LEARN TO CODE & DO <BIG> SOMETHING </BIG>

Review Programming

and Command Line Fun!

welcome.java

```
if (student == “sitting”) {
  puts “Welcome to Tech Talent South! Please boot up your computer!”
} else if (student == “standing”) {
  puts “Please find a seat!”
} else {
  puts “Please stop laying on the floor, it’s gross down there…”
}
```

Lesson Overview

1. Introductions!
2. Ice Breaker
3. Keys to Success
4. Intro to Programming, Markup and Terminal
5. Java and Spring Boot
6. Environment

Introductions

<Image Source>

1. Tell us your name and a little bit about yourself.
2. What types of software are you interested in developing, and why?
3. Why this class?
4. Tell us one cool/interesting thing about you!

Keys To Success!

- Get in the right mindset!
  - Think about the long-term big picture and don't let the short -term frustration distract you.
- You ***must*** put in the time.
- Take ownership of your education.
- Try teaching. To teach is to learn!
- Pair programming: two minds are always better than one! This type of thing is done a lot in professional environments.
- Don't be afraid to fail. Most code doesn't run the first time. It's about the process, not about getting it to work the first time.
- Get out of your comfort zone. The more time you spend out of your comfort zone, the bigger it gets.
- Be active and engaged in the development community. Go to meet-ups and stay on top of new technologies. Always be learning!

What You Should Expect

Every class, you will get materials, step through ***code*** with us, work on challenges, and do your homework.

- Materials will be published in **TiTuS**.
- We'll review homework at the start of each class, to answer any questions.
- We'll start on time - instructors will be here before class.

Intro To Programming

<Image Source: https://media.giphy.com/media/u0LBtLR7h3RXa/giphy.gif>

Computer programming is:

Telling a computer what to do via a given set of instructions.

What are some things that come to mind when you hear the words "computer programming" or "writing software"?

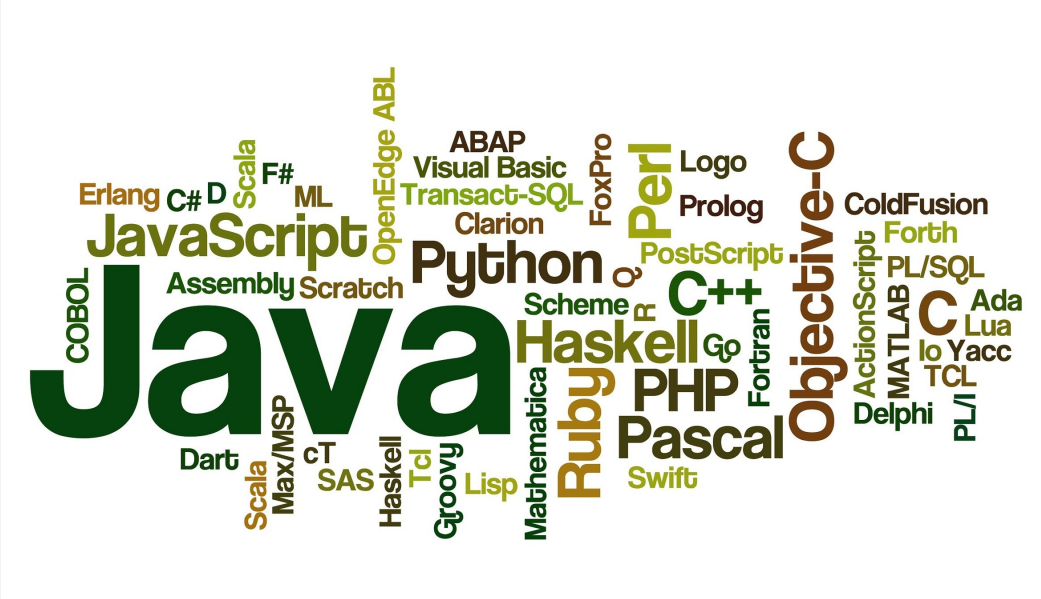
What are some "tech" terms you've heard of that you are curious about?

Are Computers Smart?

Computers are fast, not smart. They only do what they are told - no more, no less. Computers may seem smart, but that's because they were programmed to behave that way. Computers are really great at calculations, but should probably leave the higher level thinking to us.

