# Operators

## Unary Operators

- +, -, , /, //, * and % are known as operators and these operators can be unary or binary
- A unary operator has only one operand
- The unary - (minus) operator yields the negation of its numeric argument
- The unary + (plus) operator yields its numeric argument unchanged
- The unary ~ (invert) operator yields the bit-wise inversion of its plain or long integer argument

### Example

### The - (minus) operator is used to negate any positive number

In [1]:

```
-15      # in this case the - (minus) operator is acting as a unary operator
```

Out[1]:

```
-15
```

In [2]:

```
100 - 40 # The -(minus) operator is acting as a binary operator
```

Out[2]:

```
60
```

## Identity operators

- Identity operators are used to compare if two objects are same with the same memory location
- They are usually used to determine the data type of a variable
- The identity operators are 'is' and 'is not'

### ‘is’ operator - evaluates to true if the variables on either side of the operator point to the same object and false otherwise

In [3]:

```
a = 15
if (type(a) is float):
    print ("true")
else:
    print ("false") # returns false because the data type of 'a' is 'int' not 'float'
```

```
false
```

### ‘is not’ operator - evaluates to false if the variables on either side of the operator point to the same object and true otherwise

In [4]:

```
b = 15.6
if (type(b) is not float):
    print ("true")
else:
    print ("false") # returns false because the data type of 'b' is 'float' not 'int'
```

```
false
```

## Membership Operators

- Membership operators are operators used to validate the membership of a value in a sequence
- The membership operators are 'in' and 'not in'

### 'in' operator - checks if a value exists in a sequence or not

### It evaluates to true if it finds a variable in the specified sequence and false otherwise

In [5]:

```
x = [1,2,3,4,5]

print(4 in x) # returns True because 4 exists in the x
```

```
True
```

### ‘not in’ operator - evaluates to true if it does not find a variable in the specified sequence and false otherwise

In [6]:

```
y = [1,2,3,4,5]

print(8 not in y) # returns True because 8 doesn't exists in the y
```

```
True
```

# Bitwise Operators

| Operator | Name | Description | Syntax |
|---|---|---|---|
| & | Bitwise AND | Sets each bit to 1 if both bits are 1 | x & y |
| \| | Bitwise OR | Sets each bit to 1 if one of two bits is 1 | x \| y |
| ~ | Bitwise NOT | Inverts all the bits | ~x |
| ^ | Bitwise XOR | Sets each bit to 1 if only one of two bits is 1 | x ^ y |
| >> | Bitwise right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off | x>> |
| << | Bitwise left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off | x<< |

# END OF SCRIPT