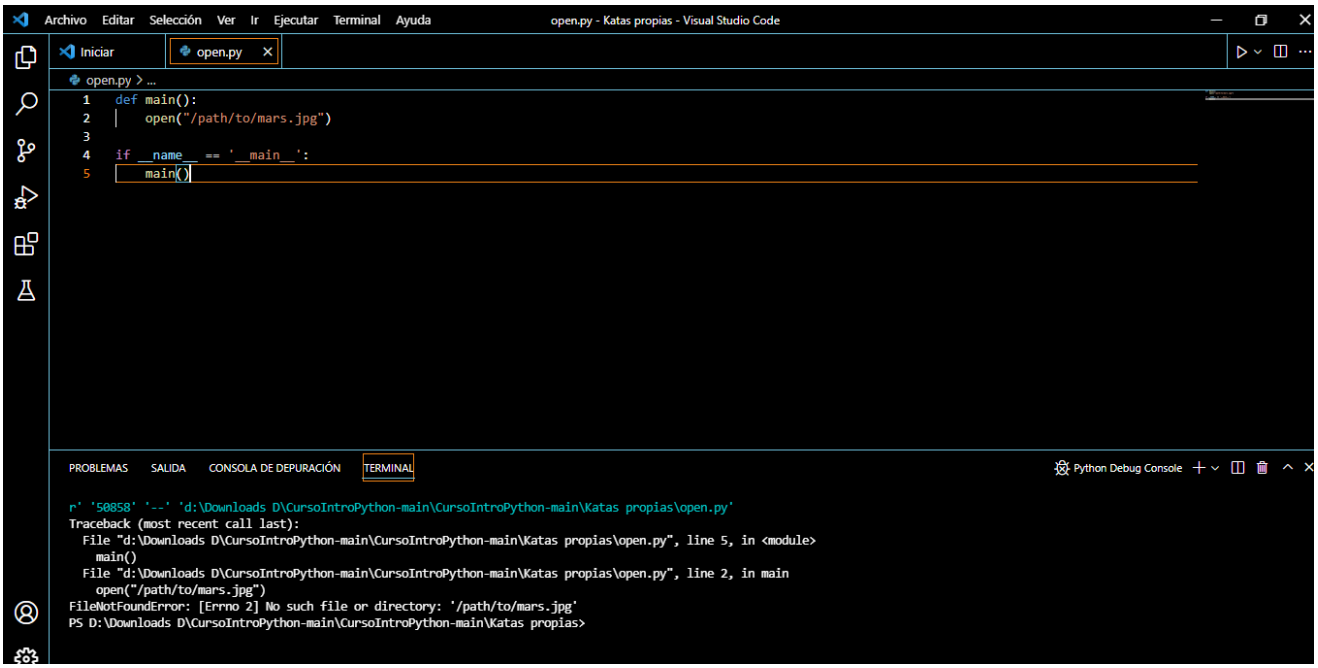
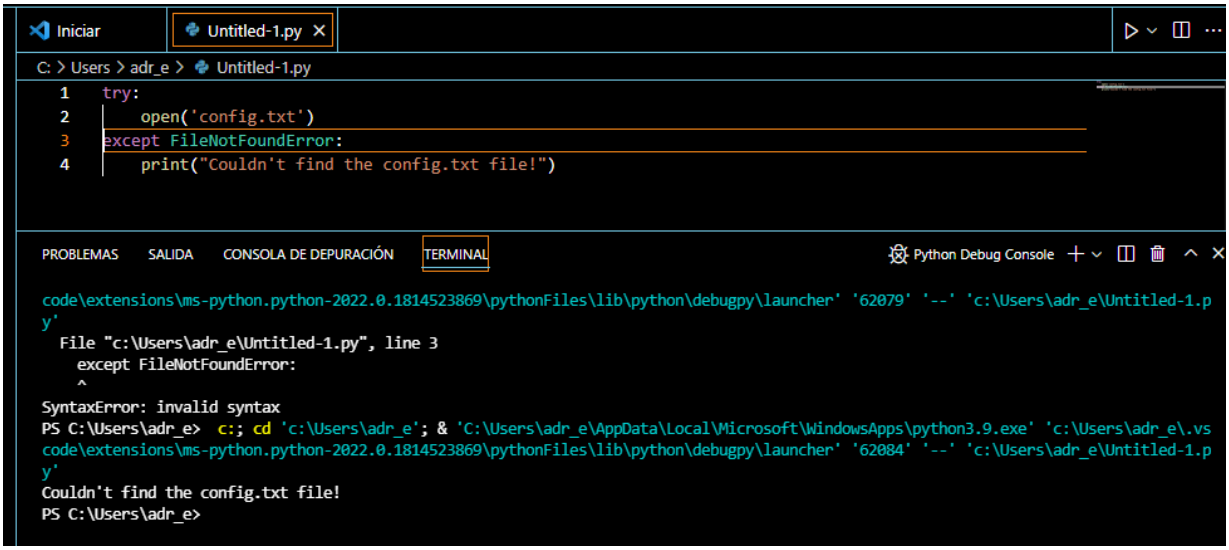
KATA 10

