

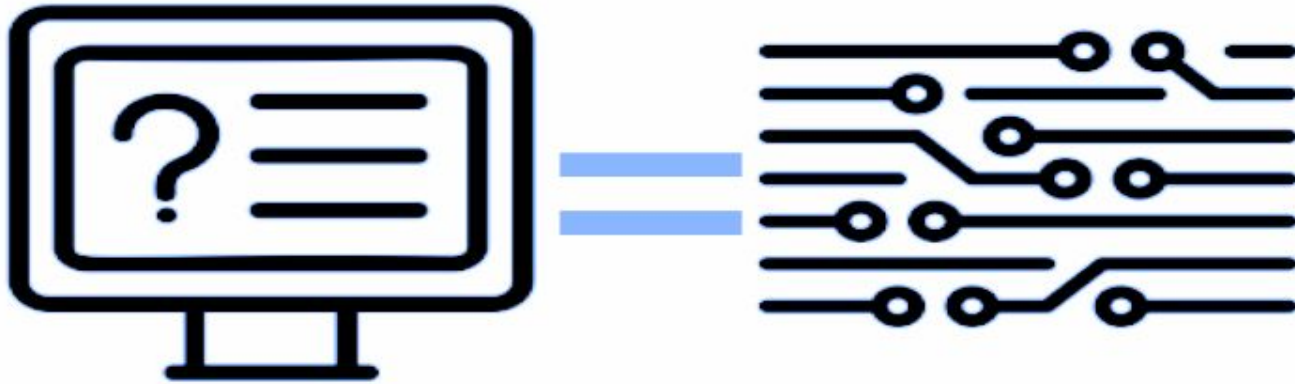
Module 1

Intro to Python & Computer Programming

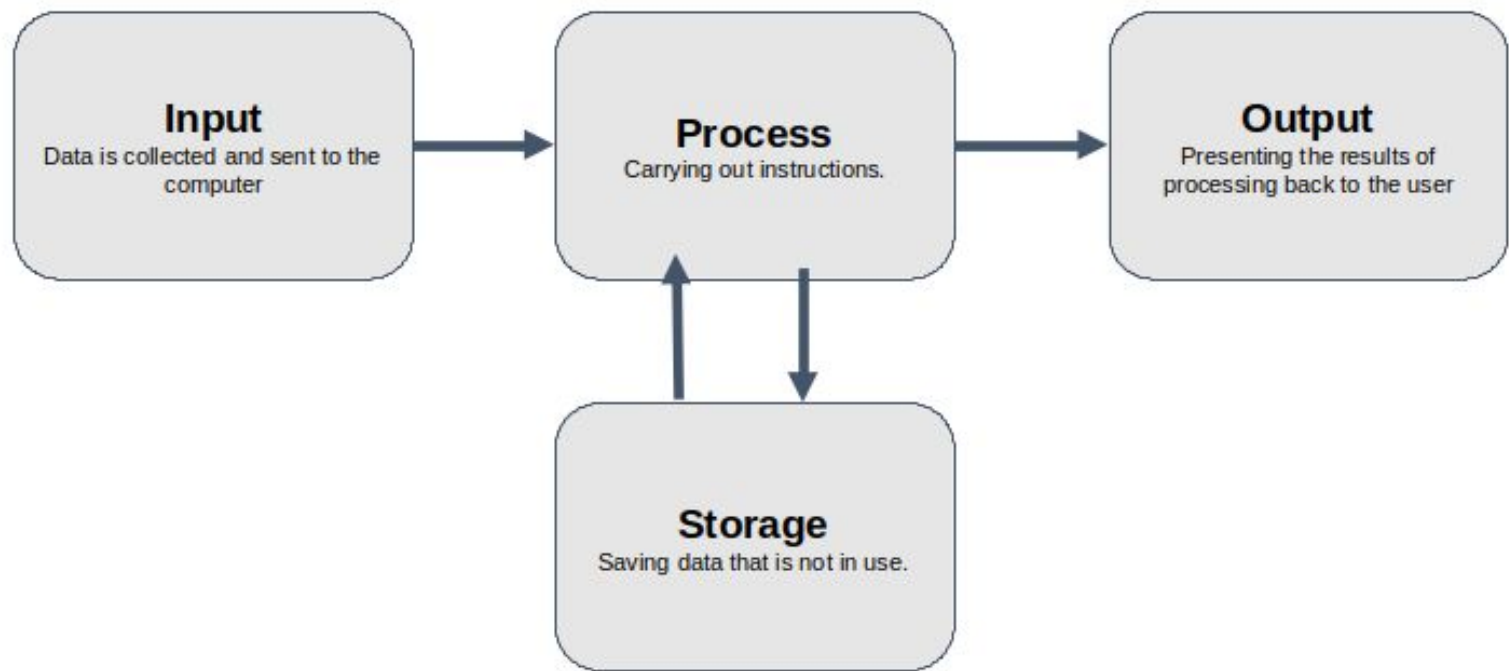
- Algorithms & Programs
- Natural Language vs Programming Language
- Machine Language vs High-Level Language
- The Python Programming Language
- Downloading and Installing Python
- Write your first program
- Errors and Debugging

