



EXPLORER

OPEN EDITORS

User Settings C:\Users\pcurtis7...
python_openpyxl_tutorial.py ...
percipio65_traceback_objects...
test.py

PYTHON

percipio33_properties.py
percipio34_properties_with_i...
percipio35_operator_overloa...
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
09_Exceptions
percipio60_catching_all_exce...
percipio61_catching_specific...
percipio62_the_exception_hie...
percipio62b_all_python_obje...
percipio63_exception_payloa...
percipio64_creating_new_exc...
percipio65_traceback_objects...
percipio66_assertions.py
percipio67_chaining_exceptio...
10_Automation Programming
percipio68_identifying_how_to...
percipio69_installing_python...
percipio70_managing_packa...
percipio71_using_variables_a...
percipio72_working_with_fu...
percipio73_using_lists_tuples...
percipio74_creating_and_run...
percipio75_performing_file_in...
percipio76_manipulating_csv...
percipio77_processing_json_f...
percipio78_parsing_xml_files...
percipio79_retrieving_data_fr...
percipio80_managing_cron.j...
percipio81_creating_zipped_...
percipio82_exercise_using_py...
Python Projects_2014
CMD_Python_Set-Path.txt
excel_code_.py

User Settings python_openpyxl_tutorial.py percipio65_traceback_objects.py x test.py

'''
THERE ARE ERRORS WITH THIS OUTPUT.
percipio65_traceback_objects.py
Percipio video: Exceptions; Traceback Objects

* Demonstrate using traceback objects
* How to capture a traceback object
* How to print a traceback object without stopping the program
* Storing in log files
* Capture into different objects and then manipulate those objects in the main program.
'''

nl = '\n'
import sys # sys module gets the traceback object using 'sys.exc_info' code
import traceback # tracencback module prints the traceback, and the stack objects, and is able
to extract them
import pprint # lets the extracted objects print nicely

def raise_exception(): # function that simply raises an exception
raise RuntimeError('The root cause of the problem') # raises runtimeError

def call_raise_exception(): # another function that executes the raise_exception() function
and added here so there's more than one function to trace
raise_exception()

def trace_call_raise_exception(): # final function
global extracted_traceback, extracted_stack # these two variables are declared global to
be able to access the objects that get created
try: # exception handling is implemented to capture the exception and it's traceback
call_raise_exception() # this function is called, which calls the raise_exception()
function, which raises a RuntimeError
except RuntimeError as exception: # this is exception block handling with the except
block for the RuntimeError so that it can be caught as an object named exception
exc_type, exc_value, exc_traceback = sys.exc_info() # to extract the type of error,
message or value, and traceback object, executing 'sys.exc_info()' will return
these 3 things

print('Handling %s: %s' % (exc_type, exc_value), nl) # prints handling that
'exc_type' or runtime error with a value, which is the message, 'The root cause
of the problem'

test.py



OPEN EDITORS

{} User Settings C:\Users\pcurtis7...	33
python_openpyxl_tutorial.py ...	34
percipio65_traceback_objects...	
test.py	35

PYTHON

percipio33_properties.py	36
percipio34_properties_with_i...	37
percipio35_operator_overloa...	
06_Working-with-Files	38
07_Comprehensions	39
08_iterables-and-Generators	40
09_Exceptions	
percipio60_catching_all_exce...	41
percipio61_catching_specific...	
percipio62_the_exception_hie...	42
percipio62b_all_python_obje...	43
percipio63_exception_payloa...	44
percipio64_creating_new_exc...	
percipio65_traceback_objects...	45
percipio66_assertions.py	46
percipio67_chaining_exceptio...	
10_Automation Programming	47
percipio68_identifying_how_to...	
percipio69_installing_python...	48
percipio70_managing_packa...	49
percipio71_using_variables_a...	50
percipio72_working_with_fu...	51
percipio73_using_lists_tuples...	52
percipio74_creating_and_run...	
percipio75_performing_file_in...	53
percipio76_manipulating_csv...	
percipio77_processing_json_f...	54
percipio78_parsing_xml_files....	55
percipio79_retrieving_data_fr...	
percipio80_managing_cron_j...	56
percipio81_creating_zipped_...	57
percipio82_exercise_using_py...	
Python Projects_2014	
CMD_Python_Set-Path.txt	58

