Background-color property –by this we can change the back ground of the color

```
.one{
background-color: red;
}
```

$ Background-color property – by this we can change the back ground of the color

```
.one{
background-color: red;
}
```

$ background-image property – by this we can set image in the background

```
body{
background-image:url(“one.png”);
}
```

If the image is same then by default it will repeat in the background

$ Background repeat property

```
body{
background-repeat: repeat-x / repeat-y/ no-repreat;
}
```

$ Background size property

```
body{
background-size: auto/ cover/ contain/ 100% / 20px 30px;      width height
}
```

$ Background position property

```
body{
background-postion: left top;
}
```

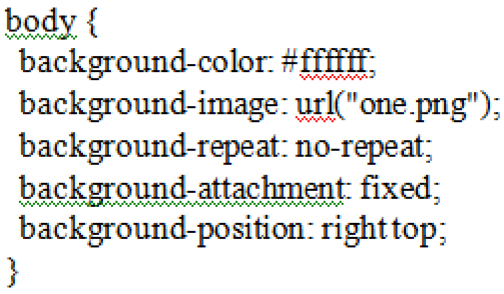
$ Background attachment property

```
body{
background-image: url("one.png");
background-repeat: no-repeat;
background-attachment: fixed;
```

}

$ Background Shorthand –

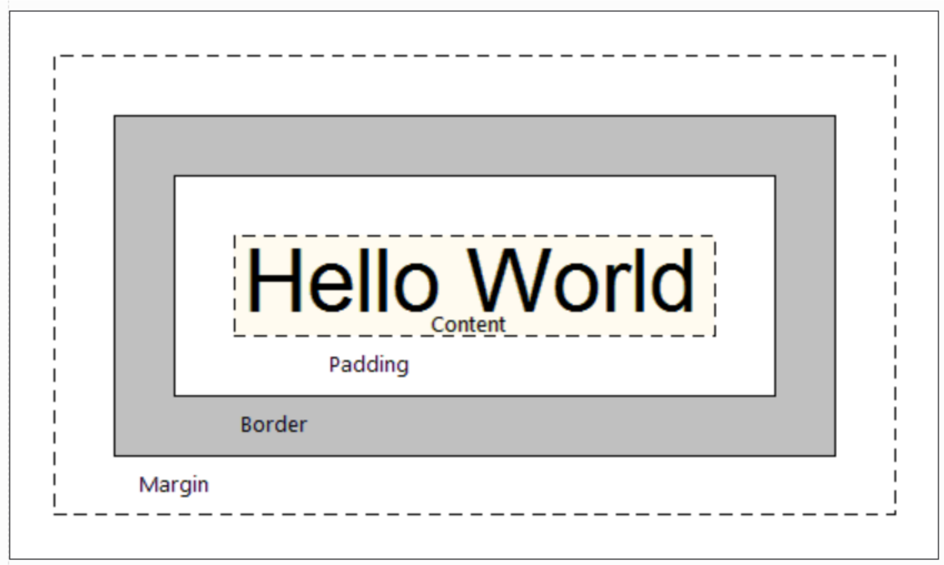
```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

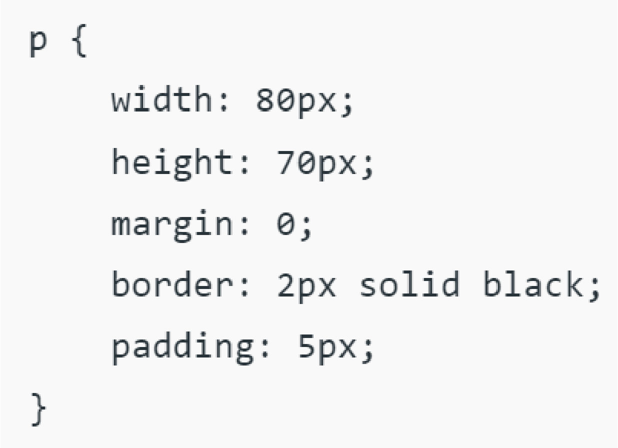


```
body {
  background-color: #ffffff;
  background-image: url("one.png");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: right top;
}
```

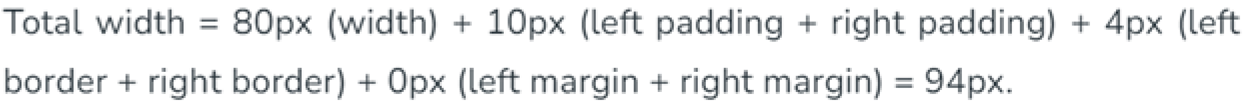
$ CSS Box model

The CSS box model is a container that contains multiple properties including borders, margins, padding, and the content itself. It is used to create the design and layout of web pages.

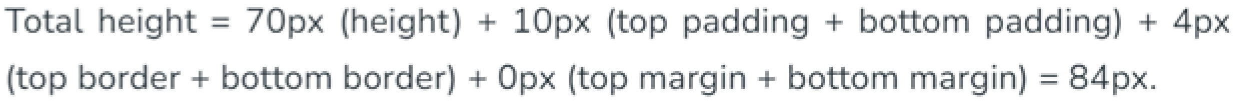




