Data Analytics
Intro to Python

What is a programmer?

- A creator
- A problem solver
- A thinker

What is an analyst?

- A creator
- A problem solver
- A thinker

That is why they fit together so well!

What will you learn along the way?

- To think methodically
- Solve Problems
- Create small programs to practice your coding skills

Algorithms

What is an algorithm?

- Series of steps to solve the problem at hand

What is an program?

- Series of instructions telling a computer how to solve a problem

Programming is the skill that allows a computer scientist to take an algorithm and represent that solution in a program that can be followed by a computer!

Programs are written in programming languages such as Python, html, JavaScript etc.

Python Learning Materials

- Python For Everyone -- PY4E
- This program was created by Dr. Charles Severance (a.k.a. Dr. Chuck).
- He's a Clinical Professor at the *University of Michigan School of Information Technology*.
- **https://www.py4e.com/**

PYTHON

Python is considered a high-level language

- Programs must be processed before they can run
- The lowest level languages are known as Machine Languages.
- Machine Languages encode instructions in binary that is easily executed by computers

Easier to use a high-level language

- More like natural language
- Takes less time to write
- Shorter and easier to read
- More likely to be correct

- High-level languages need to be “translate” into machine language for computers to understand.

Two “ways” to Process Code

Interpreter

- Reads a high-level program and executes it, meaning it does what the program says.
- Processes a little at a time, reading lines and performing computations

Compiler

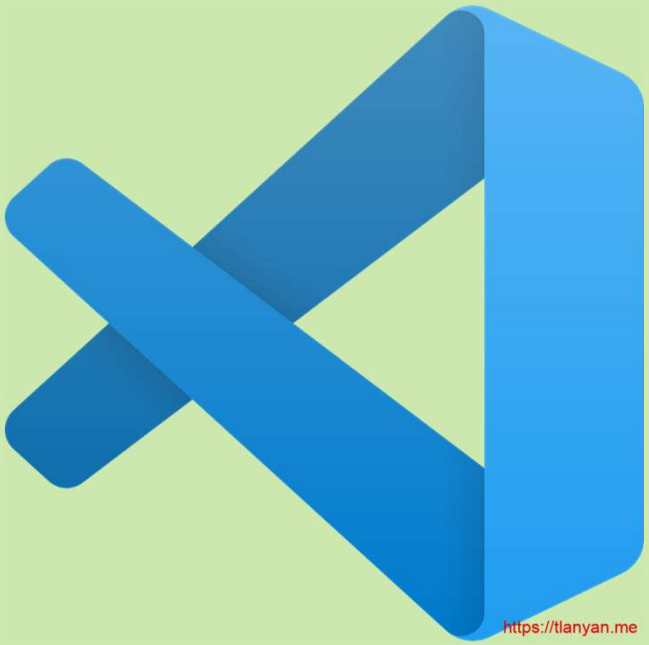
- Reads the program and translates it completely before starting to run it.

Python is an Interpreted language

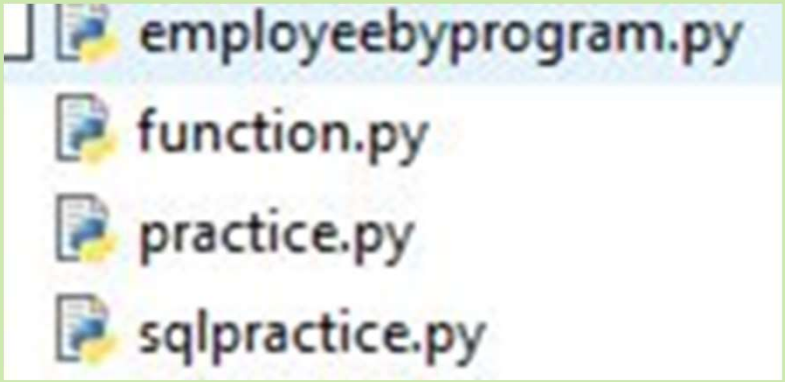
- The source code of a Python program is converted into ***bytecode*** that is then executed by the Python virtual machine.
- Python is different from major compiled languages, such as Java, C and C + +; Python code is not required to be built and linked like code for these languages.