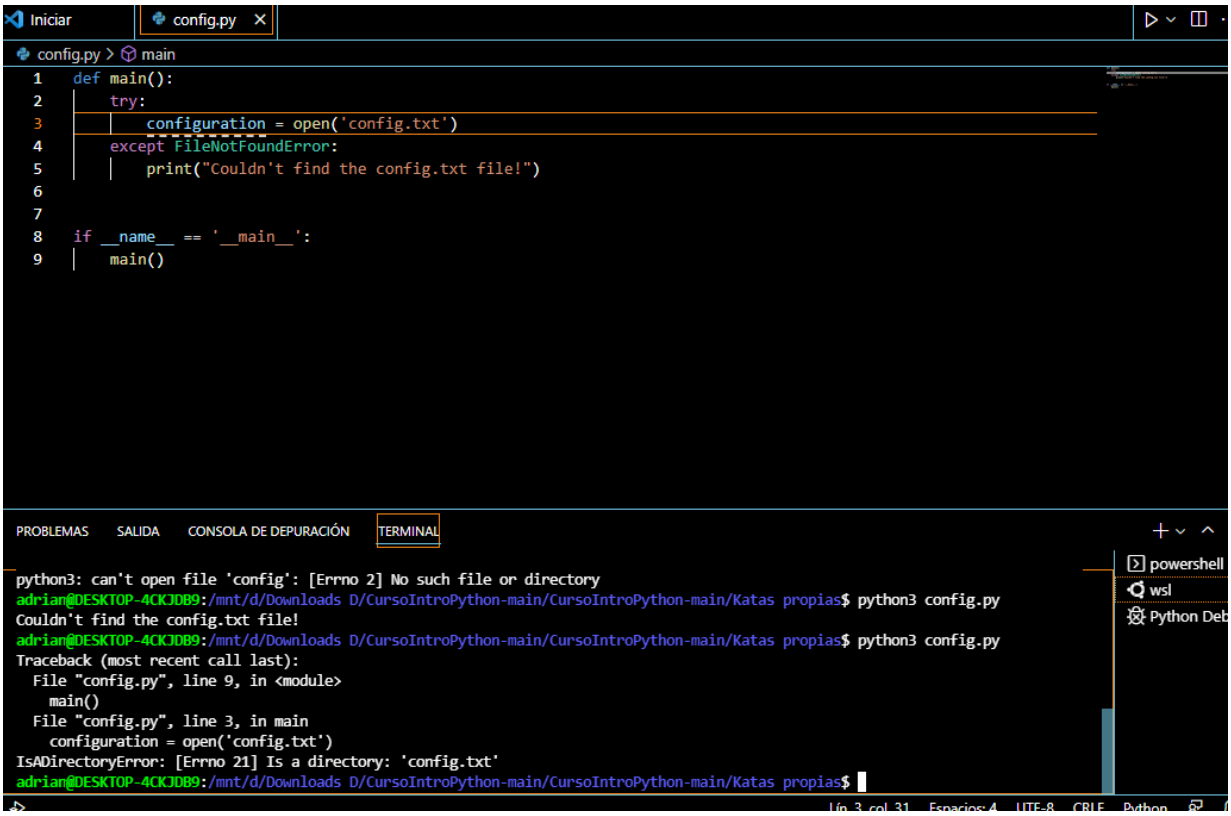


Primer archivo creado en la kata de Excepciones, llamado “open.py”, se genera ese error debido a que no existe el archivo que estamos intentando acceder

