Intro to Data Management with Python

Course Description: “Data Management with Python, a ***Primer***” is for users who are brand new to using Python and want to get exposure to Python OR existing Python users who want to check out the coursework’s data management methods. This class will cover; introductory Python concepts/methods, basic module usage, debugging tips, Juypter notebooks, Pandas data fames, and more.

Date: June 25th, 2020

Contents

[Intro to Python – Course Section 1 3](#_Toc43920255)

[What & Why 3](#_Toc43920256)

[Creating a Python script from scratch & the Print Statement 4](#_Toc43920257)

[IDE 4](#_Toc43920258)

[python Versions 5](#_Toc43920259)

[Exercise 1: Hello World! 6](#_Toc43920260)

[If the “Edit with IDLE” option is not showing (e.g. Anaconda install) 7](#_Toc43920261)

[Debugging 9](#_Toc43920262)

[Exercise 1b: Hello World Debugging 10](#_Toc43920263)

[Python Topics for Data Management using IDLE – Course Section 2 12](#_Toc43920264)

[Commenting 12](#_Toc43920265)

[Indenting 13](#_Toc43920266)

[Variables 14](#_Toc43920267)

[More on String Variables 16](#_Toc43920268)

[Functions, methods, properties 17](#_Toc43920269)

[Python Classes 18](#_Toc43920270)

[Python Modules 18](#_Toc43920271)

[Python packages and Libraries 18](#_Toc43920272)

[Exercise 2A: Time, Time, tell me what time! 19](#_Toc43920273)

[For Loops, Iterating 20](#_Toc43920274)

[For Loops, Iterating through a list Examples 20](#_Toc43920275)

[Conditional Statements 21](#_Toc43920276)

[User-Defined functions 22](#_Toc43920277)

[Exercise 2B: Manage them files 23](#_Toc43920278)

[Exercise 2C: Extra Credit, Keeping it DRY 24](#_Toc43920279)

[Topics to explore on your own 26](#_Toc43920280)

Instructors/Teaching Assistants

Nathan Zimpher nathan\_zimpfer@fws.gov, MB,

Erin Butts erin\_butts@fws.gov, SA/Fisheries, Portland, OR

Brent Frakes brent\_frakes@fws.gov – Refuges I&M – Fort Collins, CO

Matt Heller matthew\_heller@fws.gov – SA - Bozeman, MT

Setup:

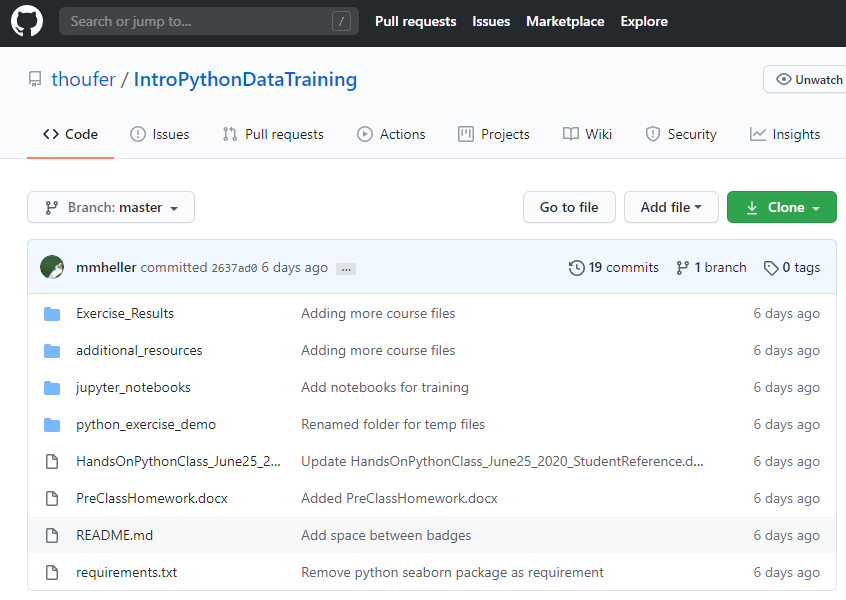
1. Create a C:\temp\_PythonEx2bEx2c\_files folder
2. Copy/download files from <https://github.com/thoufer/IntroPythonDataTraining/tree/master/python_exercise_demo>

to

C:\temp\_PythonEx2bEx2c\_files

Class Format and Materials

* Attendees, please use the chat box early and often.  Instructors will do their best to answer questions.
* The class will consist of instruction, 5 short hands on exercises and a final challenge
* Materials
  + Sections 1 and 2 attendee reference: HandsOnPythonClass\_June25\_2020\_StudentReference.docx
  + Exercise Results for Sections 1 and 2: Exercise\_Results/(various files)
  + Exercise Section 2 files: python\_exercise\_demo/(various files)
  + Jupyter notebook files
    - jupyter\_notebooks/PANDAS TRAINING SOLUTIONS.ipynb
    - jupyter\_notebooks/PANDAS TRAINING.ipynb
  + Powerpoint file: additional\_resources/HandsOnPythonClass\_June25\_2020.pptx
  + Additional Python resources: additional\_resources/Addtional Resources.docx
  + Pandas cheat sheet: additional\_resources/Pandas\_Cheat\_Sheet.pdf
  + Jupyter cheat sheet: additional\_resources/weidadeyue\_jupyter-notebook.bw.pdf



Class Outline

* Intro to Python (15 min)
  + Exercises 1A, 1B
* Python Topics for Data Management using IDLE (35 min)
  + Exercise 2A, 2B, 2D Extra Credit
* Python Topics for Data Management using Jupyter (40 min)
  + Final Challenge Exercise

# Intro to Python – Course Section 1

## What & Why

What is Python - python.org definition:

“Python is an easy to learn, powerful language... (with) high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing...make it an ideal language for scripting...in many areas and on most platforms.” *ESRI*

What is Python & Why use it

Free, cross-platform, easy to learn,

Widely useful, great community ESRI

Personal Reason Why…

Interoperable

USFWS and DOI allow using it

Allows me (shareable to ) to do powerful processing

A ton of help resources

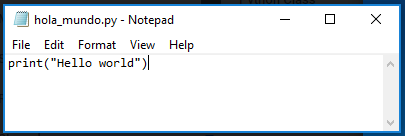
Personal Why not…

GUI development (however there are companion frameworks that can be added e.g. Tkinter, Django)

Code with multiple modules (especially ArcPY) can at times be difficult to transfer to other systems

## Creating a Python script from scratch & the Print Statement

* Creating from scratch easy,
  + Python file is a text file with a .py extension
  + Can also copy/paste old python files



* Print Statement
  + Python statement to enter text in the shell (command line window to execute python)
  + Useful for providing information while the script runs

