

Python Training Workshop 2019

An introduction course to Python

Jan 31, 2019

Ryan Leung

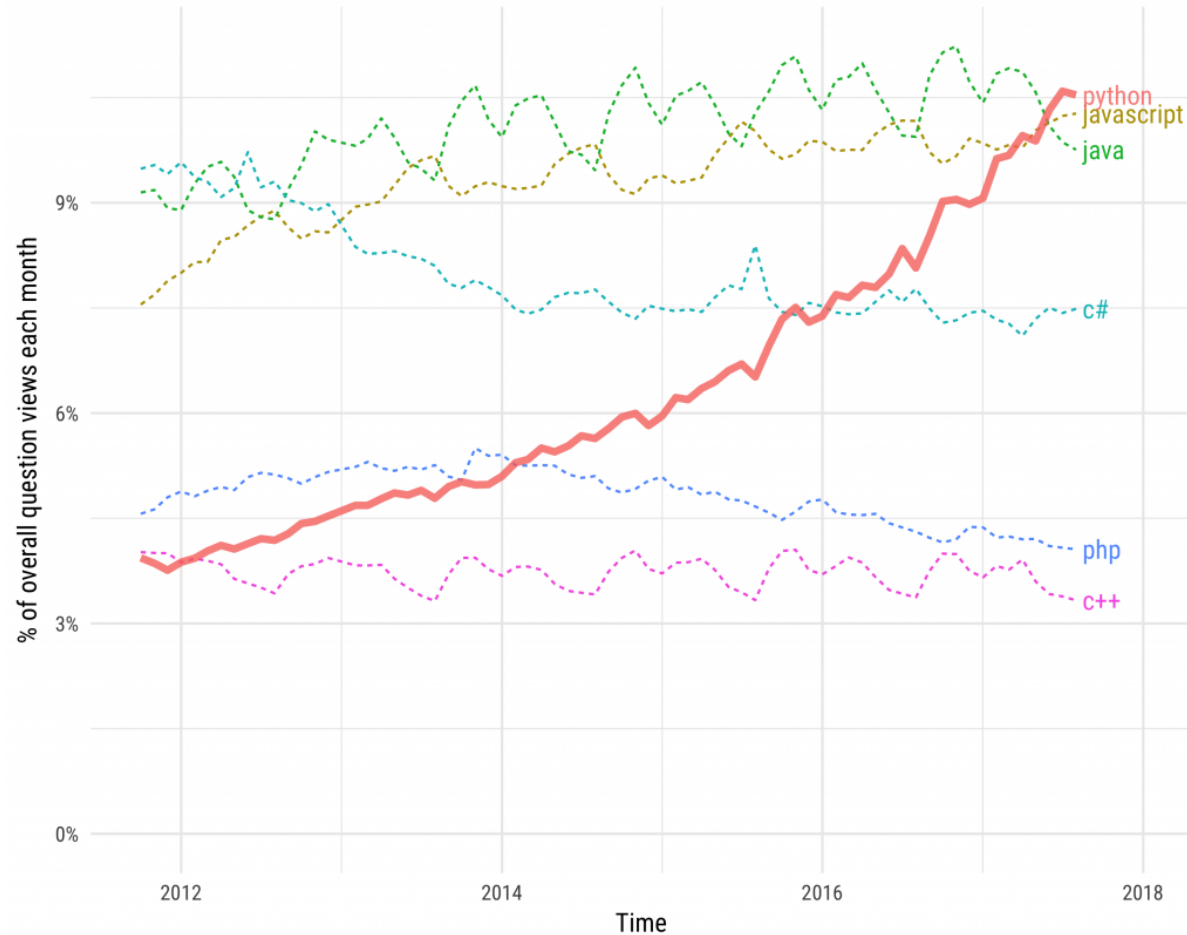
(yanyan.ryan.leung@gmail.com)

Please go to <http://goo.gl/> for the materials. :)

Python: a very fast-growing language

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



A new era of computing

- Varieties of programming languages
- Multi-core CPU and GPU support
- Easily-accessible cloud computing
- Cloud microservices

Python: a versatile language



- high-level
- object-oriented, and
- Interpreted

programming language.

Python: a "High-level language"

- "Low level language": C, Fortran, Basic
- **Level** means the accessibility to system resources.
- **High Level** :
 - care less about memory management or proper declaration of variables
 - less abstract than low-level language
 - less time to write and compile
 - relatively slower running time than some low-level language (not always true).

