## **The Box Model**

### **Introduction**

Now that you understand the basic syntax of HTML and CSS, we’re going to get serious. The most important skills you need to master with CSS are *positioning* and *layout*. Changing fonts and colors is a crucial skill, but being able to put things exactly where you want them on a webpage is even more crucial. After all, how many webpages can you find where absolutely every element is just stacked one on top of another?

### **Learning Outcomes**

* You’ll learn all about *the box model*.
* You’ll learn how to make sure elements are just the right size with margin, padding, and borders

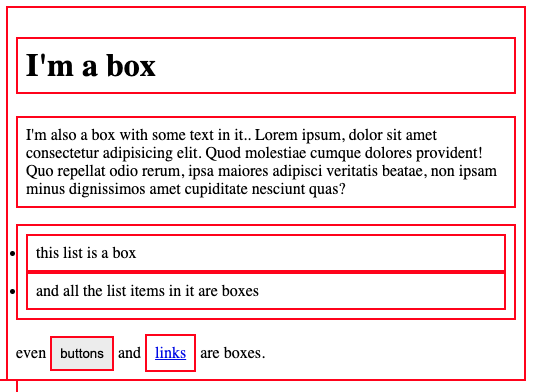
Learning to position elements on a webpage is not that difficult once you understand just a few key concepts. Unfortunately many learners race through learning HTML and CSS to get to JavaScript, and end up missing these fundamental concepts. This leads to frustration, pain ([and funny gifs](https://giphy.com/gifs/css-13FrpeVH09Zrb2)), because all of the JavaScript skills in the world are meaningless if you can’t stick your elements on the page where you need them to be. So with that in mind, let’s get started.

### **The Box Model**

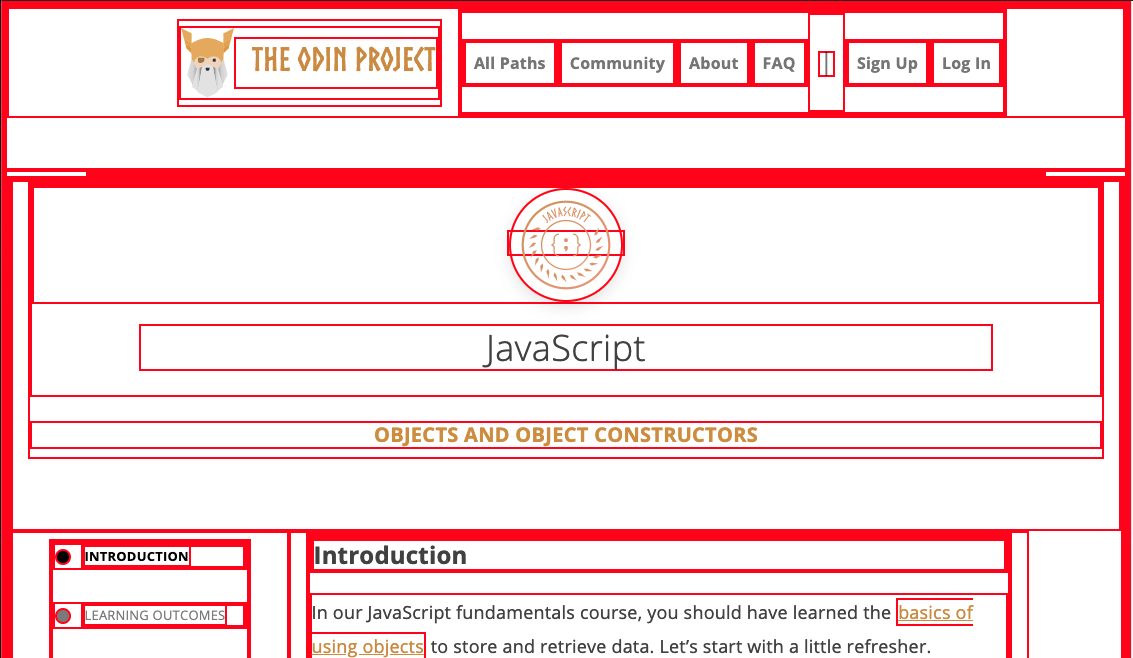
The first important concept that you need to understand to be successful in CSS is the box model. It isn’t complicated, but skipping over it now will cause you much frustration down the line.

Basically, every single thing on a webpage is a rectangular box. These boxes can have other boxes in them and can sit alongside one another. You can get a rough idea of how this works by sticking a border on every item on the page like this:

\* { border: 2px solid red }

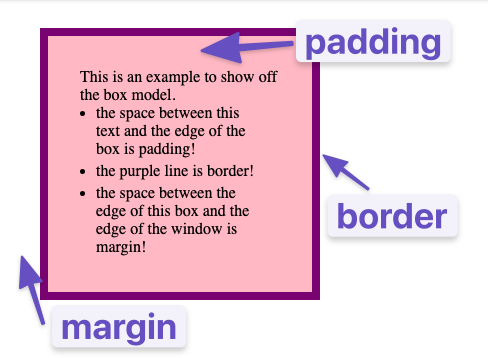


You can use the browser’s inspector to add the CSS above to this web page if you want. Boxes in boxes!



OK, so there might be some circles in the above image… but when it comes to layout, they fit together like rectangular boxes and not circles. In the end, laying out a webpage and positioning all of its elements is basically just deciding how you are going to nest and stack these boxes.

The only real complication here is that there are multiple ways to manipulate the size of these boxes, and the space between them, using margin, padding, and border. The assigned articles go into more depth on this concept, but to sum it up briefly: padding increases the space between the edge of a box and the content inside of it; margin increases the space between a box and any others that sit next to it; and border adds space (even if it’s only a pixel or two) between the margin and the padding. Be sure to study the diagrams carefully.



1. [This video](https://www.youtube.com/watch?v=rIO5326FgPE) is a straightforward overview of the box model, padding and margin. Go ahead and watch this now; it informs everything else.
2. Because the box model concept is so incredibly fundamental, check out [this lesson from MDN](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model). It covers the same material as the video above, but it goes a little further in depth. Pay close attention to the examples and take the time to experiment with their in-browser editor!
3. [This CSS Tricks page](https://css-tricks.com/almanac/properties/m/margin/) has some further information about the margin property that you’ll find useful. Specifically, the sections about auto and margin collapsing contain things you’ll want to know.

### **Additional Resources**

This section contains helpful links to other content. It isn’t required, so consider it supplemental.

### **Knowledge Check**

This section contains questions for you to check your understanding of this lesson. If you’re having trouble answering the questions below on your own, review the material above to find the answer.

* What does the box-sizing CSS property do?
* Would you use margin or padding to create more space between 2 elements?
* Would you use margin or padding to create more space between the contents of an element and its border?