

Guía Programación

A resolver en Python Notebook

De resolver algún ejercicio utilizando una librería, documentar los métodos utilizados y resolver de otra forma aparte sin uso de librerías salvo que se indique lo contrario en el ejercicio.

- [Guía Programación](#)
 - [A resolver en Python Notebook](#)
 - [Parte 1: Funciones](#)
 - [Ejercicio 1](#)
 - [Ejercicio 2](#)
 - [Ejercicio 3](#)
 - [Ejercicio 4](#)
 - [Ejercicio 5](#)
 - [Ejercicio 6](#)
 - [Ejercicio 7](#)
 - [Ejercicio 8](#)
 - [Ejercicio 9](#)
 - [Ejercicio 10](#)
 - [Parte 2: Archivos](#)
 - [Ejercicio 11: Pandas](#)
 - [Ejercicio 12: PIL](#)
 - [Parte 3: Numpy](#)
 - [Ejercicio 13](#)
 - [Ejercicio 14](#)
 - [Ejercicio 15](#)
 - [Ejercicio 16](#)
 - [Ejercicio 17](#)
 - [Ejercicio 18](#)
 - [Ejercicio 19](#)
 - [Ejercicio 20](#)
 - [Bonus](#)

Parte 1: Funciones

Ejercicio 1

Escribir una función que reciba una lista de números y devuelva una lista con los números pares.

Ejercicio 2

Escribir una función que tome una lista de números y devuelva el numero mas grande. Si la lista está vacía, devolver 0.

Ejercicio 3

Escribir una función que tome una lista de números y devuelva el numero mas chico. Si la lista está vacía, devolver 0.

Ejercicio 4

Escribir una función que tome una lista de números y devuelva el promedio de los números. Si la lista está vacía, devolver 0.

Ejercicio 5

Escribir una función que tome una lista de números y devuelva el mayor número que no sea divisible entre otro número. Si la lista está vacía, devolver 0.

Ejercicio 6

Escribir una función que tome una lista de listas de la forma `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]` y devuelva una lista con los números de todas las listas: `[1, 2, 3, 4, 5, 6, 7, 8, 9]`. Si la lista está vacía, devolver una lista vacía.

Bonus: Resolver con una comprensión de listas.

Ejercicio 7

Escribir una función que tome una lista de listas de la forma `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]` y devuelva una lista con el mayor numero de cada

sublista: [3, 6, 9] .

Bonus: Resolver con una comprensión de listas.

Ejercicio 8

Escribir una función `es_vocal` que tome un caracter y devuelva `True` si es una vocal y `False` si no lo es. Sugerencia: usar la función `ord` de Python.

Ejercicio 9

Usando la función `es_vocal` implementada en el ejercicio 8, escribir una función que tome un string y devuelva una lista con los caracteres que no son vocales. Sugerencia: usar la función `map` de Python.

Ejercicio 10

Escribir una función que tome una lista de números y los normalice a un rango dado.

Ejemplo:

```
normalizar([1, 2, 3, 4, 5], 0, 10)
# [0.1, 0.2, 0.3, 0.4, 0.5]

normalizar([2, 4, 6, 8], 0, 8)
# [0.25, 0.5, 0.75, 1.0]

normalizar([30, 60, 90, 120, 150], 30, 150)
# [0.0, 0.25, 0.5, 0.75, 1.0]
```

Parte 2: Archivos

Ejercicio 11: Pandas

Leer el archivo `usuarios.csv` y obtener un `DataFrame` con los datos.

- Obtener una Serie con los nombres de todos los usuarios.
- Obtener una Serie con los nombres de todos los usuarios que tengan edad menor al promedio de edad.

- c. Obtener el promedio de edad de todos los usuarios.
- d. Graficar las edades usando `matplotlib.pyplot`
- e. Dada una edad, obtener el nombre de todos los usuarios que tengan esa edad.
- f. Obtener los nombres y edades de los usuarios que sean mas altos que el promedio de usuarios.

Ejercicio 12: PIL

Leer la imagen `imagen.png` .

- a. Obtener la cantidad de pixeles blancos de la imagen. (Usar la función `getpixel` de PIL).
- b. Implementar una función que tome una imagen y devuelva un diccionario con la cantidad de ocurrencias de cada color en la imagen.
- c. Graficar los valores de los colores mas ocurrentes de la imagen usando `matplotlib`. En el eje x se debe mostrar el color y en el eje y la cantidad de ocurrencias. Sugerencia: usar la función `hist` de `matplotlib.pyplot`
- d. Voltrear la imagen. (Usar la función `transpose` de PIL)
- e. Resizear la imagen a un tamaño de $100 * 100$. (Usar la función `resize` de PIL)
- f. Convertir la imagen a escala de grises. (Usar la función `convert` de PIL)
- g. Cropear la imagen a una región rectangular de $100 * 100$. (Usar la función `crop` de PIL)
- h. Rotar la imagen 90° grados. (Usar la función `rotate` de PIL)
- i. Aplicar una máscara a la imagen.
- j. Guardar alguna de las imágenes obtenidas en los puntos "d" en adelante en un archivo `'nueva_imagen.png'`. (Usar la función `save` de PIL)

Parte 3: Numpy

Ejercicio 13

Crear una matriz a partir de una lista. Luego ver la forma de la matriz usando `shape`. (Usar la función `array` de Numpy)

Ejercicio 14

Crear una matriz a partir de una lista de listas. Luego ver la forma de la matriz usando `shape`. (Usar la función `array` de Numpy)

Ejercicio 15

Comparar dos matrices elemento a elemento. (Usar la función `equal` de Numpy)

Ejercicio 16

Crear una función que devuelva una matriz de numpy de $n \times m$ dimensiones pasadas por parámetro con valores aleatorios entre 0 y 1 en sus posiciones. Sugerencia: usar la función `random.random` de numpy.

Ejercicio 17

Utilizando la función anterior, crear una matriz de numpy de $n \times m$ dimensiones pasadas por parámetro con valores aleatorios entre 0 y k en sus posiciones. Sugerencia: usar la función `multiply` de numpy.

Ejercicio 18

Tomando una matriz de forma (n) , crear una matriz de forma (\sqrt{n}, \sqrt{n}) . Ejemplo: Si la matriz es de forma (4) , la matriz debe ser de forma $(2,2)$. Bonus: Especificar que sucede si la matriz tiene de forma (n) a un numero que no es un cuadrado perfecto.

$$[1, 2, 3, 4] \implies [[1, 2], [3, 4]]$$

Ejercicio 19

Tomando una matriz de forma (n, n) , crear una matriz de forma (n^2) .
Ejemplo: Si la matriz es de forma $(3, 3)$, la matriz debe ser de forma (9) .

$$[[1, 2, 3], [4, 5, 6], [7, 8, 9]] \implies [1, 2, 3, 4, 5, 6, 7, 8, 9]$$

Ejercicio 20

Crear una función que tome una matriz de numpy y devuelva una matriz de numpy con los valores absolutos de los elementos de la matriz original.

Bonus

- a. Generar con Numpy un archivo `usuarios.csv` para ser usado en el ejercicio 11.
- b. Resolver el ejercicio 12 usando funciones de numpy y opcionalmente OpenCV.