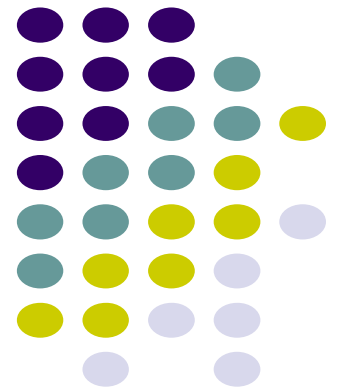
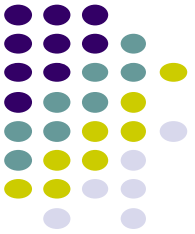


CSS Box model and its properties

Dr. Arul Xavier V M
Assistant Professor





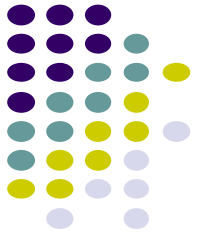
The CSS Box Model

- All HTML elements can be considered as boxes.
- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element.

Every element in web design is a “rectangular box”

The CSS Box Model

- The box model allows us to add a border around elements, and to define space between elements.



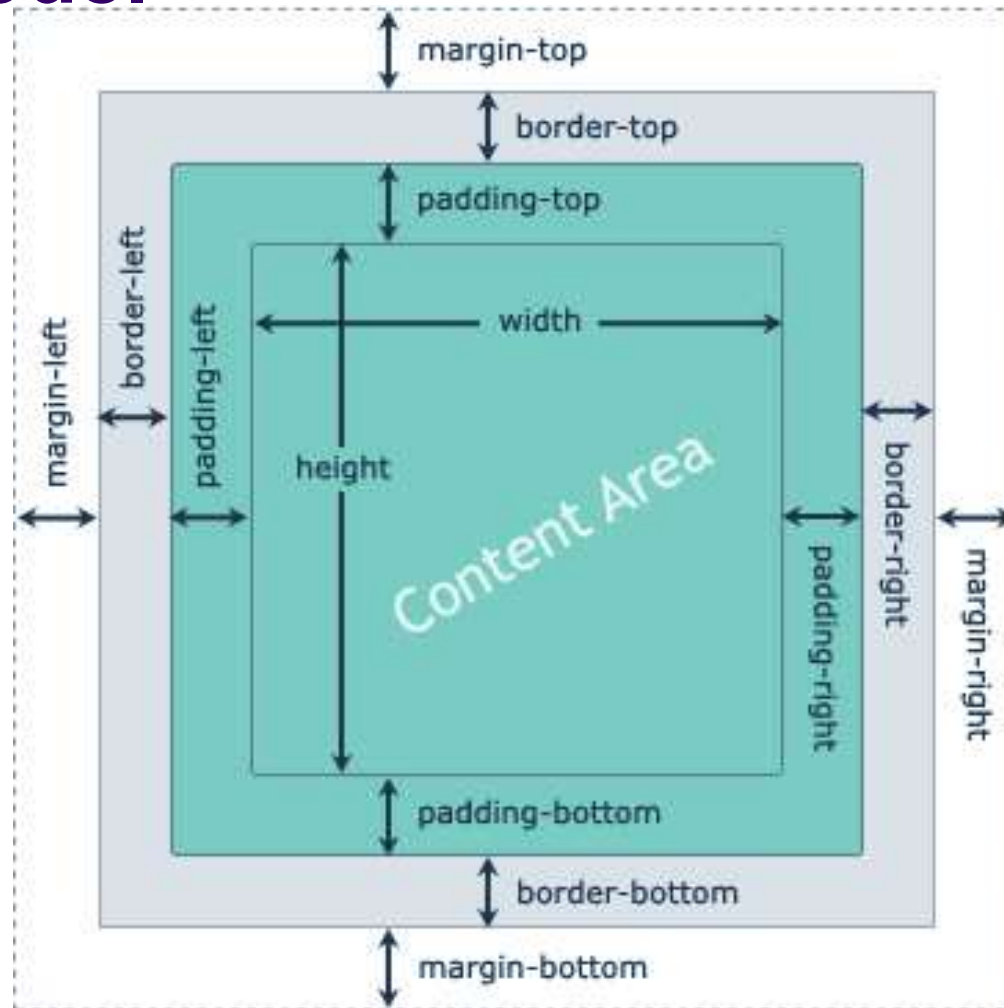
Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

