



[Course](#) > [Modul...](#) > [5.5 CSS...](#) > [Managi...](#)

Managing element size

Managing element size

As you start to leverage borders, background colors, and the other decorative CSS properties we have seen in the previous sections you will need to become more aware of the element size and how to manage it.

The wrong way

A common trap that newbies fall into is to discover the `width`, `height`, `left` and `top` CSS properties and to start blindly using them.

These are useful properties, however, they should **not** be your first choice. These CSS properties depend upon *other* properties before they can even be used, and they can have unintended consequences. We'll explore them more in next module when covering Layout.

Here is a simple example of a common error:

| CSS | Result |
|---|---|
| <pre>div { width: 40px; border: 2px solid red; font-size: 18px; }</pre> | <div>CSS is Awe som e</div> |

Padding - The right way

The best way to control an element's size for the purpose of controlling how a border or background extends relative to the item itself is with the `padding` properties.

Padding is a way of making an element a little bigger than it would normally be. Similar to margin, there are four padding properties. The values are dimension units that can be 0 or a positive value; negative values are not allowed.

```
padding-top:    0;  
padding-right:  1em;  
padding-bottom: 10px;  
padding-left:   12px;
```

Similar to `margin`, this can be abbreviated with the `padding` property.

```
padding: <top> <right> <bottom> <left>;  
padding: <top and bottom> <right and left>;  
padding: <all>;
```

When decorative CSS is not used by many CSS newbies, use padding like margin, to space things out. Note: That is not correct. Margins make space between elements and padding makes an element larger.

In this example, note that the use of padding does not make the text deviate from the baseline. In addition, note that in the first paragraph we apply the same padding to all four sides. However, in the second paragraph we use different padding for different sides, thus placing the rectangle of the element relative to the text itself.

HTML

```
<p class="even-pad">My friends <span>Mr. Thomas</span> and <span>Mr. Joyce</span>  
both hail from Ireland.</p>  
  
<p>But <span class="proust">Mssr. Proust</span> and <span class="voltaire">Mssr.  
Arouet</span> are from France.</p>
```

CSS

Result

```
span {  
  background-color: tan;  
}  
.even-pad span {  
  padding: 15px;  
}  
.proust {  
  padding: 2px 15px 15px  
2px;  
}  
.voltaire {  
  padding: 15px 2px 2px  
15px;  
}
```

My friends Mr. Thomas and Mr. Joyce both hail from
Ireland.

But Mssr. Proust and Mssr. Arouet are from France.

I'm confused - are you saying that I shouldn't use the `width` property to change the width?

Essentially yes. Let's qualify that a bit so there is no confusion.

This section has been about the decorative CSS properties, like borders, background colors, background images, and shadows. In the context of these things users often want to make an element "bigger", or keep the border a bit away. And in this context the `padding` property is absolutely the correct property that should be used to increase the elements size. Do not use the `width` or `height` properties for this.

Furthermore, there are other issues associated with the `width` and `height` properties. We will discuss these in-depth in the next module. Here is a quick rundown:

- `height` and `width` properties do not work on inline elements.
- Many elements have natural behaviors that occur when `height` and `width` are **not** set. These are generally advantageous. However, by setting the `width` and `height` you **lose** those advantages. We'll understand this in next module.
- Most Web pages are viewed in a variety of browser sizes, especially on mobile devices like phones or tablets. Overuse of explicit `width` and `height` can make your page unviewable on smaller devices.
- The flexbox layout system (which we will see next) is incredibly powerful, but by over-determining explicit heights and widths you reduce its usefulness to you and your viewer.