**Carpeta: PYx03**

**BLOQUE 1: LISTAS Y TUPLAS**

1. **0-pendientes.py:** crear y modificar una lista

Estás construyendo una lista de tareas para hoy. Crea una lista vacía y añade las siguientes tareas:

* Estudiar Python
* Ir al Gym
* Leer 30 minutos

Luego

* Inserta al inicio la tarea 'Desayunar'
* Elimina la tarea 'Leer 30 minutos'
* Imprime cuantas tareas quedan

Limites

* No uses **input** (hazlo todo desde código)
* Usa sólo métodos de listas (**append, insert, remove, len,** etc)

1. **1-duplicar.py:** Usar comprensión de listas y bucles

Dada una lista de números del 1 al 10, crea una nueva lista donde cada número esté duplicado (ej: 2, 4, 6)

* Debes usar **comprensión de listas** para crearla
* Imprime la lista original y la nueva

Limites

* No uses funciones externas
* No modifiques la lista original

1. **2-coordenadas.py:** Crear y acceder a tuplas

Representa las coordenadas de 3 puntos en un plano con tuplas: **(x, y)** para cada punto. Luego imprime solo las coordenadas **x** de todos.

Limites

* No puedes usar listas
* Solo debes usar tuplas y bucles

**BLOQUE 2: DICCIONARIOS**

1. **3-contactos.py:** Crear y modificar diccionarios

Crea un diccionario llamado **agenda** con al menos 3 contactos, donde:

* La clave sea el nombre del contacto
* El valor sea su número de teléfono

Luego

* Agrega un contacto nuevo
* Cambia el número de un contacto existente
* Elimina un contacto

Limites

* No usar **input**
* Usa **del**, asignación directa y **.update()** si quieres

1. **4-contar\_palabras.py:** Contar repeticiones con diccionarios

Dada una frase fija (ej: **“hola hola mundo esto es Python mundo mundo”**), crea un diccionario que cuente cuántas veces aparece cada palabra

Limites

* Usa **.split()** para separar palabras
* Usa **in** para verificar si la palabra ya existe en el diccionario

1. **5.lista\_dict.py:** Usar **zip()** para construir un diccionario

Teniendo 2 listas:

nombres = [“Ana”, “Luis”, “Carlos”]

edades = [25, 32, 19]

Crea un diccionario con cada nombre como clave y su edad como valor

Limites

* Debes usar **zip()** y **dict()**
* No puede usar bucles

**BLOQUE 3: FUNCIONES**

1. **6-jaja\_saludos.py:** Definir funciones con parámetros

Escribe una función **saludar(nombre)** que imprima:

*Hola, <nombre>. ¡Bienvenido a Python!*

Limites

* No usar **input**
* Sólo crea la función y el llamado a la misma

1. **7-sumar\_lista.py:** Recibir una lista como parámetro

Escribe una función **sumar\_lista(lista)** que reciba una lista de números y devuelva la suma total.

Limites

* No usar **sum()**, hazlo con un bucle
* Retorna el resultado como un número flotante, no lo imprimas

1. **8-calificaciones.py:** Función con entrada y salida de diccionarios

Crea una función **analizar\_calificaciones(calificaciones)** que reciba un diccionario con alumnos y sus notas (ej: **{"Ana": 4.5, "Luis": 3.7}** ) y devuelva:

* El promedio de cada alumno
* El número de aprobados (**nota >= 30**)

**Salida:** otro diccionario con estas métricas

Limites

* No usar librerías externas
* Usa **.items()** y **return**