

Lesson:

Typography in CSS and Google Fonts



Topics Covered

- Height and width
- Min and max-height
- Min and max-width
- Box sizing

Note: In every example, we will use the most common CSS absolute unit called a pixel, denoted by "px." In the next section, we will see each CSS unit (absolute unit and relative unit) with an example.

height and width

The "**height**" and "**width**" properties are used to set the size of an element. The "**height**" property sets the height of the content area of an element, not including any padding, borders, or margins. The "**width**" property sets the width of the content area of an element, again not including any padding, borders, or margins.

Here's an example of how to use the "height" and "width" properties in CSS:

```
div {
    width: 300px;
    height: 200px;
}
```

Min and max-height:

min-height

The "**min-height**" property sets the minimum height that an element can have. If the content inside the element requires more height than the specified minimum, the element will expand to accommodate the content. If the content inside the element requires less height than the specified minimum, the element will still have the minimum height.

Example-

index.html

```
<h1>The min-height Property</h1>
<h2>min-height: none (default):</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper
diam at erat pulvinar, at pulvinar felis blandit...</p>
<h2>min-height: 200px:</h2>
<p class="minHeight">Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit...</p>
```

style.css

```
.minHeight {
  min-height: 100px;
}
```

Browser Output-

The min-height Property

min-height: none (default):

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit...

min-height: 200px:

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit...

If the content is smaller than the minimum height, the minimum height will be applied.
If the content is larger than the minimum height, the min-height property has no effect.

max-height

The "**max-height**" property sets the maximum height that an element can have. If the content inside the element requires less height than the specified maximum, the element will shrink to fit the content. If the content inside the element requires more height than the specified maximum, the element will still have the maximum height.

Example

Index.html

```
<h1>The max-height Property</h1>
<h2>max-height: none (default):</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.</p>

<h2>max-height: 50px:</h2>
<p class="maxHeight">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.</p>
```

style.css:

```
.maxHeight {
    max-height: 50px;
}
```

Browser output:

The max-height Property

max-height: none (default):

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

max-height: 50px:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

If the content is larger than the maximum height, it will overflow. If the content is smaller than the maximum height, the max-height property has no effect.

min and max-width:

min-width

The "min-width" property sets the minimum width that an element can have. If the content inside the element requires more width than the specified minimum, the element will expand to accommodate the content. If the content inside the element requires less width than the specified minimum, the element will still have the minimum width.

Index.html

```
<h1>The min-width Property</h1>
<h2>min-width: none (default):</h2>
<span>Lorem ipsum dolor sit amet...</span>

<h2>min-width: 500px:</h2>
<span class="minWidth">Lorem ipsum dolor sit amet...</span>
```

style.css:

```
span {
    background-color: yellow;
}

.minLength {
    min-width: 500px;
    display: inline-block;
}
```

Browser output:

The min-width Property

min-width: none (default):

Lore ipsum dolor sit amet...

min-width: 500px:

Lore ipsum dolor sit amet...

The min-width property defines the minimum width of an element.

If the content is smaller than the minimum width, the minimum width will be applied.

If the content is larger than the minimum width, the min-width property has no effect.

max-width

The "max-width" property sets the maximum width that an element can have. If the content inside the element requires less width than the specified maximum, the element will shrink to fit the content. If the content inside the element requires more width than the specified maximum, the element will still have the maximum width.

Index.html

```
<h1>The max-width Property</h1>
<h2>max-width: none (default):</h2>
<p>Lore ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper
diam at erat pulvinar, at pulvinar felis blandit.</p>

<h2>max-width: 150px:</h2>
<p class="maxWidth">Lore ipsum dolor sit amet, consectetur adipiscing
elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit.</p>
```

style.css:

```
.maxWidth {  
    max-width: 150px;  
}
```

Browser output:

The max-width Property

max-width: none (default):

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit.

max-width: 150px:

 Lorem ipsum dolor sit
 amet, consectetur
 adipiscing elit. Etiam
 semper diam at erat
 pulvinar, at pulvinar
 felis blandit.

