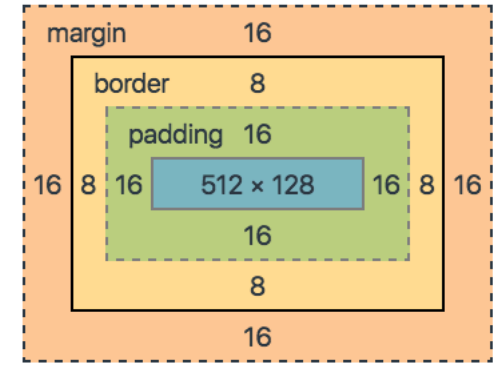
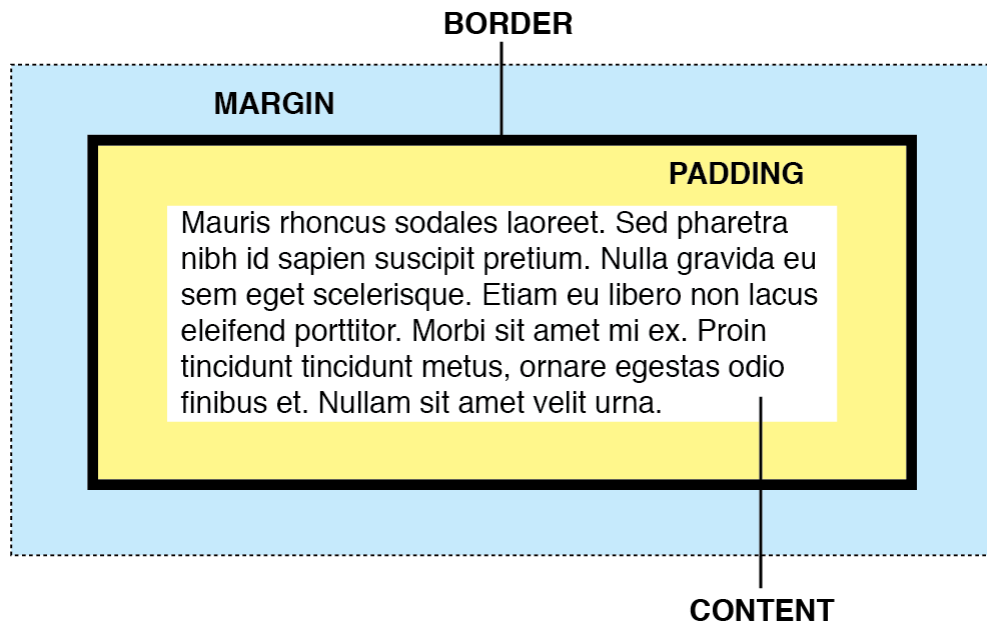


UI Basics

Box Model

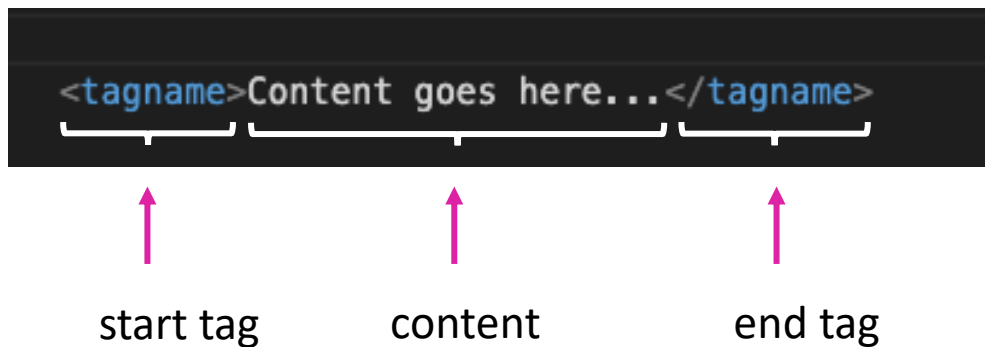
- CSS Padding
- CSS Border & Border-Radius
- CSS Margin
- CSS Box-sizing
- CSS Height & Width
- CSS Display
- CSS Visibility
- CSS Overflow
- CSS Float
- CSS Clear

