

Intro to HTML & CSS

WOMEN WHO

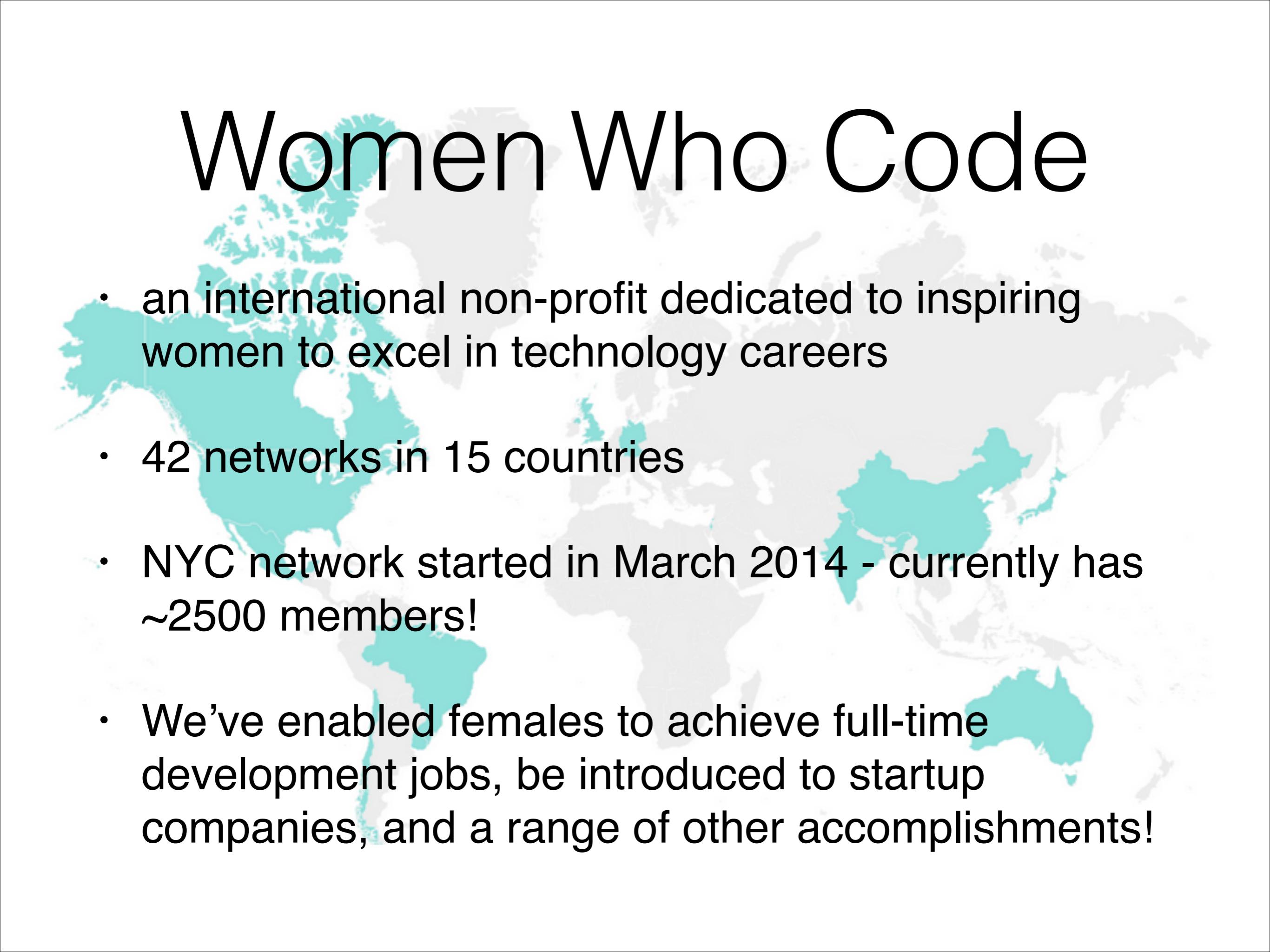
001110
00110010
000100000
10110
01101
11011
10010
00011
10101100
0011000
10110

1010000
010000011
1101101101
1001000
00000
01000
01000
11000
11011
0110111000
101111001
01010
10000
10111
01010
00001
0001100101
00010000

1100100
1101101111
0011011100
01110
11010
01010
10000
10111
01010
00001
0000110111
0001100101
00010000

00011011
011010110
000011000
1101
10010000
0110111
01101
1101
01000011
00000111
01101101

Women Who Code

A world map with countries highlighted in teal where Women Who Code networks are active. The highlighted areas include North America, parts of Europe, India, China, Australia, and South America.

