

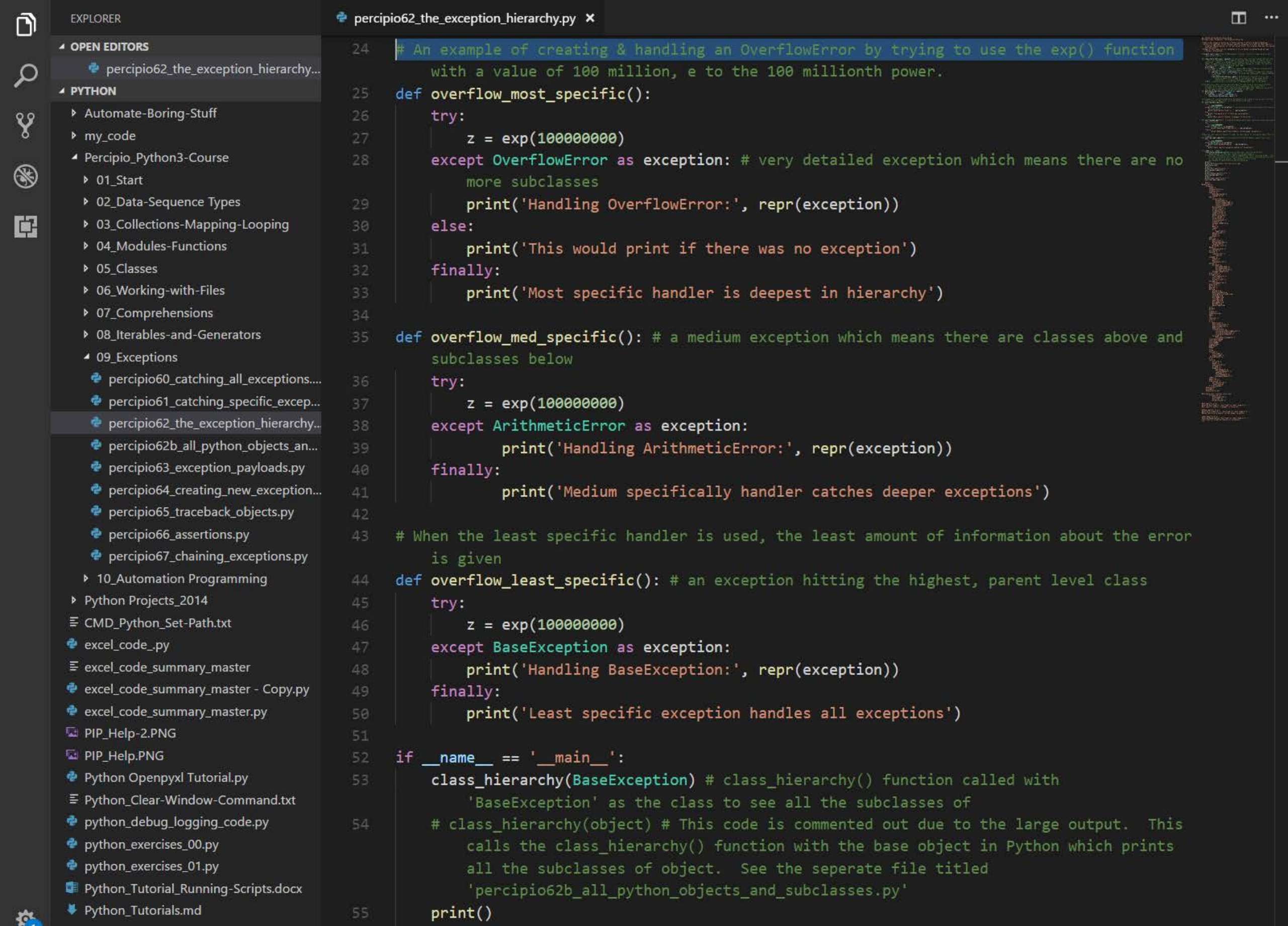


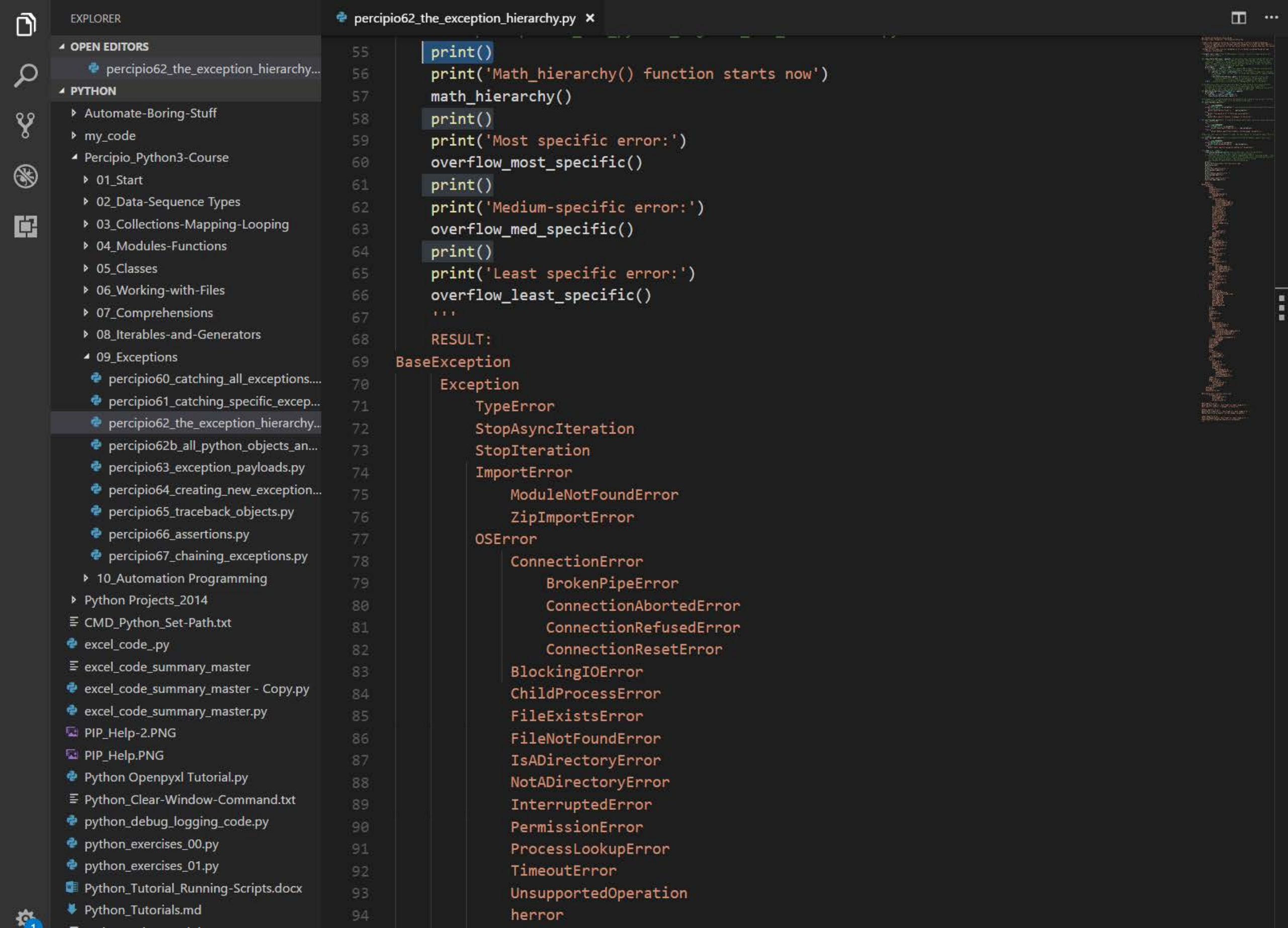
EXPLORER

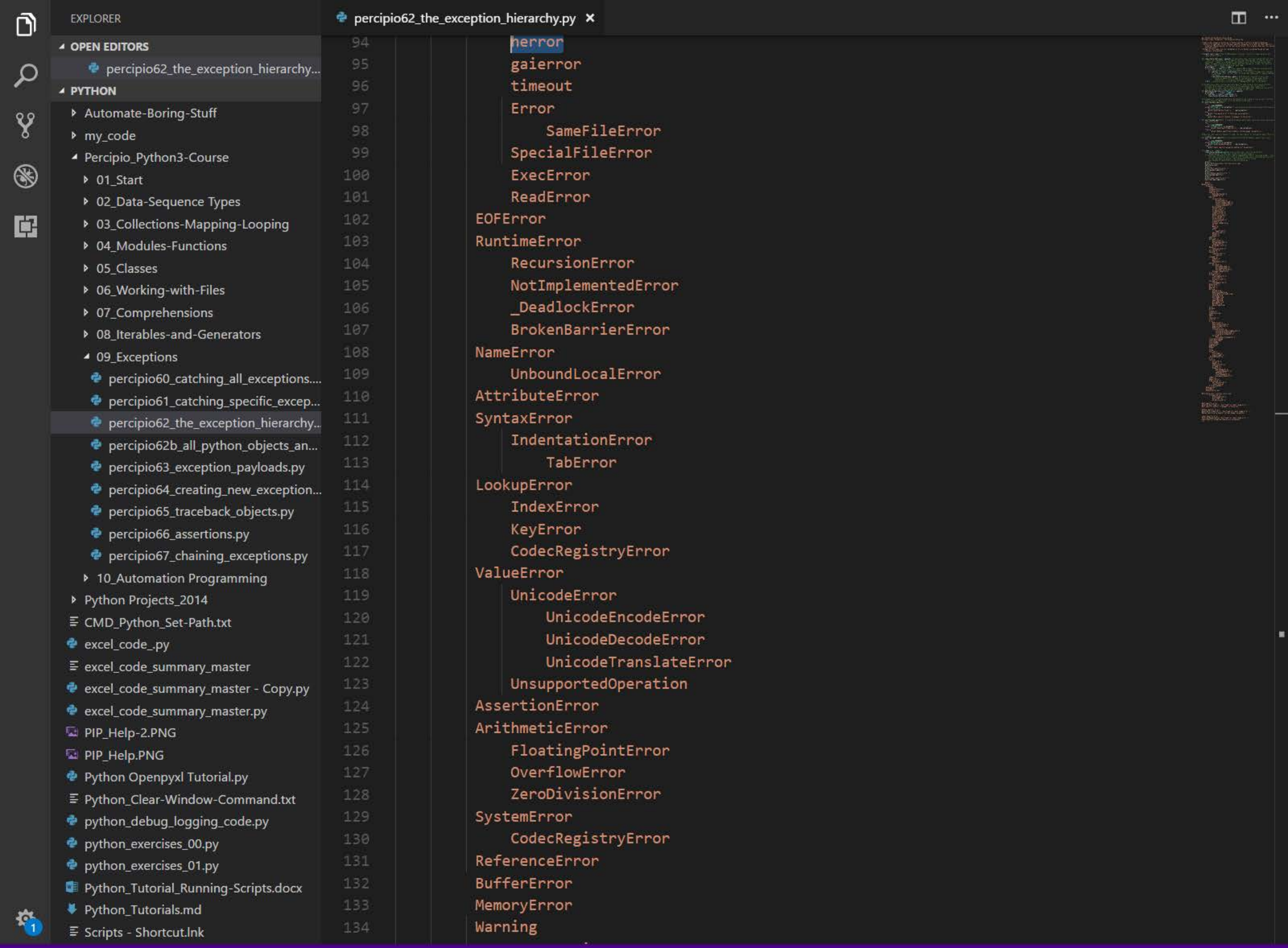
- OPEN EDITORS
  - percipio62_the_exception_hierarchy...
- PYTHON
  - Automate-Boring-Stuff
  - my_code
  - Percipio_Python3-Course
    - 01_Start
    - 02_Data-Sequence Types
    - 03_Collections-Mapping-Looping
    - 04_Modules-Functions
    - 05_Classes
    - 06_Working-with-Files
    - 07_Comprehensions
    - 08_Iterables-and-Generators
    - 09_Exceptions
      - percipio60_catching_all_exceptions....
      - percipio61_catching_specific_excep...
      - percipio62_the_exception_hierarchy...
      - percipio62b_all_python_objects_an...
      - percipio63_exception_payloads.py
      - percipio64_creating_new_exception...
