Unit Three
CSS The Box Model in Depth

AIM Code School
Vanessa Kasun

Displaying Elements

Before we discuss the box model, we need to better understand how elements are laid out and displayed on a webpage.

Does anyone remember the difference between block and inline elements?! ANYONE?

Let’s Review...

Block & inline level elements

Every element has a default display property value.

We can change an elements display property value by selecting that elements in CSS and declaring a new **display** property value.

Block

Inline

Inline-block

None

inline-block

The inline block value will allow an element to behave as a block level element, accepting all box model properties. (we'll cover that soon). However, the element will be displayed inline with the other elements, and it will **not** begin on a new line by default.

★ One important distinction with inline-block elements is that they are not always touching, or displayed directly against one another. Usually a small space will exist between two inline-block elements. This space, though perhaps annoying, is normal. We'll discuss why this space exists and how to remove it in the next lesson.

Width & Height

By now we understand that every element has a default height and width value that is rendered by the browser. Depending on how an element is displayed, the default may be suitable for the desired layout. However, if an element is key to the layout of a page, you may need to be more specific with height and width property values.

Width

To get the default value of an element, it depends upon the element itself since each elements has a different default value.

Block Level Elements have a default width of 100%. Taking up the entire horizontal space available

Inline and inline-block elements expand and contract horizontally to accommodate their content. Inline-level elements cannot have a fixed size, thus the width and height properties are only relevant to non-inline elements. To set a specific width for a non-inline element, use the width property:

```
1   div {
2     width: 400px;
3   }
```

Height

The elements default height is **determined by its content.** Like width, it with expand and contrast but vertically, as needed to accommodate its content.

To set height for a non-inline element, use the height property:

```
1    div {
2      height: 100px;
3    }
```

Padding

- Padding is empty space that surrounds content.
- It is similar to margin, however it is not the space between the tags, it's the content that the tags contain.

```
1    div {
2      padding: 20px;
3    }
```

Margin

- Margin surrounds everything, and is similar to padding in that it is empty space.
- Margin should be thought of as empty space between tags themselves, and is used to push tags apart from one another on the screen

```
1    div {
2      margin: 20px;
3    }
```

Margin Extras

To set one value for the top and bottom and another value for the left and right sides of an element, specify two values: top and bottom first, then left and right. Here we are placing margins of 10 pixels on the top and bottom of a and margins of 20 pixels on the left and right:

```
1    div {
2      margin: 10px 20px;
3    }
```

Borders

- Borders fall between margin and padding, giving an outline around an element.
- The border property requires three values : width, style, and color.

```
1    div {
2      border: 6px solid #949599;
3    }
```

Box Sizing

- This CSS property sets how the total width and height of an element is calculated.

The box sizing property accepts three different values:

1. Border-box
2. Content-box
3. padding-box

Content- Box:

Gives you the default CSS box-sizing behavior. If you set an element's width to 100 pixels, then the element's content box will be 100 pixels wide, and the width of any border or padding will be added to the final rendered width, making the element wider than 100px.

Border-Box:

tells the browser to account for any border and padding in the values you specify for an element's width and height. If you set an element's width to 100 pixels, that 100 pixels will include any border or padding you added, and the content box will shrink to absorb that extra width. This typically makes it much easier to size elements.