BOX MODEL

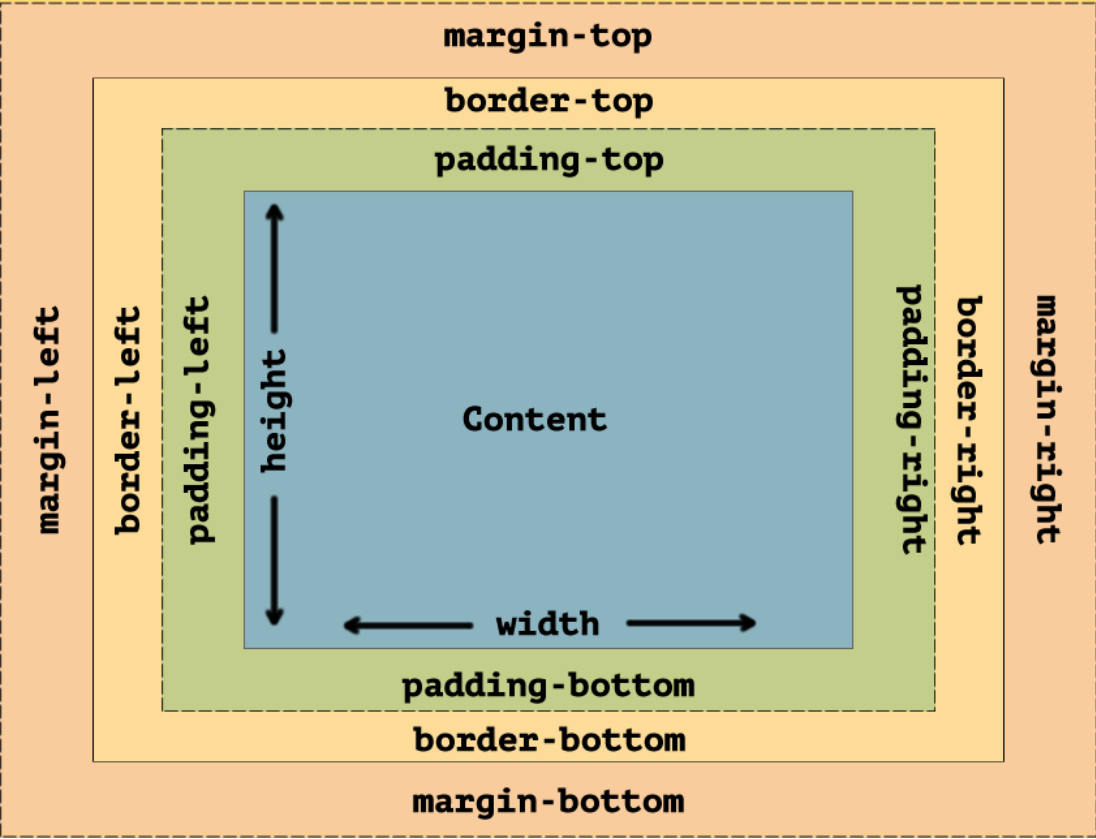
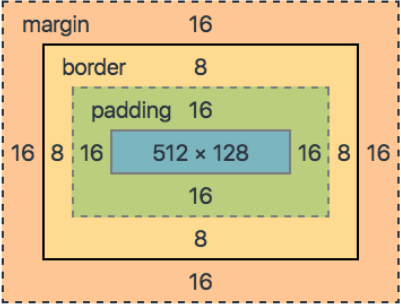
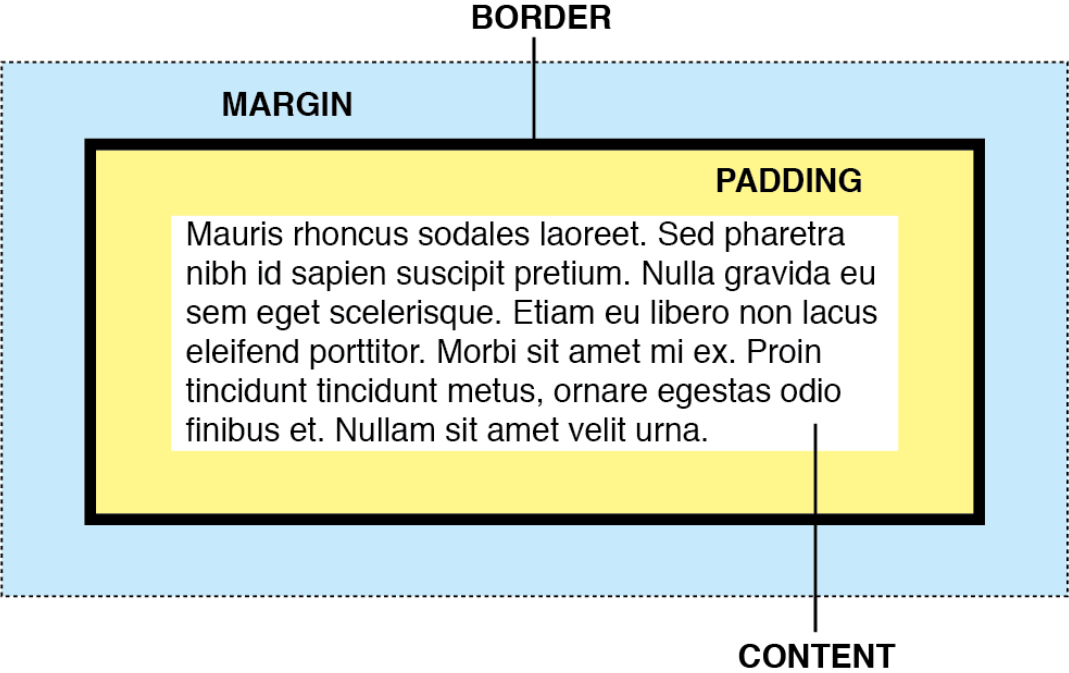


