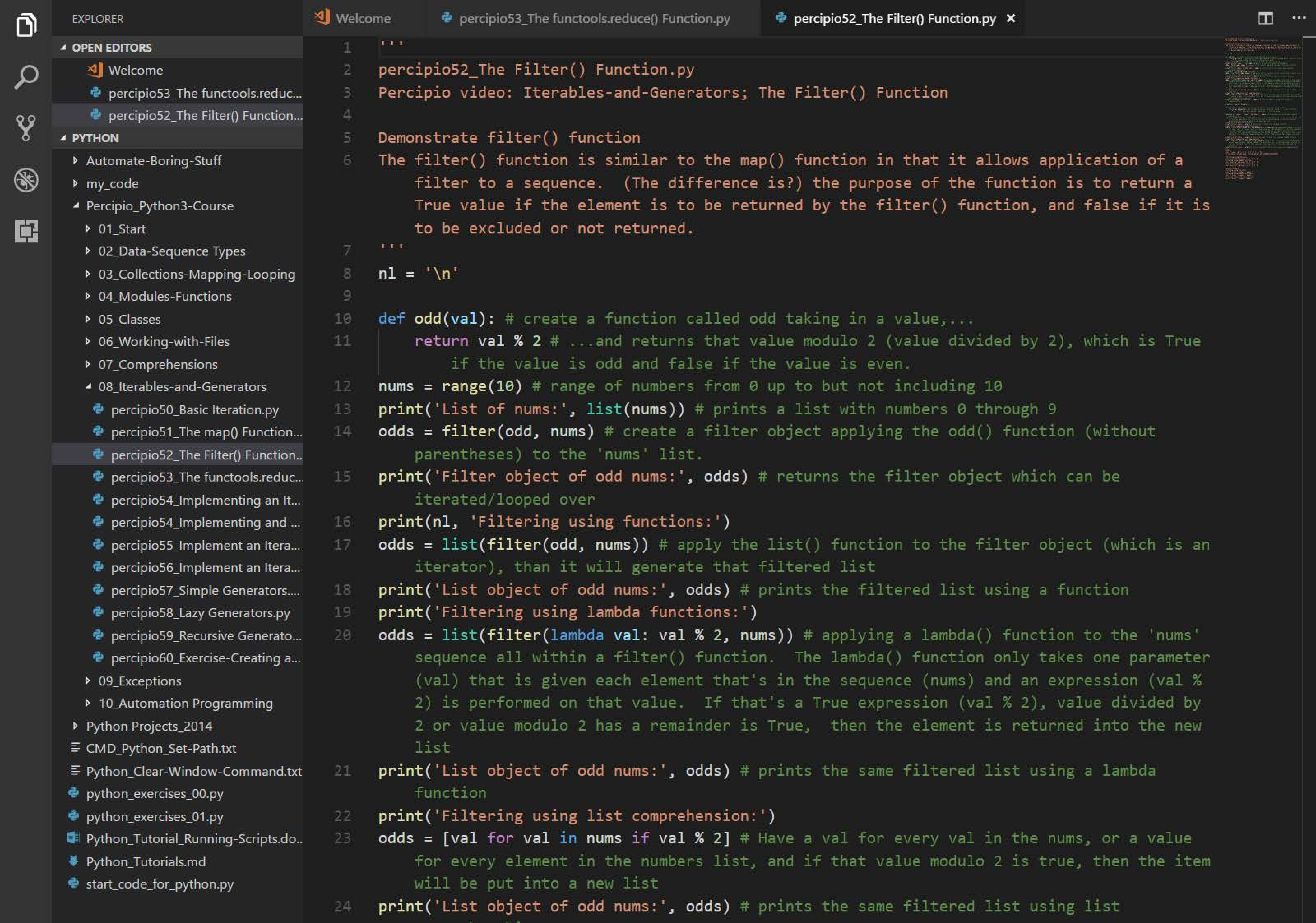
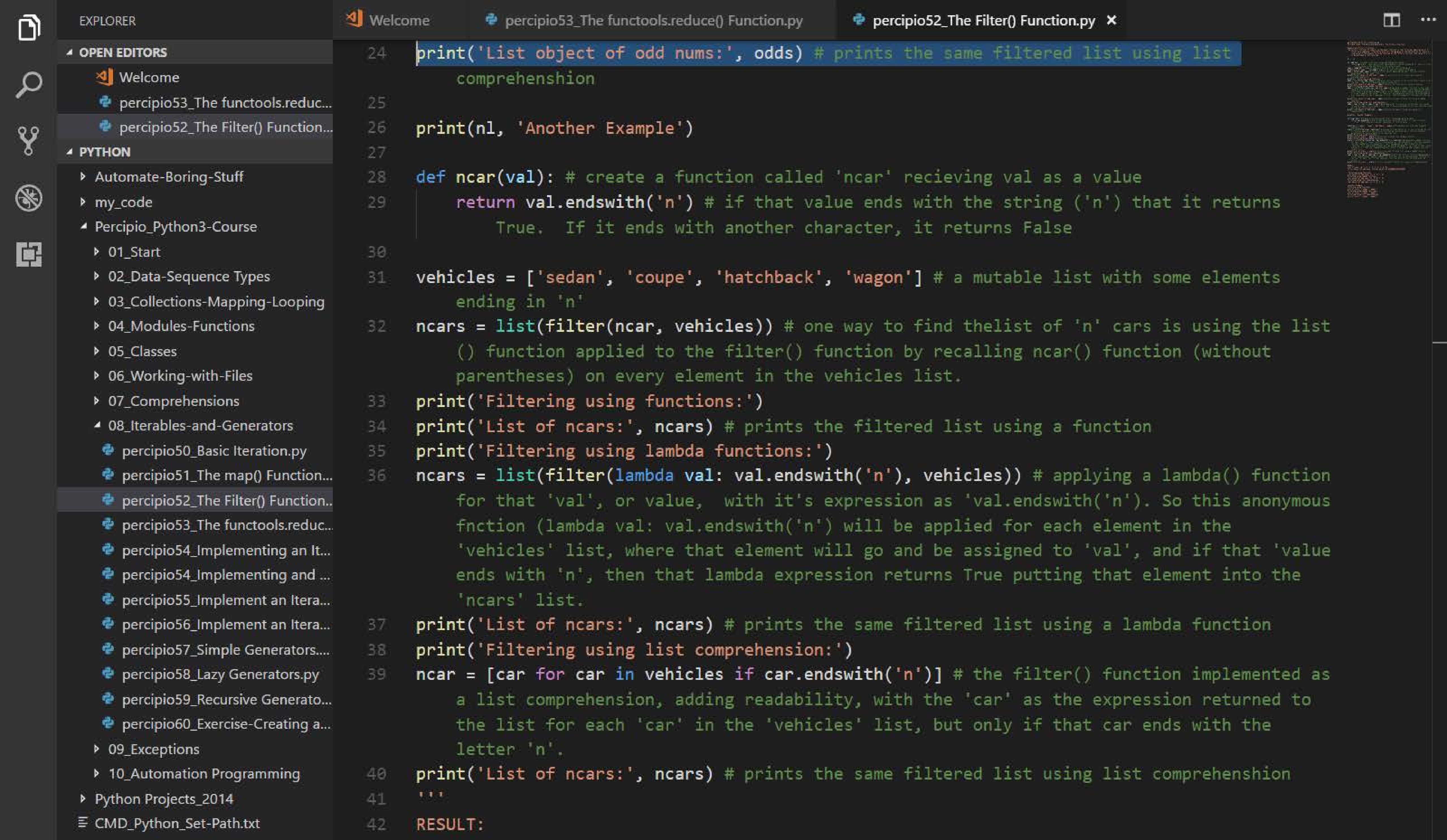

EXPLORER
Welcome
percipio53_The functools.reduce() Function.py
percipio52_The Filter() Function.py
OPEN EDITORS
Welcome
percipio53_The functools.reduc...
percipio52_The Filter() Function...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function...
percipio52_The Filter() Function...
percipio53_The functools.reduc...
percipio54_Implementing an It...
percipio54_Implementing and ...
percipio55_Implement an Itera...
percipio56_Implement an Itera...
percipio57_Simple Generators....
percipio58_Lazy Generators.py
percipio59_Recursive Generato...
percipio60_Exercise-Creating a...
