Python Libraries? Let's Check it out...

Corey Fuller

Welcome to another ACM E-Python Workshop!

- I hope you all have been doing well in this E-Semester!!
- Even though we may not meet physically for the rest of the semester the Sessions will still continue!
- How this will happen
    a. The slides will be posted on Github along with an interactive worksheet
    b. The worksheet will have some sort of program at the end that will produce output, which you will paste into the attendance form for the extra credit
    c. There may or may not be a live kahoot depending on participation

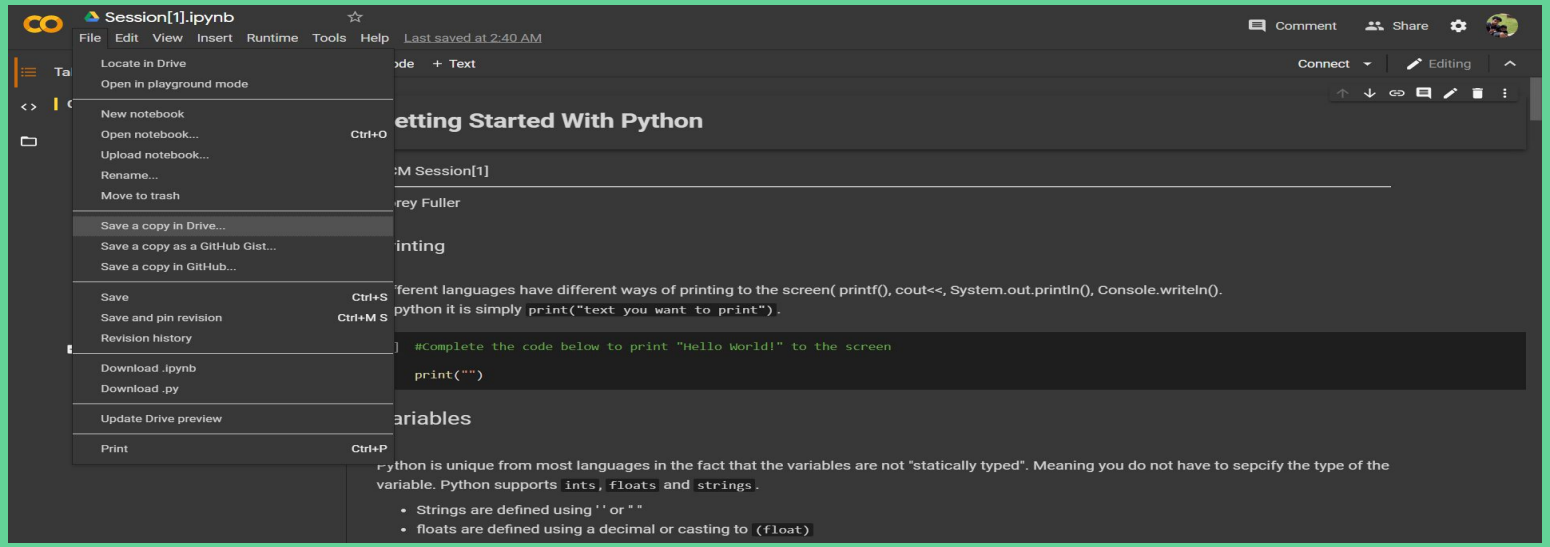
To Make Things Simpler...

- Go this link https://github.com/coreyFuller/ACM-Python.git
  - Click the Session3 Folder
    - Click **Session3.md**
      - Click "**Click here to access Session[3]'s interactive lesson file.**"
- All future slides will be posted on GitHub, uploaded to the discord, and sent via email.

https://github.com/coreyFuller/ACM-Python.git

Working with Google Colaboratory

- Google Colab allows us to make these sessions more interactive
- Once the notebook is open (you've clicked the link), save a copy of the notebook to your Drive
  - This will allow you to edit your own personal copy and follow along



Let’s Get Started!!!

A Review

- Most of you will recall last session we covered a few topics
  - Functions
  - Classes
  - Objects
  - Dictionaries
- The link to that session can be found here.
- The first thing on the worksheet is review practice from last session. If you're good, you can just skip by it.
- If you want some extra practice go ahead and try it out.

Imports

- This week we're going to talk about something that will help as we gear closer towards our final lecture: Libraries
- If you're at all familiar with programming you seen libraries before in many ways
- In C **#include <stdio.h>**, **#include <stdlib.h>**
- In C++ **#include <iostream>**, **#include <vector>, #include <string>**
- In Java **import java.util.***, **import java.lang.***
- In python, you use import statements almost similar to java to utilizes different modules, packages, and libraries

What are modules?