DevTools

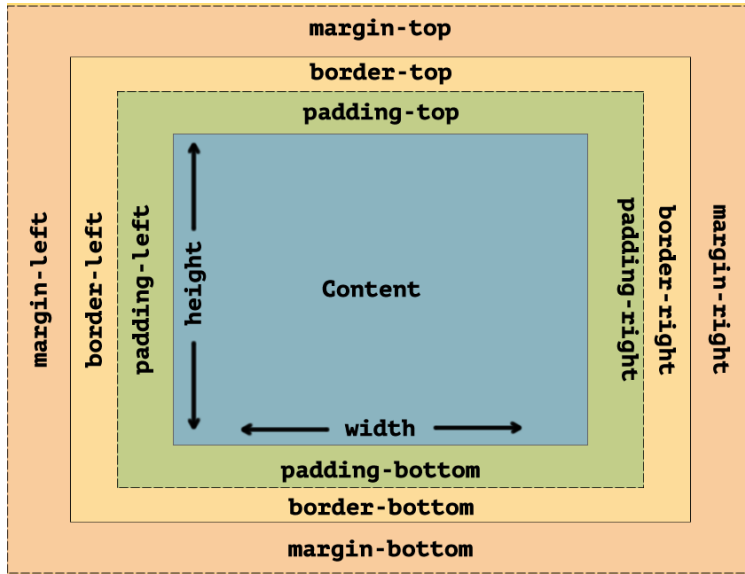
- The CSS **padding** properties are used to generate space around an element's content, inside of any defined borders.
- The CSS **margin** properties are used to create space around elements, outside of any defined borders.



BOX MODEL



CSS Padding



`padding-top: 50px;`

`padding-right: 30px;`

`padding-bottom: 50px;`

`padding-left: 80px;`

`padding: 100px 150px 100px 80px; /* padding top - right - bottom - left */`

`padding: 100px 150px 70px; /* padding top - right and left - bottom */`

`padding: 100px 150px; /* padding top and bottom - right and left */`

`padding: 100px; /* padding for all sides */`

length : specifies a padding in **px**, cm, etc.

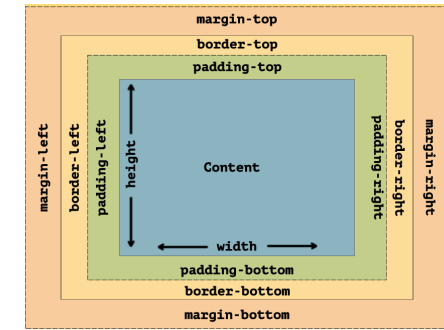
% : specifies a padding in % of the width of the containing element

inherit : specifies that the padding should be inherited from the parent element

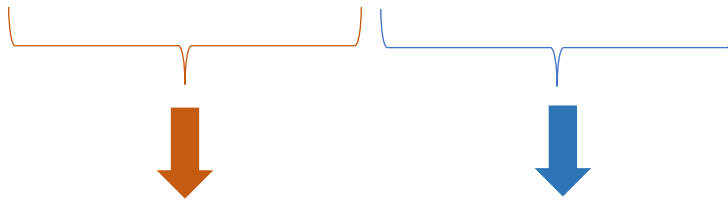
NOTE: Negative values are not allowed.

CSS Border

The CSS **border** properties allow you to specify the style, width, and color of an element's border.