## IDE

* An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger…
* The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated… *https://en.wikipedia.org/wiki/Integrated\_development\_environment*

Why use IDE’s

* Easier code editing (longer scripts, expanding/contracting code, code navigation, editing tools, text colors, etc.,)
* Debugging (various tools)

Common Python IDE’s

* IDLE -  bundled with the default install of Python, Shell window, Editor Window, Debug Control Window
* MS Visual Studio – my favorite!!!!
* PyScripter
* PyCharm
* Many many more…

Jupyter Notebooks (discussed in depth later in this)

* Non-traditional IDE
* Visualization capabilities (graphs, tables, etc.)
* Sharing

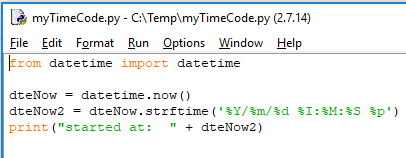
## python Versions

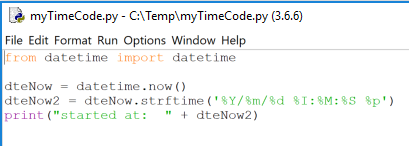
2x vs 3x

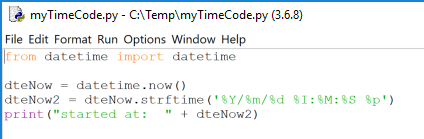
* Syntax
* ArcGIS Desktop vs ArcGIS Pro
* IDLE startup .bat file path examples
  + 2x
    - C:\Python27\ArcGIS10.6\Lib\idlelib\idle.bat
    - C:\Program Files\MySQL\MySQL Shell 8.0\lib\Python2.7\Lib\idlelib\idle.bat
  + 3x
    - C:\Users\#username#\Anaconda3\Lib\idlelib\idle.bat
    - C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\Lib\idlelib\idle.bat
* IDE considerations – make sure the IDE is configured with the Python version you want

**Feel free to find the idle.bat file on your system during this presentation!!!!**

IDE considerations – make sure the IDE is configured with the Python version you want





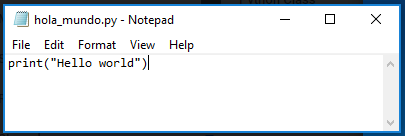


## Exercise 1: Hello World!

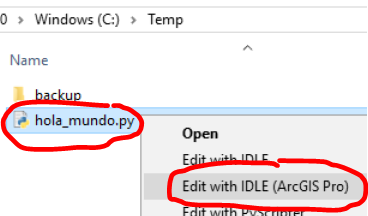
Introductory Python concepts/ methods

* What is Python & Why use Python
* Creating a Python Script from Scratch
* The Print Statement

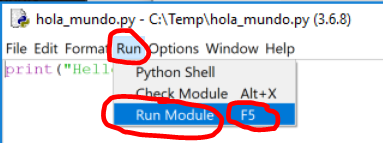
1. Open a text editor (e.g. MS Notepad)
2. Enter print(“Hello world”) 🡪 save as filename hola\_mundo.py on C:\temp



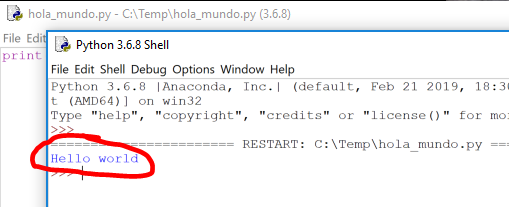
1. In a file browser, right click 🡪 hola\_mundo.py 🡪 select “Edit with IDLE…”



1. Note: explore intellisense functionality. Enter the text “pr” 🡪 ctrl-spacebar 🡪 tab
2. Under the Run menu, click Run Module OR click the F5 shortcut key

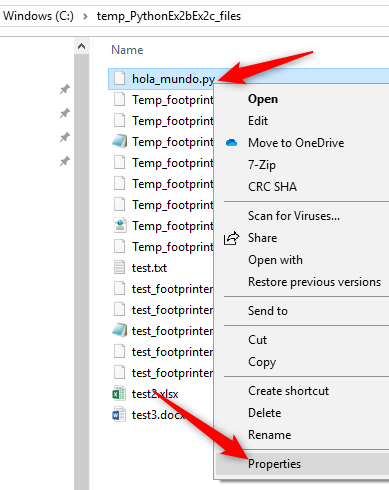


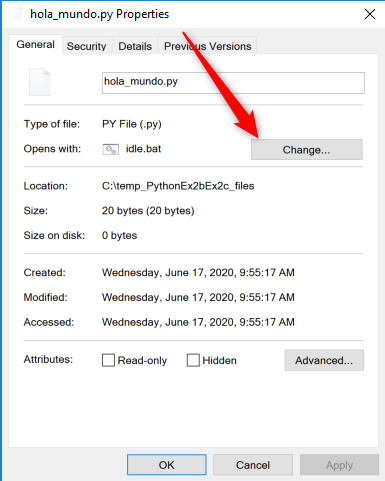
1. Rerun from step 3 however select “Edit with IDLE”, do you notice a difference?

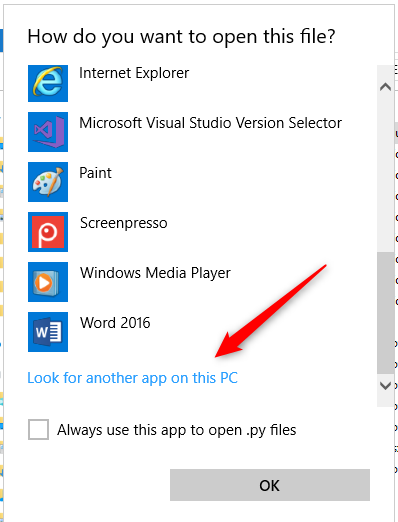


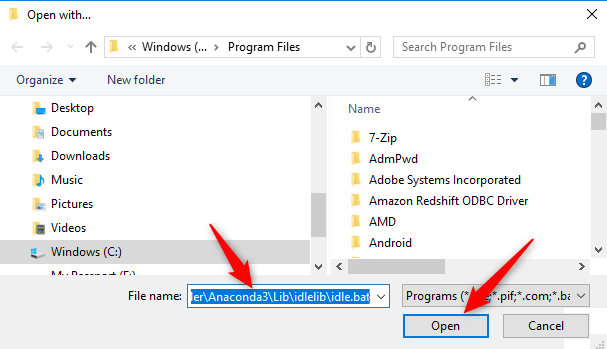
## If the “Edit with IDLE” option is not showing (e.g. Anaconda install)

