**Lesson 2 - Front End:**

| **Title of Unit** | Foundations | **Grade Level** | 11-12 |
| --- | --- | --- | --- |
| **Subject** | Mobile App Development | **Time Frame** |  |
| | **Description** | In this section you'll spend a good deal of time getting familiar with the major client-side (browser-based) languages like HTML, CSS, and JavaScript. You'll get to build a webpage with HTML/CSS and learn some programming fundamentals with JavaScript. | | --- | --- | | | | |
| **Stage 1 - Identify Desired Results** | | | |
| **Learning Outcomes**  What relevant goals will this unit address? | | | |
| Computer and Information Sciences, General.  **CIP#**: 11.0101  Pathway Competencies   * **Algorithms & Programming**: | | | |
| **Enduring Understandings** | | **Essential Questions** | |
| *Students will understand…*   * *The differences between a web page, web server, web browser and search engine.* * *What is the role of HTML in a web page?* * *What is the role of CSS in a web page?* * *What is the role of JavaScript in a web page?* | | *Content specific….*   * *Which language is responsible for the semantic structure of a document?* * *Which language can change the font, text-size, or background-color of an element?* * *Name a few behaviors that javascript can be responsible for on a website.* * *What is the purpose of an opening tag and closing tag in an HTML element?* * *What is the selector in a CSS rule set?* * *What is the property and property value in a CSS rule set?* * *What is the declaration in a CSS rule set?* * *How do you reference a file in the same directory as your HTML file?* * *How do you reference a file in a directory below your HTML file?* * *How do you reference a file in the directory above your HTML file?* * *What is the difference between HTML and CSS?* * *For accessibility in HTML, what is the attribute used to describe an image (on screen readers or if it fails to load)?* * *What is the difference between CSS Grid and Flexbox?* * *For a responsive website, should it be designed mobile-first or desktop-first?* * *Describe the components of the CSS Box Model.* * *In CSS, what is a breakpoint?* * *What is a div and how are they used?* * *What are the two main groups of CSS properties that control typography style?* * *What is the “query string” in a URL and what does it do?* * *What is the difference between “pixels” and “em”?* * *How does inheritance work for CSS styles, i.e. how does an element get its “default” styles?* * *What are two CSS attributes you can change to push an element around on the page?* * *What are the three different ways to include a CSS stylesheet in your project or use CSS to style a particular element?* * *What is the “default stylesheet” or “user agent stylesheet”?* * *What is the purpose of a CSS reset file?* * *How do you open developer tools in your browser?* * *How do you select a specific element on your page with your browser’s developer tools?* * *How do you change CSS in real time on specific elements of a web page with your browser’s developer tools?* * *What does a strikethrough in a CSS element mean in your browser’s developer tools?* * *How do you check every inherited style for an element in your browser’s developer tools?* * *How do you edit HTML in real time in your browser’s developer tools?* * *How do you toggle responsive design mode in your browser’s developer tools?* * *Are changes made in your browser’s developer tools permanent?* * *How do you open the console in your browser’s developer tools?* * *What is the best resource for learning about your browser’s developer tools?* | |
| **Knowledge:** | | **Skills:** | |
| *Students will know how to...*   * *Why do we separate HTML and CSS?* * *What are classes and IDs (and how are they different)?* * *What are elements?* * *What are tags?* * *What are attributes?* * *What are forms?* * *What is a div?* * *What are selectors?* * *What are properties?* * *What are values?* * *What is the “query string” in a URL and what does it do?* * *What is the difference between “pixels” and “ems”?* * *How do CSS styles for a particular element get inherited? i.e. how does an element get its “default” styles?* * *What are two CSS attributes you can change to push an element around on the page?* * *What are the three different ways to include CSS in your project or use CSS to style a particular element?* * *What is the “default stylesheet” or “user agent stylesheet”?* * *Why use a CSS reset file?* | | *Students will be able to…*   * *Create a basic HTML webpage using HTML tags* * *Style an HTML using CSS* * *Identify and use CSS Selectors* * *You can open the developer tools in your browser.* * *You can select a specific HTML element on your page with your browser’s developer tools.* * *You can use developer tools to experiment with the CSS on specific elements of a web page.* | |

| **Stage 2 – Assessment Evidence** | | |
| --- | --- | --- |
| **Performance Task** | | |
| **PROJECT: GOOGLE HOMEPAGE****Introduction** For this mini-project, you’ll deconstruct an existing web page and rebuild it. Don’t worry if the links don’t go anywhere and the search box doesn’t do anything when you submit it. The goal is to start thinking about how elements get placed on the page and roughly how they get styled and aligned. For some of you, this may be the first time you’ve actually tried to “build” something in HTML (very exciting!).  Using the browser’s developer tools (right-clicking something on the page and clicking “inspect element” will get you there) will be your best friend. Build the page in a .html text file and open it in your browser to check it out. | | |
