

```
manejoErrores.ipynb X
launch_x > modulo_10 > manejoErrores.ipynb > open("/path/to/mars.jpg")
+ Code + Markdown | ▶ Run All | Clear Outputs of All Cells | ⏮ Restart | ⏹ Interrupt | Variables | Outline | ...

▶
open("/path/to/mars.jpg")
[1] 0.5s
...
-----
FileNotFoundError                                Traceback (most recent call last)
c:\launch_x\modulo_10\manejoErrores.ipynb Cell 1' in <module>
----> 1 open("/path/to/mars.jpg")

FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
```

```
open.py X
launch_x > modulo_10 > open.py > ...
1 def main():
2     open("/path/to/mars.jpg")
3
4 if __name__ == '__main__':
5     main()
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x> & C:/Python310/python.exe c:/launch_x/modulo_10/open.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\open.py", line 5, in <module>
    main()
  File "c:\launch_x\modulo_10\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:\launch_x> █
```

```
open.py X
launch_x > modulo_10 > open.py
1 try:
2     open('config.txt')
3 except FileNotFoundError:
4     print("Couldn't find the config.txt file!")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x> & C:/Python310/python.exe c:/launch_x/modulo_10/open.py
Couldn't find the config.txt file!
PS C:\launch_x> █
```

```
config.py X
launch_x > modulo_10 > 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 if __name__ == '__main__':
8     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x> & C:/Python310/python.exe c:/launch_x/modulo_10/config.py
Couldn't find the config.txt file!
PS C:\launch_x> 
```

```
config.py X
launch_x > modulo_10 > 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 if __name__ == '__main__':
8     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/config.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\config.py", line 8, in <module>
    main()
  File "c:\launch_x\modulo_10\config.py", line 3, in main
    configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS C:\launch_x\modulo_10> 
```

```
config.py X
launch_x > modulo_10 > config.py > main
1 def main():
2     try:
3         configuration = open('config.txt')
4     except Exception:
5         print("Couldn't find the config.txt file!")
6
7 if __name__ == '__main__':
8     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/config.py
Couldn't find the config.txt file!
PS C:\launch_x\modulo_10>

config.py X
launch_x > modulo_10 > 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     except IsADirectoryError:
7         print("Found config.txt but it is a directory, couldn't read it")
8
9 if __name__ == '__main__':
10    main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/config.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\config.py", line 10, in <module>
    main()
  File "c:\launch_x\modulo_10\config.py", line 3, in main
    configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS C:\launch_x\modulo_10>
```

```
config.py X
launch_x > modulo_10 > 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      except IsADirectoryError:
7          print("Found config.txt but it is a directory, couldn't read it")
8
9  if __name__ == '__main__':
10     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch\_x\modulo\_10> & C:/Python310/python.exe c:/launch\_x/modulo\_10/config.py  
Couldn't find the config.txt file!  
PS C:\launch\_x\modulo\_10> █

```
config.py X
launch_x > modulo_10 > 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      except IsADirectoryError:
7          print("Found config.txt but it is a directory, couldn't read it")
8      except (BlockingIOError, TimeoutError):
9          print("Filesystem under heavy load, can't complete reading configuration file")
10
11 if __name__ == '__main__':
12     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch\_x\modulo\_10> & C:/Python310/python.exe c:/launch\_x/modulo\_10/config.py  
Couldn't find the config.txt file!  
PS C:\launch\_x\modulo\_10> █

```
config.py X
launch_x > modulo_10 > config.py > main
1 def main():
2     try:
3         open("config.txt")
4     except OSError as err:
5         if err.errno == 2:
6             print("Couldn't find the config.txt file!")
7         elif err.errno == 13:
8             print("Found config.txt but couldn't read it")
9
10 if __name__ == '__main__':
11     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/config.py
Couldn't find the config.txt file!
PS C:\launch_x\modulo_10>

water_astronauts.py X
launch_x > modulo_10 > water_astronauts.py > ...