`border: border-width border-style border-color;`

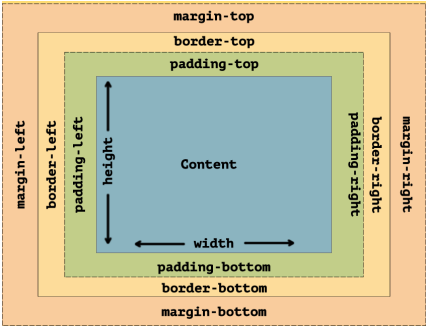


specific size (px, cm, em, etc)
thin
medium
thick

dotted	- Defines a dotted border
dashed	- Defines a dashed border
solid	- Defines a solid border
double	- Defines a double border
groove	- Defines a 3D grooved border.
ridge	- Defines a 3D ridged border.
inset	- Defines a 3D inset border.
outset	- Defines a 3D outset border.
none	- Defines no border
hidden	- Defines a hidden border

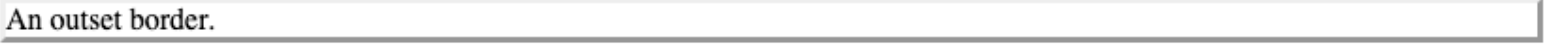
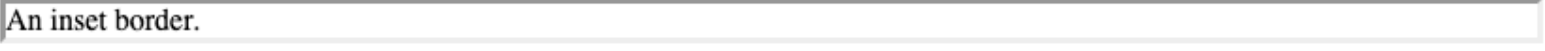
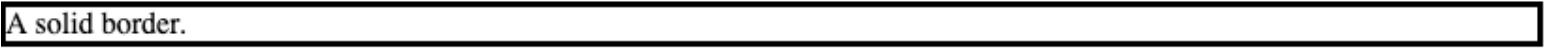
NOTE: border-style - required

CSS Border



border-style

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```



No border.

A hidden border.



CSS Border - radius

The **border-radius** property is used to add **rounded borders** to an element

```
p.normal {  
  border: 2px solid red;  
}
```

```
p.round1 {  
  border: 2px solid red;  
  border-radius: 5px;  
}
```

```
p.round2 {  
  border: 2px solid red;  
  border-radius: 8px;  
}
```

```
p.round3 {  
  border: 2px solid red;  
  border-radius: 12px;  
}
```

```
<p class="normal">Normal border</p>  
<p class="round1">Round border</p>  
<p class="round2">Rounder border</p>  
<p class="round3">Roundest border</p>
```

Normal border

Round border

Rounder border

Roundest border

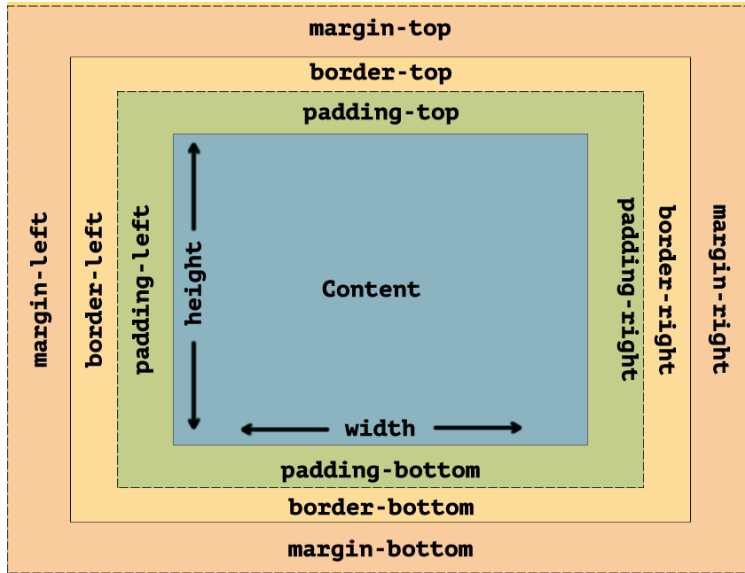
CSS Border - radius

```
.radius {  
  border: 2px solid red;  
  border-radius: 50px; /* all corners */  
  border-radius: 50px 10px; /* top left - bottom right / top right - bottom left*/  
  border-radius: 50px 0 10px; /* top left / top right - bottom left / bottom right*/  
  border-radius: 50px 0 10px 30px; /* top left / top right / bottom right / bottom left*/  
}
```

```
<p class="radius">A mix border with differet side properties.</p>
```

A mix border with differet side properties.

