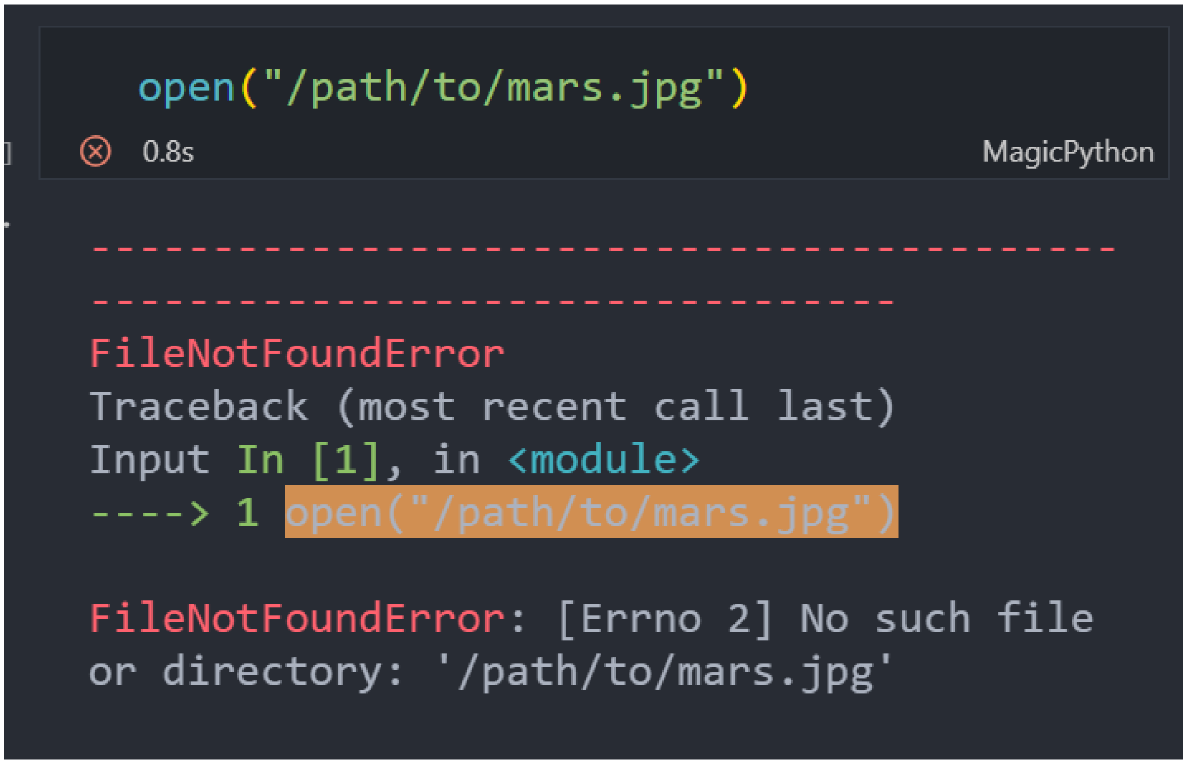
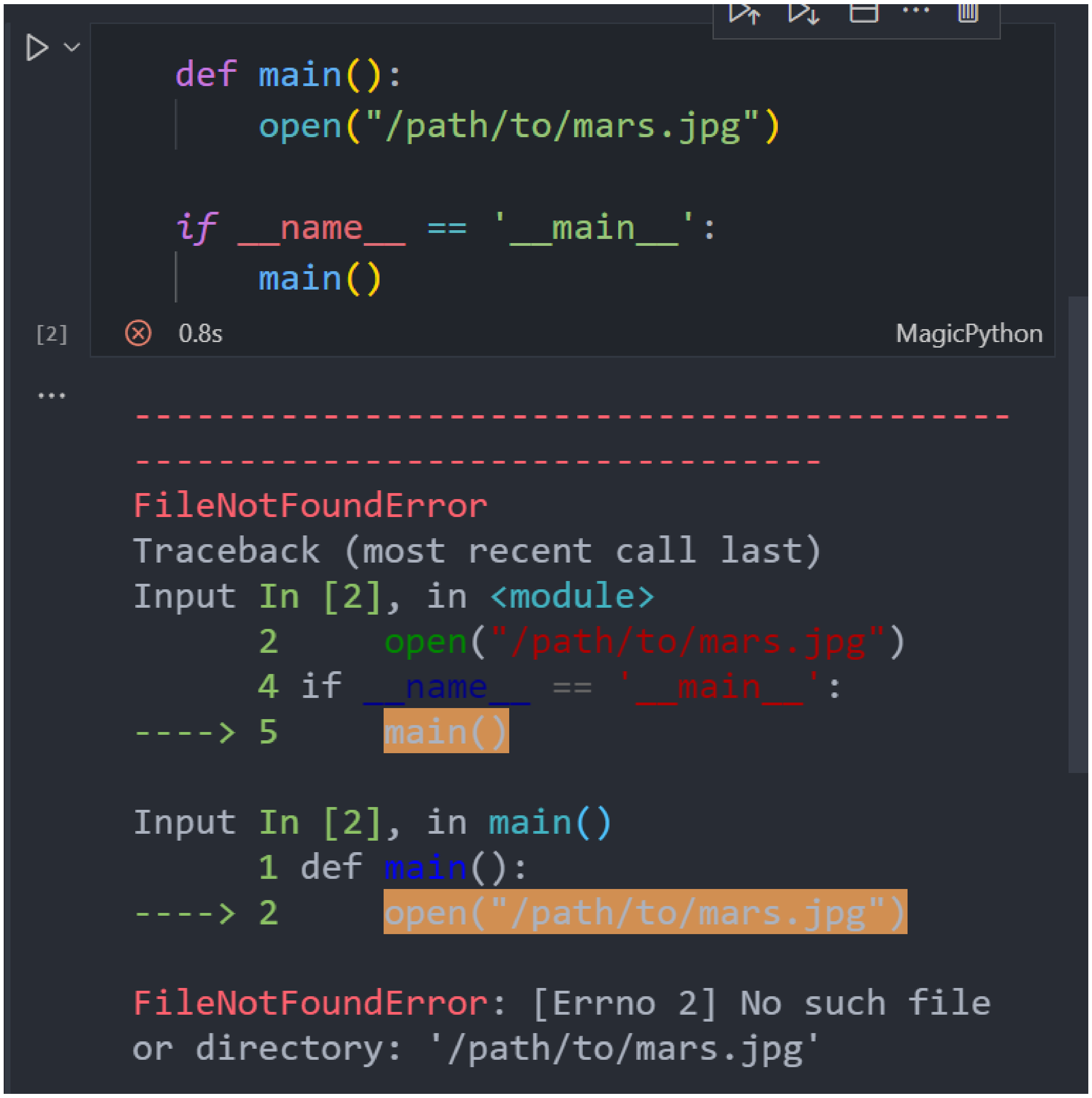


```
open("/path/to/mars.jpg")
```