If the content is larger than the maximum width, it will automatically change the height of the element.
If the content is smaller than the maximum width, the max-width property has no effect.

Box sizing

The box-sizing property sets how the total width and height of an element are calculated.

The syntax of the box-sizing property is given as

```
box-sizing: border-box; // for border-box  
box-sizing: content-box; // for content-box
```

By default box-sizing is set to “content-box”, When using this value, the element's width and height properties will only consider the content, and will not include the margin, padding, or border values.

For example, if we specify the width of an element as 200px, the content will be exactly 200px wide, while the border and padding will be added to the final display, making the element wider than 200px.

The formula for calculating box size in the case of **box-sizing: "content-box"** is

```
Actual Width = Width + Horizontal Padding + Horizontal Border  
Actual Height = Height + Vertical Padding + Vertical Border
```

It means if we create a box of some specific height and width and then add padding and a border to it, it will look wider than the actual width.

Example: Let's create two div elements with the same height and width, but different borders and padding.

Index.html

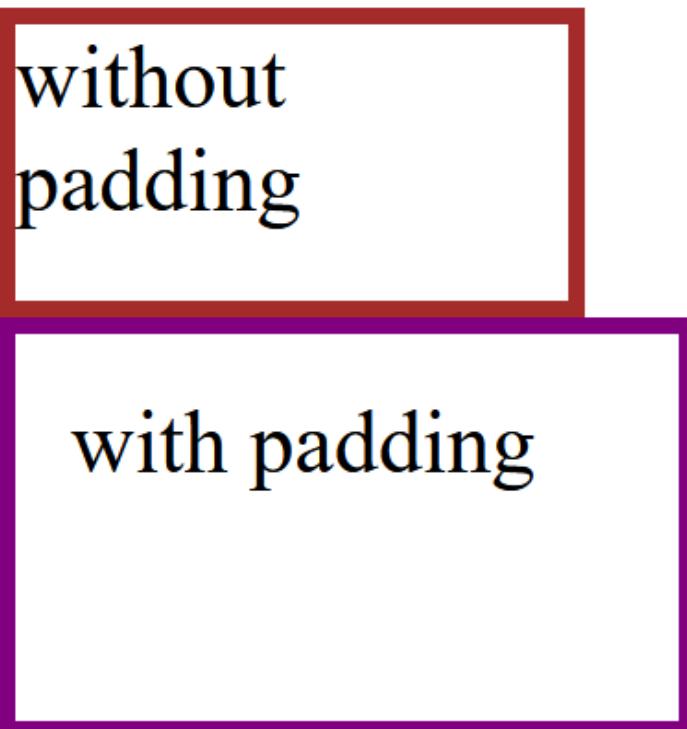
```
<div class="withoutPadding">without padding</div>
<div class="withPadding">with padding</div>
```

Style.css

```
.withoutPadding {
    height: 50px;
    width: 100px;
    border: solid brown;
}

.withPadding {
    height: 50px;
    width: 100px;
    border: solid purple;
    padding: 10px;
}
```

Browser output:



From the above output, it's clear that the two divs have different sizes, even though we specified the same size for both. The second div appears larger than the first one.

This issue can be resolved by changing box-sizing property to "**border-box**"

The "**border-box**" value for the **box-sizing** property is commonly used to tell the browser to **fit the element's border and padding within its specified width and height**.

For example, if we set an element's width to 200px, the border and padding we specify will be included in that 200px, and the content box will adjust to any extra width.

Example: Let's compare border-box and content-box with the below example.

In the below example, we have two containers, one with **box-sizing: "border-box"** and another with default **box-sizing:"content-box"**(no need to set explicitly)

index.html

```
<div class="withBoxSizing">box-sizing:border-box</div>
<div class="withoutBoxSizing">without box-sizing</div>
```

style.css

```
.withBoxSizing {
    height: 100px;
    width: 200px;
    border: solid 2px;
    padding: 10px;
    box-sizing: border-box;
}

.withoutBoxSizing {
    height: 100px;
    width: 200px;
    border: solid 2px;
    padding: 10px;
}
```

Browser output-

