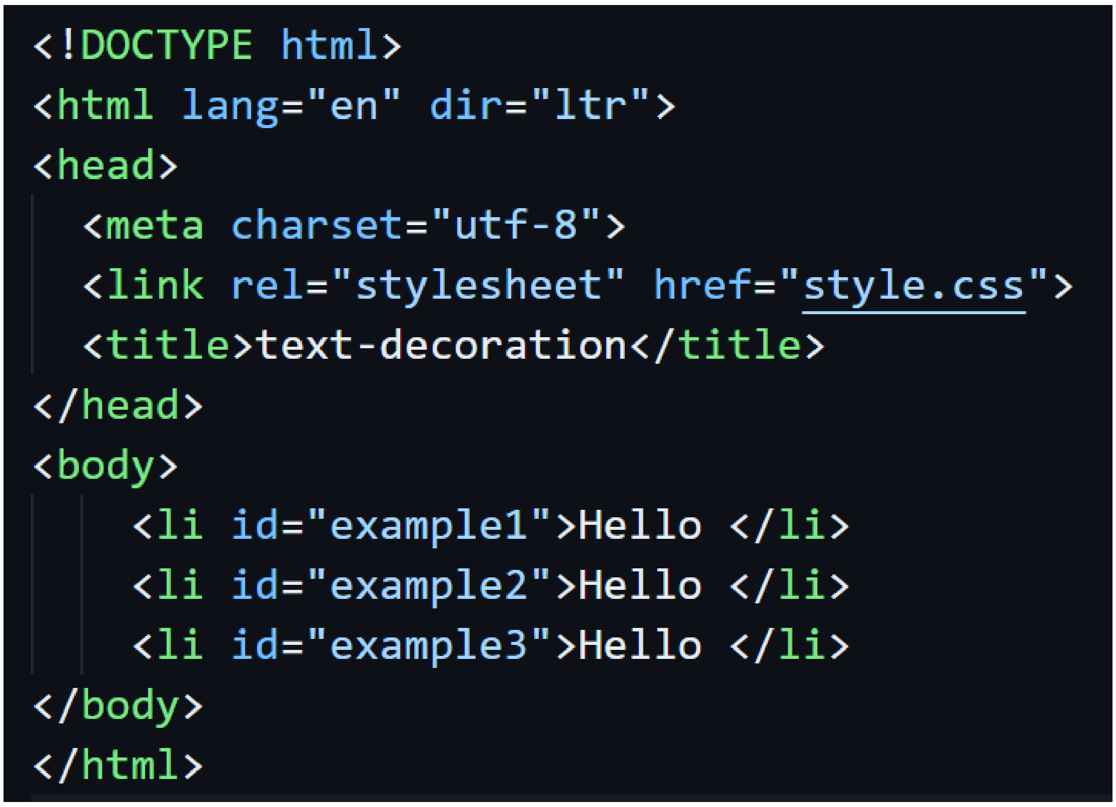
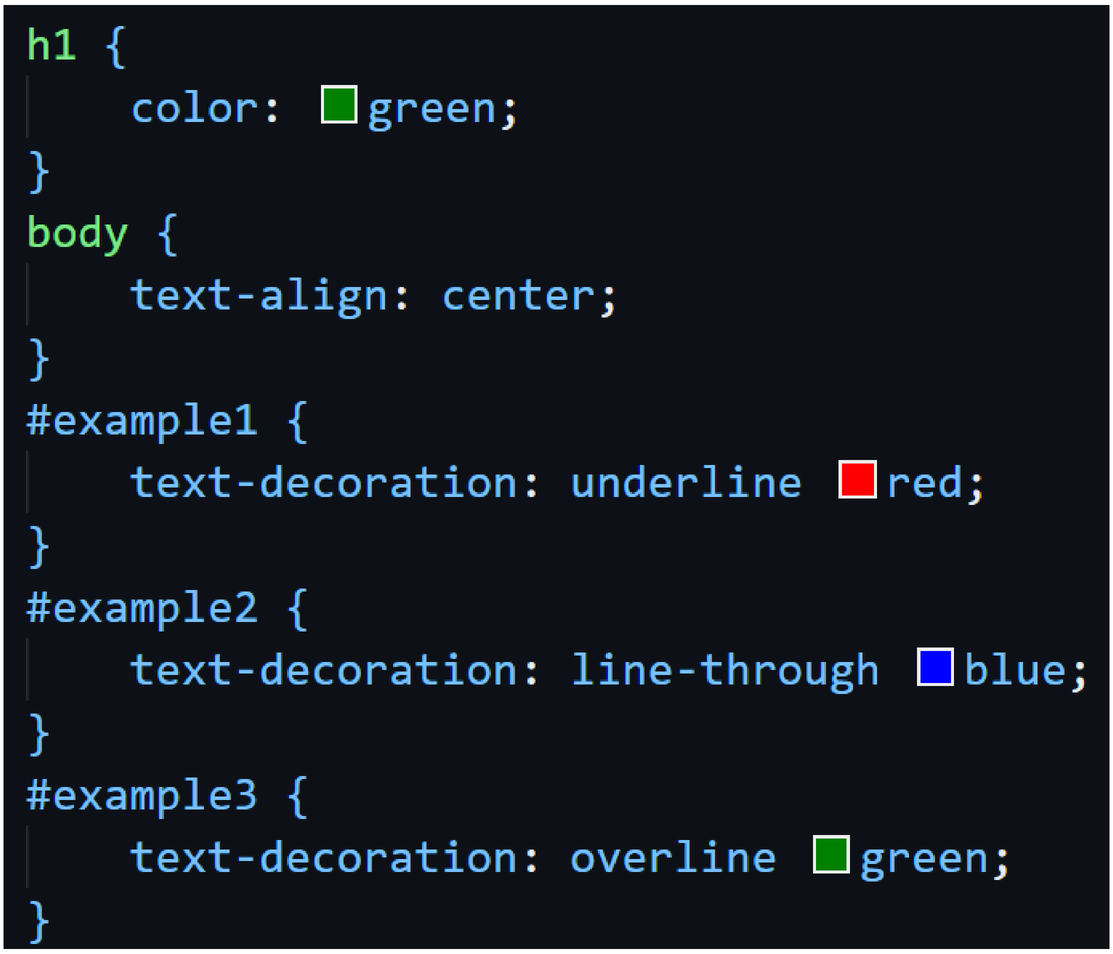
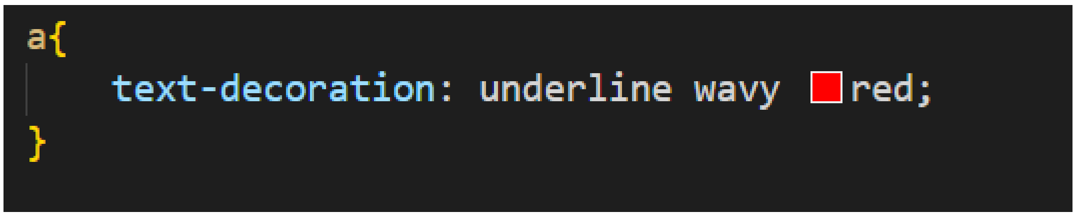
text-decoration Property- none | underline | line-through | overline | inherit



```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
  <title>text-decoration</title>
</head>
<body>
    <li id="example1">Hello </li>
    <li id="example2">Hello </li>
    <li id="example3">Hello </li>
</body>
</html>
```



```css
h1 {
    color: green;
}
body {
    text-align: center;
}
#example1 {
    text-decoration: underline red;
}
#example2 {
    text-decoration: line-through blue;
}
#example3 {
    text-decoration: overline green;
}
```

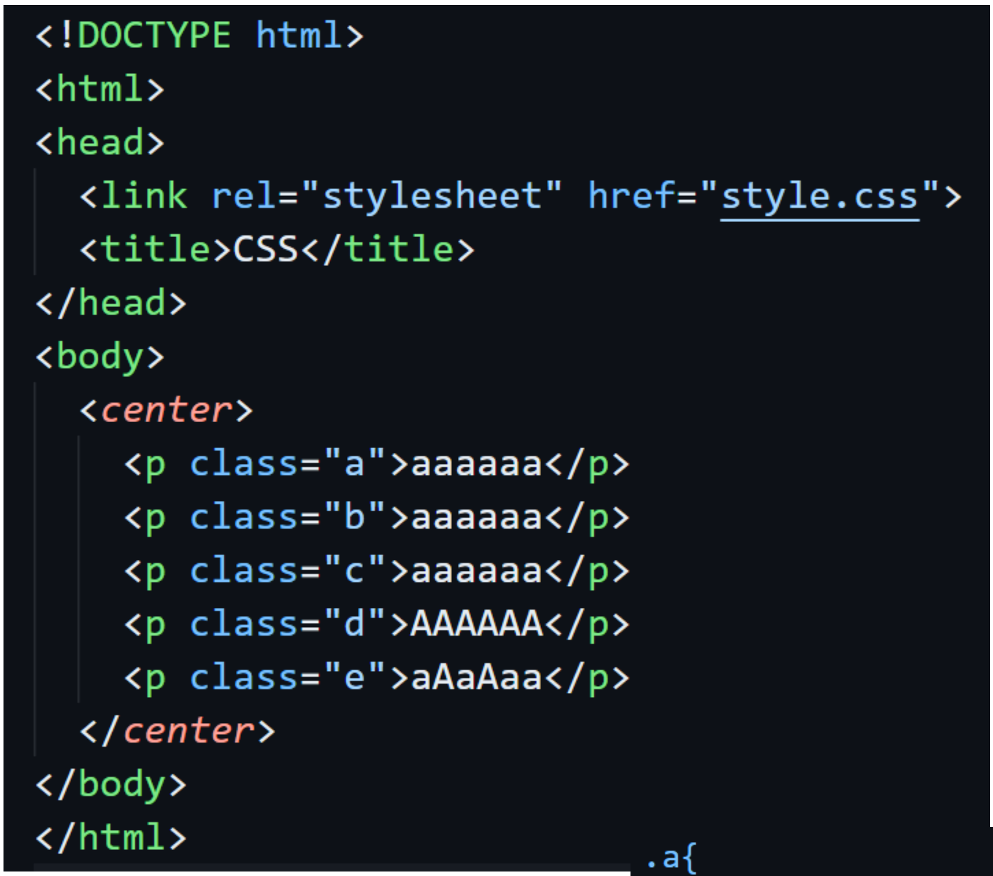


```css
a{
    text-decoration: underline wavy red;
}
```

