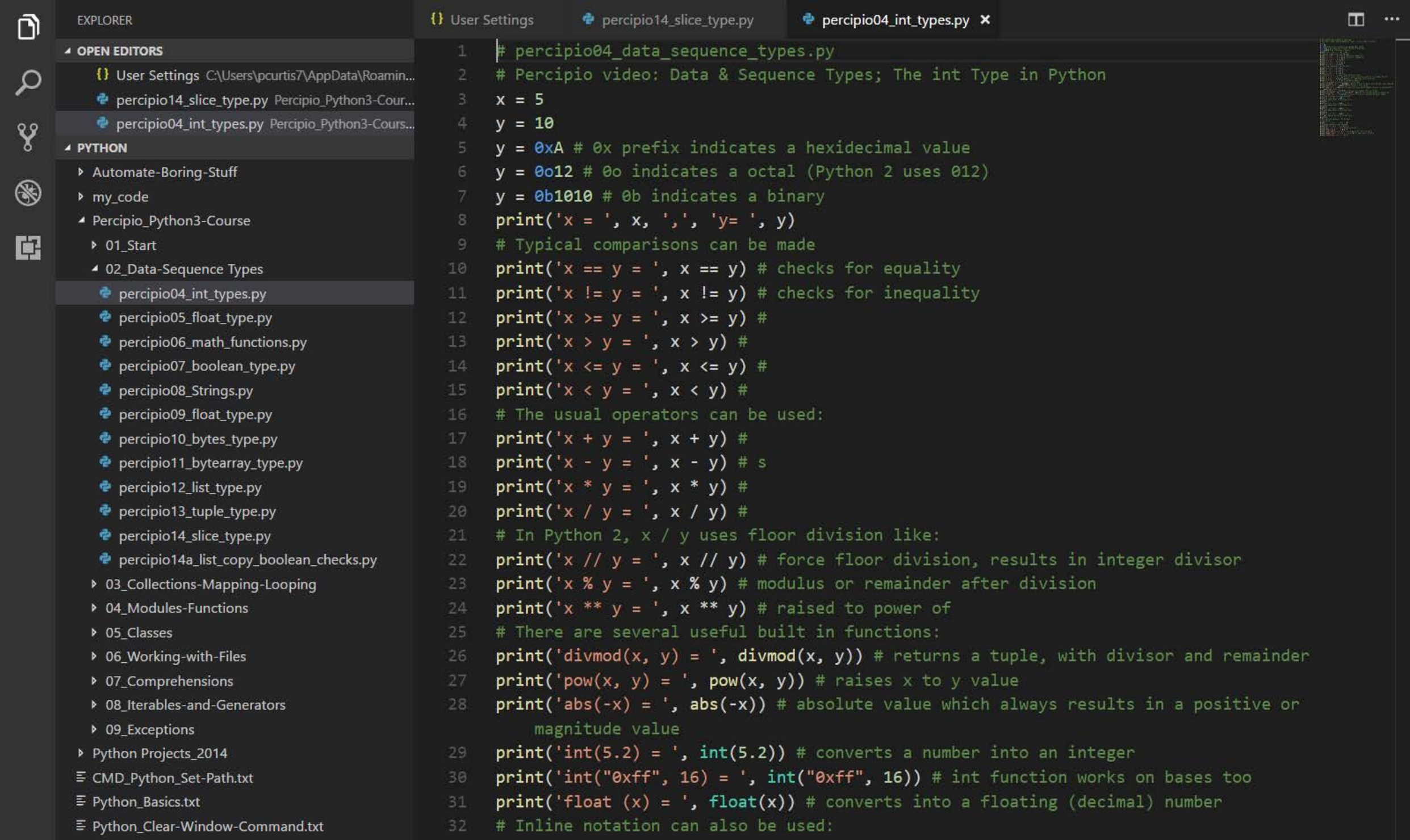
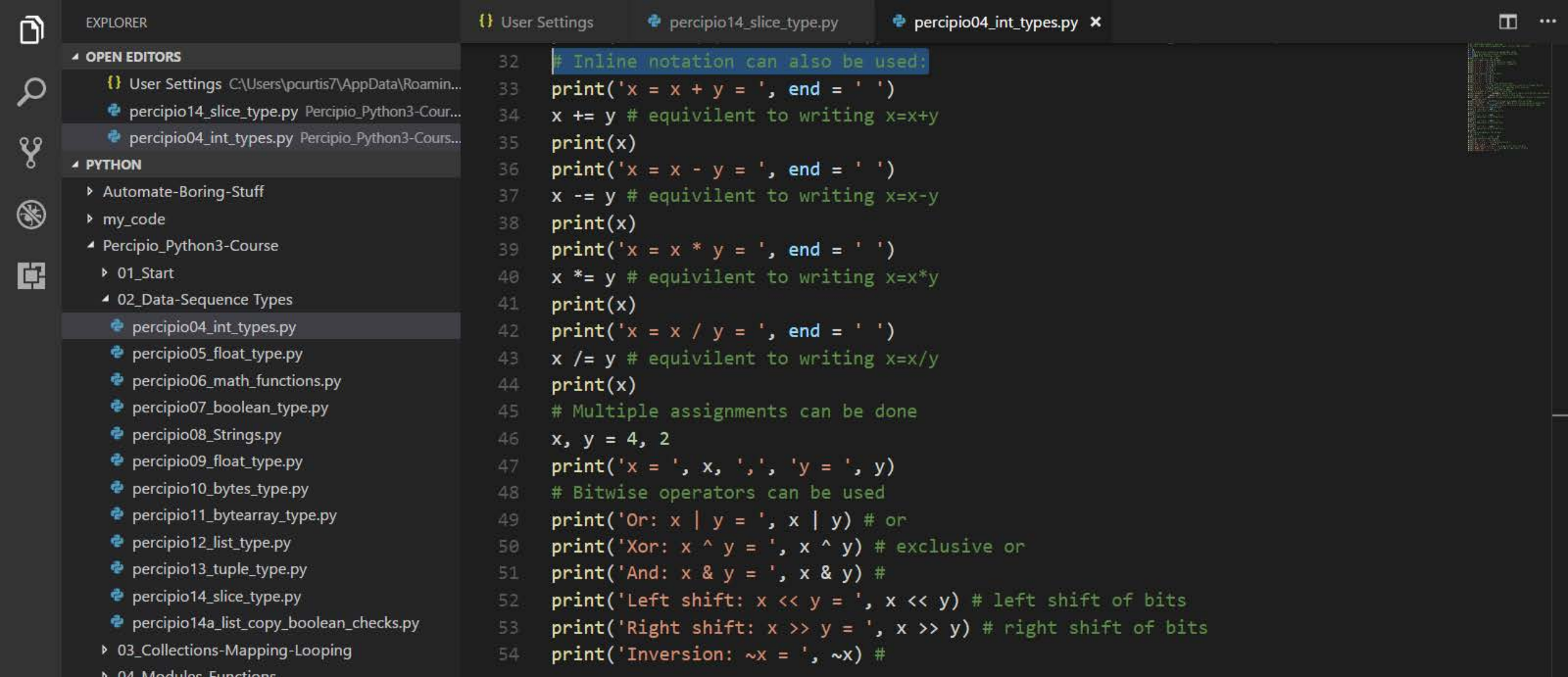

EXPLORER
User Settings
percipio14_slice_type.py
percipio04_int_types.py
OPEN EDITORS
User Settings C:\Users\pcurtis7\AppData\Roamin...
percipio14_slice_type.py Percipio_Python3-Cour...
