Operators : Operators are the symbols which are used to perform operations on values and variables

Types of Operators in Python :

- Arithmetic Operators
- Assignment Operators
- Comparision/Relational Operators
- Logical Operators
- Identity Operators
- Membership Operators

Assignment operator:-

In [1]:
```
a = 100
a
```

Out[1]: 100

In [3]:
```
# += :- Assignment & Arithmetic:-
a = 10
b = 20
print(a)
print(b)
a += b  # a += b -> a = a+b
print(a)
```

```
10
20
30
```

In [4]:
```
c = a+b
c
```

Out[4]: 50

In [6]:
```
# -=:-
a = 10
b = 20
print(a)
print(b)
a -= b  # a -= b -> a = a-b
print(a)
```

```
10
20
-10
```

In [7]:
```
# *=:-
a = 10
b = 20
print(a)
print(b)
a*= b  # a*= b -> a = a*b
print(a)
```

```
10
20
200
```

Comparision Operators:- used to compare the values.

In [8]:
```
100 > 67
```

Out[8]: True

In [9]:
```
66 == 66
```

Out[9]: True

In [10]:
```
'cat' != 'CAT'
```

Out[10]: True

In [11]:
```
100 <= 100
```

Out[11]: True

In [12]:
```
100 > 10
```

Out[12]: True

In [13]:
```
'Z' < 'z'
```

Out[13]: True

Logical Operators:- used to combine multiple conditions at a time.

In [15]:
```
# and:- if all the conditions are True then the result will be True.
('cat' == 'cat') and ('dog' != 'cat')
```

Out[15]: True

In [16]:
```
# or:- if any of the condition is True it will result True.
(15 % 5 == 0) or (15 % 4 == 0)
```

Out[16]: True

In [18]:
```
# not :- it is an unary op, that result True condition as False and Vice-versa.
```

```
not(15 % 5 == 0)
```

Out[18]: False

Special Operators:-

Membership Operators:-

In [30]:
```
# in :- returns True if an element is present in the sequence (string,list,set,tupl
'hello' in 'hello world'
```

Out[30]: True

In [31]:
```
# not in:- returns True if an element is not present in the sequence (string,list,s
12 not in [1,2,3,4,45]
```

Out[31]: True

Q. Wapp to return True if a character is a vowel.

In [1]:
```
char = input()
char in 'aeiouAEIOU'
```

Out[1]: True

Identity Operator:- used to compare the address of any variable.

In [3]:
```
# is:- return True if the addresses of both variables are same.
x = 190
y = 190
print( x is y )
# to check the address of any variable we can use id().
print(id(x))
print(id(y))
```

```
True
140728376285896
140728376285896
```

In [5]:
```
x = 1900
y = 1900
print( x is y )
print(id(x))
print(id(y))
```

```
False
2502842872368
2502842874832
```

In [6]:
```
# is not:- return True if addresses of the variables are different.
a = [1,2,3]
b = [1,2,3]
```

```
print( a is not b )
print(id(a))
print(id(b))
```

```
True
2502844167936
2502843230592
```

In [7]:
```
a = [1,2,3]
b = a
print( a is not b )
print(id(a))
print(id(b))
```

```
False
2502844166784
2502844166784
```

Q. Wapp to return True if a year is leap else retrun False.

In [8]:
```
year = int(input("Enter the year: "))
print((year%400==0) or ((year%4==0) and (year%100!=0)))
```

```
False
```

Indentation : It is one of the most important feature in Python. It indicates the block of code.

In [9]:
```
if 'A' == 'A':
    print("both values are equal")
```

```
both values are equal
```

In [10]:
```
if 'A' == 'A'
print("both values are equal")
```

```
  Cell In[10], line 1
    if 'A' == 'A'
                 ^
SyntaxError: expected ':'
```

In [14]:
```
if 'A' == 'A':
    print("both values are equal")
    print('hello')
print('out of if block')
```

```
both values are equal
hello
out of if block
```

Conditional Statement :- It is used to the check the flow of control according to the given condition.

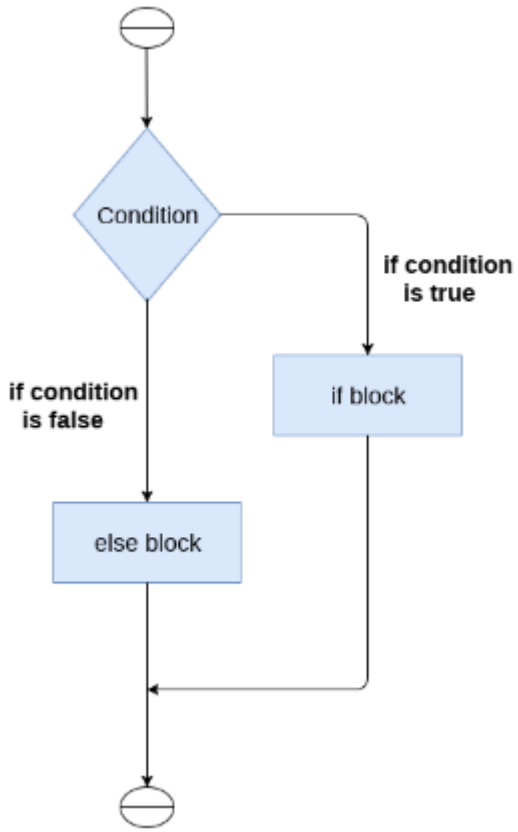
- Single/Simple if

- if-else
- Nested if
- if-elif-else

If Statement	The if statement is used to test a specific condition. If the condition is true, a block of code (if-block) will be executed.
If - else Statement	The if-else statement is similar to if statement except the fact that, it also provides the block of the code for the false case of the condition to be checked. If the condition provided in the if statement is false, then the else statement will be executed.
Nested if Statement	Nested if statements enable us to use if ? else statement inside an outer if statement.

if statement : The if statement is used to test a particular condition and if the condition is true, it executes a block of code known as if-block.

Syntax: if expression:
statement
statement



In [15]:
```
char = input("Enter a character: ")
if char in 'aeiouAEIOU':
    print(char,'is a vowel')
```

```
O is a vowel
```

In [16]:
```
char = input("Enter a character: ")
if char in 'aeiouAEIOU':
    print(char,'is a vowel')
```

If -Else:- if checks the truthness as well as the falsiness of the program.

In [17]:
```
char = input("Enter a character: ")
if char in 'aeiouAEIOU':
    print(char,'is a vowel')
else:
    print(char,'is not a vowel')
```

```
r is not a vowel
```

In [18]:
```
char = input("Enter a character: ")
if char in 'aeiouAEIOU':
    print(char,'is a vowel')
else:
    print(char,'is not a vowel')
```

```
i is a vowel
```

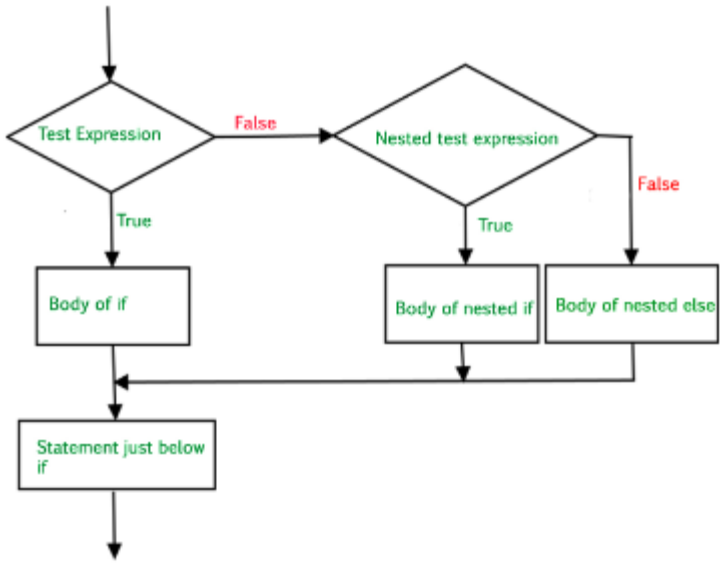
Nested if : to check more than one condition at a time.

Syntax:

```
if condition:
       # block of statements

       if condition:
          # block of statements

       else:
          # block of statements
  else:
     # block of statements
```



In [19]:
```
tenth = float(input("Enter 10th mark: "))
twelve = float(input("Enter 12th mark: "))
btech = float(input("Enter Btech's mark: "))
if tenth >= 80:
    if twelve >= 80:
        if btech >= 80:
            print("you are eligible for scholarship")
```

```
you are eligible for scholarship
```

In [21]:
```
tenth = float(input("Enter 10th mark: "))
twelve = float(input("Enter 12th mark: "))
btech = float(input("Enter Btech's mark: "))
if tenth >= 80:
    if twelve >= 80:
        if btech >= 80:
            print("you are eligible for scholarship")
        else:
            print("you scored less in btech")
    else:
        print("aur kar masti 12th mey")
else:
    print("padhae kar leta board hai 10th ka")
```

```
padhae kar leta board hai 10th ka
```

if-elif-else: The elif statement enables us to check multiple conditions and execute the specific block of statements depending upon the true condition among them. We can have any number of elif statements in our program depending upon our need.

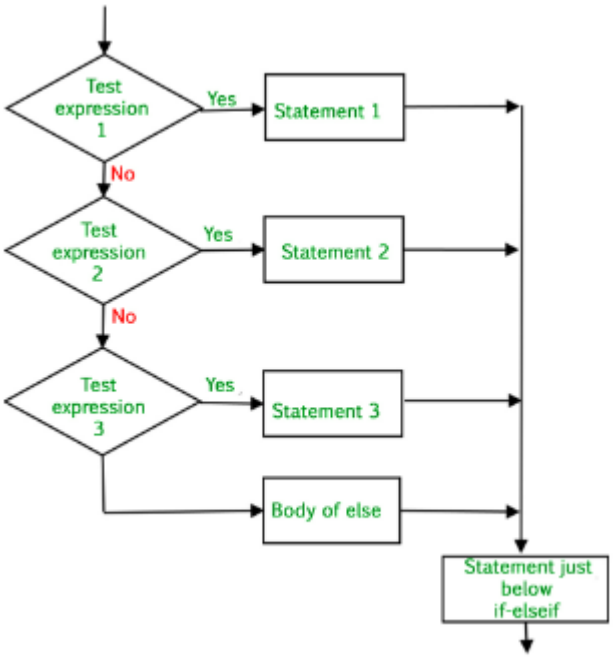
Syntax:

```
if expression 1:
    # block of statements
```

```
elif expression 2:
    # block of statements

elif expression 3:
    # block of statements

else:
    # block of statements
```



In [25]:
```
treat = input("Enter what you wanna eat: ").lower()
if treat == 'burger':
    print("we will have burger party ")
elif treat == 'pizza':
    print("we can have pizza party today")
elif treat == 'biriyani':
    print("let's head to our favourite biriyani stop")
elif treat == 'pakhala':
    print("Tu ghar aakar Pakhala khale")
else:
    print('Tu bhaad mey jaa')
```

```
let's head to our favourite biriyani stop
```

Q. Wapp to check if the entered character is a alphabet, digit or special character.

In [29]:
```
char = input("Enter a character: ")
if char.isalpha():
    print('alphabet')
elif char.isdigit():
    print('digit')
elif char.isspace():
```

```
    print('space')
else:
    print('special character')
```

```
special character
```

Q. A company decided to give 5% bonus to its employees if his/her service is more than 5 years. Ask user for their salary and year of experience and print the net bonus amount.