**OPERATORS :**

There are 6 types of operators :

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Special Operators
7. Identity Operators
8. Membership Operators

# =============================================================================

**Arithmetic Operators :**

a) + Addition: adds two operands x + y

b) - Subtraction: subtracts two operands x - y

c) \* Multiplication: multiplies two operands x \* y

d) / Division (float): divides the first operand by the second x / y

e) // Division (floor): divides the first operand by the second x // y

f) % Modulus: returns the remainder when first operand is divided by the second x % y

# =============================================================================

print("Arithmetic Operators : \n ")

a = 10

b = 5

c = a+b

print("Addition of a+b is:" ,c)

d = a-b

print("Subtraction of a-b is:" ,d)

e = a\*b

print("Multiplication of a+b is:" ,e)

f = a/b

print("Division of a+b is:" ,f)

g = a//b

print("Division (Floor) of a+b is:" ,g)

h = a%b

print("Modulus of a+b is:" ,h)

# =============================================================================

**Relational Operators:**

1. > Greater than: True if left operand is greater than the right x > y
2. < Less than: True if left operand is less than the right x < y
3. == Equal to: True if both operands are equal x == y

d) != Not equal to - True if operands are not equal x != y

e) >= Greater than or equal to: True if left operand is greater than or equal to the right x >= y

f) <= Less than or equal to: True if left operand is less than or equal to the right x <= y

# =============================================================================

print("Relational Operators : \n ")

if a>b:

print ("a is Greater than b:",a)

else:

print(False)

if a<b:

print (a,b)

else:

print("b is less than a:",b)

i = 1

j = 1

if i==j:

print(True)

else:

print(False)

k = 2

if i!=k:

print(True)

else:

print(False)

l = 13

m = 12

if l>=m:

print(True)

else:

print(False)

if m<=l:

print(True)

else:

print(False)

# =============================================================================

**Logical Operators:**

1. and Logical AND: True if both the operands are true x and y
2. or Logical OR: True if either of the operands is true x or y

c) not Logical NOT: True if operand is false not x

# =============================================================================

print("Logical Operators : \n ")

a = True

b = True

print (a and b)

a = True

b = False

print(a or b)

a = True

b = False

print( not b)

# =============================================================================

**Bitwise Operators:**

1. & Bitwise AND x & y
2. | Bitwise OR x | y
3. ~ Bitwise NOT ~x
4. ^ Bitwise XOR x ^ y
5. >> Bitwise right shift x>>
6. << Bitwise left shift x<<

# =============================================================================

print("Bitwise Operators : \n ")

x = 5 # 000 101

y = 7 # 000 111

print("Bitwise AND of x and y is :",x & y) #100

print("Bitwise OR of x and y is :",x | y) # 111

print("Bitwise NOT of x is :", ~ x ) #111 010

print( "Bitwise XOR of x and y is :",x ^ y ) #000 010

print( "Bitwise right shift of x is :",x >> 0)

print( "Bitwise left shift of x is :",x << 1)

# # =============================================================================

**Assignment Operators:**

1. = Assign value of right side of expression to left side operand x = y + z
2. += Add AND: Add right side operand with left side operand and then assign to left operand a+=b a=a + b
3. -= Subtract AND: Subtract right operand from left operand and then assign to left operand a-=b a=a-b
4. \*= Multiply AND: Multiply right operand with left operand and then assign to left operand a\*=b a=a\*b
5. /= Divide AND: Divide left operand with right operand and then assign to left operand a/=b a=a/b
6. %= Modulus AND: Takes modulus using left and right operands and assign result to left operand a%=b a= a % b
7. //= Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand
8. a//=b a=a//b
9. \*\*= Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand a\*\*=b
10. a=a\*\*b
11. &= Performs Bitwise AND on operands and assign value to left operand a&=b a=a&b
12. |= Performs Bitwise OR on operands and assign value to left operand a|=b a=a|b
13. ^= Performs Bitwise xOR on operands and assign value to left operand a^=b a=a^b
14. >>= Performs Bitwise right shift on operands and assign value to left operand a>>=b a=a>>b
15. <<= Performs Bitwise left shift on operands and assign value to left operand a <<= b a= a << b

