

# DAT12SYD

May 2018



# Welcome!

DAT12SYD

# Who am i?

1. Founder of [The Cognitive Company](#)
2. Data Science Instructor at General Assembly
3. Manager Deloitte Consulting - Analytics and Information Management
4. Dropped out of 2 PhD programmes



@RoboAndy



@RoboAndy



Andrew Szvec



andrew@thecognitiveco.com

# Who am i?

1. Data Scientist @ \_Servian
2. General Assembly Data Science Immersive
3. National Account Manager @Lavazza
4. Master degree in Econometrics and Quantitative Analysis



Stefano Rossi

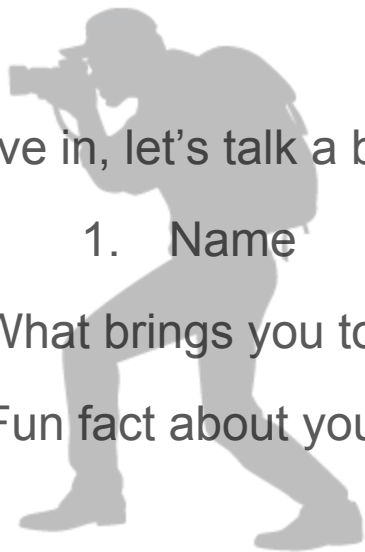


sroujo@yahoo.com

# About you?

Before we dive in, let's talk a bit about you!

1. Name
2. What brings you to GA?
3. Fun fact about yourself!



# Expectations

While I am presenting don't be afraid to ask questions

If a class member is speaking please respect them

During the coding activities ask the person to your left and right for the answer  
before asking Stefano or I

# Agenda Day 1

1. Install and Setup
2. Intro to Data Science and core topics
3. Python Fundamentals

# Learning Objectives





# Learning Objectives

- Describe the roles and components of a successful development environment.
- Define data science and the data science workflow.
- Apply the data science workflow to solve a task.
- Discuss common data science terminology and processes.
- Define what a type is and what kinds exist in Python.
- Define a function and identify common functions in Python.
- Define control flow and some common examples in Python.

# Install Fest



# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Anaconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil

# Slides

Download links

from

<https://goo.gl/gVvmkQ>

WiFi

GA-Guest  
yellowpencil

# Development Env Check

## Part 1. Operating System

You can be a data scientist on any operating system. In general, most professionals choose a UNIX-type OS; typically Apple's OS X or a popular Linux distribution, such as Ubuntu. If you're already using Mac or Linux, great! Skip ahead to Part 2 and get started with your installs.

However, there is a growing need for (and interest in) data science in industries that traditionally use PCs. If you're on a Windows machine, that's ok too! You'll just need to install an additional piece of software to provide a development environment similar to OS X and Linux.

Click [here to download the Git Bash shell](#). This will allow you to emulate most of the common commands and functions native to OS and Linux systems.

# Anaconda 101



# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Ananconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil



# Development Env Check

## Part 2. Anaconda Installation

In this course, we'll be working closely with tools that utilize the Python programming language. Anaconda is a popular cross-platform tool that helps install and manage Python-related data science libraries.

1. [Download Anaconda](#) and follow the installation instructions package for your operating system. Please make sure that you're downloading the latest stable version for Python 3!
2. Agree to the terms and let Anaconda complete its default installation.
3. Once installed, navigate to your command line (on OS X, this is the terminal application; on Windows, use your new `Git Bash` shell) and confirm that it's installed by typing in the `which conda` command.





# Activity Make Anaconda Env

```
conda create -n dat2018 python=3.6
```

```
conda info --envs
```

```
source activate dat2018
```

```
pip install numpy sklearn pandas
```

```
source deactivate
```



EXERCISE



# Development Env Check

You should see:

```
$ which conda
/Users/USERNAME/anaconda3/bin/conda
```

- If the command line returns a file path (like in the example below), you've successfully installed Anaconda.
  - If the command line returns nothing (and sends you back to the prompt), check in with your instructor.
    - Note: Your file path may look different.
    - Note: You'll often see commands that look like: `$ which conda` above — when you see those, type in everything except the dollar sign. The dollar sign is used to denote a code prompt in your window.
1. Once installed, run the following command to ensure that some frequently used libraries are installed. Anaconda may also update your packages at this time (which is OK!).
  - 2.

```
conda install jupyter notebook python matplotlib nltk numpy pip setuptools scikit-learn scipy statsmodels
```

# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Ananconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil

# Development Env Check



## Part 3. Git Configuration

1. To check if your Git installation was successful, open a new terminal window and try to run Git from the command line:

```
$ git --version
```

The output should be something like this:

```
$ git --version  
git version 2.X.X
```

# Development Env Check



## Part 3. Git Configuration

1. Next, you'll need to provide Git with your name and email. Make sure to use the same email address that you registered at <https://git.generalassemb.ly>:

```
$ git config --global user.name "Your Name"
$ git config --global user.email your.name@example.com
```

These identifiers will be added to your commits and show up when you push your changes to [GitHub](#) from the command line!

# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Ananconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil

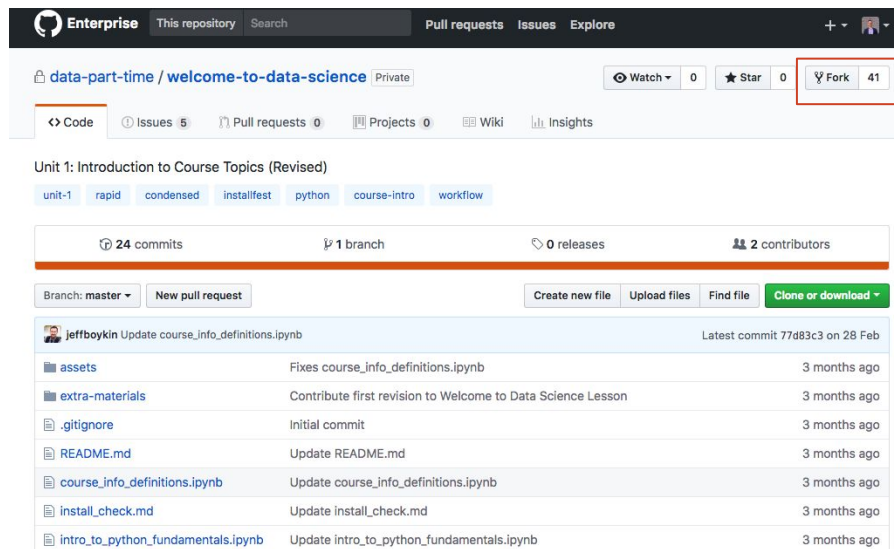
# GitHub



Navigate here:

<https://git.generalassemb.ly/andrewszwec-ga/welcome-to-data-science.git>

Fork repo:



The screenshot shows the GitHub interface for the repository 'data-part-time / welcome-to-data-science'. The repository is private and has 0 stars and 41 forks. The 'Fork' button is highlighted with a red box. Below the repository name, there are tabs for 'Code', 'Issues' (5), 'Pull requests' (0), 'Projects' (0), 'Wiki', and 'Insights'. The main content area shows 'Unit 1: Introduction to Course Topics (Revised)' with sub-tabs for 'unit-1', 'rapid', 'condensed', 'installfest', 'python', 'course-intro', and 'workflow'. Below this, there are statistics: 24 commits, 1 branch, 0 releases, and 2 contributors. At the bottom, there is a table of recent commits by 'jeffboykin'.

Commit	Message	Time
assets	Fixes course_info_definitions.ipynb	3 months ago
extra-materials	Contribute first revision to Welcome to Data Science Lesson	3 months ago
.gitignore	Initial commit	3 months ago
README.md	Update README.md	3 months ago
course_info_definitions.ipynb	Update course_info_definitions.ipynb	3 months ago
install_check.md	Update install_check.md	3 months ago
intro_to_python_fundamentals.ipynb	Update intro_to_python_fundamentals.ipynb	3 months ago

(<https://git.generalassemb.ly/data-part-time/welcome-to-data-science>)

# Github



Use the terminal to clone repo:

```
$git clone https://git.generalassemb.ly/<your-name-here>/welcome-to-data-science.git
```

```
$cd welcome-to-data-science
```

Configure your local clone to point to the official course repository

```
$ git remote -v
```

```
$ git remote add upstream https://git.generalassemb.ly/andrewszwec-ga/welcome-to-data-science.git
```

```
$ git remote -v
```



