Box Model

CSS3

What is the Box Model?

Every visible element in a web page follows the box model. This model essentially illustrates which parts of the element we can manipulate, size, colour, and/or style to achieve our goals.

There are two types of boxes that our elements may typically be:

1. Block Think "larger" boxes that contain text or graphic content.
    - appears on a new-line
    - width will fill the available space by default (auto)
    - height will be just large enough to fit the element's contents by default (auto)
    - width and height can be adjusted to specific values
    - examples of default block elements: H1, SECTION, P, and LI

2. Inline Think hyperlinks, text, and graphics within a block of text.
    - appears beside other inline elements
    - width and height match the text or graphic contents by default
    - width and height cannot be overridden or adjusted manually
    - only horizontal padding, margins, and borders will take up space and push other inline elements away (vertical will still display but not affect other elements)
    - examples of default inline elements: A, STRONG, EM, and TIME

Parts of a Box

All boxes possess the following parts:

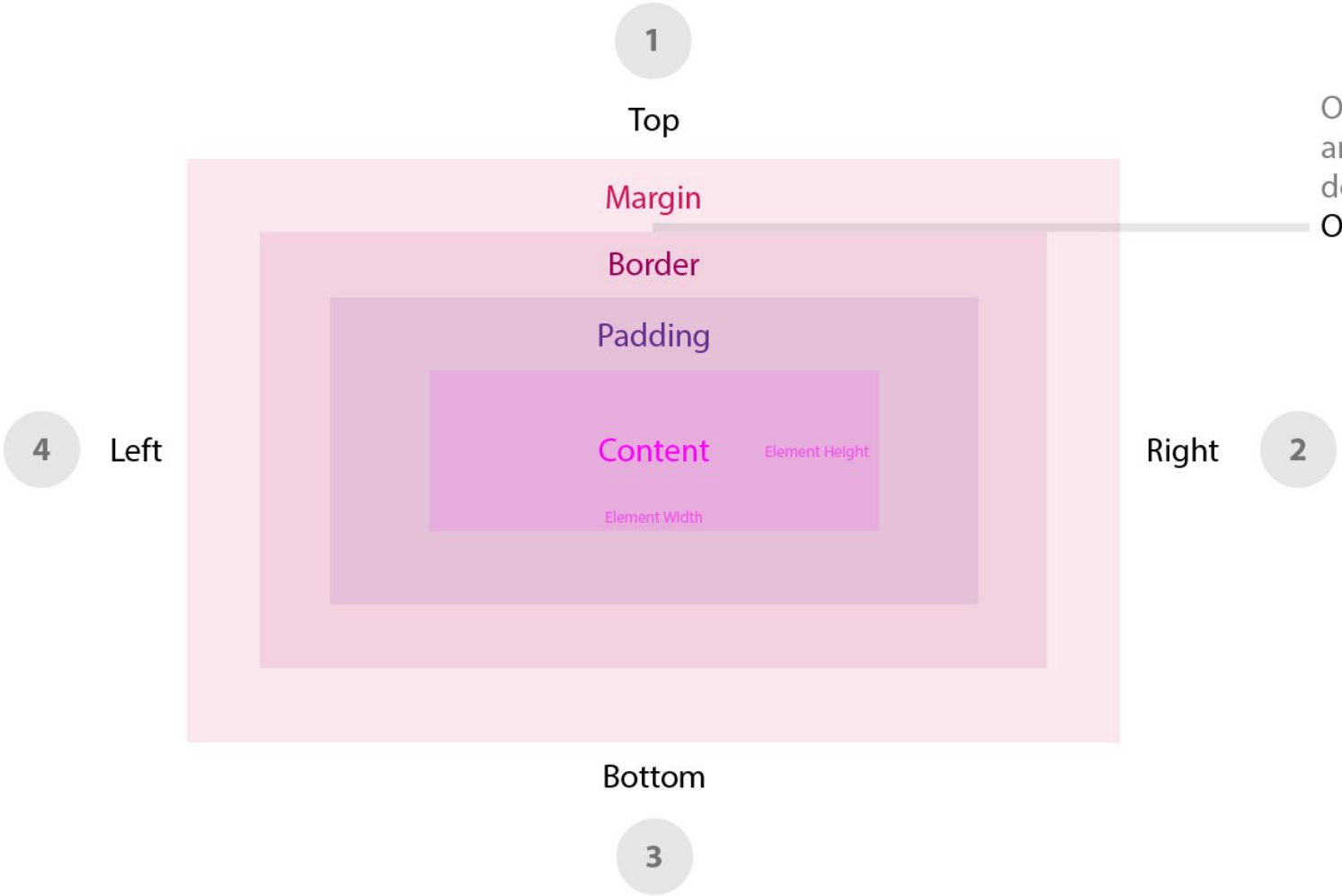
1. Content
   This is the text, graphics, or other elements inside of the element. The width and height applied to an element apply here.

