PYTHON 2.X IS DEAD, LONG LIVE PYTHON 3.X

BEING A VERY-SIMPLEST INTRODUCTION TO THOSE BEAUTIFUL METHODS GENERALLY CALLED BY THE TERRIFYING NAMES PYTHON 3.X AND PYTHON 2.X

Some Housekeeping

- Next Week: ACM Presents: Git, version control
  Southwick 240
- Week After: TBA
  Southwick 240
- Sign In At signin.umlacm.org
- Join Us On Slack: slack.umlacm.org

Structure Of This Talk

- First 15-Min: Lecture
- Next 30-Min: Live Coding
- Final 15-Min: Questions

What This Talk Is Not

- An Introduction to Computing
- Applications Programming in Python

What This Talk Will Cover

- A brief description of Python as a Language
- Some Common Applications For Python
- A few ways to get started using Python, and a couple libraries
- And the current hot topic of 2.x vs 3.x
- References to sites where to learn more about Python

What Is Python?

- Interpreted
- High-level
- General-Purpose
- Dynamically Typed
- Multi-Paradigm

Interpreted

- A.K.A. Scripting Language
- Code Read and Processed at Run-Time
- No Compilation Step Before Processing
- Biggest example: JavaScript; also CSS to an extent, and technically HTML (though for other reasons it’s not a ‘programming’ language)

High Level

- A Loose Term
- Vaguely Implies it does not have direct access to hardware and memory
- Also has the implication that it can be quite slow

General Purpose

- This is fairly self explanatory
- Can Do Anything A ‘Programming Language’ Can
- Does NOT imply it’s great at everything

Dynamically Typed

- Variable Type Enforcement does not exist innately
- Casts are, mostly, implied

Multi-Paradigm

- There are many ways to approach Python Programming
- Functional Programming
- Symbolic Programming (LISP)
- Object Oriented Programming (Like C++,Java)
- Procedural Programming (Like C)

One Page History of Python

- 1.0 Released In 1991
- C is underlying language
- 2.0 Released In 2000
- 3.0 Released In 2008
- 2.x being sun-set at the end of 2019

Why is Python 2.x dying?

- Backwards compatibility is a pain to maintain
- Some of the C-code is very insecure
- Most major contributors to the project request 3.x features
- Many popular libraries dropped 2.x support
- Most new libraries are in 3.x

Where Does Python Get Used?

- Financial Industry
- Defense Industry
- Big Tech

What does Python Get Used For?

- Back End Server Management
- Front End Page Delivery
- Automation and Scripting
- Testing
- Mathematical Proofs
- Data Analytics and Science

Getting Started In Python

- Command-Line
- Jupyter Notebooks
- Anaconda
- Google Colabratory

courtneycaldwell — -bash — 80×24

```
Last login: Tue Sep 24 16:52:58 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Epictetus:~ courtneycaldwell$ brew install python
```

INSTALLING PYTHON - COMMAND LINE

ON LINUX: APT-GET INSTALL PYTHON

ANACONDA.

Products | Why Anaconda? | Solutions | Resources | Company | Download

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform Download

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

Windows | macOS | Linux

Anaconda 2019.07 for macOS Installer

Python 3.7 version

Download

64-Bit Graphical Installer (653 MB)
64-Bit Command Line Installer (435 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (634 MB)
64-Bit Command Line Installer (408 MB)

INSTALLING ANACONDA

GO TO ANACONDA.COM/DISTRIBUTION/

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Table of contents | Code snippets | Files

Welcome to Colaboratory!

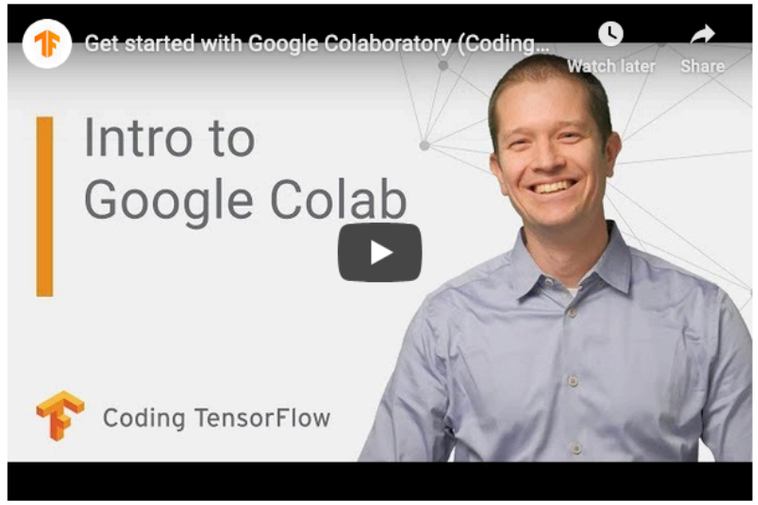
Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

[ ]

Introducing Colaboratory

This 3-minute video gives an overview of the key features of Colaboratory:



Getting Started

The document you are reading is a Jupyter notebook, hosted in Colaboratory. It is not a static page, but an interactive environment that lets you write and execute code in Python and other languages.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter".

All cells modify the same global state, so variables that you define by executing a cell can be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week
```

```
604800
```

GOOGLE'S COLABORATORY

COLAB.RESEARCH.GOOGLE.COM

Resources for Learning

- Udacity
- Codecademy
- Codewars
- Learn X in Y minutes
- HackerRank