# **17.2 Lesson Plan: The HTML/CSS Box Model**

## **1. Instructor Do: Welcome and Objectives (5 min)**

📌 Your goal is to introduce students to the HTML/CSS box model and to help bring organization and structure to their HTML and CSS.

Welcome the class and introduce today's topic: the HTML/CSS box model.

Open up the [17-Week/01-Slides/17.2 HTML and CSS Box Model Slides](https://docs.google.com/presentation/d/1blcIGbK39lUuYEzLoVLX0OynGNZOy9yWJp0U7pGN4zQ/) for the day.

Put today's lesson in context for students.

**~~Say,~~** ~~"Today, you'll build a style guide for the U.S. Postal Service (USPS)."~~

~~We'll apply the concepts discussed in class today to create a basic webpage for a USPS style guide. While it might seem simple, there is plenty of work to be done!~~

~~Review today's learning objectives.~~

**~~Review concepts from the last class.~~**

**~~Instructor note:~~** ~~Students who are new to programming will benefit from frequent reviews of what we have covered. Take the time to go over concepts as part of the post-activity recap.~~

~~Before you move on to the first lecture of the day, review a couple of concepts from last class and take questions to be sure that the fundamentals of HTML and CSS are sinking in for students.~~

**Ask:**

**What does HTML stand for? How is it used?**

*Answer:* Hypertext Markup Language. It is used to create tags that provide the structure for the web.

**What does CSS stand for? How is it used?**

*Answer:* Cascading Style Sheets. CSS is used to style HTML tags.

**How do HTML and CSS work together?**

*Answer:* HTML provides the structure of the web. CSS makes it look and feel good by making it visually appealing as opposed to barebones HTML elements.

**Front-End Development Resources**

Tell students a bit about the value of development resources. Development resources are a big part of how beginners learn web development. Reading tutorials and documentation is key to becoming a strong developer. During front-end development projects (or any coding project), your students will inevitably encounter problems. Researching and learning from problems will make your students great front-end developers.

**TAs,** share the following resources with students via Slack:

* [W3Schools - CSS Box Model Tutorial](https://www.youtube.com/watch?v=lPm8lK4C7nc)
* [CSS Tricks - The Difference Between IDs and Classes](https://css-tricks.com/the-difference-between-id-and-class/)
* [Stack Exchange - What Is the Difference Between Section and Div?](https://stackoverflow.com/questions/6939864/what-is-the-difference-between-section-and-div)
* [W3Schools - HTML Semantic Elements](https://www.w3schools.com/html/html5_semantic_elements.asp)

Take any questions students may have about HTML and CSS basics, then move on to the first lecture.

## **SLIDE 14**

## **2. Instructor Do: Targeting With Classes and IDs (5 min)**

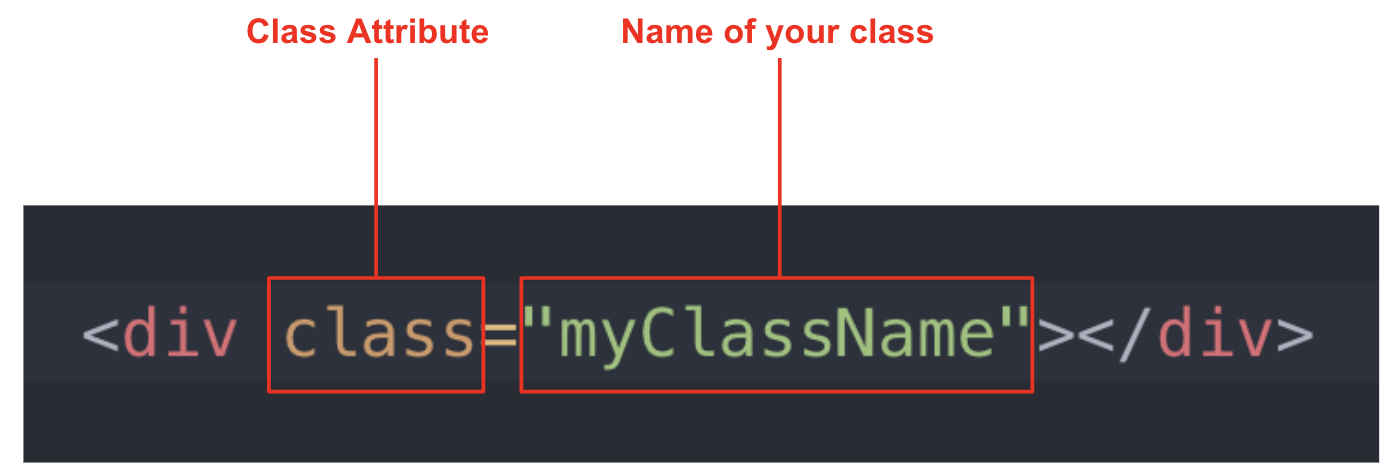
📌 The goal of this section is to introduce students to classes and IDs in CSS. Students will learn the differences between the two and how they are targeted. This section supports learning objective No. 1: apply CSS to target HTML tags with IDs and classes.

To style containers that we create in our HTML, we use two different CSS attributes, the class and ID attributes. Attributes are used to target unique HTML elements, while classes make our CSS reusable and target multiple HTML elements that need the same style.

## **SLIDE 15**

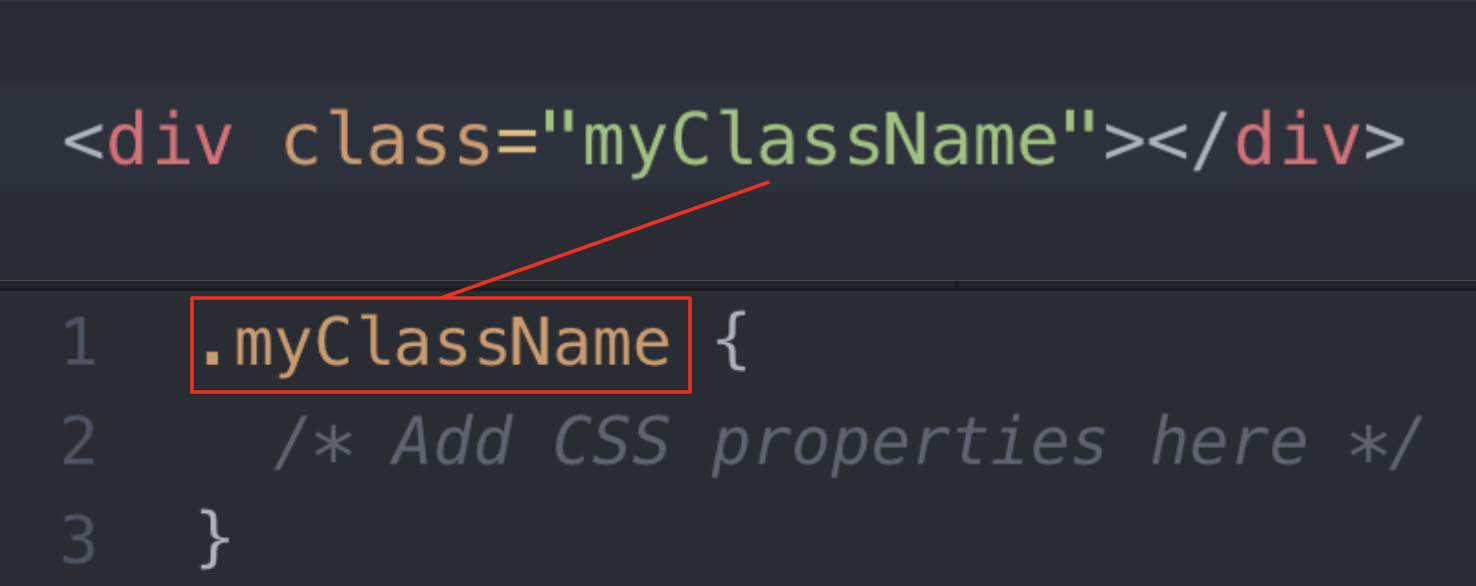
### **CSS Classes**

**Class attribute:** A class is an HTML attribute that is used to target multiple elements across the same page using CSS. You specify a class on an HTML element by typing "class" followed by the equals sign with the name of your class between the quotation marks.