# # =============================================================================

print(" Assignment Operators : \n")

a=5

b=4

a += c

print("The value of Add AND (+=) is :",a)

a =5

b= 2

a -= b

print("The value of Subtract AND (-+) is :",a)

a =5

b= 2

a \*= b

print("The value of Multiply AND (\*=) is :",a)

a =5

b= 2

a /= b

print("The value of Divide AND (/=) is :" ,a)

a =5

b= 2

a %= b

print("The value of Modulus AND (%=) is :",a)

a =5

b= 2

a //= b

print("The value of Divide (Floor) (//=) is",a)

a =5

b= 2

a \*\*= b

print("The value of Exponent AND (\*\*=) is",a)

a =5

b= 2

a &= b

print("The value of Bitwise AND (&=) is" ,a)

a =5

b= 2

a |= b

print("The value of Bitwise OR (|=) is", a)

a =5

b= 2

a^=b

print("The value of Bitwise XOR (^=) is", a)

a =5

b= 2

a<<=b

print("The value of Bitwise Right Shift (<<=) is", a)

a =5

b= 2

a>>=b

print("The value of Bitwise Left Shift (>>=) is" ,a)

# =============================================================================

**Special Operators: identity and Membership**

1. **Identity operators** :

* is and is not are the identity operators both are used to check if 2 values are located on the same part of the memory. Two variables that are equal does not imply that they are identical
* is True if the operands are identical
* is not True if the operands are not identical

# =============================================================================

print("Identity Operators : \n")

a1 = 3

b1 = 3

a2 = 'GeeksforGeeks'

b2 = 'GeeksforGeeks'

a3 = [1,2,3]

b3 = [1,2,3]

print(a1 is not b1)

print(a2 is b2)

# Output is False, since lists are mutable.

print(a3 is b3)

# =============================================================================

# Membership Operator:

# - in and not in are the membership operators; used to test whether a value or variable is in a sequence.

# in True if value is found in the sequence

# not in True if value is not found in the sequence

# =============================================================================

print("Membership Operator : \n ")

x = 'Geeks for Geeks'

y = {3:'a',4:'b'}

print('G' in x)

print('geeks' not in x)

print('Geeks' not in x)

print(3 in y)

print('b' in y)

# =============================================================================

# Control Flow Statements :

# Conditions : if,elif,elif ladder and else

# Iterations : while,for,break and continue

# =============================================================================

print("Control Flow Statements : \n" )

a= 4

b =5

if a>b:

print('A is bigger',a)

else:

print('B is bigger',b ,"\n")

print("if elif else example : \n")

a=5

b=7

c=9

if a>b and a>c:

print('A is Bigger',a)

elif b>c:

print('B is Bigger',b)

else:

print('C is Bigger',c,"\n")

print("elif ladder : \n")

x = -15

if x == 0:

print(x, "is zero")

elif x > 0:

print(x, "is positive")

elif x < 0:

print(x, "is negative")

else:

print(x, "is unlike anything I've ever seen... \n")

print("While Loop : \n")

n=1

while True:

print (n)

n+=1

if n==5:

break

print("After Break \n")

print("For Loop Example: \n")

for str in "Python":

if str == "t":

break

print(str)

print("Exit from loop \n")

print("Continue keyword example using for while : \n")

n=0

while n < 5:

n+=1

if n==3:

continue

print (n, "\n")

print("Continue keyword example using for loop: \n")

n=0

for n in range(5):

n+=1

if n==3:

continue

print(n)

print("Loop Over \n")

# For loop example

print("For Loop : \n")

word = "Android"

for letter in word:

print(letter)

# While loop

print("\n While Loop : \n" )

a = 0

while a < 10:

a = a + 1

print(a)