- 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 2,500 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?

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

Tools That We'll Use



Browser - Google Chrome

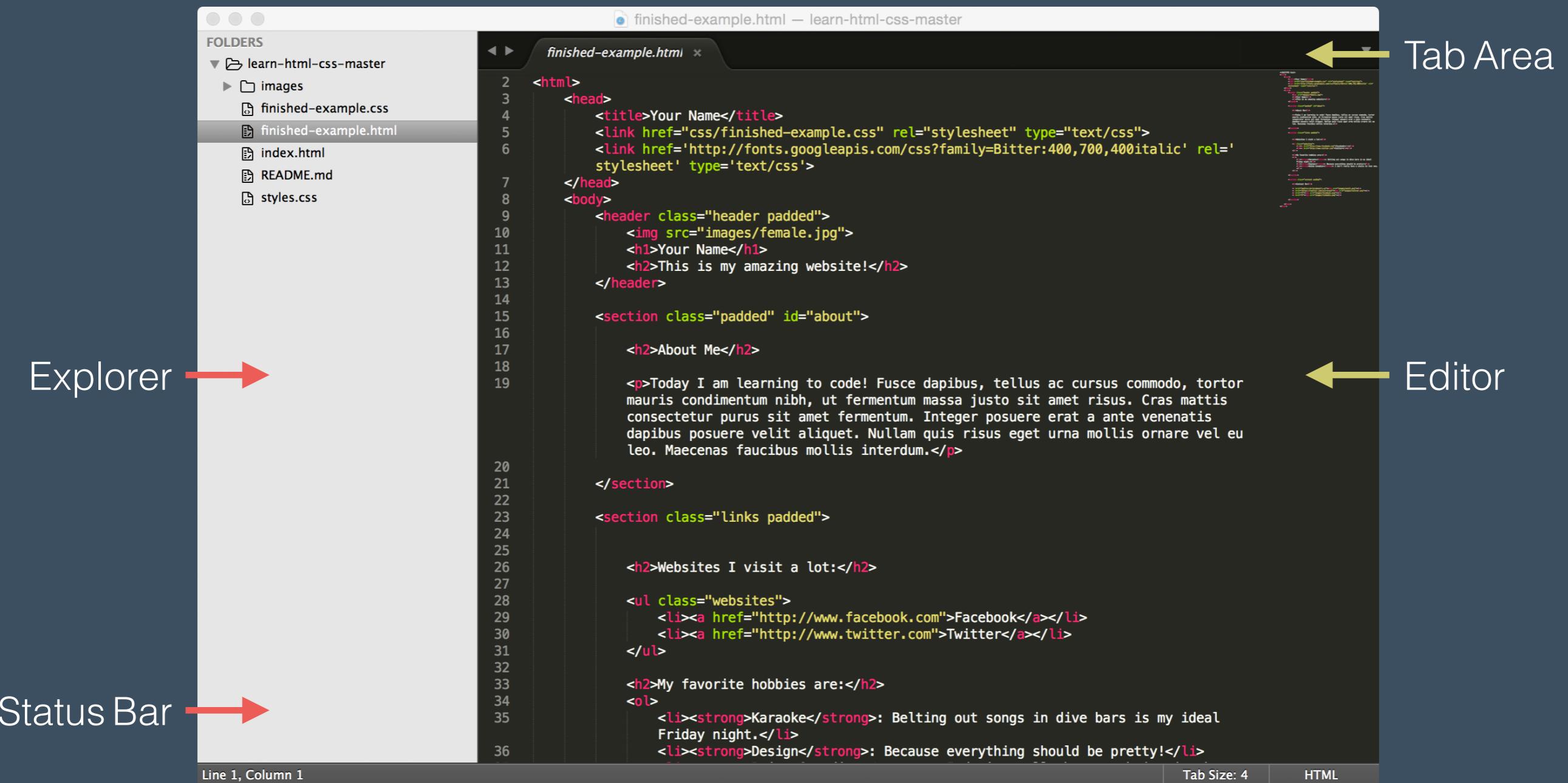
Used for viewing and debugging our site



Text Editor - Sublime Text

Used to edit HTML and CSS files

Quick Sublime Overview



Download The Course Files

If you did not download the course files get them here

<https://github.com/Nyiriland/learn-html-css>

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

Let's Get to Work!

