

PROGRAMMING BASICS

Intro to The Command Line and Coding
Open Data Day 2016

Purpose of Today's Class

In today's class we learn about command line environments, getting started with GitHub, and basic programming logic.

What we will Accomplish Today

- We will create our own GitHub profile
- Learn about the basics of a command line environment
- We will use cloud9 to play around with our Linux environment
- Learn some core programming concepts
- Write our first program
- Run our first program
- Learn how to set up our own environment on our home machine

Today We Will Learn How Learn to Code



Why Code?

“I think everyone should get a little exposure to computer science because it really forces you to think in a slightly different way, and it’s a skill that you can apply in life in general, whether you end up in computer science or not” - Tony Hsieh

“Software touches all of these different things you use, and tech companies are revolutionizing all different areas of the world...from how we shop to how farming works, all these things that aren't technical are being turned upside down by software. So being able to play in that universe really makes a difference” - Drew Houston

“Coding is today’s language of creativity. All our children deserve a chance to become creators instead consumers of computer science” - Maria Klawe

What do you wish you knew before you got started coding?

“Be more fearless and break your computer. You learn the most in the debugging” – Tanya Tarr

“Learning a new pattern is much more valuable than a new language” – Travis Hoppe

“Even the most experienced Developers utilize a lot of trial and error. Struggling through code and asking google is natural” – Nicholas Marcouiller

My Story

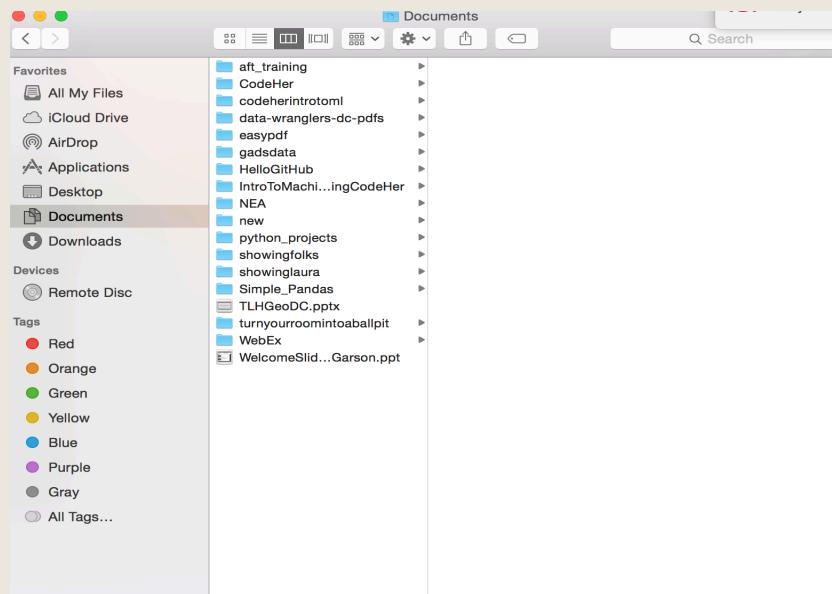


Introductions – Turn to someone you don't know and ask as follows

- Your name
- Favorite movie
- What brings you here today

What is the command line?

- The command line is a way of interacting with the computer but without the graphical interface we are used to in programs like finder on the computer



```
JessicasMacBook:documents jessicagarson$ ls
CodeHer
HelloGitHub
IntroToMachineLearningCodeHer
NEA
Simple_Pandas
TLHGeoDC.pptx
WebEx
WelcomeSlide For DC - Jessica Garson.ppt
aft_training
codeherintrotml
data-wranglers-dc-pdfs
easypdf
gadsdata
new
python_projects
showingfolks
showinglaura
turnyourroomintoaballpit
~$TLHGeoDC.pptx
~$WelcomeSlide For DC - Jessica Garson.ppt
~$aft_first_model.xlsx
~$codeher_first_model.xls
JessicasMacBook:documents jessicagarson$
```

Linux, Unix and Other Systems

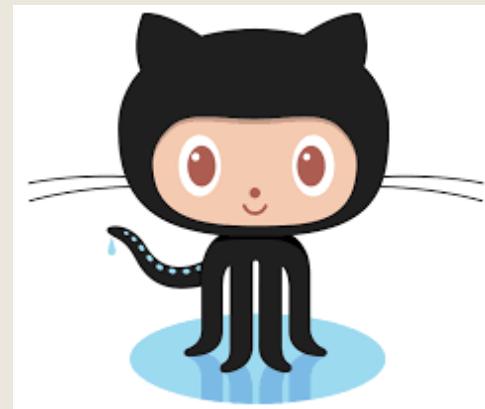


Linux



UNIX

Git vs Github



What is Open Source software?



