


Tracebacks

D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 >  open.py > ...

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL


PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> **python** open.py

Traceback (most recent call last):

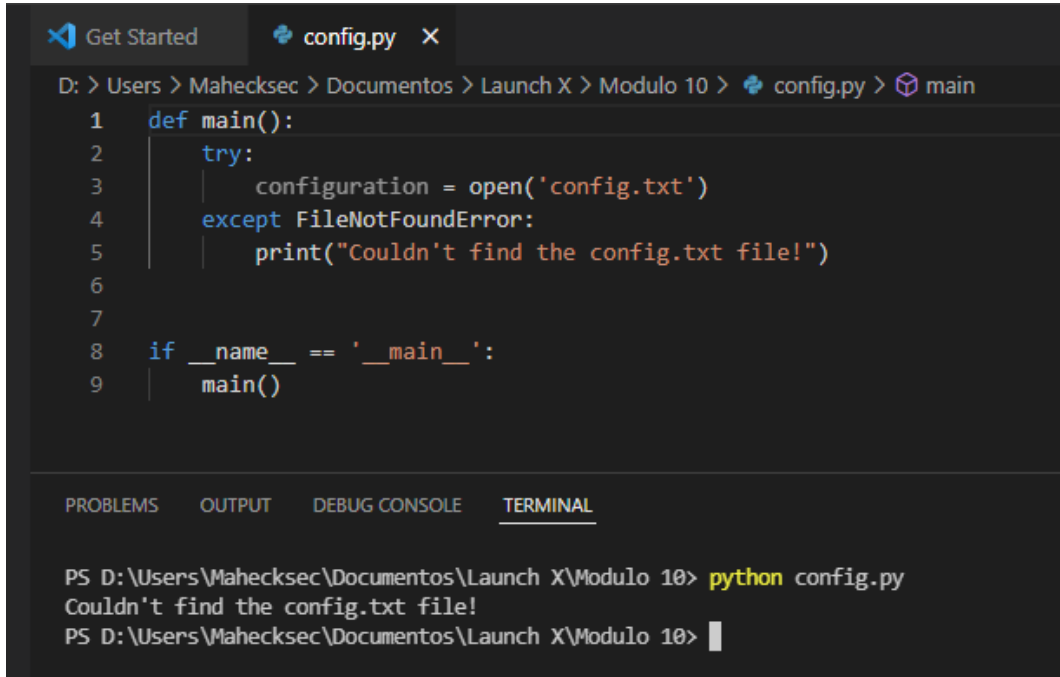
File "open.py", line 5, in <module>
 main()

File "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:\users\Mahecksec\Documentos\Launch X\Modulo 10> 

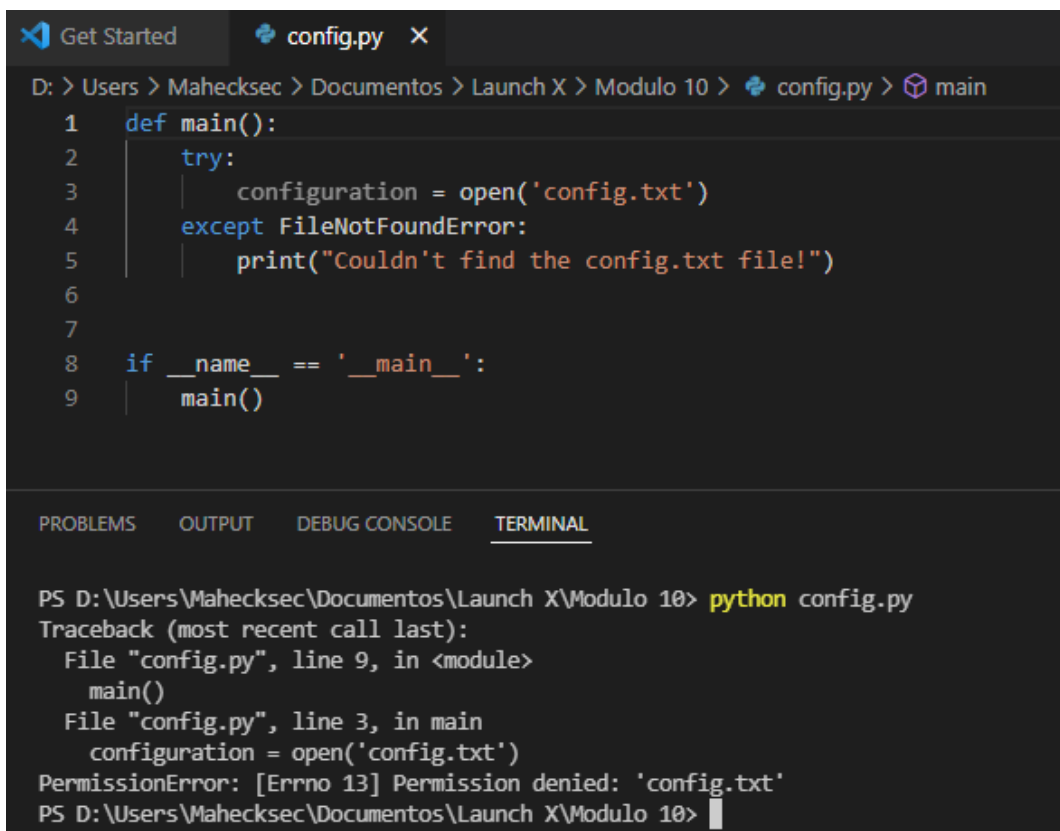
Controlando las excepciones



```
Get Started config.py X
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > 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
7
8 if __name__ == '__main__':
9     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python config.py
Couldn't find the config.txt file!
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> 
```