CSS Margin



margin-top: 50px;

margin-right: 30px;

margin-bottom: 50px;

margin-left: 80px;

margin: 100px 150px 100px 80px; /* margin top - right - bottom - left */

margin: 100px 150px 70px; /* margin top - right and left - bottom */

margin: 100px 150px; /* margin top and bottom - right and left */

margin: 100px; /* margin for all sides */

auto : the browser calculates the margin

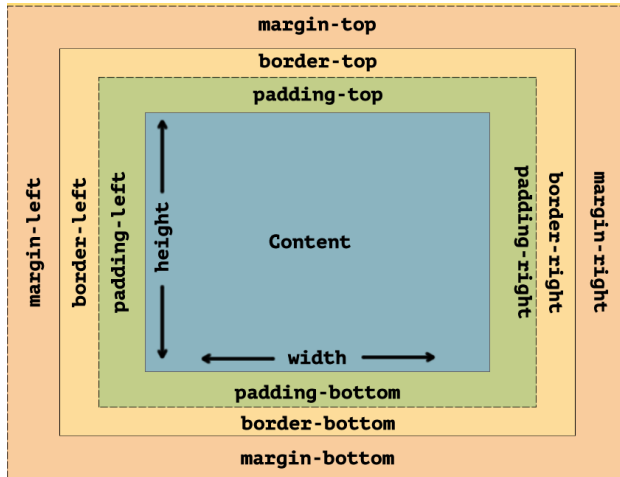
length : specifies a margin in **px**, cm, etc.

% : specifies a margin in % of the width of the containing element

inherit : specifies that the margin should be inherited from the parent element

NOTE: Negative values are not allowed.

CSS Margin - auto



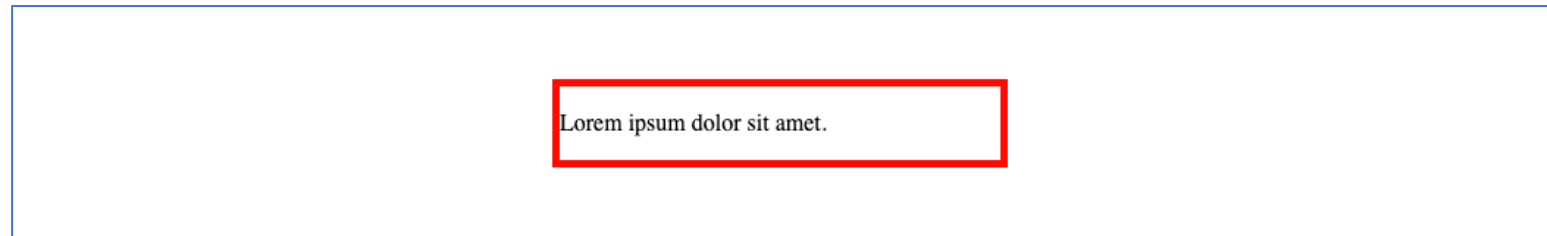
```
.box2 {  
  border: 5px solid red;  
  width: 300px;  
  margin: 50px auto;  
}
```

```
<div class="box2">  
  <p>Lorem ipsum dolor sit amet.</p>  
</div>
```

margin: 50px auto;