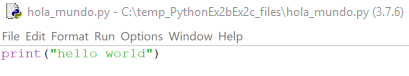
1. Find C:\Users\#username#\Anaconda3\Lib\idlelib\idle.bat







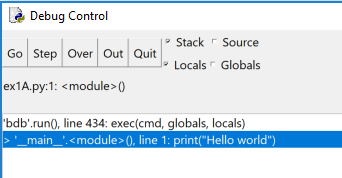


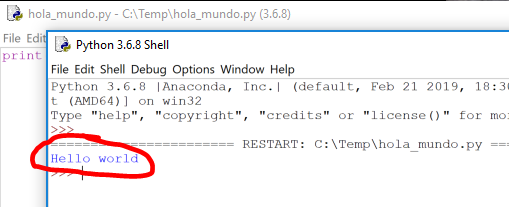


## Debugging

* What – run the code while giving details on most all portions of the code
* Why – gives much more ability to get python code to run properly
* How
  + Control speed of execution
  + Break points
  + Inspect variables
  + Step into, over out
* What is the Python IDLE Shell: Python interpreter in interactive mode is commonly known as Python Shell.

[*https://overiq.com/*](https://overiq.com/)



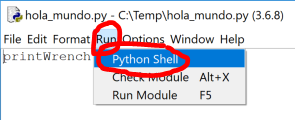


## Exercise 1b: Hello World Debugging

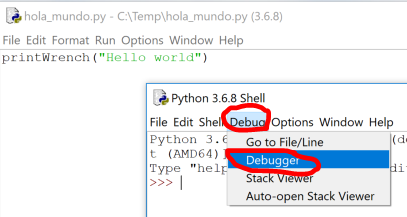
Introductory Python concepts/ methods

* IDE
* Python Versions
* Debugging

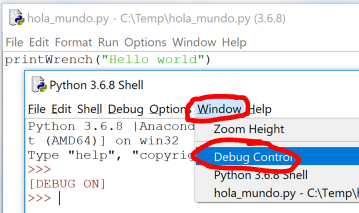
1. Change print command to printWrench 🡪 save. Note: this intentionally produces an error
2. From IDLE, click Run 🡪Python Shell



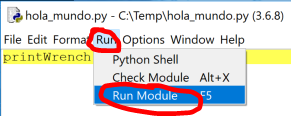
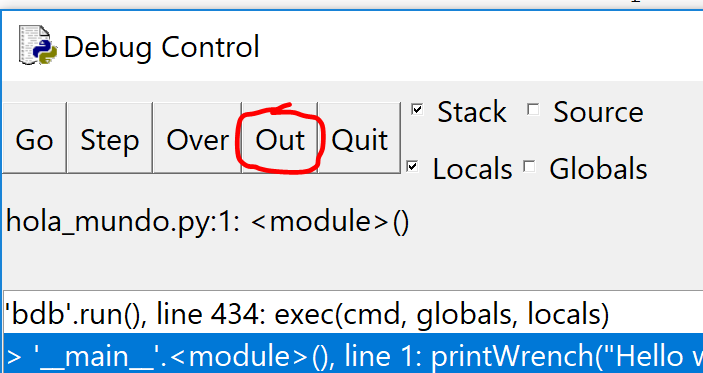
1. From Shell, click Debug 🡪 Debugger



1. Click Window 🡪 Debug Control

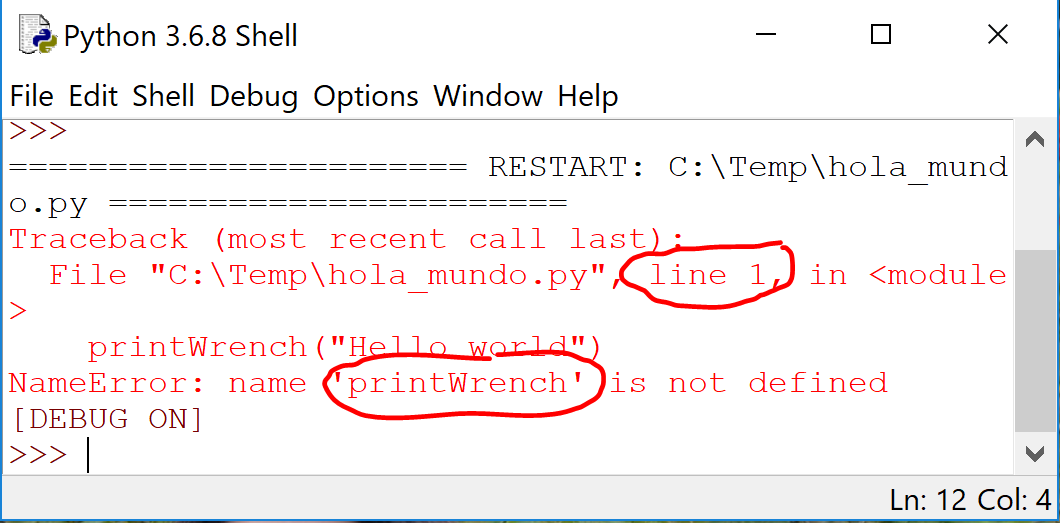


1. From IDLE, click Run 🡪 Run Module & from the Debug Control, click Out

1. Observe the line number and error message
2. Change printWrench back to print 🡪 save 🡪 Run

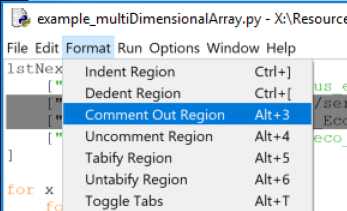
*Note: “Run module” would have shown the same error. Ex 1B was to begin using debugging.*



# Python Topics for Data Management using IDLE – Course Section 2

## Commenting

* What is it: Indicate text in a file that is not to be executed when running a script



* When to use it:
  + Script header

#title :hola\_mundo.py

#description :This is a simple hello world script

#author,org :Your name, Your org name

#date :1/22/2020

#version :0.1

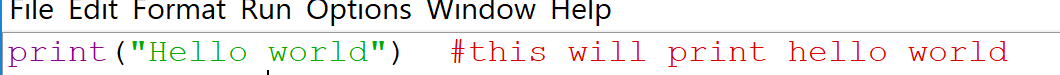
#usage :python hola\_mundo.py

#notes :

