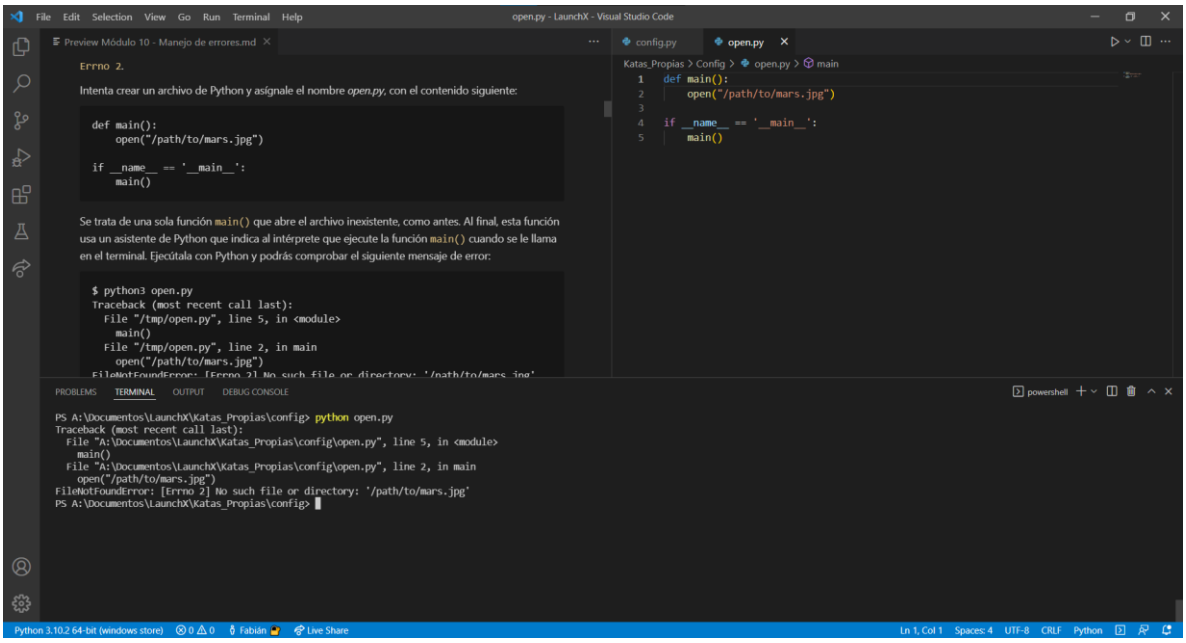
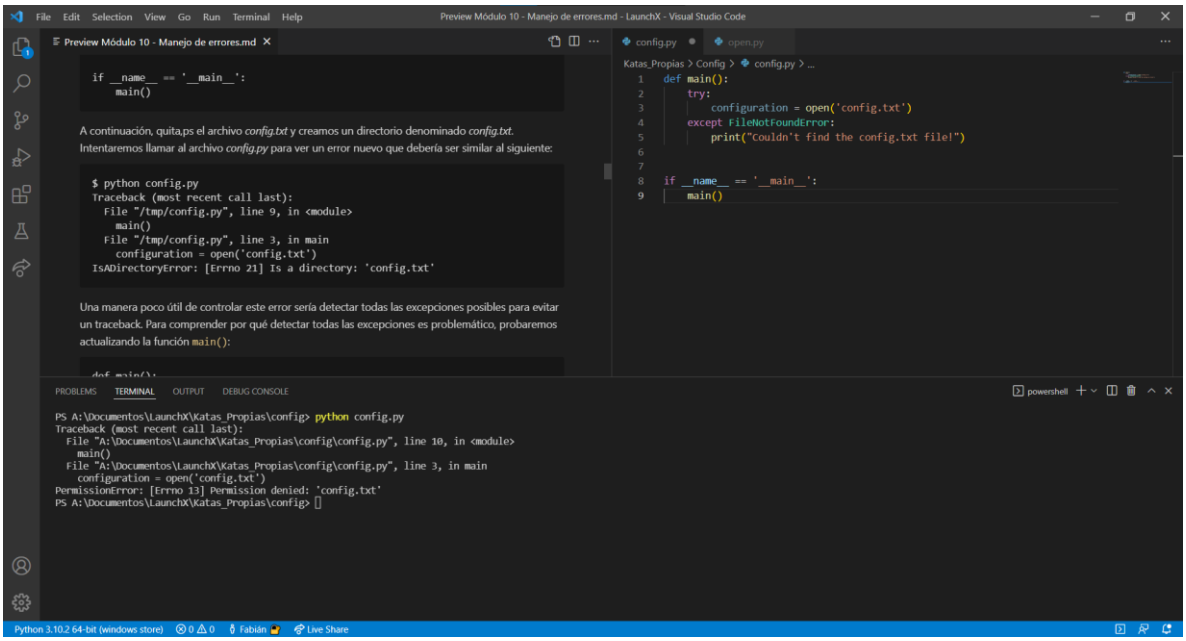
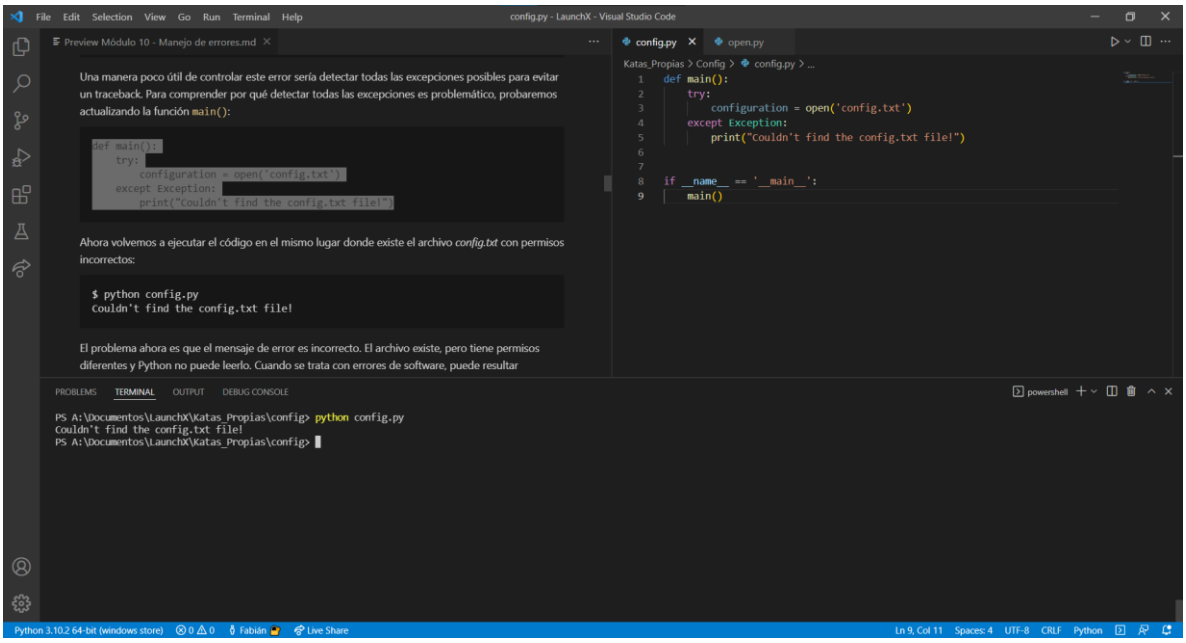
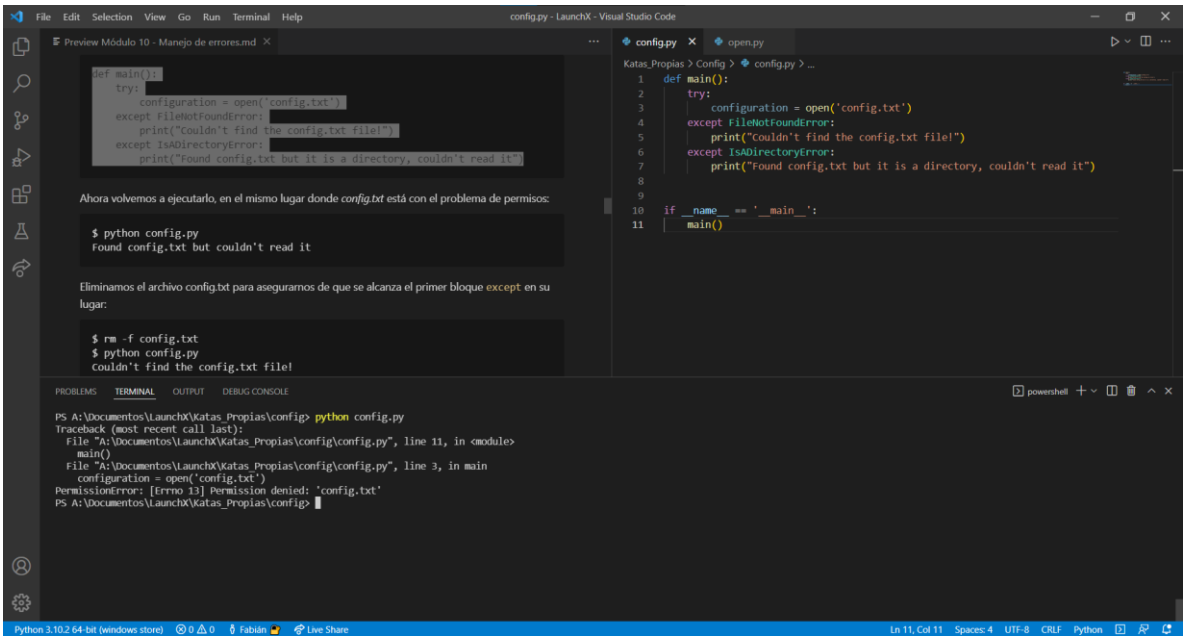
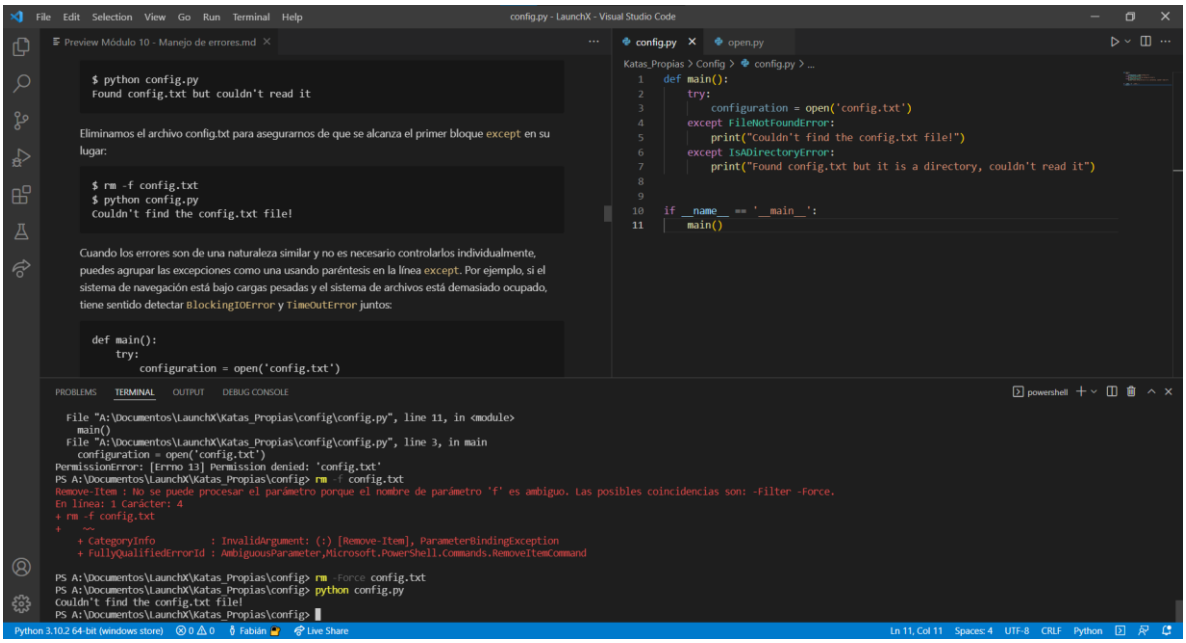
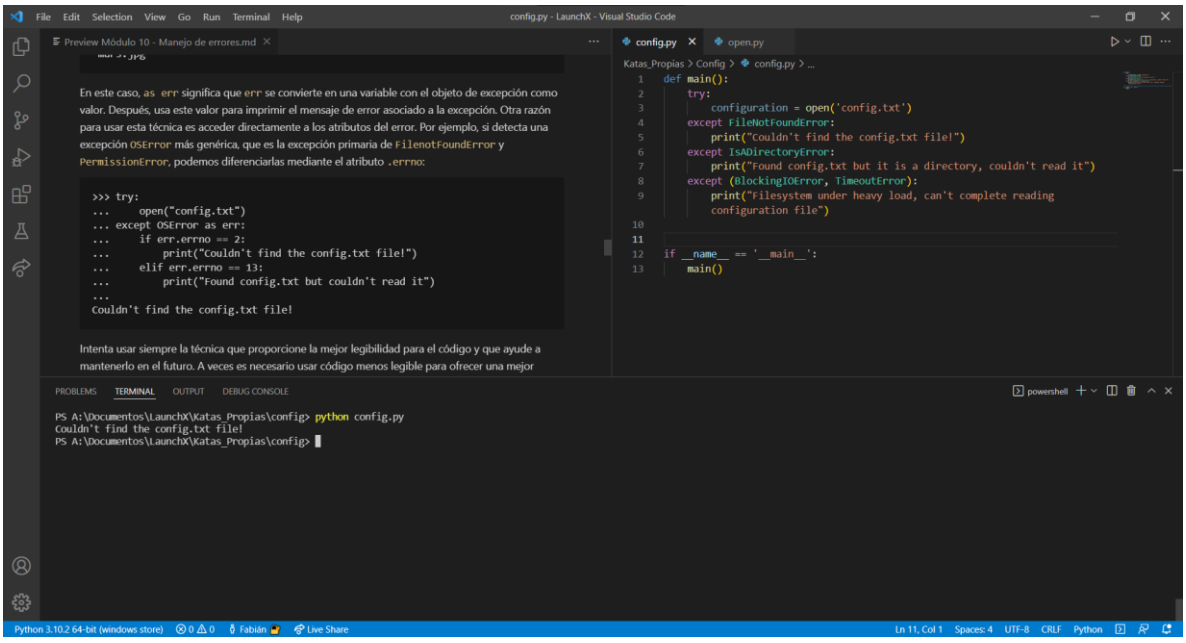

File Edit Selection View Go Run Terminal Help
open.py - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
Errno 2.
Intenta crear un archivo de Python y asignale el nombre open.py, con el contenido siguiente:
def main():
open("/path/to/mars.jpg")
if __name__ == '__main__':
main()
Se trata de una sola función main() que abre el archivo inexistente, como antes. Al final, esta función usa un asistente de Python que indica al intérprete que ejecute la función main() cuando se le llama en el terminal. Ejecútala con Python y podrás comprobar el siguiente mensaje de error:
$ python3 open.py
Traceback (most recent call last):
