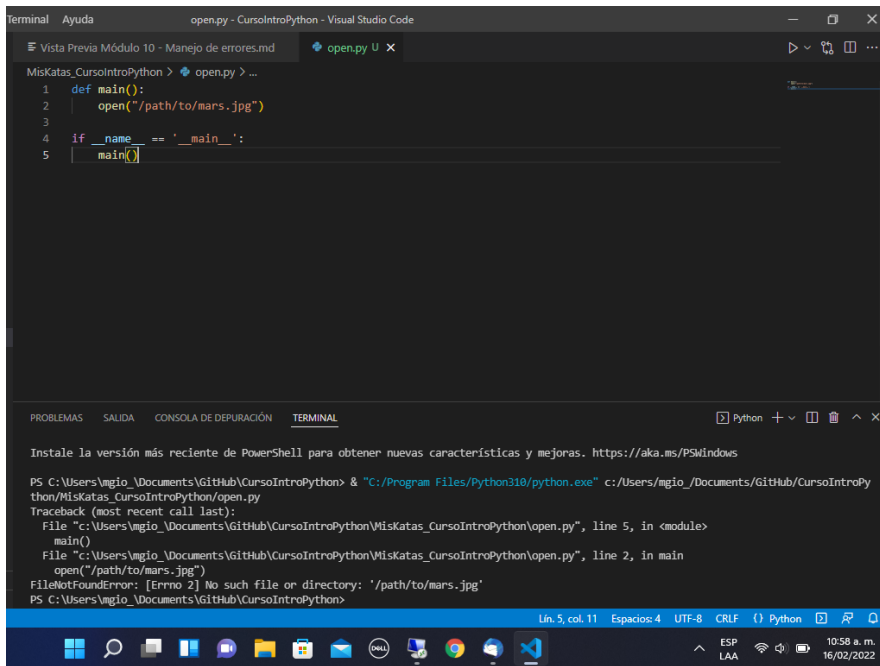


Prueba con error de archivo no encontrado.



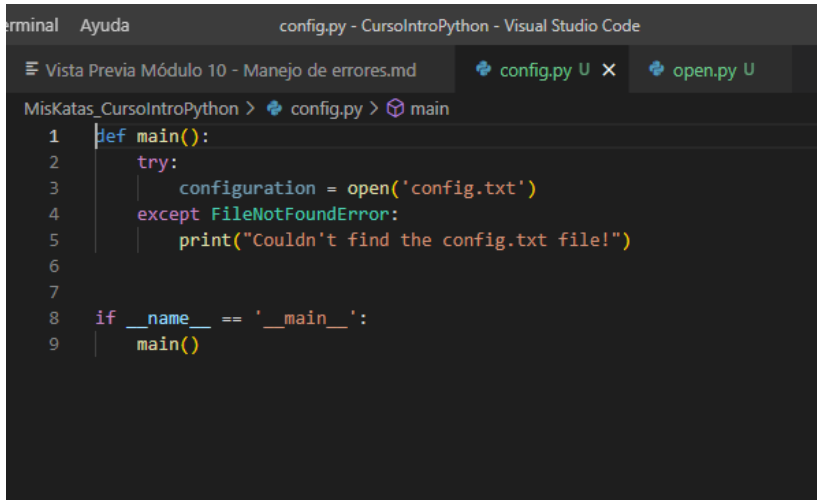
The screenshot shows the Visual Studio Code interface with a file named `open.py` open. The code in the editor is as follows:

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

The terminal at the bottom shows the command to run the script and the resulting error:

```
PS C:\Users\mgio_\Documents\GitHub\CursoIntroPython> & "C:/Program Files/Python310/python.exe" c:/Users/mgio_/Documents/GitHub/CursoIntroPython/MisKatas_CursoIntroPython/open.py
Traceback (most recent call last):
  File "c:/Users/mgio_/Documents/GitHub/CursoIntroPython/MisKatas_CursoIntroPython/open.py", line 5, in <module>
    main()
  File "c:/Users/mgio_/Documents/GitHub/CursoIntroPython/MisKatas_CursoIntroPython/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\mgio_\Documents\GitHub\CursoIntroPython>
```

Manejo de error de archivo no encontrado.