```
p {
    width: 80px;
    height: 70px;
    margin: 0;
    border: 2px solid black;
    padding: 5px;
}
```



Total width = 80px (width) + 10px (left padding + right padding) + 4px (left border + right border) + 0px (left margin + right margin) = 94px.



Total height = 70px (height) + 10px (top padding + bottom padding) + 4px (top border + bottom border) + 0px (top margin + bottom margin) = 84px.

Margin-

box{

margin: 7px 10px 5px 9px;

top right bottom left

}

box{

margin: 7px 9px;

top & bottom left & right

}

box{

margin-top:7px;

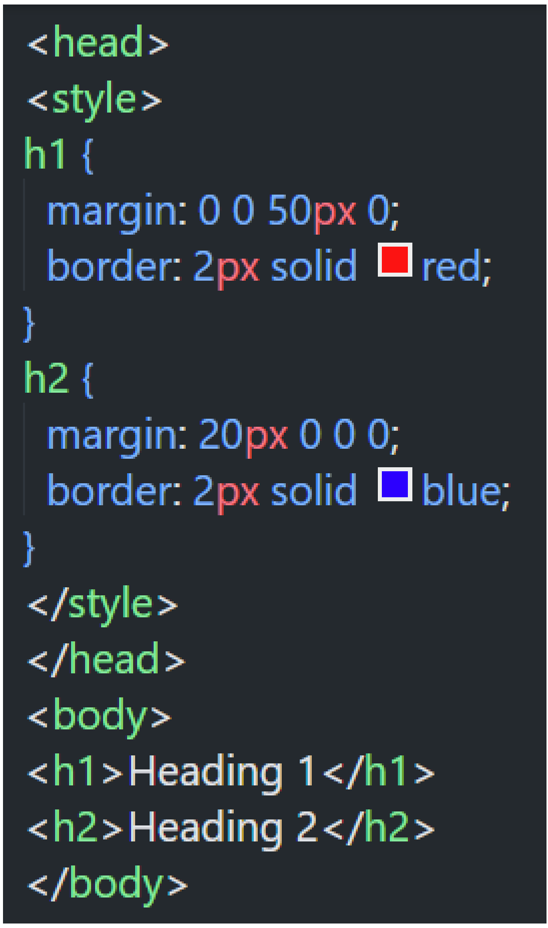
margin-bottom:7px;

margin-left:7px;

margin-right:7px;

}

Margin collapse – when two margins from different elements overlaps the real margin is the greater one. This is called margin collapse.



```
<head>
<style>
h1 {
  margin: 0 0 50px 0;
  border: 2px solid 🟥red;
}
h2 {
  margin: 20px 0 0 0;
  border: 2px solid 🟦blue;
}
</style>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
</body>
```

box-sizing –

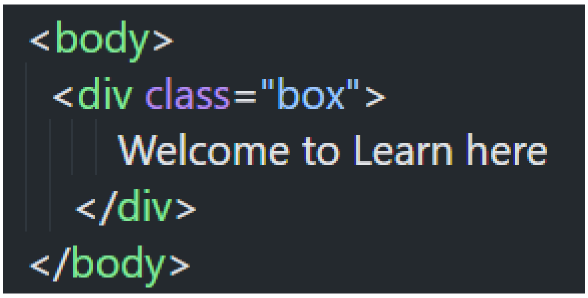
box{

box-sizing: content-box / border-box;

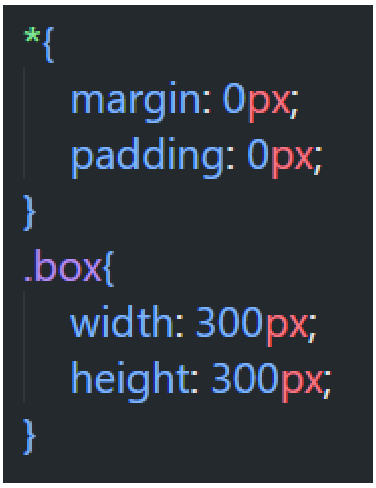
}

content-box: This is the default value of the box-sizing property. In this mode, the width and height properties include only the content.

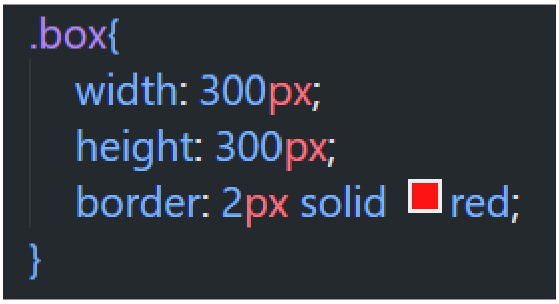
border-box: The content width and height include content + padding + border.



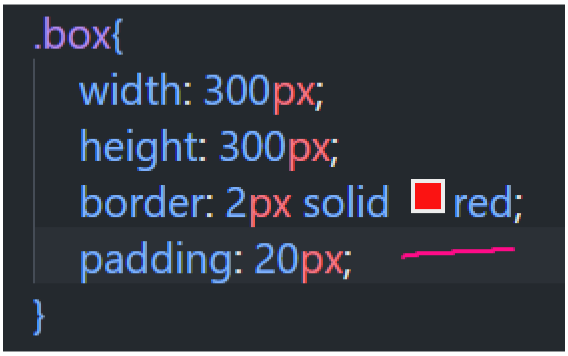
```
<body>
  <div class="box">
      Welcome to Learn here
   </div>
</body>
```



```
*{
    margin: 0px;
    padding: 0px;
}
.box{
    width: 300px;
    height: 300px;
}
```

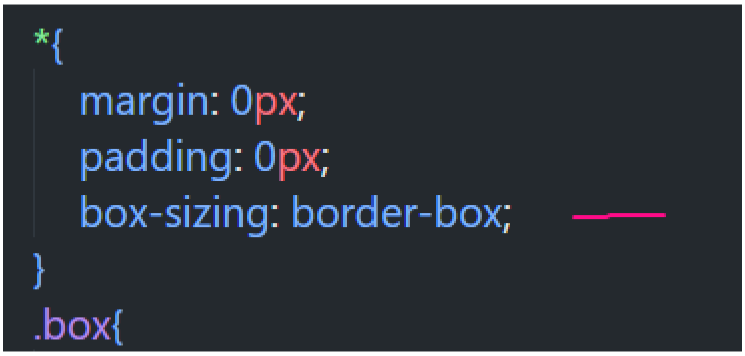


