

----: Numeric in Python:----

The int data type represents values or numbers without decimal values. In python, there is no limit for the int data type. It can store very large values conveniently.

```
a=10
```

```
type(a) O/P:- <class 'int'>
```

Note: In Python 2nd version long data type was existing but in python 3rd version long data type was removed.

We can represent int values in the following ways

1. Decimal form (**bydefault**)
2. Binary form
3. Octal form
4. Hexa decimal form

1. Decimal form(base-10):

It is the default number system in Python. The allowed digits are: 0 to 9

Ex: a =10

2. Binary form(Base-2):

The allowed digits are : 0 & 1

Literal value should be prefixed with 0b or 0B

Eg: a = 0B1111

a =0B123

a=b111

3. Octal Form(Base-8):

The allowed digits are : 0 to 7

Literal value should be prefixed with 0o or 0O.

Ex: a=0o123

a=0o786

4. Hexa Decimal Form(Base-16):

The allowed digits are : 0 to 9, a-f (both lower and upper cases are allowed)

Literal value should be prefixed with 0x or 0X.

Ex: a=0X9FcE

a=0x9aDF

Note: Being a programmer we can specify literal values in decimal, binary, octal and hexa

decimal forms. But PVM will always provide values only in decimal form.

```
# binary data type(Base-2)
x= 0b1111
y= 0B1010

# by default converted into decimal
print(x) # O/P-15
print(y) # O/P-10

# octal data type(Base-8)
x=0o765
y=0O542
# by default converted into decimal
print(x) # O/P-501
print(y) # O/P-354

# decimal data type(Base-10)---default
x=10
y=50
# by default converted into decimal
print(x) # O/P-10
print(y) # O/P-50

# Hexadecimal data type(Base-16)
x=0X8EA
z=0x5eA
# by default converted into decimal
print(x) # O/P-2282
print(y) # O/P-50
```

Base Conversions:--- Python provide the following in-built functions for base conversions

```

# Base Conversions
# bin()
print(bin(15)) # o/p- 0b1111
print(bin(0o11)) # o/p- 0b1001
print(bin(0X10)) # o/p- 0b10000

# oct()
print(oct(10)) # o/p-0o12
print(oct(0B1111)) # o/p-0o17
print(oct(0X123)) # o/p-0o443

# hex()
print(hex(100)) # o/p-0x64
print(hex(0B111111)) # o/p-0x3f
print(hex(0o12345)) # o/p- 0x14e5

```

Float Data Type in Python:

The float data type represents a number with decimal values. floating-point numbers can also be written in scientific notation. e and E represent exponentiation. where e and E represent the power of 10. For example, the number $2 * 10^{pow2}$ is written as 2E2, such numbers are also treated as floating-point numbers.

```

salary = 50.5
print(salary)
print(type(salary))

O/P---
50.5
<class 'float'>

```

Example: Print float values

```

a = 2e2 # 2*10^2 e stands for 10 to the power
b = 2E2 # 2*10^2
c = 2e3 # 2*10^3

```

```
d = 2e1
print(a)
print(b)
print(c)
print(d)
print(type(a))
```

O/P----

200.0

200.0

2000.0

20.0

<class 'float'>

Complex Data Type in python:

The complex data type represents the numbers that are written in the form of $a+bj$ or $a-bj$, here a is representing a real part of the number and b is representing an imaginary part of the number. The suffix small j or upper J after b indicates the square root of -1 . The part “ a ” and “ b ” may contain integers or floats.

```
a = 3+5j
b = 2-5.5j
c = 3+10.5j
print(a)
print(b)
print(c)
print()
print("A+B=",a+b)
print("B+C=",b+c)
print("C+A=",c+a)
print("A*B=",a*b)
print("B*C=",b*c)
print("C*A=",c*a)
print("A+B+C=", a+b+c)
print("A/B=",a/b)
```

O/P---

(3+5j)

(2-5.5j)

(3+10.5j)

$$A+B = (5-0.5j)$$

$$B+C = (5+5j)$$

$$C+A = (6+15.5j)$$

$$A*B = (33.5-6.5j)$$

$$B*C = (63.75+4.5j)$$

$$C*A = (-43.5+46.5j)$$

$$A+B+C = (8+10j)$$

$$A/B = (-0.6277372262773723+0j)$$