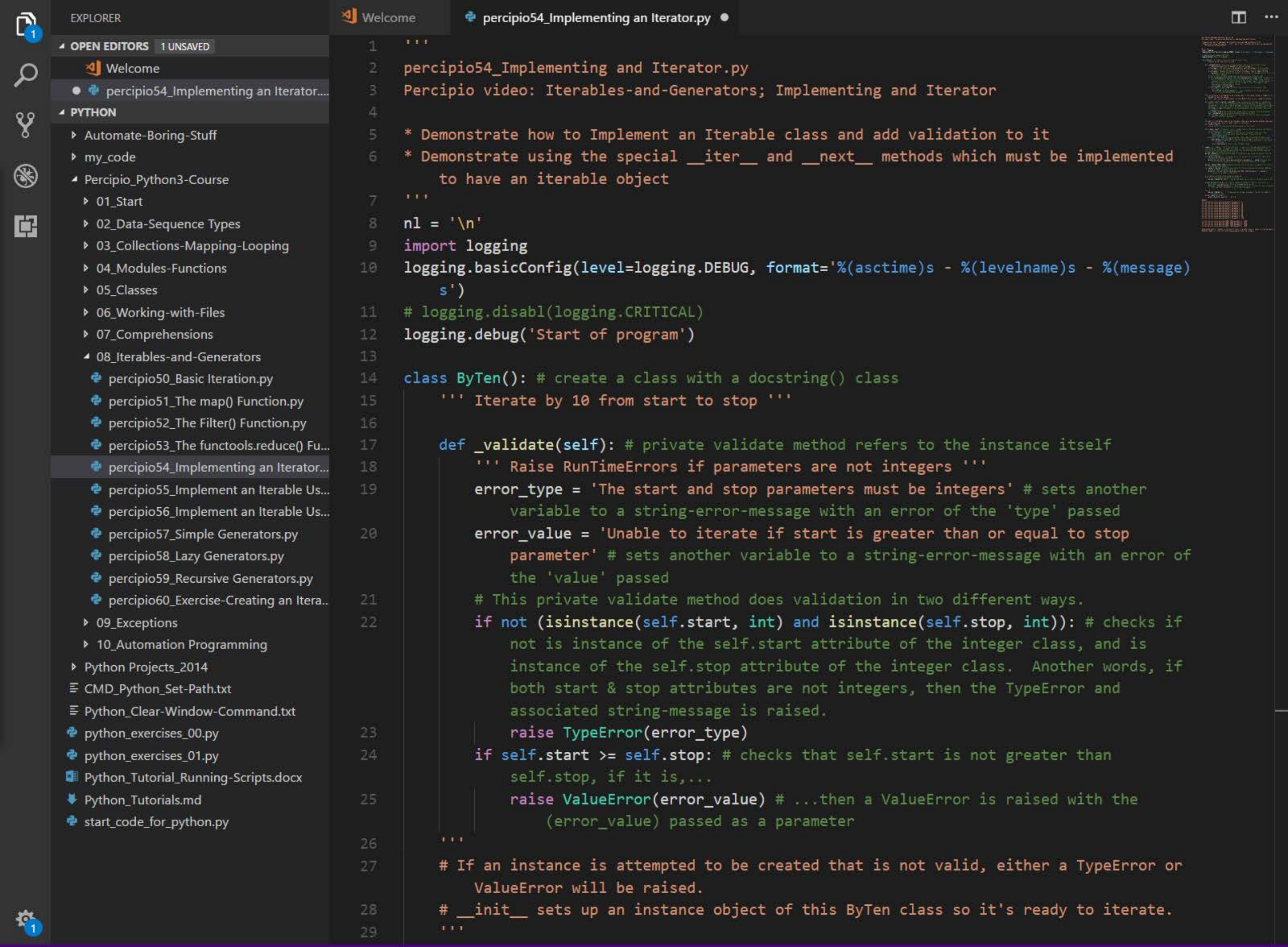
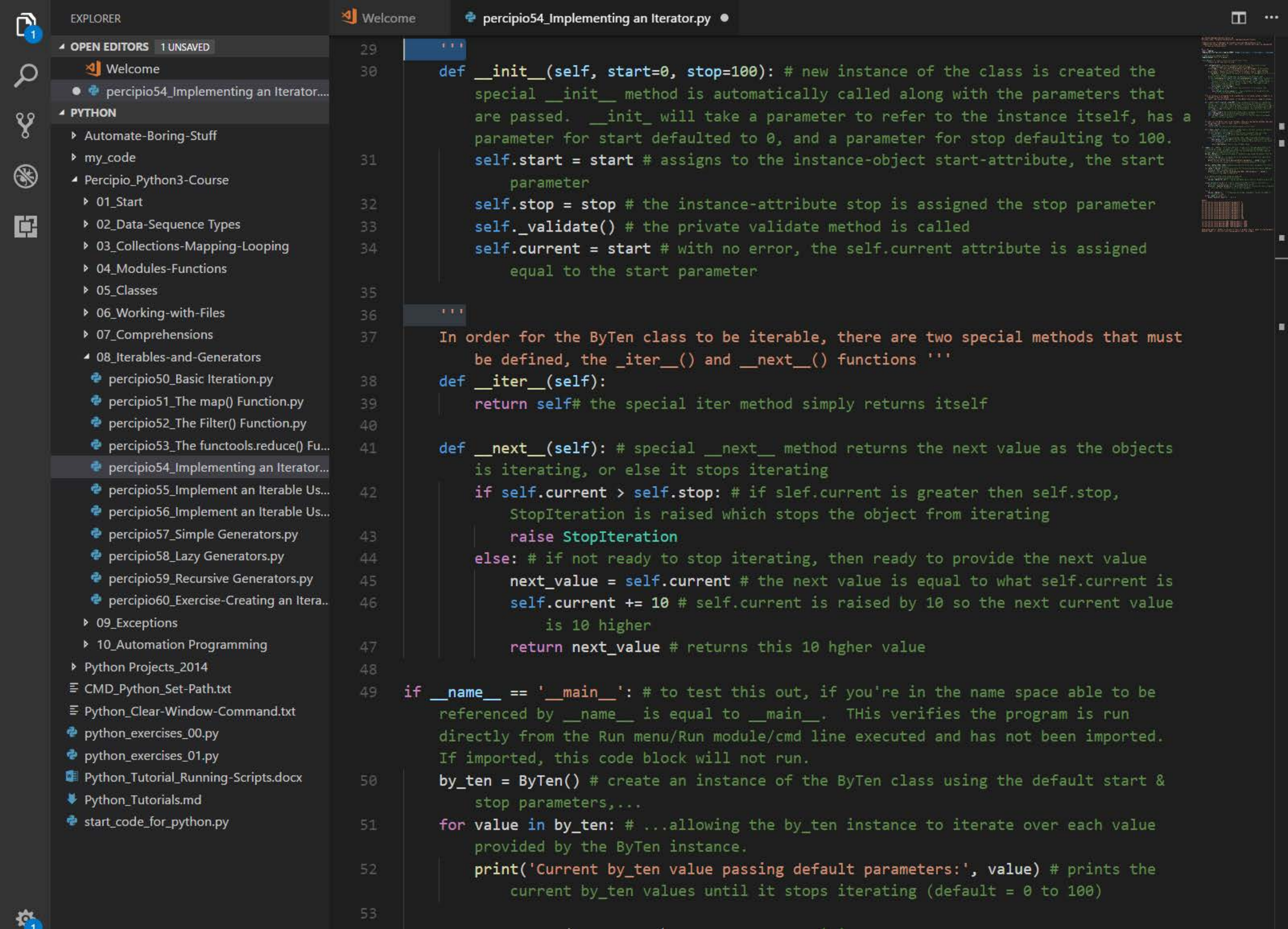
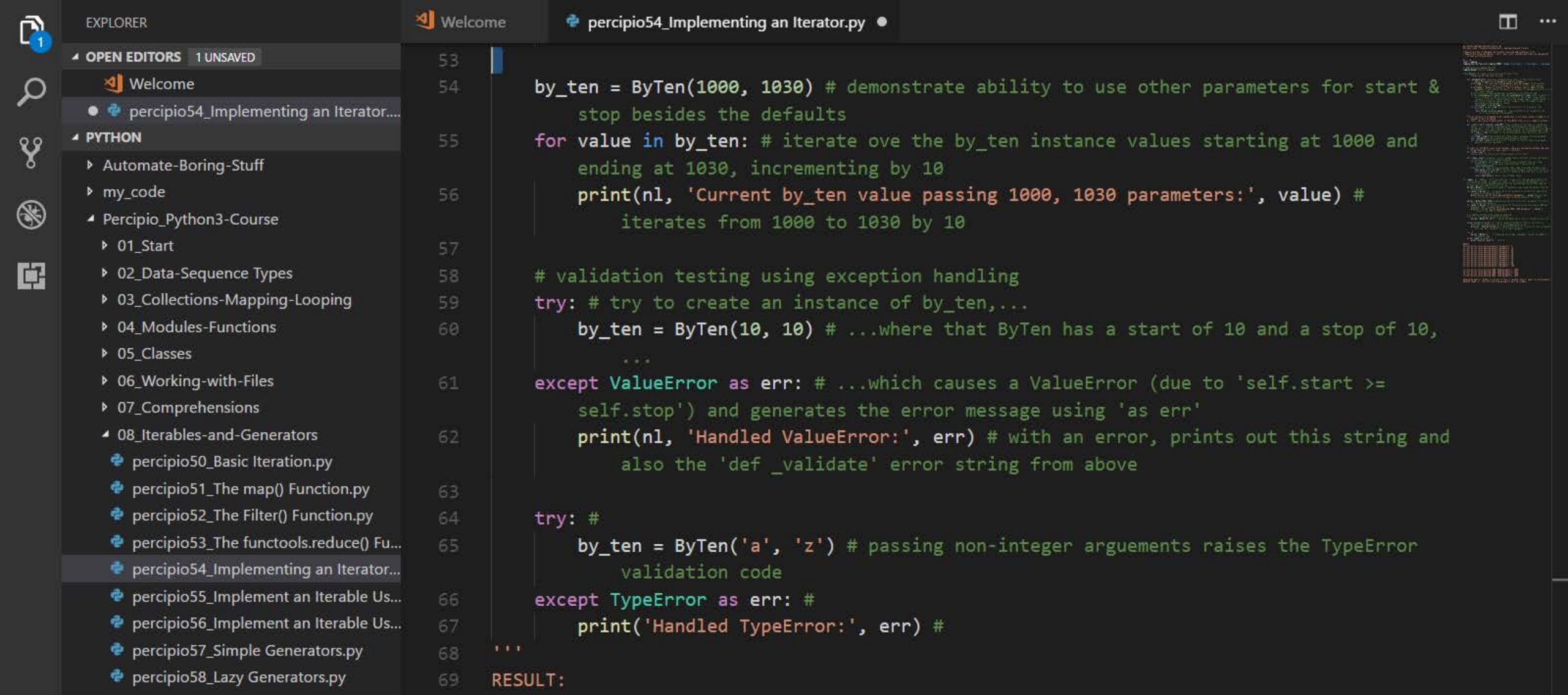
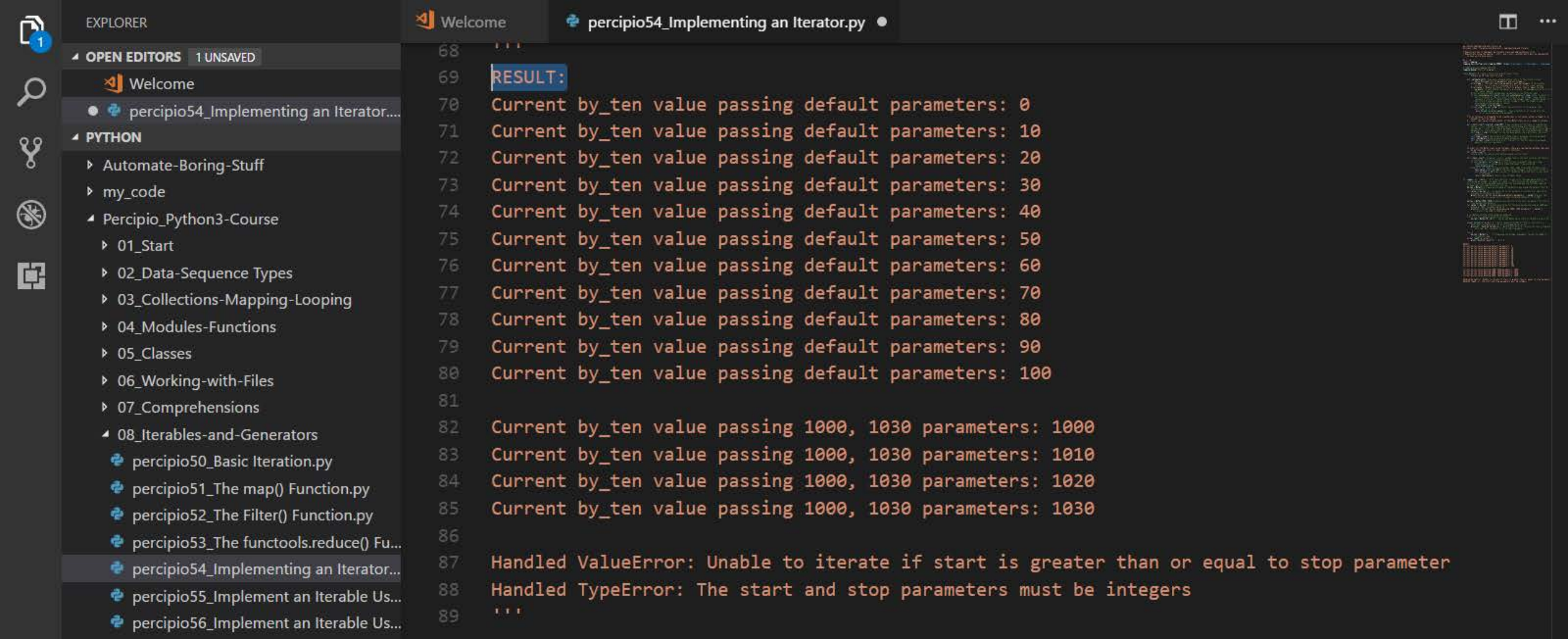

EXPLORER
OPEN EDITORS 1 UNSAVED
Welcome
percipio54_Implementing an Iterator....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function.py
percipio52_The Filter() Function.py
percipio53_The functools.reduce() Fu...
percipio54_Implementing an Iterator...
percipio55_Implement an Iterable Us...
percipio56_Implement an Iterable Us...
percipio57_Simple Generators.py
percipio58_Lazy Generators.py
percipio59_Recursive Generators.py
percipio60_Exercise-Creating an Itera..
09_Exceptions
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
Python_Clear-Window-Command.txt
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
start_code_for_python.py
Welcome
percipio54_Implementing an Iterator.py
