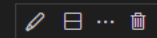


Ejercicio Kata 10



Tracebacks

```
>>> 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'
```

Python

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

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

Python

```
$ python3 open.py
Traceback (most recent call last):
  File "/tmp/open.py", line 5, in <module>
    main()
  File "/tmp/open.py", line 2, in main
```

Try y Except de los bloques

```
>>> try:
...     open('config.txt')
... except FileNotFoundError:
...     print("Couldn't find the config.txt file!")
...
Couldn't find the config.txt file!
```

Python

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

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

Python

```
$ python config.py
Traceback (most recent call last):
  File "/tmp/config.py", line 9, in <module>
    main()
```

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

Python

```
$ python config.py
Couldn't find the config.txt file!
```

Python

```
def main():
    try:
        configuration = open('config.txt')
    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")
```

Python

```
$ python config.py
Found config.txt but couldn't read it
```

Python

Generación de excepciones

```
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"
```

Python

```
>>> water_left(5, 100, 2)  
'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"
```

Python

```
>>> water_left(5, 100, 2)  
Traceback (most recent call last):
```

```

    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"

```

Python

```

>>> water_left("3", "200", None)
Traceback (most recent call last):
  File "<stdin>", line 5, in water_left
TypeError: unsupported operand type(s) for /: 'str' and 'int'

During handling of the preceding exception, another exception occurred:

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 9, in water_left
TypeError: All arguments must be of type int, but received: '3'

```

Python