# GitHub



Ensure you're in the master branch

```
$ git checkout master
```

Grab the latest changes from the master

```
$ git fetch upstream
```

Merge the master changes with your repo

```
$ git merge upstream/master
```

*WARNING: Be careful not to overwrite files you have already changed in your repo unless you want to replace them with the master versions!*

*(Consider renaming yours or doing a PULL REQUEST.)*

# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Ananconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil

# Install Fest

15<sub>min</sub>

Install the following software for your OS:

- Install Ananconda 3.6 (<https://www.anaconda.com/download/> )
- Install Git (<https://git-scm.com/downloads>)
- Install sublime text 3 (<https://www.sublimetext.com/3> )
- Make a github account (<https://github.com> )
- Make a Slack account (<https://slack.com/>, @dat12syd.slack.com )

WiFi  
GA-Guest  
yellowpencil

# Intro to Data Science




# Intro to Data Science

course\_info\_definitions\_v2.ipynb

Slide Type

Slide



## What is Data Science?

*Authors: Alexander Egorenkov (DC), Amy Roberts (NYC)*

Slide Type

Slide

### Learning Objectives

*After this lesson, you will be able to:*

- Describe the roles and components of a successful development environment.
- Define data science and the data science workflow.
- Apply the data science workflow to solve a task.
- Discuss common data science terminology and processes.


# Python Fundamentals



# Python Fundamentals

intro\_to\_python\_fundamentals\_v2.ipynb

Slide Type Slide



## Introduction to Python Fundamentals ¶

*Authors: Kiefer Katovich (San Francisco), Dave Yerrington (San Francisco), Joseph Nelson (Washington, D.C.), Sam Stack (Washington, D.C.)*

Slide Type Slide

### Learning Objectives

*After this lesson, you will be able to:*

- Define what a type is and what kinds exist in Python.
- Define a function and identify common functions in Python.
- Define control flow and some common examples in Python.

# Terminal 101





# Terminal 101

- What is the terminal => interactive console
- Why do we need it => more powerful than UI
- TL;DR - What are the key commands (all lowercase, case sensitive):
  - cd, mv, cp, mkdir, pwd, rm # Navigating the terminal
  - ls, ls -lsa
  - grep, ps aux, piping |
  - nano, vim, touch
  - wc -l
  - tar czvf, tar xzvf, zip, unzip # Compressing Files
  - ssh, scp
  - whoami, which

# Terminal 101 - What is it?

A way to manipulate and interact with your computer It's entirely text-based

Not the **W.I.M.P** (Windows, Icons, Menus and Pointers) style!

# Terminal 101 - Why use it?

- It's (eventually) very fast
- It's automatable and flexible No interruptions
- It gives us what we expect Sometimes it is the only way
  - Command Line Interaction (C.L.I.)
  - Web servers

# Terminal 101 - The Bash Shell

- Bash is a regular program on your computer It was created to take commands from you
- We talk to it using the **Bash Shell Language**
- When I say "shell", it's just that program we were talking about before
- It's an interface to interact with other programs

# Terminal 101 - What can you do with it?

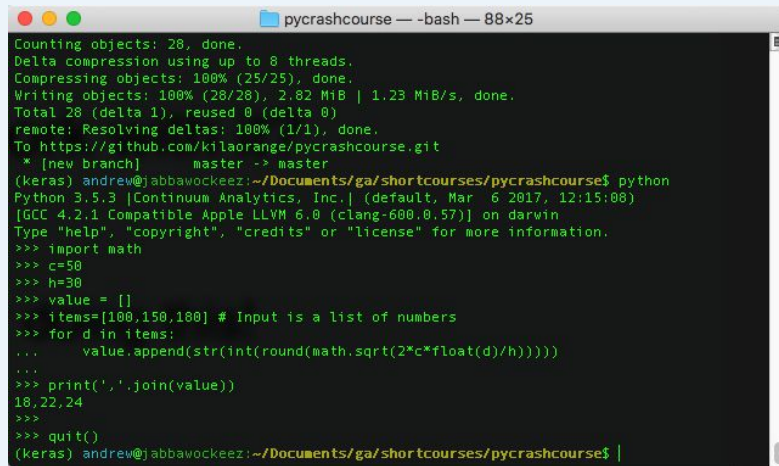
- Most of you will have a lot of experience with the WIMP (Windows, Icons, Menus, Pointer) style of system
- That's not the only way. We are going to be using a text-only "console" or "terminal"
- This is going to seem alien and primitive but you will soon see the power!

