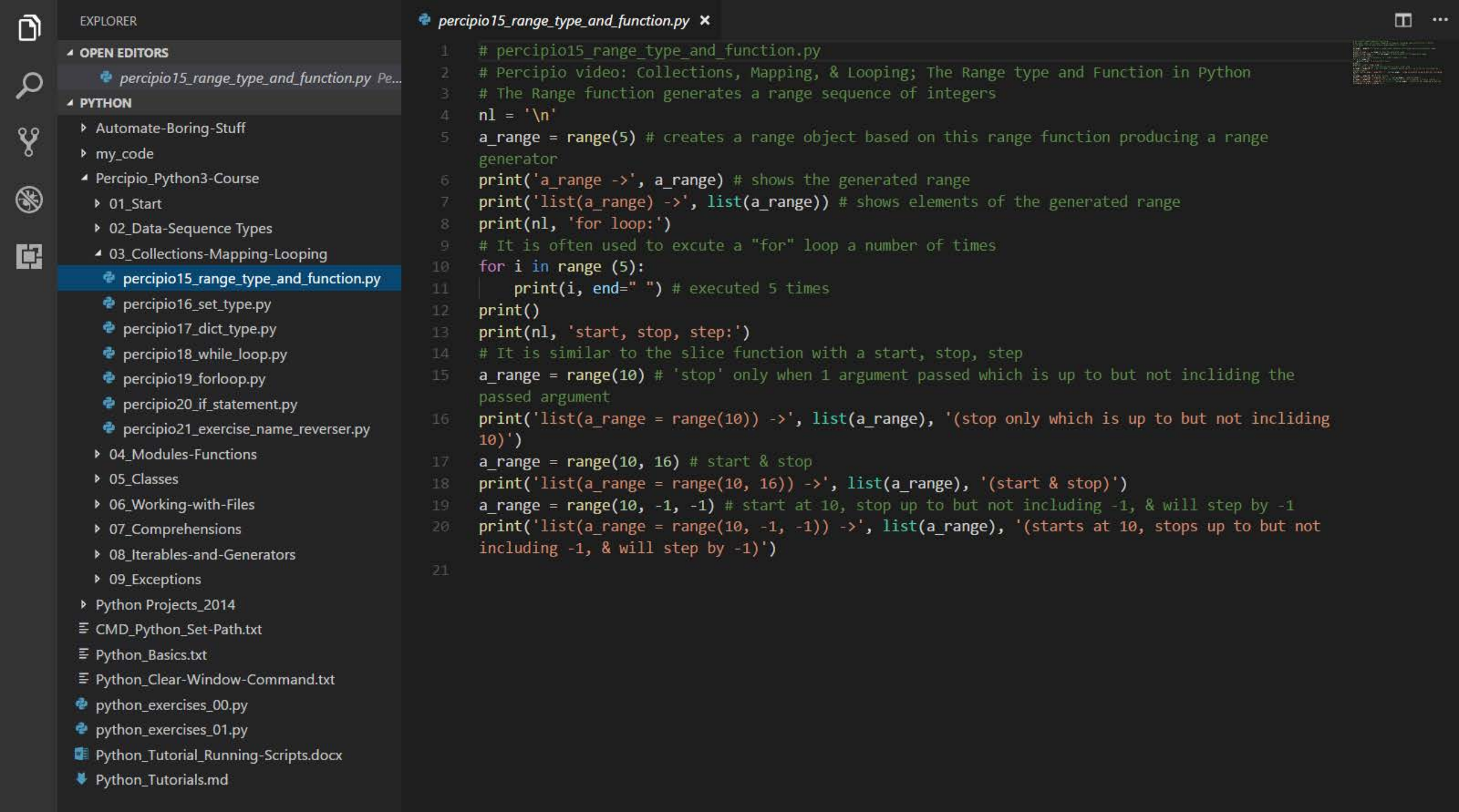
EXPLORER
OPEN EDITORS
percipio15_range_type_and_function.py Pe...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
percipio15_range_type_and_function.py
percipio16_set_type.py
percipio17_dict_type.py
percipio18_while_loop.py
percipio19_forloop.py
percipio20_if_statement.py
percipio21_exercise_name_reverser.py
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
percipio15_range_type_and_function.py
# percipio15_range_type_and_function.py
# Percipio video: Collections, Mapping, & Looping; The Range type and Function in Python
# The Range function generates a range sequence of integers
nl = '\n'
a_range = range(5) # creates a range object based on this range function producing a range generator
print('a_range ->', a_range) # shows the generated range
print('list(a_range) ->', list(a_range)) # shows elements of the generated range
print(nl, 'for loop:')
# It is often used to excute a "for" loop a number of times
for i in range (5):
print(i, end=" ") # executed 5 times
print()
print(nl, 'start, stop, step:')
# It is similar to the slice function with a start, stop, step
a_range = range(10) # 'stop' only when 1 argument passed which is up to but not incliding the passed argument
print('list(a_range = range(10)) ->', list(a_range), '(stop only which is up to but not incliding 10)')
a_range = range(10, 16) # start & stop
print('list(a_range = range(10, 16)) ->', list(a_range), '(start & stop)')
a_range = range(10, -1, -1) # start at 10, stop up to but not including -1, & will step by -1
print('list(a_range = range(10, -1, -1)) ->', list(a_range), '(starts at 10, stops up to but not including -1, & will step by -1)')