



EXPLORER

OPEN EDITORS

- percipio63\_exception\_payloads...

PYTHON

- 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\_except...
    - percipio61\_catching\_specific\_e...
    - percipio62\_the\_exception\_hier...
    - percipio62b\_all\_python\_object...
    - percipio63\_exception\_payloads..**
    - percipio64\_creating\_new\_exce...
    - percipio65\_traceback\_objects.py
    - percipio66\_assertions.py
    - percipio67\_chaining\_exception...
  - 10\_Automation Programming
- Python Projects\_2014

CMD\_Python\_Set-Path.txt

- excel\_code\_py
- excel\_code\_summary\_master
- excel\_code\_summary\_master - Cop...
- excel\_code\_summary\_master.py
- PIP\_Help-2.PNG
- PIP\_Help.PNG
- Python Openpyxl Tutorial.py
- Python\_Clear-Window-Command.t...
- python\_debug\_logging\_code.py
- python\_exercises\_00.py
- python\_exercises\_01.py

percipio63\_exception\_payloads.py

```
1 '''
2 percipio63_exception_payloads.py
3 Percipio video: Exceptions; Exception Payloads
4
5 Demonstrate raising exceptions with different payloads in Python
6     * Raising errors explicitly by the raise keyword and the error class name with
       appropriate parameters for the particular class
7     * Typically when raising an error or exception, at least one parameter is used to display an
       error message along with the class name
8 '''
9 nl = '\n'
10 import errno # imported to help identify error numbers and use constants that have names
                associated to those numbers
11 import os # imported os module to use the path module
12
13 print(nl, '#1 Check-path example:', nl)
14 def check_file(filepath): # created check_file() function takes a file path as an argument
15     if not os.path.isfile(filepath): # os module->path module->isfile() function can
                determine if a file path is a valid file, here asking if NOT a file, continue
16         raise FileNotFoundError # not recommended to use only the exception name, provide a
                message
17 try:
18     check_file('invalid.path') # invalid.path causes an error because no file path is found
19 except FileNotFoundError as exception:
20     print('#1 Handled FileNotFoundError:', exception)
21
22 print(nl, '#2 FileNotFoundError example:', nl)
23 def check_file_message(filepath): # filepath causes an error because no file path is found
24     if not os.path.isfile(filepath):
25         raise FileNotFoundError('%s' not found' % filepath) # here the given file path is
                substituted for the quoted string %s
26 try:
27     check_file_message('invalid.path')
28 except FileNotFoundError as exception:
29     print('#2 Handled FileNotFoundError:', exception)
30
31 print(nl, '#3 argument example:', nl)
```





EXPLORER	
OPEN EDITORS	
percipio63_exception_payloads...	
PYTHON	
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_except...	
percipio61_catching_specific_e...	
percipio62_the_exception_hier...	
percipio62b_all_python_object...	
percipio63_exception_payloads...	
percipio64_creating_new_exce...	
percipio65_traceback_objects.py	
percipio66_assertions.py	
percipio67_chaining_exception...	
10_Automation Programming	
Python Projects_2014	
CMD_Python_Set-Path.txt	
excel_code_py	
excel_code_summary_master	
excel_code_summary_master - Cop...	
excel_code_summary_master.py	
PIP_Help-2.PNG	
PIP_Help.PNG	
Python Openpyxl Tutorial.py	
Python_Clear-Window-Command.t...	
python_debug_logging_code.py	
python_exercises_00.py	
python_exercises_01.py	
Python_Tutorial_Running-Scripts.d...	
Python_Tutorials.md	
Scripts - Shortcut.lnk	

percipio63\_exception\_payloads.py

```
31 print(nl, '#3 argument example:', nl)
32 def check_file_args(filename):
33     if not os.path.isfile(filename):
34         raise FileNotFoundError(errno.ENOENT, # the error number with ENOENT for an error of
            no entry, from that errno module is the specific error number related with a
            file that's not found.
35         os.strerror(errno.ENOENT), # os module strerror() function takes one of the error
            numbers and shows the associated operating system message.
36         filename, # the original filename
37         2, # another parameter for a number, the WinError number (NOTE: there is no module,
            like errno, which defines all the WinErrors, which are errors specific to the
            Windows platform)
38         'filename2' ) # 2nd filename that's an available parameter for file not found
39 try:
40     check_file_args('invalid.path')
41 except FileNotFoundError as exception:
42     print('#3 Handled FileNotFoundError:', exception)
43     print('Arguments of exception:', exception.args)
44
45 print(nl, '#4 actual-error example:', nl)
46 def check_file_real(filename): # actual file check with an actual error
47     f = open(filename) # assigns variable 'f', a file object, by opening that filename
48     try:
49         check_file_real('invalid.path') # try check_file_real() function with an invalid.path,
            the error is handled correctly
50     except FileNotFoundError as exception: # actual error caused by FileNotFoundError set as an
        exception object
51         print('#4 Handled FileNotFoundError:', exception) # exception,...
52         print('Arguments of exception:', exception.args) # ...and it's arguemts are called
53
54 print(nl, 'Help file starts:', nl)
55 # Output from help() can accept a particular exception class which shows data descriptors
    that can be used from it's parent class. Shows the different parameters that can be
    passed and other methods available with that passed-in class
56 help(FileNotFoundError)
57 '''
58 RESULT:
59 #1 Check-path example:
60 #1 Handled FileNotFoundError:
61
```





EXPLORER

OPEN EDITORS 1 UNSAVED

percipio63\_exception\_payloads...

