

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

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE Python +

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Luis\OneDrive\1 Launch X> & C:/Users/Luis/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Luis/OneDrive/1 Launch X/config.p
PS C:\Users\Luis\OneDrive\1 Launch X> & C:/Users/Luis/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Luis/OneDrive/1 Launch X/open.py"
Traceback (most recent call last):
  File "c:\Users\Luis\OneDrive\1 Launch X\open.py", line 5, in <module>
    main()
  File "c:\Users\Luis\OneDrive\1 Launch X\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\Luis\OneDrive\1 Launch X>
```

```
config.py > main
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")

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE Python + -

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Luis\OneDrive\1 Launch X> & C:/Users/Luis/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Luis/OneDrive/1 Launch X/config.py"
PS C:\Users\Luis\OneDrive\1 Launch X>
```

```
Code | Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline

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"

water_left(5, 100, 2)

[4] ✓ 0.5s

... 'Total water left after 2 days is: -10 liters'
```

```
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"

water_left(5, 100, 2)

[5] ⚠ 0.8s Python

...
-----
RuntimeError                                Traceback (most recent call last)
c:\Users\Luis\OneDrive\1 Launch X\Kata-10-parte2.ipynb Cell 1' in <module>
      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 water_left(5, 100, 2)

c:\Users\Luis\OneDrive\1 Launch X\Kata-10-parte2.ipynb Cell 1' in water_left(astronauts, water_left, 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"

RuntimeError: There is not enough water for 5 astronauts after 2 days!
```

```
# TypeError 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"

water_left("3", "200", None)

⚠ 0.1s

-----
TypeError                                Traceback (most recent call last)
c:\Users\Luis\OneDrive\1 Launch X\Kata-10-parte2.ipynb Cell 1' in water_left(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

TypeError: unsupported operand type(s) for /: 'str' and 'int'

During handling of the above exception, another exception occurred:

TypeError                                Traceback (most recent call last)
c:\Users\Luis\OneDrive\1 Launch X\Kata-10-parte2.ipynb Cell 1' in <module>
     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 water_left("3", "200", None)

c:\Users\Luis\OneDrive\1 Launch X\Kata-10-parte2.ipynb Cell 1' in water_left(astronauts, water_left, days_left)
      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

TypeError: All arguments must be of type int, but received: '3'
```

