INTRODUCTION TO PYTHON

Dr. Jay Wang

July 2020

$ whoami

wcj365.github.io

Software Engineer
Java
Python
Cloud Computing

Project Manager
People
Process
Technology

Systems Thinker
Systems
Systems of Systems
Complex Adaptive Systems

Data Scientist
Data Engineering
Data Visualization
Data for Social Good

Why Learn?

To live is to learn.

Why Code?

Coding is fun.

Why Python?

“Life is short (You need Python)”

- Bruce Eckel, author of Thinking in C++

But I am a Systems Engineer, Why Me?

name a system that is not:

software-centric and data-driven

The Plan

Audience
Interested in Python

Prior Exp. not Required

Local Env. not Required

Objectives
Code in Python

Use Linux/Bash Shell

Use Git/GitHub

Approach
Hands-on Practices

Eng. Best Practices

Cloud-hosted Env

Three One-Hour Sessions, Monday 8 – 9 AM

Preparation

python.org

Read about Python

Try Interactive Shell

GitHub

Create an account

Explore features

Google Colab

Create an account

Explore features

Three Sessions

1. Elementary (Interactive Shell)
2. Exploratory (Jupyter Notebook)
3. Engineering (Python Modules)

Session #1 – Elementary Python via Interactive Shell

Scalar Types & Operators	int, float, bool, None =, +, - , *, / , %, ** (arithmetic)
Collection Types & Operations	string and string operations (strip, split, join, replace, find) tuple, list, dictionary, set, range(start, stop, step) In (relational, membership), comprehension
Conditions, Loops & Controls	Relational: ==, !=. >, <, >=, <=, and, or, not, is, in If, elif, else, for loop, while loop, pass/break/continue
Built-in functions & standard libraries	len(), type(), int(), str(), min(), max(), round(), range(), os.listdir(), sys.path, math.sqrt(), math.pi random - randint(), choice(), randrange(), gauss()

Getting Help in Interactive Shell

```
>>> help(<module or function>)
```

Examples:

```
>>> help(len)
>>> import math
>>> help(math.sqrt)
```

Python Summary

- Python is interpreted, not compiled
- Python is dynamically typed, not statically/ typed
- Python supports procedural, functional, and OOP
- Python’s flexibility is both a blessing and a curse.
    - More friendly to learning and exploration
    - Less rigorous for software engineering.
- Python is case-sensitive
- Python index is zero-based
- Python does not use curly brackets ({})
    - Use colon “:” along with indentation
    - Indentation can have any number of spaces
    - 4 space indentation is the industry standard
- Python interactive shell has limited functionality

Refresh and Repeat

String

+ (concatenation)

strip()

split()

Join()

replace()

lower()

upper()

Substring via index and slicing

Misc.

int(), str()

