In [ ]:
```
# Transfer statement

1- break

2- pass

3- continue
```

In [2]:
```
numbers = [1,2,4,4,5,6,7,8]

for num in numbers:
    if num == 5:
        break
    print(num)
```

```
1
2
4
4
```

In [3]:
```
i = 0
while i < 10:
    i = i +  1  -> assignment operator
    print(i)
    break
```

```
1
```

In [6]:
```
# continue

for i in range(10):
    if i  == 5:
        print("Pause and restart")
        continue
    print(i)
```

```
0
1
2
3
4
Pause and restart
6
7
8
9
```

In [7]:
```
# pass
name = "John"
if name == "John":
    pass

for i in range(10):
    pass

i = 0
while i < 10:
```

```
    pass

def add():
    pass
```

Operator

In [ ]:
```
Python Operator are  special symbols or keywords that perform on one or more ope


Types:

1- Arithematic Operator --> Perform the mathematical calculations( +, - ,/ ,*, /

2- Comparison Operator -- compare values and return boolean values ( ==, !=,<,>,

3- Assignment operator  - --> assign value to the variable( =, +=, -=)

4- logical operator   ---> Combine or modify boolean expression (and, or, not)

5- Bitwise Operator  --- > bits of integer (| ,&,^)

6- Membership Operator --> check for presence within a sequence( in , not in)

7- identity Operator  --> Compare object memory address( is ,is not)
```

Arithematic Operator

In [15]:
```
x =  10
y = 5

result  = x  + y
print(result)

result  = x  - y
print(result)

result  = x  / y
print(result)

result  = x  * y
print(result)

result  = x  % y
print(result)

result  = x  // y
print(result)

result  = x  ** y
print(result)
```

```
15
5
2.0
50
0
2
100000
```

Comparison Operator

In [20]:
```python
age  = 25

if age >= 18:
    print("Eligible for work")

if age == 25:
    print("same value")

if age < 50:
    print("age is less than 50")
```

```
Eligible for work
same value
age is less than 50
```

Assignment operator

In [ ]:
```python
count = 0

count = count + 1 # 1

count += 1  # 2---> count = count + 1

count -=1 ---> count = count -1
```

logical operator

In [22]:
```python
age = 25
if age >=18 and age <=60:
    print("They have to work")


if age >=18 or age <=60:
    print("They have to work")
```

```
They have to work
They have to work
```

Bitwise Operator

In [32]:
```python
2 & 3
```

Out[32]: 2

In [27]:
```
4 & 5
```

Out[27]: 4

In [28]:
```
2 | 3
```

Out[28]: 3

In [29]:
```
2 | 2
```

Out[29]: 2

In [33]:
```
x = 2
y = 3
```

In [35]:
```
x & y
```

Out[35]: 2

In [38]:
```
bin(9)
```

Out[38]: '0b1001'

Membership Operator

In [45]:
```
#in , not in

lst = [1,2,3]

for i in lst:
    print(i)

if 21 in lst:
    print("Yes")
else:
    print("No")

if 21 not in lst:
    print("Yes")
else:
    print("No")
```

```
1
2
3
No
Yes
```

identity Operator

In [52]:
```
a = 10
b = 5
```

```
a is not b

b is a

age  = 20
if age is 20:
    print("age is 2 year")
```

```
age is 2 year
```

```
<>:9: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:9: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
C:\Users\jitud\AppData\Local\Temp\ipykernel_15080\2348820079.py:9: SyntaxWarning:
"is" with 'int' literal. Did you mean "=="?
  if age is 20:
```

convert number to binary

In [56]:
```
print(bin(23))
print(bin(50))
print(bin(22))
```

```
0b10111
0b110010
0b10110
```

convert binary to number

In [59]:
```
print(0b10111)
print(0b110010)
print(0b10110)
```

```
23
50
22
```

In [63]:
```
#number to hexa

hex(8)
oct(6)
```

Out[63]:
```
'0o6'
```

In [ ]:
```

```