- Open Sublime
- Go to ‘File’ -> ‘Open’ then choose the folder ‘learn-html-css’ we put on the desktop
- Open the ‘index.html’ file in Sublime and delete everything
- Open ‘index.html’ in your browser

Learn HTML

HTML - Hyper Text Markup Language

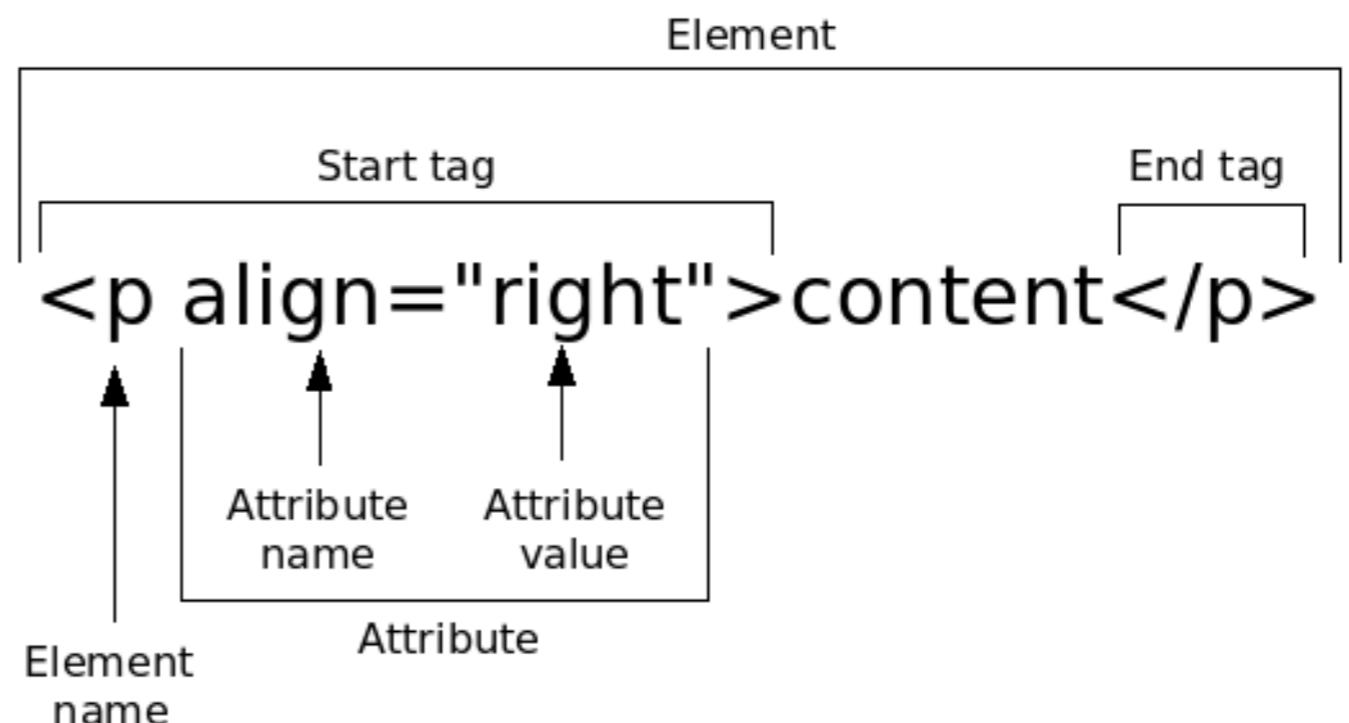
HTML is made up of elements which consist of tags

