Intro to HTML & CSS

Setup & Schedule

1. Download Text editor: sublimetext.com
2. Download and unzip the class files https://github.com/WomenWhoCodeNYC/HTML-CSS/archive/master.zip
3. Download Chrome: chrome.google.com
4. Sign up for Github at github.com
5. Download SourceTree at sourcetreeapp.com

10:30am–12:30pm:
Learn HTML
12:30pm-1:30pm:
Lunch Break
1:30pm–4pm:
Learn CSS & GitHub

Hi I’m Cristiana Yambo

twitter: @CristianaYambo

Thanks for Coming!

WOMEN WHO CODE

Women Who Code

- an international non-profit dedicated to inspiring women to excel in technology careers
- 42 networks in 15 countries
- NYC network started in March 2014 - currently has ~2500 members!
- We've enabled females to achieve full-time development jobs, be introduced to startup companies, and a range of other accomplishments!

Why WWC

- Only 20% of computer science degree grads are female
- LARGER PROBLEM is 56% of females already in the tech industry are exiting mid career (35-40yr), which is DOUBLE the rate of males

SO Women Who Code is dedicated to supporting females already in the tech industry+beginners looking to transition into tech

+ or learn some tech skills (you!)

WWC NYC

Started last March, already at nearly 3,000 members
twitter.com/womenwhocodenyc

WWC NYC Events

- Weekly Front-End Study Group (JS + JS Frameworks)
- Monthly HTML + CSS Saturday Workshop
- Monthly Hacknight
- Monthly Algorithms
- Quarterly Lightning Talks
- Quarterly Volunteer Dinner
- Propose an event or event series!

Meet your TAs

… and let’s

introduce YOU!

Quick Survey

How advanced are you?

Download The Course Files

If you did not download the course files get them here

https://github.com/WomenWhoCodeNYC/HTML-CSS/archive/master.zip
(https://goo.gl/g09G8G)

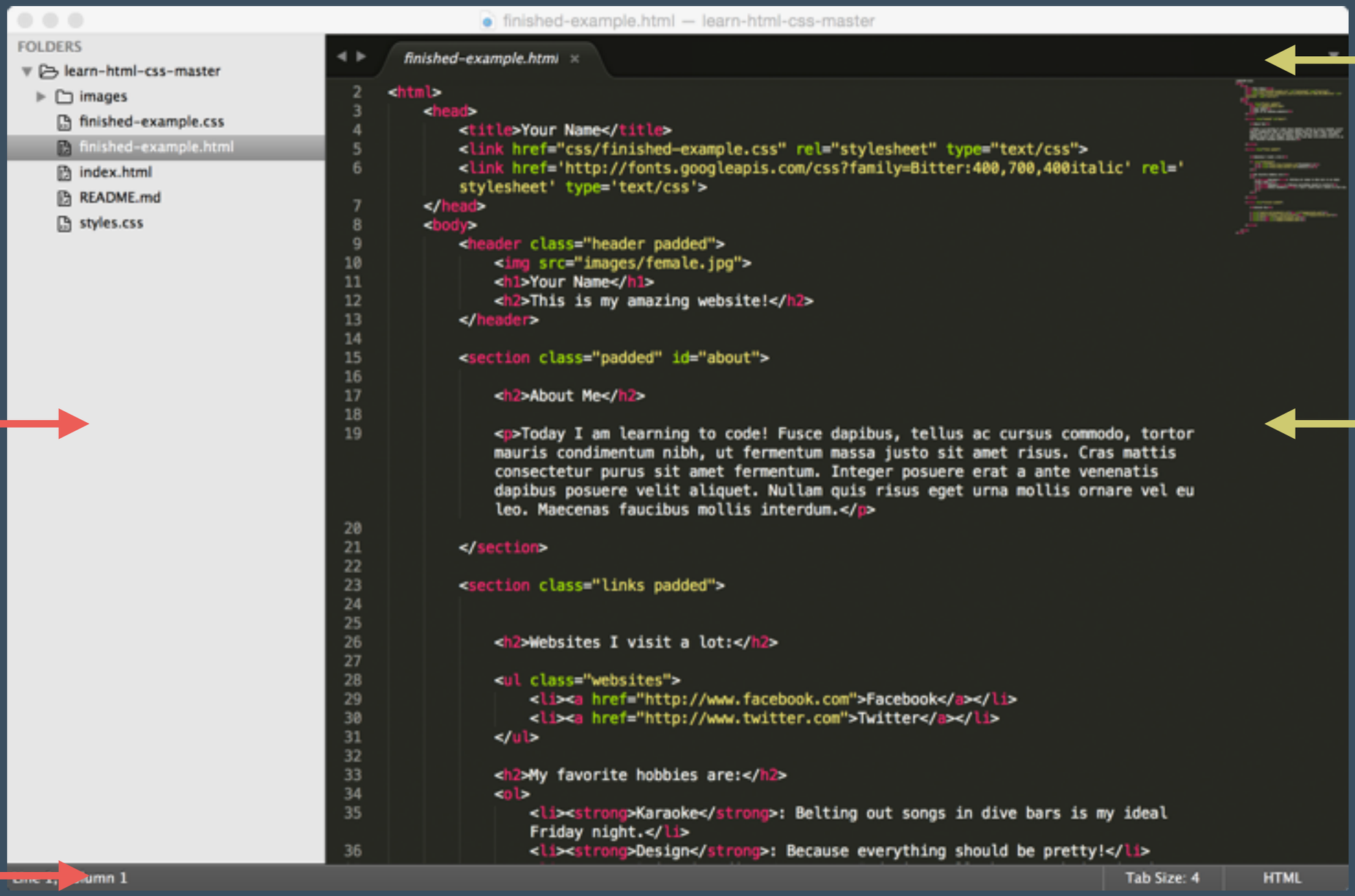
Download the zip file, open it and move the folder to your Desktop

Tools That We'll Use

Browser - Google Chrome
Used for viewing and debugging our site

Text Editor - Sublime Text
Used to edit HTML and CSS files

Quick Sublime Overview



An Introduction to Web Applications

What Makes This Possible?

Front End

HTML

+

CSS

+

JS

Browser

Back End

PHP

Ruby

Java

Python

Database

SQL

Server

We are going to learn HTML & CSS

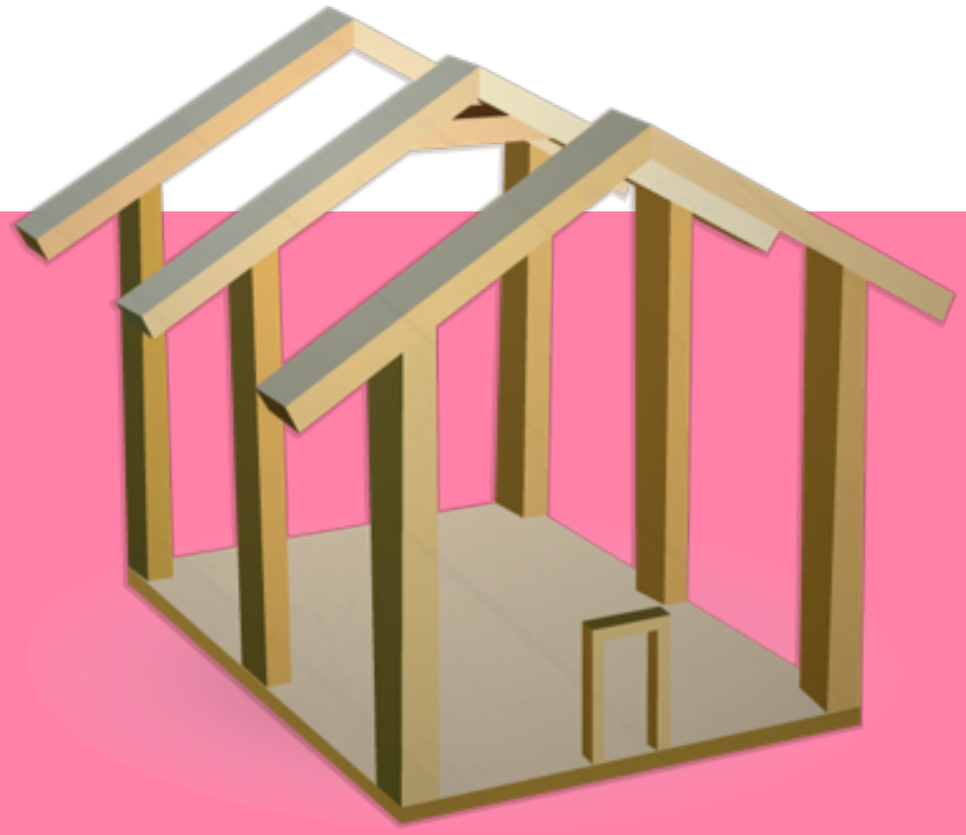
HTML = Hyper Text Markup Language

- Used for page structure
- The elements on the page: headings, paragraphs, links, images

CSS = Cascading Style Sheet

- Used to control the page design and layout
- The style of the page: colors, sizes, fonts

Think of a Web Page Like A House



The HTML is the frame
and
The CSS is everything else

So in a house, you can repaint the walls, and move furniture around without changing the frame. You can do the same with HTML and CSS

CSS Zen Garden is a great resource to demonstrate the flexibility of CSS

Let’s Get to Work!