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.do...
Python_Tutorials.md
start_code_for_python.py

```python
'''
percipio52_The Filter() Function.py
Percipio video: Iterables-and-Generators; The Filter() Function

Demonstrate filter() function
The filter() function is similar to the map() function in that it allows application of a filter to a sequence.  (The difference is?) the purpose of the function is to return a True value if the element is to be returned by the filter() function, and false if it is to be excluded or not returned.
'''
nl = '\n'

def odd(val): # create a function called odd taking in a value,...
    return val % 2 # ...and returns that value modulo 2 (value divided by 2), which is True if the value is odd and false if the value is even.
nums = range(10) # range of numbers from 0 up to but not including 10
print('List of nums:', list(nums)) # prints a list with numbers 0 through 9
odds = filter(odd, nums) # create a filter object applying the odd() function (without parentheses) to the 'nums' list.
print('Filter object of odd nums:', odds) # returns the filter object which can be iterated/looped over
print(nl, 'Filtering using functions:')
odds = list(filter(odd, nums)) # apply the list() function to the filter object (which is an iterator), than it will generate that filtered list
print('List object of odd nums:', odds) # prints the filtered list using a function
print('Filtering using lambda functions:')
odds = list(filter(lambda val: val % 2, nums)) # applying a lambda() function to the 'nums' sequence all within a filter() function.  The lambda() function only takes one parameter (val) that is given each element that's in the sequence (nums) and an expression (val % 2) is performed on that value.  If that's a True expression (val % 2), value divided by 2 or value modulo 2 has a remainder is True,  then the element is returned into the new list
print('List object of odd nums:', odds) # prints the same filtered list using a lambda function
print('Filtering using list comprehension:')
odds = [val for val in nums if val % 2] # Have a val for every val in the nums, or a value for every element in the numbers list, and if that value modulo 2 is true, then the item will be put into a new list
print('List object of odd nums:', odds) # prints the same filtered list using list
```

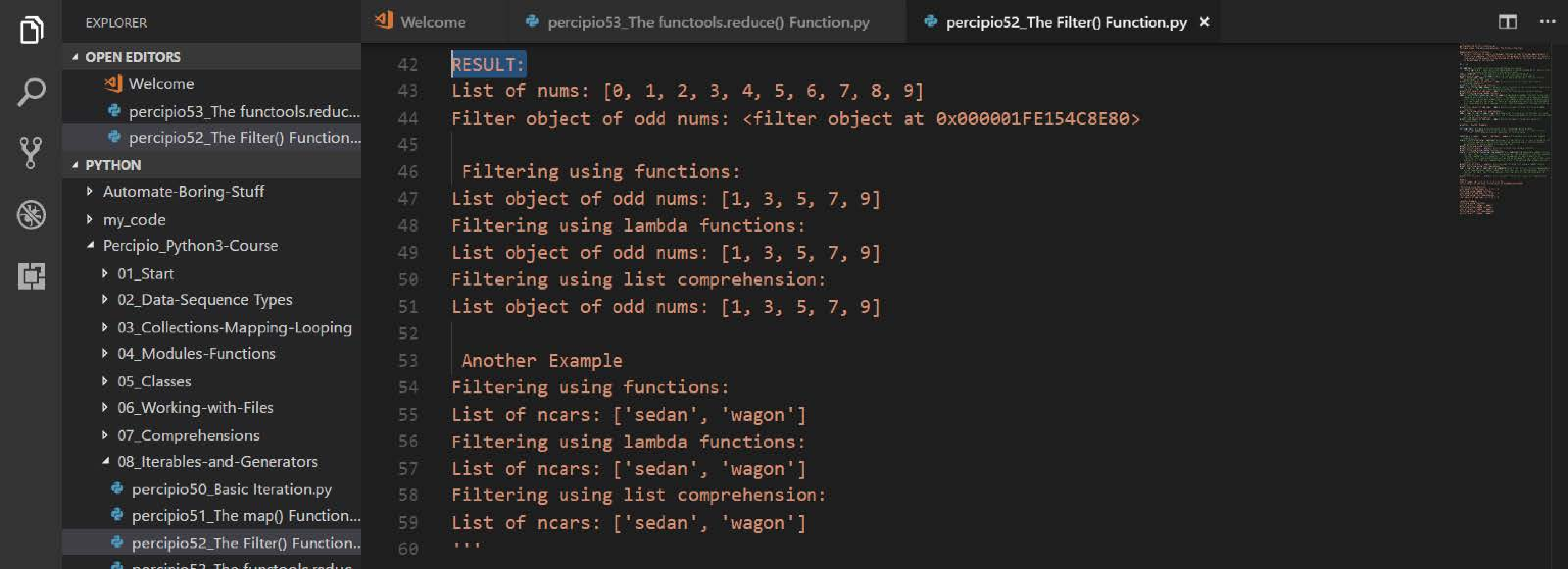

EXPLORER
OPEN EDITORS
Welcome
percipio53_The functools.reduc...
