Introduction to Python Programming

MIRAN BABAN

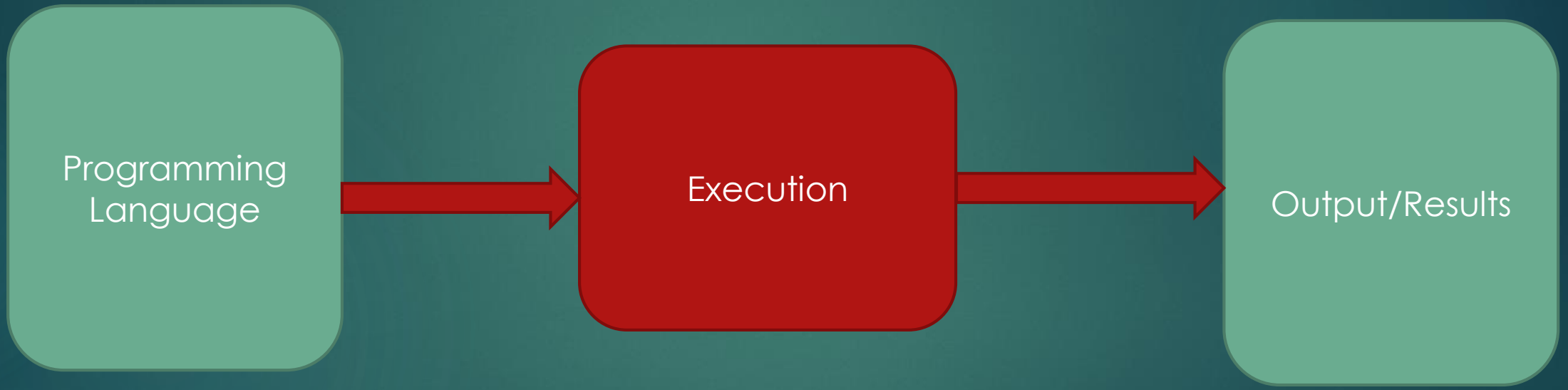
KOMAR UNIVERSITY

What is computer

- Electronic device
- Takes input and produce output
- The input go to the set of process to generate output

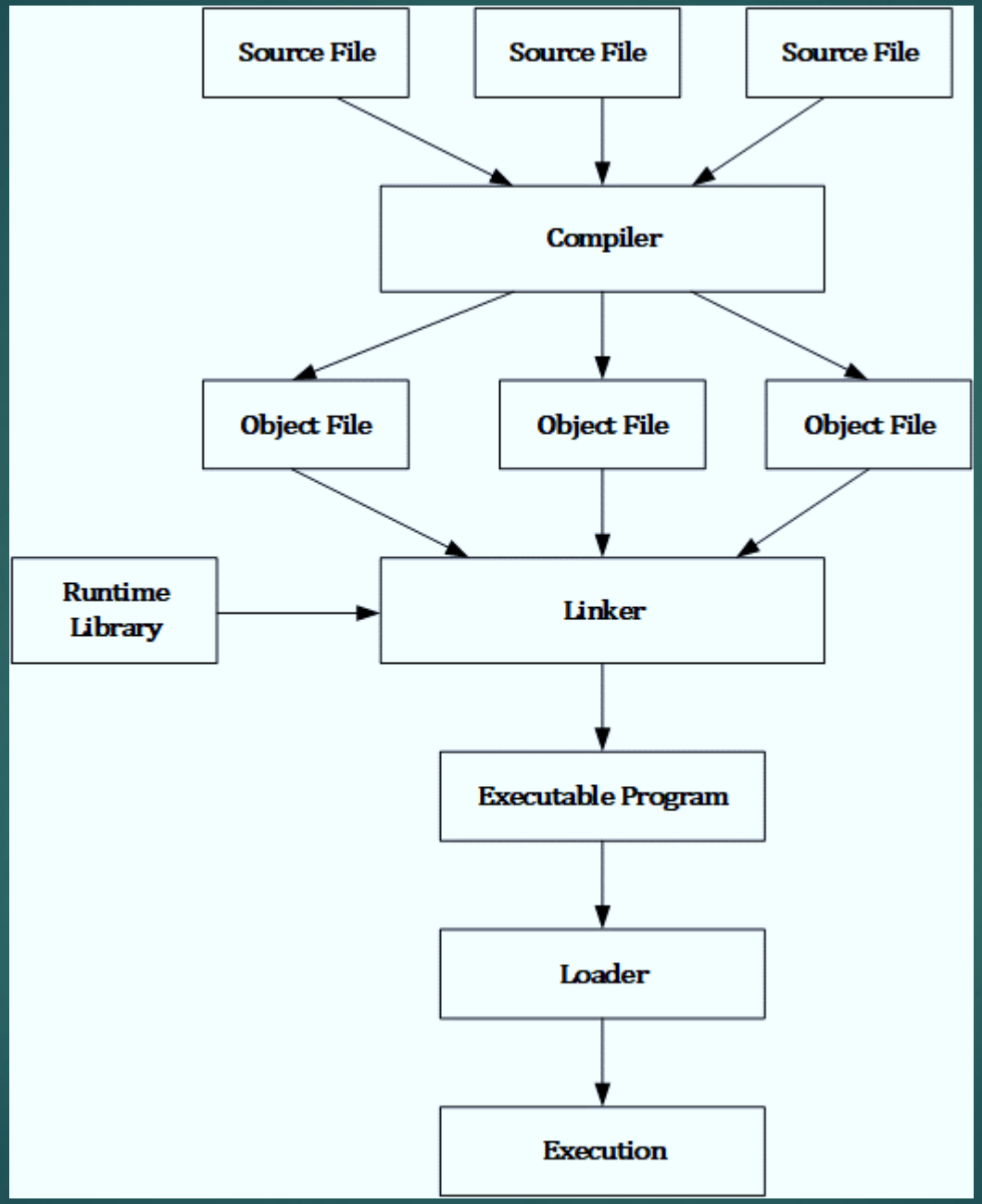
What is execution program

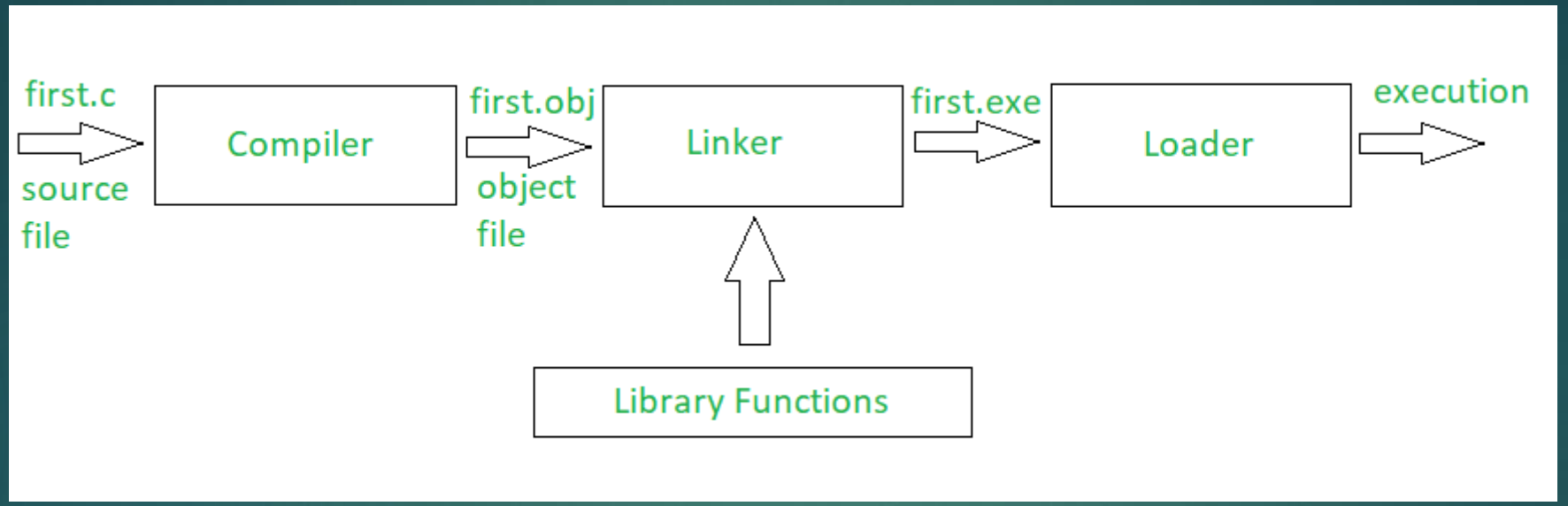
- In most cases, coding is done in either a high-level or a low-level language (assembly language)
- These languages must be translated into machine language for the computer to understand them.
- A compiler/interpreter (for high-level languages) or an assembler does the translation (for assembly language program).
- The resulting machine language code can be saved and executed immediately or later.