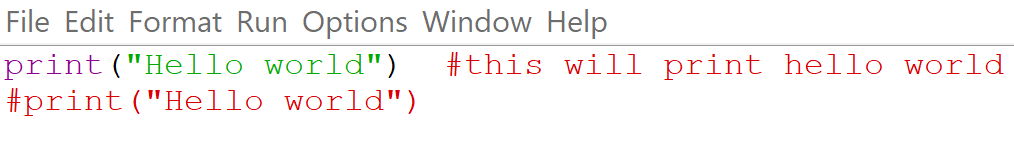
#python\_version :3.6.8

#======================================

* + Describe what’s going on in parts of the script



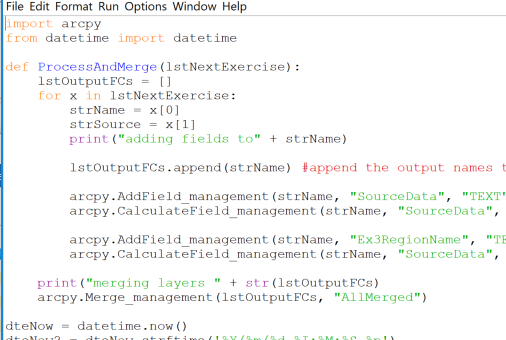
* + Save code that you don’t want to execute

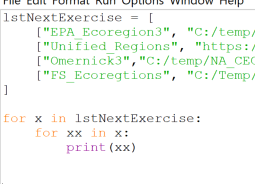


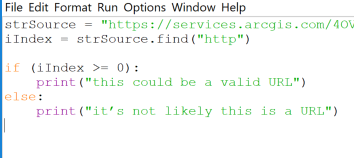
Note: most IDE’s will display commented text in a different color

## Indenting

* What is it: Indenting text in a document using tabs or blank spaces
* Why: Python requires consistent indenting for blocks of code (functions, conditional statements, loops, etc.)



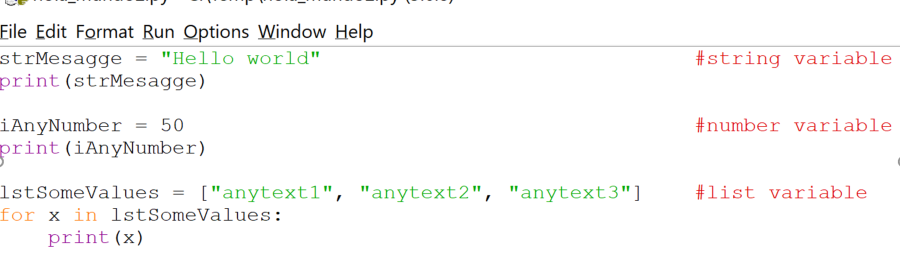


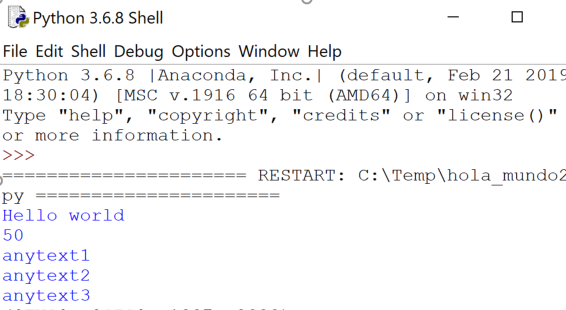


## Variables

* What is it: A name that stores a value; assign using =  *ESRI*
* Variable types - variables have data types
  + **String** - Strings are defined either with a single quote or a double quotes.
  + **Number** - Python supports two types of numbers - integers and floating point numbers. (It also supports complex numbers, which will not be explained in this tutorial).
  + **List** – Lists are very similar to arrays. They can contain any type of variable, and they can contain as many variables as you wish. Lists can also be iterated over in a very simple manner.

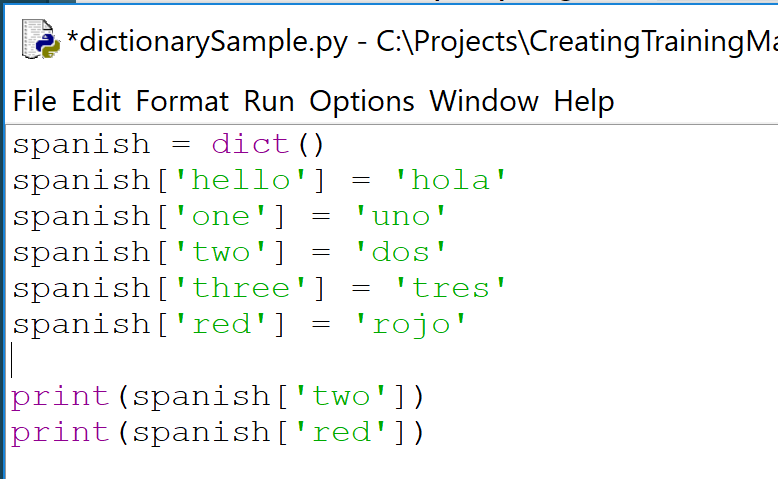
[*https://www.learnpython.org/*](https://www.learnpython.org/)

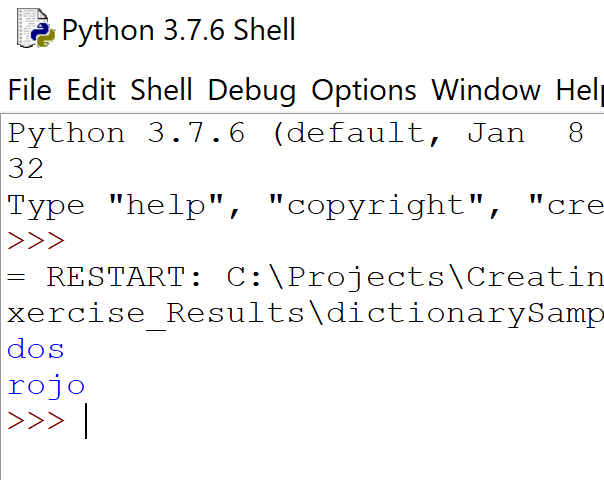




* + **Dictionary** - a collection of words matched with their definitions. Given a word, you can look up its definition.