```
.box{
    width: 300px;
    height: 300px;
    border: 2px solid 🟥red;
}
```

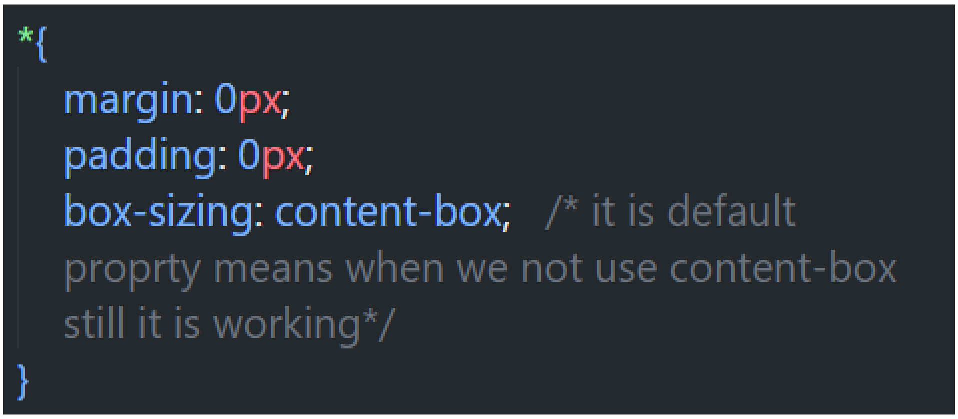


```
.box{
    width: 300px;
    height: 300px;
    border: 2px solid 🟥red;
    padding: 20px;
}
```

Now we add box sizing- border box by this the actual dimension of our box never change. But content size will change.



```
*{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
.box{
```



```
*{
    margin: 0px;
    padding: 0px;
    box-sizing: content-box;   /* it is default
    proprty means when we not use content-box
    still it is working*/
}
```

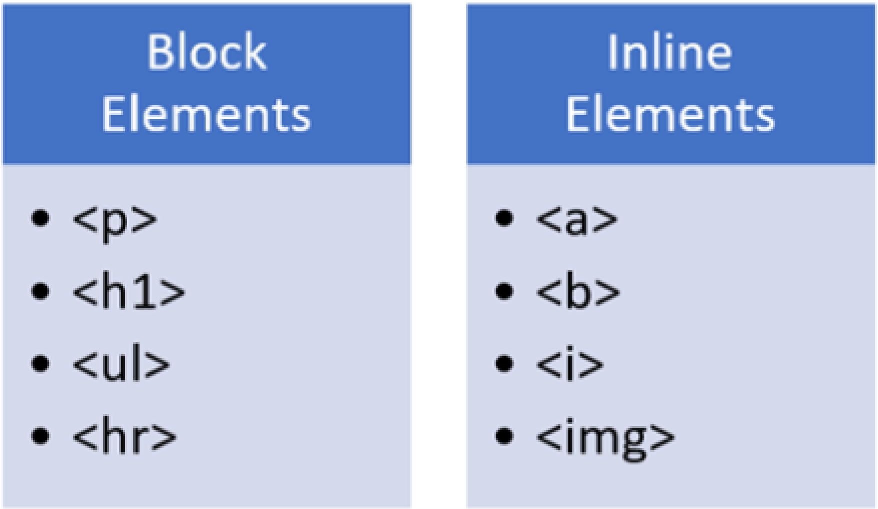
Display property- It will defines how the components(div, hyperlink, heading, etc) are going to be placed on the web page. Block, inline, inline-block, none

Display property is use to display different different elements(div, p, a, h1) on our screen. On different different elements different different display property are applicable or on some elements some display properties are not applicable.

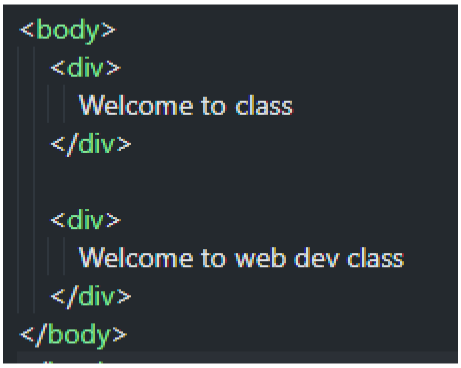
https://css-tricks.com/almanac/properties/d/display/ web site

Block elements =>

We also have block elements like div, p, h1

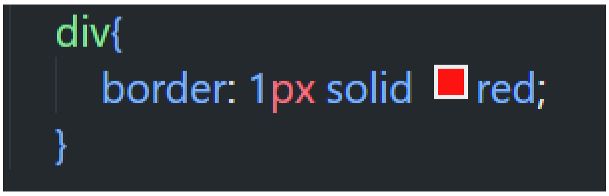


Block Elements	Inline Elements
<p>	<a>
<h1>	<b>
<ul>	<i>
<hr>	<img>



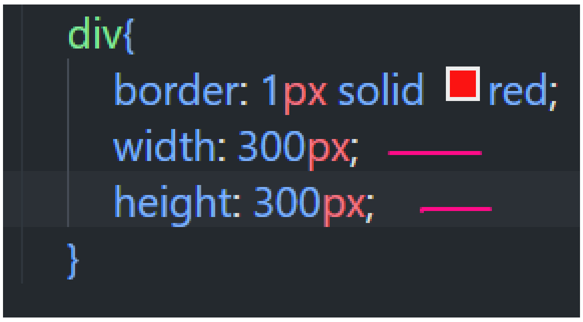
```
<body>
    <div>
        Welcome to class
    </div>

