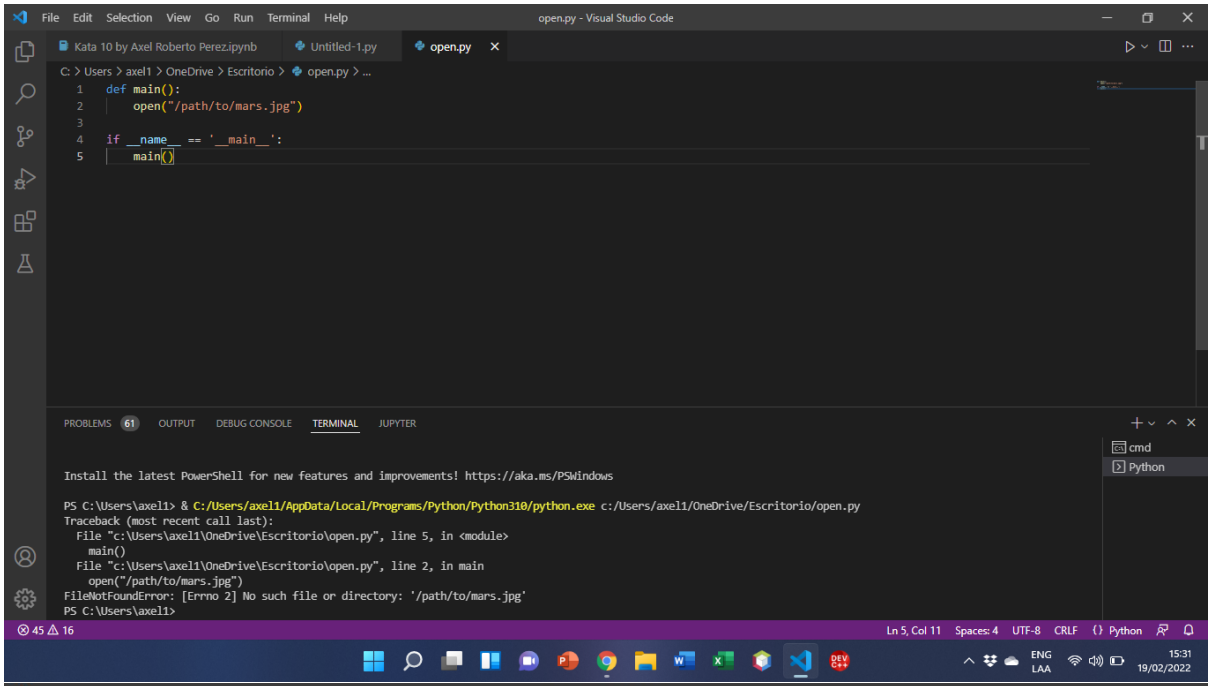
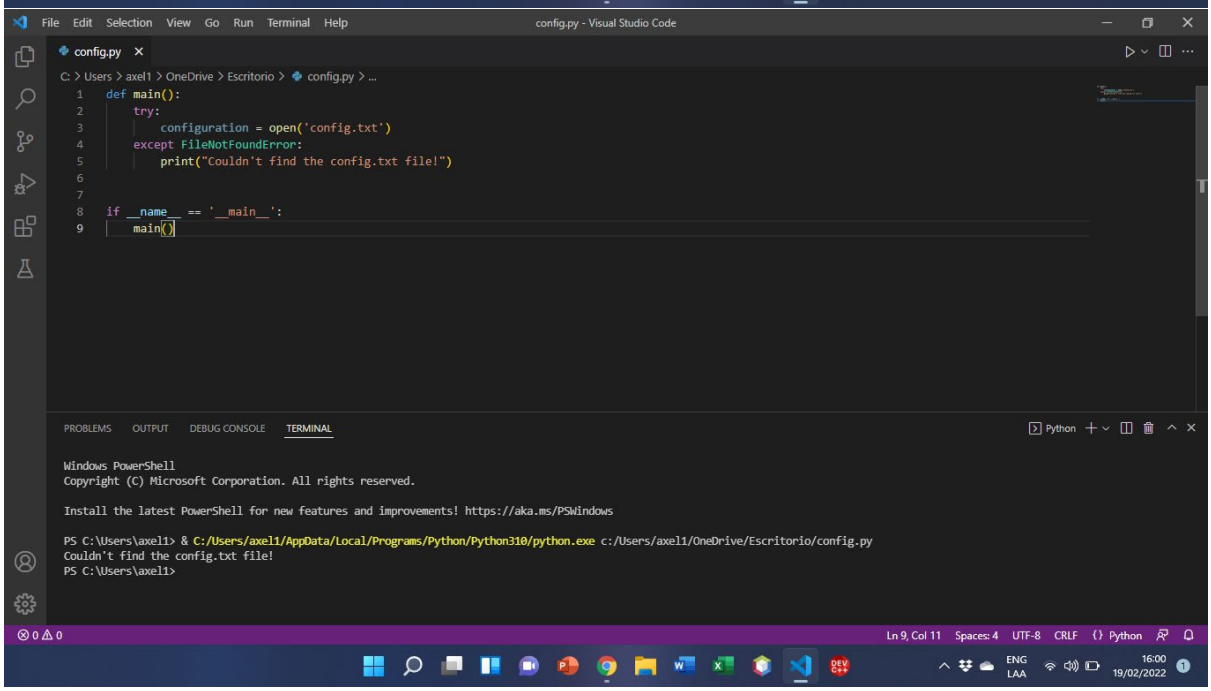