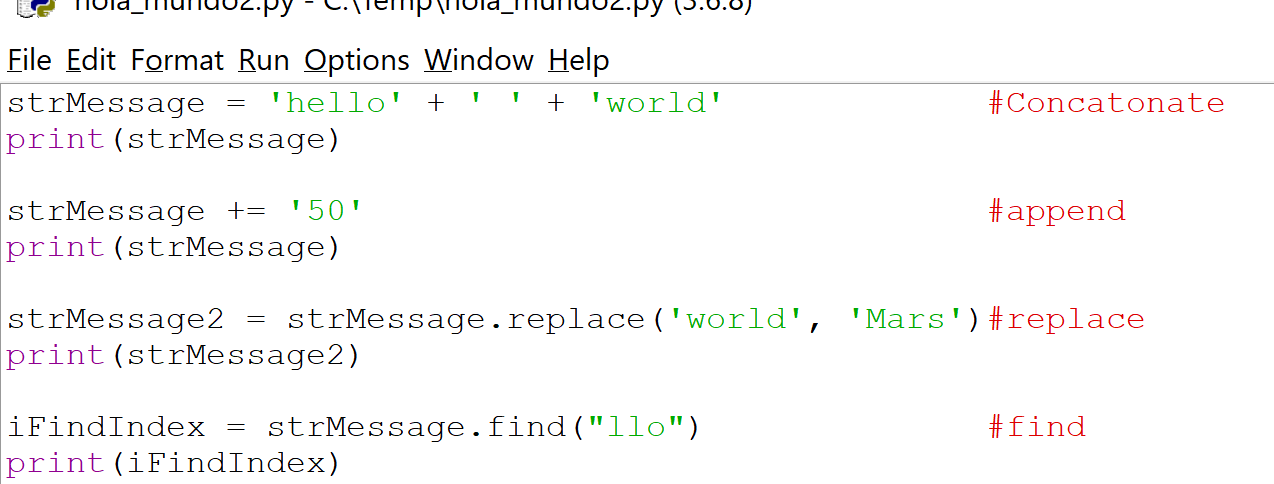
*http://anh.cs.luc.edu/*





## More on String Variables

* Manipulating string – useful when bulk processing of data
  + Concatenate
  + Append
  + Replace
  + Find



* Valid use of paths in Python

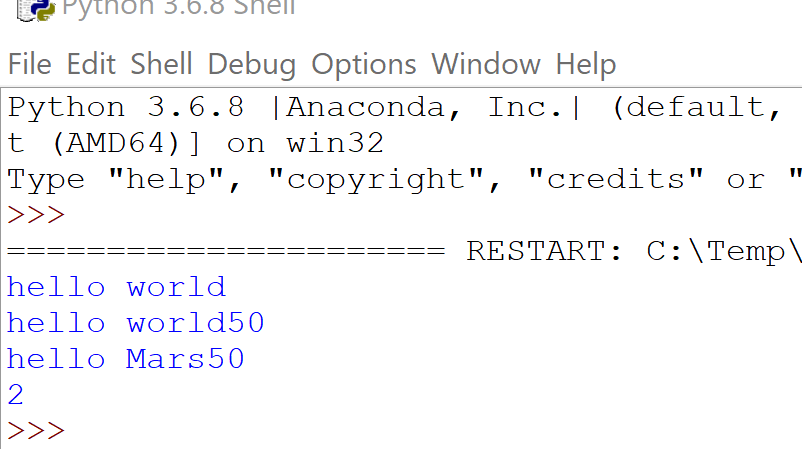
strPath = “c:/temp/roads.shp”

strPath = “c:\\temp\\roads.shp”

strPath = r“c:\temp\roads.shp”

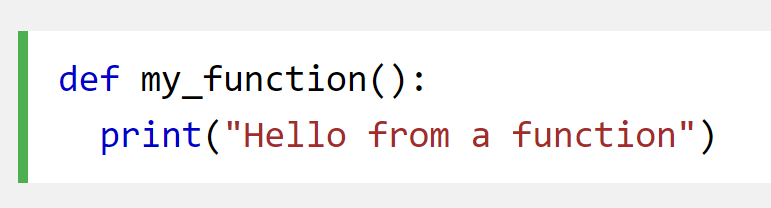
* Not Valid use of paths in Python

strPath = “c:\temp\roads.shp”



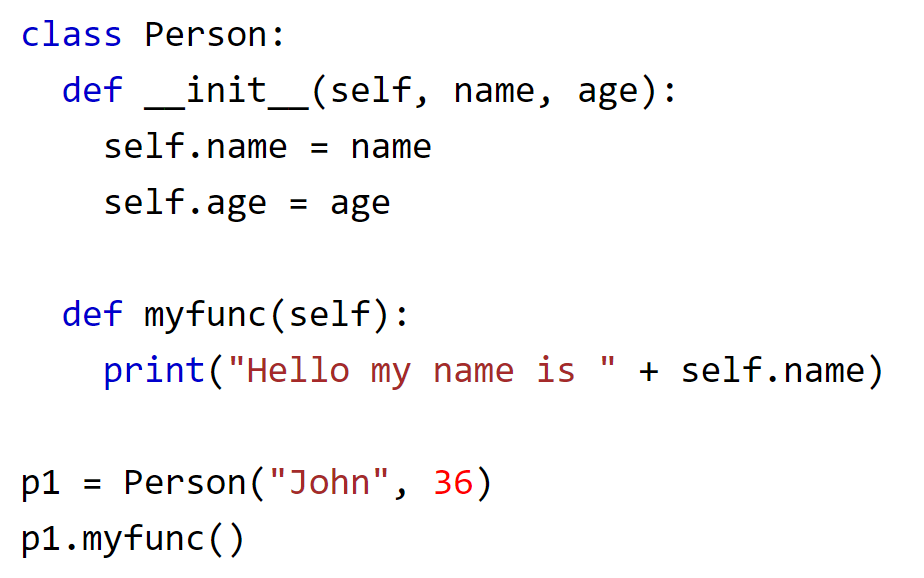
## Functions, methods, properties

* Python Function:
  + A function is a block of code which only runs when it is called.
  + You can pass data, known as parameters, into a function.
  + A function can return data/variables as a result
  + https://www.w3schools.com/



* Python Method:
  + Objects can contain methods. Methods in objects are functions that belong to the object..

<https://www.w3schools.com>



* Python Properties (and attributes):
  + Objects can contain attributes. Attributes are described by data variables for example like name, age, height etc. Properties are special kind of attributes which have getter, setter and delete methods

https://www.tutorialspoint.com/

## Python Classes

* What is a it: Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating object.

https://www.w3schools.com/

* Why: Classes provide a means of bundling data and functionality together. Creating a new class creates a new *type* of object, allowing new *instances* of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state.

*https://docs.python.org/*

## Python Modules

* What is it: File consisting of Python code of statements, functions, classes, variables, and/or other modules.
* How it is used:
  + Line of code added: import #module name#
  + First python (interpreter) searches for a built-in module (i.e. dir, os, sys) then in a list of system directories, by the environment variable sys.path (folder containing the input script, PYTHONPATH shell variable PATH, install-dependent default)
  + Python can cache complied versions of .py files, as .pyc files, for speed of loading
  + Python comes with library of standard modules (e.g. sys)
* Why use it:
  + Simplicity (bite size), Maintainability (updates when needed), Reusability (no copies needed), Scoping (avoid overlap with other areas of coding)
  + Benefit from the development of additional modules in numerous niches of Python by Data Manager and GIS professionals and programmers from many different disciplines.

https://knowpapa.com/

## Python packages and Libraries

* Python Package: a collection of Python modules
* Python Library: the term library does not have any specific contextual meaning in Python. When used in Python, a library is used loosely to describe a collection of the core modules. “Additional libraries” refer to those optional components that are commonly included in Python distributions.