- Open Sublime
- Go to ‘File’ -> ‘Open’ then choose the unzipped class files
- Open the ‘index.html’ file in Sublime and delete everything
- Open ‘index.html’ in your browser

Learn HTML

HTML - Hyper Text Markup Language

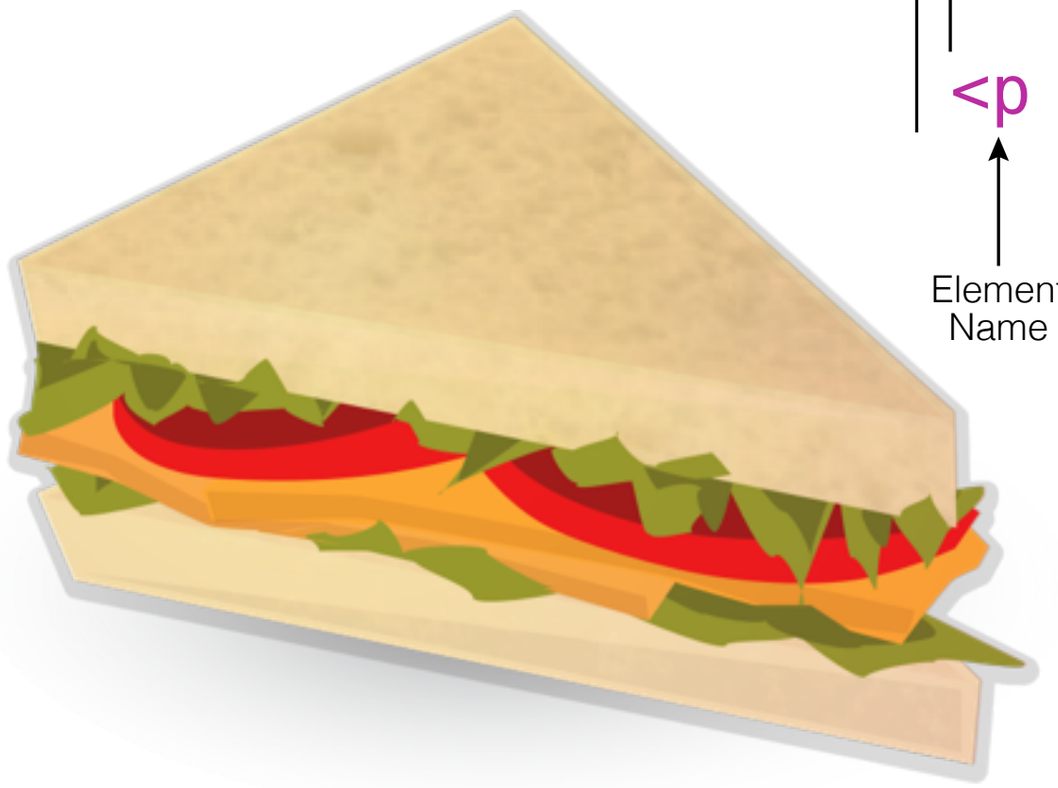
HTML is made up of elements which consist of tags

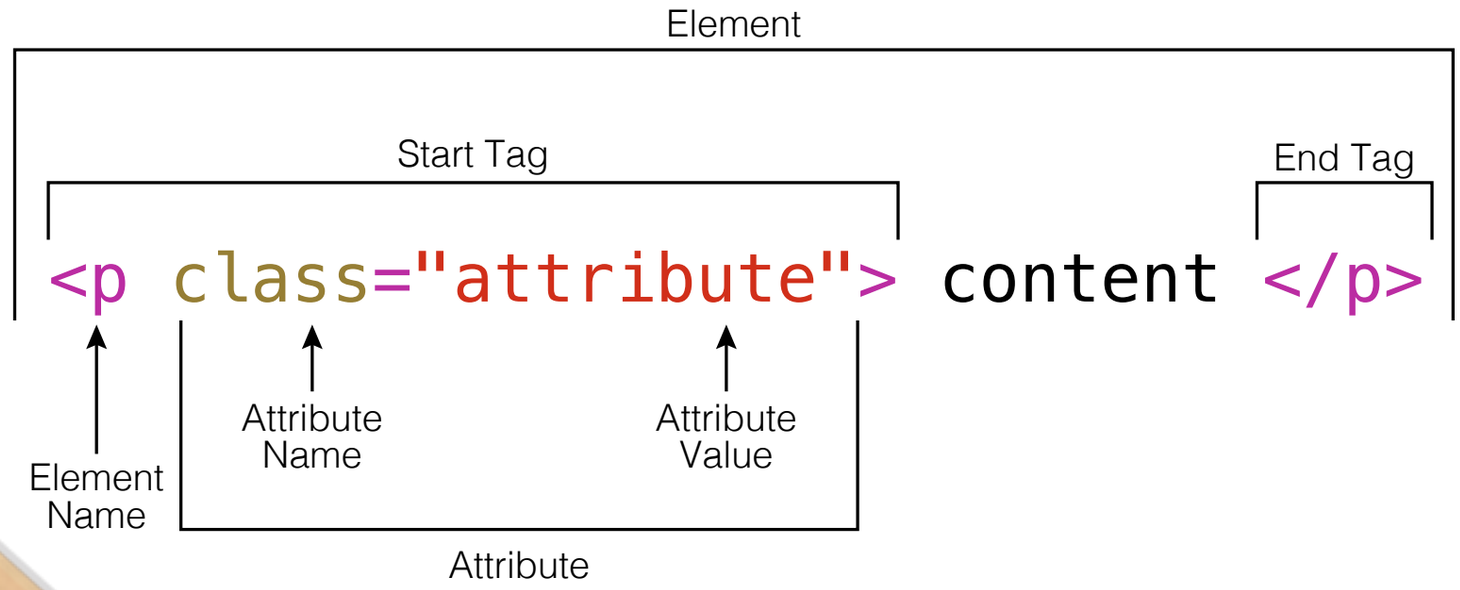
HTML elements are the building blocks of web pages

```
<h1></h1>
<h2></h2>
<p></p>
<strong></strong>
<em></em>
<ul></ul>
<img />
```

HTML Elements

Has an opening tag content, and a closing tag, like a sandwich

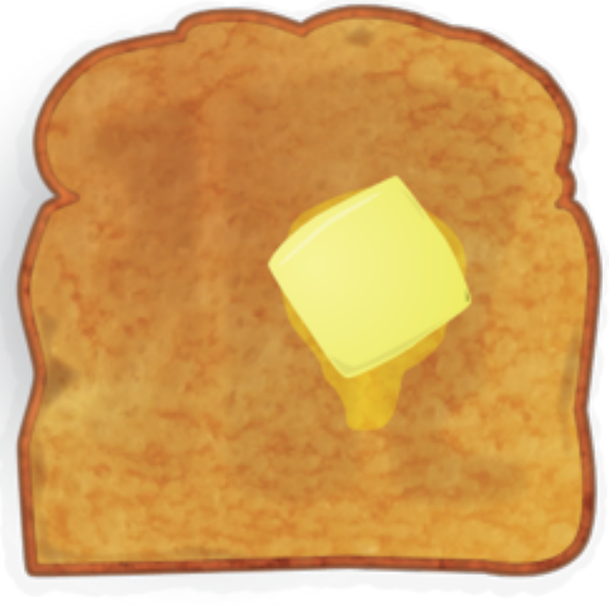




HTML Elements

Some standalone elements such as ‘br’ have no content. But, they need to be closed like any other element. You close it by adding a slash as such:

```
<br />
```



Instead of a sandwich it is like a piece of toast. But don’t forget the slash ‘/‘. The slash is like a piece of butter, you need it there, just as nobody wants dry toast.

All Web Pages Are Made With HTML

1. Go to any web page

2. Right Click on the page

3. Click on ‘View Page Source”

4. You are looking at HTML!

DOCTYPE <html> & <body>

`DOCTYPE` = Document type. Tells the browser which version of HTML you're using. - required

`<html>` = the container for your web page. -required

`<head>` = metadata for the browser. - important

`<body>` = contains the actual content of the web page. Everything nested inside `<body>` and `</body>` shows up in the web browser - necessary

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
    </body>
</html>
```

Nesting

HTML elements can contain other elements. That is called ’nesting’. The ‘head’, and ‘body’ elements are nested within the ‘html’ element, and the ‘title’ element is nested within the ‘head’.

Exercise: add all of these elements into the index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
    </body>
</html>
```

Cardboard Boxes

HTML elements are like cardboard boxes. You can nest different size boxes within each other, or put boxes next to each other in a larger box.

Shortcuts

Save your file: Ctrl + S (windows) or Command ⌘ + S (mac)

Open your index.html by right clicking in the editor, OR dragging your file into your browser from explorer or finder.

Refresh your page in the browser: Ctrl/Command ⌘+ R

We are going to keep this browser tab open and refresh the page instead of opening a new tab every time. Efficiency!

To switch between apps quickly, use Ctrl/ Command ⌘ + tab. THIS IS A LIFE SAVER once you get the hang of it.