CSS Box Model



CSS Box Model

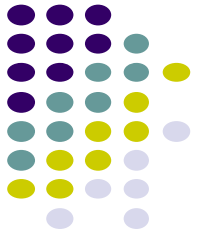
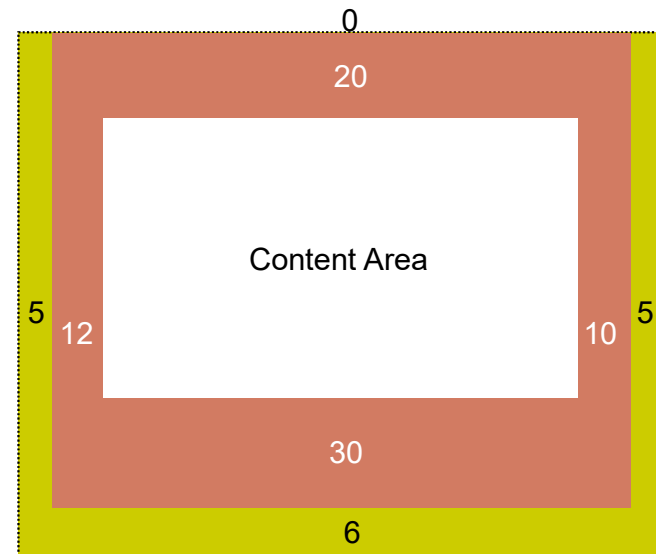


- Every CSS box is divided into regions, consisting of:
 - **Content**
 - Text & web page elements in the container
 - **Padding**
 - Space between the content and the border
 - **Border**
 - Space between the padding and the margin
 - **Margins**
 - Empty space between the element and adjacent elements

CSS Box Model

```
div {  
  margin-top: 0;  
  margin-right: 5px;  
  margin-bottom: 6px;  
  margin-left: 5px;  
  padding-top: 20px;  
  padding-right: 10px;  
  padding-bottom: 30px;  
  padding-left: 12px;  
}
```

<div>Content Area</div>



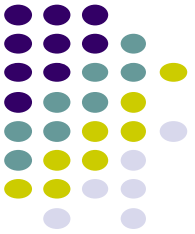
Setting width and Height of the Box



```
<head>
  <style>
    .b1 {
      background-color: yellow;
      width: 300px;
      height: 100px;
    }
    .b2 {
      background-color: blue;
      width: 500px;
      height: 200px;
    }
  </style>
</head>
<body>
  <div class="b1">Box1 (width=300px and Height=100px)</div>
  <div class="b2">Box2 (width=500px and Height=200px)</div>
</body>
```



Borders



- The core border properties are:
 - **Width:** absolute (px, in, cm, or 'thin', 'medium', 'thick'), or relative (em)
 - **Style:** dotted, dashed, solid, double, groove, ridge, inset, outset, hidden, etc
 - **Color:** 'blue', 'red', #FF9900, etc
- You can also create the effect of a border by using a graphic image in a CSS background property, instead of the border property

Setting border properties

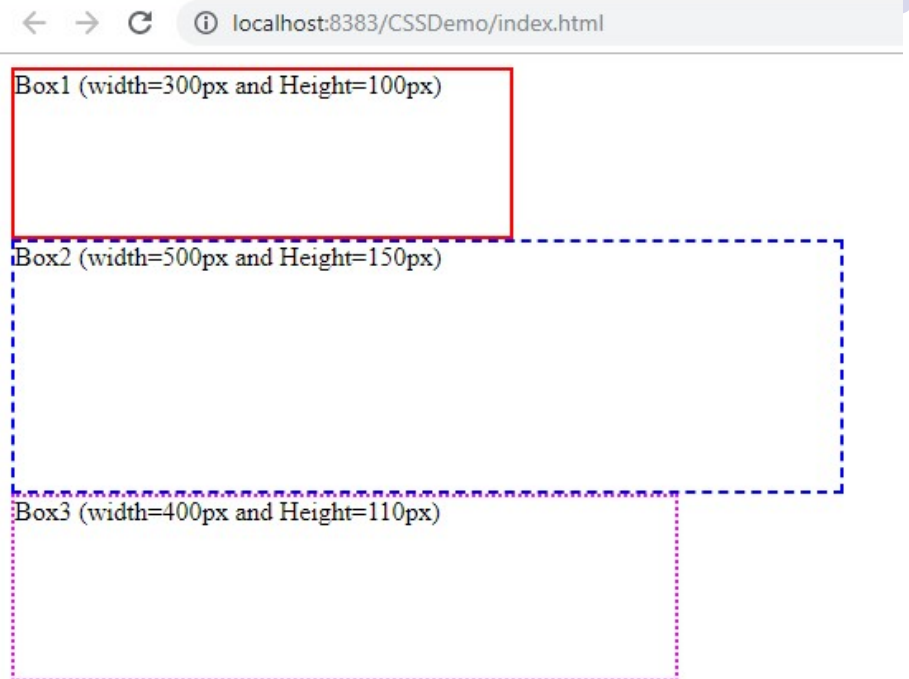


```
<style>
```

```
.b1 {  
  border-color: red;  
  border-style: solid;  
  border-width: 2px;  
  width: 300px;  
  height: 100px;  
}  
.b2 {  
  border-color: blue;  
  border-style: dashed;  
  border-width: 2px;  
  width: 500px;  
  height: 150px;  
}  
.b3 {  
  /*border shorthand property*/  
  border: 2px dotted magenta;  
  width: 400px;  
  height: 110px;  
}
```