      - percipio65_traceback_objects.py
      - percipio66_assertions.py
      - percipio67_chaining_exceptions.py
    - 10_Automation Programming
  - Python Projects_2014
  - CMD_Python_Set-Path.txt
  - excel_code_.py
  - excel_code_summary_master
  - excel_code_summary_master - Copy.py
  - excel_code_summary_master.py
  - PIP_Help-2.PNG
  - PIP_Help.PNG
  - Python Openpyxl Tutorial.py
  - Python_Clear-Window-Command.txt
  - python_debug_logging_code.py
  - python_exercises_00.py
  - python_exercises_01.py
  - Python_Tutorial_Running-Scripts.docx
  - Python_Tutorials.md

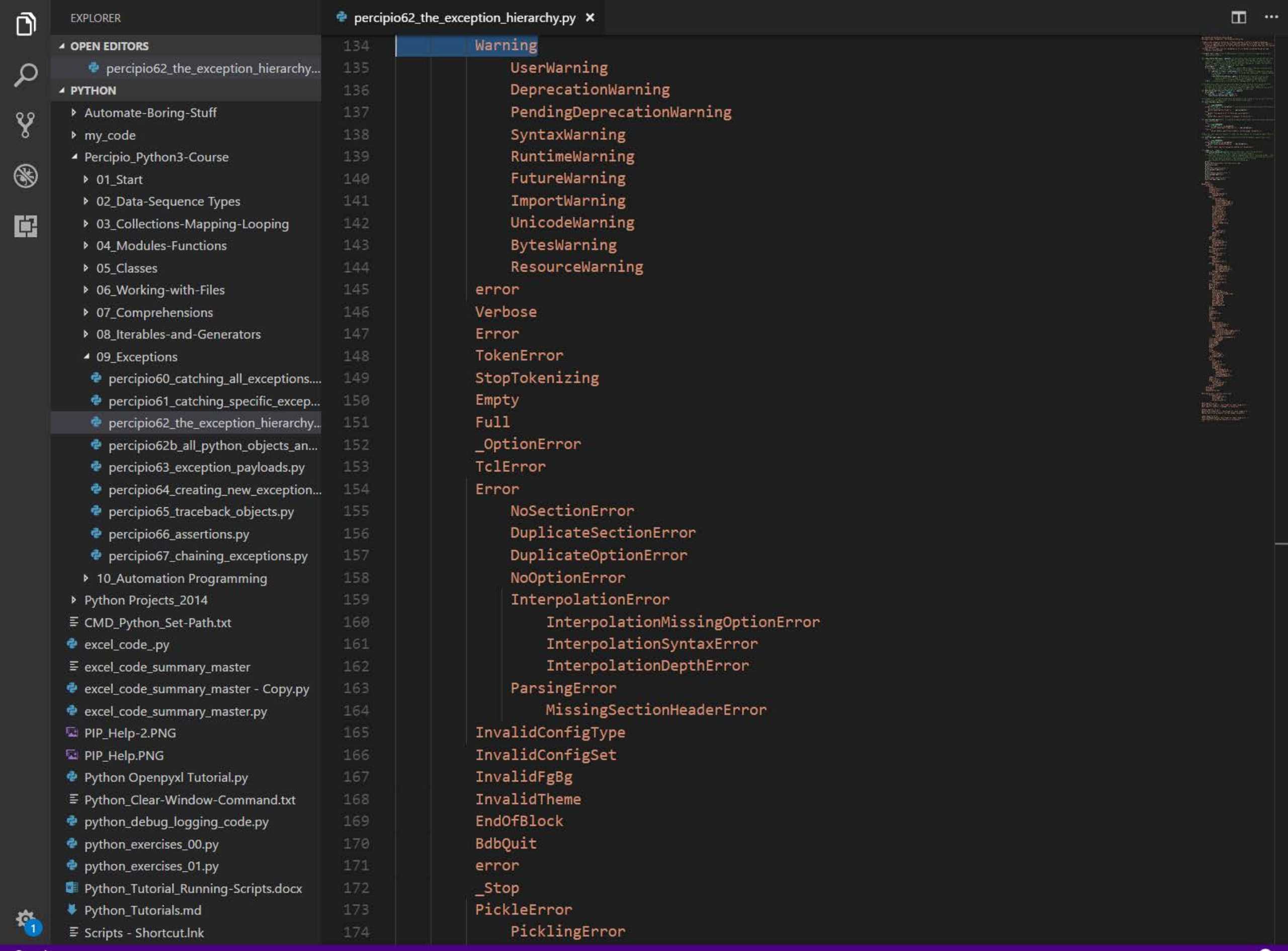
percipio62_the_exception_hierarchy.py ×

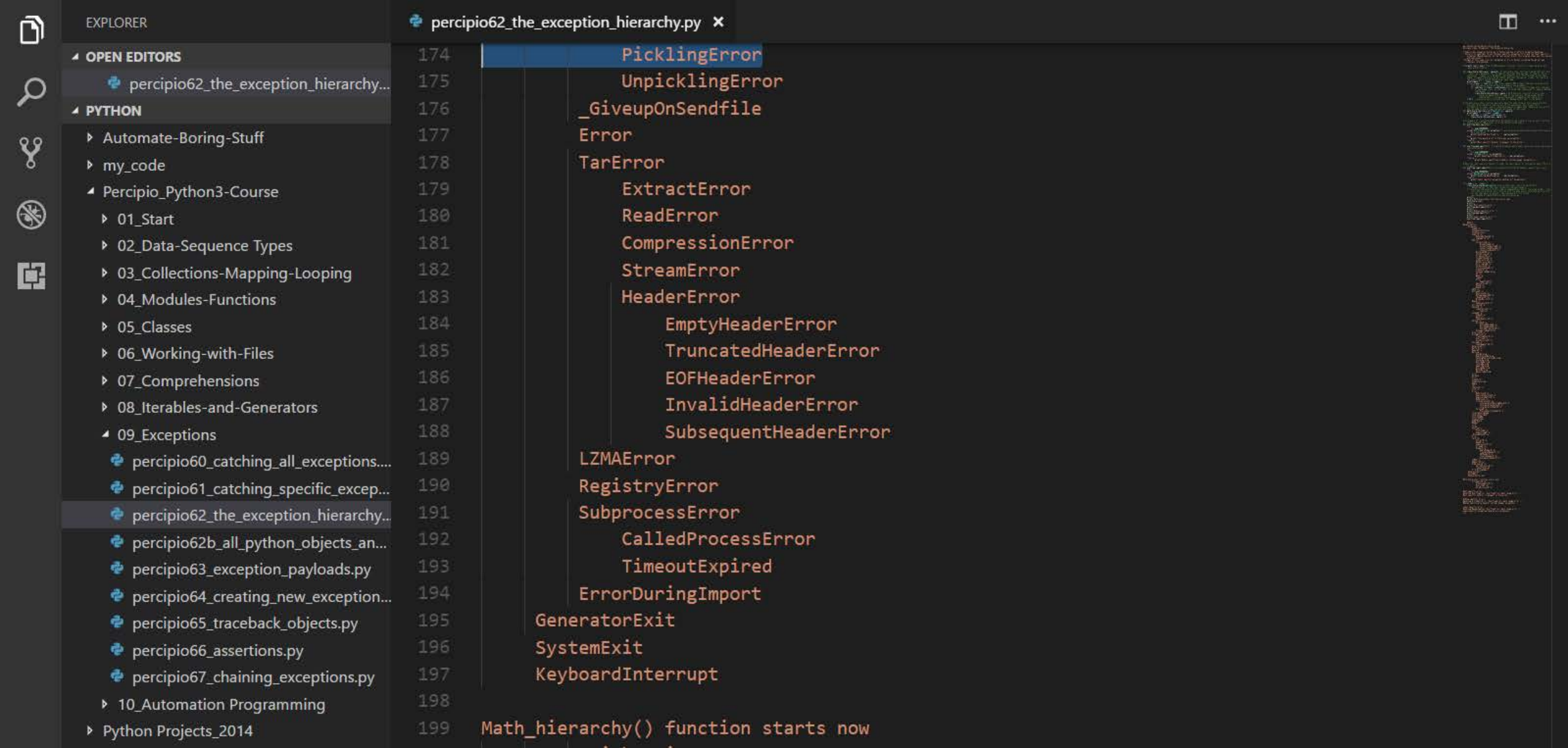
```
'''
percipio62_the_exception_hierarchy.py
Percipio video: Exceptions; The Exception Hierarchy

* Demonstrate exception hierarchy in Python and how it affects Exception Handling
* Since it is important to use the most specific exception first and the least specific
    exception (BaseException) as the last except block, this program shows you the classes
    and subclasses.