Programming Language
Execution
Output/Results

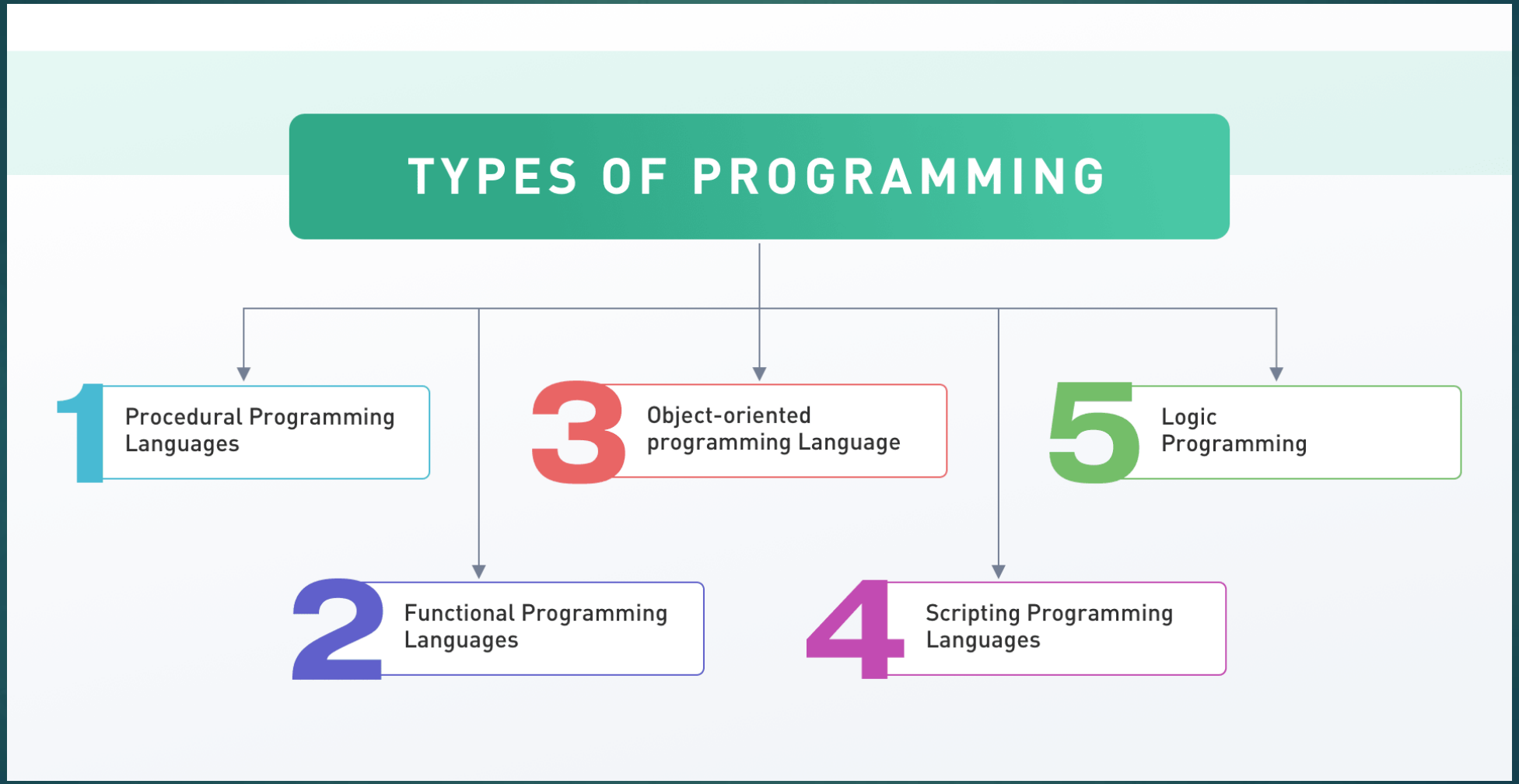
Execution process

- Before becoming an executable program, source code must go through numerous phases.
- The source code is verified for syntax issues in the first step.
- After the syntax mistakes have been identified, the source file is run through a compiler, which converts the high-level language into object code (A machine code not ready to be executed).
- The object code is then linked with pre-compiled library functions by a linker, resulting in an executable program.
- After that, the executable program is loaded into memory and executed.


Source File
Source File
Source File
Compiler
Object File
Object File
Object File
Runtime Library
Linker
Executable Program
Loader
Execution

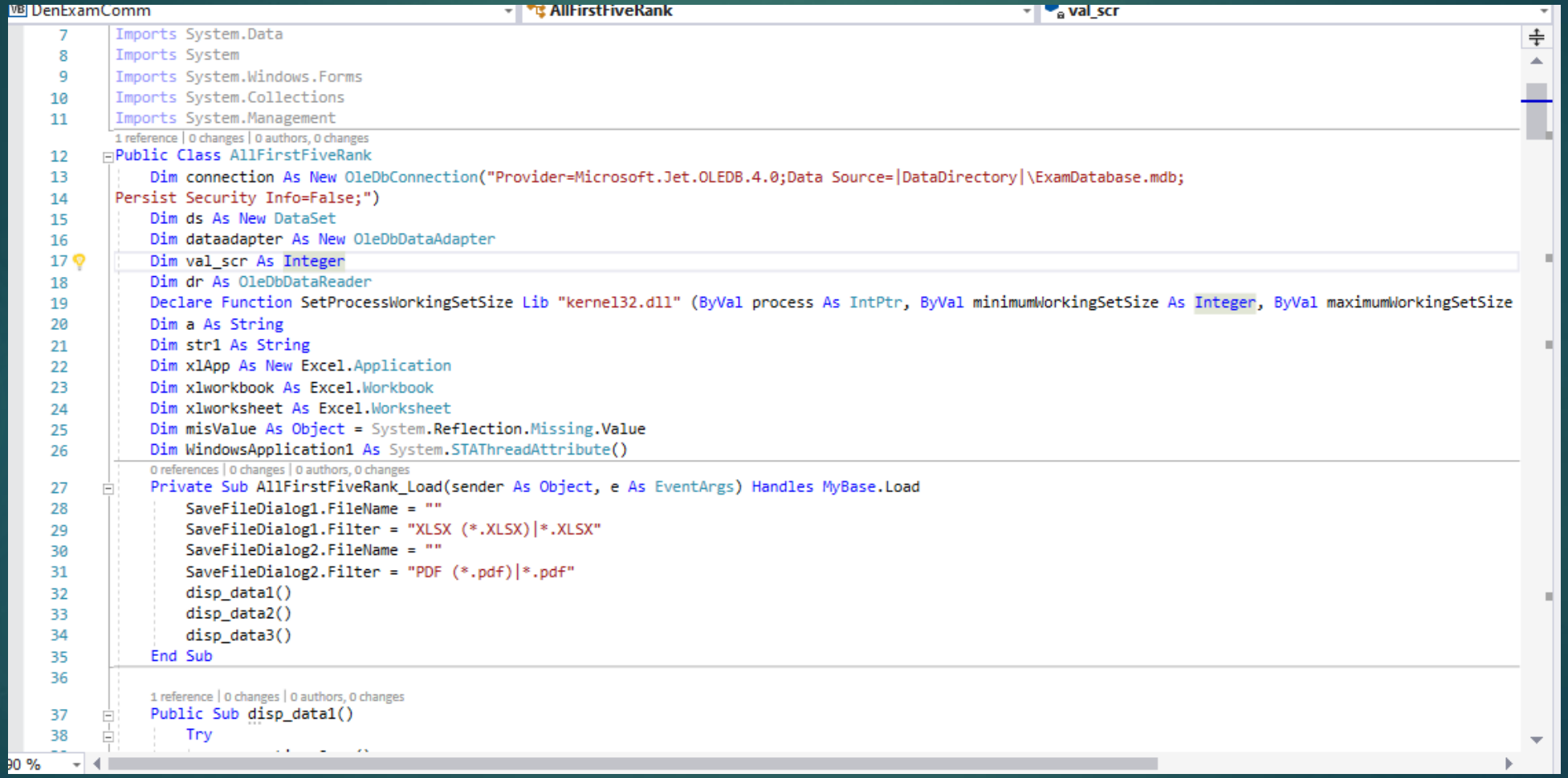

first.c
source
file
Compiler
first.obj
object
file
Linker
first.exe
Loader
execution
Library Functions

Types of Languages

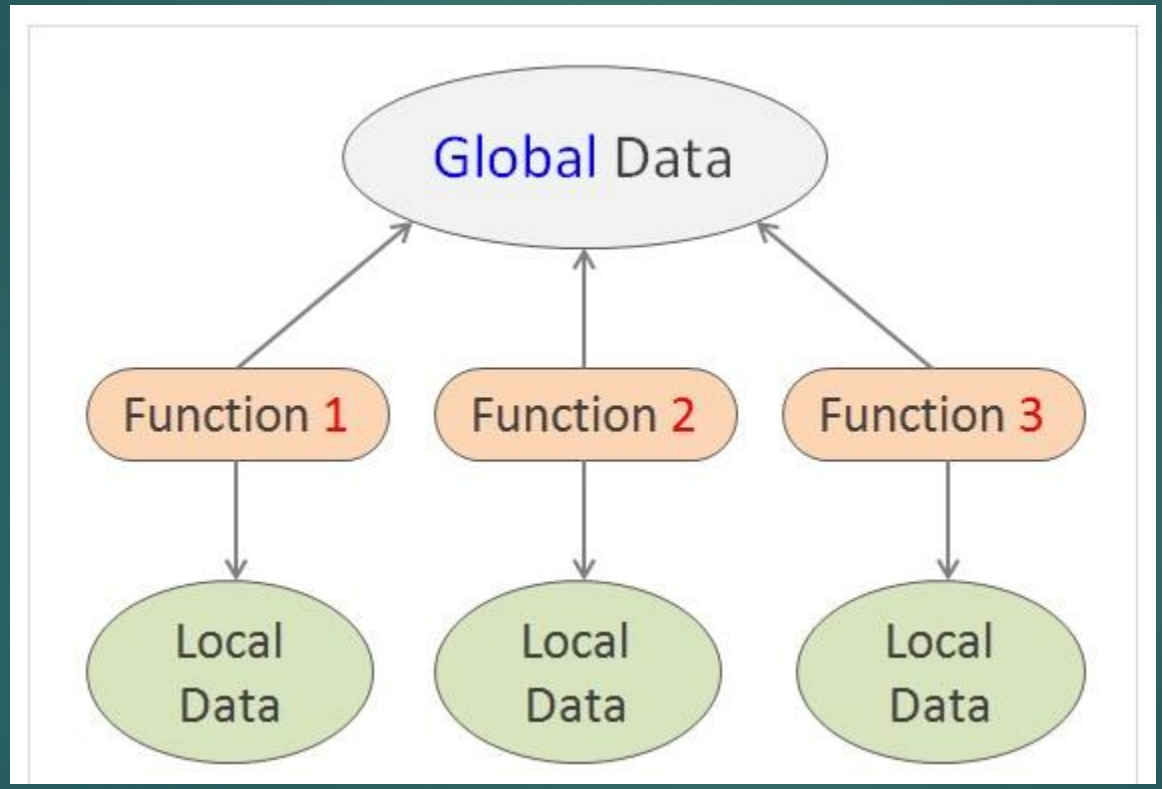


Procedural Programming

- Procedural Programming is likely to be a new developer's first programming paradigm. Procedural code, in its most basic form, is the code that tells a device how to do a task in logical steps.
- This paradigm takes a top-down, linear approach to data and methods, and sees them as two distinct things.
- Procedural Programming separates a program into procedures, which are sometimes known as routines or functions and simply comprise a set of actions to be carried out, based on the concept of a procedure call.


