

HTML / CSS Worksheet 3

This week's worksheet we will look at:

CSS

linking a CSS stylesheet

selectors

styling elements

defining fonts

border,
padding

flexbox (an intro)

linking html sheets
together

adding a second
stylesheet

Last week we made an html only document and populated it with some content. Today we'll make it look pretty, kind of. You will also download a pre-populated html file and you will style it step by step for an intro to Flexbox. This is a large worksheet, so just take it one step at a time.

Motivation? You get a week off next week.

****Note** in order to succeed, read the ENTIRE STEP FIRST, THEN code everything by hand or by direction. It's the only way to be sure.**

Have you ever made a pizza from scratch?

Yeah, but have you really thought about it?

Like, how if you put the sauce on first, before you make the base it doesn't work?

Or, if you put the cheese on then the sauce it's messed up too.

Making a pizza is similar to coding. There are certain things we have to do first. Like making the base, letting it rise, mould the base out, add the sauce, add the toppings, add the cheese, bake, and cut and serve. Not all pizzas end up the same, but they do start out in the same way as long as you follow the order correctly. You can have everything you need to make many different pizzas using the same base structure that is set up before hand.

This is how you make a pizza

There are some specific steps you need to take to make a pizza. We will list them here.

- I. make the pizza base
- II. add the tomato sauce
- III. add the toppings
- IV. add the cheese
- V. bake
- VI. cut
- VII. serve

The toppings are different; they change for every occasion and taste. There is no order to toppings. Do we will do them like this.

• Pepperoni
• olives
• peppers (never green)
• mushrooms
• onions
• jalapeños
• pineapple (only if you are crazy)
• bacon

Here's the [WIKIMEDIA](#) article about pizza.

But wait, there's something I like even more than pizza. *It's cake.*

I love cake. I'm not afraid to shout it to the mountain top.

That's right. I love all kinds of cake. Chocolate, vanilla, cherry, and fruit based. If there is extra frosting or marzipan, all the better. I would happily run a 5k if there was cake at the finish line.

To show my love of cake, I'm going to put even more images in here but this time, they will be rectangular! (like a poster)

Here are six different versions of cake and cake-like products. I would happily eat them all. Macarons are different than macaroons, and are considerably more difficult to make from scratch. They were all the rage in Montreal when I lived there. As an aside, these images are proportionate to an A3 piece of paper. Handy to know for one of your tasks. Think about it.

An conclusion note. Learn it. Alas, it is this easy to plan and make a website with images of products and some feedback from your audience. That's why you can do this. Don't take a break or think you're finished. You deserve to work on the shiny parts.

1

Open VSC. Go to File > Open Folder then navigate to **last weeks worksheet folder** on your desktop.

We will be working from within last weeks folder / repository. If you haven't finished [week 2 worksheet](#). Go back and finish up. It's no problem, we'll wait.

This should bring up your entire folder in VSC.
If you can't see it, make sure you click on the double page icon in the top left of VSC and then the folder list should show up. (I have some invisible files showing up, if you don't see those, don't worry. ie .vscode / .gitattributes)

