CSS Introduction

<u>Before we begin, there will be assigned roles for this assignment. Each of the roles will be responsible for helping the group get through the document</u>

Switch up the roles! They should not be the same from the last session

Roles

Reference Guide: Role specific for person who knows HTML*: When the group is stuck on a question, your job is to guide them towards the answer without outright telling them.

Manager: Keeps the team on task, makes sure that the conversation doesn't go too off topic

Quality Control: Makes sure there is an accurate record of correct/final answer for the team and that all other members in the group understand the right answer

Timekeeper: Makes sure that the activities are being done in a timely manner and vocalizes to the instructor if more time is needed

Speaker: Reads through the packet out loud for the group so everyone can follow along. Communicates with the facilitator (in this case Sean) in case the group has a question or is confused about a part of the assignment

Reference Guide:

Manager:

Quality Control:

Timekeeper:

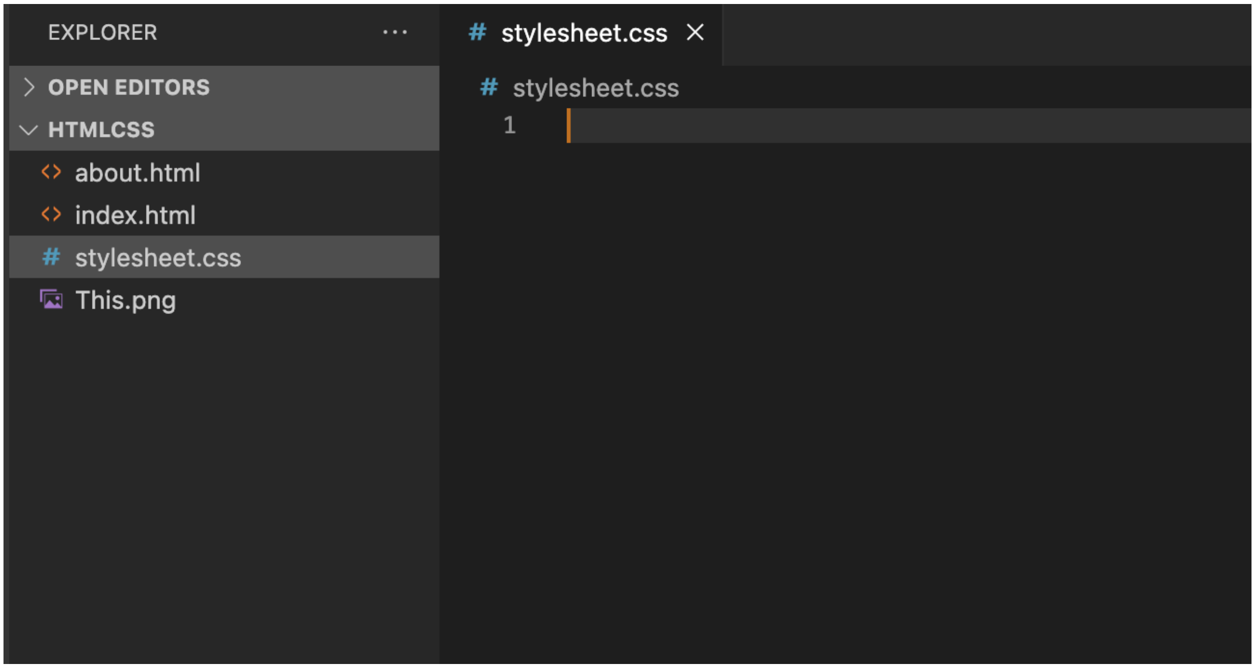
Speaker:

Now to begin the assignment

What is CSS? What is a Stylesheet?

CSS, or Cascading Style Sheets, is a markup language that works together with HTML to stylize it, but has its own syntax

Similar to creating a new HTML file, we can create a new style sheet by adding a new file into the same folder as our HTML, giving it a name, and giving it the file type of .css



Once we have this file created, we’re going to connect our stylesheet together with our HTML document

To do this we’re going to go into the head of our HTML document and input a <link></link> element, with an attribute for rel and href

rel = stands for relationship, it points out to the code editor that this link is for a stylesheet

href = the name of your stylesheet. Make sure if your stylesheet is in a different folder (or subfolder) than your HTML to add in the entire path to your stylesheet file

It should end up looking like this

<link rel="stylesheet" href="THENAMEOFYOURFILE.css">

We’re going to test this connection by inputting some CSS

But first, we have to learn about the syntax of CSS

Syntax of CSS

CSS is made up a series of **selectors**, which each contain an **opening curly bracket**, a **closing curly bracket**, a **property**, a **colon**, a **value,** and a **semi-colon**

OK but what does that mean?

