

# Box Model



# TABLE OF CONTENTS

- 01 **Box Model**
- 02 **CSS properties** (padding, margin, border)
- 03 **CSS properties** (box-sizing, height, width, border-radius)
- 04 **CSS properties** (display, visibility, overflow, box-shadow)

# CSS Box Model

---

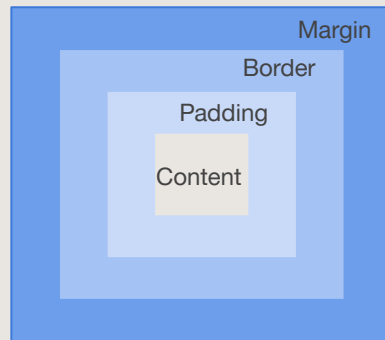




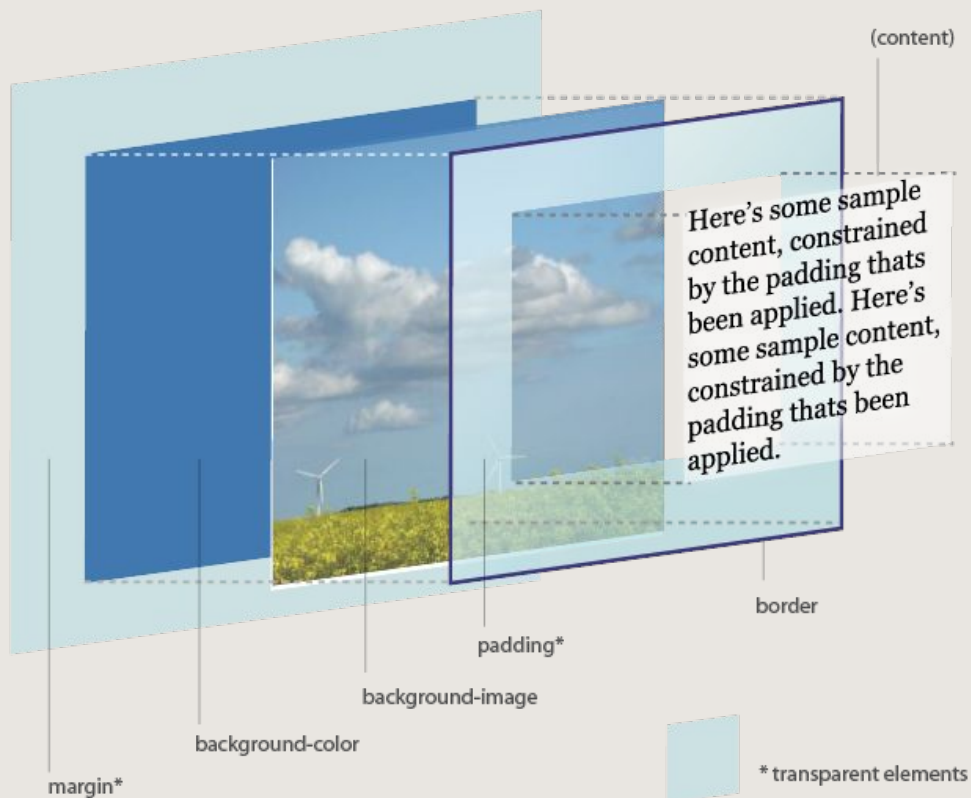
- In CSS, the term "**box model**" is used when talking about **design and layout**.
- The CSS box model is essentially a **box that wraps around every HTML element**.
- All HTML elements can be considered as **boxes**.
- It consists of: **margin**, **border**, **padding**, and the actual **content**.

## Parts of the CSS box model:

- **Margin** - Clears an area **outside the border**.
- **Border** - A border **around the padding and content**
- **Padding** - Clears an area **around the content**.
- **Content** - **where text and images appear**.



# CSS Box Model



# CSS properties

---



- padding
- margin
- border



- The padding properties are used to **generate space around an element's content**.
- CSS has properties for specifying the **padding** for **each side** of an element:
  - padding-**top**
  - padding-**right**
  - padding-**bottom**
  - padding-**left**
- **padding** shorthand property
  - **padding**: 25px /all sides/
  - **padding**: 25px 30px /top & bottom - right & left/
  - **padding**: 25px 30px 35px /top - right & left - bottom/
  - **padding**: 25px 30px 35px 40px /top - right - bottom - left/
- All **padding** properties can have the following values:
  - **length** - specifies a padding in px, pt, cm, etc.
  - **%** - specifies a padding in % of the width of the containing element