0.8s MagicPython

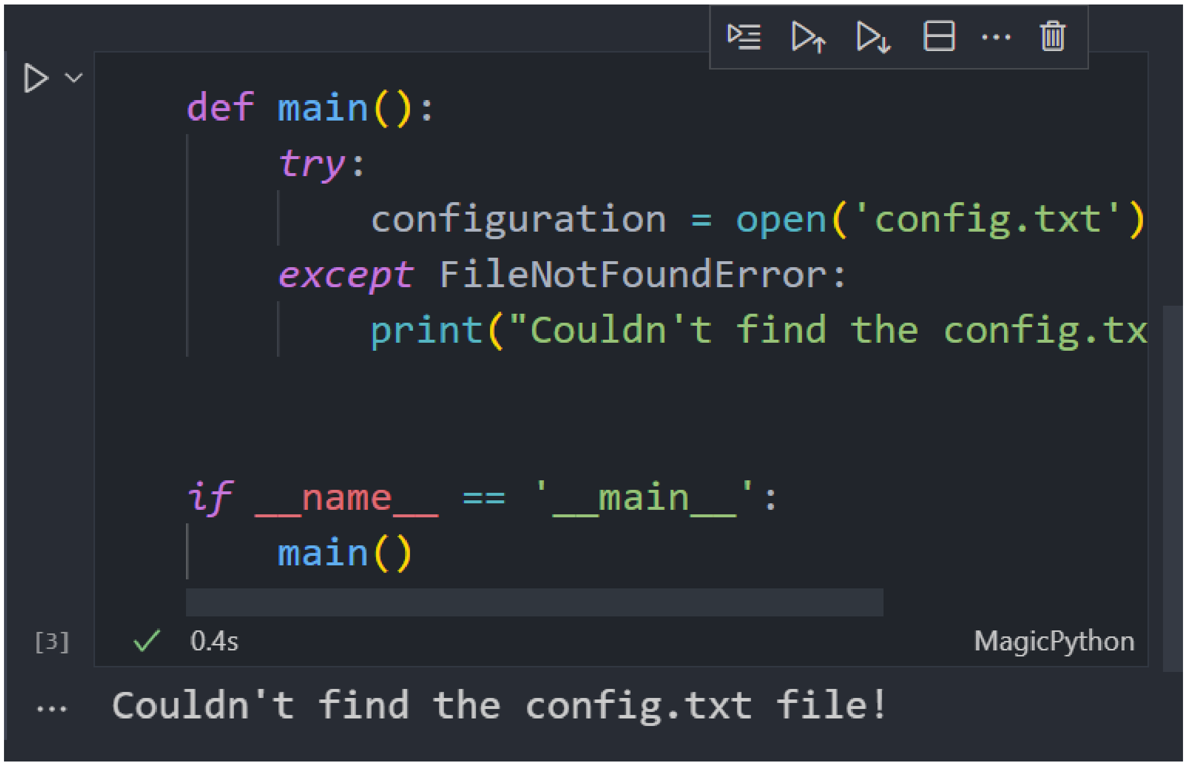
```
---------------------------------------------------------------------------
FileNotFoundError
Traceback (most recent call last)
Input In [1], in <module>
----> 1 open("/path/to/mars.jpg")

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



