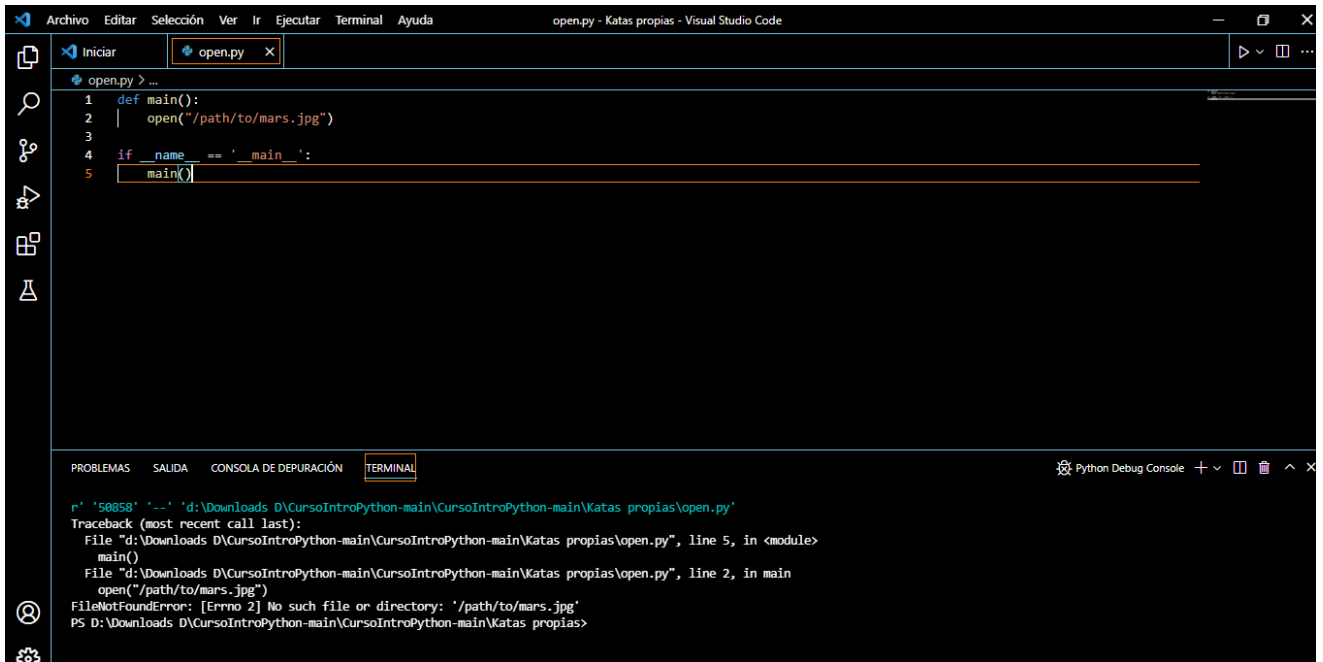


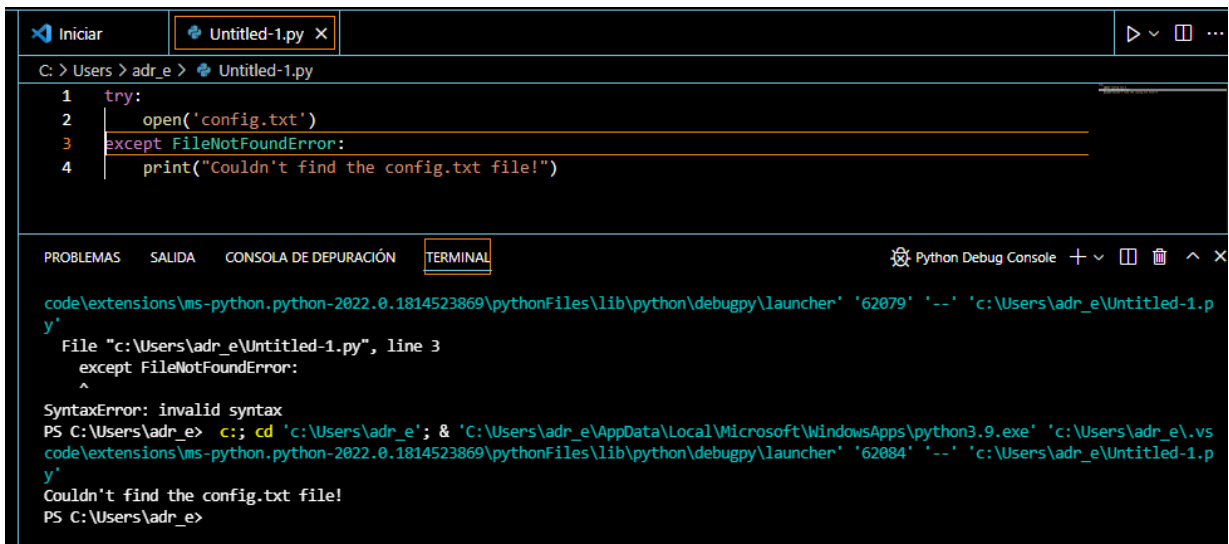
KATA 10



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

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
Python Debug Console
r' '50858' '--' 'd:\Downloads D\CursoIntroPython-main\CursoIntroPython-main\Katas propias\open.py'
Traceback (most recent call last):
  File "d:\Downloads D\CursoIntroPython-main\CursoIntroPython-main\Katas propias\open.py", line 5, in <module>
    main()
  File "d:\Downloads D\CursoIntroPython-main\CursoIntroPython-main\Katas propias\open.py", line 2, in main
    open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
PS D:\Downloads D\CursoIntroPython-main\CursoIntroPython-main\Katas propias>
```

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



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

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
Python Debug Console
code\extensions\ms-python.python-2022.0.1814523869\pythonFiles\lib\python\debugpy\launcher '62079' '--' 'c:\Users\adr_e\Untitled-1.py'
File "c:\Users\adr_e\Untitled-1.py", line 3
except FileNotFoundError:
^
SyntaxError: invalid syntax
PS C:\Users\adr_e> c:: cd 'c:\Users\adr_e'; & 'C:\Users\adr_e\AppData\Local\Microsoft\WindowsApps\python3.9.exe' 'c:\Users\adr_e\vs
code\extensions\ms-python.python-2022.0.1814523869\pythonFiles\lib\python\debugpy\launcher' '62084' '--' 'c:\Users\adr_e\Untitled-1.p
y'
Couldn't find the config.txt file!
PS C:\Users\adr_e>
```

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

```
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()
```

python3: can't open file 'config': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias\$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias\$ python3 config.py
Traceback (most recent call last):
 File "config.py", line 9, in <module>
 main()
 File "config.py", line 3, in main
 configuration = open('config.txt')
IsADirectoryError: [Errno 21] Is a directory: 'config.txt'
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias\$

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.

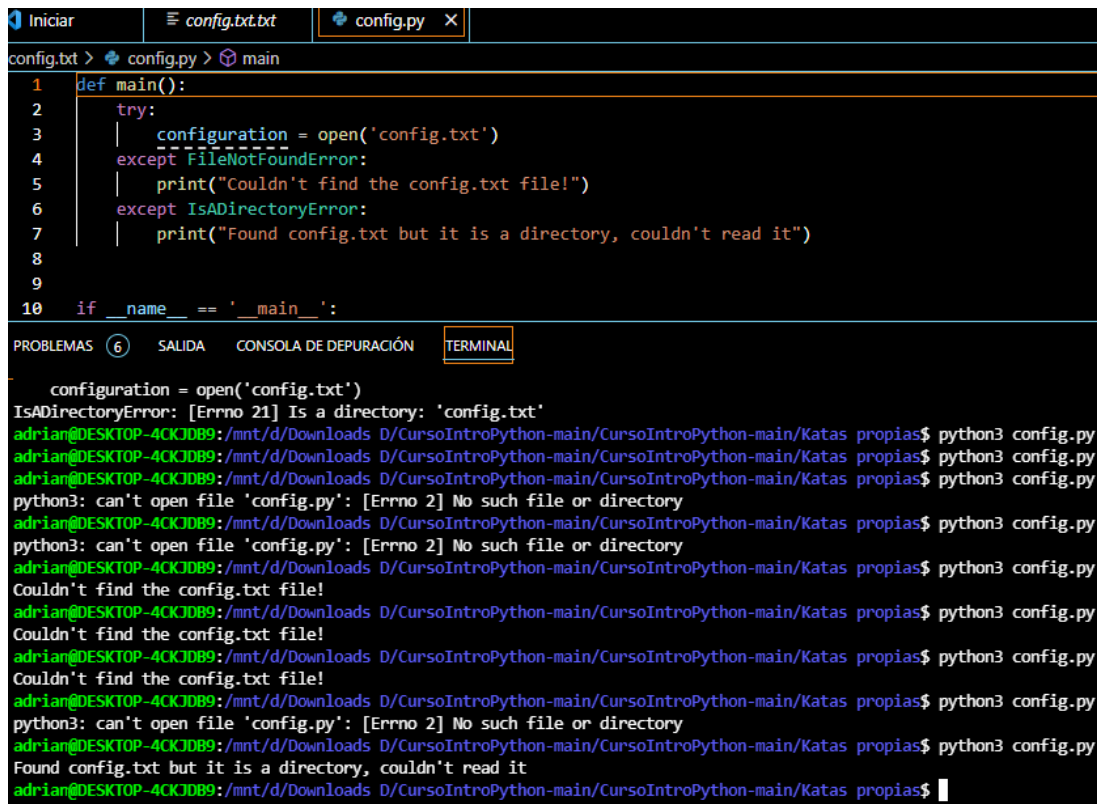
```
config.py > main

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")
8
9
10 if __name__ == '__main__':

configuration = open('config.txt')
IsADirectoryError: [Errno 21] Is a directory: 'config.txt'
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
python3: can't open file 'config.py': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
python3: can't open file 'config.py': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$
```

