

Manipulación de ficheros con Python

Fundamentos de Computación

Grado en Ingeniería en Tecnologías Industriales

16 de Noviembre 2021

Universidad de Cantabria

- Es útil conocer cómo podemos acceder desde un programa a los datos que contiene un archivo para manejarlos de manera automatizada.
- También es conveniente almacenar en ficheros los datos que devuelve un algoritmo, para poder utilizarlos con posterioridad, publicarlos o compartirlos con otras aplicaciones.

Analizaremos algunas operaciones básicas de edición de archivos de texto, tratándolos como objetos dentro de un programa Python. Es importante saber (o exigir) el formato que presenta el fichero de entrada.

Cuanto menos rígida sea esta especificación, más complicado será el trabajo del programador.

```
_____ gastos.txt _____  
1500  
350  
2600  
_____
```

```
fichero = open('gastos.txt','r')  
total = 0  
for linea in fichero:  
    num = int(linea)  
    total += num  
fichero.close()  
print ("TOTAL", total)
```

- 1 fichero=open("gastos.txt","r") Con esta instrucción *preparamos* el archivo para ser leído (parámetro r, read) y lo asociamos a la variable fichero.
- 2 fichero.close(). Es importante *cerrar* los archivos al final.
- 3 for linea in fichero:. Un fichero de lectura puede interpretarse como una lista formada por cadenas de texto (sus líneas).

El método `.readlines()` retorna una lista con cada línea del archivo de texto.

```
>>> fichero = open("gastos.txt","r")  
>>> fichero.readlines()  
['1500\n', '350\n', '2600']
```

Existen dos modos adicionales a `r` (lectura) para abrir un archivo: `w` (escritura) y `a` para añadir datos al final de un archivo. Con el modo `w` se borran los datos que el archivo contuviera anteriormente.

```
print("Vayan escribiendo sus nombres.")
print('Escriban "FIN" cuando no quede nadie más.')
```

`cadena = "PERSONAL\n"``fichero = open("lista.txt","w")``while cadena != "FIN":` `fichero.write(cadena+"\n")` `cadena = input()``fichero.close()`

```
>>>
Vayan escribiendo sus nombres.
Escriban "FIN" cuando no quede nadie más.
Claude E. Shannon
George Boole
Bertrand Russell
FIN
```

```
_____ lista.txt _____
PERSONAL
```

```
Claude E. Shannon
George Boole
Bertrand Russell
```

El sistema de D'Hondt

Gobierna la asignación de escaños parlamentarios en muchos procesos electorales. Construiremos un programa para automatizarlo, calculando, a partir de un archivo que contenga la distribución del voto entre las distintas candidaturas, otro donde aparezcan los representantes obtenidos por cada una de ellas.

votos.txt

P.A. 314

P.B. 1678

P.C. 501

P.D. 10

```
1 archivo = open('votos.txt','r')
2 for linea in archivo:
3     print(linea)
4 archivo.close()
```

P.A. 314

P.B. 1678

P.C. 501

P.D. 10

Vamos a construir una lista de *tuplas*: cada elemento de la lista será un par (nombre del partido, número de votos).

```
archivo = open('votos.txt', 'r')
lista = []
for linea in archivo:
    par = linea.split()
    lista.append((par[0], int(par[1])))
archivo.close()
print (lista)
```

Observa que, para que este fragmento de código funcione como esperamos, es preciso que cada línea del archivo de entrada se atenga al formato TEXTO NÚMERO. Obtendremos un error si, por ejemplo, el nombre de un partido consta de varias palabras.

```
[('P.A.', 314), ('P.B.', 1678), ('P.C.', 501), ('P.D.', 10)]
```

En las elecciones al Congreso de los Diputados, un primer requisito para obtener representación por una circunscripción es recibir al menos el 3% de los votos. Contémoslos (utilizando una variable llamada `suma`, por ejemplo) y quedémonos después con las candidaturas que cumplan esta condición:

```
suma = 0
for x in lista:
    suma += x[1]

umbral = suma*.03
pasan = []
for x in lista:
    if x[1] >= umbral:
        pasan.append(x)
print(pasan)
```

```
[('P.A.', 314), ('P.B.', 1678), ('P.C.', 501)]
```