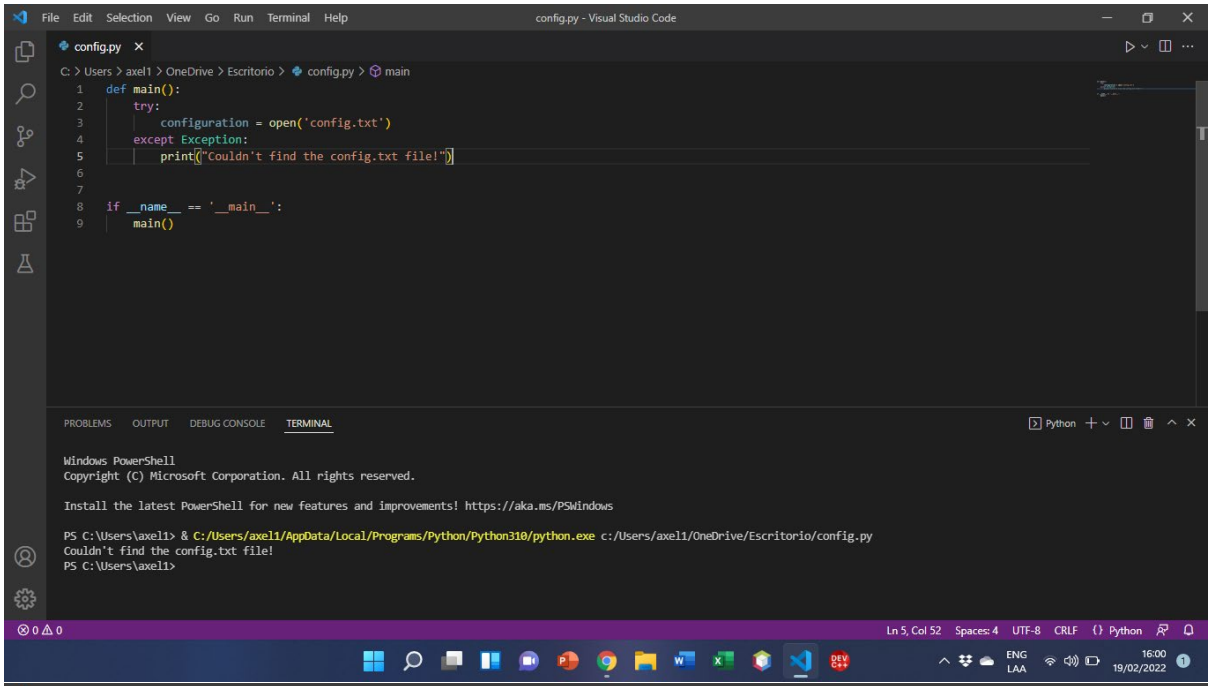