1 def water_left(astronauts, water_left, days_left):
2     daily_usage = astronauts * 11
3     total_usage = daily_usage * days_left
4     total_water_left = water_left - total_usage
5     return f"Total water left after {days_left} days is: {total_water_left} liters"
6     print(water_left(5,100,2))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/water_astronauts.py
Total water left after 2 days is: -10 liters
PS C:\launch_x\modulo_10>

water_astronauts.py X
launch_x > modulo_10 > water_astronauts.py > ...
1 def water_left(astronauts, water_left, days_left):
2     daily_usage = astronauts * 11
3     total_usage = daily_usage * days_left
4     total_water_left = water_left - total_usage
5     if total_water_left < 0:
6         raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
7     return f"Total water left after {days_left} days is: {total_water_left} liters"
8     print(water_left(5,100,2))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/water_astronauts.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\water_astronauts.py", line 8, in <module>
    print(water_left(5,100,2))
  File "c:\launch_x\modulo_10\water_astronauts.py", line 6, in water_left
    raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
RuntimeError: There is not enough water for 5 astronauts after 2 days!
PS C:\launch_x\modulo_10>
```

```
water_astronauts.py X
launch_x > modulo_10 > water_astronauts.py > ...
1 def water_left(astronauts, water_left, days_left):
2     daily_usage = astronauts * 11
3     total_usage = daily_usage * days_left
4     total_water_left = water_left - total_usage
5     if total_water_left < 0:
6         raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
7     return f"Total water left after {days_left} days is: {total_water_left} liters"
8
9 try:
10     water_left(5,100,2)
11 except RuntimeError as err:
12     print(err)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/water_astronauts.py
There is not enough water for 5 astronauts after 2 days!
PS C:\launch_x\modulo_10>

water_astronauts.py X
launch_x > modulo_10 > water_astronauts.py > ...
1 def water_left(astronauts, water_left, days_left):
2     daily_usage = astronauts * 11
3     total_usage = daily_usage * days_left
4     total_water_left = water_left - total_usage
5     if total_water_left < 0:
6         raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
7     return f"Total water left after {days_left} days is: {total_water_left} liters"
8
9 try:
10     water_left("5","100",None)
11 except RuntimeError as err:
12     print(err)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/water_astronauts.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\water_astronauts.py", line 10, in <module>
    water_left("5","100",None)
  File "c:\launch_x\modulo_10\water_astronauts.py", line 3, in water_left
    total_usage = daily_usage * days_left
TypeError: can't multiply sequence by non-int of type 'NoneType'
PS C:\launch_x\modulo_10>
```

```
water_astronauts.py X
launch_x > modulo_10 > water_astronauts.py > water_left
1 def water_left(astronauts, water_left, days_left):
2     for argument in [astronauts, water_left, days_left]:
3         try:
4             # If argument is an int, the following operation will work
5             argument / 10
6         except TypeError:
7             # TypeError will be raised only if it isn't the right type
8             # Raise the same exception but with a better error message
9             raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
10    daily_usage = astronauts * 11
11    total_usage = daily_usage * days_left
12    total_water_left = water_left - total_usage
13    if total_water_left < 0:
14        raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
15    return f"Total water left after {days_left} days is: {total_water_left} liters"
16
17 water_left("5", "100", None)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\launch_x\modulo_10> & C:/Python310/python.exe c:/launch_x/modulo_10/water_astronauts.py
Traceback (most recent call last):
  File "c:\launch_x\modulo_10\water_astronauts.py", line 5, in water_left
    argument / 10
TypeError: unsupported operand type(s) for /: 'str' and 'int'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "c:\launch_x\modulo_10\water_astronauts.py", line 17, in <module>
    water_left("5", "100", None)
  File "c:\launch_x\modulo_10\water_astronauts.py", line 9, in water_left
    raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
TypeError: All arguments must be of type int, but received: '5'
PS C:\launch_x\modulo_10> 
```