Operators

this notes is made by- Rehyan yadav

*logical and bitwise not operator on boolean

code:

#a python code that users logical not or! on #boolean

a= not true

b=not true

print a

print b

*Bitwise NOT(or~)

Bitwise: denoting an operator in a programming language which manipulates the individuals bits in a byte or word.

*operator

*description

&

Bitwise AND

|

Bitwise OR

^

Bitwise XOR

~

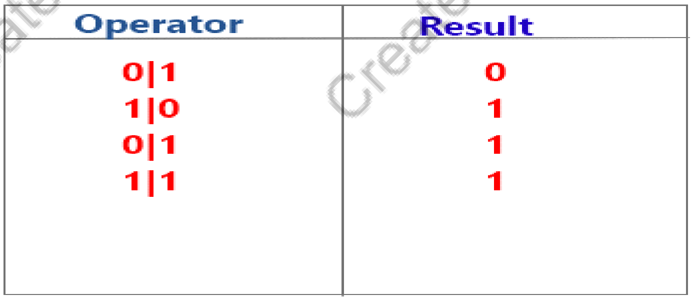
Bitwise NOT

<<

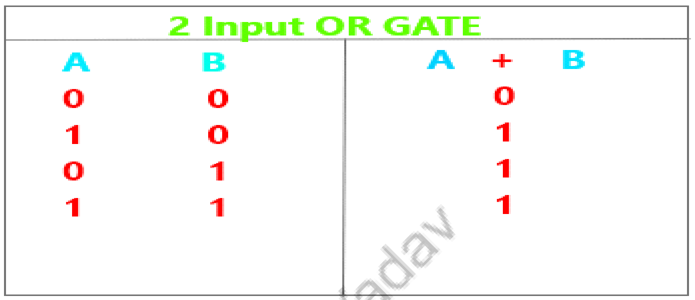
Bitwise left shift

>>

Bitwise right shift



Operator	Result
0|1	0
1|0	1
0|1	1
1|1	1



2 Input OR GATE

A	B	A + B
0	0	0
1	0	1
0	1	1
1	1	1

***Ternary operators :it is also know as condition are**

operator that evaluate something based on a condition being true or false.

syntax: [on_true] if[expression]else

[on_false]

1)simple methode to use ternary operators

code:

```
#program to demonstrate    condition operator
a,b = 10,20
#copy value of a in min if a<b else copy b
min =a    if a<b else b
print(min)
```

2)direct method by using tuples dictionary and labda

code:

```
#python code to demonstrate
a,b= 10,20
print((b,a)[a>b])
print({true:a,false:b}[a<b])
print((lambda : b,lambda: a)[a<b]))
```

3)ternary operator can be written as nested if else

code :

```
a,b= 10, 20
print("both a&b are equal ")
if a==b else "a is greater than b"
if a>b else "b is greater than a"
```

4)Increament and Decrement Operators

*for variable_name in range (start,strap,step)

i)start:Optional.an integer number specifying at
which position to start default is 0.

ii)stop:An int number specifying at which position
to end.

iii)step:Optional.An integer number specifying the
increment deafault 1.

code:

```
print('incriment for loop ")
for i in range(0,5):
  print i
print ("\n decrement for loop")
for i in range (4,-1,-1):
print i
```

* DIVISION operator in python

*code:(// real floor division)

```
  print(5//2)
```

```
print(-5//2)
```

***code:**

```
print (50/2)
print(-5.0/2)
```

python code:

```
print (5//2)
print (-5//2)
print (5.0//2)
print (-5.0//2)
```

***Any all in python**

Any : returns true if any of the the items is true

returns false if any of the items is false

syntax- any(list of literables)
code :

```
print(any([false,false,false,false]))
print(any([false,true,false,false]))
print(any([true,false,false,false]))
```

All: returns true if all the items are true or

literabels is empty.

syntax : all(list of literables)

code:

```
print(all([true,true,true,true]))

print(all([false,true,true,false]))

print(al([false,false,false]))
```

practical examples

```
#this code explains how can we use 'any' function

#on list

list 1=[]

list 2=[]

#index range from 1 to 10 to multiply

for i in range(1,11):

list 1.append(4*i)

for i in range(0,10):

list2.append(list 1[i]%5==0)

print('see whether at least one number is divisible

            by 5 in list 1=>')

print(any(list=2)
```

*INPLACE VS STANDARD OPETRATOR IN PYTHON

1)The_add_method does simple addition,takes two agrument,returns the sum and stores it in other variable without modifying any of the argument.

2)_iadd_method also takes two arguments,but it makes in -plane change in 1st arguments passed by storing the sum in it .