<Image Source: https://img.memecdn.com/scumbag-computer_c_393995.jpg>

Programming Languages

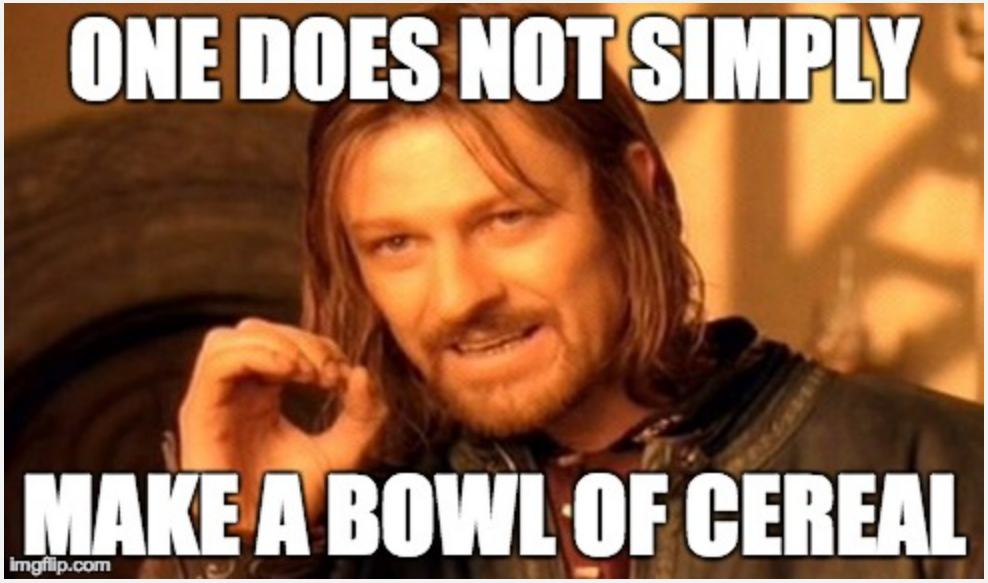


<Image Source: https://i0.wp.com/www.learnsteps.com/wp-content/uploads/2017/04/programming-languages.jpg?zoom=2&resize=840%2C485&ssl=1>

In order to program a computer, we have to use a programming language. There are many kinds of programming languages. For our class, we will be using a programming language called "Java". It is a very powerful object-oriented programming language.

Classroom Challenge!

Write instructions (*a program*) for making a bowl of cereal.



<Image source: https://i.imgflip.com/gnxt5.jpg>

Hint: Be specific. Nothing can be assumed.

Classroom Challange

OK, as you can see, computers can only do things that you tell them to do, and you have to be very specific and detailed when you do so.

So, how do computers seem so smart when they can only understand such boring and long instructions?

Programmers!!

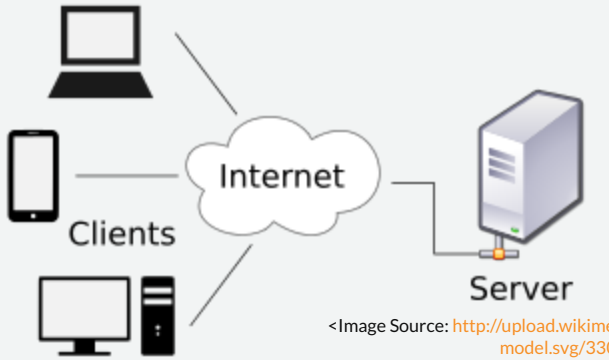
Markup Languages

Markup languages are used to change the way that basic text is displayed. They do this by using "tags" or elements to control how the data is presented. It can be text, images, forms, etc.

Below is how HTML is written but is then rendered and viewable in its final form in a browser. What is a browser?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Markup Languages</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

Client-Server Model



<Image Source: http://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Client-server-model.svg/330px-Client-server-model.svg.png>

When browsing the web, the client makes requests to the server which then sends back a payload of information. One of the most familiar client-side applications is the **browser**. Its purpose is to organize the information that is returned from the server. In web development, programming languages might run on the client side or the server side.

Ex: JavaScript is run on the client side, but Java, Ruby, Python, and PHP are executed on the server side, then rendered using HTML.

How Does the Web Work?

The browser uses something called

HTTP (*hypertext transfer protocol*)

which is the underlying technology that makes it possible for a **client** (your computer, laptop, smartphone) to interact with a **server** over the web. The server is where your programs run (i.e. your Spring Boot programs).

What is a protocol? Can you think of more protocols?

HTTP Verbs

Pro Tip: Memorize these.

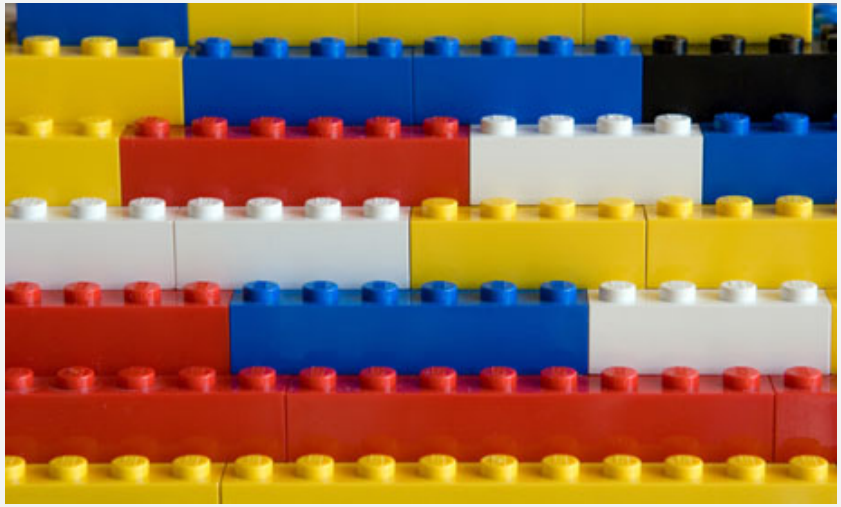
- **GET - Retrieve items from resource.**
- **POST - Create new item in resource.**
- **PUT - Replace existing item in resource.**
- **PATCH - Update existing item in resource.**
- **DELETE - Delete existing item in resource.**

This relates to RESTful design.