open source

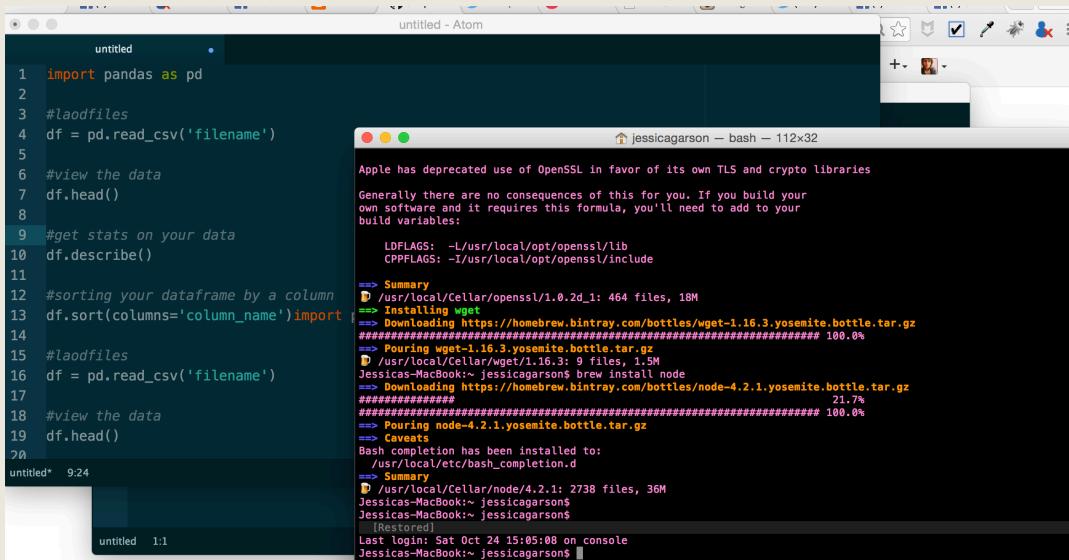
A few other key terms

- Bash scripting - A script for a computer tells the computer what it should do or say. In the context of Bash scripts we are telling the Bash shell what it should do. A Bash script is a plain text file which contains a series of commands.
- Shell - A shell is a program that provides the traditional, text-only user interface for Linux and other Unix-like operating systems.

What can we build with code

- Websites
- Games
- Predictions
- Toys
- Wearable tech
- Make our lives easier

What do you need to code?



A screenshot of a Mac desktop showing a terminal window and an Atom code editor. The terminal window is active and displays a Homebrew upgrade process. The Atom editor shows a script with code related to data processing and file operations.

```
untitled
```

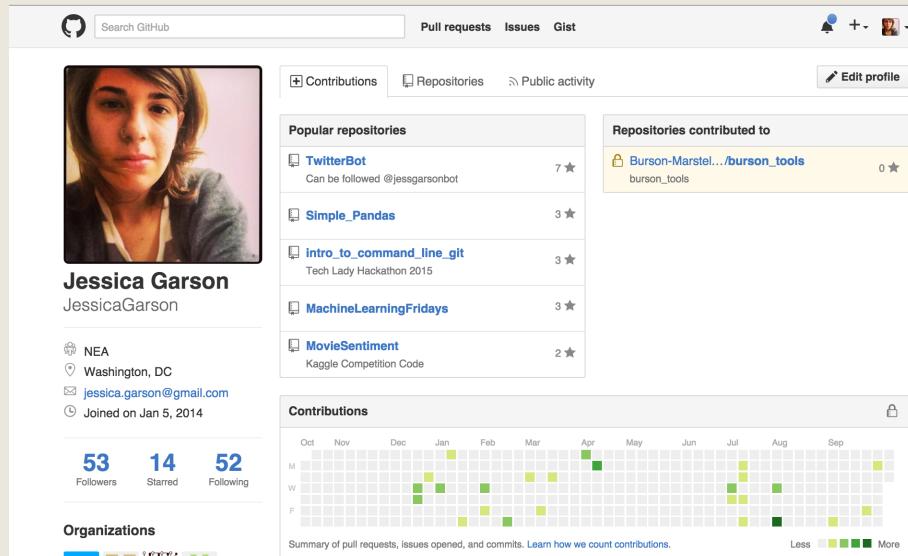
```
1 import pandas as pd
2
3 #laodfiles
4 df = pd.read_csv('filename')
5
6 #view the data
7 df.head()
8
9 #get stats on your data
10 df.describe()
11
12 #sorting your dataframe by a column
13 df.sort(columns='column_name').import
14
15 #laodfiles
16 df = pd.read_csv('filename')
17
18 #view the data
19 df.head()
20
```

```
jessicagarson — bash — 112x32
```

```
Apple has deprecated use of OpenSSL in favor of its own TLS and crypto libraries
Generally there are no consequences of this for you. If you build your
own software and it requires this formula, you'll need to add to your
build variables:
  LDFLAGS: -L/usr/local/opt/openssl/lib
  CPPFLAGS: -I/usr/local/opt/openssl/include
=> Summary
  /usr/local/Cellar/openssl/1.0.2d_1: 464 files, 18M
=> Installing wget
=> Downloading https://homebrew.bintray.com/bottles/wget-1.16.3.yosemite.bottle.tar.gz
#####
=> Pouring wget-1.16.3.yosemite.bottle.tar.gz
  /usr/local/Cellar/wget/1.16.3: 9 files, 1.5M
Jessicas-MacBook:~ jessicagarson$ brew install node
=> Downloading https://homebrew.bintray.com/bottles/node-4.2.1.yosemite.bottle.tar.gz
#####
=> Pouring node-4.2.1.yosemite.bottle.tar.gz
  21.7%
#####
=> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d
=> Summary
  /usr/local/Cellar/node/4.2.1: 2738 files, 36M
Jessicas-MacBook:~ jessicagarson$
Jessicas-MacBook:~ jessicagarson$
[Restored]
Last login: Sat Oct 24 15:05:08 on console
Jessicas-MacBook:~ jessicagarson$
```

Let's make our own GitHub profile

www.github.com



Let's Create Our Cloud 9 account
<https://c9.io/web/sign-up/free>



Activity – 20 Questions

Let's as a class play a game

If you don't remember how to play 20 questions here is a quick recap:

One person thinks of an object and the other player gets to guess up to 20 questions to guess the object

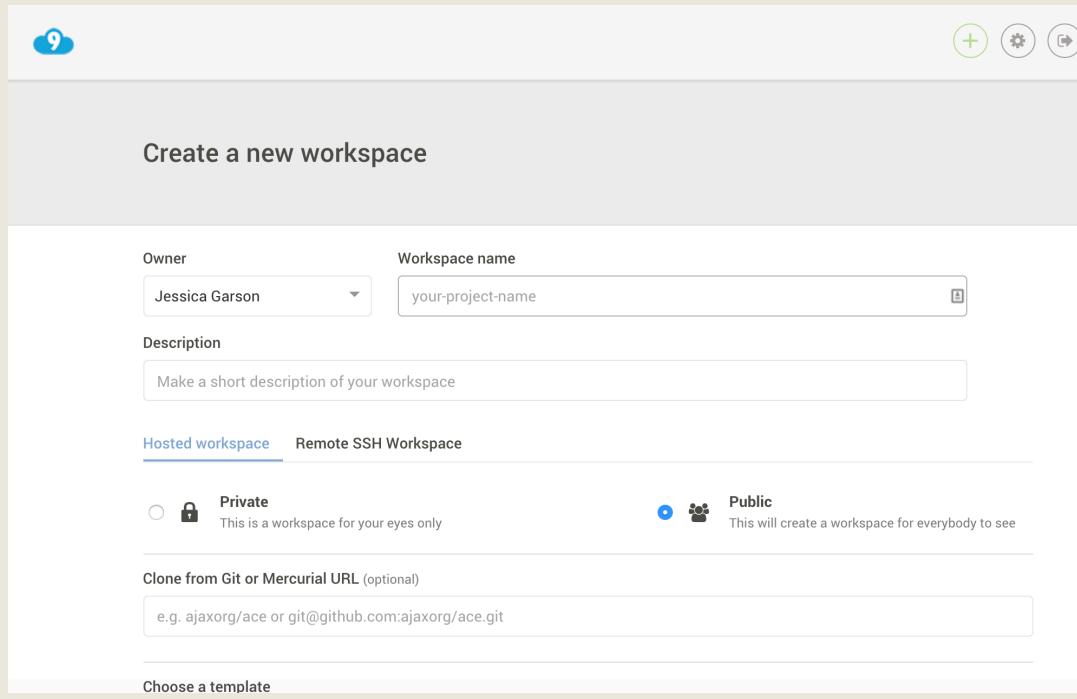
Debrief

- What logic did you use to help make a guess?
- What strategies were effective?
- What strategies proved ineffective?