Take an opportunity to fill out what you THINK the syntax should look like – talk through this as a group and come to a decision of what the syntax looks like (and why you think so). Feel free to use the white board.

For this example, make all the text in the paragraph elements in the HTML document green

{	p	:	
;	}	color	green

____ ____

____ ____ ____ ____

____

Now that you had a chance to guess the syntax of this, take the tape off this section. This is and label the parts of this CSS

```
p {
color: green;
}
```

Before trying to define which part is which, let’s lay down some definitions

Selector: points to the element in the HTML document we will be changing

Property: identifies what characteristics of the element we are going to change

Value: shows what about the element we’re going to change

What part of this do you think is the **selector**?

What part of this do you think is the **property**?

What part of this do you think is the **value?**

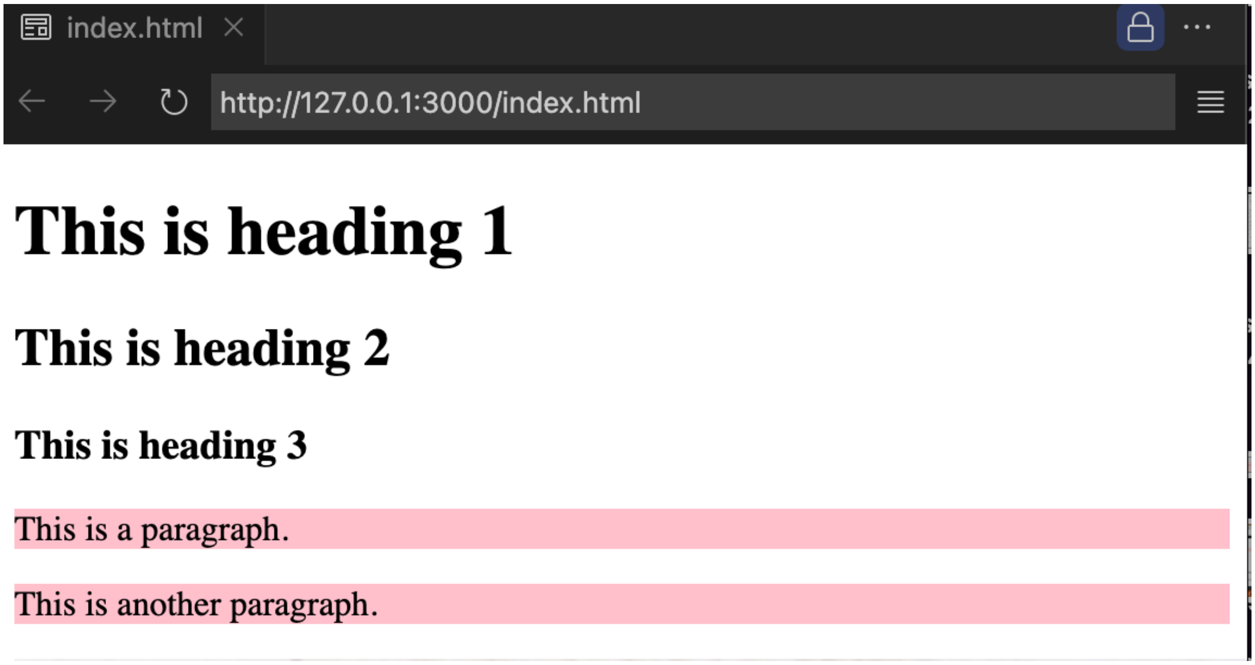
Which part is the opening **curly bracket?**

Which part is the **closing curly bracket?**

Now that we understand the syntax of CSS, we're able to learn more about the ways we can stylize our HTML documents

Go back to your CSS stylesheet- try to change the paragraph elements in your document so that the "background-color" is pink