```
def main():
    open("/path/to/mars.jpg")

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

[2] 0.8s MagicPython

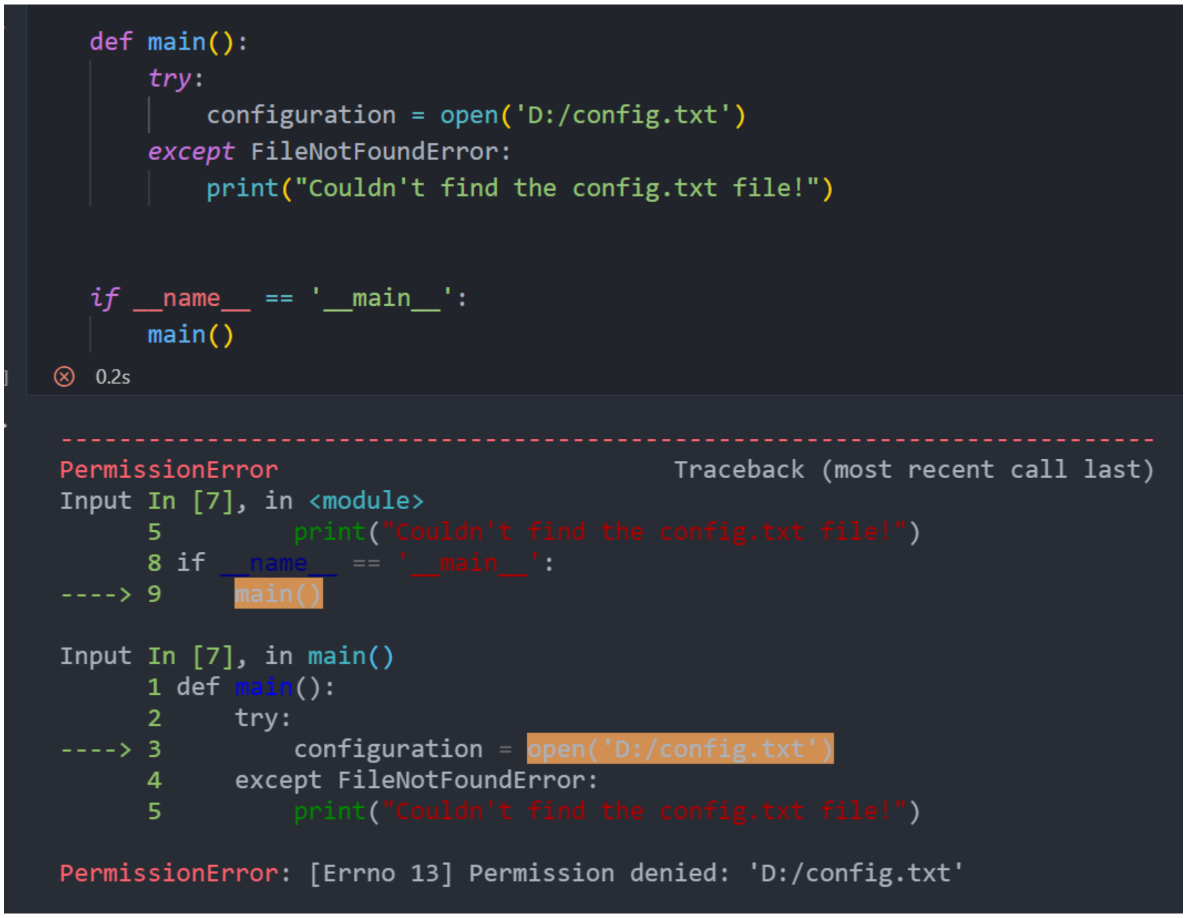
```
---------------------------------------------------------------------------
FileNotFoundError
Traceback (most recent call last)
Input In [2], in <module>
      2     open("/path/to/mars.jpg")
      4 if __name__ == '__main__':
----> 5     main()

Input In [2], in main()
      1 def main():
----> 2     open("/path/to/mars.jpg")

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