Computer Concepts We Used

- Computational thinking
- Linear search
- Binary search
- Divide and conquer
- Comparing algorithms

Setting Up in Cloud 9



The image shows the 'Create a new workspace' interface on the Cloud 9 platform. At the top, there's a header with a cloud icon and three circular buttons: a green plus sign, a gear, and a refresh arrow. Below the header, the title 'Create a new workspace' is centered. The form fields include:

- Owner:** A dropdown menu showing 'Jessica Garson'.
- Workspace name:** An input field containing 'your-project-name'.
- Description:** A text area placeholder 'Make a short description of your workspace'.
- Hosted workspace** (underlined) and **Remote SSH Workspace**.
- Visibility:** Two options: 'Private' (with a lock icon) and 'Public' (with a paw print icon). The 'Public' option is selected.
- Clone from Git or Mercurial URL (optional):** An input field with the placeholder 'e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git'.
- Choose a template:** A section with a dropdown menu currently set to 'None'.

Where am I?
pwd

```
jessicagarson1:~/workspace $ pwd  
/home/ubuntu/workspace  
jessicagarson1:~/workspace $ █
```

Show the manual page for a command
man

```
jessicagarson1:~/workspace/class $ man pwd  
jessicagarson1:~/workspace/class $ █
```

Make a new directory

mkdir

```
jessicagarson1:~/workspace $ mkdir class  
jessicagarson1:~/workspace $ █
```

Change Directory
cd directory_name

```
jessicagarson1:~/workspace $ cd class
jessicagarson1:~/workspace/class $ █
```

Some key commands for changing directories

```
jessicagarson1:~/workspace/class $ cd ..
jessicagarson1:~/workspace $ cd
jessicagarson1:~ $ █
```

Make new file
touch file_name

```
jessicagarson1:~/workspace/class $ touch new_file.txt  
jessicagarson1:~/workspace/class $ █
```

List contents of a directory

ls

```
jessicagarson1:~/workspace/class $ ls  
new_file.txt  
jessicagarson1:~/workspace/class $ █
```

List the longer contents of a directory

`ls -l`

```
jessicagarson1:~/workspace/class $ ls -l
total 8
drwxr-xr-x 2 ubuntu ubuntu 4096 Dec 17 02:29 dec/
-rw-r--r-- 1 ubuntu ubuntu   310 Nov  6 01:07 pythonsecondfile.py
```

Remove a File

rm

```
jessicagarson1:~/workspace/class $ rm jess.txt  
jessicagarson1:~/workspace/class $ █
```

Move a file

mv

```
jessicagarson1:~/workspace/class $ mv helloworld.py dec/helloworld.py  
jessicagarson1:~/workspace/class $ █
```

Clear the contents of command line window
clear

```
jessicagarson1:~/workspace/class $ clear
```

Create a tree diagram of contents

Tree

```
jessicagarson1:~/workspace/class $ tree
.
└── dec
    └── helloworld.py
    └── pythonsecondfile.py

1 directory, 2 files
```

Remove a directory
rm –r dirname

```
jessicagarson1:~/workspace/class $ rm -r dec
jessicagarson1:~/workspace/class $ █
```

Read a file cat

```
jessicagarson1:~/workspace/class $ cat pythonsecondfile.py
name_of_student = "Jess"
print name_of_student

statement = "Hello {0}, it's nice to meet you".format(name_of_student)
print statement

color = raw_input("Whats your favorite color? ")
response = "Well {0}, {1} is a great color, nice _ to meet you, have a great day".format(name_of_student, color)
```

Tells you how many lines, words and characters there are in a file

WC

```
jessicagarson1:~/workspace/class $ wc pythonsecondfile.py
    8 43 310 pythonsecondfile.py
```

A few tricks

- Tab complete
- The up arrow
- The down arrow

Let's Play Around On The Command Line

- Let's take a few minutes to create a few files, move some files and generally play around to get to know the command line.

Our First Python Program



Our First Python Program

```
1 print "Hello World"
```

```
2 |
```

Running our first program

```
jessicagarson1:~/workspace/class $ python helloworld.py  
hello world
```

Our 2nd Python Program

```
1 name_of_student = "Jess"
2 print name_of_student
3
4 statement = "Hello {0}, it's nice to meet
• you".format(name_of_student)
5 print statement
6
7 color = raw_input("Whats your favorite color? ")
8 response = "Well {0}, {1} is a great color, nice to meet
• you, have a great day".format(name_of_student, color)
9 print response
10
```

My Favorite Thing About Code

- There is no right answer here. There are many ways to write a program that accomplishes the same goals in different ways. Feel free to make changes here and make this your own.