of the problem'

```
print('Print a normal-looking traceback (in red) using traceback.print_exception():',
      nl)
traceback.print_exception(exc_type, exc_value, exc_traceback) # requires all 3
                    dataItems from sys.exc_info()

print(nl, 'Using traceback.print_exc() which should be exactly the same as above but
        with normal text', nl)
traceback.print_exc(file=sys.stdout) # file for sys-dot-standard-out(put)
''' # 'sys.stdout' causes to print output in normal text, not red
# 'print_exc' is the short form of 'print_exception' and accesses the most current
    exception
# Not necessary to pass in the 'exc_type, exc_value, exc_traceback' from
    'sys.exc_info()'
'''

print(nl, 'Using traceback.format_exception() which captures actual code-text making
        up the different lines from print_exception (or print_exc)', nl)
tb_text = traceback.format_exception(exc_type, exc_value, exc_traceback) #
''' # 'tb_text' has an object with several different lines of text in it allowing
    iteration and printing each line
# like 'print_exception', traceback.format_exception requires 'exc_type',
    'exc_value', and 'exc_traceback'
'''

for tb_line in tb_text:
    print(tb_line, end=' ', file=sys.stdout) #
'''

# * each line has a new line so to avoid extra blank lines, specifying 'end' with an
    empty string prevents an extra new line
# * sending the file to 'sys.stdout' so text looks normal, but can send it to any
    open file '''

''' # a stack_trace, or what's seen with a traceback module, print_stack, shows the
    code as its executing line-by-line
# Some things printed are related to the Python IDE (IDLE)
# With regard to the traceback or the stack, to preserve data outside of the except
    block, global variables must be used or a value returned from a function (global
    variables are 'extracted_traceback' and 'extracted_stack') '''
print(nl, 'Using traceback.print_stack() which results in a list of tuples, each
```





EXPLORER	
OPEN EDITORS	
{}	User Settings C:\Users\pcurtis7...
python	python_openpyxl_tutorial.py ...
python	percipio65_traceback_objects...
python	test.py
PYTHON	
percipio33	percipio33_properties.py
percipio34	percipio34_properties_with_i...
percipio35	percipio35_operator_overloa...
06	06_Working-with-Files
07	07_Comprehensions
08	08_Iterables-and-Generators
09	09_Exceptions
percipio60	percipio60_catching_all_exce...
percipio61	percipio61_catching_specific...
percipio62	percipio62_the_exception_hie...
percipio62b	percipio62b_all_python_obje...
percipio63	percipio63_exception_payloa...
percipio64	percipio64_creating_new_exc...
percipio65	percipio65_traceback_objects...
percipio66	percipio66_assertions.py
percipio67	percipio67_chaining_exceptio...
10	10_Automation Programming
percipio68	percipio68_identifying_how_to...
percipio69	percipio69_installing_python....
percipio70	percipio70_managing_packa...
percipio71	percipio71_using_variables_a...
percipio72	percipio72_working_with_fu...
percipio73	percipio73_using_lists_tuples...
percipio74	percipio74_creating_and_run...
percipio75	percipio75_performing_file_in...
percipio76	percipio76_manipulating_csv...
percipio77	percipio77_processing_json_f...
percipio78	percipio78_parsing_xml_files....
percipio79	percipio79_retrieving_data_fr...
percipio80	percipio80_managing_cron_j...
percipio81	percipio81_creating_zipped_...
percipio82	percipio82_exercise_using_py...
Python Projects_2014	
CMD_Python_Set-Path.txt	
excel_code_py	