# Terminal 101 - What can you do with it?

- Anything! Run programs to make all sorts of changes
  - Editing files and images
  - Converting files between types
- Creating back-ups
- Making and copying files
- Downloading, compiling, and running programs We can do a lot more with the Terminal

# Terminal 101 - How do you work with it?

- Non-interactively
- Running scripts. We are already doing this!
- Interactively
- Opening up a **REPL**

A terminal window titled 'pycrashcourse — -bash — 88x25'. The terminal shows the output of a 'git commit' command, including progress bars for counting, compressing, and writing objects. It then shows the user switching to the 'master' branch and running 'python'. The Python prompt shows the user's location and the Python version (3.5.3). The user enters a series of commands: 'import math', 'c=50', 'h=30', 'value = []', and a loop that calculates and appends values to a list. Finally, the user prints the list and quits the Python interpreter.

```
pycrashcourse — -bash — 88x25
Counting objects: 28, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (28/28), 2.82 MiB | 1.23 MiB/s, done.
Total 28 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/kilaorange/pycrashcourse.git
 * [new branch]      master -> master
(keras) andrew@jabbawockeez:~/Documents/ga/shortcourses/pycrashcourse$ python
Python 3.5.3 [Continuum Analytics, Inc.] (default, Mar  6 2017, 12:15:00)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> c=50
>>> h=30
>>> value = []
>>> items=[100,150,180] # Input is a list of numbers
>>> for d in items:
...     value.append(str(int(round(math.sqrt(2*c*float(d)/h))))))
...
>>> print(', '.join(value))
18,22,24
>>>
>>> quit()
(keras) andrew@jabbawockeez:~/Documents/ga/shortcourses/pycrashcourse$
```

# Terminal 101 - Commands

[tab]	# Autocomplete command
pwd	# Where am I? The programmer's "um"
ls	# List all files in the current directory
cd	# Change Directories
mkdir	# Make a Directory
rmdir	# Remove an empty directory
rm	# Remove a file or a directory [There is no undo]
touch	# Create a file
open	# Open a file in the default application
code	# Open the VSCode Editor (atom will open in Atom)
say	# Make your computer talk



# Advanced Commands

`ls -lsa` # long format, system blocks, view hidden files (.)

`Ls -lt` # long format, sort by time modified (most recently modified first)

`ps aux | grep <keyword>` # process status, all users, usernames, even those without controlling terminal, search for keyword e.g. "python"

`nano, vim` # editors

# Advanced Commands

`wc -l` # count number of lines in file e.g. a csv

`tar czvf, gzip, zip` # compress into archive

`tar xzvf, unzip` # decompress file

`ssh, scp` # Secure shell, secure copy

`whoami, which <keyword>` # list username, which python

# Activity - Terminal 101

1. Navigate to your home directory with **cd ~**
2. Use **pwd** to discover its name
3. Use **ls** to see what is in your home directory
4. Use **cd ~** to navigate back down to your home directory
5. Create a new directory with **mkdir** called **sandbox**
6. Navigate to your downloads with **cd ..** or **cd ~/Downloads**
7. Create a file in **Downloads** with **touch** called file.txt
8. Copy file.txt to your sandbox with **cp file.txt ~/sandbox/**
9. Rename **file.txt** to **hello.py** with **mv file.txt hello.py**
10. Change directory to **cd ~/sandbox**
11. Make a new file called **fake.py** using **nano fake.py**
12. Inside the file type `print("hello world!")`, then push Ctrl+o, Enter, Ctrl+x to save and exit
13. Make a directory called 'crash\_course' using **mkdir crash\_course**
14. Remove **fake.py** and **crash\_course** with **rm**, you will need **-f** for one of the removals
15. Well done you've finished!



EXERCISE

# Additional Resources

Read these for more info:

- [Quick Left's Tutorials](#) - start from the bottom!
- [Learn CLI the Hard Way](#)
- Track down the [Terminal City Murderer](#)
- [40 Terminal Tricks and Tips](#)

# PYTHON CRASH COURSE

---

# Q&A

---

PYTHON CRASH COURSE

---

# EXIT TICKETS

DON'T FORGET TO FILL OUT YOUR EXIT TICKET

# Appendix



# Fun Activities

Online game <http://www.pythonchallenge.com>

<https://selfdrivingcars.mit.edu/deeptraffic/>

<http://playground.tensorflow.org/>