Lesson:

CSS Box Model

PW SKILLS

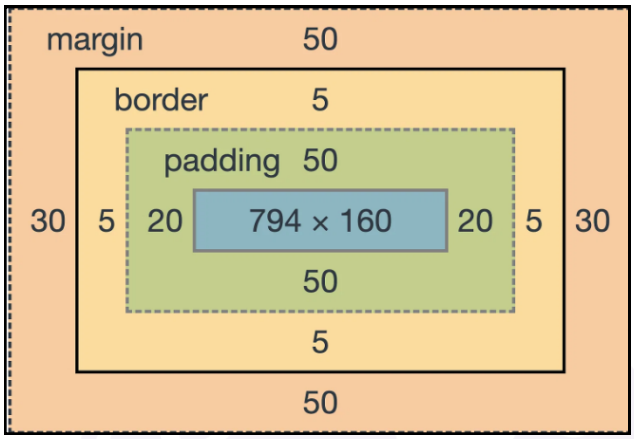
Topics Covered

- Introduction to box model
- Padding
- Border
- Margin
- Margin collapse

Introduction to box model

The CSS box model is a container that contains multiple properties, including **borders, margins, padding**, and the **content** itself. It is used to create the design and layout of web pages. According to the CSS box model, the web browser supplies each element as a square prism.

The following diagram illustrates the box model.



Properties of the box model

Content: The content area consists of content like **images, text**, or **other forms of media content**, the height and width properties help to modify the box dimensions.

Padding: The padding area is the **space around the content area** and within the border box. It can be applied to all sides of the box or to the specific, selected side(s) - top, right, bottom, and/or left.

Margin: The margin area consists of space between the border and the margin, the margin does not possess its own background color and is completely transparent, it shows the background color of the element, like the body element.

Border: The border area surrounds the padding and the content, and can be applied to all the sides of the box or to selected sides/side top, right, bottom, and left.

Note: In the examples, we will mostly use a CSS unit called "pixel" which is represented by "px". However, we can use other units, depending on the example. In the upcoming section, we will explore different units of CSS (absolute and relative) with examples.

Padding

The padding property allows you to specify how much space should appear between the content of an element and its border.

We have the following properties to set an element padding:

- **padding - Individual Sides**

We can specify the padding for each side of an element.

index.html

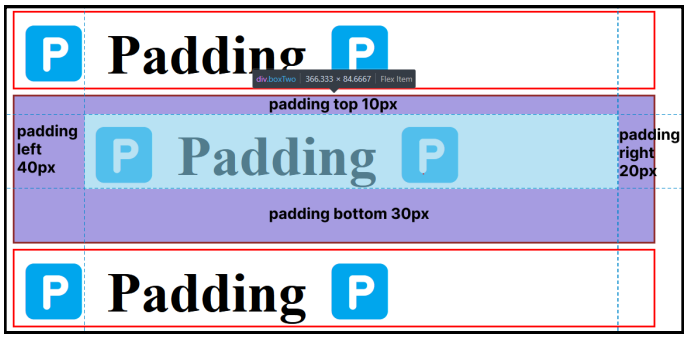
```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="/style.css" />
    <title>Document</title>
  </head>
  <body>
      <div class="boxOne">
           <h1 class="heading" >🅿 Padding  🅿 </h1>
      </div>
      <div class="boxTwo">
           <h1 class="heading" >🅿 Padding 🅿 </h1>
      </div>
      <div class="boxThree">
           <h1 class="heading" >🅿 Padding 🅿 </h1>
      </div>
  </body>
</html>
```

Note: We will use the same HTML code for all the padding examples.

style.css

```
Unset
.boxTwo{
      padding-top: 10px;
      padding-right: 20px;
      padding-bottom: 30px;
      padding-left: 40px;
}
```

Browser output:



- **Padding - Shorthand Property**

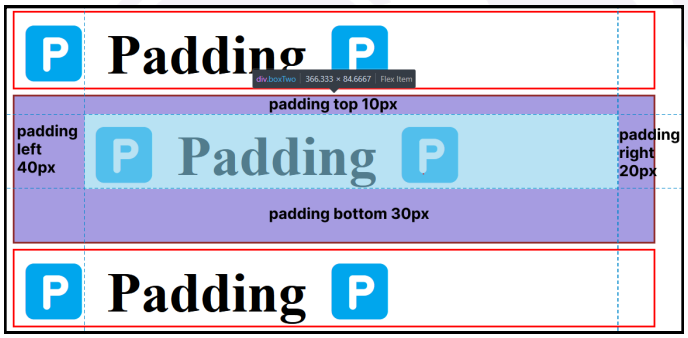
Using **padding**, we can specify all the padding properties in one property.

1. Padding property with four values

style.css:

```
Unset
.boxTwo{
      padding: 10px 20px 30px 40px;
}
```

Browser output:



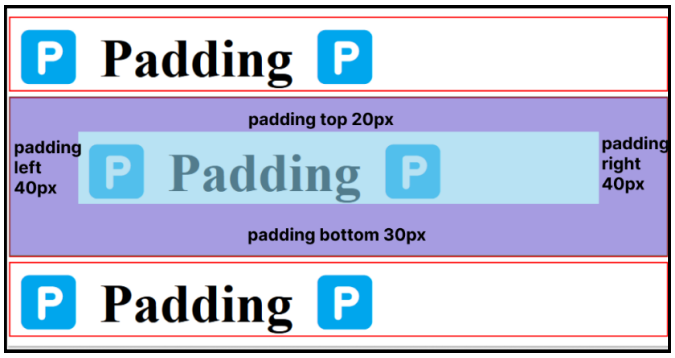
The **first value is for top padding**, the **second value is for right padding**, the **third value is for bottom padding** and the **fourth value is for left padding.** Observe the clockwise pattern.

2. Padding property with three values

style.css:

```
Unset
.boxTwo {
      padding: 20px 40px 30px;
}
```

Browser output:



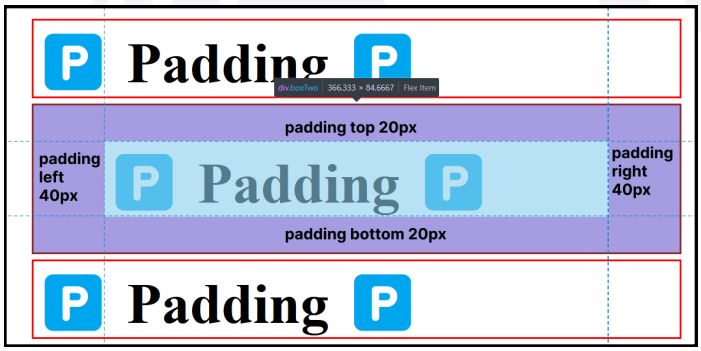
The **first value is for top padding,** the **second value is for left and right padding** and the **third value is for bottom padding.**

3. Padding property with two values

style.css:

```
Unset
.boxTwo {
      padding: 20px 40px;
}
```

Browser output:



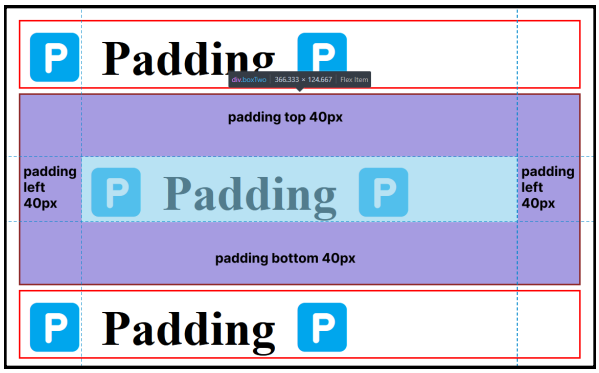
The first value is for the **padding-top and padding-bottom** and the second value is for the **padding-right and padding-left.**

4. padding property with only one value

style.css:

```
Unset
.boxTwo {
      padding: 40px;
}
```

Browser output:



All sides of an element have the same **padding value.**

Border

The border properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change.