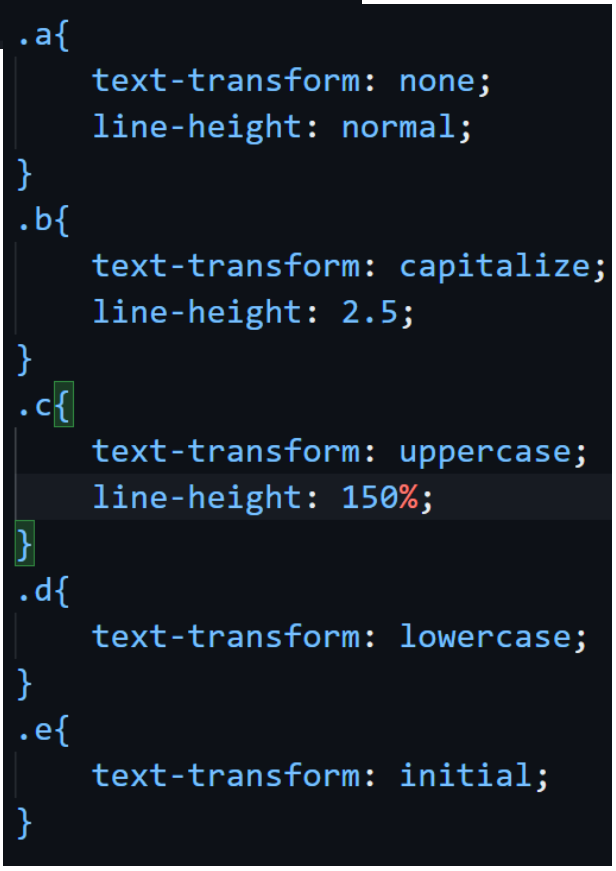
Text transformation & text height-

```
text-transform:
none|capitalize|uppercase|lowercase|initial|inherit;
```

```
line-height: normal|number|length|percentage|initial|inherit;
```

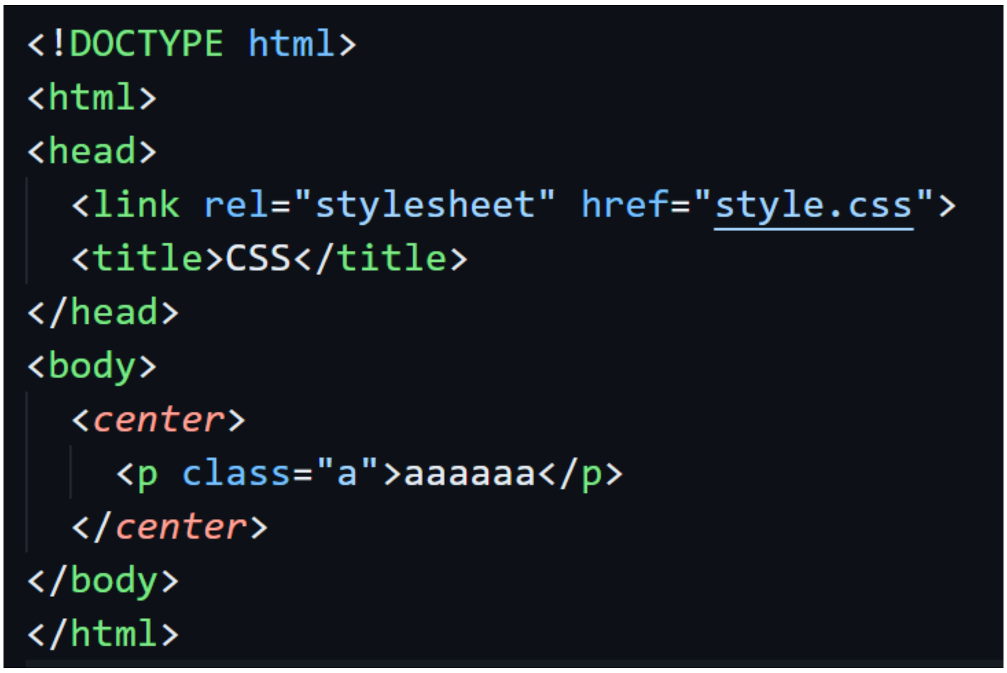


```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <title>CSS</title>
</head>
<body>
  <center>
    <p class="a">aaaaaa</p>
    <p class="b">aaaaaa</p>
    <p class="c">aaaaaa</p>
    <p class="d">AAAAAA</p>
    <p class="e">aAaAaa</p>
  </center>
</body>
</html>
```

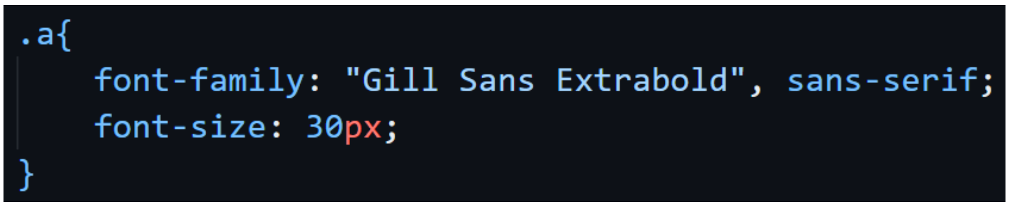


```
.a{
    text-transform: none;
    line-height: normal;
}
.b{
    text-transform: capitalize;
    line-height: 2.5;
}
.c{
    text-transform: uppercase;
    line-height: 150%;
}
.d{
    text-transform: lowercase;
}
.e{
    text-transform: initial;
}
```

Font family-



```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <title>CSS</title>
</head>
<body>
  <center>
    <p class="a">aaaaaa</p>
  </center>
</body>
</html>
```



```css
.a{
    font-family: "Gill Sans Extrabold", sans-serif;
    font-size: 30px;
}
```

Position property-

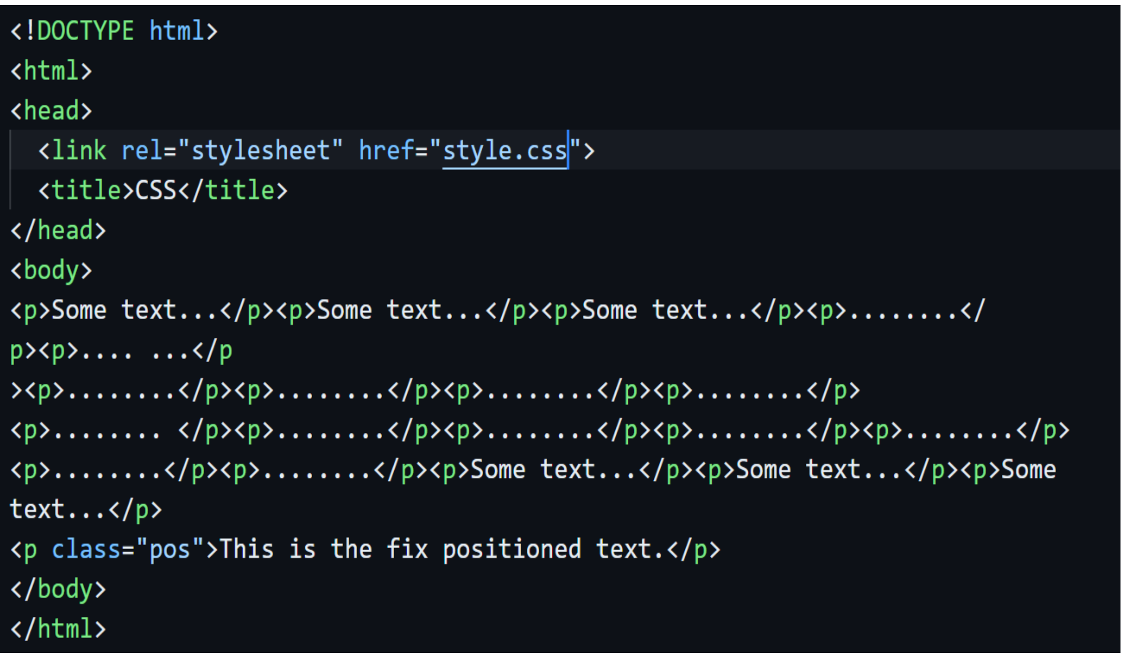
CSS Static Positioning

CSS Fixed Positioning

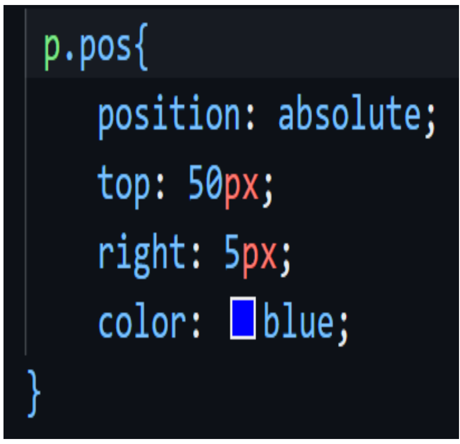
CSS Relative Positioning

CSS Absolute Positioning

CSS Sticky Positioning

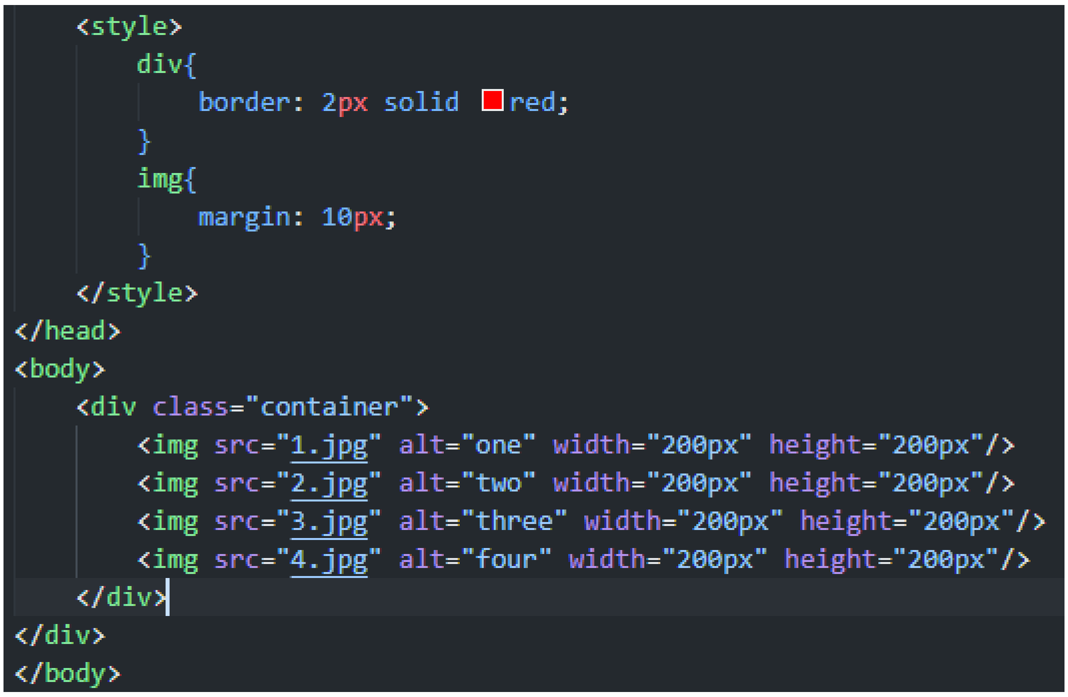


```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <title>CSS</title>
</head>
<body>
<p>Some text...</p><p>Some text...</p><p>Some text...</p><p>........</
p><p>.... ...</p
><p>........</p><p>........</p><p>........</p><p>........</p>
<p>........ </p><p>........</p><p>........</p><p>........</p><p>........</p>
<p>........</p><p>........</p><p>Some text...</p><p>Some text...</p><p>Some
text...</p>
<p class="pos">This is the fix positioned text.</p>
</body>
</html>
```

