Where am I?

Full Stack Web Development

Introductions and Course Overview

Who am I?
Bill Caffery

- Currently working as a web developer for a major media CMS based in the midwest
- Former Sr. Developer at The Buffalo News
- More than 15 years experience developing websites in a variety of languages (PHP, GoLang, JS, etc)

Who are you?

- What is your name?
- In three sentences, tell us your story and why you choose this course.
- Past web development, programming, or computer experience?
- What do you think make good qualities for a web developer?

OR

- What qualities do you possess that you think will make you a good developer?

What is this anyway?

- In this course, you will be building a real website, in groups. The design and layout of that site will be decided within your own group.

- Every Class:
  - Part 1 - Learn a front end concept
  - Part 2 - Do a lab to reinforce that lesson
  - Part 3 - Learn a back end concept related to the front end lesson
  - Part 4 - Finish up with a lab incorporating both concepts into the group site

- Each week, the lecture slides will be posted online after class.

Expectations

...No, this isn’t some relationship kind of deal, I don’t need a text every time you get home safely…..

But I will expect you to:

- Be present in class, physically and mentally.
- **Stay Up To Date**. If you miss a class, you will be lost next class, so reach out to me to catch up **before the next class**.
- Participate and complete all assignments and all tasks.
- **Ask Questions**! There are no stupid questions, only stupid assumptions.
- Use lab time wisely and if you finish a task early use that time to expand on a project or try to figure out something new. **Seriously!!**
- I also have a have full-time job, don’t wait until the last minute to ask questions, do it early enough to give me time to respond.
- You get out what you put in. Full stop.

What to Expect From This Class

- What will we build?
  - Personal “about me” page
  - Basic database driven CRUD api
  - Small 5-page team website
- What are we going to learn?
  - HTML
  - CSS
  - JavaScript
  - PHP
  - SEO / SMO
  - Working with and Building APIs
  - Web Hosting
  - Security
  - MySQL Databases
  - Git / Debugging Tools

The Fine Print

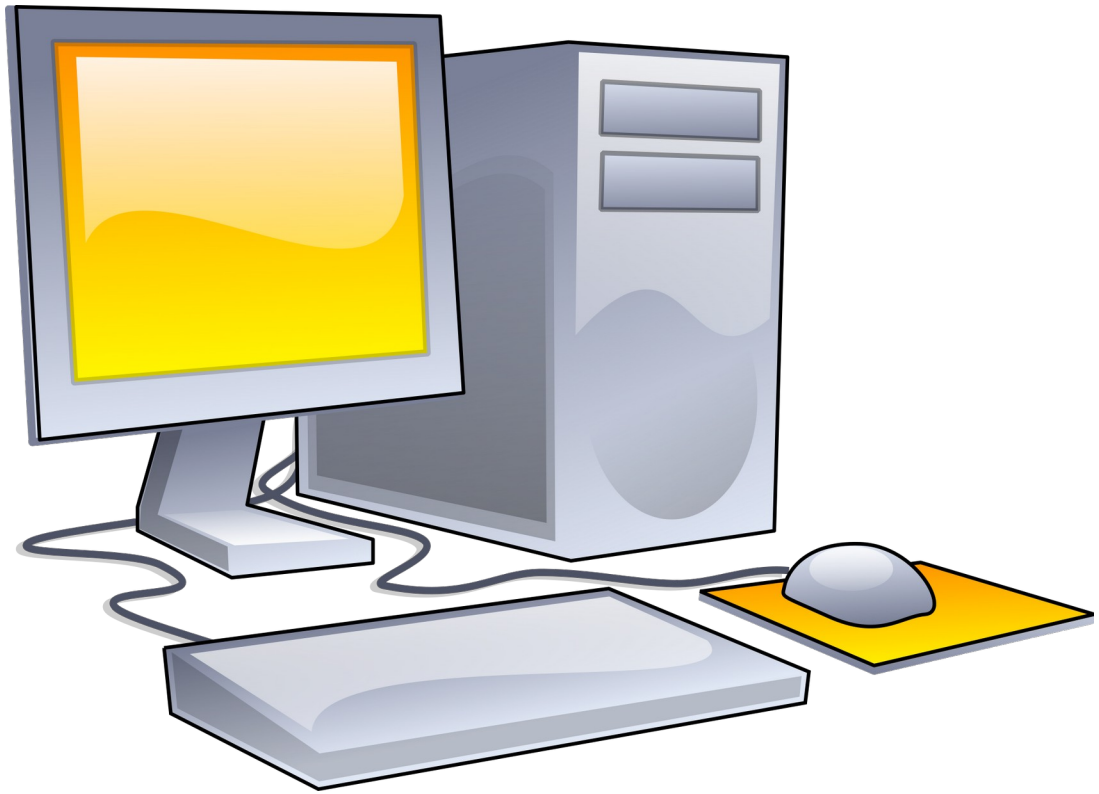
- No letter grading, this course is Pass/Fail. All work must be completed to receive your certificate of completion.
- The goals of this class are to build a portfolio for you to show potential employers, and to give you the skills necessary for an entry level position.
- Attendance will be taken during each class. If you cannot attend class, please let your instructor know.
- Expectation of a minimum of 10 hours of additional studying each week, outside of lecture/lab time.
- Bring computers to every class. We **will** be writing live code.
- Ask as many questions as you need, and ask for clarification whenever necessary.
- Again, please be present, put phones away, and give your instructor your attention.

