

55284A

# INTRODUCTION TO PYTHON 3

## ME

- Name: Ted Ng
- Background: Over 15 years software development, infrastructure, administration and operations experiences in cloud, consumer electronics and consulting industries.

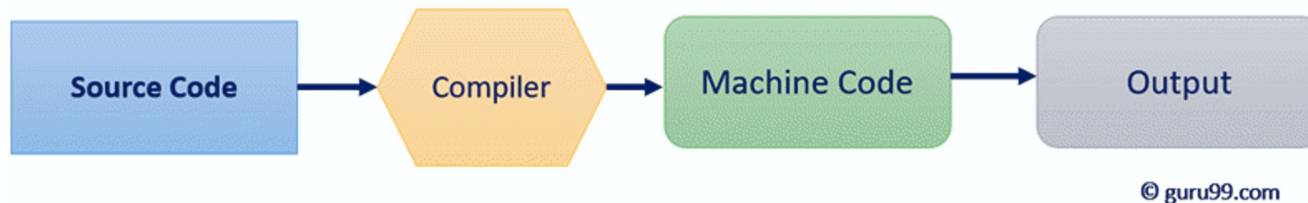
## SCHEDULE

- 9:30 AM - 12:30 PM: AM Section
- 12:30 PM - 2:00 PM: Lunch Break
- 2:00 PM - 5:00 PM: PM Section

# WHAT IS PYTHON?

- Python is an interpreted language, which means 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 C and Java as Python code is not required to be built and linked like code for these languages.

## How Compiler Works



## How Interpreter Works



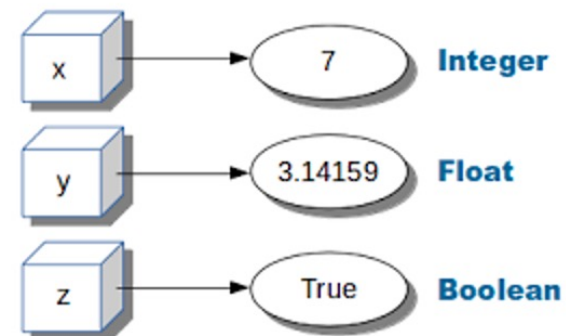
# WHAT IS PYTHON?

Interpreter translates just one statement of the program at a time into machine code. Compiler scans the entire program and translates the whole of it into machine code at once. An interpreter takes very less time to analyze the source code. However, the overall time to execute the process is much slower.

Compiler	Interpreter
<ul style="list-style-type: none"><li>• A compiler takes the entire program in one go.</li></ul>	<ul style="list-style-type: none"><li>• An interpreter takes a single line of code at a time.</li></ul>
<ul style="list-style-type: none"><li>• The compiler generates an intermediate machine code.</li></ul>	<ul style="list-style-type: none"><li>• The interpreter never produces any intermediate machine code.</li></ul>
<ul style="list-style-type: none"><li>• The compiler is best suited for the production environment.</li></ul>	<ul style="list-style-type: none"><li>• An interpreter is best suited for a software development environment.</li></ul>
<ul style="list-style-type: none"><li>• The compiler is used by programming languages such as C, C ++, C #, Scala, Java, etc.</li></ul>	<ul style="list-style-type: none"><li>• An interpreter is used by programming languages such as Python, PHP, Perl, Ruby, etc.</li></ul>