percipio52_The Filter() Function...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
percipio50_Basic Iteration.py
percipio51_The map() Function...
percipio52_The Filter() Function...
percipio53_The functools.reduc...
percipio54_Implementing an It...
percipio54_Implementing and ...
percipio55_Implement an Itera...
percipio56_Implement an Itera...
percipio57_Simple Generators....
percipio58_Lazy Generators.py
percipio59_Recursive Generato...
percipio60_Exercise-Creating a...
09_Exceptions
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
Welcome
percipio53_The functools.reduce() Function.py
percipio52_The Filter() Function.py

```python
print('List object of odd nums:', odds) # prints the same filtered list using list comprehenshion

print(nl, 'Another Example')

def ncar(val): # create a function called 'ncar' recieving val as a value
    return val.endswith('n') # if that value ends with the string ('n') that it returns True.  If it ends with another character, it returns False

vehicles = ['sedan', 'coupe', 'hatchback', 'wagon'] # a mutable list with some elements ending in 'n'
ncars = list(filter(ncar, vehicles)) # one way to find thelist of 'n' cars is using the list() function applied to the filter() function by recalling ncar() function (without parentheses) on every element in the vehicles list.
print('Filtering using functions:')
print('List of ncars:', ncars) # prints the filtered list using a function
print('Filtering using lambda functions:')
ncars = list(filter(lambda val: val.endswith('n'), vehicles)) # applying a lambda() function for that 'val', or value,  with it's expression as 'val.endswith('n'). So this anonymous fnction (lambda val: val.endswith('n') will be applied for each element in the 'vehicles' list, where that element will go and be assigned to 'val', and if that 'value ends with 'n', then that lambda expression returns True putting that element into the 'ncars' list.
print('List of ncars:', ncars) # prints the same filtered list using a lambda function
print('Filtering using list comprehension:')
ncar = [car for car in vehicles if car.endswith('n')] # the filter() function implemented as a list comprehension, adding readability, with the 'car' as the expression returned to the list for each 'car' in the 'vehicles' list, but only if that car ends with the letter 'n'.
print('List of ncars:', ncars) # prints the same filtered list using list comprehenshion
'''
RESULT:
```


EXPLORER
Welcome
percipio53_The functools.reduce() Function.py
percipio52_The Filter() Function.py
OPEN EDITORS
Welcome
percipio53_The functools.reduc...
percipio52_The Filter() Function...
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...
percipio52_The Filter() Function...

```
RESULT:
List of nums: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Filter object of odd nums: <filter object at 0x000001FE154C8E80>

 Filtering using functions:
List object of odd nums: [1, 3, 5, 7, 9]
Filtering using lambda functions:
List object of odd nums: [1, 3, 5, 7, 9]
Filtering using list comprehension:
List object of odd nums: [1, 3, 5, 7, 9]

 Another Example
Filtering using functions:
List of ncars: ['sedan', 'wagon']
Filtering using lambda functions:
List of ncars: ['sedan', 'wagon']
Filtering using list comprehension:
List of ncars: ['sedan', 'wagon']
'''
```