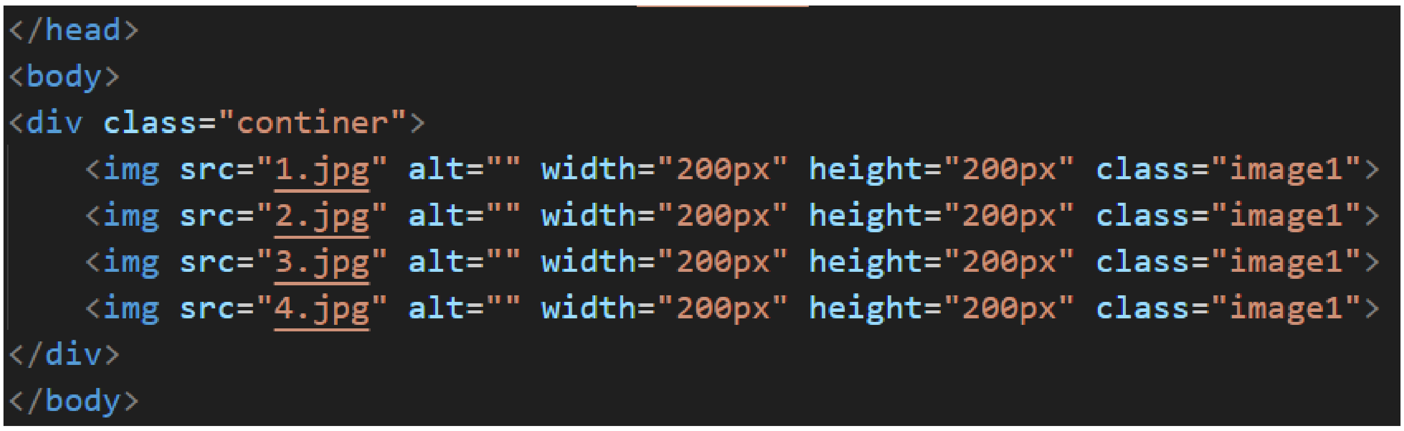


```css
p.pos{
    position: absolute;
    top: 50px;
    right: 5px;
    color: blue;
}
```

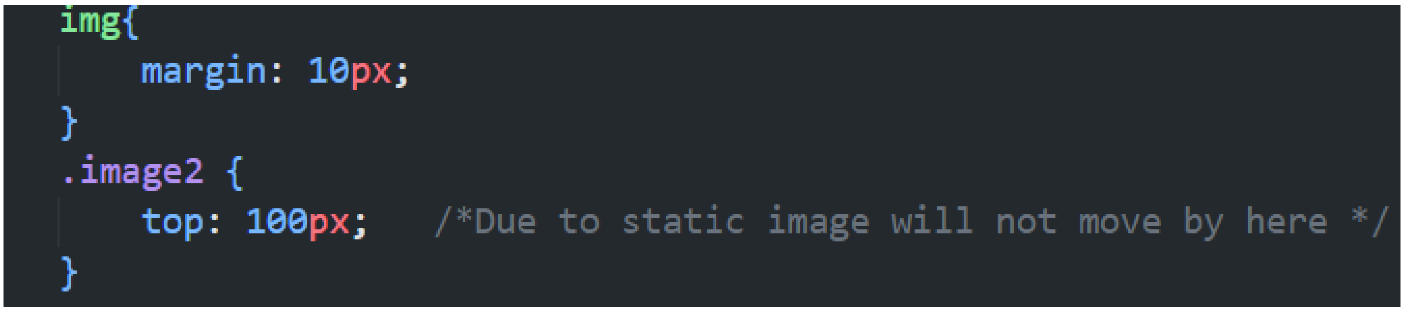
By default static property is apply so if we want to move these images due to static they will not move



```
    <style>
        div{
            border: 2px solid red;
        }
        img{
            margin: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <img src="1.jpg" alt="one" width="200px" height="200px"/>
        <img src="2.jpg" alt="two" width="200px" height="200px"/>
        <img src="3.jpg" alt="three" width="200px" height="200px"/>
        <img src="4.jpg" alt="four" width="200px" height="200px"/>
    </div>
</div>
</body>
```



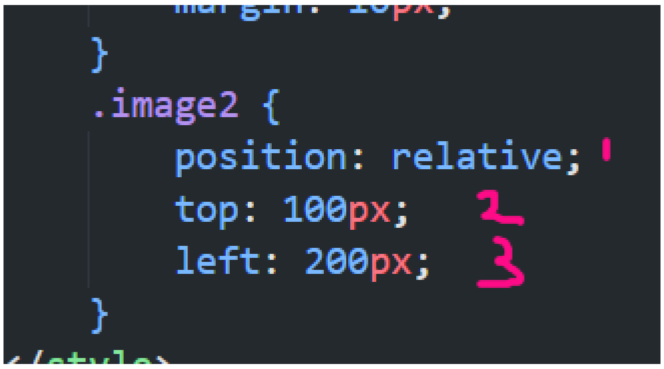
```
</head>
<body>
<div class="continer">
    <img src="1.jpg" alt="" width="200px" height="200px" class="image1">
    <img src="2.jpg" alt="" width="200px" height="200px" class="image1">
    <img src="3.jpg" alt="" width="200px" height="200px" class="image1">
    <img src="4.jpg" alt="" width="200px" height="200px" class="image1">
</div>
</body>
```



```
img{
    margin: 10px;
}
.image2 {
    top: 100px;    /*Due to static image will not move by here */
}
```

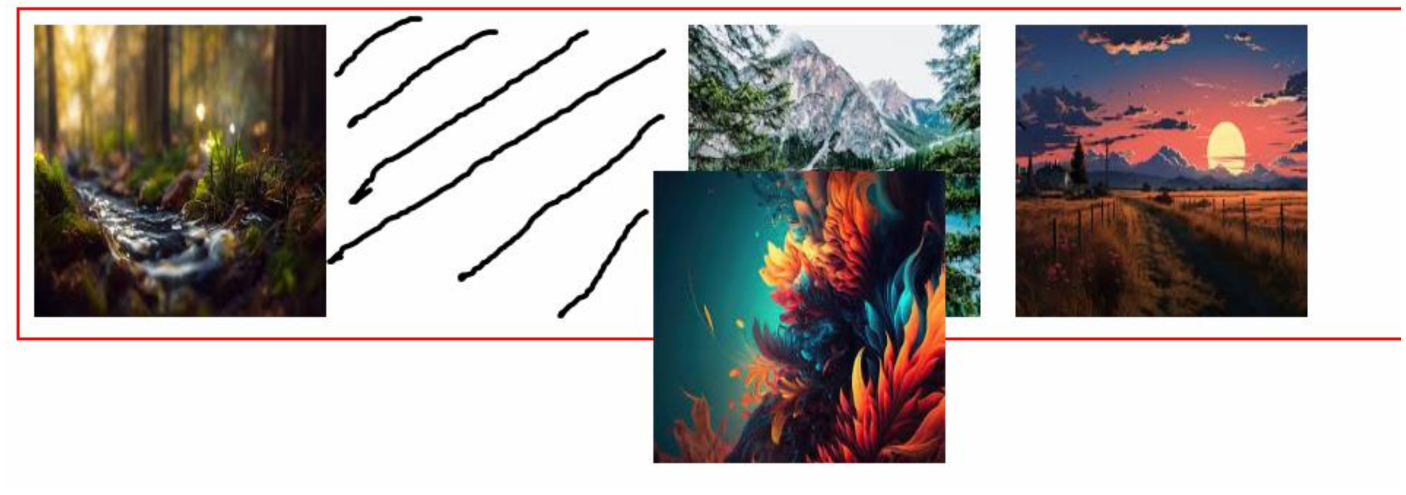
See o/p I mage will not move

But by using Relative property we can move the image and relative means in the respect of current window.

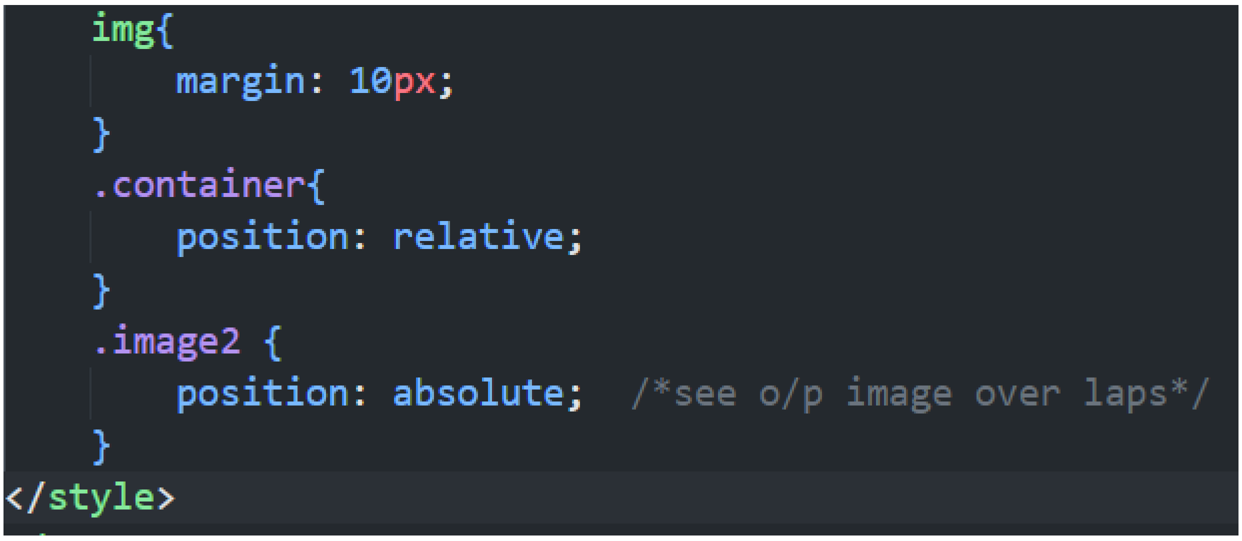


```
        margin: 10px;
    }
    .image2 {
        position: relative; 1
        top: 100px;   2
        left: 200px;  3
    }
</style>
```

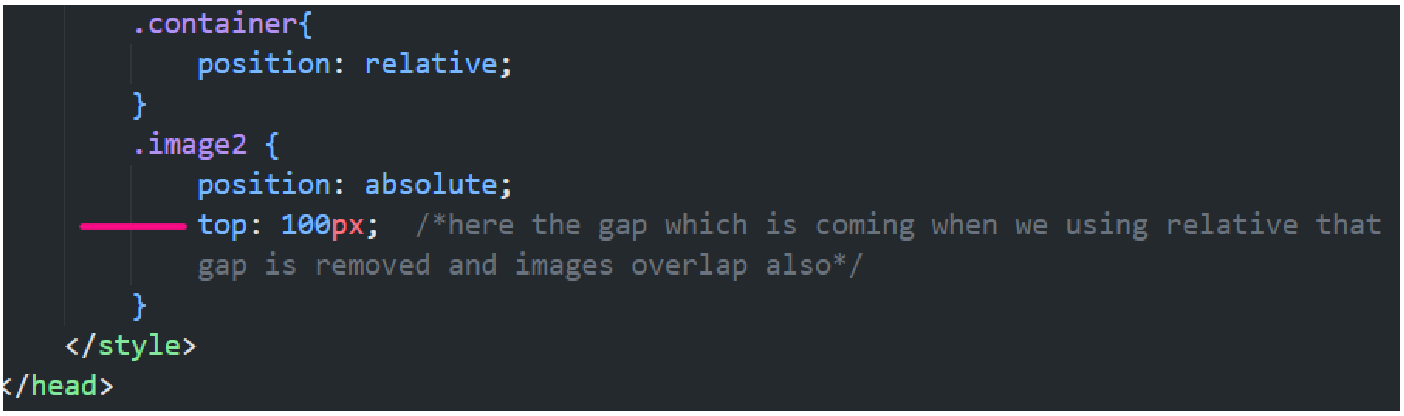
In relative we have four properties top, left, right, bottom. But when we use relative property to move any image then the previous space which is acquired by that image is remain same.