File "/tmp/open.py", line 5, in <module>
main()
File "/tmp/open.py", line 2, in main
open("/path/to/mars.jpg")
config.py
open.py
Katas_Propias > Config > open.py > main
1 def main():
2 open("/path/to/mars.jpg")
3
4 if __name__ == '__main__':
5 main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
PS A:\Documentos\LaunchX\Katas_Propias\config> python open.py
Traceback (most recent call last):
File "A:\Documentos\LaunchX\Katas_Propias\config\open.py", line 5, in <module>
main()
File "A:\Documentos\LaunchX\Katas_Propias\config\open.py", line 2, in main
open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Python


File Edit Selection View Go Run Terminal Help
Preview Módulo 10 - Manejo de errores.md - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
if __name__ == '__main__':
main()
A continuación, quita,ps el archivo config.txt y creamos un directorio denominado config.txt. Intentaremos llamar al archivo config.py para ver un error nuevo que debería ser similar al siguiente:
$ python config.py
Traceback (most recent call last):
File "/tmp/config.py", line 9, in <module>
main()
File "/tmp/config.py", line 3, in main
configuration = open('config.txt')
IsADirectoryError: [Errno 21] Is a directory: 'config.txt'
Una manera poco útil de controlar este error sería detectar todas las excepciones posibles para evitar un traceback. Para comprender por qué detectar todas las excepciones es problemático, probaremos actualizando la función main():
def main():
config.py
open.py
Katas_Propias > Config > config.py > ...
1 def main():
2 try:
3 configuration = open('config.txt')
4 except FileNotFoundError:
5 print("Couldn't find the config.txt file!")
6
7
8 if __name__ == '__main__':
9 main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
PS A:\Documentos\LaunchX\Katas_Propias\config> python config.py
Traceback (most recent call last):
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 10, in <module>
main()
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 3, in main
configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share


File Edit Selection View Go Run Terminal Help
config.py - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
Una manera poco útil de controlar este error sería detectar todas las excepciones posibles para evitar un traceback. Para comprender por qué detectar todas las excepciones es problemático, probaremos actualizando la función main():
def main():
try:
configuration = open('config.txt')
except Exception:
print("Couldn't find the config.txt file!")
Ahora volvemos a ejecutar el código en el mismo lugar donde existe el archivo config.txt con permisos incorrectos:
$ python config.py
Couldn't find the config.txt file!
