Operators

- Arithmetic
- Relational
- Assignment
- Logical
- Bitwise
- Membership
- Identity

In [1]:

```
# Arithmatic Operators (+,-,*,/,%,//,**)
4+5
```

Out[1]:

```
9
```

In [2]:

```
9-5
```

Out[2]:

```
4
```

In [3]:

```
2*2
```

Out[3]:

```
4
```

In [4]:

```
8/2
```

Out[4]:

```
4.0
```

In [5]:

```
8//2
```

Out[5]:

```
4
```

In [6]:

```
2%2
```

Out[6]:

```
0
```

In [7]:

```
2**1
```

Out[7]:

```
2
```

In [8]:

```
2**2
```

Out[8]:

```
4
```

In [9]:

```
a=6
b=8
```

In [10]:

```
a+b
```

Out[10]:

```
14
```

In [11]:

```
a-b
```

Out[11]:

```
-2
```

In [12]:

```
a/b
```

Out[12]:

```
0.75
```

In [13]:

```
# Relational(Comparison) Operators (==,!=,>,<,>=,>=)
5==6
```

Out[13]:

```
False
```

In [14]:

```
2!=2
```

Out[14]:

```
False
```

In [15]:

```
41>89
```

Out[15]:

```
False
```

In [16]:

```
4>2
```

Out[16]:

```
True
```

In [17]:

```
56<4
```

Out[17]:

```
False
```

In [18]:

```
1<2
```

Out[18]:

```
True
```

In [19]:

```
6>=4
```

Out[19]:

```
True
```

In [20]:

```
6>=6
```

Out[20]:

```
True
```

In [21]:

```
7<=2
```

Out[21]:

```
False
```

In [22]:

```
a<=b
```

Out[22]:

```
True
```

In [23]:

```
# Assignment Oprerators (=,+=,-=,*=,/=,%=,//=,**=)
a=5
b=3
c=a
```

In [24]:

```
c
```

Out[24]:

```
5
```

In [25]:

```
a
```

Out[25]:

```
5
```

In [26]:

```
a+=2#a+2
```

In [27]:

```
a
```

Out[27]:

```
7
```

In [28]:

```
b-=1
```

In [29]:

```
b
```

Out[29]:

```
2
```

In [30]:

```
a*=5
```

In [31]:

```
a
```

Out[31]:

```
35
```

In [32]:

```
a/=2
```

In [33]:

```
a
```

Out[33]:

```
17.5
```

In [34]:

```
a%=2
```

In [35]:

```
a
```

Out[35]:

```
1.5
```

In [36]:

```
c=9
```

In [37]:

```
c//=3
```

In [38]:

```
c
```

Out[38]:

```
3
```

In [39]:

```
c**=3
```

In [40]:

```
c
```

Out[40]:

```
27
```

In [42]:

```
# Bitwise Operators (&,|,^,~,<<,>>)
a= 7 # 0111
b= 6 # 0110

# logical AND (&)
#  0111
#  0110
# -----
#  0110
# O/P: 6

# logical OR (|)
#  0111
#  0110
# ------
#  0111

# O/P:- 7

# Logical XOR (^)
# 0111
# 0110
#-------
# 0001

# O/P:- 1

# Complement ~
#  0111
#-------
#  1000

# O/P:- 8

# Left Shift (<<)
# 5 ->  0101
# 5<<1
#    0101
#  --------
#    01010

# O/P:- 10

# Right Shift (>>)
# 5 -> 0101
# 5>>1
#  0101
#---------
#  0010

#O/P:2
```

In [43]:

```
# Membership Operators (in,not in)
temp="string"
'a' in temp
```

Out[43]:

```
False
```

In [44]:

```
'g' in temp
```

Out[44]:

```
True
```

In [45]:

```
'z' not in temp
```

Out[45]:

```
True
```

In [46]:

```
'i' not in temp
```

Out[46]:

```
False
```

In [52]:

```
# Identity Operators (is, is not)
a=5
b=6
print(id(a))
print(id(b))
print(a is b)
print(a is not b)
```

```
140718760671136
140718760671168
False
True
```