| **Other Evidence** | | **Student Self-Assessment** |
| * Knowledge Checks * Assignments | | * Reflection |

| **Stage 3 – Learning Plan** | | | | |
| --- | --- | --- | --- | --- |
| Activity 1 - Introduction to Front End  Activity 2 - HTML and CSS Basics  Activity 3 - Developer Tools  Project 2 - Google homepage | | | | |
| **How will you engage students at the beginning of the unit? (motivational set)** | | | | |
| **INTRODUCTION TO THE FRONT END** The “front-end languages” live in the browser. After you type an address into the address bar at the top and hit the enter/return key, your browser will receive at least an HTML file from the web server. That file will likely tell the browser to request a CSS file and a JavaScript file as well (probably many more than one, but we’ll keep it simple).  Each of these languages performs a separate but very important function and they work harmoniously together to determine how the web page is STRUCTURED (HTML), how it LOOKS (CSS), and how it FUNCTIONS (JavaScript). And keep in mind that your *browser* handles figuring out how to make these files into a functioning web page (not the server).  Front-end web development is NOT design (you won’t be playing around in Photoshop or anything), but a front-end developer *does* apply the work of designers to the web page by translating their well-designed layouts into real code. The front-end developer stands between the designer on one end and the back-end developer on the other, translating the design into code and plugging the data from the back-end developer into the right spots. They must also handle all the possible interactions that the user may need to make with the page.  On the front end, you will need to be highly conscious of who your user is and how they will be interacting with your web page, because you are building their gateway to your page or product. This may mean gaining a strong understanding of accessibility and things like responsive development down the line, but first you need to build up your toolkit and pick up the fundamentals of the front-end languages.  In the following lessons, you’ll get a healthy understanding of each of the three front-end languages. To get warmed up, we’ll start at the high level. | | | | |
| **#** | **Lesson Title** | **Lesson Activities** | **CCCs** | **Resources** |
| 1 | Introduction to Front End | 1. Read a brief intro to front-end development from this [interview with Nick Schaden](https://web.archive.org/web/20200601022721/https://generalassemb.ly/blog/what-is-front-end-web-development/). 2. Read about the [role of HTML](https://webapps-for-beginners.rubymonstas.org/html.html) from this Ruby Monstas page. 3. Get a high-level overview of how all three languages work together in MDN’s [Getting started with the web](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web). Learners should only read: [Dealing with files](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files), [HTML basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics), [CSS basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics) and [JavaScript basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics). There is no need to implement the website; simply read about the process at this point. 4. Get an overview of what you can do with these documentation pages: [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML/Element), [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Keyword_index), and [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference). You will not understand most of what is there just yet, but bookmark the pages for later reference. 5. Bookmark [DevDocs.io](https://devdocs.io/). Read the “Welcome” message. Massive API documentation collection that even works offline. Essential collection of reference material for everything covered and more. (Maintained by [FreeCodeCamp](https://www.freecodecamp.org/)) |  |  |
| 2 | HTML and CSS Basics | 1. Dive right into learning HTML and CSS with [freeCodeCamp’s interactive tutorials](https://www.freecodecamp.org/learn) to introduce you to these new concepts. Do the entire section labeled “Responsive Web Design Certification” only. It’s long and, if you’re new, might take you at least a few days to get through. This resource is meant to help you develop familiarity, so do not expect to memorize everything. Instead, expect to google these concepts in the future when you need to use them. They also have some projects at the end of this section. They’re good practice, but you can consider them optional. (A large part of their proposed ‘300 hours’ for completion comes from these projects, so it probably won’t take you that long if you only do the exercises). Note: If you are using a browser add-on such as Dark Reader (or a similar add-on that creates a “dark mode” for a website), it may interfere with certain exercises. If your solution isn’t accepted, try disabling the browser add-on and submitting it again. 2. Optional: Learn about your browser’s default stylesheet and CSS resets [in this video](https://www.youtube.com/watch?v=14Vb6tZCjEY) (resets are mentioned starting at the 2:00 mark). This is why there are some spaces that show up in your layout even if you haven’t specified CSS. Real developers almost always use a CSS reset to blow away the default stylesheet and let them work from scratch. If you’re still curious, here’s a [longer video](https://www.youtube.com/watch?v=HqRFPLP7Ffs) on resets. |  |  |