HTML elements are the building blocks of web pages

<h1></h1>
<h2></h2>
<p></p>

HTML Elements

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

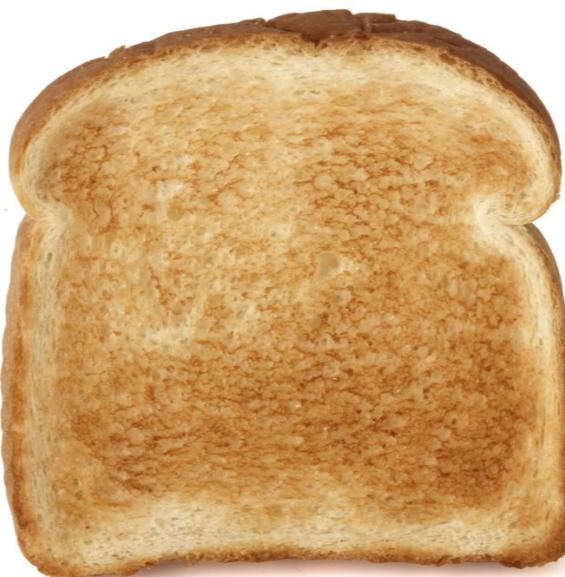


HTML Elements

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 <div>. You can even nest containers inside of containers.

<div>

<div>

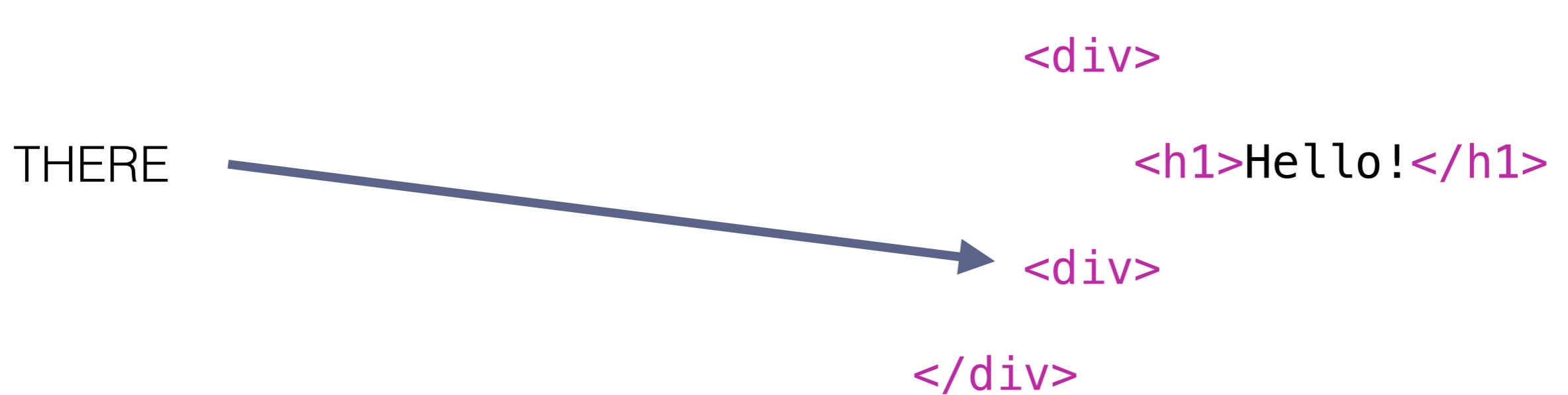
<h1>Hello!</h1>

<div>

</div>

Can you spot the error?

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>`

Heading 1

`<h3>Heading 3</h3>`

Heading 2

`<h4>Heading 4</h4>`

Heading 3

`<h5>Heading 5</h5>`

Heading 4

`<h6>Heading 6</h6>`

Heading 5

Heading 6

Paragraphs

Headings in a web page are usually followed by paragraphs.

Exercise: Add a paragraph under your `<h2>` with a description about yourself (where are you from, how many children do you have, where do you live).

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

Images

The 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

```

```

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

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-example.css" rel="stylesheet" type='text/css'>
  <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type='text/css'>
</head>
<body>
  <header class="header padded">
    
    <h1>Your Name</h1>
    <h2>This is my amazing website!</h2>
  </header>
```

Attributes

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

```

```

Exercise: Add an alt attribute and
hover over the image

Unordered Lists

An unordered list is a bulleted list of items

Exercise: Add a short list of websites you visit. 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 in your web page to include your favorite hobbies. Add an `<h2>` above this that says “My favorite hobbies are:”

```
<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 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 along with our social media links

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

More HTML Elements

Text Styling

- `` : Important Text, usually bolded
- `` : Emphasized Text, usually italicized

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
- Around our lists
- Around our social media links

Lunchtime!