* Output of this code lists all exception-errors in Python including the parent and
    subclass relationship.
'''
from math import exp # from the Math module, the exp() function is imported which will
    cause a OverflowError.

def class_hierarchy(class_, space=0): # class_hierarchy() function created that will use
    'class_' as a parameter for the name of the class that you want to view it, ans all
    subclasses.  'space' is a parameter that defaults to 0. This function prints that
    number of space characters as an indent, and then the special __name__ attribute of
    the class_ object, which should be a class name.
    print(space * ' ', class_.__name__)
    if not class_ is type: # If that class_name is NOT a type (type has a problem using
        this subclasses method without an argument and avoided),...
        for subclass in class_.__subclasses__(): # ...it iterates through every subclass
            or descended class, of the class, by using that subclasses, or special method
            subclasses,...
            class_hierarchy(subclass, space + 4) # and uses recursion to call class
                hierarchy again with the name of the subclass. Using normal Python
                indentation, it increases the amount of space by four characters.
    # NOTE: class_hierarchy() function ends at 'KeyboardInterrupt' in the output

# The math_hierarchy() function basically does the same thing as the class_hierarchy()
    function, printing out a class in all of its subclasses, but it uses the
    ArithmeticError class, and is indented 8 spaces to begin with. Otherwise this prints
    out the name of every class and any subclasses of that class.
def math_hierarchy(class_=ArithmeticError, space=8):
    print(space * ' ', class_.__name__)
    for subclass in class_.__subclasses__():
        class_hierarchy(subclass, space + 4)

# An example of creating & handling an OverflowError by trying to use the exp() function
```

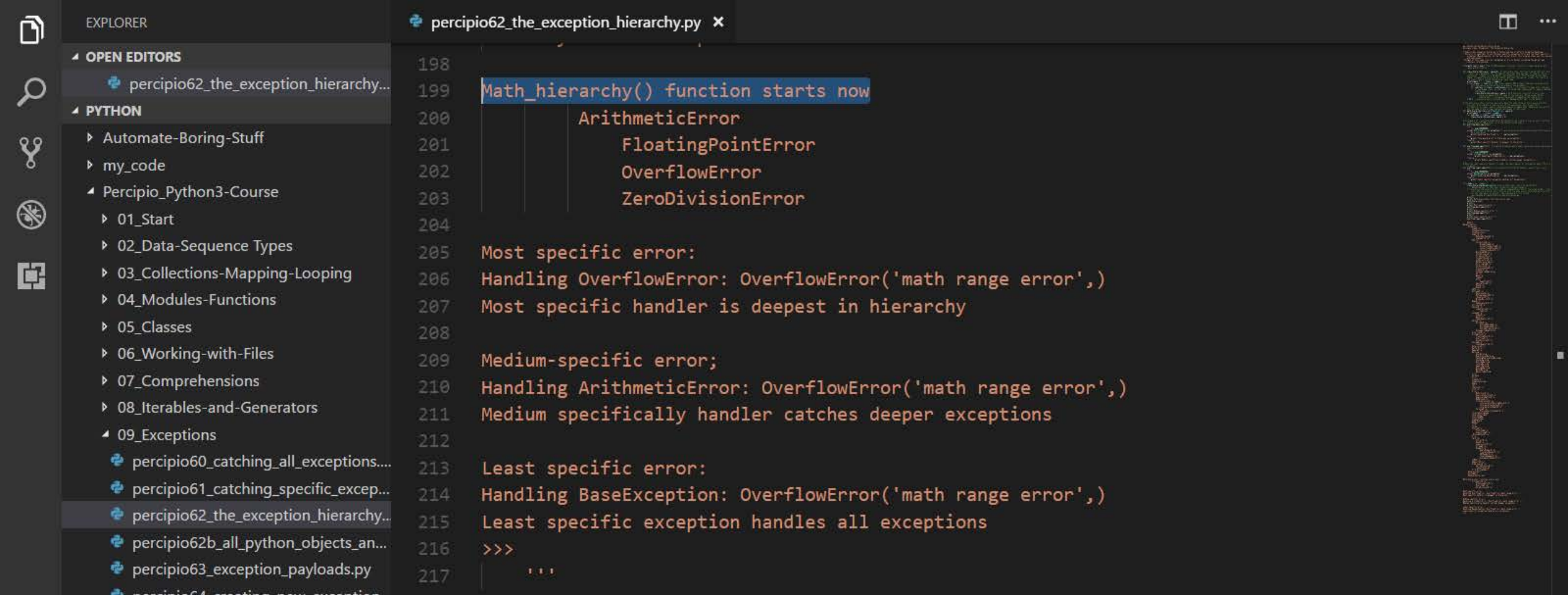

EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
PYTHON
Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-Looping
04_Modules-Functions
05_Classes
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
percipio64_creating_new_exception...
