

# Relación de ejercicios sobre listas

Introducción a la programación para Ciencias de Datos

Para generar datos aleatorios, con los que probar los ejercicios, usa `generar_lista.py`

1. Escribe una función `sum_nums_lista(numeros)` que sume todos los números de una lista. Compara el tiempo entre usar o no **range**
2. Escribe una función `contar_numeros_impares(numeros)` que cuente la cantidad de número impares que hay en una lista.
3. Escribe una función `numeros_pares(numeros)` que devuelva los números pares que hay en una lista.
4. Escribe una función `combinar_listas(l1, l2)` que devuelva una lista que esté formada por todos los elementos de l1 y a continuación todos los de l2. Por ejemplo `combinar_listas([1, 2, 8], [5, 10])` devolvería `[1, 2, 8, 5, 10]`
5. Escribe una función `mezclar(la, lb)` que dadas dos listas ordenadas devuelva una lista conteniendo los elementos de ambas listas ordenados de forma ascendente.
6. La traspuesta de una matriz se obtiene intercambiado filas y columna. Escribe una función que devuelva la traspuesta de una matriz.
7. Escribe una función `contar_letras(palabra)` que tome una palabra como argumento y devuelva una lista de pares en la que aparece cada letra junto con el número de veces que aparece esa letra en la palabra. Por ejemplo, `contar_letras('patata')` devuelve `[('a', 3), ('p', 1), ('t', 2)]`.
8. Escribe una función `eliminar(l1, l2)` que dadas dos listas devuelva una lista en la que estén todos los elementos de l1 que no están en l2.
9. Escribe una función `suma_acumulada(numeros)` a la que se le pase una lista de números y devuelva una lista en la que el elemento i-ésimo se obtiene como la suma de los elementos de la lista entre las posiciones 0 e i. Por ejemplo, para `[1, 2, 3]` sería `[1, 3, 6]`
10. Escribe una función `parejas(lista)` que calcule las parejas distintas de valores que aparecen en una lista.
11. Escribe una función `cadena_mas_larga(cadenas)` a la que se pasa una lista de palabras y que devuelva la palabra más larga.
12. Escribe una función `suma_primer_digito(numeros)` que devuelva la suma de los primeros dígitos de todos los números de la lista que se pasa como argumento.
13. Un vector disperso es aquel que tiene muchos elementos nulos. Para ese tipo de vectores, la representación más adecuada es guardar únicamente los elementos no nulos. Escribe una función `dispersa(v)` a la que se le pase una lista representando un vector disperso y que devuelva el número de elementos del vector junto con una lista de pares (pos, elem) con cada una de las posiciones en las que hay un elemento no nulo y el elemento. Ejemplo: `(1,0,0, 5, 4, 0, 0, 0)` sería `[(0,1), (3,5), (4,4)], 8)`
14. Escribe una función que saque de forma aleatoria todas las cartas de una baraja hasta que quede vacía. Para ello debe usar una lista que tenga inicialmente todas las cartas.  
`sacar_carta(baraja)`