



File Edit View Navigate Code Refactor Run Tools VCS Window Help Test_pickling.py - Exception.py

Module07 - Files and Exceptions > Exception.py

Project

- Module07 - Files and Exceptions [Module07]
 - _Mod7PythonProgrammingNotes.docx
 - Assignment07.docx
 - Assignment07_screen_capture.pptx
 - Exception.py
 - Figure13.png
 - Lab7-1_Starter.py
 - Test_demo_pickling.txt
 - Test_pickling.py
 - ~\$Assignment07_screen_capture.pptx
 - ~\$signment07.docx

Test_pickling.py × Exception.py ×

```
1 # Use this script to demonstrate Exception handling
2 # Use this site to learn about Exception handling: https://realpython.com/python-exceptions/
3 # Created by Tao Peng on 05.19.2021
4
5 # Exceptions versus Syntax Errors
6
7 print( 0 / 0 ))
8 print( 0 / 0 )
```

Python Console

```
>>> print( 0 / 0 ))
File "<input>", line 1
    print( 0 / 0 ))
            ^
SyntaxError: unmatched ')'
```

```
>>> print( 0 / 0 )
Traceback (most recent call last):
  File "<input>", line 1, in <module>
    ZeroDivisionError: division by zero
>>>
```

Run TODO Problems Terminal Python Packages Python Console

8:1 CRLF UTF-8 4 spaces Python 3.9

FileEditViewNavigateCodeRefactorRunToolsVCSWindowHelpTest_pickling.py - Exception.py

Module07 - Files and ExceptionsException.py

Project

Module07 - Files and Exceptions [Module07] C:\U
_Mod7PythonProgrammingNotes.docx
Assignment07.docx
Assignment07_screen_capture.pptx
Exception.py
Figure13.png
Lab7-1_Starter.py
Test_demo_pickling.txt
Test_pickling.py
~\$Assignment07_screen_capture.pptx
~\$Assignment07.docx
External Libraries
Scratches and Consoles

Test_pickling.py

Exception.py

4
5# Exceptions versus Syntax Errors
6print(0 / 0)
7print(0 / 0)
8
9# Raising an Exception
10x = 10
11if x > 5:
12raise Exception('x should not exceed 5. The value of x was: {}'.format(x))

Python ConsoleException

Traceback (most recent call last):
File "<input>", line 4, in <module>
Exception: x should not exceed 5. The value of x was: 10
>>>

RunTODOProblemsTerminalPython PackagesPython Console

12:57CRLFUTF-84 spacesPython 3.9

Windows Taskbar

3:06 PM5/19/2021

I am using a window laptop!

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the active file is 'Test_pickling.py - Exception.py'. The left sidebar shows the Project view with a tree structure for 'Module07 - Files and Exceptions [Module07]'. The main editor window displays the code in 'Exception.py'.

```
4
5 # Exceptions versus Syntax Errors
6 print( 0 / 0 ))
7 print( 0 / 0)
8
9 # Raising an Exception
10 x = 10
11 if x > 5:
12     raise Exception('x should not exceed 5. The value of x was: {}'.format(x))
13
14 # The AssertionError Exception
15 import sys
16 assert ('linux' in sys.platform), "This code runs on Linux only."
```

The Python Console at the bottom shows the following traceback:

```
Traceback (most recent call last):
  File "<input>", line 3, in <module>
AssertionError: This code runs on Linux only.
```

The status bar at the bottom indicates 'PyCharm 2021.1.1 available // Update... (today 12:25 PM)', '16:14 (107 chars, 2 line breaks) CRLF UTF-8 4 spaces Python 3.9', and '1 Event Log'.

```

18 # The try and except Block: Handling Exceptions
19 import sys
20 def linux_interaction():
21     assert ('linux' in sys.platform), "Function can only run on Linux systems."
22     try:
23         linux_interaction()
24     except AssertionError as error:
25         print(error)
26         print('The linux_interaction() function was not executed')

```

linux_interaction()

```

... def linux_interaction():
...     assert ('linux' in sys.platform), "Function can only run on Linux systems."
...     try:
...         linux_interaction()
...     except AssertionError as error:
...         print(error)
...         print('The linux_interaction() function was not executed')
...
Function can only run on Linux systems.
The linux_interaction() function was not executed
>>> |

```

Project

Project

- Module07 - Files and Exceptions [Module07] C:\L
- External Libraries
- Scratches and Consoles

Pickling.py

Exception.py

28

The else and finally Clause

29

import sys

30

def linux_interaction():

31

assert ('linux' in sys.platform), "Function can only run on Linux systems."

32

try:

33

linux_interaction()

34

except AssertionError as error:

35

print(error)

36

else:

37

try:

38

with open('file.log') as file:

39

read_data = file.read()

40

except FileNotFoundError as fnf_error:

41

print(fnf_error)

42

finally:

43

print('Cleaning up, irrespective of any exceptions.')

else > try

5 2 1

Python Console

Function can only run on Linux systems.

Cleaning up, irrespective of any exceptions.

>>>

Special