CURSO DE PROGRAMACIÓN FULL-STACK

Python / Guía 4

Ejercicios

Colecciones de Datos I

egg

Ejercicios de aprendizaje

Con Visual Studio Code, debes crear un archivo para cada ejercicio, denominando a cada archivo de la siguiente manera: `ejercicio_4_1.py`, `ejercicio_4_2.py` , `etc.`

1. **Tuplas.** A partir de la siguiente tupla:

   ```
   tupla = (5, 12, 7, 37, 8, 86, 19, 7, -783, 87, 188, 7, 9, 12, 7, 3982)
   ```

   Imprimir por pantalla, de forma ordenada, lo siguiente:

   a) El último elemento de tupla
   b) El número de elementos de tupla (usa la función 'len')
   c) La posición donde se encuentra el elemento 87 de tupla
   d) Un elemento que esté en la posición 8 de tupla
   e) Una lista con los últimos tres ítems de tupla
   f) El número de veces que el ítem 7 aparece en tupla

2. **Listas.** A partir de las siguientes listas:

   ```
   lista_1 = [1, 12, 123]
   lista_2 = ["Bye", "Ciao", "Agur", "Adieu"]
   ```

   a) Añade a la `lista_1` el entero `1234` y luego la cadena `"Hola"`
   b) Añade a la `lista_2` el string `“Adios”` y luego el entero `1234`
   c) Genera una `lista_3` con todos los elementos de la `lista_1` menos el último
   d) Genera una `lista_4` con todos los elementos de la `lista_2` menos el primero y el último
   e) Genera una `lista_5` con los elementos de la `lista_4` y de la `lista_3`

3. **Conjuntos.** Crear un conjunto 'usuarios' con los usuarios 'Brahms', 'Schubert', 'Vivaldi', 'Verdi' y 'Tchaikovsky'. Crear otro conjunto 'administradores', y poner como administrador a 'Brahms' y 'Verdi'. Eliminar al administrador 'Brahms' del conjunto de administradores. Agregar a 'Tchaikovsky' como nuevo administrador, pero no eliminarlo del conjunto de usuarios. Mostrar todos los usuarios por consola. Además, mostrar si cada usuario es administrador o no.

4. Escribir un programa que pregunte al usuario cuántas tareas quiere anotar. Ir agregando cada tarea que el usuario vaya ingresando a una colección. Mostrar por pantalla la lista de tareas en orden alfabético.

5. Escribir un programa que pregunte al usuario que ingrese palabras, y cada vez que ingrese una, la guarde en una lista. La palabra será guardada en mayúsculas y en orden inverso. Salir del programa cuando el usuario no ingrese nada. Mostrar por pantalla la lista completa de palabras invertidas.

6. Crear la lista 'mis_numeros'. El usuario debe introducir enteros. Cada vez que ingresa un número, guardarlo en una tupla anidada dentro de la lista 'mis_numeros'. Sale de la entrada de números cuando presiona 'enter' si no hay nada ingresado. Al terminar, mostrar por pantalla 'mis_numeros', por un lado, y por otro, los elementos desempacados de 'mis_numeros' (no aparecerán los corchetes). Finalmente, mostrar por pantalla cada elemento desempacado de cada tupla (no aparecerán los paréntesis ni la coma). Usar la función len().

Ver video "Listas"

7. **Listas.** Realizar el siguiente ejercicio, y **después** ver el video:

   A partir de la siguiente variable:

```
matriz = [[1, 5, 1],
          [2, 1, 2],
           [3, 0, 1],
           [1, 4, 4]]
```

   Debes crear un cuarto elemento en las listas anidadas, cuyo valor deber ser el resultado de la suma de los tres elementos actuales. El resultado final debe ser el siguiente:

```
matriz = [[1, 5, 1, 7],
                [2, 1, 2, 5],
                 [3, 0, 1, 4],
                 [1, 4, 4, 9]]
```

Lo debes hacer creando tres fragmentos de códigos para llegar al mismo resultado de 3 formas:

a) Usando append() y slicing.
b) Usando append() y sum()
c) Usando el operador + y sum() pero sin usar append()

Ver video “Matrices”