- OK! That’s over…
- Phew...
- Time to learn...

Computer Basics

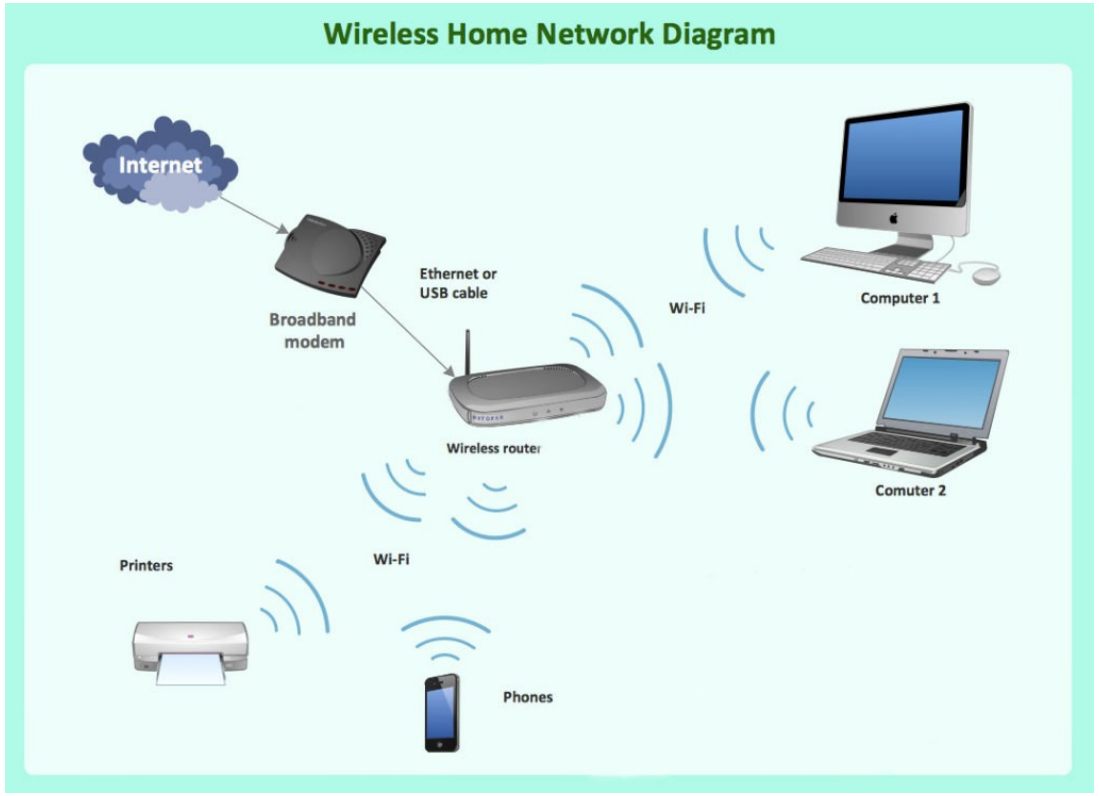
Hold on tight, this is going to go quick!

