



CSS

Introduction, Using CSS, Selectors,
Margin, Padding, Colors



Table of Contents



- ▶ What is CSS?
- ▶ History of CSS
- ▶ CSS Syntax

What is CSS?

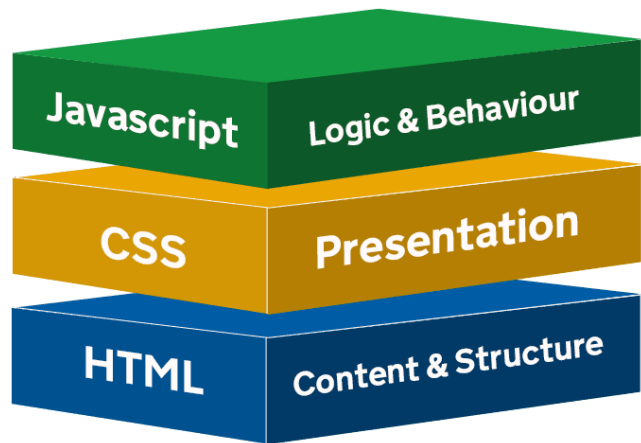


- ▶ CSS stands for Cascading Style Sheets
- ▶ It is a **markup language** that determines the style properties of HTML elements.
- ▶ Thanks to CSS, the properties of many elements can be changed **at the same** time with a single definition. While defining the skeleton and content of the page with HTML, it is **formatted with CSS**.

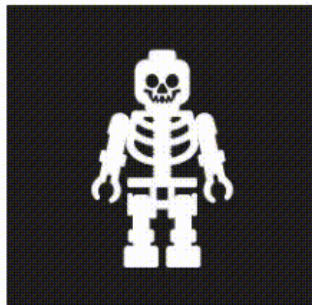
Cascading Style Sheets

What is CSS?

- ▶ CSS is presentation layer of a webpage while html is structure layer.



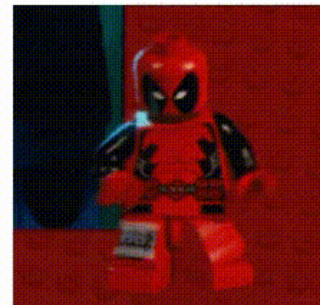
HTML
structure



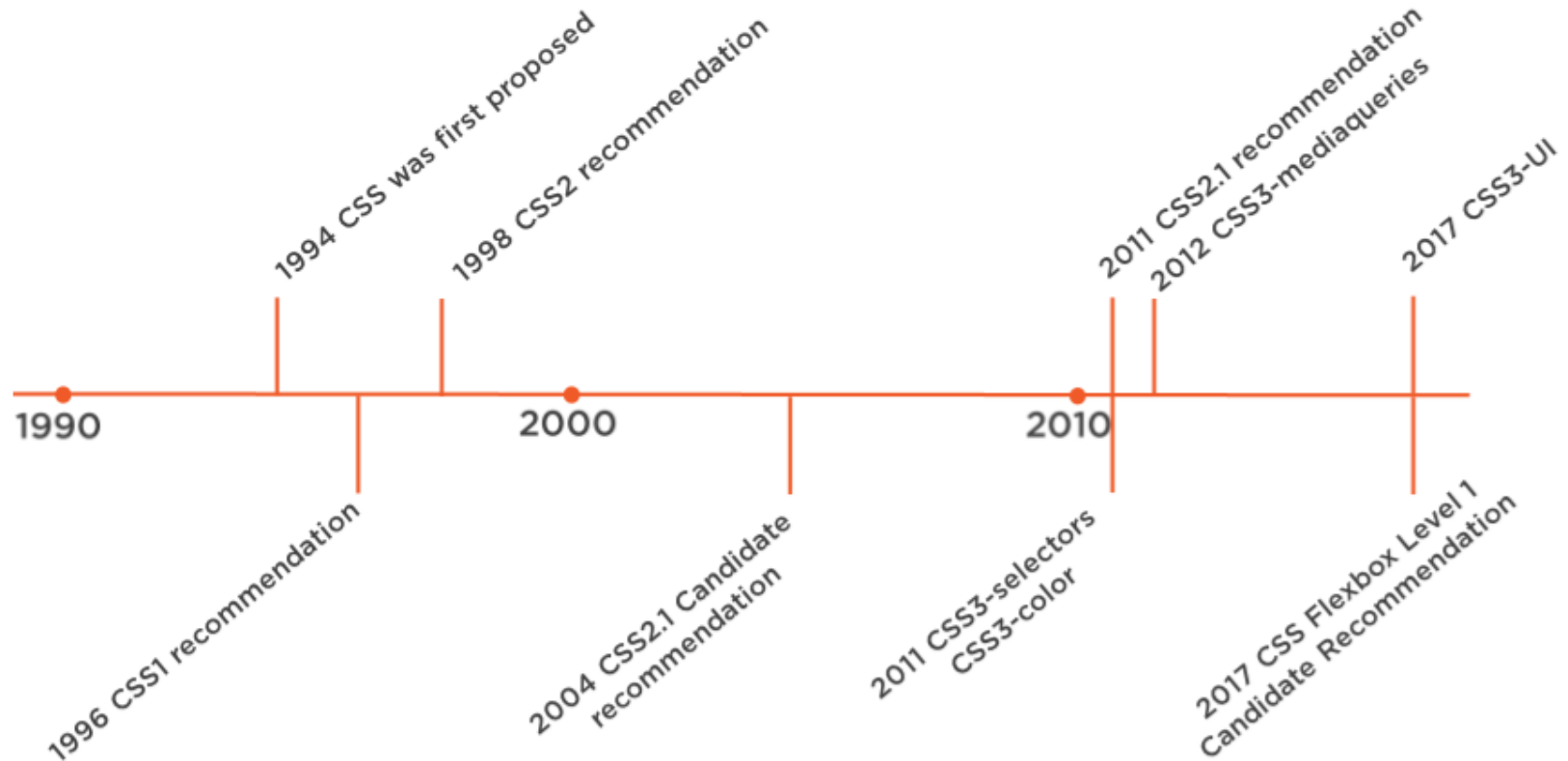
CSS
presentation/appearance



JavaScript
dynamism/action



History of CSS?



