

# Coffee Shop Assessment

In this assessment, you'll be writing HTML and CSS. You won't need to write any JavaScript and all images are provided to you.

Please feel free to reference your notes, books, or even research on the Internet. But please make sure all code is written by you alone and is original. Please don't collaborate with other developers or designers whether online or in person.

Please read all the instructions before beginning or you may misinterpret certain instructions! It's also likely to save you some time.

The assessment should take about two hours for a developer who is new to HTML and CSS.

You'll be creating this page:

MENU   LOCATION   STORY   ORDER ONLINE

Links don't have to actually go anywhere. You don't have to create these pages.

Header to serve as a site menu



"Hero" image

Overlaying the invitation on the hero image is a bonus. Place it underneath the image unless you want an extra challenge

Four featured items cards



### Muffin and espresso

These bakery-style Espresso Chocolate Chip Muffins are moist, dense and rich with a distinct coffee taste. That smooth espresso flavor combines perfectly with the sweet chocolate chips to make these muffins the perfect breakfast or afternoon snack for coffee lovers.

\$4.50

Food image



### Double latte

This delicious iced chocolate latte is perfect for a hot day. It can be topped with whipped cream or drunk plain. Could it BE more delicious?

\$2.75

Food title

Food description

Food price



### Coffee and a scone

Nothing will get you going in the morning like these scones! Paired with our special Tribiani blend dark coffee to offset the sweetness of the scones.

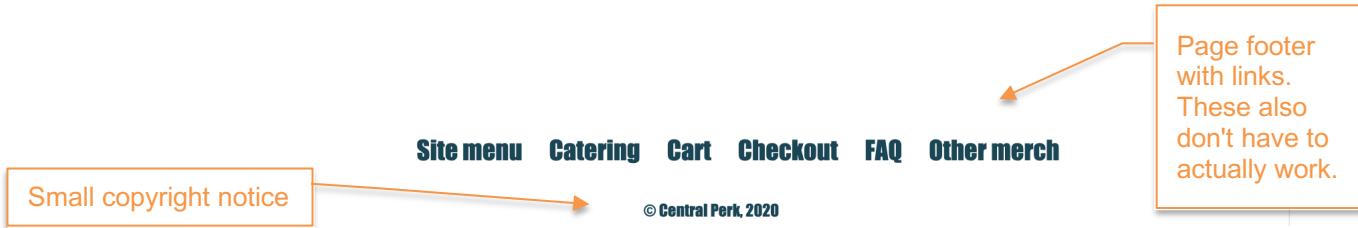
\$4.25



### Herbal tea

Our house blend of ecanacia goldenseal, dandelion, and peppermint. Zero calories, zero caffeine, zero guilt, all wonderful.

\$1.25



## Setting up

1. Create a folder for the root of your web app anywhere you like, even the desktop is fine.
2. In the starters folder you will find a folder called *coffee shop*. Copy its *assets* folder into your new root folder. All the image files you need will be found there.
3. In the root folder (not in assets), create two new files, site.css and index.html.

## Creating your page

Let's create the page without worrying about styling just yet. Do not get stuck trying to style anything. We'll do that later.

4. Make sure you have the four things that every site needs: DOCTYPE, html, head, body.
5. Add a header, a main, and a footer. (Hint: semantic tags)
6. In the header, add the logo image. It will be huge. Don't worry. We'll fix it later.
7. Also in the header, add four links; menu, location, story, and order online. The links can go anywhere you like. They do not have to actually work. Give the first three a class of *topMenu* and the last one (order online) a class of *topMenuButton*.
8. In the footer, insert this:

```
<div>
  <ul class="links">
    <li class="link">Site menu</li>
    <li class="link">Catering</li>
    <li class="link">Cart</li>
    <li class="link">Checkout</li>
    <li class="link">FAQ</li>
    <li class="link">Other merch</li>
  </ul>
</div>
```

9. In the main, add a *<div>* with an id of *heroDiv* and a section with an id of *featuredItems*.
10. In your *heroDiv*, add an image with a class of *heroImg*. In the assets you'll find an image called *hero.jpg*. Make sure you put an alt attribute on that image.
11. After the image, add a paragraph with an id of *invitation* that says "Join us on the orange couch".

Now let's look at those featured items in the middle of the page.