When you preview your HTML document it should look like this



This is heading 1

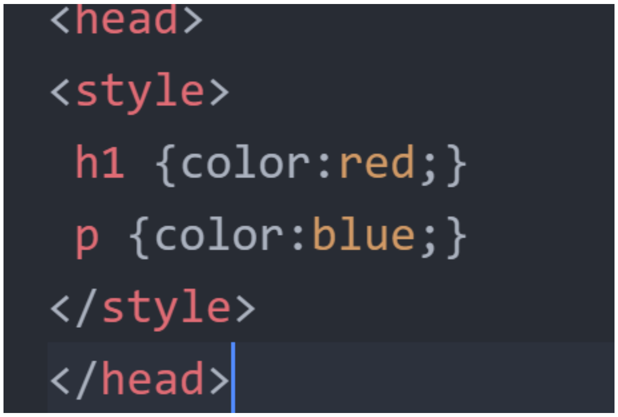
This is heading 2

This is heading 3

This is a paragraph.

This is another paragraph.

There's also a third way of doing this that involves creating a style element in the head of our HTML document



```
<head>
<style>
 h1 {color:red;}
 p {color:blue;}
</style>
</head>
```

Why do you think though linking our HTML document to a .css file is the most popular way to stylize our HTML document? What are the benefits (think like you had to stylize a whole website with multiple pages). Try to name at least two benefits.

Selectors

As I mentioned before, the purpose of a selector is to point to the element in the HTML document we will be changing. It is possible however that we can customize how it is we use the selector to point to different HTML elements

We've already discussed **element selectors**, which just points to one element

```
h1 {
color:blue;
}
```

It’s possible to also use the **multiple elements selector** if you want to give them all the same effect

```
a, h2, p {
text-align: center;
}
```

What if though you wanted to affect only SOME of the elements in the document (without also changing all of the same type of that element)?

That’s where **class** and **ID** come in

The **class selector** applies a specific style to multiple individual element

The **ID selector** applies a specific and unique style to one singular element

We are able to mark a class selector by giving it a period and a unique name (which will then be used in the attributes of the affected element)

```
.exampleclass {
color:orange;
font-size: 3em;
}
```

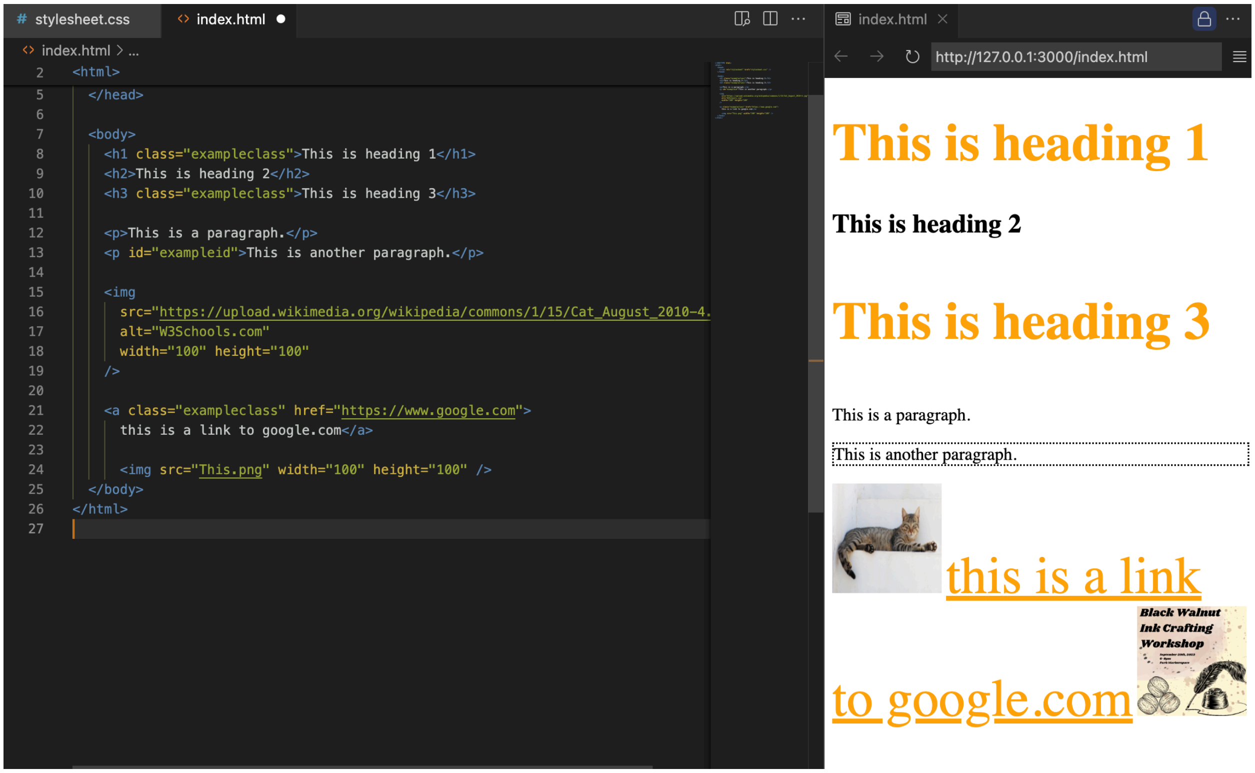
The same applies to ID, except it uses a hashtag

```
#exampleid {
Border:dotted;
}
```

