Desafío funciones y listas

Instrucciones

A continuación de detallan variados desafíos a desarrollar.

Para su correcta corrección, los programas deben ser almacenados en un comprimido `.zip` de la siguiente manera:

Desafio-funciones-listas.zip

├── velocidad.py

├── listas_uno.py

├── listas_dos.py

├── listas_tres.py

├── listas_cuatro.py

├── listas_cinco.py

├── asociar.py

Desafío - Velocidad

Se pide crear el programa `velocidad.py` que dada la información de velocidad promedio de distintos vehículos se pueda entregar cierta información.

```
velocidad = [4, 4, 7, 7, 8, 9, 10, 10, 10,
             11, 11, 12, 12, 12, 12, 13, 13,
             13, 13, 14, 14, 14, 14, 15, 15,
             15, 16, 16, 17, 17, 17, 18, 18,
             18, 18, 19, 19, 19, 20, 20, 20,
             20, 20, 22, 23, 24, 24, 24, 24, 25]
```

Se pide:

- Crear una función llamada `promedio` que devuelva la velocidad promedio de los vehículos en la lista.

tips:_

- La corrección del ejercicio funciona llamando a la función `promedio`, por lo que la función debe existir y el valor ser el promedio de cualquier arreglo ingresado
- Puede probar el programa llamando a la función y mostrando el resultado pero no es necesario que el programa entregue resultado alguno. La revisión se hace llamando a la función.

Desafío - Listas

A continuación se presenta una serie de listas.

```
auto1 = ['Mazda RX4', 21.0, 6, False, 4]
auto2 = ['Merc 240D', 24.4, 4, True, 2]
auto3 = ['Merc 280', 19.2, 6, True, 4]
auto4 = ['Valiant', 18.1, 6, True, 1]
auto5 = ['Merc 450SL', 17.3, 8, False, 3]
auto6 = ['Toyota Corolla', 33.9, 4, True, 1]
```

Se solicita lo siguiente:

1. Generar una lista anidada que contenga los 6 autos. (Utilizar esta lista para las solicitudes siguientes).

   La solución debe estar dentro del programa `listas_uno.py`.

2. Utilizar la función construída en el `Desafío - Velocidad` para obtener la media entre los 6 autos para cada uno de los campos numéricos. Puede hacer uso de un loop.

   La solución debe estar dentro del programa `listas_dos.py`.

3. Generar un loop que muestre en pantalla aquellos autos cuyo segundo campo (el número flotante) es mayor al de la media de todos los autos.

   La solución debe estar dentro del programa `listas_tres.py`.

4. Utilizar los valores booleanos de cada lista (penúltimo campo) para mostrar en pantalla **el nombre** del auto correspondiente, en caso de que este valor sea True.

   La solución debe estar dentro del programa `listas_cuatro.py`

5. Generar una lista con los valores flotantes (segundo campo) que sean mayores que el promedio, utilizando una operación funcional para filtrar estos datos.

   La solución debe estar dentro del programa `listas_cinco.py`.

Desafío - Asociar

Ahora necesitamos asociar cada registro de velocidad con la distancia recorrida. Para ello, se debe implementar la función `zip`. Ésta debe permitir asociar elementos entre dos listas (una de velocidades, y otra de distancias).

La solución debe estar dentro del programa `asociar.py`.

Ejemplo:

```
lista_1 = [1, 2, 3]
lista_2 = ['A', 'B', 'C']

for i in zip(lista_1, lista_2):
    print(i)

# Retorno
# (1, 'A')
# (2, 'B')
# (3, 'C')
```

```
velocidad = [4, 4, 7, 7, 8, 9, 10, 10, 10,
             11, 11, 12, 12, 12, 12, 13, 13,
             13, 13, 14, 14, 14, 14, 15, 15,
             15, 16, 16, 17, 17, 17, 18, 18,
             18, 18, 19, 19, 19, 20, 20, 20,
             20, 20, 22, 23, 24, 24, 24, 24, 25]

distancia = [2, 10, 4, 22, 16, 10, 18, 26, 34,
             17, 28, 14, 20, 24, 28, 26, 34, 34,
             46, 26, 36, 60, 80, 20, 26, 54, 32,
             40, 32, 40, 50, 42, 56, 76, 84, 36,
             46, 68, 32, 48, 52, 56, 64, 66, 54,
             70, 92, 93, 120, 85]
```

Con esta función, se le solicita que cuente todas las observaciones que cumplan por lo menos con una de las siguientes condiciones:

- Velocidad bajo el promedio
- Velocidad bajo el promedio y distancia sobre el promedio.
- Velocidad sobre el promedio
- Velocidad sobre el promedio y distancia bajo el promedio.

tip: Puede contar las ocurrencias declarando contadores **fuera del loop**.