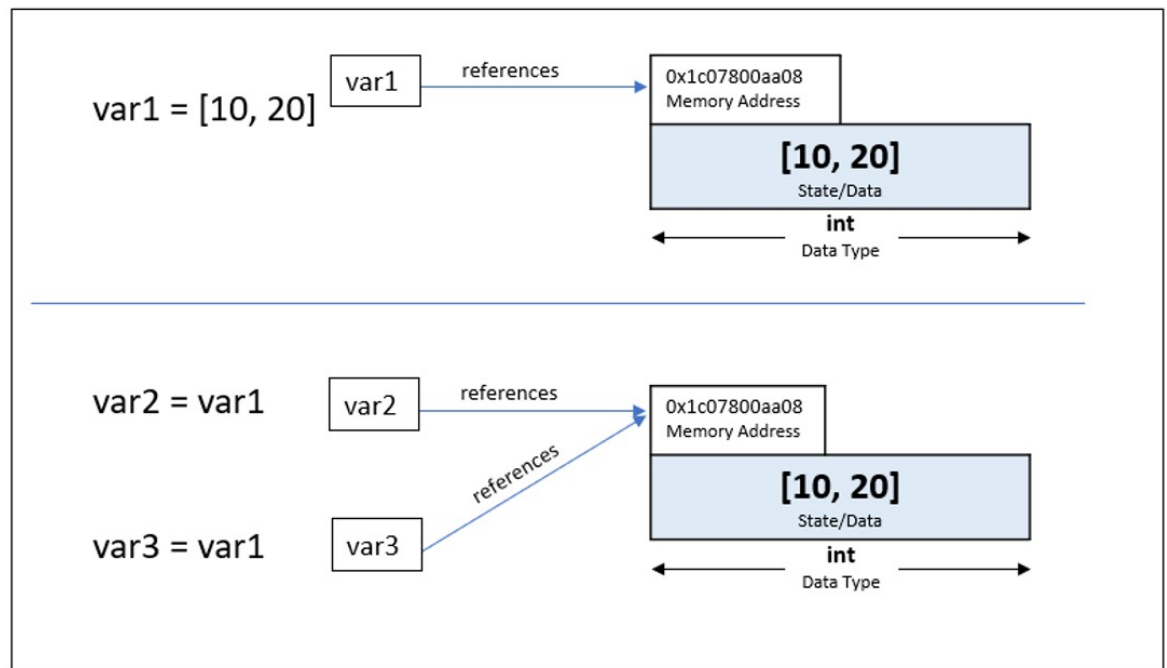
# VARIABLES

- Think of a variable as a box with a label that you can store information
- A variable is a label given to a location in memory containing a value that can be accessed or changed.



# VARIABLES

```
var1 = [10,20]
var2 = var1
var3 = var1
print(var1)
print(var2)
print(var3)
var1.append(30)
print(var1)
print(var2)
print(var3)
var1 = "hello"
print(var1)
print(var2)
print(var3)
```



# INDENTATION

- The first line of python code cannot have an indentation. Indentation is mandatory in python to define the blocks of statements. The number of spaces must be uniform in a block of code. It is preferred to use whitespaces instead of tabs to indent in python.

```
n=int(input("Enter no of Students:"))
Marks=[20,50,60,70,80]
def maxMarks(n,senu):
    max=0
    for x in range(n):
        if senu[max]<senu[x]:
            max=x
    return max
x=maxMarks(n,Marks)
print("Maximum Mark is", Marks[x], "and it's obtained by",Names[x])
```

Block 1

Block 2

Block 3

Block 2 Remaining

Block 1 Remaining

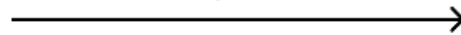


# INDENTATION

■ -> Indicates 1 Space Indentation

Statement 1  
Statement 2  
■ Statement 3  
■ ■ Statement 4  
■ Statement 5  
■ Statement 6  
Statement 7

*How the interpreter visualises*



Code Block 1 begins  
Code Block 1 continues  
    Code Block 2 begins  
        Code Block 3  
    Code Block 2 continues  
    Code Block 2 continues  
Code Block 1 continues