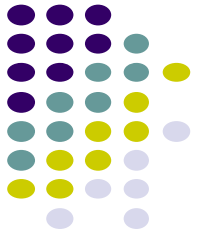
```
</style>
```

```
<div class="b1">Box1 (width=300px and Height=100px)</div>  
<div class="b2">Box2 (width=500px and Height=150px)</div>  
<div class="b3">Box3 (width=400px and Height=110px)</div>
```



Setting individual border

- You can also set the border based on the side such as top, right, bottom and left with different width, style and thickness.

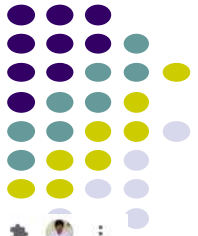


```
<style>
    .box1 {
        height: 50px;
        background-color: yellow;
        border-top: 4px solid blue;
        border-left: 4px solid red;
        border-bottom: 4px solid green;
    }
</style>
```

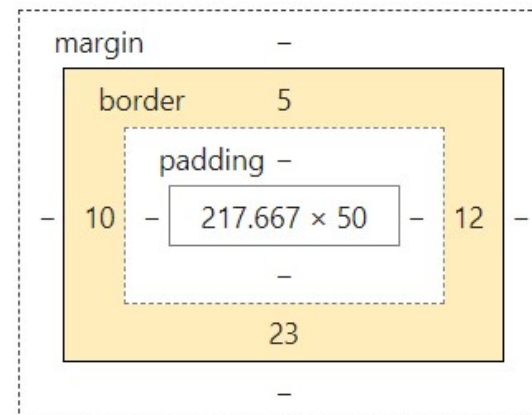
```
</head>
<body>
    <div class="box1"></div>
</body>
```



Setting border-width with multiple values



```
<meta name="viewport" content="width=device-width">
<style>
  body {margin: 0px;}
  .box1 {
    height: 50px;
    background-color: lightblue;
    border-style: solid;
    border-color: red;
    border-width: 5px 12px 23px 10px;
  }
</style>
</head>
<body>
  <div class="box1"></div>
  <div class="box2"></div>
</body>
```