    <div>
        Welcome to web dev class
    </div>
</body>
```

when we see o/p of this div we can see that both div are coming on different different lines. By this we can know that every block element cones on new like becz dive covers all available space. Block elements take width from extreme left to extreme right.



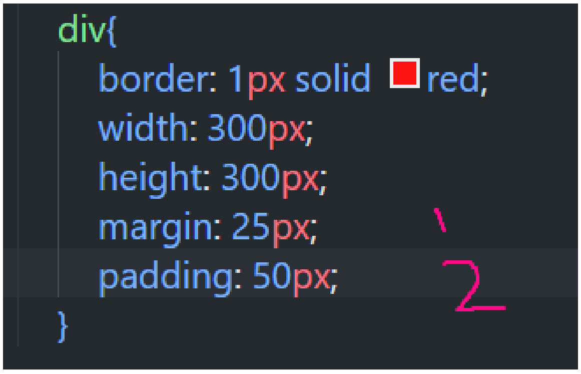
```
div{
    border: 1px solid 🟥red;
}
```

We can also give custom width and height to block elements like

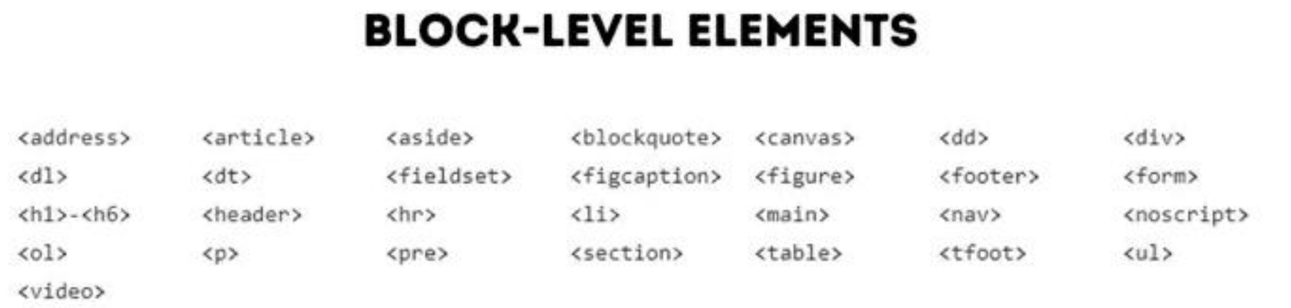


```
div{
    border: 1px solid 🟥red;
    width: 300px;
    height: 300px;
}
```

We can also give margin and padding to our block elements

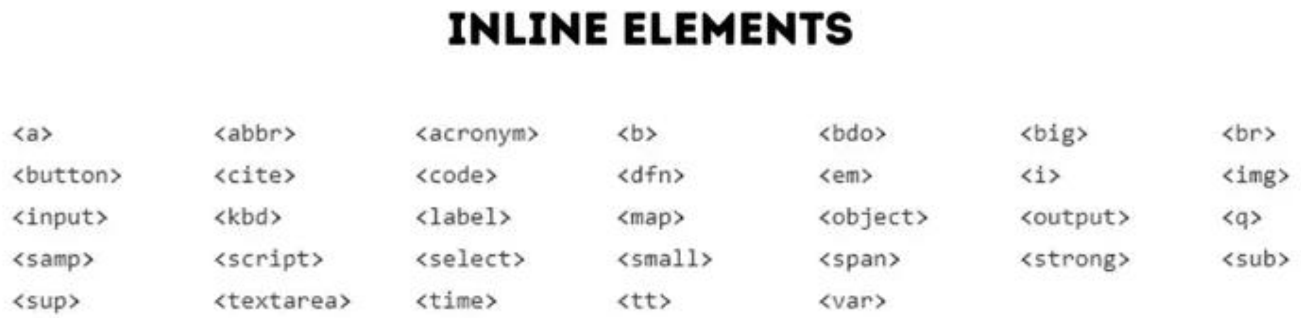