The screenshot shows the Visual Studio Code interface with a file named `config.py` open. The code in the editor is as follows:

```
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()
```

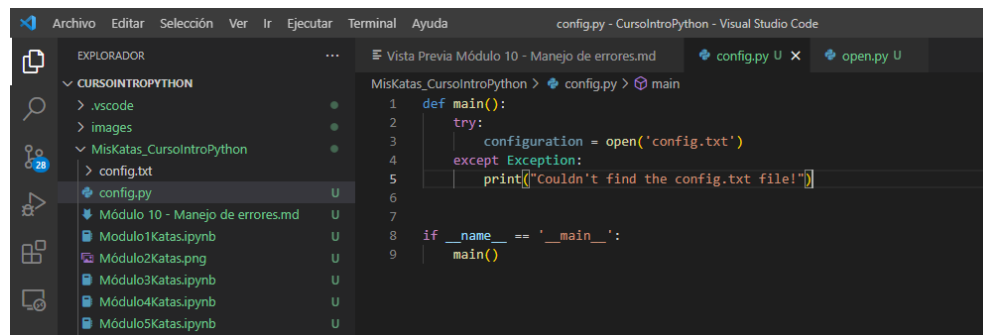
Creamos una carpeta llamada config.txt y ejecutamos el código

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.22000.493]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>python3 config.py
Traceback (most recent call last):
  File "C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython\config.py", line 9, in <module>
    main()
  File "C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython\config.py", line 3, in main
    configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>
```

Se modifica el código para manejo de excepciones



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the project structure with a folder named 'CURSOINTROPYTHON' containing files like '.vscode', 'images', 'MisKatas_CursoIntroPython', 'config.txt', 'config.py', and several 'Módulo' files. The main editor window displays the 'config.py' file with the following code:

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

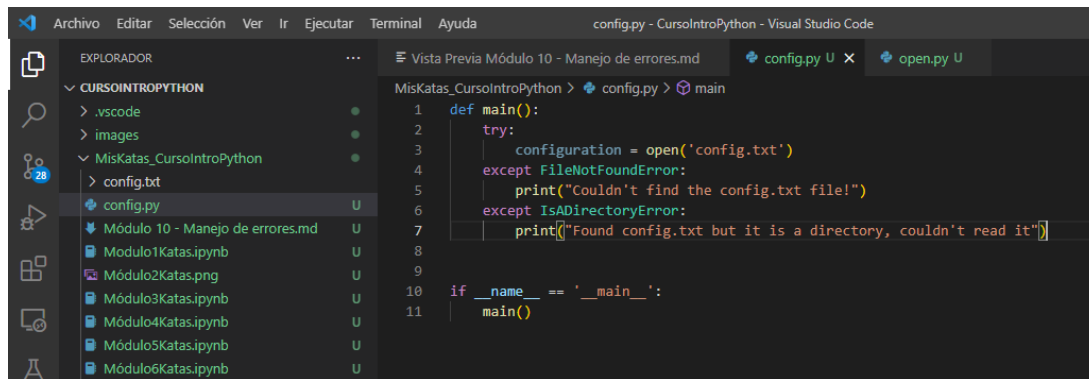
Nos muestra este mensaje

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.22000.493]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>python3 config.py
Couldn't find the config.txt file!

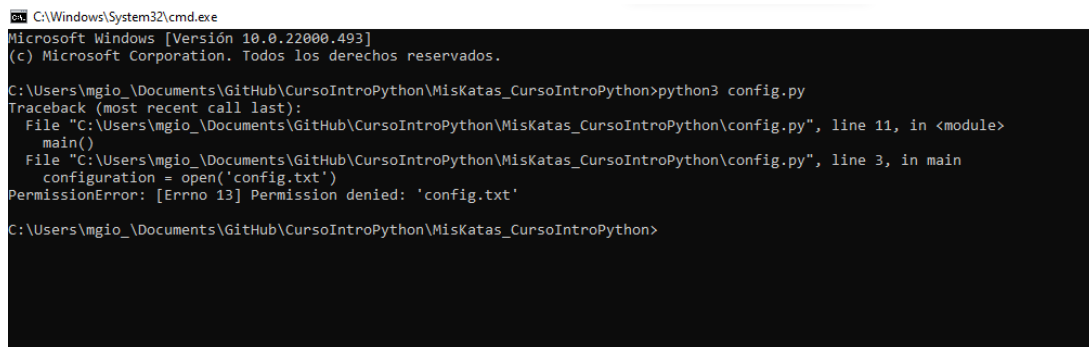
C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>_
```