```
def main():
    try:
        configuration = open('config.txt')
    except FileNotFoundError:
        print("Couldn't find the config.tx


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

[3] ✓ 0.4s MagicPython

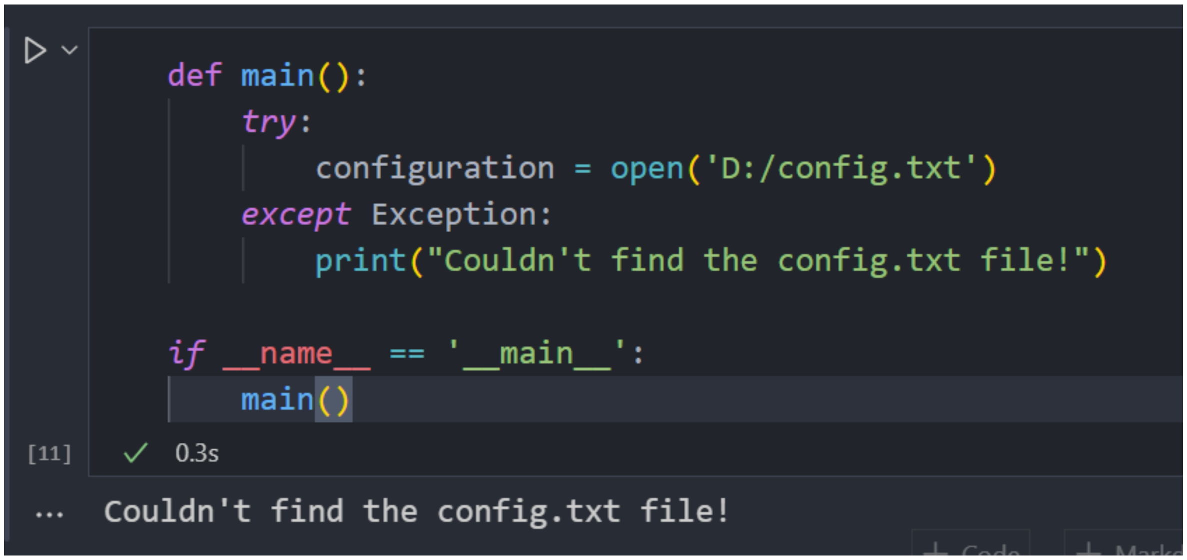
```
Couldn't find the config.txt file!
```



```
def main():
    try:
        configuration = open('D:/config.txt')
    except FileNotFoundError:
        print("Couldn't find the config.txt file!")


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

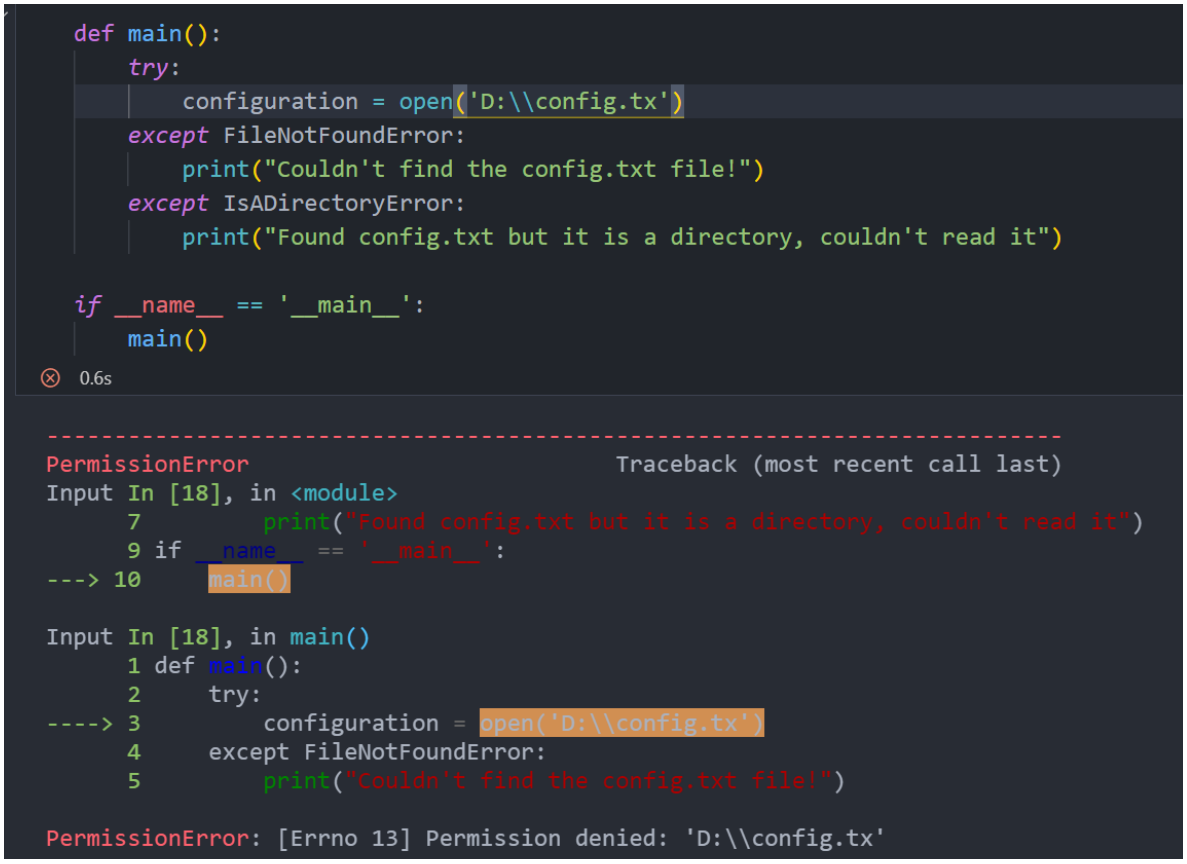
0.2s