Assignment = vs Comparison ==

Relational: And, or, not, is, in

Input/print

Import <module>

from <module> import

list comprehension

try/catch/throw

break/continue/pass

Function/lambda

Session #2 – Explore Python via Jupyter Notebook

Cloud Env

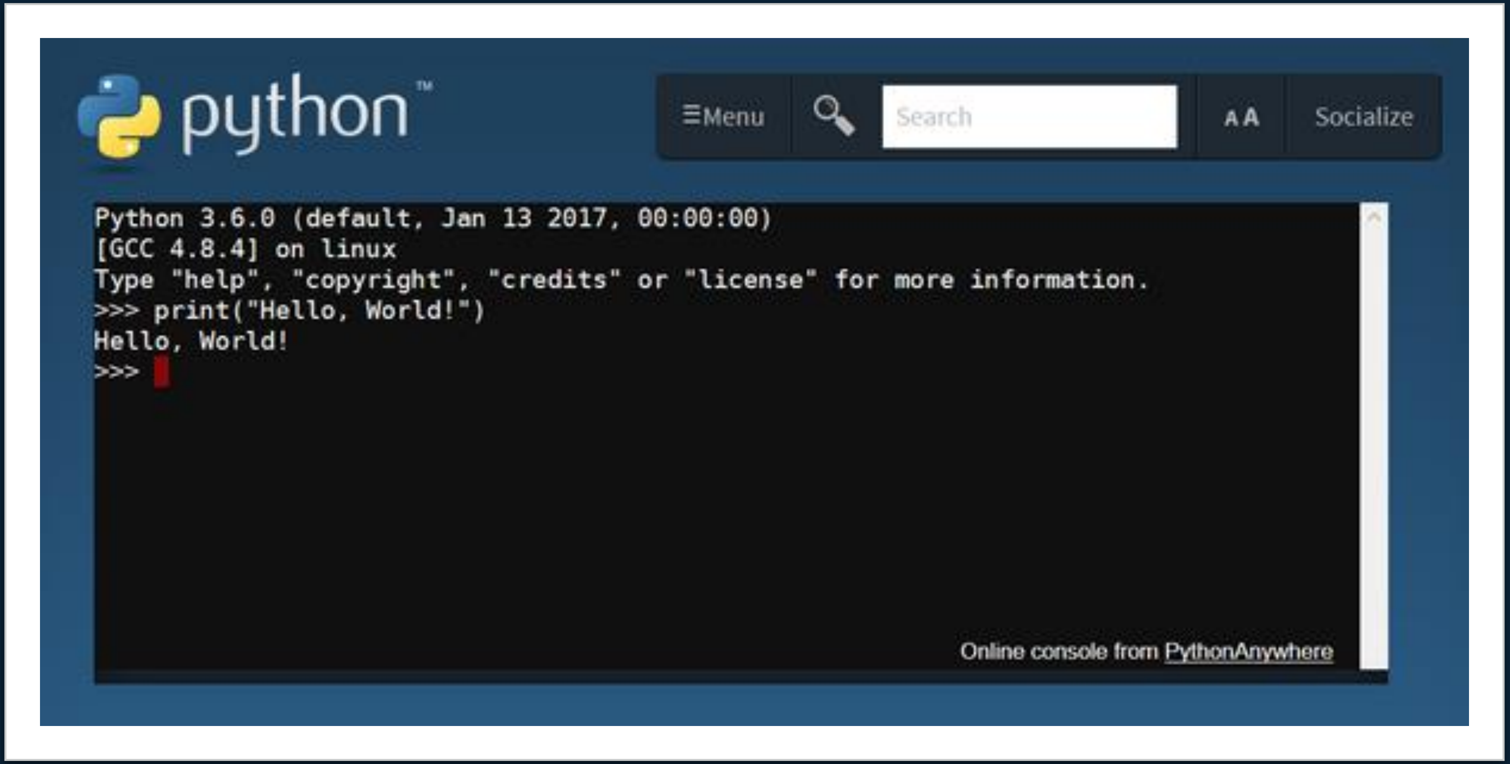
- GitHub
- Google Colab
- Kaggle

Project Mgmt.