Modificamos el código para manejar distintas excepciones



```
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__':
11     main()
```

Muestra el siguiente resultado

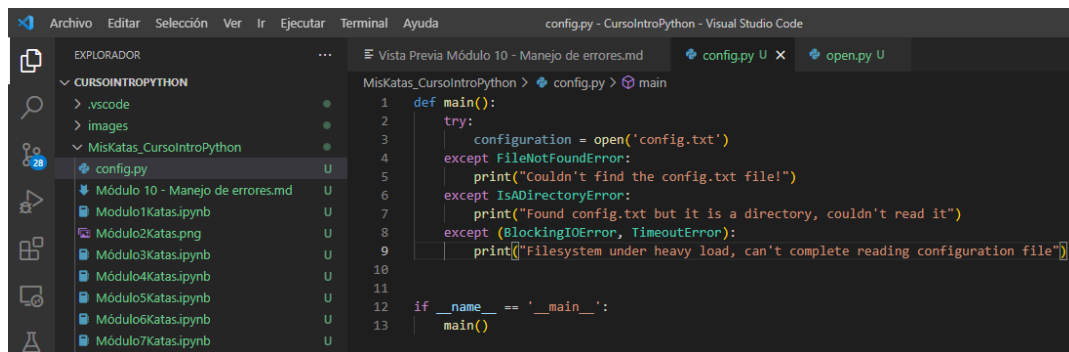


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.22000.493]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>python3 config.py
Traceback (most recent call last):
  File "C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython\config.py", line 11, in <module>
    main()
  File "C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython\config.py", line 3, in main
    configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>
```

Modificamos el manejo de distintas excepciones



```
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__':
13     main()
```

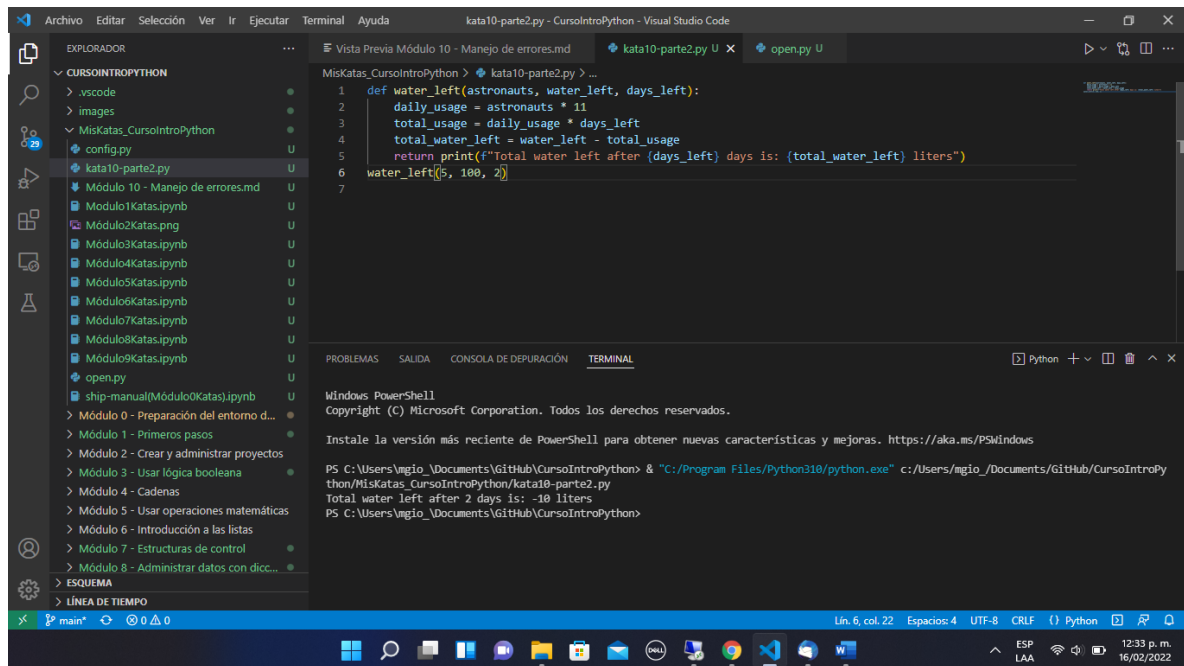
Muestra el siguiente resultado

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.22000.493]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>python3 config.py
Couldn't find the config.txt file!

C:\Users\mgio_\Documents\GitHub\CursoIntroPython\MisKatas_CursoIntroPython>_
```