2. Padding
   A space around the content part of the box. Note that it will maintain background-colour applied to the element. You can specify different padding thickness for the top, right, bottom, and left sides of the element.

3. Border
   Surrounding the content and padding is the border part of the box. Borders can have a variety of different styles, and can be coloured differently than the content/padding area. You can specify different border thickness, style, and colour for the top, right, bottom, and left sides of the element.

4. Outline
   The outline part of the box does not take up space (it won't push other elements away.) It appears as a secondary border outside of the first border, and is styled similarly. This is often used for the red glow surrounding a form field with an error (you wouldn't want that glow to push away the other fields like an added border might.) You can specify different outline thickness, style, and colour for the top, right, bottom, and left sides of the element.

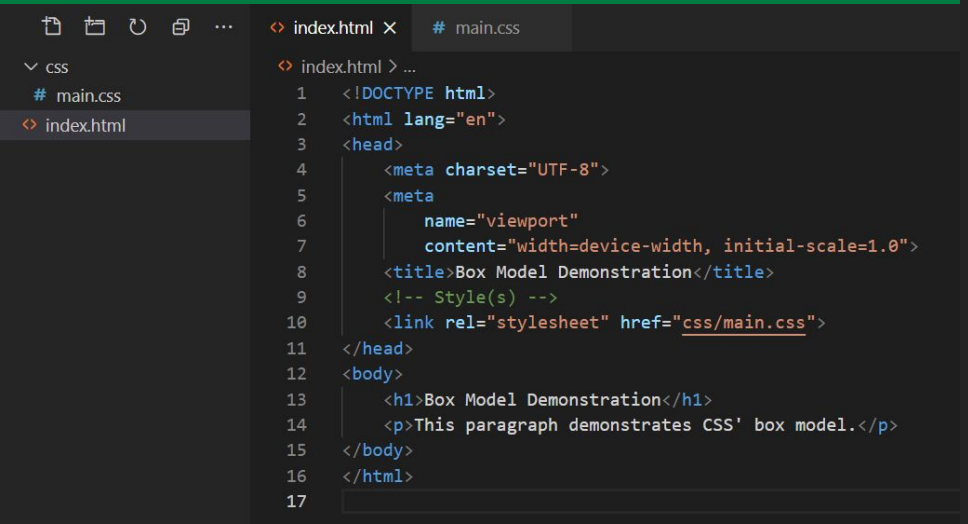
5. Margin
   A transparent space outside of the border of the element. Margins have no background-colour or visible representation, they are the space you'd like to maintain between this element and those that surround it. You can specify different margin thickness for the top, right, bottom, and left sides of the element.