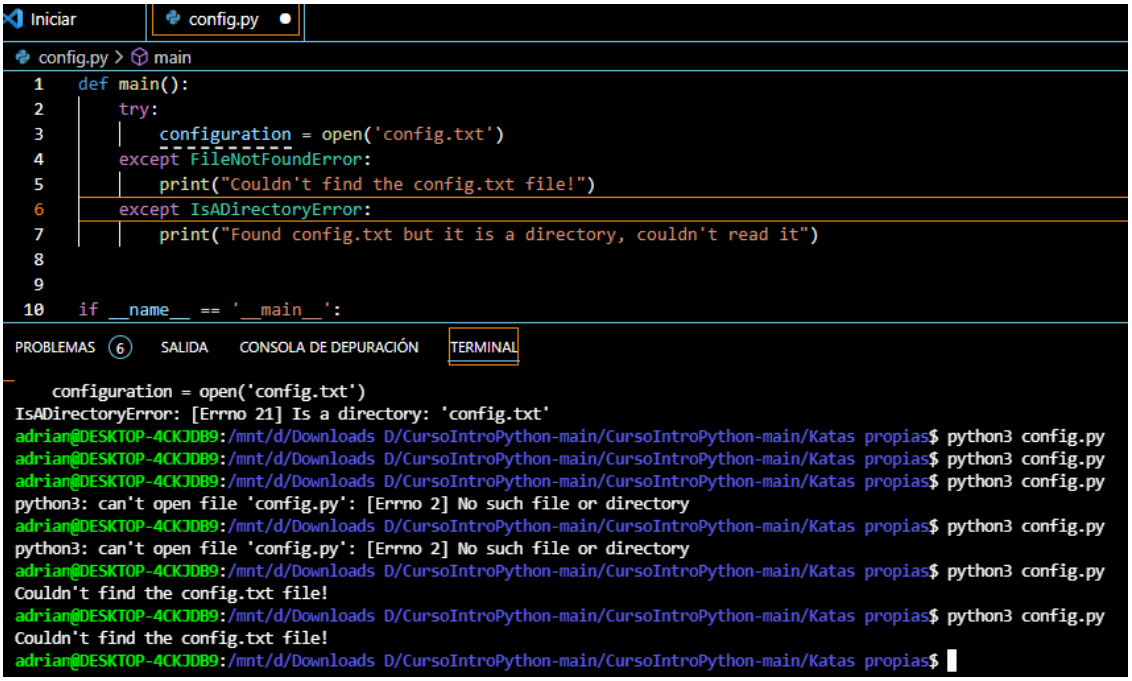


Posteriormente creamos otro archivo, en el que tratamos de controlar el error del ejercicio anterior enviando a pantalla una frase cuando nos ocurría el error, el cual nos informa del problema