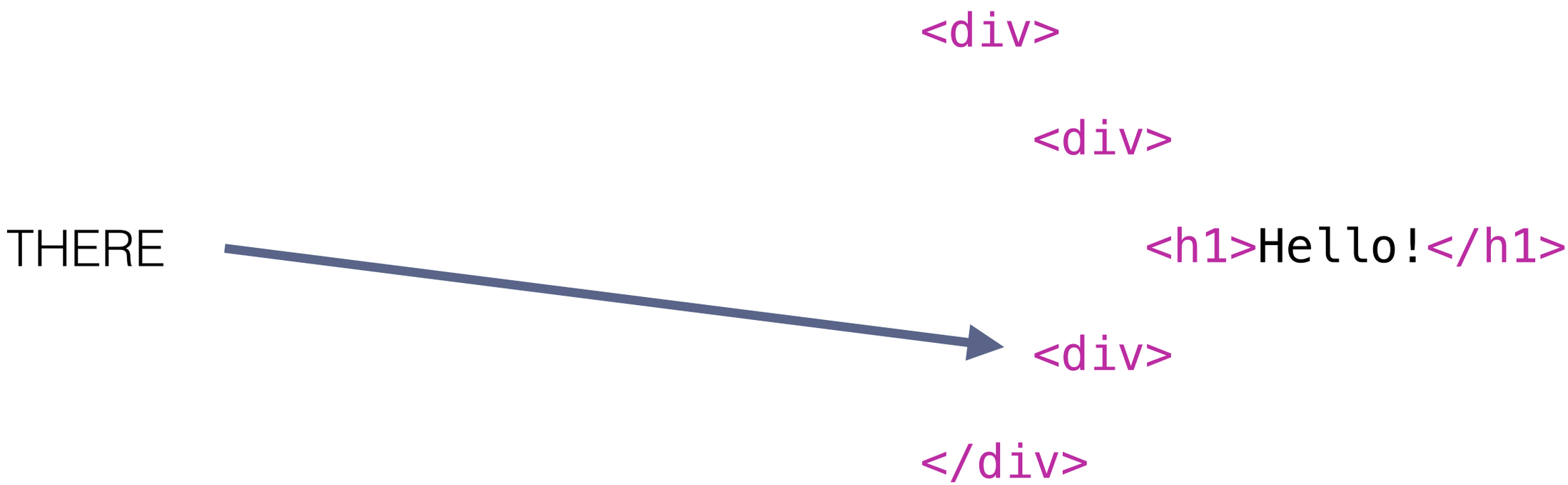
Containers

All tags are containers. The most general, all purpose container is called a . You can even nest containers inside of containers.

Can you spot the error?

```
<div>
    <div>
        <h1>Hello!</h1>
    <div>
</div>
```

Containers



Indenting

When coding HTML tags are usually indented within each other by using tabs. The tabs help you identify at a glance which elements are inside others. Keeping your code clean will help eliminate and spot possible bugs.

```
<div>
    <div>
        <h1>Hello!</h1>
    </div>
</div>
```

Comments

A comment is part of the code that help the developer remember and determine which piece of code is used for what. Comments are part of the page but are not displayed in the browser

```
<div>
    <!-- This is a Comment -->
    <div>
        <h1>Hello!</h1>
    </div>
</div>
```

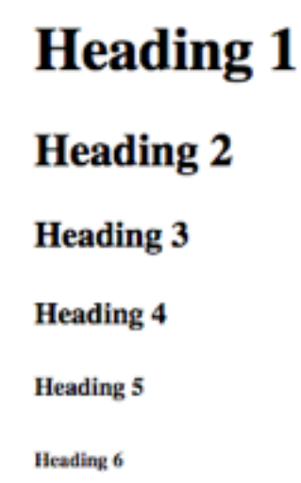
Headings

There are 6 levels of headings.

Exercise: add an <h1> heading in the <body> of your page with your name.

Next, add an <h2> underneath, your tagline.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```



Paragraphs

Headings in a web page are usually followed by paragraphs.

Exercise: Next we are going to add another <h2> below your tagline, and below that a <p> with some text about yourself

```
<h2>About Me</h2>
<p>This is a paragraph tag with
 some text in it!</p>
```

Images

The <img> element is used to add images to a page it has a ‘src’ attribute which points to the address of the image. The img tag is a standalone tag, so don’t forget the slash

```
<img src=”images/female.jpg” />
```

Exercise: Add an image above your <h1> tag. Make the img element’s src point to “female.jpg”.

What happens?

External Images

The src attribute can also point to an image anywhere on the internet. You can get the URL for an image by right clicking on it in Chrome and selecting ‘Copy Image Address’

Exercise: Find any image and paste it into the src attribute and see what happens

```
<img src="https://www.google.com/images/srpr/logo11w.png" />
```

Syntax Highlighting

You may have noticed that in your editor the text is in various colors. These colors help you quickly read through code. Syntax highlighting is essential to coding in any language.

Exercise: Remove the last quote “ from the image tag

```
<head>
  <title>Your Name</title>
  <link href="css/finished-examp
  <link href="http://fonts.googl
rel="stylesheet" type='text/css'>
</head>
<body>
  <header class="header padded">
    <img src="images/female.jpg'
    <h1>Your Name</h1>
    <h2>This is my amazing webs
  </header>
```

Attributes

Many elements have attributes.
They specify additional information about the element.
We used the src attribute with our img tag.

Exercise: Add a title attribute and hover over the image

```
<img src=”images/female.jpg”
title="female"/>
```

Unordered Lists

An unordered list is a bulleted list of items

Exercise: Add a short list of websites you visit below the paragraph. Add an <h2> above this that says “Websites I visit a lot”

```
<ul>
    <li>List Item</li>
    <li>List Item</li>
</ul>
```

Ordered Lists

An ordered list is a numbered list of items

Exercise: Add an ordered list below the unordered list in your web page to include your favorite hobbies. Add an <h2> above this that says “My favorite hobbies”

```
<ol>
    <li>List Item</li>
    <li>List Item</li>
</ol>
```

Links

One of the most important features of HTML, links let us connect to other pages and websites.

The <a> element has an href attribute which points to the website you want to link to.

```
<a href=“http://google.com">
Link Text</a>
```

Exercise: Turn your website list into actual links

More links

Next Create some social media links under your hobbies list

```
<h2>Contact Me</h2>
<a href=”http://twitter.com/yourname”>Twitter</a>
<a href=”http://facebook.com/yourname”>Facebook</a>
<a href=”http://linkedin.com/yourname”>Linked In</a>
```

And then add an email link below our social media links

```
<a href="mailto:someone@example.com">Email Me</a>
```

More HTML Elements

Text Styling

- `<strong></strong>` : Important Text, usually bolded
- `<em></em>` : Emphasized Text, usually italisized

HTML Elements to identify regions of a page

- `<header></header>` = header content, like a logo & website title
- `<section></section>` = defines a new section of content

The `<section>` and `<header>` tags act just like `<div>` tags but are used to organize your page

Sectioning the Page

Exercise: Let's use these tags to divide up our page:

Wrap our image, h1 and h2 with a `<header>` tag.

Wrap each of these areas in `<section>` tags:

- Around our About paragraph and `<h2>`
- Around the two lists
- Around our social media links and `<h2>`

Lunchtime!

Learn CSS

What is CSS?

CSS = Cascading Style Sheet

CSS applies style to your site.
It works with HTML but is not HTML

Quick exercise
Let’s look at a website together...
and then disable the CSS styles to see how it changes.

Writing CSS

CSS is written by applying properties and values to selectors.
Selectors can be html elements, classes, or IDs.

```
selector {
  property: value;
  property2: value;
  property3: value;

}
```

```
p {
    color : pink;
    background: #666;
    font-size: 30px;
}
```

CSS Selectors: elements

Any HTML element name can be a selector

```
a {
  color: orange;
  font-weight: bold;
}
```

Makes all links orange and bold

```
img {
    border: 1px solid red;
}
```

Gives all images a red border

CSS Selectors: Classes

- Many elements on a page can have the same CSS class
- Elements can have multiple classes
- Classes are case-sensitive
  In HTML elements, classes are added with the "class" attribute
- In CSS, a class is represented with a “.” (period)

Classes Continued

```
.description {
    color: blue;
}
```

CSS

```
<p class="description"></p>
<ul class="description"></ul>
```

HTML Affected

CSS Selectors: IDs

- Are unique: an ID can only be applied to one element on a page An element can only have one ID
- IDs are case-sensitive
- In HTML elements, IDs are added with the "id" attribute
- In CSS, an ID is represented with a “#”

IDs Continued

```
#my-description {
    color: blue;
}
```

CSS

```
<p id="my-description"></p>
```

HTML Affected

Adding CSS

The preferred way to add styles is with an external CSS file, called a stylesheet.

```
<!DOCTYPE html>
<html>
  <head>
      <title>My Title</title>"
      <link href="styles.css" rel="stylesheet" type="text/css">
  </head>
```

Exercise

Add the above link to your index.html file.

Then open up your styles.css file

Let’s start adding Styles!

Let’s change the font.
Add each property of CSS one at a time, save and refresh your index.html each time

CSS

```
body {
  font-family: sans-serif;
  color: gray;
}
```

HTML Affected

```
<body>
```

Styles applied to the body affect the entire page

Let’s add some styles to the header

```
header {
  text-align: center;
  background: pink;
}
```

CSS

```
<header>
```

HTML Affected

Color

There are several ways to write color values in CSS

```
header {
      background: pink;
           color: #FFC0CB;
    border-color: rgb(255, 192, 203);
}
```

Color Name

Hexadecimal

RGB

Color Notes

When using color, usually it's easiest to use hexadecimal.

