# Operators

- they are some special characters or reserve words used to perform various operations

**Types of Operators**

1. Arithmetic operators
2. Assignment operators
3. Comparison (Relational) operators
4. Logical (Boolean) operators
5. Membership Operators
6. Identity Operators

**Arithematic Operators**

- Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.
- `+ , -, *, /, %, //, **  are arithmetic operators`

> **Arthimetic operators Precedence**

- Paracentheis
- exponents
- floor division
- Multiplication or Division or Moduluo (left to right)
- Addition or Subtraction (left to right)



In [1]:

```
#addition
10+4
```

Out[1]:

14

In [2]:

```
#subtraction
10-4
```

Out[2]:

6

In [3]:

```
#multiplication
10*4
```

Out[3]:

```
40
```

In [4]:

```
#division-- always gives output as float
8 / 4
```

Out[4]:

```
2.0
```

In [5]:

```
# floor division -- computes the quotient
9 // 5
```

Out[5]:

```
1
```

In [6]:

```
# modulo - computes the remainder
1 % 6
```

Out[6]:

```
1
```

In [7]:

```
# exponent
10 ** 4
```

Out[7]:

```
10000
```

In [8]:

```
#Paracentheis
(2+3)*5
```

Out[8]:

```
25
```

**When we use arthimetic operators, the boolean values will be automatically converted to int**

In [9]:

```
True + True
```

Out[9]:

```
2
```

In [10]:

```
b = 3.9
c = False
b*c
```

Out[10]:

```
0.0
```

In [11]:

```
b = 3
c = False
b/c
```

```
---------------------------------------------------------------------
-
ZeroDivisionError                         Traceback (most recent call las
t)
Cell In[11], line 3
      1 b = 3
      2 c = False
----> 3 b/c

ZeroDivisionError: division by zero
```



**Assignment Operators**

- Assignment operators are used in Python to assign values to variables.
- (=, +=, -=, *=, /=, %=, //=, **= ) are Assignment operators
- First right side part will be executed and then value will be assigned to the left side variable

In [12]:

```
x = 5
print(x)
```

```
5
```

In [13]:

```
x = 5
x = x+10
print(x)
```

15

In [14]:

```
x=5
x+=10
print(x)
```

15

In [15]:

```
x=5
x*=10
print(x)
```

50

**Comparison/Relational Operators**

- Comparison operators are used to compare values.
- It either returns True or False according to the condition.
- ==, !=, >, <, >=, <= are comparision operators

In [16]:

```
5 == 5.0
```

Out[16]:

True

In [17]:

```
3 != 3
```

Out[17]:

False

In [18]:

```
5>3
```

Out[18]:

True

In [19]:

```
3>=3
```

Out[19]:

True

In [20]:

```
3<1
```

Out[20]:

False

In [21]:

```
3<=1
```

Out[21]:

False

In [22]:

```
45==45.00000
```

Out[22]:

True

In [23]:

```
"srk" == "SRK"
```

Out[23]:

False



**Logical Operators / Boolean Operators**

- Logical Operators are used to check multiple conditions at a time.
- It returns bool type only
- **and, or, not** are Logical operators.
- when we have, "or" operator...atleast 1 condition should be True then only overall output will be True
- when we have, "and" operator...all conditions should be True then only overall output will be True

> **Logical operators Precedence**

- Logical NOT
- Logical AND
- Logical OR

In [24]:

```
5 > 3 and 2 > 3
```

Out[24]:

False

In [25]:

```
5 > 3 or 5 < 3
```

Out[25]:

True

In [26]:

```
not False
```

Out[26]:

True

In [27]:

```
not True
```

Out[27]:

False

In [28]:

```
5 > 3 or 2 > 3 and 5 < 3
```

Out[28]:

True

In [29]:

```
False or not False and True
```

Out[29]:

True

### Identity opertors

- Identity operators are used to check if two values (or variables) are indicating to same object or no
- It returns output as boolean
- **is , is not** are the identity operators in Python.
- **is** operator (# is - True if the opernds are identical)
- **is not** operator (# is not - True if the operands are not identical)

In [30]:

```
a = 5
b = 5
print(a is b)              # iid(a)==id(b)
```

```
True
```

In [31]:

```
s1 = "satish"
s2 = "Satish"
print(s1 is s2)             # id(s1)==id(s2)
```

```
False
```

In [32]:

```
a=6
b=8
a is not b
```

Out[32]:

```
True
```

### Membership Operators

- Membership operators are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).
- It returns oputput as boolean
- **in , not in** are the membership operators.

In [33]:

```
a='venkatesh'
'esh' in a
```

Out[33]:

```
True
```

In [34]:

```
lst = [1, 2, 3, 4]
5 in lst
```

Out[34]:

```
False
```

In [35]:

```
a=[1,2,3,4,5,6]
9 not in a
```

Out[35]:

```
True
```

**only arthimetic operators, return with value**

**remaining all operators, return boolean value**

## Operators precedence:

- Arthimetic operators
- Comparision operators
- membership operators
- identity operators
- Logical operators
- Assignment operators

In [36]:

```
b = (1>2) or (3>=3) and (5<4)
b
```

Out[36]:

```
False
```