



Chapter 11 - FormattingThe Box

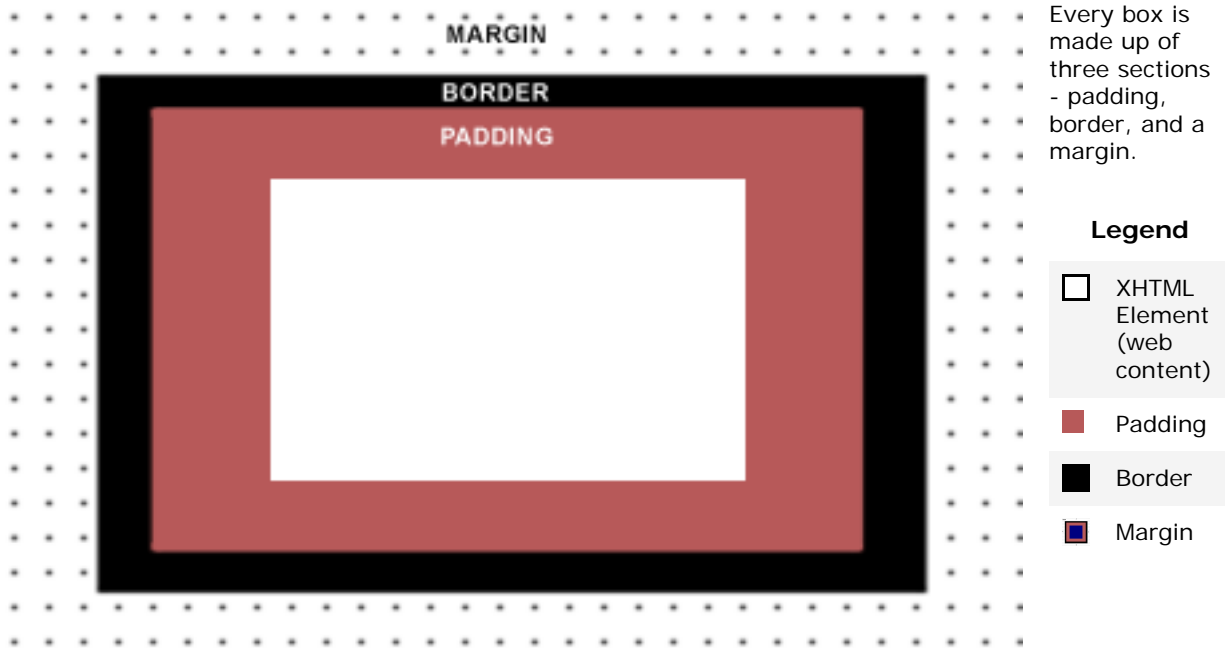
This is an excerpt from Joy Of Code's Workshop - Introduction To XHTML And CSS. For further details see joyofcode.com or contact Bud Kraus, friendly@joyofcode.com, 973 235 1452.

What Is The Box?

Surrounding every XHTML element is a **box** or better said, the space that is to the top, right, bottom, and left of the element.

What could this be? A box encases every XHTML element - paragraph, table, list, link, or embedded image - to name a few. Each is surrounded by a box. Be it a block-level or inline tag, there always is a box. The problem is, the box is invisible!!

Table 23 - The Box Model



Every box is made up of three sections - padding, border, and a margin.

Legend

- XHTML Element (web content)
- Padding
- Border
- Margin

How you use the box to provide white space for web content is what "formatting the box" is all about.