Computers & Networking



- Monitor
- Computer
  - CPU
  - Memory (RAM)
  - Graphics
  - Storage (Hard Drive)
- Mouse
- Keyboard

How does a computer connect to the Internet?



Source: http://beonservices.com

- Your computer connects to a modem, either through a wired connection or a wireless connection.
- That modem connects to an Internet Service Provider (ISP), like Spectrum or Verizon.
- The ISP connects your computer to the Internet.

But How Does that work?

- IP Addressing
  - IPv4 (255.255.255.255)
    - Decimal (0-9)
  - IPv6 (ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff)
    - Hexidecimal (0-f)
- Localhost (127.0.0.1)
- Domain Names (DNS)
  - TLD (.com, .net, .org, etc)
  - Subdomains (dev.example.com)

Protocols

- So, we have domains and IP addresses, but what are protocols?

- HTTP(S) – secure and non-secure web traffic
- FTP(S) – secure and non-secure file transfer
- SSH – secure shell access to servers

Web Browsers

Usage share of all browsers

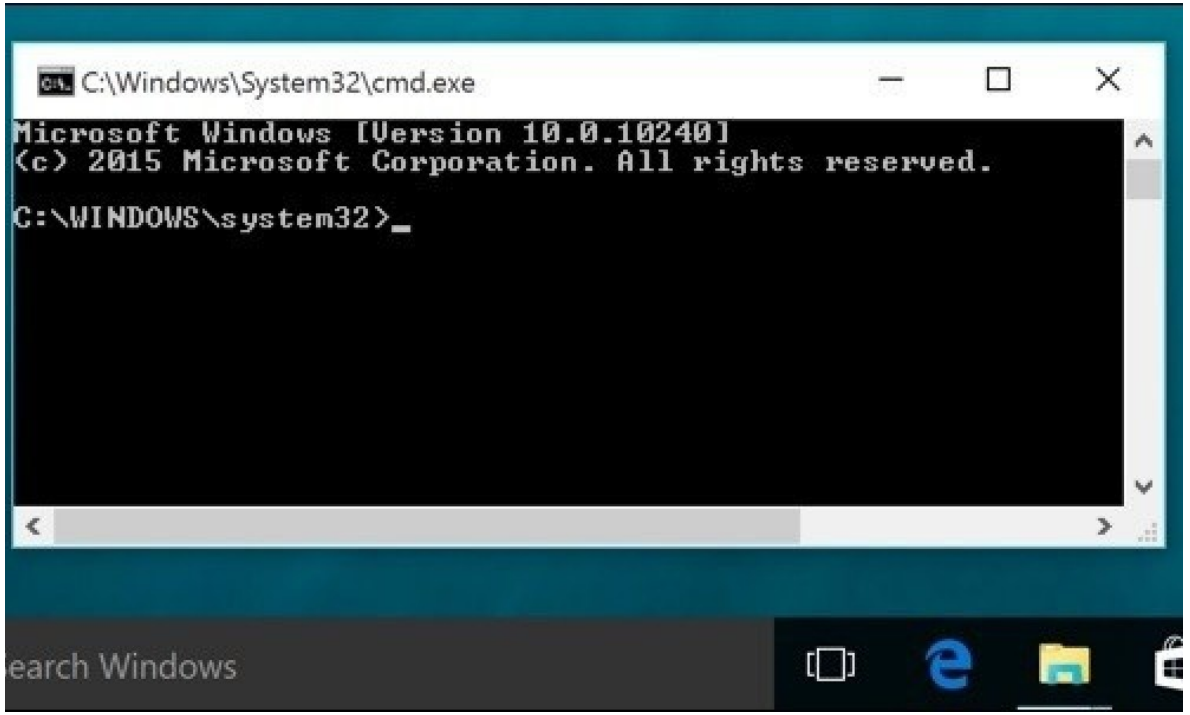
Browser	StatCounter[16] April 2020	NetMarketShare[17] April 2020	Wikimedia[18] November 2019
Chrome	62.48%	65.96%	48.7%
Safari	19.94%	17.26%	22.0%
Firefox	4.21%	3.19%	4.9%
Samsung Internet	3.40%	2.68%	2.7%
UC	1.97%	0.82%	0.3%
Opera	1.73%	0.89%	1.1%
Edge	2.67%	3.03%	1.9%
IE	1.41%	2.13%	3.9%
AOSP	0.57%	0.47%	0.2%
Others	1.62%	3.57 %	14.3%

Source: Wikipedia.com

Command Line Tools

- In Windows, click the Start button, then in the search box type **cmd** and hit enter.