margin top and bottom : 50px
margin left and right : auto

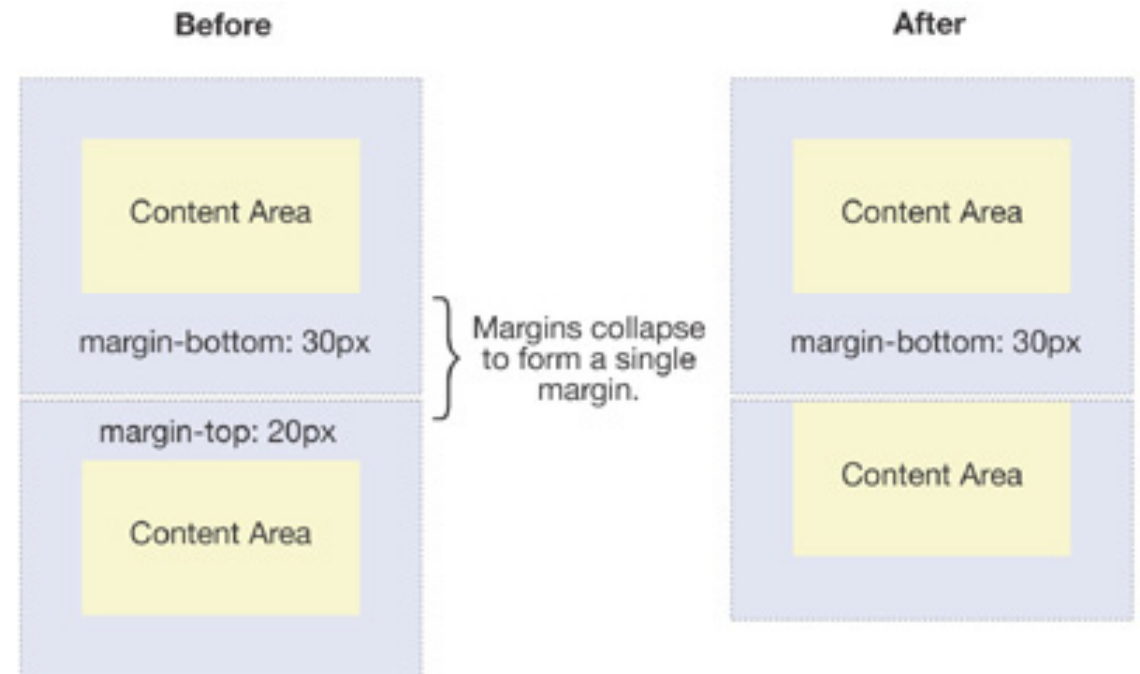
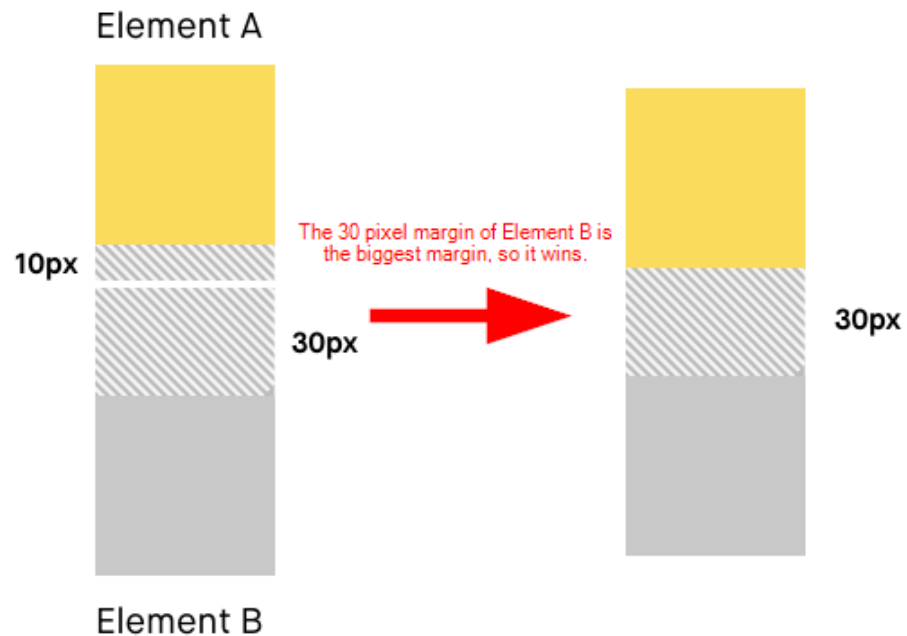
width: 300px;



NOTE: without **width** property , “margin – auto” rule does not work.

CSS Margin - Collapse

- **Top and bottom margins of elements** are sometimes collapsed **into a single margin** that is equal to the largest of the two margins.
- This does not happen on left and right margins! Only top and bottom margins!



