



Semana 8: Diccionarios y listas de listas

Programación

2020 - 2021

- Ejercicio 1.** Crea tres diccionarios: `maria`, `pedro` y `miguel`. Dale a cada diccionario las claves `nombre`, `ejerciciosSemanales`, `pruebasSemanales`. En la clave `nombre` almacena el nombre del estudiante (es decir, en el diccionario `maria` debe valer `'María'`, etc.). Las otras claves deben contener listas vacías. Imprime los resultados por pantalla. A continuación genera 10 elementos aleatorios entre 0 y 10 para una de las dos otras claves. Crea una nueva clave denominada `examen` en cada uno de ellos y rellénala con un número aleatorio entre 0 y 10.
- Ejercicio 2.** Crea un diccionario llamado `precios` y pon estos valores en él: `manzana: 5.5`, `melon: 6`, `banana: 7.2` y `naranja: 4`. Después itera por las claves y los valores, imprimiendo por pantalla cada clave junto a su valor.
- Ejercicio 3.** Dado el siguiente diccionario:
- ```
juguetes = {
 'muñecas': 500,
 'juegos': ['Guess who?', 'Clue', 'Battleship'],
 'puzzles': ['Star Wars', 'Batman', 'Ironman']
}
```
- Añade una clave `peluche` a `juguetes`, y establece su valor a `oso`, `perro` y `gato`.
- Elimina `Batman` de la lista de `puzzles`.
- Añade 100 al valor almacenado en `muñecas`.
- Ejercicio 4.** Crear una lista llamada `estudiantes` con los diccionarios del Ejercicio 1. Para cada elemento de la lista:
- Calcula la media de sus ejercicios semanales y de sus pruebas semanales. En cada uno de ellos, cambia las dos listas por una nueva lista de dos elementos. El primer elemento será la lista anterior con las notas de los ejercicios, el segundo la media.
  - Añade a cada elemento una nueva clave denominada `nota` que se calculará de la manera siguiente: `ejerciciosSemanales 10%`, `pruebasSemanales 30%` y `examen 60%`.
  - Añade una nueva clave denominada `notaLiteral` cuyo valor se calculará según la siguiente tabla:
    - `'A'`, si los puntos son 9 o más.
    - `'B'` si los puntos son 8 o más.
    - `'C'`, si los puntos son 7 o más.
    - `'D'`, si los puntos son 6 o más.
    - `'E'` si los puntos son 5 o más.
    - `'F'` en cualquier otro caso.
  - Para cada elemento de `estudiantes`, imprime por pantalla sus datos, uno en cada línea.
  - Calcula e imprime la nota media de la clase (nota media de los alumnos)
- Ejercicio 5.** Crear un diccionario cuyas claves sean los nombres de los meses del año y sus valores los días de cada mes (`'Enero': 31`, etc.). Crear otro llamado `fecha` con claves: `día`, `mes`, `año` y

bisiesto. Pedir al usuario valores para rellenar este segundo diccionario. El valor bisiesto se debe calcular automáticamente. Se debe comprobar que el mes es correcto y el día también, teniendo en cuenta años bisiestos (Febrero tiene 29 días esos años). En caso de que algún dato sea incorrecto se volverá a pedir. Imprimir al terminar la fecha en el formato: 23 de Marzo de 2019, año no bisiesto.