percipio65_traceback_objects.py
percipio66_assertions.py
percipio67_chaining_exceptions.py
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
excel_code_.py
excel_code_summary_master
excel_code_summary_master - Copy.py
excel_code_summary_master.py
PIP_Help-2.PNG
PIP_Help.PNG
Python Openpyxl Tutorial.py
Python_Clear-Window-Command.txt
python_debug_logging_code.py
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
24 # An example of creating & handling an OverflowError by trying to use the exp() function with a value of 100 million, e to the 100 millionth power.
25 def overflow_most_specific():
26 try:
27 z = exp(100000000)
28 except OverflowError as exception: # very detailed exception which means there are no more subclasses
29 print('Handling OverflowError:', repr(exception))
30 else:
31 print('This would print if there was no exception')
32 finally:
33 print('Most specific handler is deepest in hierarchy')
34
35 def overflow_med_specific(): # a medium exception which means there are classes above and subclasses below
36 try:
37 z = exp(100000000)
38 except ArithmeticError as exception:
39 print('Handling ArithmeticError:', repr(exception))
40 finally:
41 print('Medium specifically handler catches deeper exceptions')
42
43 # When the least specific handler is used, the least amount of information about the error is given
44 def overflow_least_specific(): # an exception hitting the highest, parent level class
45 try:
46 z = exp(100000000)
47 except BaseException as exception:
48 print('Handling BaseException:', repr(exception))
49 finally:
50 print('Least specific exception handles all exceptions')
51
52 if __name__ == '__main__':
53 class_hierarchy(BaseException) # class_hierarchy() function called with 'BaseException' as the class to see all the subclasses of
54 # class_hierarchy(object) # This code is commented out due to the large output. This calls the class_hierarchy() function with the base object in Python which prints all the subclasses of object. See the seperate file titled 'percipio62b_all_python_objects_and_subclasses.py'
55 print()


EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
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
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
percipio64_creating_new_exception...
percipio65_traceback_objects.py
percipio66_assertions.py
percipio67_chaining_exceptions.py
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
excel_code_.py
excel_code_summary_master
excel_code_summary_master - Copy.py
excel_code_summary_master.py
PIP_Help-2.PNG
PIP_Help.PNG
Python Openpyxl Tutorial.py
Python_Clear-Window-Command.txt
python_debug_logging_code.py
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
print()
print('Math_hierarchy() function starts now')
math_hierarchy()
print()
print('Most specific error:')
overflow_most_specific()
print()
print('Medium-specific error:')
overflow_med_specific()
print()
print('Least specific error:')
overflow_least_specific()
'''
RESULT:
BaseException
Exception
TypeError
StopAsyncIteration
StopIteration
ImportError
ModuleNotFoundError
ZipImportError
OSError
ConnectionError
BrokenPipeError
ConnectionAbortedError
ConnectionRefusedError
ConnectionResetError
BlockingIOError
ChildProcessError
FileExistsError
FileNotFoundError
IsADirectoryError
NotADirectoryError
InterruptedError
PermissionError
ProcessLookupError
TimeoutError
UnsupportedOperation
herror


EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
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
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
percipio64_creating_new_exception...
