

Operators in Python

In [4]:

```
# arithmetic operator
# (+, -, *, /, //, %, **)

x = 20
y = 4
print("value of x+y :",x+y)
print("value of x-y :",x-y)
print("value of x*y :",x*y)
print("value of x/y :",x/y) # gives float in output
print("value of x//y :",x//y) # float division operator (it rounds down the value)
print("value of x%y :",x%y) # modulo operator gives reminder
print("value of x**y :",x**y) # x^y
```

```
value of x+y : 24
value of x-y : 16
value of x*y : 80
value of x/y : 5.0
value of x//y : 5
value of x%y : 0
value of x**y : 160000
```

In [5]:

```
33.7+5
```

Out[5]:

```
38.7
```

In [6]:

```
True+6
```

Out[6]:

```
7
```

In [7]:

```
print("nikhil"+"vishwakarma") # concatienation
```

```
nikhilvishwakarma
```

In [8]:

```
print("nikhil"+"vishwakarma"+"aids")
```

```
nikhilvishwakarmaaids
```

In [9]:

```
print("nikhil"+"vishwakarma"+88)
```

TypeError

Traceback (most recent call last)

t)

Cell In[9], line 1

```
----> 1 print("nikhil"+"vishwakarma"+88)
```

TypeError: can only concatenate str (not "int") to str

In [10]:

```
"s"*6 # imp
```

Out[10]:

```
'ssssss'
```

In [11]:

```
"nikhil "*10
```

Out[11]:

```
'nikhil nikhil nikhil nikhil nikhil nikhil nikhil nikhil nikhil nikhil '
```

In [13]:

```
print("nikhil\n"*10)
```

```
nikhil
nikhil
nikhil
nikhil
nikhil
nikhil
nikhil
nikhil
nikhil
nikhil
```

In [14]:

```
"nikhil "*10*" "*9
```

TypeError

Traceback (most recent call last)

t)

Cell In[14], line 1

```
----> 1 "nikhil "*10*" "*9
```

TypeError: can't multiply sequence by non-int of type 'str'

In [20]:

```
print(30/4) # always gives float value
print(30.6/4)
print(30//4) # perform division ,floors down the value,return int bcz both operants ar
print(30.6//4) # perform division ,floors down the value,return float bcz one operant i
print(30.6//4.2) # perform division ,floors down the value,return float bcz both operan
print(30//4.2) # perform division ,floors down the value,return float bcz one operan
```

```
7.5
7.65
7
7.0
7.0
7.0
```

In [26]:

```
print(66%6)
print(66.7%6)
print(66%6.2)
print(66.5%6.1)
print(-66%6)
print(67%-6)
print(-67%-6)
print(-66.45%6)
```

```
0
0.70000000000000028
3.9999999999999982
5.50000000000000036
0
-5
-1
5.549999999999997
```

In [27]:

```
# relational (comparison) operator

# (<, >, >=, <=, ==, !=) it also returns boolean value

a = 44
b = 6
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
print(a==b)
print(a!=b)
```

False
True
False
True
False
True
False

In [29]:

```
a = "m" # it uses unicode for comparison
b = "n"
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
print(a==b)
print(a!=b)
```

True
False
True
False
False
True

In [30]:

```
a = "Nikhil" # it uses unicode for comparison
b = "nikhil"
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
print(a==b)
print(a!=b)
```

True
False
True
False
False
True

In [32]:

```
a = "Nikhim"
b = "Nikhil"
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
```

```
False
True
False
True
```

In [34]:

```
a = "Nikhim"
b = "Nikhil"
print(a==b)
print(a!=b)
```

```
False
True
```

In [35]:

```
a = "Nikhim"
b = 5
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
```

```
-----
-
TypeError                                Traceback (most recent call las
t)
Cell In[35], line 3
      1 a = "Nikhim"
      2 b = 5
----> 3 print(a<b)
      4 print(a>b)
      5 print(a<=b)

TypeError: '<' not supported between instances of 'str' and 'int'
```

In [36]:

```
a = "N"
b = 14
print(a<b) # it see datatype
print(a>b) # it see datatype
print(a<=b) # it see datatype
print(a>=b) # it see datatype
```

```
-----
-
TypeError                                Traceback (most recent call las
t)
Cell In[36], line 3
      1 a = "N"
      2 b = 14
----> 3 print(a<b)
      4 print(a>b)
      5 print(a<=b)
```

