CSS Box Model

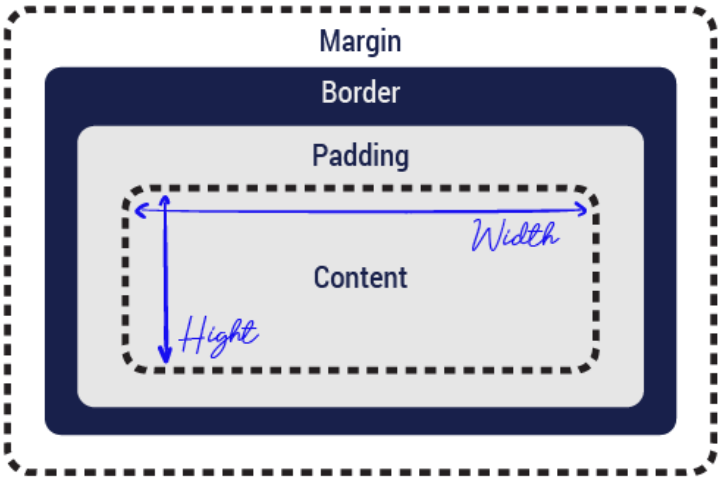
Every element in web design is a rectangular box (even if it is a circle).

The CSS box model is defined by the four layers below:

- Content: Is where text and images appear.
- Padding: Is transparent. It clears area around the content.
- Border: Goes around the padding and content. We have already seen it earlier.
- Margin: Clears an area outside the border. The margin is transparent.

Also, we can determine the dimension of any html element through the `width` and the `height`

We'll get to know them more individually later.



Padding

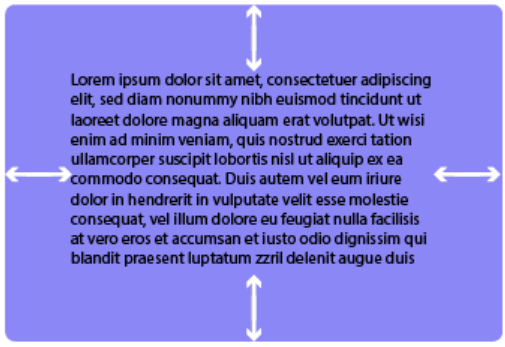
- We use the padding property to create spacing between an element's content area and border. The padding property applies this spacing in different ways depending on how many values you provide.

  The Padding is the INSIDE of the element.

  Suppose we have a div with many HTML elements inside. The problem is, we want to move

the elements inside without moving the div itself. How can we accomplish that? Padding is our hero here!

```
div{
    padding-left: 50px;
    padding-right: 50px;
    padding-bottom: 50px;
    padding-top: 50px;
}
```



- We can treat the padding value separately as we have seen in the previous example, or we can make it in one line like the following code.

```
.box {
  padding: <padding-top> || <padding-right> || <padding-bottom> || <padding-left>
}
```

Margin

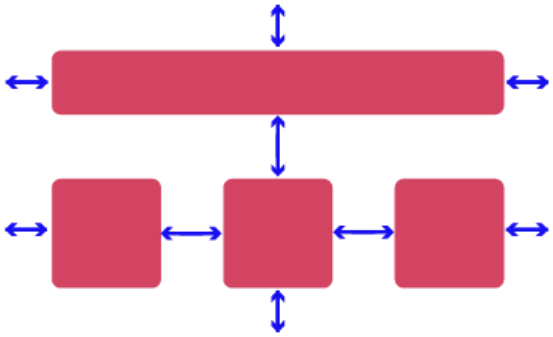
As a first step, go back to the CSS box model scheme to see where the margin is positioned.

Basically, margins allow us to move our tag wherever we want, in any direction: top, bottom, right, or left.

The Margin is the OUTSIDE of the element.