Learn CSS

What is CSS?

CSS = Cascading Style Sheet

CSS applies style to your site.
It works *with* **HTML**, but it's 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;  
    property: value;  
    property: value;  
}
```

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

CSS Selectors: elements

Any HTML element name can be a selector.

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

Makes all links
orange and bold

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

Gives all images a
red border

CSS Selectors: classes

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)

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

HTML
affected

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

CSS

CSS Selectors: IDs

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 “#”

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

HTML
affected

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

CSS

Adding CSS

The preferred way to add styles is with an external CSS file (“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 line of CSS one at a time, save, and refresh your index.html file.

**Styles applied to the body affect the entire page.*

```
<body>
```

HTML
affected

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

CSS

Let's add some styles to the header.

```
<header>
```

HTML
affected

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

CSS

Color

There's more than one way to write color values in CSS.

```
header {  
    background: pink; ← Color name  
    background: #FFC0CB; ← Hexadematical  
    background: rgb(255,192,203); ← RGB  
}
```

Usually it's easiest to use hexadecimall.

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

Resources

- A list of css color names:
http://www.w3schools.com/cssref/css_colornames.asp
- An easy color picker:
<http://www.colorpicker.com/>

Let's change up some more styles.

Add each one at a time, save, and refresh your page.

```
body {  
    colour: #444444;  
}  
  
a {  
    colour: #38A88F;  
}  
  
h1 {  
    colour: #ffffff;  
    font-size: 48px;  
}
```

Body color:
a darker gray

Link color: teal

h1: white and
larger

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:

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

HTML

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

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

HTML

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

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

HTML
affected

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

CSS

What happens if you apply multiple backgrounds 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="padded about">
```

Add this in
your HTML

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

CSS:
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.**

Element specificity

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

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

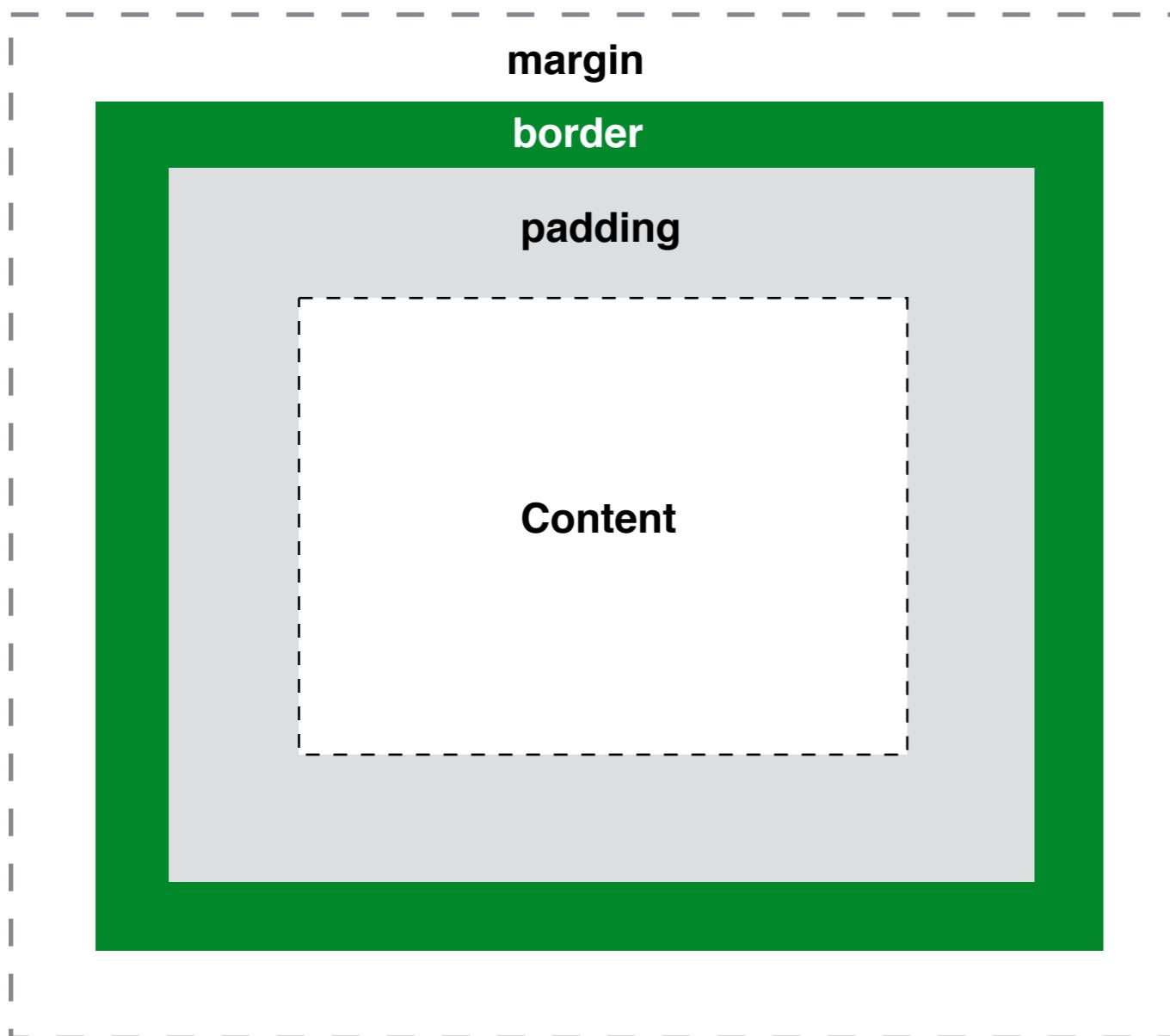
HTML
affected

```
.links h2 {  
    font-family: sans-serif;  
    colour: #FF572B;  
}
```

