Python Essentials

Assignment I

Dhananjay Porwal
EN19CS301110

1 Introduce Python.

Ans Python is a widely used general-purpose, high level programming language. It was created by Guido von Rossum in 1991 & further developed by the Python Software foundation. It was designed with an emphasis on code readability, & its syntax allows programmers to express their concepts in fewer lines of code.

Their are two major Python versions: Python 2 & Python 3. Both are quite different. features of python are -

- Works on dif platforms (Win / Mac / Linux / Raspberry Pi.)
- Syntax is similar to English language.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written.
- Python can be treated in a procedural way,

an object-oriented way or a functional way.

2 Explain Basic data types with example?

Ans

Data Types	Examples	Explanation	Is mutable
• Strings	"Hello", "12.1"	Text - anything b/w " " becomes string	Immutable
• Integers	5364	Whole numbers	Immutable
• Floats	3.1415	Decimal Numbers	Immutable
• Booleans	True, false	Truth values that represent Yes/No	Immutable
• Lists	[1,2,3,4]	A collection of data, sits between []	Mutable
• Tuples	(1,2,3,4)	A collection of data, sits between ()	Immutable
• Dictionaries	{"a":1, "b":2}	A collection of data, sits between {}	Mutable

3 Application of Python?

Ans

- Web & Internet Development
  a) Frameworks such as Django & Pyramid.
  b) micro-frameworks such as Flask & Bottle.

- Scientific & Numeric

a) SciPy is a collection of package for mathematics, science, & engineering.

b) Pandas is a data analysis & modelling library.

- Desktop GUIs

a) Tk library is included with most binary distro. of python.

b) Platform-specific toolkits are also available like GTK+, etc.

- Software Development

a) Scons for build control.

b) Roundup or Taiac for bug tracking & project management.

- Business Applications

a) Python is also used to build ERP & e-commerce systems.

b) Tryton is a three-tier high-level general purpose application platform.

4. **Implement 5 programs on operator.**

Arithmetic Operators

```
# Examples of Arithmetic Operator

a = 9
b = 4

# Addition of numbers
add = a + b

# Subtraction of numbers
sub = a - b

# Multiplication of number
mul = a * b

# Division(float) of number
div1 = a / b

# Division(floor) of number
div2 = a // b

# Modulo of both number
mod = a % b

# Power
p = a ** b

# print results
print(add)
print(sub)
print(mul)
print(div1)
print(div2)
print(mod)
print(p)
```

Comparison Operators

```
# Examples of Relational Operators
a = 13
```

```
b = 33

# a > b is False
print(a > b)

# a < b is True
print(a < b)

# a == b is False
print(a == b)

# a != b is True
print(a != b)

# a >= b is False
print(a >= b)

# a <= b is True
print(a <= b)
```

Logical Operators

```
# Examples of Logical Operator

a = True
b = False

# Print a and b is False
print(a and b)

# Print a or b is True
print(a or b)

# Print not a is False
print(not a)
```

Bitwise Operators

```
# Examples of Bitwise operators

a = 10
b = 4

# Print bitwise AND operation
```

```
print(a & b)

# Print bitwise OR operation
print(a | b)

# Print bitwise NOT operation
print(~a)

# print bitwise XOR operation
print(a ^ b)

# print bitwise right shift operation
print(a >> 2)

# print bitwise left shift operation
print(a << 2)
```

Assignment Operators

```
# Examples of Assignment Operators

a = 10

# Assign value
b = a
print(b)

# Add and assign value
b += a
print(b)

# Subtract and assign value
b -= a
print(b)

# multiply and assign
b *= a
print(b)

# bitwise lishift operator
b <<= a
print(b)
```

Identity Operators

```
a = 10

b = 20
c = a

print(a is not b)
print(a is c)
```

Membership Operators

```
# Python program to illustrate

# not 'in' operator
x = 24
y = 20
list = [10, 20, 30, 40, 50]

if (x not in list):
print("x is NOT present in given list")
else:
print("x is present in given list")

if (y in list):
print("y is present in given list")
else:
print("y is NOT present in given list")
```

Precedence and Associativity of Operators

```
# Examples of Operator Precedence


# Precedence of '+' & '*'
expr = 10 + 20 * 30
print(expr)

# Precedence of 'or' & 'and'
name = "Alex"
age = 0

if name == "Alex" or name == "John" and age >= 2:
print("Hello! Welcome.")
else:
```

```
print("Good Bye!!")
```

Operator Associativity

```
# Examples of Operator Associativity


# Left-right associativity
# 100 / 10 * 10 is calculated as
# (100 / 10) * 10 and not
# as 100 / (10 * 10)
print(100 / 10 * 10)

# Left-right associativity
# 5 - 2 + 3 is calculated as
# (5 - 2) + 3 and not
# as 5 - (2 + 3)
print(5 - 2 + 3)

# left-right associativity
print(5 - (2 + 3))

# right-left associativity
# 2 ** 3 ** 2 is calculated as
# 2 ** (3 ** 2) and not
# as (2 ** 3) ** 2
print(2 ** 3 ** 2)
```

5 Explain Variable, print(), input(), range().

Ans • Variable - A variable is created the moment you first assign a value to it.

for example

```
x = 5
y = "John"
```

• print() - Prints to the Standard output device.

for example

```
print("Hello World")
```

• input() - Allowing user input

for example

```
x = input()
```

- range() - Returns a sequence of numbers, starting from 0 & increment by 1 (by default)

For example

```
x = range(6)
for n in x:
    print(n)
```