Introduction to Coding with Python

What is Programming?

- Programming is a set of instructions that tells the computer what to do and how to do it.
- Everything on your computer, from applications like Word to smaller applications like the Calculator are all based on code.
- Computers communicate through Binary, which is a series of 1's (On) and 0's (Off). It's far too impractical to instruct the computer through Binary.
    - The word "Hello" would be translated as
        - 01001000 01100101 01101100 01101100 01101111
        - It would be too much of hassle to communicate with a computer in this way
- Instead we use translators to communicate with the computer. A translator converts our invented programming languages into machine code (binary), thus translating our code into a set of instructions that the computer can understand and complete.
- These are the Translators"
    - Interpreters
    - Compilers
    - Hybrids if interpreters and compliers
    - Assembly

Interpreters

- Interpreters processes the code line by line.
    - After it translates a line, it sends the instructions to the computer and then moves on to the next line
    - Your code begins running before all the code is fully translated.
- The code starts running until it encounters an error.
    - The script is then stopped and the error is reported for the programmer to fix.

Compilers

- Converts all of your code into machine code through a compilation process.
- If there is an error, it will be detected during the compilation process and flagged before the code is executed.
    - This interrupts the compilation process, and thus no binary is generated.

Interpreters vs Compilers

- Interpreters
  - The programmer receives instant feedback as code is written
  - Interpreters run in the background and can use up a computer's momory resources to run.
- Compilers
  - Programs made using compilers, tend to run faster and are much more efficient.
  - The complier never runs on the computer while the program is being executed.
    - This the reason why compiler languages like C++ are used in game development. Unlike an interpreter, the complier would not be running the background and taking allocated resources away from a video game that is currently running.

Hybrids

- An example would be Java
- Hybrids compile your code through an intermediate format named Bytecode.
- The bytecode is then interpreted and executed by the runtime engine, known as a virtual machine.
  - A Java Virtual Machine enables a computer to run Java programs as well as other programs written in other languages that are also complied with Java bytecode.

Assemblers

- Translate low-level assembly language into machine code.
- An assembler enables software and application developers to access, operate and manage a computer's hardware architecture and components.

Python

- Python is often used to introduce coding concepts to beginners.
- It's because Python is easy to learn and read
- It can help explain basic and complex coding concepts
- Also with a glance you can confer what a code is doing.
- It is used for
  - Web Development
  - Software Development
  - Mathematics
  - System Scripting

Python can be used:

- On a server to create web applications.
- Alongside software to create workflows.
- To connect to database systems. It can also read and modify files.
- To handle big data and perform complex mathematics
- For rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- Python has a simple syntax that's similar to the English language.
- The syntax also allows developers to write programs with fewer lines.
- Python runs on an interpreter system, meaning that the code can be executed as soon as it's written.

How to Get Started

- https://notepad-plus-plus.org/
    - A text and source code editor.
- https://www.python.org/downloads/
- https://www.onlinegdb.com/online_python_compiler
    - Browser based complier

Hello World!

- The goal of this exercise is to have the complier output this phrase "Hello World!"
- To do so, please enter this code line:
    - *print ("Hello World!")*
- Print instructs the computer to output what is in the parenthesis to the console. The quotations signify a string, without the quotations the complier will assume Hello World! is a variable. This would cause an error.

Python Indentation

- Indentation refers to the spacing at the beginning of a line of code.
- In other programs, indentation in code is used for readability. In Python, it can cause an error if a line is not properly indented.

Comments in Code

- Comments are lines of text that are ignored by the complier.
- Comments are used to help explain what a line or block of code is supposed to do. This helps both yourself and others to know what that line or block of code is doing, or what you were trying to get it to do.
- In Python, comments are preceded by the # sign:
- Example:
    - #This is a comment
- They can be placed above a line of code or at the end of one.

Creating Variables

- Variables are containers for storing data values
- Unlike other programming languages, Python has no command for declaring variables.
- Instead, a variable is created the moment you first assign a value to it.