''' percipio54_Implementing and Iterator.py Percipio video: Iterables-and-Generators; Implementing and Iterator * Demonstrate how to Implement an Iterable class and add validation to it * Demonstrate using the special __iter__ and __next__ methods which must be implemented to have an iterable object '''
nl = '\n'
import logging
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s - %(message)s')
# logging.disabl(logging.CRITICAL)
logging.debug('Start of program')
class ByTen(): # create a class with a docstring() class
''' Iterate by 10 from start to stop '''
def _validate(self): # private validate method refers to the instance itself
''' Raise RunTimeErrors if parameters are not integers '''
error_type = 'The start and stop parameters must be integers' # sets another variable to a string-error-message with an error of the 'type' passed
error_value = 'Unable to iterate if start is greater than or equal to stop parameter' # sets another variable to a string-error-message with an error of the 'value' passed
# This private validate method does validation in two different ways.
if not (isinstance(self.start, int) and isinstance(self.stop, int)): # checks if not is instance of the self.start attribute of the integer class, and is instance of the self.stop attribute of the integer class. Another words, if both start & stop attributes are not integers, then the TypeError and associated string-message is raised.
raise TypeError(error_type)
if self.start >= self.stop: # checks that self.start is not greater than self.stop, if it is,...
raise ValueError(error_value) # ...then a ValueError is raised with the (error_value) passed as a parameter
'''
If an instance is attempted to be created that is not valid, either a TypeError or ValueError will be raised.
__init__ sets up an instance object of this ByTen class so it's ready to iterate.
'''


EXPLORER
OPEN EDITORS 1 UNSAVED
Welcome
percipio54_Implementing an Iterator....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function.py
percipio52_The Filter() Function.py
percipio53_The functools.reduce() Fu...
percipio54_Implementing an Iterator...
percipio55_Implement an Iterable Us...
percipio56_Implement an Iterable Us...
percipio57_Simple Generators.py
percipio58_Lazy Generators.py
percipio59_Recursive Generators.py
percipio60_Exercise-Creating an Itera..
09_Exceptions
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
Python_Clear-Window-Command.txt
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
start_code_for_python.py
Welcome
percipio54_Implementing an Iterator.py
29 '''
30 def __init__(self, start=0, stop=100): # new instance of the class is created the special __init__ method is automatically called along with the parameters that are passed. __init_ will take a parameter to refer to the instance itself, has a parameter for start defaulted to 0, and a parameter for stop defaulting to 100.