The margin property is very similar to the padding property, except it allows you to define the spacing around the outside of an HTML element past the border. Like padding, it allows you to define single or multiple values.

```
div{
    margin-top:20px;
    margin-bottom:30px;
    margin-left:25px;
    margin-right:40px;
}
```



Height & Width

As we have seen, every element in HTML is a rectangular box. By the laws of geometry, every rectangle is characterized by its width and height.

We can alter these two properties of any HTML content element by using the CSS attribute width or height.

One last thing to always keep in mind:

`Total_width= width+padding_right+padding_left+margin_right+margin_left`

To change the value of these properties, just follow the following example:

- HTML

```
<div>
   <p>I want to get bigger</p>
</div>
```

- CSS

```
div{
    height: 150px;
    width: 90%;
}
```

Border

Like its name implies, the border CSS property sets the border of an element.The border property is a shorthand syntax in CSS that accepts multiple values for drawing a line around the element it is applied to.

```
border:  <border-width> || <border-style> || <color>
```

- `border-width`: specifies the thickness of the border.
    - <length>: A numeric value measured in px, em, rem, vh and vw units.
    - thin: the equivalent of 1px
    - medium: the equivalent of 3px
    - thick: the equivalent of 5px
- border-style: Specifies the type of line drawn around the element, including:
    - solid: a solid, continuous line.
    - none (default): No line is drawn.
    - hidden: a line is drawn, but not visible. This can be useful for adding a little extra width to an element without displaying a border.
    - dashed: a line that consists of dashes.
    - dotted: a line that consists of dots.
    - double: adds two lines around the element
    - groove: adds a bevel based on the color value in a way that makes the element appear pressed into the document.
    - idge: similar to groove, but reverses the color values in a way that makes the element appear raised.
    - inset: adds a split tone to the line that makes the element appear slightly depressed.
        - outset: similar to inset, but reverses the colors in a way that makes the element appear slightly raised.
- color: specifies the color of the border and accepts `<rgb()>`, `<rgba()>`, `<code><hsl()>`, `<hsla()>`, `<hex-color>`, `<named-color>`

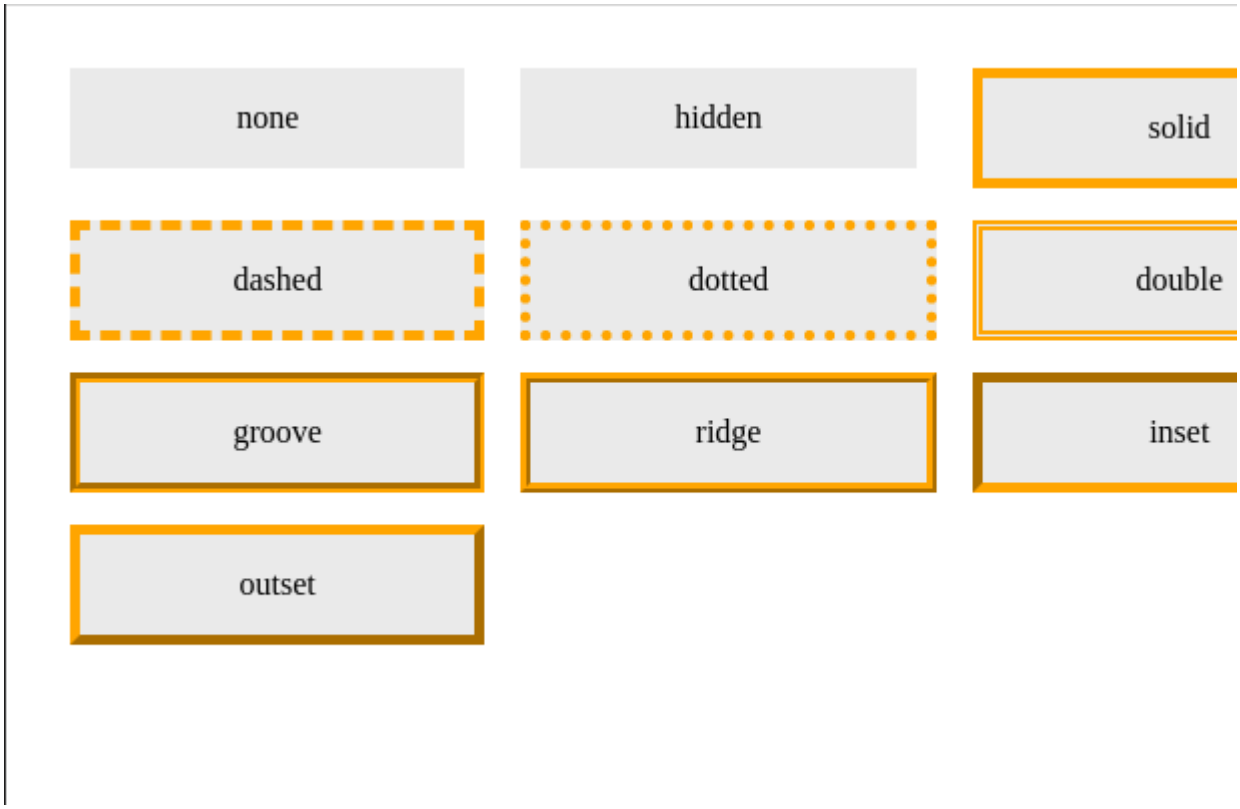
- CSS

```
.box-1 {
  border: none;
}
```

```
.box-2 {
  border: 5px hidden red;
}

.box-3 {
  border: 5px solid orange;
}

.box-4 {
  border: 5px dashed orange;
}

.box-5 {
  border: 5px dotted orange;
}

.box-6 {
  border: 5px double orange;
}

.box-7 {
  border: 5px groove orange;
}

.box-8 {
  border: 5px ridge orange;
}

.box-9 {
  border: 5px inset orange;
}

.box-10 {
```

```
  border: 5px outset orange;
}
```