Ejecución de programa con resultado negativo



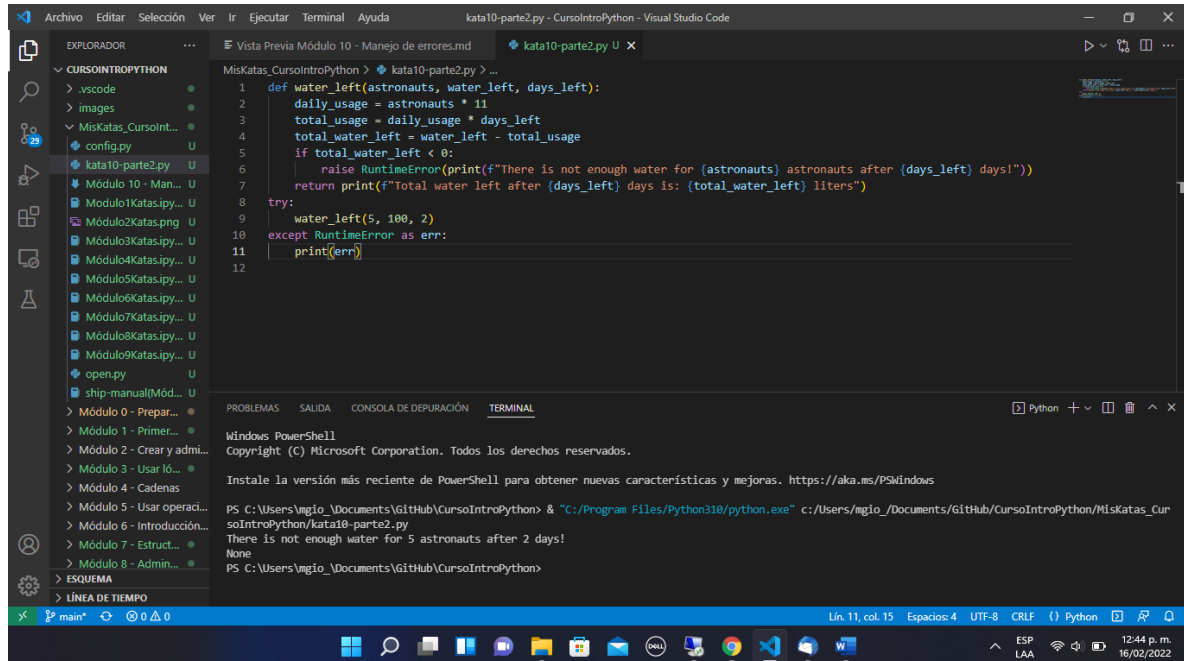
The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure for 'CURSORINTROPYTHON'. The main editor window shows the file 'kata10-parte2.py' with the following Python code:

```
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 print(f"Total water left after (days_left) days is: {total_water_left} liters")
6
7 water_left([5, 100, 2])
```

The terminal at the bottom shows the execution of the script in a Windows PowerShell environment. The output is:

```
PS C:\Users\mgio_\Documents\Github\CursoIntroPython> & "C:/Program Files/Python310/python.exe" c:/Users/mgio_\Documents/Github/CursoIntroPython/Miskatas_CursoIntroPython/kata10-parte2.py
Total water left after 2 days is: -10 liters
PS C:\Users\mgio_\Documents\Github\CursoIntroPython>
```

