

Welcome to Learn to Code Seattle

Last updated: 1/2/2017 powered by  galvanize

About Galvanize

Dynamic learning community for technology

- Web Development
- **Workspace**
- Data Science
- **Networking**

To learn more,
visit galvanize.com



powered by  **galvanize**

But first...



powered by galvanize

Let's get to know each other

Turn to the person next to you and ask:

- 1) What is your name?
- 2) Why did you come here?
- 3) What is the nerdiest thing about you?

You have 2 minutes to complete this mission!

powered by galvanize

Intro to Python

For Data Science

Last updated: 1/2/2017

powered by  galvanize

In this course you will learn

- ❏ Set up your computer
- ❏ Setting up iPython/Jupyter
- ❏ Basic Python commands
- ❏ Explorations in NumPy, Pandas, and Matplotlib
- ❏ Sandbox time!

Gut check, **Galvanize** style!



- This course is for beginners
- Feel free to move ahead
- Help others when you can
- Be patient and nice
- We'll all get through it!

Want to move ahead? No problem!

Go to: [github.com/
GalvanizeOpenSource/
Python-Resources/](https://github.com/GalvanizeOpenSource/Python-Resources/)

1. Installation *
2. Intro_Scripts *
3. Basic_Resources *
4. Exploring_Data



* Covering this today

What IS Python?

('Tis a silly language...)

The origins of Python

- Created by this guy (Guido Van Rossum) in the 1990s
- Totally named after Monty Python, so pun away!
- Open-source project with a huge community

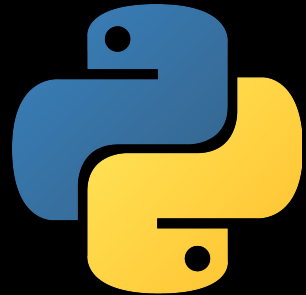


Why Python for data science?

Easy to learn, scalable language

Popular libraries:

- SciPy.org (Math, Science, Engineering)
- StatsModels (Statistics)
- Pandas (Frameworks)
- SciKit-Learn (Machine Learning)
- GGplot, Matplotlib, Plot.ly (Graphics)



Setting up your computer

(Brace yourself...)

We're going to set up...

- Anaconda
 - Software containing our Interactive Python Notebook (Jupyter) & Python 2.7 (3.5 also OK)
- Git (optional)
 - Working in the command prompt. You may already have this on Mac (Terminal)
 - Anaconda should work in Windows Command Prompt

1: Install Anaconda!

Download here (2.7.x):
continuum.io/downloads

Follow the instructions in
the website - they vary
per platform



Anaconda is an
open-source platform
for Python, powered by
Continuum Analytics.

2: Install Git (optional)

Git is a member project of Software Freedom Conservancy, which handles legal and financial needs for the project. Conservancy is currently raising funds to continue their mission. Consider [becoming a supporter](#)!

Go to this link to download

<https://git-scm.com/downloads>

(Most MacBooks already have Bash!)



git

--everything-is-local

About

Documentation

Blog

Downloads

GUI Clients

Logos

Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Mac OS X Windows

Linux Solaris

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients

Git comes with built-in GUI tools ([git-gui](#), [gitk](#)), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)



Pictures of Sloths Wearing Onesies

Setting up your computer can take time...



If you've done the following:

- ❑ Install Anaconda
- ❑ Install Git (optional)

You're ready to move on to the next step!

In this course you will learn

- ~~❏ Set up your computer~~
- ❏ Setting up iPython/Jupyter
- ❏ Basic Python commands
- ❏ Explorations in NumPy, Pandas, and Matplotlib
- ❏ Sandbox time!

Activating Jupyter

(Wait, I thought we were using Python...)

Project Jupyter: better coding xp



Began as IPython, short for “Interactive Python”


Way for you to code within a browser in an interactive way

Now offering support beyond Python...

Let's initialize Jupyter

1. Open up your Git terminal
2. Navigate to your working directory
3. Type “jupyter notebook” into the prompt
Some computation should happen...
4. Go to your browser and type in this URL
 - a. <http://localhost:8888/>

What you should see...



