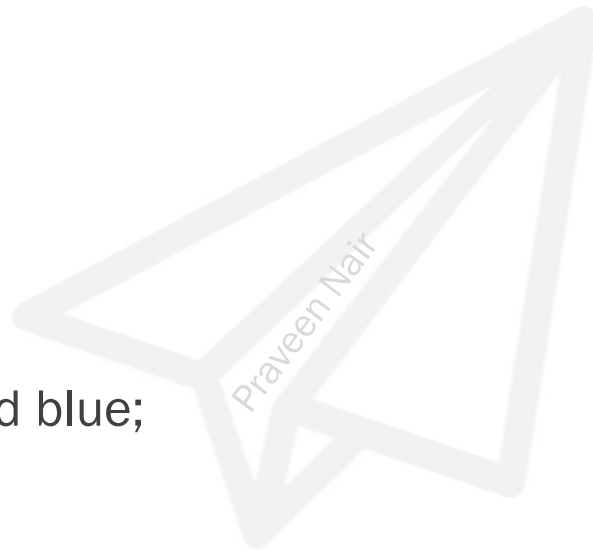
Media queries

```
<style>
.container {
  display: flex;
  flex-wrap: wrap;
}
@media only screen and (max-width: 600px) {
  /* For phones: */
  .box {
    color: green;
    width: 100%;
  }
}
@media only screen and (min-width: 600px) {
  /* For tablets: */
  .box {
    color: blue;
    width: 50%;
  }
}
@media only screen and (min-width: 768px) {
  /* For desktop: */
  .box {
    color: red;
    width: 25%;
  }
}
@media only screen and (orientation: landscape) {
  .box {
    background-color: lightblue;
  }
}
@media only screen and (orientation: portrait) {
  .box {
    background-color: lightgreen;
  }
}
</style>
<body>
<div class="container">
  <div class="box">Hello</div>
  <div class="box">Hello</div>
  <div class="box">Hello</div>
  <div class="box">Hello</div>
  <div class="box">Hello</div>
</div>
</body>
```




Position fixed

```
div {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 500px;  
    border: 10px solid blue;  
}
```



Position sticky

```
div {  
    position: sticky;  
    top: 0;  
    background-color: green;  
    border: 2px solid #4caf50;  
}
```



Background-image

```
body {  
    background-image: url("colors.png");  
    background-repeat: no-repeat; /* repeat */  
    background-position: center; /* right top, left top */  
    background-attachment: scroll; /*fixed*/  
    background-size: cover;  
}
```

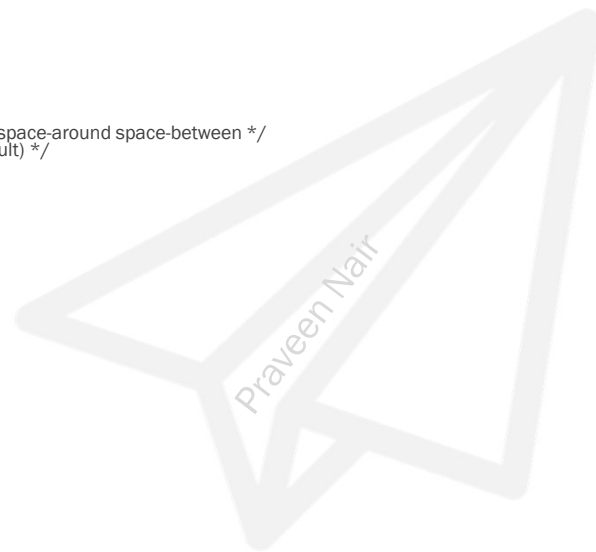
```
.....  
body {background: #eeeeee url("colors.png") no-repeat right top;}
```

```
.....  
background-color, background-image, background-repeat, background-  
attachment, background-position
```


Flex box

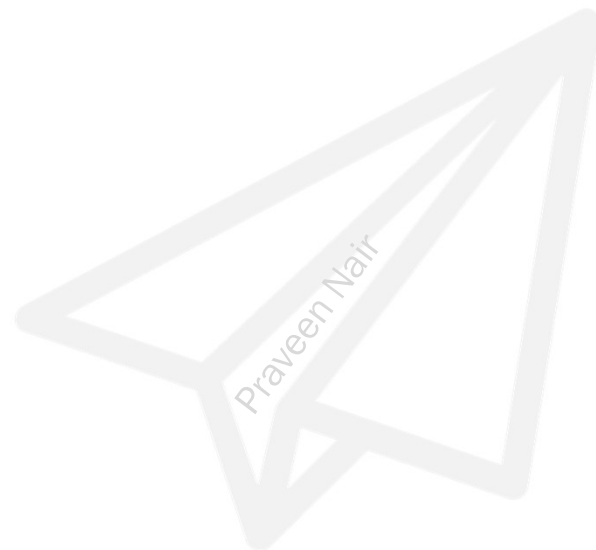
```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  display: flex;
  height: 100px;
  /*flex-direction: row;*/ /* try column */
  background-color: silver;
  /* flex-wrap: wrap; */
  /* flex-flow: row wrap; */
  /* justify-content: right; center left flex-start flex-end space-around space-between */
  /* align-items: center; flex-start flex-end stretch(default) */
  /* align-content: flex-start; */
}

.container > div {
  background-color: orange;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 20px;
}
</style>
</head>
<body>
<div class="container">
  <div>HTML</div>
  <div>CSS</div>
  <div>JavaScript</div>
</div>
</body>
</html>
```



Grid

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  display: grid; /*inline-grid */
  background-color: silver;
  grid-template-columns: auto auto auto;
  column-gap: 20px;
  row-gap: 30px;
  /* gap: 20px 30px; for both row and col */
}
.container > div {
  background-color: orange;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 20px;
}
</style>
</head>
<body>
<div class="container">
  <div>HTML</div>
  <div>CSS</div>
  <div>JavaScript</div>
  <div>HTML</div>
  <div>CSS</div>
  <div>JavaScript</div>
  <div>HTML</div>
  <div>CSS</div>
  <div>JavaScript</div>
</div>
</body>
</html>
```



Gradients

```
div {  
    /*background-image: linear-gradient(maroon, yellow);*/  
    /*background-image: linear-gradient(to right,red,pink);*/  
    /*background-image: linear-gradient(to bottom right,maroon, yellow);*/  
    /*background-image: linear-gradient(90deg,maroon, yellow);*/  
    /*background-image: linear-gradient(maroon, yellow,pink);*/  
    /*background-image: radial-gradient(red, yellow, green);*/  
    /*background-image: radial-gradient(maroon 10%, yellow 20%, green 30%);*/  
    /*background-image: radial-gradient(circle,red, pink, green);*/  
    background-image: repeating-radial-gradient(maroon, pink 10%, green 15%);  
}
```

Transformation and Transition

```
<!DOCTYPE html>
<html>
<head>
<style>
.c {
  transform: translate(50px, 100px);
  /* transform: rotate(30deg); */
  /* transform: scale(1.2, 1.2); */
  /* transform: scaleX(1.3); */
  /* transform: scaleY(1.3); */
  /* transform: skew(10deg, 20deg); */
  /* transform: skewX(10deg); */
  /* transform: skewY(20deg); */
  /* transform: matrix(1, -0.3, 0, 1, 20, 40); */
  /* transform: matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY()) */
}
</style>
</head>
<body>


</body>
</html>
```

Thank You

- PRAVEEN NAIR