BOX MODEL

CSS Box-sizing

```
selector {  
  box-sizing: content-box / border-box;  
}
```



„content-box“:
Width: 230px
Padding: 15px
Border: 5px

Width + 2x Padding + 2x Border



„border-box“:
Width: 230px
Padding: 15px
Border: 5px

Width - 2x Padding - 2x Border

BOX MODEL

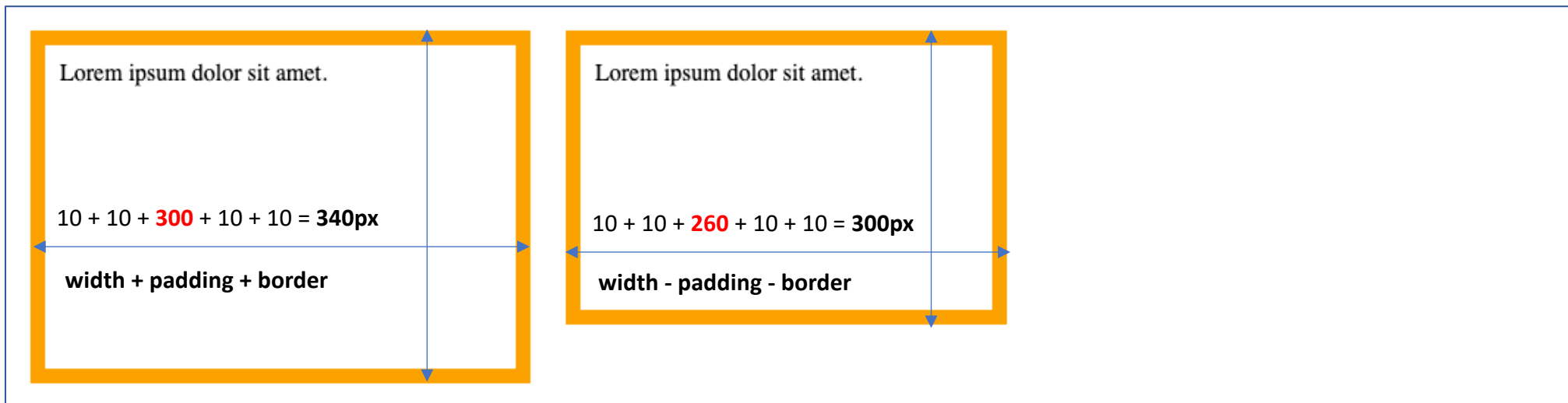
CSS Box-sizing

height + padding + border

$$10 + 10 + 200 + 10 + 10 = 240\text{px}$$

height - padding - border

$$10 + 10 + 160 + 10 + 10 = 200\text{px}$$



```
.box_sizing_content_box {  
  display: inline-block;  
  width: 300px;  
  height: 200px;  
  padding: 10px;  
  margin: 10px;  
  border: 10px solid orange;  
  box-sizing: content-box;  
}
```

```
.box_sizing_border_box {  
  display: inline-block;  
  width: 300px;  
  height: 200px;  
  padding: 10px;  
  margin: 10px;  
  border: 10px solid orange;  
  box-sizing: border-box;  
}
```

```
<div class="box_sizing_content_box">Lorem ipsum dolor sit amet.</div>  
<div class="box_sizing_border_box">Lorem ipsum dolor sit amet.</div>
```

CSS Height & Width

- The **height** and **width** properties are used to set the height and width of an element.
- The **height** and **width** properties **do not include padding, borders, or margins**.
(By default – box-sizing: content-box)
- It sets the height/width of the area **inside the padding, border, and margin** of the element.
(By default – box-sizing: content-box)

auto - This is default. The browser calculates the height and width

length - Defines the height/width in px, cm etc.

% - Defines the height/width in percent of the containing block

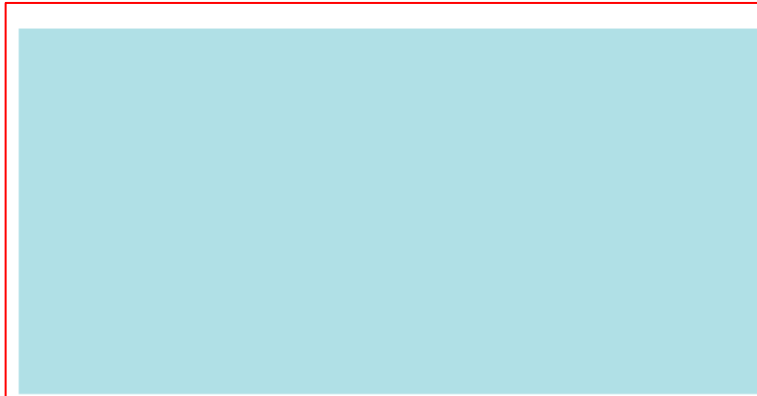
initial - Sets the height/width to its default value

inherit - The height/width will be inherited from its parent value

BOX MODEL

CSS Height & Width

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: powderblue;  
}
```



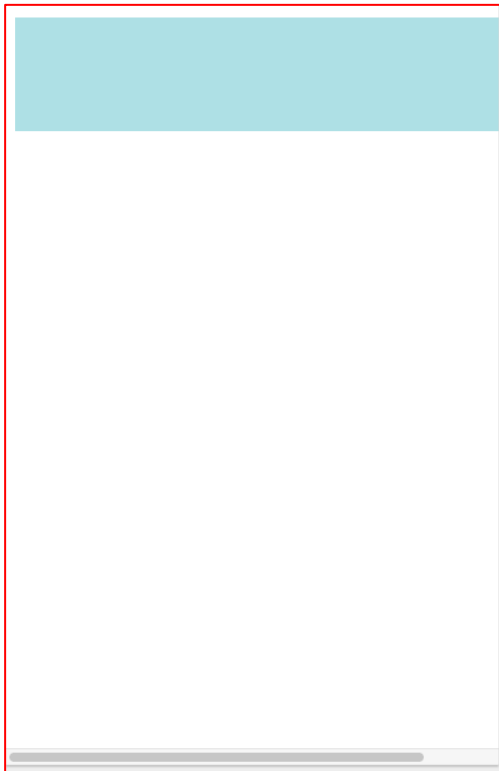
```
div {  
  height: 100px;  
  width: 500px;  
  background-color: powderblue;  
}
```



BOX MODEL

CSS Height & Width – max-width

```
div {  
  height: 100px;  
  width: 500px;  
  background-color: powderblue;  
}
```