Top = 5px
Right = 12px
Bottom = 23px
Left = 10px

Setting margin properties

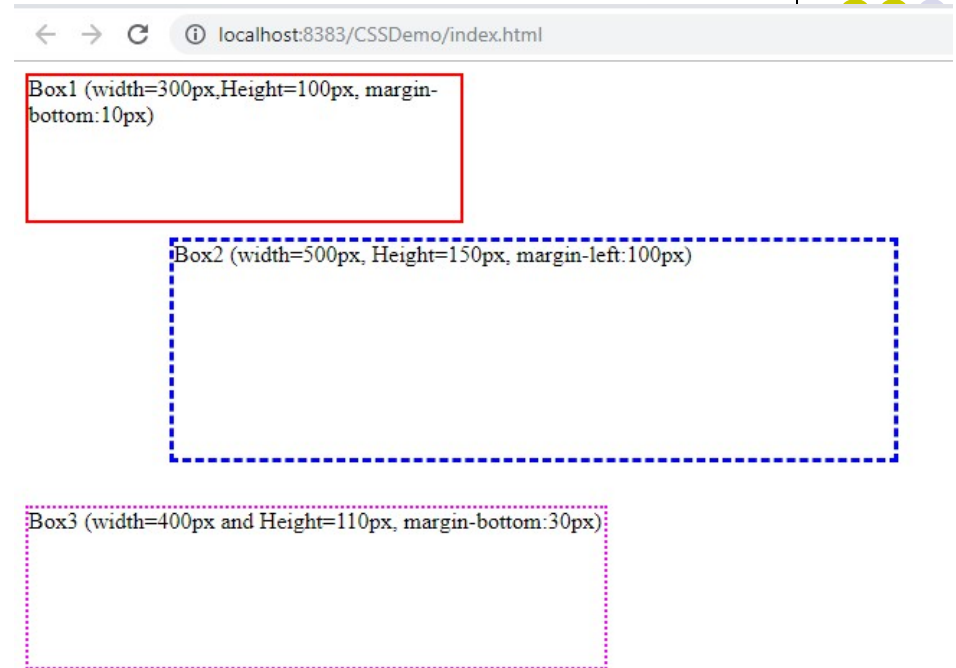


```
<style>
```

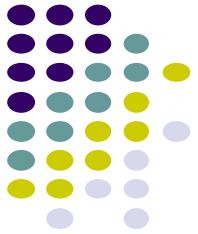
```
.b1 {  
  margin-bottom: 10px;  
  border: 2px solid red;  
  width: 300px;  
  height: 100px;  
}  
.b2 {  
  margin-left: 100px;  
  border: 3px dashed blue;  
  width: 500px;  
  height: 150px;  
}  
.b3 {  
  margin-top: 30px;  
  border: 2px dotted magenta;  
  width: 400px;  
  height: 110px;  
}
```

```
</style>
```

```
<div class="b1">Box1 (width=300px, Height=100px, margin-bottom:10px)</div>  
<div class="b2">Box2 (width=500px, Height=150px, margin-left:100px)</div>  
<div class="b3">Box3 (width=400px and Height=110px, margin-bottom:30px)</div>
```



Setting margin



- There are 4 ways you can set margin values(in pixels)
 - **Single value**
 - This single value will be applied to all margins such as **top**, **right**, **bottom** and **left**.
 - **Double value**
 - First value applied to **top** and **bottom**. Then second value applied to **right** and **left**.
 - **Three values**
 - First value to **top**, second to **right**, third to **bottom** and for left the second value will be applied
 - **Four values**
 - First value to **top**, second to **right**, third to **bottom**, and fourth to **left**

The above rule is applicable to border and padding also.

Setting Margin – Example – Single Value



```
<style>
  body {margin: 0px;}
  .box1 {
    width: 200px;
    height: 50px;
    background-color:blue;
    margin: 10px;
  }
  .box2 {
    width: 200px;
    height: 50px;
    background-color:yellow;
  }
</style>
```

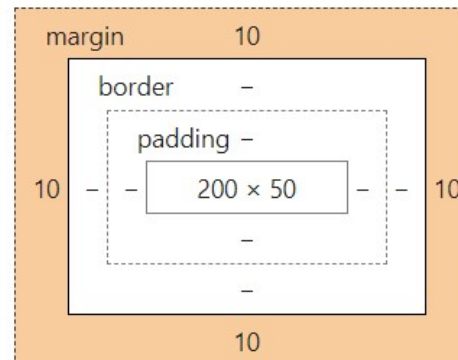
```
</head>
```

```
<body>
```

```
  <div class="box1"></div>
```

```
  <div class="box2"></div>
```

```
</body>
```

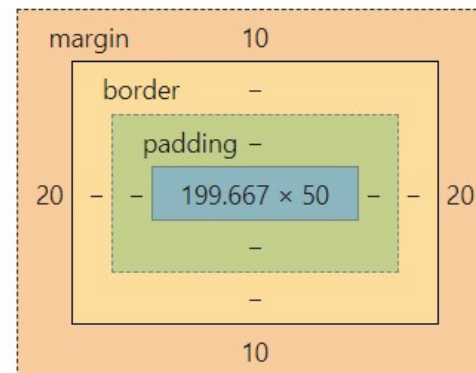
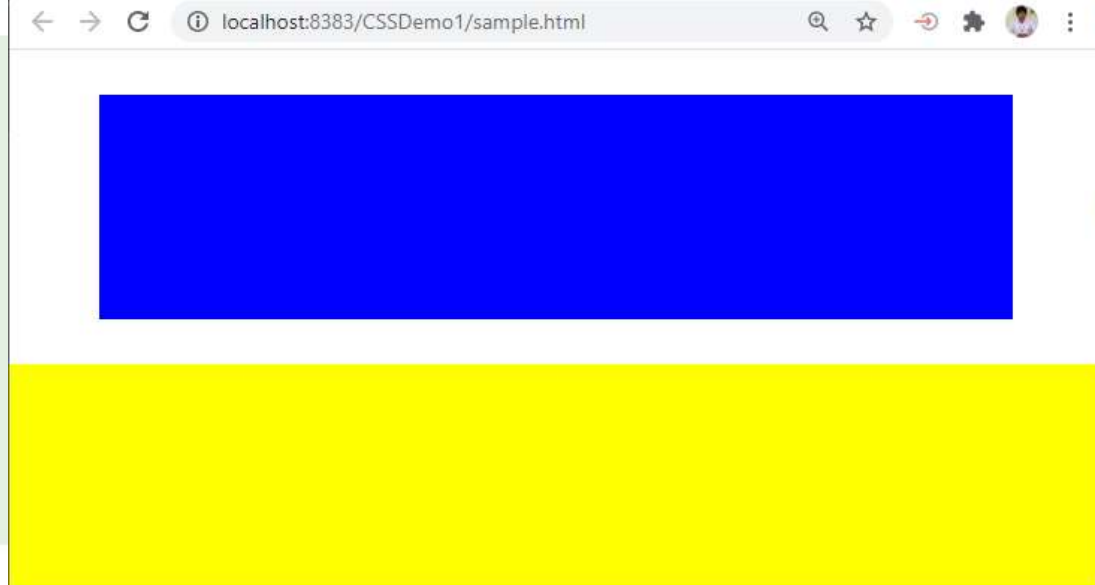