Please be aware that your unique name is case sensitive (meaning exampleid and EXAMPLEID are not the same, capitalization and lowercasing matters)

Both of the above samples would be put into our CSS file

Inside of our HTML file, we would mark all of the elements we want to be affected by the class selector and the single one we want to be affected by ID. We would do this by giving them an attribute of either class or id


stylesheet.css
index.html
index.html > ...
index.html
http://127.0.0.1:3000/index.html
Black Walnut Ink Crafting Workshop

```
<html>
  </head>

  <body>
    <h1 class="exampleclass">This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3 class="exampleclass">This is heading 3</h3>

    <p>This is a paragraph.</p>
    <p id="exampleid">This is another paragraph.</p>

    <img
      src="https://upload.wikimedia.org/wikipedia/commons/1/15/Cat_August_2010-4.
      alt="W3Schools.com"
      width="100" height="100"
    />

    <a class="exampleclass" href="https://www.google.com">
      this is a link to google.com</a>

      <img src="This.png" width="100" height="100" />
  </body>
</html>
```

This is heading 1

This is heading 2

This is heading 3

This is a paragraph.

This is another paragraph.

this is a link to google.com

In your HTML Document and CSS Stylesheet, add a class which makes two of your paragraph elements and one of your headings have a blue colored text with an orange background

Properties and Values

Now that we’ve had a chance to understand how to create CSS, let’s dive into how we can use CSS further to stylize our HTML document

Font

We can change certain aspects of our font by using the font-size and font-family properties

The font-size

```
p {
font-size: 2 rem
}
```

The typical values you can use to affect the font-size are px or em px you will probably be most familiar with, since it’s the same sizing you would use on a Microsoft Word Document or a Google Doc. 16px is the default font size.

By using em, this allows users to resize the text in the browser menu. This can be helpful for those who have certain accessibility requirements and need either a bigger or smaller font.

1em is the default font size so 16px = 1em

The best practice to use however is rem

rem (also called root em) is used because em has a compounding problem, where it adds on the font size from the parent element on top of the child element. Rem fixes this by having it relative to just the element.

Try seeing this in action – make the font size for the body 1em and the paragraph element 2em

Then try making just the paragraph 2rem

What’s the difference you can, see? (make sure also when you type it out, don’t put a space between the number and (r)em

For sizing things in your HTML document, use rem but for things not tied to the font (which we will go over later) feel free to use px

Font-family

```
p {
font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

What you’ll notice in the example above is when we’re picking out the font we want to use for our HTML document (in this case it’s Verdana), we’re including not just the font value we want to use but a whole list of fonts

The other font values besides Verdana that are separated by commas are our font alternatives. Not all computers have the same font library – while you may Verdana on yours, someone using either a different operating system or an older model may not.

By including a list, the browser will select the first font in the list that is installed or that can be downloaded

You should always include at least one generic family name in a font-family list, since there's no guarantee that any given font is available (in the above case it's sans-serif)

What are two other generic family names besides sans-serif (feel free to google this)?

Color

We've used color as an example many times so far, but now we're going to jump further into color and background color

Color

When inputting the value for color, most of the times we've used the name of the color but we can get even more specific

Use the color picker available here https://www.w3schools.com/colors/colors_picker.asp

In your CSS Stylesheet, use the color picker to find a hex color code and turn one of your paragraph elements a specific color

An example of a hexcolor might be

