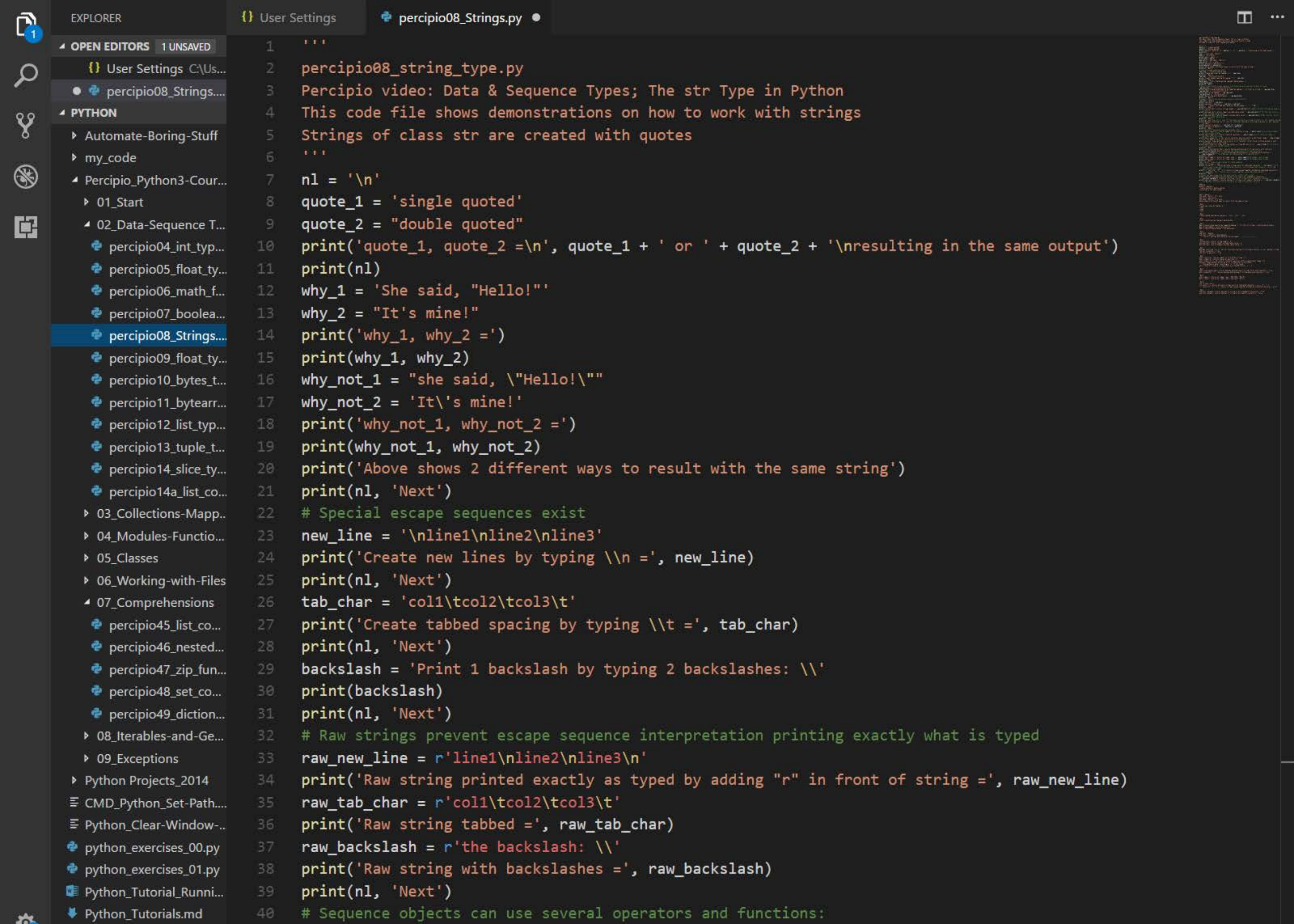
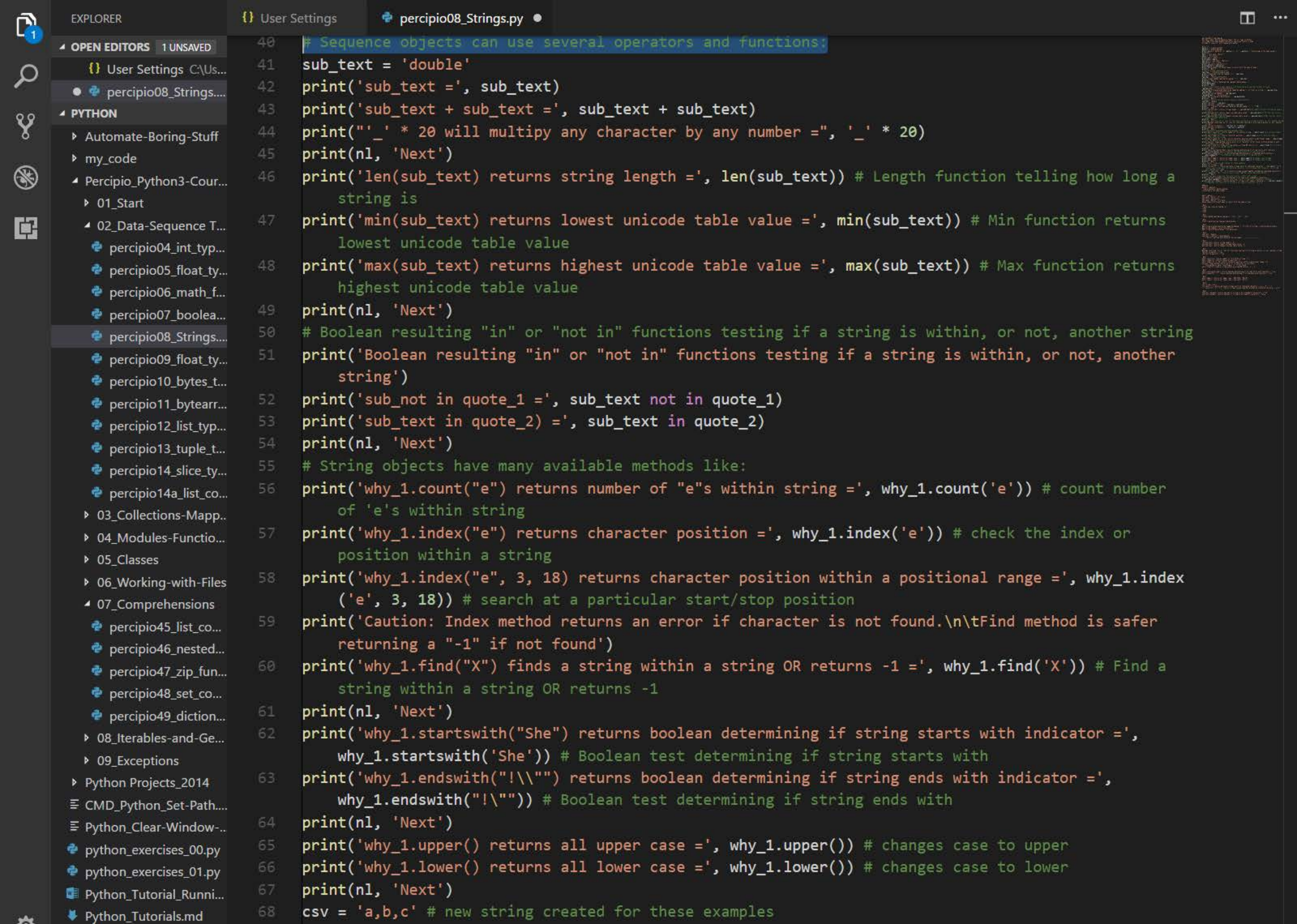
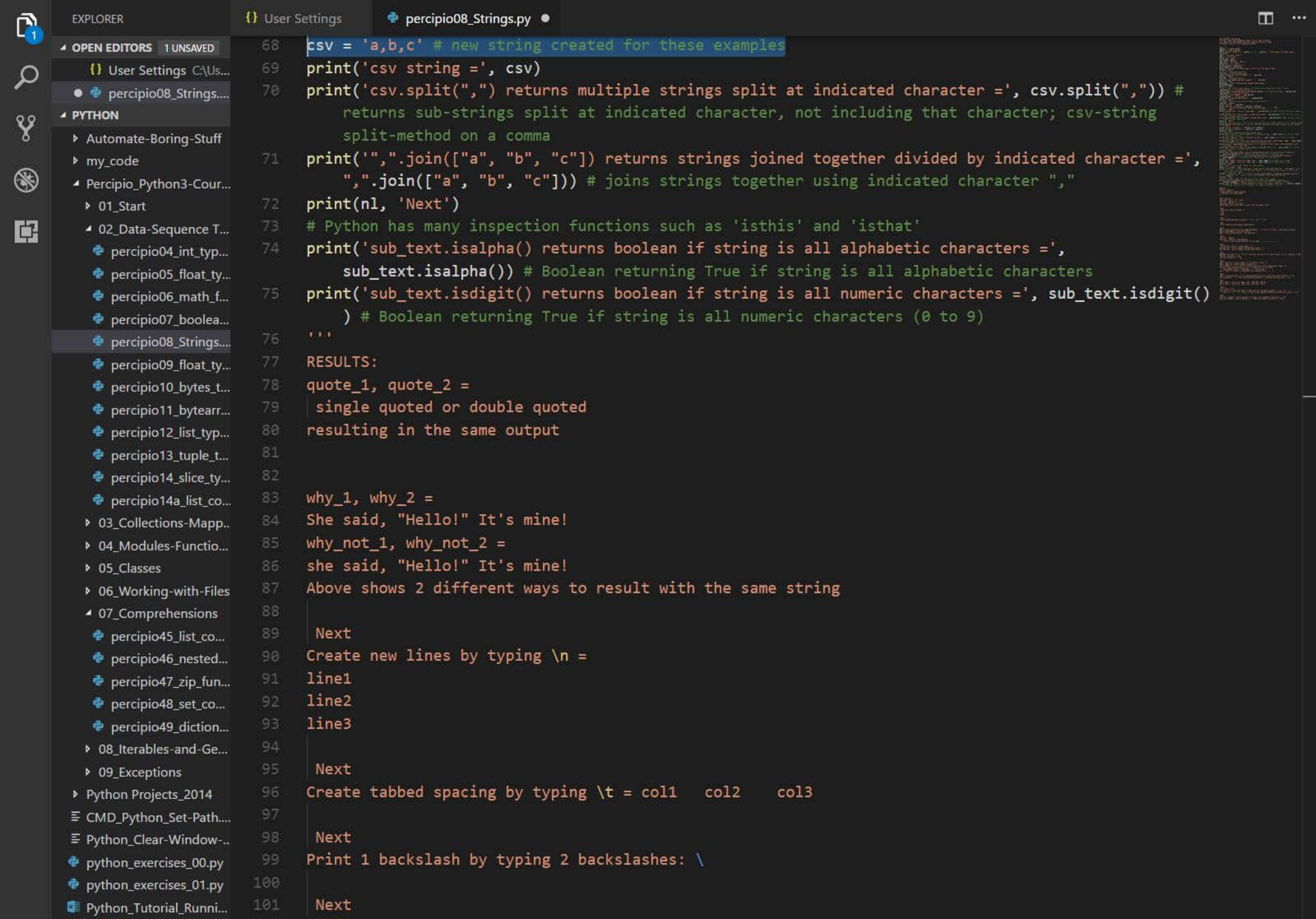
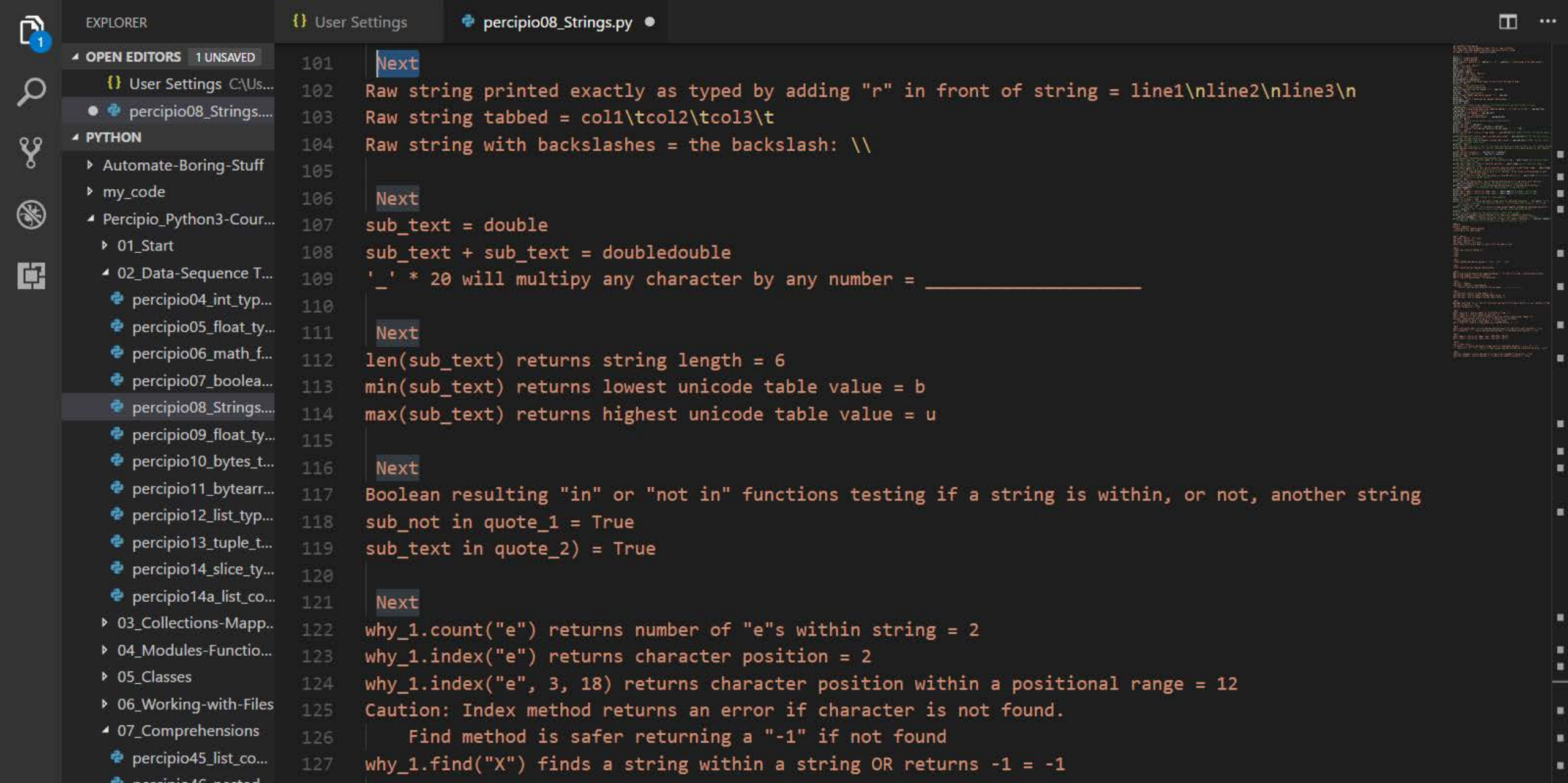
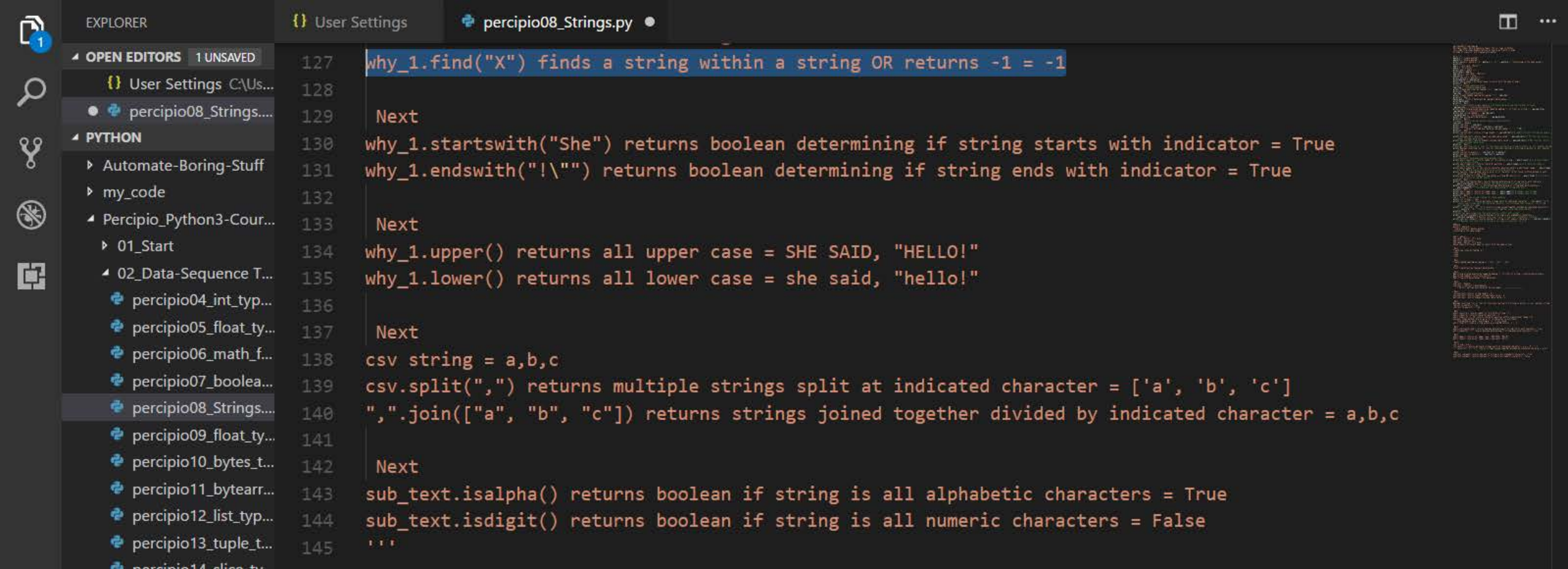

EXPLORER
{} User Settings
percipio08_Strings.py
OPEN EDITORS 1 UNSAVED
{} User Settings C:\Us...
percipio08_Strings....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Cour...
01_Start
02_Data-Sequence T...
percipio04_int_typ...
percipio05_float_ty...
percipio06_math_f...
percipio07_boolea...
percipio08_Strings....
percipio09_float_ty...