Top = 10px
Right = 10px
Bottom = 10px
Left = 10px

Setting Margin – Example – Two Values



```
<style>
  body {margin: 0px;}
  .box1 {
    height: 50px;
    background-color:blue;
    margin: 10px 20px;
  }
  .box2 {
    height: 50px;
    background-color:yellow;
  }
</style>
</head>
<body>
  <div class="box1"></div>
  <div class="box2"></div>
</body>
```



Top = 10px
Right = 20px
Bottom = 10px
Left = 20px

Setting Margin – Example – three Values



```
<style>
  body {margin: 0px;}
  .box1 {
    height: 50px;
    background-color:blue;
    margin: 10px 20px 12px;
  }
  .box2 {
    height: 50px;
    background-color:yellow;
  }
</style>
```

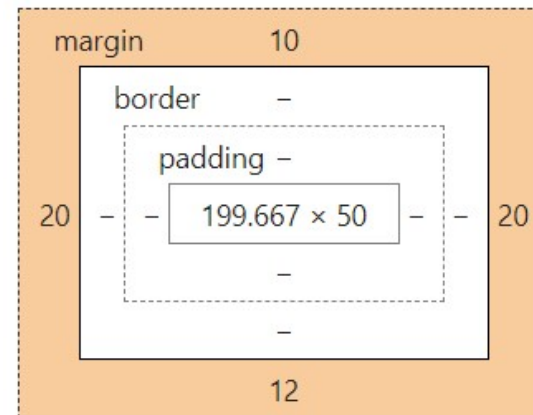
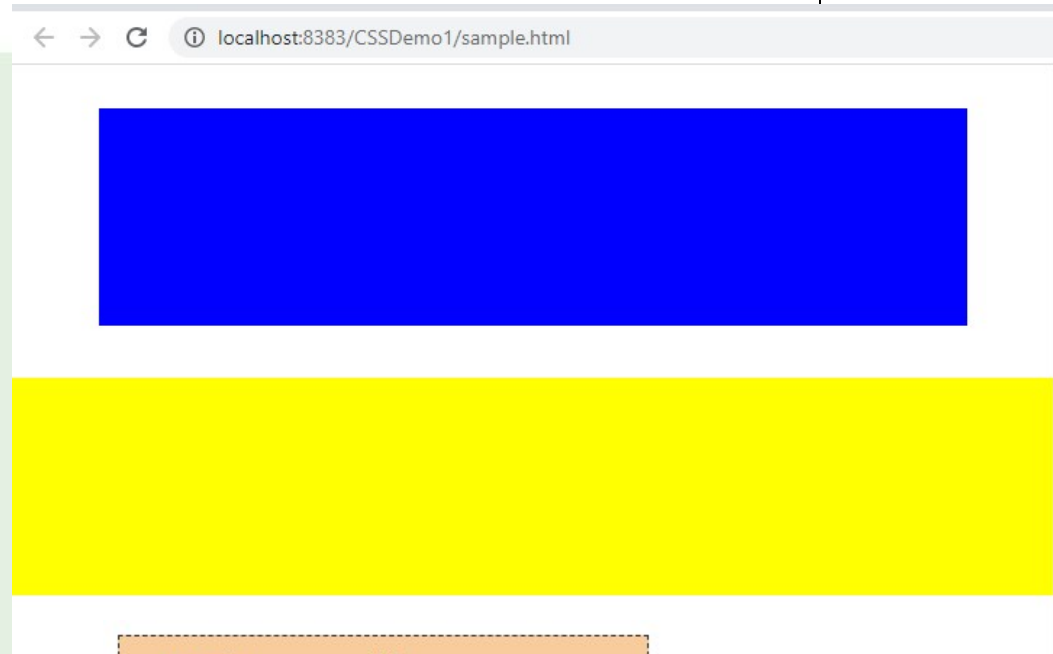
```
</head>
```

```
<body>
```

```
  <div class="box1"></div>
```

```
  <div class="box2"></div>
```

```
</body>
```



Top = 10px
Right = 20px
Bottom = 12px
Left = 20px

Setting Margin – Example – Four Values



```
<style>
  body {margin: 0px;}
  .box1 {
    height: 50px;
    background-color:blue;
    margin: 10px 20px 12px 8px;
  }
  .box2 {
    height: 50px;
    background-color:yellow;
  }
</style>
```