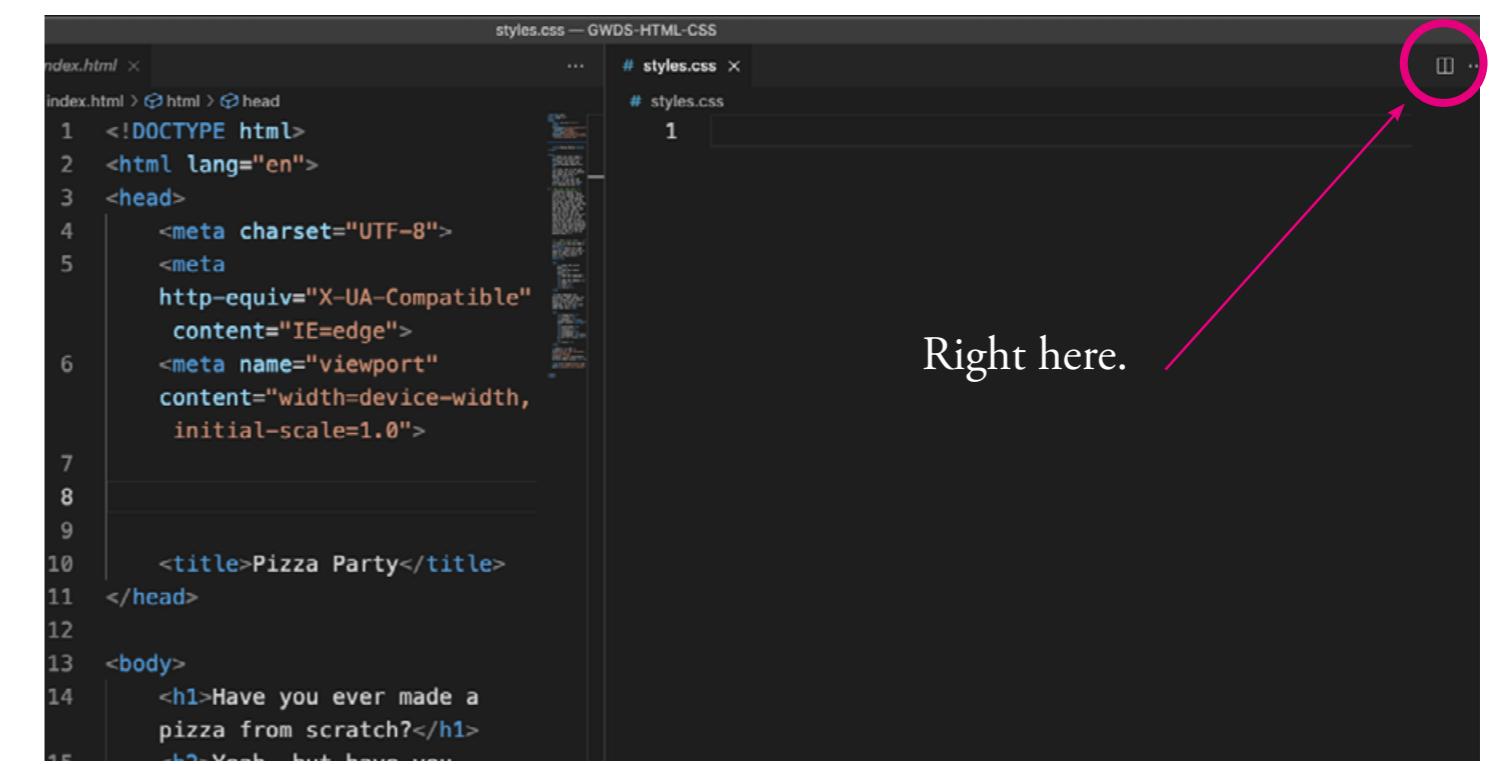
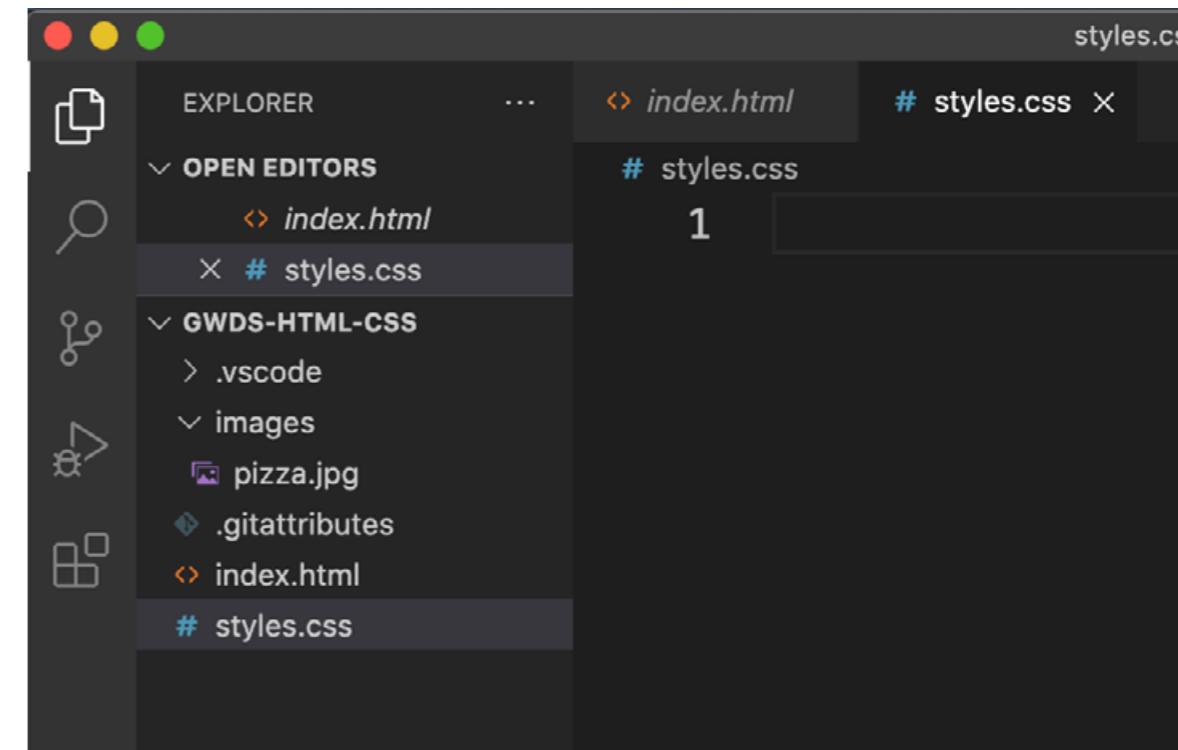
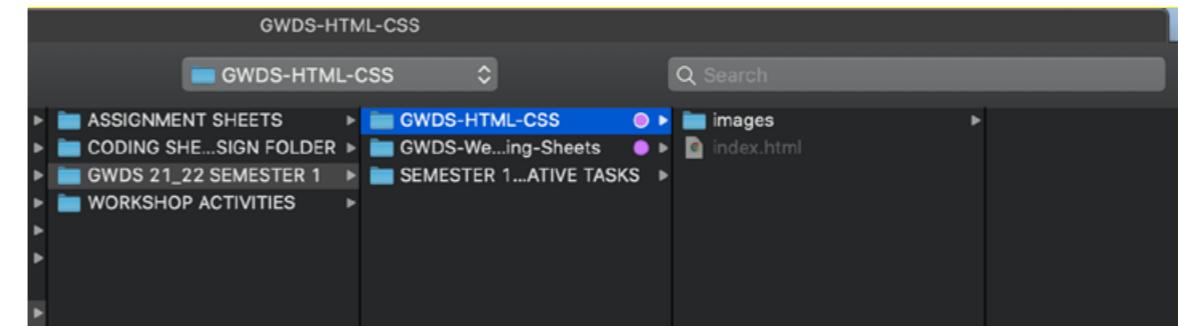
Now in VSC go to File > New File and **save** a new file named **styles.css**

Make sure in the format drop down you declare it as a **CSS file**.

Navigate to the folder where your index.html file is and save it in there. It should now show up in your VSC list.

In VSC, You can split your screen vertically, so you can see both files at the same time. The picture on the right shows you how to do it. This is extremely helpful when looking at how your html and css interact together.

In GitHub Desktop, **commit** and **push**.



2

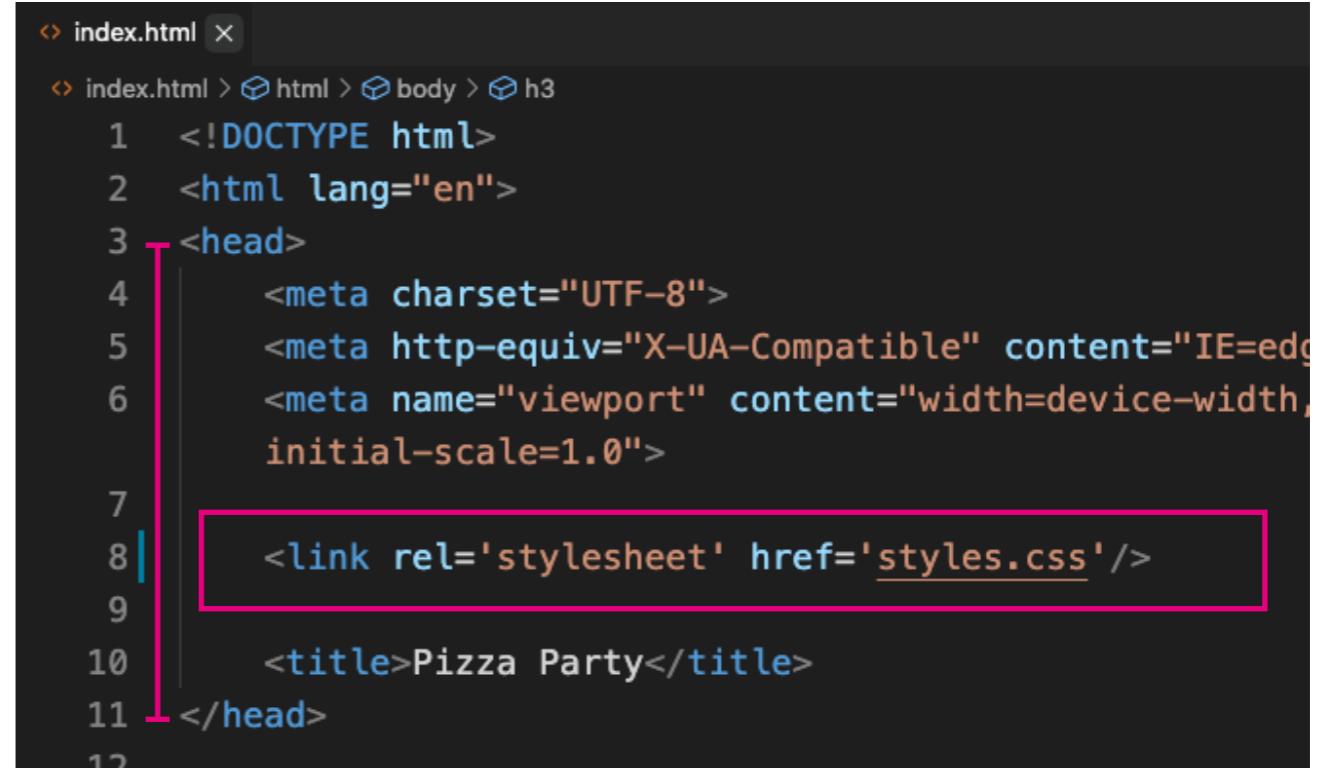
In your index.html file inside the `<head></head>` tag type in the code that links your styles.css to your html file. **Save**. In GitHub, **commit**, and **push**.

Before we move forward, let's test to see if your link works.

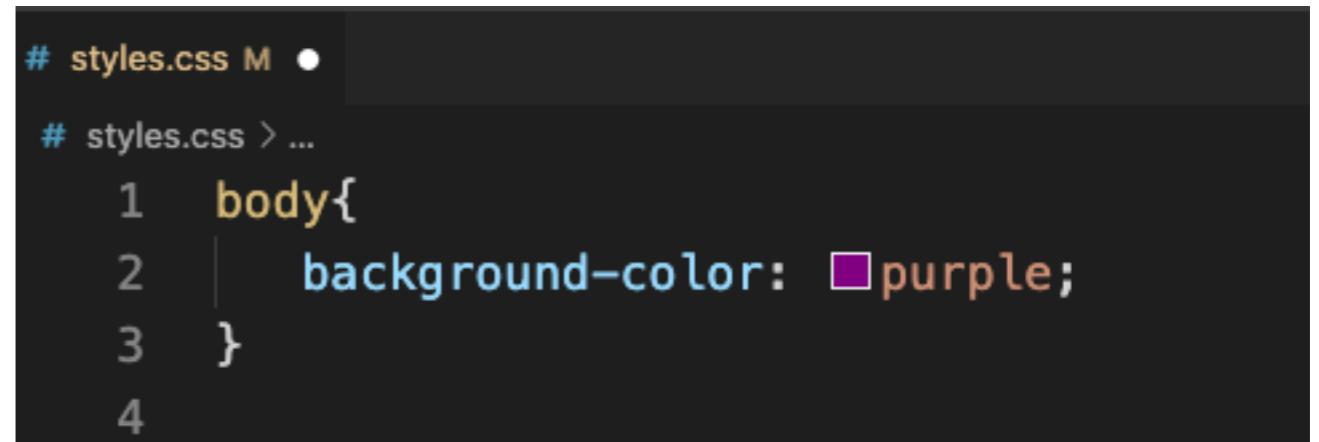
In your new **styles.css** file type in the following in the second image. After you type in your first curly bracket, hit return, and VSC will format it for you. Then continue typing in between the two brackets.

If you double click on your index.html file in your local folder your background colour should be purple. So what exactly is going on here?

- 1) When we added the styles.css link inside your index.html we created a link between your html file and your css file. They now can talk to each other.
- 2) We added a style to the `<body></body>` tag, so everything inside the `<body></body>` in your index.html will be styled. In this case the **background-color** is purple.
- 3) CSS always has this format. Curly brackets, colon, semi colon, then closing curly bracket.
- 4) **You can read about styles in css [here](#)**. Because we are only touching of some of power of CSS, I recommend you have a look at what you can do.



```
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
7
8   <link rel='stylesheet' href='styles.css'>
9
10 <title>Pizza Party</title>
11 </head>
12
```



```
# styles.css M ●
# styles.css > ...
1 body{
2   background-color: purple;
3 }
```



3 Let's add a few more styles below your body style to your styles.css and see what happens. Type in what you see to the right into your styles.css page After each image, **save** and refresh your local browser page.

```
p {  
    font-family: monospace;  
    text-align: center;  
    color: #lightpink  
}
```

What do you see?

What we did here was lump up styles, so they will be applied together. You can start with applying, say your `<h>` tags uniformly. Now from `<h1><h2><h3><h4><h5><h6>` they are all styled the same. Also your `<p>` (paragraph) tags should have a monospace font, be centered, and be light pink.

Remember to **commit** and **push**.

4 I want to make my `<h1>` tag and my `<h2>` tag different from all the other `<h>` tags we just styled. So let's do that now. **Save**. Refresh your local browser and have a look.

```
h1,h2,h3,h4,h5,h6 {  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    color: #lightyellow;  
    font-style: italic;  
    text-align: right;  
}
```



