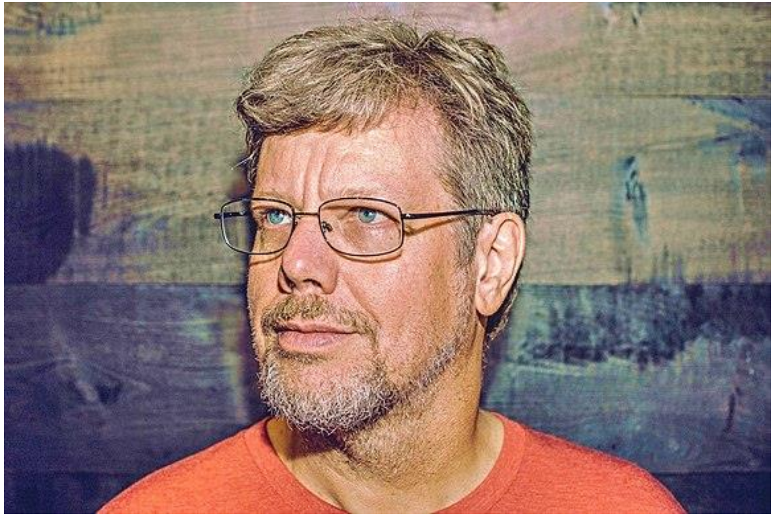
Python Overview
&
Development Setup

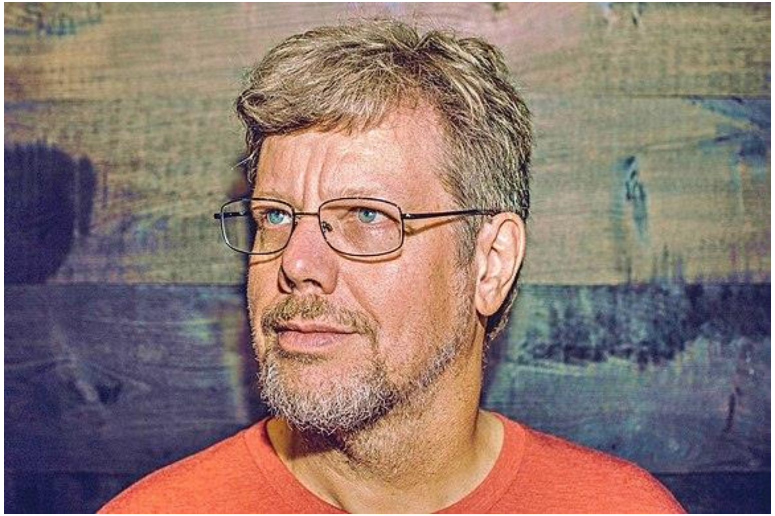
Lectures 1 and 2

- In this lecture we will do a brief overview of what Python is, why choose Python for programming, and what you can do with Python.
- This lecture in particular is geared towards people new to programming.

- Brief History of Python
  - Created in 1990 by Guido van Rossum
  - Python 3 released in 2008



- Brief History of Python
    - Specifically designed as an easy to use language
    - High focus on readability of code



- Why Choose Python?
    - Designed for clear, logical code that is easy to read and learn.
    - Lots of existing libraries and frameworks written in Python allowing users to apply Python to a wide variety of tasks.

- Why Choose Python?
    - Focuses on optimizing developer time, rather than a computer’s processing time.
    - Great documentation online:
        - **docs.python.org/3**

- What can you do with Python?
    - This course first focuses on “base” Python, which consists of the core components of the language and writing scripts and small programs.
    - Later we begin to learn about outside libraries and frameworks that greatly expand Python’s capabilities.

- What can you do with Python?
  - Automate simple tasks
    - Searching for files and editing them
    - Scraping information from a website
    - Reading and editing excel files
    - Work with PDFs
    - Automate emails and text messages
    - Fill out forms

- What can you do with Python?
  - Data Science and Machine Learning
    - Analyze large data files
    - Create visualizations
    - Perform machine learning tasks
    - Create and run predictive algorithms

- What can you do with Python?
  - Create websites
    - Use web frameworks such as Django and Flask to handle the backend of a website and user data
    - Create interactive dashboards for users

- Once you understand base Python and begin working with a few libraries, you’ll quickly begin to see the vast potential Python has for your own projects!
- Let’s get started with setting you up for the course!

Installing Python

1. Get it from the official website (www.python.org) – you'll get only the basic stuff

2. Get it from the Anaconda website –
   you'll get most of the things needed in single installation package

3. Use a "no-install" option – no need to INSTALL anything on your PC, instead, use a website to access and execute code (e.g., Google Colab) – you need a live internet connection while working

   Google Collaboratory
   URL: https://research.google.com/colaboratory/
   You need a Google account to create Python notebooks
   (think of them as Python source code files)

- There are many ways to run Python!
- Later on we’ll explore the difference between running a Python .py script or running Python code in a notebook environment.
    - ***Either way, we will still want to install Python!***

- Installation Lecture:
    - Install Anaconda Distribution for Python.
        - Anaconda installs Python **and** an easy to use development environment and navigator launch tool.
    - Briefly run Jupyter Notebook.
    - Explore “no install” online options.

- To install Python we will use the free Individual Anaconda distribution.
- This distribution includes Python as well as many other useful libraries, including Jupyter Notebook environment.
- Anaconda can also easily be installed on to any major OS, Windows, MacOS, or Linux.

www.anaconda.com/downloads

- **Free “No Install” Options:**
    - jupyter.org/try
    - Google Collab Online Notebooks
    - Repl.it
        - Google Search:
            - “Python Interpreter Online”

Running Python Code

- There are several ways to run Python code.
- First let’s discuss the various options for development environments
- There are 3 main types of environments:
    - Text Editors
    - Full IDEs
    - Notebook Environments

- Text Editors
  - General editors for any text file
  - Work with a variety of file types
  - Can be customized with plugins and add-ons
  - Keep in mind, most are not designed with only Python in mind.

Most popular:

VS Code, Notepad++, Sublime Text

- Full IDEs
    - Development Environments designed specifically for Python.
    - Larger programs.
    - Only community editions are free.
    - Designed specifically for Python, lots of extra functionality.

Most popular: PyCharm and Spyder

- Notebook Environments
  - Great for learning.
  - See input and output next to each other.
  - Support in-line markdown notes, visualizations, videos, and more.
  - Special file formats that are not .py

Most popular is Jupyter Notebook.

- Most important note:
  - Development Environments are a personal choice highly dependent on personal preference.

Choose whichever development environment you prefer!

- Let’s now explore how to run Python code:
  - First with an editor to create a .py script and run the file at your command line.
  - Then with a Jupyter Notebook.