```
p {
color: #33cc33
}
```

For rgb (which stands for red,blue,green) pick a value from 0 to 255 and see what you get. Each of the values for rgb is determining what amount of these colors are "mixed" together to create a custom color. Share with the group what color you get.

Example:

```
p {
color: rgb(40, 50,100)
}
```

Background-color

```
p {
background-color: rgb(100, 11 ,200)
}
```

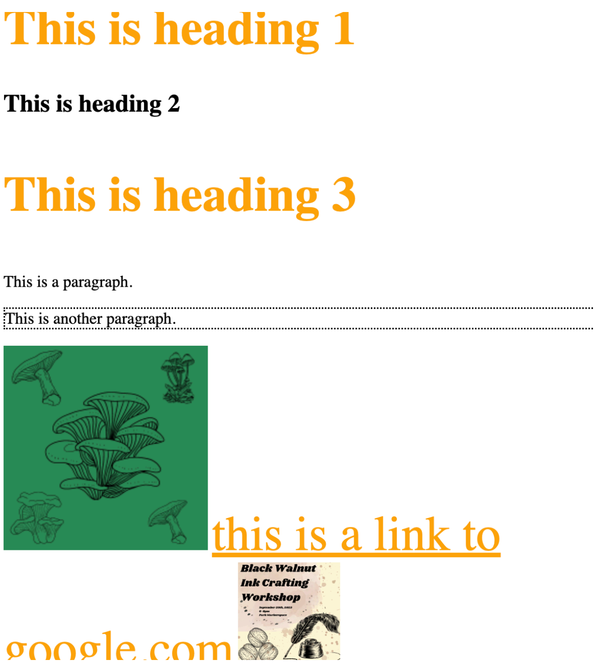
This impacts the color behind the content. We’ve often used this with elements that are text based, but it’s also possible to use it with other elements as well

In your HTML document, add an image with an absolute link to this URL
https://skeenan520.github.io/coding-sandbox/images/mushroom.png

Add any background color to this image (which has a transparent background) and share with the group

Box Model

If your doc looks anything like mine, while playing around with all of these features, you may have noticed that your HTML document looks a bit squished together



This is heading 1

This is heading 2

This is heading 3

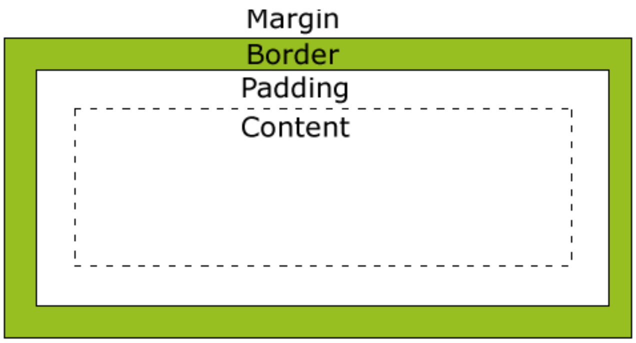
This is a paragraph.

This is another paragraph.

this is a link to google.com

Yikes!

Fortunately, using the box model, we're able to stretch out the spacing in our document a little better