Classes are targeted using a dot (.) in a CSS file. After the dot, type out the name of your class to style an HTML element.

🔑 **Key tech point:** In the examples above and below, you might have noticed the caps at the start of each word. This is known as camel casing, which makes code more legible. It is a common naming convention in web development.



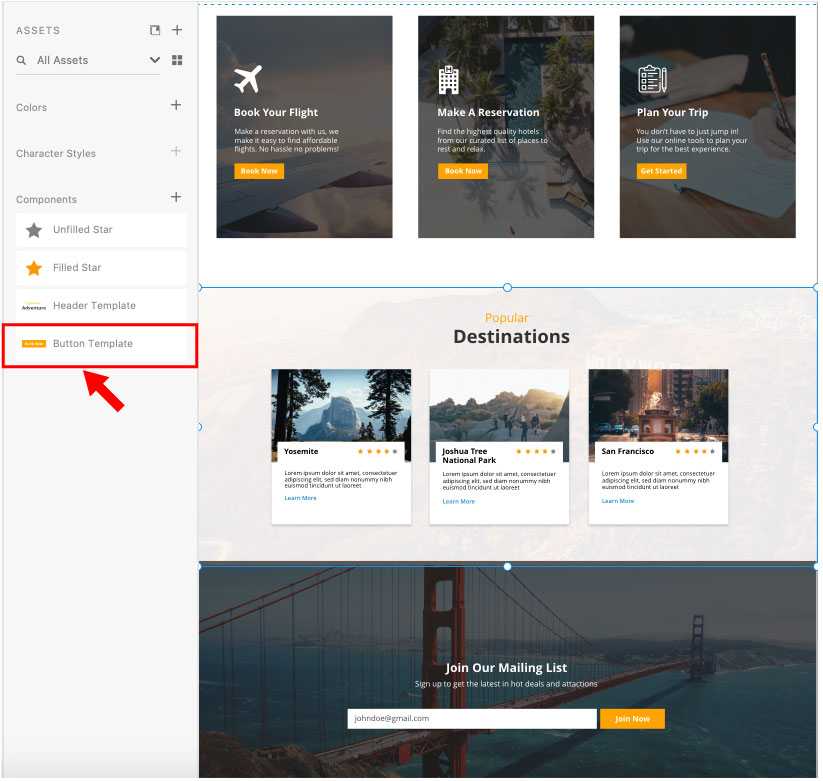
## **SLIDE 16**

Think of classes like components in Adobe XD. You create a component once and then reuse it throughout the design. The same concept applies to CSS with UI elements.

For example, consider the button component in the screenshot (Slide 14).

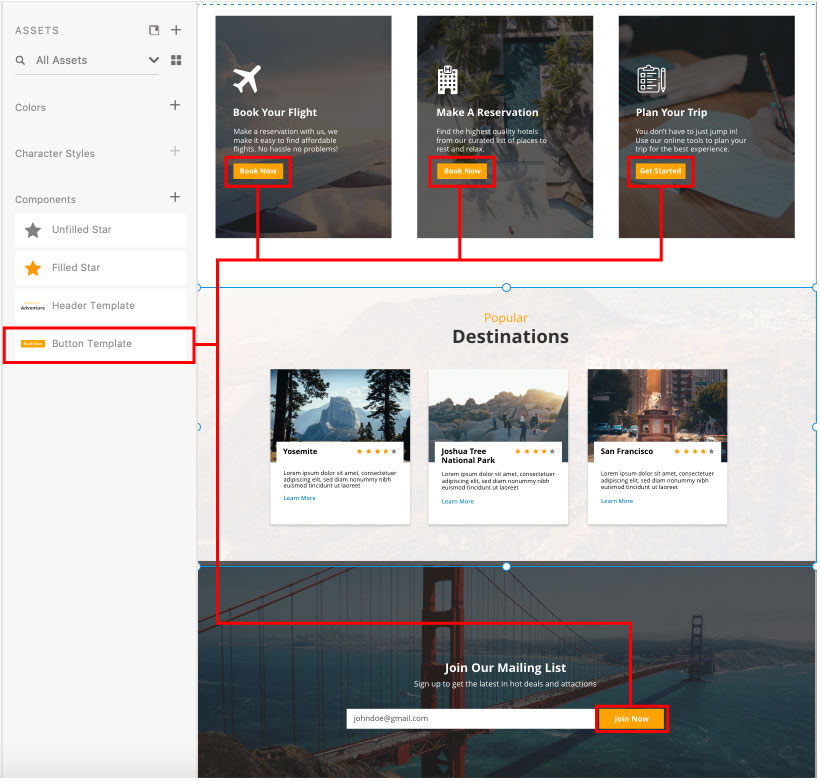
## **SLIDE 17**

**Question:** How many times is it reused on the page (also Slide 14)?



*Answer:* This component was used four separate times on the page.

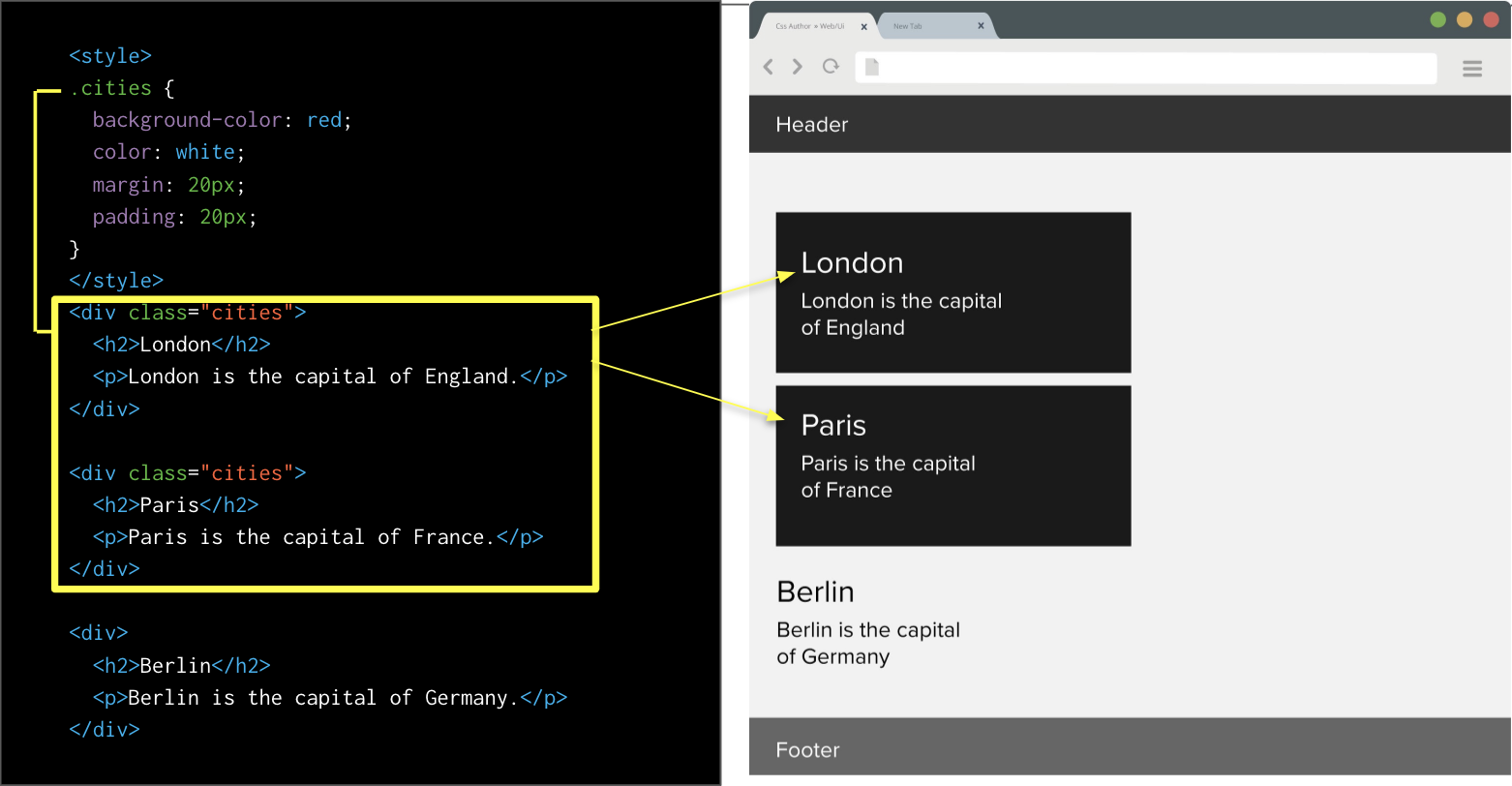
Imagine you are coding this page. Would you code each button individually or just once by using a class?