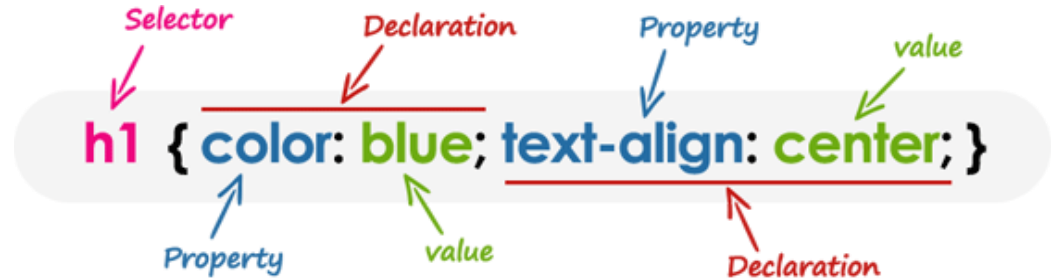


CSS Syntax



- ▶ In CSS, a declaration is the **key-value** pair of a CSS property and its value.
- ▶ Each declaration includes a CSS property name and a value, separated by a **colon**.

```
h1{  
  color:blue;  
  font-size: 12px;  
}
```



CSS Properties



- ▶ CSS supports more than 200 CSS properties. Here's a complete list.

<https://www.hostinger.com/tutorials/css-cheat-sheet>

```
<p style="color: red">Hello</p>
```

```
<h1 style='background-color: blue'>CSS</h1>
```

Use of CSS



- ▶ Inline CSS
- ▶ Internal CSS
- ▶ External CSS
- ▶ Multiple Styles and Cascading Order



Inline CSS



- ▶ An inline style may be used to apply a **unique style** for a **single element**.
- ▶ To use inline styles, add the **<style>** attribute to the relevant element.

`<p style="font-size:20px">inline</p>`

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <p style="font-size: 20px">This is an inline styling</p>
  </body>
</html>
```



Internal CSS



- ▶ An internal style sheet may be used if one single HTML page has a unique style.
- ▶ The internal style is defined inside the `<style>` element, inside the **head section**.

```
<head>
  <style>
    p{
      color:blue;
    }
  </style>
</head>
```

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        color:blue;
      }
    </style>
  </head>
  <body>
    <p>This is an internal styling</p>
  </body>
</html>
```

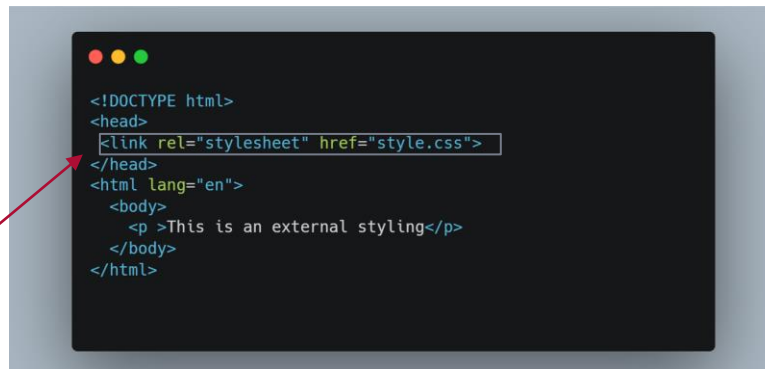
External CSS



- ▶ With an external style sheet, you can change the look of an entire website by changing just one file!
- ▶ Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.



```
<head>
  <link rel="stylesheet" href="style.css">
</head>
```



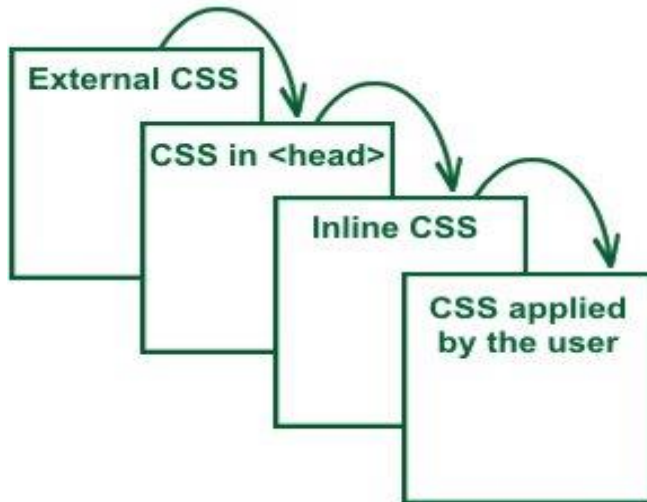


External CSS



1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.



Selectors in Css



Element Selector

```
h2{  
  color:blue;  
}
```

Universal Selector "*"

```
* {  
  color:blue;  
}
```

ID Selector "#"

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

Class Selector "."

```
.main {  
  color:blue;  
}
```



Element Selector



- ▶ An element selector uses element **name** to select the **HTML element**. A markup language defines the structure and presentation of raw text.

```
<!DOCTYPE html>

<html lang="en">
  <body>
    <h1>This is element selector</h1>
  </body>
</html>
```

```
h1{
  font-size:25px;
  color:red;
}
```





Universal Selector

- ▶ The universal selector (*) selects all HTML elements on the page.



```
*{  
margin:0;  
padding:0;  
font-family: 'Courier New', Courier, monospace;  
}
```



Id Selector



The **id selector** uses **id attribute** to select the element in HTML file.