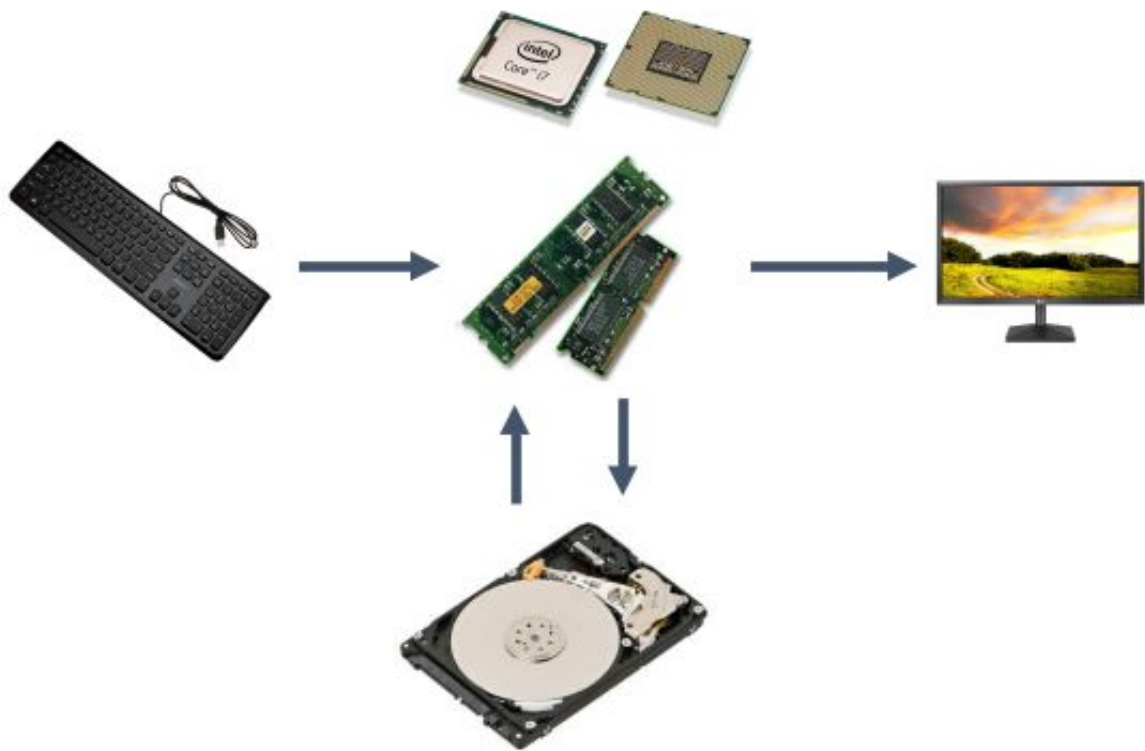
Algorithms & Programs

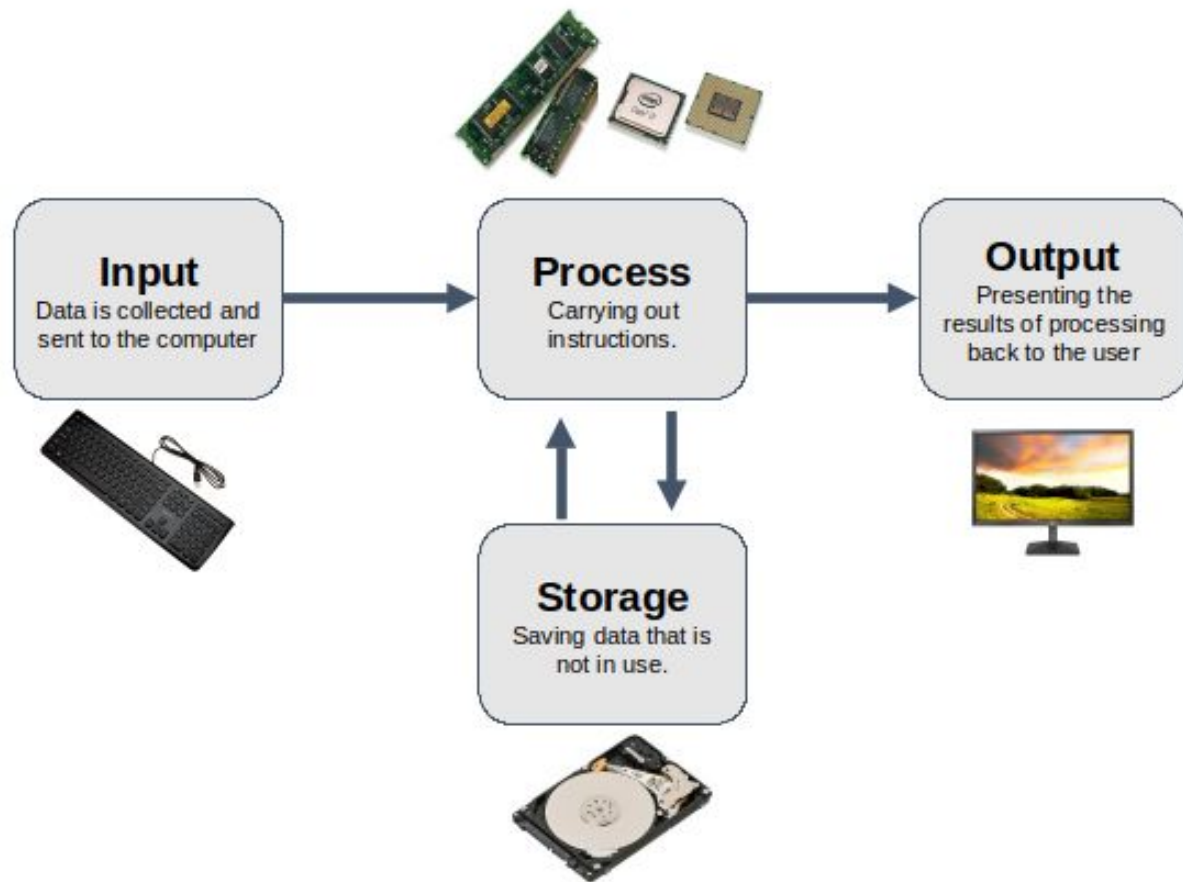
What is a Computer?



A device that can be programmed to accept data (**input**), process it into useful information (**output**) and store it away (**storage**)







Algorithm

- An algorithm is a step by step procedure or set of rules designed to solve a specific problem or perform a particular task.
- Algorithms can be expressed using natural language, flowcharts, etc.
- Algorithms are not always for computers.
- Algorithms can be simple, like adding two numbers together, or sophisticated enough to enable self-driving cars and robots to navigate their environments safely.