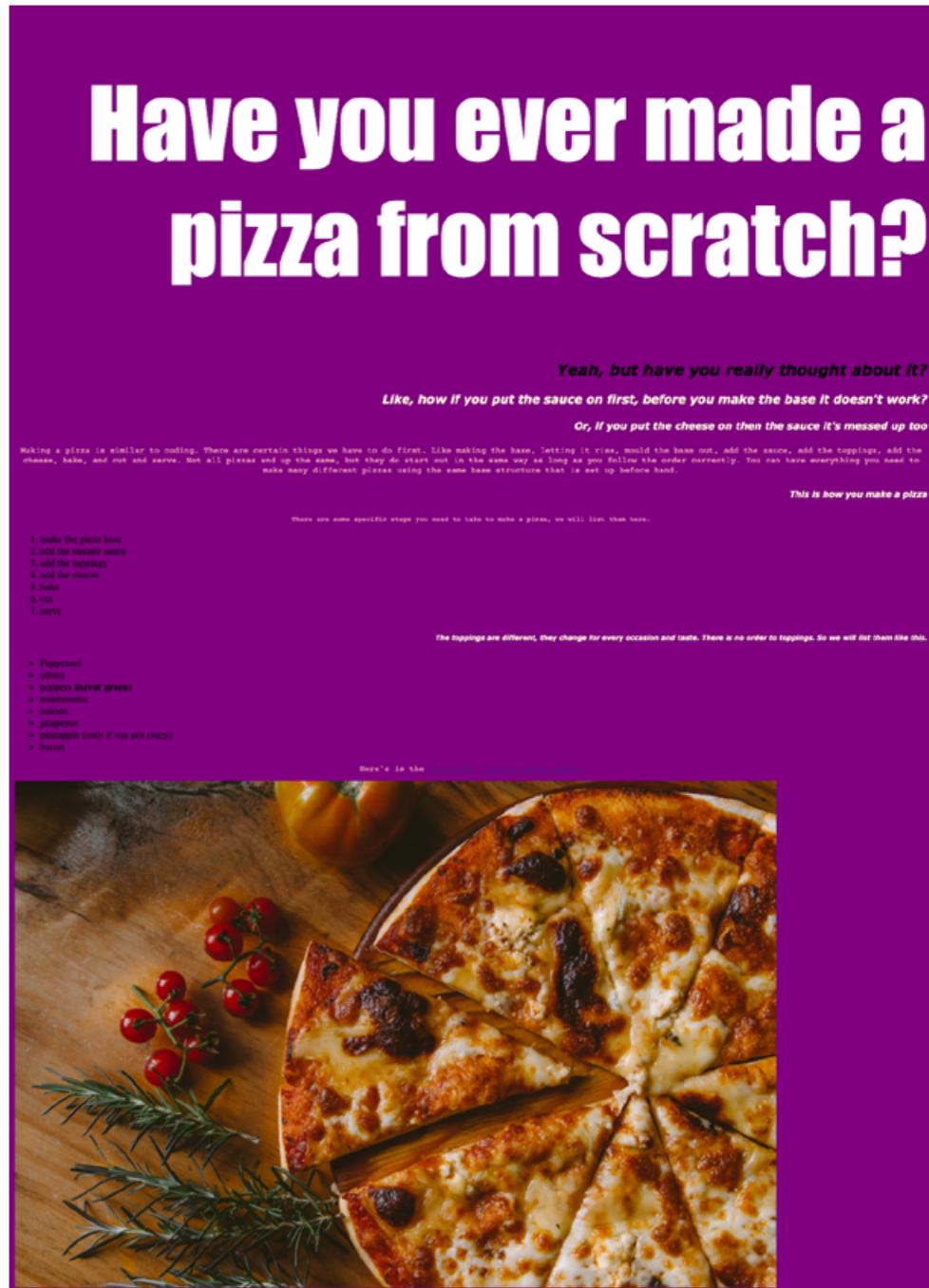
```
h1 {  
    font-family: Impact, Haettenschweiler, 'Arial  
    Narrow Bold', sans-serif;  
    font-size: 150px;  
    color: #white;  
    font-style: normal;  
}
```

```
h2 {  
    color: #black;  
}
```

Because the stylesheet “cascades” down, we can “over-ride” the css code as we move down the styles.css page. This is helpful as we get more and more specific with what we want our style changes to be.

Question: what happens when you change the order? If you moved the grouped **h** styles below the **h1** and **h2** in your styles.css?

commit and **push** in GitHub Desktop.



5

We will now add what is called a **class selector**. A class selector allows you to apply a style to different tags. We will add one at the bottom. When you create a class selector you always use a **.** in front of what you name it. It can be named anything.

Now we will head over to our **index.html** file. Add this to the paragraph about how making pizza is similar to coding, just like this:

Save. What happened?

Because our **.special** is placed below our **p** in the cascade of the document, the **font-family** and **text-align** is overridden by our **.special**.

Now go to your **<h5>** in your **index.html** file

**<h5>This is how you make
a pizza </h5>**

and add the **.special** just the same as you did with the **<p>** above but attach it to **<h5>** (don't change it to a **<p>** tag).

Something like a class selector can be used on different tags to create visual consistency, or to reach a specific design goal.

Commit and **push** in GitHub Desktop

```
.special {  
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri,  
    'Trebuchet MS', sans-serif;  
    font-weight: bold;  
    font-size: 40px;  
    text-align: left;  
}
```

```
<!-- Add some copy here -->  
<p class="special">Making a pizza is similar to coding. There  
are certain things we have to do first. Like making the base,  
letting it rise, mould the base out, add the sauce, add the  
toppings, add the cheese, bake, and cut and serve. Not all  
pizzas end up the same, but they do start out in the same way  
as long as you follow the order correctly. You can have  
everything you need to make many different pizzas using the  
same base structure that is set up before hand.</p>
```

```
p {  
    font-family: monospace;  
    text-align: center;  
    color: lightpink  
}  
  
.special {  
    font-family: 'Gill Sans', 'Gill  
    'Trebuchet MS', sans-serif;  
    font-weight: bold;  
    font-size: 40px;  
    text-align: left;  
}
```

```
h1, h2, h3, h4, h5, h6 {  
    font-family: Verdana, Geneva, Ta  
    color: lightyellow;  
    font-style: italic;  
    text-align: right;  
}  
  
.special {  
    font-family: 'Gill Sans', 'Gill  
    'Trebuchet MS', sans-serif;  
    font-weight: bold;  
    font-size: 40px;  
    text-align: left;  
}
```



6

Next we will style our lists. Type what's in the images and **save**.

And we'll make a class selector in CSS for our favourite ingredients. Highlight a few of your favourite ingredients in the html by adding a

```
<li class="fave">
```

in your index.html file and **save**.

Commit and **push** in GitHub.

```
ul {  
    list-style-type: square;  
}
```

```
ol {  
    list-style-type: upper-roman;  
}
```

```
li {  
    color: white;  
    font-size: 20px;  
    font-weight: bold;  
}
```

```
.fave{  
    color: orange;  
}
```

- I. make the pizza base
- II. add the tomato sauce
- III. add the toppings
- IV. add the cheese
- V. bake
- VI. cut
- VII. serve

- Pepperoni
- olives
- peppers (never green)
- mushrooms
- onions
- jalapenos
- pineapple (*only if you are crazy*)
- bacon



7

Though I admire the alignment of the pizza in this photograph, and how it works with the rule of thirds, it's too big in our web page and needs to be a bit smaller. Instead of going into Photoshop and guessing how much smaller we want it, we can do that in CSS. Add the class selector in CSS, then attach it to your `img` in the html page. **Save** and refresh your local web page. You can change the % to whatever you like. You can also set a fixed pixel size too. Whatever works for your design. Try it now.

