Python and Machine learning: An Introduction

ISSAA 2022

Day #1

Vishal Upendran
IUCAA

Intended Takeaways

1. Open data files and visualize them.
2. Perform vectorized operations on this data.
3. ML-Astro: Have a basic understanding of techniques used.
   a. Be able to critique data preprocessing, choice of algorithm, etc. in research papers.
   b. Able to phrase the correct problem to use ML.
   c. Have a basic understanding of the effectiveness of different techniques when applied to a given task.

Menu for today

1. What *is* python?
2. Where do I code?
3. Some data types and their usage.
4. What are packages?
5. Simple examples?
6. What does astro data look like?
7. Exercise!

What is python?

Python is an *interpreted* high-level general-purpose programming language. Python's design philosophy emphasizes *code readability* with its notable use of *significant indentation*.

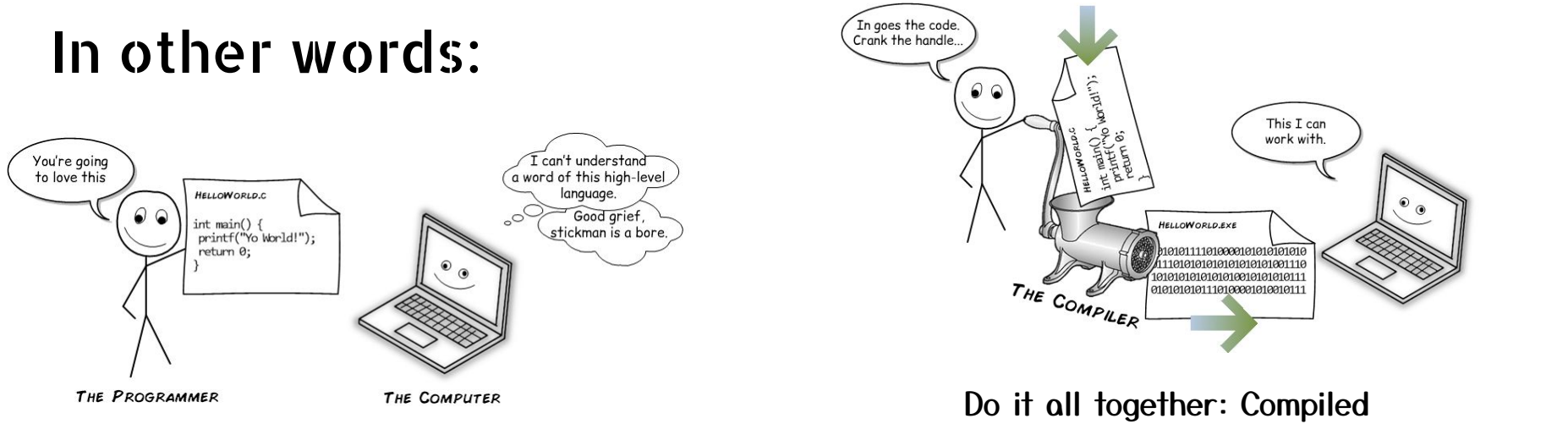- *Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1*

- Interpreted language.
- Code readability.
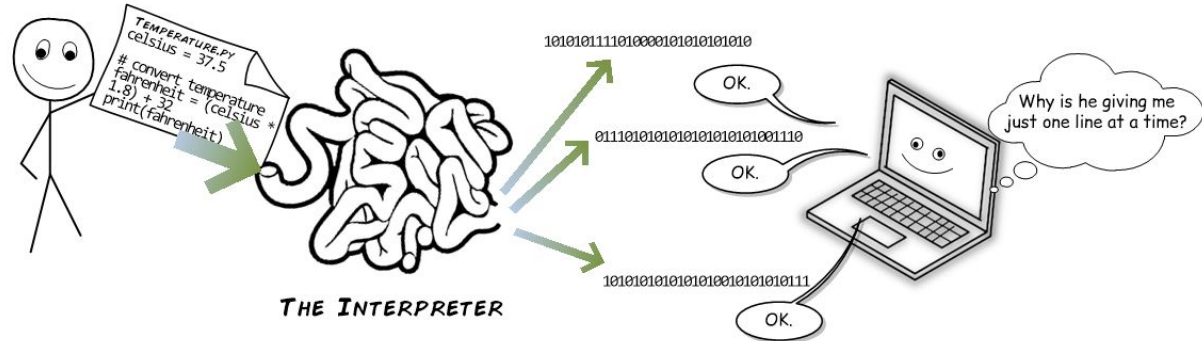- Indentation.

Why python?

- Open source.
- Support community.

Source: Young.Coder The Difference Between Compiled and Interpreted Languages

In other words:

You're going to love this

HelloWorld.c

```
int main() {
printf("Yo World!");
return 0;
}
```

THE PROGRAMMER

THE COMPUTER

I can't understand a word of this high-level language.

Good grief, stickman is a bore.

Your task: Show "Yo World" on monitor

In goes the code. Crank the handle...

This I can work with.

HelloWorld.c
int main() { printf("Yo World!"); return 0;

THE COMPILER

HelloWorld.exe

1010101111010000101010101010
1110101010101010101001110
1010101010101010101010111
01010101011101000010101010111

Do it all together: Compiled

Do it line by line: Interpreted

Temperature.py
celsius = 37.5

```
# convert temperature
fahrenheit = (celsius *
1.8) + 32
print(fahrenheit)
```

THE INTERPRETER

1010101111010000101010101010

OK.

0111010101010101010101001110

OK.

1010101010101010010101010111

OK.

Why is he giving me just one line at a time?

Isn't that bad?

- Line by line: easily understandable code.
- Execute blocks of code together.
  - Easy to debug.
  - Easy to prototype.
  - Easy to teach!
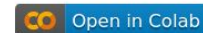- Interpreting on the fly: compatibility issues reduced.

An example?

```cpp
# include <iostream>
void main()
{
  cout<<"Hello world!";
}
```
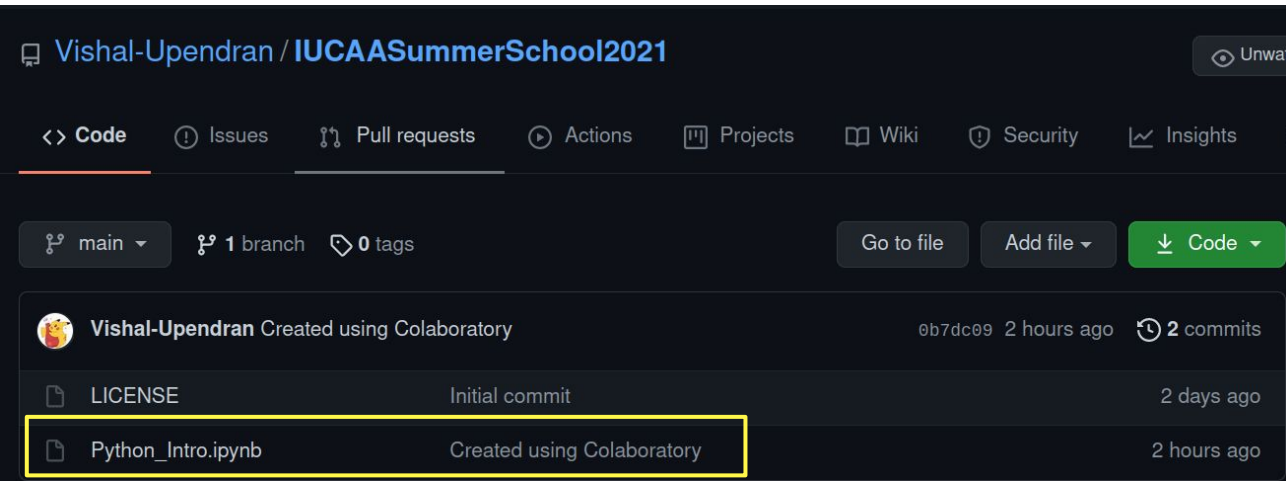
```python
print("Hello world!")
```

Where to write code?

Jupyter Notebook and Google Colab

Open in Colab

This section has:

1. Some basic intro to python, and OOP.
2. Data types - what is nan?
3. Lists, dictionaries, dataframes.

Vishal-Upendran / **IUCAASummerSchool2021**

Unwat...

<> Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights

main   1 branch   0 tags   Go to file   Add file ▾   ⬇ Code ▾

Vishal-Upendran Created using Colaboratory   0b7dc09 2 hours ago   ⏱ 2 commits

LICENSE	Initial commit	2 days ago
Python_Intro.ipynb	Created using Colaboratory	2 hours ago

Let us move on to Jupyter →

Some references and important packages

1. Jupyter: Today's codes were written in Jupyter notebook.
2. Numpy: For numerical computation, vector operations; bread & butter.
3. Matplotlib: Plotting and visualization library.
4. Scipy: Scientific-python - curve fitting, special functions, statistics, etc.
5. Pandas: Data loading and manipulation package. If you use GPU, check out Dask.
6. Skimage: Image processing in python.
7. Astropy: Core astronomy packages, coordinate systems, FITS generation. Very important for astronomers.
8. Sunpy: Associated packages for image astro manipulation, plotting, analysis

Always checkout the documentation of the packages if needed.
Stackoverflow is your best friend to venture into python.