31 self.start = start # assigns to the instance-object start-attribute, the start parameter
32 self.stop = stop # the instance-attribute stop is assigned the stop parameter
33 self._validate() # the private validate method is called
34 self.current = start # with no error, the self.current attribute is assigned equal to the start parameter
35
36 '''
37 In order for the ByTen class to be iterable, there are two special methods that must be defined, the _iter__() and __next__() functions '''
38 def __iter__(self):
39 return self# the special iter method simply returns itself
40
41 def __next__(self): # special __next__ method returns the next value as the objects is iterating, or else it stops iterating
42 if self.current > self.stop: # if slef.current is greater then self.stop, StopIteration is raised which stops the object from iterating
43 raise StopIteration
44 else: # if not ready to stop iterating, then ready to provide the next value
45 next_value = self.current # the next value is equal to what self.current is
46 self.current += 10 # self.current is raised by 10 so the next current value is 10 higher
47 return next_value # returns this 10 hgher value
48
49 if __name__ == '__main__': # to test this out, if you're in the name space able to be referenced by __name__ is equal to __main__. THis verifies the program is run directly from the Run menu/Run module/cmd line executed and has not been imported. If imported, this code block will not run.
50 by_ten = ByTen() # create an instance of the ByTen class using the default start & stop parameters,...
51 for value in by_ten: # ...allowing the by_ten instance to iterate over each value provided by the ByTen instance.
52 print('Current by_ten value passing default parameters:', value) # prints the current by_ten values until it stops iterating (default = 0 to 100)
53


EXPLORER
Welcome
percipio54_Implementing an Iterator.py
OPEN EDITORS 1 UNSAVED
Welcome
percipio54_Implementing an Iterator....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function.py
percipio52_The Filter() Function.py
percipio53_The functools.reduce() Fu...
percipio54_Implementing an Iterator...
percipio55_Implement an Iterable Us...
percipio56_Implement an Iterable Us...
percipio57_Simple Generators.py
percipio58_Lazy Generators.py
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
by_ten = ByTen(1000, 1030) # demonstrate ability to use other parameters for start & stop besides the defaults
for value in by_ten: # iterate ove the by_ten instance values starting at 1000 and ending at 1030, incrementing by 10
print(nl, 'Current by_ten value passing 1000, 1030 parameters:', value) # iterates from 1000 to 1030 by 10
validation testing using exception handling
try: # try to create an instance of by_ten,...
by_ten = ByTen(10, 10) # ...where that ByTen has a start of 10 and a stop of 10, ...
except ValueError as err: # ...which causes a ValueError (due to 'self.start >= self.stop') and generates the error message using 'as err'
print(nl, 'Handled ValueError:', err) # with an error, prints out this string and also the 'def _validate' error string from above
try: #
by_ten = ByTen('a', 'z') # passing non-integer arguements raises the TypeError validation code
except TypeError as err: #
print('Handled TypeError:', err) #
'''
RESULT:


EXPLORER
Welcome
percipio54_Implementing an Iterator.py
OPEN EDITORS 1 UNSAVED
Welcome
percipio54_Implementing an Iterator....
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function.py
percipio52_The Filter() Function.py
percipio53_The functools.reduce() Fu...
percipio54_Implementing an Iterator...
percipio55_Implement an Iterable Us...
percipio56_Implement an Iterable Us...
'''
RESULT:
Current by_ten value passing default parameters: 0
Current by_ten value passing default parameters: 10
Current by_ten value passing default parameters: 20
Current by_ten value passing default parameters: 30
Current by_ten value passing default parameters: 40
Current by_ten value passing default parameters: 50
Current by_ten value passing default parameters: 60
Current by_ten value passing default parameters: 70
Current by_ten value passing default parameters: 80
Current by_ten value passing default parameters: 90
Current by_ten value passing default parameters: 100
Current by_ten value passing 1000, 1030 parameters: 1000
Current by_ten value passing 1000, 1030 parameters: 1010
Current by_ten value passing 1000, 1030 parameters: 1020
Current by_ten value passing 1000, 1030 parameters: 1030
Handled ValueError: Unable to iterate if start is greater than or equal to stop parameter
Handled TypeError: The start and stop parameters must be integers
'''