Now lets use absolute position. Try not to use this property becz when we use this property and adjust image position then we will face some problem while making website responsive. We use this property when we want to do overlapping. The image we move by using absolute position property that I mage will move according to its closest position ancestors like we have div and inside it we have four images then the image will move according to this div. so when we use absolute property on any image then we will mark relative to its closest position ancestor like we make div as relative and then we mark image as absolute. We can do without this also by using some tricks.

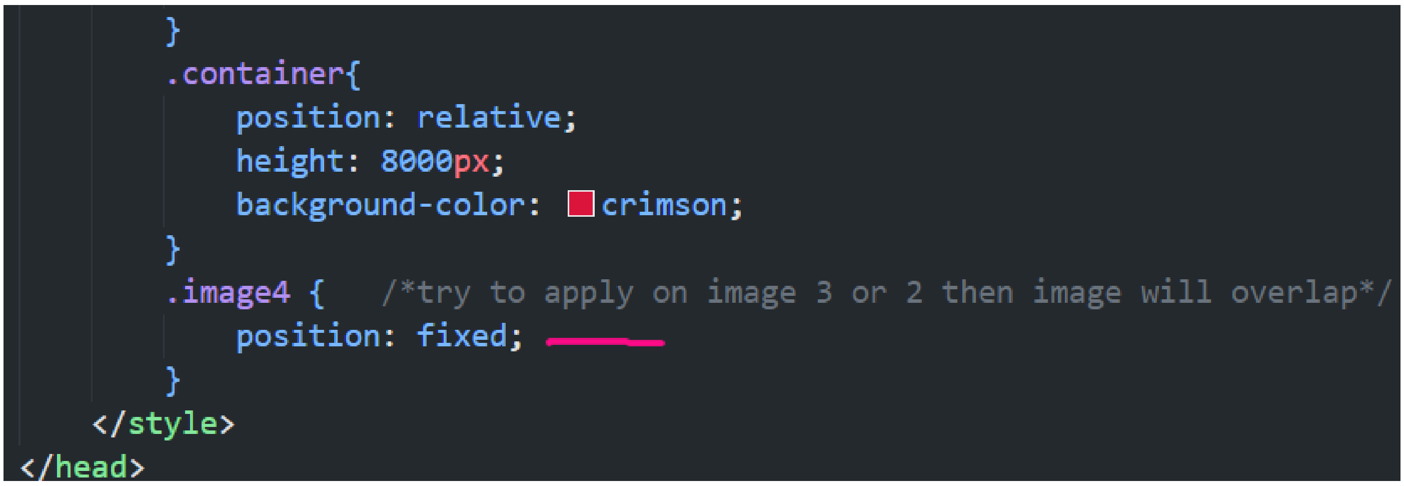


```
    img{
        margin: 10px;
    }
    .container{
        position: relative;
    }
    .image2 {
        position: absolute;  /*see o/p image over laps*/
    }
</style>
```

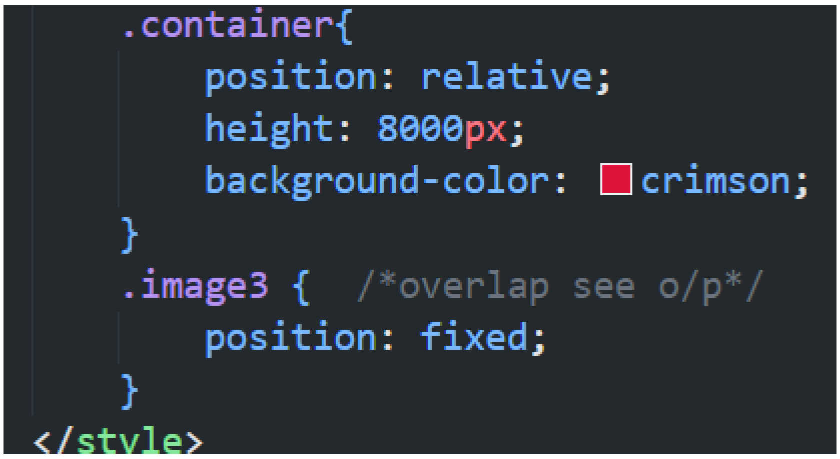


```
        .container{
            position: relative;
        }
        .image2 {
            position: absolute;
            top: 100px;  /*here the gap which is coming when we using relative that
            gap is removed and images overlap also*/
        }
    </style>
</head>
```

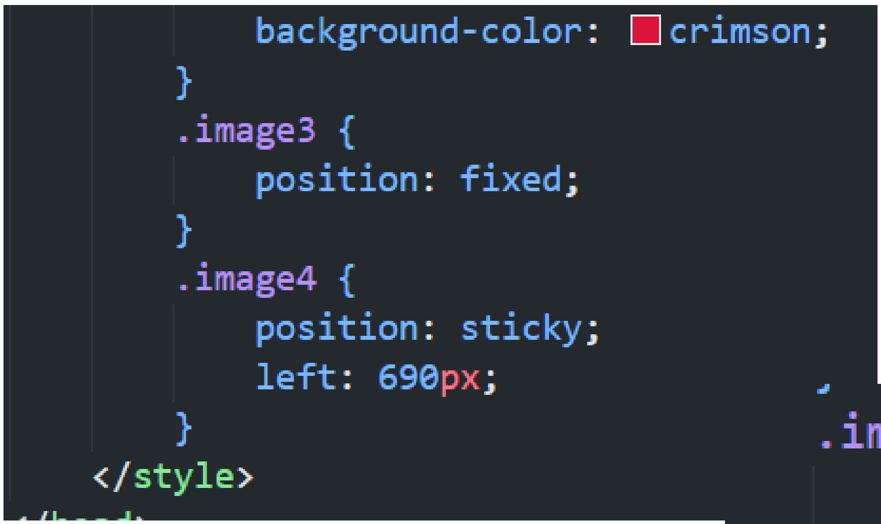
Fixed position => by this our image will fixed at that position with respect to entire web page so if we scroll other images will move but that particular image will remain at same location. Example sign in or chat button on right side bottom corner



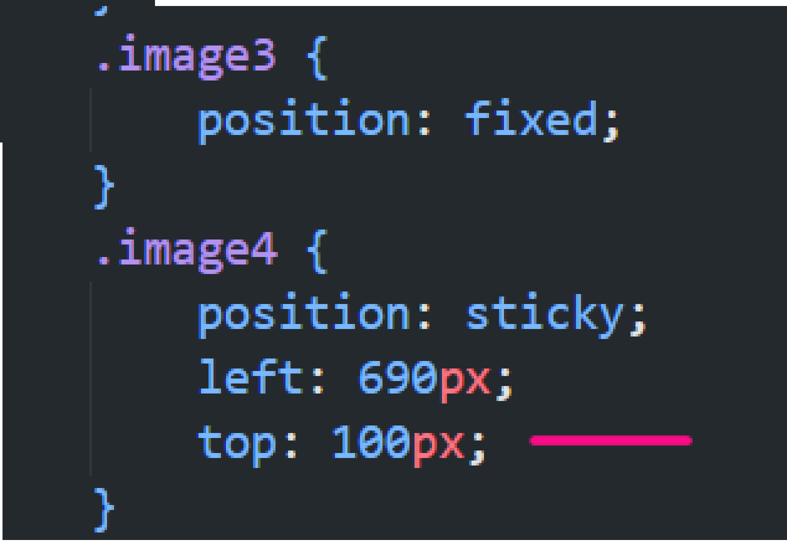
```
        }
        .container{
            position: relative;
            height: 8000px;
            background-color: 🟥crimson;
        }
        .image4 {   /*try to apply on image 3 or 2 then image will overlap*/
            position: fixed;
        }
    </style>
</head>
```



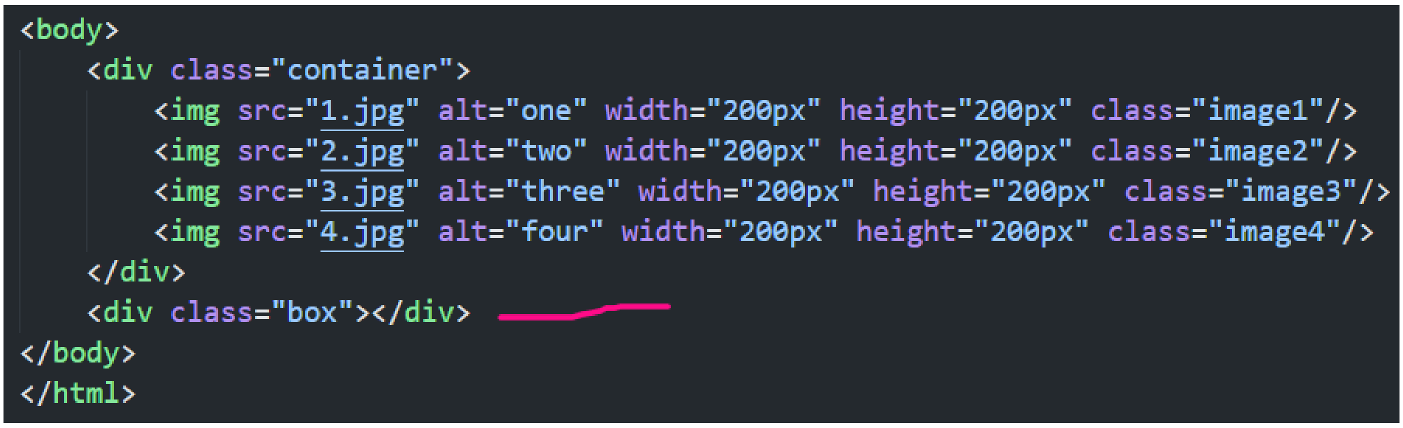
```
    .container{
        position: relative;
        height: 8000px;
        background-color: 🟥crimson;
    }
    .image3 {  /*overlap see o/p*/
        position: fixed;
    }
</style>
```