Commit and **push** in GitHub Desktop.

8

One more thing before we head over to our populated index.html file, let's add a link. At first won't go anywhere, but will soon.

Go to your index.html file, and add the following `<h2>` below your pizza image, but before the `</body>` tag. **Save**.

You have a link now, but it's hard to read. Your other wiki link is hard to read too. Let's fix that using CSS!

Commit and **push**.

```
.pizza{
```

```
    width: 50%;
```

```
}
```

```

```



```

```

```
<h2>But wait, there's something I like even more than pizza. <a href="cake.html" target="_blank">It's cake.</a></h2>
```

```
    < a {
```

```
        color: white;
```

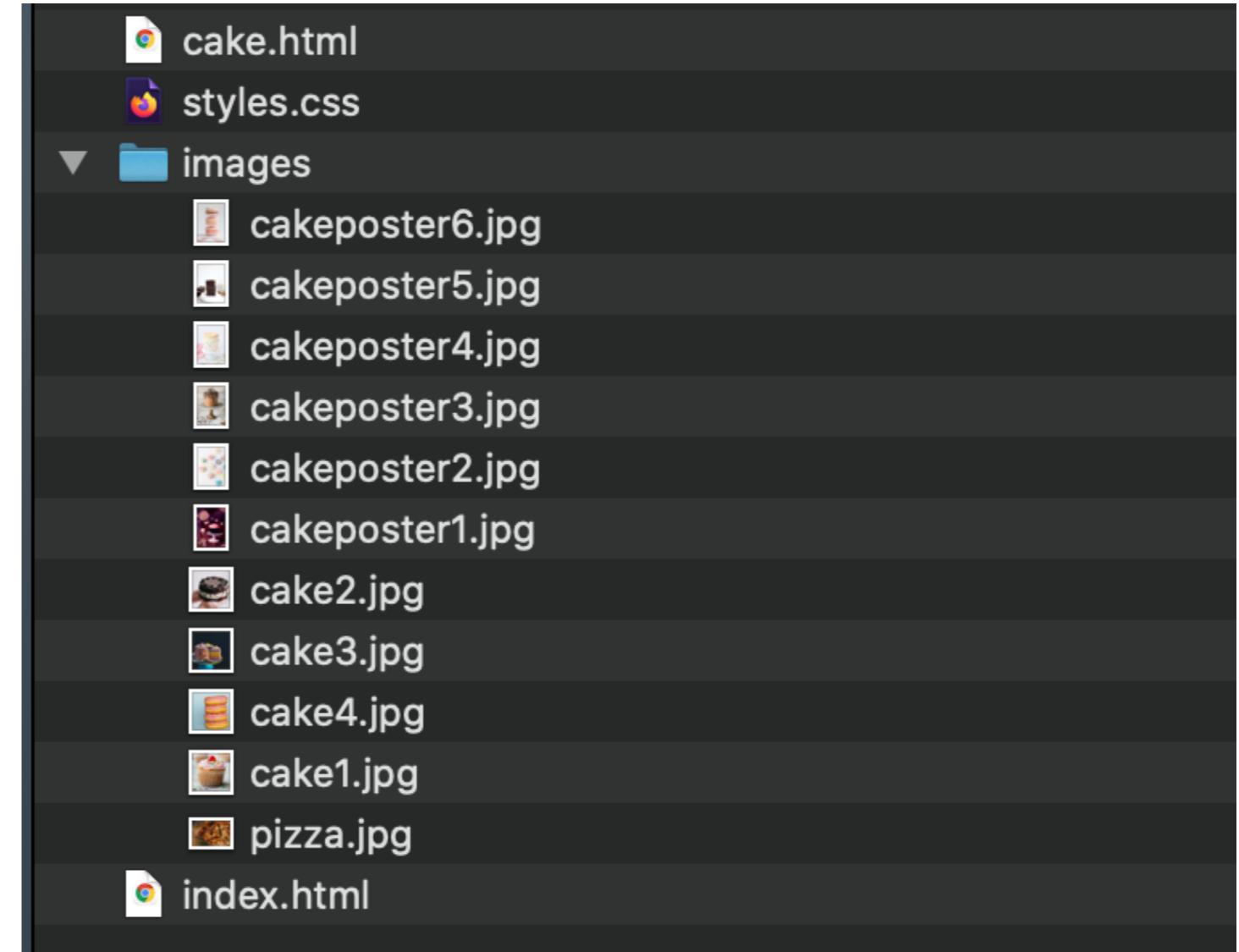
```
}
```



9

At the moment, the link goes nowhere. So download the [following zip](#) and put the files and photographs as shown in this image. Make yours look like mine.

Commit and **push** in GitHub Desktop.