**Note that hexadecimal can be written in upper OR lowercase*

- A list of css color names:
  http://www.w3schools.com/cssref/css_colornames.asp

- An easy color picker:
  http://www.colorpicker.com/

More Styling

Add each style one a time. Saving, and refreshing your page after each addition

```
body {
    color: #333;
}
```

Body text color: a darker gray

```
a {
    color: #339988;
}
```

Link color: teal

```
h1 {
    color: #ffffff;
    font-size: 48px;
}
```

h1: white and larger

Color Shorthand

When using hexadecimal colors you can use 3 or 6 characters in the color. When using 3 characters it means that each character is repeated once to make it 6 characters.

For example:

`#3A2`

is equivalent to

`#33AA22`

Borders

CSS properties can have multiple values.
Borders have 3 property values: width, border-style, and color.

You can also apply borders to each side separately: the two examples below produce the same result.

Try experimenting with "dashed" or "dotted" border styles, and width size.

```
img {
    border: 2px solid blue;
}

img {
    border-top: 2px solid blue;
    border-right: 2px solid blue;
    border-bottom: 2px solid blue;
    border-left: 2px solid blue;
}
```

Add some classes & IDs

In your HTML file, add these classes and ID to your appropriate sections:

This goes in the about section

```
<section id="about">
```

This goes in the section with your hobbies and websites

```
<section class="links">
```

This goes in the contact section

```
<section class="contact">
```

Now we can target our sections by class or ID.
Add some new CSS to make our About section gold:

```
#about {
    background: #FFCA59;
}
```

CSS

```
<section id="about">
```

HTML Affected

The Cascade

What happens if you apply multiple background colors?

```
#about {
    background: #FFCA59;
    background: red:
    background: green;
}
```

Styles added lower in the stylesheet will overwrite the ones above them. This is what "cascading" refers to.

Now let's add a new "about" class to our About section. What happens if we give that a background color?

```
<section id="about" class="about">
```

Add this to your HTML

```
#about {
    background: #FFCA59;
}

.about {
    background: blue;
}
```

Add the ‘about’ class below the ‘about’ ID

Even if they are higher in the CSS file, ID styles will override class styles on the same element.

Please remove the about class from your HTML

Element Specificity

What if we only want to style the h2 tags in our links section?

To do this you put h2 after .links, this means we are targeting only h2 tags inside anything with the links class

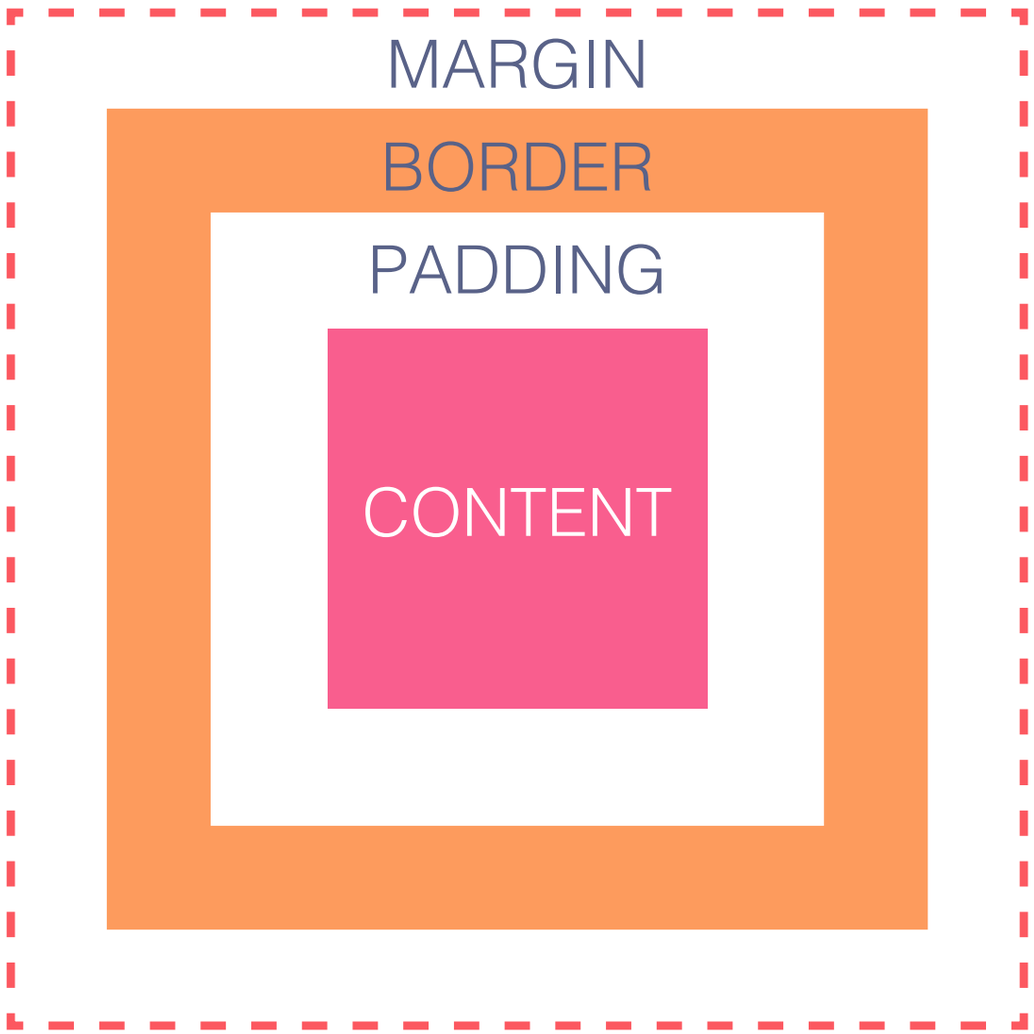
```
.links h2 {
    color: #FF572B
}
```

CSS

```
<section class="links"><h2>
```

HTML Affected

The Box Model



content: <img>, for example

padding: the space between the content and its border

margin: the space between the element's border and other elements on the page

Margins & Defaults

If you notice there is a white border surrounding the page. That is because like every other element the body tag has margins and padding. However, some elements have a default setting that is loaded before your CSS is loaded.
You can see this by inspecting your page and selecting the body tag. It has a margin that you did not put in your CSS. We can remove the margin with this CSS:

```
body {
    margin: 0px;
}
```

Padding

Next, add the class “stacked” to all our sections and our header. Separate multiple classes with a space:

```
<section class="links stacked">
```

Let’s add some padding to our “spacer” classes

```
.stacked {
    padding: 10px
}
```

CSS

```
<section class="stacked">
```

HTML Affected

Add even more padding! Each will create the same result.

```
.stacked {
    padding-top: 50px;
    padding-bottom: 50px;
    padding-left: 100px;
    padding-right: 100px;
}
.stacked {
    padding: 50px 100px 50px 100px;
}

.stacked {
    padding: 50px 100px;
}
```

Chrome Developer Tools

Browser Developer Tools are indispensable for any front-end web developer.

- Go to your page in the browser, right/ctrl-click and select "Inspect Element".
- Use the Styles tab to see your CSS, and the Computed
- tab to see an element's box model calculations.
- Try toggling some styles on and off, and editing styles in
- your browser.

Background Images

Now we will add a background image to our contact section

```css
.contact {
    text-align: center;
    color: #FFF;
    background: url(images/dark_embroidery.png);
}
```

CSS

```
<section class="contact">
```

HTML Affected

Background Properties

Try adding these background properties to our section one at a time to see what happens.

```
.contact {
    background: url(images/dark_embroidery.png);
    background-repeat: none;
    background-repeat: repeat-x;
    background-position: bottom center;
}
```

Images

Now let's change our social media links to images.

```
<section class=”contact padded”>
    <h2>Contact me</h2>

    <a href="..."><img src="images/twitter.png"></a>
    <a href="..."><img src="images/facebook.png"></a>
    <a href="..."><img src=“images/linkedin.png"></a>
    <a href="..."><img src="images/email.png"></a>
</section>
```

Sizing

Our images look pretty huge... let's change some settings in this section. We can set element sizes with CSS, too.

```
.contact img {
    width: 100px;
    margin: 10px;
}
```

You can set a width and a height, but if you only set one it will scale it proportionally.