```
div {  
  max-width: 500px;  
  height: 100px;  
  background-color: powderblue;  
}
```



BOX MODEL

CSS Display

Block-Level Elements

```
<div>  
<p>  
<h1>–<h6>  
<ol>  
<ul>  
<li>  
<table>  
<tfoot>  
<dl>  
<dt>  
<dd>
```

```
<address>  
<article>  
<nav>  
<section>  
<aside>  
<main>  
<header>  
<footer>  
<blockquote>  
<pre>
```

```
<canvas>  
<figcaption>  
<figure>  
<video>  
<form>  
<fieldset>  
<hr>  
<noscript>
```

By default, display: block;

BOX MODEL

CSS Display

Inline-Level Elements

```
<a>  
<button>  
<img>  
<input>  
<label>  
<select>  
<textarea>  
<span>
```

```
<b>  
<em>  
<i>  
<strong>  
<br>  
<abbr>  
<acronym>  
<bdo>  
<big>  
<cite>
```

```
<code>  
<dfn>  
<kbd>  
<map>  
<object>  
<output>  
<q>  
<samp>
```

```
<script>  
<small>  
<sub>  
<sup>  
<time>  
<tt>  
<var>
```

By default, display: inline;

CSS Display

What is the difference between `display: block;` vs. `display: inline-block;` vs. `display: inline;` ?

```
span.a {  
  display: inline; /* the default for span */  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

```
span.b {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

```
span.c {  
  display: block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.

Aliquam

venenatis

gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

CSS Display vs. CSS Visibility

What is the difference between `display: none;` vs. `visibility: hidden;` ?

```
div.none ul li:nth-of-type(2) {  
  display: none;  
}  
  
div.hidden ul li:nth-of-type(2) {  
  visibility: hidden;  
}
```

```
<div class="none">  
  <ul>  
    <li> One</li>  
    <li>Two</li>  
    <li>Three</li>  
  </ul>  
</div>  
<div class="hidden">  
  <ul>  
    <li>One</li>  
    <li>Two</li>  
    <li>Three</li>  
  </ul>  
</div>
```

- One
- Three
- One
- Three

The **overflow** property specifies **what should happen if content overflows an element's box**.

visible	The overflow is not clipped . It renders outside the element's box. This is default
hidden	The overflow is clipped , and the rest of the content will be invisible
scroll	The overflow is clipped , but a scroll-bar is added to see the rest of the content
auto	If overflow is clipped , a scroll-bar should be added to see the rest of the content

CSS Overflow

```
div.ex1 {  
  background-color: lightblue;  
  width: 110px;  
  height: 110px;  
  overflow: scroll;  
}  
  
div.ex2 {  
  background-color: lightblue;  
  width: 110px;  
  height: 110px;  
  overflow: hidden;  
}  
  
div.ex3 {  
  background-color: lightblue;  
  width: 110px;  
  height: 110px;  
  overflow: auto;  
}  
  
div.ex4 {  
  background-color: lightblue;  
  width: 110px;  
  height: 110px;  
  overflow: visible;  
}
```

```
<div class="ex1">Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</div>  
  
<br>  
  
<div class="ex2">Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</div>  
  
<br>  
  
<div class="ex3">Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</div>  
  
<br>  
  
<div class="ex4">Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</div>
```

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit,
sed diam



overflow: scroll;

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit,
sed diam
nonummy nibh



overflow: hidden;

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit,
sed diam
nonummy



overflow: auto;

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit,
sed diam
nonummy nibh
euismod
tincidunt ut
laoreet dolore
magna aliquam
erat volutpat.



overflow: visible; **by default**

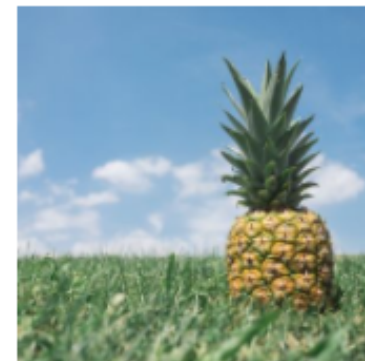
CSS Float

The **float** property specifies how an element **should float**.

```
img {  
  float: right;  
}
```

```
<p>  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,  
nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim  
ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor  
vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula,  
facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus  
interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper  
ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare  
turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla  
congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo  
purus. Mauris quis diam velit.</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



BOX MODEL

CSS Float & Clear

```
img.left {  
  float: left;  
  margin-right: 15px;  
}
```

```
<div class="float-clear">  
    
  <p>  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,  
    nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula  
    venenatis dolor.  
  </p>  
</div>  
  
<h2>New Heading</h2>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

New Heading



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

New Heading