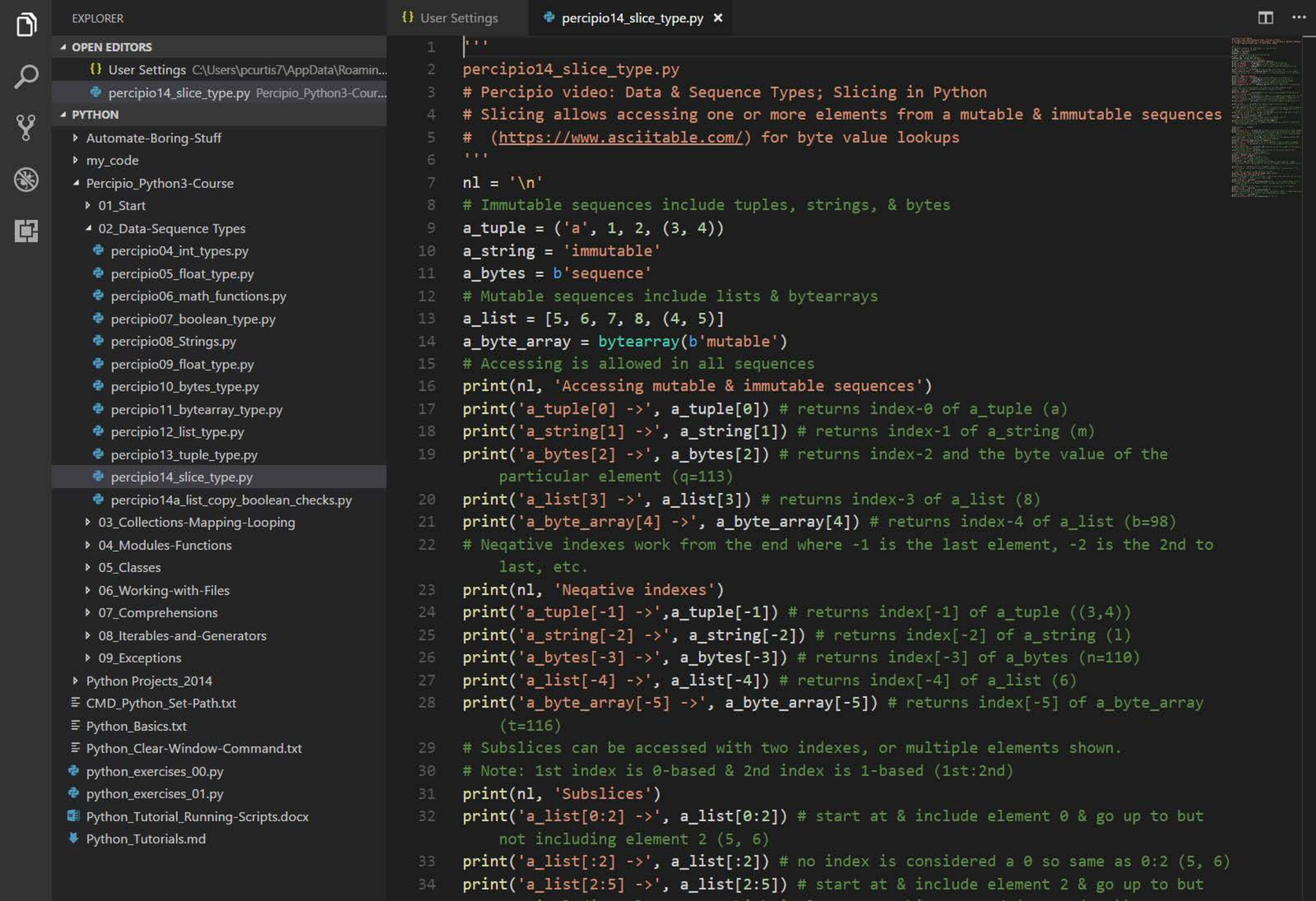
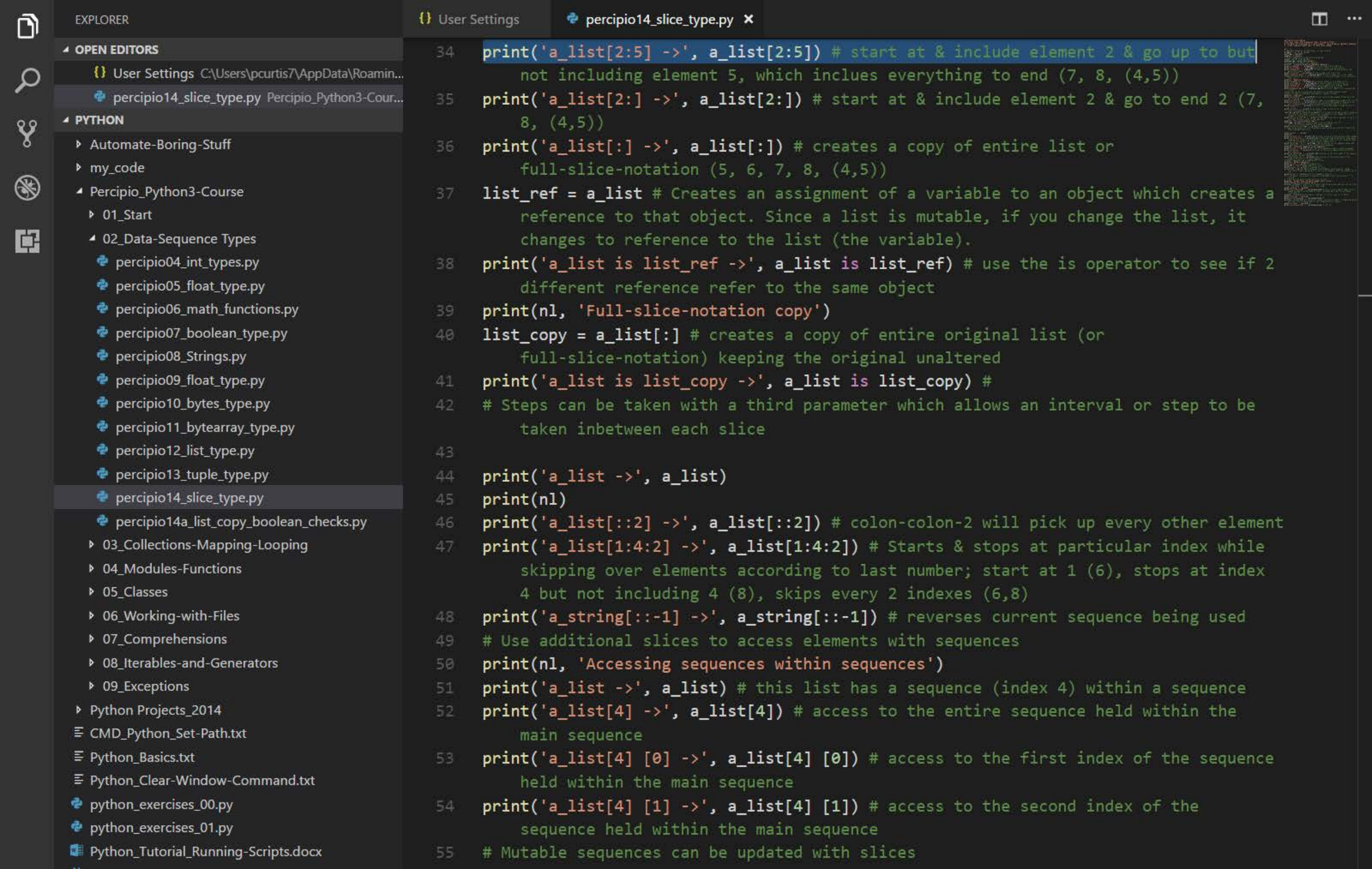
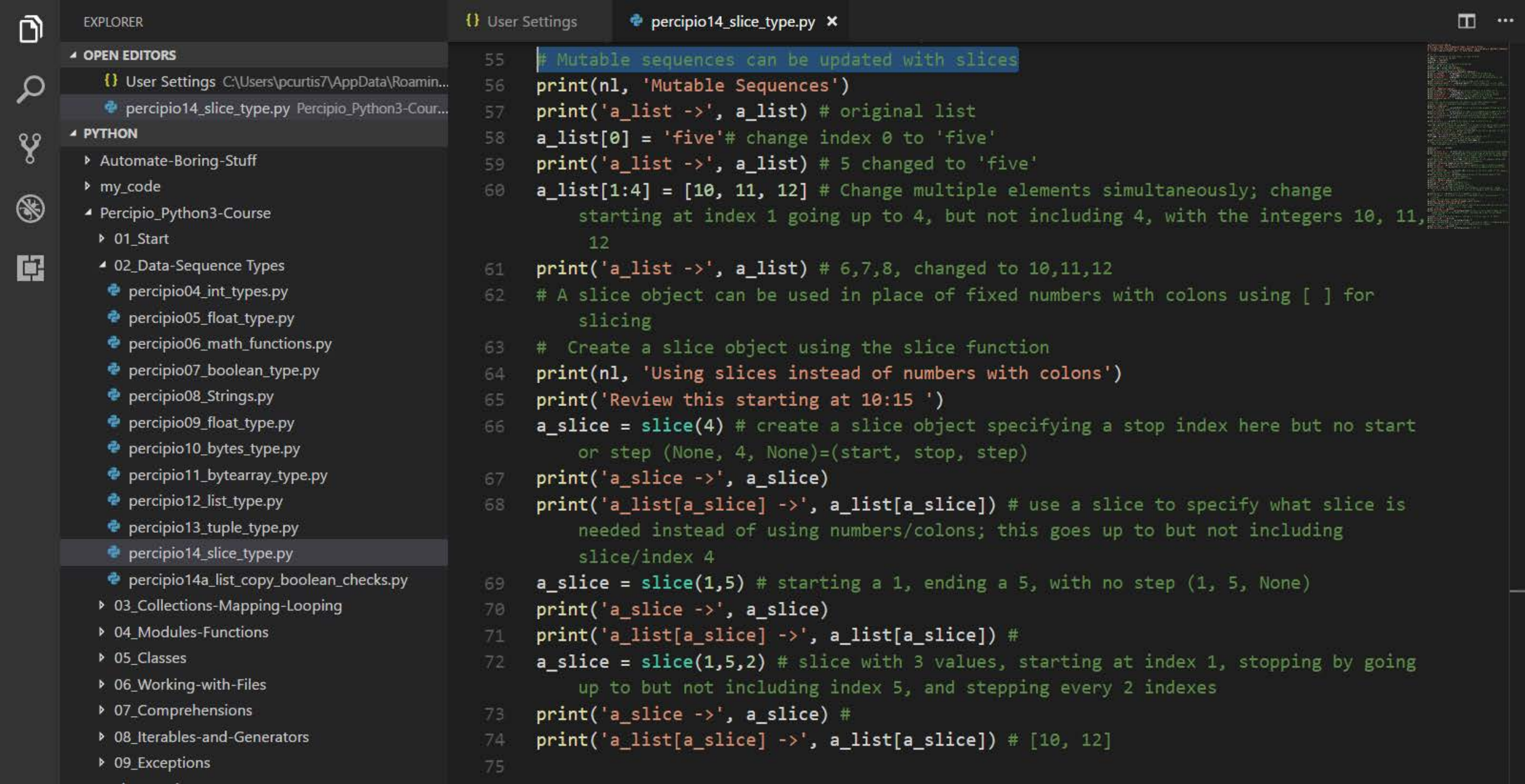

EXPLORER
OPEN EDITORS
User Settings C:\Users\pcurtis7\AppData\Roamin...
percipio14_slice_type.py Percipio_Python3-Cour...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
percipio04_int_types.py
percipio05_float_type.py
percipio06_math_functions.py
percipio07_boolean_type.py
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
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
User Settings
percipio14_slice_type.py
'''
percipio14_slice_type.py
Percipio video: Data & Sequence Types; Slicing in Python