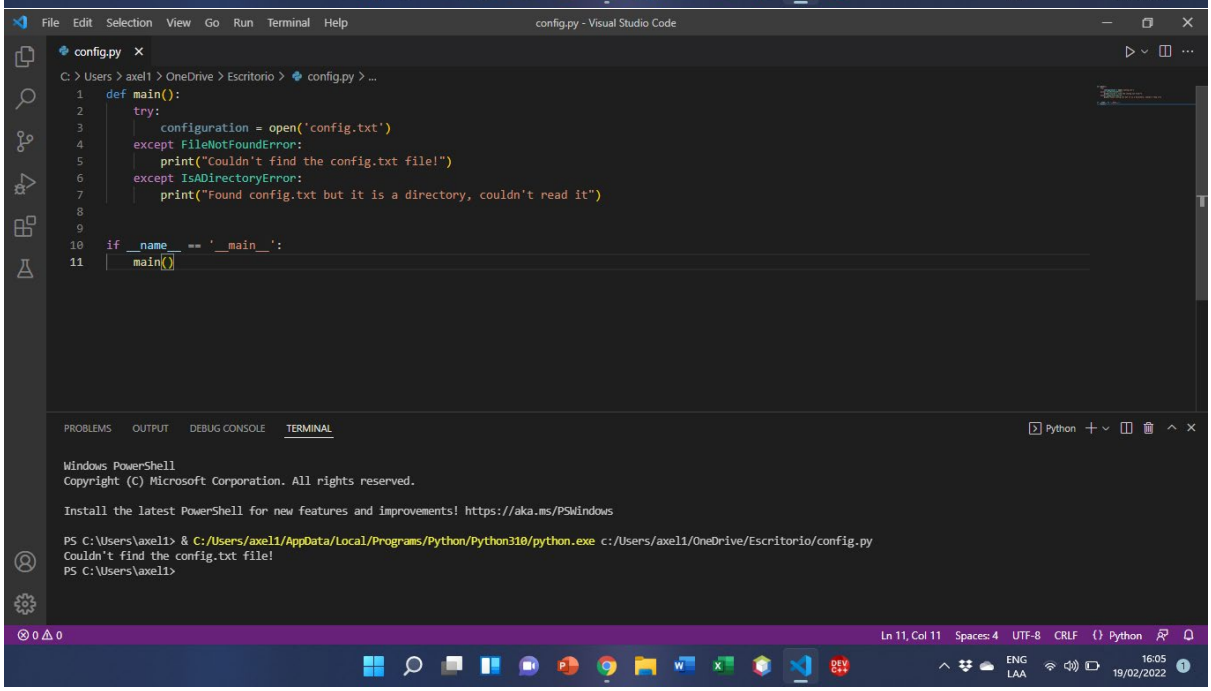
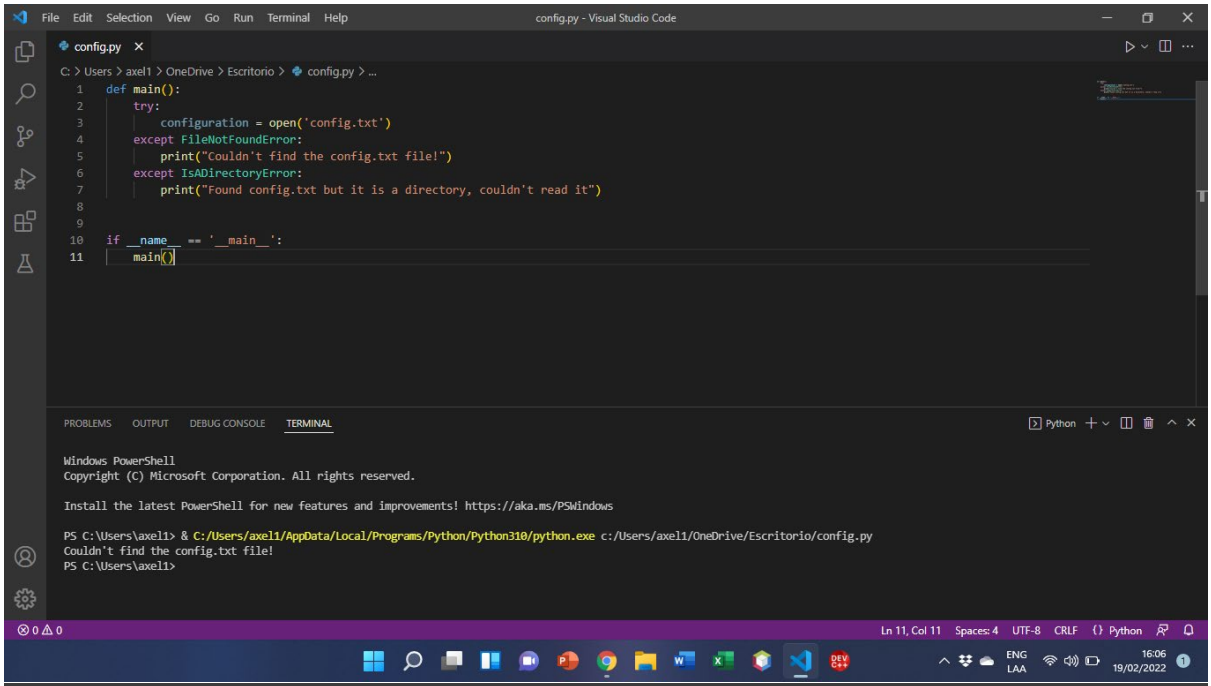