The screenshot shows the JupyterLab web interface. At the top, there are tabs for 'Files', 'Running', 'Clusters', and 'Conda'. Below these is a message: 'Select items to perform actions on them.' A list of files and folders is displayed, including 'anaconda', 'Applications', 'Chocolate-Chip-Cookies', 'dataviz-pj', 'Desktop', 'Documents', 'Downloads', 'ds-scratch', 'ds-scratch-jg', 'express-intro', 'fishbowl', 'flask-web-dev', 'h2o-3', 'hello-world', and 'heroku-dump'. In the top right corner, there are buttons for 'Upload', 'New', and a refresh icon. The 'New' button is circled in red, and a red arrow points to it from the text overlay.

The “New” button will come in handy next!

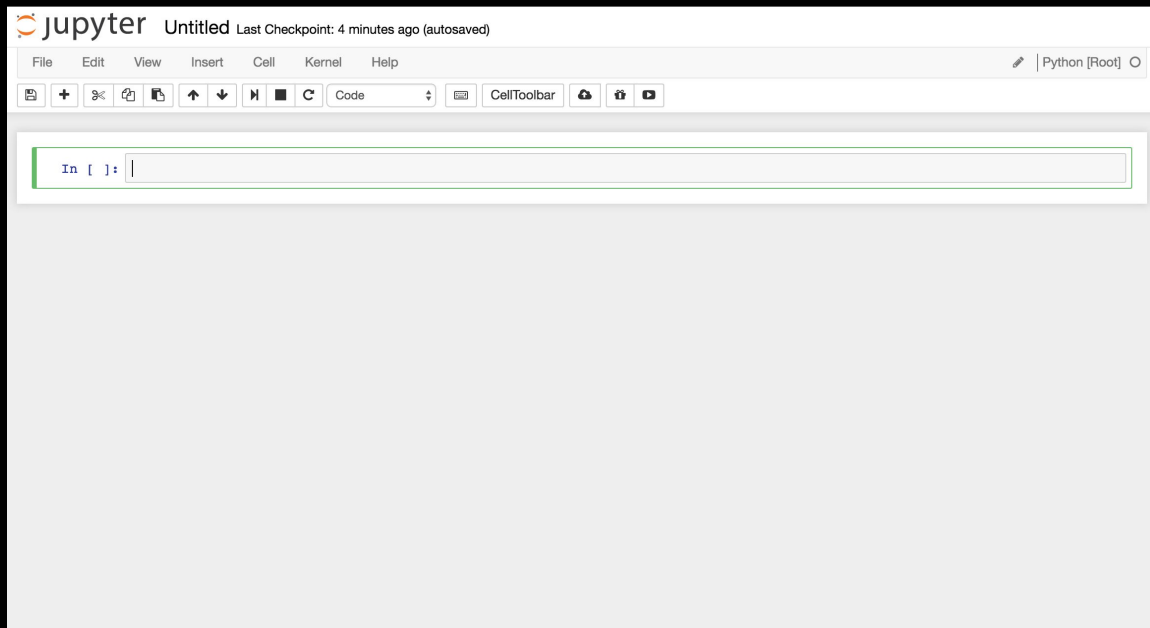
Create a new Jupyter Notebook

5. Click on “New” in the top right corner

6. Select under “Notebooks” > “Python
[root]”

Something should initialize immediately...

What you should see...



If you see this,
you are good
to go!

If not, raise
your hand!

If you've done the following:

- ❑ Initialize Jupyter on your computer
- ❑ Created your own blank Jupyter Notebook

You're ready to move on to the next step!

In this course you will learn

- ~~❏ Set up your computer~~
- ~~❏ Setting up iPython/Jupyter~~
- ❏ Basic Python commands
- ❏ Explorations in NumPy, Pandas, and Matplotlib
- ❏ Sandbox time!

Basic Python Commands

(Finally, let's code!)

Gut check, **Galvanize** style!



- This course is for beginners
- Feel free to move ahead
- Help others when you can
- Be patient and nice
- We'll all get through it!

Data Types - the very basics

Five data types:

- *int* - integer value
- *float* - decimal value
- *bool* - True/False
- *complex* - imaginary
- *NoneType* - null value

```
>>> type(None)
<type 'NoneType'>
>>> type(1)
<type 'int'>
>>> type(2.5)
<type 'float'>
>>> type(True)
<type 'bool'>
>>> type(2+3j)
<type 'complex'>
```

LET'S CODE!

Predict the data types of the following.

1. 3.1415
2. 10
3. 4+8j
4. False
5. ''

Check with the `type()` command in Jupyter.

(Hint: hit Shift+Enter to run in Jupyter.)