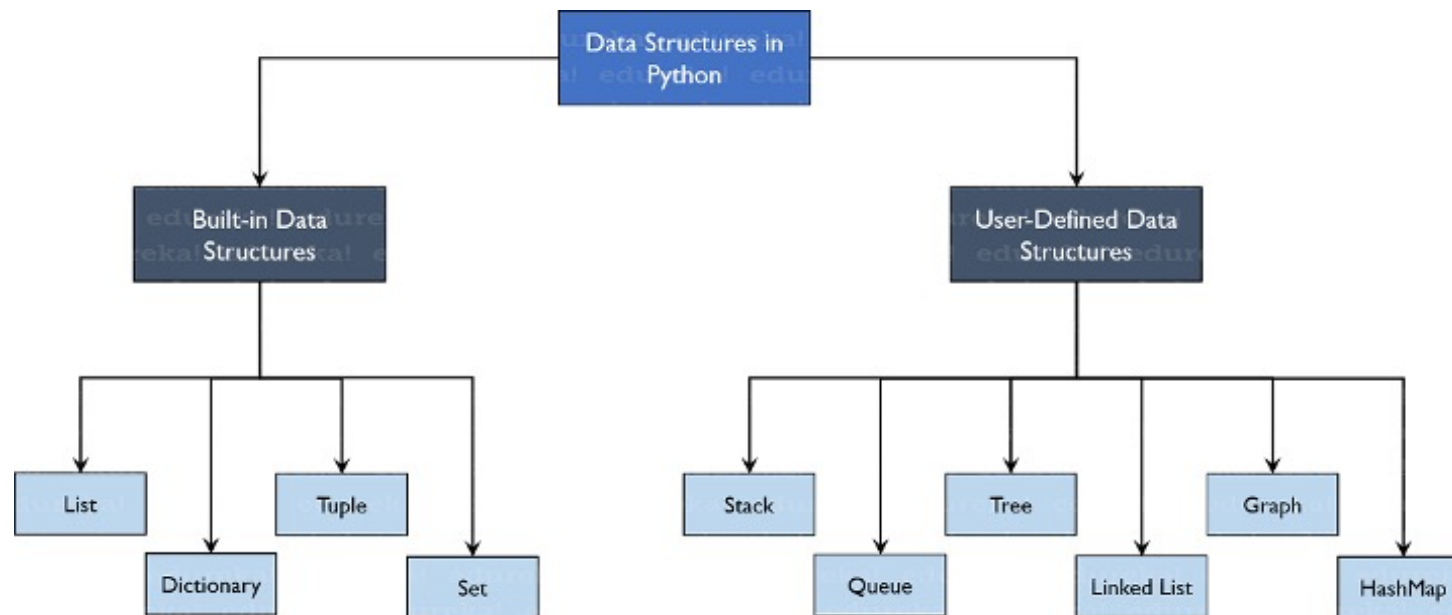
**Here:**

Statements 1, 2, 7 belong to code block 1 as they are at the same distance to the right.  
Statements 3, 5, 6 belong to code block 2  
Statement 4 belongs to code block 3

Execution happens in the same order.

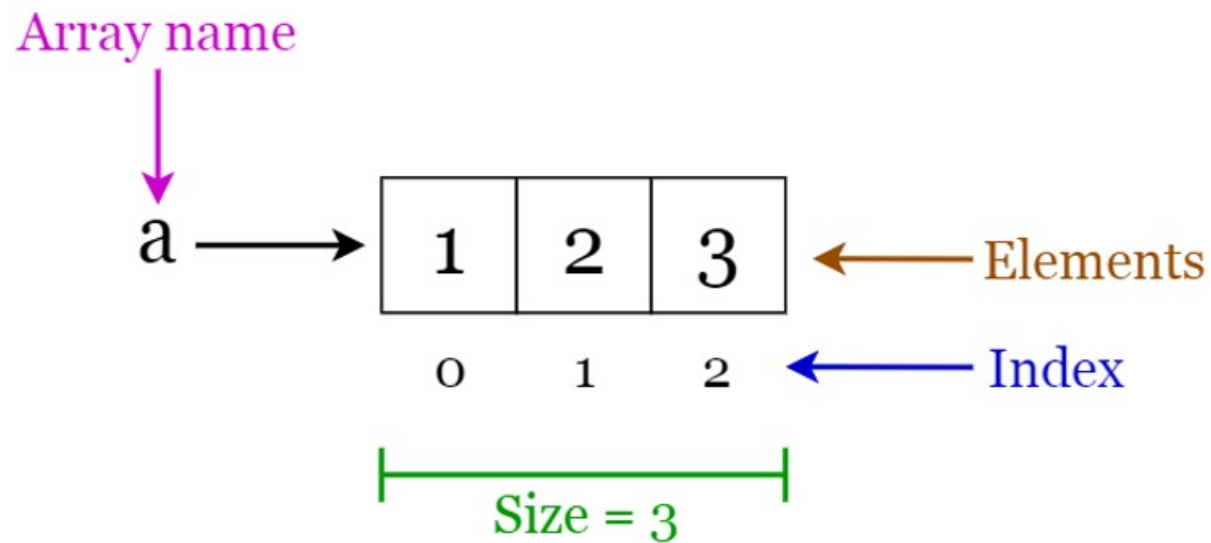
# DATA STRUCTURE

- Data organization, management, and storage format that enables efficient access and modification

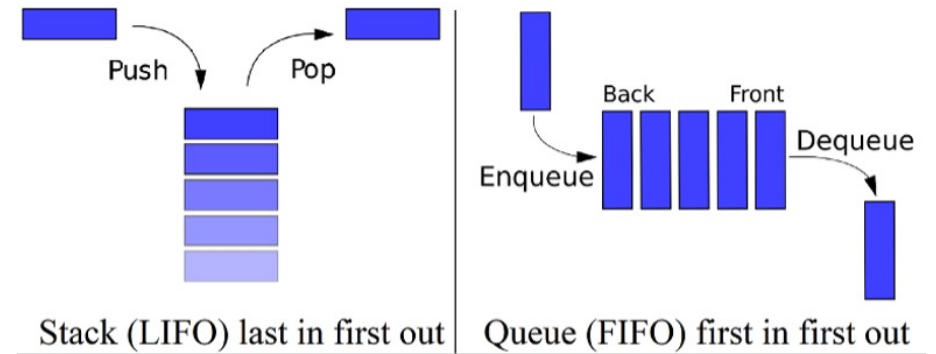
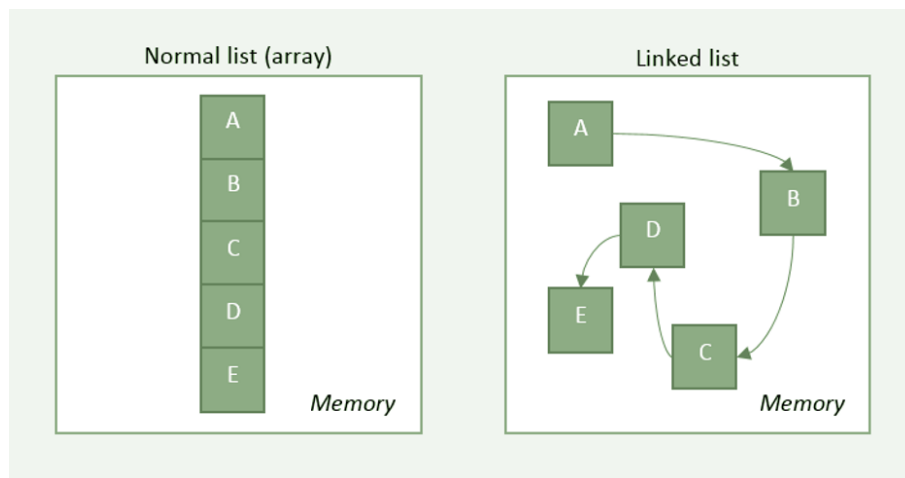