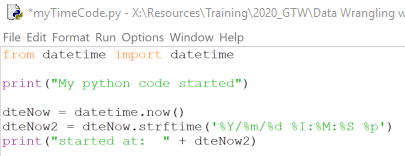
<https://knowpapa.com/>

## Exercise 2A: Time, Time, tell me what time!

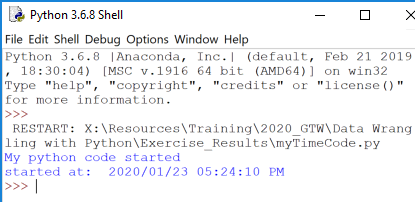
Introductory Python Concepts/Methods

* Commenting
* Variables
* More on String Variables
* Python Classes, Modules, Packages
* Methods

1. Make a python file that “print’s” the text “My python code started” 🡪 save as “myTimeCode.py”
2. Before the print statement, import the datetime module with the date time classes, enter “import datetime”
3. After the import, enter
   * 1. dteNow = datetime.now()
     2. dteNow2 = dteNow.strftime('%Y/%m/%d %I:%M:%S %p')
     3. print("started at: " + dteNow2)



1. Save 🡪 Run



## For Loops, Iterating

* What is it: A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
* Python supports different kinds of loops:
  + While - execute a block of statements repeatedly until a given a condition is satisfied.
  + For in - For loops are used for sequential traversal (list, string, dictionaries, etc.)
  + Iterating by index of sequences (Range) - We can also use the index of elements in the sequence to iterate
  + else statement with for loops - Combine else statement with for loop like in while loop
  + Nested - one loop inside another loop
  + Loop Control Statements
    - Continue Statement - returns the control to the beginning of the loop
    - Break Statement - brings control out of the loop
    - Pass Statement - pass statement to write empty loops

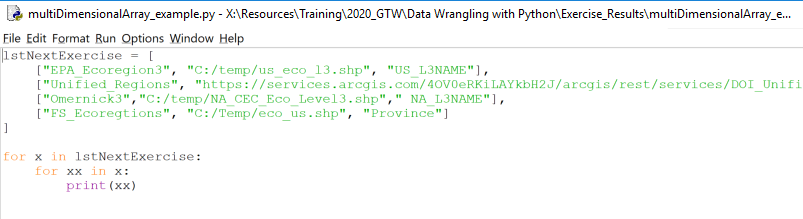
https://www.geeksforgeeks.org/

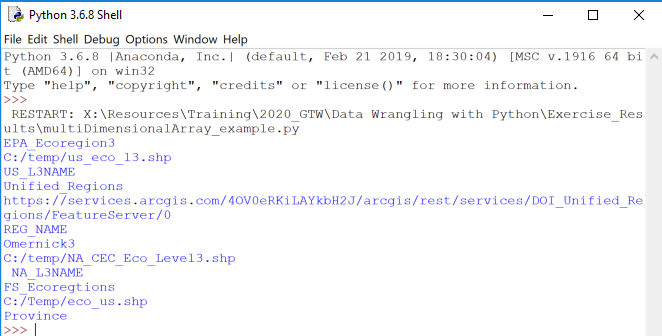
* Why: Handle repetitive tasks
* What is a multi-dimensional list: Multi-dimensional lists are the lists within lists

https://www.geeksforgeeks.org/

## For Loops, Iterating through a list Examples

Example of looping through a 2 dimensional list



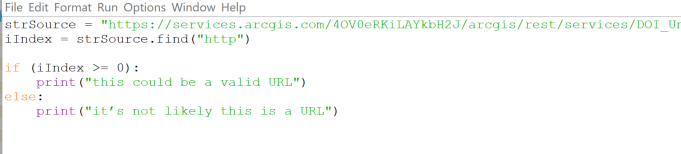


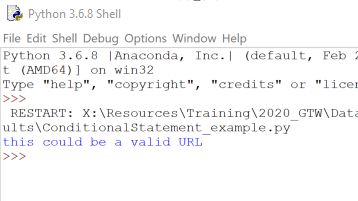
## Conditional Statements

* What is it: The “If” statement
* Why: In programming, very often we want to check the conditions and change the behavior of the program.

*https://www.pythonforbeginners.com/*

* Example of simple conditional statement:





## User-Defined functions

* What it is: Python includes many built-in functions,  users can also define functions. A function is a reusable block of programming statements designed to perform a certain task.

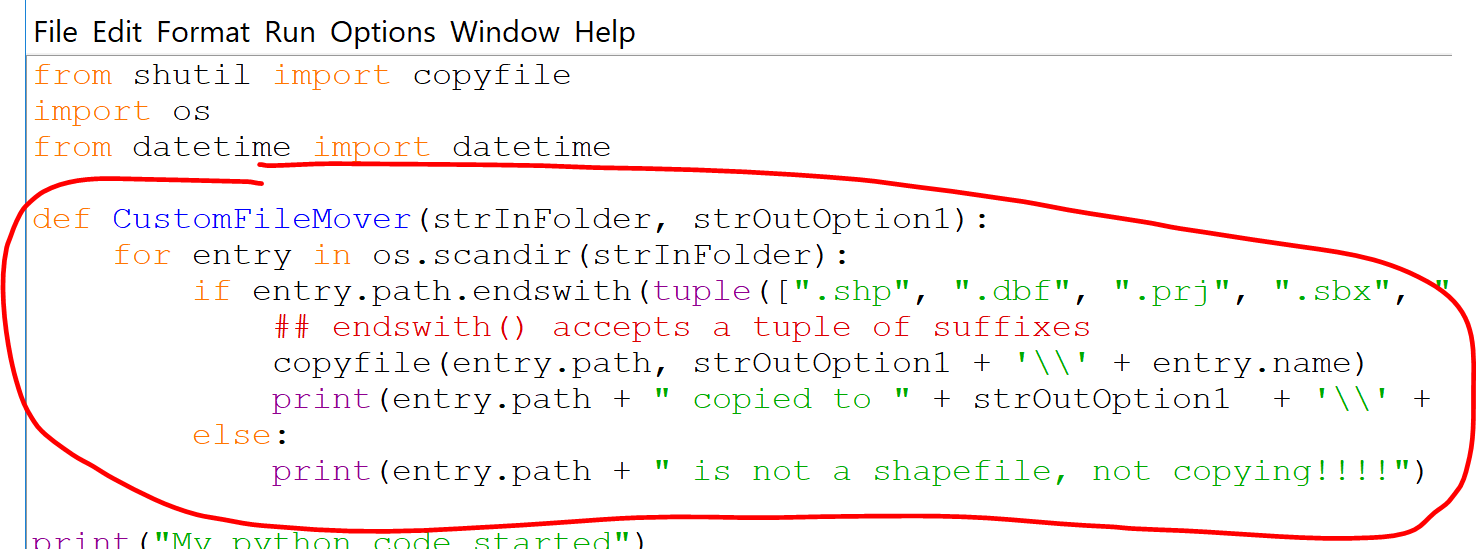
*https://www.tutorialsteacher.com/*

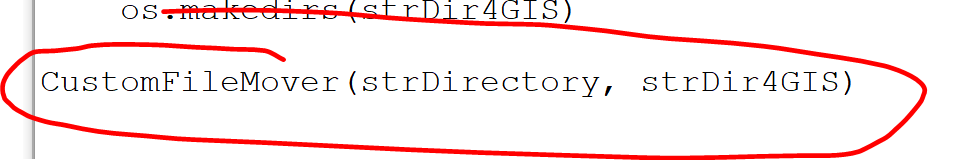
* Why:
  + Organize code for readability and reusability
  + Helps with DRY

***Don't repeat yourself****(****DRY****, or sometimes****do not repeat yourself****) is a principle of software development aimed at reducing repetition of software patterns, replacing it with abstractions or using data normalization to avoid redundancy.*

[*https://en.wikipedia.org/*](https://en.wikipedia.org/)

* + How: To define a function, Python provides the def keyword, add colon, proper indentation, call the function



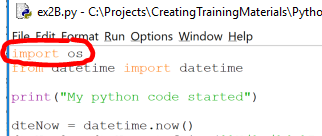


## Exercise 2B: Manage them files

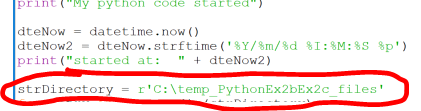
Introductory Python concepts/methods

* For Loops, Iterating through a dictionary
* Conditional Statements
* Python os module usage

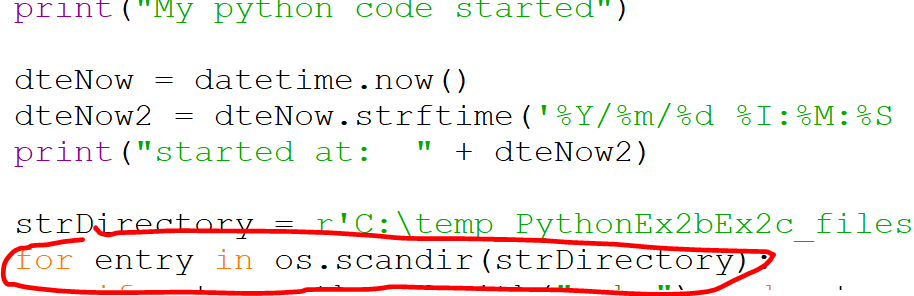
1. Copy the Exercise 2A resulting Python file and rename ex2B.py
2. Insert a new line at the top of the .py file and add “import os” to import the os module



1. After the last print statement, Declare a string variable named strDirectory and set it to equal… r'C:\temp\_PythonEx2bEx2c\_files‘ Note: these are the temporary exercise files



1. Enter the first line of the loop by copy/pasting … for entry in os.scandir(strDirectory):



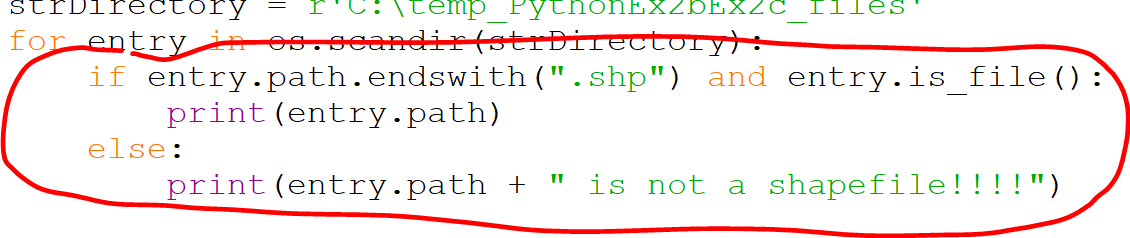
1. Create a conditional statement that checks the file extension for .shp and that the it is indeed a file (vs. a folder) by copy/pasting …

if entry.path.endswith(".shp") and entry.is\_file():

print(entry.path)

else:

print(entry.path + " is not a shapefile!!!!")



1. Save 🡪 Run 🡪 view the PythonEx3A.gdb contents

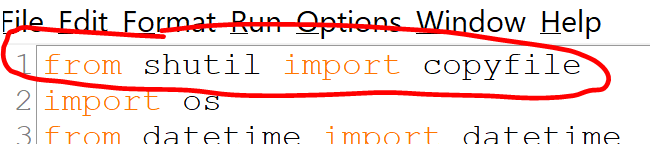
<https://www.newbedev.com/python/howto/how-to-iterate-over-files-in-a-given-directory/>

## Exercise 2C: Extra Credit, Keeping it DRY

Introductory Python concepts/ methods

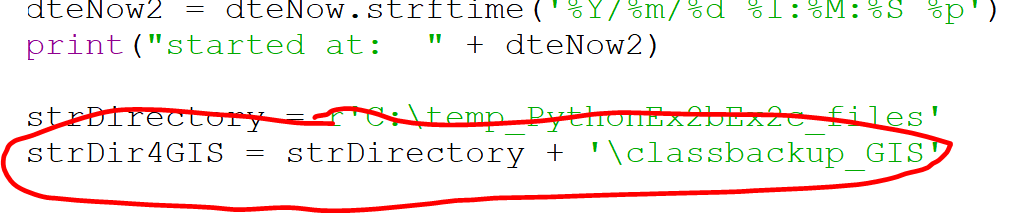
* More methods
* User defined function

1. Copy the Exercise 2B resulting Python file and rename ex3C.py
2. Insert a new line at the top of the .py file and add “from shutil import copyfile” to specify importing the copyfile method from the module



1. After the string variable add a new variable strDir4GIS and set it to equal…

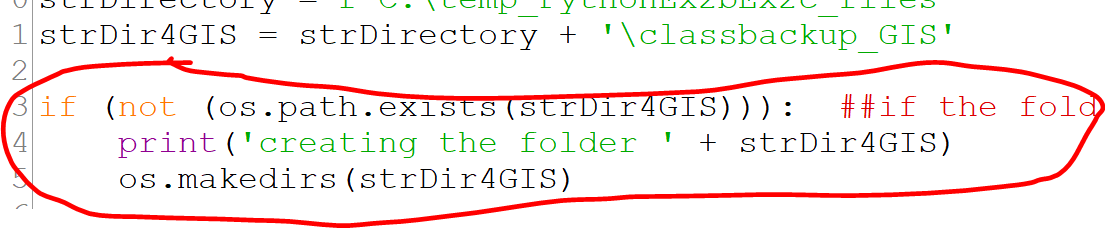
strDirectory + '\classbackup\_GIS‘



1. Add the following 3 lines of code to create a new folder if does not exist
2. if (not (os.path.exists(strDir4GIS))): ##if the folder does not exist