10

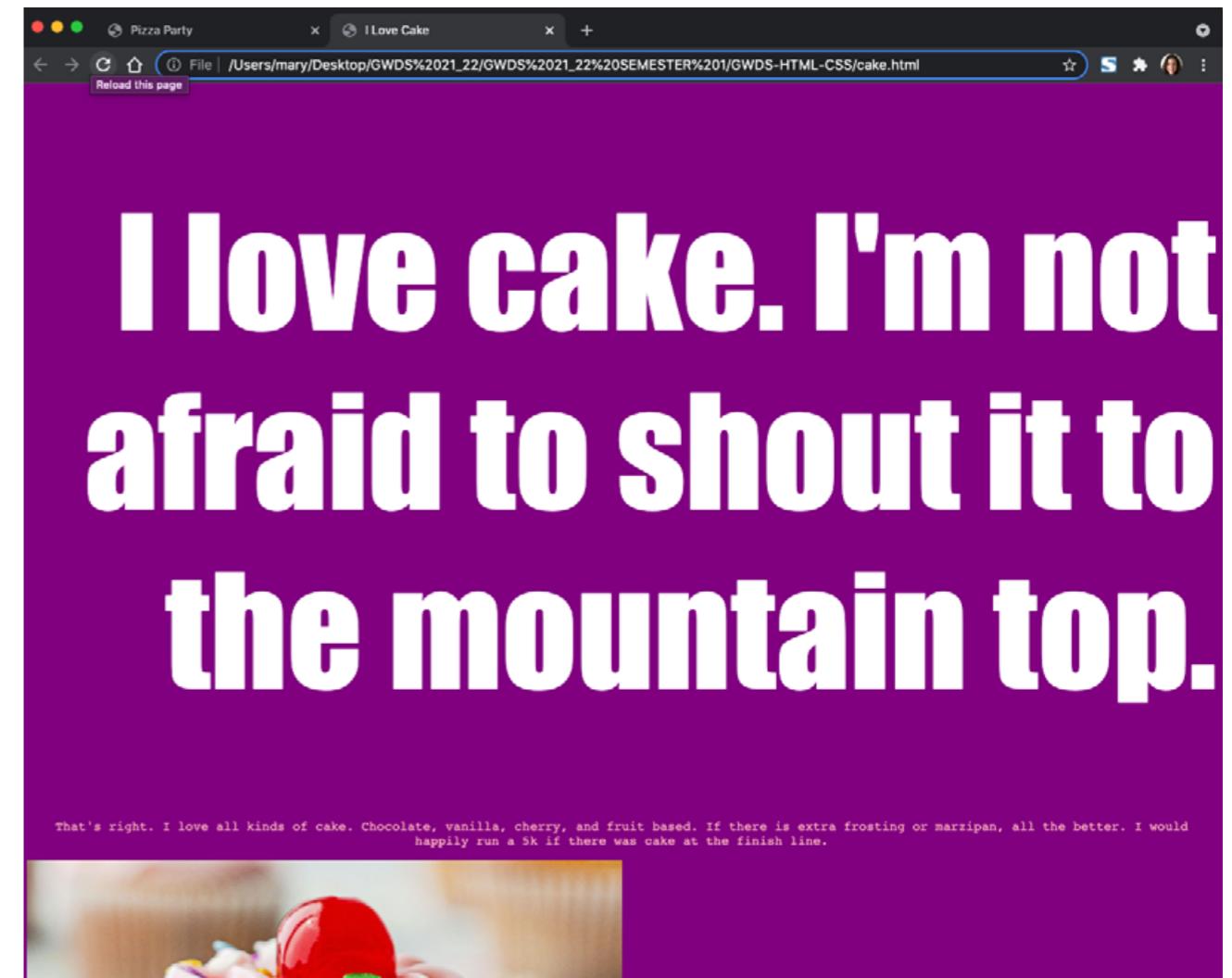
Yikes, that's kinda hokey. And BIG. Never fear, that is what our CSS file is for. As you look at the html I have placed inside, you'll see some stuff may have not seen before. That's because this html file uses the power of FLEXBOX.

Let's link our styles.css. We will do it in the same way we did for the index.html document.

Save.

The **styles.css** styles we made are showing up here. But I don't want to make it look the same, I want some of it to look different.

```
cake.html x
cake.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9      <link rel='stylesheet' href='styles.css' />
10
11     <title>I Love Cake</title>
12 </head>
13
14 <body>
15
```



11

After the last style you created in your styles.css add a **comment**. Like the html comments, your browser won't display them, but it is like a note for your future self. You can also use these to turn on and off styles to see what the heck is going on. It's a good way to organise your css file and know where things are. They are styled like this.

```
color: white;  
}  
  
/* Cake page */
```



12

Here is our `<p>` style. While that is fine on our pizza page, let's make this page a little nicer. Right under our comment, in our **CSS file**, put this code in. **Save** and refresh your local browser.



```
.cake-text {  
    font-family: Verdana, Geneva, Tahoma,  
    sans-serif;  
    text-align: left;  
    color: white;  
    font-size: 25px;  
    font-weight:bold;  
}
```

to the mountain top.</h1>

<p class="cake-text">That's right. I love
all kinds of cake. Chocolate, vanilla,
cherry and fruit based. If there is extra

Now when you refresh you will see something a bit different.

Now that text is a bit bigger and a bit bolder. Which is exactly how I want all my body copy. But now all my `<p>` tags are really big, and I might want a few options. We will clean that all up in time.

Commit and **push**. (Has it become a habit yet?)

**That's right. I love all kinds of cake. C
based. If there is extra frosting or ma
happily run a 5k if there was cake at t**



In my cake.html file structure, I know I have a **header**, **two sections and a footer**.

The `<header></header>` is above, outside of the `<main></main>` tag.

The two `<section></section>` tags are nested together, one after the other, inside the `<main>` tag.

The `<footer></footer>` is outside the `<main>` at the bottom. **Look for these in your cake.html file.**

And it's all nested inside the `<body></body>` tag.

Again, I say look for this structure in the cake.html file.
You will see it and the lines that VSC uses to connect the opening and closing tags.

13

We are going to work on the first section where we have some square images. Right under `.cake-text` we will put `.section-one`. Now you will see that the whole of the section has a white border, and a slate blue background, with a bit of the purple of the body on the outside, because the section is nested inside the body.

Save, commit and push.

```
<main>
  <!-- begin square images -->
  <section class="section-one">
    <div class="container">
      <div class="cake-group">
        
        <p class="cake-text caption">Cupcakes are a preferred way to enjoy cake. Many versions, all in little bite sized joy. Have one, have five! They are excellent.</p>
      </div>
    </div>
  </section>
</main>
```

```
.section-one {
  border: 1px solid #white;
  background-color: #darkslateblue;
}
```

marzipan, all the better. I would happily run a 5k if there was cake at the finish line.



14

My captions are horsey. I will add a class selector on to the end before I move on.



Donuts get a bad rap, but I think donuts totally qualify as cake. Donut holes from Tim Horton's in Canada are the absolute best with a Double Double on the way to work in the snowy North.

So now my captions have the cake-text style on them and they are smaller and italic. Keeping in the family, but still different in its hierarchy.

```
<div class="cake-group">
  

  <p class="cake-text caption">Cupcakes are a preferred way to enjoy cake. Many versions, all in little bite sized joy. Have one, have five! They are excellent.</p>
</div>
```

```
.caption {
  font-size: 15px;
  font-style: italic;
}
```



Donuts get a bad rap, but I think donuts totally qualify as cake. Donut holes from Tim Horton's in Canada are the absolute best with a Double Double on the way to work in the snowy North.



15

I want my photographs aligned horizontally, and if the web page gets too narrow (phone) then they will stack. **Enter flexbox.** Flexbox is huge, and we will look at it further, but right now I want you to just try a few things out. A few things to understand:

Flexbox requires a container. You will see that both our sections have the `<container>` tag.

