CURSO DE PROGRAMACIÓN FULL-STACK

Python / Guía 5

Ejercicios

Colecciones de Datos II

egg

Ejercicios de aprendizaje

Con Visual Studio Code, debes crear un archivo para cada ejercicio, denominando a cada archivo de la siguiente manera: `ejercicio_5_1.py`, `ejercicio_5_2.py`, etc.

1. Escribir un programa que almacene en una lista los números del 1 al 10 y los muestre por pantalla en orden inverso.

2. Escribir un programa que enumere los países de la siguiente lista:
   `["Canada", "USA", "Mexico", "Australia", "Argentina", "China", "India"]`

3. Pedirle a un usuario que ingrese un número, y devolver los dígitos totales del número. Por ejemplo, si el número es 75869, la salida debería ser 5

4. Escribir un programa que pida al usuario un número entero del 0 al 9 dentro de un bucle. Comprobar si el número se encuentra en la lista de números válidos y notificarlo:
   `numeros_validos = [1, 3, 6, 9]`

5. Crear una lista de países en letras minúsculas. Luego, por medio de la comprensión de listas, crear una nueva lista con las mismas cadenas, cuya primera letra sea en mayúsculas.

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

   Dadas dos listas, debes generar una tercera con todos los elementos que se repitan en ellas, pero no debe repetirse ningún elemento en la nueva lista.

   ```
   lista_1 = ["h", "o", "r", "o", " ", "s", "o", "l", "a"]
   lista_2 = ["h", "o", "l", "a", " ", "l", "u", "n", "a"]
   ```

Ver video “Recorrer listas”

7. Programar las siguientes instrucciones de forma ordenada. Inicialmente, el diccionario es: `animales = {"elefante": ""}`

   a) Añadir al diccionario las claves: "perro", "tigre" y "mono", con sus respectivos valores: "Bobby", "Peepe" y "Homero"

   b) Modificar las claves: "elefante" y "delfín" con los valores: "Trompis" y "Manolo" respectivamente.

8. A partir del siguiente diccionario:

   ```
   {"Euro": "€", "Dólar": "$", "Yen": "¥"}
   ```

   Preguntar al usuario por una moneda, y luego mostrar su símbolo, o un mensaje si la moda no está en el diccionario.

9. Pregunta al usuario por su nombre, edad, dirección y teléfono. Guardar los datos en un diccionario. Mostrar por pantalla: <nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>.

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

    Crear un diccionario con la siguiente estructura:

    ```
    {
        "artículo": {
            int: {"nombre": str, "precio": float, "stock": int},
        }
    }
    ```

    El programa permitirá cargar artículos. La clave <int> del diccionario anidado, corresponde al código de un artículo. Cada vez que se añada un nuevo dato, debe imprimirse el contenido del diccionario.

Cuando el diccionario es grande, es difícil leerlo. Te aconsejo que uses el módulo "pprint" que proporciona la capacidad de "imprimir de forma bonita" estructuras de datos arbitrarias de Python. La representación formateada mantiene los objetos en una sola línea siempre que sea posible y los divide en varias líneas si no encajan dentro del ancho permitido.en la primera línea del código escribas:

```
from pprint import pprint
pprint(diccionario)
```

Crear varios artículos y simular una venta. Informar si no hay stock y el precio final de la compra.

Ver video “Diccionarios”