**Front-End Academy  
  
CSS**

**Exercise Guide**

Version 1.1 ITC

.

### Introduction

#### This document contains the all of the exercises you will complete during the two days you are working through the CSS content

#### Day 1

#### Videos:

#### Styline Websites in CSS

#### Working with Colors and Images in CSS

#### Exercises:

#### Using CSS Selectors

#### Styling Text

#### Using Custom Fonts

#### Style the Travel Site

#### Day 2

#### 

#### Videos:

#### Creating Layouts with CSS

#### 

#### Exercises:

#### Creating Sticky Footers

#### Play Flexbox Froggy

#### Using the Flexbox

#### (Optional) Pluralsight Interactive Courses

**Day 1**

### Exercise - Using CSS Selectors

There are some great websites where you can practice your CSS selectors without having to create a web page.

In this exercise, we will use the **CSS Diner** application at [https://flukeout.GitHub.io/](https://flukeout.gitlab.io/%20) and try to do "at least" the first 12 exercises. You may do more if you feel it is productive!

Feel free to talk to other learners in your cohort for help. Get together and make sure everyone understands why each exercise works!

### Exercise - Styling Text

Continue working in HTML-experiments. In an earlier exercise, you created a story.html page. We will style the text on that page.

**Step 1.** Create CSS

Create a folder called CSS and add a styles.css file to it. Then, include styles.css in the page

**Step 2.** Use CSS to bold some text

In your first paragraph, select a phrase that you want to bold. Surround it with a <span> and set its id to “bigdeal”

Using an id selector, create a CSS rule to bold that text

View the page in the browser

**Step 3.** Center the page heading

If the page heading is not an h1, make it an h1.

Using a tag selector, create a CSS rule to center the page heading

View the page in the browser

**Step 4.** Set the background color of "Sneak Peak" and "Main Story" headers

Use a CSS rule to set the background color of these headings to a tan or light gray color. We leave the type of selector to use up to you!

View the page in the browser

**Step 5.** Add text shadowing to the "Review Questions" heading

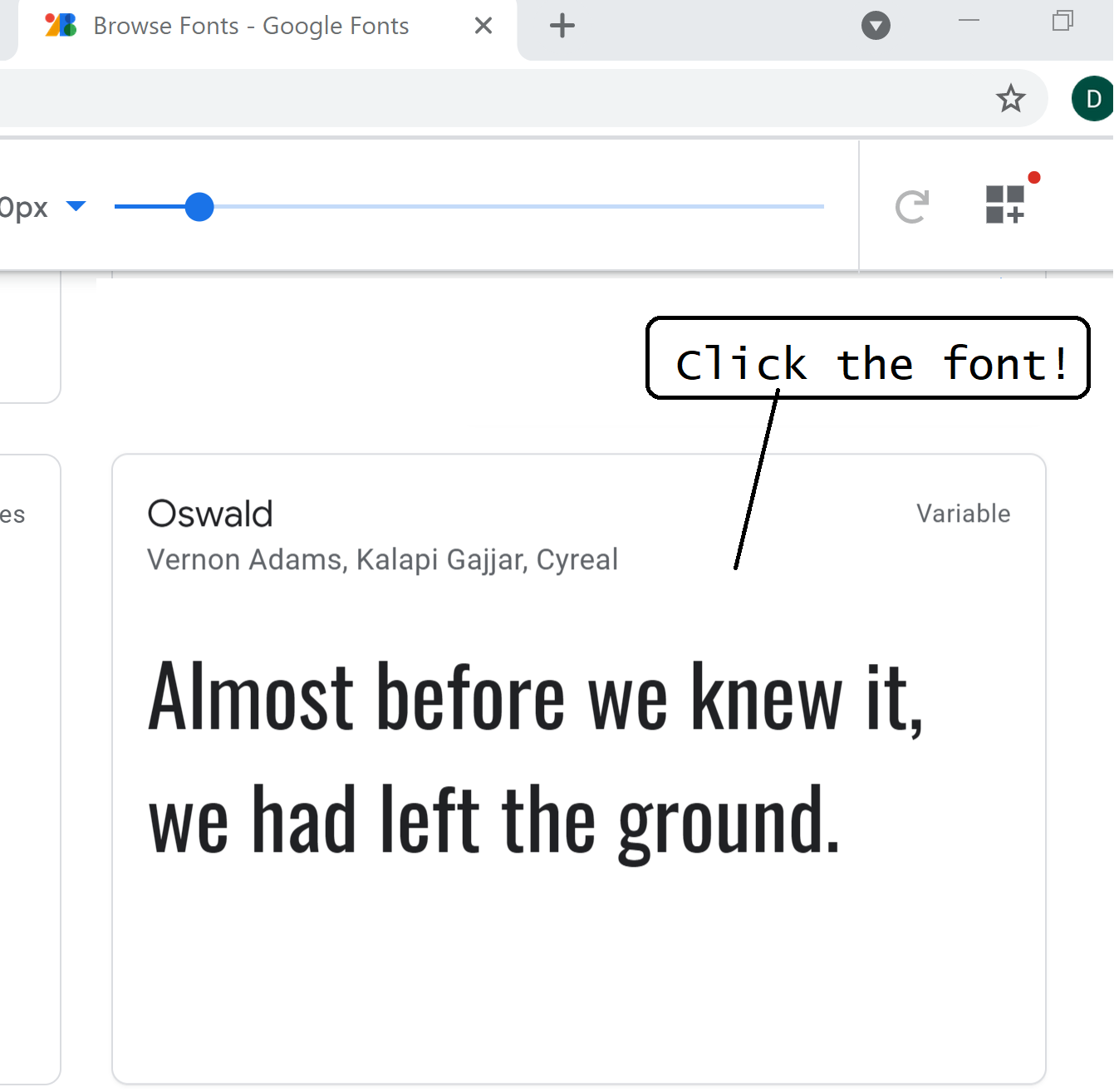
Research how to add 2px red text shadowing to the heading then create the CSS rule to make it happen. Note: the Review Questions heading should NOT be affected by the CSS rule in step 4.

View the page in the browser.

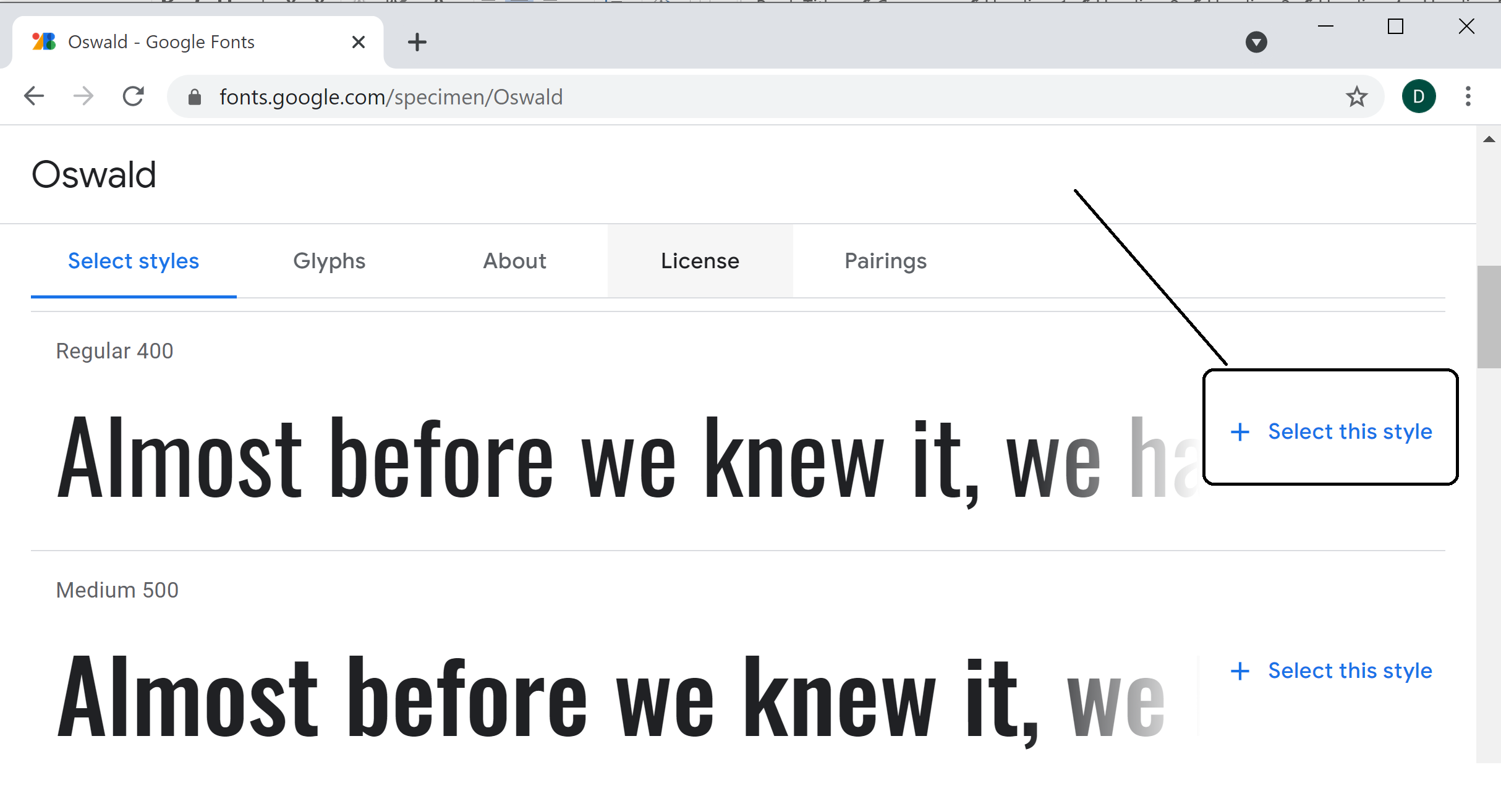
**Step 4.** Commit your changes and use the message "Styled Text". Push to GitHub.