# LIST

- A list can be defined as an abstract data type in which the elements are stored in an ordered manner for easier and efficient retrieval of the elements.

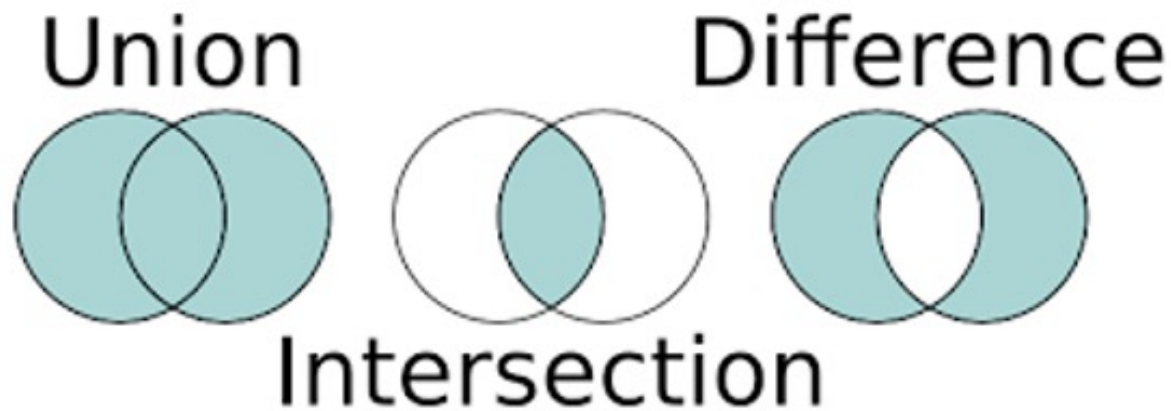


# LIST



# SET

- A set is a data structure that can store any number of unique values in any order you so wish. Set only allow non-repeated, unique values within them.