## **SLIDE 19**

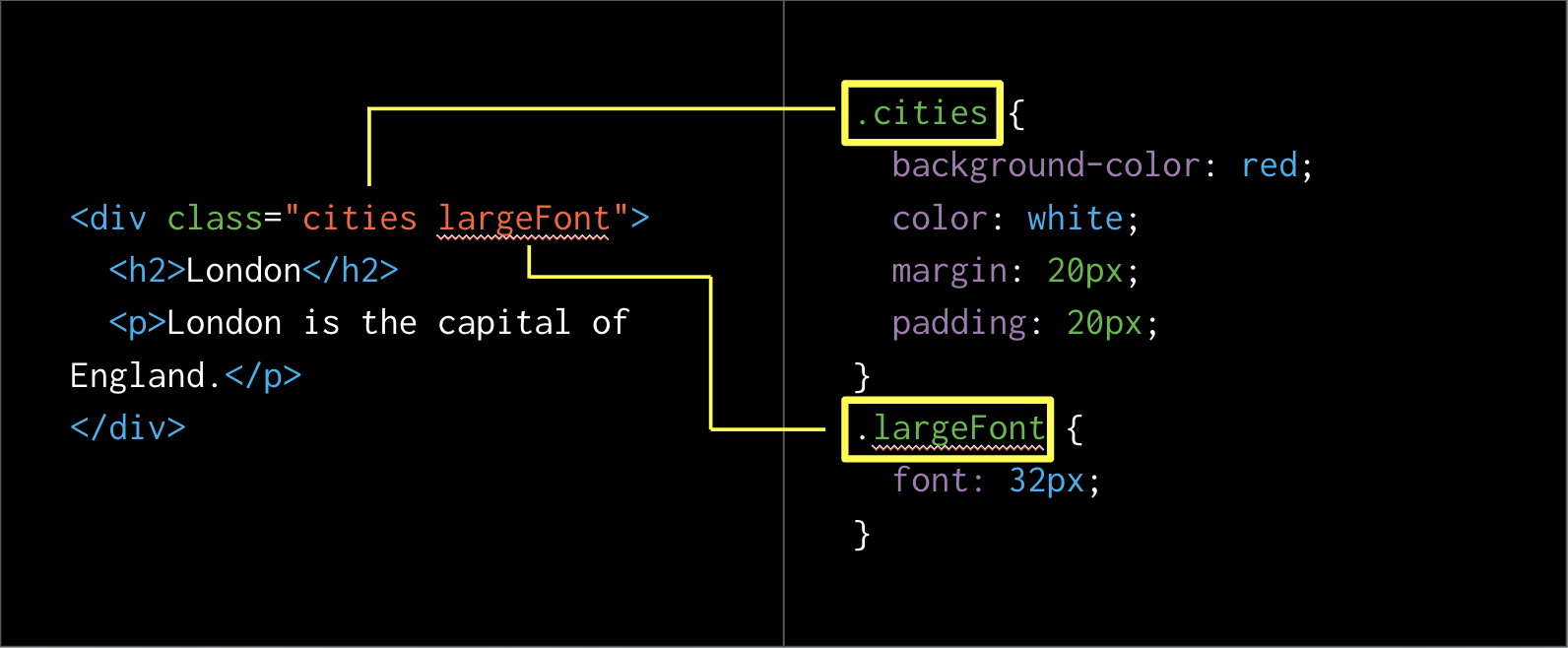
Classes are used to target multiple elements across the same page.

This means that you can reuse a class to apply the same style to all particular elements of a page.



In the picture above, we have the CSS selector of .cities styling two HTML elements with a background-color, font-size, and spacing with margin and padding.

HTML elements can have multiple classes. You can add two classes to an HTML element by simply adding another name for your class after your first.



## **SLIDE 20**

## **SLIDE 21**

**Key takeaways for classes:**

* Classes are not unique. Classes are designed to be reused just like components in Figma/Adobe XD.
* You can use the same class on multiple elements.
* You can use multiple classes on the same element.

## **SLIDE 23**

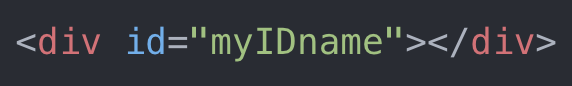
### **CSS IDs**

**ID attribute:** ID stands for an identifier. An ID is an HTML attribute used to create unique instances of a CSS selector. IDs are *unique* to a page, so you can only have one ID per page.

You should not have two IDs that are named the same. Identifiers are very useful when writing jQuery code. IDs will be the primary way that we target HTML elements in jQuery.

## **SLIDE 24**

IDs are HTML attributes that can be added to any HTML element by typing "id" followed by the equals sign and quotations containing the name of your ID.



## **SLIDE 25**

With CSS, you target IDs with a hash (#) before the name of the selector.



## **SLIDE 26**

**Key takeaways for IDs:**

* An ID is a unique instance of an HTML element.
* Each element can have only one ID.
* Each page can have only one element with that ID.

💎 **Designer Insight:** Help students appreciate the relevance of knowing these tags by relating them to work you have done with front-end developers.

Take questions to be sure these concepts are clear, then move onto the next tags.

## **SLIDE 28**

## **3. Instructor Do: HTML Div and Section Tags (5 min)**

📌 The goal of this section is to introduce students to two new tags: the div and section tags. These are the basic building blocks for all layouts, so be sure to introduce the differences between the two. This section supports learning objective 3: code the CSS box model by manipulating the height, width, margin, and padding to create containers for a style guide.

* The section tag needs no explanation, as the name says it all. This tag is used to build specific sections of a webpage.
  + Generally, these containers define the boundaries of your website, and divs are positioned inside them to create content that users interact with.
  + Section tags can be used to define sections in a document, such as footers, hero images, or sections that showcase features or products specifically.

## **SLIDE 29**

## **SLIDE 30**

## **SLIDE 31**

* Div stands for division. Divs are used to create content meant to be positioned inside a container. The div tag is used to group block and inline elements to be formatted with CSS.
* Think of a sign-in form. The form's container is probably a div positioned inside a section.
* Knowing when to use a div or section tag helps you write semantic code—HTML that introduces meaning to the web page rather than just presentation. For example, a p tag indicates that the content within is a paragraph.

## **SLIDE 32**

This makes your code easier to understand for yourself and your fellow developers by labeling HTML elements with tags that fit what they are.

## **SLIDE 33**

## **SLIDE 34 – form div**

**SLIDE 35**

**SLIDE 36**

You are now going to move on to a demo of Chrome's Web Inspector. This is a natural next step in the lesson because it will show students how to apply box model properties, which will set them up for success by showing students how to visually test out CSS properties.

## **SLIDE 38**

## **4. Instructor Do: Demo - Chrome's Web Inspector (5 min)**

📌 Your goal for this demo is to help students feel empowered to tackle building containers and structure for their USPS style guides. This demo supports learning objective 2: employ Chrome's Web Inspector to experiment with CSS in the browser.

**Instructor note:** A common frustration among CSS beginners is that they want to see how it looks immediately after they have made changes, so the "additional step" of refreshing their browser can feel time-consuming.

