

Excepciones. Ejercicios

Agenda básica

Tenemos un archivo en el disco con una agenda, algo así:

```
agenda.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
639232983 # blaky@ucm.es # calle BB 45 8-3-3 Madrid
659163091 # fer@upm.es # ctra. Húmera 45 8-3-3 Pozuelo
654091275 # elena@gmail.com # calle BB 12 8-3-3 Praga
234560123 # artu@ucm.es # calle Aravaca 1 12-3 Madrid
```

Si lo abrimos con un nombre erróneo, falla la cosa:

In [1]:

```
def cargar_agenda(nombre_archivo):
    agenda = dict({})
    el_archivo = open(nombre_archivo, "r")
    for l in el_archivo:
        datos = l.split(" # ")
        agenda[datos[0]] = {
            "email" : datos[1],
            "direc" : datos[2].rstrip()
        }
    el_archivo.close()
    return agenda

agenda = cargar_agenda("agenda.txt")
print(agenda)

# Esto va a fallar: el archivo no está en el disco
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-1-abd9852ccd19> in <module>
    11     return agenda
    12
--> 13 agenda = cargar_agenda("agenda.txt")
    14 print(agenda)
    15

<ipython-input-1-abd9852ccd19> in cargar_agenda(nombre_archivo)
     1 def cargar_agenda(nombre_archivo):
     2     agenda = dict({})
--> 3     el_archivo = open(nombre_archivo, "r")
     4     for l in el_archivo:
     5         datos = l.split(" # ")

FileNotFoundError: [Errno 2] No such file or directory: 'agenda.txt'
```

Arréglalo, manejando esa situación con una excepción.

In [2]:



```
def cargar_agenda(nombre_archivo):
    agenda = dict({})
    try:
        f = open(nombre_archivo, 'r')
    except:
        print('El archivo dado no está a mano')
        return
    el_archivo = open(nombre_archivo, "r")
    for l in el_archivo:
        datos = l.split(" # ")
        agenda[datos[0]] = {
            "email" : datos[1],
            "direc" : datos[2].rstrip()
        }
    el_archivo.close()
    return agenda

agenda = cargar_agenda("agenda.txt")
print()
agenda = cargar_agenda("agenda.txt")
print(agenda)
```

El archivo dado no está a mano

```
{'639232983': {'email': 'blaky@ucm.es', 'direc': 'calle BB 45 8-3-3 Madri
d'}, '659163091': {'email': 'fer@upm.es', 'direc': 'ctra. Húmera 45 8-3-3 Po
zuelo'}, '654091275': {'email': 'elena@gmail.com', 'direc': 'calle BB 12 8-3
-3 Praga'}, '234560123': {'email': 'artu@ucm.es', 'direc': 'calle Aravaca 1
12-3 Madrid'}}
```

Aserciones

Además de las excepciones, tenemos la función `assert`, para asegurarnos de que los requisitos de una función se cumplen.

Diseña un par de funciones con la instrucción `assert` y con excepciones para calcular el mcd de dos números, pensando en un usuario que pueda algún dato no entero, o nulo, o negativo... y compara su funcionamiento cuando la función es llamada desde otra.

In [3]:



```
def mcd(a, b):
    assert isinstance(a, int) and isinstance(b, int), "Los parámetros han de ser enteros"
    assert a != 0 and b != 0, "Los parámetros han de no nulos"
    a, b = abs(a), abs(b)
    while a != b:
        if a > b:
            a = a-b
        else:
            b = b-a
    return a

print(mcd(2.3, 4))
```

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-3-a0ca75462829> in <module>
     10     return a
     11
--> 12 print(mcd(2.3, 4))

<ipython-input-3-a0ca75462829> in mcd(a, b)
      1 def mcd(a, b):
--> 2     assert isinstance(a, int) and isinstance(b, int), "Los parámetro
s han de ser enteros"
      3     assert a != 0 and b != 0, "Los parámetros han de no nulos"
      4     a, b = abs(a), abs(b)
      5     while a != b:

AssertionError: Los parámetros han de ser enteros
```

In [4]:



```
def mcd(a, b):
    assert isinstance(a, int) and isinstance(b, int), "Los parámetros han de ser enteros"
    assert a != 0 and b != 0, "Los parámetros han de no nulos"
    a, b = abs(a), abs(b)
    while a != b:
        if a > b:
            a = a-b
        else:
            b = b-a
    return a

def prueba_mcd(m, n):
    print(m, n)
    try:
        cd = mcd(m, n)
        print(cd)
    except AssertionError as e:
        print("Algo va mal en los parámetros:", e)
        print()

prueba_mcd(2.3, 4)
prueba_mcd(2, "pepito")
prueba_mcd(-20, 0)
prueba_mcd(-20, 30)
```

2.3 4

Algo va mal en los parámetros: Los parámetros han de ser enteros

2 pepito

Algo va mal en los parámetros: Los parámetros han de ser enteros

-20 0

Algo va mal en los parámetros: Los parámetros han de no nulos

-20 30

10