



Ejercicios Avanzados – Python (Parte 1)

Prof. Juan Ignacio Bonini

1. Función: Verificar si un número es primo

Instrucciones:

Escribe una función `es_primo(n)` que reciba un número entero y devuelva `True` si el número es primo, y `False` en caso contrario.

2. Función + Loop: Mostrar todos los primos del 1 al 100

Instrucciones:

Usando la función del ejercicio anterior, crea otra función que recorra los números del 1 al 100 e imprima solamente los números primos.

3. Mutabilidad: Modificar una lista dentro de una función

Instrucciones:

Crea una función `agregar_elemento(lista, valor)` que reciba una lista y un valor, y lo agregue al final.

Probá pasarle una lista y fijate si se modifica afuera de la función. ¿Por qué pasa eso?

4. Copias: Prevenir efectos colaterales con listas

Instrucciones:

Modificá el ejercicio anterior para que la función no modifique la lista original.

Tip: usá `lista.copy()` dentro de la función.

5. Función con bucle `while`: Menú interactivo

Instrucciones:

Escribe una función `menu()` que muestre estas opciones:

1. Saludar
2. Mostrar hora actual
3. Salir

Debe repetirse hasta que el usuario elija salir (3). Usá el módulo `time` para mostrar la hora.

6. Función acumuladora: Promedio hasta que se escriba "fin"

Instrucciones:

Escribí una función `promediar()` que:

- Pida al usuario que ingrese números uno por uno.
- Termine cuando el usuario escriba "`fin`".
- Devuelva el promedio de los valores ingresados.

7. Función que filtra: Números positivos

Instrucciones:

Crea una función `filtrar_positivos(lista)` que reciba una lista de números y retorne una nueva lista con solo los positivos.

No modifiques la lista original.

8. Función con `input()` y `try/except`: Validador de enteros

Instrucciones:

Crea una función `pedir_entero(mensaje)` que:

- Muestre un mensaje al usuario con `input()`.
 - Devuelva el número solo si el usuario ingresó un entero válido.
 - Si no, muestre un mensaje de error y vuelva a pedirlo (usar `while`).
-

9. Mutabilidad avanzada: Lista de listas

Instrucciones:

Dada una lista de listas como `[[1, 2], [3, 4]]`, creá una función que:

- Agregue un número a todas las sublistas.
 - Explicá qué pasa si se hace una copia superficial (`[:]`) vs una copia profunda (`copy.deepcopy()`).
-

10. Funciones y lógica: Contador de vocales

Instrucciones:

Escribí una función `contar_vocales(texto)` que reciba una cadena y devuelva un diccionario con la cantidad de veces que aparece cada vocal.

Ejemplo:

```
contar_vocales("Hola mundo")
{'a': 1, 'e': 0, 'i': 0, 'o': 2, 'u': 1}
```