Slicing allows accessing one or more elements from a mutable & immutable sequences
(https://www.asciitable.com/) for byte value lookups
'''
nl = '\n'
# Immutable sequences include tuples, strings, & bytes
a_tuple = ('a', 1, 2, (3, 4))
a_string = 'immutable'
a_bytes = b'sequence'
# Mutable sequences include lists & bytearrays
a_list = [5, 6, 7, 8, (4, 5)]
a_byte_array = bytearray(b'mutable')
# Accessing is allowed in all sequences
print(nl, 'Accessing mutable & immutable sequences')
print('a_tuple[0] ->', a_tuple[0]) # returns index-0 of a_tuple (a)
print('a_string[1] ->', a_string[1]) # returns index-1 of a_string (m)
print('a_bytes[2] ->', a_bytes[2]) # returns index-2 and the byte value of the particular element (q=113)
print('a_list[3] ->', a_list[3]) # returns index-3 of a_list (8)
print('a_byte_array[4] ->', a_byte_array[4]) # returns index-4 of a_list (b=98)
# Negative indexes work from the end where -1 is the last element, -2 is the 2nd to last, etc.
print(nl, 'Negative indexes')
print('a_tuple[-1] ->',a_tuple[-1]) # returns index[-1] of a_tuple ((3,4))
print('a_string[-2] ->', a_string[-2]) # returns index[-2] of a_string (l)
print('a_bytes[-3] ->', a_bytes[-3]) # returns index[-3] of a_bytes (n=110)
print('a_list[-4] ->', a_list[-4]) # returns index[-4] of a_list (6)
print('a_byte_array[-5] ->', a_byte_array[-5]) # returns index[-5] of a_byte_array (t=116)