User Settings		python_openpyxl_tutorial.py	percipio65_traceback_objects.py	test.py
58			<pre>print(nl, 'Using traceback.print_stack() which results in a list of tuples, each containing 4 different items', nl)</pre>	
59			<pre>traceback.print_stack(file=sys.stdout)</pre>	
60			<pre>extracted_traceback = traceback.extract_tb(exc_traceback) # global variable used to receive from the 'traceback' module, 'extract_tb', using the 'exc_traceback' object which will be an object that is a traceback object.</pre>	
61			<pre>''' # This results in a list of tuples, with each tuple containing 4 different items:</pre>	
62			<pre># the file</pre>	
63			<pre># the line number</pre>	
64			<pre># the call that was made</pre>	
65			<pre># what that call was referring to</pre>	
66			<pre>'''</pre>	
67			<pre>extracted_stack = traceback.extract_stack() # global variable to extract a stack by using the traceback module extract_stack() function</pre>	
68				
69			<pre>trace_call_raise_exception()</pre>	
70			<pre>printer = pprint.PrettyPrinter()</pre>	
71				
72			<pre>print(nl, 'Extracted Traceback:', nl)</pre>	
73			<pre>printer.pprint(extracted_traceback)</pre>	
74				
75			<pre>print('Extracted Stack:', nl) #</pre>	
76			<pre>printer.pprint(extracted_stack) # since this is a global variable (extracted_stack), PrettyPrint can be used</pre>	
77				
78			<pre>'''</pre>	
79			<pre>RESULT:</pre>	
80			<pre>Handling <class 'RuntimeError'>: The root cause of the problem</pre>	
81				
82			<pre>Print a normal-looking traceback (in red) using traceback.print_exception():</pre>	
83				
84			<pre>Traceback (most recent call last):</pre>	
85			<pre>File "C:/Python36x64/test.py", line 26, in trace_call_raise_exception</pre>	
86			<pre>call_raise_exception() # this function is called, which calls the raise_exception() function, which raises a RuntimeError</pre>	
87			<pre>File "C:/Python36x64/test.py", line 21, in call_raise_exception</pre>	
88			<pre>raise_exception()</pre>	
89			<pre>File "C:/Python36x64/test.py", line 18, in raise_exception</pre>	
90			<pre>raise RuntimeError('The root cause of the problem') # raises runtimeError</pre>	
91			<pre>RuntimeError: The root cause of the problem</pre>	
92				





EXPLORER

1 OPEN EDITORS 1 UNSAVED

User Settings C:\Users\pcurtis7...

python_openpyxl_tutorial.py ...

percipio65_traceback_objects...

test.py

PYTHON

percipio33_properties.py

percipio34_properties_with_i...

percipio35_operator_overloa...

06_Working-with-Files

07_Comprehensions

08_iterables-and-Generators

09_Exceptions

percipio60_catching_all_exce...

percipio61_catching_specific...

percipio62_the_exception_hie...

percipio62b_all_python_obje...

percipio63_exception_payloa...

percipio64_creating_new_exc...

percipio65_traceback_objects...

percipio66_assertions.py

percipio67_chaining_exceptio...

10_Automation Programming

percipio68_identifying_how_to...

percipio69_installing_python....

percipio70_managing_packa...

percipio71_using_variables_a...

percipio72_working_with_fu...

percipio73_using_lists_tuples...

percipio74_creating_and_run...

percipio75_performing_file_in...

percipio76_manipulating_csv...

percipio77_processing_json_f...

percipio78_parsing_xml_files....

percipio79_retrieving_data_fr...

percipio80_managing_cron_j...

percipio81_creating_zipped_...

percipio82_exercise_using_py...

Python Projects_2014

CMD_Python_Set-Path.txt

excel_code_py

User Settings python_openpyxl_tutorial.py percipio65_traceback_objects.py test.py

92

93 Using traceback.print_exc() which should be exactly the same as above but with normal text

94

95 Traceback (most recent call last):

96 File "C:/Python36x64/test.py", line 26, in trace_call_raise_exception

97 call_raise_exception() # this function is called, which calls the raise_exception()

98 function, which raises a RuntimeError

99 File "C:/Python36x64/test.py", line 21, in call_raise_exception

100 raise_exception()

101 File "C:/Python36x64/test.py", line 18, in raise_exception

102 raise RuntimeError('The root cause of the problem') # raises runtimeError

103 RuntimeError: The root cause of the problem

104

105 Using traceback.format_exception() which captures actual code-text making up the different

106 lines from print_exception (or print_exc)