Circular Images

With CSS you can turn a square image into a circle. It's done by adding a border with a radius of 50% to the image

```
.contact img {
    border-radius: 50%;
    border:none;
}
```

If you try different percentages you can make crazy shapes. For example, try:

```
border-radius: 80% 5%;
```

We will now add a background image to our header

```
header {
    background: url(images/background-small.jpg);
}
```

CSS

<header>

HTML Affected

Let’s stretch out the image to fill our header

```
header {
    background-size: cover;
}
```

The image is still looking pixelated... let's swap it for a larger one, and add a fixed position to make it a bit fancier.

```
header {
    background: url(images/background.jpg);
    background-attachment: fixed;
}
```

Now let's make that header image fancier

```
header img {
    border: 3px solid #fff;
    border-radius: 50%;
    padding: 10px;
}
```

CSS

```
<header><img />
```

HTML Affected

Pseudo Classes

What about if we want to make a hover state for our contact section links? Let's add the :hover pseudo class.

```
.contact img:hover {
    opacity: 0.5;
}
```

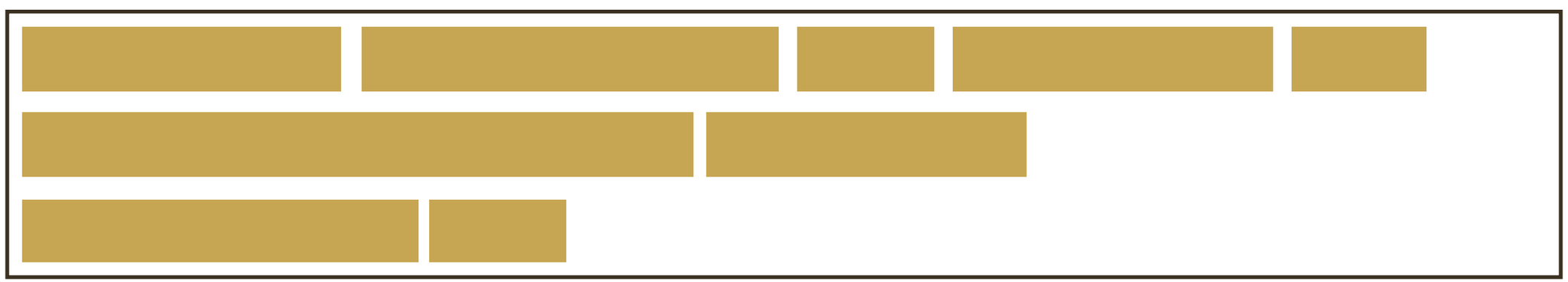
Now, when we hover over the contact image it will be slightly transparent

Inline vs Block elements

Block elements live on their own lines. They include <p>, <ul>, <h1>, <li>, <ul>, <section>, <header>, , and many other elements, and they arrange themselves to take up the full width of their container.

Inline vs Block elements

Inline elements appear on the same line upon which they are written. They include , <img />, <a> among others



Making link-style Buttons

Let's turn our website links into buttons. First, add a class to our unordered list, then remove the default list styles.

```
<ul class="websites">
```

Add this HTML

```
ul.websites {
    padding: 0;
}

ul.websites li {
    list-style: none;
    display: inline-block;
}
```

And CSS

Now we'll style our links to look like buttons

```
.websites a {
  background: #38A88F;
  padding: 10px 20px;
  color: #fff;
  text-decoration: none;
  border-radius: 5px;
  font-size: 24px;
  display: block;
}

.websites a:hover {
  background: #666;
}
```

Line Height

Line height is the spacing between lines of text. For example, in a word processor you can make your text double spaced, and you can do the same with a line height of 2.0

Now we are going to add some spacing to our text

```
p, ul, ol {
  line-height: 1.5;
}
```

Simple Test And Fix

When working with HTML you sometimes have to do some tests to see if your page functions in different screen sizes.

Go to your browser and make it as skinny as possible. Now look through the page. You see our link buttons are all stacked on top of each other.

Exercise: How would you make the buttons so they don’t touch?

Fix the Buttons

One way to fix the buttons is to add a margin to the bottom.

```
ul.websites li {
    margin-bottom: 5px;
}
```

There are a ton of other solutions to the problem, when debugging try to always use the simplest solution to cover all the cases

Custom fonts

Choose a font from: http://www.google.com/fonts/

- In the trio of buttons beside "Add to Collection", click the middle button.
- Choose the style you want to use. (Adding styles you don't need increases file sizes/page load times.)
- Copy the "Add this code to your website" link, and paste in your HTML <head>

```
<link href="http://fonts.googleapis.com/css?family=Bitter:
400,700,400italic" rel="stylesheet" type="text/css">
```

Next, look at the "Integrate the fonts into your CSS" link. You'll want to copy the font family name into your CSS.

```
h1, h2 {
  font-family: "Bitter", sans-serif;
}
```

Note: Depending on your font, you may need to also declare font-weight: normal in your CSS. h1 and h2 tags are font-weight: bold by default.

Font-family can be a never-ending list of fonts. If the first font is not found on the user's system, the browser will try to find the the next one, and so on. A common set is:

```
font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
```

More Resources

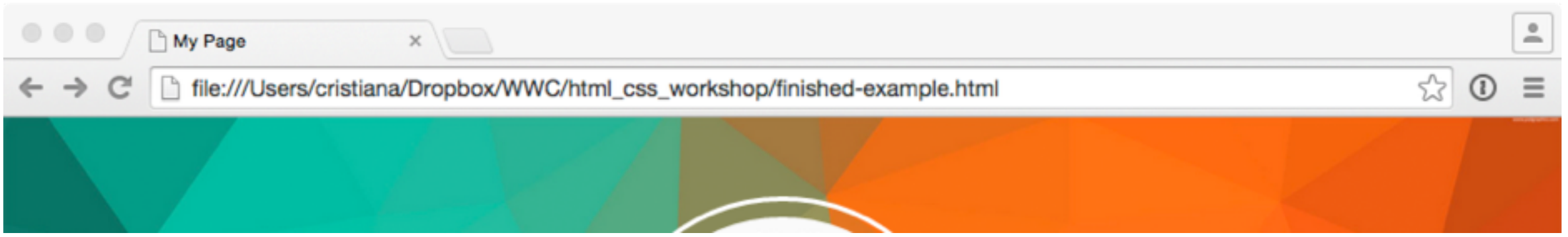
Here's a reference list of CSS properties:
http://www.htmldog.com/reference/cssproperties/

CSS Zen Garden:
http://www.csszengargen.com/

GitHub

file:// ?

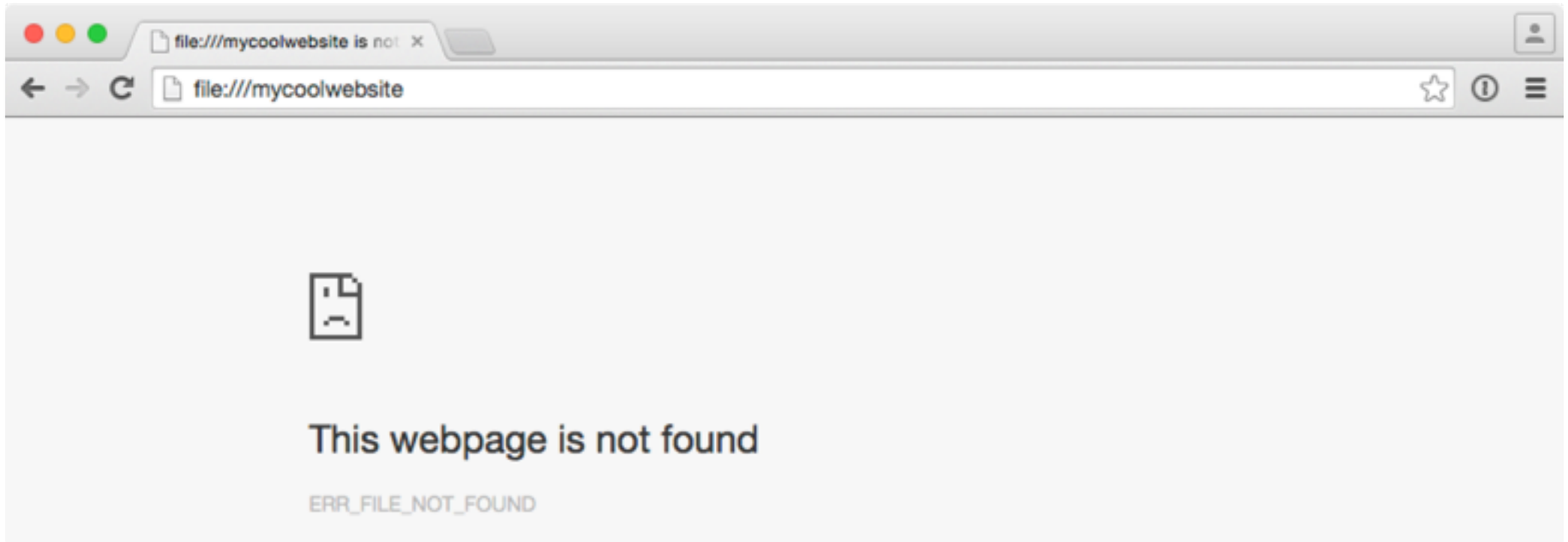
You may have noticed that the address bar shows file:// instead of http:// like you are used to.



This means you are viewing the file on your computer versus a server on the internet

Why does this matter?

A local html page behaves in different ways than it would if it was on a server. Plus, If you send a link to your page nobody could view it.