Inside of our HTML document there are invisible boxes that surround all of our content- we are able to shape these boxes in order to build a more customized CSS page

It's important for us to better define what these aspects of the box model mean

As defined by the W3Schools page on Box Models
https://www.w3schools.com/css/css_boxmodel.asp

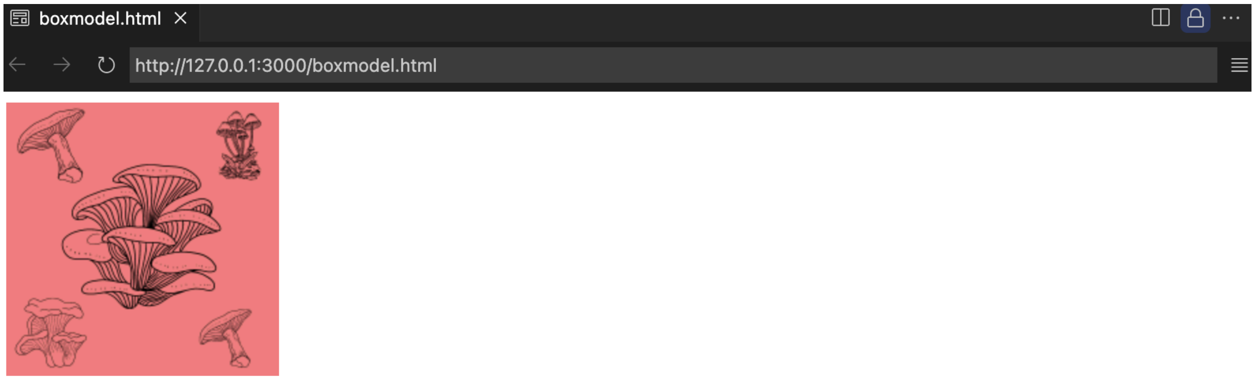
- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The best way to play around with the box model in an easy way to understand would be to

-Create a new HTML file
-Add an image element with the mushroom image
<img src=https://skeenan520.github.io/coding-sandbox/images/mushroom.png >
-Create a new CSS stylepage and connect the new HTML and CSS file together
-In your new CSS stylepage, give the mushroom image a background color

From there, this allows us to see the invisible borders of the box model a little more easily

Right now, our image is in the upper left hand corner

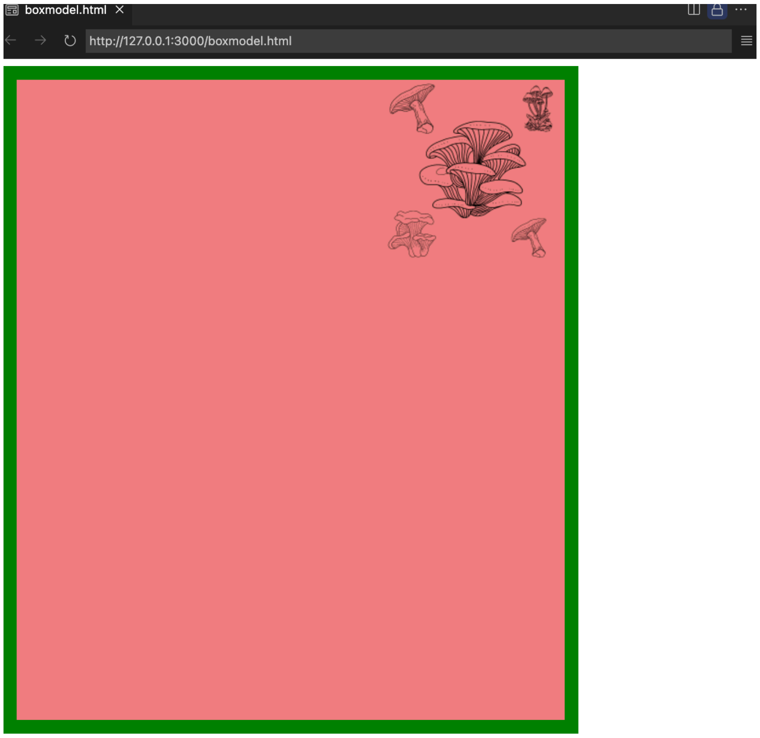


Let’s try to get a better understanding of these borders

Add to your CSS

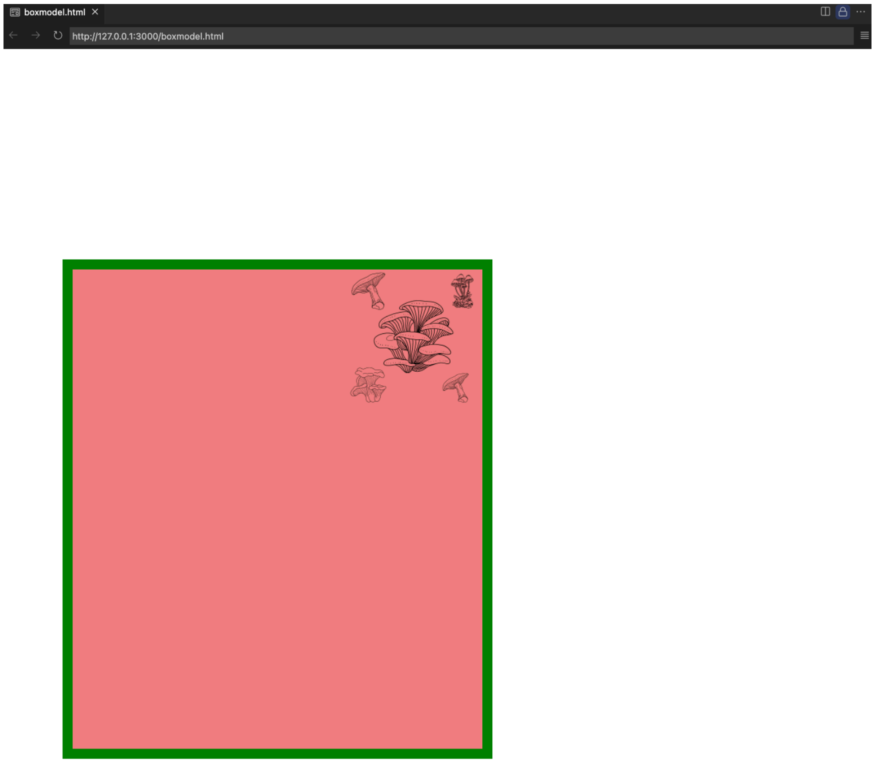
```
img {
border: 15px solid green;
padding-bottom: 50px;
padding-left:100px;
}
```

What changes do you notice?



How about when I add in