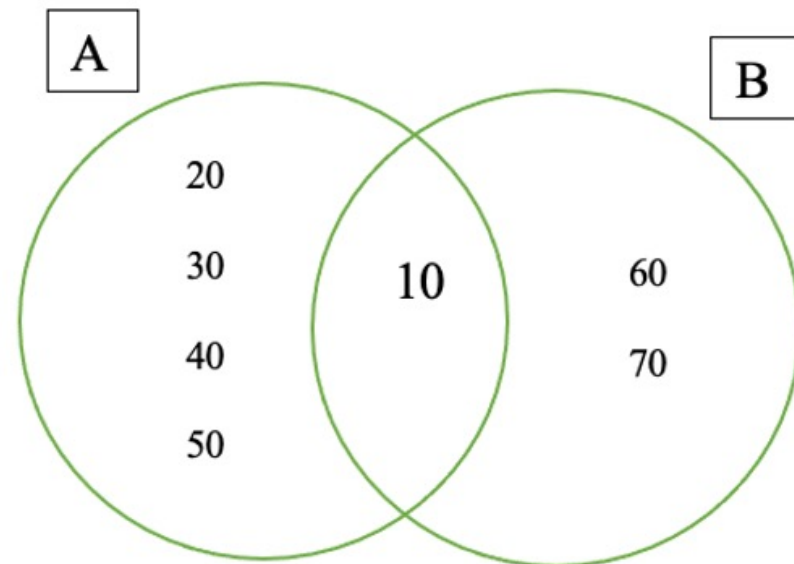


# SET

```
a = set([10,20,30,40,50])  
b = set([10,60,70])  
print(a)  
print(b)
```

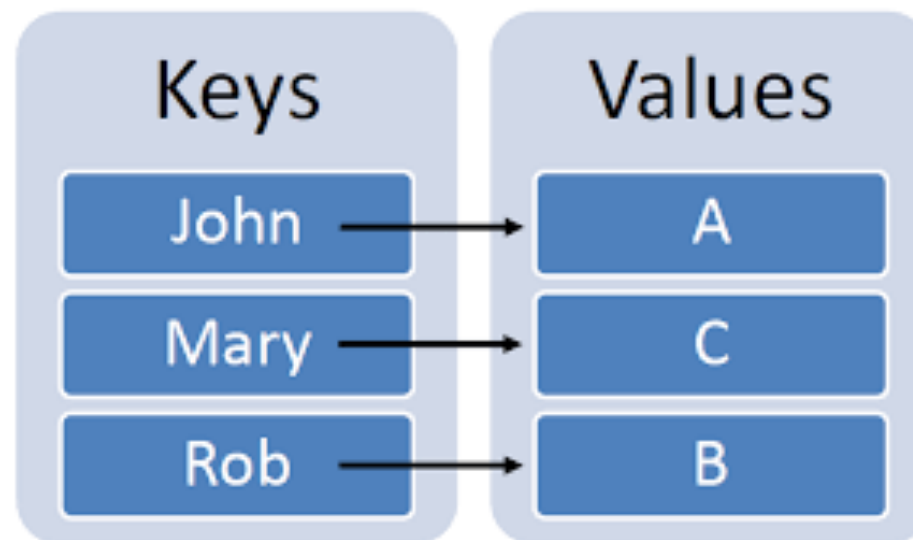
```
print(a.intersection(b))  
print(a.difference(b))  
print(a.union(b))
```

```
print(b.intersection(a))  
print(b.difference(a))  
print(a.union(b))
```

