



# Absolute Beginner's Guide to Web Development!

Led by: Shonna Dorsey



# Group Introductions

- Name
- Department / Team
- What you hope to get out of today's session
- What are you most excited about in 2019?



# Research Time

Break into groups of 2 or 3 (partner with someone you don't know :))

Hint: Google and W3Schools.com are great resources!

For each of the following, please share a brief definition and its role in front end web development:

1. Group 1: HTML
2. Group 2: CSS
3. Group 3: JavaScript
4. Group 3: jQuery

# HTML - hypertext markup language

---



# What is HTML?

Hypertext Markup Language - Made up of tags, content and hyperlinks

**Skeleton of the web - without HTML, we cannot see the content on your web pages!**

Tag Examples: `<strong>` `<em>` `<div>`

Semantic Tags: `<aside>` `<article>` `<section>` `<nav>`

<http://www.w3schools.com/tags/>

# HTML Page Breakdown - Document Kickoff and the <head> Section

```
<!DOCTYPE html>
<html>
  <head>
    <!--info page needs prior to loading-->
  </head>
  <body>
    <!--visible page content-->
  </body>
</html>
```

Includes <meta> tags, references to external content (CSS documents, JavaScript files), title for the page)



# <!DOCTYPE>

The <!DOCTYPE html> declaration is an instruction about which version of HTML the page is written in.

Currently, HTML is on version 5.x. Using <!DOCTYPE html> tells the browser that your page content is utilizing the latest version of HTML. Check [w3.org](https://www.w3.org/) for more information.



## <head>

The <head> section includes information the page needs to load and data about the page (meta tags). An example of content within a <head> section follows:

```
3    <head>
4        <meta charset="UTF-8">
5        <meta name="description" content="Training Website">
6        <meta name="keywords" content="HTML,CSS,JavaScript">
7        <meta name="author" content="Shonna Dorsey">
8        <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10       <title>My Page</title>
11
12       <link rel="stylesheet" href="css/style.css" />
13
14       <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
15       <script src="js/script.js"></script>
16
17    </head>
```





## <meta>

<meta> tags are used for indexing, searching, etc. The more (relevant) information here, the better.

```
4      <meta charset="UTF-8">
5      <meta name="description" content="Training Website">
6      <meta name="keywords" content="HTML,CSS,JavaScript">
7      <meta name="author" content="Shonna Dorsey">
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# <meta>

One other important new attribute in the <meta> section is the ‘**viewport**’ attribute. Viewport is important in Responsive Web Design because it describes the page appearance based on the device size.

From w3schools, the example on the left is what a page looks like without viewport:



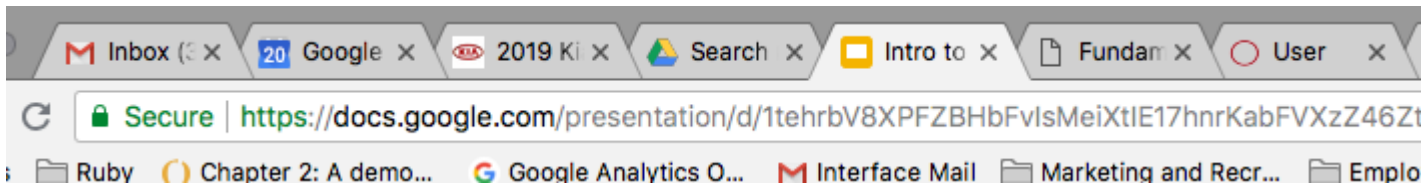
**Without the viewport meta tag**



**With the viewport meta tag**

## <title> - friendly name for URLs

Every time you load a web page, you will see a 'friendly name' in the tab for the page. Here's a screenshot from my current tabs (I know. It's a lot :))



Each of those tabs appears with a 'friendly name' because of the title tag. Make it meaningful. Keep it short.



## External Content - CSS References

CSS Stylesheets and JavaScript content can be entered directly into an HTML document.

This practice is **ok** for small projects. I recommend storing CSS and JS in external files - much easier to manage.

For CSS the format is:

```
<link rel="stylesheet" href="css/style.css" />
```

- rel = relationship
- href= hyperlink reference
- /> = self closing tag.



# External Content - JavaScript References

CSS Stylesheets and JavaScript content can be entered directly into an HTML document.

This practice is **ok** for small projects. I recommend storing CSS and JS in external files - much easier to manage.

For JS, the format is:

1. ***<!--library reference-->***
2. `<script src="https://code.jquery.com/jquery-3.3.1.js"></script>`
3. ***<!--custom file-->***
4. `<script src="js/script.js"></script>`

**IMPORTANT:** Any library references must appear above the custom file



## Let's review in Github

- Visit the following page: **github.com**
- Click the **sign in** button:
  - Username: **mooweb**
  - Password: **Pass@word123!**
- Once you are signed in, go to the following link:  
**tiny.cc/jan15-files**
- Click on your name
- Click on index.html to view the <head> section



## <body> section - where your content appears

Think about it like this:

- <head>: info the page needs prior to loading (**brain** of the page - behind the scenes)
- <body>: visible content on the page (**visible** attributes of a person - hair, clothes, shoes, etc.)

# <body> content - images, text, links

The <body> section is made up of tags, attributes and content. A couple of examples follow:

## Bold Text

Before bold `<strong>` bold text `</strong>` after bold

All of the content between the opening and closing `<strong>` tags will be **bold** in your browser

## Linked Text

`<a href = "http://google.com">`Best Search Engine!`</a>`

**a** = anchor, **href**=hyperlink reference, **Best Search Engine** = visible text, `</a>` = closing tag



# <body> - images, text, links, lists

## Images

**Format:** ``

**Example:** ``

**img** = image tag

**src** = link to image file

**alt** = text that appears if the image doesn't show up. Can also be used by screen readers for accessibility purposes.

# <body> - images, text, links, lists...oh my



## Text

You can use `<p>` tags to add text to a page and line breaks `<br/>` to add a single line of text between content. Examples:

1. `<p>`Here is some text. A little more text. Just a little more text.
2. `<br/>`
3. Adding one more line of content below the first line and closing out the paragraph `</p>`

## Comments

Helpful notes which do not appear on a live web page, are called comments. Comment notation follows:

`<!--here is a hidden comment-->` This text (outside of the `<!--comment-->`), will appear on the page!

# **<body> - images, text, links, lists...oh my**

## Lists

There are two list types:

- `<ul>` Unordered lists (bulleted lists)
- `<ol>` Ordered lists (numbered lists)

### Unordered List Example

```
<ul>
```

```
  <li>Bread</li>
```

```
  <li>Cheese</li>
```

```
</ul>
```

### Ordered List Example

```
<ol>
```

```
  <li>Heat Skillet</li>
```

```
  <li>Cook Grilled Cheese</li>
```

```
</ol>
```

# <body> - images, text, links, lists...oh my

## Headings

Default heading sizes are <h1> - largest through <h6> - smallest. These tags are used to highlight important summary data within your HTML documents.

### Example HTML Content:

```
<h2>Grilled Cheese Ingredients</h2>
<ul>
  <li>Bread</li>
  <li>Cheese</li>
  <li>Bacon</li>
</ul>
```

### Example Page View:

## **Grilled Cheese Ingredients**

- Bread
- Cheese
- Bacon



## Let's review in Github

- Once you are signed in, go to the following link:  
**[tiny.cc/jan15-files](https://tiny.cc/jan15-files)**
- Click on your name
- Click on index.html to view the <body> section
  - Let's update the text, images and links

# CSS - Cascading Style Sheets

---



# Cascading Style Sheets - Pretty up the html

A Cascading Style Sheet (CSS) is used to add fonts, color, spacing, responsive design components, animations, etc., to html documents. NOTE: 1em = 16px. Used for responsive resizing.

CSS format follows:

## Format

```
selector {  
  property: value;  
}
```

## Example

```
p {  
  color: red;  
  font-size: 2em;  
}
```



# Customizing a single element/group of elements

In HTML, use ID and class attributes to customize CSS content. A couple of examples follow:

## Example (Bold and Gold Text)

- **HTML:** `<strong><p id="gold">Gold text.</p></strong>` Text without formatting.
- **CSS:** `p { color: gold; }`
- **Output:** **Gold text.** Text without formatting.





## Let's review in Github

- Once you are signed in, go to the following link:  
**[tiny.cc/jan15-files](https://tiny.cc/jan15-files)**
- Click on your name
- Click on the css folder, then style.css to view the custom CSS file:
  - Let's update the font color, font size, background color, and image corners of elements in the index.html document

**jQuery - write less. do more.**

---



# jQuery - Write Less. Do More.

JavaScript is a powerful and flexible scripting language, but can be intimidating for new developers. jQuery is an excellent entry point into interactive front-end web development.

The tagline for jQuery is 'Write less. Do more.'

In this section of the deck, we are going to walk through setting up a jQuery document and customizing a couple of simple functions.



## jQuery Setup - index.html and script.js

In our index.html file, you will notice a reference to a jQuery library hosted by jquery.com through a Content Delivery Network or CDN.

The library reference **must appear before** the custom file. Think about it like checking out a book from the library

```
14     <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
15     <script src="js/script.js"></script>
```



# jQuery Methods, Functions, etc.

jQuery methods, functions, etc., can be activated upon page load or after the user interacts with your page in a predefined way.

**Example: Display hidden images when the page loads:**

```
$('#img').fadeIn(slow);
```

**Example: Hide images when a user clicks a button:**

```
$('#button').click(function(){  
    $('#img').hide();  
});
```



## Let's review in Github

- Once you are signed in, go to the following link:  
**[tiny.cc/jan15-files](https://tiny.cc/jan15-files)**
- Click on your name
- Click on the js folder, then script.js to view the custom JS file:
  - Let's make it so all images on the page disappear when we click the button
  - Let's hide the background image on the page (<body>) when the page loads

---

**Thank you!!**