```
div{
    border: 1px solid 🟥red;
    width: 300px;
    height: 300px;
    margin: 25px;
    padding: 50px;
}
```



BLOCK-LEVEL ELEMENTS

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>	<div>
<dl>	<dt>	<fieldset>	<figcaption>	<figure>	<footer>	<form>
<h1>-<h6>	<header>	<hr>	<li>	<main>	<nav>	<noscript>
<ol>	<p>	<pre>	<section>	<table>	<tfoot>	<ul>
<video>						

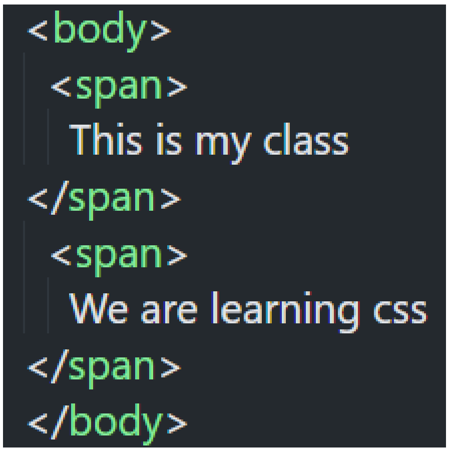


INLINE ELEMENTS

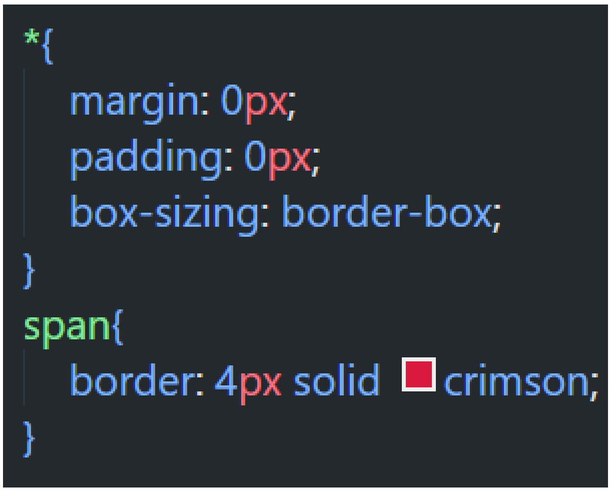
<a>	<abbr>	<acronym>	<b>	<bdo>	<big>	<br>
<button>	<cite>	<code>	<dfn>	<em>	<i>	<img>
<input>	<kbd>	<label>	<map>	<object>	<output>	<q>
<samp>	<script>	<select>	<small>	<span>	<strong>	<sub>
<sup>	<textarea>	<time>	<tt>	<var>		

Inline elements =>

Inline elements always trying to come in single line becz now we are using only two span but if we have so many spans then rest of span goes to next line.



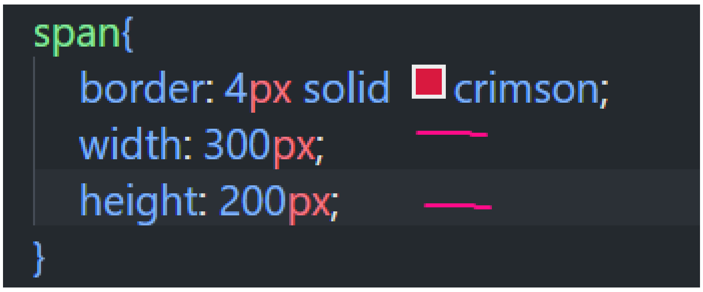
```
<body>
  <span>
    This is my class
</span>
  <span>
    We are learning css
</span>
</body>
```



```
*{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
span{
    border: 4px solid crimson;
}
```

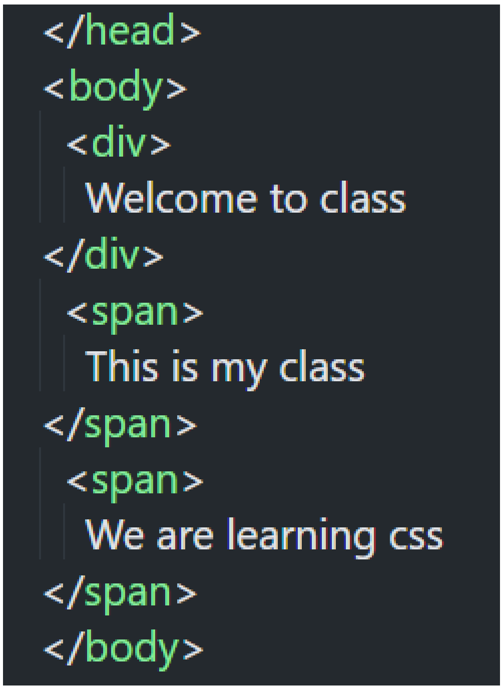
By observing the border of above code we can say that the width of our inline elements are same as we are have content width. Inline elements are not like block elements

We cannot apply custom width and custom height on inline elements.

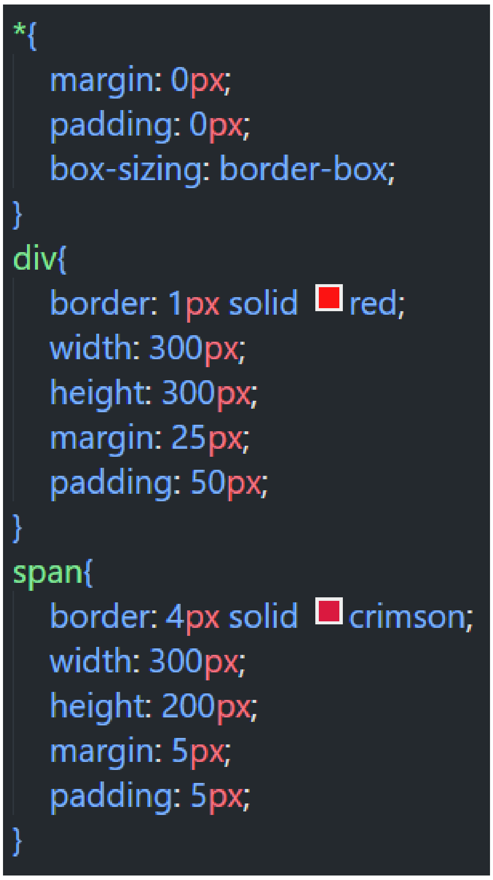


```
span{
    border: 4px solid crimson;
    width: 300px;
    height: 200px;
}
```

We cannot apply proper custom margin in the case of inline element becz custom margin in inline elements are browser dependent. But we can apply custom padding on inline elements

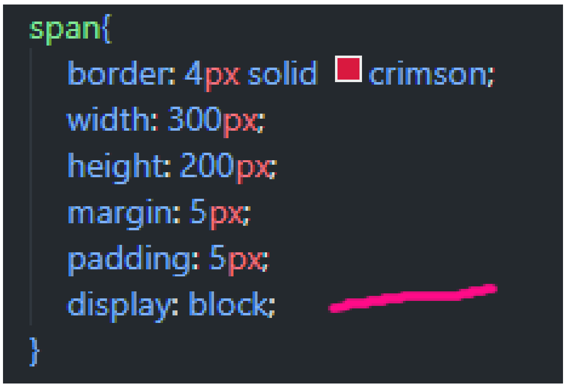