# Subslices can be accessed with two indexes, or multiple elements shown.
# Note: 1st index is 0-based & 2nd index is 1-based (1st:2nd)
print(nl, 'Subslices')
print('a_list[0:2] ->', a_list[0:2]) # start at & include element 0 & go up to but not including element 2 (5, 6)
print('a_list[:2] ->', a_list[:2]) # no index is considered a 0 so same as 0:2 (5, 6)
print('a_list[2:5] ->', a_list[2:5]) # start at & include element 2 & go up to but


EXPLORER
User Settings
percipio14_slice_type.py
OPEN EDITORS
User Settings C:\Users\pcurtis7\AppData\Roamin...
percipio14_slice_type.py Percipio_Python3-Cour...
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
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
34 print('a_list[2:5] ->', a_list[2:5]) # start at & include element 2 & go up to but not including element 5, which inclues everything to end (7, 8, (4,5))
35 print('a_list[2:] ->', a_list[2:]) # start at & include element 2 & go to end 2 (7, 8, (4,5))
36 print('a_list[:] ->', a_list[:]) # creates a copy of entire list or full-slice-notation (5, 6, 7, 8, (4,5))
37 list_ref = a_list # Creates an assignment of a variable to an object which creates a reference to that object. Since a list is mutable, if you change the list, it changes to reference to the list (the variable).
38 print('a_list is list_ref ->', a_list is list_ref) # use the is operator to see if 2 different reference refer to the same object
39 print(nl, 'Full-slice-notation copy')
40 list_copy = a_list[:] # creates a copy of entire original list (or full-slice-notation) keeping the original unaltered
41 print('a_list is list_copy ->', a_list is list_copy) #
42 # Steps can be taken with a third parameter which allows an interval or step to be taken inbetween each slice
43
44 print('a_list ->', a_list)
45 print(nl)
46 print('a_list[::2] ->', a_list[::2]) # colon-colon-2 will pick up every other element
47 print('a_list[1:4:2] ->', a_list[1:4:2]) # Starts & stops at particular index while skipping over elements according to last number; start at 1 (6), stops at index 4 but not including 4 (8), skips every 2 indexes (6,8)
48 print('a_string[::-1] ->', a_string[::-1]) # reverses current sequence being used
49 # Use additional slices to access elements with sequences
50 print(nl, 'Accessing sequences within sequences')
51 print('a_list ->', a_list) # this list has a sequence (index 4) within a sequence
52 print('a_list[4] ->', a_list[4]) # access to the entire sequence held within the main sequence
53 print('a_list[4] [0] ->', a_list[4] [0]) # access to the first index of the sequence held within the main sequence
54 print('a_list[4] [1] ->', a_list[4] [1]) # access to the second index of the sequence held within the main sequence
55 # Mutable sequences can be updated with slices


EXPLORER
OPEN EDITORS
User Settings C:\Users\pcurtis7\AppData\Roamin...
percipio14_slice_type.py Percipio_Python3-Cour...
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
User Settings
percipio14_slice_type.py
55 # Mutable sequences can be updated with slices
56 print(nl, 'Mutable Sequences')
57 print('a_list ->', a_list) # original list
58 a_list[0] = 'five'# change index 0 to 'five'
59 print('a_list ->', a_list) # 5 changed to 'five'
60 a_list[1:4] = [10, 11, 12] # Change multiple elements simultaneously; change starting at index 1 going up to 4, but not including 4, with the integers 10, 11, 12
61 print('a_list ->', a_list) # 6,7,8, changed to 10,11,12
62 # A slice object can be used in place of fixed numbers with colons using [ ] for slicing
63 # Create a slice object using the slice function
64 print(nl, 'Using slices instead of numbers with colons')
65 print('Review this starting at 10:15 ')
66 a_slice = slice(4) # create a slice object specifying a stop index here but no start or step (None, 4, None)=(start, stop, step)
67 print('a_slice ->', a_slice)
68 print('a_list[a_slice] ->', a_list[a_slice]) # use a slice to specify what slice is needed instead of using numbers/colons; this goes up to but not including slice/index 4
69 a_slice = slice(1,5) # starting a 1, ending a 5, with no step (1, 5, None)
70 print('a_slice ->', a_slice)
71 print('a_list[a_slice] ->', a_list[a_slice]) #
72 a_slice = slice(1,5,2) # slice with 3 values, starting at index 1, stopping by going up to but not including index 5, and stepping every 2 indexes
73 print('a_slice ->', a_slice) #
74 print('a_list[a_slice] ->', a_list[a_slice]) # [10, 12]
75