*# Operators in Python  
# 1) Arithmatic Operators  
# 2)Assignmint Operators  
# 3)Comparison Operators  
# 4)Logical Operators  
# 5)Identity Operators  
# 6)MemberShip Operators  
#Bitwise operators  
  
# 1) Aritmetic Operators  
'''  
Arithmetic operators are the symbols that represent arithmetic math operations.  
Examples include + (addition operator), - (subtraction operator),  
\* (multiplication operator), and / (division operator)  
'''*print(**"Aritmetic Operators"**)  
print(**"4 + 8 is"**, 4 + 8) *#Addition Operator*print(**"4 - 8 is"**, 4 - 8) *#Subtraction Operators*print(**"4 \* 8 is"**, 4 \* 8) *#Multiplication Operators*print(**"4 / 8 is"**, 4 / 8) *#Division Operators*print(**"23 // 8 is"**, 23 // 8) *#Double Division operators (Point ka bad wali value skip karta ha*print(**"4 \*\* 8 is"**, 4 \*\* 8) *#Exponantial Operators*print(**"4 % 8 is"**, 4 % 8) *#Remainder Operators  
  
# 2) Assignment Operators  
# Assignment operators are used to assigning value to a variable.  
# The left side operand of the assignment operator is a variable and right side operand of the  
# assignment operator is a value. ... This operator is used to assign the value on the right to  
# the variable on the left.*print(**"Assignment Operator"**)  
n = 10 *#(n is variable, = is Assignment Operator, 10 is jis ko ham na assign karna ha)*print(n)  
n += 5 *#(+= bhi aik assignment operator ha is ko ya purana n ma add kar da ga isi tarha baki bhi ha -,\*,/...)*print(n)  
n -= 6  
print(n)  
  
*# 3)Comparison operators***'''  
Comparison operators — operators that compare values and return true or false .  
The operators include: > , < , >= , <= , === , and !== . Logical operators —   
operators that combine multiple boolean expressions or values and provide a  
single boolean output.  
'''**print(**"Comparision Operators"**)  
i = 12  
print(i==13) *#(Comparision operators compare karta ha do values ko agar comparision sahi ho g ato True or agar ghalt ho ga to False return kara ga*print(i!=13)  
print(i>13)  
print(i<13)  
print(i>=13)  
print(i<=13)  
  
*# 4) Logical Operators***"""  
A logical operator is a symbol or word used to connect two or more expressions such that the value of the compound  
expression produced depends only on that of the original expressions and on the meaning of the operator.   
Common logical operators include AND, OR, and NOT.  
"""**print(**"Logical Operators"**)  
a = **True**b = **False**print(a **and** b) *#and ka mtlb ha ka agar dono True ho ga to Ture print ho ga lkn agar aik bhi False ho ga to False print hoga*print(a **or** b) *#or ka mtlb ha ka agar dono ma sa koi aik bhi True ho ga to Ture print ho ga  
  
# 5) Identity Operators***'''  
The identity operators in Python are used to determine whether a value is of a certain class or type.  
They are usually used to determine the type of data a certain variable contains.   
For example, type(3) is int evaluates to True because 3 is indeed an integer number.  
'''**print(**"Identity Operators"**)  
a = 49  
b = 10  
print(a **is** b) *#a is b ka mtlb k agar a b ho ga to True Return kara ga warna False*print(a **is not** b) *#a is not b ka mtlb ha k agar a b nhi ho ga to True Return kara ga warna False  
  
#6 ) MemberShip Operators***'''  
A Membership Operator in Python can be defined as being an operator that is used to validate the membership of a value.  
This operator is used to test memberships in variables such as strings, integers as well as tuples.  
 Membership Operators as a whole contain a number of different operators.  
'''**print(**"MemberShip Operators"**)  
List = [1,325,4,688,456,346]  
print(688 **and** 4 **in** List)  
  
*# 7) Bitwise Operators***'''  
In Python, bitwise operators are used to perform bitwise calculations on integers.  
The integers are first converted into binary and then operations are performed on bit by bit, hence the name bitwise operators.  
Then the result is returned in decimal format. ... Bitwise AND operator: Returns 1 if both the bits are 1 else 0.  
'''**print(**"Bitwise Operators"**)