- The margin properties are used to **create space around elements**.
- CSS has properties for specifying the **margin** for **each side** of an element:
  - **margin-top**
  - **margin-right**
  - **margin-bottom**
  - **margin-left**
- **margin** shorthand property
  - **margin: 25px /all sides/**
  - **margin: 25px 30px /top & bottom - right & left/**
  - **margin: 25px 30px 35px /top - right & left - bottom/**
  - **margin: 25px 30px 35px 40px /top - right - bottom - left/**
- All **margin** properties can have the following values:
  - **length** - specifies a margin in px, pt, cm, etc.
  - **%** - specifies a margin in % of the width of the containing element
  - **auto** - the browser calculate the margin





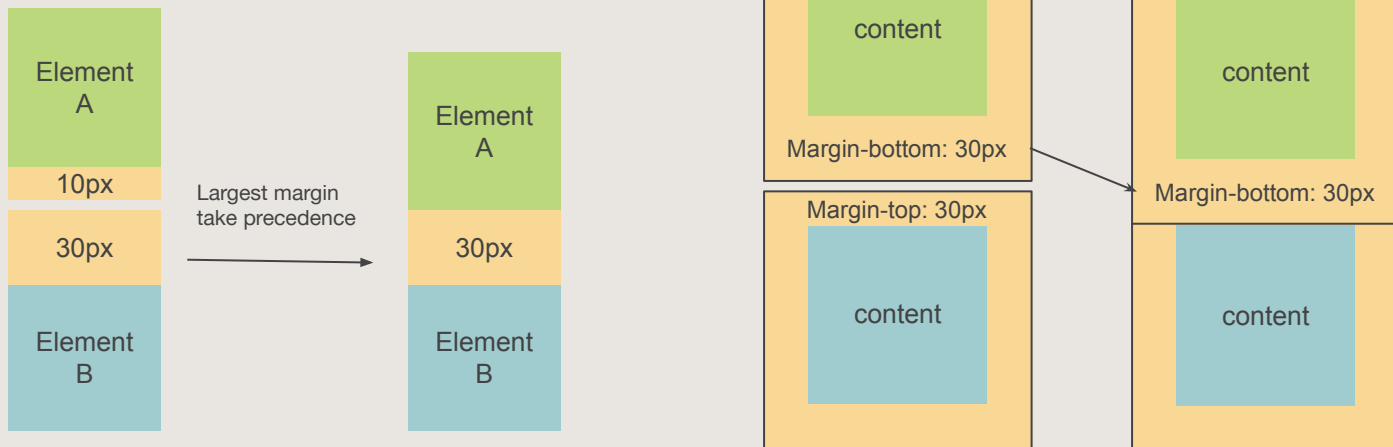
- You can set the margin property to **auto** to **horizontally center the element** within its container.
- The element will take up the **specified width**, and the remaining space will be **split equally between the left and right** margins.
- “**margin – auto**” rule will only work if the **width property** of the element is specified

```
h1 {  
  width: 300px;  
  margin: 50px auto;  
  border: 3px solid blue;  
}
```

This is a heading



- Top and bottom margins of elements collapse into a single margin that is equal to the largest one.
- This does not happen on left and right margins, only top and bottom margins.





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

border **shorthand** property

`h1 {border: 1px solid green}`

border properties

**border-width**

**border-style**

**border-color**

3px by default

Required

black by default

```
h1 {  
  border: 1px solid #423d3d;  
}
```

```
h1 {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #423d3d;  
}
```

(Px, cm, vh, etc)

Thin  
Medium  
thick

Dotted, dashed,  
solid, double,  
groove, ridge,  
inset, outset,  
none, hidden

# CSS properties

---



- box-sizing
- height & width
- border-radius



- The box-sizing CSS property sets how the **total width and height** of an element are calculated.
- It allows us to **include the padding and border** in an element's **total width and height**.
- The default value of **box-sizing** is content-box
- **By default**, the width and height of an element are calculated like this:
  - **width + padding + border** = actual width of an element
  - **height + padding + border** = actual height of an element

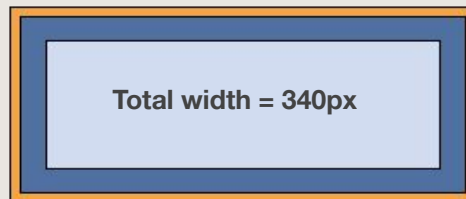
```
selector {  
  box-sizing: content-box/ border-box;  
}
```



