

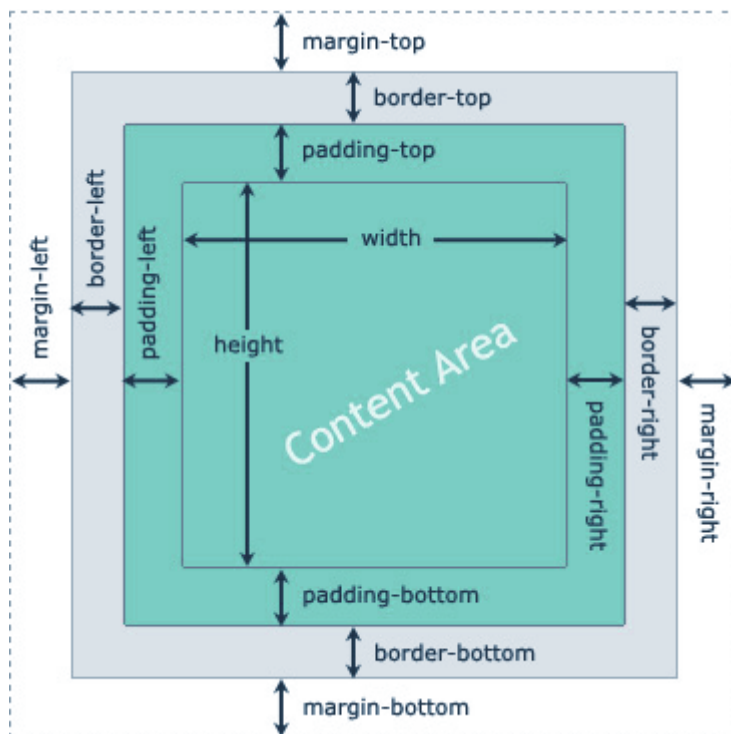
CSS 3 – PART 2

CSS Box Model

What is Box Model?

Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS box model typically describes how these rectangular boxes are laid out on a web page. These boxes can have different properties and can interact with each other in different ways, but every box has a **content area** and optional surrounding **padding**, **border**, and **margin areas**.

The following diagram demonstrates how the width, height, padding, border, and margin CSS properties determines how much space an element can take on a web page.



Padding is the transparent space between the element's content and its border (or edge of the box, if it has no border), whereas margin is the transparent space around the border.

Also, if an element has the [background color](#) it will be visible through its padding area. The margin area is always remain transparent, it is not affected by the

element's background color, however, it causes the background color of the parent element to be seen through it.

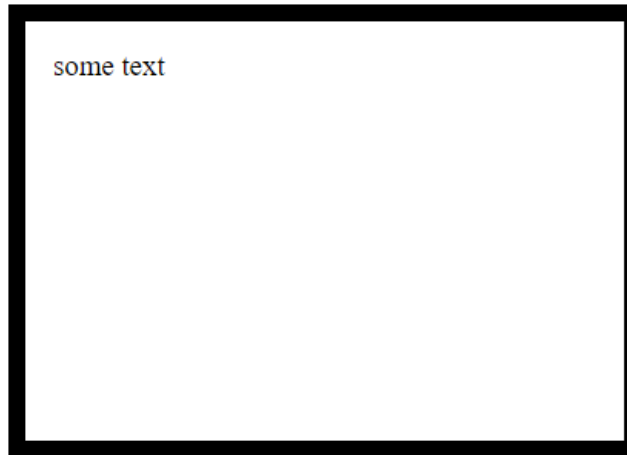
Width and Height of the Elements

Usually when you set the width and height of an element using the CSS `width` and `height` properties, in reality you are only setting the width and height of the content area of that element. The actual width and height of the element's box depends on the several factors.

The actual space that an element's box might take on a web page is calculated like this:

Box Size	CSS Properties
Total Width	<code>width + padding-left + padding-right + border-left + border-right + margin-left + margin-right</code>
Total Height	<code>height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom</code>

