

# HTML



# CSS



## HTML & CSS: LEVEL 1

Instructor: Beck Johnson

[beckjohnson@gmail.com](mailto:beckjohnson@gmail.com)

Week 1 - February 23, 2017

<https://kweeket.github.io>



# INTRODUCTIONS

- Who are you?
- What do you do/study/etc?
- What is your experience with web development?
- What are you hoping to get out of this class?



# CLASS SCHEDULE

- Thursdays, 2/23 - 3/23, 6:30 - 9:30pm
- 10 minute break somewhere in the middle
- No grades, no tests
- Questions and feedback highly encouraged.



# CLASS OVERVIEW

- The basic rules of HTML and CSS
- Using CSS to style web pages
- Website structure, navigation and file organization
- Intro to related technologies (Javascript, Git, Hosting)



# SESSION OVERVIEW

- Overview of a website
- Code editors and web authoring tools
- Anatomy of an HTML Document
- Intro to Basic HTML Elements
- HTML Attributes
- Code and file organization and best practices
- Build your first webpage!



## ODDS AND ENDS

- Save your work!
- Best way is to use a thumb drive!
- Other options:
  - Dropbox
  - Class folder server on computer (gets deleted at end of quarter!)



## ODDS AND ENDS

<https://kweeket.github.io>

- Class files and slides will be posted here.
- Download the zip of “start” files and save them on your flash drive or on Dropbox and work off of there.



# COMPONENTS OF A WEBSITE



# CONTENT, DESIGN, & CODE



**Content** is the most important part of any website



**Design** is critical to the best user experience.



**Code** is the language of the browser to bring the content and design to life.



# CONTENT

What am I presenting?

## TEXT

- Articles
- Links
- Captions
- Lists

## MEDIA

- Images
- Videos
- Audio
- Games

## What is the experience?

### **USER EXPERIENCE & INFORMATION ARCHITECTURE**

- Layout
- Navigation
- User flows
- Ease of use

### **GRAPHIC DESIGN**

- Colors
- Fonts
- Backgrounds
- Borders
- Icons

# </> CODE

How does the computer understand?

- **HTML** structures and organizes **CONTENT**
- **CSS** stylizes the content and creates layout.
- **Javascript** adds interactivity.

# PRACTICE, STUDY, PRACTICE

- Learning to write code is hard!
- Practice as much as you can. To get good at web development, it takes lots of repetition!
- Ask questions

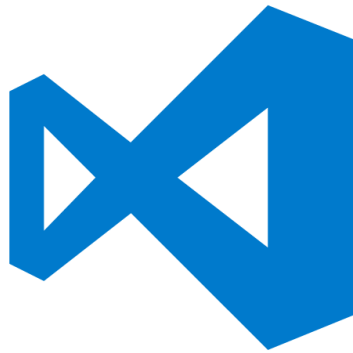


# CODE EDITING TOOLS

# WEB BROWSERS

- **Chrome** – All platforms
- **Firefox** – All platforms
- **Safari** – Mac, iPhone, iPad
- **Internet Explorer** – PC/Windows

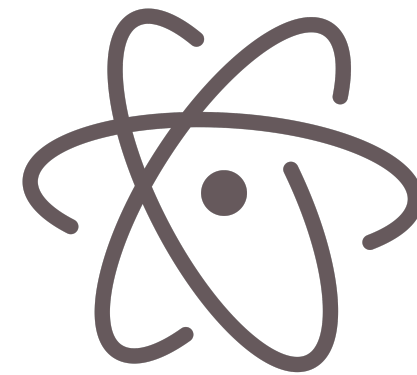
# CODE EDITORS



VS Code



Brackets



Atom




Sublime Text



Coda



# DEVELOPER TOOLS

- Firefox: plugin  **Firebug**
  - <http://getfirebug.com/>
- Chrome: **Developer Tools**
  - Right click > Inspect
  - Or View > Developer > Developer Tools
- Safari: **Developer Tools**
  - Open Preferences > Advanced > Show Develop menu in menu bar
  - Right click > Inspect Element
- Internet Explorer: **Developer Tools**
  - F12 key

<html>

# HTML DOCUMENTS

# HTML DOCUMENT

```
<!doctype html>

<html>

<head>

    <meta charset="UTF-8">
    <title>My First Page</title>

</head>

<body>
    <h1>The body is what the browser sees.</h1>
    <p>Several ways to format text.</p>
</body>
</html>
```

# HTML ELEMENTS

<!doctype html>

- HTML elements have tags in <> brackets.
- Most times, an HTML tag will have an **opening** tag and a **closing** tag.
- Some elements are “self-closing”

<element>Content goes in here</element>

<p>I'm a paragraph!</p>

# HTML RULES

- Tags are ALWAYS written in lowercase
  - `<a>` **NOT** `<A>`
- Tags must be closed
  - `<p>Stuff in here.</p>`
  - `<div>Content in here</div>`
  - `` (“self-closing”)
- Indent tags that are nested for readability.

# DOCTYPE DECLARATION

<!doctype html>

- The very first thing in any HTML document.
- Goes before <html> tag
- Tells the browser what version of HTML the document is written in.

# HTML DECLARATION

`<html>`

- The top line after `<doctype>` declaration.
- Tells the browser “This is where everything starts!”

```
<html><!-- everything else --></html>
```

# HEAD ELEMENT

`<head></head>`

- Must include a `<title>` for the document.
- Information about the document not visible to user.
- Styles in the form of CSS (inline or linking a stylesheet)
- Javascript (inline or linking a script file)

`<head>`

`<!-- metadata and resources -->`

`</head>`



# META TAGS

```
<meta charset="UTF-8"/>
```

- Used to specify "meta" information to the browser
- Page title, page description, author, search engine keywords, and character encoding.
- UTF-8 represents Unicode, a system to handle text consistently in a variety of languages.

```
<head>  
  <meta charset="UTF-8">  
  <meta name="author" content="Your Name">  
  <meta name="description" content="A thrilling page" />  
</head>
```

# TITLE TAG

```
<title>My First Page</title>
```

- Meta tag, goes in the <head>
- Required in all HTML documents.
- Defines the title in the browser toolbar.
- Provides title when page is bookmarked.
- Displays a title for the page in search results. (Google, Bing)

# <!--HTML COMMENTS-->

- Can be used anywhere in the HTML document.
- Are not visible to the user.
- Great for leaving notes for yourself or other developers.

<!-- Comments are great for everyone! -->

# BODY ELEMENT

`<body></body>`

- Area of the HTML document visible to the user
- Contains all content of the document, such as tags, links, images, tables, etc.

`<body>`

`<!-- all my content -->`

`</body>`

<html>

**PRACTICE TIME!**

# MAJOR BODY ELEMENTS

- **Headings** for dividing up your page and content.
- **Paragraphs** of text.
- Bulleted, ordered, unordered **lists**.
- **Images**
- **Links** to other pages, websites or resources.

# HEADINGS

`<h1>`My Page Header`</h1>`

- **Headings** range from most important to least important
  - `<h1>` to `<h6>`
- Search engines use the `<h1>` to determine important information about the page.

# PARAGRAPHS

`<p>Hi! I'm a paragraph!</p>`

- Most text in the document. (main content)



# LIST ELEMENTS

`<ul>`

`<li>Puppies</li>`

`<li>Kittens</li>`

`</ul>`

- Common types are **unordered** `<ul>` (aka **bulleted**) and ordered `<ol>` (aka **ordered**)
- Lists always contain **list items** (`<li>`)

# IMAGES

```

```

- Images do not have a closing tag.
- Images have two required **attributes**:
  - **src** is where the file lives (local or external)
  - **alt** is a description of the image (used for screen readers, search engines, etc.)

# LINKS

```
<a href="http://google.com">Google</a>
```

- Defines an "anchor" or link that creates a path to other pages or websites.
- Have a **required** `href` attribute that says where the link should send the user.
- Anything inside `<a>` tags are clickable.
- Can be text, an image, or any valid HTML.
- **target="\_blank"** → opens link in a new tab.

# FILE ORGANIZATION

- HTML files (.html)
- CSS files (.css)
- Javascript files (.js)
- Image files (.png, .jpg, .gif, etc)

# FILE NAMING RULES

- NO spaces in filenames
- Capitalization matters (kittens.png is not the same as Kittens.png)
- Use only letters, numbers, hyphens (-) or underscores (\_)
- Always start with a letter for file names.
- Your homepage is always **index.html**
- Recommended to use lowercase and hyphens (**about-us.html**)

# FILE STRUCTURE

- On the web, folders are called **directories**. This is what dictates paths to resources such as html files, images, and CSS and Javascript files.
- HTML should usually go in the **main (root) directory**.
- Make **subdirectories** for media, CSS, and Javascript files.

# FILE PATHS

- On the web, all resources have a Universal Resource Locator (URL)
- **Absolute paths** are URLs that start with **http**
- If someone renames or deletes the file, your link will be broken.

# FILE PATHS (RELATIVE)

- Relative paths are URLs that go to a resource in relation to what page you're on. (e.g **css/layout.css**)
- Resources “local” to you should all be relative paths. (your images, fonts, CSS and JS files)

`` image is in the same folder as this file

`` image is in a subfolder named “images”

`` image is in a superior folder named “images”





**PRACTICE TIME!**

# ASSIGNMENT

- Create a website that has 2 pages (home page and about page)
- Each page should link to the other.
- Each page should contain a link to outside website
  - Bonus: have the link open in a new tab!
- Use 3 heading tags.
- Create a paragraph under each heading.
- Show 2 images, (one local, and one outside resource).
- Add one HTML comment

# “HOMEWORK”

- Practice!
- Read the MDN Introduction to HTML
- Optional Reading
  - *HTML5 for Web Designers ch. 1-2*