percipio10_bytes_t...
percipio11_bytearr...
percipio12_list_typ...
percipio13_tuple_t...
percipio14_slice_ty...
percipio14a_list_co...
03_Collections-Mapp..
04_Modules-Functio...
05_Classes
06_Working-with-Files
07_Comprehensions
percipio45_list_co...
percipio46_nested...
percipio47_zip_fun...
percipio48_set_co...
percipio49_diction...
08_Iterables-and-Ge...
09_Exceptions
Python Projects_2014
CMD_Python_Set-Path....
Python_Clear-Window-...
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Runni...
Python_Tutorials.md
'''
percipio08_string_type.py
Percipio video: Data & Sequence Types; The str Type in Python
This code file shows demonstrations on how to work with strings
Strings of class str are created with quotes
'''
nl = '\n'
quote_1 = 'single quoted'
quote_2 = "double quoted"
print('quote_1, quote_2 =\n', quote_1 + ' or ' + quote_2 + '\nresulting in the same output')
print(nl)
why_1 = 'She said, "Hello!"'
why_2 = "It's mine!"
print('why_1, why_2 =')
print(why_1, why_2)
why_not_1 = "she said, \"Hello!\""
why_not_2 = 'It\'s mine!'
print('why_not_1, why_not_2 =')
print(why_not_1, why_not_2)
print('Above shows 2 different ways to result with the same string')
print(nl, 'Next')
# Special escape sequences exist
new_line = '\nline1\nline2\nline3'
print('Create new lines by typing \\n =', new_line)
print(nl, 'Next')
tab_char = 'col1\tcol2\tcol3\t'
print('Create tabbed spacing by typing \\t =', tab_char)
print(nl, 'Next')
backslash = 'Print 1 backslash by typing 2 backslashes: \\'
print(backslash)
print(nl, 'Next')
# Raw strings prevent escape sequence interpretation printing exactly what is typed
raw_new_line = r'line1\nline2\nline3\n'
print('Raw string printed exactly as typed by adding "r" in front of string =', raw_new_line)
raw_tab_char = r'col1\tcol2\tcol3\t'
print('Raw string tabbed =', raw_tab_char)
raw_backslash = r'the backslash: \\'
print('Raw string with backslashes =', raw_backslash)
print(nl, 'Next')
# Sequence objects can use several operators and functions:


EXPLORER
User Settings
percipio08_Strings.py
OPEN EDITORS 1 UNSAVED
User Settings C:\Us...
percipio08_Strings....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Cour...