Community of Python users

- Web backend developers
- Data science
- Machine learning

Global Community

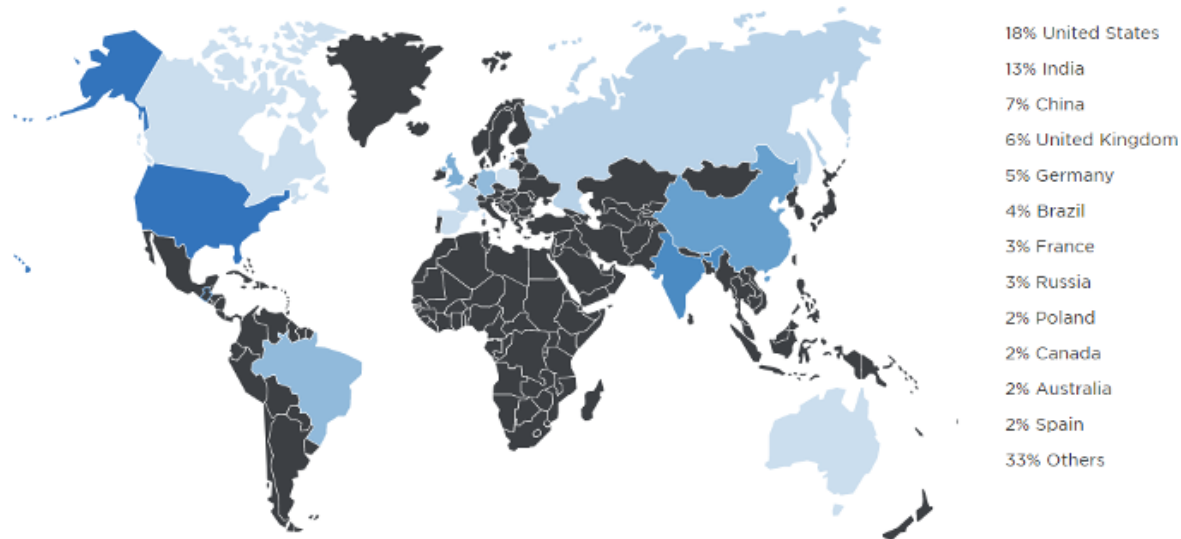


Image courtesy of the Python Developers Survey 2017 Results website

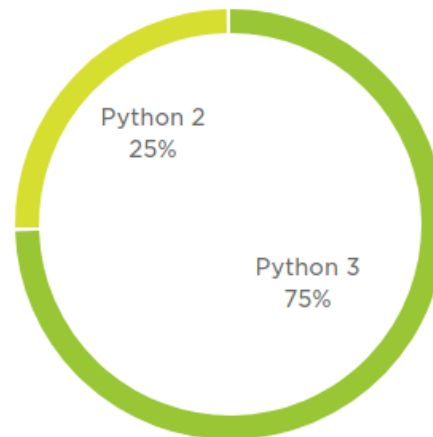
Python 2 vs Python 3

Results are quoted from

<https://www.jetbrains.com/research/devecosystem-2018/python/>

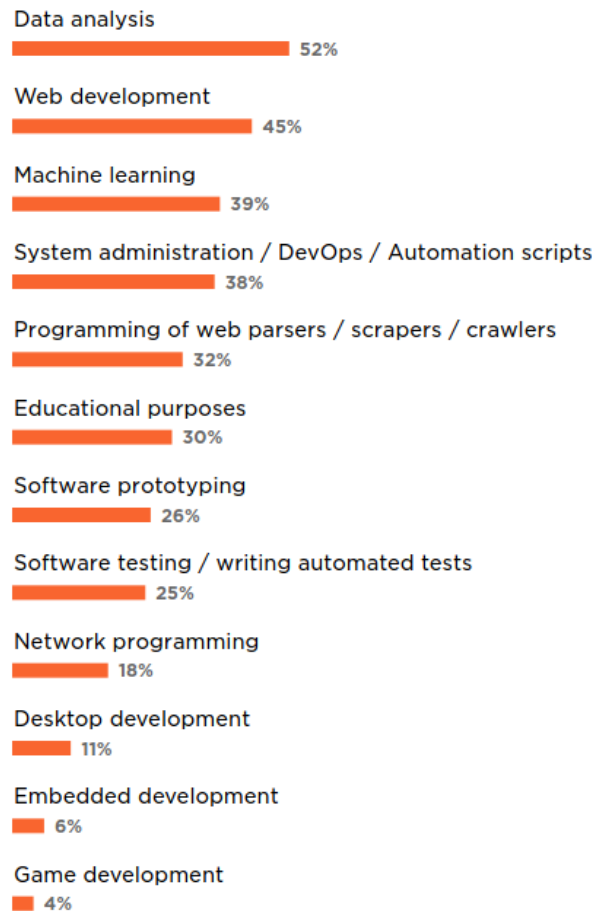


Which version of
Python do you use
the most?