**Ejercicio 6.** Crear una programa que pida por pantalla los elementos a introducir en dos matrices M1 y M2 de diferentes tamaños (representadas como listas) y muestre los elementos de M1 que están incluidos en M2. Ejemplo de ejecución:

```
Introduce el número de filas de la matriz A y pulsa Enter: 2
Introduce el número de columnas de la matriz A y pulsa Enter: 3
Introduce el número de la posición 0, 0: 1
Introduce el número de la posición 0, 1: 3
Introduce el número de la posición 0, 2: 5
Introduce el número de la posición 1, 0: 7
Introduce el número de la posición 1, 1: 9
Introduce el número de la posición 1, 2: 11
La matriz A es:
1 3 5
7 9 11
Introduce el número de filas de la matriz B y pulsa Enter: 3
Introduce el número de columnas de la matriz B y pulsa Enter: 2
Introduce el número de la posición 0, 0: 1
Introduce el número de la posición 0, 1: 2
Introduce el número de la posición 1, 0: 3
Introduce el número de la posición 1, 1: 4
Introduce el número de la posición 2, 0: 5
Introduce el número de la posición 2, 1: 6
La matriz B es:
1 2
3 4
5 6
El elemento 1 está incluido en ambas matrices
El elemento 3 está incluido en ambas matrices
El elemento 5 está incluido en ambas matrices
```

**Ejercicio 7.** Crear una matriz NxN representada como un lista de N elementos, siendo cada uno de ellos otra lista con N números:  $a_1, a_2 \dots a_{n-1}, a_n$  siendo N un número aleatorio entre 3 y 20 y cada  $a_i$  un número aleatorio entre 1 y 10, excepto  $a_n$  que será la suma de los anteriores. Imprimir la matriz como se muestra en el ejemplo para una matriz de 4x4:

```
10 + 9 + 3 = 22
4 + 8 + 7 = 19
1 + 9 + 7 = 17
2 + 8 + 8 = 18
```

**Ejercicio 8.** Dada una matriz de 3x3. Rellenarla con números aleatorios entre el 1 y el 30, teniendo en cuenta que ningún número se puede repetir.

|   |    |    |
|---|----|----|
| 7 | 3  | 21 |
| 9 | 12 | 1  |
| 8 | 13 | 5  |

Una vez creada se deben poner sus elementos en orden sin utilizar las funciones `max()`, `min()` ni el método `sort()`.

|    |    |    |
|----|----|----|
| 1  | 3  | 5  |
| 7  | 8  | 9  |
| 12 | 13 | 21 |

- Ejercicio 9.** Una carpintería vende tableros de madera de 10 tamaños diferentes, denominados tamaño 1, tamaño 2, etc. Crear un programa que:
- Genera 10 números aleatorios entre 0 y 10, que representan la cantidad de tableros de cada tamaño que hay, y los imprime según se muestra en el ejemplo.
  - Pide al usuario el número de clientes que hay. Debe asegurarse de que hay al menos un cliente.
  - Para cada cliente genera aleatoriamente 10 números entre 0 y 4 que representan la cantidad de tablonos de cada medida que quiere comprar, e imprime los pedidos.
  - Vende tablonos a cada cliente, empezando por el primero, reduciendo el número de tablonos de cada tipo según se van vendiendo.
  - Imprime los pedidos pendientes de cada cliente que no haya podido comprar todos los tablonos que quería al haberse agotado. Estos pedidos pendientes reflejarán el número de tablonos de cada tipo que le quedan por comprar.

```
Almacenados: tamaño0 tamaño1 tamaño2 tamaño3 tamaño4 tamaño5 tamaño6 tamaño7 tamaño8 tamaño9
 1 6 4 5 5 8 1 8 7 2
Introduce el número de clientes: 3
Pedidos originales
Cliente 0: 3 3 1 0 3 0 2 3 3 3
Cliente 1: 0 3 3 1 1 0 0 1 3 0
Cliente 2: 2 3 0 1 0 2 0 0 0 3

Pedidos pendientes
Cliente 0: 2 0 0 0 0 0 1 0 0 1
Cliente 2: 2 3 0 0 0 0 0 0 0 3
```

## Normas de entrega

Los ejercicios se deben subir a Aula Global antes de las 7:00 de la mañana del próximo día con clase de laboratorio. ). Se deberá subir un fichero comprimido **zip** con un fichero para cada programa, nombrados ejercicio1.py, ejercicio2.py, etc. El nombre del fichero será "s8-iniciales-del-alumno.zip" (por ejemplo Lucía Pérez Gómez subirá un archivo llamado s8-lpg.zip).