percipio04_int_types.py Percipio_Python3-Cours...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
percipio04_int_types.py
percipio05_float_type.py
percipio06_math_functions.py
percipio07_boolean_type.py
percipio08_Strings.py
percipio09_float_type.py
percipio10_bytes_type.py
percipio11_bytearray_type.py
percipio12_list_type.py
percipio13_tuple_type.py
percipio14_slice_type.py
percipio14a_list_copy_boolean_checks.py
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
09_Exceptions
Python Projects_2014
CMD_Python_Set-Path.txt
Python_Basics.txt
Python_Clear-Window-Command.txt
1 # percipio04_data_sequence_types.py
2 # Percipio video: Data & Sequence Types; The int Type in Python
3 x = 5
4 y = 10
5 y = 0xA # 0x prefix indicates a hexidecimal value
6 y = 0o12 # 0o indicates a octal (Python 2 uses 012)
7 y = 0b1010 # 0b indicates a binary
8 print('x = ', x, ',', 'y= ', y)
9 # Typical comparisons can be made
10 print('x == y = ', x == y) # checks for equality
11 print('x != y = ', x != y) # checks for inequality
12 print('x >= y = ', x >= y) #
13 print('x > y = ', x > y) #
14 print('x <= y = ', x <= y) #
15 print('x < y = ', x < y) #
16 # The usual operators can be used:
17 print('x + y = ', x + y) #
18 print('x - y = ', x - y) # s
19 print('x * y = ', x * y) #
20 print('x / y = ', x / y) #
21 # In Python 2, x / y uses floor division like:
22 print('x // y = ', x // y) # force floor division, results in integer divisor
23 print('x % y = ', x % y) # modulus or remainder after division
24 print('x ** y = ', x ** y) # raised to power of
25 # There are several useful built in functions:
26 print('divmod(x, y) = ', divmod(x, y)) # returns a tuple, with divisor and remainder
27 print('pow(x, y) = ', pow(x, y)) # raises x to y value
28 print('abs(-x) = ', abs(-x)) # absolute value which always results in a positive or
magnitude value
29 print('int(5.2) = ', int(5.2)) # converts a number into an integer
30 print('int("0xff", 16) = ', int("0xff", 16)) # int function works on bases too
31 print('float (x) = ', float(x)) # converts into a floating (decimal) number
32 # Inline notation can also be used:


EXPLORER
OPEN EDITORS
User Settings C:\Users\pcurtis7\AppData\Roamin...
percipio14_slice_type.py Percipio_Python3-Cour...
percipio04_int_types.py Percipio_Python3-Cours...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
percipio04_int_types.py
percipio05_float_type.py
percipio06_math_functions.py
percipio07_boolean_type.py
percipio08_Strings.py
percipio09_float_type.py
percipio10_bytes_type.py
percipio11_bytearray_type.py
percipio12_list_type.py
percipio13_tuple_type.py
percipio14_slice_type.py
percipio14a_list_copy_boolean_checks.py
03_Collections-Mapping-Looping
User Settings
percipio14_slice_type.py
percipio04_int_types.py
32 # Inline notation can also be used:
33 print('x = x + y = ', end = ' ')
34 x += y # equivilent to writing x=x+y
35 print(x)
36 print('x = x - y = ', end = ' ')
37 x -= y # equivilent to writing x=x-y
38 print(x)
39 print('x = x * y = ', end = ' ')
40 x *= y # equivilent to writing x=x*y
41 print(x)
42 print('x = x / y = ', end = ' ')
43 x /= y # equivilent to writing x=x/y
44 print(x)
45 # Multiple assignments can be done
46 x, y = 4, 2
47 print('x = ', x, ',', 'y = ', y)
48 # Bitwise operators can be used
49 print('Or: x | y = ', x | y) # or
50 print('Xor: x ^ y = ', x ^ y) # exclusive or
51 print('And: x & y = ', x & y) #
52 print('Left shift: x << y = ', x << y) # left shift of bits
53 print('Right shift: x >> y = ', x >> y) # right shift of bits
54 print('Inversion: ~x = ', ~x) #