KATA 10

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

0.4s Python

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Untitled-1.ipynb Cell 2' 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()
```

0.4s Python

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
c:\Users\Diego\Desktop\Launch X\Katas Propias\open.ipynb Cell 1' in <module>
      2     open("/path/to/mars.jpg")
      4 if __name__ == '__main__':
----> 5     main()

c:\Users\Diego\Desktop\Launch X\Katas Propias\open.ipynb Cell 1' in main()
      1 def main():
----> 2     open("/path/to/mars.jpg")

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

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

0.2s Python

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

```python
def main():
    open("config.py")

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

0.9s Python

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
c:\Users\Diego\Desktop\Launch X\Katas Propias\open.ipynb Cell 1' in <module>
      2     open("config.py")
      4 if __name__ == '__main__':
----> 5     main()

c:\Users\Diego\Desktop\Launch X\Katas Propias\open.ipynb Cell 1' in main()
      1 def main():
----> 2     open("config.py")

FileNotFoundError: [Errno 2] No such file or directory: 'config.py'
```

```python
r_left(astronauts, water_left, days_left):
y_usage = astronauts * 11
l_usage = daily_usage * days_left
l_water_left = water_left - total_usage
rn f"Total water left after {days_left} days is: {total_water_left} liters"
```

0.5s Python

```
>>> water_left(5, 100, 2)
'Total water left after 2 days is: -10 liters'
```

0.6s Python

```
'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
    return f"Total water left after {days_left} days is: {total_water_left} liters"
```

✓ 0.5s Python

```
>>> water_left(5, 100, 2)
'Total water left after 2 days is: -10 liters'
```

⊗ 0.6s Python

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Untitled-1.ipynb Cell 3' in <module>
----> 1 water_left(5, 100, 2)
      2 'Total water left after 2 days is: -10 liters'

Untitled-1.ipynb Cell 4' 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):
    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"
```

✓ 0.5s Python

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

⊗ 0.8s Python

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Untitled-1.ipynb Cell 5' in <module>
      1 try:
----> 2     water_left(5, 100, 2)
      3 except RuntimeError as err:

Untitled-1.ipynb Cell 4' in water_left(astronauts, water_left, days_left)
      5 if total_water_left < 0:
----> 6     raise RuntimeError(f"There is not enough water for {astronauts} as
tronauts after {days_left} days!")
      7 return f"Total water left after {days_left} days is: {total_water_lef
t} 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)
Untitled-1.ipynb Cell 5' in <module>
      2     water_left(5, 100, 2)
      3 except RuntimeError as err:
----> 4     alert_navigation_system(err)

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

```
>>> water_left("3", "200", None)
```

0.5s Python

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Untitled-1.ipynb Cell 6' in <module>
----> 1 water_left("3", "200", None)

Untitled-1.ipynb Cell 4' 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'
```

```python
    for argument in [astronauts, water_left, days_left]:
        try:
            # If argument is an int, the following operation will work
            argument / 10
        except TypeError:
            # TypError 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"
```

0.4s

```
>>> water_left("3", "200", None)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Untitled-1.ipynb Cell 7' 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     # TypError 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 8' in <module>
----> 1 water_left("3", "200", None)

Untitled-1.ipynb Cell 7' in water_left(astronauts, water_left, days_left)
      5         argument / 10
      6     except TypeError:
      7         # TypError 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'
```