Two Ways to use Python

Shell



Files



You've chosen to learn another language!

Be patient, and keep practicing!

Types of Languages

Natural

- Form on their own
- Spoken languages
- Slang - great example

Formal

- Are designed by people!
- Programming languages fall under this
- There are rules about grammar, syntax, and punctuation

Common Vocabulary in Python

- **VALUE** – a number or string etc. that can be stored in a variable or computed in an expression.
- **VARIABLE** – a name that refers to a value
- **VARIABLE NAME** - A name given to a variable. Variable names consist of a sequence of letters and digits that begin with a letter.

"In best programming practice, variable names should describe their use in a program, making the program self documenting."

- **Str** = a Python data type that holds a string of characters
- **Operators** = special symbols that represent a simple computation
- **Data Type** = a set of values
- **Comment** = information in a program that is meant for other programmers (used with ## infront)
- **Input** = a command in a program that prompts the user to put in an answer
- **Output** = the result of a program

Let’s write our First Program!

print(“Hello, World!”

print(Hello, World!”)

(“Hello, World!”)

Input(What is your name?”)

inputHow you?)

input(“What you names?”)

Debugging

The process of detecting and removing existing and potential errors
(aka 'bugs')
from software code.

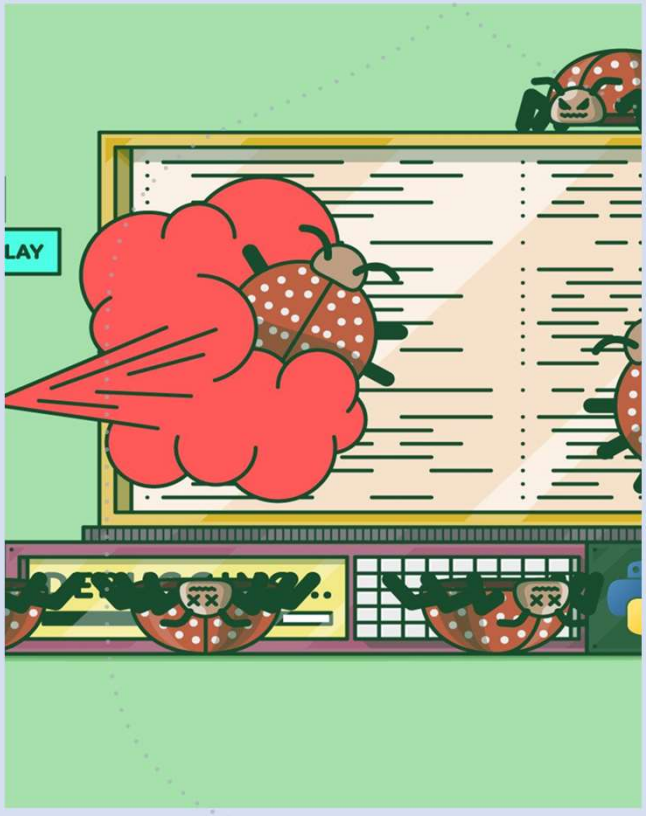
Errors could cause code to behave unexpectedly or crash.

Examples of Error Types

- **SYNTAX** – mistakes in the code like spelling, punctuation, spacing, incorrect labels, etc. Won't run.
- **RUNTIME** – occurs at the time of running or executing a program; program may hang or crash.
- **SEMANTIC** – code is grammatically correct but doesn't make any sense.
- **LOGICAL** –when instructions given do not accomplish the intended goal. EG: wrong calculations

Debugging is a VERY useful skill in programming!

- It's a science and an art
- Think of yourself as a detective looking for the clues as to what went wrong
- Consider all evidence
- Sometimes issues are minor and are quick, but others can take a while to find / fix.
- #1 Rule: Complexity is the enemy of efficiency



Helpful debugging tips

- Regularly test and review code as you are writing it
- Chunk code into smaller sections
- Explain your code aloud to yourself
- Work backward to try finding the issue
- Take an break and come back to it
- Ask someone to help you look at it

Review

- What kind of language is Python considered?
- What's a natural language?
- What are Algorithms?
- What is the program we use to write Python in?
- What's the file extension of a Python file?
- What is the area called where we put in our commands?
- What is the #1 rule of coding / debugging?