

## Práctico de Repaso

En este práctico, deberá resolver todos los ejercicios de modo artesanal, sin utilizar comandos que ya dé el propio lenguaje de Python, como `reverse`, `count`, `split`, `zip`, `shuffle`, etc. Sigue siendo lo más importante y foco de la asignatura desarrollar y/o mejorar vuestra capacidad de razonamiento.

Este práctico hará fuerte hincapié en el uso de bucles, archivos y algoritmización.

SIEMPRE considere organizar segmentos de código en funciones o procedimientos que hiciesen el código más ordenado, legible, re utilizable y lindo.

1. Crear una función que determine si un número es par o no.
2. Crear una función que retorne la suma de todos los valores dentro de una lista de 8 números que usted puede inventar.
3. Dada una frase ingresada por el usuario, mostrar en pantalla, las letras de la misma, una por línea. Luego mostrar solo las letras en posiciones pares, una por línea. Y finalmente, mostrar las letras en posiciones impares, una por línea.
4. Dada una frase ingresada por el usuario, determinar cuantas palabras de longitud par contiene. (*La frase "hola Frodo, y Bilbo?" contiene 1 sola palabra de longitud par*)
5. Obtener un cuento corto, desde un archivo de texto llamado `cuento.txt` (*Busque algún párrafo o cuento corto clásico y guardelo en un archivo de texto con dicho nombre*). Realizar el siguiente análisis y mostrar en pantalla un informe de:
  - a) Cuantas palabras contiene en total.
  - b) Cuantas frases contiene en total. (*¿Cómo deberá distinguir una frase de otra?*)
  - c) Cuantos nombres propios o nombres de lugares hay en total.
  - d) El total de vocales que contiene ese cuento.
  - e) Cuantas palabras de longitud par existen.
  - f) Cuantas palabras de longitud impar contiene.
6. Pida al usuario una frase e inviertala. Implemente al menos 3 modos diferentes de lograrlo, sin utilizar el comando `reverse`.
7. Implementar una función que reciba dos vectores y retorne el producto escalar de ambos.
8. Pedir números al usuario y almacenarlos en una lista, hasta que el usuario ingrese la letra `q` para salir.
9. Implementar una función llamada *signos* que reciba un vector de números enteros, y retorne un vector de 1 y -1, según si el número en cada posición es positivo o negativo.
10. ¿Qué diferencia hay entre listas y arreglos? Explique.
11. Haga un listado de comandos básicos de la librería Numpy, relacionados a crear vectores y matrices. El resto de las operaciones de Numpy que no sean específicos para la creación de arrays, no serán utilizados. Se deberán utilizar bucles para manipular los valores de las matrices.

12. Dada una matriz de 10 x 10 retornar la suma de todos sus valores. Utilizar Numpy.
13. Dada una matriz de 6 x 6 valores numéricos, implementada de dos modos, con listas anidadas, y con arrays de Numpy, realice un programa que sume los elementos de la diagonal y muestre el resultado. Los valores iniciales deben ser puestos aleatoriamente. (Repase cómo utilizar random)
14. Generar la siguiente matriz e imprimirla en pantalla, tanto como dejarla almacenada en un archivo de texto llamado *matriz.txt*:

---

```
1 0 0 0 0 0
0 2 0 0 0 0
0 0 3 0 0 0
0 0 0 4 0 0
0 0 0 0 5 0
0 0 0 0 0 6
```

---

### Resumen - Tarea a entregar aparte

15. Realice manuscrito en su cuaderno, una síntesis de comandos con una breve explicación de su función, o cualquier comentario o aprendizaje que haya tenido en la realización de este práctico. Deberá subirlo al campus. Si son varias páginas, arme un archivo PDF y suba un link.