Challenge

- Create your own text based game

Let's play and around and adjust the 2nd program

- Make changes as you see fit
- Run the program when you are ready
- Let me know if you have any questions

Do This at Home

Shell Environments

- On a mac – there is a Linux environment that comes pre-installed called the terminal
 - *To get to the terminal go to finder select utilities and inside the folder you should*
- On a PC – There are many options but I really like babun - <http://babun.github.io/>

Text Editors

- Sublime Text - <http://www.sublimetext.com/>
- Atom - <https://atom.io/>
- Notepad ++ for PC only - <https://notepad-plus-plus.org/>

Installing Python on your Computer – Macs

- Python comes bundled with Macs already, so there's no extra setup for now. You'll want to check which version you have, but it's not super important for writing your first scripts.

Installing Python on your Computer – PC and Linux

Running Python in Windows or Linux

Download and install Python from here: <http://www.python.org/download/releases/2.7.6/>

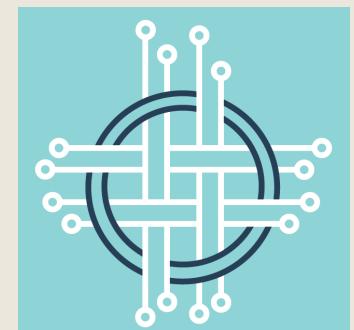
I prefer to use IDLE (bundled with Python) over the command prompt when possible -- simply put, it's easier to use. If you need to run Python scripts from the command line, the Mac instructions below will work with minimal adjustment.

When you run IDLE, you'll automatically be in the interactive interpreter mode where you can run Python commands one at a time.

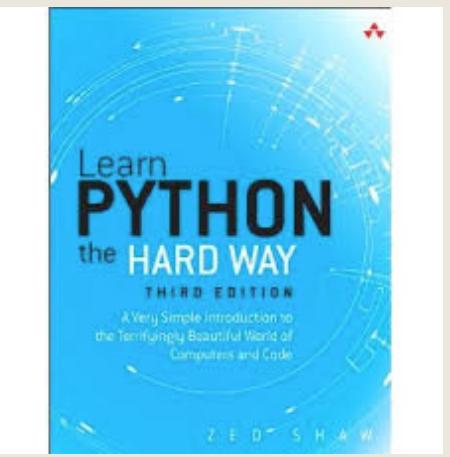
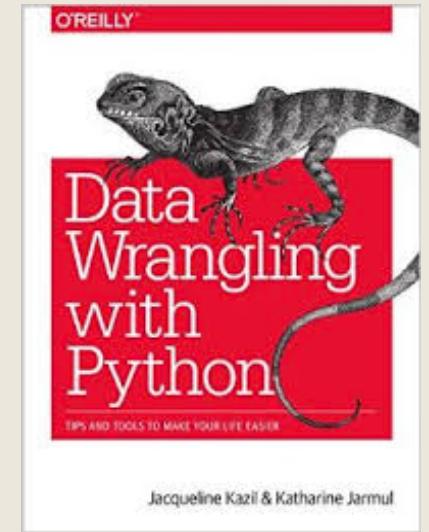
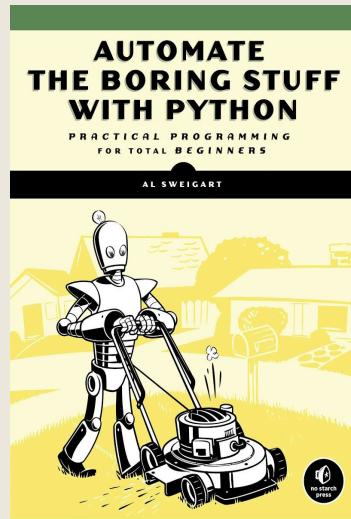
To create a new file from IDLE, go to File > New Window. A new, blank screen will open up where you can create your Python projects.

When you have a Python file open in IDLE, you can run it at any time by pressing **F5**.

The coolest thing about DC right now



A few resources I recommend



kaggle™

Resources for Learning How to Code

- What have you tried?
- What's worked well for you?
- What hasn't worked for you?
- What can we try?

Questions, Comments, Feedback

- Let us know if you have any

Contact

Jessica Garson

Jessica.Garson@gmail.com

Jessica.Garson@bm.com

Twitter: @jessicagarson