

Ejercicios de preparación



Miniproyecto 3

Se recomienda en realizar estos ejercicios **antes** de desarrollar el miniproyecto 3, ya que les facilitará enormemente la tarea. Se sugiere además hacer los 5 ejercicios en el orden dado.

En el miniproyecto van a utilizar lo aprendido con estos ejercicios.

Ejercicio 1

Escribe un programa que lea un archivo de texto llamado "amigos.txt" que contiene en cada línea el nombre y la edad de cada uno de tus amigos(as). Algo como lo siguiente:

```
Pedro 32
Julia 26
Ema 31
Luis 39
```

El objetivo es imprimir una lista que contiene a cada amigo y su edad como una lista dentro de ella. En el ejemplo anterior el resultado sería: `[['Pedro', '32'], ['Julia', '26'], ['Ema', '12'], ['Luis', '39']]`

Ayuda

- a) Para leer el archivo se usa la función `open` que recibe como primer parámetro el nombre del archivo y como segundo parámetro si solo se va a leer (`'r'`) o también escribir (`'w'`). Lo que devuelve el `open` es un descriptor de archivo que puede ser usado posteriormente para leerlo. En nuestro caso sería:

```
>> f = open("amigos.txt", "r")
```

Luego para leer una línea del archivo se usa el método `readline()` sobre `f`.

```
>> linea = f.readline()
```

- b) Recuerda que el contenido de cada línea del archivo será un string que podemos separar usando la función `split()`.

Si `linea = "Pedro 32"`, entonces `linea.split()` = `['Pedro', '32']`

c) Para agregar elementos al final de una lista podemos usar el método `append`.

Si `lalista = []` y `elItem = ['Pedro', '32']`, entonces `lalista.append(elItem)` será `[['Pedro', '32']]`.

Si le agrego ahora `otroItem = ['Julia', '26']`, entonces `lalista.append(otroItem)` será `[['Pedro', '32'], ['Julia', '26']]`.

Ejercicio 2

Escribe un programa lea un archivo `ramos.csv` (los ramos de cada línea están separados por comas) y en forma similar al ejercicio 1 produzca una lista con las listas correspondientes a cada línea del archivo.

Por ejemplo, si el archivo `ramos.csv` contiene:

```
Python, Bases_de_Datos, Estadistica, Ciencia_de_Datos
Estadistica, Ciencia_de_Datos, Inteligencia Artificial
Tenis, Futbol, Música
```

Debe imprimir:

```
>> [['Python', 'Bases_de_Datos', 'Estadistica', 'Ciencia_de_Datos'],
      ['Estadistica', 'Ciencia_de_Datos', 'Inteligencia_Artificial'],
      ['Tenis', 'Futbol', 'Musica']]
```

Ayuda

- a) Nada especial, es muy similar al ejercicio 1. Un archivo `csv` es un archivo de texto especial.
- b) Ojo que el `split` debe ir con una `,` en lugar de `" "` porque los ramos están separados por comas.
- c) Es conveniente eliminar el caracter de `new line`, al final con `rstrip('\n')`.

Ejercicio 3

Construye un diccionario Python para almacenar información de tu biblioteca de forma que las llaves son las categorías de libros y los valores el número de libros en la categoría. Así por ejemplo una entrada en el diccionario tendrá por ejemplo como clave 'Python' y como valor 3 (indicando que tienes 3 libros de Python).

A partir de una lista con los pares llave y valor construye un diccionario con la información. Luego usa el diccionario para imprimir cuantos libros de Python tienes.

Por ejemplo, si el input es:

```
list = [['Python', '3'], ['R', '1'], ['Elixir', '4']]
```

Entonces el diccionario generado es:

```
dic = {'Python': 3, 'R': 1, 'Elixir': 4}
```

Y, el valor a imprimir es `dic['Python']`, es decir 3.

Ayuda

- a) Parte con un diccionario vacío en la forma:

```
dic = { }
```

- b) Recuerda que, para agregar un par llave, valor a un diccionario `dic` debes hacer:

```
dic[llave] = valor
```

- c) Cada ítem de la lista tiene llave y valor, recuerda que el valor lo debes convertir en `int` antes de insertarlo en el diccionario.

Ejercicio 4

Supón que en el ejercicio 2, cada línea en el archivo de ramos comienza con el número de estudiante que toma esos ramos.

Por ejemplo, el archivo `ramos.csv` contiene:

```
1234567, Python, Bases_de_Datos, Estadistica, Ciencia_de_Datos
3245673, Estadistica, Ciencia_de_Datos, Inteligencia_Artificial
3454232, Tenis, Futbol, Musica
```

Lo que queremos, es poner esta información en un diccionario Python, donde la llave es el número de alumno y el valor la lista con los ramos.

llave	valor
'1234567'	['Python', 'Bases_de_Datos', 'Estadística', 'Ciencia_de_Datos']
'2345673'	['Estadística', 'Ciencia_de_Datos', 'Inteligencia_Artificial']
'3454232'	['Tenis', 'Fútbol', 'Música']

Ayuda

- a) Si tenemos una lista que incluye el número de alumno y los ramos podemos separar el número de alumno usando el método `pop(0)` sobre la lista. Eso extrae el primer elemento y deja la lista con lo resultante.

Si `lista = ['1234567', 'Python', 'Bases_de_Datos', 'Estadística', 'Ciencia_de_Datos']`, entonces al hacer `first = lista.pop(0)` dejará a `first` con el valor `'1234567'` y la lista con el resto de los elementos.

Para eliminar espacios y fines de línea, se recomienda usar `lista.pop(0).strip().rstrip('\n')`

- b) Al tener ya separados el número de alumno (`first`) y la lista (`lista`) podemos agregarlos fácilmente en un diccionario con:

```
dic[first] = lista
```

Ejercicio 5

Supón ahora que tenemos ya armados un diccionario de recetas y un diccionario de ingredientes. En el diccionario de recetas la clave es el nombre de la receta y el valor es la lista de ingredientes necesarios para hacerla. En el diccionario de ingredientes la clave es el nombre del ingrediente y el valor es la cantidad disponible de él.

Escribe un programa que pida el nombre de una receta y busque en el diccionario de recetas los ingredientes para luego buscar las disponibilidades de cada uno de ellos en el diccionario de ingredientes. De acuerdo a ello imprima: "a la orden" si es posible hacer la receta y "sorry, por el momento no es posible" si no lo es. Si la receta no existe imprimir "sorry, pero no preparamos esa receta"

Estos son los diccionarios cargados inicialmente:

```
dic_ingred = {'Tomate': 5, 'Lechuga': 2, 'Hamburguesa': 10, 'Carne': 1, 'Espárragos': 8, 'Pan': 10, 'Papa': 5, 'Cebolla': 12}
```

```
dic_recetas = {'HamburguesaCasera': ['Hamburguesa', 'Tomate', 'Pan'], 'PastelDeCarne': ['Carne', 'Papa', 'Cebolla'], 'EnsaladaEspecial': ['Lechuga', 'Espárragos', 'Tomate']}
```

Ayuda

- a) a) Primero verificar si la receta es válida viendo si está en el diccionario (recuerda que se usa `in`), `receta_pedida in dic_recetas` debe ser verdadero.
- b) Obtener la lista de ingredientes de la receta y para cada uno de ellos acceder al diccionario de ingredientes. Si la cantidad es cero no se puede preparar, si no es cero descontar en uno y seguir con el siguiente ingrediente de la receta.