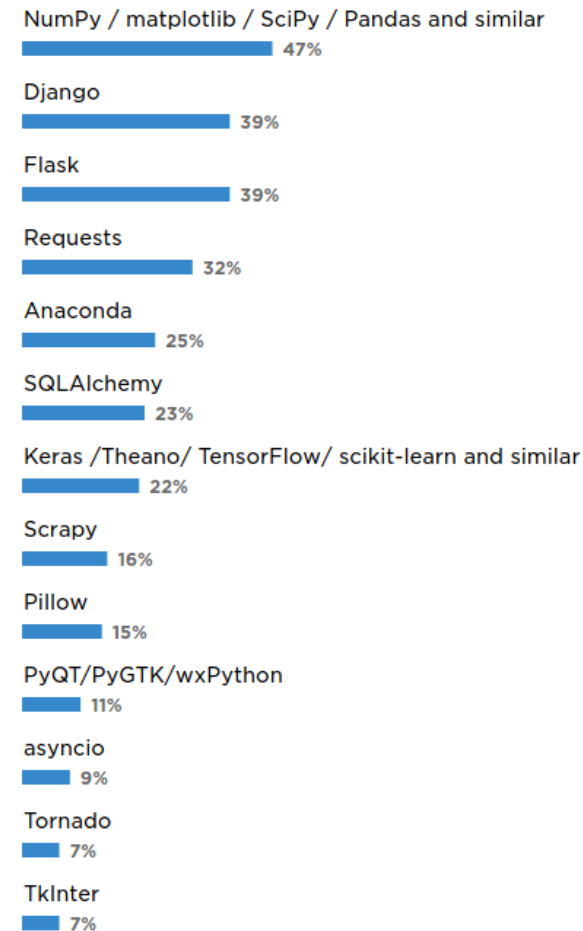


Python Usages

What do you use Python for?



What libraries and/or frameworks do you use in addition to Python, if any?



Installation

- Refer to another guide
- Recommendation:
 - Anaconda
 - Google colab

Install packages (with anaconda)

- `conda search xxxxxx`
- `conda install xxxxxx`

Install packages (with `pip`)

`pip` is a package management system in Python

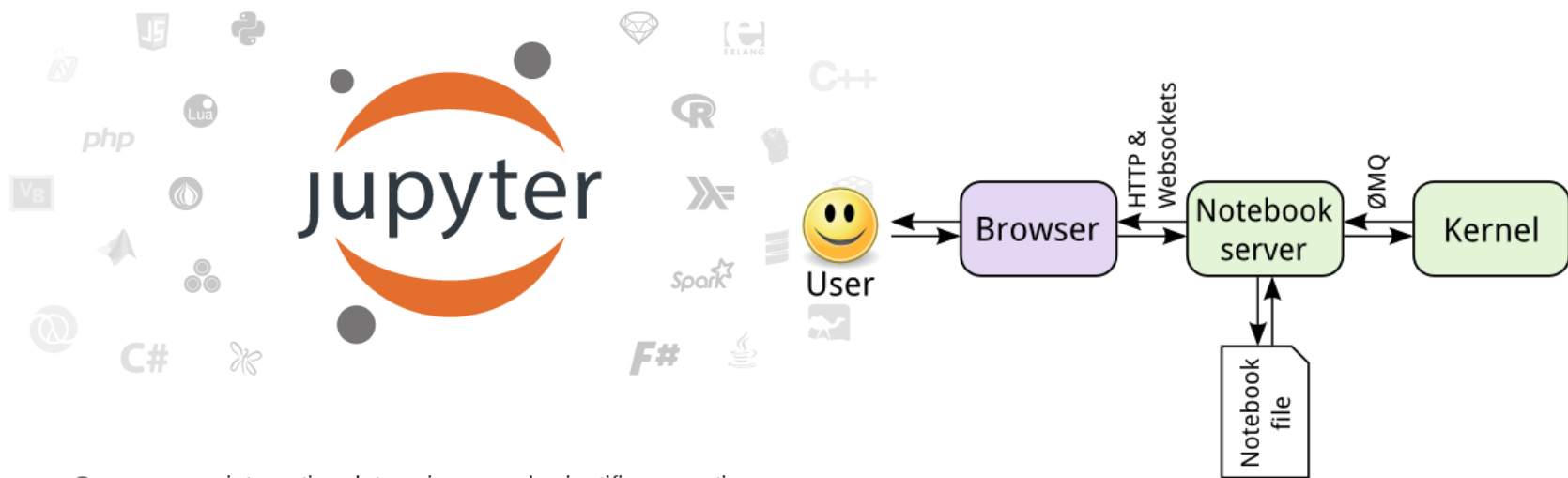
To search/install packages:

- Search package : `pip search xxxxxx`
- Install package : `pip install xxxxxx`
- Upgrade package : `pip install --upgrade xxxxxx`
- Uninstall package : `pip uninstall xxxxxx`
- Install wheel package : `pip install xxxxxx.whl`

Jupyter Notebook

You may want to run a Jupyter notebook when:

- You want to try out a new experiment or analysis with an existing Jupyter notebook from someone.
- You want to develop an algorithm that run on a large software.
- You have only ten minutes to download a data, plot a graph and send the email to your supervisor in a neat format.



Open source, interactive data science and scientific computing across over 40 programming languages.

Open Jupyter in Linux/MacOS

Type

```
jupyter notebook
```

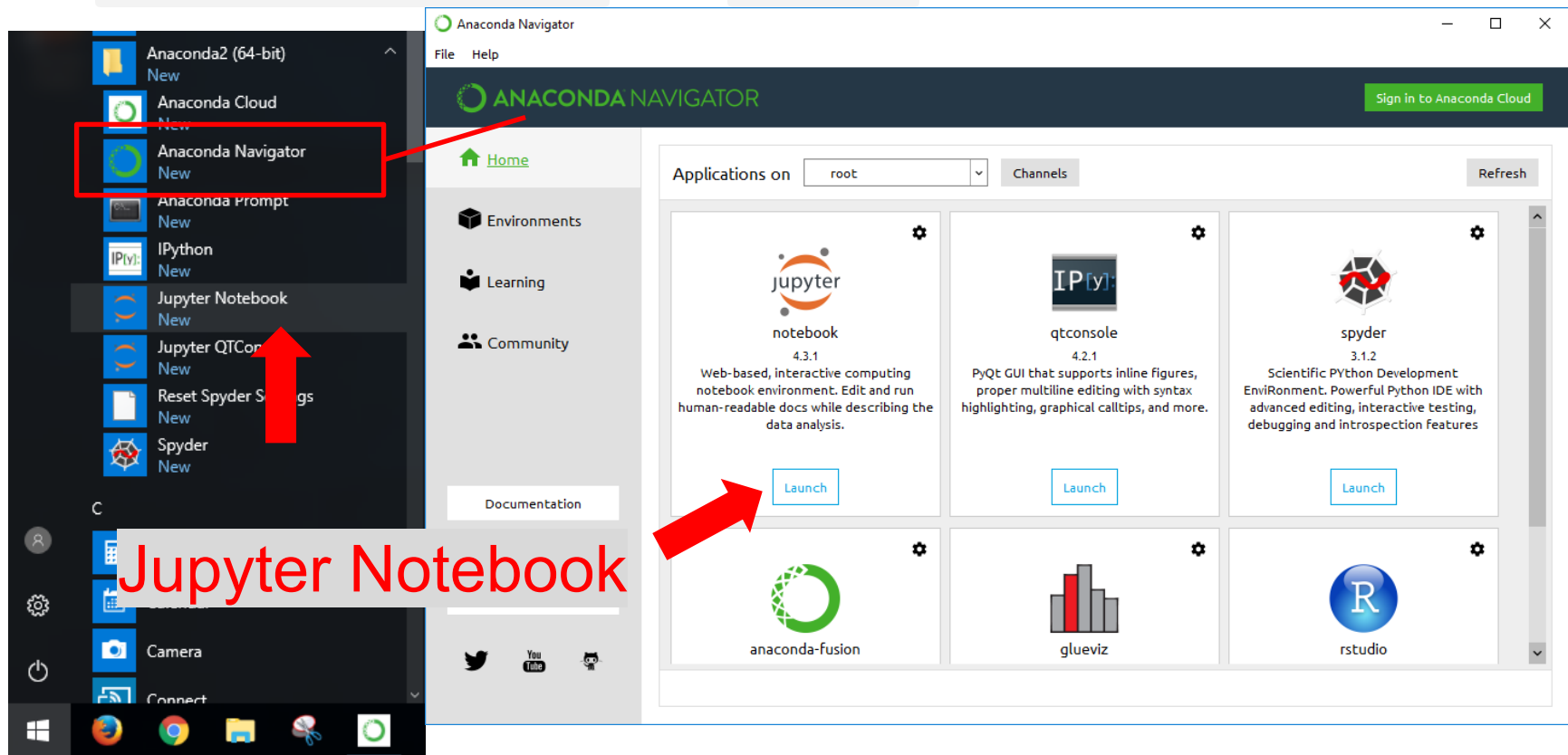
```
# yanyan @ vela in ~/workspace [12:37:07]
$ jupyter notebook
jupyter notebook

# yanyan @ vela in ~/workspace [12:37:07]
$ jupyter notebook
[I 12:38:14.082 NotebookApp] Serving notebooks from local directory: /home/yanyan/workspace
[I 12:38:14.082 NotebookApp] 0 active kernels
[I 12:38:14.082 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=40a1e1aa7783bb15e5178ec870a0f8bb07470e94d0a02da0
[I 12:38:14.082 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 12:38:14.083 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=40a1e1aa7783bb15e5178ec870a0f8bb07470e94d0a02da0
Gtk-Message: Failed to load module "pk-gtk-module"
Created new window in existing browser session.
[I 12:38:15.159 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Open Jupyter in Windows

Open your Start menu, goes to **Anaconda** Folder,
Click the **Jupyter Notebook** shortcut (Recommended). Or start
the **Anaconda Navigator** and **Launch**



Hand's on Session

The hand's on session requires a working python installations with Jupyter installed. The following links are read-only, they do not run calculations in your computer.

First Session:

- Python Syntax
- Python Data Structures
- Python Numpy Array


Second Session:



- Introduction to Pandas
- Python Functions and Class
- Python Matplotlib
- Python plotting with Astropy and AplPy

Online Platforms


Here are some online Python platform that are quite good indeed.


- c9.io
- repl.it
- [Microsoft Azure Notebooks](https://azure.microsoft.com/en-us/services/notebooks/)
- [Google Colab](https://colab.research.google.com/)






@YanYanLeung/PhonyProudConure 
 No description


my replscommunityBETAYanYanLeu... ▾

 saved


share 


run 


 