A few rules of safety...

Convention over Configuration

CONVENTION (i.e., placing Lego Blocks neatly together) OVER **CONFIGURATION** (i.e., using glue, paper machete, etc. = Hacking it together, inefficient way of getting things done)



<Image Source: http://static.guim.co.uk/sys-images/Arts/Arts_/Pictures/2012/12/21/1356105794135/Fits-the-bill---a-stack-o-010.jpg>

Don't Repeat Yourself

This is primarily to maintain clean code, which in turn:

- Makes coding **FUN** rather than **TEDIOUS**
- Streamlines production and collaboration.
- Reduces errors.

How?

- Reuse code, don't duplicate. (When possible!)
- Take advantage of the tools you have!
  - Those given to you by Java and SpringBoot

Environment Setup

Make sure you install:

An IDE

- Eclipse

A Browser

- We highly recommend Chrome
- Firefox will work as well.
- Safari is passable; IE should ***never*** be used.

A Command Line Processor

- Default Mac is called Terminal, or the "Shell"
- Default Windows is Called Command Prompt
  - You can also use Windows Powershell
  - Or the git bash shell

Environment Setup

Install Java:

https://java.com/en/download/help/download_options.xml

*Windows should use the online installation

Install the JDK (Java Developer Kit):

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Select the download option on the left

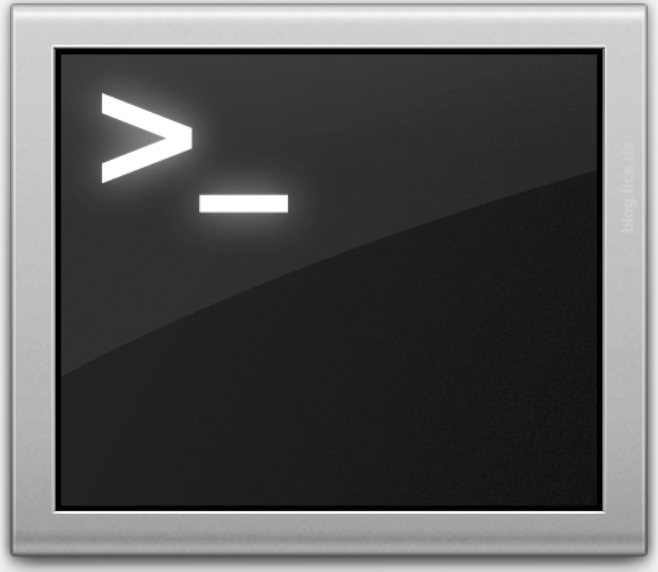
lol JDK?

The Java Developer Kit (JDK) is the development environment for Java. It packages together the Java Runtime Environment (JRE), an interpreter (java), and a compiler (javac) among other useful tools.

We'll experience all of these things in more depth further down the road!

Navigating Your Machine via Command Line

Getting comfortable with the command line is an essential part of becoming a skilled web developer. Once you get the hang of it, you'll be able to navigate to directories, run programs, and launch applications much faster than pointing and clicking. It may feel a bit awkward at first, after years of relying on a GUI interface, but the more you use the command line, the more efficiently you can get things done. Saving 10 seconds here and 30 seconds there starts to really add up!



<Image Source: http://newsourcemedia.com/blog/wp-content/uploads/2011/09/img_Terminal.png>

Essential Commands

cd (followed by directory name) - change directory

cd .. - go up / back a directory

ls (dir *in Windows***)** - list contents of current directory

pwd - print working directory

mkdir - make directory

touch (null >> in Windows followed by file name + file extension) - make a new file

open (start in Windows**) -** opens file in default program

Essential Commands

mv (move in Windows**) [from_path] [to_path] -** move file or directory, also used to rename

cp (copy in Windows**) [from_path] [to_path]** - copy file or directory

rm (del in Windows**) [filename]** - deletes a file, and with some flags can delete a directory as well (**rmdir** is also available for directory removal)

Up Arrow Key - will bring back recent commands you've typed in. This is a great shortcut since you'll often be using the same commands again and again.

Activity

- **cd** or **cd ~** to get to our root
- **ls**
- **mkdir LionKing** (can't have spaces or it will make 2 directories)
- **cd LionKing**
- **mkdir goodLion badLion**
- **ls**
- (open up our GUI to see what is actually going on)
- **cd goodLion**
- **touch simba.html**
- **touch mufasa.html nala.jpg**
- **pwd**
- **up arrow, down arrow** (cycle through previous commands)
- **mv nala.jpg scar.jpg**
- **mv scar.jpg ../badLion**
- **rm *.html**
- **cd**
- **rm -r LionKing**