```
---------------------------------------------------------------------------
PermissionError                           Traceback (most recent call last)
Input In [7], in <module>
      5         print("Couldn't find the config.txt file!")
      8 if __name__ == '__main__':
----> 9     main()

Input In [7], in main()
      1 def main():
      2     try:
----> 3         configuration = open('D:/config.txt')
      4     except FileNotFoundError:
      5         print("Couldn't find the config.txt file!")

PermissionError: [Errno 13] Permission denied: 'D:/config.txt'
```



```
def main():
    try:
        configuration = open('D:/config.txt')
    except Exception:
        print("Couldn't find the config.txt file!")

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

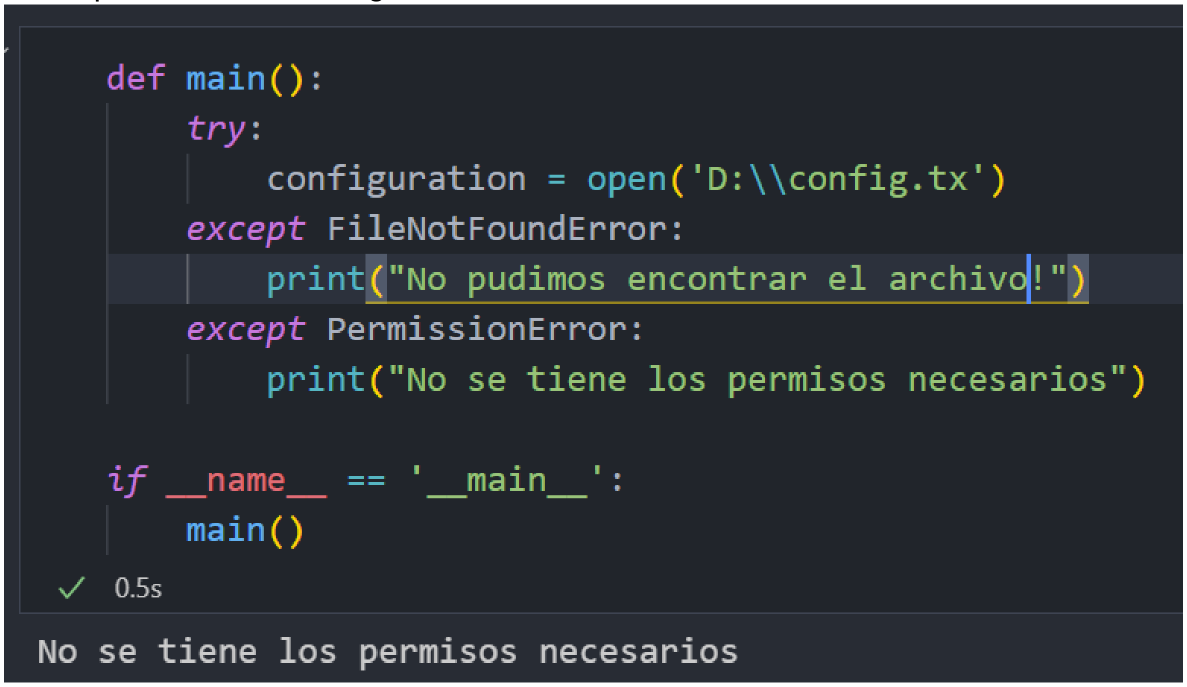
```
[11]  ✓ 0.3s
...  Couldn't find the config.txt file!
```



```
def main():
    try:
        configuration = open('D:\\config.tx')
    except FileNotFoundError:
        print("Couldn't find the config.txt file!")
    except IsADirectoryError:
        print("Found config.txt but it is a directory, couldn't read it")

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