Data Types - “Arrays” in Python

Five types (iterable data):

- *str* - string/varchar immutable value
- *list* - collection of elements
- *tuple* - immutable list
- *dict* - unordered key-value pairs, keys are unique and immutable
- *set* - unordered collection of unique elements

Data Types - “Arrays” in Python

Five types (iterable data):

- *str* - defined with quotes = 'abc'
- *list* - defined with brackets = ['a', 'b']
- *tuple* - defined with parentheses = ('a', 'b')
- *dict* - defined with braces = {'a':1, 'b':2}
- *set* - defined with braces = {'a', 'b'}

LET'S CODE!

Create a list called **doc** with the following data in this order:

1. 'Gigawatts'
2. 88
3. 'miles per hour'
4. 1.21

doc is a variable here, so you have to set it equal to a list of these values

When you're done, type **doc[2]**. What happens?

Control Flow - If-else statements

```
x, y = False, False
if x:
    print('apple')
elif y:
    print('orange')
else:
    print('sandwich')
```

Try this in your notebook (ignore the stuff on the left as you type it) - what do you think will be printed out?

Control Flow - While loops

```
x = 0
while True:
    print('Hello!')
    x += 1
    if x >= 3:
        break
```

Try this in your notebook. What do you think will be printed out?

Control Flow - For loops

```
for k in range(4):  
    print(k**3)
```

Try this in your notebook. What do you think will be printed out?

Functions - Creating ways to use objects

```
def x_plus_4(x):  
    return x + 4
```

```
def subtract(x,y):  
    return x - y
```

Try creating your own mathematical functions like this.

Import - important for our purposes!

```
import math
```

```
math.pi
```

What is the output?

```
from math import sin
```

```
sin(math.pi/2)
```

What is the output?

Importing is crucial when using libraries and frameworks, such as....

If you've done the following:

- ❑ Reviewed the basic data types
- ❑ Reviewed control flows
- ❑ Reviewed functions and importing

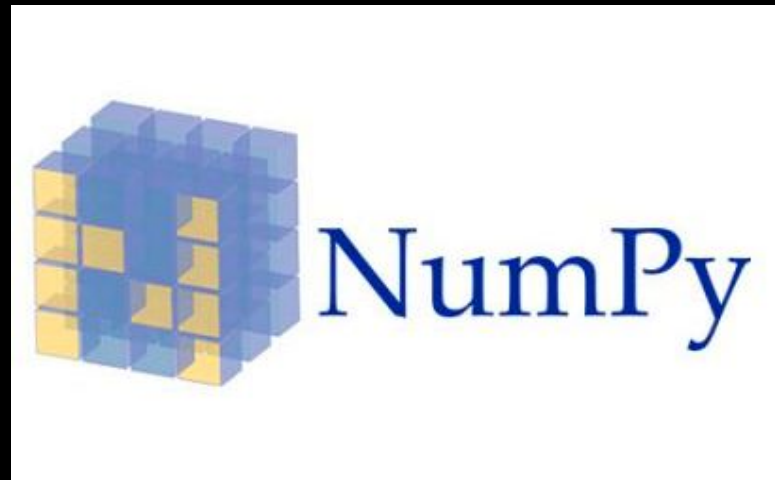
You're ready to move on to the next step!

In this course you will learn

- ~~❏ Set up your computer~~
- ~~❏ Setting up iPython/Jupyter~~
- ~~❏ Basic Python commands~~
- ❏ Explorations in NumPy, Pandas, and Matplotlib
- ❏ Sandbox time!

What's NumPy?

Python library of
mathematical functions
that allow us to
operate on huge arrays
and matrices of data



LET'S CODE! Array object in 1D

```
import numpy as np  
a = np.array([0,1,5,7,6])  
a[3]
```

What is the output?

LET'S CODE! Array object in 2D

```
a = np.array([[1,2,3],[4,5,6]])  
a.shape
```

What is the output?

LET'S CODE! Array object in 2D

```
a.T # transposing the array
```

```
a.T.shape
```

What is the output?

LET'S CODE! Array object in 2D

```
b = np.array([6,7])
```

```
np.dot(a.T,b) # matrix multiplication
```

What is the output?

LET'S CODE! Array object in nD

```
aa = np.array(  
    [[1,2,3],[4,5,6],[1,2,3],[4,5,6]]  
)  
  
bb = np.array(  
    [[[3],[4],[6]],[[6],[5],[7]]]  
)
```

LET'S CODE! Array object in nD

```
aa.shape, bb.shape # why do we check?
```

What is the output?

```
np.dot(aa, bb)
```

What is the output?

