AMIT ACADEMY
for Computer Education

SUBJECT – PROGRAMMING WITH PYTHON

Faculty Name – Ankit Rami

Unit-1

Introduction of Python

AMIT ACADEMY FOR COMPUTER EDUCATION

Nr Vardayini Mataji Temple, Rupal, Gandhinagar-382630

Email – amitacademy1117@gmail.com

Mobile No – 8460467193

YouTube Link –
https://www.youtube.com/@ankitramijoinar

Instagram Link –
https://www.instagram.com/amitacademy17/

Facebook Link –
https://www.facebook.com/aramitacademy/

What is Python?

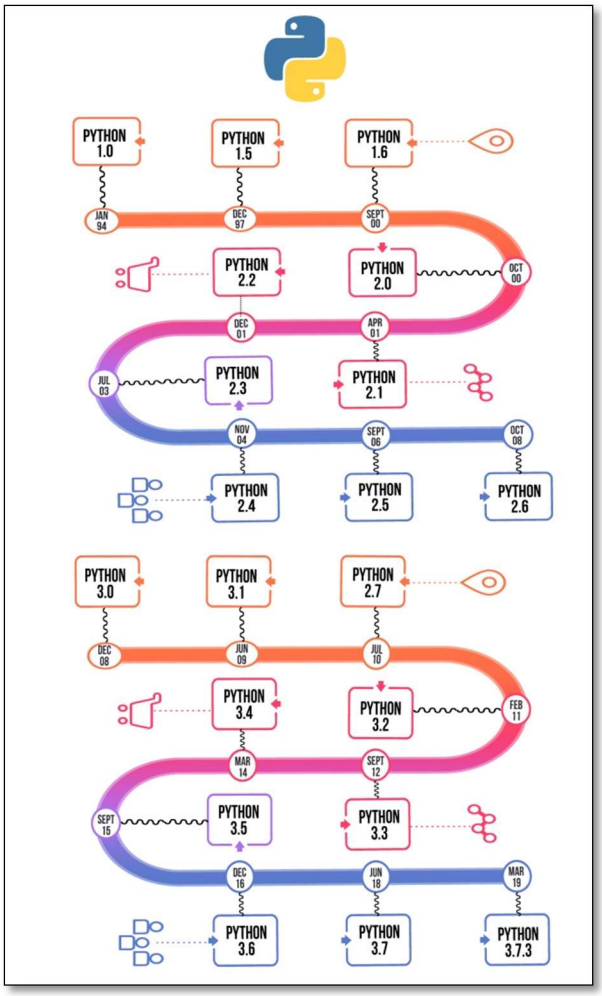
- ✓ Python is an interpreter, object-oriented, high-level programming language.
- ✓ It was created by Guido van Rossum, and released in 1991.
- ✓ Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- ✓ Python has a simple syntax similar to the English language.
- ✓ Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development

History of Python?

- ✓ There is a fact behind choosing the name Python. Guido van Rossum was reading the script of a popular BBC comedy series "Monty Python's Flying Circus". It was late on-air 1970s.
- ✓ Van Rossum wanted to select a name which unique, sort, and little-bit mysterious. So he decided to select naming Python after the "Monty Python's Flying Circus" for their newly created programming language.
- ✓ The comedy series was creative and well random. It talks about everything. Thus it is slow and unpredictable, which made it very interesting.
- ✓ Python is also versatile and widely used in every technical field, such as Machine Leaning, Artificial Intelligence, Web Development, Mobile Application, Desktop Application, Scientific Calculation, etc.
- ✓ Python laid its foundation in the late 1980s.
- ✓ The implementation of Python was stated in December 1989 by Guido Van Rossum at CWI in Netherland.
- ✓ In February 1991, Guido Van Rossum published the code (labeled version 0.9.0) to alt. sources.

PROGRAMMING WITH PYTHON

- ✓ In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- ✓ Python 2.0 added new features such as list comprehension, garbage collection systems.
- ✓ On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
- ✓ ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.
- ✓ The following programming languages influence Python:
    1. ABC language.
    2. Modula-3.



Need of Python (Why you should learn python)

✓ Python is an object-oriented and open-source language. Tech Giants like Cisco, IBM, Mozilla, Google, Quora, Hewlett-Packd, Dropbox, and Qualcomm e using this language owing to its simplicity and elegance. Most developers prefer Python over the plethora of programming languages out there because of its emphasis on readability and efficiency. There e several reasons why you should consider Python. In this Python can provide numerous benefits to the users.

1. **Data Science**
2. **Easy to Learn**
3. **Cross-Platform and Open Source**
4. **Versatile Language and Platform**
5. **Vast Libraries**
6. **Flexibility**
7. **High Salary**
8. **Scripting and Automation**
9. **Artificial Intelligence**
10. **Computer Graphics**
11. **Testing Framework**
12. **Web Development**

1. Data Science

- ✓ Python is the preferred programming language of most data scientists. Be it IT ops, software development or marketing, currently every job makes use of data and depends on it to drive their operations. With the release of 'Numpy' and 'Pandas', Python rose to prominence in the world of data. Python also handles statistical, table and matrix data and also visualizes it with Libraries like 'Matplotlib' and 'Seaborn'.

2. Easy to Learn

- ✓ Python is an easy language to master. This is chiefly because of its resemblance to the English language. Python's syntax is characterized by very few rules and special cases. It's safe to say that in Python the focus is on what you want to do with the code, not on language intricacies. Anybody can master Python easily. With practice, newbies can build a basic game in mere days using python. Another attractive aspect of this programming language is its efficiency and readability.

3. Cross-Platform and Open Source

- ✓ It's been more than 20 yes since this language has been running cross-platform and open source. Be it Linux, Windows or MacOS, Python code works on every platform. Another remarkable thing about Python is that it's supported by decades of bug-squashing and kink-straightening which ensures that its code works as intended whenever the user runs it.

4. Versatile Language and Platform

- ✓ Python remains very relevant today as it can be used in any operations scenario or software development, be it in managing local and cloud infrastructure, working against a SQL database, developing a custom function for Hive & Pig, supporting object-oriented design or even developing a small tool for the user.

5. Vast Libraries

- ✓ Python is supported by PyPI which has 85,000+ python scripts and modules accessible to the user. These modules provide pre-packaged functionality available to the users in their local Python environment. It can solve diverse problems such as executing advanced data analytics like developing RESTful web services or sentiment analysis and establishing computer vision.

6. Flexibility

- ✓ Python has several powerful applications integrated with other programming languages.
- ✓ Details about these e given as follows:
  - .Net and C# compatible: IronPython
  - A version with C: CPython
  - Python combined with Ruby: RubyPython
  - Python integrated with Java: Jython
  - Python written with Objective C toolkits: PyObjc

7. High Salary

- ✓ The salary of Python engineers is comparatively higher as compared to others in the industry. In the United States, a Python developer earns an average of $116,028 per year.

8. Scripting and Automation

- ✓ What most people don't know about Python is that it can be used as a scripting language. In scripting, the code is written in script form and gets executed. So the code is read and interpreted by the machine and errors e checked during runtime. After the code is checked, it can be used many times. It is also possible to automate specific tasks in a problem by automation.

9. Artificial Intelligence

- ✓ Without any dispute, Artificial intelligence is going to lead the future IT. Python's libraries such as Keras and Tensor Flow enable machine learning functionality. Also, libraries like OpenCV aids in computer vision or image recognition.

10. Computer Graphics

- ✓ Python can be employed in small, large, online or offline projects. It is used to develop GUI and desktop applications. It's 'Tkinter' library enables simple and rapid application development. This programming language is also used in game development where the logic is written using a module 'pygame' which can also run on android devices.

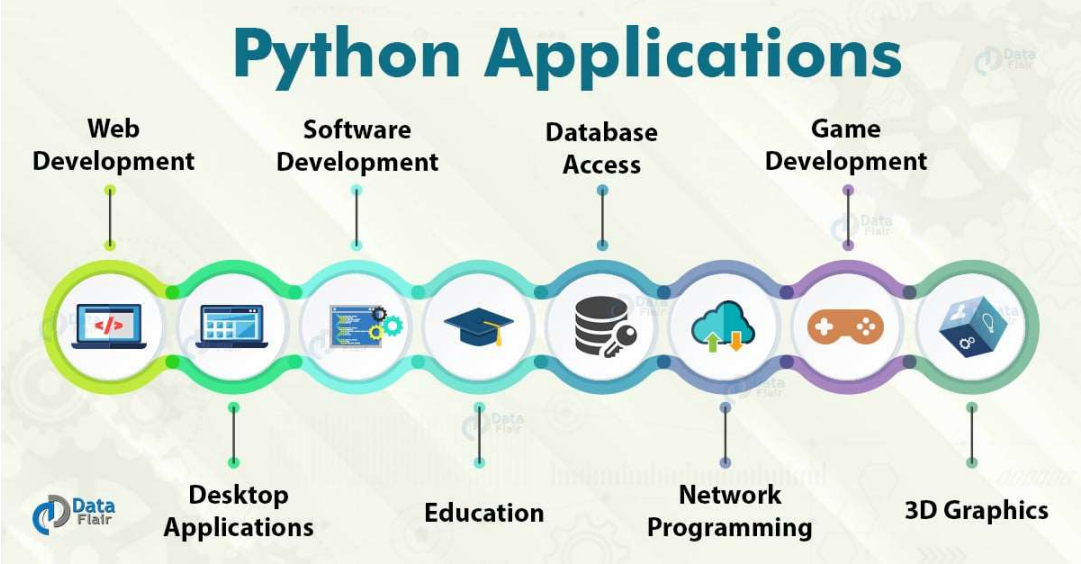
11. Testing Framework

- ✓ This language is an excellent tool for validating the products or ideas for established enterprises. Python has numerous built-in testing frameworks that deal with debugging and rapid workflows. Its tools and modules such as Selenium and Splinter work to make things easier. Python also supports cross-platform and cross-browser testing with frameworks like PyTest and Robot framework.

12. Web Development

- ✓ Python's different frameworks support website development. Python has an array of frameworks for developing websites. Popular frameworks such as Django, Flask, and Pylons are characterized by faster and stable code; this is because they are written in Python. Using Python, the users can perform web scraping which means fetching details from other websites.

Applications of Python



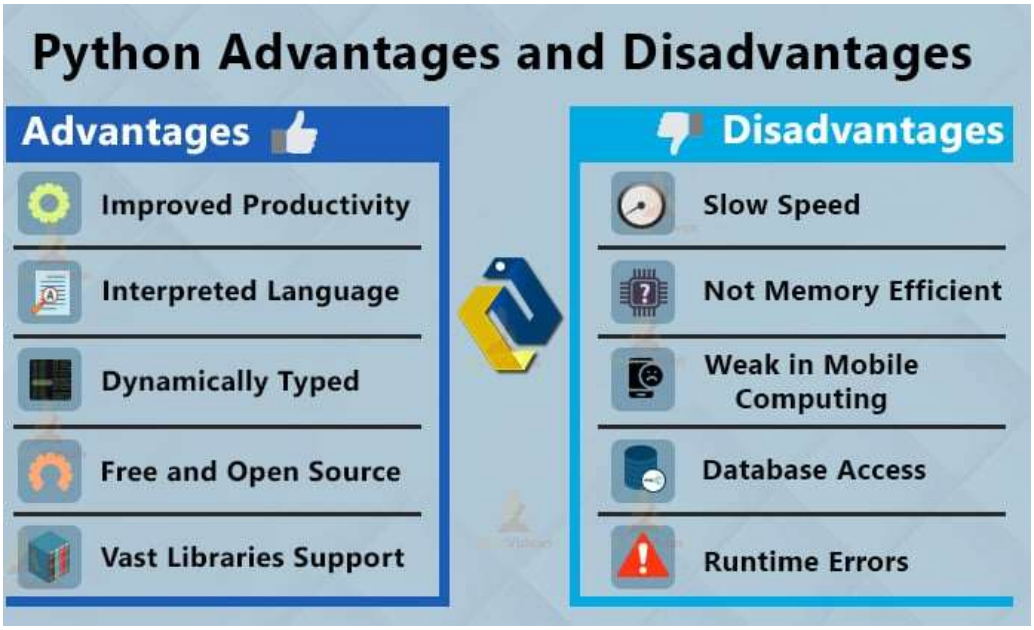
1. **Web and Internet Development**
2. **Desktop GUI Applications**
3. **Science and Numeric**
4. **Software Development**
5. **Education**
6. **Database Access**
7. **Network Programming**
8. **Games and 3D Graphics**

Note - All Topic All Ready Discusses in Why Need of Python Topic

Reference Link –

https://data-flair.training/blogs/python-applications/

Python Advantages and Disadvantages



Advantages of Python

1. Easy to Read, Len and Write

- ✓ Python is a high-level programming language that has English-like syntax. This makes it easier to read and understand the code.
- ✓ Python is really easy to pick up and learn, that is why a lot of people recommend Python to beginners. You need less lines of code to perform the same task as compared to other major languages like C/C++ and Java.

2. Improved Productivity

- ✓ Python is a very productive language. Due to the simplicity of Python, developers can focus on solving the problem. They don't need to spend too much time in understanding the syntax or behavior of the programming language. You write less code and get more things done.

3. Interpreted Language

- ✓ Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred.

- ✓ Python shows only one error even if the program has multiple errors. This makes debugging easier.

4. Dynamically Typed

- ✓ Python doesn't know the type of variable until we run the code. It automatically assigns the data type during execution. The programmer doesn't need to worry about declaring variables and their data types.

5. Free and Open-Source

- ✓ Python comes under the OSI approved open-source license. This makes it free to use and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behavior and use their version for development.

6. Vast Libraries Support

- ✓ The standard library of Python is huge, you can find almost all the functions needed for your task. So, you don't have to depend on external libraries.
- ✓ But even if you do, a Python package manager (pip) makes things easier to import other great packages from the Python package index (PyPi). It consists of over 200,000 packages..

7. Portability

- ✓ In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

Disadvantages of Python

1. Slow Speed

- ✓ We discussed above that Python is an interpreted language and dynamically-typed language. The line by line execution of code often leads to slow execution.
- ✓ The dynamic nature of Python is also responsible for the slow speed of Python because it has to do the extra work while executing code. So, Python is not used for purposes where speed is an important aspect of the project.

2. Not Memory Efficient

- ✓ To provide simplicity to the developer, Python has to do a little tradeoff. The Python programming language uses a large amount of memory. This can be a disadvantage while building applications when we prefer memory optimization.

3. Weak in Mobile Computing

- ✓ Python is generally used in server-side programming. We don't get to see Python on the client-side or mobile applications because of the following reasons. Python is not memory efficient and it has slow processing power as compared to other languages.

4. Database Access

- ✓ Programming in Python is easy and stress-free. But when we e interacting with the database, it lacks behind.
- ✓ The Python's database access layer is primitive and underdeveloped in comparison to the popular technologies like JDBC and ODBC.
- ✓ Huge enterprises need smooth interaction of complex legacy data and Python is thus rarely used in enterprises.

5. Runtime Errors

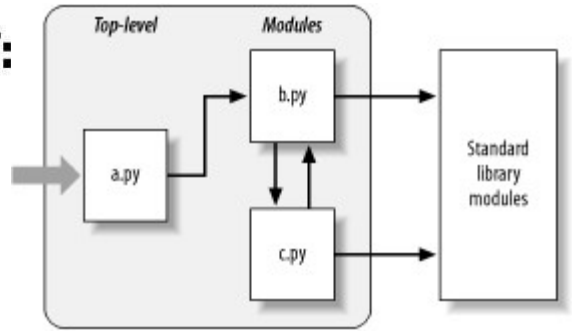
- ✓ As we know Python is a dynamically typed language so the data type of a viable can change anytime. A viable containing integer number may hold a string in the future, which can lead to Runtime Errors.

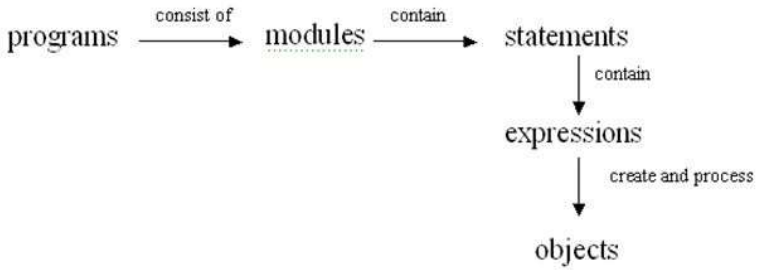
- ✓ Therefore Python programmers need to perform thorough testing of the applications.

Python Program Structure

Python programs consist of:

- Modules
- Statements
- Expressions
- Objects





Attributes and Imports:

- ✓ Import in python is similar to #include header_file in C/C++. Python modules can get access to code from another module by importing the file/function using import. The import statement is the most common way of invoking the import machinery, but it is not the only way.
- ✓ import module_name

Ex- import math
pie = math.pi
print("The value of pi is : ",pie)

- ✓ When the import is used, it searches for the module initially in the local scope by calling __import__() function. The value returned by the function is then reflected in the output of the initial code.

Statements

- ✓ A statement is an instruction that the Python interpreter can execute. We have seen two kinds of statements: print and assignment.
- ✓ When you type a statement on the command line, Python executes it and displays the result, if there is one. The result of a print statement is a value. Assignment statements don't produce a result.
- ✓ A script usually contains a sequence of statements. If there is more than one statement, the results appear one at a time as the statements execute.

Ex-

```
print 1
x = 2
print x
```

Expression

- ✓ An expression is a combination of operators and operands that is interpreted to produce some other value. In any programming language, an expression is evaluated as per the precedence of its operators.
- ✓ A combination of operands and operators is called an expression. The expression in Python produces some value or result after being interpreted by the Python interpreter. An expression in Python is a combination of operators and operands.
- ✓ An example of expression can be : x=x+10. In this expression, the first 10 is added to the variable x. After the addition is performed, the result is assigned to the variable x.

Ex-

```
x = 15+3.1
print x
```

Object

- ✓ Python is an object-oriented programming language. Everything is in Python treated as an object, including variable, function, list, tuple, dictionary, set, etc. Every object belongs to its class. For example - An integer variable belongs to integer class. An object is a real-life entity. An object is the collection of various data and functions that operate on those data. An object contains the following properties.
- ✓ **State -** The attributes of an object represents its state. It also reflects the properties of an object.
- ✓ **Behavior -** The method of an object represents its behavior.
- ✓ **Identity -** Each object must be uniquely identified and allow interacting with the other objects.

Reference Link

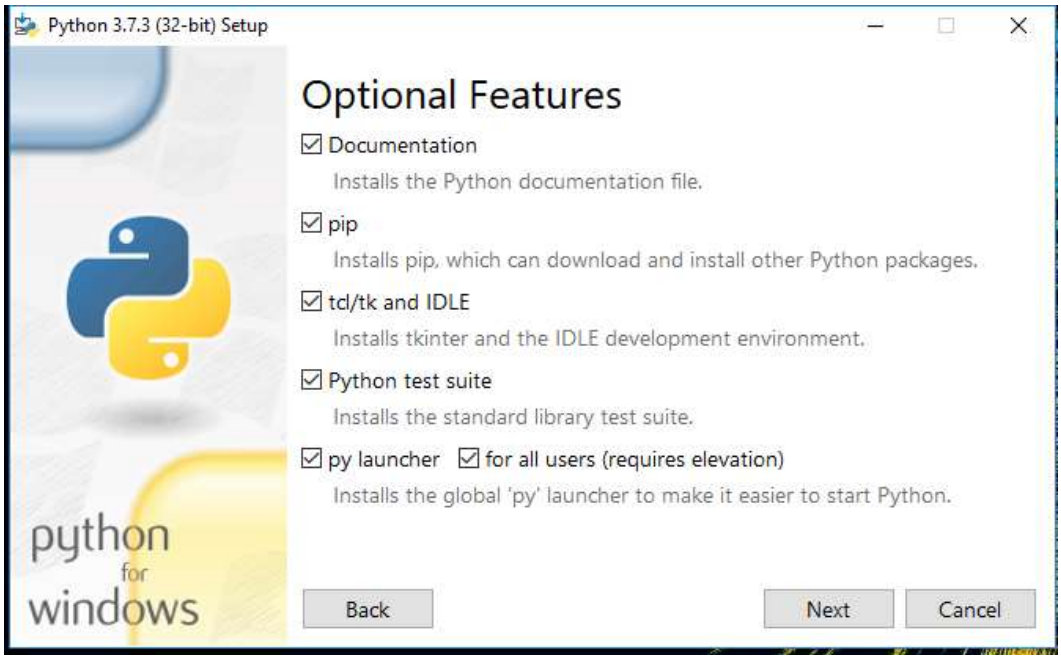
1. https://onlineitguru.com/blog/programming-structure-of-python#:~:text=The%20structure%20Python%20Program%20consists,top%20when%20it%20is%20launched.
2. https://www.geeksforgeeks.org/structuring-python-programs/
3. https://blog.sap-press.com/what-is-the-basic-structure-of-a-python-program

How to Install Python

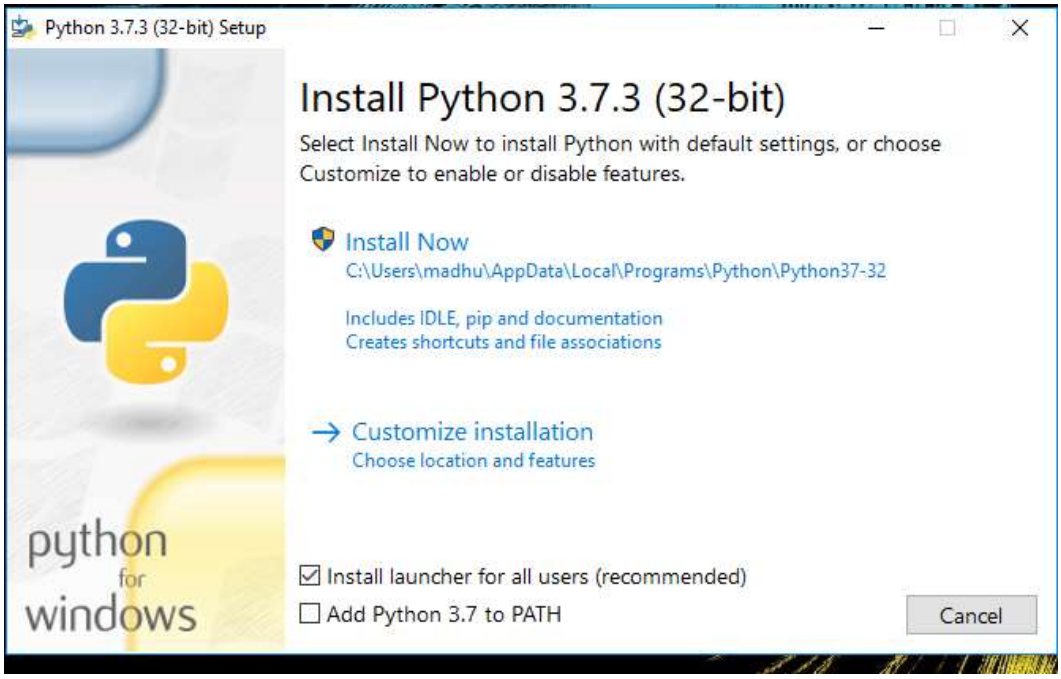
Step 1: Download the Python Installer

Link - https://www.python.org/downloads/

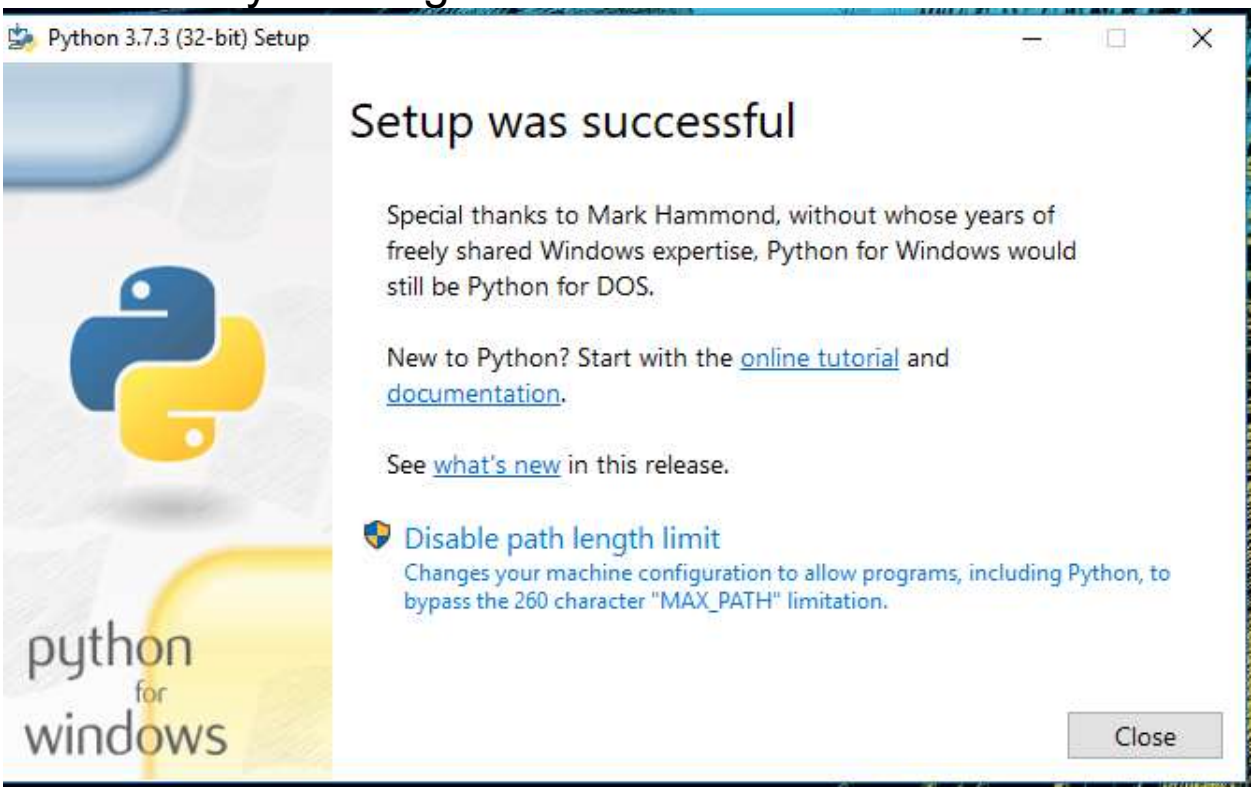
Step 2: Run the Executable Installer



Step 3: Select all Option Installation Time



Step 4: if latest version than all option automatic selected so not need to follow step-3 Than Show Installation Successfully Message.



Reference Link -
https://www.digitalocean.com/community/tutorials/install-python-windows-10

Identifiers in Python

- ✓ Identifier is a user-defined name given to identities like class, functions, variables, modules, or any other object in Python.
- ✓ Identifiers are the name given to variables, classes, methods, etc.

Rules for Naming an Identifier

- ✓ Identifiers cannot be a keyword.
- ✓ Identifiers are case-sensitive.
- ✓ It can have a sequence of letters and digits. However, it must begin with a letter or _. The first letter of an identifier cannot be a digit.
- ✓ It's a convention to start an identifier with a letter rather _.
- ✓ Whitespaces are not allowed.
- ✓ We cannot use special symbols like !, @, #, $, and so on.

Python Valid Identifiers Example

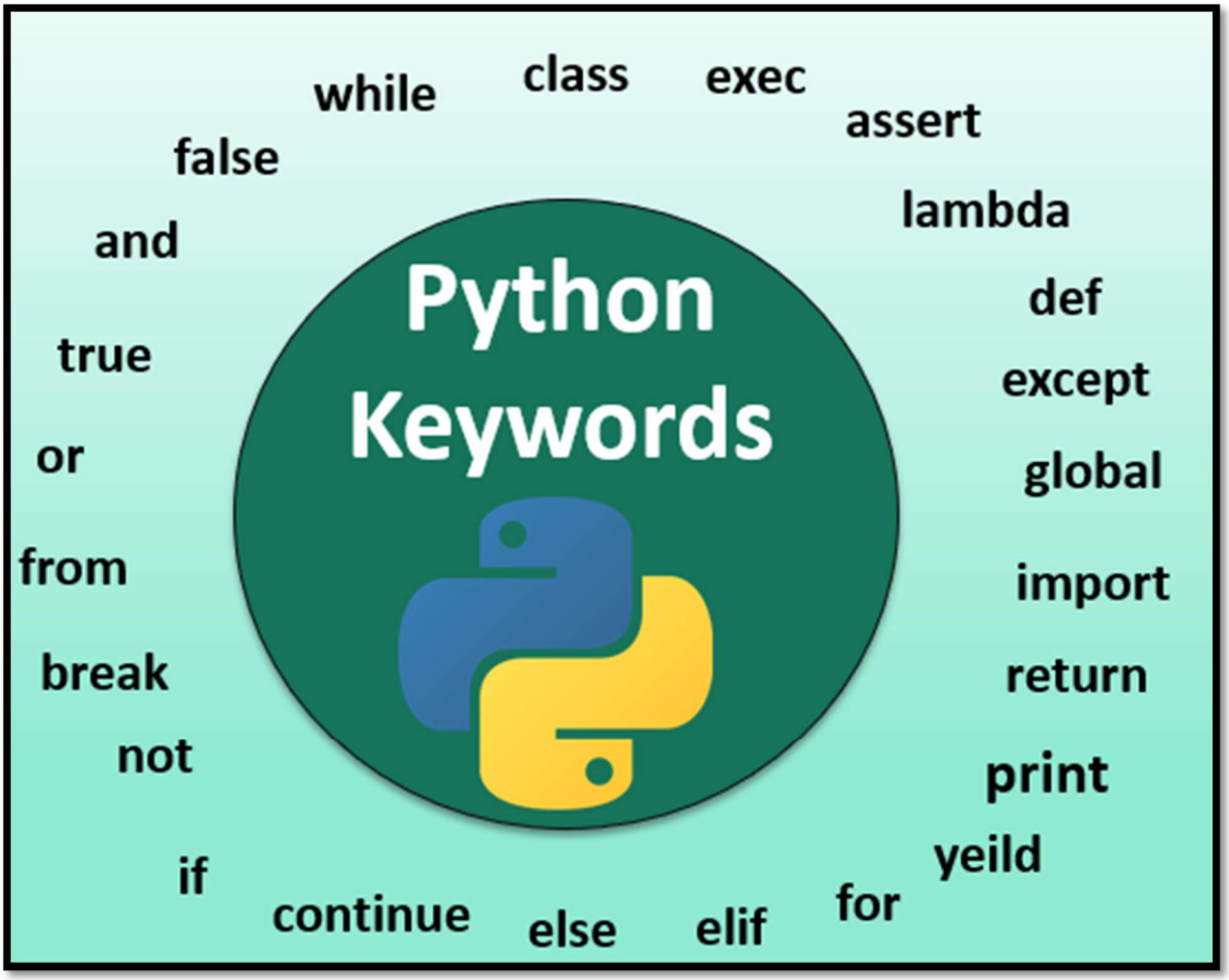
- abc123
- abc_de
- _abc
- ABC
- abc

Python Invalid Identifiers Example

- 123abc
- abc@
- 123
- For

Keywords in Python

- ✓ Keywords are some predefined and reserved words in python that have special meanings.
- ✓ Keywords are used to define the syntax of the coding.
- ✓ The keyword cannot be used as an identifier, function, and variable name.
- ✓ All the keywords in python are written in lower case except True and False.
- ✓ There are 33 keywords in Python 3.7



PROGRAMMING WITH PYTHON
Python Keywords Description

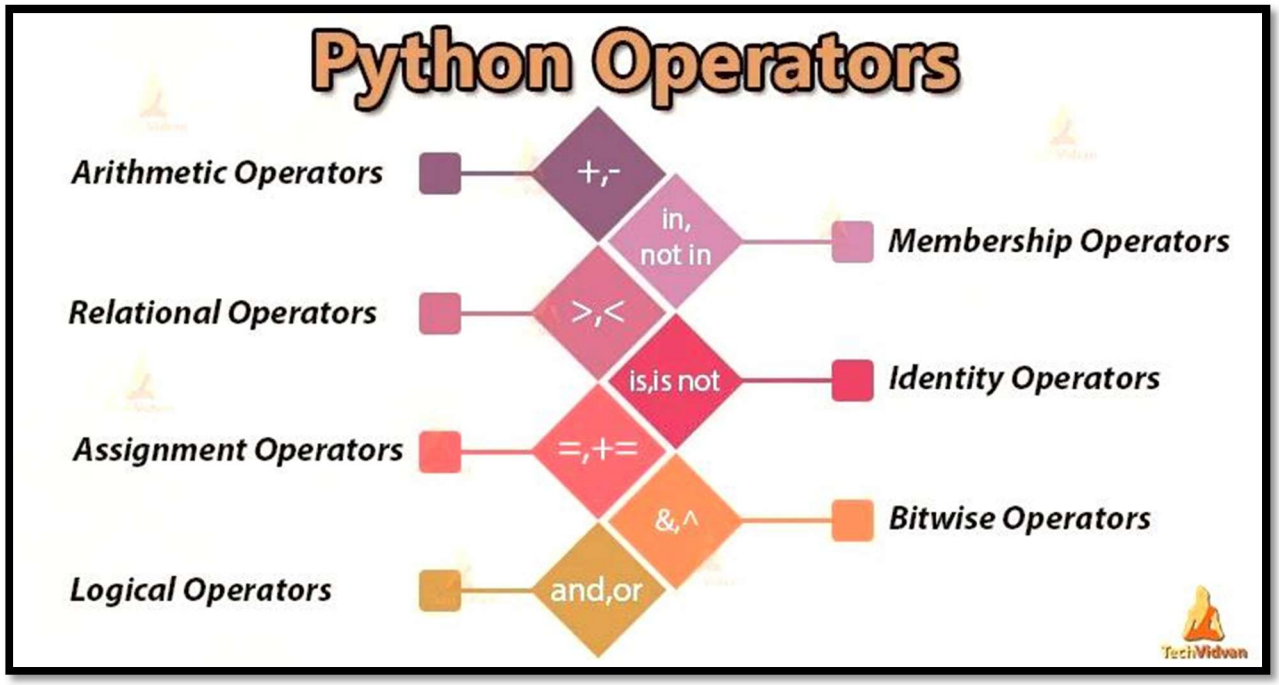
and	A logical operator
as	To create an alias
assert	For debugging
break	To break out of a loop
class	To define a class
continue	To go to the next iteration of a loop
def	To define a function
del	To delete an object
elif	A conditional statements, like else if
else	A conditional statements
except	Used with exceptions, what to do when an exception occurs
False	Boolean value
finally	Used with exceptions, will be executed no matter if there is an exception or not
for	To create a for loop
from	To import specific parts of a module
global	To declare a global variable
if	To make a conditional statement
import	To import a module
in	To check if a value is in a list, tuple
is	To test if two variables are equal
lambda	To create an anonymous function
None	Represents a null value
nonlocal	To declare a non-local variable
not	A logical operator
or	A logical operator
pass	A statement that will do nothing (null)
raise	To raise an exception
return	To exit a function and return a value
True	Boolean value
try	To make a try...except statement
while	To create a while loop
with	Used to simplify exception handling
yield	To end a function, returns a generator

Operators in Python.

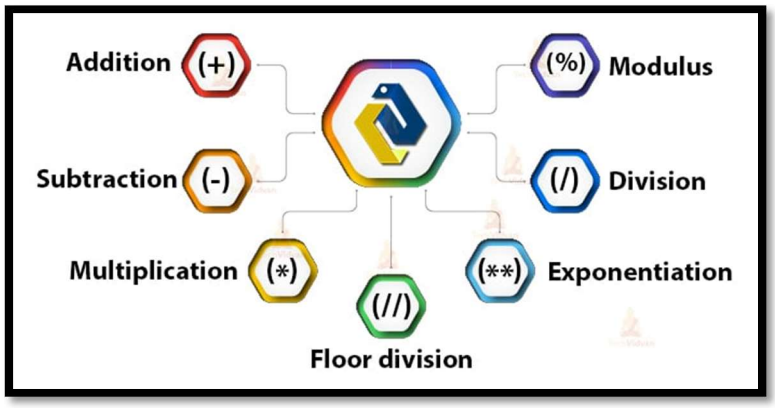
What is Operators?

- ✓ An operator is a symbol that will perform mathematical operations on variables or on values.
- ✓ Operators operate on operands (values) and return a result.

Python has 7 types of operators.



1. Arithmetic Operators



- ✓ arithmetic operators are used with numeric values to perform common mathematical operations

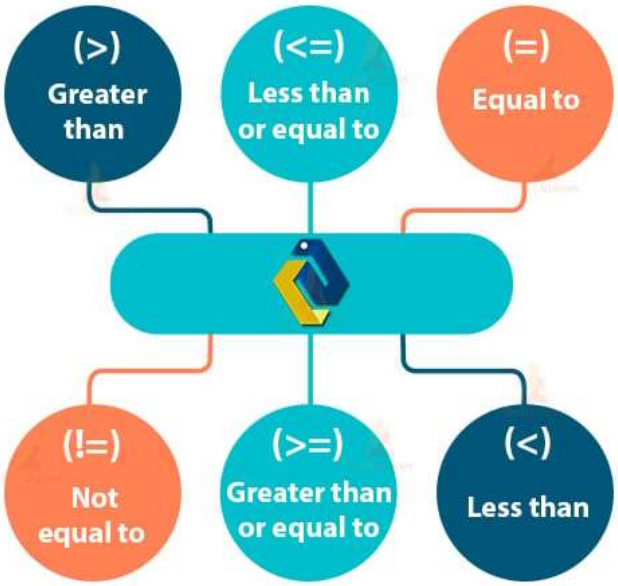
✓ **Ex-**

```
a=50
b=2
print("Add ans is - ",(a+b))
print("Sub ans is - ",(a-b))
print("Mult ans is - ",(a*b))
print("Divi ans is - ",(a/b))
print("Mod ans is - ",(a%b))
print("Expo ans is - ",(a**b))
print("FloDiv ans is - ",(a//b))
```

Output:-

```
Add ans is -  52
Sub ans is -  48
Mult ans is -  100
Divi ans is -  25.0
Mod ans is -  0
Expo ans is -  2500
FloDiv ans is -  25
```

2. Relational Operators



- ✓ They are also called comparison operators and they compare values.
- ✓ Comparison operators are used to compare two values.
- ✓ Python has 6 relational operators:
  - > (Greater than)
  - < (Less than)
  - == (Equal to)
  - != (Not equal to)
  - >= (Greater than or equal to)
  - <= (Less than or equal to)
- ✓ **Ex-**

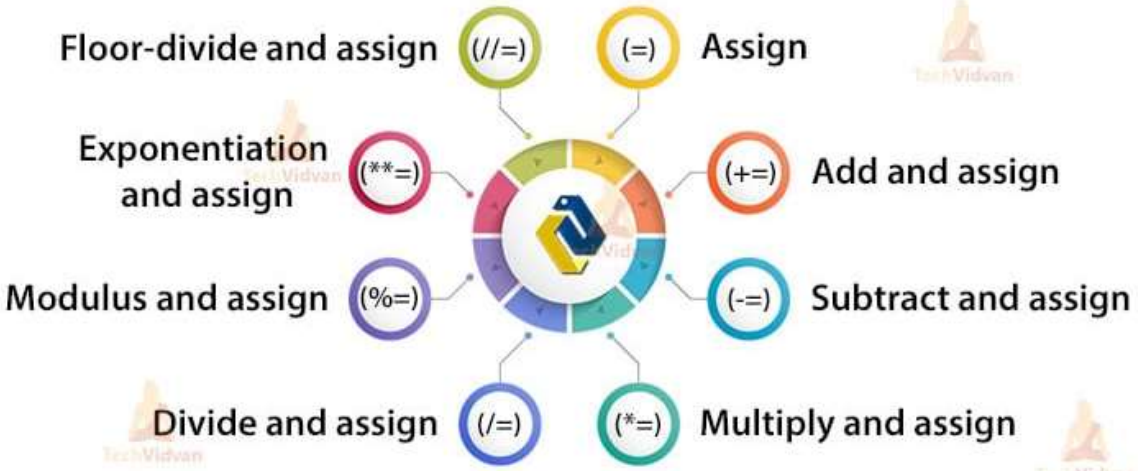
```
a=5
#< and <= 1 2 3 4
if(a<=5):
    print("Value Match")
else:
    print("Value Not Match")
```

```
#> and >= 1 2 3 4
b=45
if(b>=5):
    print("Value Match")
else:
    print("Value Not Match")
#==
c=4
if(c==5):
    print("Value Match")
else:
    print("Value Not Match")
#!=
d=7
if(d!=5):
    print("Value Match")
else:
    print("Value Not Match")
```

Output:-

```
Value Match
Value Match
Value Not Match
Value Match
```

3. Assignment Operators



- ✓ Assignment operators are used to assign values to variables
- ✓ Python has 8 assignment operators:
  - = (Assign)
  - += (Add and assign)
  - -= (Subtract and assign)
  - *= (Multiply and assign)
  - /= (Divide and assign)
  - %= (Modulus and assign)
  - **= (Exponentiation and assign)
  - //= (Floor-divide and assign)
- ✓ **Ex-**

```
a=50
print("Print A value - ",a)
a+=50
print("Print A value After += Ope",a)
a-=50
```

```
print("Print A value After -= Ope",a)
a*=2
print("Print A value After *= Ope",a)
a/=2
print("Print A value After /= Ope",a)
"""a%=2"""
#print("Print A value After %= Ope",a)]
```

Output:-

```
Print A value -  50
Print A value After += Ope 100
Print A value After -= Ope 50
Print A value After *= Ope 100
Print A value After /= Ope 50.0
```

4. Logical Operators

✓ They can combine conditions.

✓ Python has 3 logical operators:

- and (Logical and)
- or (Logical or)
- not (Logical not)

✓ **Ex-**

```
un="ankit"
pw="ar"
# and both check
if(un=="ankit" and pw=="ar"):
    print("Login Success")
else:
    print("Login Fail")
# or match any singel value
```

```
if(un=="smit" or pw=="ar"):
    print("Login Success")
else:
    print("Login Fail")
# Not
if(un!="ankit" and pw!="ar"):
    print("Login Success")
else:
    print("Login Fail")
```

Output:-

```
Login Success
Login Success
Login Fail
```

5. Membership Operators

✓ Membership operators check whether a value is in another.

✓ Python has 2 membership operators:

- in
- not in

✓ **Ex-**

```
x=('a','b','c','d','e')
print('a' in x)
print('z' not in x)
print('d' not in x)
```

Output:-

True
True
False

6. Identity Operators

✓ Identity operators check whether two values are identical.

✓ Python has 2 identity operators as well:

- is
- is not

✓ **Ex-**

```
a=5
b=a
c='5'
d=5
print(a is b)
print(a is not c)
```

Output:-

True
True

7. Bitwise Operators

✓ They operate on values bit by bit.

✓ Python has 6 bitwise operators:

- & (Bitwise and)
- | (Bitwise or)
- ^ (Bitwise xor)
- ~ (Bitwise 1's complement)
- << (Bitwise left-shift)
- >> (Bitwise right-shift)

✓ **Ex-**

```
print(3&4)
print(3|4)
print(3^4)
print(~3)
print(4<<2)
print(4>>2)
```

Output:-

```
0
7
7
-4
16
1
```

Reference Link

1. https://techvidvan.com/tutorials/python-operators/
2. https://www.w3schools.com/python/python_operators.asp
3. https://www.geeksforgeeks.org/python-operators/
4. https://www.programiz.com/python-programming/operators

Any Query Contact Us

Faculty Name- Ankit Rami

Email – ankitramiblog@gmail.com

Contact No – +91 8460467193

Website - amit.arinfoway.com

Subscribe Our YouTube Channel

https://www.youtube.com/channel/UCWbJh2iQ8w-8nrU0Xpjpw7g