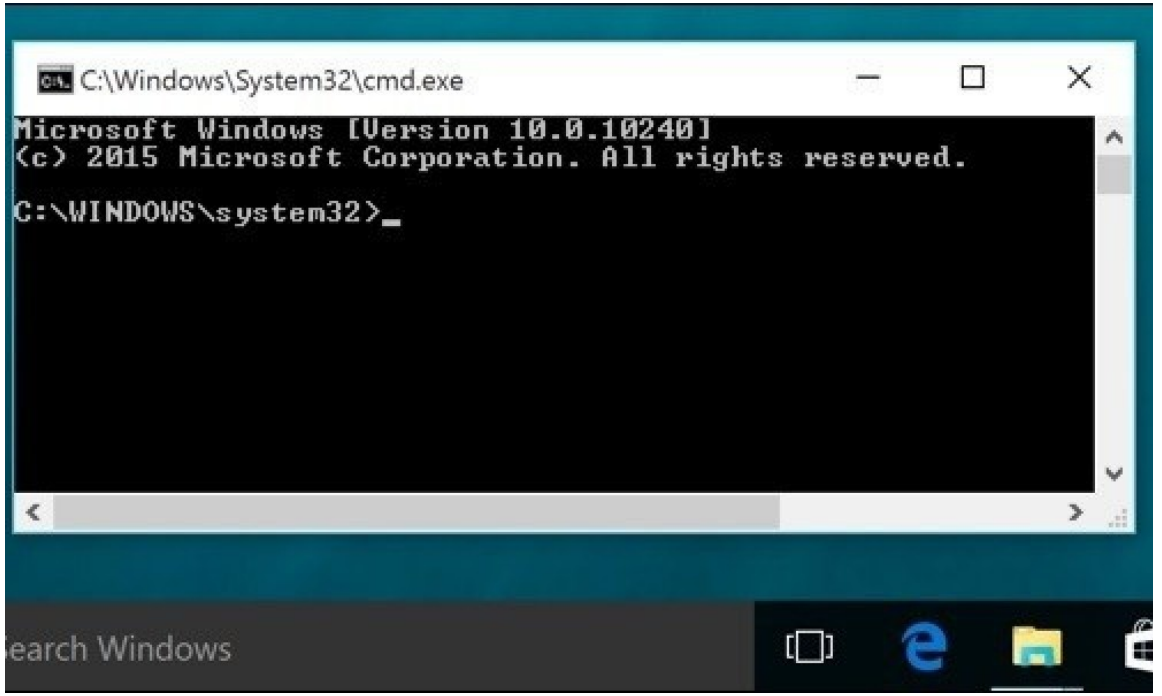
- You should see the window at the right.



Command Line Tools

Common Commands:

cd – change directory
dir / ls - list
copy / cp – copy
move / mv – move
del / rm – delete