In comes GitHub

We can use GitHub to host our pages so the whole world can view it

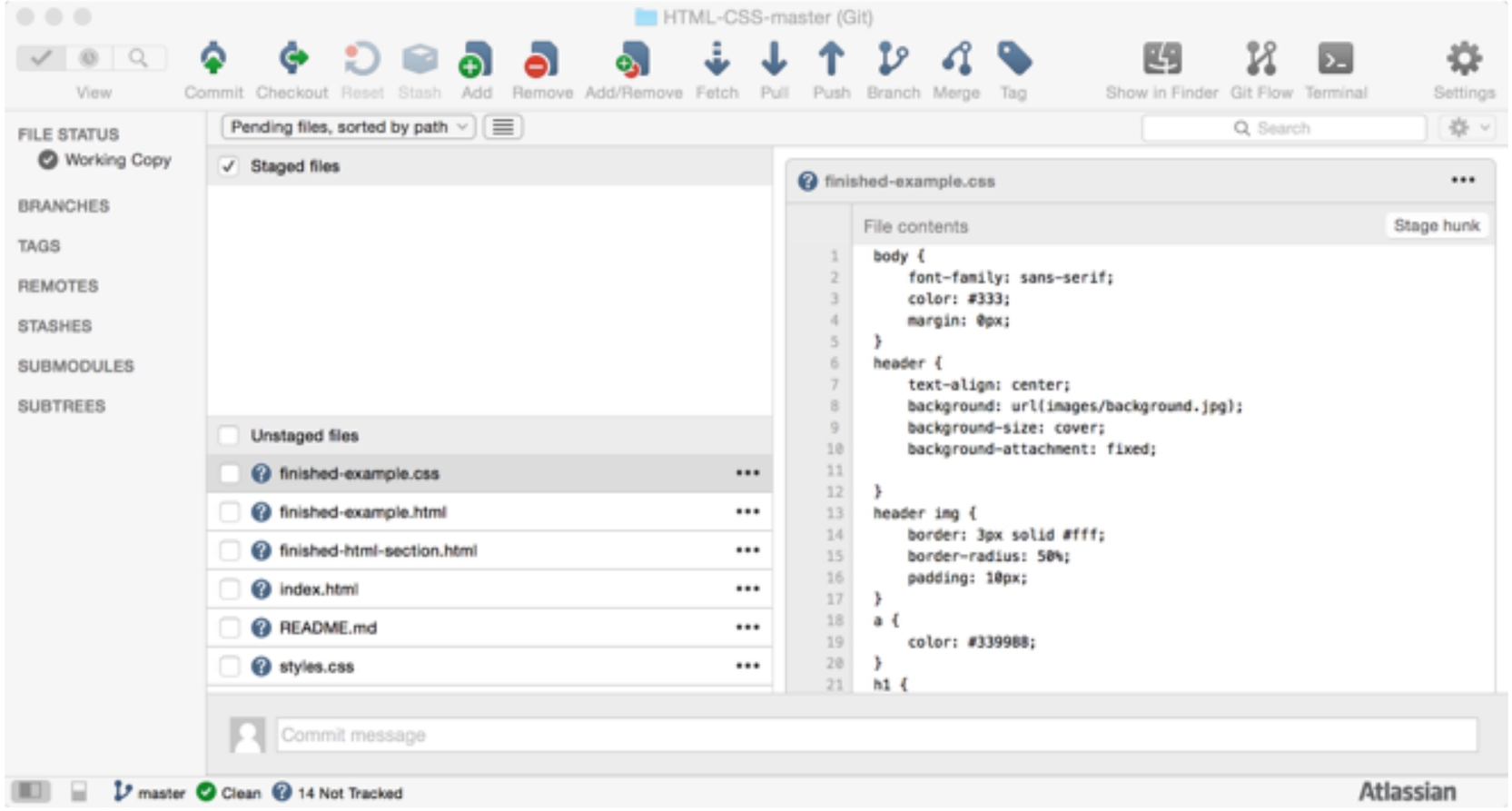
Creating a Local Repo

Load up SourceTree and click on New Repository.
Then click create local repository.

Now find the folder where your html files are and select that and click ‘OK’

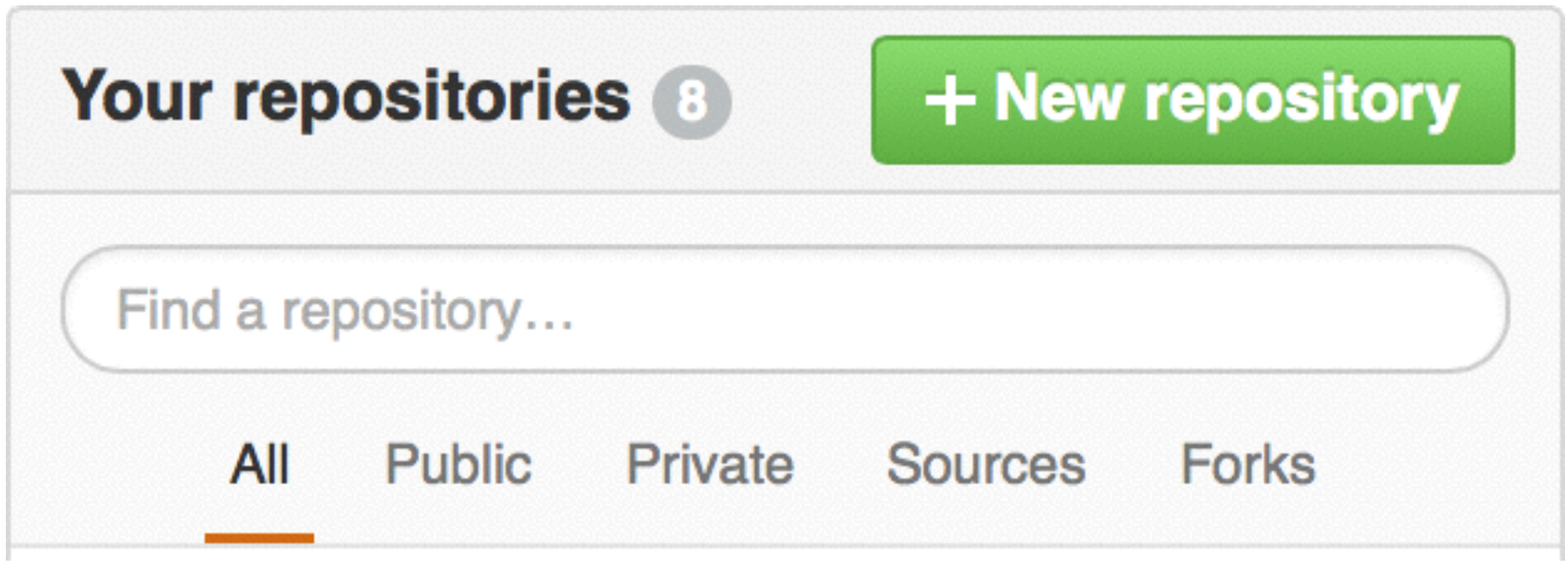
Creating Continued…

Now you will see a window containing all of your files.



Creating a GitHub Repo

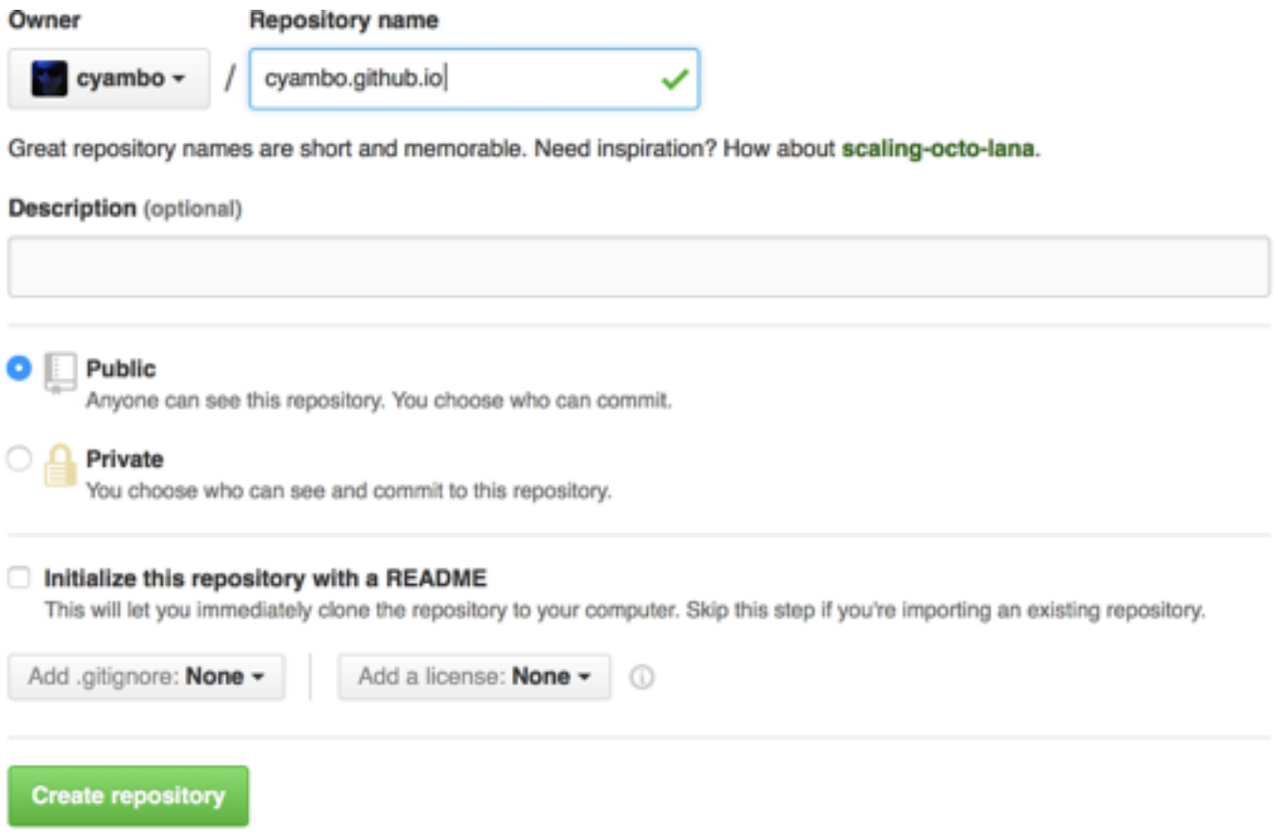
Go to Github and click on the ‘New repository’ button



A repository is a place where you can store your site on the GitHub Servers

Naming the Repo

The repository needs to be named in a specific way:
[github username].github.io



Make sure the repository is public, then click ‘Create Repository’.