TypeError: '<' not supported between instances of 'str' and 'int'

In [38]:

```
a = "N"
b = 14
print(a==b) # it does not see datatype
print(a!=b) # it does not see datatype
```

False
True

In [39]:

```
# chaining is allowed in python

1<2<3<5<7
```

Out[39]:

True

In [40]:

```
1<2<3<5>7
```

Out[40]:

False

In [41]:

```
x= 1
y= 5
z = 9
print(x<y<z)
print(x<y>z)
print(x>y>z)
print(x>y<z)
```

True
False
False
False

In [42]:

```
# logical operators (and , or, not)
# it applies on bool

print(True and True)
print(True and False)
print(False and True)
print(False and False)
```

True
False
False
False

In [43]:

```
print(True or True)
print(True or False)
print(False or True)
print(False or False)
```

True
True
True
False

In [44]:

```
print(not True)
print(not False)
```

False
True

In [46]:

```
# python speciality
x = 3
y = 6
print(x and y)
print(x or y)
```

6
3

In [50]:

```
x = 6
y = 3
print(x and y)
print(x or y)
```

3
6

In [47]:

```
x = 0
y = 6
print(x and y)
print(x or y)
```

0
6

In [48]:

```
x = 3
y = 0
print(x and y)
print(x or y)
```

0
3

In [54]:

```
5<10>15<33   #(True>15 is False) if any of condition is false then the output is False
```

Out[54]:

False

In [52]:

```
"Red"or"White"
```

Out[52]:

'Red'

In [53]:

```
"Red"and"White"
```

Out[53]:

```
'White'
```

In [55]:

```
# assignment operator
# =, Compound assignment oprator (+=, -=, *=, /=, )

x = 66 # assign the value 66 in identifier x
print(x)
x+=2
print(x)
x-=3
print(x)
x*=2
print(x)
x/=2
print(x)
x//=2
print(x)
```

```
66
68
65
130
65.0
32.0
```

In [56]:

```
a,b,c=10,20,30
print(a)
print(b)
print(c)
```

```
10
20
30
```

In [57]:

```
x++ # not available in python
```

Cell In[57], line 1

```
x++ # not available in python
  ^
```

SyntaxError: invalid syntax

In [59]:

```
# swap two values
print(a)
print(b)
a,b=b,a
print(a)
print(b)
```

```
20
10
10
20
```

Programs on Operators in Python

In [69]:

```
# 1) write a program to remove Last digit from a given number

n = int(input("Enter a number:"))
k = n
k //=10
print("Your value is:",k)
```

```
Enter a number:254
Your value is: 25
```

In [70]:

```
# 2) write a program to get Last digit from a given number

n = int(input("Enter a number:"))
k = n
k %=10
print("Last digit is:",k)
```

```
Enter a number:25634
Last digit is: 4
```

In [73]:

```
# 3) write a program which takes a 3 digit number from user and display only its 1st dig

n = int(input("Enter a number:"))
k = n
k //=10
k //=10
print("1st digit is:",k)
```

```
Enter a number:125
1st digit is: 1
```

In [72]:

```
# 4) write a program which takes a 3 digit number from user and display only its middle digit

n = int(input("Enter a number:"))
k = n
k //=10
k %=10
print("Middle digit is:",k)
```

Enter a number:245
Middle digit is: 4

In [75]:

```
# 5) write a program to swap data of two variables

a1 = input("Enter 1st value:")
b1 = input("Enter 2nd value:")
k = a1
a1 = b1
b1 = k
print("After Swapping\n1st value:",a1)
print("2nd value:",b1)
```

Enter 1st value:256
Enter 2nd value:nikhil
After Swapping
1st value: nikhil
2nd value: 256

In [2]:

```
# ^ swap values without using temporary variable

a =10
b =20
print(a,b)
a=a+b
b=a-b
a=a-b
print(a,b)
```

10 20
20 10

In [3]:

```
a =True
b =True
print(a-b)
```

0

In [5]:

```
a =False
b =True
print(a-b)
```

-1

In [6]:

```
b =False
a =True
print(a-b)
```

1

In [8]:

```
# bitwise Operator
```

```
print(4&5) # 4->100, 5->101 it apply bit by bit (1&1)->1, (0&0)->0, (0&1)->0 ==> 100->4
print(4|5) # 4->100, 5->101 it apply bit by bit (1|1)->1, (0|0)->0, (0|1)->1 ==> 101->5
```