El problema ahora es que el mensaje de error es incorrecto. El archivo existe, pero tiene permisos diferentes y Python no puede leerlo. Cuando se trata con errores de software, puede resultar
config.py
open.py
Katas_Propias > Config > config.py > ...
def main():
try:
configuration = open('config.txt')
except Exception:
print("Couldn't find the config.txt file!")
if __name__ == '__main__':
main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
PS A:\Documentos\LaunchX\Katas_Propias\config> python config.py
Couldn't find the config.txt file!
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share
Ln 9, Col 11
Spaces: 4
UTF-8
CRLF
Python


File Edit Selection View Go Run Terminal Help
config.py - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
Ahora volvemos a ejecutarlo, en el mismo lugar donde config.txt está con el problema de permisos:
$ python config.py
Found config.txt but couldn't read it
Eliminamos el archivo config.txt para asegurarnos de que se alcanza el primer bloque except en su lugar:
$ rm -f config.txt
$ python config.py
Couldn't find the config.txt file!
config.py
open.py
Katas_Propias > Config > config.py > ...
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
if __name__ == '__main__':
main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
PS A:\Documentos\LaunchX\Katas_Propias\config> python config.py
Traceback (most recent call last):
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 11, in <module>
main()
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 3, in main
configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share
Ln 11, Col 11
Spaces: 4
UTF-8
CRLF
Python


File Edit Selection View Go Run Terminal Help
config.py - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
$ python config.py
Found config.txt but couldn't read it
Eliminamos el archivo config.txt para asegurarnos de que se alcanza el primer bloque except en su lugar:
$ rm -f config.txt
$ python config.py
Couldn't find the config.txt file!
Cuando los errores son de una naturaleza similar y no es necesario controlarlos individualmente, puedes agrupar las excepciones como una usando paréntesis en la línea except. Por ejemplo, si el sistema de navegación está bajo cargas pesadas y el sistema de archivos está demasiado ocupado, tiene sentido detectar BlockingIOError y TimeOutError juntos:
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
if __name__ == '__main__':
main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 11, in <module>
main()
File "A:\Documentos\LaunchX\Katas_Propias\config\config.py", line 3, in main
configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS A:\Documentos\LaunchX\Katas_Propias\config> rm -f config.txt
Remove-Item : No se puede procesar el parámetro porque el nombre de parámetro 'f' es ambiguo. Las posibles coincidencias son: -Filter -Force.
En línea: 1 Carácter: 4
+ rm -f config.txt
+ ~~
+ CategoryInfo : InvalidArgument: (:) [Remove-Item], ParameterBindingException
+ FullyQualifiedErrorId : AmbiguousParameter,Microsoft.PowerShell.Commands.RemoveItemCommand
PS A:\Documentos\LaunchX\Katas_Propias\config> rm -Force config.txt
PS A:\Documentos\LaunchX\Katas_Propias\config> python config.py
Couldn't find the config.txt file!
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share
Ln 11, Col 11
Spaces: 4
UTF-8
CRLF
Python


File Edit Selection View Go Run Terminal Help
config.py - LaunchX - Visual Studio Code
Preview Módulo 10 - Manejo de errores.md
En este caso, as err significa que err se convierte en una variable con el objeto de excepción como valor. Después, usa este valor para imprimir el mensaje de error asociado a la excepción. Otra razón para usar esta técnica es acceder directamente a los atributos del error. Por ejemplo, si detecta una excepción OSError más genérica, que es la excepción primaria de FilenotFoundError y PermissionError, podemos diferenciarlas mediante el atributo .errno:
>>> try:
... open("config.txt")
... except OSError as err:
... if err.errno == 2:
... print("Couldn't find the config.txt file!")
... elif err.errno == 13:
... print("Found config.txt but couldn't read it")
...
Couldn't find the config.txt file!
Intenta usar siempre la técnica que proporcione la mejor legibilidad para el código y que ayude a mantenerlo en el futuro. A veces es necesario usar código menos legible para ofrecer una mejor
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
except (BlockingIOError, TimeoutError):
print("Filesystem under heavy load, can't complete reading configuration file")
if __name__ == '__main__':
main()
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powershell
PS A:\Documentos\LaunchX\Katas_Propias\config> python config.py
Couldn't find the config.txt file!
PS A:\Documentos\LaunchX\Katas_Propias\config>
Python 3.10.2 64-bit (windows store)
Fabián
Live Share
Ln 11, Col 1
Spaces: 4
UTF-8
CRLF
Python