Understanding the Box Model

**What is the Box Model?**

In the last unit, you learned how to organize HTML code with IDs, classes, and divs. All of these concepts are necessary to understand advanced CSS styling techniques. An understanding of CSS is not complete, however, without a study of the *box model*.

In some of the past exercises, you've unknowingly seen aspects of the box model. For example, when you set the background color of an element, you may have noticed that the background color was applied not only to the area directly behind the element, but also to the area to the right of the element. In another exercise, you learned how to align the text. How did the browser know how to align the text?

All HTML elements live within a box. Elements on a web page are understood by the browser as "living" inside of a container, or a box. This is what is meant by the *box model*.

When you changed the background color of an element, you changed the background color of its entire box. When you aligned text, the text was aligned relative to the element's entire box. To truly create custom websites, you'll have to understand the box model.

In this unit, you'll learn about the following aspects of the box model:

1. The dimensions of an element's box
2. The borders of an element's box
3. The content within an element's box
4. The area around all four sides of an element's box

Let's begin!

# Width & Height

An element's box has two dimensions: a height and a width. In HTML, all boxes have default dimensions. These default dimensions are automatically set to hold the raw contents of the box.

To modify the default dimensions an element's box in CSS, you can use the width and height properties.

These two properties can be specified with the following units of measurement:

1. Pixels - You learned about pixels when you learned about fonts. This unit lets you set the exact size of an element's box.
2. Ems - This unit sets the dimensions of the box relative to the size of the text within the box.
3. Percentages - This unit sets the dimensions of the box relative to the size of the box that it is encased in. For example, consider an element (a box) of class blue set to a height of 200 pixels and a width of 200 pixels. Inside of blue, consider another box of class red, set to a height of 37% and a width of 45%. The resulting dimensions of the red box would be a height of 74 pixels and a width of 90 pixels.

Developers often use ems or percentages to set box dimensions. Because many web pages are now accessed on mobile devices and other displays of differing sizes, ems and percentages allow boxes to scale depending on the size of a user's display.

**Width: Minimum & Maximum**

Because a web page can be viewed through displays of differing screen size, the content on the web page can suffer from those changes in size. To avoid this problem, CSS offers two properties that can limit how narrow or how wide an element's box can be sized to.

1. min-width - this property ensures a minimum width for an element's box.
2. max-width - this property ensures a maximum width for an element's box.

p {

min-width: 300px;

max-width: 600px;

}

In the example above, the width of all paragraphs will not shrink below 300 pixels, nor will the width exceed 600 pixels.

Content, like text, can become difficult to read when a browser window is narrowed or expanded. These two properties ensure that content is legible by limiting the minimum and maximum widths of an element.

**Height: Minimum & Maximum**

You can also limit the minimum and maximum *height* of an element.

1. min-height - this property ensures a minimum height for an element's box.
2. max-height - this property ensures a maximum height for an element's box.

p {

min-height: 150px;

max-height: 300px;

}

In the example above, the height of all paragraphs will not shrink below 150 pixels and the height will not exceed 300 pixels.

What will happen to the contents of an element's box if the max-height property is set too low? It's possible for the content to spill outside of the box, resulting in content that is not legible. You'll learn how to work around this issue in the next exercise.

**Overflow**

When the value of the max-height property is set too low, the contents will spill outside of the box. How can we ensure that this doesn't happen?

The overflow property controls what happens to content when it spills, or *overflows*, outside of its box. It can be set to one of the following values:

1. hidden - when set to this value, any content that overflows be hidden from view.
2. scroll - when set to this value, a scrollbar will be added to the element's box so that the rest of the content can be viewed by scrolling.

p {

min-width: 300px;

max-width: 600px;

min-height: 150px;

max-height: 300px;

overflow: scroll;

}

In the example above, minimum and maximum widths and heights have been set. If any of the paragraph content overflows (perhaps a user resizes their browser window), a scrollbar will appear so that users can view the rest of the content.