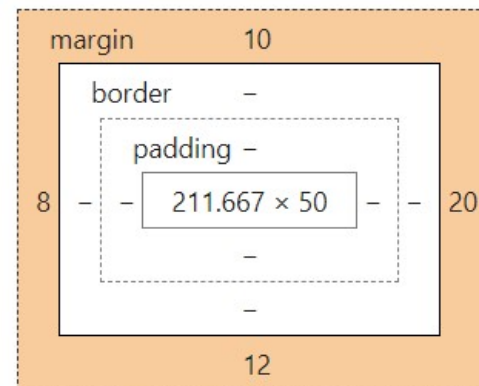
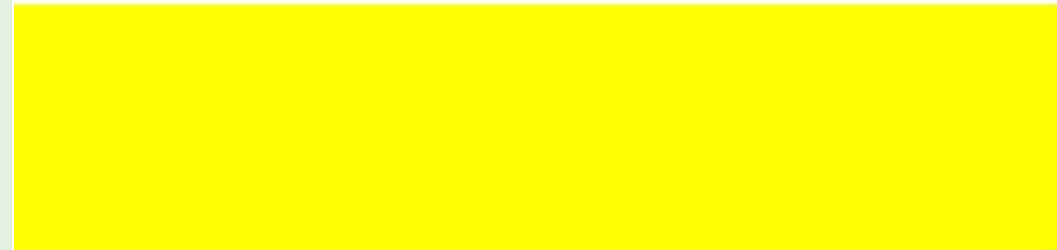
```
</head>
```

```
<body>
```

```
  <div class="box1"></div>
```

```
  <div class="box2"></div>
```

```
</body>
```



Top = 10px
Right = 20px
Bottom = 12px
Left = 8px

Setting padding properties



```
<style>
```

```
.b1 {  
    padding-left: 40px;  
    padding-top: 50px;  
    border: 2px solid red;  
    width: 500px;  
    height: 200px;  
}
```

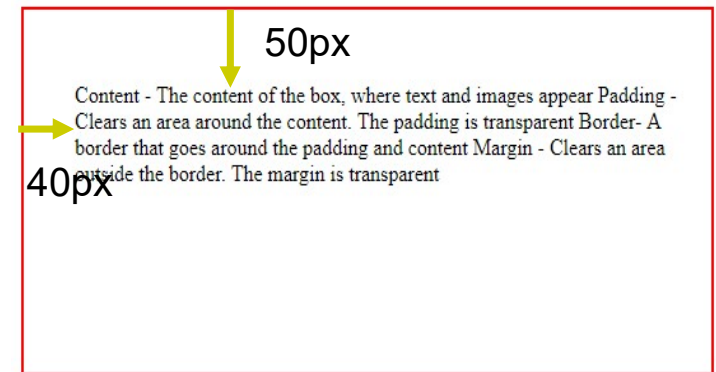
```
</style>
```

```
<div class="b1">
```

Content - The content of the box, where text and images appear
Padding - Clears an area around the content. The padding is transparent
Border- A border that goes around the padding and content
Margin - Clears an area outside the border. The margin is transparent

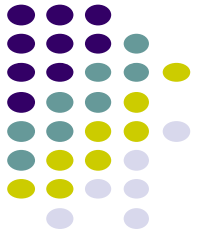
```
</div>
```

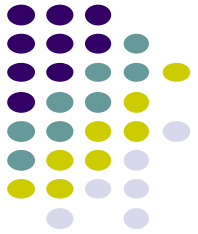
← → ↻ ⓘ localhost:8383/CSSDemo/index.html



Setting padding using **padding** property

- There are 4 ways you can set margin values(in pixels)
 - **Single value**
 - This single value will be applied to all padding such as top, right, bottom and left.
 - **Double values**
 - First value applied to top and bottom. Then second value applied to right and left.
 - **Three values**
 - First value to top, second to right, third to bottom and for left the second value will be applied
 - **Four values**
 - First value to top, second to right, third to bottom, and fourth to left





box-sizing property

- By default **padding** increases the **width** and **height** of the box.
- $\text{width} = \text{border} + \text{padding} + \text{width of the content}$
- $\text{height} = \text{border} + \text{padding} + \text{height of the content}$
- To prevent this default behavior, you can use **box-sizing** property to “**border-box**”

```
<style>
.b1 {
  padding-left: 40px;
  padding-top: 50px;
  border: 2px solid red;
  width: 500px;
  height: 200px;
  box-sizing: border-box;
}
</style>
```