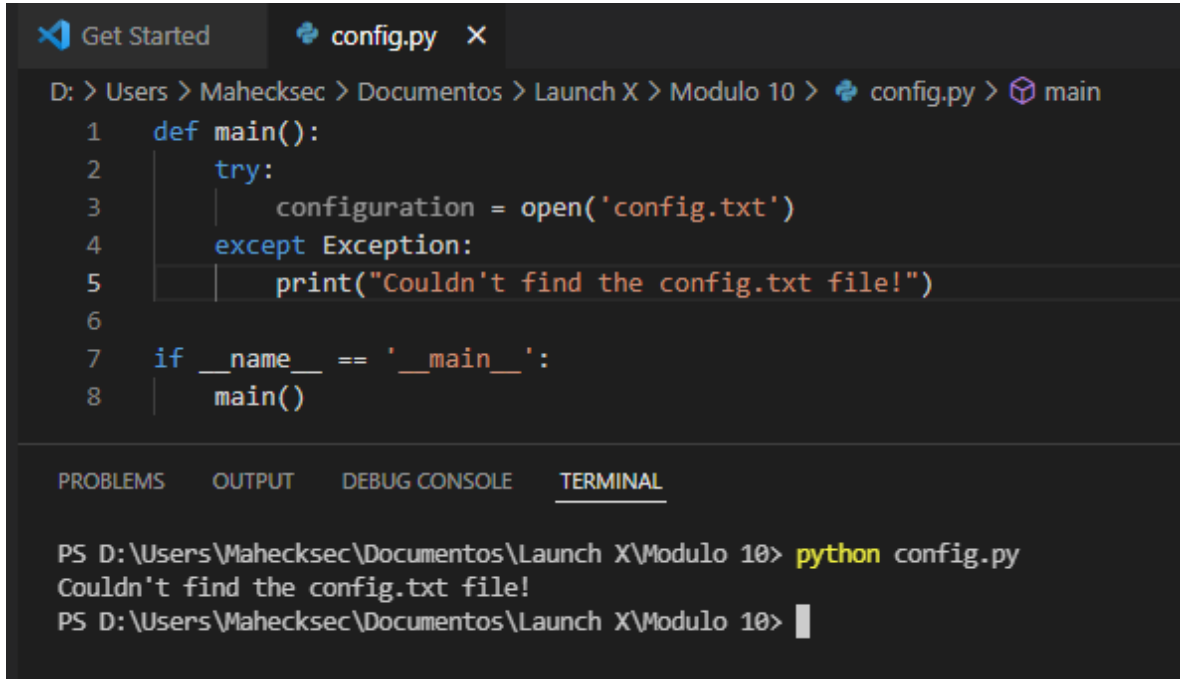
Creando un directorio 'config.txt'



```
Get Started config.py X
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > 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
7
8 if __name__ == '__main__':
9     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python 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')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> 
```