| 3 | Developer Tools | 1. Watch [this 10-minute video](https://www.youtube.com/watch?v=wcFnnxfA70g) that goes over the most useful features of the Chrome DevTools in some detail. 2. Watch this [awesome tutorial](https://www.youtube.com/watch?v=Z3HGJsNLQ1E) from LearnCode.academy on how to use developer tools effectively when working with your CSS. It talks a bit about Bootstrap, which you may not know about yet. Don’t worry about it at this point; just check out the stuff he’s doing to CSS in the DevTools. In particular, editing CSS *in the browser* in real time is a serious productivity booster compared to using your text editor and continuously refreshing to see the changes. 3. Skim through the [official Chrome DevTools docs](https://developers.google.com/web/tools/chrome-devtools/) [(or Firefox’s!)](https://developer.mozilla.org/en-US/docs/Tools). Chrome and Firefox are constantly enhancing and updating their developer tools. The basic functionality is going to be the same of course, but it’s not unusual for things to move around or change appearance as the tools evolve. These docs should be your up-to-date resource. If your version of DevTools doesn’t look like the videos above, reference these docs to find out where everything went. |  |  |
| P2 | Google homepage | **Easy Version: Build the** [**Google.com**](http://www.google.com/) **homepage** (the simple one with just a search box).  Inside your project folder, create your index.html file.   1. Tips:    * DON’T BE A PERFECTIONIST! You’re just trying to make it *look* like google.com, not actually function like it and it doesn’t have to be spaced exactly the same way to the pixel. Any dropdown menus or form submissions or hover-highlighting should be ignored.    * USE GOOGLE! You’ll probably run into roadblocks where you can’t figure out how to do something so do what all good devs do… Google it!    * Now is a good time to [set up the live server extension](https://youtu.be/mGORIVStWWc) in VSCode, if you haven’t already. This extension will save you from the hassle of refreshing the browser window repeatedly to see changes made in your code and will automatically load the most recent change in the browser window.    * If you’re frustrated with trying to get buttons or inputs to style the way you want (for instance, they seem to just not respond to any styles), look into the CSS property -webkit-appearance: none; or -moz-appearance if you’re using Firefox.    * [Here’s a link to the Google Logo just in case they temporarily replace it with a Doodle!](https://www.google.com/images/branding/googlelogo/1x/googlelogo_color_272x92dp.png)    * [Here’s a cached page of the Google home page for reference in case the original logo is not there.](https://web.archive.org/web/20191130234759/https://www.google.com/)    * [Here’s a link to the Google Material Icons page. This is a great resource for free to use open source icons.](https://fonts.google.com/icons) 2. Start with just putting the main elements on the page (the logo image and search form), then get them placed horizontally. You can either download the Google logo or link directly to its URL on the web in your <img> tag. 3. Next do the navbar across the top, first building the content and then trying to position it. Check out [how to build a horizontal CSS navbar](http://www.w3schools.com/css/css_navbar.asp) if you’re lost. 4. Finally, put in the footer, which should be very similar to the top navbar. 5. In general, do as much on your own as you can before relying on the developer tools (or viewing the page’s source code) to help you along. 6. Push your project to GitHub using the instructions above! Then consider sharing your project with the community by submitting a link to your repository in the Solutions section below.  **Difficult Version (optional): Build the** [**Google.com search results page**](https://www.google.com/search?q=build+this+webpage) You should be able to reuse much of your code from before if you started with that project. Again, don’t worry about links to nowhere and forms that won’t submit and hard coding the search results (which you’ll have to do of course), just focus on placement and order of items on the page.  Note: All the classes and id’s and names of elements that you inspect on Google’s home page are nonsensical strings (like <div class='srg'>). This is because the code was Minified ([see the Wikipedia entry here](http://en.wikipedia.org/wiki/Minification_(programming))), which removes or shortens unnecessary characters and names to help the page load faster. The HTML (or JavaScript or CSS) file will be smaller but the browser can still read it just fine. |  |  |

| **Stage 4 - Assess and Reflect** | |
| --- | --- |
| **Considerations** | **Comments** |
| **Is there alignment between outcomes, performance assessment and learning experiences?** |  |
| **Have I made purposeful adjustments to the curriculum content (not outcomes), instructional practices, and/or the learning environment to meet the learning needs and diversities of all my students?** | For struggling students:                    For students who need a challenge: |
| **Do I use a variety of teacher directed and student centered instructional approaches?** |  |
| **Do the students have access to various resources on an ongoing basis?** |  |
| **Have I nurtured and promoted diversity while honoring each child’s identity?** |  |