- Repositories
- Project Structure
- Markdown

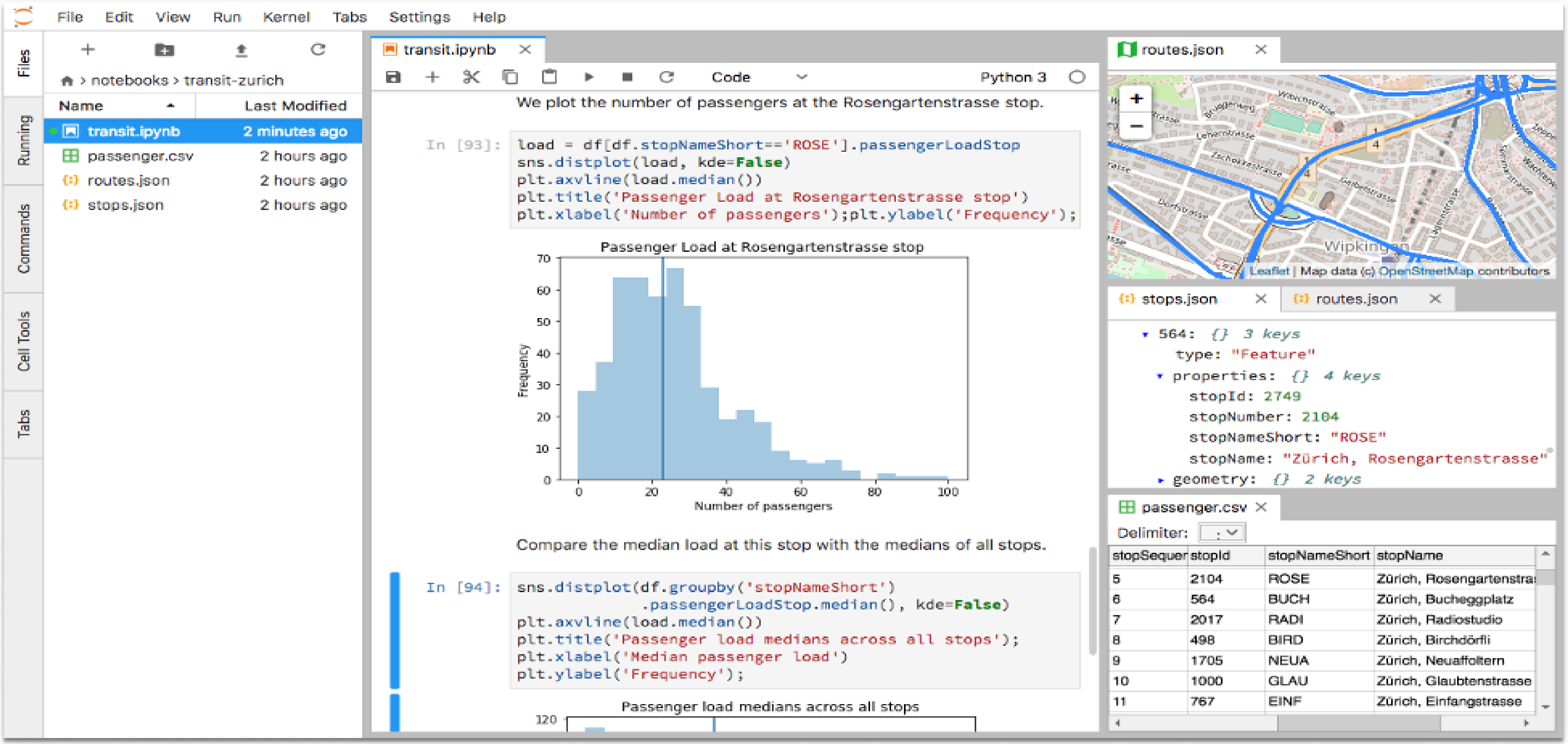
More Python

- try/except/throw
- Input/output/format
- functions/lambdas



```
Python 3.6.0 (default, Jan 13 2017, 00:00:00)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>> 
```

Interactive shell is helpful but limiting.


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
notebooks > transit-zurich
transit.ipynb
Code
Python 3
routes.json
Passenger Load at Rosengartenstrasse stop
Frequency
Number of passengers
Passenger load medians across all stops
Wipkingen
Leaflet | Map data (c) OpenStreetMap contributors
stops.json
routes.json
passenger.csv
Delimiter:

Name	Last Modified
transit.ipynb	2 minutes ago
passenger.csv	2 hours ago
routes.json	2 hours ago
stops.json	2 hours ago

We plot the number of passengers at the Rosengartenstrasse stop.

```
In [93]: load = df[df.stopNameShort=='ROSE'].passengerLoadStop
         sns.distplot(load, kde=False)
         plt.axvline(load.median())
         plt.title('Passenger Load at Rosengartenstrasse stop')
         plt.xlabel('Number of passengers');plt.ylabel('Frequency');
```