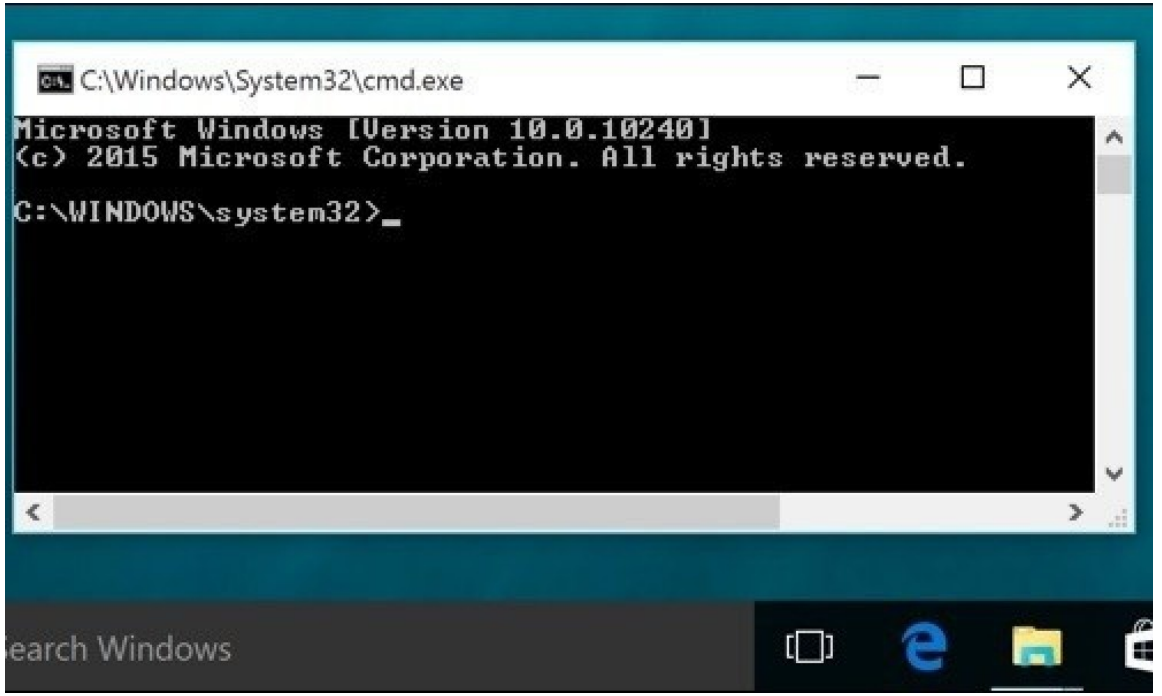
More Information:
https://www.thomas-krenn.com/en/wiki/Cmd_commands_under_Windows

Command Line Tools

Tutorial:

https://www.youtube.com/playlist?list=PL6gx4Cwl9DGDV6SnbINlVUd0o2xT4JbMu

9 Videos that show the basics of how the command line in Windows works. It's about an hour and worth it!



What is Web Development anyway?

What is Front-End development?

No, not that front end...

Actually,

- Building web applications that perform logic on a user's device directly (computer, phone, etc.)
- Communicate with the back-end systems that handle secure data and do complex analysis
- Focuses on design and user-interaction, not on algorithmic logic and databases.
- Works heavily with APIs and third-party services
- Requires hyper-focus on user experience and less on behind-the-scenes databases and structure.
- Accept user input and display data in a visually pleasing manner

What is Back-End Development?

……No, not that back end……

Instead,

- Working with databases and processing/storing data
- Big focus on algorithms, how to make code fast and efficient
- build for use cases that may not be clearly seen at the beginning of a project

- Builds APIs for front-end developers to interact with
- Requires a strong focus on security, structure, data aggregation
- These are the devs that “make it work”
- The part of the iceberg you don’t see

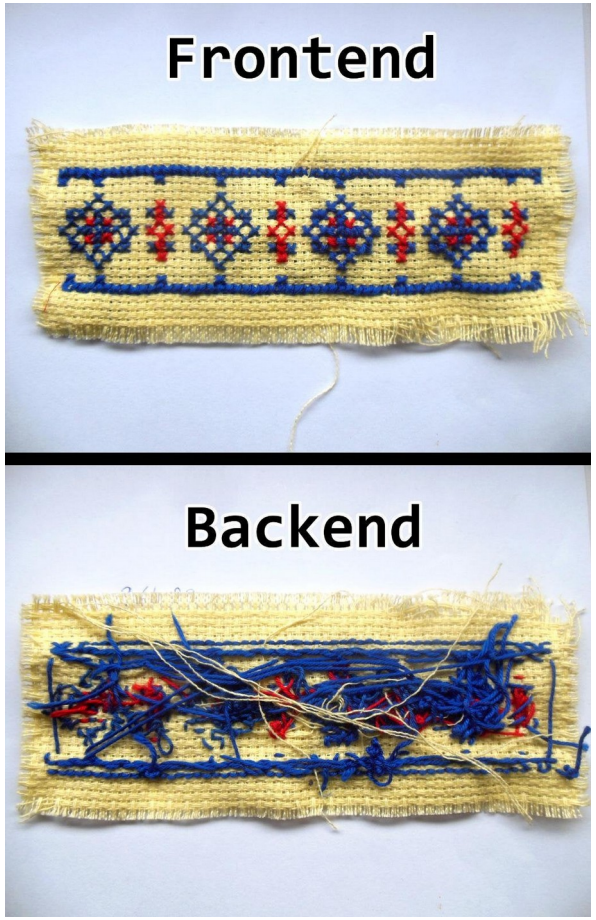
So, Full Stack is?

