

# Data Types

---

Data Type represent the type of data present inside a variable.

In Python we are not required to specify the type explicitly. Based on value provided, the type will be assigned automatically. Hence Python is Dynamically Typed Language.

## Data types:

### 1. Numeric

#### 1. Integer

#### 2. Float

#### 3. Complex

### 2. Mapped

#### 1. Dictionary

#### 3. Unordered

#### 1. Set

#### 2. Frozenset

### 4. Boolean

## Fundamental Data Types in Python

In Python, the following data types are considered as Fundamental Data types,

#### 1. Int

#### 2. Float

#### 3. Complex

#### 4. Bool

#### 5. Str

## Integer data type

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

## Example

```
a=10
print(a)
print(type(a))
```

*# Output*

```
10
<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 (by default)**
2. **Binary**
3. **Octal**
4. **Hexa decimal**

### **Decimal(base-10):**

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

Example:

```
x=10
```

### **Binary(Base-2):**

The allowed digits are : 0 & 1

Literal value should be prefixed with 0b or 0B

Example:

```
a = 0B1111
a = 0B123
a = b111
```

### **Octal (Base-8):**

1. The allowed digits are : 0 to 7
2. Literal value should be prefixed with 0o or 0O.

Example:

```
a=0o123
a=0o786
```

### **Hexa Degits(Base-16):**

1. The allowed digits are : 0 to 9, a-f (both lower and upper cases are allowed)
2. Literal value should be prefixed with 0x or 0X.

Example:

```
a=0X9FcE
a=0x9aDF
```

## Base Conversions

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

Example:

```
# bin()
print(bin(15))
print(bin(0o11))
print(bin(0X10))

# oct()
print(oct(10))
print(oct(0B1111))
print(oct(0X123))

# hex()
print(hex(100))
print(hex(0B111111))
print(hex(0o12345))

# Output
0b1111
0b1001
0b10000

0o12
0o17
0o443

0x64
0x3f
0x14e5
```

## Float Data Type:

The float data type denotes 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  $2 * 10^2$  is written as 2E2, such numbers are also treated as floating-point numbers.

Example:

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

*# Output*

```
50.5
<class 'float'>
```

Example:

```
a = 2e2 # 2*10^2  e stands for 10 to the power
b = 2E2 # 2*10^2
c = 2e3 # 2*10^3
d = 2e1 # 2*10^1
```

```
print(a)
print(b)
print(c)
print(d)
print(type(a))
print(type(b))
print(type(c))
print(type(d))
```

*# Output*

```
200.0
200.0
2000.0
20.0
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
```

## Complex Data Type

The complex data type denotes 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.

Example:

```
a = 3+5j
b = 2-5.5j
c = 3+10.5j
print(a)
print(b)
print(c)
print(type(a))
print(type(b))
print(type(c))
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)
```

*# Output*

```
(3+5j)
(2-5.5j)
(3+10.5j)
<class 'complex'>
<class 'complex'>
<class 'complex'>
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+0.7737226277372262j)
```