Compare the median load at this stop with the medians of all stops.

```
In [94]: sns.distplot(df.groupby('stopNameShort')
                      .passengerLoadStop.median(), kde=False)
         plt.axvline(load.median())
         plt.title('Passenger load medians across all stops');
         plt.xlabel('Median passenger load')
         plt.ylabel('Frequency');
```

```
▾ 564: {} 3 keys
    type: "Feature"
  ▾ properties: {} 4 keys
      stopId: 2749
      stopNumber: 2104
      stopNameShort: "ROSE"
      stopName: "Zürich, Rosengartenstrasse"
  ▸ geometry: {} 2 keys
```

stopSequer	stopId	stopNameShort	stopName
5	2104	ROSE	Zürich, Rosengartenstra
6	564	BUCH	Zürich, Bucheggplatz
7	2017	RADI	Zürich, Radiostudio
8	498	BIRD	Zürich, Birchdörfli
9	1705	NEUA	Zürich, Neuaffoltern
10	1000	GLAU	Zürich, Glaubtenstrasse
11	767	EINF	Zürich, Einfangstrasse

Welcome to the world of Jupyter Notebook

Getting Help in Jupyter Notebook

<module or function>? Click “Run”

<module or function>press shift + tab

Session #3 – Python Programs and Software Engineering

Command Line

Linux/Bash/Git

vi/ssh/scp

pythonanywhere.com

Coding Style

The Zen of Python

PEP 8

PEP 257

Python Modules

Scripts & Modules

Import Modules

Install Packages

The view through a window is nice, but I would rather get my feet wet right now!

Welcome to the world of Linux!

Common Commands of Linux Shell

```
$ whoami
$ python --version
$ cat /proc/cpuinfo
$ cat /proc/meminfo
$ cal
$ date
```

```
$ pwd
$ cd
$ ls –al
$ cp, rm,  rmdir, rm -r
$ ssh (secure shell)
$ scp (secure copy)
```

Basic Git Operations

One Time

To start with a remote git repo

- $ git clone <URL of a remote repository>

To start with a local dev folder

- $ git init

Recurring

Track changes in staging area

- $ git add .

Commit changes to local repo

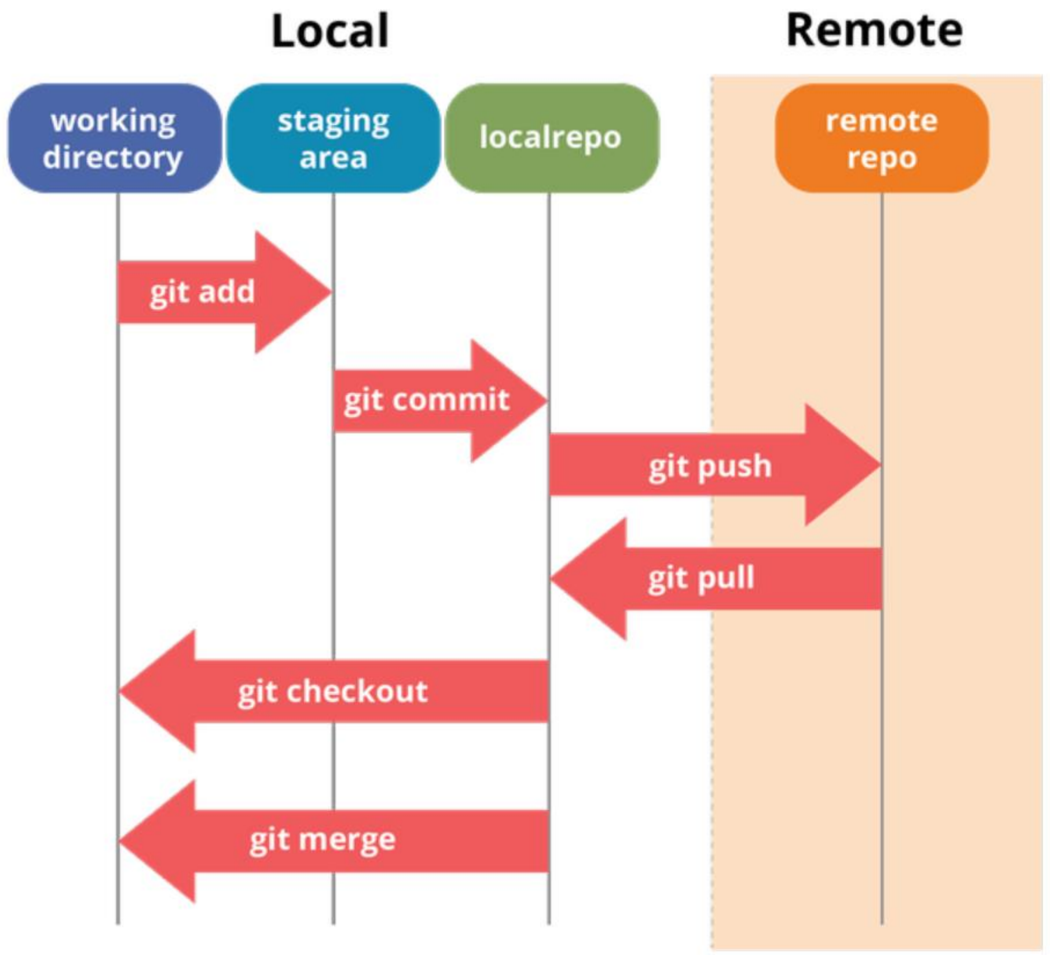
- $ git commit –m “blah blah…”