### Quick Lesson - Custom Fonts

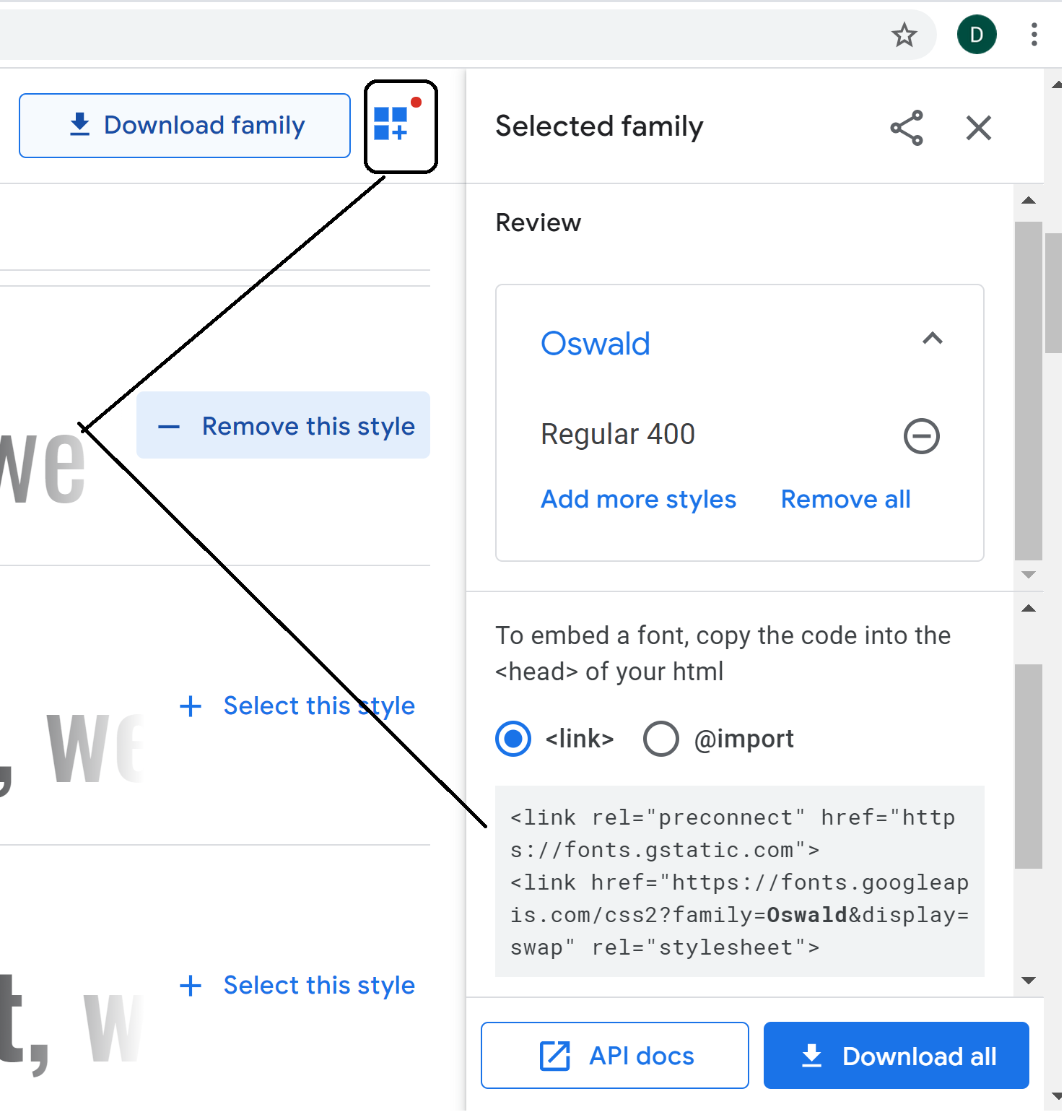
* **Google has many custom fonts available free for your use**
* <https://fonts.google.com/>
* **To learn about the font, click the font**



* **Then select the font with the characteristics you want**



* **Finally, view the tags to need to use to link the font into your HTML**



* **The preconnect link tells the browser to connect to the server ahead of time to make the download go faster**
* Note: Add crossorigin to the preconnect link

