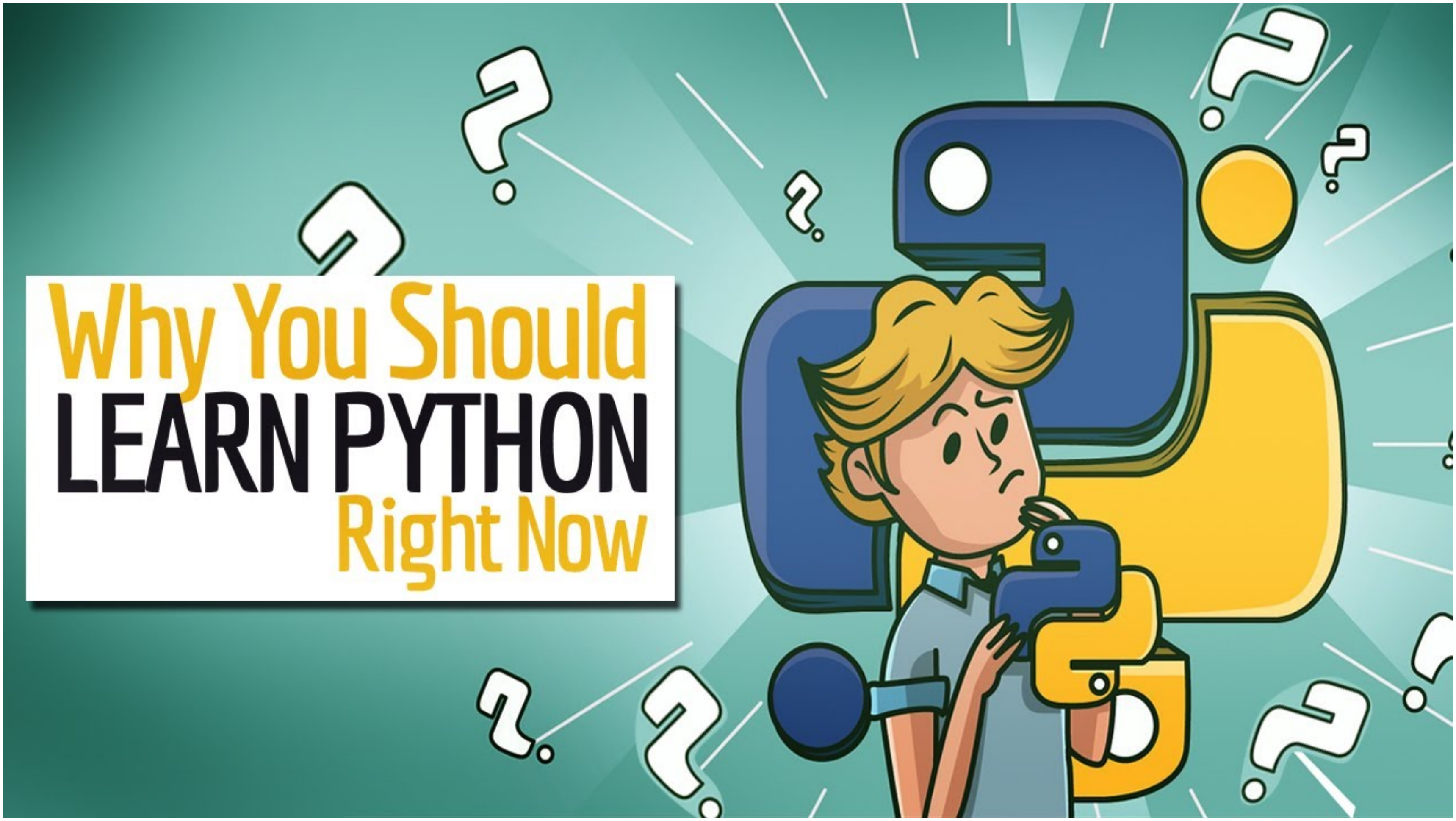
Parul® University

Python Programming

PU

CHAPTER-1

Introduction to Python


Why You Should
LEARN PYTHON
Right Now

What is Python?

- High Level Programming Language
- Shorter Language(3-5 lines than java, 5-10 lines than c++)
- Since 1991, consistently in top 10 most popular computing language

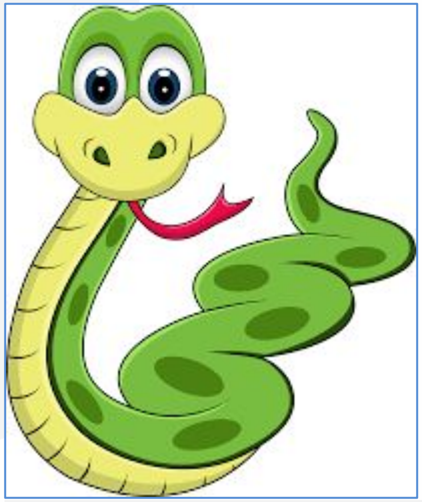


Image source : https://www.nicepng.com/

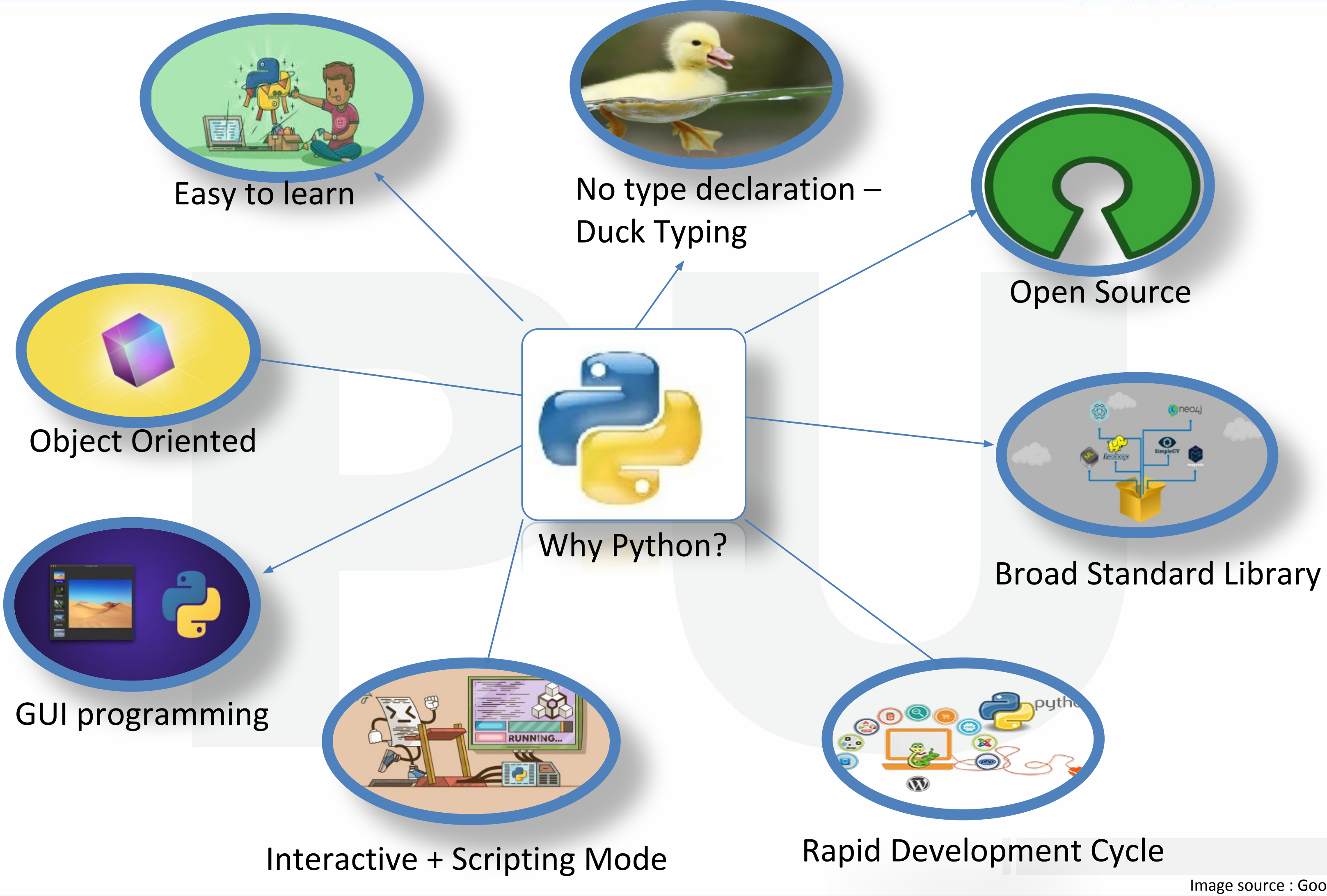
History

- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning
- Managed by Python Software Foundation

- 1994
  - Python 1.0
- 2000
  - Python 2.0
- 2008
  - Python 3.0

Figure 1.1 Python Release

PU



Why Python?