107

108 Traceback (most recent call last):

109 File "C:/Python36x64/test.py", line 26, in trace_call_raise_exception

110 call_raise_exception() # this function is called, which calls the raise_exception()

111 function, which raises a RuntimeError

112 File "C:/Python36x64/test.py", line 21, in call_raise_exception

113 raise_exception()

114 File "C:/Python36x64/test.py", line 18, in raise_exception

115 raise RuntimeError('The root cause of the problem') # raises runtimeError

116 RuntimeError: The root cause of the problem

117

118 Using traceback.print_stack() which results in a list of tuples, each containing 4 different

119 items

120

121 File "<string>", line 1, in <module>

122 File "C:\Python36x64\lib\idlelib\run.py", line 142, in main

123 ret = method(*args, **kwargs)

124 File "C:\Python36x64\lib\idlelib\run.py", line 460, in runcode

125 exec(code, self.locals)

126 File "C:/Python36x64/test.py", line 67, in <module>

127 trace_call_raise_exception()

128 File "C:/Python36x64/test.py", line 57, in trace_call_raise_exception

129 traceback.print_stack(file=sys.stdout)

130

131 Extracted Traceback:



EXPLORER

1 OPEN EDITORS 1 UNSAVED

User Settings C:\Users\pcurtis7...

python_openpyxl_tutorial.py ...

percipio65_traceback_objects...

test.py

PYTHON

percipio33_properties.py

percipio34_properties_with_i...

percipio35_operator_overloa...

06_Working-with-Files

07_Comprehensions

08_Iterables-and-Generators

09_Exceptions

percipio60_catching_all_exce...

percipio61_catching_specific...

percipio62_the_exception_hie...

percipio62b_all_python_obje...

percipio63_exception_payloa...

percipio64_creating_new_exc...

percipio65_traceback_objects...

percipio66_assertions.py

percipio67_chaining_exceptio...

10_Automation Programming

percipio68_identifying_how_to...

percipio69_installing_python...

percipio70_managing_packa...

percipio71_using_variables_a...

percipio72_working_with_fu...

percipio73_using_lists_tuples...

percipio74_creating_and_run...

percipio75_performing_file_in...

percipio76_manipulating_csv...

percipio77_processing_json_f...

percipio78_parsing_xml_files....

percipio79_retrieving_data_fr...

percipio80_managing_cron_j...

percipio81_creating_zipped_...

percipio82_exercise_using_py...

Python Projects_2014

CMD_Python_Set-Path.txt

excel_code_py

User Settings python_openpyxl_tutorial.py percipio65_traceback_objects.py test.py

128

129 [`<FrameSummary file C:/Python36x64/test.py, line 26 in trace_call_raise_exception>`,

130 `<FrameSummary file C:/Python36x64/test.py, line 21 in call_raise_exception>`,

131 `<FrameSummary file C:/Python36x64/test.py, line 18 in raise_exception>]`

132 Extracted Stack:

133

134 [`<FrameSummary file <string>, line 1 in <module>>`,

135 `<FrameSummary file C:\Python36x64\lib\idlelib\run.py, line 142 in main>`,

136 `<FrameSummary file C:\Python36x64\lib\idlelib\run.py, line 460 in runcode>`,

137 `<FrameSummary file C:/Python36x64/test.py, line 67 in <module>>`,

138 `<FrameSummary file C:/Python36x64/test.py, line 65 in trace_call_raise_exception>]`

139 Handling `<class 'RuntimeError'>`: The root cause of the problem

140

141 Print a normal-looking traceback (in red) using `traceback.print_exception()`:

142

143 Traceback (most recent call last):

144 File "C:/Python36x64/test.py", line 104, in trace_call_raise_exception

145 call_raise_exception() # this function is called, which calls the raise_exception()

146 function, which raises a RuntimeError

147 File "C:/Python36x64/test.py", line 99, in call_raise_exception

148 raise_exception()