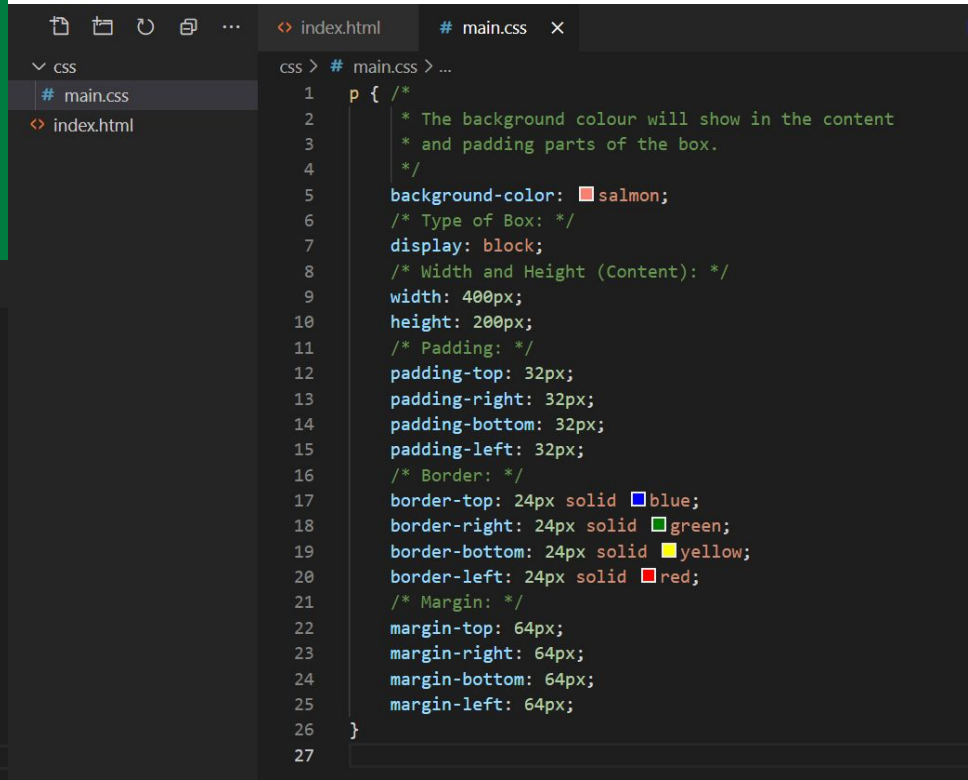


Outside of border there is an additional border that doesn't take up space:
Outline

Box Model Demonstration

Set up a basic website, and we can experiment with the box model! Try out the following code—feel free to make some changes and see what happens.