KATA 10

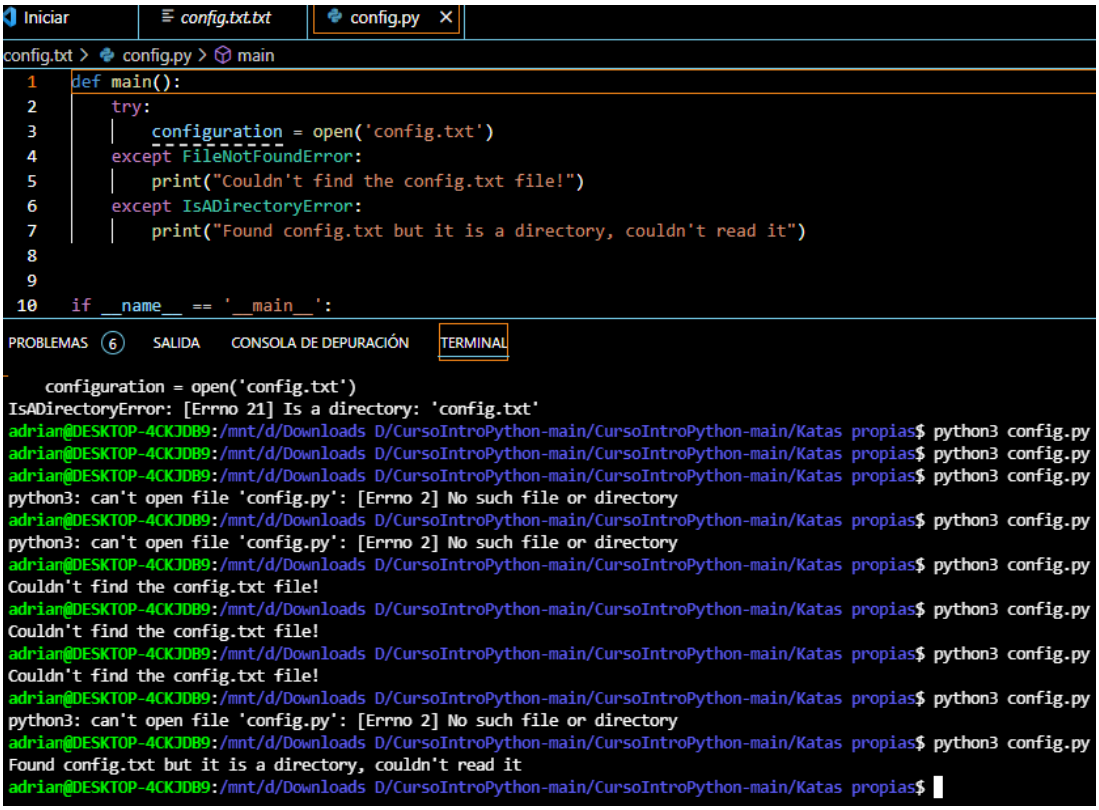


Sin embargo, podía ocurrir casos en los que no tenemos los permisos o como en este caso, creamos un directorio, en vez de un archivo, entonces en lugar de tratar de abrir un archivo, era una dirección por lo que generaba un error como se puede apreciar.

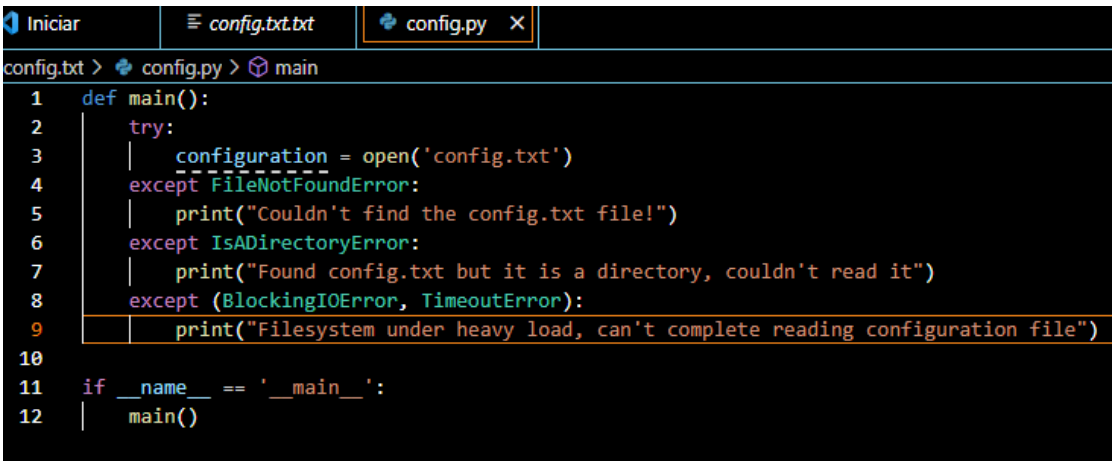


KATA 10

Ahora cambiamos el archivo py junto con el archivo config.py al cual no tenemos permisos, y ahora ya no se genera el problema de que sea un directorio, así que tenemos que agregar un control de excepciones para especificar cual es el error ahora

