

FUNDAMENTOS DE INFORMÁTICA

INTRODUCCIÓN A LA PROGRAMACIÓN

Listas II

1 Pruebas iniciales (20 min)

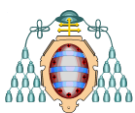
Usando el intérprete python, crea una variable llamada `texto` que contenga “Esto es una prueba”, y otra llamada `fecha` que contenga la cadena “16/10/2000”.

1.1 Operaciones básicas sobre cadenas

1. Usa `len()` para averiguar la longitud del texto
2. Usa el operador `+` para concatenar las dos variables (`fecha` y `texto`), con un guión separándolas y guarda el resultado en la variable `resultado`. Al final, debe contener: “16/10/2000 – Esto es una prueba”
3. Usa el operador `in` para averiguar si el `texto` contiene la palabra “prueba” o la palabra “esto” (con la “e” minúscula)
4. Ejecuta `texto.split()` y observa la lista que resulta. ¿Qué hace `split()`? Prueba `texto.split(“e”)` ¿qué significa el parámetro que pasas a `split()`?
5. Usa `split()` sobre `fecha` para separar su contenido en tres variables `dia`, `mes` y `anyo`.
6. Usa `help(str.replace)` para averiguar cosas sobre el método `replace` que puedes usar sobre una cadena. Úsalo sobre la variable `texto` para cambiar “es” por “era”.

1.2 Listas y ficheros

7. Prepara un fichero que contenga varias líneas de texto. Abre el fichero sobre una variable `f`, mediante `f=open("nombre-del-fichero.txt")`
8. Ejecuta `datos=f.readlines()` y observa la variable `datos`. Es una lista en la que cada elemento es un string que contiene una línea del fichero (no olvides `f.close()` después).
9. Usa `len(datos)`. El resultado es el número de líneas del fichero. ¿Qué resultado te da `len(datos[0])` y qué significa?
10. Observa `datos[0]` y verás que tiene un `\n` al final. Este carácter representa el “retorno de carro” que hay al final de cada línea del fichero. Normalmente no lo queremos. Las cadenas tienen el método `rstrip()` que elimina todos los caracteres invisibles (espacios, retornos de carro, etc) que pueda haber al final de una cadena. No modifican la propia cadena (no pueden), sino que retornan la versión sin esos caracteres finales. Así: `linea=datos[0].rstrip()`. Si miras la variable `linea`, verás que ya no tiene al final el `\n` (aunque `datos[0]` sigue teniéndolo).



2 Ejercicios (40 min)

► EJERCICIO 1. LEER UNA LISTA DE NÚMEROS DE UN FICHERO

Descarga el fichero llamado datos.txt. Observa que este fichero se compone de una serie de líneas, en cada una de las cuales hay un número entero.

Escribe una función que reciba como parámetro el nombre del fichero y devuelva como resultado una lista con los números leídos del fichero. Aplica la función sum() de python sobre esta lista para averiguar la suma de todos los datos que has leído¹. Debe salirte 49195.

► EJERCICIO 2. ANALIZAR DATOS EN UN FICHERO

Descarga el fichero llamado alumnos.txt. Contiene un listado de todos los alumnos matriculados en la asignatura “Fundamentos de Informática” en la EPI-Gijón, en el curso 2011-2012, pero por motivos de privacidad de los datos se han intercambiado aleatoriamente los apellidos y los nombres entre los alumnos. Si das un vistazo al fichero, abriéndolo con el notepad++, verás que contiene un alumno por línea, figurando en primer lugar el apellido y después, separado por coma, el nombre.

Realiza los ejercicios siguientes sobre los datos de este fichero.

- Escribe código python para determinar el número total de alumnos
- Escribe una función que reciba dos parámetros: una línea leída del fichero y un apellido de persona, y la función debe retornar “True” si cualquiera de los dos apellidos de la persona coinciden con el apellido dado. Por ejemplo:

```
>>> coincide_apellido("Lopez Alvarez, Andres", "Lopez")
True
>>> coincide_apellido("Lopez Alvarez, Andres", "Suarez")
False
>>> coincide_apellido("Lopez Alvarez, Andres", "Andres")
False
```

- Usando la función anterior, determina qué porcentaje de alumnos tiene “Fernandez” como alguno de sus dos apellidos (debe salirte el 15,256%)
- Escribe una función que reciba como parámetro una línea del fichero y retorne un entero que indique cuántas palabras tiene el nombre. Por ejemplo:

```
>>> palabras_en_nombre("Lopez Alvarez, Andres")
1
>>> palabras_en_nombre("Suarez Fernandez, Jose Luis")
2
>>> palabras_en_nombre("de Marichalar y Borbon, Felipe Juan Froilan de Todos los Santos")
7
```

- Usando la función anterior, escribe un programa que cuente cuántos alumnos tienen nombre simple (1 palabra), cuántos de 2 palabras, etc. Puedes usar una lista para cada posible número de palabras, de modo que numero_palabras[1] por ejemplo almacene cuántos tienen nombre simple, numero_palabras[2] cuántos tienen 2 palabras, etc.. Puedes asumir que no habrá nadie con más de 7 palabras y por tanto darle a la lista un tamaño prefijado de 8 elementos (ya que el elemento [0], aunque no lo usaremos, está ahí)

¹ Cuidado! Si no te funciona ¿has tenido en cuenta que los elementos de la lista han de ser de tipo int?



Tu programa comenzaría poniendo 8 ceros en esa lista, para después ir leyendo líneas del fichero y para cada una de ellas llamar a `palabras_en_nombre()` que le dirá que ese alumno tiene X palabras en su nombre, y seguidamente incrementarás `numero_palabras[X]` para contar que hay un alumno más con X palabras en su nombre.

Al final imprimes la lista `numero_palabras` (o si prefieres, solo los elementos con valor distinto de cero). Debe salirte que hay 767 nombres simples, 120 con dos palabras, 9 con tres palabras y 2 con cuatro palabras.

3 Ejercicio extra, opcional

► EJERCICIO 3. IMPORTAR Y PROCESAR DATOS DE EXCEL

Descarga el fichero llamado `datos-paises.xls`. Ábrelo en Excel y verás que contiene una lista de nombres de país, de número de habitantes de cada uno de ellos, y de número de teléfonos móviles². Desde Excel guárdalo con la opción “Guardar como...” y elige “CSV separado por comas”. Eso te creará el fichero `datos-paises.csv`

Si abres dicho fichero `datos-paises.csv` con cualquier editor de texto (por ejemplo con `notepad++`) podrás ver que simplemente consta de una línea por cada fila de la hoja Excel, y en cada línea se han volcado los contenidos de las celdas, separados por punto y coma.

Desde el editor, borra la primera línea del fichero que has generado, pues no contiene datos de ningún país, sino la cabecera de la tabla.

A la vista del formato csv, escribe programas en python que te permitan:

1. Calcular el número total de habitantes (suma de todos los países)
2. Calcular el número promedio de habitantes por país (suma de habitantes entre número de países)
3. Generar otro archivo csv que contenga la lista de países y el número promedio de teléfonos móviles por habitante de cada uno.
4. Encontrar los países cuyo número de habitantes sea cercano al número promedio calculado en el paso 2 (entenderemos que es cercano si la diferencia es menor de un millón).
5. ¡Otras cosas que se te ocurran! (por ejemplo ¿cuál es el país que tiene más teléfonos móviles por habitante?)

² Datos reales obtenidos de <https://www.cia.gov/library/publications/the-world-factbook/>