Monroe Township Library Coding Bootcamp

Class 7 Notes

* Variables and Scope
* File handling, reading and writing to files
* Importing modules
* Python Standard Library

**Variables and Scope:**

* Scope in Python refers to the availability of variables in your code
* Variables declared in your python file, outside of any functions, are considered to be in the global scope
  + Global scope means the variable can be accessed both inside and outside of functions
* Variables declared inside of a function are considered to be in local scope, meaning they can only be accessed within that function
  + Trying to access a variable outside of the function it was created in will give you an error
  + If a function contains a local variable with the same name as a global variable, it will use the value of the ***local*** variable, the global variable will not be changed
  + If you do need to modify the global variable inside of a function, you can use the global keyword before the variable name at the top of your function, however this is usually considered bad practice
    - If you have a larger program with multiple functions, it is hard to keep track of which functions are modifying global variables, and it is not obvious to others who might read your code

**File Handling:**

* We can use the open function to do basic file handling with text files
  + The open function takes 2 string arguments, the first one is the name of the file and the second is a command (“r” for read, “w” for write, “a” for append)
  + Always remember to close a file when you’re done with it using the close() method
* In ‘write’ mode, contents of the file can be overwritten, and if a file doesn’t already exist with that name, it will be created
  + The write() function takes a single string as an argument
  + The writelines() function takes a list as an argument and will write each individual element to the file
* In ‘read’ mode, the file can be read but not edited
  + The read() function returns the entire contents of the file as a string
  + The readline() function returns a single line and then moves to the next line in the file
  + The readlines() function returns all the lines of the document as a list, where each line is an element of the list
  + Using the seek() function with an argument 0 will take you back to the start of the file
* In ‘append’ mode, you can write to the end of a file but the original contents won’t be overwritten
  + You can still use the same functions as ‘write’ mode but anything you write to the file will be appended instead of erasing old data
* You can use the with keyword as a shortcut that opens and closes files automatically using the following format:
  + with open([filename], [command]) as [variable name]:

[whatever you want to do with your file goes here]

* in this case, you don’t need to call close(), the file closes automatically when it reaches the end of the code block

**Importing modules:**

* You can access functions, variables, and classes from other python modules using the import keyword and the name of the module you are importing
  + Typically all imports are done at the top of your file
* Elements of a module (functions, classes, etc.) can be accessed using dot notation with the name of the module followed by the desired element
* You can use import as to change the name of the module you’re importing (i.e. import tkinter as tk)
  + When you reference the module in your code, you would use the name you chose instead of the module name
* You can also just import specific elements from a module using from, import
  + In this case, you are specifically importing those elements, so you don’t need to use dot notation with the module name to access them

**Project: Hangman**

**Check class files at** [**github.com/monroecoding**](https://github.com/monroecoding)

Chris DiFazio

Adult Services Librarian

cdifazio@monroetpl.org