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS Box Formatting Model</title>
<style>
  div {
    width: 300px;
    height: 200px;
    padding: 15px;
    border: 10px solid black;
    margin: 20px auto;
  }
</style>
</head>
<body>
  <div>some text</div>
</body>
</html>
```



Note: In CSS box model; the content area of an element's box is the area where its content, e.g., text, image, video, etc. appears. It may also contain descendant elements boxes.

CSS Dimension

Setting Element Dimensions

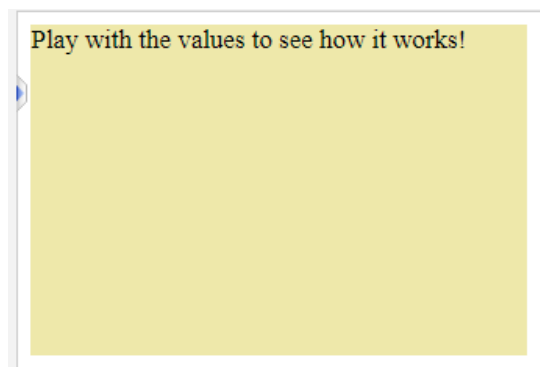
CSS has several dimension properties, such as `width`, `height`, `max-width`, `min-width`, `max-height`, and `min-height` that allows you to control the width and height of an element. The following sections describe how to use these properties to create a better web page layout.

Setting the Width and Height

The `width` and `height` property defines the width and height of the content area of an element.

This width and height does not include paddings, borders, or margins. See the [CSS box model](#) to know how the effective width and height of an element's box is calculated.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>CSS Set Width and Height of an Element</title>
    <style>
      div {
        width: 300px;
        height: 200px;
        background: #eee8aa;
      }
    </style>
  </head>
  <body>
    <div>Play with the values to see how it works!</div>
  </body>
</html>
```



The above style rules assign a fixed width of 300 pixels and height of 200px to the `<div>` element.

The `width` and `height` properties can take the following values:

- *length* - specifies a width in px, em, rem, pt, cm, etc.
- % - specifies a width in percentage (%) of the width of the containing element.
- auto - the browser calculates a suitable width for the element.
- initial - Sets the width and height to its default value, which is `auto`.
- inherit - specifies that the width should be inherited from the parent element.

You can not specify negative values to the width and height properties.

Tip: Typically when you create a block element, such as `<div>`, `<p>`, etc. browser automatically set their width to 100% of the available width, and height to whatever is needed to show all the content. You should avoid setting a fixed width and height unless it is necessary.

Setting Maximum Width and Height

You can use the `max-width` and `max-height` property to specify the maximum width and height of the content area. This maximum width and height does not include paddings, borders, or margins.

An element cannot be wider than the `max-width` value, even if the `width` property value is set to something larger. For instance, if the `width` is set to 300px and the `max-width` is set to 200px, the actual width of the element will be 200px. Let's check out an example:

```
div {
  width: 300px;
  max-width: 200px;
  background: #bbb3ff;
}
p {
  float: left;
  max-width: 400px;
  background: #eee8aa;
}
```

Note: If the `min-width` property is specified with a value greater than that of `max-width` property, in this case the `min-width` value will in fact be the one that's applied.

Likewise, an element that has `max-height` applied will never be taller than the value specified, even if the `height` property is set to something larger. For example, if the `height` is set to 200px and the `max-height` set to 100px, the actual height of the element will be 100px.

```
div {
  height: 200px;
  max-height: 100px;
  background: #bbb3ff;
}
p {
  max-height: 100px;
}
```

```
background: #eee8aa;  
}
```

Note: If the `min-height` property is specified with a value greater than that of `max-height` property, in this case the `min-height` value will in fact be the one that's applied.

CSS Padding

CSS Padding Properties

The CSS padding properties allow you to set the spacing between the content of an element and its border (or the edge of the element's box, if it has no defined border).

The padding is affected by the element's `background-color`. For instance, if you set the background color for an element it will be visible through the padding area.

Define Paddings for Individual Sides

You can specify the paddings for the individual sides of an element such as top, right, bottom, and left sides using the CSS `padding-top`, `padding-right`, `padding-bottom`, and the `padding-left` properties, respectively. Let's try out an example to understand how it works:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8">  
<title>CSS Padding for Individual Sides</title>  
<style>  
  h1 {  
    padding-top: 50px;  
    padding-bottom: 100px;  
    background: lime;  
  }  
  p {  
    padding-left: 75px;  
    padding-right: 75px;  
    background: lime;  
  }  
</style>
```

```
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a simple paragraph of text.</p>
  <p><strong>Note:</strong> Play with the padding property value to see how it
works.</p>
</body>
</html>
```

The padding properties can be specified using the following values:

- *length* - specifies a padding in px, em, rem, pt, cm, etc.
- % - specifies a padding in percentage (%) of the width of the containing element.
- inherit - specifies that the padding should be inherited from the parent element.

Unlike [CSS margin](#), values for the padding properties cannot be negative.

The Padding Shorthand Property

The `padding` property is a shorthand property to avoid setting padding of each side separately, i.e., `padding-top`, `padding-right`, `padding-bottom` and `padding-left`.

Let's take a look at the following example to understand how it basically works:

```
h1 {
  padding: 50px; /* apply to all four sides */
}
p {
  padding: 25px 75px; /* vertical | horizontal */
}
div {
  padding: 25px 50px 75px; /* top | horizontal | bottom */
}
pre {
  padding: 25px 50px 75px 100px; /* top | right | bottom | left */
}
```


This shorthand notation can take one, two, three, or four whitespace separated values.

- If *one value* is specified, it is applied to **all four sides**.
- If *two values* are specified, the first value is applied to the **top and bottom** side, and the second value is applied to the **right and left** side of the element's box.
- If *three values* are specified, the first value is applied to the **top**, second value is applied to **right and left** side, and the last value is applied to the **bottom**.
- If *four values* are specified, they are applied to the **top, right, bottom** and the **left** side of the element's box respectively in the specified order.

It is recommended to use the shorthand properties, it will help you to save some time by avoiding the extra typing and make your CSS code easier to follow and maintain.

Effect of Padding and Border on Layout

When creating web page layouts, adding a padding or border to the elements sometimes produce unexpected result, because padding and border is added to the width and height of the box generated by the element

For instance, if you set the width of a `<div>` element to 100% and also apply left right padding or border on it, the horizontal scrollbar will appear. Let's see an example:

```
div {  
    width: 100%;  
    padding: 25px;  
    background: violet;  
}
```

To prevent padding and border from changing element's box width and height, you can use the CSS `box-sizing` property. In the following example the width and height of the `<div>` box will remain unchanged, however, its content area will decrease with increasing padding or border.

```
div {
```

```
width: 100%;  
padding: 25px;  
background: violet;  
box-sizing: border-box;  
}
```

CSS Border

CSS Border Properties

The CSS border properties allow you to define the border area of an element's box.

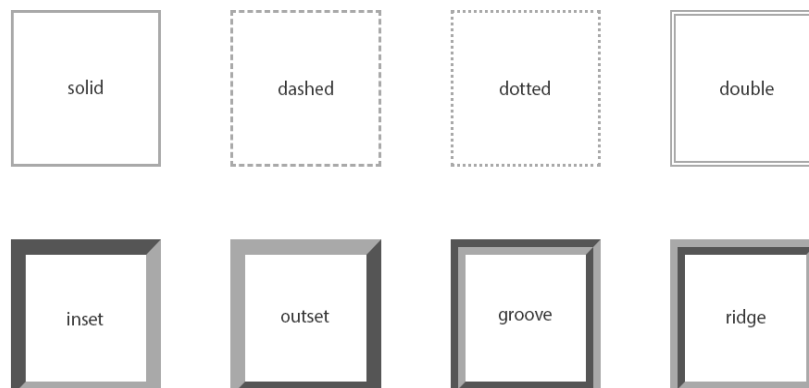
Borders appear directly between the margin and padding of an element. The border can either be a predefined style like, solid line, dotted line, double line, etc. or [an image](#).

The following section describes how to set the style, color, and width of the border.

Understanding the Different Border Styles

The `border-style` property sets the style of a box's border such as: `solid`, `dotted`, etc. It is a shorthand property for setting the line style for all four sides of the elements border.

The `border-style` property can have the following values: `none`, `hidden`, `solid`, `dashed`, `dotted`, `double`, `inset`, `outset`, `groove`, and `ridge`. Now, let's take a look at the following illustration, it gives you a sense of the differences between the border style types.

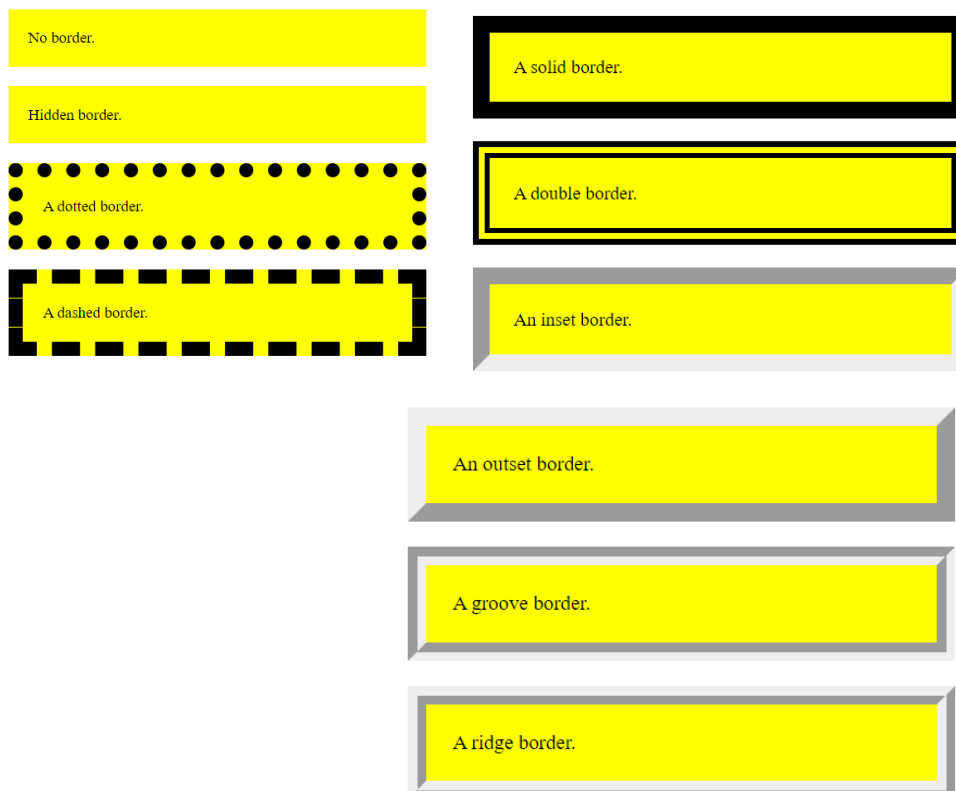


The values `none` and `hidden` displays no border, however, there is a slight difference between these two values. In the case of table cell and border collapsing, the `none` value has the *lowest* priority, whereas the `hidden` value has the *highest* priority, if any other conflicting border is set.

The values `inset`, `outset`, `groove`, and `ridge` creates a 3D like effect which essentially depends on the `border-color` value. This is typically achieved by creating a "shadow" from two colors that are slightly lighter and darker than the border color. Let's check out an example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>CSS border-style Property</title>
    <style>
      p {
        border-width: 15px;
        background: yellow;
        padding: 20px;
        margin: 20px;
      }
      p.none {
        border-style: none;
      }
      p.hidden {
        border-style: hidden;
      }
      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;
    }
</style>
</head>
<body>
  <p class="none">No border.</p>
  <p class="hidden">Hidden border.</p>
  <p class="dotted">A dotted border.</p>
  <p class="dashed">A dashed border.</p>
  <p class="solid">A solid border.</p>
  <p class="double">A double border.</p>
  <p class="inset">An inset border.</p>
  <p class="outset">An outset border.</p>
  <p class="groove">A groove border.</p>
  <p class="ridge">A ridge border.</p>
</body>
</html>
```