PYTHON

- 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\_except...
    - percipio61\_catching\_specific\_e...
    - percipio62\_the\_exception\_hier...
    - percipio62b\_all\_python\_object...
    - percipio63\_exception\_payloads...
    - percipio64\_creating\_new\_exce...
    - percipio65\_traceback\_objects.py
    - percipio66\_assertions.py
    - percipio67\_chaining\_exception...
  - 10\_Automation Programming
- Python Projects\_2014
- CMD\_Python\_Set-Path.txt
- excel\_code\_py
- excel\_code\_summary\_master
- excel\_code\_summary\_master - Cop...
- excel\_code\_summary\_master.py
- PIP\_Help-2.PNG
- PIP\_Help.PNG
- Python Openpyxl Tutorial.py
- Python\_Clear-Window-Command.t...
- python\_debug\_logging\_code.py
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Running-Scripts.d...
- Python\_Tutorials.md
- Scripts - Shortcut.lnk

percipio63\_exception\_payloads.py

```
61
62 #2 FileNotFoundError example:
63 #2 Handled FileNotFoundError: "invalid.path" not found
64
65 #3 argument example:
66 #3 Handled FileNotFoundError: [WinError 2] No such file or directory: 'invalid.path' ->
    'filename2'
67 Arguments of exception: (2, 'No such file or directory')
68
69 #4 actual-error example:
70 #4 Handled FileNotFoundError: [Errno 2] No such file or directory: 'invalid.path'
71 Arguments of exception: (2, 'No such file or directory')
72
73 Help file starts:
74 Help on class FileNotFoundError in module builtins:
75 class FileNotFoundError(OSError)
76 | File not found.
77 |
78 | Method resolution order:
79 |     FileNotFoundError
80 |     OSError
81 |     Exception
82 |     BaseException
83 |     object
84 |
85 | Methods defined here:
86 |
87 | __init__(self, /, *args, **kwargs)
88 |     Initialize self. See help(type(self)) for accurate signature.
89 |
90 | -----
91 | Methods inherited from OSError:
92 |
93 | __new__(*args, **kwargs) from builtins.type
94 |     Create and return a new object. See help(type) for accurate signature.
95 |
96 | __reduce__(...)
97 |     helper for pickle
98 |
99 | __str__(self, /)
```



EXPLORER	
1	OPEN EDITORS 1 UNSAVED
percipio63_exception_payloads...	
PYTHON	
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_except...	
percipio61_catching_specific_e...	
percipio62_the_exception_hier...	
percipio62b_all_python_object...	
percipio63_exception_payloads...	
percipio64_creating_new_exce...	
percipio65_traceback_objects.py	
percipio66_assertions.py	
percipio67_chaining_exception...	
10_Automation Programming	
Python Projects_2014	
CMD_Python_Set-Path.txt	
excel_code.py	
excel_code_summary_master	
excel_code_summary_master - Cop...	
excel_code_summary_master.py	
PIP_Help-2.PNG	
PIP_Help.PNG	
Python Openpyxl Tutorial.py	
Python_Clear-Window-Command.t...	
python_debug_logging_code.py	
python_exercises_00.py	
python_exercises_01.py	
Python_Tutorial_Running-Scripts.d...	
Python_Tutorials.md	
Scripts - Shortcut.lnk	

percipio63\_exception\_payloads.py

```
99 | __str__(self, /)
100 |     Return str(self).
101 |
102 | -----
103 | Data descriptors inherited from OSError:
104 |
105 | characters_written
106 |
107 | errno
108 |     POSIX exception code
109 |
110 | filename
111 |     exception filename
112 |
113 | filename2
114 |     second exception filename
115 |
116 | strerror
117 |     exception strerror
118 |
119 | winerror
120 |     Win32 exception code
121 |
122 | -----
123 | Methods inherited from BaseException:
124 |
125 | __delattr__(self, name, /)
126 |     Implement delattr(self, name).
127 |
128 | __getattr__(self, name, /)
129 |     Return getattr(self, name).
130 |
131 | __repr__(self, /)
132 |     Return repr(self).
133 |
134 | __setattr__(self, name, value, /)
135 |     Implement setattr(self, name, value).
136 |
137 | __setstate__(...)
138 |
```







- EXPLORER
- OPEN EDITORS
- percipio63\_exception\_payloads...
- PYTHON
- 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\_except...
      - percipio61\_catching\_specific\_e...
      - percipio62\_the\_exception\_hier...
      - percipio62b\_all\_python\_object...
      - percipio63\_exception\_payloads...
      - percipio64\_creating\_new\_exce...
      - percipio65\_traceback\_objects.py
      - percipio66\_assertions.py
      - percipio67\_chaining\_exception...

```
percipio63_exception_payloads.py x
137 | __setstate__(...)
138 |
139 | with_traceback(...)
140 |     Exception.with_traceback(tb) --
141 |         set self.__traceback__ to tb and return self.
142 |
143 | -----
144 | Data descriptors inherited from BaseException:
145 |
146 |     __cause__
147 |         exception cause
148 |
149 |     __context__
150 |         exception context
151 |
152 |     __dict__
153 |
154 |     __suppress_context__
155 |
156 |     __traceback__
157 |
158 |     args
159 | ...
```