***S~~ay,~~*** ~~"It's a drag to reload, but it's the only way—and all developers have to."~~

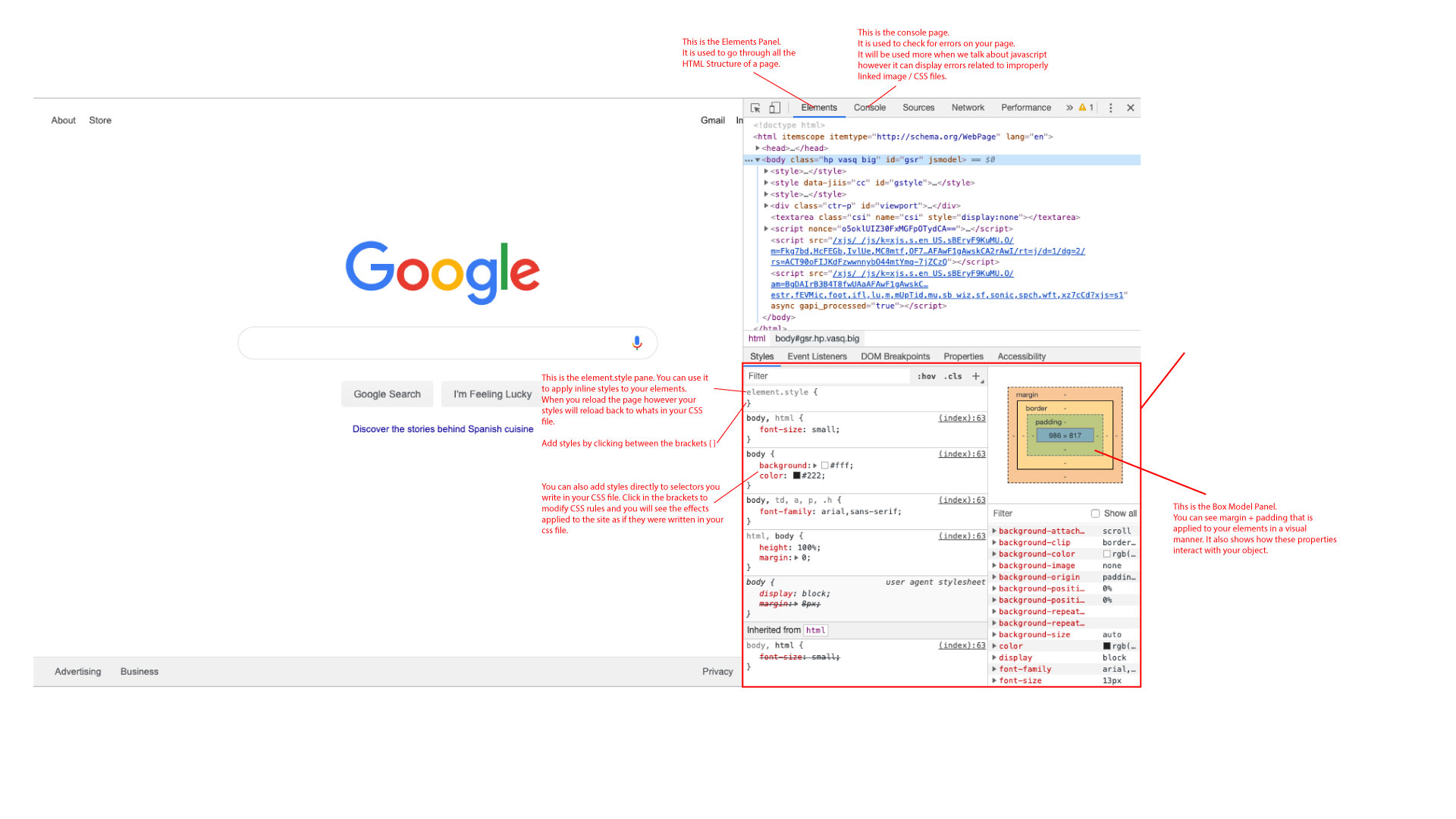
~~Before you begin, put this demo into context by explaining how front-end developers use the Web Inspector to build UIs visually. Earlier in the lesson, we discussed section and div tags. During this demo we'll show students how to use the Web Inspector to apply box model properties to our HTML elements—visually.~~

* Enter Chrome's Web Inspector tools. The Web Inspector can be used to view CSS visually, and it lets you play with your design.
* The Web Inspector is a game-changer. It allows you to see what you are doing when you apply specific CSS properties.

In Chrome, to view the Web Inspector, you can do a couple of things:

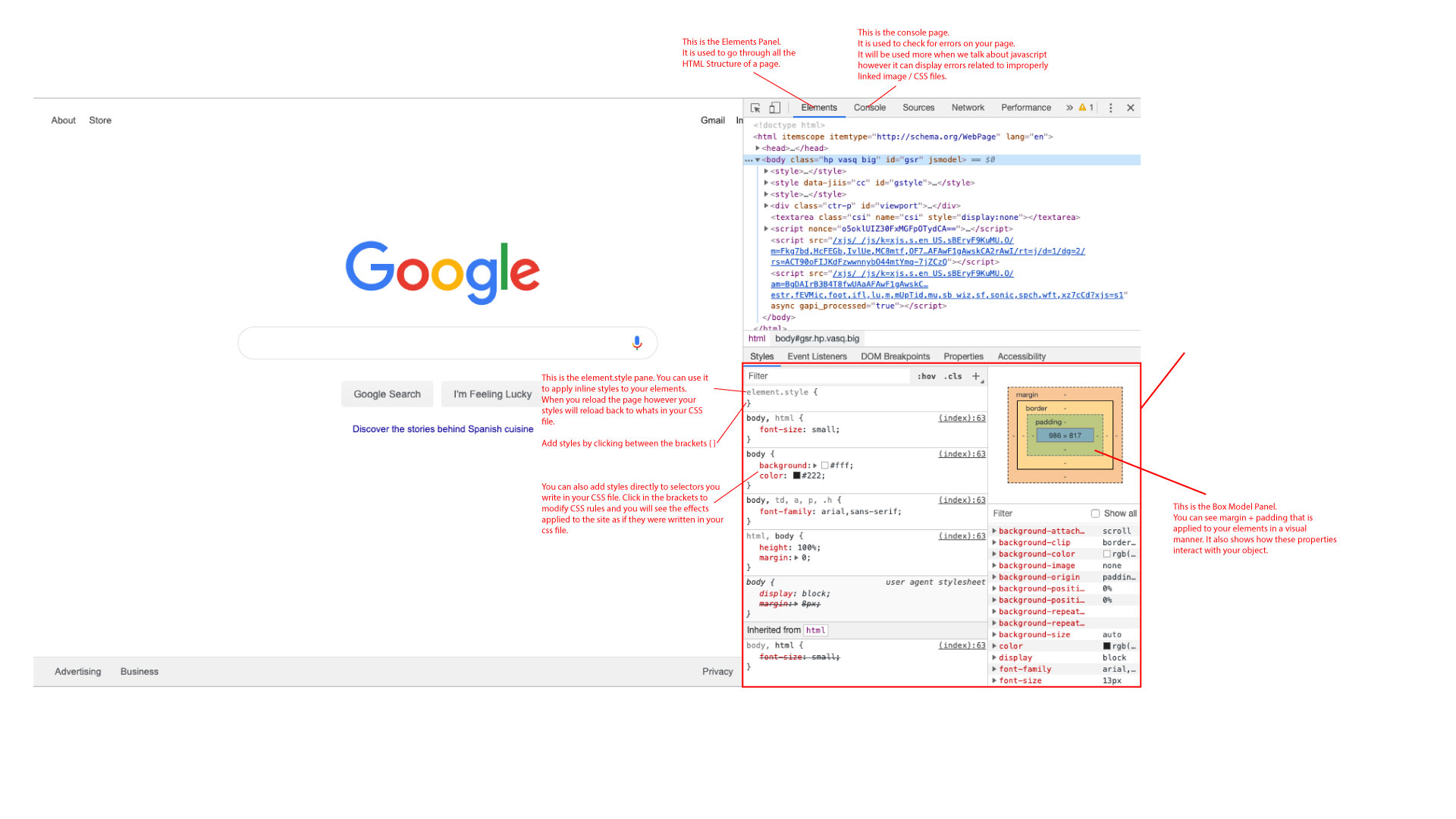
* Right-click on an element and click inspector.
* Use the hotkey (Mac: CMD + Option + i; Windows: Shift + ctrl + i).

Once you have the Inspector open, you will see the following interface:



**Walk through the following annotated Web Inspector image:**

* Open webInspecter\_annotated.jpg in front of the class.



**Instructions:**

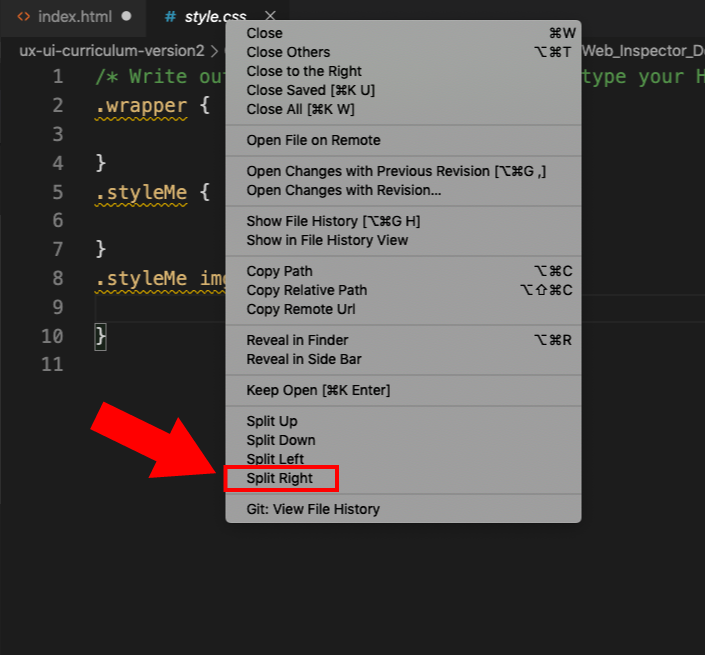
1. This is the elements panel. It is used to go through all the HTML structure of a page.
2. This is the console panel. It is used to check for errors on your page. It will be used more when we talk about JavaScript. However, it can display errors related to improperly linked images and CSS files.
3. This is the styles panel. It displays what styles are being inherited by your HTML elements.
4. This is the box model panel. You can see margin and padding that is applied to your elements in a visual manner. It also shows how these properties interact with your object.
5. This is the element style panel. You can use it to apply inline styles to your elements. When you reload the page, however, your styles will reload back to what's in your CSS file.
6. Add styles by clicking between the brackets { }.
7. You can also add styles directly to the selectors you write in the CSS file. Click in the brackets to modify CSS rules and you will see the effects applied to the site as if they were written in your CSS file.

**Demo Chrome's Web Inspector tool for the class.**

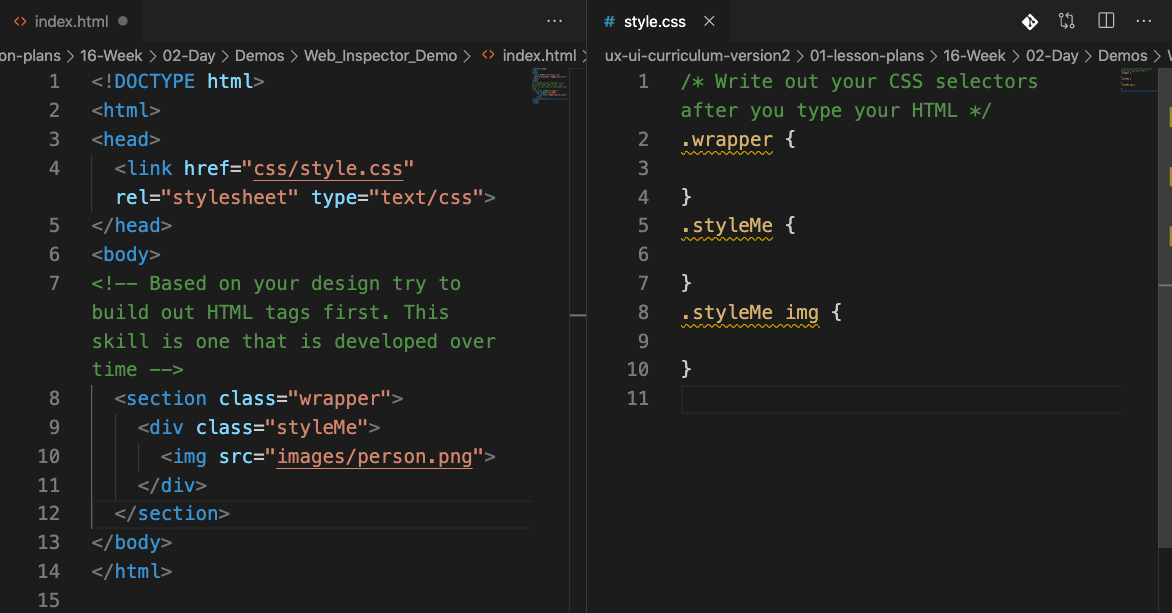
**Instructor note:** During this demo, the goal is for students to see how you can use Chrome's Web Inspector to test CSS and to build out front-end websites visually.

**~~Instructions:~~**

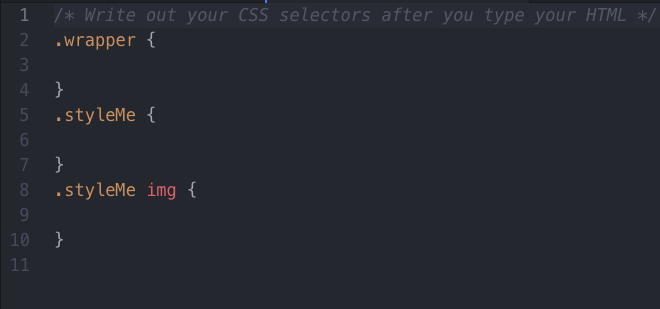
1. ~~Open the index.html file located in~~ [~~17-Week/02-Day/Demos/Web\_Inspector\_Demo/index.html~~](https://github.com/coding-boot-camp/ux-ui-curriculum-version2/tree/master/01-lesson-plans/17-Week/02-Day)~~.~~
2. ~~Open the style.css file located in~~ [~~17-Week/02-Day/Demos/Web\_Inspector\_Demo/css/index.html~~](https://github.com/coding-boot-camp/ux-ui-curriculum-version2/tree/master/01-lesson-plans/17-Week/02-Day)~~.~~  
   ~~Say: "When writing front-end code, you should split your HTML and CSS files so you can code them in plain view of each other."~~
3. ~~Right-click on the style.css tab and click split right from the drop-down menu.~~



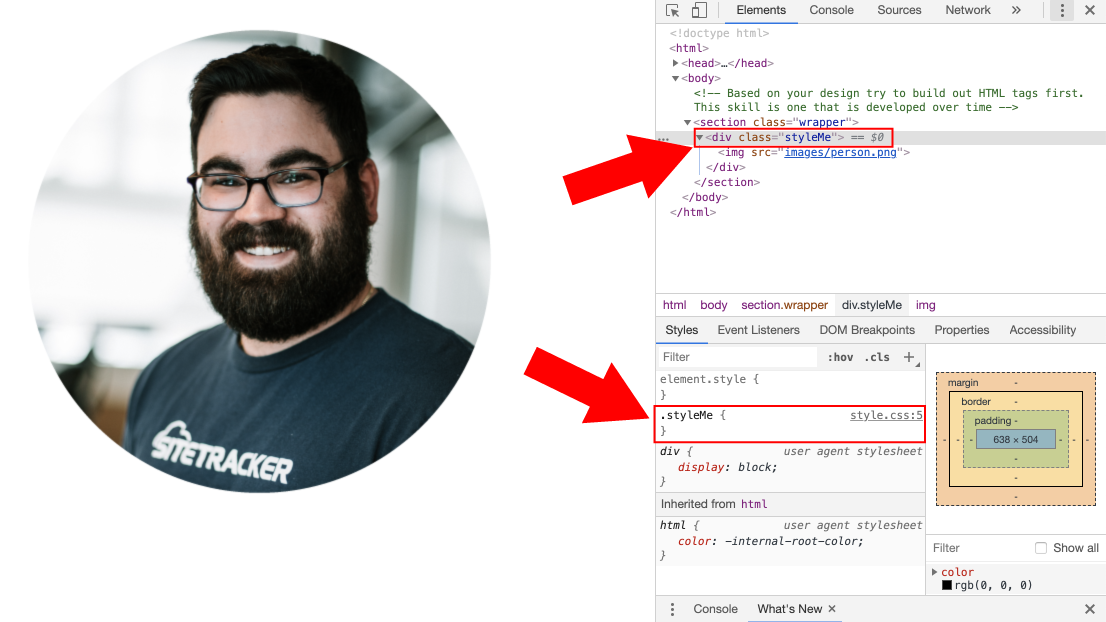
Your workspace should now look like the following:



**Say,** "Now, we can view both our HTML and our CSS while we work. Generally, you want to pre-write your CSS selectors (classes & IDs) before you start styling."



1. Open the index.html file in Google Chrome.
2. Right-click on the picture of the person and click inspect. The Inspector window will pop up to the right (default).
3. Click on the div with the class of .styleMe.



**Say,** "Notice how the Web Inspector can see that we have written a selector without any styles."

1. Type out the following CSS properties into the .styleMe selector.

.styleMe {

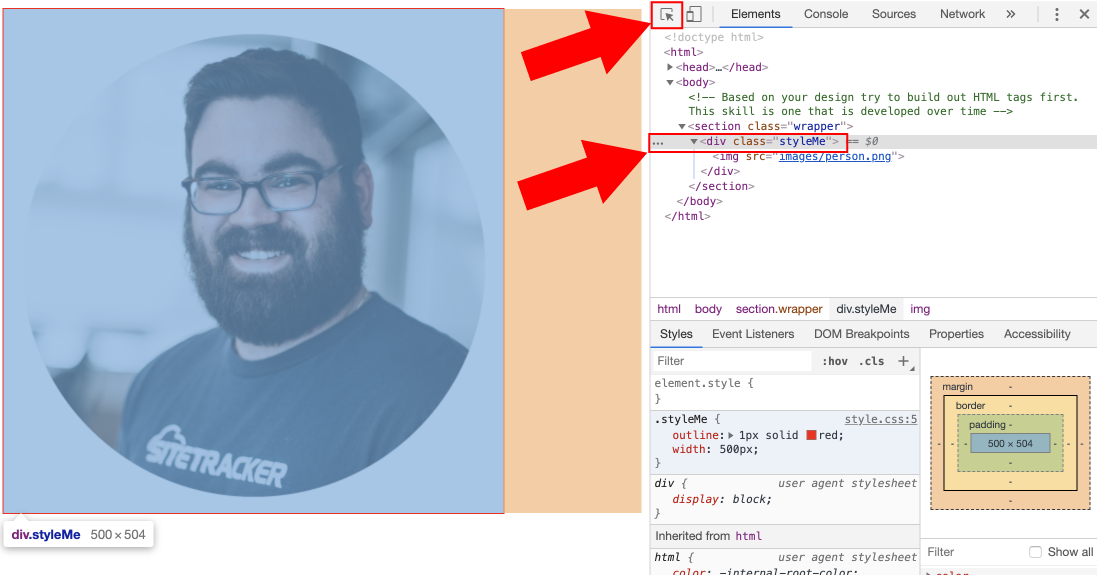
outline: 1px solid red;

width: 500px;

}

**Say,** "Notice how the changes we make happen in real time right before our eyes."

1. Highlight the CSS you wrote in .styleMe and copy it. Paste it into your CSS file. Save your file and reload your page.
2. Hover over the .styleMe HTML element with your mouse and show the class that you can visually see the box model.



* Alternatively, you can click the arrow on the top left of the Web Inspector panel and hover over elements on your screen.

**Instructor note:** the elements panel can be used to locate elements in your HTML. You always have the option of right-clicking and hitting inspect again to view the CSS of whatever you are inspecting.

1. Tell the class that the web is literally made up of boxes. Even shapes that don't appear to be boxes are, in fact, boxes.

**Say,** "You can use the Web Inspector to style and experiment with CSS code."

**Instructor note:** Students need to play with CSS in the Web Inspector. It helps their understanding of how CSS works by letting them see the results of their code visually in front of their eyes.

Now that students have seen how they can use the Web Inspector to build front-end webpages, it's time for them to test their skills by building HTML content for their USPS style guide.

## **~~5. Student Do: Containers and Structure Activity (40 min)~~**

**~~TAs,~~** ~~share a copy of the activity with the class:~~ [~~17-Week/02-Activities/17.2/17.2-01 Containers and Structure~~](https://docs.google.com/document/d/1APsBg734B73WG3vo7kMRT6hHRqOFBfCGYj427TnNxSk/)~~.~~

**~~Summary~~**

~~Students put their HTML skills to the test by adding structure to a style guide.~~

**~~Instructions~~**

~~First, students will open their index.html file and their style guide .xd file. Next, they will add section tags to their style guides.~~

*~~If students struggle with this activity~~*~~: First, double-check student tags for syntax errors. Second, check if a student has constructed their HTML as described in the activity document. As an instructor, you can lean on the solved file to showcase to students how their HTML was supposed to be constructed. Alternatively, encourage students to Google anything they are ensure of. Internet research is a normal part of developers' workflow. For example, if a student doesn't know what a break tag is, they can search for “break tag w3schools.” W3School is an excellent resource for complete beginners to HTML / CSS.~~

*~~If students finish the activity early~~*~~: If students finish early, have them research CSS selectors, as they will be applying them in later activities during this lesson.~~

**~~Be available to students.~~**

~~Monitor students and offer help where needed, but also encourage them to look up tags or documentation during this activity.~~

**~~Deliverable~~**

~~When all tags in the activity file have been added, students’ work is done.~~

## **~~6. Instructor Do: Containers and Structure Activity Review (5 min)~~**

~~Call attention back to your screen and lead a review of the container and structure activity.~~

~~Provide actionable feedback that students can use, for example:~~

* ~~Knowing when to correctly use section and div tags will help even long HTML files be more digestible for yourself and colleagues.~~
* ~~Always double-check your HTML structure for closing tags. Getting in the habit of closing a div (or section) tag whenever you create one will save you time debugging your HTML.~~
* ~~Learning how to construct HTML tags will help you build designs you create yourself later on down the road.~~

~~To test the students for competency, ask them the following questions about div and section tags:~~

* ~~What is a div tag used for? When should you use this tag?~~
  + ~~Answer: A <div> tag is used to create UI elements on a page. A <div> could be anything you want it to be.~~
* ~~What is a section tag used for? When should you use this tag?~~
  + ~~Answer: Sections are used to create whole sections of webpages. Sections contain <div> that all relate to the overarching theme of the section.~~
* ~~How can you use div and section tags together?~~
  + ~~Answer: Sections are used to create themes for groups of content on a page. <div> are then created inside the section to create content for our section.~~

~~Students now understand how div and section tags are used to bring structure to the content on a webpage. They're ready to add an additional structural element to their work with containers—that's next, after the break.~~

## **7. Break (15 min-Weekday) | (45 min-Saturday)**

Break time!

Take 15 minutes (weekdays) or 45 minutes (Saturday) to relax.

Let students know that when they're back from break, you'll talk about the box model, a way to bring structure to a webpage.

## **SLIDE 43**

## **8. Instructor Do: Intro to the CSS Box Model (5 min)**

~~📌 The goal of this section is to help students become familiar with the box model and how it affects the structure of our content. This section supports learning objective 3: use height, width, margin, and padding to create containers for the USPS style guide. This section also supports learning objective 4: build a style guide for USPS.~~

~~The box model in HTML is used to construct the containers that we have practiced building earlier in the lesson. Students have already been applying the box model when they constructed the HTML for their USPS style guide. They just didn't know it yet.~~

~~Students need the box model to be defined so they understand how and why their HTML reacts when experimenting with CSS properties. Understanding how the box model affects our HTML elements is essential for students new to front-end development. It is a fundamental concept when learning how to construct layouts.~~

~~Before we define the properties that affect the box model, you or your TAs should share your experience constructing HTML elements.~~

~~💎~~ **~~Designer Insight:~~** ~~Ask a TA to share their experience constructing HTML elements using the CSS box model. Alternatively, speak to the on-the-job relevance of how understanding the basics of the box model helped you build and design HTML elements.~~

~~As stated earlier, there are five properties that students need to understand that affect the CSS box model. The properties are height and width, margin, padding, and border. When a web developer builds out content, the first step is understanding what properties they can use to build out the basic look and feel of a layout.~~

~~Below, you will walk through each property so that students can understand how these properties affect HTML content.~~

🔑 **Key tech point:** Memorizing what properties affect the box model will improve your skills as a web developer.

TAs, share the following resources with your class:

* [MDN - CSS Box Model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model)
* [internetingishard - CSS Box Model](https://internetingishard.com/html-and-css/css-box-model/)
* [CSS Tricks - CSS Box Model](https://css-tricks.com/the-css-box-model/)
* [W3Schools - CSS Box Model](https://www.w3schools.com/css/css_boxmodel.asp)

The web is made up of boxes. We saw this concept earlier during the Web Inspector tutorial.

## **SLIDE 44**

**Introduce the Box Model**

Each HTML element you create in your HTML file is represented as a rectangular box. The box's content, padding, border, and margin are built up around one another like the layers of an onion. All the box's CSS calculates on the page to display the way content lays out on a page.

* Go to any webpage and use the Chrome Inspect tool.
  + Point out the box model.
* The web is made of a bunch of boxes.
  + CSS treats each element in your HTML document as a box with a bunch of different properties that determine where it appears on the page.
    - Understanding this idea will help you become a much better CSS developer and a better UI designer.

## **SLIDE 45**

## **SLIDE 46**

Let's walk through the properties that affect the box model:

## **SLIDE 47**

1. **Content inside your containers:** Text and images will both cause your box to expand to contain the content inside. The height and width can also be explicitly defined by setting a height and width for a container.
2. **Height/width:** Adding height and width to any of your elements will cause it to expand depending on what properties you set.
3. **Padding:** Padding properties are used to expand space *inside* the box model. This is fundamentally the opposite of margin, which creates space outside the element.
4. **Margin:** Margin properties are used to create space *outside* elements. Margin will move your box away from other elements that surround it.

* The box model is the building block of CSS. Everything you see on the web is a rectangle.

## **SLIDE 48**

## **SLIDE 49**

**~~Instructor note:~~** ~~Pause for questions to be sure that students are grasping the CSS box model concept.~~

~~Next, move on to the activity—students will get hands-on with the CSS box model.~~

## **~~9. Student Do: Apply the CSS Box Model Activity (25 min)~~**

**~~TAs,~~** ~~share a copy of the activity with the class:~~ [~~17-Week/02-Activities/17.2/17.2-02 Apply the Box Model Google Doc~~](https://docs.google.com/document/d/10oOl6ISjokvHhXa52HcVqdT4YvYXDx2vnE5wk_yx0Pg/)~~.~~

**~~Summary~~**

~~Students will create a box model of their USPS style guide, giving structure to their content.~~

**~~Instructions~~**

~~Students will start by opening their style.css file. From there, they will add various elements to create their box: header, primary colors, typography, and UI elements.~~

*~~For students who struggle~~*~~: Double-check students' CSS to make sure they are using the correct CSS selector. Students may target a class using a dot (.) instead of a hashtag (#) or vice versa. If students are having trouble building out their layout, lean on the solved file to show the student what CSS properties are used to build out the HTML element the student is tripped up on.~~

*~~For students who complete the activity early~~*~~, encourage them to study the box model and the differences among inline, block, and inline block level elements, as they are directly related to one another~~

**~~Be available to students.~~**

~~With TAs, keep an eye on students as they work through the activity. Point them to the links to documentation resources in the activity file as needed.~~

**~~Deliverable~~**

~~Once students have added all the elements listed in the activity document, they’re done.~~

## **~~10. Instructor Do: Apply the CSS Box Model Activity Review (5 min)~~**

~~Call attention back to your screen and lead a review of the box model activity.~~

~~Provide actionable feedback that students can use, for example:~~

* ~~The box model is a fundamental concept for building HTML elements.~~
* ~~Content inside containers can have their height and width defined by the content inside of them. Setting height and width to a fixed pixel width can cause content to break outside of its container.~~

**~~Ask the class:~~**

* **~~What CSS properties affect the CSS box model?~~***~~Answer:~~* ~~Height, width, border, margin, and padding.~~
* **~~What is the difference between margin and padding?~~***~~Answer:~~* ~~Margin affects the space around the element; padding affects the inside of our element.~~

**~~Instructor note:~~** ~~Take questions: Be sure that all students have had an opportunity to ask questions before moving on to the next activity.~~

## **~~11. Student Do: Style Our Style Guide Activity (10 min)~~**

**~~TAs,~~** ~~Slack out the following instructions to students:~~ [~~17-Week/02-Activities/17.2/17.2-03-Style Our Styleguide Google Doc~~](https://docs.google.com/document/u/1/d/10ZDF6o2hg9Mwhl7uvmahfgPTPLDeycRUmKma_IKhp-E/)~~.~~

**~~Summary.~~**

~~At this point, the students’ style guide is structurally sound, but it lacks color. Now they will use CSS to add color to their HTML elements.~~

**~~Instructions~~**~~.~~

~~First, students will open their style.css file. Next, they will add selectors for a variety of colors according to the activity file.~~

*~~For students who struggle with this activity~~*~~: It is common for students to write incorrect selectors during this activity, so keep an eye out for IDs that should be classes (and vice versa). Sometimes students will also type out CSS properties incorrectly. For example, a student might type out font-color as opposed to just color. If you encounter students incorrectly writing CSS properties, tell them to Google their problem—use Google to search for "CSS font color property."~~

*~~If students complete the activity early~~*~~, ask them if there is anything different in terms of the design of the site. Tell the students to practice their UI or front-end skills by redesigning the Adobe XD file located here:~~ [~~17-Week/02-Activities/17.2/17.2-USPS-Style-Guide.xd~~](https://drive.google.com/open?id=1fHOaNCrTEWmZy6orEvnO1xRQD0iX-P5B) ~~or by practicing their front-end skills by continuing to style and alter the USPS style guide.~~

**~~Be available to students.~~**

~~Monitor students' progress throughout the activity. Demo the code on your own machine if that will help students.~~

**~~Deliverable.~~**

~~Students will share a screenshot of their work via Slack.~~

~~Before we move onto the next topic, let's take a moment to review what we just learned.~~

## **~~12. Instructor Do: Style Our Style Guide Activity Review (5 min)~~**

~~Call attention back to your screen and lead a review of the style guide activity.~~

*~~Place this activity in the context of something designers do every day on the job~~*~~: At the completion of this activity, students will have gone through a very realistic experience of work a junior front-end developer might do. Students have:~~

1. ~~Created HTML from a design file (it's also realistic for students to have created the design themselves, then built it).~~
2. ~~Styled their HTML to match the design file.~~

~~Review the work of two to three students, and consider the following questions as the basis for discussion.~~

**~~Ask:~~**

* ~~What do we like about this style guide?~~
* ~~What could be done differently in an iteration of this style guide?~~

~~Be sure to provide feedback that is actionable, for example:~~

1. ~~Building the HTML first makes styling UIs easier later on.~~
2. ~~Always make sure students use the appropriate selector when targeting HTML elements. Students will often mistake classes for IDs when writing CSS selectors (and vice versa).~~

~~Invite the TAs to comment as well.~~

~~💎~~ **~~Designer Insight:~~** ~~This is a great moment to share your professional experience with style guides or a story from your first attempt at a style guide.~~

~~Next, you're going to move on to adding font families with Google Fonts to our webpage.~~

## **~~13. Instructor Do: Google Fonts and Adding Font Families (5 min)~~**

~~📌 The goal of this section is for students to learn that they can add fonts to their webpage. Font families can be both inherited in your browser and hosted on the web. This section supports learning objective No. 5: demonstrate competency with Google Fonts by adding it to our USPS style guide.~~

~~Fonts have a powerful effect on users. An effective font can completely alter the "personality" of a design or webpage and even be what subconsciously keeps a user on a site for a longer period of time.~~

~~Websites look better with custom typography—think of a design file. Designs are always better when designed with custom typefaces and color. Imagine trying to create beautiful UI work if your only choices are Arial and Helvetica, 16-point font, and black as your only color.~~

~~Arial and Helvetica are the default fonts on the web. In the past, you were forced into choosing between 15 "web-safe" fonts depending on your browser. Web-safe fonts are fonts that come installed into the browser by default.~~

~~💎~~ **~~Designer Insight:~~** ~~What experience do students have with fonts? What are their frustrations or their favorite fonts? Where have they sourced some interesting fonts?~~

## **SLIDE 56**

Google realized this and created [Google fonts](https://fonts.google.com/).

Google Fonts allows you to host any font that is hosted on their servers on your webpage. They also have literally a thousand fonts for you to pick from that allow you to customize your webpage.

Students will have an opportunity to do this hands-on in the next activity, following your demo.

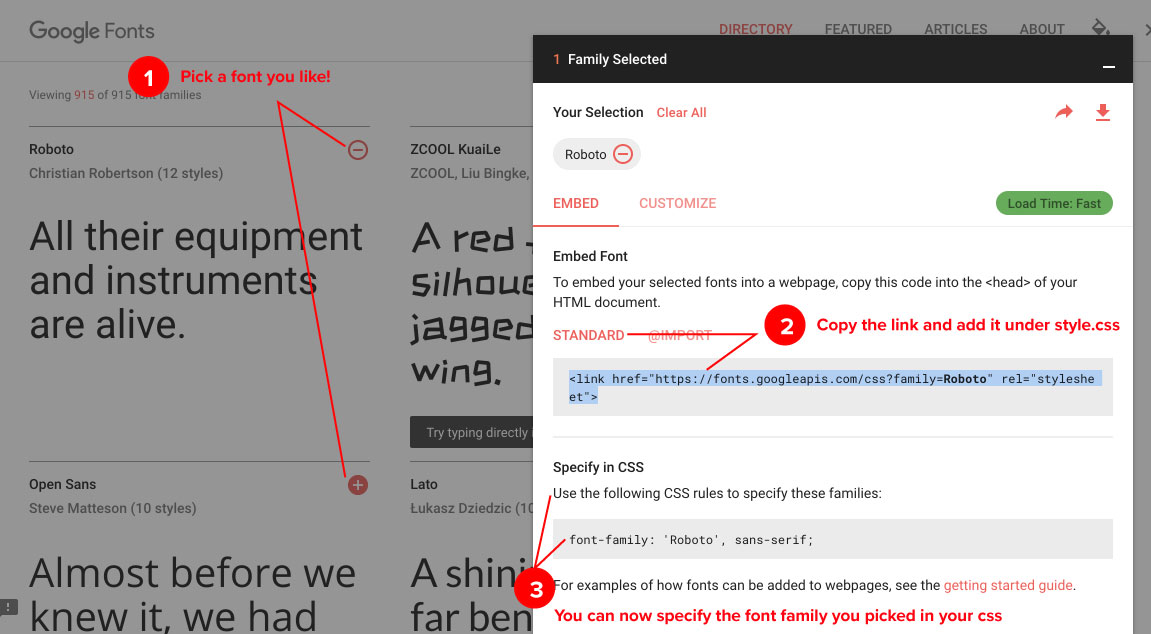
## **14. Instructor Do: Demo - Adding Google Fonts to Your Webpage (5 min)**

Give students a quick demo of how easy it is to add Google Fonts to a webpage.

**Say,** "What I'm about to show you will be very easy for you to execute on your own."

Google Fonts are so easy to use, students can add them to their website in three easy steps!

## **SLIDE 58**



**Instructions**

1. Open the solved ID and classes activity: [02-Day/Activities/04\_Add-A-Custom-Font/solved/index.html](https://github.com/coding-boot-camp/ux-ui-curriculum-version2/blob/master/01-lesson-plans/17-Week/02-Day/Activities/04_Add-A-Custom-Font/solved/index.html).
2. Open [https://fonts.google.com](https://fonts.google.com/).
3. Pick a font you like by pressing the + button.
4. On the bottom right, you will see the font you selected and a link tag.
5. Copy the link tag.
6. Paste it over the custom style sheet.
7. Open index.css located in the CSS folder.
8. Create a CSS selector that targets the body tag:

body {

}

1. Add the font-family for the custom font to the body tag. Here is an example:

body {

font-family: 'Open Sans', sans-serif;

}

1. Be sure to tell the students that they can add as many fonts as they want, but too many will slow down the website.

Easy, right?

Google Fonts allow you to host any font that is hosted on their servers on your webpage. There are nearly a thousand fonts for you to pick from, allowing you to customize your webpage like a pro!

## **15. Student Do: Custom Fonts on Your Webpage Activity (15 min)**

**TAs,** Slack out the following instructions to students: [17-Week/02-Activities/17.2/17.2-04-Custom Fonts On Your Webpage](https://docs.google.com/document/d/1USDlv3FePscFqtPjs2-bL2BfBCDB-DzfVVUbisOI1do/).

**Summary**

Students will add a font from Google Fonts to their webpage, then style it with CSS, using the activity instruction doc as their reference.

**Instructions**

In Part 1, students will add a Google Font to their index.html file.

In Part 2, students will add CSS to their page.

*If students finish early,* you should recommend that they start or continue to work on their homework. Students now have the basic skills needed to create the base HTML/CSS of their portfolio page.

**Deliverable**

Students will have added a font to their webpage (no need to share anything in Slack).

**Support students as they work.**

With TAs, monitor students' progress and pause to help students who seem to get stuck. Remind them to use Google or a classmate as resources, too.

## **16. Instructor Do: Custom Fonts on Your Webpage Activity Review (5 min)**

Call everyone's attention back to your screen and lead a discussion of the custom fonts activity.

**Instructor note:** Encourage students to explore Google Fonts further at home.

* **TAs,** Slack out 20 Best Web Fonts From Google for students to read a home: <https://www.awwwards.com/20-best-web-fonts-from-google-web-fonts-and-font-face.html>.

Some questions you could ask to stimulate reflection and conversation:

* "What is the CSS box model? Why is it important to understand it?"
  + *answer:* The box model is how the browser renders elements of HTML and CSS. Developers can control each parameter of the box.
* "Why would you want to reset your browser-specific styles?"
  + *answer:* Unfortunately, not every web browser renders your CSS the same way. Just try viewing a webpage in explorer 10 vs Safari vs Chrome.
* "What is the Web Inspector? How do front-end developers use it?"
  + *answer:* Web development tools allow a web developer to use the DOM to quickly render a webpage element and test front-end styles quickly.
* "What is the difference between a <section> tag and a <div> tag ?"
  + \_answer: The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document, whereas the <div> tag defines a division in an HTML document. The <div> tag is used to group block-elements to format them with CSS. Think of it like parent-child. <section> is the parent and <div> is the child.

**Instructor note:** Pause for questions.

This is the final section of the class, so be sure to invite students who want feedback on their work to stay online for office hours.

Thank students for their hard work and move on to review the concepts learned today.

## **17. Instructor Do: Recap and End Class (5 min)**

We covered a lot of concepts today, so let's do a quick recap.

**Review the concepts students learned today.**

Spend a couple of minutes helping students anchor today’s learning. Ask students to raise their hands—or post answers in Slack—and offer definitions for the following concepts, all learned today in class:

* "What is the box model? Why is it important to understand the box model?"
  + Answer: The box model is four CSS properties that create the containers of HTML. It is important to understand because everything you build revolves around the box model.
* "What is the Web Inspector? How do front-end developers use it?"
  + Answer: The Web Inspector is a tool used to inspect HTML in your web browser. Front-end developers use it to test out CSS properties and JavaScript on their website before making it live.
* "What is the difference between a div tag and a section tag? When should you use a div tag? When should you use a section tag?"
  + Answer: Sections are used to create themes for groups of content on a page. Divs are then created inside the section to create content for our section.

Well done!

Let the class know that they'll be learning more about CSS and HTML next class.

Say good night and dismiss the class.