

Práctica Python

18-X-2024

Enunciado

En esta práctica nos enfocaremos en consolidar el uso de estructuras de datos fundamentales en Python, como tuplas, listas y diccionarios, aplicándolas en contextos relacionados con la gestión empresarial y la economía. La práctica está dividida en dos partes bien diferenciadas: Ejercicios rápidos y Resolución de un problema práctico.

- Esta práctica se debe entregar antes del **3 de Noviembre de 2024**.
- El resultado del trabajo es un fichero *notebook Python* (.ipynb).
- Cada ejercicio se debe realizar en una celda independiente. Un ejercicio se puede resolver con un único bloque de código o una sola función, pero si lo creéis conveniente también es posible crear diferentes funciones.
- La práctica debe publicarse en el correspondiente apartado de la página de la asignatura y en **GitHub**. El enlace a vuestro repositorio debe estar en la primera celda **markdown** de vuestro documento. Se ha de entregar también el fichero de texto volcado de **git log**.

Ejercicios rápidos

1. Realizar una función que recibe 3 números (d, m, a) que representan una fecha (día, mes y año). El programa debe devolver el día (la fecha) siguiente a la recibida por parámetro. Se debe tener en cuenta los días de cada mes y los años bisiestos.
2. Dada la siguiente lista:

```
palabras_espanol = ["Perro", "Gato", "Casa", "Coche",  
                    "Árbol", "Sol", "Luna", "Mar", "Montaña", "Río",  
                    "Playa", "Libro", "Computadora", "Teléfono",  
                    "Amigo", "Familia", "Trabajo", "Comida", "Película",  
                    "Música", "Viaje", "Deporte", "Juego", "Amor", "Odio",  
                    "Felicidad", "Tristeza", "Alegría", "Dolor", "Sueño",
```

```
"Noche", "Día", "Agua", "Fuego", "Aire", "Tierra",  
"Cielo", "Estrella", "Flor", "Fruta", "Verdura",  
"Chocolate", "Café", "Té", "Vino", "Cerveza",  
"Dinero", "Tiempo", "Reloj", "Reloj"]
```

Construye un programa que muestre por pantalla cuantas veces aparece cada una de ellas.

3. Dado el siguiente diccionario con información del precio de diferentes productos por día.:

```
precio_origen = {  
    "cebolla": [30.3, 20.3, 25.9, 39.0], # céntimos/kg  
    "patata": [0.33, 0.43, 0.34, 0.12], # euro/kg  
    "naranja": [0.45, 0.56, 0.38, 0.43] # euro/kg  
}
```

Responde a las siguientes preguntas:

- ¿Cuál es el precio máximo de cada producto?
- ¿Cuál es el producto más caro?
- ¿Qué día son más caras las naranjas?

Problema

Una empresa tiene un inventario de productos con diferentes categorías, y cada producto tiene un precio y una cantidad en stock. El objetivo es gestionar el inventario de forma eficiente. Instrucciones:

1. Crea un diccionario donde la clave sea el nombre de una categoría de producto y el valor sea una lista de tuplas. Cada tupla contendrá el nombre del producto, el precio y la cantidad en stock.
2. Crea una función que calcule el valor total de inventario por categoría (precio por cantidad para cada producto).
3. Muestra la categoría que tiene el valor total más alto en inventario.
4. Añade una opción para actualizar el stock de un producto en una categoría específica.



Anexo

Uno de los requisitos de la entrega de la práctica es la entrega de un fichero de texto volcado de `git log`. Para obtener este fichero, en vuestra terminal, y una vez estéis en la carpeta de vuestro repositorio, tenéis que hacer la siguiente instrucción:

```
git log >> dump.txt
```

El resultado es un fichero `dump.txt` con la información de los diferentes `commits` realizados.