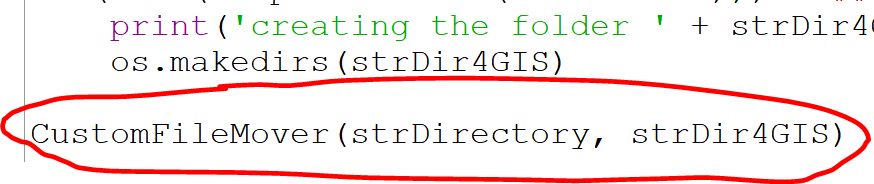
print('creating the folder ' + strDir4GIS)

os.makedirs(strDir4GIS)



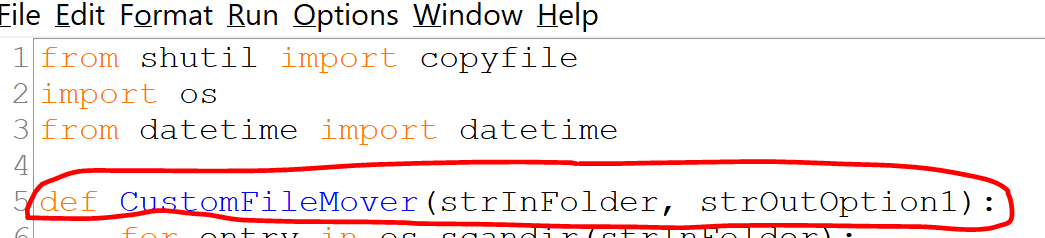
1. Add a line of code, to call the user defined function that you will create next, and pass the 2 variables with folder pathing…

CustomFileMover(strDirectory, strDir4GIS)

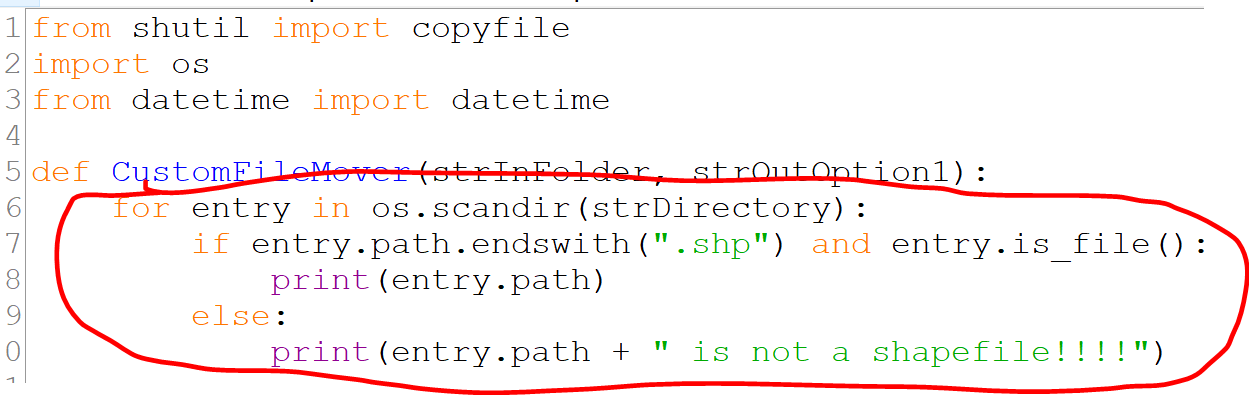


1. Create the first line of the user defined function, under the From datetime import datetime line…

def CustomFileMover(strInFolder, strOutOption1):



1. Cut/paste the for loop content and nest (using the tab key) under the user defined function’s 1st line



1. Swap out the for loop’s content with the following…

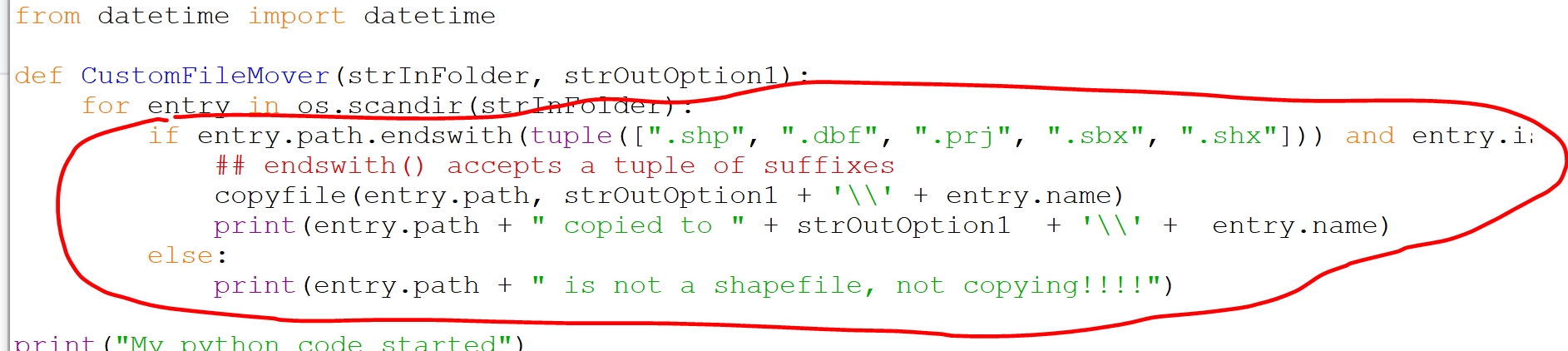
if entry.path.endswith(tuple([".shp", ".dbf", ".prj", ".sbx", ".shx"])) and entry.is\_file(): ## endswith() accepts a tuple of suffixes

copyfile(entry.path, strOutOption1 + '\\' + entry.name)

print(entry.path + " copied to " + strOutOption1 + '\\' + entry.name)

else:

print(entry.path + " is not a shapefile, not copying!!!!")



1. Save 🡪 Run 🡪 view the changes in the C:\temp\_PythonEx2bEx2c\_files folder

## Topics to explore on your own

* Error Trapping
* Python Arguments
* Packaging into tools for ArcGIS Pro/ArcMap
* Module installation, installing Python modules & packages with pip