Bootcamp Syllabus

Description: The world has changed a lot in the last decade: Python is one of the most popular languages in the world, the cloud is ubiquitous, and data is hot. Another big change is everyone needs some type of automation. This course covers fiercely effective ways to “get stuff done” in Python. This will be taught using Colab, an interactive Jupyter environment, with hands-on code examples throughout the course. With software systems, for every failed perfect solution that never got implemented, there is a reasonable solution that could have improved things. Learn to embrace imperfect, but useful, automation and you will never look at a boring task the same way again.

Target Audience: You are a DevOps engineer and want to automate tasks more effectively. You perform IT and want to get into DevOps. You are a developer who wants to become better at DevOps

Technologies: Having a google account will be helpful to run the colab notebook. Alternatively you can follow along with a Jupyter Notebook instance you have access to. Additionally, it will be helpful to have free accounts for AWS, Azure and Google Cloud for sections that use those resources.

Prerequisites: Beginning Python skills (or proficiency in a scripting language and a basic understanding of Python syntax) Basic understanding of cloud computing concepts. Comfort working with Linux

Student References: Download or clone the code repository for the course: https://github.com/noahgift/python-devops-course. A free Google account will be needed to access notebooks in Google Colab. While a Google account is not required, it is strongly recommended to get the benefit of full participation with the exercises. (If you are unable to use Google Colab, a downloadable notebook will be available via a Github repo.)

Outline

Day 1

Introduction to Python for DevOps (60-120 minutes)

- Presentation: 5-10 Minutes
  - Poll: Experience Level With DevOps?
  - What is DevOps anyway?
- Presentation: 20 Minutes
  - Poll: Experience Level With Python?
  - Procedural Statements
  - Lists and Dictionaries
- Presentation: 20 Minutes
  - Poll: Experience Level with IPython?
  - Functions
  - Lazy Expressions
  - Async and Concurrency
  - IPython terminal
- Q&A: 5-10 Minutes

- Break: 5-10 Minutes

Automating Text and Filesystem (60-120 minutes)

- Presentation: 20 Minutes
  - Poll: Experience level with File I/O in Python?
  - Reading, Writing and Using files: .txt, csv and yaml.
  - Managing files and directories using os.path and pathlib
- Presentation: 20 Minutes
  - Poll: Experience level with searching a file system in Python?
  - Walking directory trees using os.walk
  - Getting stat information on files and directories
  - Finding files: duplicates, globbing, and patterns
- Q&A: 5-10 Minutes
- Break: 5-10 Minutes

Day 2

Developing with the Command Line (60-120 minutes)

- Presentation: 20 Minutes
  - Poll: Experience level with command-line?
  - Poll: Experience level with pip?
  - Setting up a Python project with VSCode and pip
  - Using sys.argv
  - Using click CLI Framework
- Presentation: 20 Minutes
  - Poll: Experience level with VSCode?
  - Using argparse [Only mention it exists]
  - subprocess.run
  - Mixing shell and Python together
- Q&A: 5-10 Minutes
- Break: 5-10 Minutes

Continuous Integration and Delivery (60-120 minutes)

- Presentation: 20 Minutes
  - Poll: Experience with Pylint and Pytest?
  - Creating Makefiles
  - Creating Python virtualenv
  - Linting and testing Python code w/: Pylint and PyTest
- Presentation: 20 Minutes
  - Poll: Experience with CircleCI?
  - Poll: Experience with Continuous Integration and Delivery?
  - Configuring CircleCI Continuous Integration with Github
  - What is Continuous Delivery?
- Q&A: 5-10 Minutes