The id of an element must be unique in a page. A **hash (#)** character is used to select an element with a specific id.

```
<!DOCTYPE html>

<html lang="en">
  <body>
    <p id="content">This is Id selector</p>
  </body>
</html>
```

```
#content{
  color:red;
}
```


Class Selector



The **class selector** uses class attribute to select all elements of that class in HTML file.

A **period (.)** character is used to select elements with a specific class.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<body>
```

```
<p class="main">This is class selector</p>
```

```
</body>
```

```
</html>
```

```
.main{
```

```
background-color:green;
```

```
}
```

CSS Comments

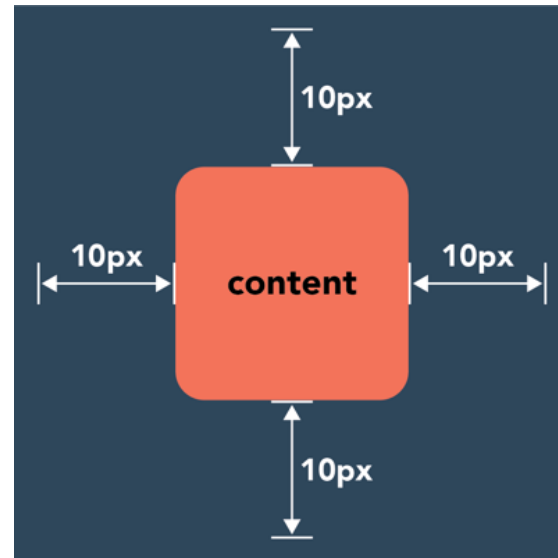


- ▶ The browser does not display comments, but with comments, we can place warnings, descriptions and other notes.
- ▶ Comments are also imported for debugging CSS. We can use comments for searching errors.
- ▶ CSS Comments can add the text between `/*` and `*/` characters.





