

# Kata del Módulo 10

## Contenido

Kata del Módulo 10 .....	1
Rastreos .....	1
Archivo open.py .....	2
Try y Except de los bloques .....	2
Archivo config.py .....	3
BlockingIOErrory TimeOutErrorjuntos: .....	4
Except as .....	5
Except as err .....	5
Generación de excepciones .....	5
RuntimeError .....	6
TypeError .....	7

## Rastreos

```
[1] open("/path/to/mars.jpg")
    0.4s Python

...
-----
FileNotFoundError                                Traceback (most recent call last)
Untitled-1.ipynb Cell 1' in <module>
----> 1 open("/path/to/mars.jpg")

FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
```

## Archivo open.py

```
Untitled-1.ipynb • config.txt • config.py • open.py •
```

```
C: > Users > Anny Rios > Documents > CursoLaunch > open.py > ...  
1  from distutils.command.config import config  
2  
3  
4  def main():  
5      open("/path/to/mars.jpg")  
6  
7  
8  if __name__ == '__main__':  
9      main()  
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python + - [ ] [X] ^ X


```
PS C:\Users\Anny Rios> & "C:/Users/Anny Rios/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Anny Rios/Documents/CursoLaunch/open.py"  
c:\Users\Anny Rios\Documents\CursoLaunch\open.py:1: DeprecationWarning: The distutils package  
is deprecated and slated for removal in Python 3.12. Use setuptools or check PEP 632 for poten  
tial alternatives  
  from distutils.command.config import config  
Traceback (most recent call last):  
  File "c:\Users\Anny Rios\Documents\CursoLaunch\open.py", line 9, in <module>  
    main()  
  File "c:\Users\Anny Rios\Documents\CursoLaunch\open.py", line 5, in main  
    open("/path/to/mars.jpg")  
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'  
PS C:\Users\Anny Rios>
```

## Try y Except de los bloques

```
▷ ▾  
• ▾ try:  
    | open('config.txt')  
    | except FileNotFoundError:  
    |     print("Couldn't find the config.txt file!")  
[13] ✓ 0.6s Python
```

... Couldn't find the config.txt file!

## Archivo 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()
```

```
PS C:\Users\Anny Rios> & "C:/Users/Anny Rios/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Anny Rios/Documents/CursoLaunch/config.py"
Couldn't find the config.txt file!
```

```

12 def main():
13     try:
14         configuration = open('config.txt')
15     except Exception:
16         print("Couldn't find the config.txt file!")
17
18
19 if __name__ == '__main__':
20     main()
22 def main():
23     try:
24         configuration = open('config.txt')
25     except FileNotFoundError:
26         print("Couldn't find the config.txt file!")
27     except IsADirectoryError:
28         print("Found config.txt but it is a directory, couldn't read it")
29
30
31 if __name__ == '__main__':
32     main()
PS C:\Users\Anny Rios> & "C:/Users/Anny Rios/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Anny Rios/Documents/CursoLaunch/config.py"
Couldn't find the config.txt file!
Couldn't find the config.txt file!
Couldn't find the config.txt file!
PS C:\Users\Anny Rios>

```

10.2 64-bit 0 0 Ln 46, Col 1 Spaces: 4 UTF-8 CRLF Python

## BlockingIOError TimeoutError juntos:

```

34 def main():
35     try:
36         configuration = open('config.txt')
37     except FileNotFoundError:
38         print("Couldn't find the config.txt file!")
39     except IsADirectoryError:
40         print("Found config.txt but it is a directory, couldn't read it")
41     except (BlockingIOError, TimeoutError):
42         print("Filesystem under heavy load, can't complete reading config")
43
44 if __name__ == '__main__':
45     main()
46

```

## Except as

```
18] try:
    open("mars.jpg")
except FileNotFoundError as err:
    print("got a problem trying to read the file:", err)

... got a problem trying to read the file: [Errno 2] No such file or directory:
'mars.jpg'
```

## Except as err

```
[19] try:
    open("config.txt")
except OSError as err:
    if err.errno == 2:
        print("Couldn't find the config.txt file!")
    elif err.errno == 13:
        print("Found config.txt but couldn't read it")

... Couldn't find the config.txt file!
```

[+ Code](#) [+ Markdown](#)

## Generación de excepciones

```
1] 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"
```

```
2] water_left(5,100,2)

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

```
3] 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"
```

## RuntimeError

```
water_left(5,100,2)

[4] ✖ 0.5s

...

-----
RuntimeError                                Traceback (most recent call last)
Untitled-1.ipynb Cell 8' in <module>
----> 1 water_left(5,100,2)

Untitled-1.ipynb Cell 7' 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!
```

```
▷ ▾
    try:
    |     water_left(5, 100, 2)
    | except RuntimeError as err:
    |     (err)

[9] ✔ 0.1s
```

# TypeError

```
water_left("3","200",None)
[6] 0.1s

...
-----
TypeError                                Traceback (most recent call last)
Untitled-1.ipynb Cell 10' in <module>
----> 1 water_left("3","200",None)

Untitled-1.ipynb Cell 7' 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:
            # 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"
[7] 0.8s
```

```
water_left("3","200",None)
[8] 0.1s

...
-----
TypeError                                Traceback (most recent call last)
Untitled-1.ipynb Cell 11' 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)
Untitled-1.ipynb Cell 12' in <module>
----> 1 water_left("3","200",None)

Untitled-1.ipynb Cell 11' 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'
```