149 File "C:/Python36x64/test.py", line 96, in raise_exception

150 raise RuntimeError('The root cause of the problem') # raises runtimeError

151 RuntimeError: The root cause of the problem

152

153 Using `traceback.print_exc()` which should be exactly the same as above but with normal text

154

155 Traceback (most recent call last):

156 File "C:/Python36x64/test.py", line 104, in trace_call_raise_exception

157 call_raise_exception() # this function is called, which calls the raise_exception()

158 function, which raises a RuntimeError

159 File "C:/Python36x64/test.py", line 99, in call_raise_exception

160 raise_exception()

161 File "C:/Python36x64/test.py", line 96, in raise_exception

162 raise RuntimeError('The root cause of the problem') # raises runtimeError

163 RuntimeError: The root cause of the problem

164

165 Using `traceback.format_exception()` which captures actual code-text making up the different

166 lines from `print_exception` (or `print_exc`)

167

168 Traceback (most recent call last):



EXPLORER

OPEN EDITORS 1 UNSAVED

User Settings C:\Users\pcurtis7...
python_openpyxl_tutorial.py ...
percipio65_traceback_objects...
test.py

PYTHON

percipio33_properties.py
percipio34_properties_with_i...
percipio35_operator_overloa...
06_Working-with-Files
07_Comprehensions
08_Iterables-and-Generators
09_Exceptions
percipio60_catching_all_exce...
percipio61_catching_specific...
percipio62_the_exception_hie...
percipio62b_all_python_obje...
percipio63_exception_payloa...
percipio64_creating_new_exc...
percipio65_traceback_objects...
percipio66_assertions.py
percipio67_chaining_exceptio...
10_Automation Programming
percipio68_identifying_how_to...
percipio69_installing_python...
percipio70_managing_packa...
percipio71_using_variables_a...
percipio72_working_with_fu...
percipio73_using_lists_tuples...
percipio74_creating_and_run...
percipio75_performing_file_in...
percipio76_manipulating_csv...
percipio77_processing_json_f...
percipio78_parsing_xml_files....
percipio79_retrieving_data_fr...
percipio80_managing_cron_j...
percipio81_creating_zipped_...
percipio82_exercise_using_py...

User Settings

python_openpyxl_tutorial.py

percipio65_traceback_objects.py

test.py

Traceback (most recent call last):

```
File "C:/Python36x64/test.py", line 104, in trace_call_raise_exception  
    call_raise_exception() # this function is called, which calls the raise_exception()  
    function, which raises a RuntimeError
```

```
File "C:/Python36x64/test.py", line 99, in call_raise_exception  
    raise_exception()
```

```
File "C:/Python36x64/test.py", line 96, in raise_exception  
    raise RuntimeError('The root cause of the problem') # raises runtimeError
```

```
RuntimeError: The root cause of the problem
```

Using traceback.print_stack() which results in a list of tuples, each containing 4 different items

```
File "<string>", line 1, in <module>
```

```
File "C:\Python36x64\lib\idlelib\run.py", line 142, in main
```

```
    ret = method(*args, **kwargs)
```

```
File "C:\Python36x64\lib\idlelib\run.py", line 460, in runcode
```

```
    exec(code, self.locals)
```

```
File "C:/Python36x64/test.py", line 145, in <module>
```

```
    trace_call_raise_exception()
```

```
File "C:/Python36x64/test.py", line 135, in trace_call_raise_exception
```

```
    traceback.print_stack(file=sys.stdout)
```

Extracted Traceback:

```
[<FrameSummary file C:/Python36x64/test.py, line 104 in trace_call_raise_exception>,  
<FrameSummary file C:/Python36x64/test.py, line 99 in call_raise_exception>,  
<FrameSummary file C:/Python36x64/test.py, line 96 in raise_exception>]
```

Extracted Stack:

```
[<FrameSummary file <string>, line 1 in <module>>,  
<FrameSummary file C:\Python36x64\lib\idlelib\run.py, line 142 in main>,  
<FrameSummary file C:\Python36x64\lib\idlelib\run.py, line 460 in runcode>,  
<FrameSummary file C:/Python36x64/test.py, line 145 in <module>>,  
<FrameSummary file C:/Python36x64/test.py, line 143 in trace_call_raise_exception>]  
...
```