More Setup

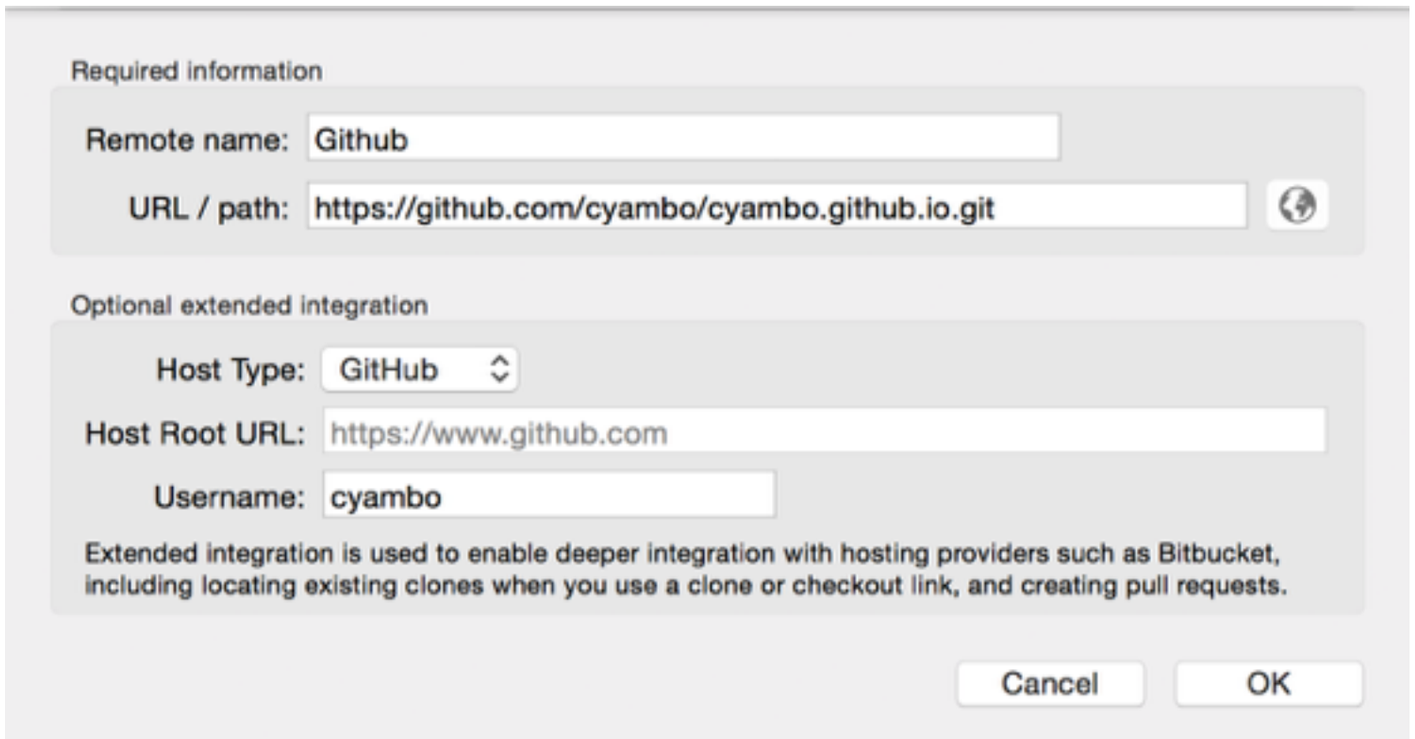
On the next screen you will see additional information about your repository

Click on the URL of your repository and copy it with CTRL or CMD (⌘) C. It looks like this:

https://github.com/cyambo/cyambo.github.io.git

Back to SourceTree

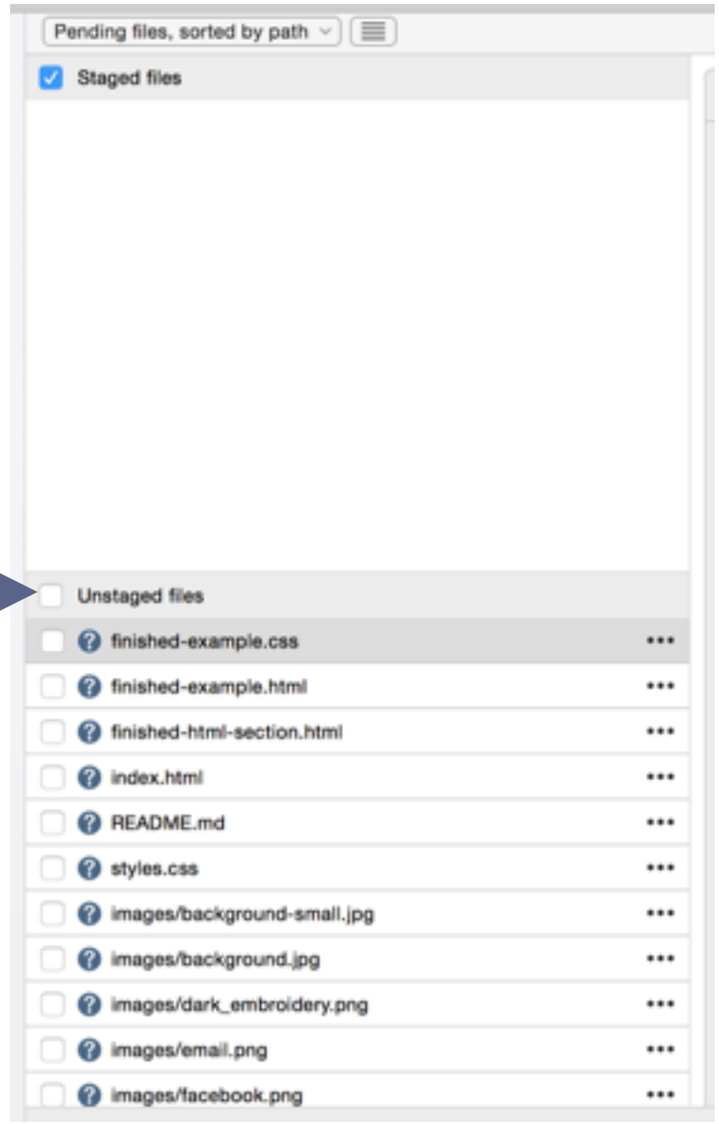
In SourceTree go to the Repository menu and click ‘Add Remote’

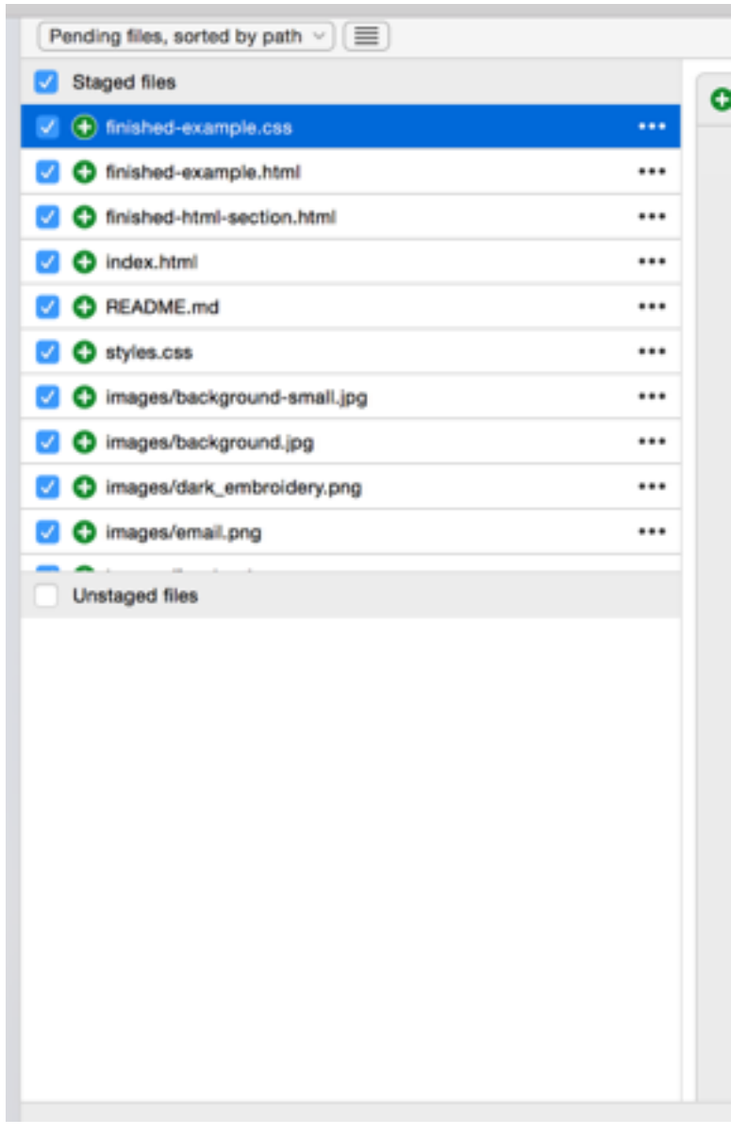


Paste the Github URL in the URL field. Then name the remote ‘Github’ and select the host type and fill in your Github username. Then click ‘OK’, and click ‘OK’ again on the next screen.