```
</head>
<body>
  <div>
   Welcome to class
</div>
  <span>
   This is my class
</span>
  <span>
   We are learning css
</span>
</body>
```



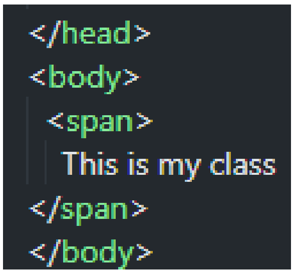
```
*{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
div{
    border: 1px solid 🟥red;
    width: 300px;
    height: 300px;
    margin: 25px;
    padding: 50px;
}
span{
    border: 4px solid 🟥crimson;
    width: 300px;
    height: 200px;
    margin: 5px;
    padding: 5px;
}
```

Can we convert inline elements into block elements. Yes can convert inline element into block element by explicitly using display-block property. And vice versa also like display-inline.

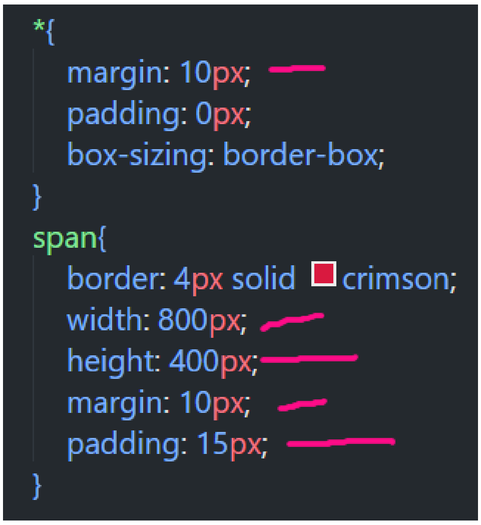


```
span{
    border: 4px solid 🟥crimson;
    width: 300px;
    height: 200px;
    margin: 5px;
    padding: 5px;
    display: block;
}
```

These is another property which is mixture of both inline and block element that is display-inline-block. This inline-block property is behave like inline but we can use our custom width and custom height.



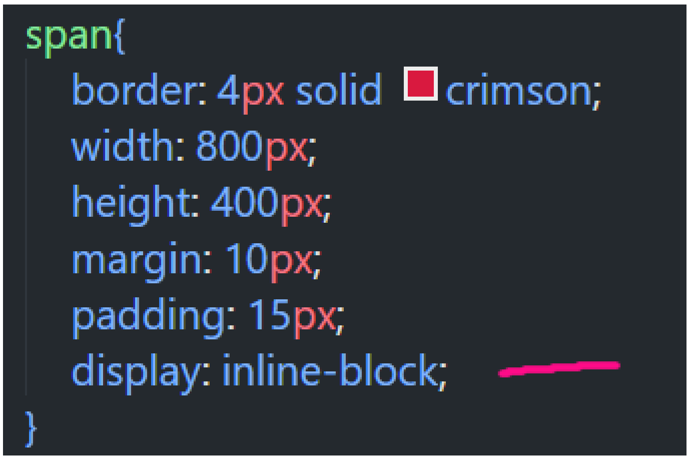
```
</head>
<body>
  <span>
    This is my class
</span>
</body>
```



```
*{
    margin: 10px;
    padding: 0px;
    box-sizing: border-box;
}
span{
    border: 4px solid crimson;
    width: 800px;
    height: 400px;
    margin: 10px;
    padding: 15px;
}
```

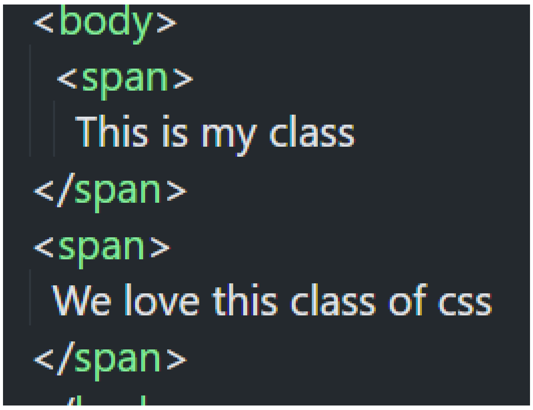
When we see its o/p then we are not able to see 800px and 400px

Now use inline-block

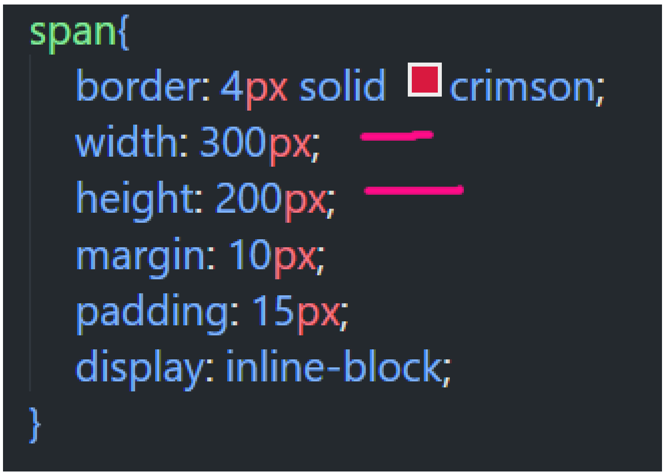