1. Border-style

The border-style property allows you to select one of the following styles of border.

- **"none"** - no border.
- **"solid"** - the border is a single solid line.
- **"dotted"** - the border is a single dotted line.
- **"dashed"** - the border is a single dashed line.
- **"double"** - the border is two solid lines.
- **"groove"** - the border looks like it is carved into a page.
- **"ridge"** - the border makes.
- **"inset"** - The border makes the box look like it is embedded in the page.
- **"Outset"** - The border makes the box look like it is coming out of the canvas.

We can also **individually change the border style** of the **left, right, bottom**, and **top** borders of an element by using **border-left-style, border-right-style, border-top-style,** and **border-bottom style.**

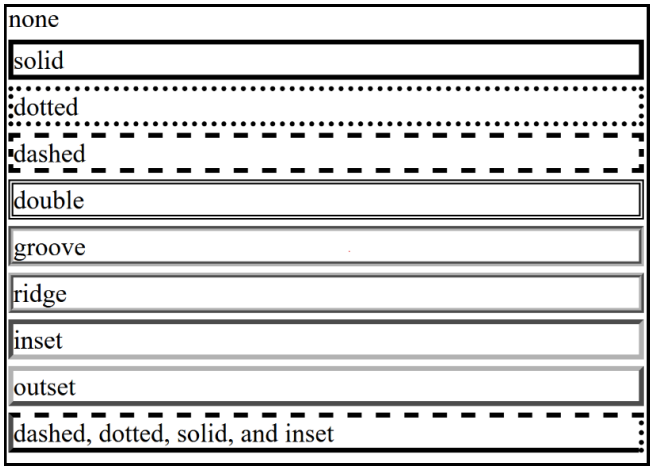
Index.html

```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./style.css" />
    <title>Document</title>
  </head>
  <body>

     <div class="none">none</div>

     <div class="solid">solid</div>

     <div class="dotted">dotted</div>

     <div class="dashed">dashed</div>

     <div class="double">double</div>

     <div class="groove">groove</div>

     <div class="ridge">ridge</div>

     <div class="inset">inset</div>

     <div class="outset">outset</div>

     <div class="mixup">dashed, dotted, solid,and inset</div>

     <div class="mixup">dashed, dotted, solid,and inset</div>

   </body>
 </html>
```

style.css

```
Unset
.none {
  border-style: none;
}
.solid {
  border-style: solid;
}
.dotted {
  border-style: dotted;
}
.double {
  border-style: double;
}
.dashed {
  border-style: dashed;
}
.groove {
  border-style: groove;
}
.ridge {
  border-style: ridge;
}
.inset {
  border-style: inset;
}
.outset {
  border-style: outset;
}
.mixup{
    border-top-style: dashed;
    border-right-style: dotted;
    border-bottom-style: solid;
    border-left-style: inset
}
```

Browser output:



2. Border-width:

The border-width **property allows you to set** the **width of an element's border**. The value of this property could be either a length in **px**, **pt**, or **cm**, or it should be set to **thin, medium**, or **thick**.

