

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> ''' CS Practical '''
' CS Practical '
>>> ''' Task : Execution of Expressions involving Arithmetic, Relational, Logical
and Bitwise operators in shell window of Python IDLE.'''
' Task : Execution of Expressions involving Arithmetic, Relational, Logical and
Bitwise operators in shell window of Python IDLE.'
>>>
>>>
>>> '''Arithmetic Operators'''
'Arithmetic Operators'
>>>
>>> #Addition
>>>
>>> 1 +2
3
>>>
>>> # Subtraction
>>> 1 - 2
-1
>>> # Multiplication
>>> 1*2
2
>>> #Division
>>> 1/2
0.5
>>>
>>> #Integer Division
>>> 1//2
0
>>>
>>> #Modulo
>>> 1 % 2
1
>>>
>>> # Exponentiation
>>> 1**2
1
>>>
>>> ''' Relational Operators'''
' Relational Operators'
>>> #Greater than
>>> 2 > 1
True
>>>
>>> # Less than
>>> 2 < 1
False
```

```

>>>
>>> #Equal to
>>> 2 == 1
False
>>>
>>> # Greater than or equal to
>>> 2 >=1
True
>>>
>>> # Less than or equal to
>>> 2 <=1
False
>>>
>>> # Not equal to
>>> 2!=1
True
>>>
>>> '''Bitwise Operators'''
'Bitwise Operators'
>>> #Bitwise AND
>>> bin(10)
'0b1010'
>>> bin(12)
'0b1100'
>>> 10 & 12
8
>>> bin(8)
'0b1000'
>>>
>>> #Bitwise OR
>>> bin(10)
'0b1010'
>>> bin(12)
'0b1100'
>>> 10 | 12
14
>>> bin(14)
'0b1110'
>>>
>>> #Bitwise NOT
>>> bin(10)
'0b1010'
>>> ~10
-11
>>> bin(-11)
'-0b1011'
>>> # bin(10) = 01010 ,Bitwise NOT (01010) = 10100 and 10100 is -11 in 2's
complement form
>>> '''bin(10) = 01010 ; Bitwise NOT(01010) = 10101 and 10101 is -11 in 2's
Complement representation.'''

```

```

"bin(10) = 01010 ; Bitwise NOT(01010) = 10101 and 10101 is -11 in 2's Complement
representation."
>>> #Bitwise XOR
>>> bin(10)
'0b1010'
>>> bin(12)
'0b1100'
>>> 10 ^ 12
6
>>> bin(6)
'0b110'
>>> ''' bitwise XOR (1010 , 1100) = 0110 which is equal to 6'''
' bitwise XOR (1010 , 1100) = 0110 which is equal to 6'
>>> #Biwise right shift
>>> bin(10)
'0b1010'
>>> 10>>
SyntaxError: invalid syntax
>>> 10>>2
2
>>> bin(2)
'0b10'
>>> '''bin(10) = 1010, Shifting two places to the right, we get 0010, which is 2'''
'bin(10) = 1010, Shifting two places to the right, we get 0010, which is 2'
>>> #Bitwise left shift
>>> bin(10)
'0b1010'
>>> 10<<2
40
>>> bin(40)
'0b101000'
>>> '''bin(10) = 1010, Shifting two places to the left, we get 101000, which is
40.'''
'bin(10) = 1010, Shifting two places to the left, we get 101000, which is 40.'
>>> '#####'
'#####'
>>> '''Logical Operators'''
'Logical Operators'
>>> # And Operator
>>> True and False
False
>>>
>>> # Or Operator
>>> True or False
True
>>> # Not Operator
>>> not True
False
>>>

```