- Modules in python are a single python file that contains functions, classes, variables or all of the above.
- We actually made modules in the last sessions.
- We have some file, my_file.py for example that has some functions defined inside of it
- In another file we can import those functions by saying

  import name_of_file_we're_importing_from without the ".py" at the top of the file.

- And then in the code we can use the module functions with the **. operator** with the name of the import.
  - **name_of_file_we're_importing_from.function()**

Packages

- If we have a file with a long name like
- **longggggggggggggg_filename.py** for example. We can use the “**as**” keyword to make an alias so we don’t have to use **longggggggggggggg_filename.function()** everytime we want to use a function.
- **import longggggggggggggg_filename as lfn**
  - lfn.function1(), lfn.function2.()
- **A package is a collection of modules**
- If you’re familiar with Java packages in python work kind of similar. They are a grouping of related files, methods, and data
- Check out the example code in the Session3 folder on Github

Libraries

- Libraries in Python can be a collection of modules and/or packages
- We’re going to go a few basic, libraries and modules in python
- Python has a standard library like many other languages that contains many modules such as random, sys, math, and many many functions that you’ve already used

Random module

The random module allows you to generate random numbers as well as pick a random element from a list, etc. or when making your password database more secure, to name a few.

Import random (import the module)

Here are some useful functions

- random.Randint(0, 5) - accepts two parameters, a lowest and highest number and generates integers between that range.
- random.choice([‘red’, ‘black’, ‘green]) - chooses a random value from a list, etc.
- random.shuffle(list) - shuffles the elements in list in place, so they are in a random order.

Math module

The math module is essential for doing mathematical calculations in python

import math (importing the math library)

Here are some useful functions

- math.factorial(number) - only accepts positive integer values and returns the factorial of that number.
- math.pow(2, 5) - the first argument is the base value and the second argument is the power value. This returns the power of 2^5 which is 32.
- math.log(number) - can get the natural log (to the base e) of the number that is passed in.
- math.sqrt(number) - calculates the square root value of the number that is passed in

Sys module

This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Import sys(importing sys module)

Here are some basic sys methods and data

- Sys.copyright - A string containing the copyright pertaining to the Python interpreter
- Sys.path - A list of strings that specifies the search path for modules.
- Sys.byteorder - An indicator of the native byte order(big or little endian)
- sys.exit(1) - Exits from the code. The integer it takes in is the **status** upon exit

Numpy Library

Numpy is a library that's particularly useful for data science. It allows us to create arrays from lists perform some pretty cool calculations and view our data.

For this slide we'll be referring to numpy with the alias np so let's

Import numpy as np

Some useful functions

- np.array(list) - this function takes in a list and creates a numpy array
- type(name_of_np_array) - returns the type of the numpy array
- np.shape() - returns the shape of the array (rows, columns)

Numpy cont.

- The np.array function can take in multiple lists as arguments to give you a multidimensional array
  - np.array([a,b,c], [1,2,3], [x,y,z])
  - Printing the numpy array will structure it in a matrix of width x height
- You can index into this using pretty easily
  - my_npArray = np.array([a,b,c],[1,2,3],[x,y,z]
  - print(my_npArray[1,2]) , prints 2
- Numpy makes working with whole arrays pretty simple. For example I can print all the elements in my array that are larger than a certain value by doing
  - print(array(array > 2))
- Or return an array of booleans that correspond to values greater than a number in my array

Pandas (<- Yes that's the real name) Library

Pandas is a useful for analyzing data with python

For this slide we'll be referring to pandas with the alias pd so let's

Import pandas as pd

Here are some useful functions

- pd.read_csv(filename) - this function reads date from a csv, "csv" can be substituted with "table", "excel", "sql", "json", "html", and other sources
- df.to_csv(filename) - this function exports a DataFrame, a 2D tabular table structure with labeled axes (rows and columns). Csv can be substituted for the other sources above

Pandas Cont.

- df.head(n) - the first n rows of the DataFrame
- df.tail(n) - the last n rows of the DataFrame
- df.shape - the number of rows and columns
- df.describe() - summary statistics for numerical columns
- df.mean() - returns the mean of all the columns
- df.corr() - returns the correlation between columns in a DataFrame

Closing

- In this presentation we just covered some basic libraries, but there are thousands of 3rd party or open source libraries available to download
- Python is so large and used by so many people it would be impossible to cover all of the libraries available
- The good news is there are alot of documentation and examples online for using libraries so you hardly have to memorize anything

I encourage you to do some exploring in your free time.

Paste the output from the worksheet into the last question to receive extra credit