We can also **individually set the border width** of the **left, right, bottom**, and **top sides** of an element by using **border-left-width, border-right-width, border-top-width,** and **border-bottom-width.**

html.css

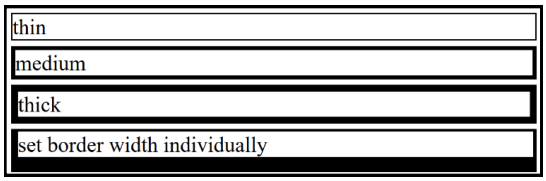
```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./style.css" />
    <title>Document</title>
  </head>
  <body>
    <div class="thin">thin</div>
    <div class="medium">medium</div>
    <div class="thick">thick</div>
    <div class="unitValue">set border width individually</div>
  </body>
</html>
```

Style.css

```
Unset
.thin {
  border-style: solid;
  border-width: thin;
}

.medium {
  border-style: solid;
  border-width: medium;
}

.thick {
  border-style: solid;
```

```
  border-width: thick;
}

.unitValue {
  border-style: solid;
  border-top-width: 2px;
  border-right-width: 2pt;
  border-bottom-width: 0.3cm;
  border-left-width: 5px;
}
```

Browser Output:



3. Border-color:

The border color property allows us to change the border color, we can also change the border-bottom-color, border-top color, border-right-color, and border-left color of an element individually.

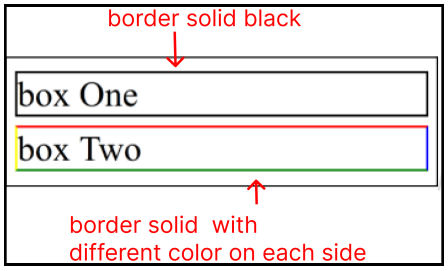
index.html

```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./style.css" />
    <title>border color</title>
  </head>
  <body>
    <div class="boxOne">box One</div>
    <div class="boxTwo">box Two</div>
  </body>
</html>
```

Style.css

```
Unset
.boxOne {
  border-color: black;
  border-style: solid;
}
.boxTwo {
  border-style: solid;
  border-color: red blue green yellow;
}
```

Browser Output:



4. Border-radius

Using the border-radius property, we can set the rounded borders and provide the rounded corners around an element, tag, or div. The border-radius defines the radius of the corners of an element.

Border radius - individual side

Note: We will use the same HTML code for all the border-radius examples.

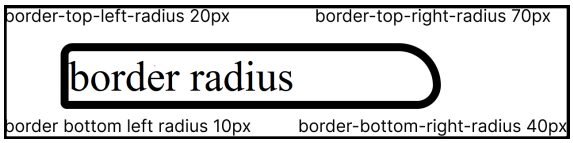
index.html

```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./style.css" />
    <title>Document</title>
  </head>
  <body>
    <div class="box">border radius</div>
  </body>
</html>
```

Style.css

```
Unset
.box {
  border-style: solid;
  border-top-left-radius: 20px;
  border-top-right-radius: 70px
  border-bottom-right-radius: 40px
  border-bottom-left-radius: 10px
}
```

Browser Output:



Border radius - shorthand property

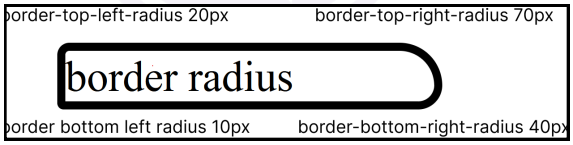
We can specify all the border-radius properties in one property.

I. Border radius with four values

style.css

```
Unset
.box {
  border-style: solid;
  border-radius: 20px 70px 40px  10px;
}
```

Browser Output:

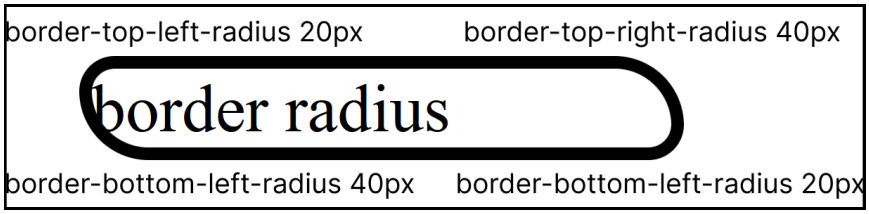


II. Border radius with two values

style.css

```
Unset
.box {
  border-style: solid;
  border-radius: 20px 40px ;
}
```