```
img {
border: 15px solid green;
padding-bottom: 400px;
padding-left: 500px;
margin: 300px 500px 30px 80px;
}
```



You’ll notice in the examples I wrote out padding using specific properties like padding-left and padding-bottom versus with margins I wrote out all the values for each of the sides.

Both are portraying information about the margins and the padding, just in different ways

When writing out how many px you want the sides to be, you can either call them out individually (example: padding-top, margin-bottom) or use the shorthand values (example: padding 25px 100px 50px 100px)

The shorthand goes in clockwise manner, meaning that the first number represents padding-top, the second padding-right, the third padding-bottom, and the fourth padding-left

There are ways you can write the shorthand using just three, two, and one value
https://www.w3schools.com/css/css_margin.asp
https://www.w3schools.com/css/css_padding.asp

As for the border, you’ll notice we’re able to affect how thick the border is in pixels, give it a border style, and change its color

Try giving your border a dotted appearance, make it purple, and have the dots be 5px wide

To find border styles check here https://www.w3schools.com/css/css_border.asp

If you're confused, I recommend during our sandbox time later today you take a look at this interactive box model link here https://codepen.io/carolineartz/full/ogVXZj

Comments

Similarly to how you are able to make comments in HTML, you can also leave comments in your CSS stylesheet

```
/* This is a comment in CSS. */
```

CSS Critical Concepts

Before we move on to building up our own HTML and CSS websites, it's important that we also go over some rules to keep in mind some of the basic CSS rules

Cascading

This means that later rules that are added to a document override rule made earlier on

Meaning if my CSS document has at the beginning

```
p {
color: green;
}
```

and at the end

```
p {
color: yellow;
}
```

My text is going to be yellow

Specificity

This rule ties closely to our discussion of class and id selectors

This is why if you have a paragraph element with a class selector saying the element is purple and in your CSS document you say that all paragraph tags are green, that one specific paragraph element will be purple

Just like also in our Intro to HTML worksheet, if you use the style attribute for a particular element in the HTML document, it will override any changes in the CSS document.

Inheritance

This rule ties into what we’ve learned about parent and child relationships

Meaning if I say that

```
Body {
	Background-color: blue;
}
```

That means all of its child elements (in this case, everything on the webpage) will have a blue background

Final CSS/HTML Challenge:

Before you get started on your sandbox, I want everyone in the group to pick one HTML or CSS concept that wasn’t covered and take 5 minutes to research it. You can discuss topics such as….

CSS Display
HTML Tables
CSS Selectors that we didn’t go over
CSS max-width / max-height
HTML button element
CSS Opacity
CSS Position

Talk to the group about your findings (or questions that popped up when you researched it!) You don’t have to have a full understanding of the concept, just try to briefly explain how it works.