El sistema de D'Hondt asigna sucesivamente los escaños que hay que repartir. El primer escaño le corresponde a la lista que tenga más votos. Amplíamos el programa para que localice la candidatura más votada.

```
def ganador(lista):
    record = 0
    for i in range(len(lista)):
        if lista[i][1] > record:
            record = lista[i][1]
            gana = i
    return gana
```

```
i = ganador(pasan)
print('El primer escaño va para',pasan[i][0]+'..')
```

```
El primer escaño va para P.B..
```


Cada uno de los escaños restantes, hasta completar los que hay para repartir, también se otorga a la candidatura que tenga un número asociado mayor. El número que se asocia a cada candidatura es el cociente entre los votos que ha recibido y el número de escaños que ya se le han asignado más uno.

Por ejemplo, para asignar el segundo escaño, se ordenan las candidaturas según su número de votos, salvo la que tiene más (y que ya ha recibido un escaño), cuyo número de votos se divide por dos. Para asignar el tercero, habrá que dividir por tres los votos de la candidatura ganadora, si le han correspondido los dos escaños anteriores; o por dos los de las dos primeras, si son dos los partidos que ya tienen representante.

```
def maxi(lista):
    record = 0
    for i in range(len(lista)):
        if lista[i] > record:
            record = lista[i]
            gana = i
    return gana
```

```
print ('Escaños totales: ')
total = int(input())

cuentas = []
asig = []
for x in pasan:
    cuentas.append(x[1])
    asig.append(0)

# Guardamos en la lista «cuentas» una copia de los votos de cada
# candidatura, para ir haciendo aquí las divisiones oportunas.
# En la lista «asig» iremos anotando los escaños que se asignen.

for i in range(1,total+1):
    j = maxi(cuentas)
    asig[j] += 1
    cuentas[j] = pasan[j][1] / (asig[j]+1)
print(asig)
```

```
Escaños totales:
```

```
5
```

```
[0, 4, 1]
```

```
Escaños totales:
```

```
8
```

```
[1, 6, 1]
```

Este programa pregunta directamente al usuario cuántos parlamentarios hay en juego, pero sería más conveniente que esta información estuviera ya contenida en el fichero de entrada. Modificaremos el programa según esta especificación.

Una vez calculados los resultados, los colocamos en un fichero de texto.

```
archivo = open('resultados.txt','w')
for i in range(len(pasan)):
    if asig[i] > 0:
        archivo.write('{0:10s} {1:1d}\n'.format(pasan[i][0]+':',asig[i]))
archivo.close()
```

Escaños totales:

12

[1, 9, 2]

resultados.txt

P.A.: 1

P.B.: 9

P.C.: 2

Podemos leer varios archivos y haga las cuentas para cada uno. Si prevemos tener en un directorio las votaciones de varias provincias, en archivos que comienzan por c_, podemos hacer lo siguiente:

```
from os import getcwd, listdir
ruta = getcwd()
archivos = listdir(ruta)
datos = []
for x in archivos:
    if x[:2] == 'c_':
        datos.append(x)
for provincia in datos:
    archivo = open(provincia, 'r')
    (...)
    n_salida = 'res'+provincia[1:]
    salida = open(n_salida, 'w')
    (...)
```

Otros ficheros(.jpg,.pdf,..)

Podemos leer y escribir en ficheros distintos de .txt.

```
import urllib.request
from urllib.error import HTTPError
web = 'http://web.unican.es/centros/minas/ImagenesExposicion/'
for index in range(1,20):
    nombre = str(index)+".jpg"
    url = web + nombre
    try:
        f = urllib.request.urlopen(url)
        archivo = open(nombre, "wb")
        archivo.write(f.read())
        archivo.close()
    except HTTPError :
        print("HTTP Error:", url)
```