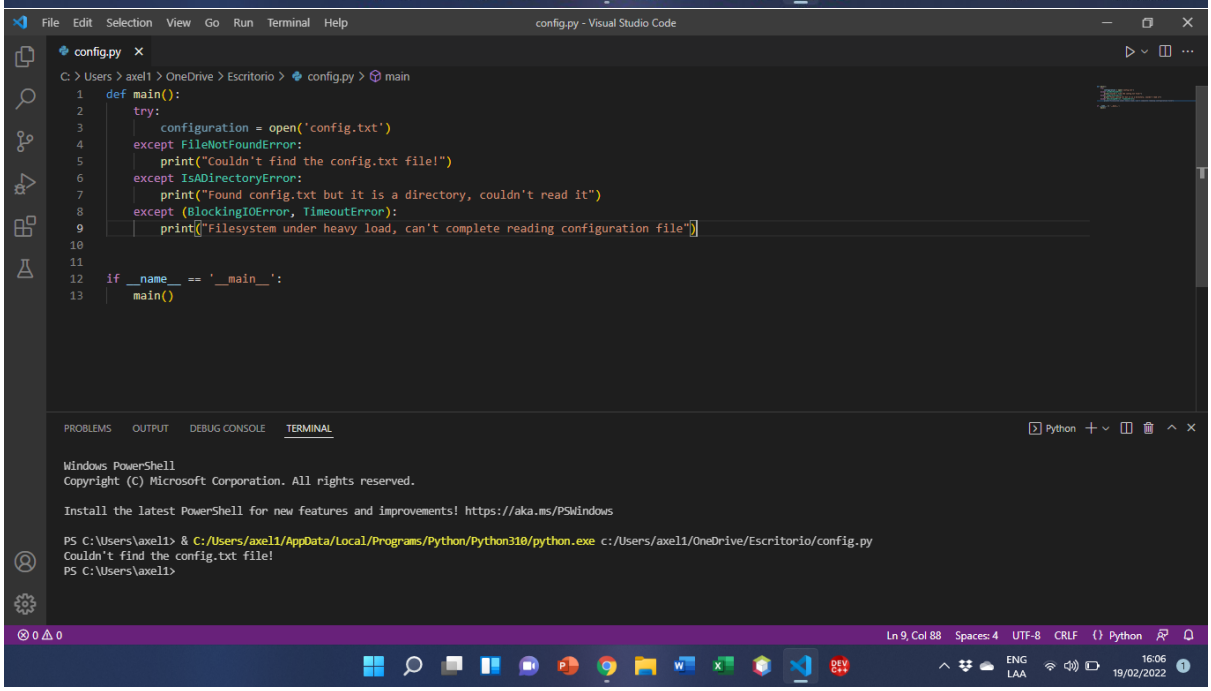