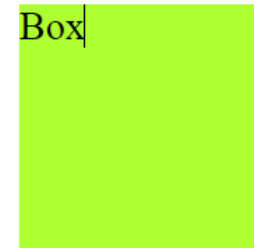
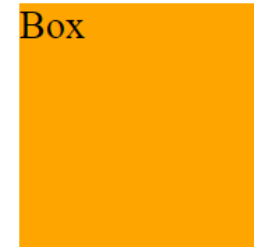
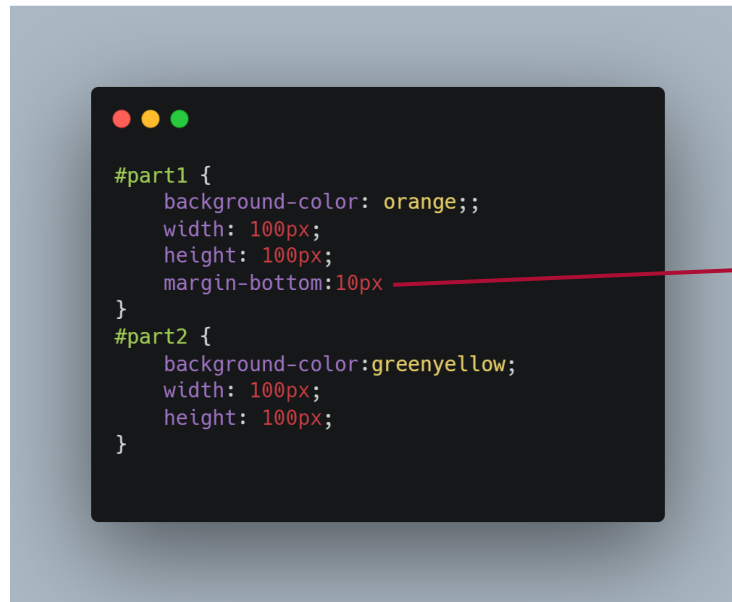
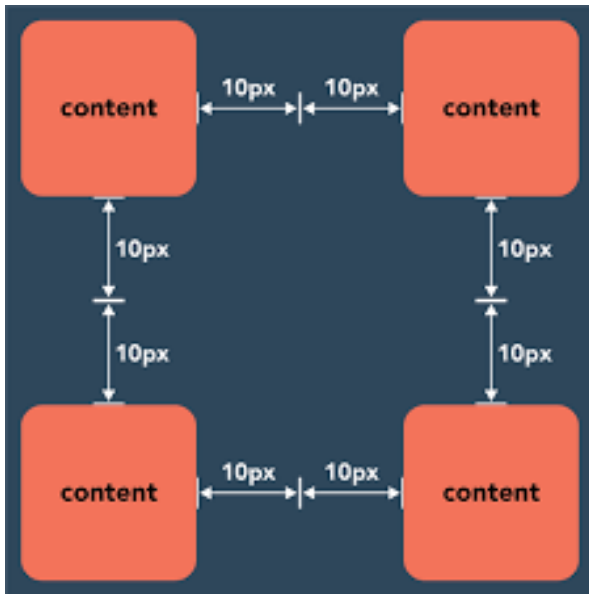
CSS Margin





Margin

- ▶ Margins are used to create **space around elements**, outside of any defined borders.



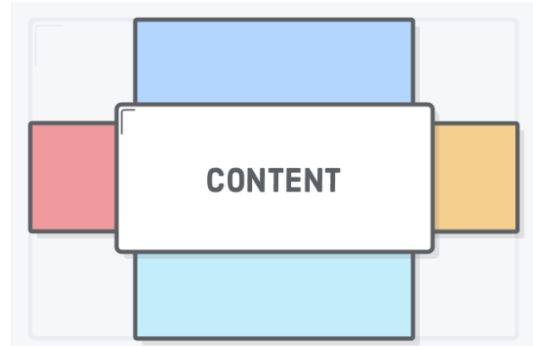


Margin



There are 4 different properties for setting the margin for each side of an element:

- ▶ margin-top
- ▶ margin-right
- ▶ margin-bottom
- ▶ margin-left



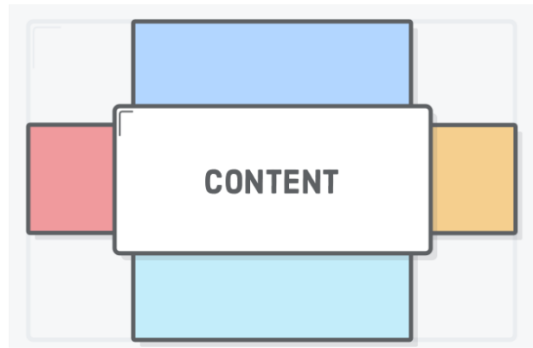
MARGIN : TOP RIGHT BOTTOM LEFT



4 margin Values

If 4 margin values are used, the order is top,right,bottom,left

- ▶ margin-top
- ▶ margin-right
- ▶ margin-bottom
- ▶ margin-left



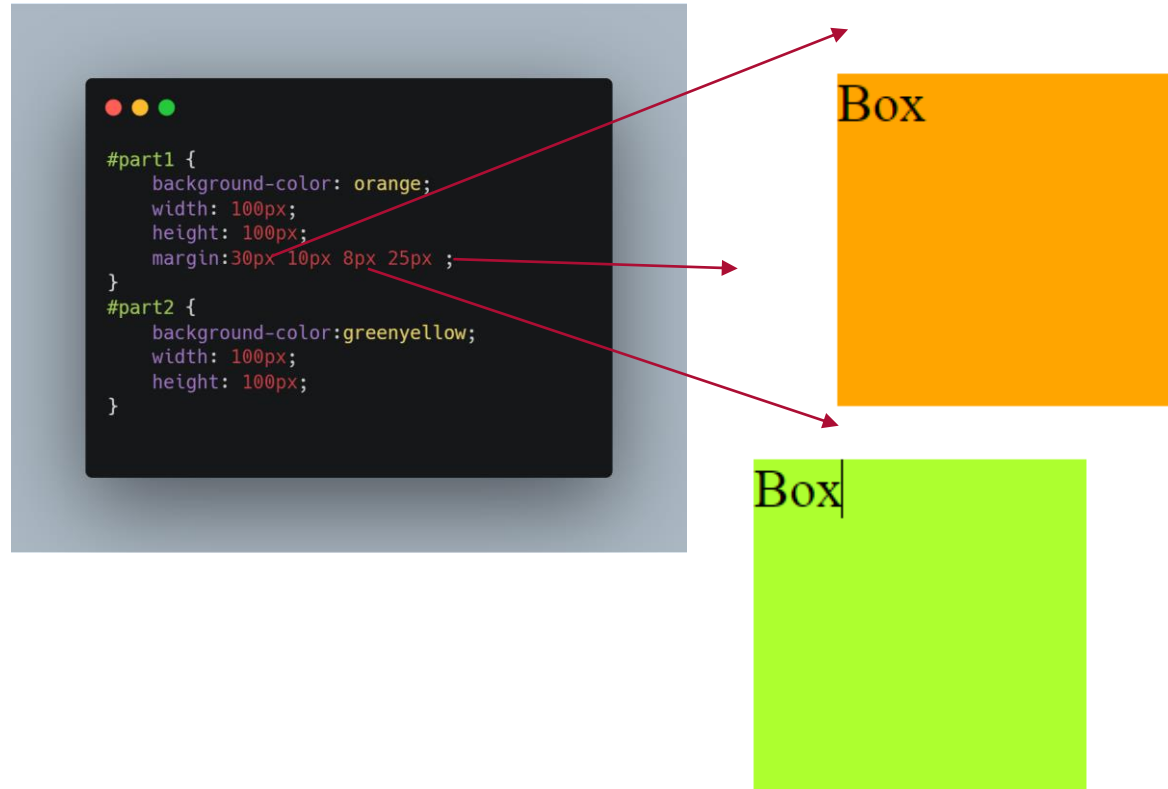
MARGIN : TOP RIGHT BOTTOM LEFT



4 margin Values

If 4 margin values are used, the order is top, right, bottom, left

- ▶ margin-top
- ▶ margin-right
- ▶ margin-bottom
- ▶ margin-left