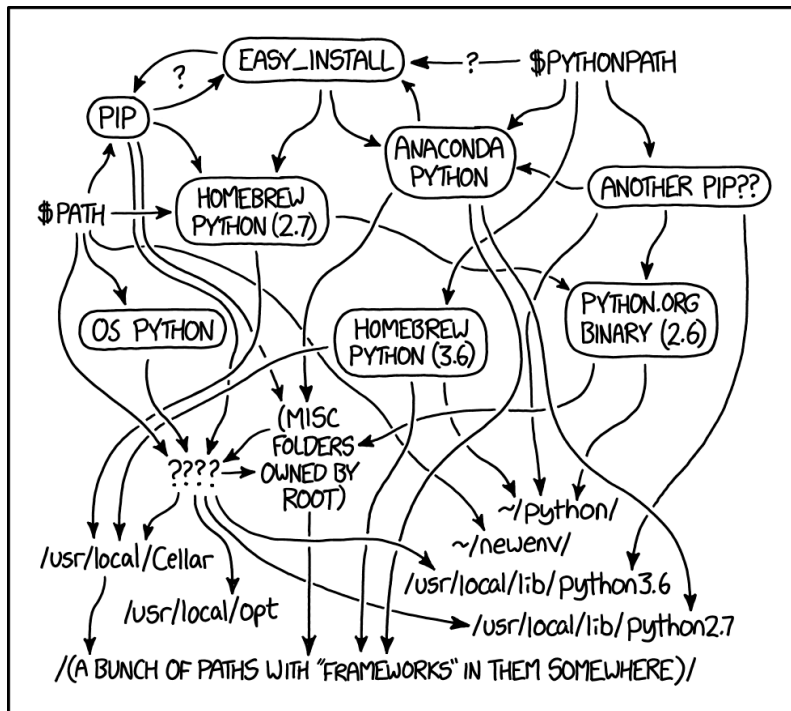


# DICTIONARY

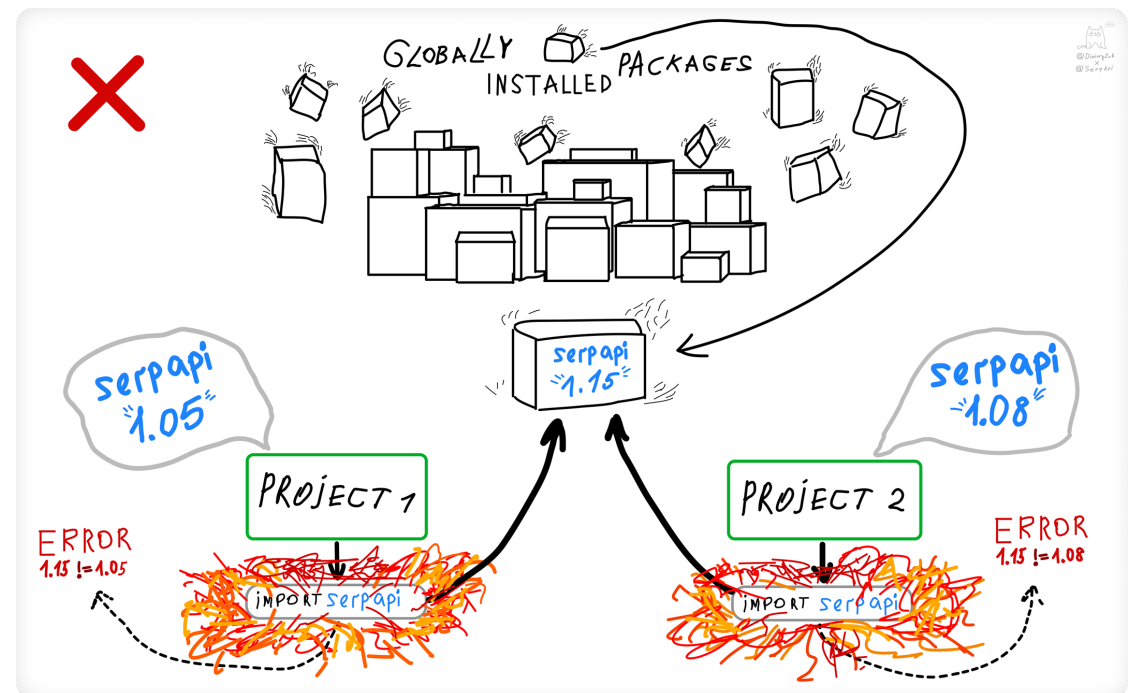
A dictionary is a general-purpose data structure for storing a group of objects. A dictionary has a set of keys and each key has a single associated value. When presented with a key, the dictionary will return the associated value.



# VIRTUAL ENVIRONMENT



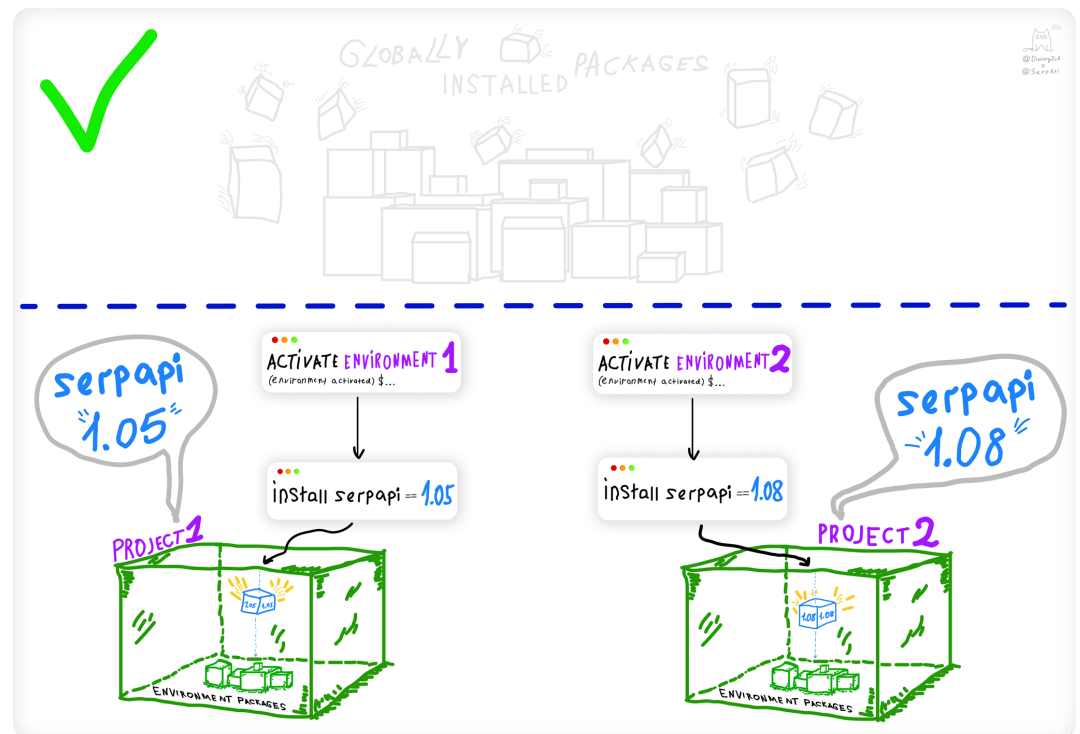
MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