```
⊗ 0.6s

---------------------------------------------------------------------------
PermissionError                           Traceback (most recent call last)
Input In [18], in <module>
      7         print("Found config.txt but it is a directory, couldn't read it")
      9 if __name__ == '__main__':
---> 10     main()

Input In [18], in main()
      1 def main():
      2     try:
----> 3         configuration = open('D:\\config.tx')
      4     except FileNotFoundError:
      5         print("Couldn't find the config.txt file!")

PermissionError: [Errno 13] Permission denied: 'D:\\config.tx'
```

La carpeta se llama ‘config.tx’

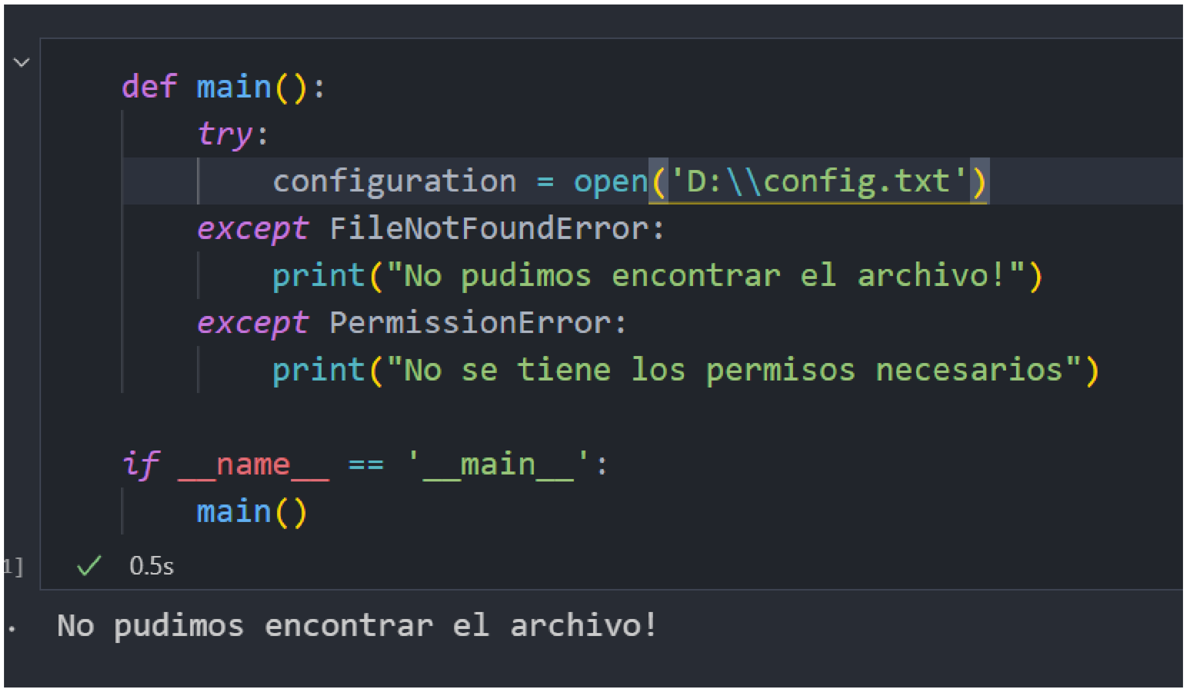


```python
def main():
    try:
        configuration = open('D:\\config.tx')
    except FileNotFoundError:
        print("No pudimos encontrar el archivo!")
    except PermissionError:
        print("No se tiene los permisos necesarios")