CSS

By adding multiple selectors, we can target more specific parts of our HTML.

The Box Model



content: , 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

Let's add some padding to our "padded" sections.

```
<section class="padded">
```

HTML
affected

```
.padded {  
  padding: 10px;  
}
```

CSS

Let's add MORE padding! These all create the same result:

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

Let's add a background image to our contact section:

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

HTML
affected

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

CSS

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;  
}
```

Resources

- Great site for patterns: <http://subtlepatterns.com/>

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

```
<section class="contact padded">
  <h2>Contact me</h2>
  <a href="#"> </a>
  <a href="#"> </a>
  <a href="#"> </a>
  <a href="#"> </a>
</section>
```

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

```
.contact h2 {  
    color: #fff;  
}  
  
.contact img {  
    width: 100px;  
    margin: 10px;  
    border-radius: 50%;  
    border: none;  
}
```

Let's add a background image to our header, too:

```
<header>
```

HTML
affected

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

CSS

Let's stretch out the image to fill our header. Add this:

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

The image is still looking pixelly... 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-position: fixed;  
}
```

And let's fancy up that header image, too.

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

HTML
affected

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

CSS

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: .5;  
}
```

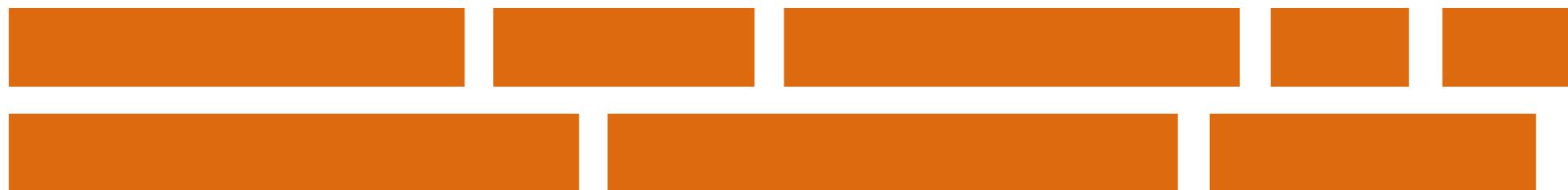
Inline vs Block elements

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



Inline vs Block elements

Inline elements appear on the same line upon which they are written. They include ``, ``, `<a>`, and `
`, 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 in
the HTML

```
ul.websites {  
    padding: 0;  
}  
ul.websites li {  
    list-style: none;  
    display: inline-block;  
}
```

CSS

Next, 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;  
    margin-right: 10px;  
}  
.websites a:hover {  
    background: #666;  
}
```

Custom fonts & font-families

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', 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 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.csszengarden.com/>