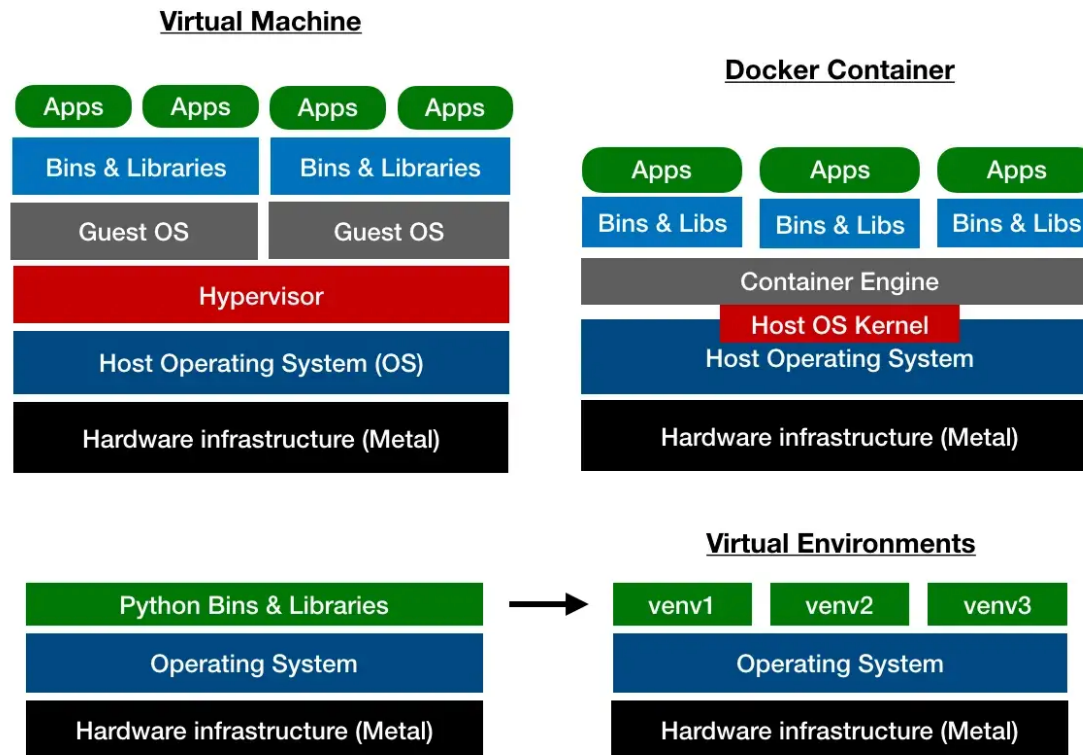


# VIRTUAL ENVIRONMENT

Python virtual environment is basically a separate folder that creates an independent set of installed packages, Python binaries in its own directory, that isolates any other installation of Python.

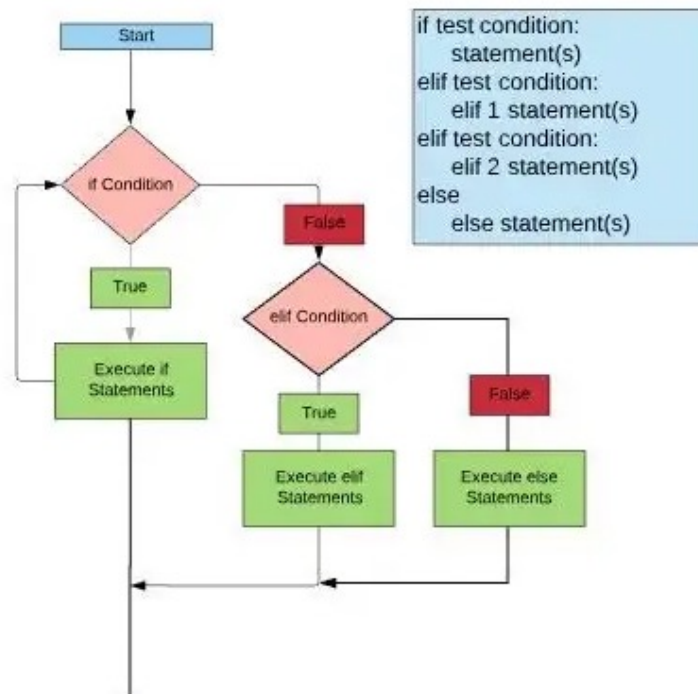


# VIRTUAL ENVIRONMENT



# FLOW CONTROL

- Python IF, IF ELSE, ELIF & Nested IF Statements



# FLOW CONTROL

Logical Operators: and, or, not

X	Y	X <b>and</b> Y
False	False	False
False	True	False
True	False	False
True	True	True

X	Y	X <b>or</b> Y
False	False	False
False	True	True
True	False	True
True	True	True

