



EXPLORER

OPEN EDITORS

- percipio60_catching_all_exceptions...

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_hierarch...
 - percipio63_exception_payloads.py
 - percipio64_creating_new_exceptio...
 - 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_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
- start_code_for_python_master.py

percipio60_catching_all_exceptions.py x

```
1 '''
2 percipio60_catching_all_exceptions.py
3 # Percipio video: Exceptions; Catching All Exceptions
4
5 * Demonstrate how to handle all errors within Python
6     * errors that occur when calling a function from another function, or when calling a
7       function from the main program
8     * erros that occur within a function
9 * BaseException() function is the parent-class of all error exceptions
10    * Write more specific cases first before more generic cases like BaseException
11    * When the 'except' code block executes, the 'else' block will not, but the 'finally'
12      code block always executes
13
14 * repr() function gives detailed information on what is the cause of the exception
15 * Try block does not require any other code block, not except, else, or finally. But
16   without errors will not be identified
17 '''
18 nl = '\n'
19 # Demonstrate error handling when calling a function
20 def all_far(x,y):
21     z = x/y
22     return z
23
24 # Demonstrate error handling within a function
25 def all_close(x, y):
26     try: # NOTE: Generally acceptable to only have statements within the Try block that
27         are likely to cause an error
28         print('Calculating all_close()-function with %s/%s' % (x,y))
29         z = x / y
30     except BaseException as exception: # BaseException catches every type of error & by
31         capturing it as an object named exception, information can be gained from it.
32         print('Unexpected all_close()-function string error ->', str(exception)) # prints
33         the string of the exception object
34         print('Unexpected all_close()-function representation error ->', repr(exception))#
35         prints the representation of the exception object which names the actual error
36         type
37     else:
38         print('No all_close()-function exception occurred')
39         return z
40     finally: # executes every time the Try block is attempted, whether error or not
41         print('"finally" code block run in all_close()-function, with or without errors')
```




EXPLORER

OPEN EDITORS 1 UNSAVED

percipio60_catching_all_exceptions...

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_hierarch...

percipio63_exception_payloads.py

percipio64_creating_new_exceptio...

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_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

start_code_for_python_master.py

percipio60_catching_all_exceptions.py

```
33
34 if __name__ == '__main__': # all code indented under name-equal-main only executes when
    the module is run directly
35     print('Calling all_close(5,2)-function returned:', all_close(5,2))
36     print(n1)
37     print('Calling all_close(2,0)-function returned:', all_close(2,0))
38     print(n1)
39
40 # Use error handling for calling function all_far(), which is from far away, requires
    writing the Try/except block every time.
41     print('Calling all_far(5,2):')
42     z = None
43     try:
44         z = all_far(5, 2)
45     except: # a naked except block not recommended due to no error identifier, but will
        catch every error
46         print('Unexpected all_far(5,2)-function Error')
47     else:
48         print('No all_far(5,2)-function exception occurred')
49     finally:
50         print('"finally" code block run in all_far()-function, with or without errors')
51     print('Calling all_far(5,2) returned:', z)
52     print()
53
54     print('Calling all_far(5,0):')
55     z = None
56     try:
57         z = all_far(5, 0)
58     except BaseException as exception: #
59         print('Unexpected all_far(5,0)-function Representation Error:', repr(exception))
60     else:
61         print('No all_far(5,0)-function exception occurred')
62     finally:
63         print('"finally" code block run in all_far(5,0)-function, with or without errors')
64     print('Calling all_far(5,0) returned:', z)
65     print()
66
67 # Without error handling, the following occurs:
68     print('Without error handling, the program stops & the following occurs.  Read these
        errors from the bottom up.')
```




EXPLORER

1

OPEN EDITORS 1 UNSAVED

- percipio60_catching_all_exceptions...

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_hierarch...
 - percipio63_exception_payloads.py
 - percipio64_creating_new_exceptio...
 - 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_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
- start_code_for_python_master.py

percipio60_catching_all_exceptions.py

```
errors from the bottom up.')
69     print('Calling all_far(0, z) without error handling:')
70     z = all_far(0, 'z')
71     print('This print-line will not print because the program has stopped')
72     print('Calling all_far(0,z) returned:', z)
73 '''
74 RESULT:
75 Calculating all_close()-function with 5/2
76 No all_close()-function exception occurred
77 "finally" code block run in all_close()-function, with or without errors
78 Calling all_close(5,2)-function returned: 2.5
79
80 Calculating all_close()-function with 2/0
81 Unexpected all_close()-function string error -> division by zero
82 Unexpected all_close()-function representation error -> ZeroDivisionError('division by
    zero',)
83 "finally" code block run in all_close()-function, with or without errors
84 Calling all_close(2,0)-function returned: None
85
86 Calling all_far(5,2):
87 No all_far(5,2)-function exception occurred
88 "finally" code block run in all_far()-function, with or without errors
89 Calling all_far(5,2) returned: 2.5
90
91 Calling all_far(5,0):
92 Unexpected all_far(5,0)-function Representation Error: ZeroDivisionError('division by
    zero',)
93 "finally" code block run in all_far(5,0)-function, with or without errors
94 Calling all_far(5,0) returned: None
95
96 Without error handling, the program stops & the following occurs.  Read these errors from
    the bottom up.
97 Calling all_far(0, z) without error handling:
98 Traceback (most recent call last):
99   File "C:/Python36x64/test.py", line 70, in <module>
100     z = all_far(0, 'z')
101   File "C:/Python36x64/test.py", line 17, in all_far
102     z = x/y
103 TypeError: unsupported operand type(s) for /: 'int' and 'str'
104 '''
```