DenExamComm
AllFirstFiveRank
val_scr
Imports System.Data
Imports System
Imports System.Windows.Forms
Imports System.Collections
Imports System.Management
1 reference | 0 changes | 0 authors, 0 changes
Public Class AllFirstFiveRank
Dim connection As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirectory|\ExamDatabase.mdb;
Persist Security Info=False;")
Dim ds As New DataSet
Dim dataadapter As New OleDbDataAdapter
Dim val_scr As Integer
Dim dr As OleDbDataReader
Declare Function SetProcessWorkingSetSize Lib "kernel32.dll" (ByVal process As IntPtr, ByVal minimumWorkingSetSize As Integer, ByVal maximumWorkingSetSize
Dim a As String
Dim str1 As String
Dim xlApp As New Excel.Application
Dim xlworkbook As Excel.Workbook
Dim xlworksheet As Excel.Worksheet
Dim misValue As Object = System.Reflection.Missing.Value
Dim WindowsApplication1 As System.STAThreadAttribute()
0 references | 0 changes | 0 authors, 0 changes
Private Sub AllFirstFiveRank_Load(sender As Object, e As EventArgs) Handles MyBase.Load
SaveFileDialog1.FileName = ""
SaveFileDialog1.Filter = "XLSX (*.XLSX)|*.XLSX"
SaveFileDialog2.FileName = ""
SaveFileDialog2.Filter = "PDF (*.pdf)|*.pdf"
disp_data1()
disp_data2()
disp_data3()
End Sub
1 reference | 0 changes | 0 authors, 0 changes
Public Sub disp_data1()
Try
90 %

Procedural Programming

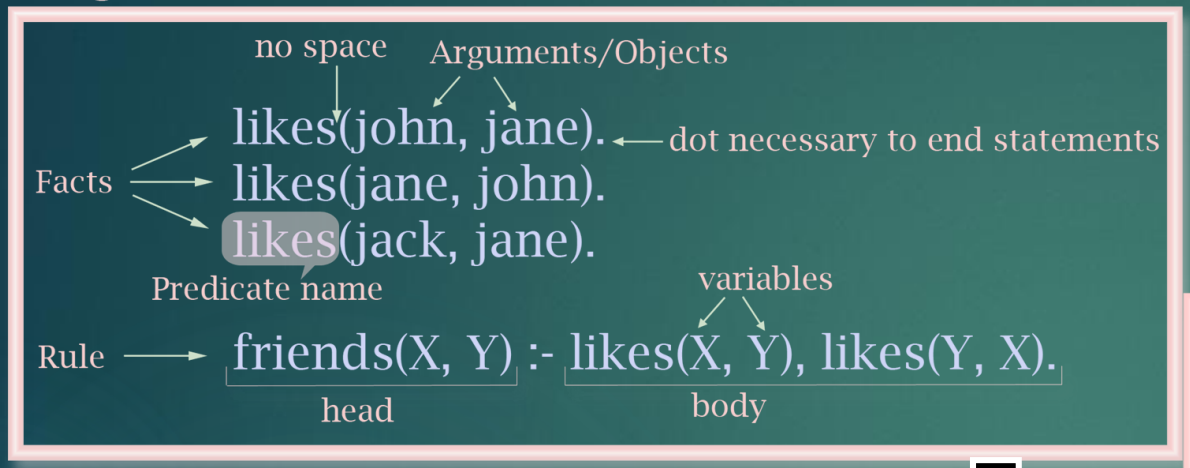


Logic Programming

- Formal logic underpins much of the programming paradigm.
- The language does not direct the machine what to do, but it does place limitations on what it can accomplish.
- The major logic programming languages include PROLOG, ASAP(Answer Set Programming), and Datalog, in which rules are represented as classes.

Example of Logic Programming

Program Window



Query Window

?- likes(john, jane). ← dot necessary

true. ← answer from prolog interpreter

sign on prolog query prompt

variables

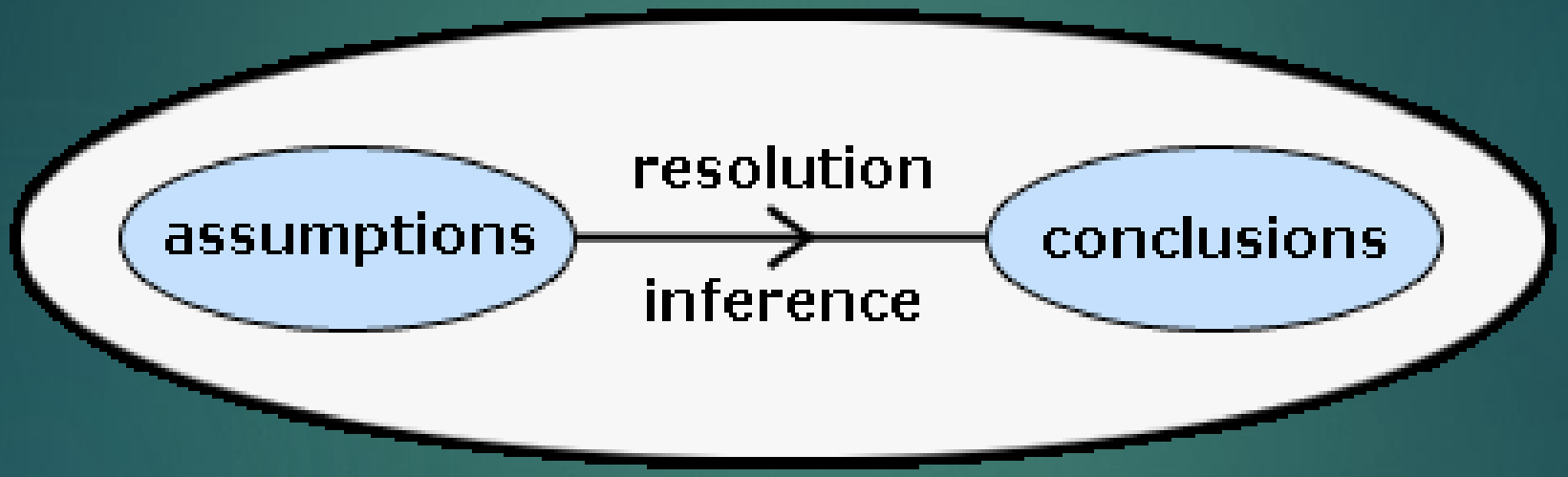
?- friends(X, Y).

X = john,

Y = jane ; ← type ; to get next solution

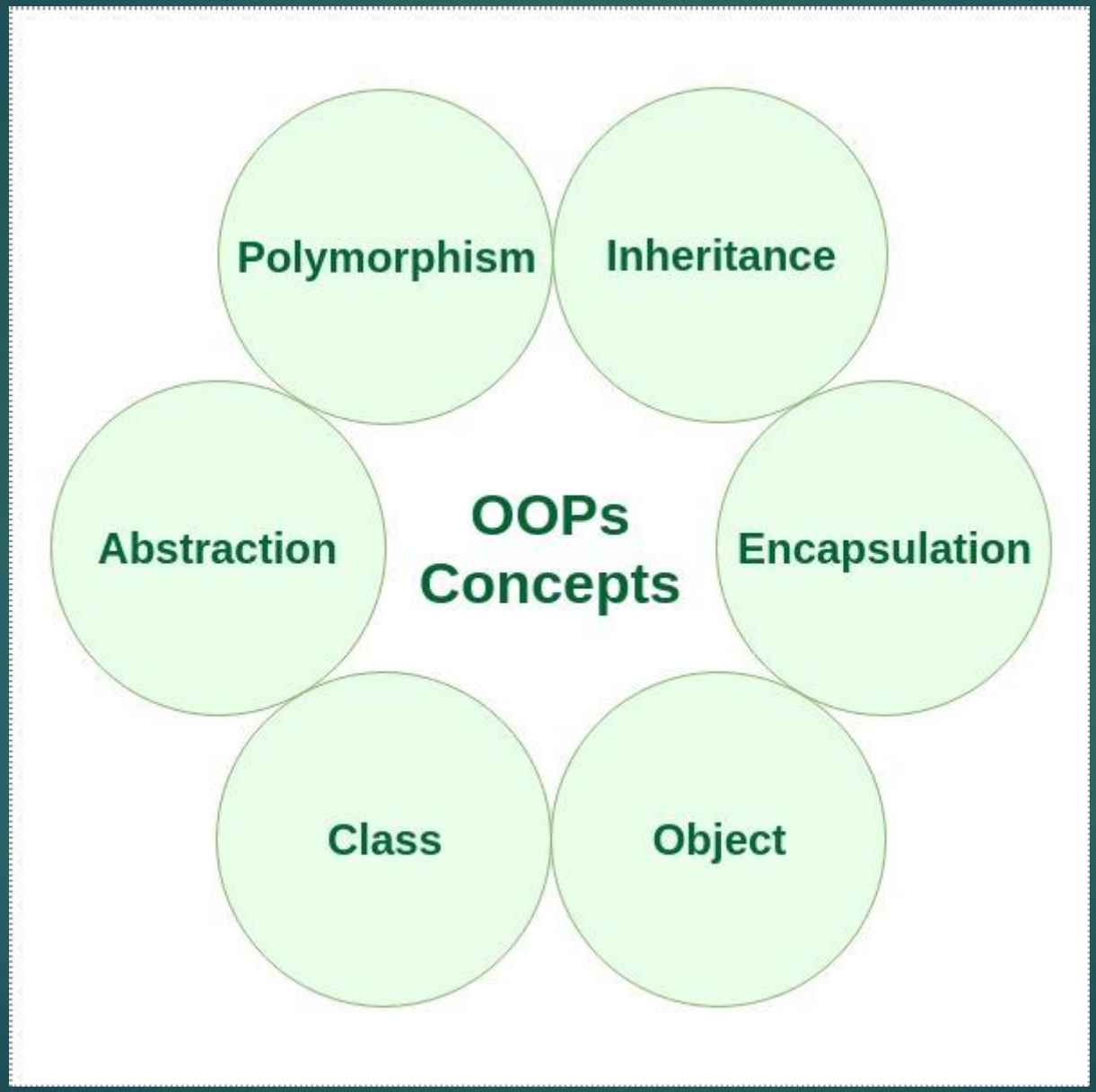
X = jane,

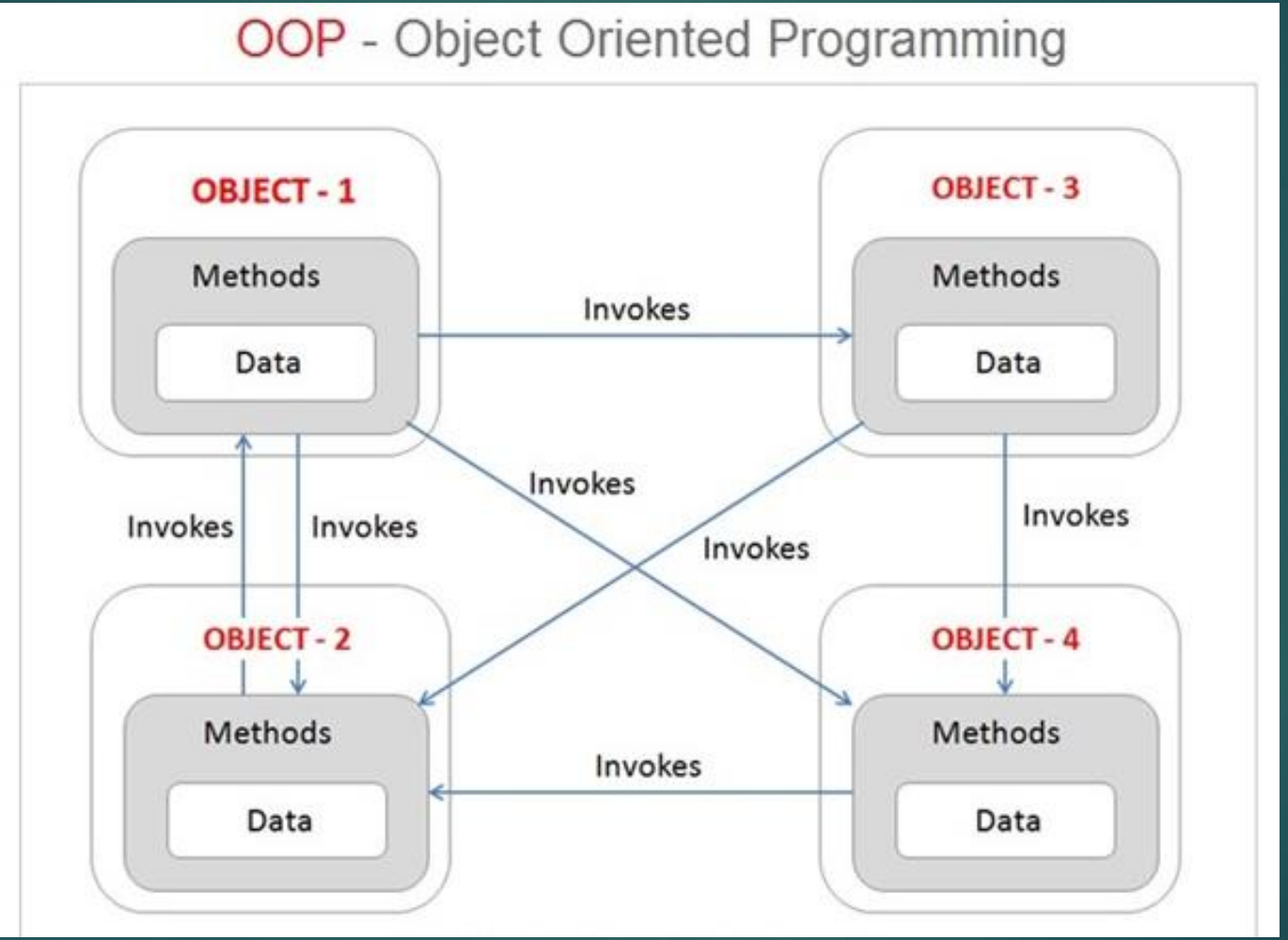
Y = john.


assumptions
resolution
inference
conclusions

OOP Programming

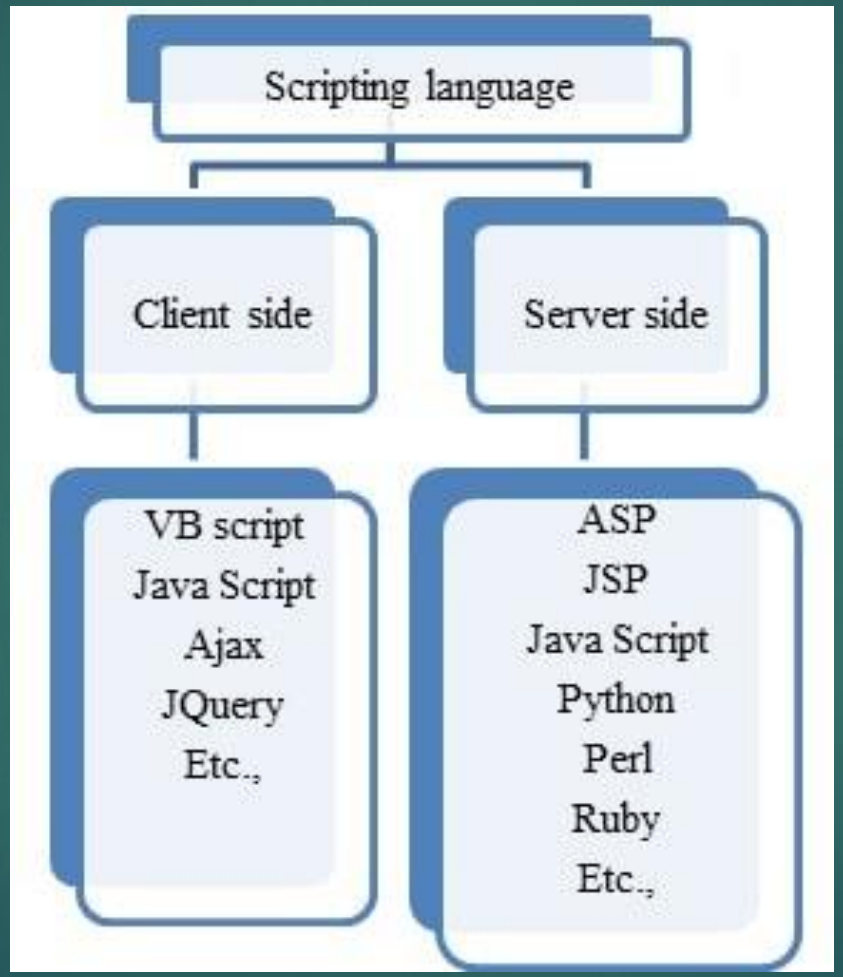
- OOP is a programming style that understands life as a collection of objects that work together to solve a specific problem.
- The most important aspect of OOP is encapsulation, which is the concept that each object that holds the program is self-sustaining, meaning that all of the components that make up the object are contained within it.
- Now that each module in this paradigm is self-sustaining, objects from one program can be taken and used to address another problem with little or no changes.


Polymorphism
Inheritance
Abstraction
OOPs Concepts
Encapsulation
Class
Object


OOP - Object Oriented Programming
OBJECT - 1
Methods
Data
OBJECT - 3
Methods
Data
Invokes
Invokes
Invokes
Invokes
Invokes
Invokes
OBJECT - 2
Methods
Data
OBJECT - 4
Methods
Data
Invokes

Scripting Languages

- Scripting language (also known as scripting or script) is a set of commands that may be run without having to compile them.
- Not all programming languages are scripting languages, and not all scripting languages are programming languages.
- Scripting languages include PHP, Perl, and Python, to name a few.


Scripting language
Client side
Server side
VB script
Java Script
Ajax
JQuery
Etc.,
ASP
JSP
Java Script
Python
Perl
Ruby
Etc.,

```
1  <html>
2      <head>
3          <title>Page Title</title>
4      </head>
5      <body>
6          <?php
7              echo "Hello World";
8          ?>
9      </body>
10 </html>
```

What is Python??

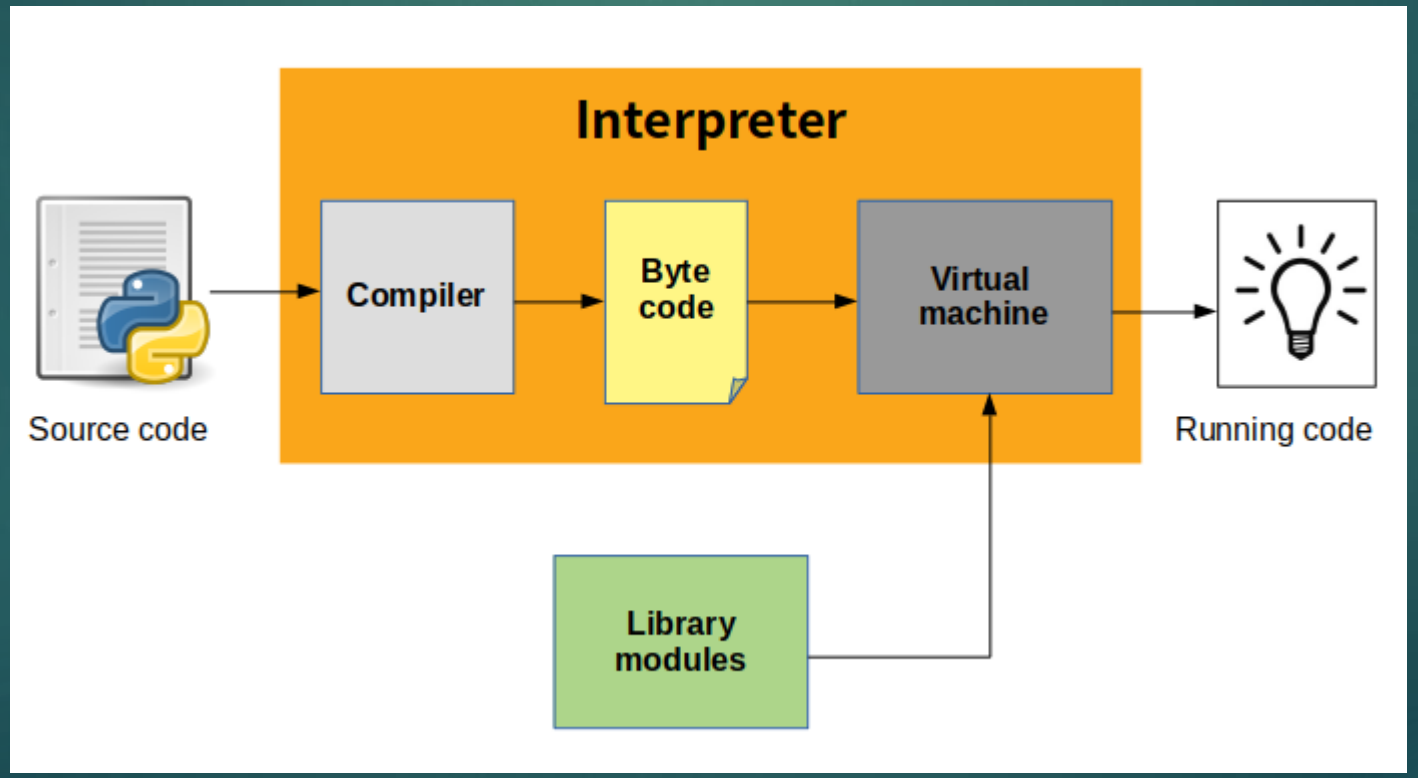
What is Python

- Python is a dynamically semantic, interpreted, object-oriented high-level programming language.
- Python is a multifunctional programming language that may be used almost everywhere data, mathematical computation, or lines of code are used.
- This means that, unlike Java, Python isn't limited to few limitation purposes.

Guido van Rossum

What is Python

- designed by Guido van Rossum in 1991 and developed by Python Software Foundation
- NASA, Google, Netflix, Spotify,



Usages of Python

- AI and Machine Learning
- Data Analytics
- Data Visualization
- Programming Applications
- Web Development
- Game Development

Advantages

- Python is easy to learn
- Improved Productivity
- Interpreted Language
- Dynamically Typed
- Open-Source
- Libraries Support
- Portability

Disadvantages

- Slow Execution Time
- Memory Efficient
- Runtime Errors
- Weak in Mobile Computing

Application of Using Python

- Web Application
- Game Development
- Machine Learning and Artificial Intelligence
- Data Science and Data Visualization
- Desktop GUI
- Web Scraping Applications
- Business Applications
- Audio and Video Applications
- Embedded Applications

```java
class PrintNumber {
	int left;
	int right;

	PrintNumber(int left, int right) {
	    this.left = left;
	    this.right = right;
	}

	public int getleft() {
	    return left;
	}
	public int getRight() {
	    return right;
	}
}

public class Print5 {

	public static void main(String[] args) {
	    PrintNumber printNumber = new PrintNumber (3,2);
	    String sum = Integer.toString(printNumber.getleft()
	          + printNumber.getRight() );
	    System.out.println("3+2=" + sum);
	}
}
```

```python
class Number:
  def __init__(self, left, right):
      self.left = left
      self.right = right

number = Number(3, 2)

print("3+2=", number.left + number.right)
```

Variables

- Variables are just reserved memory places where values can be stored. This means that when you make a variable, you set aside some memory for it.
- The interpreter allocates memory and decides what can be placed in the reserved memory based on the data type of a variable.
- As a result, you can store integers, decimals, or characters in variables by assigning multiple data types to them.

```
# This is integer Value
IntValue = 56

# This floating point
FloatMiles = 1500.67

# This A string value
StringName = "PythonProgram"

# Printing Values

print(IntValue)
print(FloatMiles)
print(StringName)
```

Variables

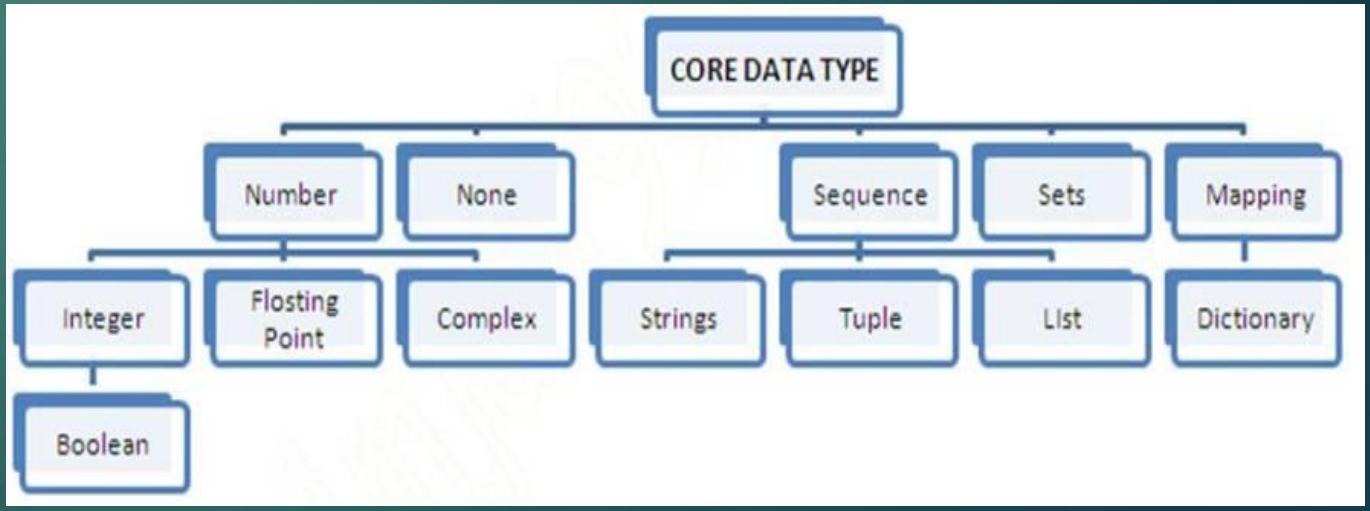
- Different way to set values to variables
- a=b=c=100
- a,b,c=1, 2.115, ‘Dimensions’
- a=1
  b=1.5
  c=‘Computer’

Data Types

- In the Python Numbers category, we can find complex numbers, floating point numbers, and integers.
- In Python, complex numbers are defined as a complex class, floating point numbers as a float, and integers as an int.

Data Types

- There are five different standard data types:
- Numbers
- String
- List
- Tuple
- Dictionary

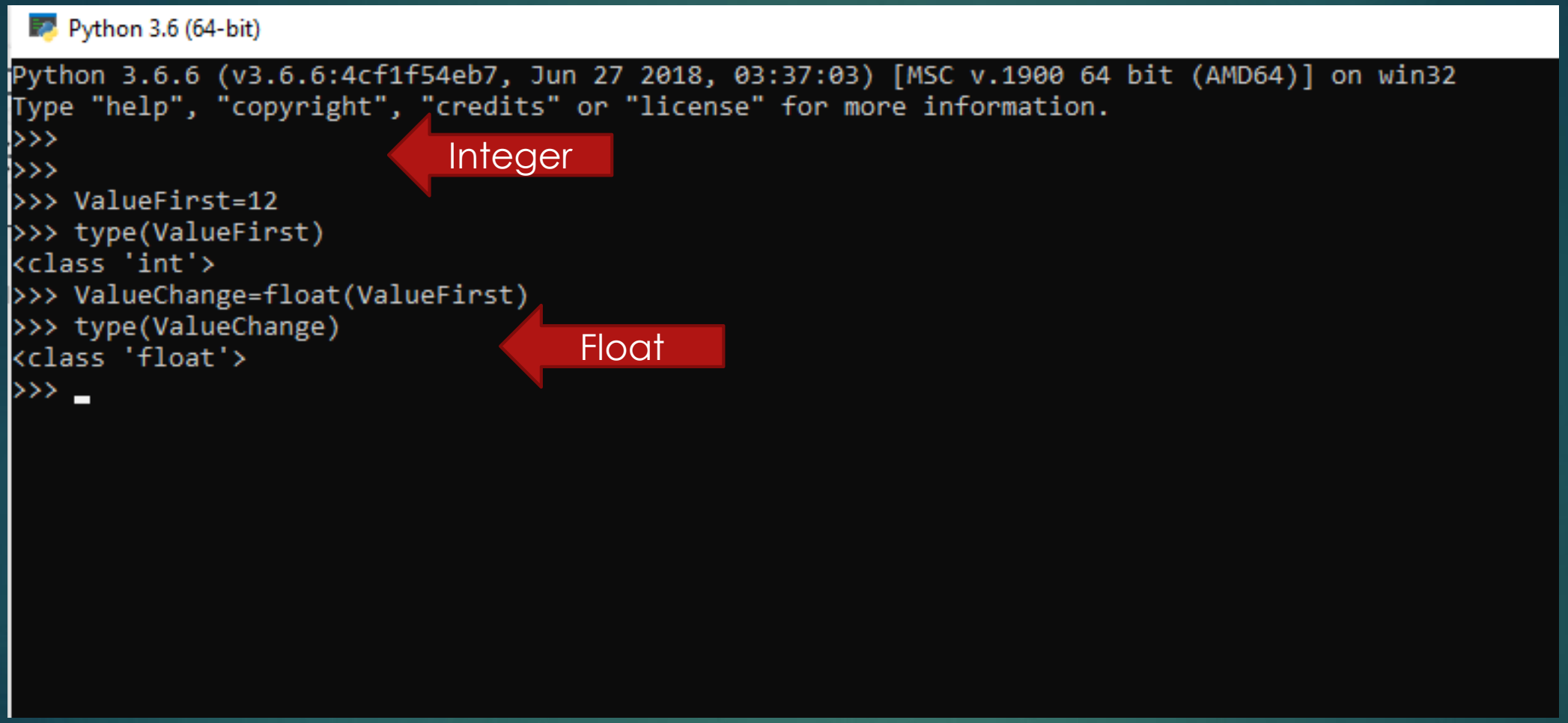


Casting in Python

- The conversion of an item from one data type to another is known as type conversion.
- The Python interpreter does implicit type conversion automatically.
- Python prevents data loss during Implicit Type Conversion.
- Explicit Type Conversion, also known as Type Casting, is when the user converts the data types of objects using predefined functions.
- As we coerce the object to a certain data type with Type Casting, data loss is possible.

```
Python 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> IntValue=12
>>> FloatValue=5.9
>>> StringValue="Computer"
>>> BoolValue=True
```

```
>>> type(IntValue)
<class 'int'>
>>> type(FloatValue)
<class 'float'>
>>> type(StringValue)
<class 'str'>
>>> type(BoolValue)
<class 'bool'>
>>>
```


Python 3.6 (64-bit)
Python 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> ValueFirst=12
>>> type(ValueFirst)
<class 'int'>
>>> ValueChange=float(ValueFirst)
>>> type(ValueChange)
<class 'float'>
>>>
Integer
Float

Comparison operators

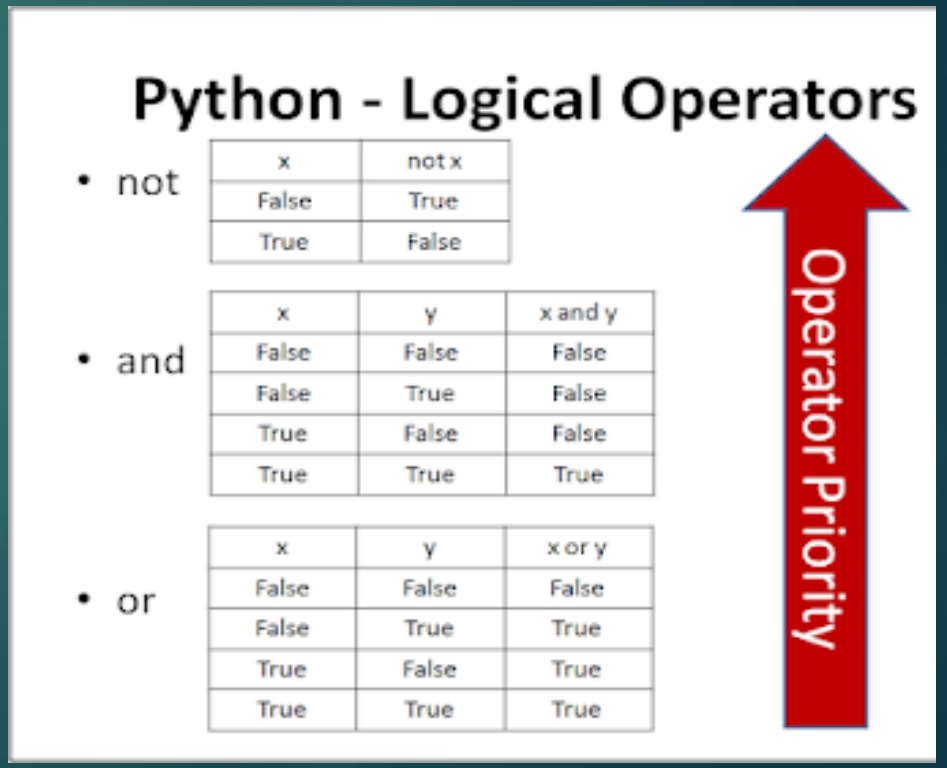
- These operations compare the values on both sides and determine their relationship. Relational operators are another name for them.

Operator types	Operator Meaning
>	Greater than
<	Less than
>=	Greater or equal
<=	Less or Equal
==	Equal
!=	Not Equal

Logical Operator

- Operation on values and variables is done through operators. These are the special symbols that allow arithmetic and logical computations to be performed. Operand is the value on which the operator operates.

- AND
- OR
- Not



Arithmetic Operations

- Mathematical operations such as addition, subtraction, multiplication, and division are performed using arithmetic operators.

Arithmetic Operators

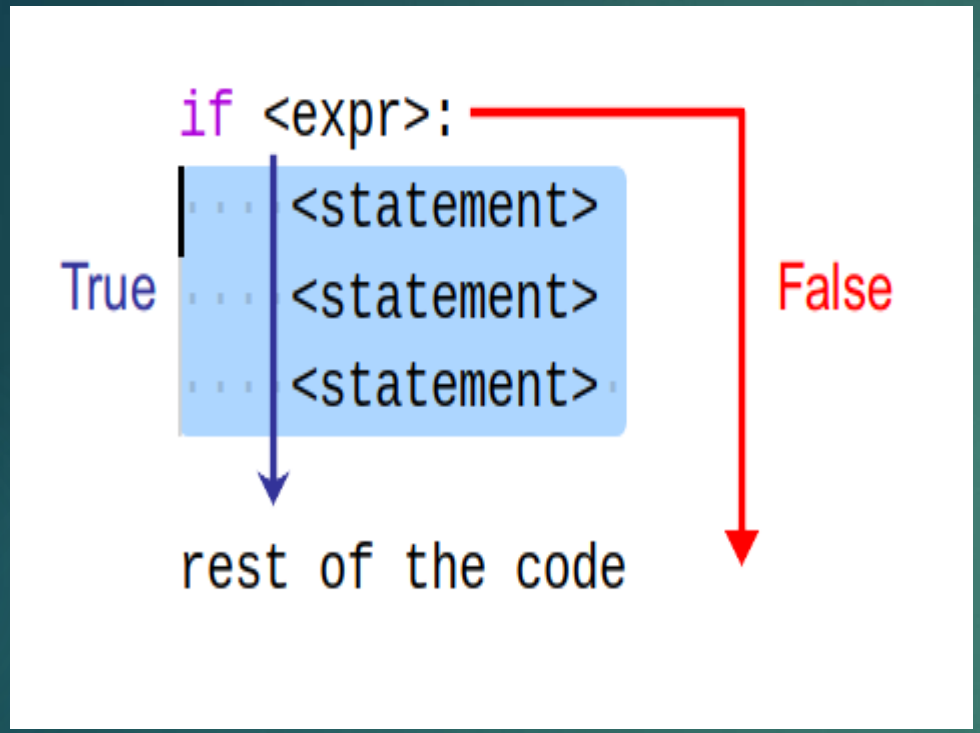
Operator	Meaning	Example
+	Addition	4 + 7 ⟶ 11
-	Subtraction	12 - 5 ⟶ 7
*	Multiplication	6 * 6 ⟶ 36
/	Division	30 / 5 ⟶ 6
%	Modulus	10 % 4 ⟶ 2
//	Quotient	18 // 5 ⟶ 3
**	Exponent	3 ** 5 ⟶ 243

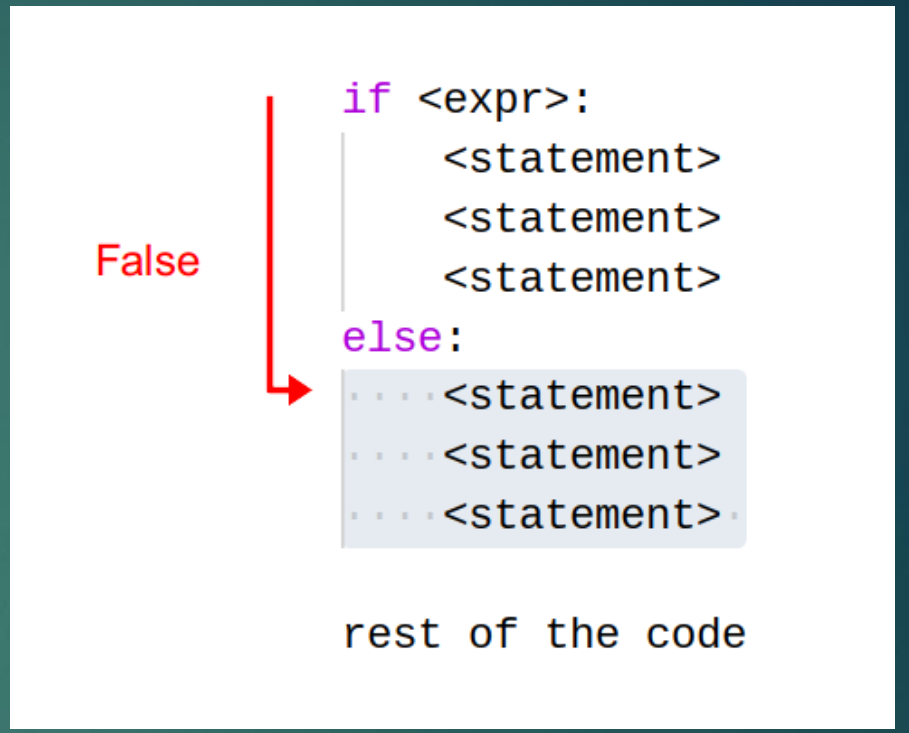
Python 3.6 (64-bit)

```
Python 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> # Addition
... 1+1+7
9
>>> # Subtract
... 1-8
-7
>>> #Multiplication
... 1*8*9
72
>>> #module
... 6%2
0
>>> # Exponent
... 2**2
4
>>> # Quotient
... 16//2
8
>>> # Quotient
... 16//3
5
>>> # Division
... 3/2
1.5
>>>
```

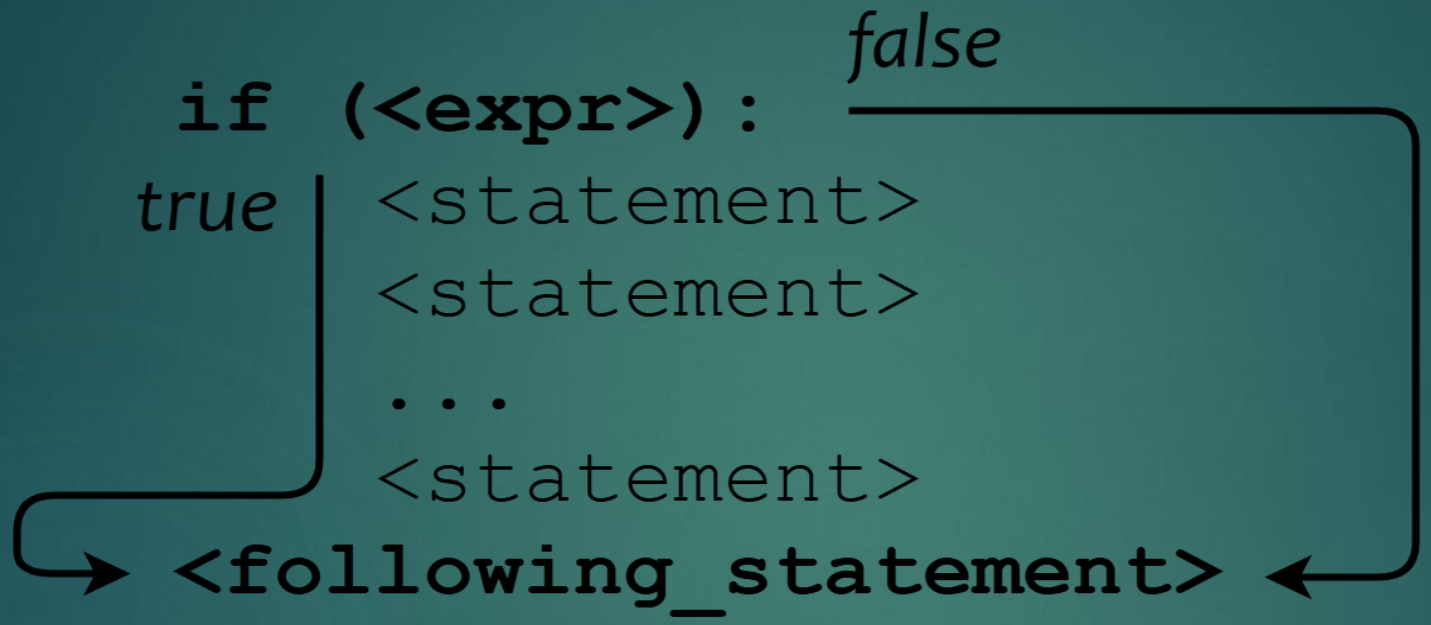
Conditional Statements

- In programming languages, we frequently have to regulate the flow of execution of our program in large projects, and we want to execute a certain group of statements only if a specific condition is met, and an other set of statements if it is not met.
- Decision-making statements are also known as conditional statements.
- If the specified condition is true or false, we must use these conditional expressions to execute the specific block of code.
  - if statements
  - if-else statements
  - elif statements
  - Nested if and if-else statements


if <expr>:
<statement>
<statement>
<statement>
True
False
rest of the code

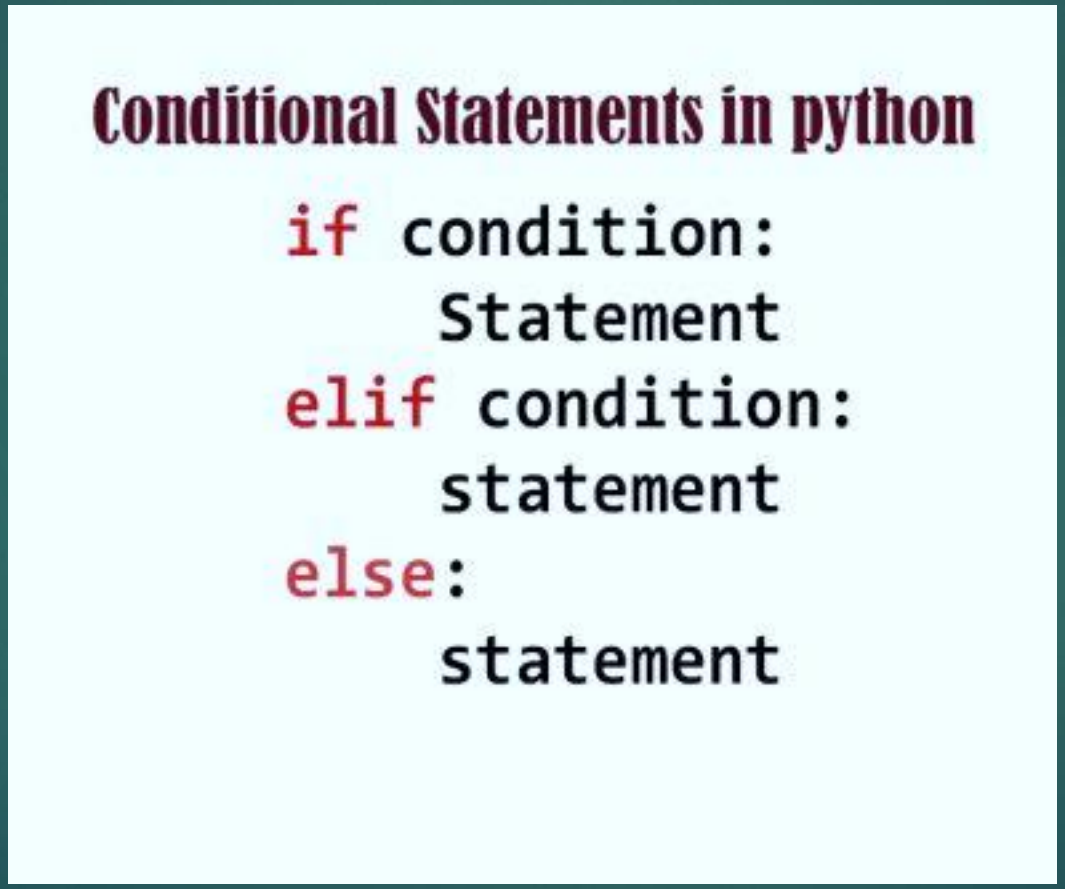

False
if <expr>:
<statement>
<statement>
<statement>
else:
<statement>
<statement>
<statement>
rest of the code

Conditional Statement



In Python, depending on whether a given Boolean constraint evaluates to true or false, conditional statements perform different computations or actions. In Python, IF statements handle conditional statements.

Conditional Statement types



Looping Statement

```
for i in range(1, 5):
  print("i=",i)
  for j in range(i):
   print(j)
  print()
```

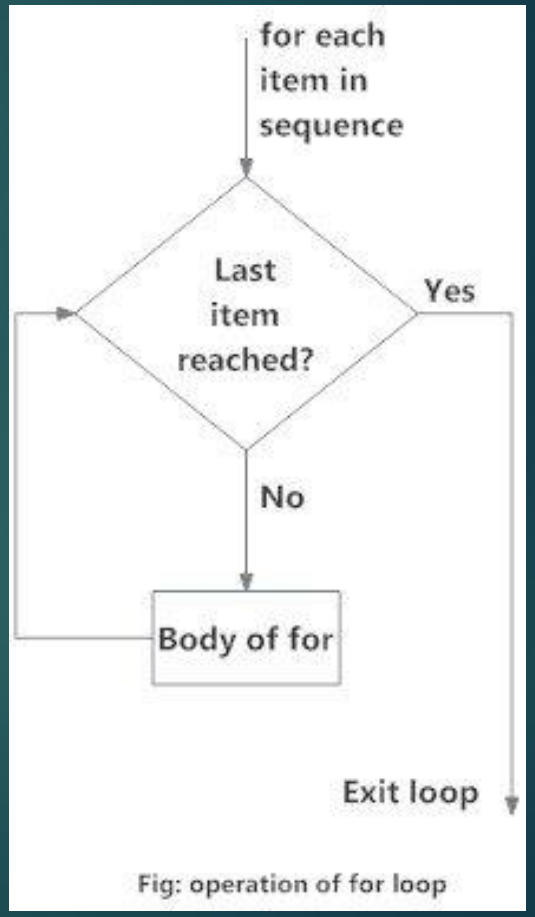


Fig: operation of for loop

Conditions with Looping Statement

```
count = 0
while (count < 3):
  if count==2:
    break
  count = count + 1
  print("Python Program")
```

```
for i in range(1, 5):
  print("i=",i)
  for j in range(i):
   print(j)
  print()
```

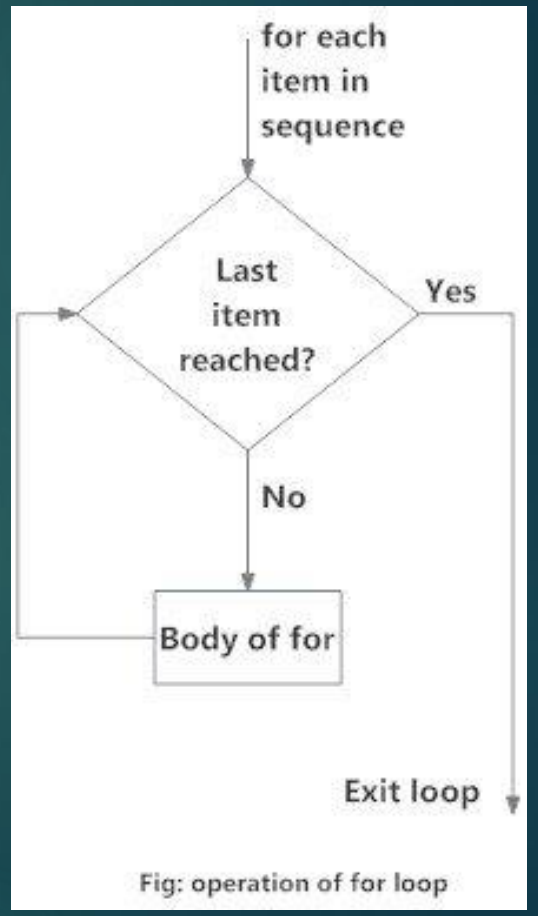
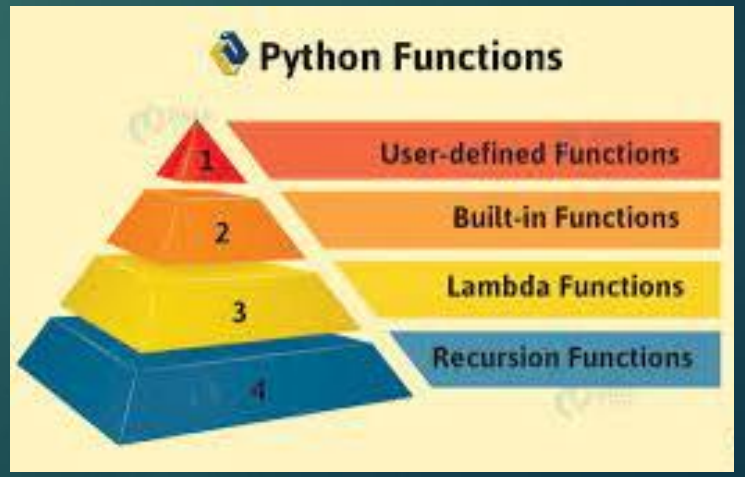


Fig: operation of for loop

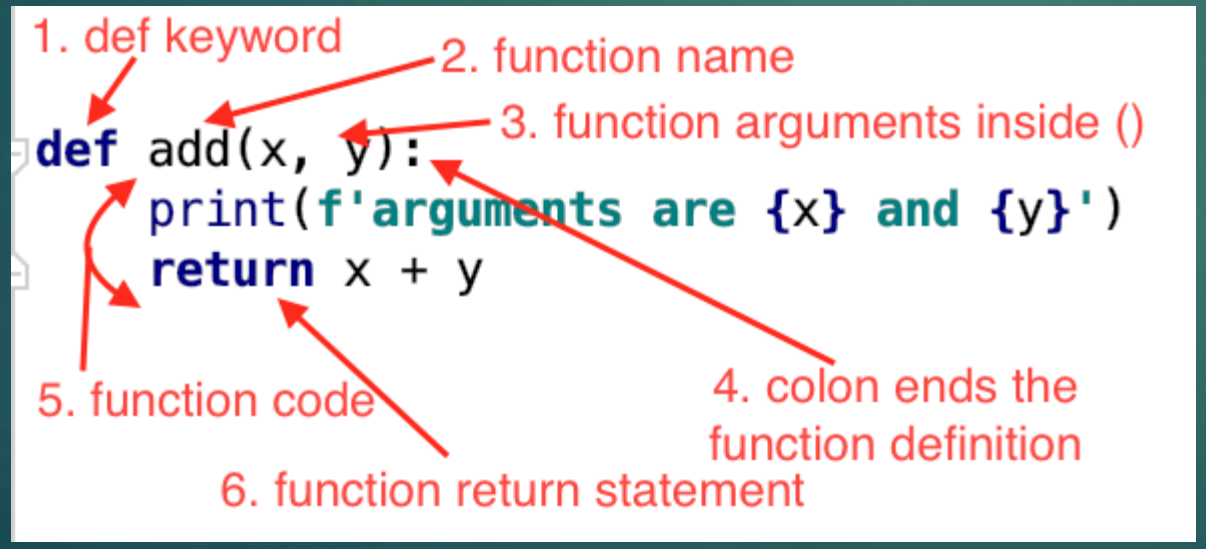
Creating Functions

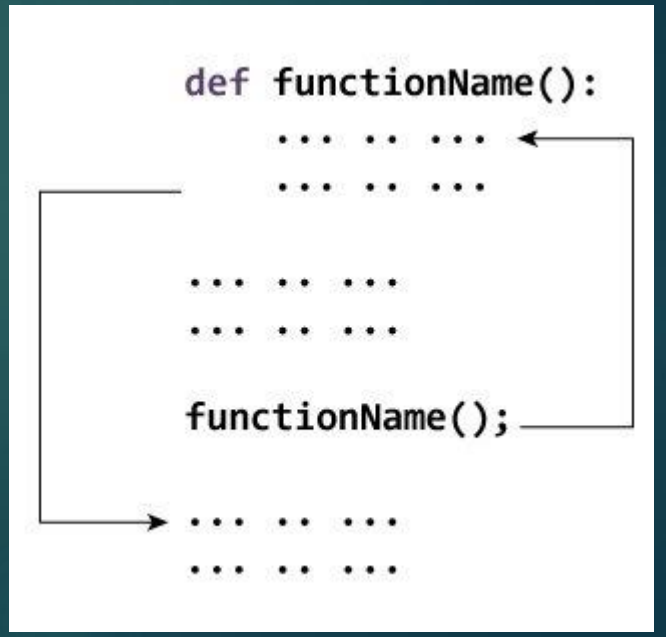
- A function is a reusable, ordered chunk of code that performs a single, connected activity. Functions provide your program more modularity and allow you to reuse a lot of code.
- Many built-in functions, such as print(), are available in Python, but you can also write your own. User-defined functions are what they're called.



How to create Function

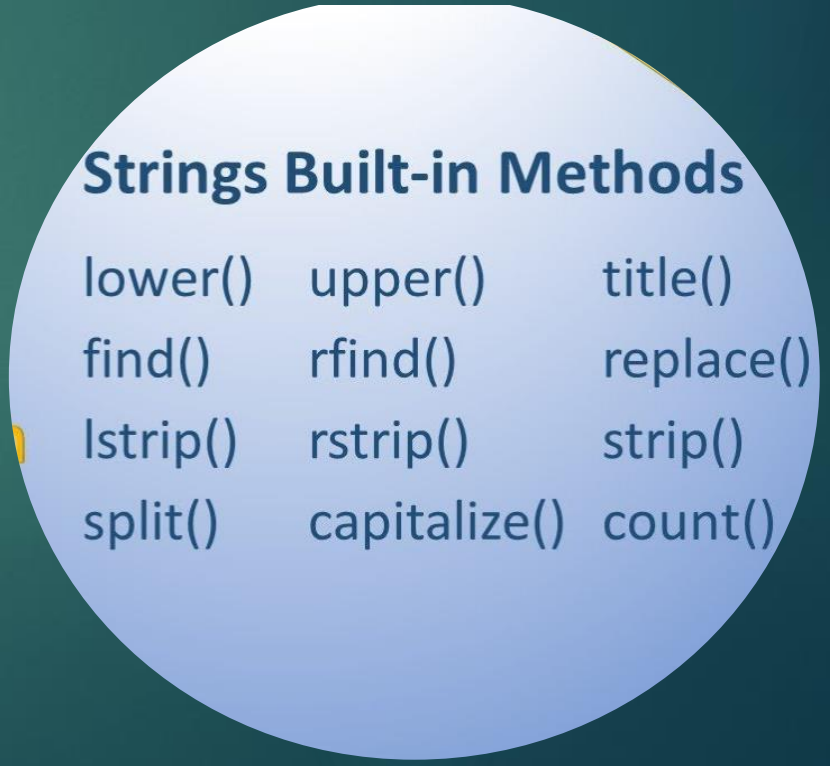
- The term def is followed by the function name and parentheses (())
  in a function block.





Built in Function

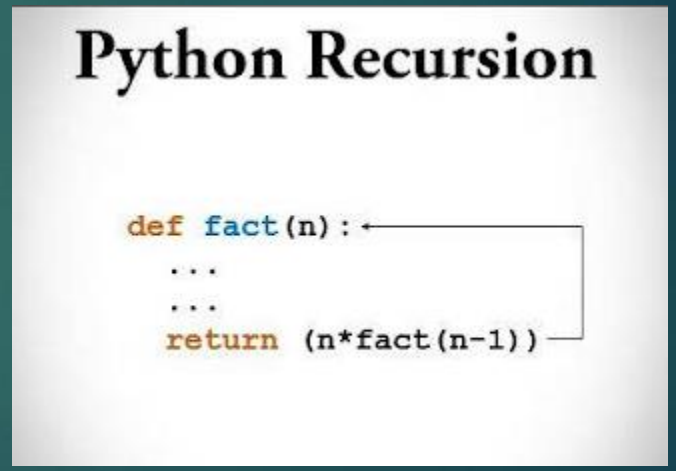
- Python built-in functions are defined as functions with pre-defined functionality in Python. T
- he Python interpreter provides a number of built-in functions that can be used at any time. Built-in Functions are the name for these functions.



Recursive Function

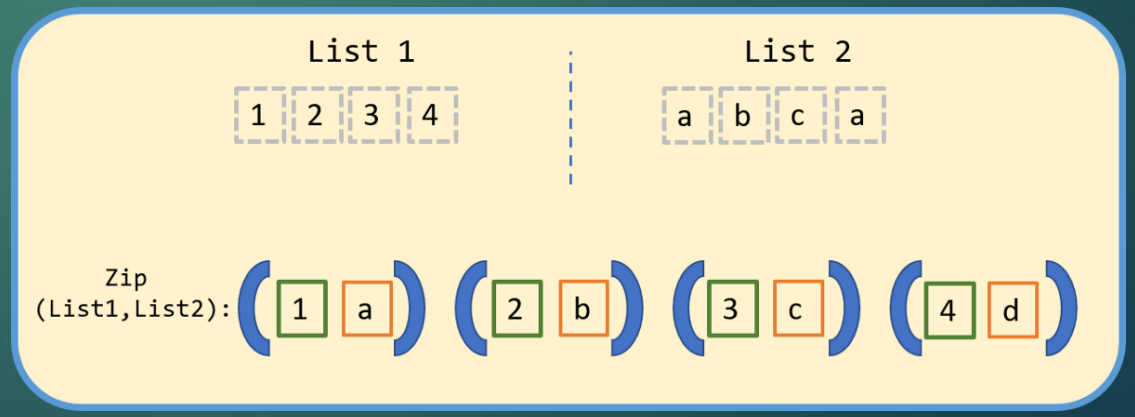
- A recursive function is one that calls itself while it is being executed. The process may be repeated numerous times, with each iteration ending with the output of the result.

```
function Count (integer N)
if (N <= 0) return "Must be a Positive Integer";
if (N > 9) return "Counting Completed";
else return
Count (N+1);
end function
```



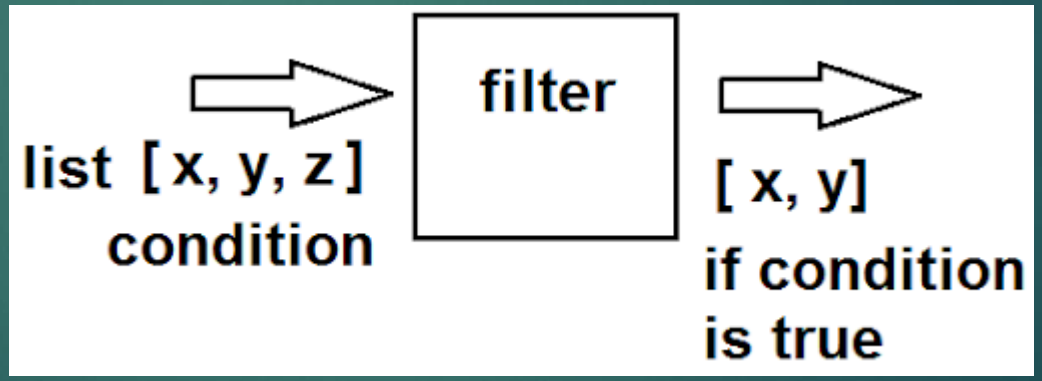
Other Function

- **Zip Function**
- The zip() function returns a zip object, which is an iterator of tuples in which the first item in each provided iterator is coupled together, and so on.
- If the lengths of the provided iterators differ, the length of the new iterator is determined by the iterator with the fewest items.



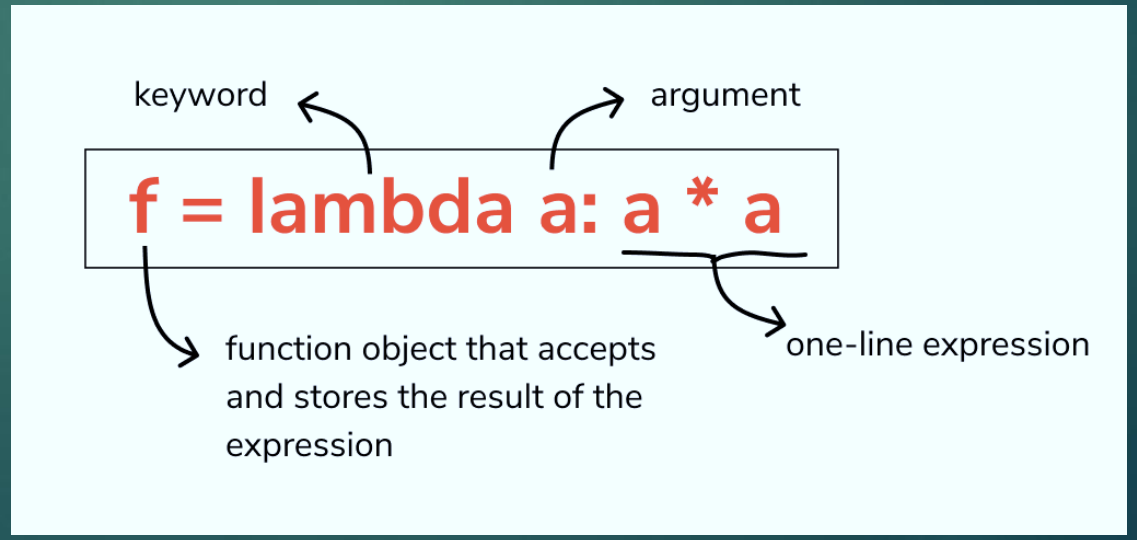
Other Function

- **Filter Function**
- The filter() method filters a series using a function that checks if each element in the sequence is true or not.



Other Function

- **Lambda Function**
- Lambda Functions, sometimes known as 'Anonymous Functions,' are similar to conventional Python functions except that they can be defined without a name.



Other Function

- **Map Function**
- A Lambda Function, commonly known as a "anonymous function," is a Python function that can be defined without a name.