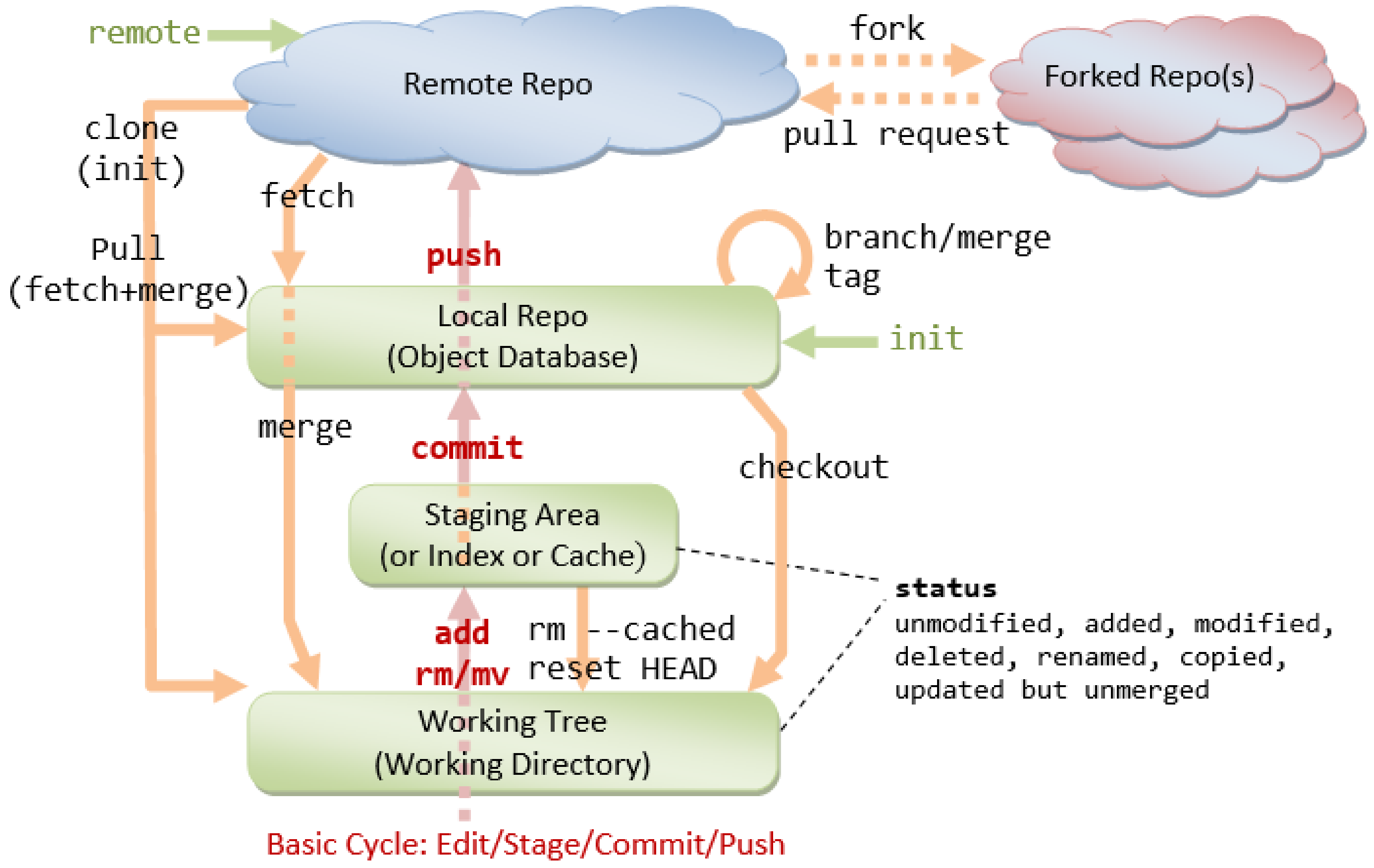
Push changes to remote repo

- $ git push –u origin master

Pull changes from remote repo

- $ git pull


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge



remote
Remote Repo
fork
pull request
Forked Repo(s)
clone
(init)
fetch
Pull
(fetch+merge)
push
branch/merge
tag
Local Repo
(Object Database)
init
merge
commit
checkout
Staging Area
(or Index or Cache)
status
unmodified, added, modified,
deleted, renamed, copied,
updated but unmerged
add
rm/mv
rm --cached
reset HEAD
Working Tree
(Working Directory)
Basic Cycle: Edit/Stage/Commit/Push

Practice #1 – Math Quizzes (Addition)

- Write a program that generates 10 random quizzes on addition of two numbers
- Numbers should be between 0 and 100
- Display quiz like 2 + 5 = ? and check the user input
  - If non integer is entered, prompt for re-entry
  - If answer is wrong, allow the user to try 2 more times
  - if answer is correct, move on to a new quiz
- At the end of the 10 quizzes, print a summary report
  - Display quizzes along with correct answers: 2 + 5 = 7
  - Display user answer only it is wrong: 2 + 5 = 7 (you answered: 8)
  - Display the number of correct answers: You fared 7 out of 10

Practice #2 – Math Quiz (Subtraction)

- Modify the program in practice #1 for subtraction
- Make sure the quizzes do not result in negative answers
- Not valid
  - 2 – 7 = ?
- Valid
  - 7 – 2 = ?

Practice #3 – Math Quiz (Multiplication)

- Modify the program in practice #1 for multiplication

Practice #4 – Math Quiz (Division)

- Modify the program in practice #2 for division
- Make sure the quizzes do not result in fractional answers
- Not valid
    - 7 / 2 = ?
- Valid
    - 8 / 2 = ?

Practice #5 – Math Quiz (Modular)

- Modify the program in practice #2 for modular operation
- Make sure the first operand is not less than the second operand
- Not valid
  - 10 % 15 = ?
- Valid
  - 15 % 10 = ?

Practice #6 – Math Quiz (Squared Root)

- Modify the program in practice #2 for squared root
- Make sure the quizzes do not result in fractional answers
- Not valid
  - Squared root of 10 = ?
- Valid
  - Squared root of 9 = ?

Practice #7 – Math Quiz (All-in-one)

- Consolidate all six operations (+, -, *, /, %, sqrt) into one product
- For each quiz, randomly pick an operation from the six choices

Challenge

- Scrape data from https://www.countries-ofthe-world.com
- Build a list of countries and their capitals for reference

Practice #8 – Raffle

Use Python to Implement the Following:

1. Create a list of workshop participant names
2. Assign each a random # between 1 and the list size
3. Randomly pick one # from the list, do this 1000 times
4. Calculate the frequency of each # being picked
5. Print the name whose # has the most occurrences
6. If there are ties, repeat the random drawing

https://made4dev.com/products/life-is-short-use-python-t-shirt-for-developers

Next Step - Python for Data Science

Pandas for Data Analytics

Plotly for Data Visualizations

Dash for Interactive Dashboards

PySpark for Big Data Analytics

Resources

Markdown

- https://guides.github.com/pdfs/markdown-cheatsheet-online.pdf
- https://commonmark.org/help/tutorial/

Bash/Git

- https://github.com/dlab-berkeley/BashGit
- https://about.gitlab.com/images/press/git-cheat-sheet.pdf
- http://feineigle.com/static/books/2018/git_essentials/Git-Essentials.pdf
- file:///C:/Users/cjwang/Downloads/essential_git_for_developers.pdf

Python

- http://python.org
- https://www.w3schools.com/python/default.asp
- https://www.practicepython.org/exercise/2014/01/29/01-character-input.html
- https://docs.python-guide.org/
- https://github.com/dlab-berkeley/python-fundamentals

Python for Data Science

- https://www.youtube.com/watch?v=r-uOLxNrNk8&feature=emb_title&fbclid=IwAR0qJPcqezRICXAx1TAR28Ca5hrw_H9HkMOzyUWPFhyer4G2Lxxs4YdwYIY

Summary of Fundamental Python

1. Data Types	2. Flow Controls	3. Inputs Outputs	4. Functions	5. Modules
Scalar: int, float, bool (True, False), None. =, +=, -=, +, -, *, /, %, **.	**Relational:** ==, != <, ,<=, >, >=, is, is not, in, is in, not in. **Logical:** and, or, not	**Console I/O:** print("hello", end=""), input("your name?")	**Built-in:** type(), len(), int(), str(), range(), min(), max(), round()	**Standard:** math, os, sys, random, datetime
Collection: str, range, tuple, list, dict, set. **Index & slicing:** x[index], x[start:stop]	**Loops:** for i in range(5), while x < 5, while True	**File I/O:** *with open(path, mode) as:* "t" - text, "b" - binary, "r" - read, "w" - write, "x" - create, "a" - append	**Named:** def func(args), return, pass	**Third-party:** numpy, statistics, scipy, pandas
Conversion: int(), str(), list(), set()	**Conditions:** if, elif, else	**File Input:** read(), readline(), readlines(). **File Output:** write()	**Anonymous:** lambda args : expression	**Import:** import, from x import y
String operations: strip, split, replace, join, format, index, find, upper, lower	**Controls:** break, continue, return, pass	**Output formatting:** format()	**Arbitrary args:** *args, **kwargs	**Packages:** pypi.org, pip, conda, venv
List comprehension: [x for x in C if condition]	**Exceptions:** try, except, throw	**Web scraping:** requests, BeautifulSoup, selenium, pandas.read_html()	**FP:** func is object, func as argument, zero side-effect, map(), filter()	**OOP:** class/type, instance, abstract base class (ABC)

Free Cloud-based Environments for Data Science

Envvironment\Feature	Bash Shell	Markdown	SQL	Interactive Python Shell	Python Script/Module	Jupyter Notebook	Big Data (PySpark)	Web App	Polyglot
GitHub/GitLab		X							
python.org				X					
Google Colab		X			X	X	X		
pythonanywhere.com	X		X	X	X	X($)		X	
Kaggle.com		X				X			
databricks.com		X				X	X		
mode.com		X	X			X			
glitch.com		X		X				X	
repl.it		X	X	X	X				X