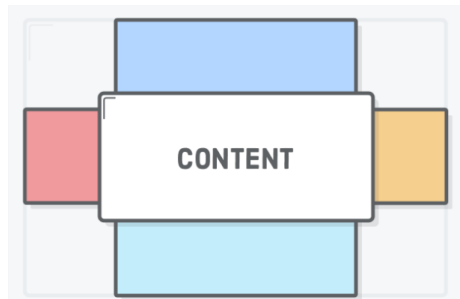




3 margin Values

If 3 margin values are used, the order is top,right-left,bottom

- ▶ margin-top
- ▶ margin-right-left
- ▶ margin-bottom



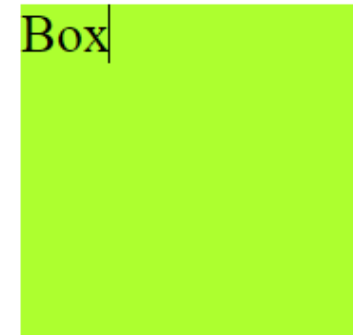
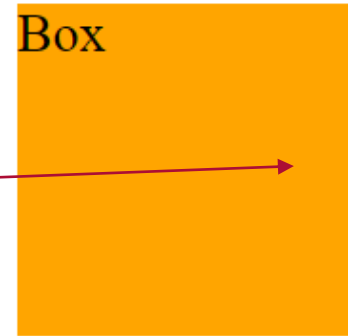
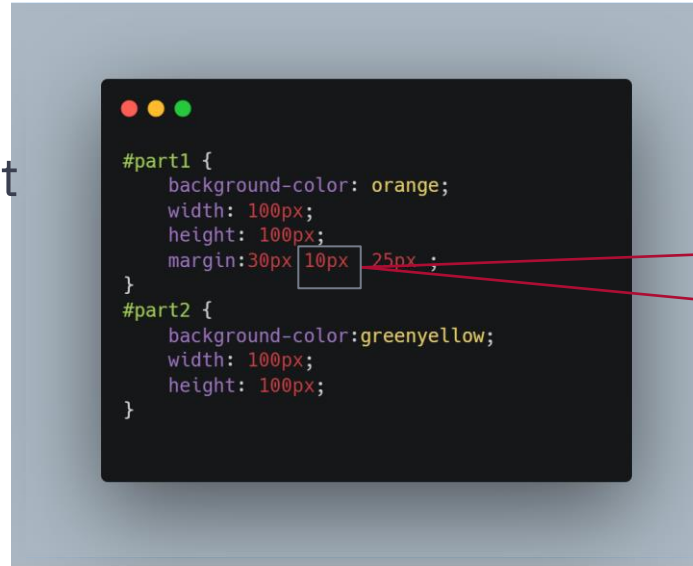
MARGIN : TOP RIGHT LEFT BOTTOM



3 margin Values

If 3 margin values are used, the order is top, right, bottom, left

- ▶ margin-top
- ▶ margin-right-left
- ▶ margin-bottom

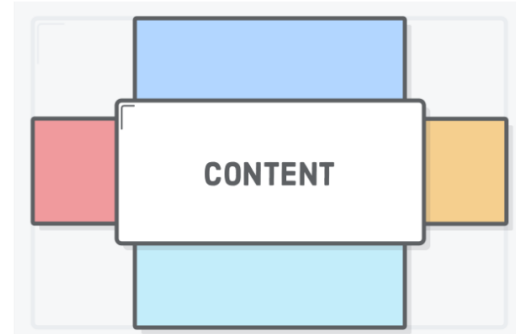


2 margin Values



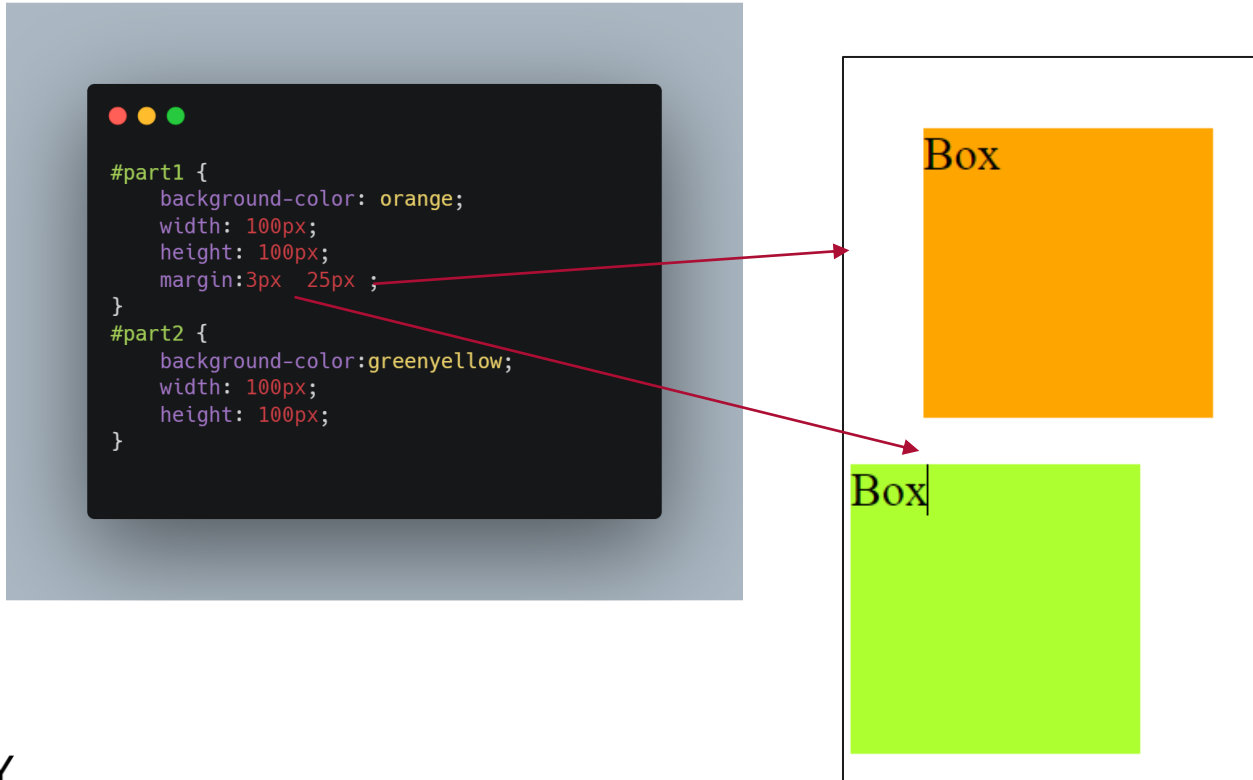
If 2 margin values are used, the order is top-bottom, right-left,

- ▶ margin-top-bottom
- ▶ margin-right-left



2 margin Values

If 2 margin values are used, the order is top-bottom, right-left

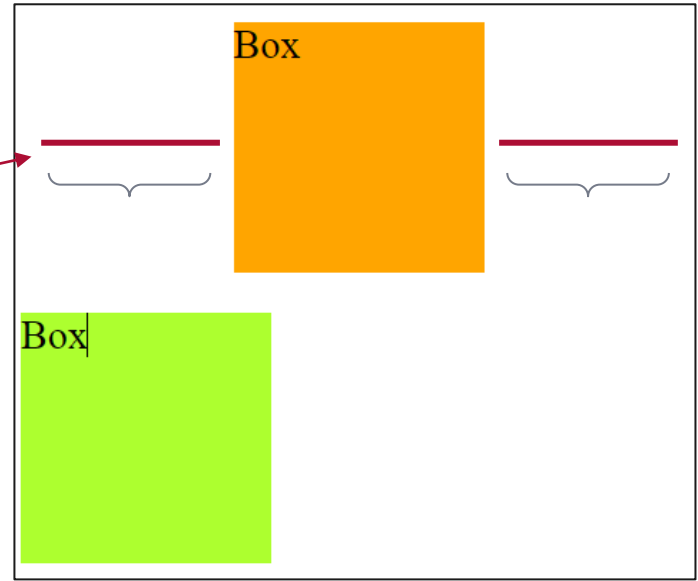


Auto Value



- ▶ The margin property can be set to **auto**. The **browser will calculate** the margin and horizontally center the element.