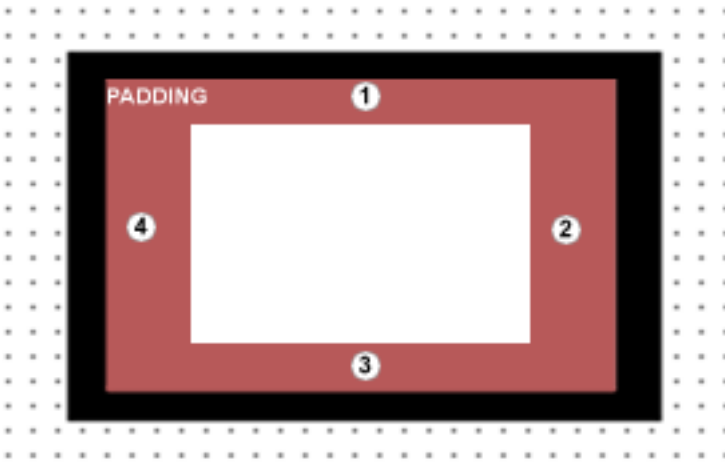


Padding

Immediately adjacent to your content - such as a paragraph, table, image or list - is the space known as Padding. Think of this space as something that starts from the edge of an XHTML element and ends at the border (which I'll get to in a minute).

Padding, like most CSS properties, has four distinct sides - top, right, bottom, and left. The size of each of these can be defined in relative units, such as percentages, or in absolute units, like pixels.

Table 24 - Padding



The padding space is defined as top, right, bottom, and left. It lies between the XHTML element (web content) and the border.

Legend

- 1 padding-top
- 2 padding-right
- 3 padding-bottom
- 4 padding-left

Suppose you want to pad several consecutive paragraphs as a group on a page. The `<div>` `</div>` container let's us break a page into logical sections. It will come in real handy here.

Example Using Padding

Code View

```
<div style="padding-top: 15px; padding-right: 10px; padding-bottom: 15px; padding-left: 10px; background-color: #B75959;">
```

```
<p>It's hard to believe that we're coming to the end of the Joy Of Code Workshop. Hope you've gotten everything you wanted from this experience and more.</p>
```

```
<p>Developing and designing web sites are skills. It takes time and practice to get good at it, but it will come. I hope that over time you will use what you're learning to put your own special stamp on the World Wide Web.</p>
```

```
</div>
```

Browser View

It's hard to believe that we're coming to the end of the Joy Of Code Workshop. Hope you've gotten everything you wanted from this experience and more.

Developing and designing web sites are skills. It takes time and practice to get good at it, but it will come. I hope that over time you will use what you're learning to put your own special stamp on the World Wide Web.

See more [padding property-value pairings](#).

1. While you can use a shorthand for padding, I decided to specify each area - top, right, bottom and left - as shown above. Notice the order - top, right, bottom and left - which is clockwise. It's not required that you write the rule this way, but you'll find the pattern convenient when defining padding dimensions.

Had I chosen to use a shorthand instead, all I'd have to do is:

```
padding: 15px 10px 15px 10px;
```

Following the padding property are the four values in clockwise order - top, right, bottom, and left.

The same idea applies when defining space for border and margin which you'll see later on.

2. I used fixed sizing here, that is, the unit of measurement selected was pixels. I frequently use fixed dimensions when defining padding, border, and margin, but you are free to try and use proportional values.
3. I used a local style rule to the div element to make it easier to see the effect. You can write your style rules for padding, border, and margin in the same locations as you learned in Chapter 8 - Formatting Text, that is, you can still write them in an External .css File, as an Internal Rule to an .html file, or as a Local Rule as I did above.
4. You don't have to choose a background-color for padding. I'm doing it for illustrative purposes. Had I not picked a color, it would have been white, which is what I specified for body in my external .css file, global.css.

Actually, in the interest of full disclosure, that's not 100% true. I wanted this example to track with (look like) the graphic box in Table 24 above. So, if you look at the source code (in your browser) of the padding example, you'll see this code (right after the div element containing the padding markup and just before the first paragraph):

```
<!-- this div section was added to make it easier to see the box -->  
<div style="background-color: #ffffff; color: #000000;">
```

Had I not done that, the entire example would have had a pink (#B75959) background, i.e., the text would have had a pink background too.

And while you're looking at the source, note the second `</div>`. That's right. I'm nesting a `<div>` `</div>` container.