In [53]:

```
c=5
d=5
c is d
```

Out[53]:

```
True
```

In [54]:

```
# Logical Operators (and, or , not)
print(a,b)
```

```
5 6
```

In [56]:

```
a >b and a>b
```

Out[56]:

```
False
```

In [57]:

```
4 and 0
```

Out[57]:

```
0
```

In [58]:

```
87 and 45
```

Out[58]:

```
45
```

In [59]:

```
0 and 12
```

Out[59]:

```
0
```

In [60]:

```
2 or 3
```

Out[60]:

```
2
```

In [61]:

```
0 or 5
```

Out[61]:

```
5
```

In [62]:

```
False or 0
```

Out[62]:

```
0
```

In [63]:

```
not False
```

Out[63]:

```
True
```

In [64]:

```
not 0
```

Out[64]:

```
True
```

In [41]:

```
# single line comment in python
""" This is a multi line comment in python"""
''' This is also a multi line comment in python  '''
```

Out[41]:

```
' This is also a multi line comment in python  '
```

In [65]:

```
# Input and Output
```

In [ ]:

```
# C language
# scanf() # I/p
# printf() #o/p

# Python
# input() # i/p
# print()  # o/p
```

In [1]:

```
# Input() in python
#temp= input() # default string datatype is String
# string input
name =input()
```

```
satyanarayana
```

In [2]:

```
name
```

Out[2]:

```
'satyanarayana'
```

In [3]:

```
type(name)
```

Out[3]:

```
str
```

In [4]:

```
# inter input
a =input()
b=int(a)
```

```
45
```

In [5]:

```
print(type(a))
print(type(b))
```

```
<class 'str'>
<class 'int'>
```

In [8]:

```
temp =int(input())
```

```
abcd

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-d2524d9c2a36> in <module>
----> 1 temp =int(input())

ValueError: invalid literal for int() with base 10: 'abcd'
```

In [7]:

```
temp
```

Out[7]:

```
85
```

In [9]:

```
# Float value
t =float(input())
```

```
56.56
```

In [10]:

```
print(type(t))
```

```
<class 'float'>
```

In [11]:

```
temp1= int(input())
```

```
45.23

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-11-e90432ee23f3> in <module>
----> 1 temp1= int(input())

ValueError: invalid literal for int() with base 10: '45.23'
```

In [12]:

```
temp1= int(float(input()))
```

```
452.36
```

In [13]:

```
temp1
```

Out[13]:

```
452
```

In [14]:

```
a=input("Enter Something")
```

```
Enter Somethinghello
```

In [15]:

```
b=input(123456)
```

```
123456999
```

In [16]:

```
b
```

Out[16]:

```
'999'
```

In [17]:

```
# print() O/P
```

In [18]:

```
print("I am Print function")
```

```
I am Print function
```

In [19]:

```
print('I am print func')
```

```
I am print func
```

In [20]:

```
print(45)
```

```
45
```

In [21]:

```
print(63.89)
```

```
63.89
```

In [22]:

```
print("first line")# \n excuted default at end of print
print('second line')
```

```
first line
second line
```

In [23]:

```
# print() have 2 keyword arguments (end,sep)
print('one',end=' ')

print('two')
```

```
one two
```

In [25]:

```
print(1,2,3,4,5,6,sep=',') #when min 2 arguments have in print then only sep is excuted
```

```
1,2,3,4,5,6
```

In [26]:

```
print(1,sep=',')
```

```
1
```

In [27]:

```
print(1,2,sep='<--->')
```

```
1<--->2
```

In [28]:

```
print(sep=',',1,2,3)
```

```
  File "<ipython-input-28-6c745bd55374>", line 1
    print(sep=',',1,2,3)
                  ^
SyntaxError: positional argument follows keyword argument
```

In [29]:

```
print(1,2,3,sep=',',4,5,6)
```

```
  File "<ipython-input-29-5c6b22e07607>", line 1
    print(1,2,3,sep=',',4,5,6)
                        ^
SyntaxError: positional argument follows keyword argument
```

In [30]:

```
print(end='\t',"hello")
```

```
  File "<ipython-input-30-d65a574efeee>", line 1
    print(end='\t',"hello")
                   ^
SyntaxError: positional argument follows keyword argument
```

In [32]:

```
print("ramu",'srinu','kiran',sep='<-sep->',end='\t')
print('sitha')
```

```
ramu<-sep->srinu<-sep->kiran    sitha
```

Conditional Statments

- if
- if else
- if elif

- Nested If

In [ ]:

```
'''
# if syntax
if condition: # {
    statment1
    statment2
    ...
    ...         #  }
statemt

'''
```

In [34]:

```
if 5<6:
    print('Highest number is 6')
    print("This is a if block because I have tab spce & :")
```

```
Highest number is 6
This is a if block because I have tab spce & :
```

In [36]:

```
# if else Example
temp = int(input("Enter Number: "))
if temp>0:
    print("Given Number is Positive Number")
else:
    print("Negative Number")
```

```
Enter Number: -1
Negative Number
```