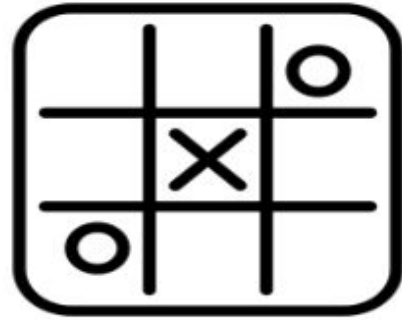
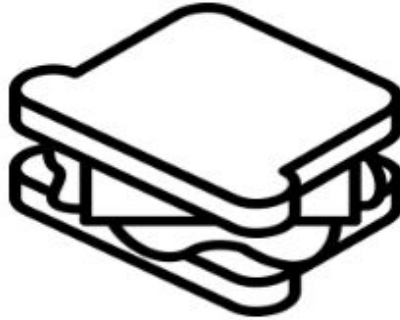
Algorithm

Algorithm (Steps for finding the largest number in a list):

- Start with the first number in the list and call it the "largest number."
- Compare the "largest number" with the next number in the list.
- If the next number is larger than the "largest number," update the "largest number" to be the next number.
- Repeat steps 2 and 3 for each remaining number in the list.
- Once you have compared all the numbers in the list, the "largest number" will be the largest number in the list.