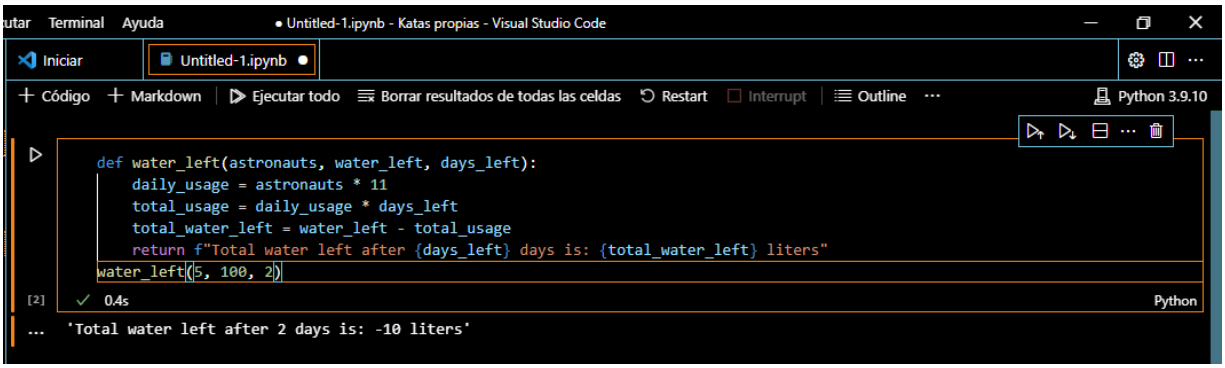


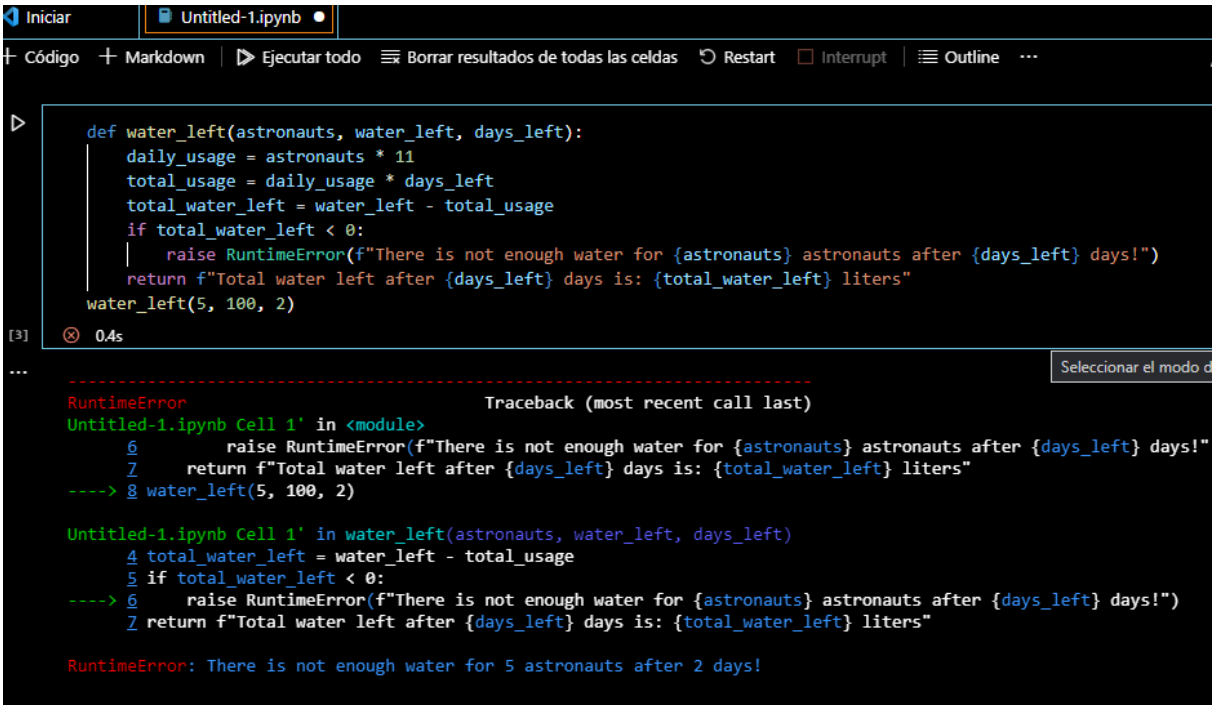
Observamos que, si funciona nuestro control de excepciones, y por ultimo solo añadiremos una excepción mas en caso de encontrar el archivo y no tener los permisos necesarios



Parte 2

KATA 10


utar Terminal Ayuda
• Untitled-1.ipynb - Katas propias - Visual Studio Code
Iniciar
Untitled-1.ipynb
+ Código
+ Markdown
Ejecutar todo
Borrar resultados de todas las celdas
Restart
Interrupt
Outline
Python 3.9.10
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)
[2]
0.4s
Python
'Total water left after 2 days is: -10 liters'


Iniciar
Untitled-1.ipynb
+ Código
+ Markdown
Ejecutar todo
Borrar resultados de todas las celdas
Restart
Interrupt
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
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)
[3]
0.4s
Seleccionar el modo d
RuntimeError Traceback (most recent call last)
Untitled-1.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)
Untitled-1.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!

KATA 10

```
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("3", "200", None)
```

⊗ 0.5s

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Untitled-1.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("3", "200", None)

Untitled-1.ipynb Cell 1' in water_left(astronauts, water_left, days_left)
      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:

TypeError: can't multiply sequence by non-int of type 'NoneType'
```

```
def water_left(astronauts, water_left, days_left):
    for argument in [astronauts, water_left, days_left]:
        try:
            argument / 10
        except TypeError:
            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)
```

⊗ 0.5s

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Untitled-1.ipynb Cell 1' in water_left(astronauts, water_left, days_left)
      3 try:
----> 4     argument / 10
      5 except TypeError:

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

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last)
Untitled-1.ipynb Cell 1' in <module>
```

KATA 10

```
TypeError                                 Traceback (most recent call last)
Untitled-1.ipynb Cell 1' in <module>
     11         raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
     12     return f"Total water left after {days_left} days is: {total_water_left} liters"
---> 13 water_left("3", "200", None)

Untitled-1.ipynb Cell 1' in water_left(astronauts, water_left, days_left)
      4         argument / 10
      5     except TypeError:
----> 6         raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
      7 daily_usage = astronauts * 11
      8 total_usage = daily_usage * days_left

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