File Edit Selection View Go Run Terminal Help
open.py - Visual Studio Code
Kata 10 by Axel Roberto Perez.ipynb
Untitled-1.py
open.py
C: > Users > axel1 > OneDrive > Escritorio > open.py > ...
1 def main():
2 open("/path/to/mars.jpg")
3
4 if __name__ == '__main__':
5 main()
PROBLEMS 61 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
cmd
Python
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/open.py
Traceback (most recent call last):
File "c:\Users\axel1\OneDrive\Escritorio\open.py", line 5, in <module>
main()
File "c:\Users\axel1\OneDrive\Escritorio\open.py", line 2, in main
open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
PS C:\Users\axel1>
45 16
Ln 5, Col 11 Spaces: 4 UTF-8 CRLF Python
ENG LAA
15:31
19/02/2022

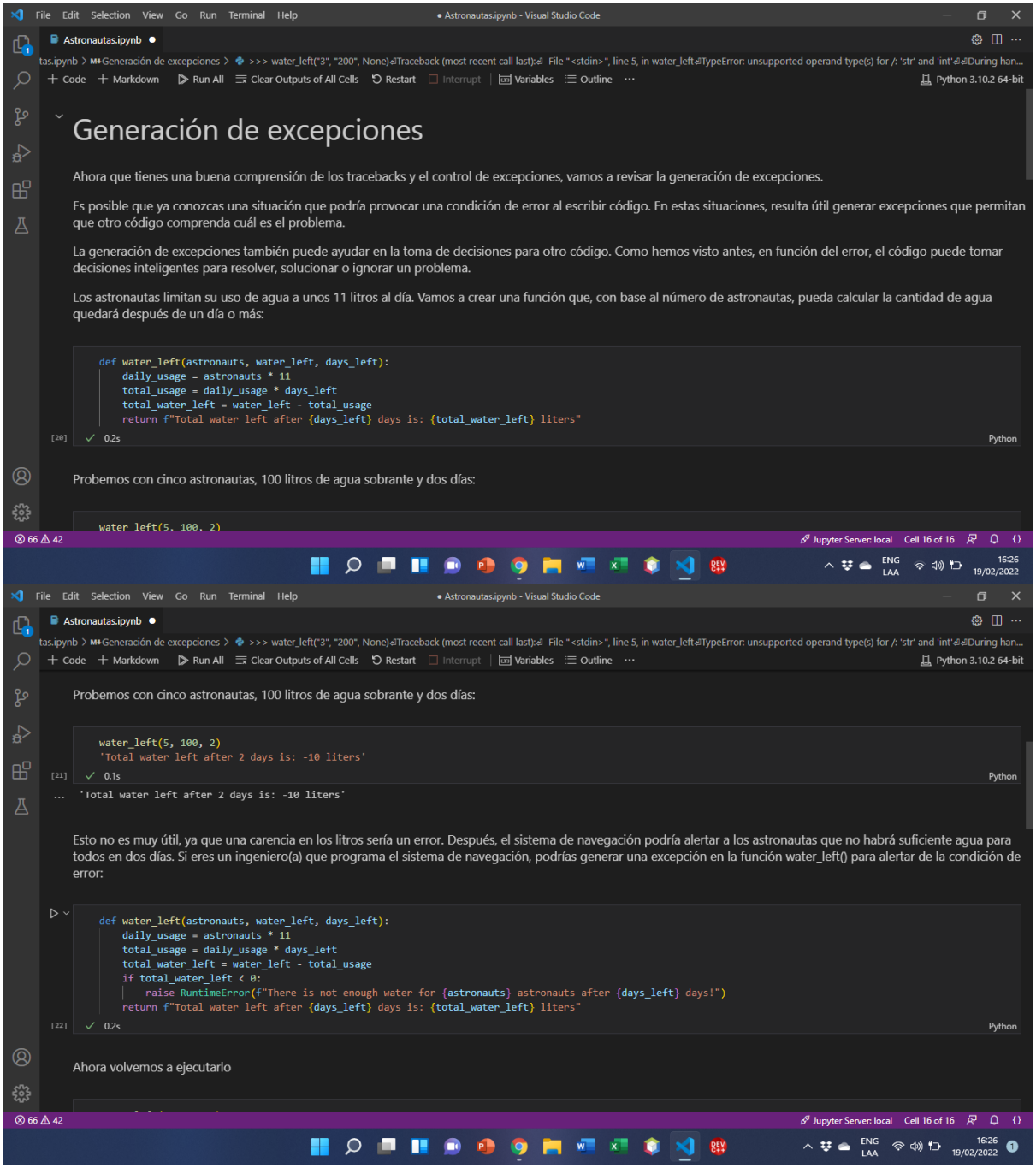

File Edit Selection View Go Run Terminal Help
config.py - Visual Studio Code
config.py
C: > Users > axel1 > OneDrive > Escritorio > config.py > ...
1 def main():
2 try:
3 configuration = open('config.txt')
4 except FileNotFoundError:
5 print("Couldn't find the config.txt file!")
6
7
8 if __name__ == '__main__':
9 main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/config.py
Couldn't find the config.txt file!
PS C:\Users\axel1>
0 0
Ln 9, Col 11 Spaces: 4 UTF-8 CRLF Python
ENG LAA
16:00
19/02/2022


File Edit Selection View Go Run Terminal Help
config.py - Visual Studio Code
config.py
C: > Users > axel1 > OneDrive > Escritorio > config.py > main
def main():
try:
configuration = open('config.txt')
except Exception:
print("Couldn't find the config.txt file!")
if __name__ == '__main__':
main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/config.py
Couldn't find the config.txt file!
PS C:\Users\axel1>
Ln 5, Col 52 Spaces: 4 UTF-8 CRLF Python
ENG
LAA
16:00
19/02/2022

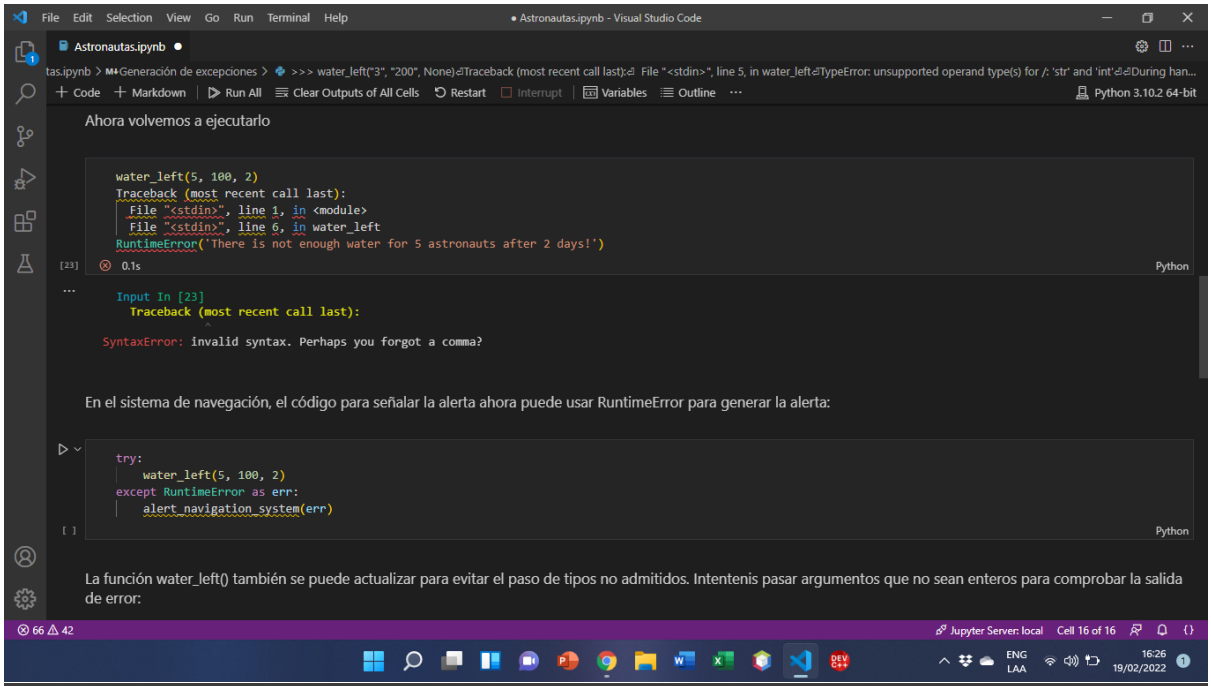

File Edit Selection View Go Run Terminal Help
config.py - Visual Studio Code
config.py
C: > Users > axel1 > OneDrive > Escritorio > config.py > ...
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
if __name__ == '__main__':
main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/config.py
Couldn't find the config.txt file!
PS C:\Users\axel1>
Ln 11, Col 11 Spaces: 4 UTF-8 CRLF Python
ENG
LAA
16:05
19/02/2022

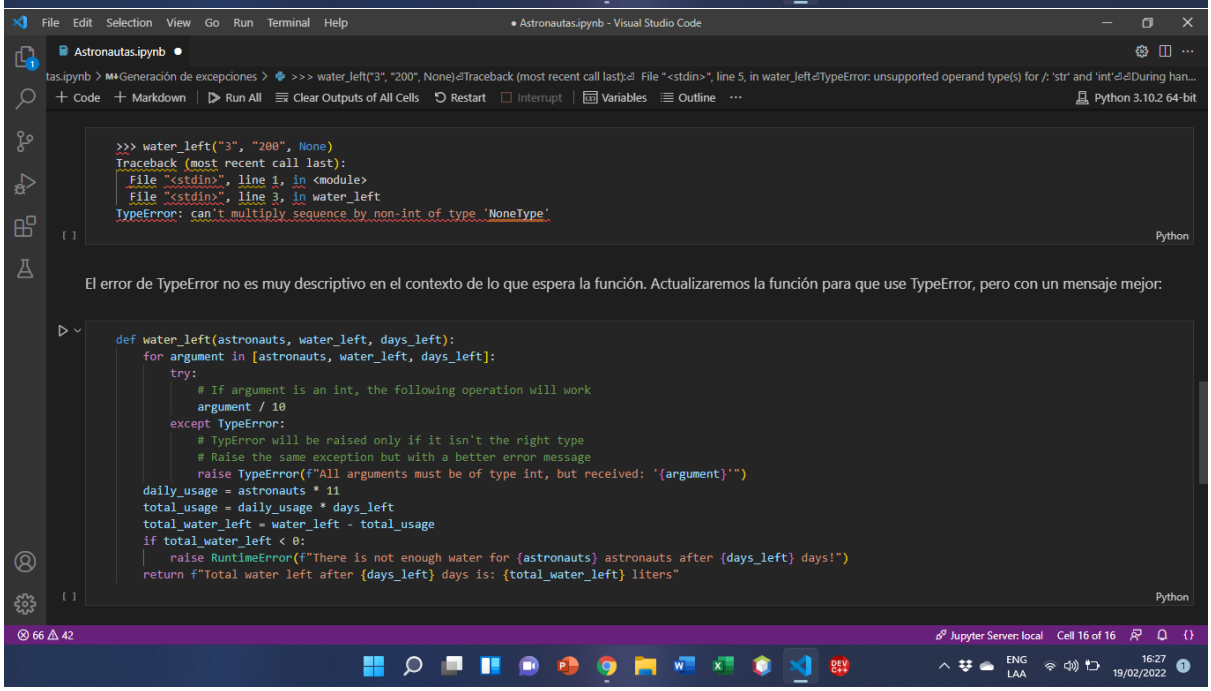

File Edit Selection View Go Run Terminal Help
config.py - Visual Studio Code
config.py
C: > Users > axel1 > OneDrive > Escritorio > config.py > ...
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found config.txt but it is a directory, couldn't read it")
if __name__ == '__main__':
main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/config.py
Couldn't find the config.txt file!
PS C:\Users\axel1>
Ln 11, Col 11 Spaces: 4 UTF-8 CRLF Python
ENG
LAA
16:06
19/02/2022

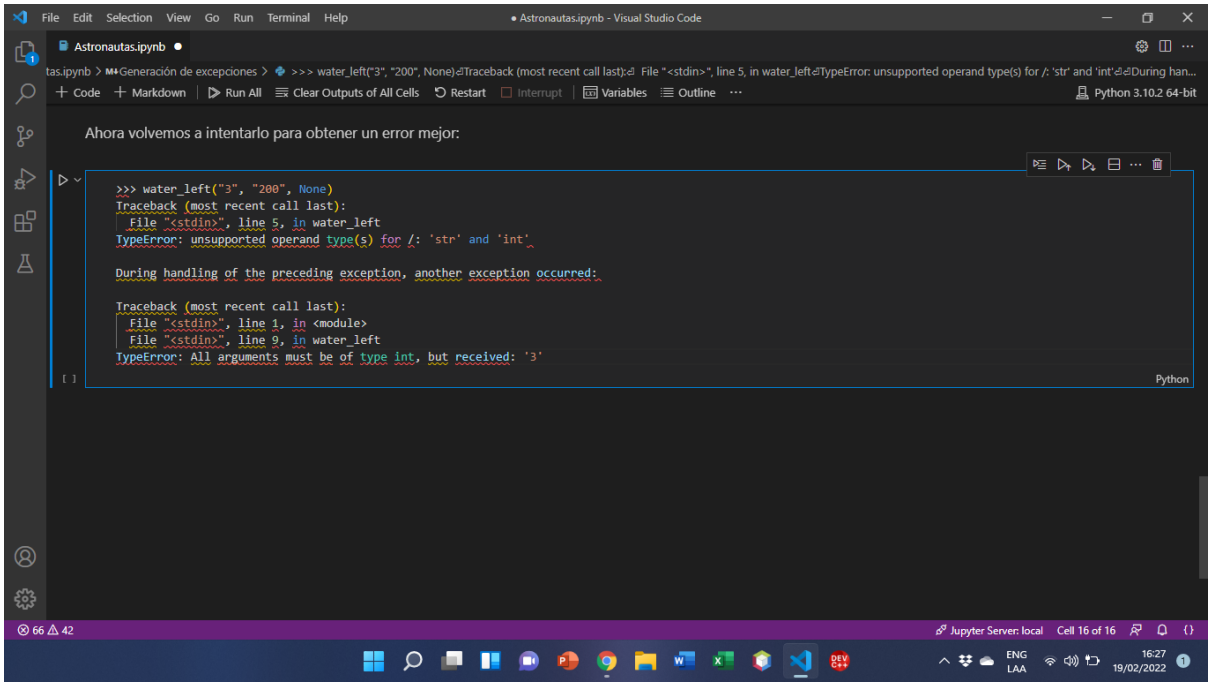

File Edit Selection View Go Run Terminal Help
config.py - Visual Studio Code
config.py
C: > Users > axel1 > OneDrive > Escritorio > config.py > main
def main():
try:
configuration = open('config.txt')
except FileNotFoundError:
print("Couldn't find the config.txt file!")
except IsADirectoryError:
print("Found 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 OUTPUT DEBUG CONSOLE TERMINAL
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\axel1> & C:/Users/axel1/AppData/Local/Programs/Python/Python310/python.exe c:/Users/axel1/OneDrive/Escritorio/config.py
Couldn't find the config.txt file!
PS C:\Users\axel1>
Ln 9, Col 88 Spaces: 4 UTF-8 CRLF Python
ENG
LAA
16:06
19/02/2022