The examples below work the same way with the height property

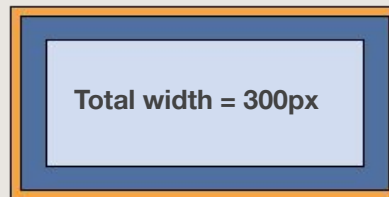
```
div {  
  width: 300px;  
  padding: 15px;  
  border: 5px;  
  box-sizing: content-box;  
}
```

width + 2x padding + 2x border



---

```
div {  
  width: 300px;  
  padding: 15px;  
  border: 5px;  
  box-sizing: border-box;  
}
```





- The height and width properties are used to set the **height and width of an element**.
- The height and width properties may have **different value according to box-sizing** property. **(By default – box-sizing: content-box)**
- It sets the height/width of the area including text content, padding and border of the element.

```
selector {  
  width: value;  
}  
  
selector {  
  height: value;  
}
```

### Values of height & width:

- **auto**: The browser calculates the height and width **(default)**
- **length**: Defines the height/width in **px, cm ...etc.**
- **%**: Defines the height/width **relative to the parent** element
- **initial**: Sets the height/width back to its **default value**
- **inherit**: The height/width will be **inherited from the parent**



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

```
<h1>Border-radius</h1>
```

```
h1 {border-radius:5px}
```

Border-radius

### **border-radius** shorthand property

- **border-radius:** 25px /all corners/
- **border-radius:** 25px 30px /top left - bottom right / top right - bottom left/
- **border-radius:** 25px 30px 35px /top left / top right - bottom left / bottom right/
- **border-radius:** 25px 30px 35px 40px /top left / top right / bottom right / bottom left/



# CSS properties

---



- display
- visibility
- overflow
- box-shadow



- The display property specifies the **display behavior** (the type of rendering box) of an element.
- In HTML, the default display property value is taken from the **HTML specifications** or from the **browser/user default style sheet**.
- The display property accept many different values, the **most important** are:
  - **block**
  - **inline-block**
  - **inline**
  - **flex**
  - **grid**



**Block-level Elements.** By default `display: block;`

```
<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>
```



Inline-level Elements. By default `display: inline;`

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

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

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

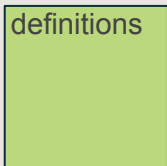
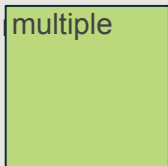
```
<script>  
<small>  
<sub>  
<sup>  
<time>  
<tt>  
<var>
```



- Displays an element as an **inline-level block container**.
- The **element itself** is formatted as an **inline element**, but you can apply **height and width** values.

```
span {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid #434343;  
  background-color: yellowgreen;  
}
```

Design is the method of putting form and content together. Design, just as art, has multiple definitions; there is no single definition. Design can be art.





Displays an element as an **inline element (like <span>)**. Any height and width properties will have no effect

```
span {  
  display: inline;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid #434343;  
  background-color: yellowgreen;  
}
```

Design is the method of putting form and content together. Design, just as art, has **multiple** **definitions**; there is no single definition. Design can be art.

# CSS properties

## Display: block



Displays an element as a **block element (like <p>)**. It starts on a **new line**, and takes up the **whole width**.

```
span {  
  display: block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid #434343;  
  background-color: yellowgreen;  
}
```

Design is the method of putting form and content together. Design, just as art, has

multiple

definitions

; there is no single definition. Design can be art.



The **visibility** CSS property **shows or hides** an element **without changing the layout** of a document.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

```
h3 {
  display: none;
}

h5 {
  visibility: hidden;
}
```

**Heading 1**

**Heading 2**

Heading 4

Heading 6





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

### Property values:

- **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 properties

## overflow



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

← `overflow: scroll;`

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

← `overflow: hidden;`

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

← `overflow: auto;`

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

← `overflow: visible;`



The **box-shadow** property adds shadow to an element.

**box-shadow:** h-offset v-offset blur-radius spread color inset;  
Required Optional

**Negative values** work the same like in text-shadow.

<b>none</b>	Default value. No shadow is displayed
<b>h-offset</b>	The position of the <b>horizontal shadow</b> . Negative values are <b>allowed</b>
<b>v-offset</b>	The position of the <b>vertical shadow</b> . Negative values are <b>allowed</b>
<b>blur-radius</b>	How much the shadow is <b>stretched</b> which causes a blur effect
<b>spread-radius</b>	How much the shadow will expand or shrink. Negative values are <b>allowed</b>
<b>color</b>	The color of the shadow. <b>In Safari</b> the color parameter is <b>required</b>
<b>inset</b>	Changes the shadow from an outer shadow (outset) to an inner shadow

End of the presentation