12	5	34	6	1	21	11	3
----	---	----	---	---	----	----	---

Algorithms In the Real World



Computer Program

- A program is a concrete implementation of an algorithm using a programming **language**.
- Algorithm that are converted into code becomes computer programs.
- Programs are written by software developers to translate an algorithm's logical steps into a format that a computer can understand and execute.
- It contains the instructions and details required for a computer to execute the algorithm and produce the desired results.



Computer Program

```
python Copy code

def find_largest_number(numbers):
    largest_number = numbers[0] # Start with the first number in the list

    for number in numbers[1:]: # Loop through the remaining numbers
        if number > largest_number:
            largest_number = number

    return largest_number

# Example usage
numbers_list = [5, 12, 9, 20, 3]
result = find_largest_number(numbers_list)
print("The largest number is:", result)
```

12	5	34	6	1	21	11	3
----	---	----	---	---	----	----	---

Natural Language vs Programming Language

Natural Language vs Programming Language

- **Natural languages are the languages that people speak**, such as English, Spanish, and French.
- They were not designed by people; they evolved naturally.
- Formal languages are languages that are designed by people for specific applications.
- **Programming languages are formal languages** that have been designed to express computations.

Natural Language vs Programming Language

Semantics: Semantics is about the meaning of the sentence. - "I ate a doughnut" makes sense, but "A doughnut ate me" doesn't.

- the program has to make sense.

Syntax: Syntax is about the structure or the grammar of the language - "I am a python" is a syntactically correct phrase, while "I a python am" isn't.

- each language has its rules and they must be obeyed.

Ambiguity: is the possibility of having more than one answer. Natural Languages often does, programming language is designed to be nearly or completely unambiguous.

Machine Language vs High-Level Language

- A **high-level language** is one which cannot be understood directly by our machine.
- High-level languages are languages in which computer programs are written
- Computers are generally designed to read **machine code**.
- High level languages are **converted to bytecode** which is then converted to machine code before the program can be executed.
- **Compilation** and **Interpretation** are two different ways of transforming a program from high-level language into machine language