Ejecución con control de error por número negativo



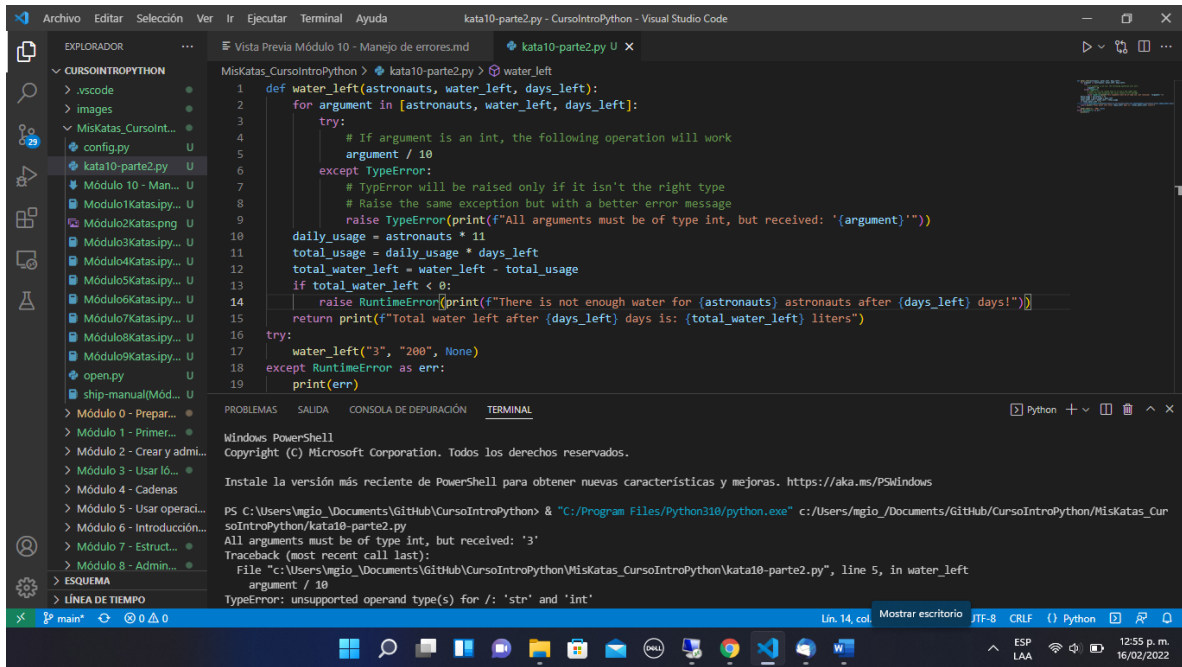
The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure for 'CURSORINTROPYTHON'. The main editor window shows the file 'kata10-parte2.py' with the following Python code:

```
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(print(f"There is not enough water for {astronauts} astronauts after (days_left) days!"))
7     return print(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)
```

The terminal at the bottom shows the execution of the script in a Windows PowerShell environment. The output is:

```
PS C:\Users\mgio_\Documents\Github\CursoIntroPython> & "C:/Program Files/Python310/python.exe" c:/Users/mgio_\Documents/Github/CursoIntroPython/Miskatas_CursoIntroPython/kata10-parte2.py
There is not enough water for 5 astronauts after 2 days!
None
PS C:\Users\mgio_\Documents\Github\CursoIntroPython>
```

Ejecución con control de error de tipo de dato



The screenshot shows the Visual Studio Code interface with a Python file named `kata10-parte2.py` open. The file contains a function `water_left` that calculates the remaining water for astronauts. The function uses a try-except block to handle a `TypeError` when the `days_left` argument is not an integer. The terminal window shows the error message: `TypeError: unsupported operand type(s) for /: 'str' and 'int'`.

```
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(print(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(print(f"There is not enough water for {astronauts} astronauts after {days_left} days!"))
15    return print(f"Total water left after {days_left} days is: {total_water_left} liters")
16
17 try:
18     water_left("3", "200", None)
19 except RuntimeError as err:
20     print(err)
```

Terminal output:

```
PS C:\Users\mgio_\Documents\GitHub\CursoIntroPython> & "C:/Program Files/Python310/python.exe" c:/Users/mgio_/Documents/GitHub/CursoIntroPython/MisKatas_CursoIntroPython/kata10-parte2.py
All arguments must be of type int, but received: '3'
Traceback (most recent call last):
  File "c:/Users/mgio_/Documents/GitHub/CursoIntroPython/MisKatas_CursoIntroPython/kata10-parte2.py", line 5, in water_left
    argument / 10
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```