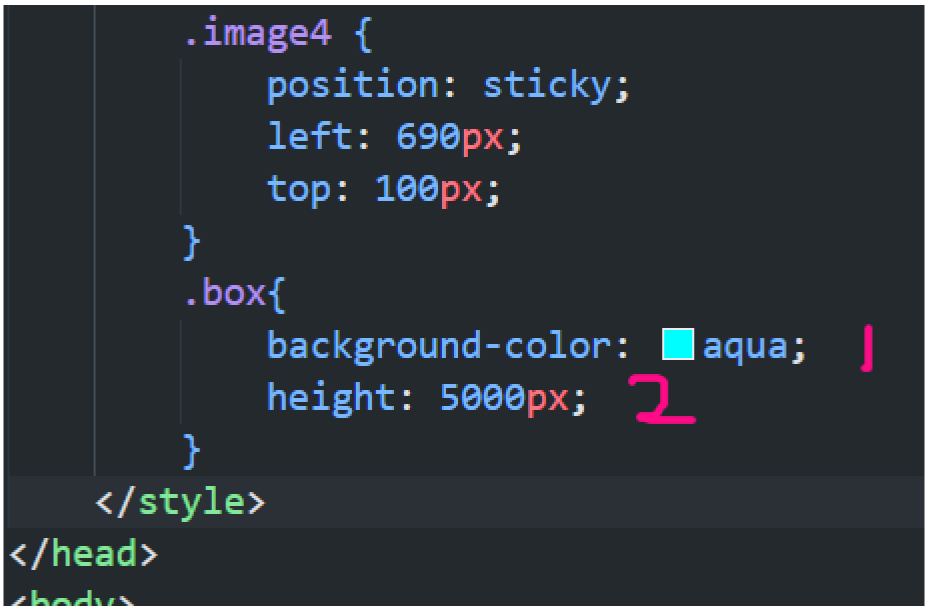
```
            background-color: 🟥crimson;
        }
        .image3 {
            position: fixed;
        }
        .image4 {
            position: sticky;
            left: 690px;
        }
    </style>
```



```
.image3 {
    position: fixed;
}
.image4 {
    position: sticky;
    left: 690px;
    top: 100px;
}
```



```
<body>
    <div class="container">
        <img src="1.jpg" alt="one" width="200px" height="200px" class="image1"/>
        <img src="2.jpg" alt="two" width="200px" height="200px" class="image2"/>
        <img src="3.jpg" alt="three" width="200px" height="200px" class="image3"/>
        <img src="4.jpg" alt="four" width="200px" height="200px" class="image4"/>
    </div>
    <div class="box"></div>
</body>
</html>
```

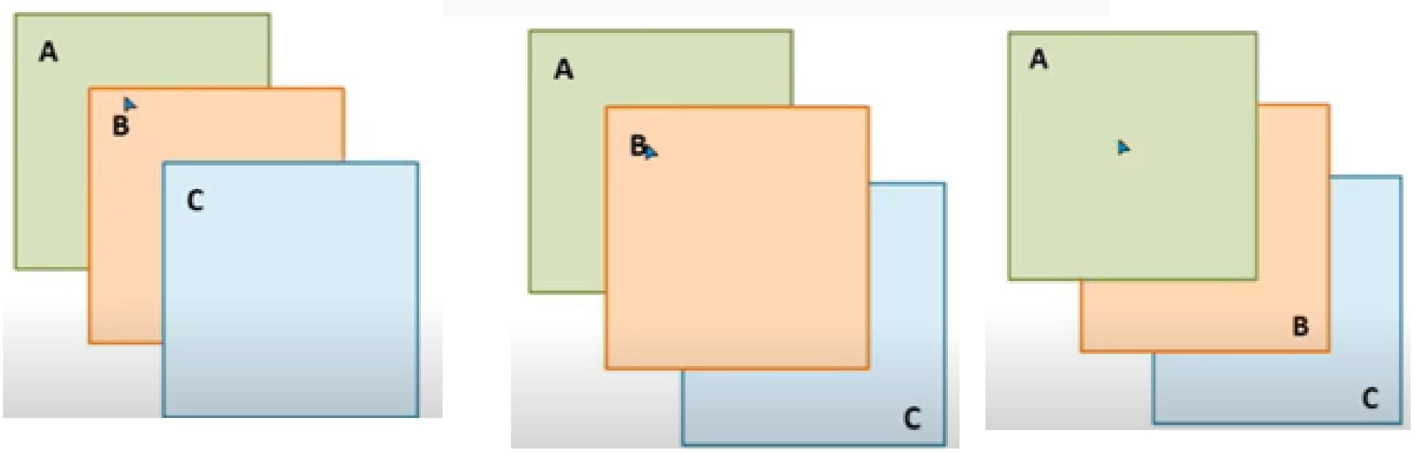


```
        .image4 {
            position: sticky;
            left: 690px;
            top: 100px;
        }
        .box{
            background-color: aqua;   1
            height: 5000px;  2
        }
    </style>
</head>
```

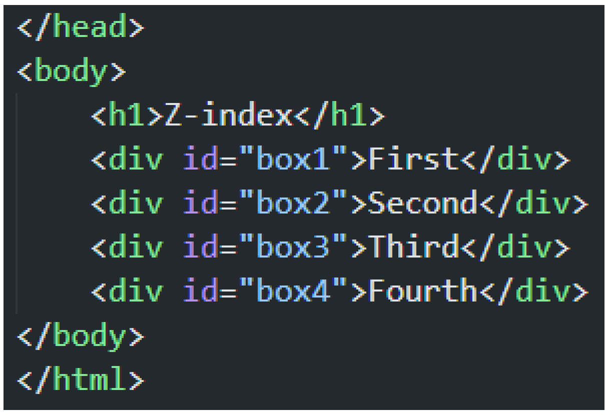
z-index Property- this property is only applicable when we use position property. Position should not be static

The z-index property is used to displace elements on the z-axis i.e in or out of the screen.

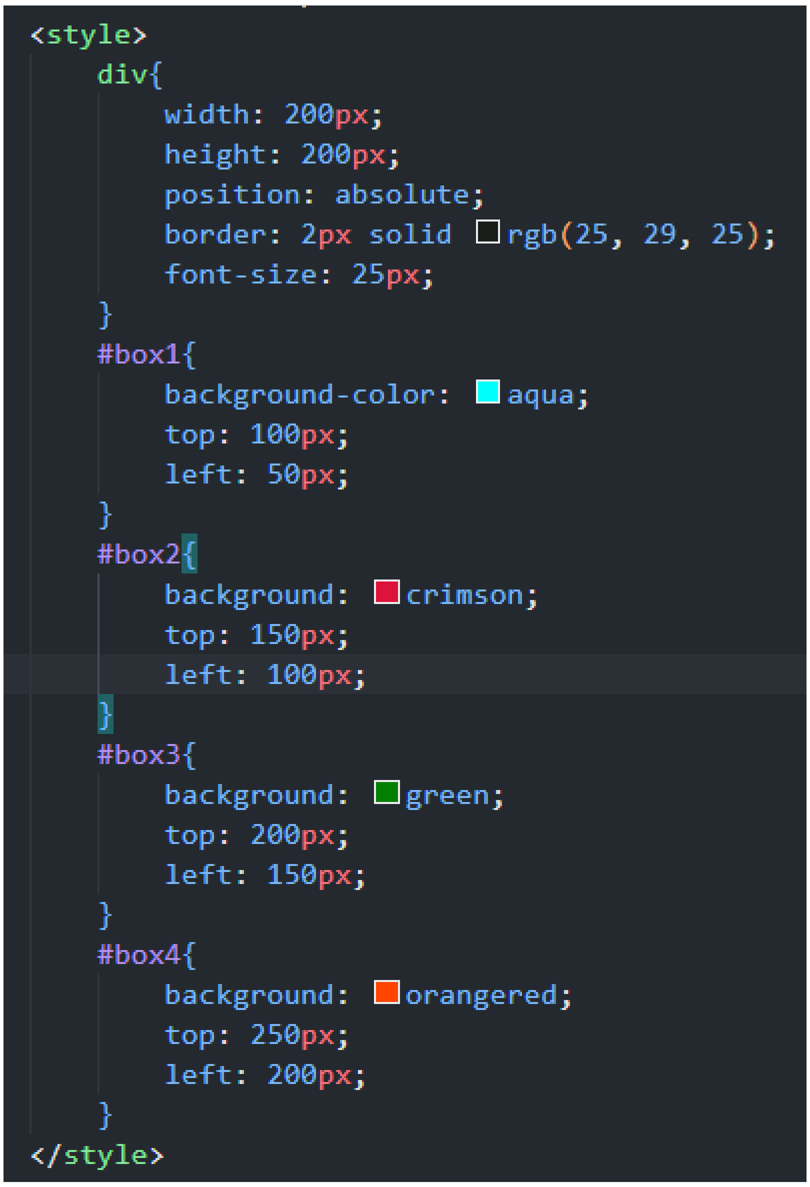
```
z-index: auto|number|initial|inherit;
```



By default z-index is auto means 0, 1 come to front, -1 goes to back



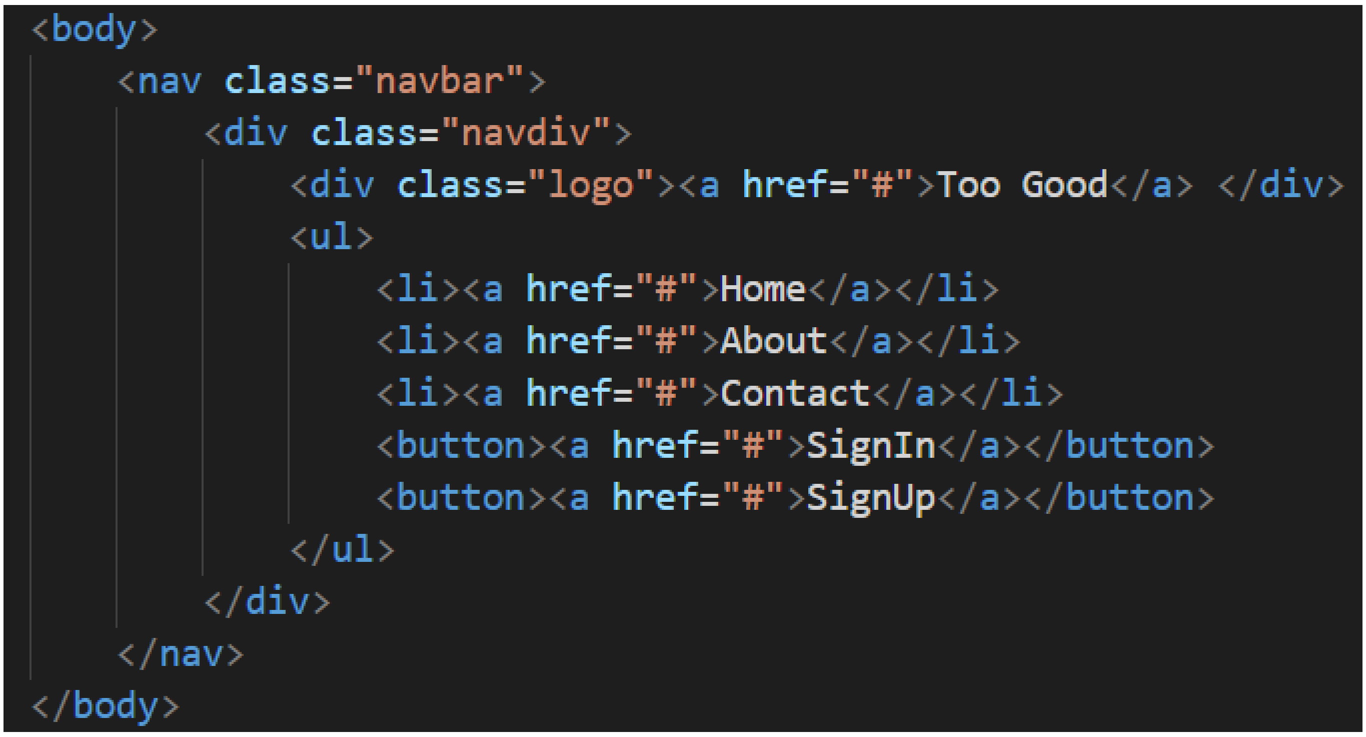
```
</head>
<body>
    <h1>Z-index</h1>
    <div id="box1">First</div>
    <div id="box2">Second</div>
    <div id="box3">Third</div>
    <div id="box4">Fourth</div>
</body>
</html>
```



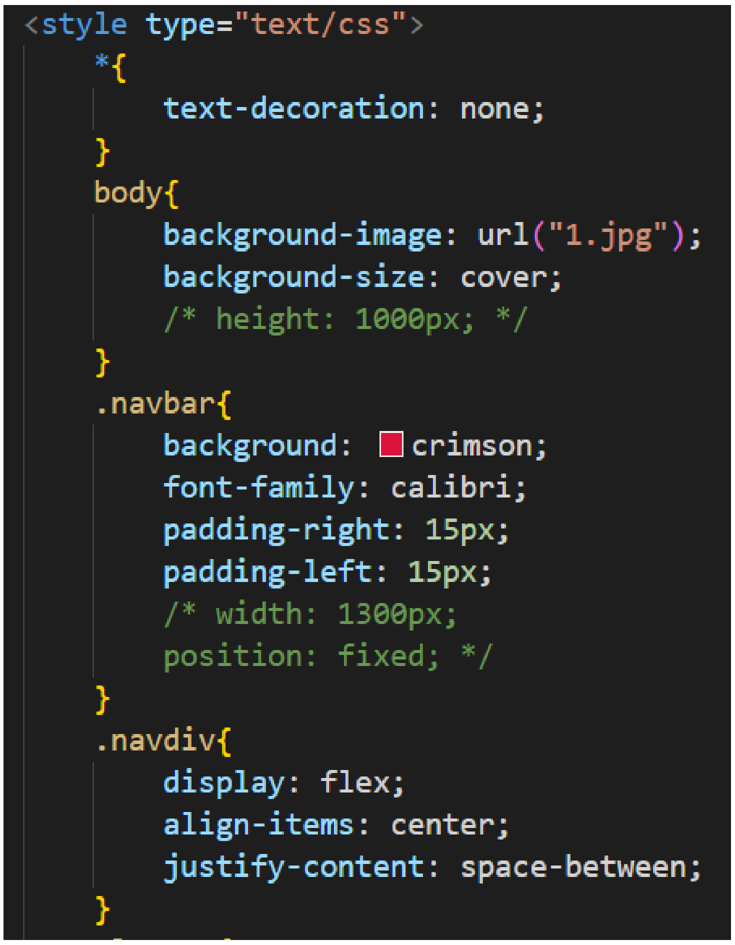
```
<style>
    div{
        width: 200px;
        height: 200px;
        position: absolute;
        border: 2px solid rgb(25, 29, 25);
        font-size: 25px;
    }
    #box1{
        background-color: aqua;
        top: 100px;
        left: 50px;
    }
    #box2{
        background: crimson;
        top: 150px;
        left: 100px;
    }
    #box3{
        background: green;
        top: 200px;
        left: 150px;
    }
    #box4{
        background: orangered;
        top: 250px;
        left: 200px;
    }
</style>
```

Now apply z-index 1, -1

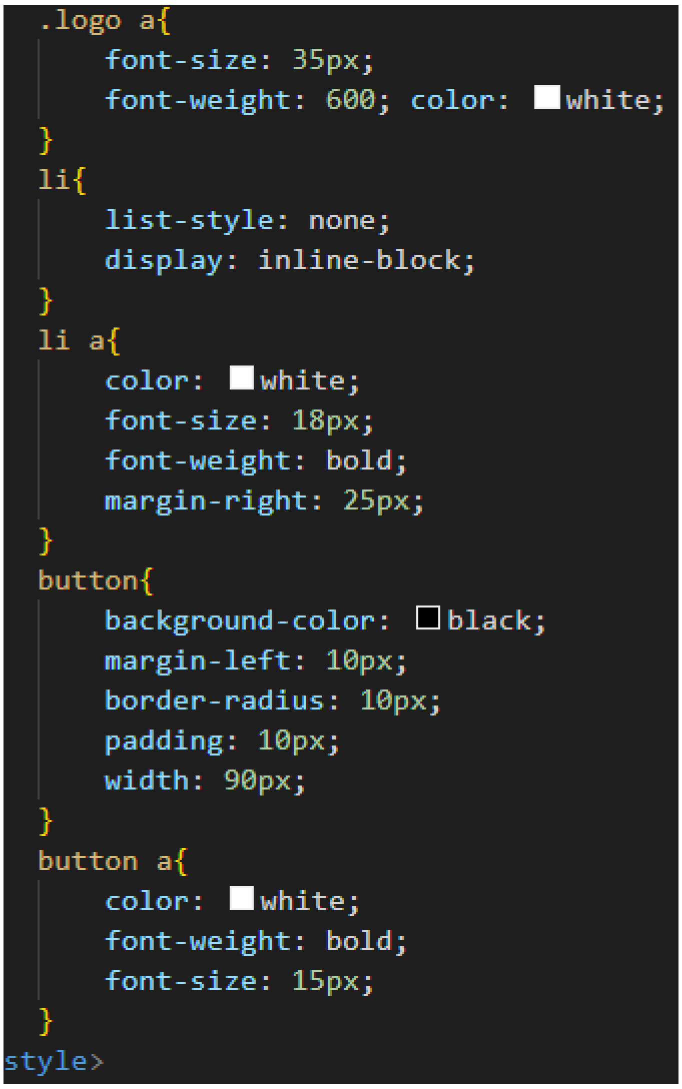
Navbar =>



```html
<body>
    <nav class="navbar">
        <div class="navdiv">
            <div class="logo"><a href="#">Too Good</a> </div>
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About</a></li>
                <li><a href="#">Contact</a></li>
                <button><a href="#">SignIn</a></button>
                <button><a href="#">SignUp</a></button>
            </ul>
        </div>
    </nav>
</body>
```

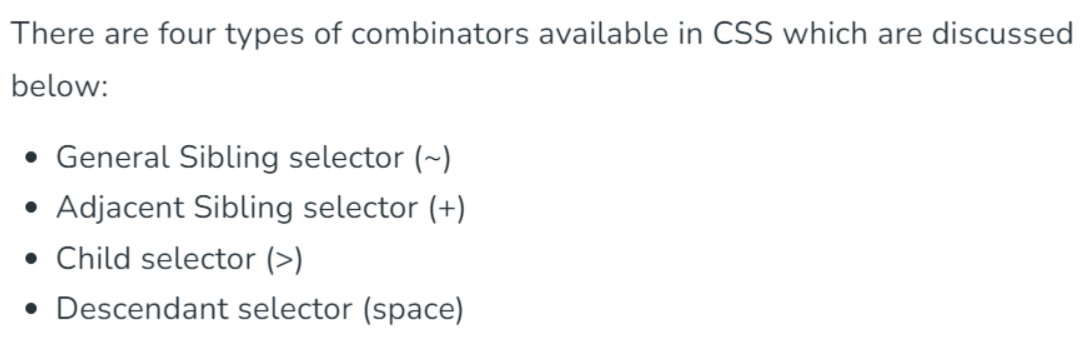


```css
<style type="text/css">
    *{
        text-decoration: none;
    }
    body{
        background-image: url("1.jpg");
        background-size: cover;
        /* height: 1000px; */
    }
    .navbar{
        background: crimson;
        font-family: calibri;
        padding-right: 15px;
        padding-left: 15px;
        /* width: 1300px;
        position: fixed; */
    }
    .navdiv{
        display: flex;
        align-items: center;
        justify-content: space-between;
    }
```



```css
.logo a{
    font-size: 35px;
    font-weight: 600; color: white;
}
li{
    list-style: none;
    display: inline-block;
}
li a{
    color: white;
    font-size: 18px;
    font-weight: bold;
    margin-right: 25px;
}
button{
    background-color: black;
    margin-left: 10px;
    border-radius: 10px;
    padding: 10px;
    width: 90px;
}
button a{
    color: white;
    font-weight: bold;
    font-size: 15px;
}
style>
```

CSS Combinators =>


There are four types of combinators available in CSS which are discussed below:

- General Sibling selector (~)
- Adjacent Sibling selector (+)
- Child selector (>)
- Descendant selector (space)

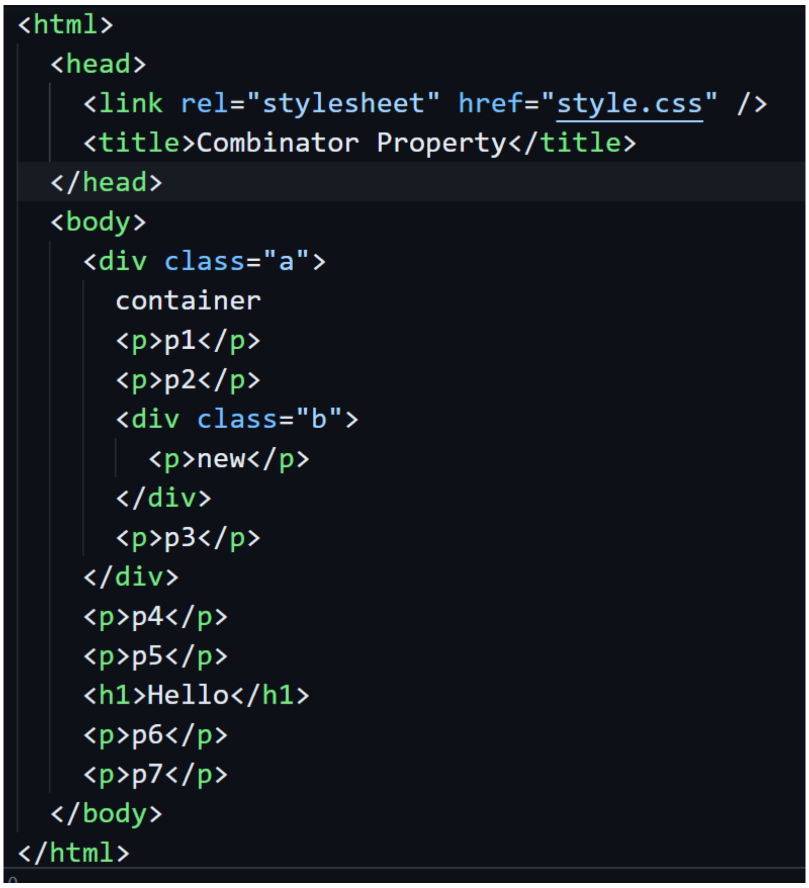
CSS combinators are explaining the relationship between two selectors. CSS selector can be a simple selector or a complex selector consisting of more than one selector connected using combinators.

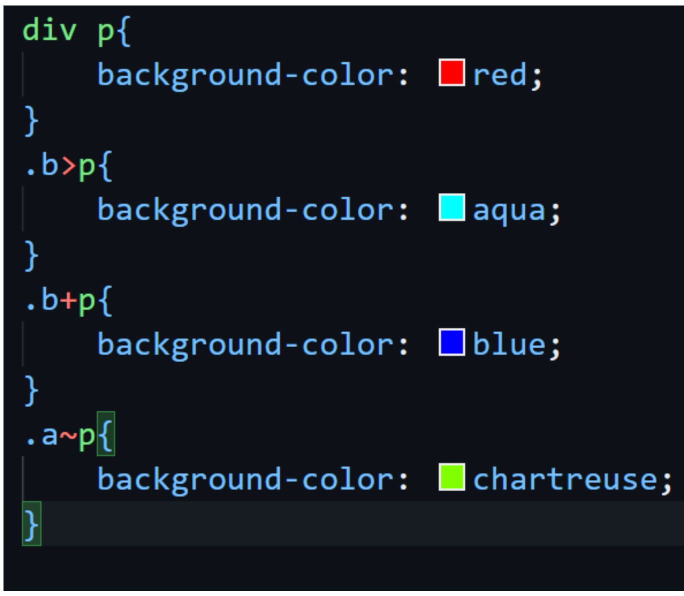
Descendant selector: This selector is used to select all the child elements of the specified tag. The tags can be the direct child of the specified tag or can be very deep in the specified tag. This combinator combines the two selectors such that selected elements have an ancestor the same as the first selector element.

Child Selector: This selector is used to select the element that is the immediate child of the specified tag. This combinator is stricter than the descendant selector because it selects only the second selector if it has the first selector element as its parent.

Adjacent Sibling selector: The Adjacent sibling selector is used to select the element that is adjacent or the element that is next to the specified selector tag. This combinator selects only one tag that is just next to the specified tag.

General Sibling selector: The general sibling selector is used to select a group of elements which is next of selected parent and that share the same parent element.

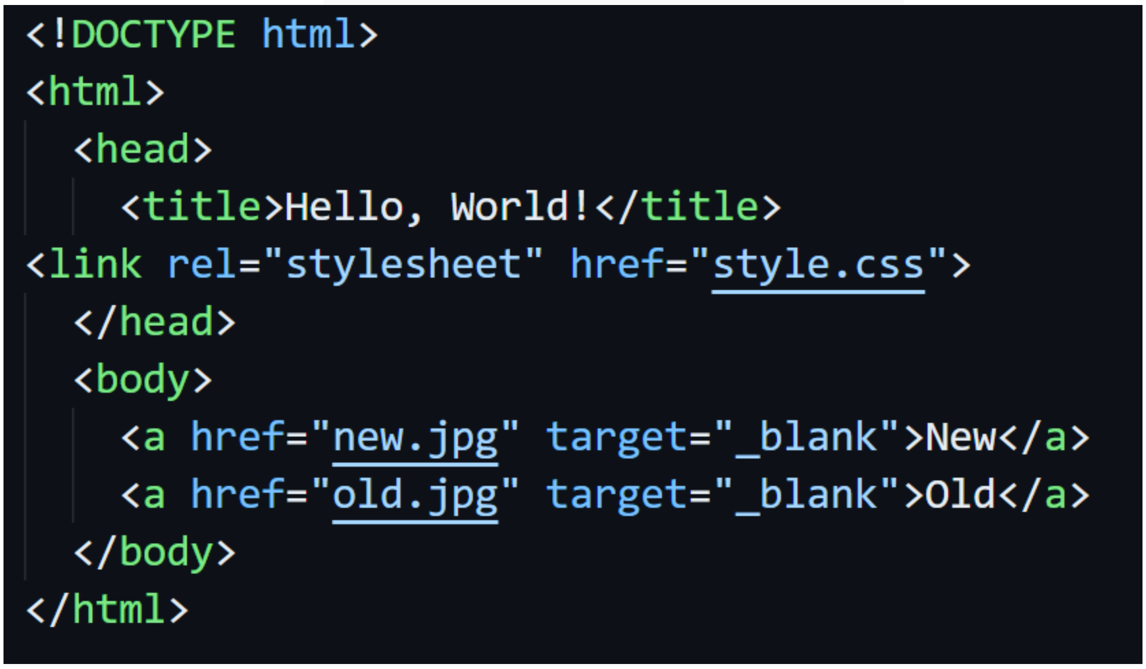

```html
<html>
  <head>
    <link rel="stylesheet" href="style.css" />
    <title>Combinator Property</title>
  </head>
  <body>
    <div class="a">
      container
      <p>p1</p>
      <p>p2</p>
      <div class="b">
        <p>new</p>
      </div>
      <p>p3</p>
    </div>
    <p>p4</p>
    <p>p5</p>
    <h1>Hello</h1>
    <p>p6</p>
    <p>p7</p>
  </body>
</html>
```


```css
div p{
    background-color: red;
}
.b>p{
    background-color: aqua;
}
.b+p{
    background-color: blue;
}
.a~p{
    background-color: chartreuse;
}
```

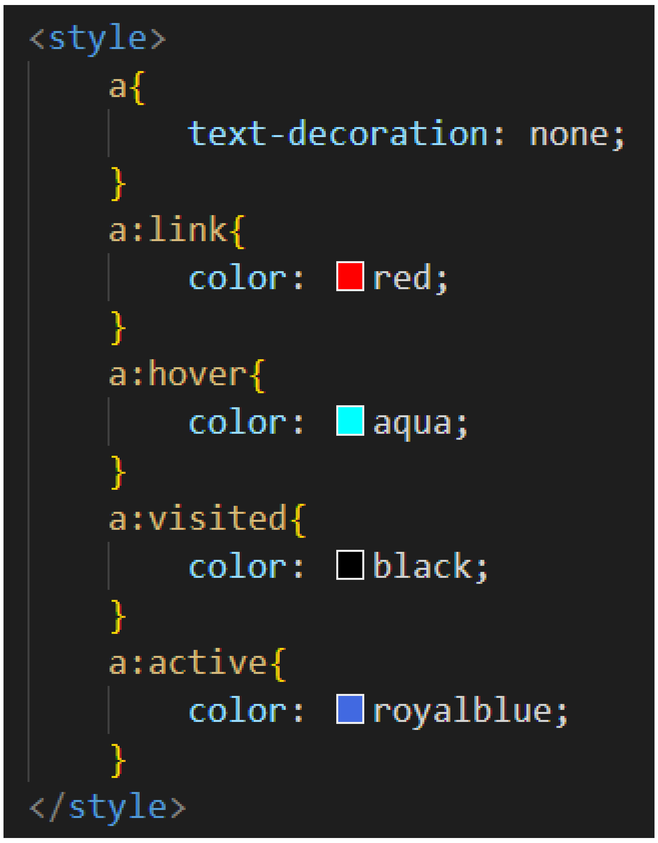
CSS Pseudo-classes =>

A Pseudo class in CSS is used to define the special state of an element. It can be combined with a CSS selector to add an effect to existing elements based on their states. For Example, changing the style of an element when the user hovers over it, or when a link is visited. All of these can be done using Pseudo Classes in CSS. :focus :not

```
selector: pseudo-class{
    property: value;
}
```

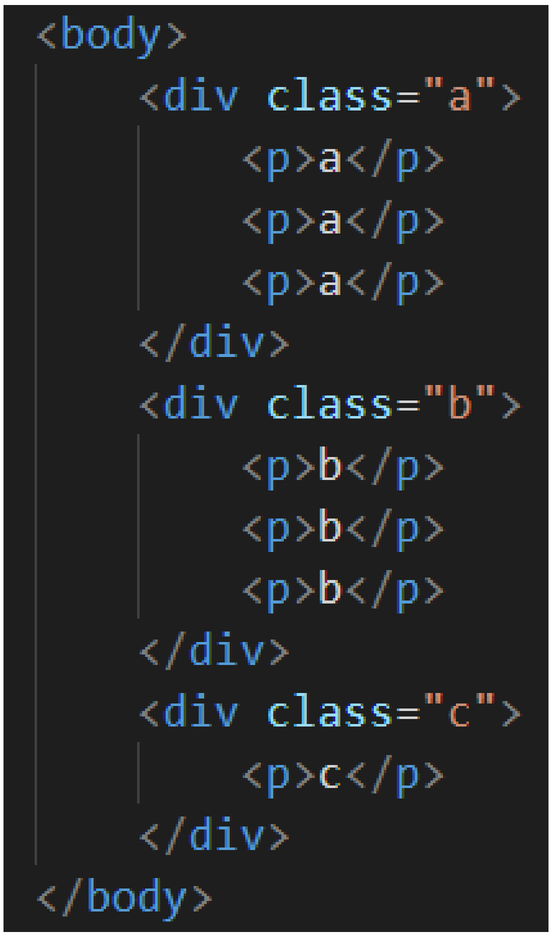


```html
<!DOCTYPE html>
<html>
  <head>
    <title>Hello, World!</title>
<link rel="stylesheet" href="style.css">
  </head>
  <body>
    <a href="new.jpg" target="_blank">New</a>
    <a href="old.jpg" target="_blank">Old</a>
  </body>
</html>
```

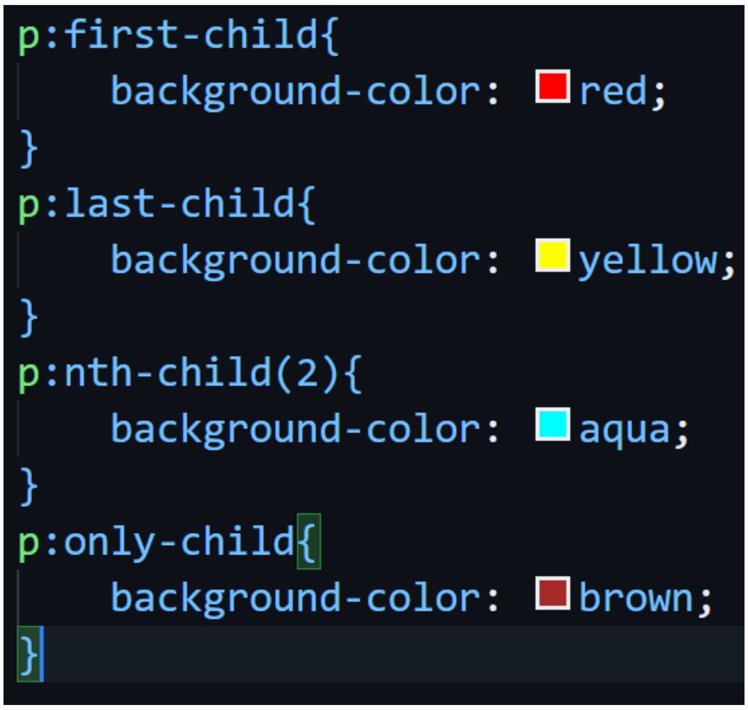


```css
<style>
    a{
        text-decoration: none;
    }
    a:link{
        color: red;
    }
    a:hover{
        color: aqua;
    }
    a:visited{
        color: black;
    }
    a:active{
        color: royalblue;
    }
</style>
```

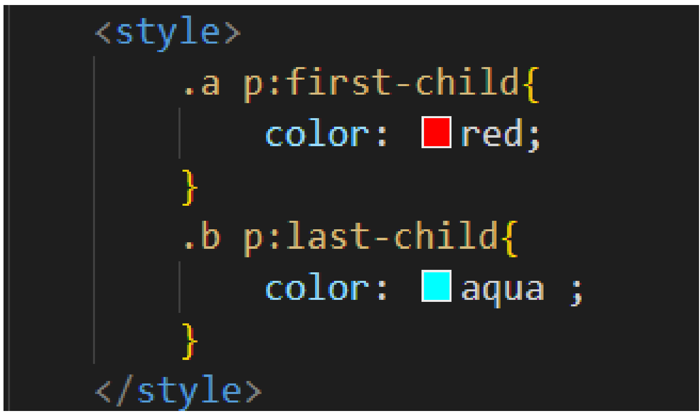
First child, Last child, nth child, Only child



```html
<body>
    <div class="a">
        <p>a</p>
        <p>a</p>
        <p>a</p>
    </div>
    <div class="b">
        <p>b</p>
        <p>b</p>
        <p>b</p>
    </div>
    <div class="c">
        <p>c</p>
    </div>
</body>
```



```css
p:first-child{
    background-color: 🟥red;
}
p:last-child{
    background-color: 🟨yellow;
}
p:nth-child(2){
    background-color: 🟦aqua;
}
p:only-child{
    background-color: 🟫brown;
}
```