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

✓ 0.5s

```
No se tiene los permisos necesarios
```

Ahora busco ‘config.txt’ el cual no existe

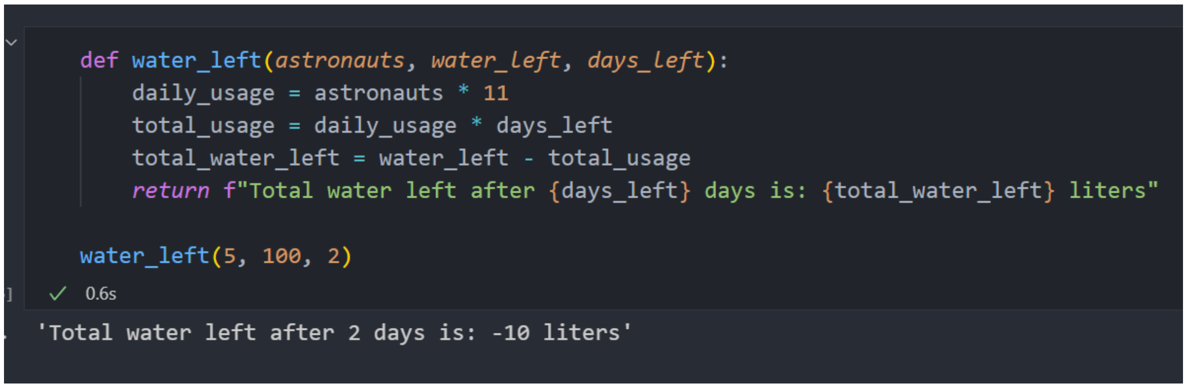


```python
def main():
    try:
        configuration = open('D:\\config.txt')
    except FileNotFoundError:
        print("No pudimos encontrar el archivo!")
    except PermissionError:
        print("No se tiene los permisos necesarios")

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

✓ 0.5s

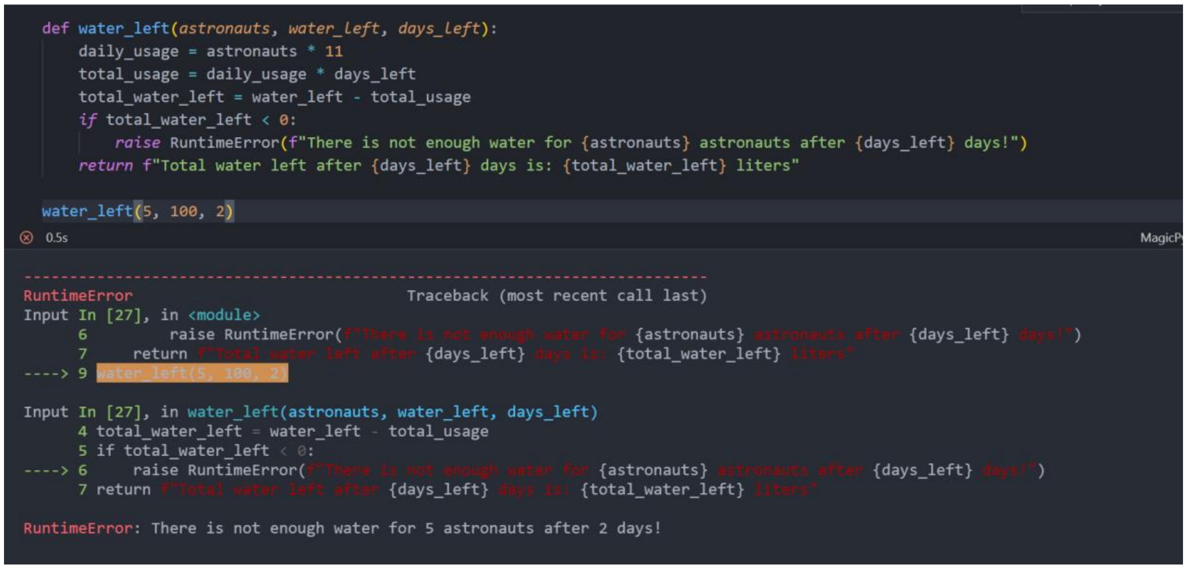
```
No pudimos encontrar el archivo!
```

Los astronautas



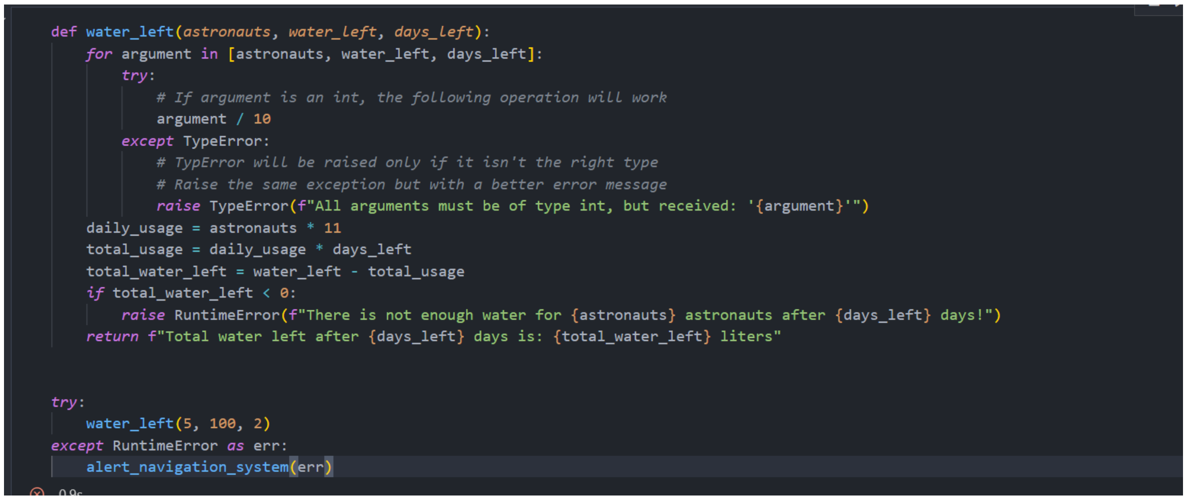
```python
def water_left(astronauts, water_left, days_left):
    daily_usage = astronauts * 11
    total_usage = daily_usage * days_left
    total_water_left = water_left - total_usage
    return f"Total water left after {days_left} days is: {total_water_left} liters"