```
1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n - 1)
6 print factorial(5)
```



input 

clear 

Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.8.2] on linux
120




repl.it

An Informal Introduction to Python

[The [source material](#) is from Python 3.5.1, but the contents of this tutorial should apply to almost any version of Python 3]

Many of the examples in this manual, even those entered at the interactive prompt, include comments. Comments in Python start with the hash character, `#`, and extend to the end of the physical line. A comment may appear at the start of a line or following whitespace or code, but not within a string literal. A hash character within a string literal is just a hash character. Since comments are to clarify code and are not interpreted by Python, they may be omitted when typing in examples.

Some examples:



```
In [1]: # This is the first comment
spam = 1 # and this is the second comment
        # ... and now a third!
text = "# This is not a comment because it's inside quotes."
```

Using Python as a Calculator

Let's try some simple Python commands.

Numbers

Azure Notebook

Hello, Colaboratory


FileEditViewInsertRuntimeToolsHelp

CODETEXTCELLCELLCOPY TO DRIVECONNECTEDEDITING

Table of contentsCode snippetsFilesX

UPLOADREFRESH

..sample_data



Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. See our [FAQ](#) for more info.

Getting Started

- [Overview of Colaboratory](#)
- [Loading and saving data: Local files, Drive, Sheets, Google Cloud Storage](#)
- [Importing libraries and installing dependencies](#)
- [Using Google Cloud BigQuery](#)
- [Forms, Charts, Markdown, & Widgets](#)
- [TensorFlow with GPU](#)
- [TensorFlow with TPU](#)
- [Machine Learning Crash Course: Intro to Pandas & First Steps with TensorFlow](#)
- [Using Colab with GitHub](#)

Highlighted Features

Seedbank

Looking for Colab notebooks to learn from? Check out [Seedbank](#), a place to discover interactive machine learning examples.

Google Colab

Online Judge

- [Sphere Online Judge \(SPOJ\)](#)
- [HackerRank](#)
- [CodeAcademy](#)
- [Aizu Online Judge \(AOJ\)](#)

Good Reference Material

- stackoverflow.com



- <https://github.com/jakevdp/PythonDataScienceHandbook>



- [Book: Python for Astronomers](#)

Credits

This tutorial have referenced the following materials:

- [Unidata's online-python-training](#)
- [Anaconda Installation Guide](#)
- And thanks Sandy Chan and Stephen Ng