Browser Output:

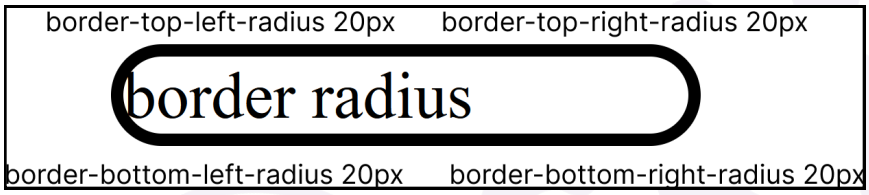


III. Border radius with one value

style.css

```
Unset
.box {
  border-style: solid;
  border-radius: 20px;
}
```

Browser Output:



Border - shorthand property
To shorten the code, it is also possible to specify all the individual border properties in one property **except border-radius.**

The border property is a shorthand property for the following individual border properties:

- border-width
- border-style (required)
- border-color

index.html

index.html

```
Unset
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./style.css" />
    <title>Document</title>
</head>

<body>
    <div class="box">border shorthand</div>
</body>

</html>
```

Style.css

```
Unset
.box {
  border: 5px solid purple;
}
```

Browser Output:

When the border's **non-required** properties are skipped, the **browser will use default values** for those properties, and the **output will be determined by these defaults.**

border-width: If you skip specifying the border width, it will typically **default to "medium,"** which is a browser-specific value, **equal to 1px.**

border-color: If you skip specifying the border color, it will **default to the current text color**, which is usually black.

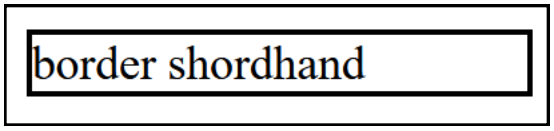
Let's see some examples where we skip specifying border shorthand non-required properties:

For the example, let's use the same HTML code that we had used in the above border shorthand property example.

Style.css

```
Unset
.box {
  border: solid;
}
```

Browser Output:



As you can see in the above example, we have specified the border style as '**solid**,' resulting in a solid line style for the border. The default border width, which is '**medium**,' and the default border color, which is the current text color (usually black), are applied.

Now let's update the above example by adding text color.

Style.css

```
Unset
.box {
  border: solid;
  color: purple;
}
```

Browser Output:

In the above case, we set the text color to purple but didn't specify the '**border-color**' in the 'border' shorthand property. As a result, it takes the current text color **(purple)** as the '**border-color'.**

Margin

The margin property defines the space around an HTML element. It is possible to use negative values to overlap content.

We have the following properties to set an element margin:

Margin - Individual Sides

We can specify the margin for each side of an element.

index.html

```
Unset
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="/style.css" />
    <title>Document</title>
  </head>
  <body>
      <div class="elementOne">Element 1</div>
      <div class="elementTwo">Element 2</div>
      <div class="elementThree">Element 3</div>
  </body>
</html>
```

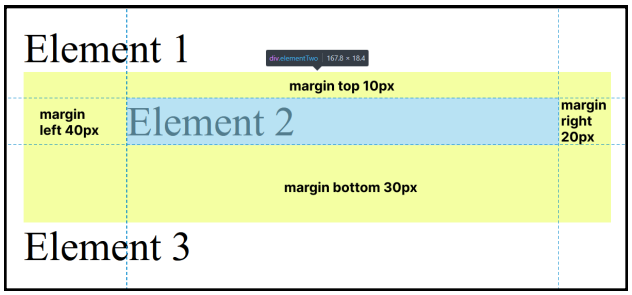
Note: We will use the same HTML code for all the margin examples.