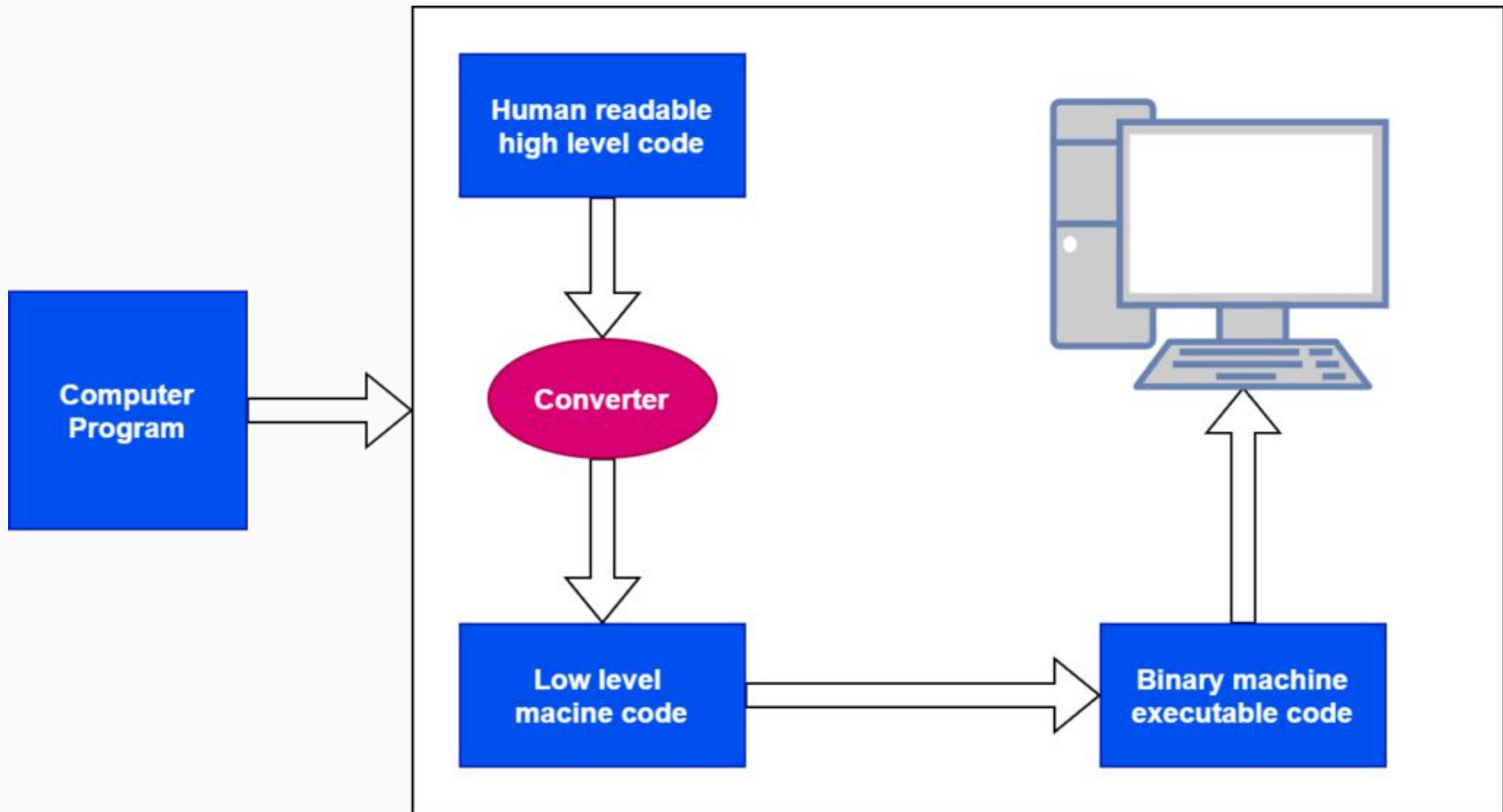
Compilation vs Interpretation

Compilation – A program is translated once (however, the translation is repeated each time you modify the source code by getting a new file (e.g., a new version of the program) intended to be run under a different operating system or hardware. The machine code. Now, the program can be distributed worldwide; the program's translation is called a compiler or translator.



Interpretation – A program is translated (or interpreted) each time it is executed. The interpreter can translate the source code (or a portion of the code) each time it is executed. This means that the interpreter has to be present on the system where the program is running. This kind of translation is called interpretation. The interpreter is intended to be executed. The program is not intended to be distributed as a single file, as it cannot just be distributed as a single file, because the end-user also needs the interpreter to execute it.