percipio65_traceback_objects.py
percipio66_assertions.py
percipio67_chaining_exceptions.py
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
excel_code_.py
excel_code_summary_master
excel_code_summary_master - Copy.py
excel_code_summary_master.py
PIP_Help-2.PNG
PIP_Help.PNG
Python Openpyxl Tutorial.py
Python_Clear-Window-Command.txt
python_debug_logging_code.py
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
Scripts - Shortcut.lnk
herror
gaierror
timeout
Error
SameFileError
SpecialFileError
ExecError
ReadError
EOFError
RuntimeError
RecursionError
NotImplementedError
_DeadlockError
BrokenBarrierError
NameError
UnboundLocalError
AttributeError
SyntaxError
IndentationError
TabError
LookupError
IndexError
KeyError
CodecRegistryError
ValueError
UnicodeError
UnicodeEncodeError
UnicodeDecodeError
UnicodeTranslateError
UnsupportedOperation
AssertionError
ArithmeticError
FloatingPointError
OverflowError
ZeroDivisionError
SystemError
CodecRegistryError
ReferenceError
BufferError
MemoryError
Warning


EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
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
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
percipio64_creating_new_exception...
percipio65_traceback_objects.py
percipio66_assertions.py
percipio67_chaining_exceptions.py
10_Automation Programming
Python Projects_2014
CMD_Python_Set-Path.txt
excel_code_.py
excel_code_summary_master
excel_code_summary_master - Copy.py
excel_code_summary_master.py
PIP_Help-2.PNG
PIP_Help.PNG
Python Openpyxl Tutorial.py
Python_Clear-Window-Command.txt
python_debug_logging_code.py
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Scripts.docx
Python_Tutorials.md
Scripts - Shortcut.lnk
134 Warning
135 UserWarning
136 DeprecationWarning
137 PendingDeprecationWarning
138 SyntaxWarning
139 RuntimeWarning
140 FutureWarning
141 ImportWarning
142 UnicodeWarning
143 BytesWarning
144 ResourceWarning
145 error
146 Verbose
147 Error
148 TokenError
149 StopTokenizing
150 Empty
151 Full
152 _OptionError
153 TclError
154 Error
155 NoSectionError
156 DuplicateSectionError
157 DuplicateOptionError
158 NoOptionError
159 InterpolationError
160 InterpolationMissingOptionError
161 InterpolationSyntaxError
162 InterpolationDepthError
163 ParsingError
164 MissingSectionHeaderError
165 InvalidConfigType
166 InvalidConfigSet
167 InvalidFgBg
168 InvalidTheme
169 EndOfBlock
170 BdbQuit
171 error
172 _Stop
173 PickleError
174 PicklingError


EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
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
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
percipio64_creating_new_exception...
percipio65_traceback_objects.py
percipio66_assertions.py
percipio67_chaining_exceptions.py
10_Automation Programming
Python Projects_2014
174 PicklingError
175 UnpicklingError
176 _GiveupOnSendfile
177 Error
178 TarError
179 ExtractError
180 ReadError
181 CompressionError
182 StreamError
183 HeaderError
184 EmptyHeaderError
185 TruncatedHeaderError
186 EOFHeaderError
187 InvalidHeaderError
188 SubsequentHeaderError
189 LZMAError
190 RegistryError
191 SubprocessError
192 CalledProcessError
193 TimeoutExpired
194 ErrorDuringImport
195 GeneratorExit
196 SystemExit
197 KeyboardInterrupt
198
199 Math_hierarchy() function starts now


EXPLORER
percipio62_the_exception_hierarchy.py
OPEN EDITORS
percipio62_the_exception_hierarchy...
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
09_Exceptions
percipio60_catching_all_exceptions....
percipio61_catching_specific_excep...
percipio62_the_exception_hierarchy...
percipio62b_all_python_objects_an...
percipio63_exception_payloads.py
198
199 Math_hierarchy() function starts now
200 ArithmeticError
201 FloatingPointError
202 OverflowError
203 ZeroDivisionError
204
205 Most specific error:
206 Handling OverflowError: OverflowError('math range error',)
207 Most specific handler is deepest in hierarchy
208
209 Medium-specific error;
210 Handling ArithmeticError: OverflowError('math range error',)
211 Medium specifically handler catches deeper exceptions
212
213 Least specific error:
214 Handling BaseException: OverflowError('math range error',)
215 Least specific exception handles all exceptions
216 >>>
217 '''