File Edit Selection View Go Run Terminal Help
• Astronautas.ipynb - Visual Studio Code
Astronautas.ipynb
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.2 64-bit
Generación de excepciones
Ahora que tienes una buena comprensión de los tracebacks y el control de excepciones, vamos a revisar la generación de excepciones.
Es posible que ya conozcas una situación que podría provocar una condición de error al escribir código. En estas situaciones, resulta útil generar excepciones que permitan que otro código comprenda cuál es el problema.
La generación de excepciones también puede ayudar en la toma de decisiones para otro código. Como hemos visto antes, en función del error, el código puede tomar decisiones inteligentes para resolver, solucionar o ignorar un problema.
Los astronautas limitan su uso de agua a unos 11 litros al día. Vamos a crear una función que, con base al número de astronautas, pueda calcular la cantidad de agua quedará después de un día o más:
def water_left(astronauts, water_left, days_left):
daily_usage = astronauts * 11
total_usage = daily_usage * days_left
total_water_left = water_left - total_usage
return f"Total water left after {days_left} days is: {total_water_left} liters"
[20] 0.2s
Python
Probemos con cinco astronautas, 100 litros de agua sobrante y dos días:
water_left(5, 100, 2)
Jupyter Server: local
Cell 16 of 16
ENG LAA
16:26 19/02/2022

File Edit Selection View Go Run Terminal Help
• Astronautas.ipynb - Visual Studio Code
Astronautas.ipynb
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.2 64-bit
Probemos con cinco astronautas, 100 litros de agua sobrante y dos días:
water_left(5, 100, 2)
'Total water left after 2 days is: -10 liters'
[21] 0.1s
Python
... 'Total water left after 2 days is: -10 liters'
Esto no es muy útil, ya que una carencia en los litros sería un error. Después, el sistema de navegación podría alertar a los astronautas que no habrá suficiente agua para todos en dos días. Si eres un ingeniero(a) que programa el sistema de navegación, podrías generar una excepción en la función water_left() para alertar de la condición de error:
def water_left(astronauts, water_left, days_left):
daily_usage = astronauts * 11
total_usage = daily_usage * days_left
total_water_left = water_left - total_usage
if total_water_left < 0:
raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
return f"Total water left after {days_left} days is: {total_water_left} liters"
[22] 0.2s
Python
Ahora volvemos a ejecutarlo
Jupyter Server: local
Cell 16 of 16
ENG LAA
16:26 19/02/2022


File Edit Selection View Go Run Terminal Help
• Astronautas.ipynb - Visual Studio Code
Astronautas.ipynb
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.2 64-bit
Ahora volvemos a ejecutarlo
water_left(5, 100, 2)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "<stdin>", line 6, in water_left
RuntimeError('There is not enough water for 5 astronauts after 2 days!')
0.1s
Python
Input In [23]
Traceback (most recent call last):
SyntaxError: invalid syntax. Perhaps you forgot a comma?
En el sistema de navegación, el código para señalar la alerta ahora puede usar RuntimeError para generar la alerta:
try:
water_left(5, 100, 2)
except RuntimeError as err:
alert_navigation_system(err)
Python
La función water_left() también se puede actualizar para evitar el paso de tipos no admitidos. Intentenis pasar argumentos que no sean enteros para comprobar la salida de error:
Jupyter Server: local
Cell 16 of 16
ENG
LAA
16:26
19/02/2022


File Edit Selection View Go Run Terminal Help
• Astronautas.ipynb - Visual Studio Code
Astronautas.ipynb
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.2 64-bit
>>> water_left("3", "200", None)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "<stdin>", line 3, in water_left
TypeError: can't multiply sequence by non-int of type 'NoneType'
Python
El error de TypeError no es muy descriptivo en el contexto de lo que espera la función. Actualizaremos la función para que use TypeError, pero con un mensaje mejor:
def water_left(astronauts, water_left, days_left):
for argument in [astronauts, water_left, days_left]:
try:
# If argument is an int, the following operation will work
argument / 10
except TypeError:
TypError will be raised only if it isn't the right type
Raise the same exception but with a better error message
raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
daily_usage = astronauts * 11
total_usage = daily_usage * days_left
total_water_left = water_left - total_usage
if total_water_left < 0:
raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
return f"Total water left after {days_left} days is: {total_water_left} liters"
Python
Jupyter Server: local
Cell 16 of 16
ENG
LAA
16:27
19/02/2022


File Edit Selection View Go Run Terminal Help
• Astronautas.ipynb - Visual Studio Code
Astronautas.ipynb
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.2 64-bit
Ahora volvemos a intentarlo para obtener un error mejor:
>>> water_left("3", "200", None)
Traceback (most recent call last):
File "<stdin>", line 5, in water_left
TypeError: unsupported operand type(s) for /: 'str' and 'int'
During handling of the preceding exception, another exception occurred:
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "<stdin>", line 9, in water_left
TypeError: All arguments must be of type int, but received: '3'
Python
Jupyter Server: local
Cell 16 of 16
ENG
LAA
16:27
19/02/2022