```
#part1 {  
  background-color: orange;;  
  width: 100px;  
  height: 100px;  
  margin:auto;  
}  
#part2 {  
  background-color:greenyellow;  
  width: 100px;  
  height: 100px;  
}
```





CSS Padding

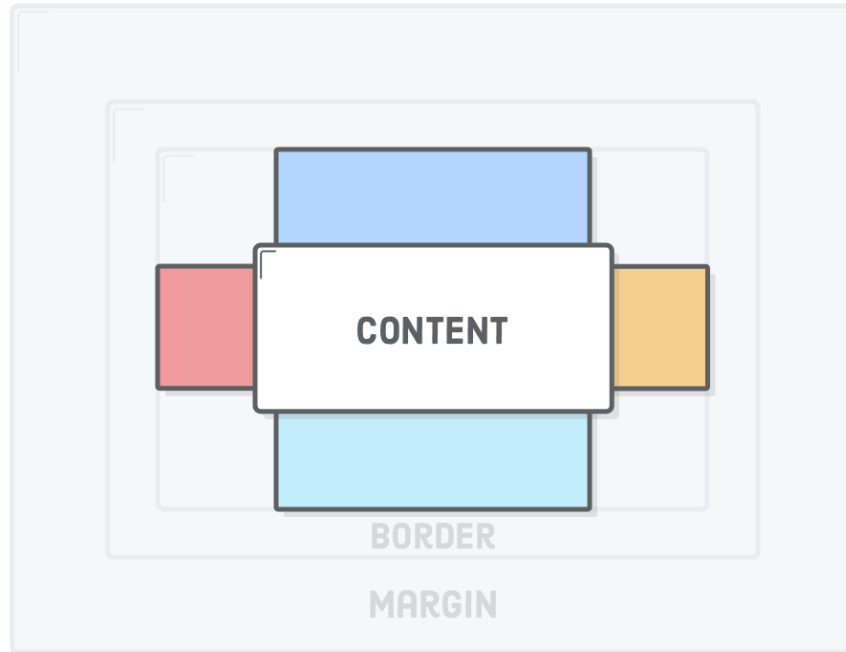




Padding



- ▶ The padding property is used to define space between the **element's content** and **the borders**.



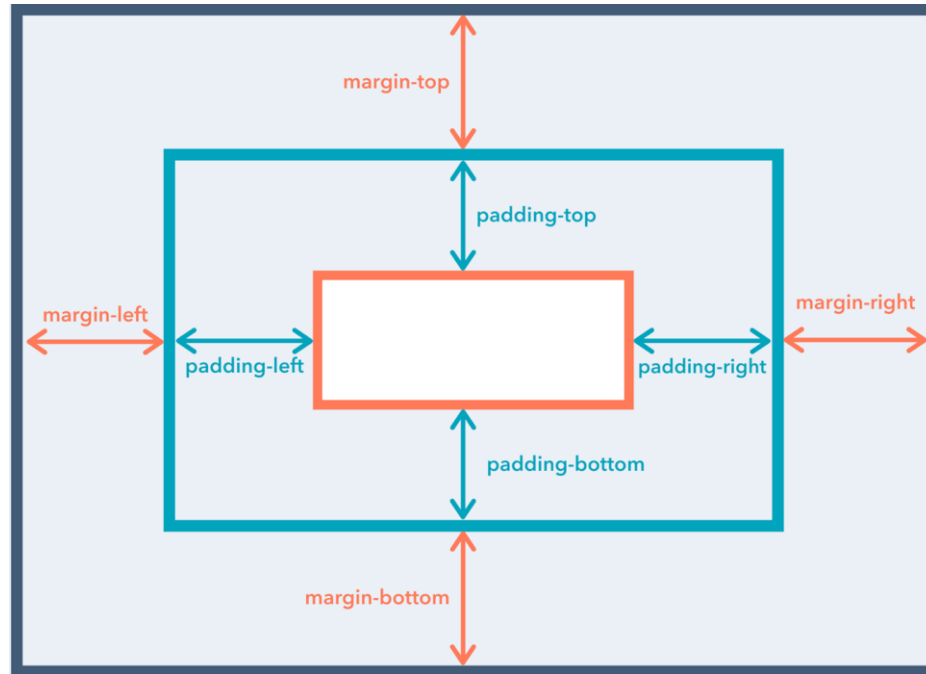


Padding



There are 4 different properties for setting the padding for each side of an element:

- ▶ padding-top
- ▶ padding-right
- ▶ padding-bottom
- ▶ padding-left

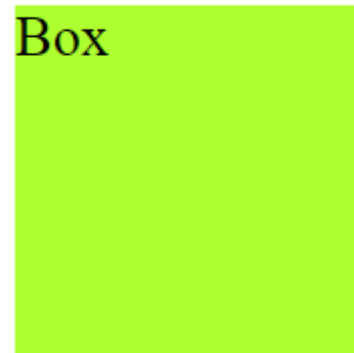
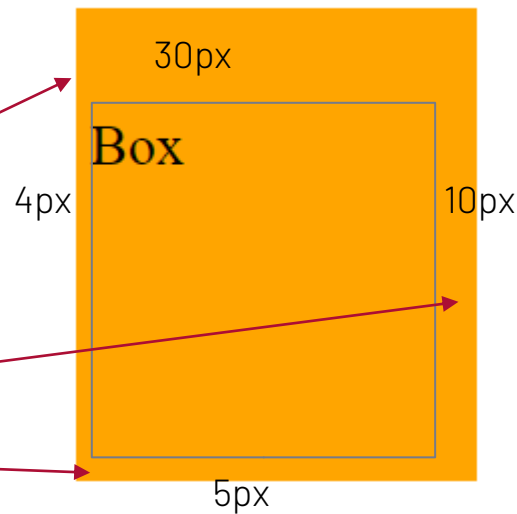




Padding