Easy to learn

No type declaration – Duck Typing

Open Source

Object Oriented

Broad Standard Library

GUI programming

Interactive + Scripting Mode

Rapid Development Cycle

Image source : Google

Moral of the Python Story

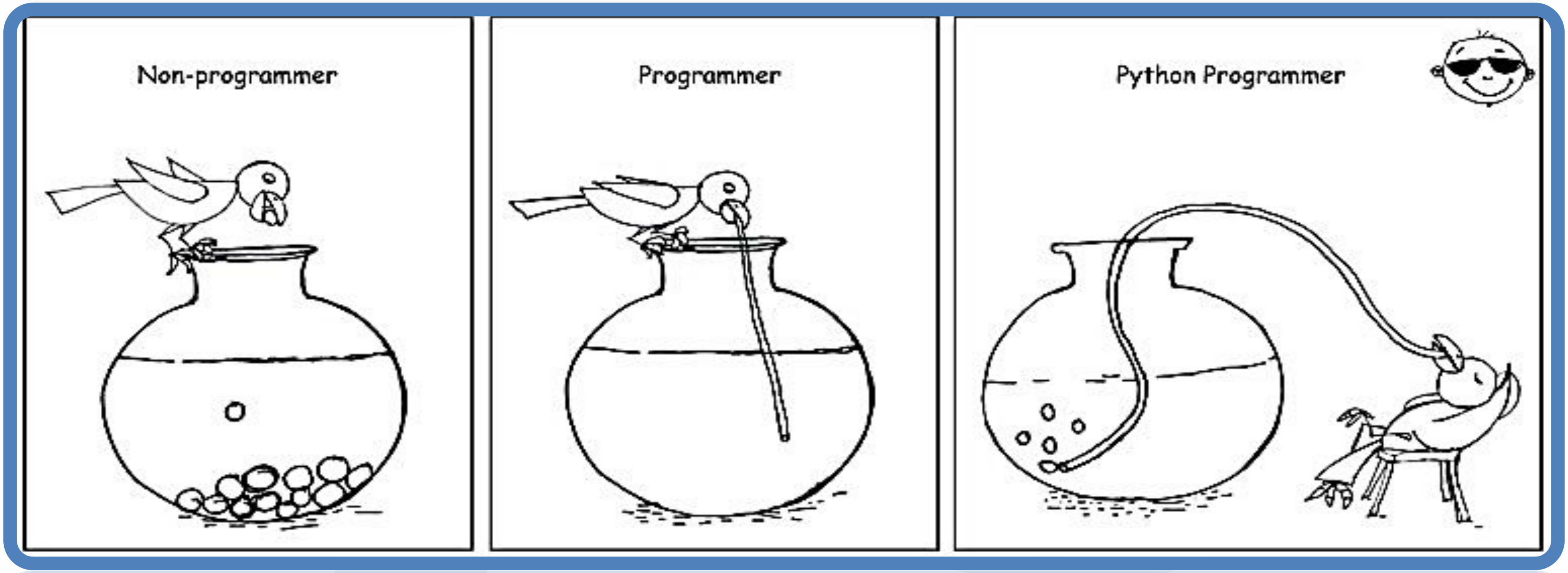
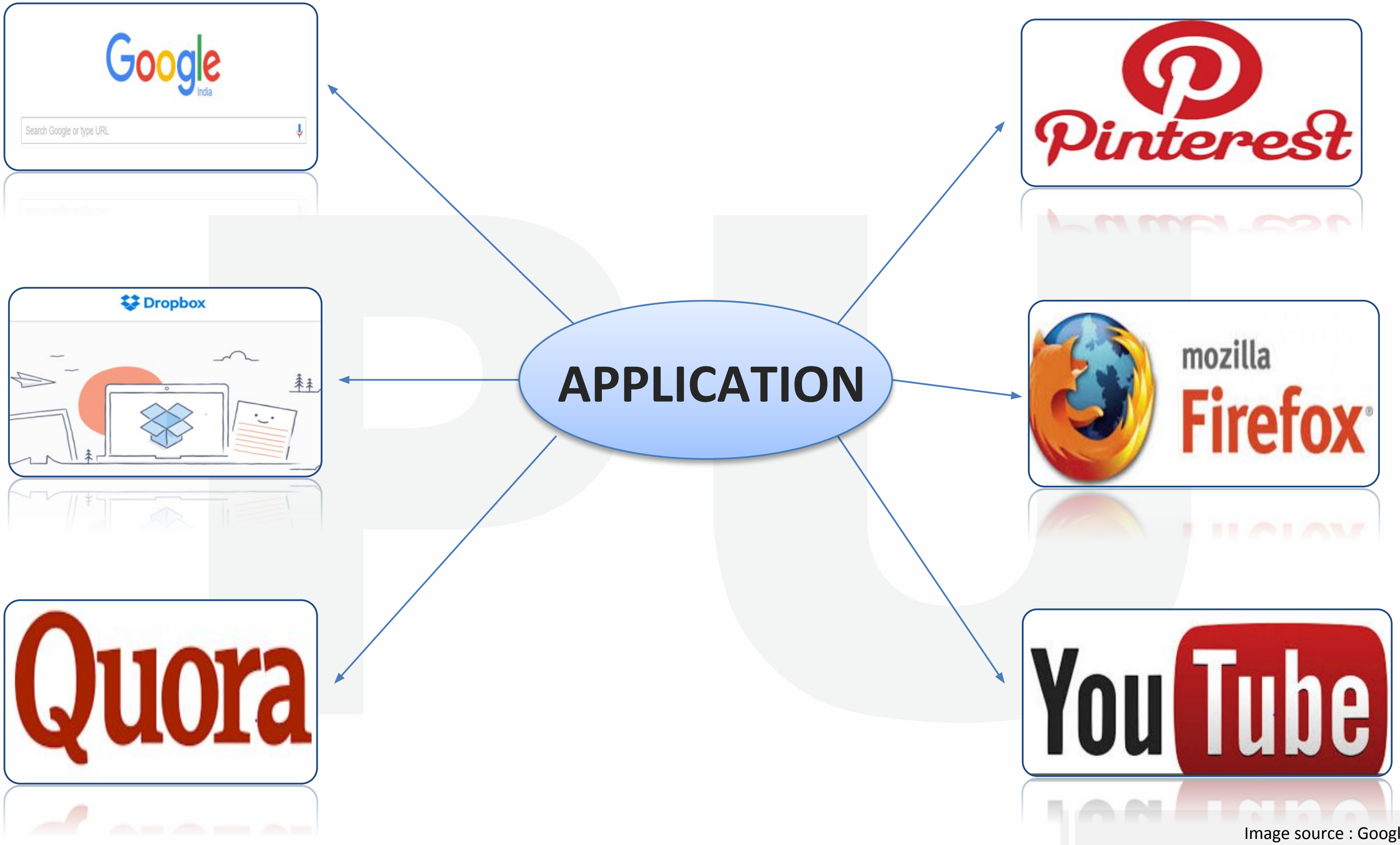


Image source : https://hackernoon.com/

Moral of the Python Story

That's Tony, our top programmer.
His code is always perfect and bug free!
I just don't know how he does it.



APPLICATION

Image source : Google

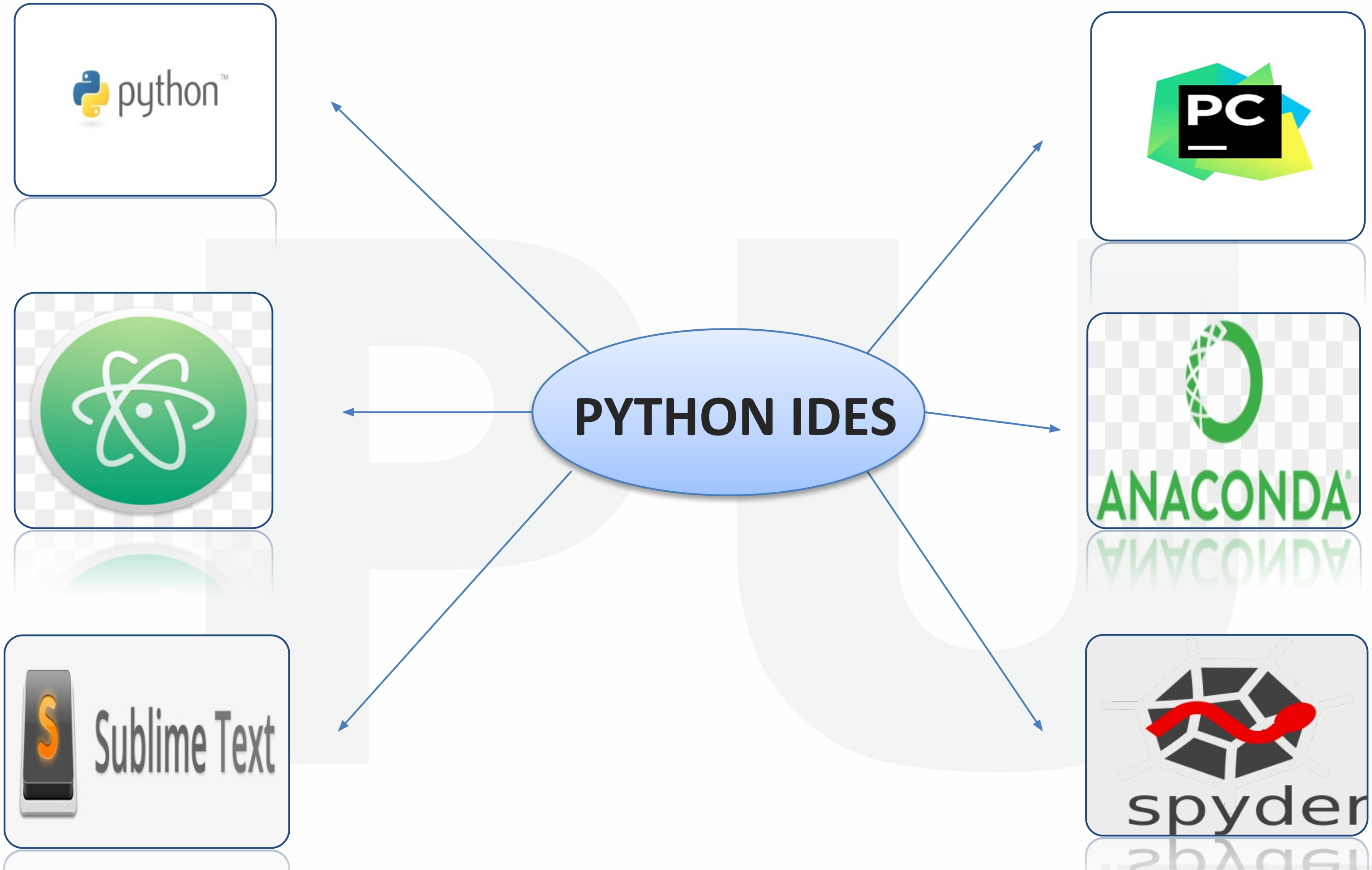
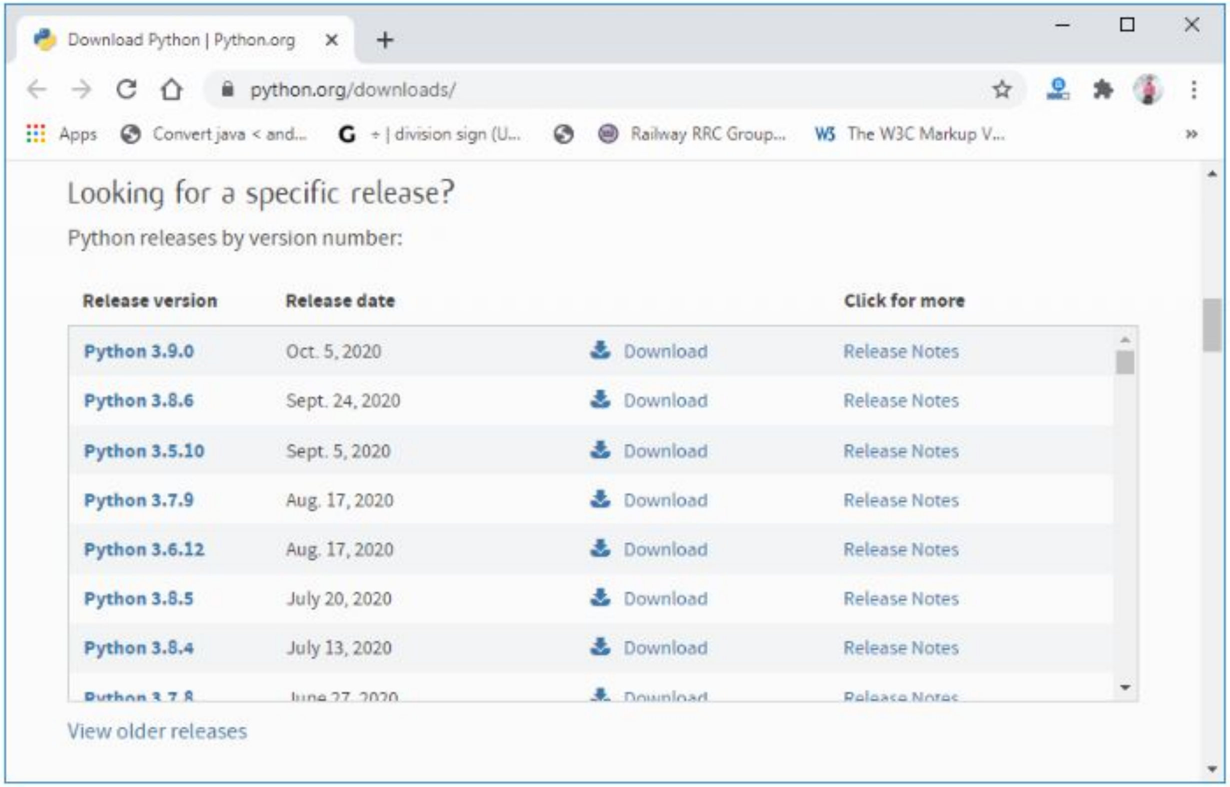