Box Sizing cont'd

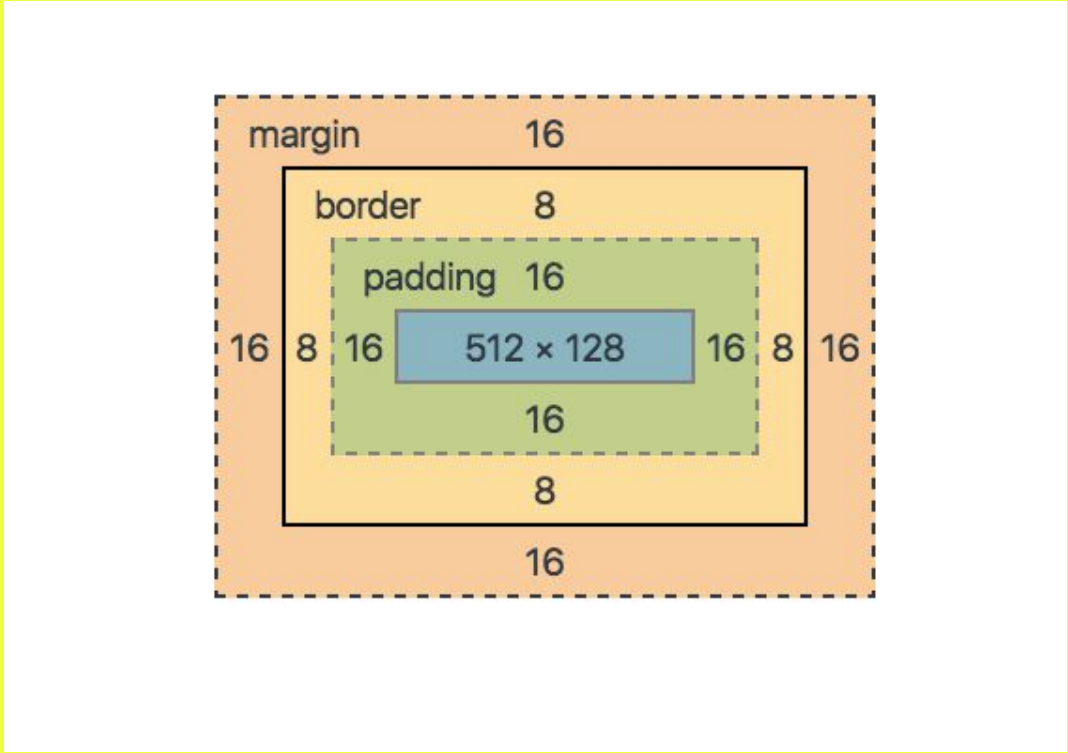
Padding-Box:

The padding-box value alters the box model by including any padding property values within the width and height of an element. When using the padding-box value, if an element has a width of 400 pixels and a padding of 20 pixels around every side, the actual width will remain 400 pixels. As any padding values increase, the content size within an element shrinks proportionately.

What exactly is the Box Model?

1. Every single element on a page is a rectangular box and may have width, height, padding, borders, and margins.

2. Every element on every page conforms to the box model, so it's incredibly important. Let's take a look at it, along with a few new CSS properties, to better understand what we are working with.



Coding according to the Box Model

The core of the box is defined by the width and height of an element, which may be determined by the display property, by the contents of the element, or by specified width and height properties. padding and then border expand the dimensions of the box outward from the element's width and height. Lastly, any margin we have specified will follow the border.

Each part of the box model corresponds to a CSS property: width, height, padding, border, and margin

Let's look at an example...

```
1   div {
2     border: 6px solid #949599;
3     height: 100px;
4     margin: 20px;
5     padding: 20px;
6     width: 400px;
7   }
```