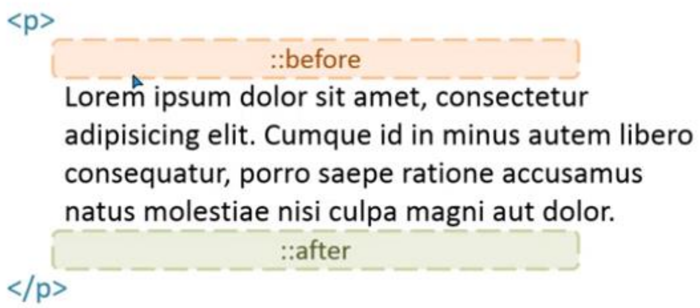


```html
<style>
    .a p:first-child{
        color: 🟥red;
    }
    .b p:last-child{
        color: 🟦aqua ;
    }
</style>
```

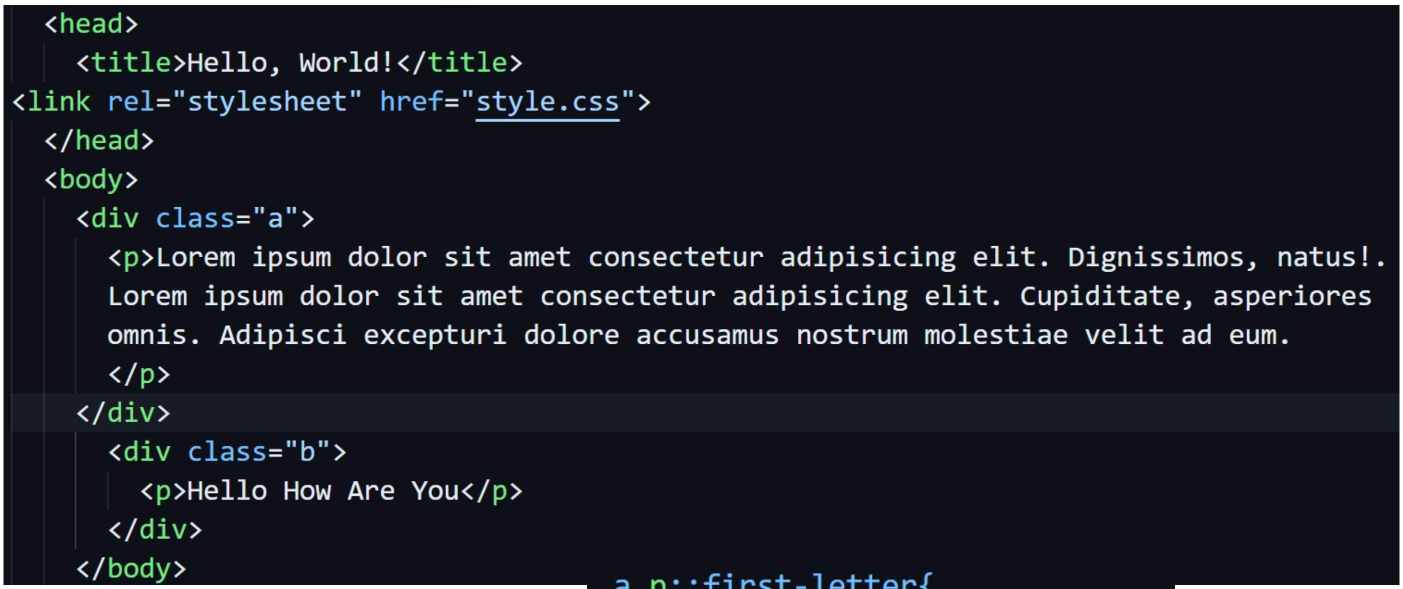
CSS Pseudo Elements =>

```
selector::pseudo-element {
      property: value;
}
```

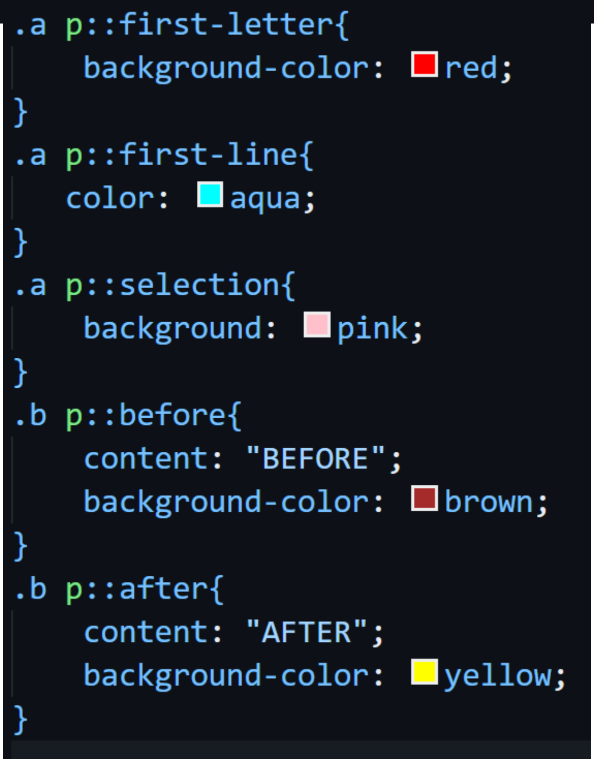
::after & ::before



<p>
::before
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cumque id in minus autem libero consequatur, porro saepe ratione accusamus natus molestiae nisi culpa magni aut dolor.
::after
</p>



```
  <head>
    <title>Hello, World!</title>
<link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="a">
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dignissimos, natus!.
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, asperiores
      omnis. Adipisci excepturi dolore accusamus nostrum molestiae velit ad eum.
      </p>
    </div>
      <div class="b">
        <p>Hello How Are You</p>
      </div>
    </body>
```



```
.a p::first-letter{
    background-color: red;
}
.a p::first-line{
   color: aqua;
}
.a p::selection{
    background: pink;
}
.b p::before{
    content: "BEFORE";
    background-color: brown;
}
.b p::after{
    content: "AFTER";
    background-color: yellow;
}
```

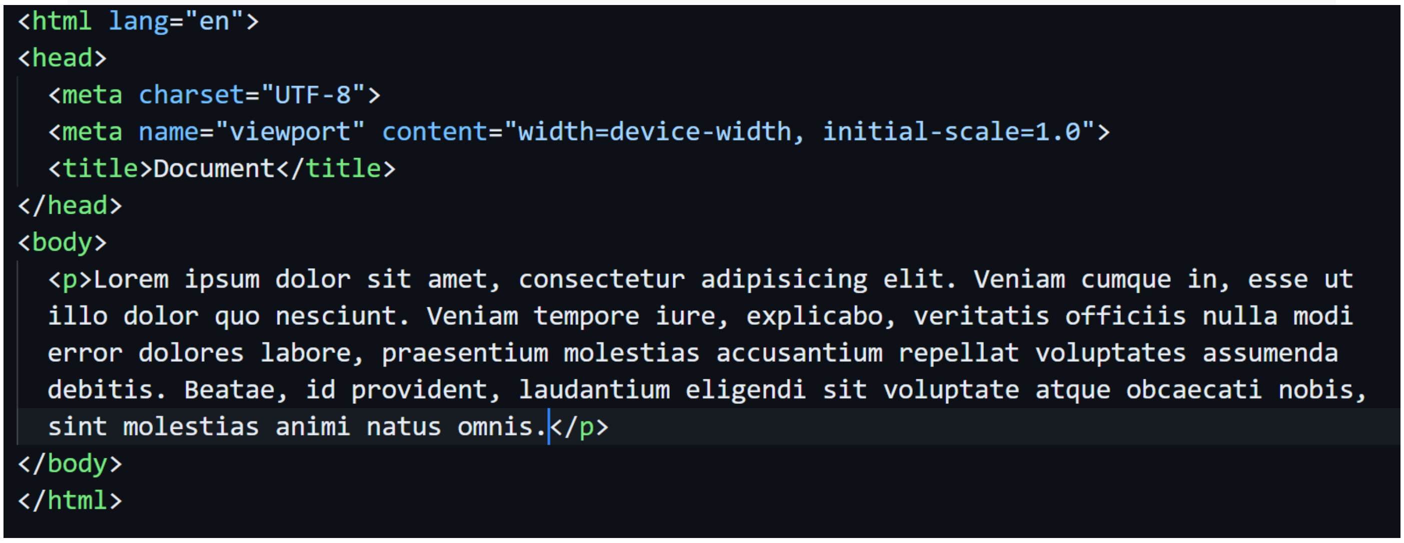
Static web Page=>

are the pages that doesn’t change the content or layout dynamically with every request to the web server. Static web pages display exactly the same information whenever anyone visits it. User sees the updated content of Static Web pages only when a web author manually updates them with a **text editor** or any web editing tool used for creating websites.

Viewport Meta tag=>

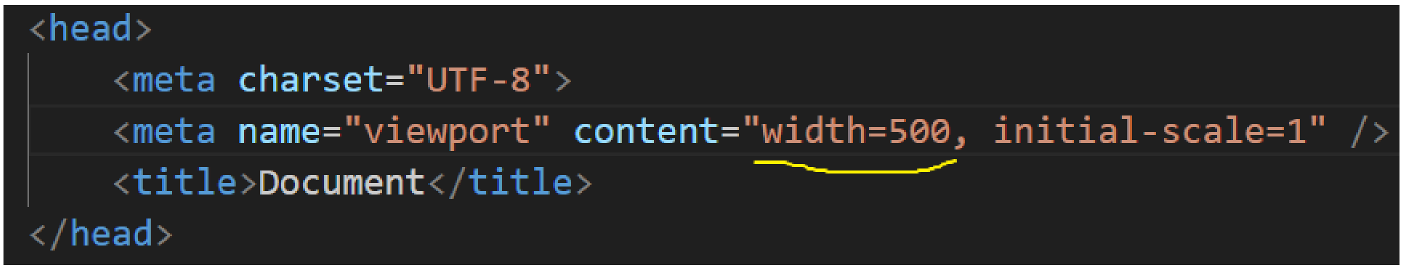
A Browser’s viewport is the area of web page in which the content is visible to the user. The viewport does not have the same size, it varies with the variation in screen size of the devices on which the website is visible. For a laptop, the viewport has a larger size as compared to a smartphone or tablet.

```
<meta name="viewport" content= "width=device-width, initial-
scale=1.0">
```



```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Veniam cumque in, esse ut
  illo dolor quo nesciunt. Veniam tempore iure, explicabo, veritatis officiis nulla modi
  error dolores labore, praesentium molestias accusantium repellat voluptates assumenda
  debitis. Beatae, id provident, laudantium eligendi sit voluptate atque obcaecati nobis,
  sint molestias animi natus omnis.</p>
</body>
</html>
```

This is the common setting of viewport used in various mobile-optimized websites. The width property governs the size of the viewport. It is possible to set it to a specific value (“width=600”) in terms of CSS pixels. Here it is set to a special value(“width= device-width”) which is the width of the device in terms of CSS pixels at a scale of 100%. The initial-scale property governs the zoom level when the page is loaded for the first time.



```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=500, initial-scale=1" />
    <title>Document</title>
</head>
```