3)normal opertor's "add()" method,impliments "a+b"

4)inplace operators "iadd()"method,implements "a+=b"

*Immutable targets : these are the objects that can't be changed

code:

```
import operator
x=5
y=6
d=5
```

```
b=6
z=operator.add(a,b)
p=opeerator.add(x,y)
print("value after adding using normal operator:",
end=" ")
print(z)
print("value after adding using inplace operators:",
end=" ")
print(a)
print("value of first arguaments using inplace
operator:", end =" ")
print(X)
```

***Mutable targets :objects which can changes state data after inception.**

code:

```
import operator
a=[1,2,4,5]
```

```
z=operator.add(a,[1,2,3])
print("value after adding using normal operator:",
                       end=" ")
print (z)
print("value of    first argument using normal
                   operator:",end =" ")
print(a)
p=operator.iadd(a,[1,2,3])
print("value after adding using inplace operator: ",
                           end= " "     )
print(p)
print("value    of    first    arrgument using inplace
                             operator:,end=" ")
print(a)
```

Set 1

1. add(a, b) :- This functions returns addition of the given arguments.

Operation – a + b.

2. sub(a, b) :- This functions returns difference of the given arguments.

Operation – a – b.

3. mul(a, b) :- This functions returns product of the given arguments.

Operation – a * b

4. truediv(a,b) :- This functions returns division of the given arguments.

Operation – a / b.

5. floordiv(a,b) :- This functions also returns division of the given arguments. But the value is floored value i.e. returns greatest small integer.

Operation – a // b.

6. pow(a,b) :- This functions returns exponentiation of the given arguments.

Operation – a ** b.

7. mod(a,b) :- This functions returns modulus of the given arguments.

Operation – a % b

8. lt(a, b) :- This function is used to check if a is less than b or not. Returns true if a is less than b, else returns false.

Operation – a < b.

9. le(a, b) :- This function is used to check if a is less than or equal to b or not. Returns true if a is less than or equal to b, else returns false.

Operation – a <= b.

10. eq(a, b) :- This function is used to check if a is equal to b or not. Returns true if a is equal to b, else returns false.

Operation – a == b

11. gt(a,b) :- This function is used to check if a is greater than b or not. Returns true if a is greater than b, else returns false.

Operation – a > b.

12. ge(a,b) :- This function is used to check if a is greater than or equal to b or not. Returns true if a is greater than or equal to b, else returns false.

Operation – a >= b.

13. ne(a,b) :- This function is used to check if a is not equal to b or is equal. Returns true if a is not equal to b, else returns false.

Operation – a != b.

code:

```
import operator
a=4
```

```
  b=3
 print("addition of number is:",end=" ");
 print(operator.add(a,b))
 print("difference of number is:",end=" ")
 print(operator.sub(a,b))
 print("the product of number is:",end=" ");
print(operator.mul(a,b))
```

Set 1

1. setitem(ob, pos, val) :- This function is used to assign the value at a particular position in the container.

Operation – ob[pos] = val

2. delitem(ob, pos) :- This function is used to delete the value at a particular position in the container.

Operation – del ob[pos]

3. getitem(ob, pos) :- This function is used to access the value at a particular position in the container.

Operation – ob[pos]

4. setitem(ob, slice(a,b), vals) :- This function is used to set the values in a particular range in the container.

Operation – obj[a:b] = vals

5. delitem(ob, slice(a,b)) :- This function is used to delete the values from a particular range in the container.

Operation – del obj[a:b]

6. getitem(ob, slice(a,b)) :- This function is used to access the values in a particular range in the container.

Operation – obj[a:b]

7. concat(ob1,obj2) :- This function is used to concatenate two containers.

Operation – obj1 + obj2

8. contains(ob1,obj2) :- This function is used to check if obj2 in present in obj1.

Operation – obj2 in obj1

9. and_(a,b) :- This function is used to compute bitwise

and of the mentioned arguments.

Operation – a & b

10. or_(a,b) :- This function is used to compute bitwise or of the mentioned arguments.

Operation – a | b

11. xor(a,b) :- This function is used to compute bitwise xor of the mentioned arguments.

Operation – a ^ b

12. invert(a) :- This function is used to compute bitwise inversion of the mentioned argument.

Operation – ~ a

Difference between == and is operator in Python

code:

```
# python3 code to
# illustrate the
# difference between
# == and is operator
# [] is an empty list
list1 = []
list2 = []
list3=list1

if (list1 == list2):
        print("True")
else:
        print("False")

if (list1 is list2):
        print("True")
else:
        print("False")

if (list1 is list3):
        print("True")
else:
        print("False")

list3 = list3 + list2

if (list1 is list3):
        print("True")
else:
        print("False")
```

---

this notes is made by- Rehyan yadav

contact number -   8102448912

this is gmail-   ritulyadav1984@gmail.com

this number is for asking any doubt.......

if you want to   add more information ask

me or contact me.

then i will provide you the .txt file or .rtf file