INTRODUCTION TO COMPUTER PROGRAMMING IN PYTHON

Lesson 1
Astro Scholars 2022

What is Computer Programming?

Computer programming is a way to give instructions to a computer about what they should do next. These instructions are known as **code**, and computer programmers write code to solve problems or perform a task.

Ways you can use code:

1. To talk directly to your local computer, using the **command line** from a **terminal**
2. To write a **program** to perform a specific task
3. To develop **software**, a set of programs or **functions** that work together or independently to perform tasks
4. To develop websites through **web-development**
5. To **visualize** data and perform large scale mathematical operations on them
6. Virtually anything you can compute by hand, you can write a program for a computer to do more efficiently! You can even train computers to recognize trends and patterns that up until recently, only humans were intelligent enough to find, by using **machine learning** algorithms.

Code is a mix between art and science, it is technical and analytical yet creative at the same time.

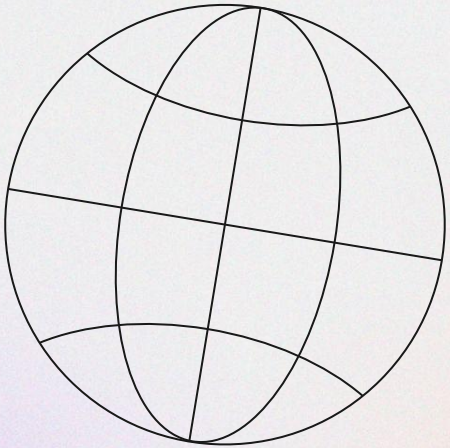
LANGUAGES

How do you “talk” to your computer?

COMMONLY USED LANGUAGES

Java

Python



C/C#/C++

Fortran

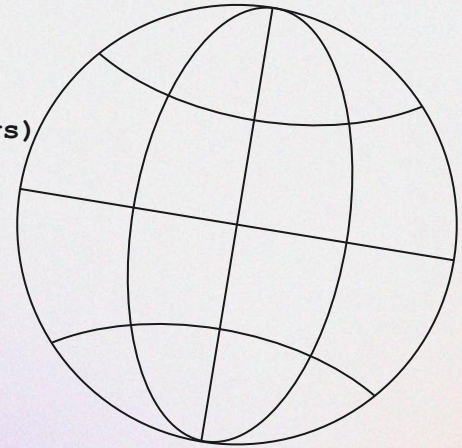
WHY PYTHON?

- Works on different platforms (Microsoft, Linux, Mac, Raspberry Pi, etc.)
- Simple syntax similar to the English Language
  - Syntax that allows developers to write programs with fewer lines than some other languages
- Code can be executed as soon as it is written
- Use of **modules** allows you to use pre-written functions to perform tasks ranging from basic calculations to complicated algorithms

COMMONLY USED LANGUAGES SYNTAX

Java

```
class HelloWorld{
 public static void main(String[] args)
{System.out.println(“Hello, World!”);
}
}
```



Python

```
print(‘Hello, World!’)
```

C

```
#include <stdio.h>
Int main(){
  printf(“Hello, World!”);
  return 0;
}
```

Fortran

```
program hello
print *, ‘Hello, World!’
end program hello
```

What kind of language is Python?

- Created by Guido van Rossum and released in 1991
- It is an **interpreted,** high-level, general purpose programming language
  - **Compiled** languages, like Java, C, and Fortran, take a program in a human-readable form (**source code**) and require a compiler to derive machine code (a form of the source code that a machine can directly execute) called an **executable**
  - Let's take a look at the power of an interpreted language!
- Emphasizes code readability with its use of **whitespace**
- It is a **dynamically** typed language
  - Instead of verifying the constraints of the code (**type checking**) at compilation, **static check**, it is verified as the code runs, **dynamic check**

```python
import numpy as numpy

def average_numbers(numbers):
    """
    This is how you write a long comment.
    Here is where you would write what is called a doc-string
    for a function. The doc-string, if well written, tells the user
    what the function does, what input it needs and what output it returns.
    For this example function:
    Input: numbers,list: a list of numbers you want to find the average of
    Ouput: average,float: the average value of your input list
    """

    #this is a short comment, these are used to leave notes within code
    #they are also used to comment out code that is not being used like below
    #print(numbers)

    sum = 0
    for i in range(len(numbers)):
        sum += numbers[i]

    manual_avg = sum/len(numbers)
    numpy_avg = np.mean(numbers)

    return manual_avg, numpy_avg

```

Example Code:

- White Space
- Indented Blocks
- Comments

Python Can Be Written in Different Environments

1. Installed on most **operating systems** by default
   a. If not, you can install Anaconda which installs Python and Python packages
2. Default Python **shell**
   a. Using the command: `python` in terminal
   b. Rarely used to develop code, mostly used to run code and check outputs
3. **Jupyter Notebook**

Let’s try running Python from the terminal!

EXERCISE 1: Running Scripts

Write a line of code that prints “Hello, World!”

Run it from the **command line** to see it print!

EXERCISE 1: Steps!

1. Open a terminal window
2. Initialize a python **session** by typing `python`
3. Write your code:
   a. `print("Hello, World!")`
4. Run your code!!
   a. Press `Enter`

JUPYTER NOTEBOOKS

- Allows you to store your code in **cells**, integrate it with **markdown** text (using **LaTeX**), and computational output (mathematical operations, plots, data visualization) in one document
- Powerful tool for developing code, interacting with data, and sharing results all in one document

How to Use:

- On a unix computer (like a Mac), use the command `jupyter notebook` from terminal
  - Or, open it from **Anaconda Navigator**
- On a Windows system, search for and start **Anaconda Jupyter notebook**
- This will open a web-based browser in the **root** directory
- To start a new notebook, click **"New"** in the top right corner, and choose **"`Python 3`"**
- To run a cell, press `Shift-Enter` at the same time
  - You can modify a cell and press `Shift-Enter` again to update the output

EXERCISE 2: Running Jupyter Notebooks

Download the Jupyter Notebooks from [LINK TO REPO]

Open the Day 1 Notebook on your computer

Practice running the cells

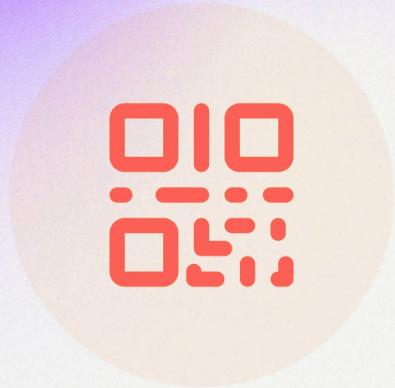
RESOURCES

Documentation on Python packages: http://www.python.org

Stack Overflow: http://www.stackoverflow.com

Using the built-in help tools in Jupyter Notebook: type `?` or `??` after any Python object to bring up the documentation or source code on that object

Exit Ticket



Join at slido.com
#587114

ⓘ Start presenting to display the joining instructions on this slide.

1. At what point(s) were you most engaged as a learner?

ⓘ Start presenting to display the poll results on this slide.

2. What concept from today's lecture would you like more elaboration or understanding about?

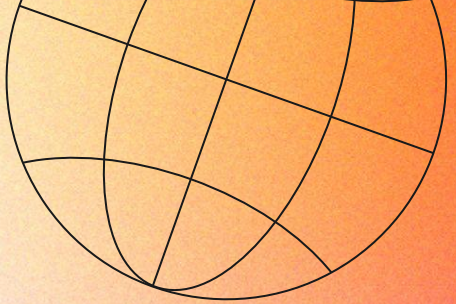
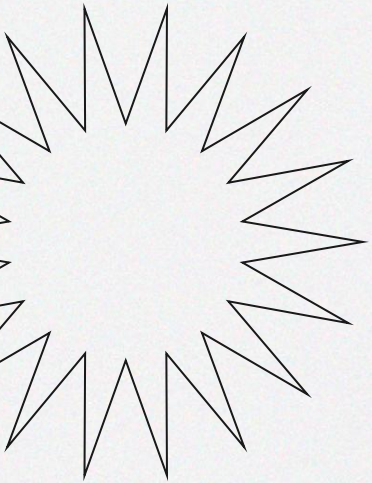
ⓘ Start presenting to display the poll results on this slide.

3. At what point(s) were you least engaged as a learner?

ⓘ Start presenting to display the poll results on this slide.

Questions?

Save to Dropbox!

Python Can Be Written in Different Environments

1. Installed on most **operating systems** by default
   a. If not, you can install Anaconda which installs Python and Python packages
   b. Python packages are commonly stored in PyPI (Python Package Index) from where you can download relevant packages using the command: `pip install [name of package]`
2. Default Python **shell**
   a. Using the command: `python` in terminal
   b. Rarely used to develop code, mostly used to run code and check outputs
3. Interactive Python **shell**
   a. Using the command: `ipython` in terminal
   b. Keeps the history of a **session**
   c. Commands can be recalled after a session is terminated and a new session is started
   d. Provides convenient tool for performance timing
   e. Useful for testing code
4. IDE (Integrated Development Environment) like Spyder, Visual Studio, or PyCharm
5. **Jupyter Notebook**

Tips to Navigate Terminal!

- Some useful terminal commands (written in **bash**) to navigate your computer:
  - To see all available files in a directory: `ls`
  - To navigate to a folder: `cd [name of directory]` or `cd [full path to directory]`
  - To navigate one directory backward: `cd ../`
    - To navigate two back: `cd ../../`
  - To see the path of the directory you are currently in: `pwd`
  - To make a directory: `mkdir [name of directory]`
  - To rename a file or to move it to a new location: `mv [name of old file] [name of new file` or `[full path of new file name in new location]`
  - To delete a file: `rm [name of file]`
    - To delete a directory: `rm -r [name of file]`
- Use a text editor to open an existing file or create a new file
  - Emacs: `emacs -nw [name of file]`
  - Nano: `nano [name of file]`
  - Set the name of the file so that it ends in `.py` -- this is how your computer knows what language it is written in
- To run a file from terminal: `python [name of file]`