Keywords in python:

- Keyword:

    - Keywords in python are reserved word that have special meanings.

In [2]:
```
# TO print the keywords in python
import keyword
print(keyword.kwlist)
print(len(keyword.kwlist))
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif',
'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'p
ass', 'raise', 'return', 'try', 'while', 'with', 'yield']
35
```

Operators:

- Operator:

    - Operators are used to perform operations on variables and values.

- In python 7 types operators:
- 1. Arithmetic Operators
- 2. Assignment Operators
- 3. Comparision Operators
- 4. Logical Operators
- 5. Identity Operators
- 6. Membership Operators
- 7. Bitwise Operators

In [3]:
```
# 1. Arithmetic Operators:
# It is used for simple mathematical calculations.
# ex: (+,-,*,/,%,//(floor division),**(power))
a = 10
b = 20
print('Addition is: ',a+b)
print('Subtraction is:',a-b)
print('Multiplication is:',a*b)
print('Division is:',a/b)
print('Floor Division:',a//b)
print('Modulus is:',a%b)
print('power is:',a**3)
```

```
Addition is:  30
Subtraction is: -10
Multiplication is: 200
Division is: 0.5
Floor Division: 0
Modulus is: 10
power is: 1000
```

In [4]:
```
print(5/2) # division
```

```
2.5
```

In [5]:
```
print(5//2) # floor division
```

```
2
```

In [6]:
```
print(7%2) # remainder
```

```
1
```

In [8]:
```
# Assignment Operators:
# To assign the value to variable..
# (+=,-=,*=,/=,//=,%=,**=)
a = 34
b = 60
a +=b # a = a+b => a = 34+60 => 94
print('a value is: ',a)
b -=a # b = b-a => 60-94 => -34
print('b value is:',b)
```

```
a value is:  94
b value is: -34
```

In [11]:
```
# Comparision Operator:
# To compare between the two values.
# (>,<,<=,>=,!=,==)
# It returns the boolean values like True or False.
n = 50
m = 78
print(n>m)
print(m>n)
print(50 == 78)
```

```
False
True
False
```

- Logical Operators:
- To check the two or more conditions at a time.
- It returns the boolean values.
- (and,or,not)
- and truth table

a b a and b T T True T F False F T False F F False

- or truth Table

a b a or b not(a) T T True False T F True False F T True True F F False True

In [16]:
```
a = 100
b = 200
print(a>b and a<b)
# false and True -> False
print(a==100 and b == 200)
print(a==100 or b<a)
# True  and False -> True
print(not(a>b))
# not(False)
```

```
False
True
True
True
```

In [18]:
```
# Identity Operators:
# is, is not
a = 'sdc'
b = 'apssdc'
print(a is b) # It is similar to equal
print(a is not b) # It is similar to not equal
```

```
False
True
```

In [21]:
```
# Membership Operators:
# It is mainly used for sequence of object..
# in, not in
# It returns the boolean values
List1 = [10,20,30,40]
print(10 in List1)
print(100 in List1)
print('h' in 'apssdc')
```

```
True
False
False
```

In [22]:
```
# bitwise operators:
# To convert the data into bitlevel.
# (&(and),|(or),^(x-or),<<(left shift),>>(right shift),~(negation))

```

Conditional Statements:

- To check the conditions
- 1. simple if-else
- 2. elif
- 3. elif-ladder
- 4. nested if-else

simple if-else syntax:

- if(condition): # always true

      statements

  else:

In [24]:
```
# To Check whether the given number is even or not?
n = int(input("enter n value: "))
if(n%2 == 0):
    print('It is a even number ')
else:
    print('it is not a even number')
```

```
enter n value: 19
it is not a even number
```

In [25]:
```
# To check whether the given two numbers are equal or not?
a = int(input("Enter a value: "))
b = int(input("enter b value: "))
if(a == b):
    print('Two numbers are equal')
else:
    print('Not equal')
```

```
Enter a value: 10
enter b value: 20
Not equal
```

In [ ]:
```

```