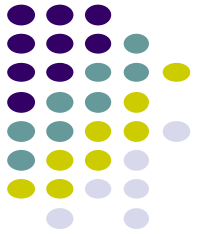
CSS **display** property



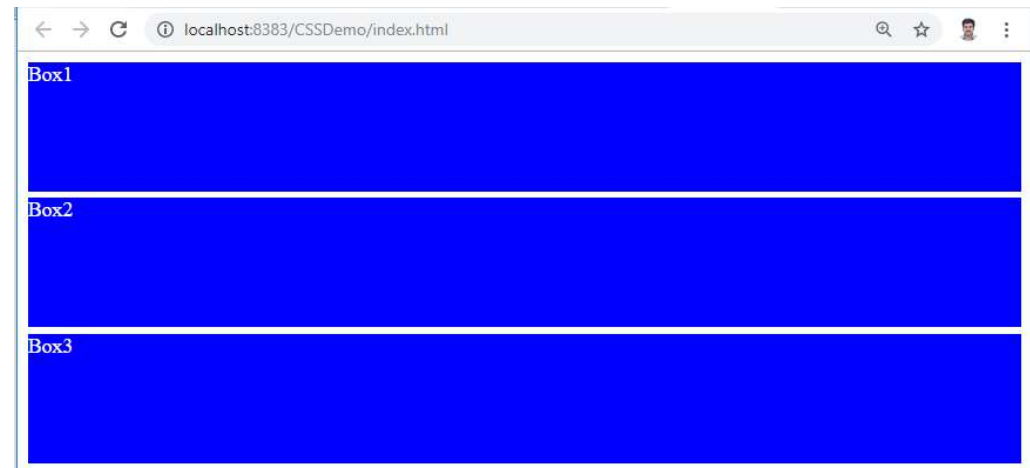
- The **display** property specifies the display behavior (the type of rendering box) of an element.
- values:
 - none (the element will not be displayed in web page)
 - inline
 - block
 - inline-block (similar to inline with block properties)

display: block

- every box laid as line by line.



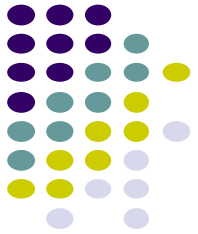
```
div{
  margin-bottom: 5px;
  height: 100px;
}
.d1 {
  display: block;
  background-color: blue;
  color:white;
}
.d2 {
  display: block;
  background-color: blue;
  color:white;
}
.d3 {
  display: block;
  background-color: blue;
  color:white;
}
```



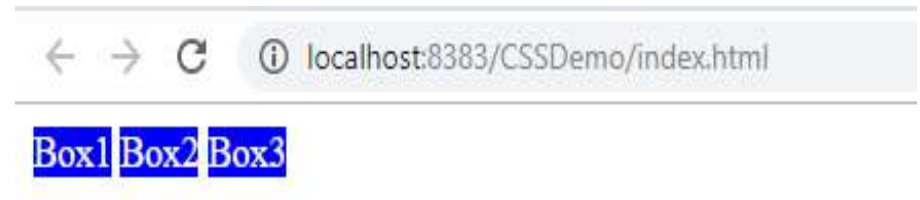
```
<body>
  <div class="d1">Box1</div>
  <div class="d2">Box2</div>
  <div class="d3">Box3</div>
</body>
```

display:inline

- elements placed in a line one after another depends its content.



```
<style>
  div{
    margin-bottom: 5px;
    height: 100px;
  }
  .d1 {
    display: inline;
    background-color: blue;
    color:white;
  }
  .d2 {
    display: inline;
    background-color: blue;
    color:white;
  }
  .d3 {
    display: inline;
    background-color: blue;
    color:white;
  }
</style>
```

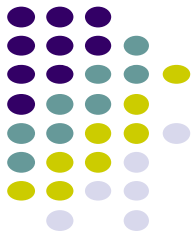


When display: inline, the box properties is not be applicable

```
<body>
  <div class="d1">Box1</div>
  <div class="d2">Box2</div>
  <div class="d3">Box3</div>
</body>
```

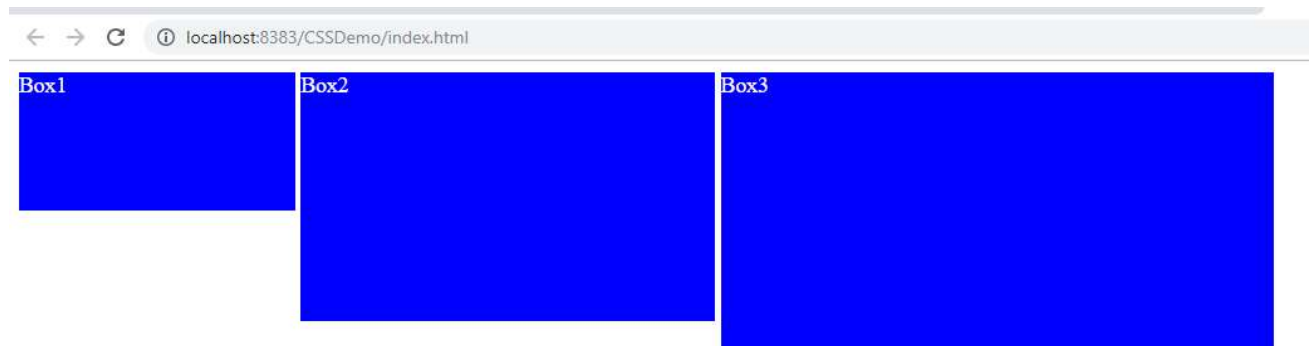

display: inline-block

- you can set box properties to every box.

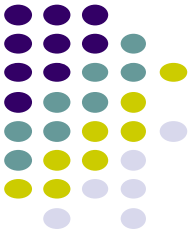


```
.d1 {  
  display: inline-block;  
  background-color: blue;  
  color: white;  
  width: 200px;  
  height: 100px;  
}  
.d2 {  
  display: inline-block;  
  background-color: blue;  
  color: white;  
  width: 300px;  
  height: 180px;  
}  
.d3 {  
  display: inline-block;  
  background-color: blue;  
  color: white;  
  width: 400px;  
  height: 200px;  
}
```

```
<div class="d1">Box1</div>  
<div class="d2">Box2</div>  
<div class="d3">Box3</div>
```



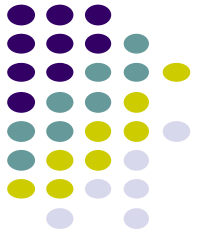
display:none



- Hiding an element can be done by setting the display **property** to **none**.
- The element will be hidden, and the page will be displayed as if the element is not there:

```
div {  
    display: none;  
}
```

opacity property



- The **opacity** property specifies the opacity/transparency of an element.

Transparent Image

The **opacity** property can take a value from 0.0 - 1.0. The lower value, the more transparent:



opacity 0.2



opacity 0.5



opacity 1
(default)

Activate V

opacity property example

```
.d1 {  
  width:400px;  
  height: 200px;  
  background-image: url(forest.jpg);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: cover;  
  color:white;  
  margin: 5px;  
  opacity: 1;  
}
```

opacity:1



```
.d2 {  
  width:400px;  
  height: 200px;  
  background-image: url(forest.jpg);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: cover;  
  color:white;  
  margin: 5px;  
  opacity: 0.2;  
}
```

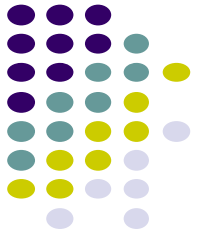
opacity:0.2



```
<body>  
  <div class="d1"> </div>  
  <div class="d2"> </div>  
</body>
```

hover effect

- Style an element when a user mouse over it.



```
<style>
```

```
.d1 {  
  width:400px;  
  height: 200px;  
  background-image: url(forest.jpg);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: cover;  
  color:white;  
  margin: 5px;  
  opacity: 0.2;  
}  
  
.d1:hover {  
  opacity: 1;  
}
```

```
</style>
```

```
<body>
```

```
  <div class="d1"> </div>
```

```
</body>
```

On mouse over it



Box-shadow

- CSS box-shadow Property
 - The CSS box-shadow property applies shadow to elements.



This is a yellow <div> element
with a black box-shadow

This is a yellow <div> element
with a grey box-shadow

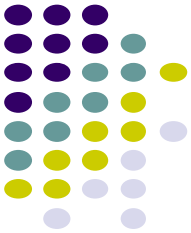
1

January 1, 2016



Hardanger, Norway

Box-shadow



```
<style>
```

```
  div {  
    background-color: yellow;  
    color: blue;  
    padding: 10px;  
    width: 200px;  
    height: 100px;  
    box-shadow: 10px 10px 2px red;  
  }
```

```
</style>
```

```
<body>
```

```
  <div>
```

```
    This is a box with box-shadow horizontal 5px,  
    vertical 5px, shadow radius 2px and shadow color red
```

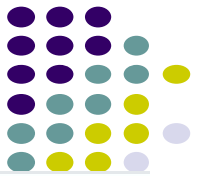
```
  </div>
```

```
</body>
```



This is a box with box-shadow
horizontal 10px, vertical 10px,
shadow radius 2px and shadow
color red

Box-shadow on hover effect



```
<style>
  div {
    background-color: yellow;
    color: blue;
    padding: 10px;
    width: 200px;
    height: 100px;
  }
  div:hover {
    box-shadow: 10px 10px 2px red;
  }
</style>
```

```
<body>
  <div>
    This is a box, on "hover" you will get box-shadow horizontal 10px,
    vertical 10px, shadow radius 2px and shadow color red
  </div>
</body>
```