01_Start
02_Data-Sequence T...
percipio04_int_typ...
percipio05_float_ty...
percipio06_math_f...
percipio07_boolea...
percipio08_Strings....
percipio09_float_ty...
percipio10_bytes_t...
percipio11_bytearr...
percipio12_list_typ...
percipio13_tuple_t...
percipio14_slice_ty...
percipio14a_list_co...
03_Collections-Mapp...
04_Modules-Functio...
05_Classes
06_Working-with-Files
07_Comprehensions
percipio45_list_co...
percipio46_nested...
percipio47_zip_fun...
percipio48_set_co...
percipio49_diction...
08_Iterables-and-Ge...
09_Exceptions
Python Projects_2014
CMD_Python_Set-Path....
Python_Clear-Window-...
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Runni...
Python_Tutorials.md
40 # Sequence objects can use several operators and functions:
41 sub_text = 'double'
42 print('sub_text =', sub_text)
43 print('sub_text + sub_text =', sub_text + sub_text)
44 print("'_' * 20 will multipy any character by any number =", '_' * 20)
45 print(nl, 'Next')
46 print('len(sub_text) returns string length =', len(sub_text)) # Length function telling how long a string is
47 print('min(sub_text) returns lowest unicode table value =', min(sub_text)) # Min function returns lowest unicode table value
48 print('max(sub_text) returns highest unicode table value =', max(sub_text)) # Max function returns highest unicode table value
49 print(nl, 'Next')
50 # Boolean resulting "in" or "not in" functions testing if a string is within, or not, another string
51 print('Boolean resulting "in" or "not in" functions testing if a string is within, or not, another string')
52 print('sub_not in quote_1 =', sub_text not in quote_1)
53 print('sub_text in quote_2) =', sub_text in quote_2)
54 print(nl, 'Next')
55 # String objects have many available methods like:
56 print('why_1.count("e") returns number of "e"s within string =', why_1.count('e')) # count number of 'e's within string
57 print('why_1.index("e") returns character position =', why_1.index('e')) # check the index or position within a string
58 print('why_1.index("e", 3, 18) returns character position within a positional range =', why_1.index('e', 3, 18)) # search at a particular start/stop position
59 print('Caution: Index method returns an error if character is not found.\n\tFind method is safer returning a "-1" if not found')
60 print('why_1.find("X") finds a string within a string OR returns -1 =', why_1.find('X')) # Find a string within a string OR returns -1
61 print(nl, 'Next')
62 print('why_1.startswith("She") returns boolean determining if string starts with indicator =', why_1.startswith('She')) # Boolean test determining if string starts with
63 print('why_1.endswith("!\\"") returns boolean determining if string ends with indicator =', why_1.endswith("!\"")) # Boolean test determining if string ends with
64 print(nl, 'Next')
65 print('why_1.upper() returns all upper case =', why_1.upper()) # changes case to upper
66 print('why_1.lower() returns all lower case =', why_1.lower()) # changes case to lower
67 print(nl, 'Next')
68 csv = 'a,b,c' # new string created for these examples


EXPLORER
{} User Settings
percipio08_Strings.py
OPEN EDITORS 1 UNSAVED
{} User Settings C:\Us...
percipio08_Strings....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Cour...
01_Start
02_Data-Sequence T...
percipio04_int_typ...
percipio05_float_ty...
percipio06_math_f...
percipio07_boolea...
percipio08_Strings....
percipio09_float_ty...
percipio10_bytes_t...
percipio11_bytearr...
percipio12_list_typ...
percipio13_tuple_t...
percipio14_slice_ty...
percipio14a_list_co...
03_Collections-Mapp...
04_Modules-Functio...
05_Classes
06_Working-with-Files
07_Comprehensions
percipio45_list_co...
percipio46_nested...
percipio47_zip_fun...
percipio48_set_co...
percipio49_diction...
08_Iterables-and-Ge...
09_Exceptions
Python Projects_2014
CMD_Python_Set-Path....
Python_Clear-Window-...
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Runni...