```
#part1 {  
  background-color: orange;;  
  width: 100px;  
  height: 100px;  
  padding: 30px 10px 5px 4px;  
}  
#part2 {  
  background-color: greenyellow;  
  width: 100px;  
  height: 100px;  
}
```





Shorthand Property

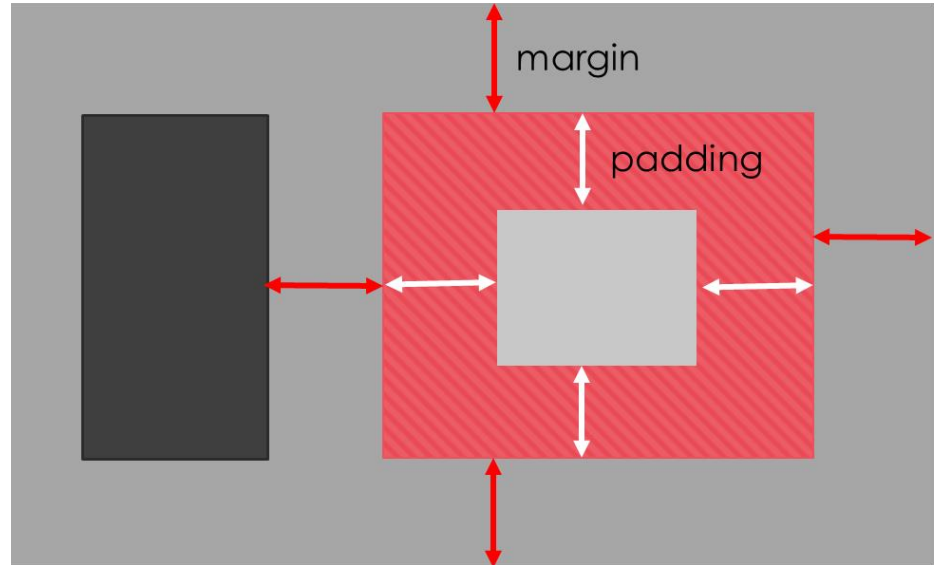


4 Padding Values



If 4 padding values are used, the order is top,right,bottom,left

- ▶ padding -top
- ▶ padding -right
- ▶ padding -bottom
- ▶ padding -left

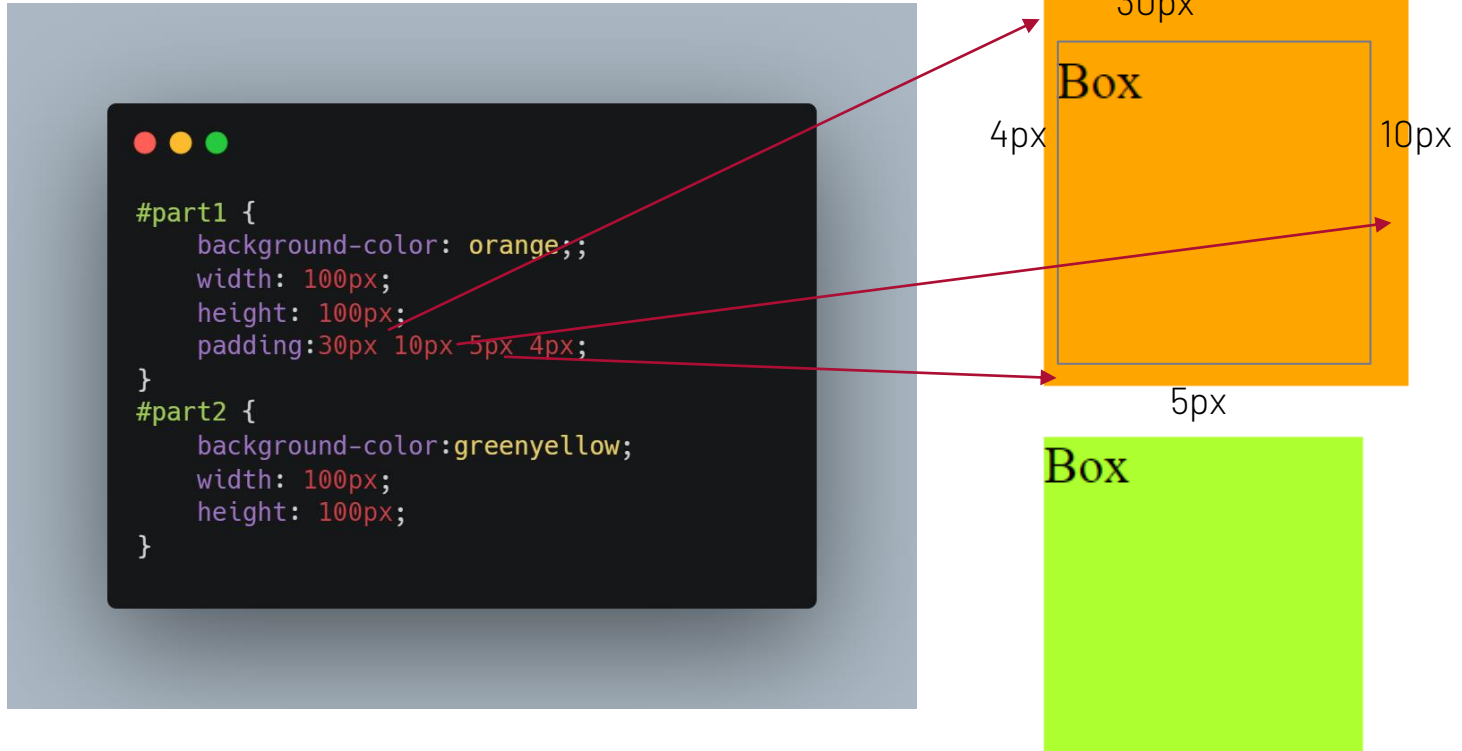




4 Padding Values



If 4 padding values are used, the order is top,right,bottom,left





3 Padding Values



If 3 padding values are used, the order is top,right-left,bottom

- ▶ padding-top
- ▶ padding-right-left
- ▶ padding-bottom





2 Padding Values



If 2 padding values are used, the order is top-bottom, right-left,

- padding -top-bottom
- padding -right-left





CSS Colors














CSS Colors



Most used color values

- ▶ Pre-defined Colors
- ▶ RGB, RGBA
- ▶ HEX
- ▶ HSL, HSLA

| Color | CSS Color Name | Hex Code #RPGGBB | Decimal Code (R,G,B) |
|---|----------------|------------------|----------------------|
|  | Red | #FF0000 | rgb (255,0,0) |
|  | Orange | #FFA500 | rgb (255,165,0) |
|  | Yellow | #FFFF00 | rgb (255,225,0) |
|  | Green | #008000 | rgb (0,128,0) |
|  | Cyan | #00FFFF | rgb (0,255,225) |
|  | Blue | #0000FF | rgb (0,0,225) |
|  | Purple | #800080 | rgb (128,0,128) |
|  | Pink | #FFC0CB | rgb (255,192,203) |
|  | Gray | #808080 | rgb (128,128,128) |
|  | Brown | #A52A2A | rgb (165,42,42) |





Pre-defined Colors



- ▶ A color can be defined by using a default color name
- ▶ CSS/HTML supports **140** standard color names.