Pancakes!

No, but it’s really:

- The ability to switch between front and back-end development easily.
- The ability to “walk a mile in their shoes” and understand what the other developer needs to be successful.
- A “jack-of-all-trades” that can easily be team lead or can be a one person army for a small company.

Which looks like



But also sometimes this

Development VS Design

- Back End
  - Makes it work
  - Focus
    - Security
    - Data management
    - Stability
- A good backend dev is one you don’t know exists

- Front End
  - Makes it pretty (still has to work tho)
  - Focus
    - Aesthetics
    - UI/UX
    - Presentation
  - A good front end dev is a digital artist

In Case You’re Interested...

- https://www.webdesignmuseum.org/web-design-history

Full Stack Web development

Lecture 1: HTML

HTML Introduction

- What is HTML?
  - Hyper Text Markup Language
  - Defines the base structure for a webpage.
  - Fundamental building block for every other web programming language.
  - Set of nested tags that define where elements display.

Basic HTML Tag

`<html>` - an HTML element

`<input type="text" value="test">` - an HTML element with two attributes

`<a href="test.htm">Link</a>` - an HTML element with one attribute and inner content

`<span>This is <em>important</em>.</span>` - two nested HTML elements

<ELEMENT ATTKEY=ATTVALUE>

Basic HTML Page

```
<!doctype html>
<html>
  <head>
  <title>Full Stack Developer</title>
  </head>
  <body>
  I am going to be a developer!
  </body>
</html>
```

BODY Section

Tangible parts of the page (the layout) go into the BODY section.

- Headings
  - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- Paragraphs
  - <pre>,
, <p>
- Text (formatting)
  - , , *<b>, <i>, <font>, *
- *Hyperlinks*
  - *<a> (Internal, External, Anchors)*
- Images
  - <img src=“” alt=“”>
- Lists
  - , <ol>, nesting
- Forms
  - <button>, <input>, <select>, <textarea>
- Layout
  - , , <table>, <th>, <tr>, <td>

BODY Section

- Attributes:
  - Regular Attributes: href, type, width, height
  - Global Attributes: draggable, style, tabindex
  - Event Attributes: onload, onclick, onblur, onfocus

- For a complete list of HTML tags and attributes:
  https://www.w3schools.com/tags/
  https://www.w3schools.com/tags/ref_attributes.asp
- HTML 5 Specification
  https://dev.w3.org/html5/html-author/

HEAD Section

Intangible parts of the page (the metadata) go into the HEAD section.

- Meta Tags
  - <meta>
    - Keywords
    - Description
    - Social Media tagging
    - Viewport
    - Cache Control
- In Page CSS Styles
  - <style>
- External Files
  - <link>
    - Prefetching
    - External CSS Files
  - <script>
    - External Javscript Files
    - Embedded Javascript Functions

Let’s Build a Page

Open a blank text editor

- Notepad
- Wordpad
- Even Word (though this has major issues later lol)

Save the file as <firstname_lastname>.htm

Reopen using File → Open and get ready to code

Lab

- Build an “about me” page that includes at least one image and multiple headings.

- Be as creative as possible
  - Describe yourself
  - Use ul or li tags
  - Look at other pages on the web and try to mimic them
  - BE ORIGINAL

Full Stack Web Development

Lecture 2: Setting up your environment

Setting up your dev environment

- What do we need to do?

- What OS are you using?
- Create your “workspace”
- Get git
- Get IDE
- Set up a local server

What are you running on?

- Windows VS Mac & *nix

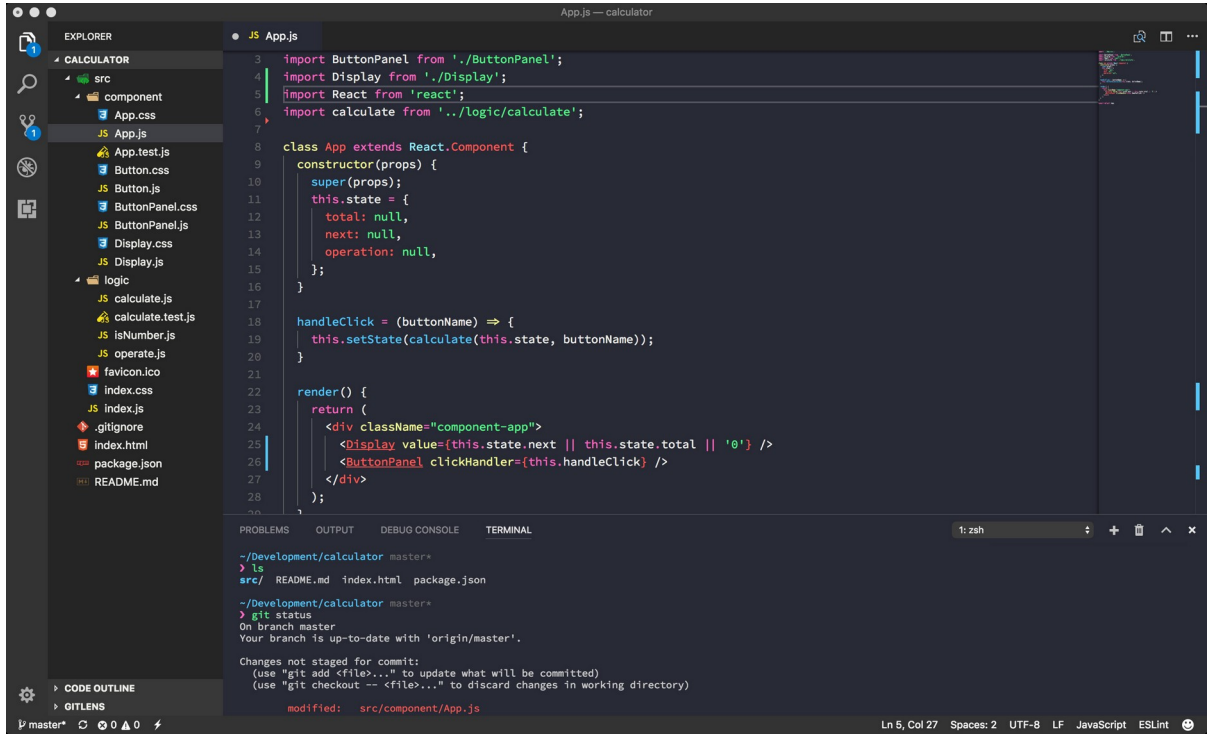
- Windows/PC
  - Easy to use
  - More difficult to develop on

- WTF is *nix?
  - Linux/Unix
  - Mac is the pretty cousin
  - CLI (command line interface)

Integrated Development Environments

- An IDE is a tool used for writing code. It offers lots of extensions that make writing web pages easier.

- In this class we will use a variety of online tools and most in-class demos will be done with VSCode (https://code.visualstudio.com)



Where does the code live?

- Set up your “workspace”

- PC users:
  - My Documents or Desktop is a good place to start
  - Right click ⟶ create new folder
- Mac/*nix
  - Start in your home directory
  - type: cd ~ (unless you did something strange, then it’s on you...)

- Now create a directory named “workspace”
- Store all coding related files in this directory

Time to download

- Git!!
  - not girls in tanktops... hr will disapprove
  - Version Control System
  - Download →https://git-scm.com/downloads
  - Install →https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
  - Interesting fact: Linus Torvalds (the guy who invented linux) sarcastically quipped about the name git (which means "unpleasant person" in British English slang): "I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'." The man page describes Git as "the stupid content tracker".

More downloading

- IDE
  - Integrated development environment
  - We will use VS Code
  - Other options exist
    - Eclipse
    - PHPStorm
    - Notepad++
    - VIM (if your a masochist)

Almost Done!

- VS Code!!
- Download → https://code.visualstudio.com/download
- Install
  - this is super straight forward
  - if you have issues let me know
- Open IDE
- Most IDEs will now ask you to set your, workspace some handle this on save (like vscode)
- Download and install any extensions or plugins as you see fit
  - these are completely subjective **you** will need to find out what works for you...

Last Thing… Maybe…

- Lastly you MIGHT need a development server esp if using windows
  - If *nix use apache
    - Google will be your friend here
  - if mac use mamp
    - http://ampps.com/mamp
  - if windows use wamp
    - https://www.wampserver.com/en/

Congrats!!!