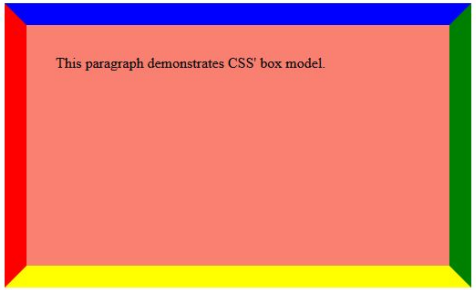




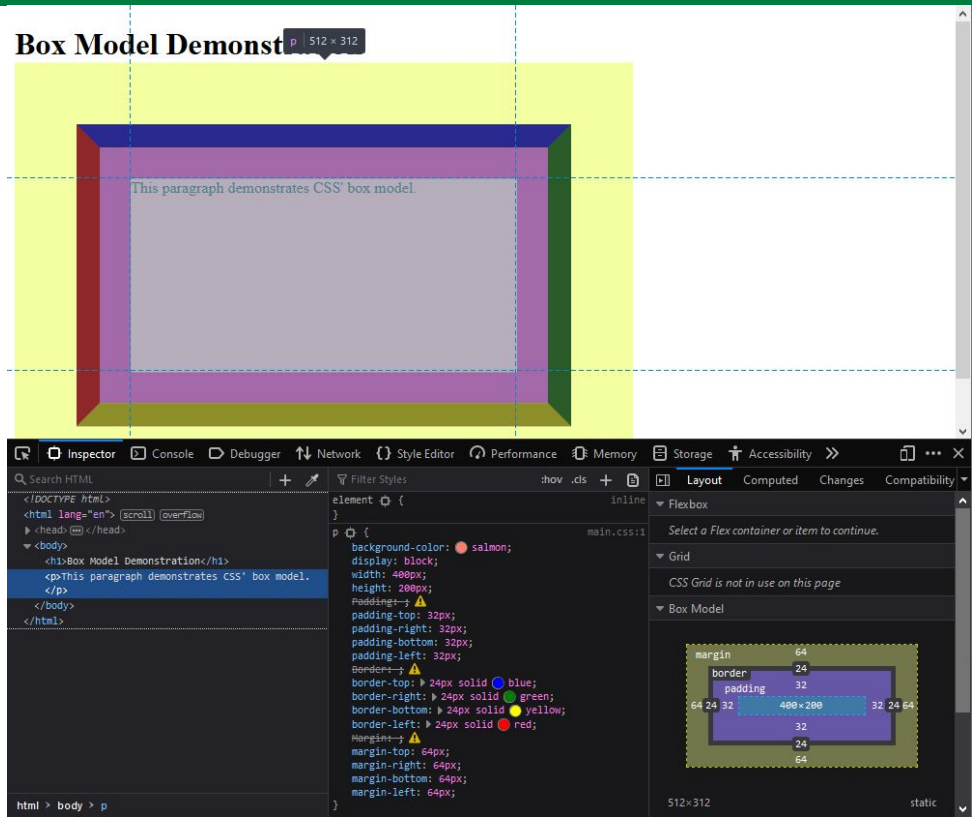
Breaking Down the Output

You should see something like the following…

Box Model Demonstration



Inspect this paragraph element, and observe the box model representation in your browser's dev tools. Most modern browsers offer a detailed CSS pane with a tab or option for the box model breakdown.



Shorthand Properties

It may seem like a lot, when adding something like padding values, like so…

```
padding-top: 32px;
padding-right: 32px;
padding-bottom: 32px;
padding-left: 32px;
```

In cases like this, there are often ways to set all these values in a single declaration via use of shorthand properties.

Shorthands may come in different shapes and sizes, but some of the ones that will come up most often are those that are related to the box model.

The padding shorthand property can accept 1, 2, 3, or 4 values. We can actually shorten the padding example to a single line, like so…

```
padding: 32px;
```

If you use 1, as we did above, it will simply apply that measure to all sides of the box.

```
padding: 32px 32px;
```

If using 2, the first represents the top/bottom, and the second represent the right/left.

Shorthand Properties (Continued)

padding: 32px 32px 32px;

If using 3 values: the first represents the top, the second represents the right/left, and the third represents the bottom.

padding: 32px 32px 32px 32px;

If using all four values: the first represents the top, the second represents the right, the third represents the bottom, and the fourth represents the left.

You'll notice this is the numbered order in the box model diagram displayed earlier! The shorthand follows a clockwise order starting from the top of the box.

Determining the Size of an Element

What is the total width of our box model demonstration? This is an important question.

We set our width to “400px,” so… it must be 400 pixels wide, right?

Nope. The total width of the element is 464 pixels. Where did that extra 64 pixels come from?

Padding counts toward the element width and height. It is important to note this, especially if you want your elements to line up properly.

If you find this inconvenient, you’re in luck! There is an alternative box model option for how the element sizing is handled. To include padding in your set width and height, simply replace the default **box-sizing: content-box;** declaration with **box-sizing: border-box;**. This is a common practice to avoid additional math during development.

Try applying this to the box model demonstration and observe the difference.

The Best of Both Worlds

There may be times where you want block elements to sit beside one another, or times where you want an inline element to have a set width and height.

Is there an in between; is there something that can help us bridge that gap when we get into such situations?

Absolutely. CSS affords us as an option for an element's display property: inline-block. Keep this in mind if you end up hitting a wall with either block or inline by themselves.

Recommended Reading

To learn more about the box model and how elements layout by default in a page:

- Meyer, E. A; Weyl, E. (October 2017). CSS: The Definitive Guide, 4th Edition. O'Reilly Media, Inc.
  - Chapter 7. Basic Visual Formatting
  - Chapter 8. Padding, Borders, Outlines, and Margins