```
<!DOCTYPE html>

<html lang="en">
  <body>
    <p id="part1">Blue</p>
    <p id="part2">Pink</p>
    <p id="part3">Yellow</p>
    <p id="part4">Orange</p>
    <p id="part5">Green</p>
  </body>
</html>

}
```

```
#part1{background-color: blue;}
#part2{background-color: pink;}
#part3{background-color: yellow;}
#part4{background-color: orange;}
#part5{background-color: green;}
```

Blue

Pink

Yellow

Orange

Green



<https://www.w3.org/wiki/CSS/Properties/color/keywords>



RGB Values



- ▶ A color can be defined with a **RGB value**.
- ▶ Each red, green, and blue parameter defines the intensity of the color between **0 and 255**.

rgb( 250 ,  54 ,  80)
Red Green Blue

rgb(0,0,0)

rgb(239, 0, 51)

rgb(171, 51, 255)

rgb(53,222,16)



RGBA Values



- ▶ RGBA color values are the RGB color values with an alpha channel that defines the **opacity** of a color.

`rgb(`  `,`  `,`  `,` `0.5)`
Red Green Blue Opacity

`rgb(0,0,0, 0.4)`

`rgb(239, 0, 51, 1)`

`rgb(171, 51, 255, 0.3)`

`rgb(53,222,16, 0.8)`



Hex Values



- ▶ A color can be defined using a hexadecimal value
- ▶ The **hexadecimal values** must be between **00** and **FF**.

#

EF

Red

54

Green

A5

Blue

#000000

#ef0033

#ab33ff

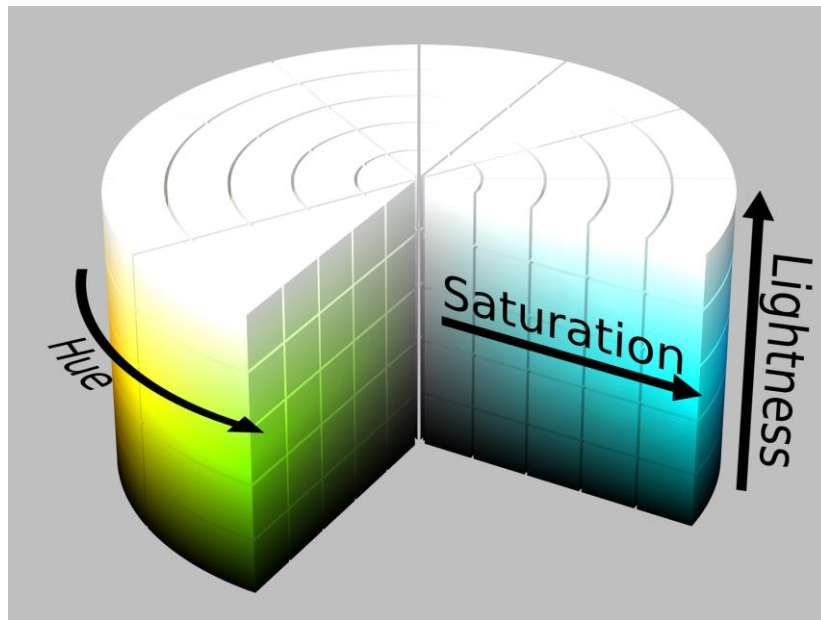
#35de10



HSL, HSLA



- ▶ Hue, Saturation, Lightness (HSL)
- ▶ Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, 240 is blue.
- ▶ Saturation is a percentage value; 0% means a shade of gray and 100% is the full color.
- ▶ Lightness is also a percentage; 0% is black, 100% is white.
- ▶ HSLA(hue, saturation, lightness, alpha)





HSL, HSLA



hsl(120, 100%, 0%)

#00FF00

Hue

Saturation

Lightness

hsl(0 , 100%, 50%)

rgb(255, 0, 0)

THANKS!