

EXPLORER

OPEN EDITORS

- percipio61\_catching\_specific\_exception\_classes...

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.py
    - percipio61\_catching\_specific\_exception\_classes..
    - percipio62\_the\_exception\_hierarchy.py
    - percipio63\_exception\_payloads.py
    - percipio64\_creating\_new\_exceptions.py
    - 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

```
percipio61_catching_specific_exception_classes.py x
1  '''
2  percipio61_catching_specific_exception_classes.py
3  Percipio video: Exceptions; Catching Specific Exception Classes
4
5  * Demonstrate how to handle specific exceptions in Python
6  * If an error is possible, using the Try/except/else/finally code blocks catches
   the error in a exception object which offers more details.
7  * Important to use the most specific exception first and the least specific
   exception (BaseException) as the last except block
8  * 'z = None' code creates a value for 'z' should a calculation error occur which
   prevents a calculated value
9  '''
10 def specific_handler(x, y):
11     print('Calculating %s/%s' % (x,y))
12     try:
13         z = x / y
14     except ZeroDivisionError as exception:
15         print('Handled ZeroDivisionError:', exception)
16         z = None
17     except ArithmeticError as exception:
18         print('Handled ArithmeticError:', exception)
19         z = None
20     except TypeError as exception:
21         print('Handled TypeError:', exception)
22         z = None
23     except BaseException as exception: # catches all other errors
24         print('Unexpected error:', repr(exception)) # use the repr() function to
           view detailed information on the cause of the exception
25     else: # prints when there is no error
26         print('No exception nor error occurred')
27         return z
28     finally: # always prints, whether error or not.
29         print('"finally" code block run in specific_handler()-function, with or
           without errors')
30
```



EXPLORER

1

OPEN EDITORS 1 UNSAVED

- percipio61\_catching\_specific\_exception\_classes...

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.py
    - percipio61\_catching\_specific\_exception\_classes..
    - percipio62\_the\_exception\_hierarchy.py
    - percipio63\_exception\_payloads.py
    - percipio64\_creating\_new\_exceptions.py
    - 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

percipio61\_catching\_specific\_exception\_classes.py

```
30
31 if __name__ == '__main__': # code block active when this program is run directly,
    and not from a module or remotely
32 # calls the specific_handler() function with given values
33     print('Calling specific_handler(5,2) returned:', specific_handler(5,2.5))
34     print()
35     print('Calling specific_handler(2,0) returned:', specific_handler(2,0))
36     print()
37     print('Calling specific_handler(2,"z") returned:', specific_handler(2,'z'))
38     ...
39     RESULTS:
40     Calculating 5/2.5
41     No exception nor error occurred
42     "finally" code block run in specific_handler()-function, with or without errors
43     Calling specific_handler(5,2) returned: 2.0
44
45     Calculating 2/0
46     Handled ZeroDivisionError: division by zero
47     "finally" code block run in specific_handler()-function, with or without errors
48     Calling specific_handler(2,0) returned: None
49
50     Calculating 2/z
51     Handled TypeError: unsupported operand type(s) for /: 'int' and 'str'
52     "finally" code block run in specific_handler()-function, with or without errors
53     Calling specific_handler(2,"z") returned: None
54     ...
```