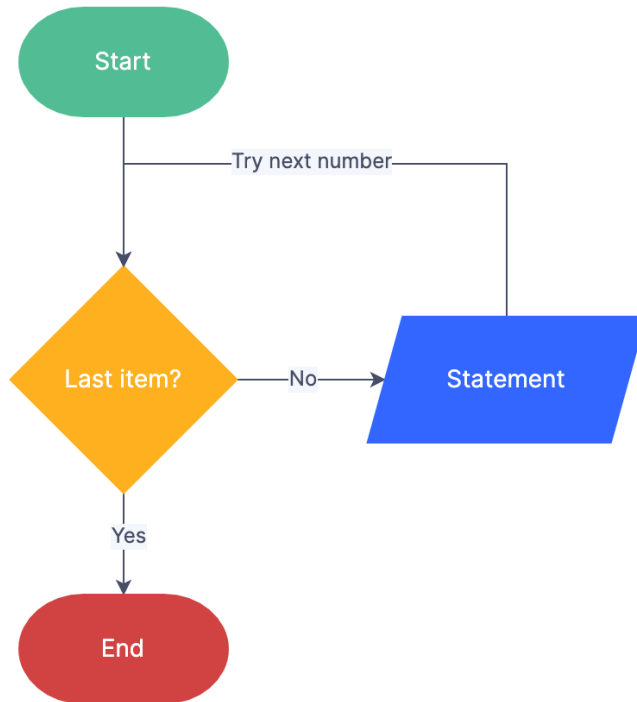
X	<b>Not</b> X
True	False
False	True

Comparison Operators

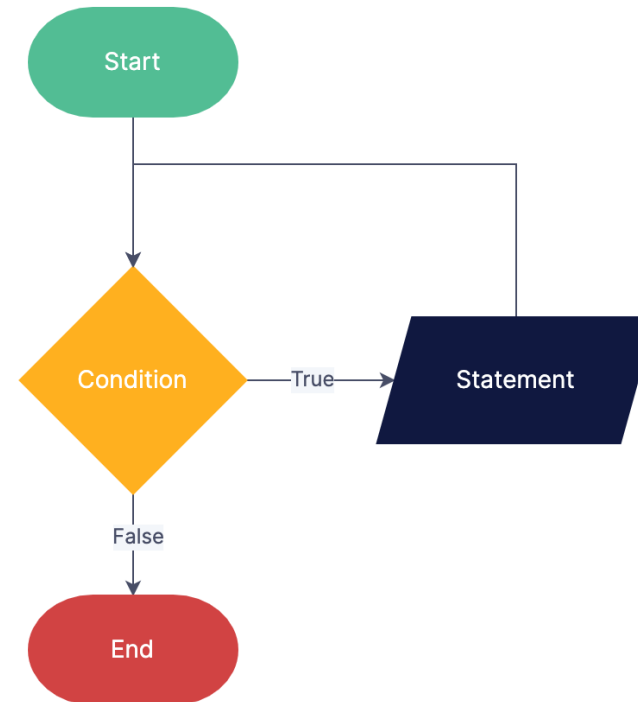
Operator	Name	Example
==	Equal	a == b
!=	Not equal	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b

# FLOW CONTROL

## For Loop Flowchart



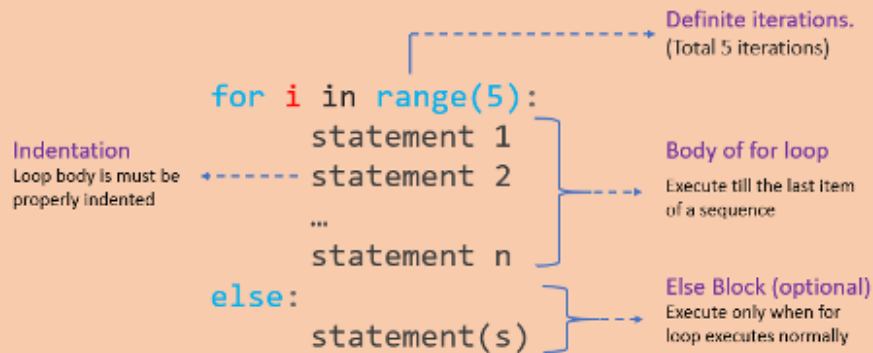
## While Loop Flowchart



# FLOW CONTROL

## Python for loop

A for loop is **used for iterating over a sequence and iterables** (like range, list, a tuple, a dictionary, a set, or a string).

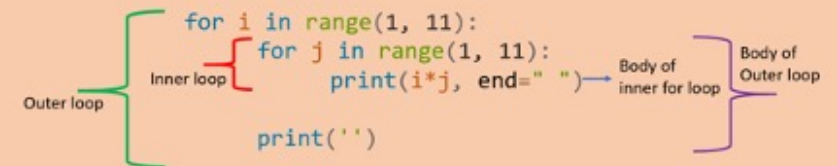


PYnative.com

## Python Nested Loop

A Loop inside a loop is known as a nested loop.

In the nested loop, the number of iterations will be equal to the number of iterations in the outer loop multiplied by the iterations in the inner loop.



PYnative.com

# FLOW CONTROL

