

# EJERCICIOS PARA LA KATA 10

- **Tracebacks**

Intentando abrir un archivo inexistente:

```
miguel@DESKTOP-RI8016I: ~$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> open("/path/to/mars.jpg")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
>>>
```

Haciéndolo desde el archivo open.py

```
(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$ "/home/miguel/Archivos/Launch X/env/bin/python" "/home/miguel/Archivos/Launch X/open.py"
Traceback (most recent call last):
  File "/home/miguel/Archivos/Launch X/open.py", line 5, in <module>
    main()
  File "/home/miguel/Archivos/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'
(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$
```

- **Controlando las excepciones**

- Try y Except de los bloques

Controlando la excepción con un bloque try y except:

```
miguel@DESKTOP-RI8016I: ~$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> try:
...     open('config.txt')
... except FileNotFoundError:
...     print("Couldn't find the config.txt file!")
...
Couldn't find the config.txt file!
>>>
```

Haciéndolo desde el archivo config.py

```
Módulo 10 - Manejo de errores.md  config.py x
Launch X > 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
8 if __name__ == '__main__':
9     main()

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$ "/home/miguel/Archivos/Launch X/env/bin/python" "/home/miguel/Archivos/Launch X/config.py"
Couldn't find the config.txt file!
(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$
```

```
config.py x Módulo 10 - Manejo de errores.md
Launch X > 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
10 if __name__ == '__main__':
11     main()

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$ "/home/miguel/Archivos/Launch X/env/bin/python" "/home/miguel/Archivos/Launch X/config.py"
Couldn't find the config.txt file!
(env) miguel@DESKTOP-RI8016I:~/Archivos/Launch X$
```

- Generación de excepciones

Probando la primera funcion con cinco astronautas, 100 litros de agua sobrante y dos días:

```
miguel@DESKTOP-RI8016k: ~  
miguel@DESKTOP-RI8016k:~$ python3  
Python 3.8.10 (default, Nov 26 2021, 20:14:08)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 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)  
'Total water left after 2 days is: -10 liters'  
>>>
```

Con la modificación:

```
miguel@DESKTOP-RI8016k: ~  
miguel@DESKTOP-RI8016k:~$ python3  
Python 3.8.10 (default, Nov 26 2021, 20:14:08)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 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)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
  File "<stdin>", line 6, in water_left  
RuntimeError: There is not enough water for 5 astronauts after 2 days!  
>>>
```

Con un error mejor:

```
miguel@DESKTOP-RI8016k: ~  
miguel@DESKTOP-RI8016k:~$ python3  
Python 3.8.10 (default, Nov 26 2021, 20:14:08)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> def water_left(astronauts, water_left, days_left):  
...     for argument in [astronauts, water_left, days_left]:  
...         try:  
...             # If argument is an int, the following operation will work  
...             argument / 10  
...         except TypeError:  
...             # 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(5, 100, 2)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
  File "<stdin>", line 14, in water_left  
RuntimeError: There is not enough water for 5 astronauts after 2 days!  
>>>
```