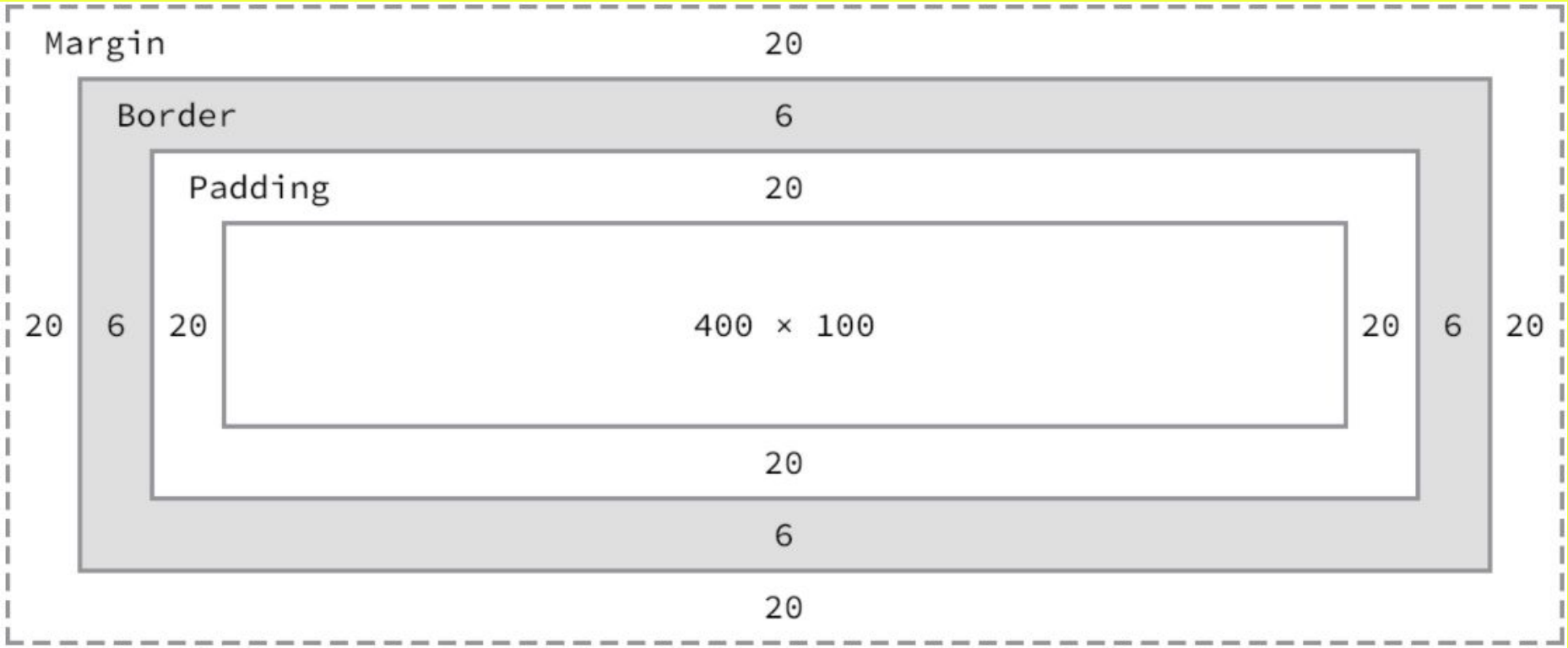
Look at these properties...

According to the box model, the total width of an element can be calculated using the following formula...

margin-right + border-right + padding-right + width + padding-left + border-left + margin-left

Similar to finding the width, we can calculate the total height of the box model using the following formula...

margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom



Using the formulas from the previous slide, we can find the total height & width of our example code
Width: 492px = 20px + 6px + 20px + 400px + 20px + 6ox + 20px
Height: 192px = 20px + 6px + 20px + 100px + 20px + 6px + 20px

So, to clarify

We set a width property value of 400px, but the actual width of our element is 429px. By default the box model is additive. Because it is additive; to determine the actual size of the box we need to take in account padding, borders, and margins for all four sides of our box.