If you've done the following:

- ❑ Used NumPy on a 1-dimensional array
- ❑ Used NumPy on a 2-dimensional array
- ❑ Used NumPy on a n-dimensional array

You're ready to move on to the next step!

Exploration of Pandas

(Cuddly cute code?)

Pandas - yeah, science!



Open-source library providing powerful data structures and analysis tools

LET'S CODE! Create a DataFrame

```
import pandas as pd
import numpy as np
dd = {
    '0': pd.Series([1,2], index=['a', 'b']),
    '1': pd.Series([15,25,35], index=['a', 'b', 'c'])
}
pd.DataFrame(dd)
```

LET'S CODE! Create another one!

```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'int_col': [1,2,6,8,-1],
    'float_col': [0.1,0.2,0.2,10.1,None],
    'str_col': ['a','b',None,'c','a']})
df
```

LET'S CODE! Do some statistics!

```
df.describe() # basic stats
```

```
df.corr() # correlation
```

```
df.cov() # covariance
```

LET'S CODE! Do some visualization!

First, we need to import matplotlib to aid.

```
%matplotlib inline  
import matplotlib.pyplot as plt
```

LET'S CODE! Do some visualization!

```
plot_df = pd.DataFrame(  
    np.random.randn(100, 2), columns=['x', 'y']  
)  
  
plot_df['y'] += plot_df['x']  
plot_df.head()
```

What is the output?

LET'S CODE! Do some visualization!

```
plot_df.plot()
```

```
plot_df.plot('x', 'y', kind='scatter')
```






```
plot_df.plot(kind='hist', alpha=0.3)
```


If you've done the following:

- ❑ Build 2 dataframes with Pandas
- ❑ Do some statistics with Pandas
- ❑ Do some visualization with Matplotlib

You're ready to move on to the next step!

In this course you will learn

-  ~~Set up your computer~~
-  ~~Setting up iPython/Jupyter~~
-  ~~Basic Python commands~~
-  ~~Explorations in NumPy, Pandas, and Matplotlib~~
-  Sandbox time!

Play around in the sandbox! Try to...

Try one of the following

- Merging and joining your data frames
- Removing and replacing some missing values
- Renaming your data columns
- Download a dataset and conduct some analysis



You did it!

You are now a data scientist...ish.
Welcome to the cool kids club.

Learn more on your own!

Go to: [github.com/
GalvanizeOpenSource/
Python-Resources/](https://github.com/GalvanizeOpenSource/Python-Resources/)

1. Installation
2. Intro_Scripts
3. Basic_Resources
4. Exploring_Data



Learn Data Science with Galvanize



Data Science Fundamentals: Intro to Python

- 6 week part-time workshop

Data Science Immersive Program

- 12 week full-time program
- Job placement: 92% within 6 months (2016)

GalvanizeU

- 12 month program in San Francisco
- Fully-accredited by the University of New Haven

powered by  galvanize

Learn Data Science with Galvanize



To learn more, visit
galvanize.com/data-science

Or contact
enrollment@galvanize.com

powered by  galvanize

Thank you for coming to galvanize

Email Lee Ngo at
info@galvanize.com

or

Visit our website at
galvanize.com



This course has been brought to you by the evangelists of Galvanize.

Intro to Python

For Data Science

Learn to Code Contributors

powered by galvanize

Learn to Code Contributor

Matt Drury

github.com/madrury

Lead Instructor & Principal
Data Scientist @ Galvanize

Usually uses Spaceman
Spiff as an avatar



Learn to Code Contributor



Roger Schmidt

Sr. Web Developer and
Instructor at Galvanize
Still uses pen and
paper

Learn to Code Contributor



Lee Ngo

github.com/lee-ngo

Galvanize Evangelist
based in Seattle

Once did a Poisson
regression on
geolocation data

Learn to Code Contributor



Julia Tao

Data Science Immersive
Alum - Seattle

Was named after Julia
Roberts.

Saralyn Ogden

github.com/SaralynOgden

g34 Student - Seattle

Only has eight pairs of
socks.

UPDATED: Has more.



Learn to Code Contributor



David Valpey

Galvanize Data Science
Alum - Seattle

Spent a sizable portion
of childhood with no
electricity and no hot
water