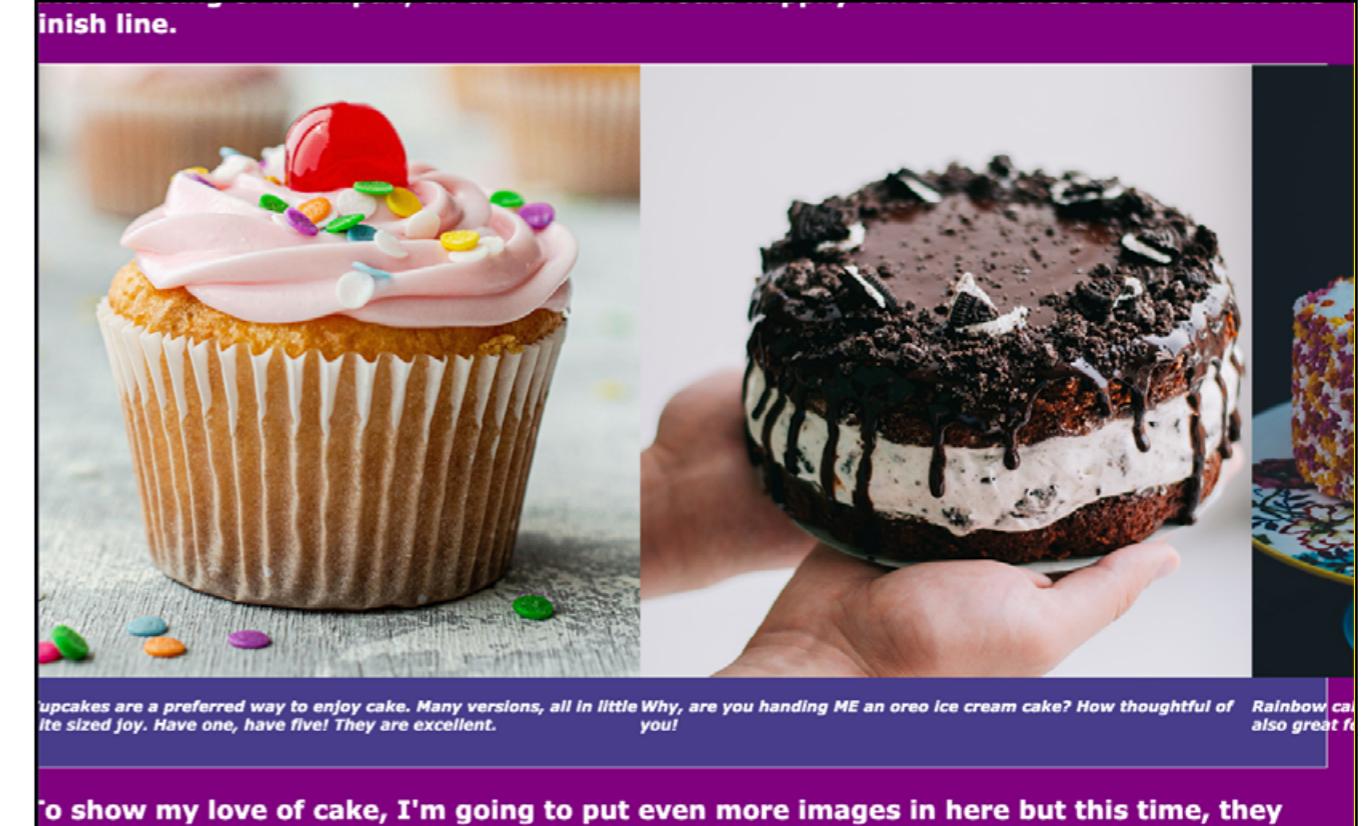
Inside that container it will have **flexbox properties**. You can read much more on Flexbox [here](#). And another great resource [here](#). And I recommend you look at both..

Add the `.container` style and **save** and refresh. We are lined up, but things aren't just right yet.

Save, commit and push.

```
<!-- begin square images -->
<section class="section-one">
  <div class="container">
    <div class="cake-group">
      
      <p class="cake-text caption">Cupcakes are a preferred way to enjoy cake. Many versions, all in little bite sized joy. Have one, have five! They are excellent!</p>
    </div>
  </div>
</section>
```

```
.container {
  display: flex;
}
```



16

Before we fix the images, we are going to look at how the html is grouped and nested within each square image. All 4 images have this.

```
<div class="cake-group">  
    
  
  <p class="cake-text caption">Cupcakes are a preferred way to  
  enjoy cake. Many versions, all in  
  little bite sized joy. Have one, have five! They are  
  excellent.</p>  
  
</div>
```

You will notice that the `<cake-text caption>` we looked at earlier is in there, but there is also `<cake-group>` and `<cake-item>` too.

`<cake-group>` wraps around the image and caption for all 4 square images.

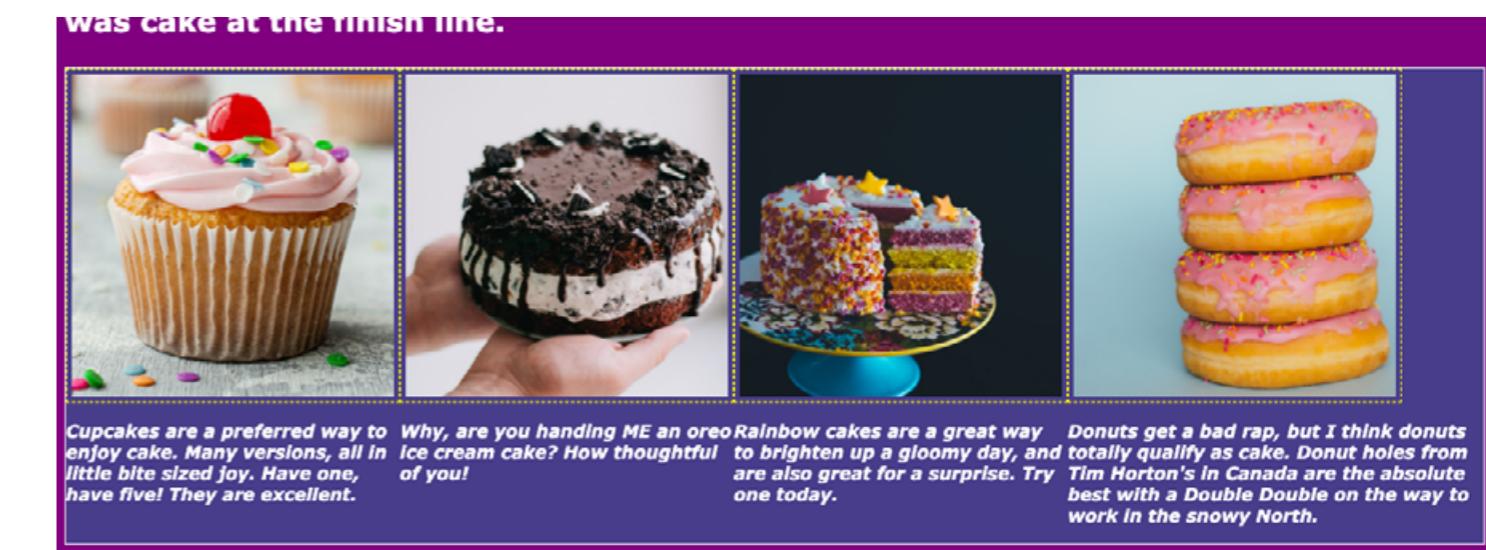
`cake-item` deals only with the image inside the group.