Tratar de detectar todas las excepciones



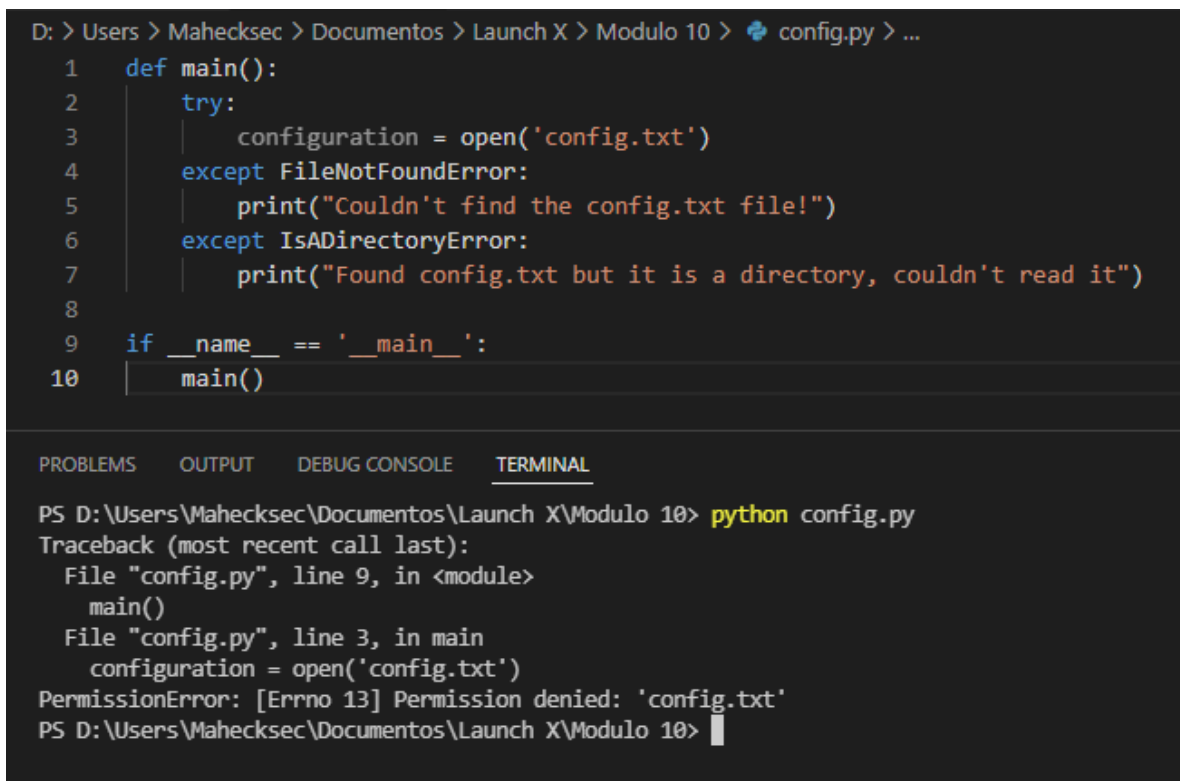
The screenshot shows a VS Code editor with a file named `config.py` open. The file path in the breadcrumb is `D:\Users\Mahecksec\Documentos\Launch X\Modulo 10\config.py`. The code in the editor is as follows:

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

Below the editor, the **TERMINAL** tab is active, showing the command `python config.py` and its output:

```
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python config.py
Couldn't find the config.txt file!
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10>
```

Abordar todas las excepciones



The screenshot shows the same VS Code editor with a modified `config.py` file. The code is:

```
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 if __name__ == '__main__':
10     main()
```

The **TERMINAL** tab shows the command `python config.py` and a detailed traceback for a `PermissionError`:

```
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python 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')
PermissionError: [Errno 13] Permission denied: 'config.txt'
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10>
```

Eliminando el directorio 'config.txt'

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > 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 if __name__ == '__main__':
10     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python config.py
Couldn't find the config.txt file!
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> |
```

Agrupar excepciones

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > 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 if __name__ == '__main__':
12     main()
```

Acceder al error asociado a la excepción

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > config.py > ...
1 def main():
2     try:
3         open("mars.jpg")
4     except FileNotFoundError as err:
5         print("got a problem trying to read the file:", err)
6
7 if __name__ == '__main__':
8     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python config.py
got a problem trying to read the file: [Errno 2] No such file or directory: 'mars.jpg'
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> |
```

Acceder a los atributos de error

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > config.py > main
1  def main():
2      try:
3          open("config.txt")
4      except OSError as err:
5          if err.errno == 2:
6              print("Couldn't find the config.txt file!")
7          elif err.errno == 13:
8              print("Found config.txt but couldn't read it")
9
10 if __name__ == '__main__':
11     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> python config.py
Found config.txt but couldn't read it
PS D:\Users\Mahecksec\Documentos\Launch X\Modulo 10> █
```

Generación de excepciones

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > excep.py > ...
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(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
9
10 try:
11     water_left(5, 100, 2)
12 except RuntimeError as err:
13     print(err)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> python excep.py
There is not enough water for 5 astronauts after 2 days!
PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> █
```

Error de tipo

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > excepy.py > ...
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(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
9
10 try:
11     water_left("5", "100", None)
12 except RuntimeError as err:
13     print(err)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> python excepy.py
Traceback (most recent call last):
  File "excepy.py", line 11, in <module>
    water_left("5", "100", None)
  File "excepy.py", line 3, in water_left
    total_usage = daily_usage * days_left
TypeError: can't multiply sequence by non-int of type 'NoneType'
PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> █
```

```
D: > Users > Mahecksec > Documentos > Launch X > Modulo 10 > excepy.py > ...
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(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(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
15    return f"Total water left after {days_left} days is: {total_water_left} liters"
16
17 water_left("3", "200", None)
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
File "excepy.py", line 5, in water_left
    argument / 10
TypeError: unsupported operand type(s) for /: 'str' and 'int'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "excepy.py", line 17, in <module>
    water_left("3", "200", None)
  File "excepy.py", line 9, in water_left
    raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
TypeError: All arguments must be of type int, but received: '3'
PS D:\users\Mahecksec\Documentos\Launch X\Modulo 10> █
```