The **margin-**top specifies the top margin of an element.
The **margin-**right specifies the right margin of an element.
The **margin-**bottom specifies the bottom margin of an element.
The **margin-**left specifies the left margin of an element.

Style.css

```
Unset
.elementTwo{
margin-top: 10px;
margin-right: 20px;
margin-bottom: 30px;
margin-left: 40px; }
```

Browser Output:



Margin - shorthand property

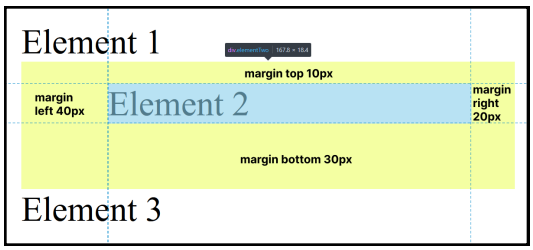
Using the margin, we can specify all the margin properties in one property.

1. Margin property with four values.

Style.css

```
Unset
.elementTwo {
  margin: 10px 20px 30px 40px;
}
```

Browser Output:



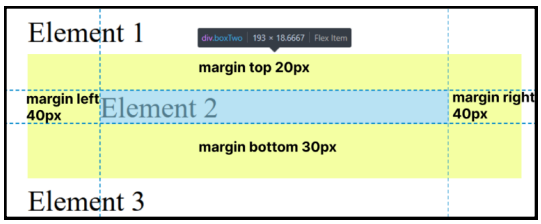
The **first value** is for the **top margin**, the **second value** is for the **right margin**, the **third value** is for the **bottom margin**, and the **fourth value** is for the **left margin**. Observe the clockwise pattern.

2. Margin property with three values.

Style.css

```
Unset
.elementTwo {
margin: 20px 40px 30px
}
```

Browser Output:



The **first value** is for the **top margin**, the **second value** is for the **right margin** and the **left margin**, and the **third value** is for the **bottom margin.**

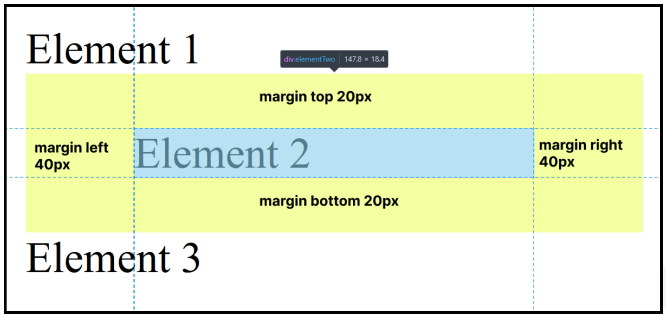
3. Margin property with two values

3. Margin property with two values

Style.css

```
Unset
.elementTwo {
margin: 20px 40px


 }
```

Browser Output:



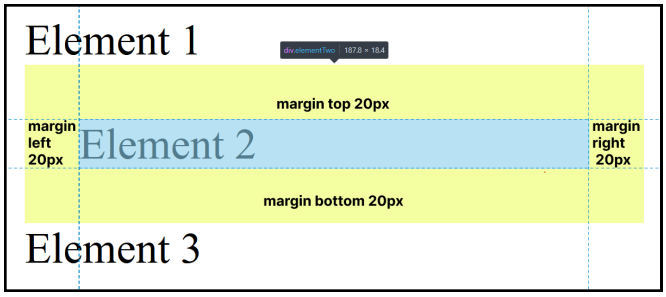
The **first value** is for the **top margin** and **bottom margin** and the **second value** is for the **right margin** and **left Margin.**

4. Margin property with only one value

Style.css

```
Unset
.elementOne{
margin: 10px
}
```

Browser Output:



Margin Collapse

The 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 the left and right margins, only top and bottom margins which means only vertical margins.

Style.css

```
Unset
.elementOne {
margin: 40px 20px


}

.elementTwo {
margin: 20px 0px
}
```

Browser Output: