What are Operators

In [ ]:
```
Operators are the Symbols that are used to perform a specific task.
Examples:
    + --> addtion
    - -->subtraction
    * --> Multiplication
```

What is an Operand and Expression?

In [ ]:
```
Operand --> A variable or a value involved in an operation is called an operand.
Example :
           c=a+b

        here a and b are operand
```

Expression --> Expression is nothing but a combination of operand and operator. Example: c=a+b

```
         here c=a+b is an Expression
```

Examples of Operand ,Expression and Operator

In [ ]:
```
a=10
b=20
c=a+b

Operator  --> =,+
Variable --> a,b,c
Operand --> a,b
Expression -->c=a+b
```

In [ ]:
```
a=10
b=20
c=a+b-a

Operator -->  _ ,+,=
Variable --> a,b,c
Operend -->a,b
Expression -->c=a+b-a
```

Types of Operators in Python

In [ ]:
```
1.Arithmetic Operators
2.Comparison or Relational Operator
3.Assignment Operator
4.Memebership Operator
5.Identity Operator
6.Logical Operator
```

Arithmetic Operators

In [ ]:
```
#Arithmetic Operator
1)--->+ -->Addition
2)--> - --> Subtraction
3)--> * --> Multiplication
4)--> / --> Division
5)--> // --> Floor Division
6)--> ** --> Power or Exponenetial
7)--> %   -->Modulus or Reminder
```

Examples of Arithmetic Operator

In [6]:
```
a=40.00
b=20
print("a+b :",a+b)   #60
print("a-b :",a-b)   #20
print("a*b :",a*b)   #800
print("a/b :",a/b)   #2
print("a//b :",a//b) #2
print("a**b :",a**b) #100000000000000000000000000000000000000000
print("a%b :",a%b) #0

Note: / division will always give you answer in floating point numbers.whereas if you are using
      floor division then if atleast one operand is float then the answer will also be in float
      else answer will always be Int.
```

```
a+b : 60.0
a-b : 20.0
a*b : 800.0
a/b : 2.0
a//b : 2.0
a**b : 1.099511627776e+32
a%b : 0.0
```

In [ ]:
```
floor number  -->bottom(lower)
ceil number --> upper value
```

In [59]:
```
import math
print(math.ceil(3.98))
print(math.floor(3.98))
```

```
4
3
```

Comparison and relational operators

In [ ]:
```
#This operator will always return True or False(Boolean)
> --> Greater Than Operator
< --> Less Than Operator
>= --> Greater than equal to operator
<= --> Less than equal to
== --> Equal to
!= --> Not Equal to
```

Examples of Comparsion Operators

In [61]:
```
a=10
b=20
print(a!=b)
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
print(a==b)
#Generally these operators are useful in case of conditional stataement.
```

```
True
False
True
False
True
False
```

In [23]:
```
#Write a python program weather the first value is greater than second or not.
x=int(input())
y=int(input())
if x>y:
    print("x is greater")
else:
    print("Y is greater")
```

```
10
20
Y is greater
```

Assignment Operators

In [ ]:
```
we can use assignement operator to assiging a value into a variable
Example:
    x=10
we can also combine more than one operator with assignment operator
x+=10 -->x=x+10
x-=10 -->x=x-10
x*=10 -->x=x*10
x//=10 -->x=x//10
x%=10 -->x=x%10
x**=10 -->x=x**10
```

In [66]:
```
x=10
y=20
x+=10
print(x)
```

```
20
```

In [69]:
```
x=10
y=20
x-=20
print(x)
```

```
-10
```

Difference between Equal to(==) and Equal(=)

In [26]:
```
== --> Equal to operator (So basically it is used to check weather two values are same or not)
= --> Assigning the value to a variable
```

Out[26]:
```
-10
```

Membership Operator

In [ ]:
```
Membership Operator:We can use memebership opertaor to check weather a given element is a member if
a sequence or not. If the element is a member of a sequence than it will return true else return
false.

Two Types of Membership operator we are having in Python:

in --> if element is present in the sequence then in operator will give you True else false
not in --> if the element is present in the sequence then not in operator will give you
          False else True

Note: Membership Operator will always return answer in True or False
     This memebership operator are used in conditional statement along with loops
```

Examples of Membership Operator

In [30]:
```
x=[10,20,30,40]
10 not in x
```

Out[30]:
```
False
```

In [73]:
```
x={10,20,30,40}
20 in x
```

Out[73]:
```
True
```

In [74]:
```
x=(20,30,40,50)
30 in x
```

Out[74]:
```
True
```

In [75]:
```
x="HelloWorld"
"H" in x
```

Out[75]:
```
True
```

Identity Operators

In [ ]:
```
Identity Operator --> Identity Operators are generally used memeory location comparison
                     (address comparsion).

There are Two Types Of Identity Operators:
    is  --> if the addresses of two variables are poiting to same object then is will return True
            else is will return False

    is not -->if the addresses of two variables are poiting to same object then is will return False
            else is will return True

Note: These are basically used to check weather two variables pointing to the same object or Not.
      Identity Operators will always return result as True or False.
```

Examples of Identity Operators

In [76]:
```
x=10.5
y=10.5
print(id(x))
print(id(y))
print(x is y)   #Address Comparison
print(x==y)     #Content Comparison
```

```
1841155692208
1841159609584
False
True
```

In [77]:
```
x=10
y=10
print(id(x))
print(id(y))
print(x is y)   #Address Comparison
print(x==y)     #Content Comparison
```

```
1841040812624
1841040812624
True
True
```

In [78]:
```
x=10+20j
y=10+20j
print(id(x))
print(id(y))
print(x is y)   #Address Comparison
print(x==y)     #Content Comparison
```

```
1841159609392
1841159609456
False
True
```

In [79]:
```
x="Hello World"
y="Hello World"
print(id(x))
print(id(y))
print(x is y)   #Address Comparison
print(x==y)     #Content Comparison
```

```
1841160035760
1841160034544
False
True
```

Difference Between is Operator and equal to(==) Operator

In [ ]:
```
#Most important Interview Question

is = basically used to compare two variables Id(Mememory location if both the memeory locations are
                                         pointing to same memoery then is will return True else False)

equal to = Compare the content of the variable
```

Logical Operators

In [ ]:
```
There are Three Types of logical operators:
    and --> Binary Operator
    or  --> Binary Operator
    not --> unary Operator it need only one operand to perform a task
```

In [ ]:
```
Truth Tables of Logical operators:
    A      B         A and B             A or B            not A        not B
    True   True      True                True              False        False
    True   False     False               True              False        True
    False  True      False               True              True         False
    False  False     False               False             True         True
```

Case 1: For boolean datatype the behaviour of logical Operator

In [ ]:
```
and -->  if both arguments are True then the result is True else False
or --> If atleast one argument is True the answer is True else Answer is False
not --> complement --> if True is an argument then not of True is False else True
```

Examples:

In [ ]:
```
True and False  --> False
True and True --> True
True or True --> True
True or False --> True
not True -->False
not False --> True
```

Case 1: For non boolean datatype the behaviour of logical Operator

In [ ]:
```
Note:
    0 and emptystring --> False
    rest all are True --> True
```

Behaviour of And Operators

In [81]:
```
0 and 20
```

Out[81]:
```
0
```

In [82]:
```
10 and 20
```

Out[82]:
```
20
```

In [83]:
```
"" and "Hello"
```

Out[83]:
```
''
```

In [84]:
```
10 and 20
```

Out[84]:
```
20
```

In [86]:
```
"Hello" and "World"
```

Out[86]:
```
'World'
```

Behaviour of OR Operator

In [ ]:
```
x or y
if x evalutes to True return x else return y
```

In [87]:
```
0 or 10
```

Out[87]:
```
10
```

In [88]:
```
"" or 50
```

Out[88]:
```
50
```

In [89]:
```
10 or 20
```

Out[89]:
```
10
```

In [91]:
```
"" or "world"
```

Out[91]:
```
'world'
```

In [92]:
```
"hello" or "world"
```

Out[92]:
```
'hello'
```

Behaviour of Not Operator

In [95]:
```
not 0
```

Out[95]:
```
True
```

In [93]:
```
not ""
```

Out[93]:
```
True
```

In [94]:
```
not 90
```

Out[94]:
```
False
```

Operator Procedence

In [ ]:
```
Operator Procedence:
    if we have multiple operators in an experssion then which operator will evalute first will be
    decided based on operator procendence.


Procedence of Operator List:

Operators                                                        Meaning

()                                                               Parentheses

**                                                               Exponent

*, /, //, %                                      Multiplication, Division, Floor division, Modulus

+, -                                                             Addition, Subtraction

==, !=, >, >=, <, <=, is, is not, in, not in    Comparisons, Identity, Membership operators

not                                                              Logical NOT

and                                                              Logical AND

or                                                               Logical OR
```

Examples

In [96]:
```
3/2*4+3+(10/5)**3-2
3/2*4+3+2.0**3-2
3/2*4+3+8.0-2
1.5*4+3+8.0-2
6.0+3+8.0-2
17.0-2
15.0
```

Out[96]:
```
15.0
```

In [97]:
```
2*3**3*3**3
2*3**3*27
2*27*27
54*27
```

Out[97]:
```
1458
```

In [98]:
```
3/2*4+3+(10/5)**3-2
```

Out[98]:
```
15.0
```

Practice Problems

In [ ]:
```
Problem 1: We need to take input from the user as Principle , amount , and rate and based on the
           given value we need to find the simple interest?

Example
input : 200
        2
        2
Output: Simple interest is 8
```

In [ ]:
```
#Solution:
principle = float(input("Enter principle amount :"))
rate = float(input("Enter the rate : "))
Time = float(input("Enter the Time in Years"))
Simple_Interest = (principle*rate*Time)//100
print(int(Simple_Interest))
```

In [ ]:
```
Problem 2: Temperature conversion in which you will have ferneheit scale and based on that scale
           you need to convert that temperature into celsious?

                T(oC)   =  ((T(oF)  -  32  ) ×  5)/9


Algorithm:
Define temperature in Fahrenheit unit.
Apply in the formula.
Print the temperature in Celsius.

Example:
Input :
Farenhrit value is 54

Output:
Temperature in Celsius value is 12.22222
```

In [ ]:
```
Problem 3: We need to take input from the user and perform addition , subtraction ,
           multiplication , division , floor division , modulus and power operations

Example:
input : 10
        2
Output:
Additon of given numbers is - 12
Subtraction of given numbers is - 8
Multiplication of given numbers is - 20
Divison of given numbers is - 5.0
```