## Adding featuredItems cards.

12. In the `featuredItems` section, we'll create four "cards" (refer to the screen shot above). Each card will hold a food item. You'll make one and then copy/paste it three more times. Do the first one like so:
13. Make a div with a class of `foodCard`.
14. In that div, add an image with a class of `foodImg`. Set its `src` to one of the food pictures in the assets folder.
15. Add an `<h2>` with a class of `foodTitle`.
16. Add a `<p>` to describe the food item. Make it say whatever you want for now.
17. Add a `<p>` with a class of `foodPrice` and put in any price.
18. Now, copy that div and paste it four times. For each of these four divs, change the image source, the `foodTitle`, the description, and the price. You can either use your imagination for those details or you can use the data from the screenshots above. Heck, put in nonsense if you like. We just need something to style later.
19. Take a deep breath and take a look at your page. If you have everything, let's move on to styling. Ready?

You'll be setting all of your styles in `site.css`, keeping your styles separate according to best practices.

## Setting global styles

20. Create a `:root {}` style. Put in these four custom properties:  
`--dark-color: rgb(26, 68, 84);`  
`--dark-color2: rgb(190, 48, 27);`  
`--light-color: rgb(255, 255, 255);`  
`--light-color2: rgb(225, 200, 175);`  
We'll be using them later.
21. Add a style for the body. Set the color to `--dark-color`, the background-color to `--light-color`, and the font-family to "Impact, Arial, sans-serif".
22. Run and test. Make sure the colors and fonts have changed properly.

## Styling the header menu

23. Resize the logo to have a width of 250px.
24. Put all of the header menu items on the same line however you want. (Hint: flexbox would be a great choice here).
25. Give the header itself a background-color of `--light-color`.
26. Make it attached to the viewport by giving it a position of fixed and a top of 0.
27. Make it stretch across by giving it a left of 0 and a right of 0.
28. Set its z-index to 1. This will ensure that it is always on top.
29. Notice that it occults the other things on the page so add a margin-top of about 200px to the main.
30. Bonus!! You can make it align nicely with `align-items: center`.
31. Make the links look sort of like the screenshot by setting their font-size, color, and margin. You don't have to get it perfect.

32. Bonus!! To remove the underlines in links, give them a style of text-decoration: none.
33. For the last link that looks like a button, give it a background-color of --dark-color, a color of --light-color, and a padding of 20px all around.
34. Run and test. Let's move on to the footer.

## **Styling the footer**

35. Turn off the bullets in the list with list-style-type.
36. Make the links be side-by-side using any method you like; float, inline-block, flex, or grid.
37. Give each link some padding or margin so they'll spread out.
38. Make them 20% larger with font-size.
39. Now for the copyright. Let's make it 25% smaller with font-size. (Hint: use ems and use a number less than one).
40. Center the copyright.

## **Styling the hero image**

41. The hero image is the huge picture that draws the eye. In our case, it is the photo of the coffee shop.
42. Make it always be the full width of the viewport. (hint: use vw as your measure. Or % will do the job as well).
43. Style your invitation paragraph to be four times the size of a normal font.
44. Bonus!! If you have extra time at the end, make the invitation overlay the heroImg like in the screen shot above. (Hints: put a position: relative on the heroDiv, position: absolute on the invitation, set the paragraph's location with top, left, or right, and change the color to light).

Lastly, let's turn to the featured items

## **Styling the featured items**

Remember, we're going to turn our <div class="card"> into what looks like a card with food details on each.

45. Give each foodCard a width of 300px. The image will still be way too big, so resize the image to be 300px by 300px.
46. Give each foodCard a margin of 50px.
47. Make the foodTitle bigger with font-size.
48. Make the foodPrice bigger and bold. You can align it to the right with text-align: right.
49. Run and test. It should look great except that the food cards are still laid out above-and-below. It would be nice to have them side-by-side.

## Laying out the food cards

The food cards should be side-by-side as long as there is room. When we run out of room horizontally the card(s) that don't fit should be pushed down to the next line.

50. Make that happen by either floating them or using flexbox or grid and making sure that they'll wrap to the next line if there isn't enough room horizontally.

## Refactoring the CSS selectors

Throughout this site you've been putting in classes and ids but these are unnecessary. Instead, you can use advanced CSS selectors.

51. Go back through the HTML file and the CSS file. Refactor them by removing the class="whatever" from the HTML and change the CSS file to point to your new class-less elements. For instance, where you put `class="link"` on the `<li>`s above, you can remove that class from the HTML and change your CSS selector from this ...

```
.link {  
    to this ...  
.links li {
```

52. Do this everywhere you think removal of a class or id makes your HTML and CSS better designed.

## Submitting the site

53. Please get these files to your instructor for evaluation. If you haven't moved the assets folder or added any other supporting files, you should be able to just email the html file and the css file to the email address he or she has provided to you.