What is a *computer program*?

- A detailed, step-by-step set of instructions telling a computer what to do.
- If we change the program, the computer performs a different set of actions or a different task.
- The machine stays the same, but the program changes!
- It is easier to change the Software than the Hardware.

PROGRAMMING BASICS

- Code/Source code
  - sequences of instructions in a program
- Syntax
  - A set of legal structures and commands used in a programming language
- Output
  - The messages that are printed to the users by the program

PROGRAMMING LANGUAGES

- High Level Language
  - designed to be used and understood by humans
  - the programming languages -Java, C, C++, Python etc.
- Low Level Language
  - machine languages/assembly languages
  - computers can only execute/run a code that is written in a low level language.
  - hence, programs written in a high level language should be converted to a LLL.

HLL	LLL
• Programmer friendly • Easy to understand • Portable • Less time to read or write	• Machine friendly • Difficult • Not portable • More time to write

Levels of Programming Languages

High-level program

```
class Triangle {
  ...
  float surface()
    return b*h/2;
  }
```

Low-level program

```
LOAD r1,b
LOAD r2,h
MUL r1,r2
DIV r1,#2
RET
```

Executable Machine code

```
0001001001000101
0010010011101100
10101101001...
```

PROGRAM TRANSLATORS

- Programs to translate the HLL to LLL.
- Assembler: ALL to MLL
- Compiler:
  - HLL to MLL
  - Entire line of codes are compiled
  - Faster
  - Errors are detected and reported during the compilation phase and displayed before the execution of a program.
- Interpreter:
  - HLL to MLL
  - Each line of code is interpreted
  - Runs slower
  - Errors are reported during the actual run time.

Program Debugging: Syntax Errors vs. Semantic Errors

- Process of finding and correcting errors(bugs) in a program.
- Syntax errors are caused by invalid syntax
- eg: prnt instead of print
- Translators cannot understand instructions containing syntax errors.
- Semantic errors/logic errors are errors in program logic.
- eg: if a program computed the average of three numbers

```
(num1 + num2 + num3) / 2.0
```

- Computers do not understand what a program is meant to do, they only follow the instructions given

PYTHON

- High-level, object-oriented, interpreted language
- Python is extremely versatile
- Data Mining, Data Science, AI, Machine Learning, Web Development, Web Frameworks, Embedded Systems, Graphic Design applications, Gaming, Network development, Product development, Rapid Application Development, Testing, Automation Scripting, the list goes on.

- **Portable:** Same code can be used in different machine. No need to rewrite for multiple platforms
- Python is **dynamically-typed** – a variable can have different types

```
a = 5
a = "Five"
```

- Python has **indentation** rules
- It is **general-purpose**.

MAJOR COMPANIES THAT USE

python

Google NETFLIX facebook Instagram

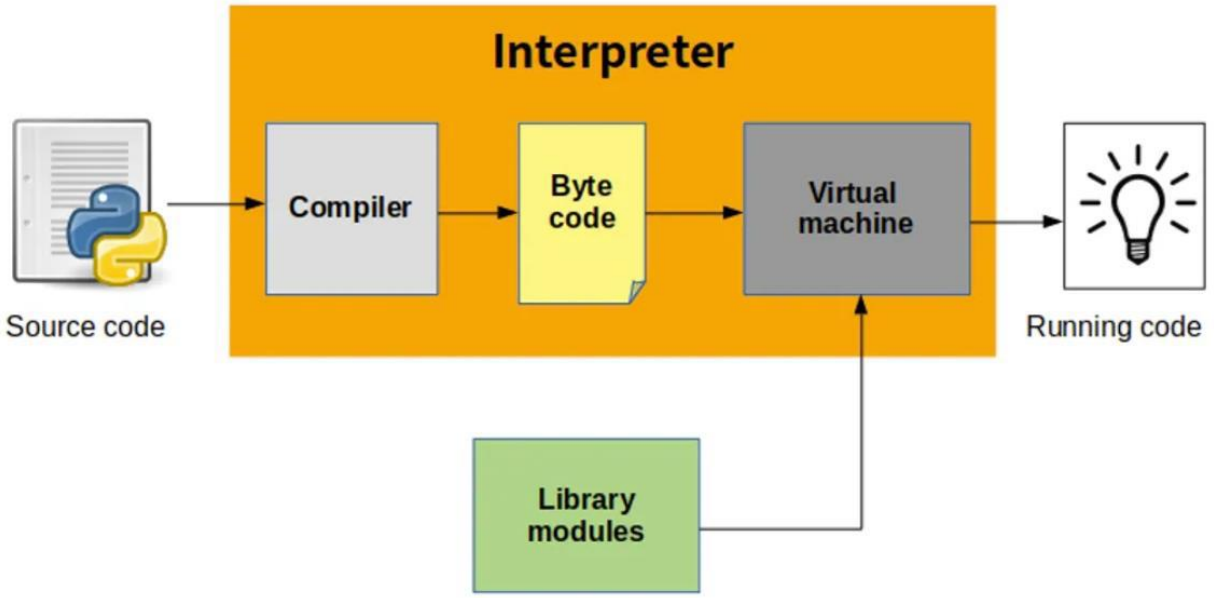
amazon Quora slack intel NASA

Dropbox ebay Spotify CapitalOne

Internal working of Python

- Uses Python Virtual Machine (PVM) to convert Python code to machine-understandable code.
- The Python Interpreter is stored in memory as a bunch of instructions. It has two components -Compiler, and Python Virtual Machine.
- Compiler - translates the source code into an intermediate language (**Byte Code.)**

- PVM - converts those byte codes into machine code.
- So that the computer can execute those machine code instructions and display the final output.



The Magic of Python

- The “>>>” is a Python *prompt* indicating that Python is ready for us to give it a command. These commands are called *statements*.
- >>> print("Hello, world“)

  Hello, world
- >>> print(2+3)

  5

```
>>> print("2+3=", 2+3)
        2+3= 5
>>>
```

Keywords

- predefined and reserved words in python that have special meanings. Keywords are used to define the syntax of the coding.

```
and       as        assert    break     class     continue  def
del       elif      else      except    finally   for       from
global    if        import    in        is        lambda    nonlocal
not       or        pass      raise     return    try       while
with      yield     false     none      true
```

Identifiers

- provide a name for a given program element with a sequence of one or more characters.
- Each variable name is an identifier.
- Rules to be followed:
  - Can only contain **alphanumeric** characters and **underscores** (a-z, A-Z, 0-9, _)
  - Must start with letters but **cannot start with a number**
  - Names are **case-sensitive**
  - Name cannot be a keyword
  - Should be one word with **no spaces** in between

- **Python does not allow special characters**
- Python is case sensitive, so rollNumber and RollNumber are two different identifiers.

Valid Identifiers		Invalid Identifiers	Reason Invalid
totalSales		'totalSales'	quotes not allowed
totalsales		total sales	spaces not allowed
salesFor2010		2010Sales	cannot begin with a digit
sales_for_2010		_2010Sales	should not begin with an underscore

Literals

- numbers, text, or other data that represent values to be stored in variables.
- Raw data assigned to variables, or, constants used while programming
  - Numeric Literals, digits 0-9, sign character and decimal point.
    - Age=22
    - Value=22.32
    - Angle=-90

- String Literals
  - delimited (surrounded) by a matching pair of either single (') or double (") quotes.

```
'A'                          - a string consisting of a single character
'jsmith16@mycollege.edu'     - a string containing non-letter characters
"Jennifer Smith's Friend"    - a string containing a single quote character
' '                          - a string containing a single blank character
''                           - the empty string
```

- Boolean Literals, assign boolean values
- Literal Collections, defines lists, tuples, dictionary, sets
- Special Literals, none literal
  - To assign null value to a variable.