Image source : Google

How to Install Python

Step 1 – Select Version of Python to Install



Looking for a specific release?

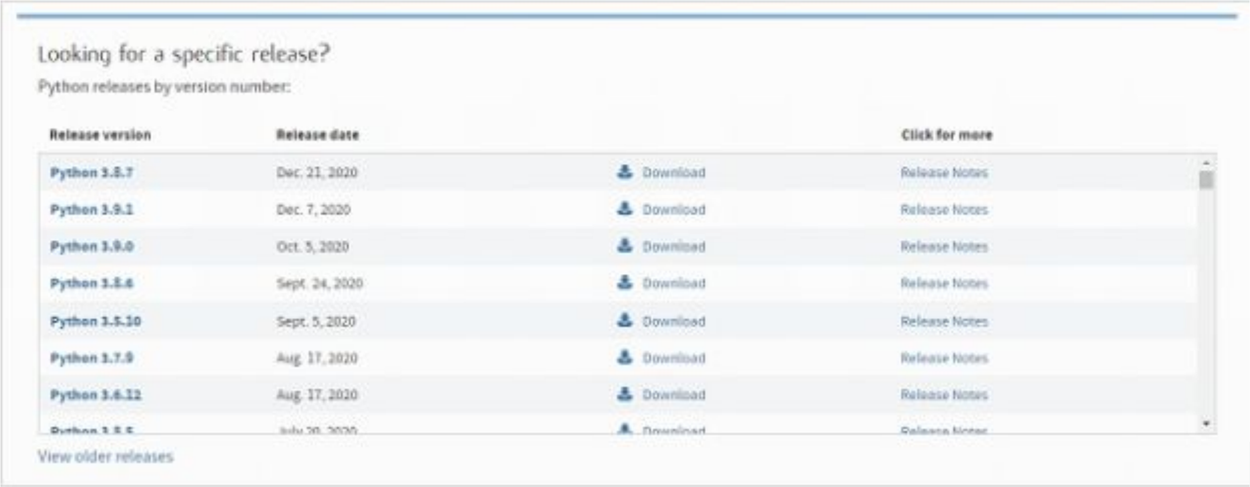
Python releases by version number:

Release version	Release date		Click for more
Python 3.9.0	Oct. 5, 2020	Download	Release Notes
Python 3.8.6	Sept. 24, 2020	Download	Release Notes
Python 3.5.10	Sept. 5, 2020	Download	Release Notes
Python 3.7.9	Aug. 17, 2020	Download	Release Notes
Python 3.6.12	Aug. 17, 2020	Download	Release Notes
Python 3.8.5	July 20, 2020	Download	Release Notes
Python 3.8.4	July 13, 2020	Download	Release Notes
Python 3.7.8	June 27, 2020	Download	Release Notes

View older releases

How to Install Python

Step 2 – Download Python Executable Installer



Looking for a specific release?

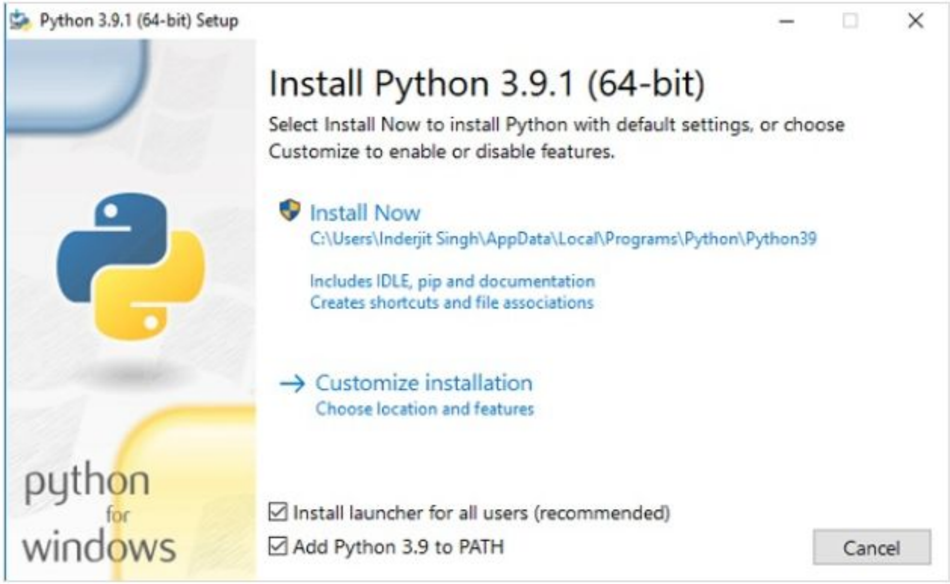
Python releases by version number:

Release version	Release date		Click for more
Python 3.8.7	Dec. 21, 2020	Download	Release Notes
Python 3.9.1	Dec. 7, 2020	Download	Release Notes
Python 3.9.0	Oct. 5, 2020	Download	Release Notes
Python 3.8.6	Sept. 24, 2020	Download	Release Notes
Python 3.5.10	Sept. 5, 2020	Download	Release Notes
Python 3.7.9	Aug. 17, 2020	Download	Release Notes
Python 3.6.12	Aug. 17, 2020	Download	Release Notes
Python 3.8.5	July 20, 2020	Download	Release Notes

View older releases

How to Install Python

Step 3 – Run Executable Installer



Python 3.9.1 (64-bit) Setup

Install Python 3.9.1 (64-bit)

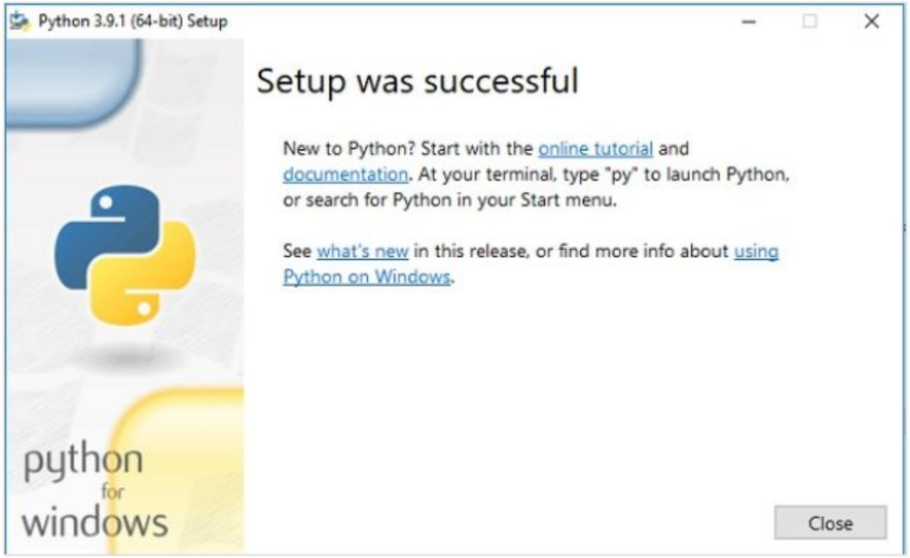
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Inderjit Singh\AppData\Local\Programs\Python\Python39

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.9 to PATH



Python 3.9.1 (64-bit) Setup

Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

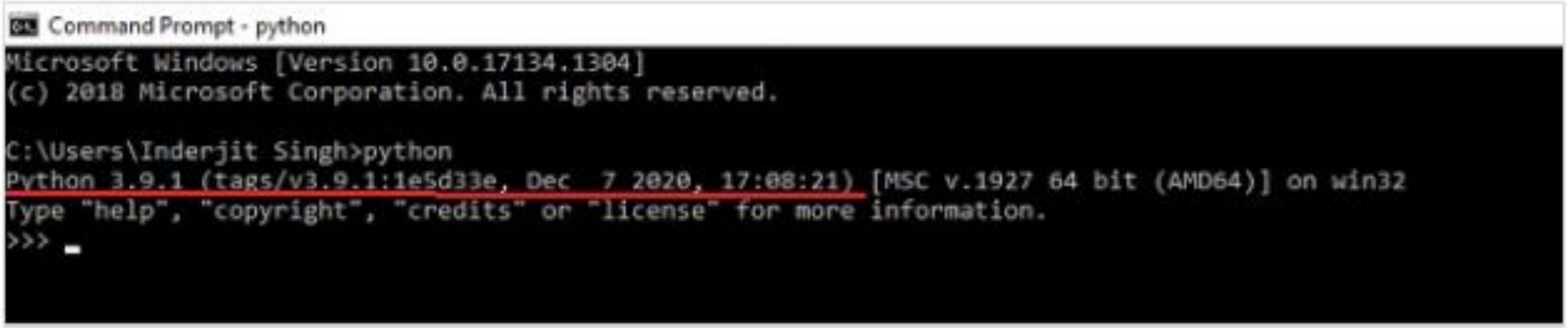
See what's new in this release, or find more info about using Python on Windows.

How to Install Python

Step 4 – Verify Python is installed on Windows

Open the command prompt.

- Type ‘python’ and press enter.
- The version of the python which you have installed will be displayed if the python is successfully installed on your windows.

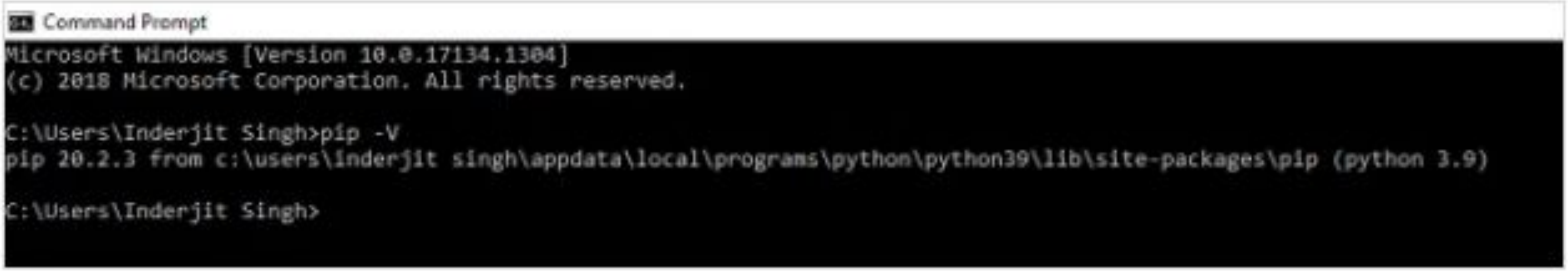


```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Inderjit Singh>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

How to Install Python

Step 5 – Verify Pip was installed

- Open the command prompt.
- Enter pip –V to check if pip was installed.
- The following output appears if pip is installed successfully.

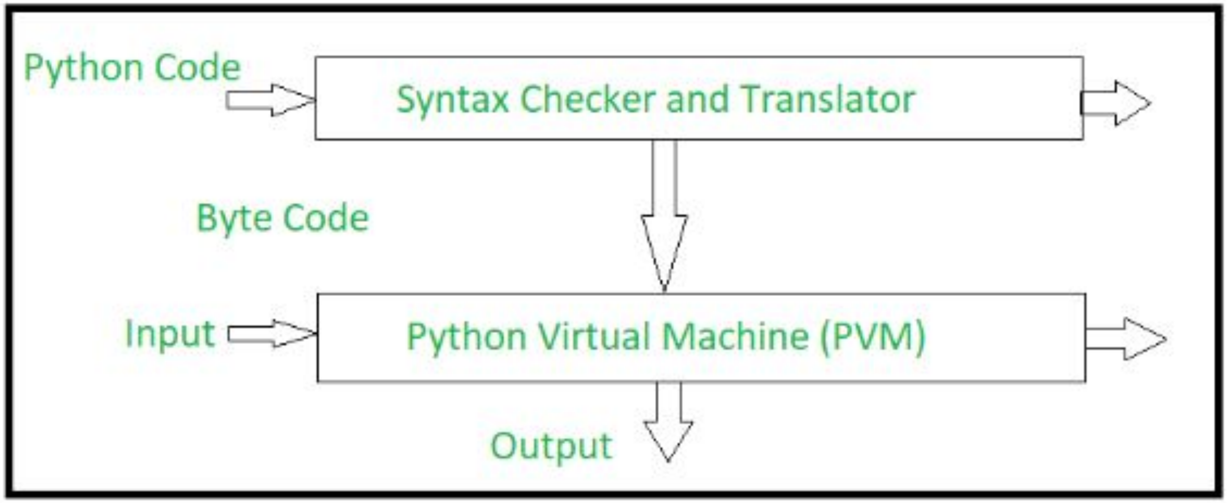


```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Inderjit Singh>pip -V
pip 20.2.3 from c:\users\inderjit singh\appdata\local\programs\python\python39\lib\site-packages\pip (python 3.9)

C:\Users\Inderjit Singh>
```

Python Interpreter and Its Working

- Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of Python.
- Python doesn’t convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So within python, compilation happens, but it’s just not into a machine language. It is into byte code (.pyc or .pyo) and this byte code can’t be understood by the CPU. So we need an interpreter called the python virtual machine to execute the byte codes.

Python Interpreter and Its Working



PU

Syntax and Semantics of Python Prog.

Syntax

- The syntax of a programming language refers to the order to which different elements are combined to from valid expressions. These elements may be words, operators, or phrases. The syntax of a programming language doesn't have any relationship with the meaning.

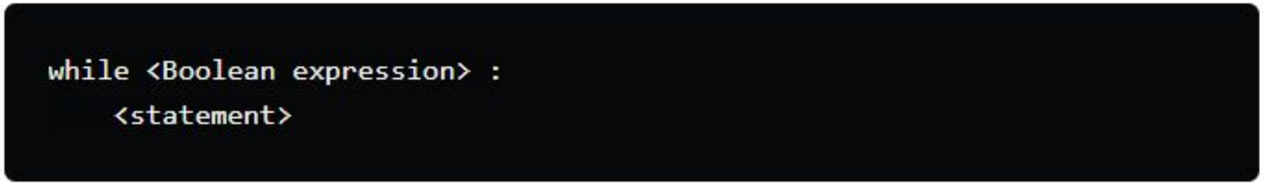
- An example of a syntax rule for programming is the assignment statement:

```
print(expression)
```

PU

Semantic

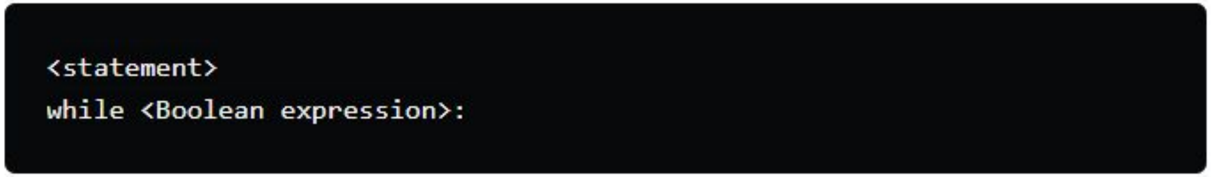
- Semantics emphasizes the meaning of a program, so it'll be understandable and easy to predict the outcome of execution.Semantics provides significant information needed to understand a program
  For example



```
while <Boolean expression> :
    <statement>
```

For the semantics, when the value of the boolean expression is met, the embedded statement would run.

Semantic



```
<statement>
while <Boolean expression>:
```

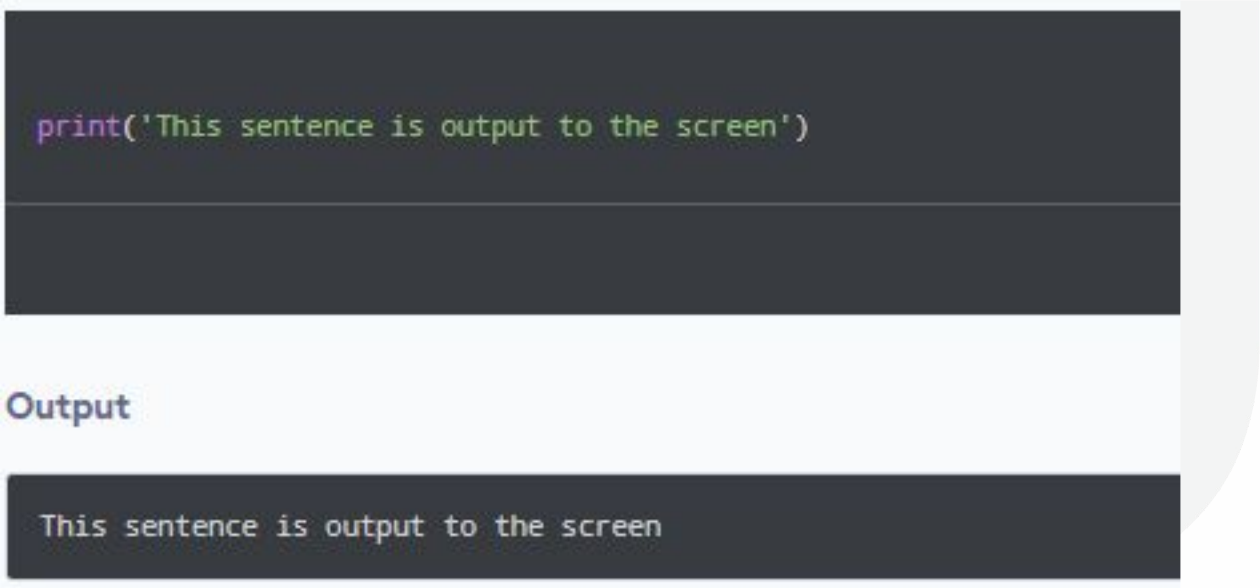
The code above has no valid meaning because, we placed the statement before the starting the While loop. In this case the syntax is correct, but the semantics is wrong.

PU

Input and Output in Python

Python Output Using print()

1. Example1



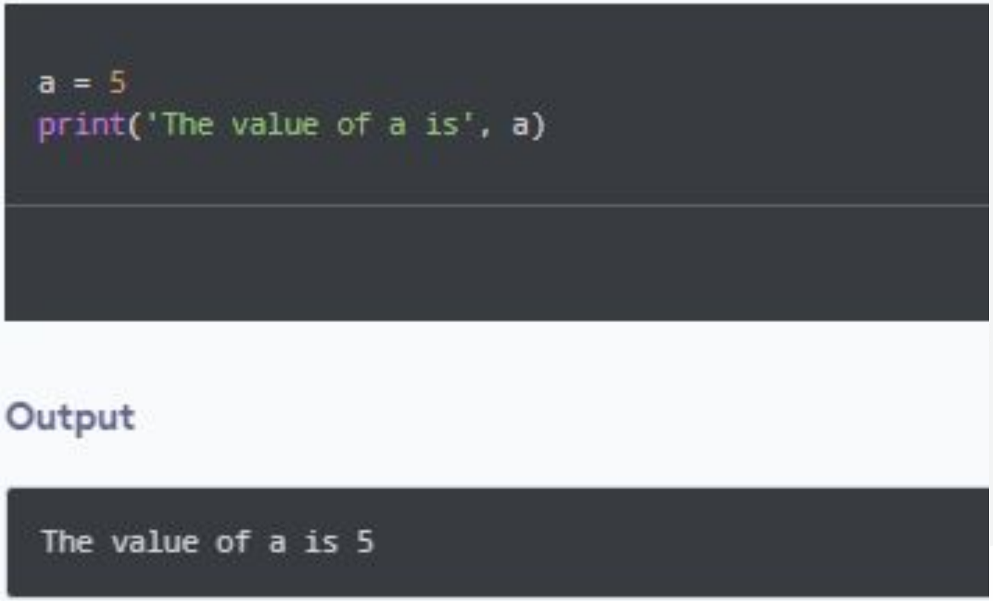
```
print('This sentence is output to the screen')
```

Output

```
This sentence is output to the screen
```

Input and Output in Python

Python Output Using print() Example2



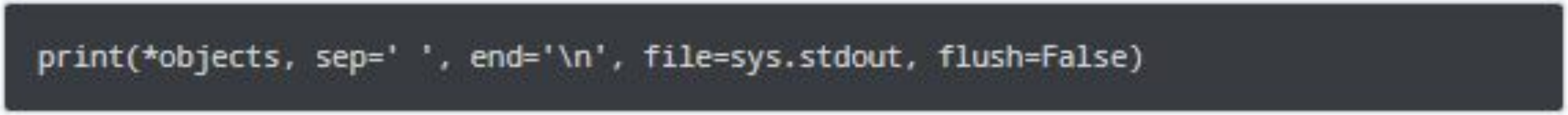
```
a = 5
print('The value of a is', a)
```

Output

```
The value of a is 5
```

Input and Output in Python

Python Output Using print() The actual syntax of the print() function is:

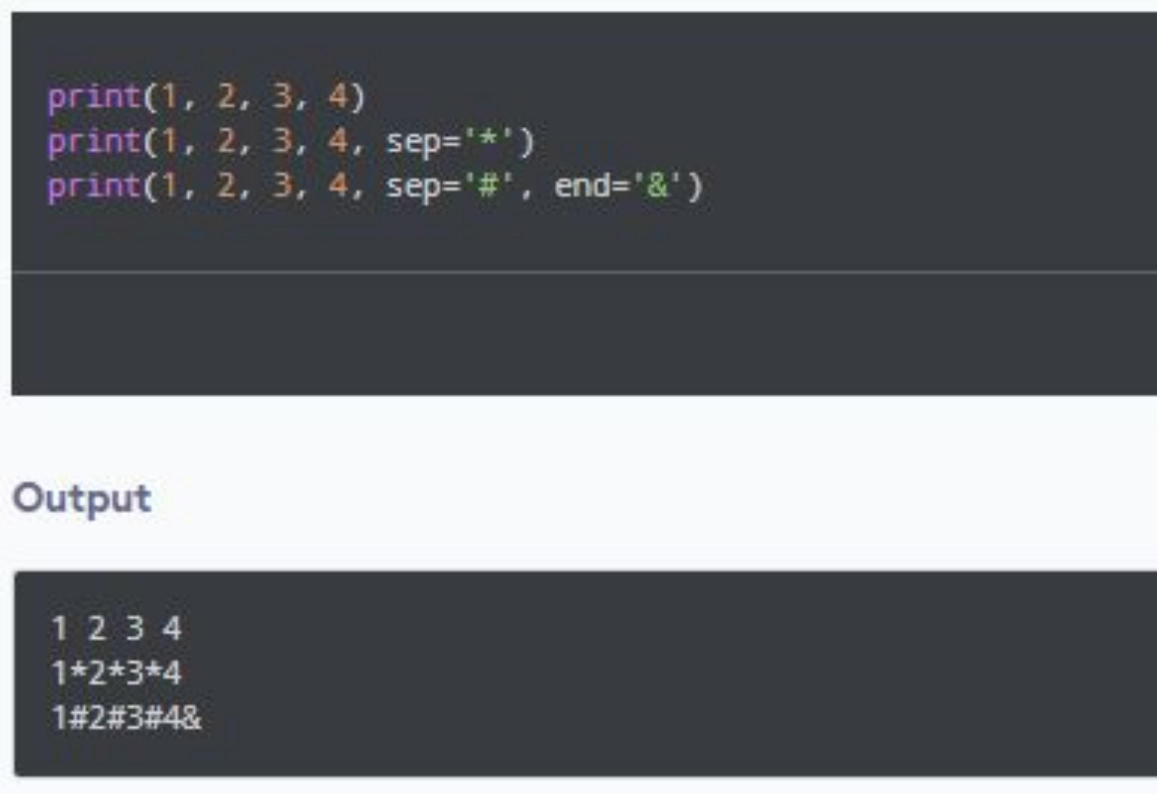


```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

- Here, ***objects*** is the value(s) to be printed.
- The ***sep*** separator is used between the values. It defaults into a space character.
- After all values are printed, ***end*** is printed. It defaults into a new line.
- The file is the object where the values are printed and its default value is ***sys.stdout*** (screen).

Input and Output in Python

Python Output Using print()



```
print(1, 2, 3, 4)
print(1, 2, 3, 4, sep='*')
print(1, 2, 3, 4, sep='#', end='&')
```

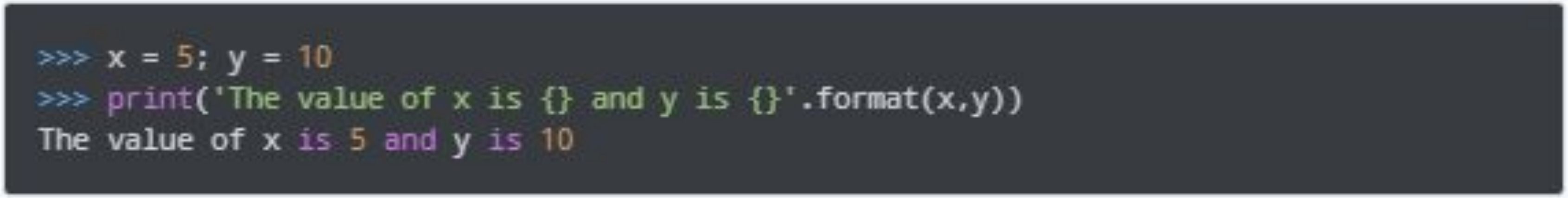
Output

```
1 2 3 4
1*2*3*4
1#2#3#4&
```

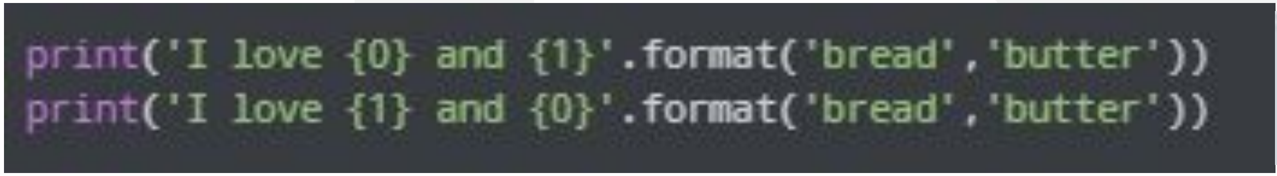
Input and Output in Python

Python Output Using print() Output formatting

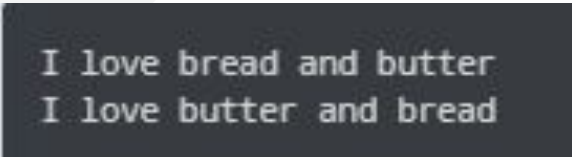
- Sometimes we would like to format our output to make it look attractive. This can be done by using the **str.format()** method. This method is visible to any string object.



```
>>> x = 5; y = 10
>>> print('The value of x is {} and y is {}'.format(x,y))
The value of x is 5 and y is 10
```



```
print('I love {0} and {1}'.format('bread','butter'))
print('I love {1} and {0}'.format('bread','butter'))
```

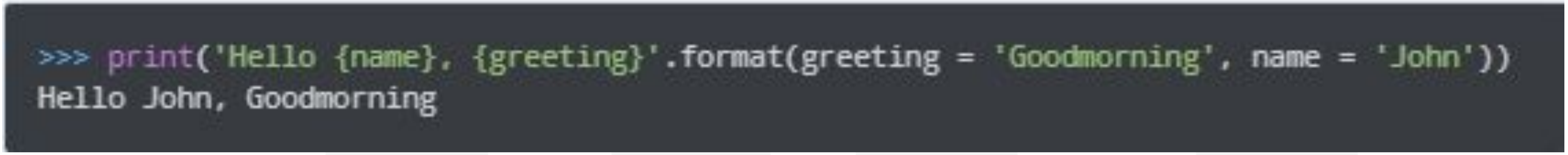


```
I love bread and butter
I love butter and bread
```

Input and Output in Python

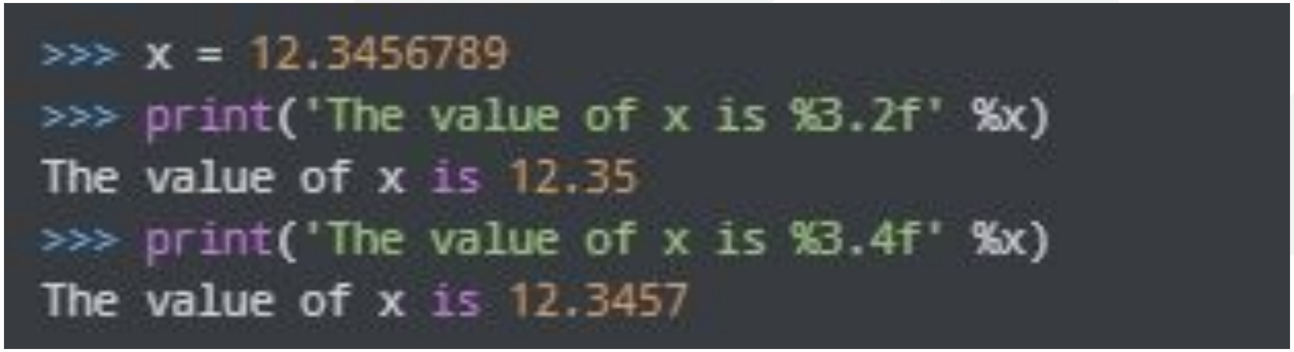
Python Output Using print()

We can even use keyword arguments to format the string



```
>>> print('Hello {name}, {greeting}'.format(greeting = 'Goodmorning', name = 'John'))
Hello John, Goodmorning
```

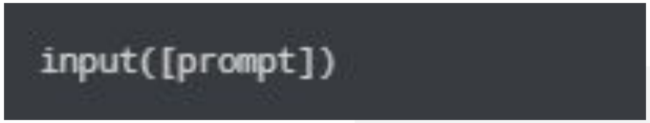
We can also format strings like the old printf() style used in C programming language. We use the % operator to accomplish this.



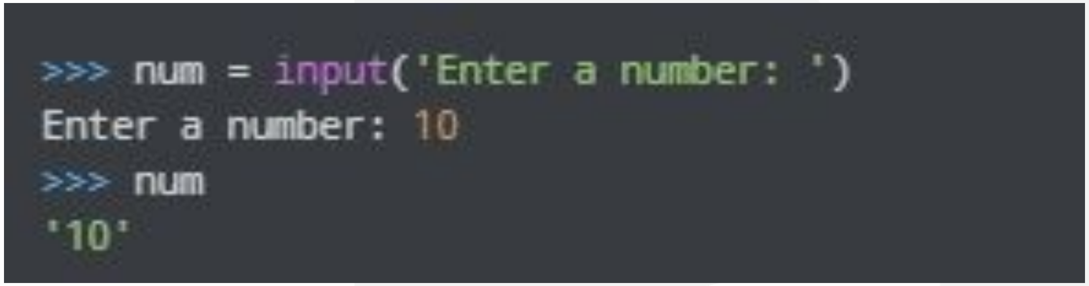
```
>>> x = 12.3456789
>>> print('The value of x is %3.2f' %x)
The value of x is 12.35
>>> print('The value of x is %3.4f' %x)
The value of x is 12.3457
```

Input and Output in Python

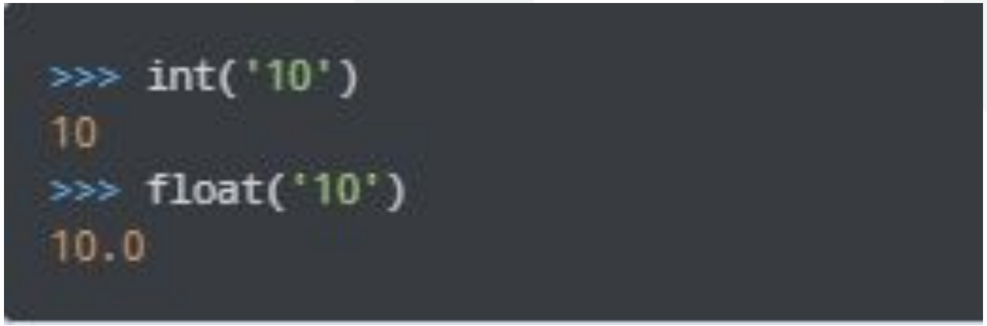
Python Input



```
input([prompt])
```



```
>>> num = input('Enter a number: ')
Enter a number: 10
>>> num
'10'
```

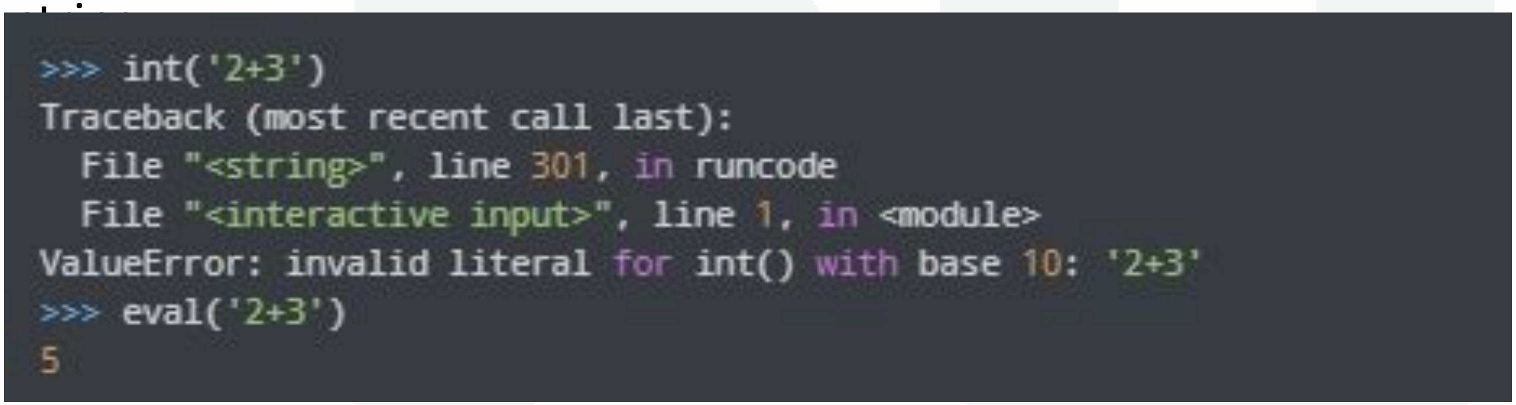


```
>>> int('10')
10
>>> float('10')
10.0
```

Input and Output in Python

Python Input

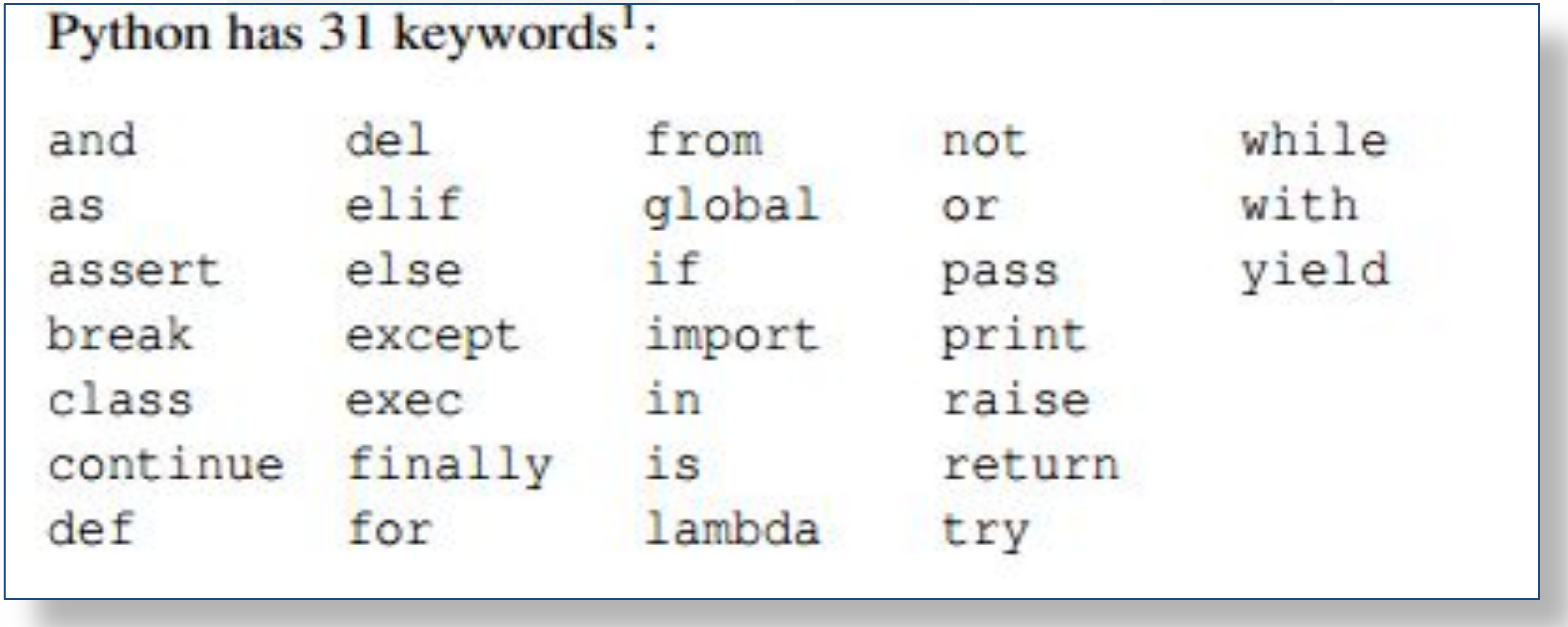
This same operation can be performed using the eval() function. But eval takes it further. It can evaluate even expressions, provided the input is a string.



```
>>> int('2+3')
Traceback (most recent call last):
  File "<string>", line 301, in runcode
  File "<interactive input>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '2+3'
>>> eval('2+3')
5
```

Reserved Words

- Reserved Words can not be used as identifiers name



Python has 31 keywords[1]:

and
as
assert
break
class
continue
def
del
elif
else
except
exec
finally
for
from
global
if
import
in
is
lambda
not
or
pass
print
raise
return
try
while
with
yield

Image source : http://python-nitol.blogspot.com/

Python Comments

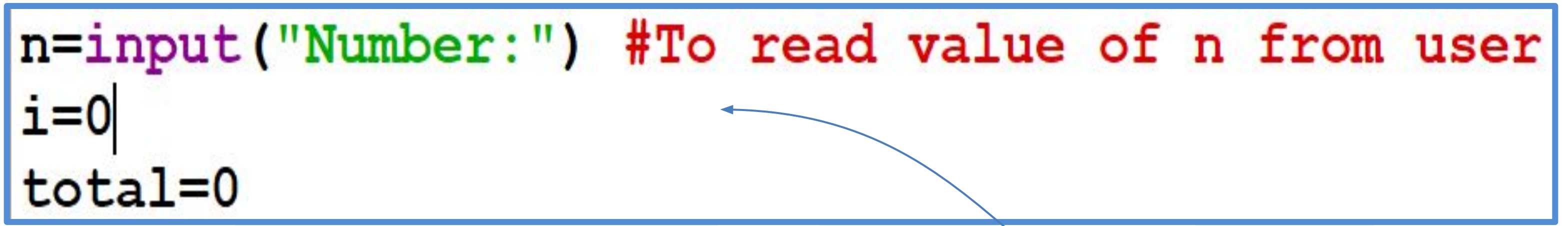
- Single line comment

  # this is single line comment

- Multiline Comments: Python does not provide the option for multiline comments. However, there are different ways through which we can write multiline comments.
  - **# Python program to demonstrate**
    # multiline comments
  - Python ignores the string literals that are not assigned to a variable so we can use these string literals as a comment.
  - **""" Python program to demonstrate**
    multiline comments"""

PU

Python Comments

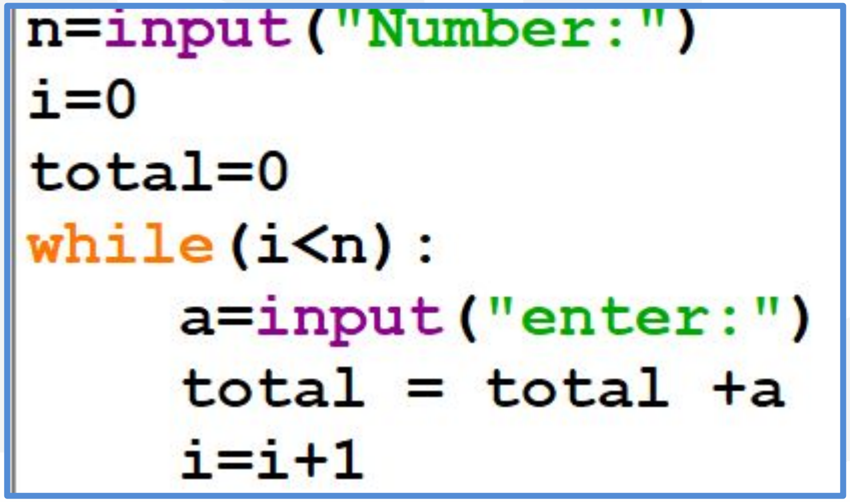


```
n=input("Number:") #To read value of n from user
i=0
total=0
```

Python comment

Indentation in Python

- Line indentation used to identify block of code



```
n=input("Number:")
i=0
total=0
while(i<n):
      a=input("enter:")
      total = total +a
      i=i+1
```

While Block

Identifier

- A name given to identify variables, user defined function, class and object

Rules

1. **Must start with letter or underscore**
2. **Must contain only letters, numbers or underscores**
3. **Case sensitive**
4. **Keywords are not allowed**

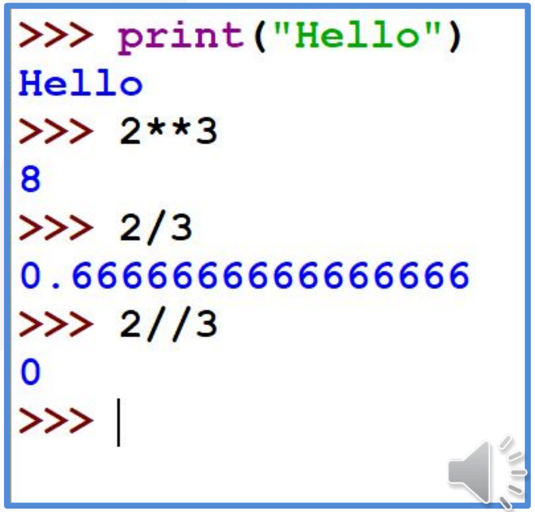
- **Valid : my_var _x var1**
- **Not Valid : my.var 2x var$**

Different : My_var my_var x X

Python As Interactive Shell

- Get immediate output after typing one python instruction
- Python Installation:
    - Get latest python release(3.0 OR above) from official website

http://www.python.org/download

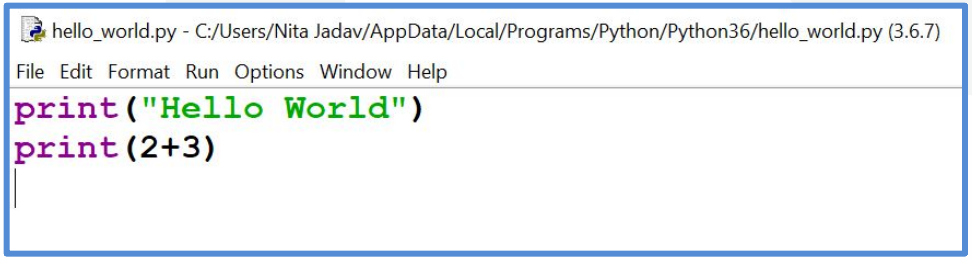


```
>>> print("Hello")
Hello
>>> 2**3
8
>>> 2/3
0.6666666666666666
>>> 2//3
0
>>> |
```

Python As Script

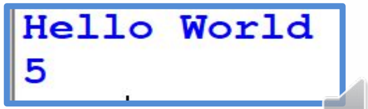
- Write sequence of statements into file and tell python to execute it
- Python file uses extension as '.py'
- Execute your python file from command prompt with command :

  '**python hello_world.py**'
- Create a script:
  - Open a file from your python idle 3.0
  - Write python statements
  - Execute file by choosing 'Run Module' or press F5



hello_world.py - C:/Users/Nita Jadav/AppData/Local/Programs/Python/Python36/hello_world.py (3.6.7)

```
print("Hello World")
print(2+3)
```



```
Hello World
5
```

Code Flow in Python

Python code is sequence of statements to perform task

Some statements can be selective based on conditions

Some statements can be iterative

Some group of statements may be used again at different places

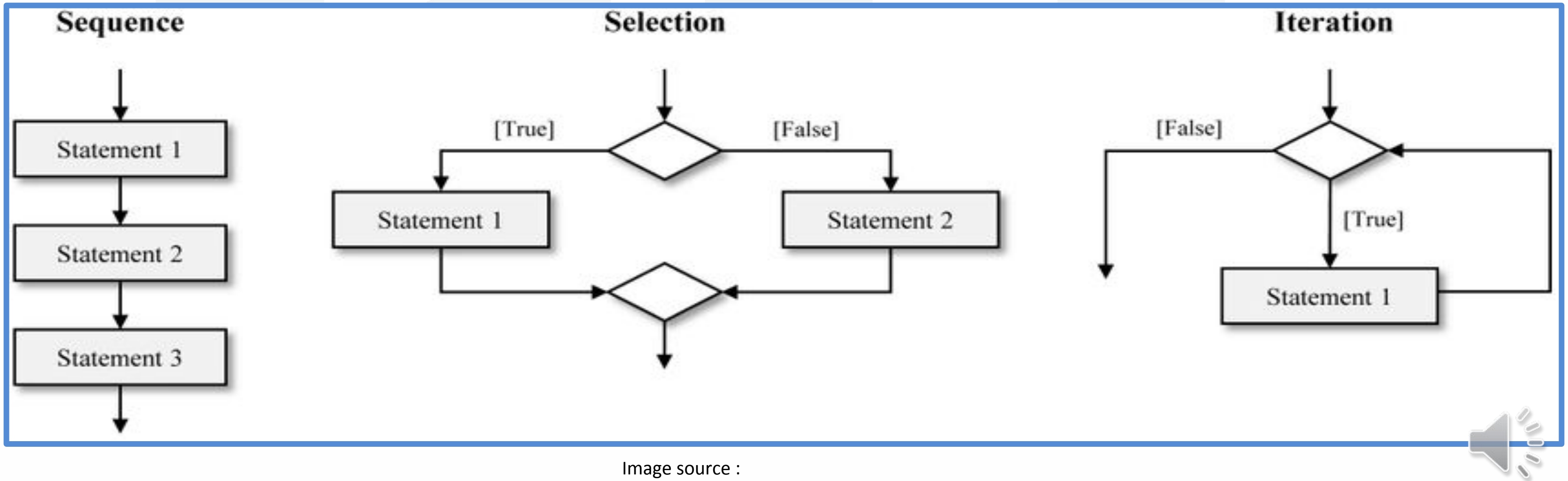
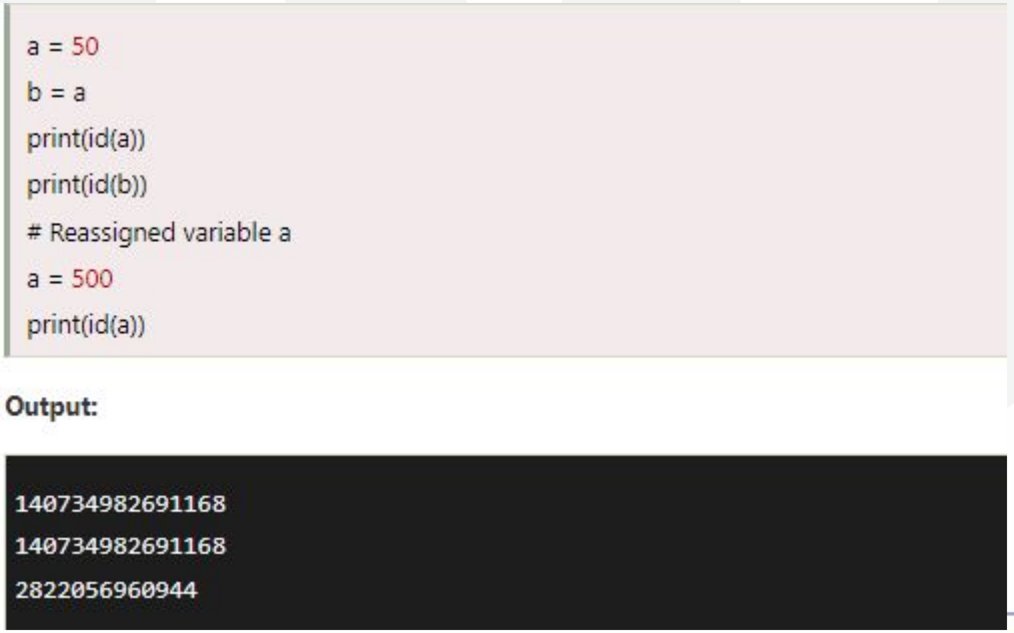


Image source : https://www.researchgate.net/

Python Variable

- Variable is a name that is used to refer to memory location. Python variable is also known as an identifier and used to hold value.
- In Python, variables are a symbolic name that is a reference or pointer to an object. The variables are used to denote objects by that name.



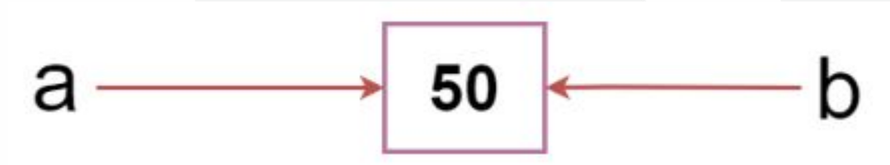
```
a = 50
b = a
print(id(a))
print(id(b))
# Reassigned variable a
a = 500
print(id(a))
```

Output:

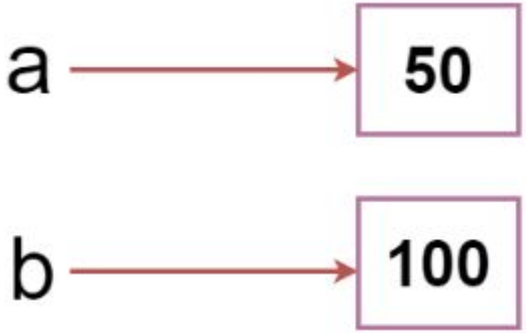
```
140734982691168
140734982691168
2822056960944
```

Python Variable

```
a = 50
b = a
```

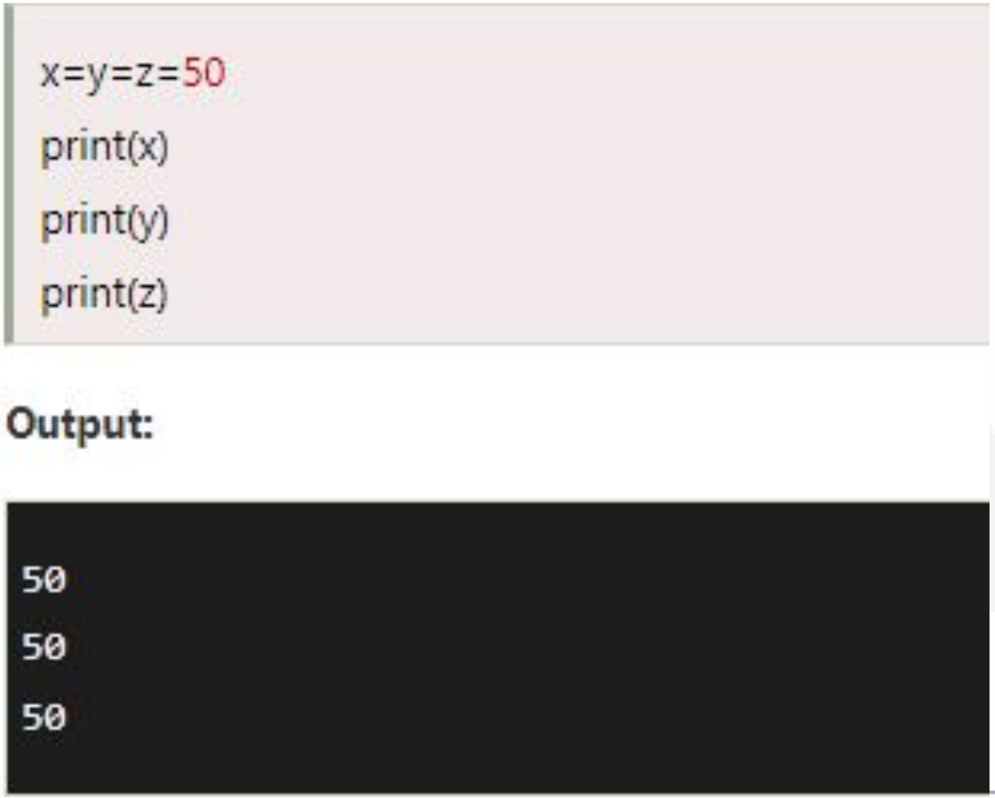


```
a = 50
b =100
```



Python Variable

- Assigning single value to multiple variables



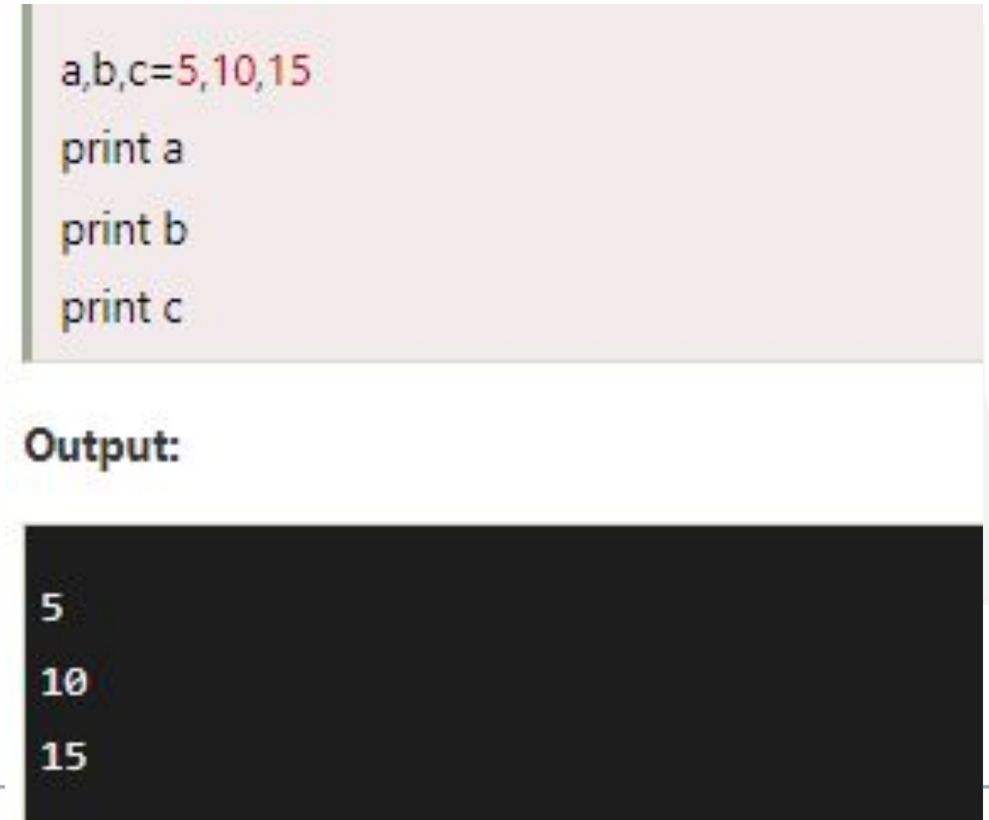
```
x=y=z=50
print(x)
print(y)
print(z)
```

Output:

```
50
50
50
```

Python Variable

- Assigning multiple values to multiple variables



```
a,b,c=5,10,15
print a
print b
print c
```

Output:

```
5
10
15
```

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in