Sending to Github

Click ‘Unstaged Files’

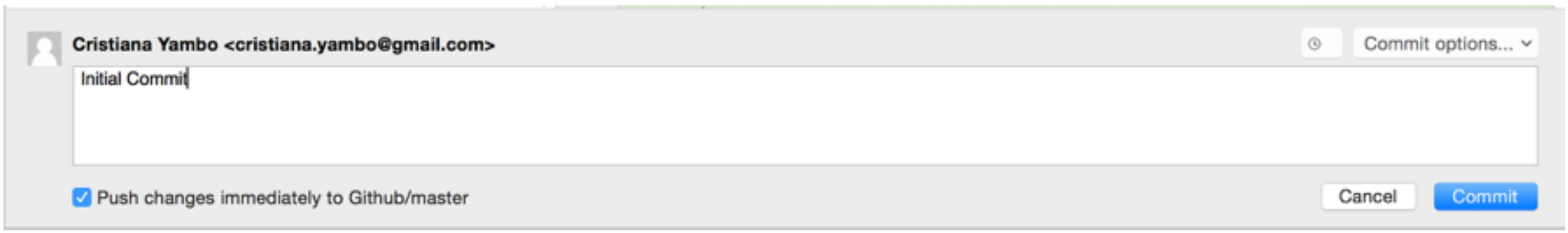




Sending to Github

Click ‘Commit’ then fill in your details. The commit log screen will then show.

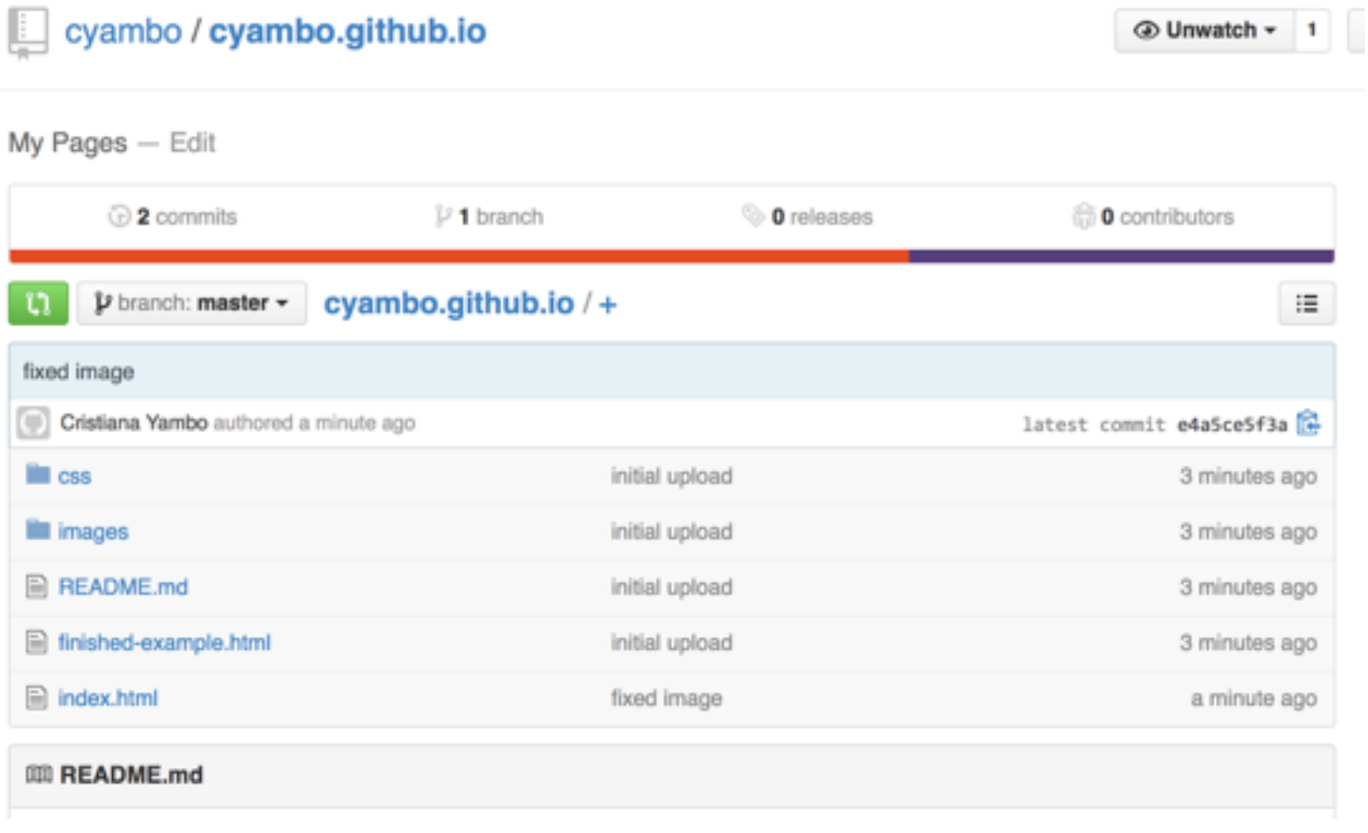
Fill in ‘Initial Commit’ and click ‘Push changes to master’ then click ‘Commit’



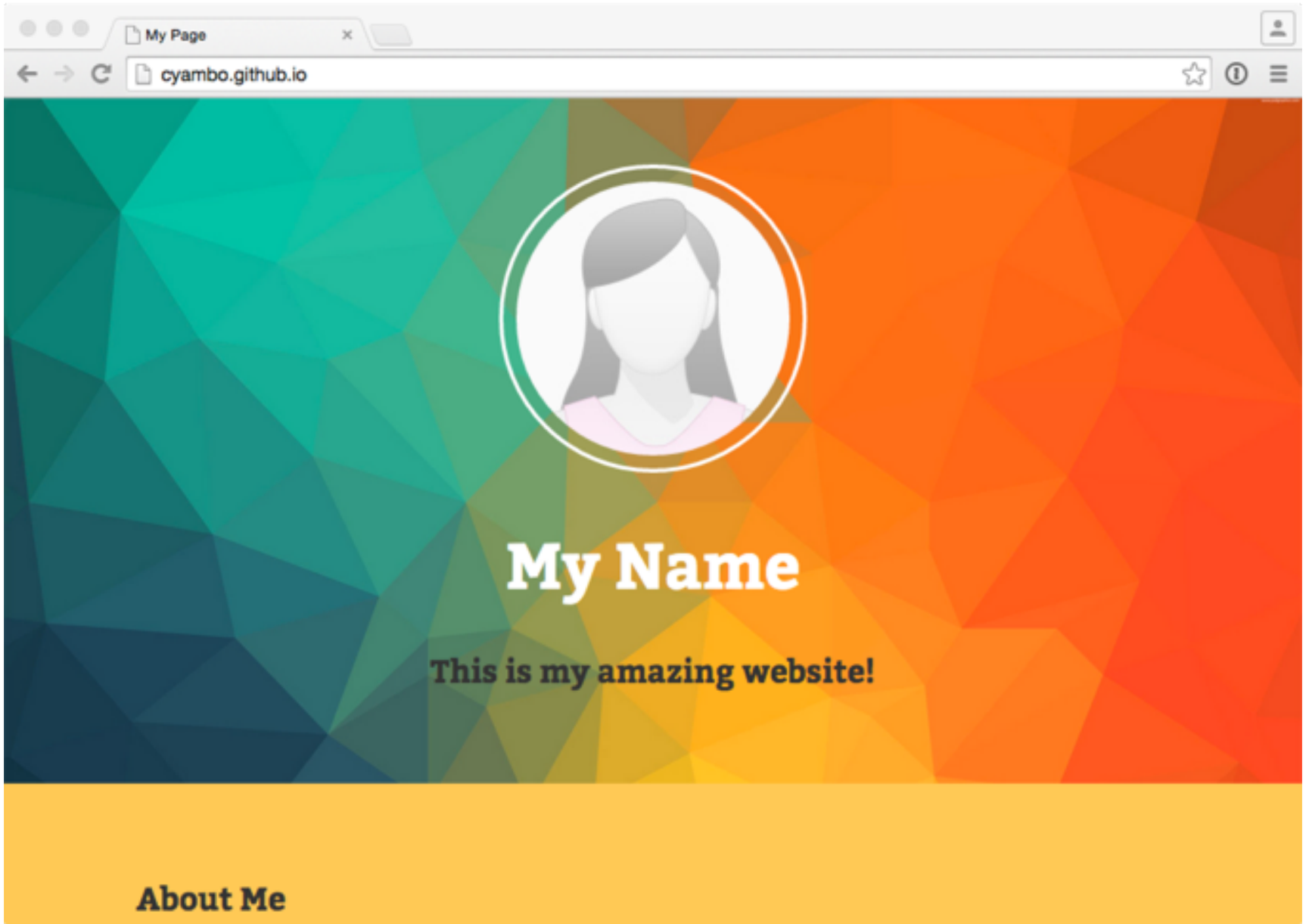
It will then ask for your Github credentials, fill them in and click ‘OK’

Viewing on Github

Now go back to GitHub and you will see a list of all the files you uploaded



View the finished Site



Thanks for Coming!