Note: You must specify a border style in order to make the border appear around an element, because the default border style is `none`. Whereas, the default border width or thickness is `medium`, and the default border color is the same as the text color.

Setting the Border Width

The `border-width` property specifies the width of the border area. It is a shorthand property for setting the thickness of all the four sides of an element's border at the same time.

Let's try out the following example to understand how it works:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>CSS border-width Property</title>
    <style>
      p {
        border-style: solid;
        padding: 20px;
        margin: 20px;
      }
      p.one {
        border-width: 5px;
      }
      p.two {
        border-width: 5px 10px;
      }
      p.three {
        border-width: 5px 10px 15px;
      }
      p.four {
        border-width: medium 10px thick 15px;
      }
    </style>
  </head>
  <body>
    <p class="one">
      <strong>one-value syntax:</strong> the single value sets the width of all
      four border sides.
    </p>
```

```
<p class="two">
  <strong>two-value syntax:</strong> the first value sets the width of the
  top and bottom border, while the second value sets the width of the right
  and left sides border.
</p>
<p class="three">
  <strong>three-value syntax:</strong> the first value sets the width of the
  top border, the second value sets the width of the right and left border,
  and the third value sets the width of the the bottom border.
</p>
<p class="four">
  <strong>four-value syntax:</strong> each value sets the width of the
  border individually in the order top, right, bottom, and left.
</p>
</body>
</html>
```

Tip: The border width can be specified using any length value, such as px, em, rem, and so on. In addition to length units, the border width may also be specified using one of three keywords: `thin`, `medium`, and `thick`. Percentage values are not allowed.

Specifying the Border Color

The `border-color` property specifies the `color` of the border area. This is also a shorthand property for setting the color of all the four sides of an element's border.

The following style rules adds a solid border of red color around the paragraphs.

```
p {
  border-style: solid;
  border-color: #ff0000;
}
```