<head>  
 <link rel="preconnect" href="<https://fonts.gstatic.com>" crossorigin>  
 <link href="https://fonts.googleapis.com/css2?family=Oswald&display=swap" rel="stylesheet">  
 <style>  
 body {  
  font-family: "Oswald", sans-serif;  
 }  
 </style>  
</head>

### Exercise - Using Custom Fonts

Continue working in the story.html page

**Step 1.** Use a custom Google font

Follow the process in the previous Quick Lesson to add a custom font to the paragraphs on the page

View the page in the browser

**Step 2.** Commit your changes and use the message "Added custom font". Push to GitHub.

### Exercise - Style the Travel Site

Revisit your simple-travel-site project. We are going to add some styling rules to your site. We will do it in a new branch so that we can test out things before merging our changes back into our main branch. If we mess things up, we won't break our original code base.

**Step 1:** Make sure you have the most recent copy locally

Use the git pull command to get the most recent version repo from GitHub.

**Step 2:** Create a new working branch

Create a new branch off main named EnhancedIndexBranch. Then switch to it.

**Step 3:** Add the enhanced styling

If you don't have a .css file for your site, create one now and then link it to your index.html page.

Using an ***element selector***, create a CSS rule that applies to all paragraphs. In it, set the font-family to one of the ones from: <https://www.w3schools.com/cssref/css_websafe_fonts.asp>. Make sure to include the fallback font of either serif on sans-serif.

Using a ***class selector***, create a CSS rule that will be applied to the HTML table. Place a border around the HTML table and its cells. If you aren't sure how, refer to <https://www.w3schools.com/css/css_table.asp>. Apply that class to the HTML table on the index page.

Test your page.

When things work well, commit your changes.

**Step 4:** Use a custom font

Use one of Google's custom fonts in your index.html page.

Test your page.

When things work well, commit your changes.

**Step 5:** Merge your changes

Switch back to the main branch. Since you are happy that your changes didn't mess anything up, merge your EnhancedIndexBranch.

You can delete your EnhancedIndexBranch.

Run git status and if all is well, push your repo to GitHub.

**Day 2**

### (Challenge) Exercise - Sticky Footers

Continue working in HTML-experiments.

**Step 1.** Create a page named sticky-footer.html

Add a <header> element to the top of the page with a faded blue background. Within it, create an h1 with the text "ACME Bird Food Company"

Add a <footer> to the bottom of the page with a light yellow background. Within it, add small centered text that shows the copyright symbol and follow it with the current year and "by ACME Bird Food Company"

Add 7 paragraphs of Lorem Ipsum text. Make sure there is enough text that your browser needs to scroll to see all of it.

View in the browser

**Step 2.** Make the footer sticky

A sticky footer always stays at the bottom of the browser window. Do some research (Google? ChatGPT?) and see if you can figure out how to make that happen.

Then, make the change and view the page in the browser

**Step 3.** Commit the changes and use the message "Added header and sticky footer". Push to GitHub.

### Exercise - Play Flexbox Froggy

The easiest way to learn about the flexbox is to play an online game that lets you try out different flexbox properties.

In this exercise, we will go to <https://flexboxfroggy.com/> and try to complete ***at least*** the first 10 levels.

Feel free to talk to other learners in your cohort for help. Get together and make sure everyone understands why each exercise works!

### Exercise - Using the Flexbox

Continue working in HTML-experiments.

**Step 1.** Make a copy of your vacations.html page and name it vacations2.html. Recall that it holds 4 pictures of your favorite vacation destinations

Using the Flexbox, redesign the page so that on a wide browser, the images appear side by side, but on a narrow browser, they wrap.

View in the browser

**Step 2.** Commit the changes and use the message "Added flexbox". Push your to GitHub.

### (Optional) Exercise - Pluralsight Interactive Courses

If you have time to play some more or want more practice, Pluralsight has several “Interactive Courses” that have skills checks and coding challenges that you can complete within the course.

We have links to a few of these courses in a tab at the very bottom of the channel that says “Optional Exercises”.

# HTML and CSS: Creating a Basic Website (Interactive)

<https://app.pluralsight.com/library/courses/html-css-basic-website/table-of-contents>

This course will lead you through creating code and making changes to existing HTML and CSS code to accomplish specific tasks.

#### CSS: Specificity, the Box Model, and Best Practices (Interactive)

<https://app.pluralsight.com/library/courses/css-specificity-box-model-best-practices/table-of-contents>

This course will test your skills in CSS through a series of challenges

### Build a Rock Climbing Company Landing Page with HTML & CSS

<https://app.pluralsight.com/code-labs/challenges/76949298-02ad-441d-b778-dd82a09d93c6?access_token=37AA5B81AD371DA91BB715305B27F9D7&step_id=c4ac7d6e-21a8-4c99-9bb2-d925432af1c3>

This course will lead you through creating code and making changes to existing HTML and CSS code to accomplish specific tasks.