Python Programming Language

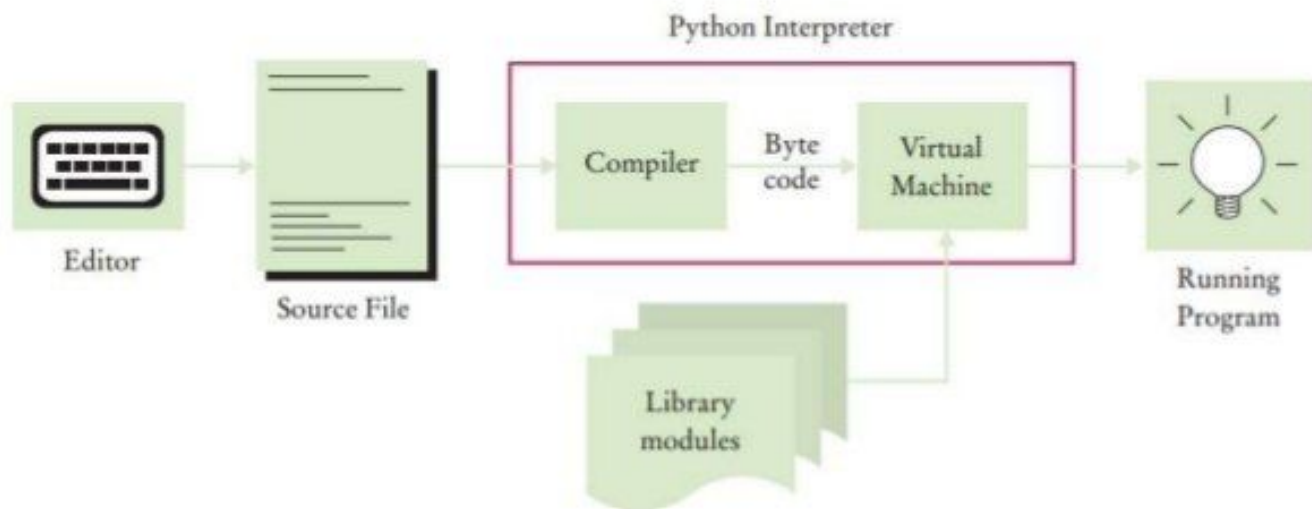
What is Python

- Developed in 1990, Python is one of the most popular *general-purpose* programming languages in modern times.
- The term “general-purpose” simply means that Python can be used for a variety of applications and does not focus on any one aspect of programming.
- Python 2.7 was widely used for a very long time, even after the release of newer versions.
- Python 2.7 has been **deprecated** as of 2020, and replaced completely by 3.xx versions, known as Python 3.

A High-Level Language

- Python falls under the category of high-level, **interpreted languages**
- Python is an interpreted language, each line is interpreted to the machine language on-the-go during execution.
- The Python interpreter is needed to program in python
- Python is free so the interpreter can easily be download and installed.

How The Python Interpreter Works



The Python Interpreter

- There are two ways to use the Python interpreter: shell mode and program mode. The example below shows the shell mode:

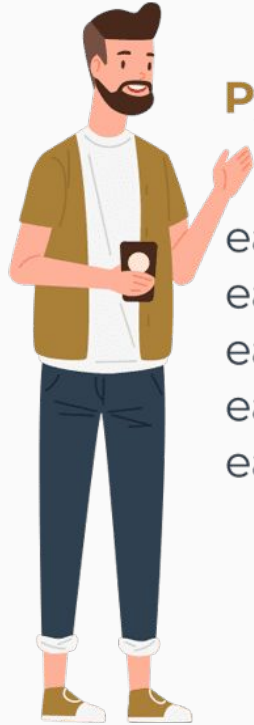
```
$ python3
Python 3.2 (r32:88445, Mar 25 2011, 19:28:28)
[GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 2 + 3
5
>>>
```

- The `>>>` is called the **Python prompt**.
- Alternatively, you can write an entire program by placing lines of python instructions in a file and then execute the file (**source code**) with an interpreter.

Python Interpreter



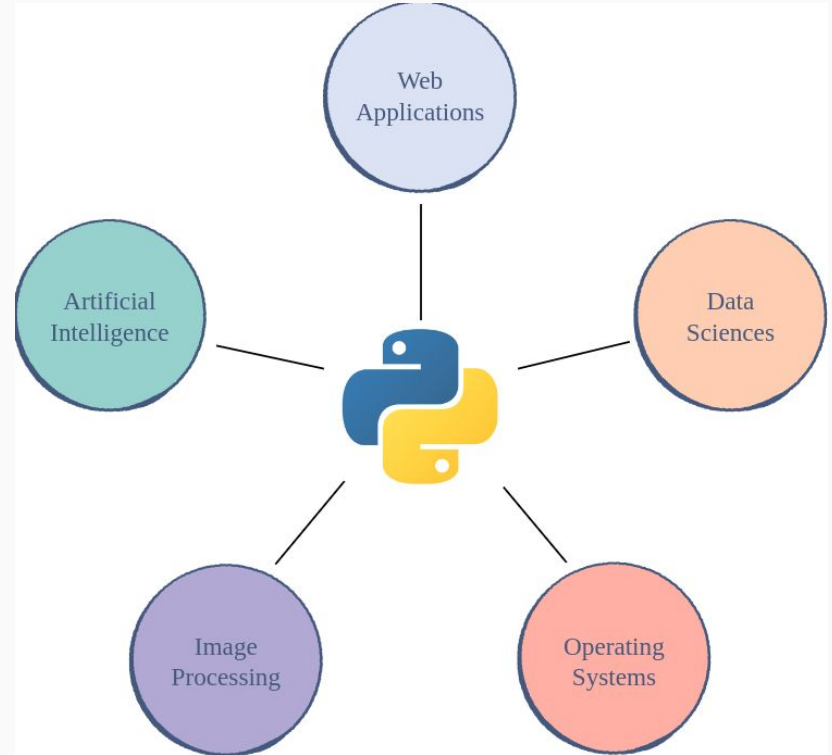
Why Python?



Python is:

easy to learn,
easy to teach,
easy to use,
easy to understand,
easy to obtain.