water_left(5, 100, 2)
```

```
✓ 0.6s
'Total water left after 2 days is: -10 liters'
```

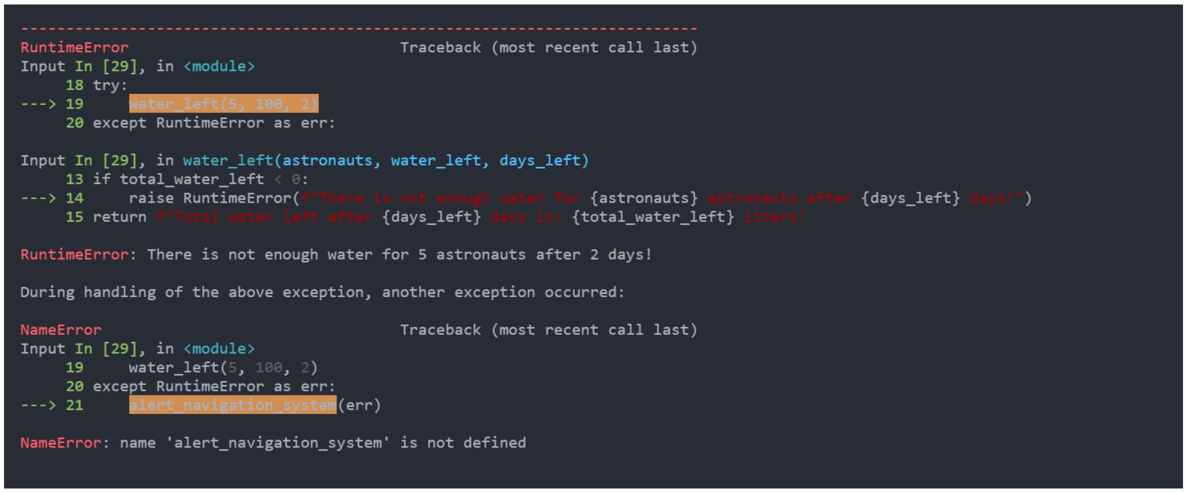


```python
def water_left(astronauts, water_left, days_left):
    daily_usage = astronauts * 11
    total_usage = daily_usage * days_left
    total_water_left = water_left - total_usage
    if total_water_left < 0:
        raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
    return f"Total water left after {days_left} days is: {total_water_left} liters"

water_left(5, 100, 2)
```

```
0.5s
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [27], 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"
----> 9 water_left(5, 100, 2)

Input In [27], 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!
```



```python
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"


try:
    water_left(5, 100, 2)
except RuntimeError as err:
    alert_navigation_system(err)
```



```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [29], in <module>
     18 try:
---> 19     water_left(5, 100, 2)
     20 except RuntimeError as err:

Input In [29], in water_left(astronauts, water_left, days_left)
     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"

RuntimeError: There is not enough water for 5 astronauts after 2 days!

During handling of the above exception, another exception occurred:

NameError                                 Traceback (most recent call last)
Input In [29], in <module>
     19     water_left(5, 100, 2)
     20 except RuntimeError as err:
---> 21     alert_navigation_system(err)

NameError: name 'alert_navigation_system' is not defined
```