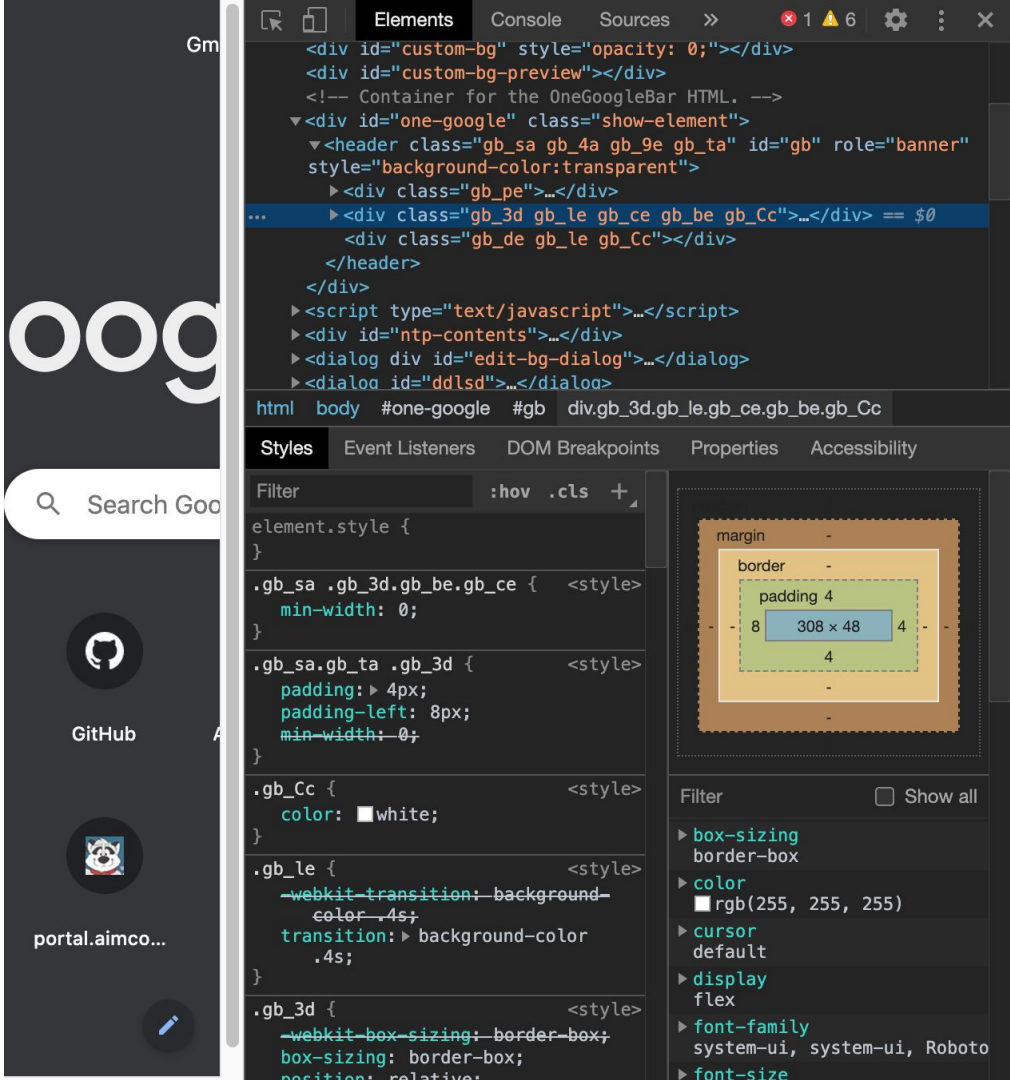
This is hands down one of the more confusing parts of HTML and CSS, but with practice and a lot of trial and error, through problem solving and thinking logically, it will become second nature.

Remember…. Obtaining challenging knowledge is fun and yes challenging, those who work hard to understand WILL understand and they WILL succeed!

If this is confusing to you in any way, ASK QUESTIONS!!!!!! Do not let these concepts discourage you from continuing. I am here to help guide you and be of assistance through this learning journey!

Debugging Developer Tools

- Web development tools, often referred to as devtools", provide a way for developers to test and debug their code.
- These tools help to inspect HTML elements on a webpage and see what css properties and values are being applied to it.
- Web development tools allow you to work with an array of different web technologies such as HTML, CSS, DOM, JavaScript, and other components that are rendered by the browser.



Accessing Developer Tools (Chrome)

That's all for now folks!

Let's get back to coding.
Open your style conference files lets work on some CSS!