There's a `<div>` class, like the `<container>`.

These are called a Content Division element. It also acts as a container, or like an empty vessel to hold a specific style of your choosing. More on that [here](#).

Add the `.cake-item` style

```
.cake-item {  
  
  border: 1px dashed yellow;  
  width: 275px;  
  height: 275px;  
  padding: 4px;  
}
```



Save, commit, push. We are almost there:

- you see the **yellow border** around the images
- we have confined the **width and height to pixels**
- we have put some **padding** around each image

17

Now we are going to look at `<cake-group>` and style it. **Save**. You'll see the green border around both the photograph and the text caption as well. Everything is in their box and not overflowing, but we don't wrap yet.

```
<div class="cake-group">
  

  <p class="cake-text caption">Cupcakes are a preferred way to
  enjoy cake. Many versions, all in
  little bite sized joy. Have one, have five! They are
  excellent.</p>

</div>
```

Go back to our `.container` in your css file.
Add the following:

Then pop to your `.caption` and add the following. **Save commit** and **push**. What do we have?

```
.cake-group{
  border: 2px solid green;
  width: 275px;
}
```



```
.container {

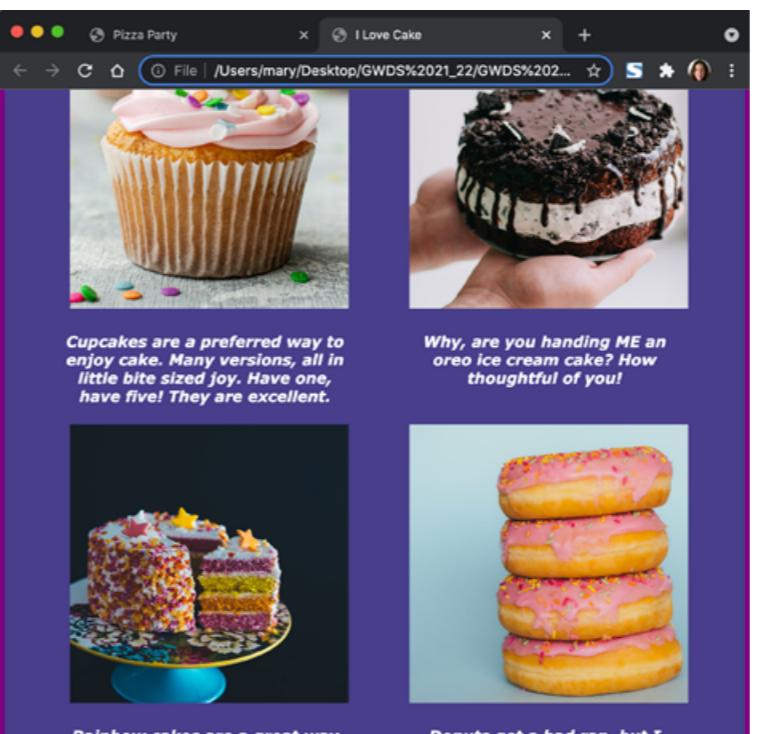
  display: flex;
  justify-content: space-evenly;
  flex-wrap: wrap;
}
```

```
.caption {
  font-size: 15px;
  font-style: italic;
  text-align: center;
}
```

If yours looks like this, then go and delete your borders you added in your CSS file. They are just for us to see what is going on.

Save and **commit** and **push**.

Then take a stretch. We have a few more things to look at before we are finished.



18

Trucking along, **now it's time for you to investigate the second part of this web page.**

I have made a second style sheet for you to [download](#). Place the file inside your folder that your index.html, cake.html, and styles.css are located. Then link it under the first stylesheet in your **cake.html** file, like you did in the beginning of the worksheet.

Every style will be commented out, and it is up to you to un-comment the style, then **save** and have a look at what changes in the second section of the website in your local browser. Don't forget to **commit** and **push** every so often.

You can attach multiple style sheets in your html files. This is handy if you want to make a global style for every page, and have separate styles for different parts of your website. Large websites will have multiple stylesheets. When you use something like Bootstrap5, it will have its own stylesheet, and you can then make a separate stylesheet to over-ride style changes of your own.

It is important to have a look at this if you want to better understand how to use CSS, stylesheets and flexbox.

```
<link rel='stylesheet' href='styles.css' />

<link rel='stylesheet' href='styles2.css' />

<title>I Love Cake</title>
```

```
# styles2.css
1
2
3 /* section {
4     padding: 8px 0px 0px 0px;
5 } */
6
7 /* .section-two {
8     padding: 0 0 10px 0;
9     background-color: deepskyblue;
10 } */
11
12 /* .cake-poster-item {
13     border: 1px solid yellow;
14     display: flex;
15     flex-wrap: wrap;
16     width: 40%;
17     width: 300px;
18 }
```



While you un-comment, follow along in your cake.html file. Check your local browser page often. Then when you are finished, remove all the borders, **save**, **commit** and **push**. Then pat yourself on the back.

```
79
80 <section class="section-two">
81
82   <p class="cake-text dark
83     center">To show my love of cake,
84     I'm going to put even more images
85     in here but this time,
86       they will be rectangular! <em>
87         (like a poster)</em> </p>
88
89 <div class="container">
90   
93
94   <img class="cake-poster-item"
95     src="images/cakeposter2.jpg"
96     alt="a photo of many cupcakes"
```

```
6  /* styles all sections */
7
8  section {
9    padding: 8px 0px 0px 0px;
10 }
11
12 /* styles only section 2 */
13
14 .section-two {
15   padding: 6px 2px 2px 8px;
16   background-color: #deepskyblue;
17 }
18
19 /* designating a % for width for an
20   image allows a rescale when web browser
21   moves */
22
23 .cake-poster-item {
24   border: 1px solid #yellow;
```