68 csv = 'a,b,c' # new string created for these examples
69 print('csv string =', csv)
70 print('csv.split(",") returns multiple strings split at indicated character =', csv.split(",")) # returns sub-strings split at indicated character, not including that character; csv-string split-method on a comma
71 print('",".join(["a", "b", "c"]) returns strings joined together divided by indicated character =', ",".join(["a", "b", "c"])) # joins strings together using indicated character ","
72 print(nl, 'Next')
73 # Python has many inspection functions such as 'isthis' and 'isthat'
74 print('sub_text.isalpha() returns boolean if string is all alphabetic characters =', sub_text.isalpha()) # Boolean returning True if string is all alphabetic characters
75 print('sub_text.isdigit() returns boolean if string is all numeric characters =', sub_text.isdigit()) # Boolean returning True if string is all numeric characters (0 to 9)
76 '''
77 RESULTS:
78 quote_1, quote_2 =
79 single quoted or double quoted
80 resulting in the same output
81
82
83 why_1, why_2 =
84 She said, "Hello!" It's mine!
85 why_not_1, why_not_2 =
86 she said, "Hello!" It's mine!
87 Above shows 2 different ways to result with the same string
88
89 Next
90 Create new lines by typing \n =
91 line1
92 line2
93 line3
94
95 Next
96 Create tabbed spacing by typing \t = col1 col2 col3
97
98 Next
99 Print 1 backslash by typing 2 backslashes: \
100
101 Next


EXPLORER
User Settings
percipio08_Strings.py
OPEN EDITORS 1 UNSAVED
User Settings C:\Us...
percipio08_Strings....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Cour...
01_Start
02_Data-Sequence T...
percipio04_int_typ...
percipio05_float_ty...
percipio06_math_f...
percipio07_boolea...
percipio08_Strings....
percipio09_float_ty...
percipio10_bytes_t...
percipio11_bytearr...
percipio12_list_typ...
percipio13_tuple_t...
percipio14_slice_ty...
percipio14a_list_co...
03_Collections-Mapp...
04_Modules-Functio...
05_Classes
06_Working-with-Files
07_Comprehensions
percipio45_list_co...
101 Next
102 Raw string printed exactly as typed by adding "r" in front of string = line1\nline2\nline3\n
103 Raw string tabbed = col1\tcol2\tcol3\t
104 Raw string with backslashes = the backslash: \\
105
106 Next
107 sub_text = double
108 sub_text + sub_text = doubledouble
109 '_' * 20 will multipy any character by any number = ____________________
110
111 Next
112 len(sub_text) returns string length = 6
113 min(sub_text) returns lowest unicode table value = b
114 max(sub_text) returns highest unicode table value = u
115
116 Next
117 Boolean resulting "in" or "not in" functions testing if a string is within, or not, another string
118 sub_not in quote_1 = True
119 sub_text in quote_2) = True
120
121 Next
122 why_1.count("e") returns number of "e"s within string = 2
123 why_1.index("e") returns character position = 2
124 why_1.index("e", 3, 18) returns character position within a positional range = 12
125 Caution: Index method returns an error if character is not found.
126 Find method is safer returning a "-1" if not found
127 why_1.find("X") finds a string within a string OR returns -1 = -1


EXPLORER
User Settings
percipio08_Strings.py
OPEN EDITORS 1 UNSAVED
User Settings C:\Us...
percipio08_Strings....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Cour...
01_Start
02_Data-Sequence T...
percipio04_int_typ...
percipio05_float_ty...
percipio06_math_f...
percipio07_boolea...
percipio08_Strings....
percipio09_float_ty...
percipio10_bytes_t...
percipio11_bytearr...
percipio12_list_typ...
percipio13_tuple_t...
why_1.find("X") finds a string within a string OR returns -1 = -1
Next
why_1.startswith("She") returns boolean determining if string starts with indicator = True
why_1.endswith("!\"") returns boolean determining if string ends with indicator = True
Next
why_1.upper() returns all upper case = SHE SAID, "HELLO!"
why_1.lower() returns all lower case = she said, "hello!"
Next
csv string = a,b,c
csv.split(",") returns multiple strings split at indicated character = ['a', 'b', 'c']
",".join(["a", "b", "c"]) returns strings joined together divided by indicated character = a,b,c
Next
sub_text.isalpha() returns boolean if string is all alphabetic characters = True
sub_text.isdigit() returns boolean if string is all numeric characters = False
'''