Python in Action



Downloading and Installing Python

Downloading and Installing Python



Your First Program

The Print Statement (function)

- Every language has a different syntax for displaying or printing something on the screen.
- In python we can use the **print** statement to print data on the terminal.

Programming – Output



```
print("your text goes here")
```

The Print Statement (function)

Programming – Output - Investigate

```
print("your text goes  
here")
```

Syntax – The Way The Code Is Written

print – no capital p.

Brackets – after print and at the end.

Text **must** go in speech marks.

Incorrect syntax = broken code that won't work = error!

First Program

The first program - the text Hello, world! is a string, so it must be enclosed in quotes.

```
print ("Hello, world!")
```

Print Multiple Pieces of Data:

To print multiple pieces of data - we just have to separate them using commas.

```
print (50, 1000, 3.142, "Hello World")
```

Print Multiple Lines:

we just have to call print() multiple times. Each call to print() will start a new line.

```
print (50)
```

```
print (100)
```

```
print (150)
```

Print on the same line:

we just have to add an **end** parameter to the print statement. The value of **end** is appended to the output and the next print will continue from here.

```
print (50, end=" ")
```

```
print (1000)
```

Comments

- Comments are pieces of text used to describe what is happening in the code
- They have no effects on the code whatsoever
- A comment can be written using the `#` character
- It's a good practice to comment your code

```
1 print(50) # This line prints 50
2 print("Hello World") # This line prints Hello World
3
4 # This is just a comment hanging out on its own!
5
6 # For multi-line comments, we must
7 # add the hashtag symbol
8 # each time
9
```

Docstrings can be used to replace multiline comments. They are encased in triple quotes, `"""`.

```
1 """ Docstrings are pretty cool
2 for writing longer comments
3 or notes about the code"""
4
```


The Print Statement



Knowledge Check


What is the most important skill for a computer scientist?

- A. To think like a computer.
- B. To be able to write code really well.
- C. To be able to solve problems.
- D. To be really good at math.



Knowledge Check

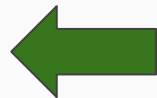
An algorithm is:

- A. A solution to a problem that can be solved by a computer.
- B. A step by step list of instructions that if followed exactly will solve the problem under consideration. 
- C. A series of instructions implemented in a programming language.
- D. A special kind of notation used by computer scientists.

Knowledge Check


A program is:

- A. a sequence of instructions that specifies how to perform a computation.
- B. something you follow along at a play or concert.
- C. a computation, even a symbolic computation.
- D. the same thing as an algorithm.




Knowledge Check

Source code is another name for:

- A. the instructions in a program, stored in a file. 
- B. the language that you are programming in (e.g., Python).
- C. the environment/tool in which you are programming.
- D. the number (or "code") that you must input at the top of each program to tell the computer how to execute your program.

Knowledge Check

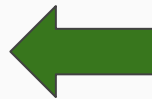
What is the difference between a high-level programming language and a low-level programming language?

- A. It is high-level if you are standing and low-level if you are sitting.
- B. It is high-level if you are programming for a computer and low-level if you are programming for a phone or mobile device.
- C. It is high-level if the program must be processed before it can run, and low-level if the computer can execute it without additional processing. 
- D. It is high-level if it easy to program in and is very short; it is low-level if it is really hard to program in and the programs are really long.

Knowledge Check


Pick the best replacements for 1 and 2 in the following sentence: When comparing compilers and interpreters, a compiler is like 1 while an interpreter is like 2.

- A. 1 = a process, 2 = a function
- B. 1 = translating an entire book, 2 = translating a line at a time
- C. 1 = software, 2 = hardware
- D. 1 = object code, 2 = byte code



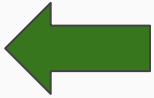
Knowledge Check

The differences between natural and programming languages include:

- A. natural languages can be parsed while formal languages cannot.
- B. ambiguity, redundancy, and literalness. 
- C. there are no differences between natural and formal languages.
- D. tokens, structure, syntax, and semantics.

Knowledge Check

What is machine code?

- A. A medium-level programming language consisting of the assembly code designed for the computer processor
- B. A low-level programming language consisting of binary digits/bits that the computer reads and understands 
- C. A high-level programming language consisting of instruction lists that humans can read and understand
- D. A low-level programming language consisting of hexadecimal digits that make up high-level language instructions