- Output:



Border Radius

We can give any element “rounded corners” by applying a border-radius through CSS. You’ll only notice if there is a color change involved. For instance, if the element has a background-color or border that is different from the element, it’s incorrect. Simple examples:

- HTML

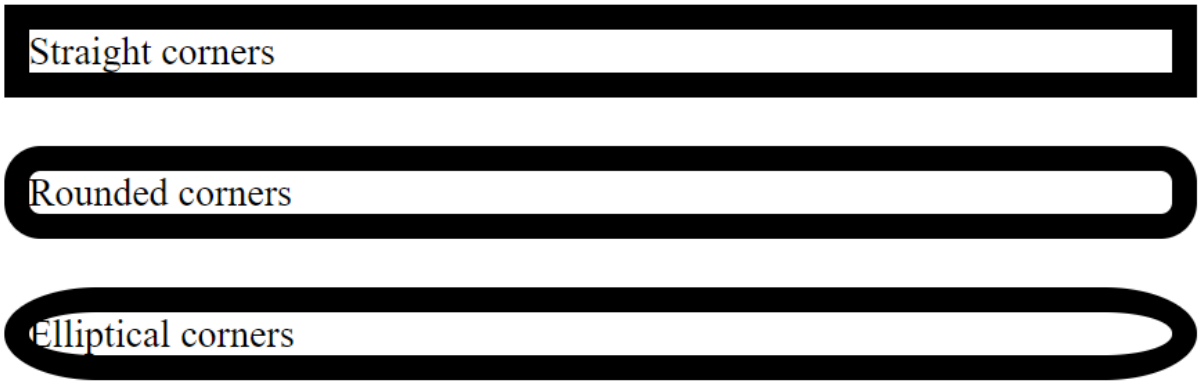
```
<div>
  <p>Straight corners</p>
  <p class="rounded">Rounded corners</p>
  <p class="elliptical">Elliptical corners</p>
</div>
```

- CSS

```
p {
  border: 10px solid black;
  margin: 20px;
```

```
}

.rounded {
  border-radius: 15px;
}

.elliptical {
  border-radius: 50px / 25px;
}
```

- Output:



Previous next