Basic Variable Types

Variable Type	Definition	Example
Integer	Whole Numbers	1, -5, 1000, 5678
Float	Decimals	3.14, 2.0, -5.8, .00001
String	Characters	“This is a string:

Second Exercise

x = 5

y = “is bigger than”

z = 2.5

print(x, y, z)

x = “Five”

print(x, y, z)

Variable Names Exercise

ageChild = 10

nameChild = “Timmy’s”

_Present = 25.25

partyDate = “February 12th, 2020.”

print(nameChild, ageChild, “th birthday party is on”, partyDate, “Please bring a gift of $”, _Present)

Removing Spaces for Output

ageChild = 10

nameChild = “Timmy”

_Present = 25.25

partyDate = “February 12th, 2020.”

print(“{}’s”.format(nameChild),“{}th birthday party is on”.format(ageChild), partyDate, “Please bring a gift of ${}”.format(_Present))

ageChild = 10

nameChild = “Timmy’s”

_Present = 25.25

partyDate = “February 12th, 2020.”

print(nameChild, “’s”, ageChild, “th birthday party is on”, partyDate, “Please bring a gift of $”, _Present, sep = ‘’)

Variable Names

- A variable can have a short or descriptive name.
- A variable name:
  - Must start with a letter or underscore character
  - Cannot start with a number
  - Can only contain an alpha-numeric characters, and underscores (0-9, A-Z)
  - Are case sensitive (age, Age, and AGE are three different variables.)

Multiple Variables

- Multiple variables can be added on the same line of code.
- **x, y, z = “Orange”, “Banana”, “Cherry”**
    - *print (x)*
    - *print (y)*
    - *print (z)*

Outputting variables

- Print is used to output a variable
- You can combine a variable with a string when using print:
- x = “awesome!”
- print(“Python is “ + x) OR print(“Python is”, x)

Operators

Operators	Syntax	Examples
Addition	+	x + y
Subtraction	-	x – y
Multiplication	*	x * y
Division	/	x / y
Absolute Value	abs()	abs(-x)
Exponent	**	x ** y

Order of Operations

Order of Operations	Examples
Parenthesis	z * (x + y)
Exponents and Roots	z ** x + sqrt(y)
Multiplication and Division	z * x / y
Addition and Subtraction	z + x - y

Comparing Numbers

Comparing Numbers	Syntax	Example
Equals	==	x == y
Does not Equal	!=	x != y
Greater Than	>	x > y
Less Than	<	x < y
Greater Than or Equal To	>=	x >= y
Less Than or Equal To	<=	x <= y

If Statement Exercise

```
if (5 > 2):
        print(“Five is greater than two!”)
```

If Statements

- Are used to set a condition to run a block of code, if the condition is proved true then the code runs. If it’s false then it fails and the complier does not run that code.
- Example:
    - num = 5
    - if num > 4:
        - print(num)
    - The variable num is giving the value of 5
    - If num(5) is greater than 4, then print(num)
        - 5 is greater than 4, so the complier runs the print which will out the value of 5 which was attached to the num variable

Else and Elif Example:

```
Total = 40.29
if total >= 50:
    print(“You get free shipping!”)
elif total >= 40:
    print(“Spend a little more to get free shipping!”)
else:
    print(“Spend $50 to get free shipping.”)
```

Else and Elif Statements

- Else is used when you want to have a different code block to run when the condition of the If statement, is not met.
- Elif (Else If) Statements
    - This is used to test for more than one condition

Learning Resources:

- https://www.w3schools.com/python/
- http://introtopython.org/
- https://www.tutorialspoint.com/python/index.htm
- https://www.youtube.com/watch?v=Q_itdXI3YeE&list=PLRJdqdXieSHN0U9AdnmwD-9QcR9hmw04d&index=1

Thank you for Attending!

Palm Beach County
Board of County Commissioners

PALM BEACH COUNTY LIBRARY SYSTEM
CONNECT. INSPIRE. ENRICH.
#pbclibrary