```
span{
    border: 4px solid crimson;
    width: 800px;
    height: 400px;
    margin: 10px;
    padding: 15px;
    display: inline-block;
}
```

Now see o/p that we are able to apply custom width and height on inline elements



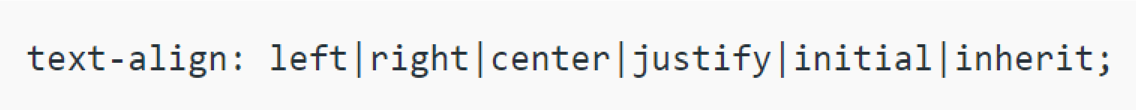
```
<body>
  <span>
    This is my class
</span>
<span>
  We love this class of css
</span>
```



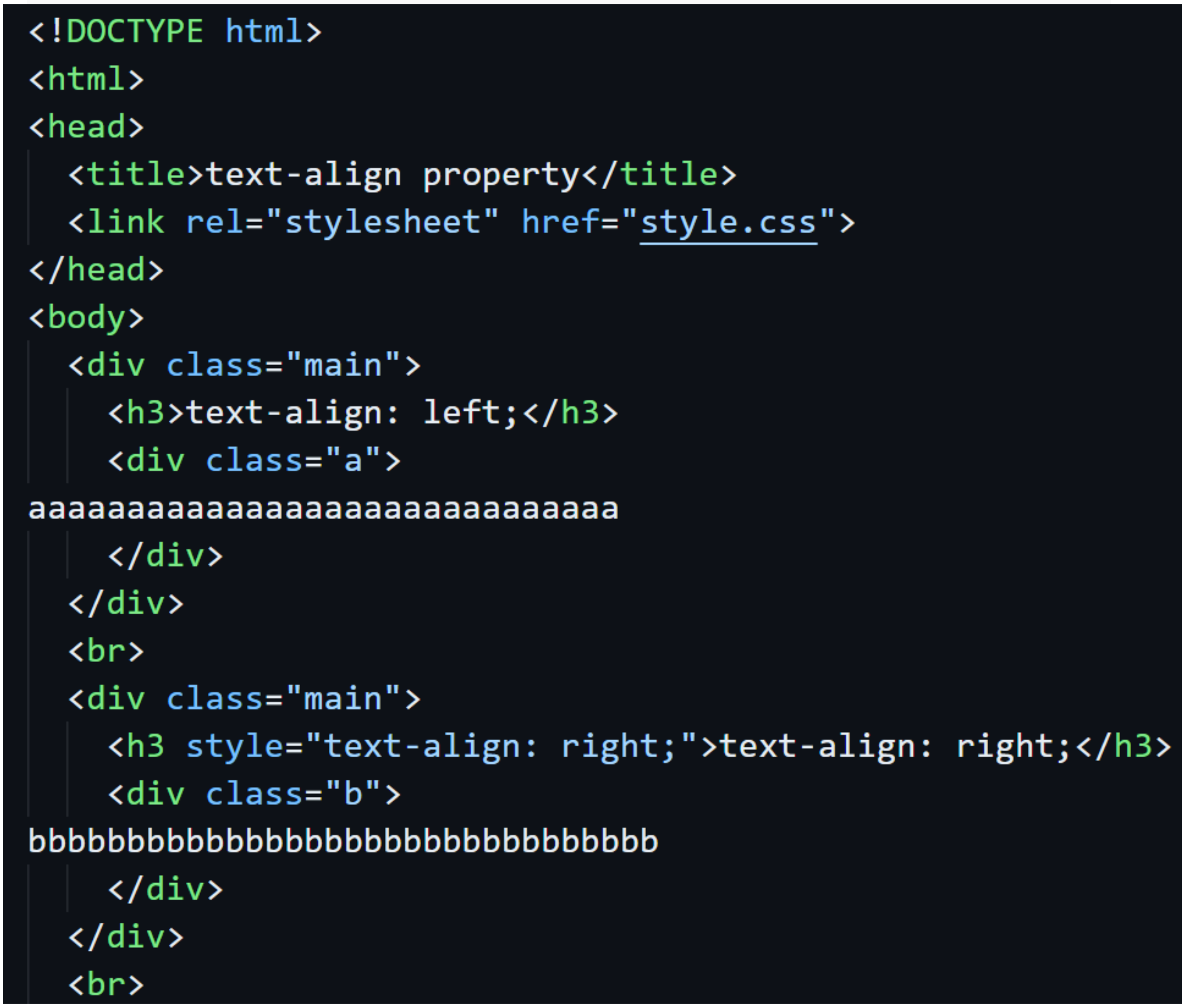
```
span{
    border: 4px solid crimson;
    width: 300px;
    height: 200px;
    margin: 10px;
    padding: 15px;
    display: inline-block;
}
```

We can see in above code how inline elements are acting like block elements.

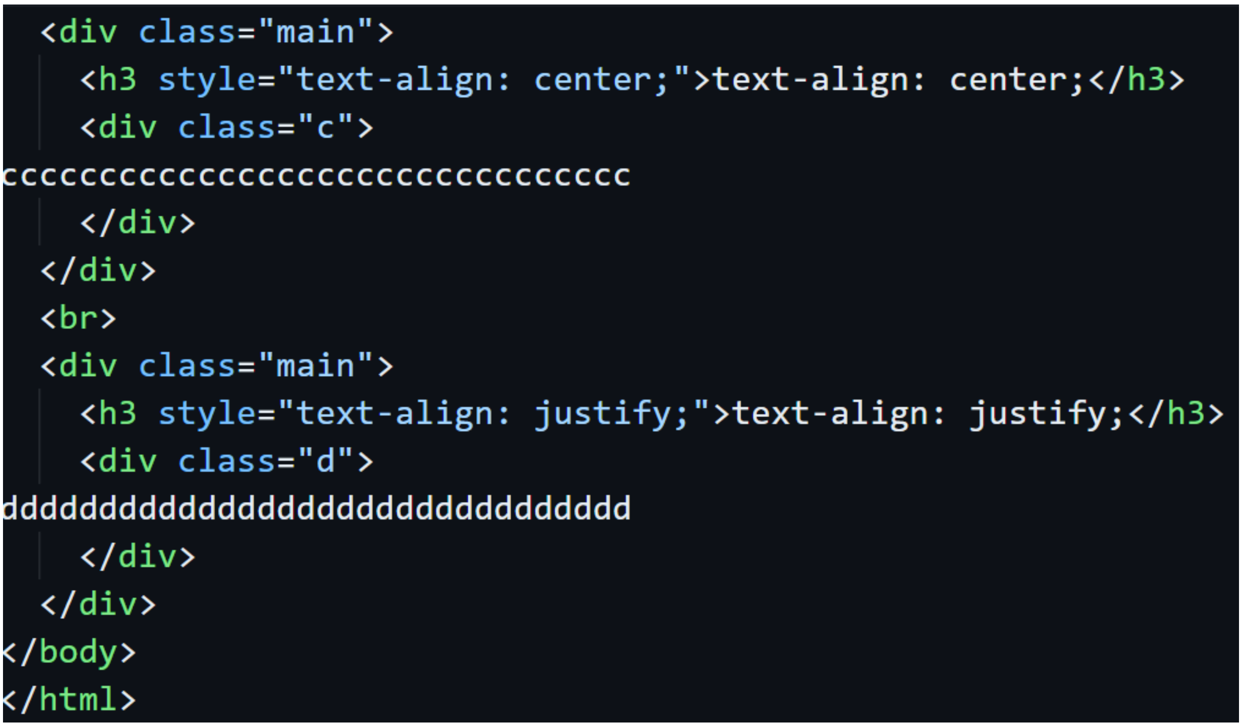
Text align-



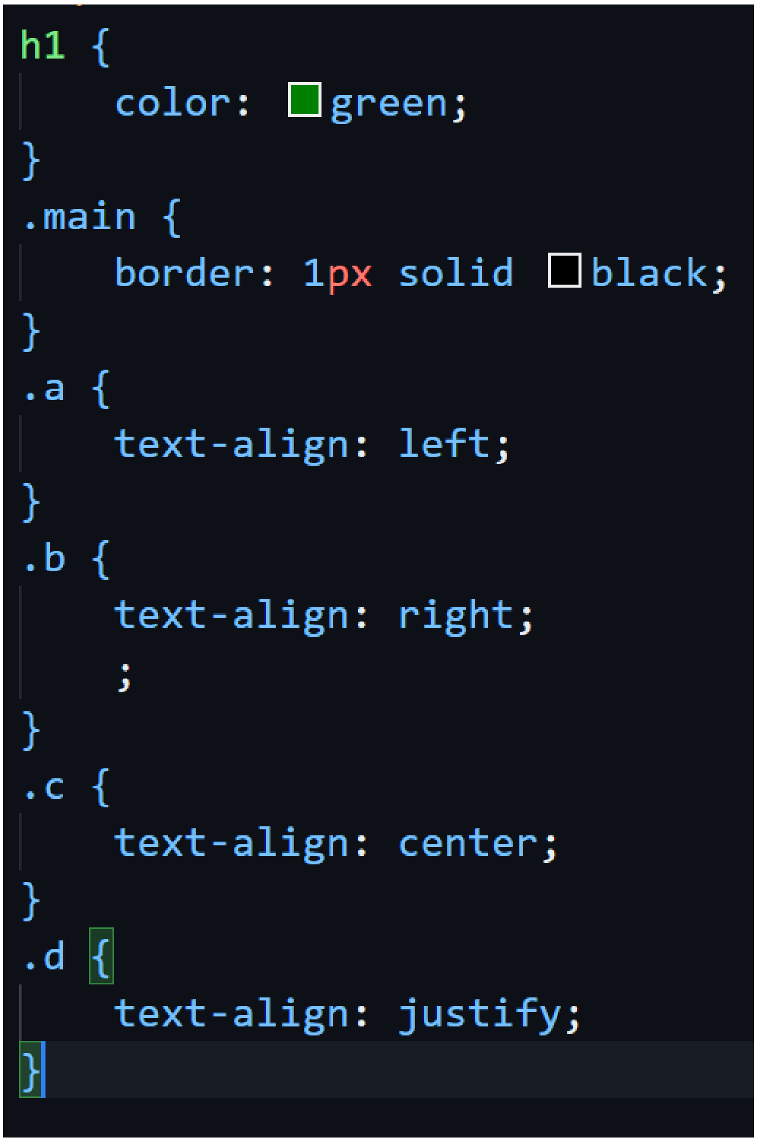
```
text-align: left|right|center|justify|initial|inherit;
```



```
<!DOCTYPE html>
<html>
<head>
  <title>text-align property</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="main">
    <h3>text-align: left;</h3>
    <div class="a">
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    </div>
  </div>
  <br>
  <div class="main">
    <h3 style="text-align: right;">text-align: right;</h3>
    <div class="b">
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
    </div>
  </div>
  <br>
```



```
  <div class="main">
    <h3 style="text-align: center;">text-align: center;</h3>
    <div class="c">
cccccccccccccccccccccccccccccccc
    </div>
  </div>
  <br>
  <div class="main">
    <h3 style="text-align: justify;">text-align: justify;</h3>
    <div class="d">
dddddddddddddddddddddddddddddddd
    </div>
  </div>
</body>
</html>
```



```
h1 {
    color: 🟩green;
}
.main {
    border: 1px solid ⬛black;
}
.a {
    text-align: left;
}
.b {
    text-align: right;
    ;
}
.c {
    text-align: center;
}
.d {
    text-align: justify;
}
```