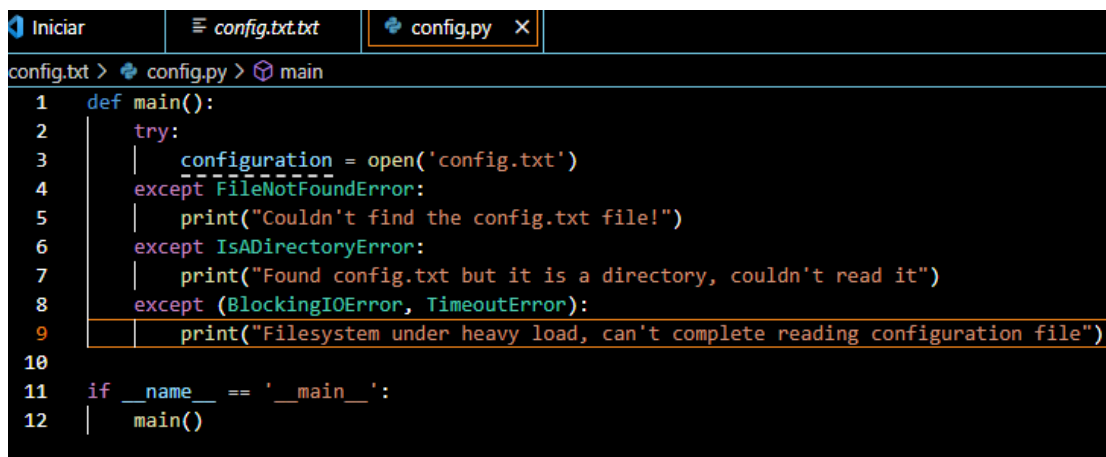
KATA 10

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



```
config.txt > 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")
8
9
10 if __name__ == '__main__':
    configuration = open('config.txt')
IsADirectoryError: [Errno 21] Is a directory: 'config.txt'
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
python3: can't open file 'config.py': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
python3: can't open file 'config.py': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Couldn't find the config.txt file!
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
python3: can't open file 'config.py': [Errno 2] No such file or directory
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$ python3 config.py
Found config.txt but it is a directory, couldn't read it
adrian@DESKTOP-4CKJDB9:/mnt/d/Downloads D/CursoIntroPython-main/CursoIntroPython-main/Katas propias$
```

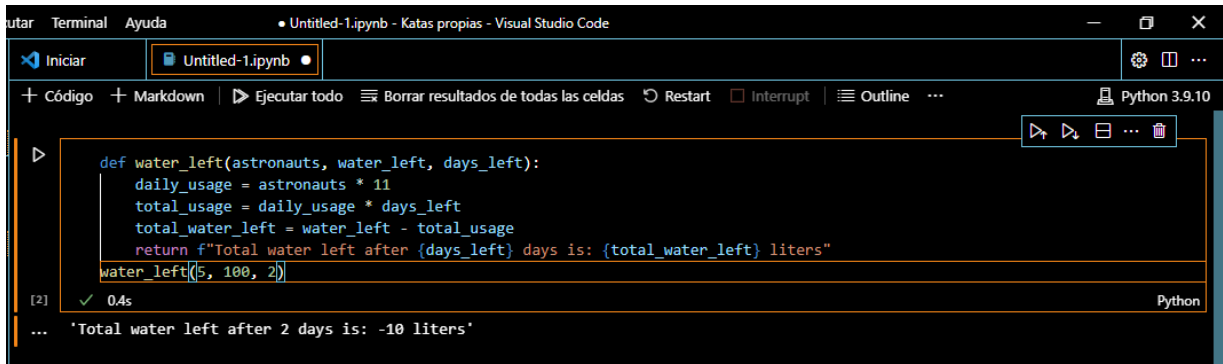
Observamos que, si funciona nuestro control de excepciones, y por ultimo solo añadiremos una excepción mas en caso de encontrar el archivo y no tener los permisos necesarios



```
config.txt > 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")
8     except (BlockingIOError, TimeoutError):
9         print("Filesystem under heavy load, can't complete reading configuration file")
10
11
12 if __name__ == '__main__':
    main()
```

Parte 2

KATA 10



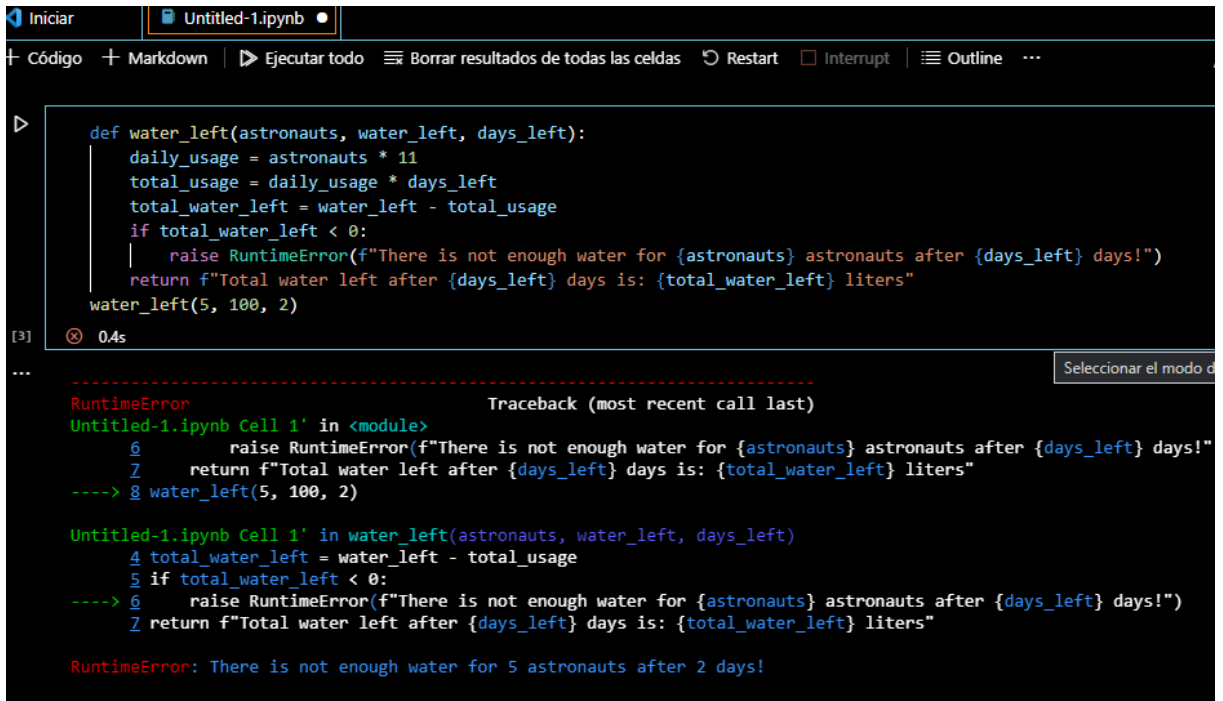
The screenshot shows a Jupyter Notebook cell in Visual Studio Code. The cell contains a Python function `water_left` and a call to it. The function calculates the total water usage for a given number of astronauts and days, and returns the remaining water. The call to the function is `water_left(5, 100, 2)`. The output of the cell is `'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
    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'



The screenshot shows a Jupyter Notebook cell in Visual Studio Code. The cell contains a Python function `water_left` and a call to it. The function calculates the total water usage for a given number of astronauts and days, and returns the remaining water. If the remaining water is less than or equal to zero, it raises a `RuntimeError` with the message "There is not enough water for {astronauts} astronauts after {days_left} days!". The call to the function is `water_left(5, 100, 2)`. The output of the cell is a `RuntimeError` with the message "There is not enough water for 5 astronauts after 2 days!".

```
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 de

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'
```