**Python Basic Tutorial**

What is python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

**It is used for:**

Web development

Software development

Mathematics

System scripting

Computations and Analysis

**What can Python do?**

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and mo

dify files.

Python can be used to handle big data and perform complex mathematics.

Python can be used for rapid prototyping, or for production‐ready software development.

### **Why Python?**

It is Dynamically typed means no need to declare the variable types.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural way, an object‐orientated way or a functional way.

### **Python Syntax compared to other programming languages**

Python was designed to for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly‐brackets for this purpose.

# **Program­ 1: Initializing Variable**

In [2]: x = 10

print(x)

print(type(x))

10

<class 'int'>

# **Program ­2: Addition**

In [15]: x = 10

y = 12.45

sum = x+y

print(sum)

22.45

# 

# 

# 

# **Program ­3: Accessing Numbers**

In [17]: x = 2

y = 3.5

z = 3j

print(type(x))

print(type(y))

print(type(z))

<class 'int'>

<class 'float'>

<class 'complex'>

# **Program­ 4: integers**

In [18]: x = 1

y = 35656222554887711

z = -3255522

Print(type(x))

Print(type(y))

Print(type(z))

<class 'int'>

<class 'int'>

<class 'int'>

# 

# 

# **Program ­5: Float**

In [21]: x = 1.10

y = 1.0

z = ‐35.59

print(type(x))

print(type(y))

print(type(z))

<class 'float'>

<class 'float'>

<class 'float'>

# **Program­ 6: Float with 'e'(power of 10)**

In [80]: x = 35e3

y = 12E4

z = ‐87.7e100

print(type(x))

print(type(y))

print(type(z))

print(x)

print(y)

print(z)

<class 'float'>

<class 'float'>

<class 'float'>

35000.0

120000.0

‐8.77e+101

# **Program ­7.1: Complex Number Basic**

In [69]: x = 3+5j

y = 5j

z = ‐5j

print(type(x))

print(type(y))

print(type(z))

*#Addition of Two Complex Numbers*

print(x+y)

<class 'complex'>

<class 'complex'>

<class 'complex'> (3+10j)

3.0

5.0

(2‐3j)

# **Program­ 7.2: Complex number using complex()**

In [70]: # Creating complex number using complex() finction

z = complex(2, -3)

print(z)

(2-3j)

# **Program­ 7.3: Complex number addition using complex()**

In [71] : # Addition of two complex numbers using complex()

print(complex(2, 3) + complex(4, 5))

(6+8j)

# **Program­ 7.3: Complex number Division using complex()**

In [72]: print ( complex (2, 3) / complex(4, 5) )

(0.5609756097560976+0.0487804878048781j)

# **Program­ 7.4: Complex number Multiplication using complex()**

# In [74] : print ( complex (1, 3) \* complex (4, 1) )

(1+13j)

# **Program­ 7.5: Real & Imaginary value of Complex Numbers**

In [75] : #Showing the real part of the complex number after addition print(x.real)

#Showing the imaginary part of the complex number after addition print(x.imag)

3.0

5.0

# **Program­ 7.6: Complex Number Length**

In [ 77 ] : #Length of a cpmplex number or finding the absolute value of a complex number.

That is, sqrt[ i^2 + j^2 ]

print ( abs ( complex(3, 4) ) )

5.0

**Program-8: Casting (Integer)**

In [ 24 ] : x = int(1) # x will be 1

y = int(2.8) # y will be 2

z = int(“3”) # z will be 3

print (x)

print (y)

print (z)

1

2

3

# **Program­ 9: Casting(Float)**

In [ 25 ] : a = float(1) # x will be 1.0

b = float(2.8) # y will be 2.8

c = float(“3”) # z will be 3.0

d = float(“4.2”) # w will be 4.2

print(a)

print(b)

print(c)

print(d)

1.0

2.8

3.0

4.2

**Program 10 : Divmod(int)**

The divmod() built-in function returns both quotient and remainder

In [4] : x = 9

y = 3

z = divmod(x,y)

print(z)

(3,0)

# **Program ­11: DecToBin**

In [34] : x = 1223

y = bin(x)

print(y)

0b10011000111

# **Program­ 12: DecToOct**

In [11] : x = 1223

y = oct(x)

print(y)

0o2307

# **Program­ 13: DecToHex**

In [13] : x = 1223

y = hex(x)

print(y)

0x4c7

# **Program­ 14: HexToDec**

In [35] : x = 0x4c7

y = int(x)

print(y)

1223

# **Program­ 15: DecToOct**

In [37] : x = 0o2307

y = int(x)

print(y)

1223

# **Program­ 16: BinToDec**

In [38] : x = 0b10011000111

y = oct(x)

print(y)

1223

# 