4

5

In [10]:

```
print(8&4)
print(8|4)
```

0

12

In [11]:

```
# identity operator
# (is)-> it checks identity
a=5
b=6
print(id(a))
print(id(b))
a is b
```

140708480324520

140708480324552

Out[11]:

False

In [13]:

```
a=5
b=5
print(id(a))
print(id(b))
a is b
```

```
140708480324520
140708480324520
```

Out[13]:

True

In [14]:

```
l =[1,5,7]
k =[2,5,6]
print(id(l))
print(id(k))
l is k
```

```
1640695947328
1640696045568
```

Out[14]:

False

In [15]:

```
l =[1,5,7]
k =[1,5,7]
print(id(l))
print(id(k))
l is k
```

```
1640696530752
1640695662720
```

Out[15]:

False

In [16]:

```
l ={1,5,7}
k ={2,5,6}
print(id(l))
print(id(k))
l is k
```

```
1640695194880
1640695805120
```

Out[16]:

False

In [17]:

```
l = {1, 5, 7}
k = {1, 5, 7}
print(id(l))
print(id(k))
l is k
```

```
1640695806240
1640695194880
```

Out[17]:

False

In [18]:

```
d1 = {"s": 1001, "u": 1002, "v": 1003}
d2 = {"a": 1009, "b": 1008, "c": 1007}
print(id(d1))
print(id(d2))
d1 is d2
```

```
1640696163712
1640696534592
```

Out[18]:

False

In [19]:

```
d1 = {"s": 1001, "u": 1002, "v": 1003}
d2 = {"s": 1001, "u": 1002, "v": 1003}
print(id(d1))
print(id(d2))
d1 is d2
```

```
1640695670912
1640695662272
```

Out[19]:

False

In [22]:

```
f1 = frozenset({"s", "u", "v"})
f2 = frozenset({"s", "u", "v"})
print(id(f1))
print(id(f2))
f1 is f2
```

```
1640696721952
1640696723072
```

Out[22]:

False

In [23]:

```
t1 = ("s", "u", "v")
t2 = ("s", "u", "v")
print(id(t1))
print(id(t2))
t1 is t2
```

```
1640702208960
1640702163456
```

Out[23]:

False

In [24]:

```
n1 =None
n2 =None
print(id(n1))
print(id(n2))
n1 is n2
```

```
140708478937840
140708478937840
```

Out[24]:

True

In [25]:

```
s1 ="nikhil"
s2 ="nikhil"
print(id(s1))
print(id(s2))
s1 is s2
```

```
1640701770480
1640701770480
```

Out[25]:

True

In [26]:

```
c1 =5+6j
c2 =5+6j
print(id(c1))
print(id(c2))
c1 is c2
```

```
1640698850480
1640698850160
```

Out[26]:

False

In [30]:

```
n1 =85.6
n2 =85.6
print(id(n1))
print(id(n2))
n1 is n2
```

```
1640695589904
1640695581616
```

Out[30]:

False

In [29]:

```
n1 =True
n2 =True
print(id(n1))
print(id(n2))
n1 is n2
```

```
140708478855712
140708478855712
```

Out[29]:

True

In [28]:

```
# in operator
print("n" in "nikhil")
```

True

In [31]:

```
for i in range(2,6):
    print(i)
```

```
2
3
4
5
```

Programs of in & is Operator

In [33]:

```
# 6) write a program to print "True" if the string entered by user containing "py"
s = input("Enter a text:")
print("py" in s)
```

Enter a text:hgpyjhuykp
True

In [42]:

```
# 7) write a program to input two strings from the user and display whether the two vari
s1 = input("Enter text1:")
s2 = input("Enter text1:")
print(id(s1))
print(id(s2))
print(s1 is s2)
```

Enter text1:nikhil
Enter text1:nikhil
1640702212912
1640702213552
False

In [41]:

```
# 8) write a program to input five integers from the user and execute this expression _<
a = int(input("enter 1st no.:"))
b = int(input("enter 2nd no.:"))
c = int(input("enter 3rd no.:"))
d = int(input("enter 4th no.:"))
e = int(input("enter 5th no.:"))

print(a<b<c>d<e)
```

enter 1st no.:5
enter 2nd no.:6
enter 3rd no.:7
enter 4th no.:5
enter 5th no.:10
True

In []: