

1. Ejercicios didácticos

- 1) Dados los siguientes números, convertirlos a base 10:
 - a) 101010 en BCD
 - b) 100110011 en complemento a 2
- 2) Escogiendo un número natural impar cualquiera en base 10, hacer lo siguiente:
 - a) representar el número en BCD
 - b) El negativo del número en signo y magnitud
 - c) El negativo del número en complemento a 2
 - d) Multiplicar por 2 el número original y representarlo en BCD
 - e) Hacer la división entera del número entre 2 y representarlo en BCD
 - f) En la ventana de comandos de python, usar 3 operadores diferentes para multiplicar y dividir enteramente por 4 el número
- 3) Hacer una lista de diccionarios que guarden información sobre alumnos, con su nombre, apellidos y número de matrícula. A partir de eso:
 - a) ordenar la lista según sus campos.
 - b) añadir a cada elemento de la lista un campo booleano de repetidor, que indica si han repetido curso. Si ninguno lo ha hecho, que salga por pantalla un mensaje indicándolo. Si alguno sí, que salgan por pantalla los repetidores y si todos son repetidores, que salga por pantalla un mensaje de “curso de repetición”
 - c) Añadir un campo a cada alumno con la fecha de matriculación en el centro
 - d) Crear un nuevo alumno, comprobar que no existe por su nombre y apellidos y añadirlo a la lista en tal caso
 - e) Sacar un alumno de la lista si este se desmatricula buscándolo por su número de matrícula
- 4)
- 5) Crear una clase vector que tenga atributos (x, y, z) y otra clase triángulo que contenga 3 elementos de tipo vector (p1, p2, p3).
 - a) En el constructor de triángulo, debemos generar un atributo más que sea la normal del plano del triángulo y normalizarla (que su módulo sea 1)
 - b) Generar un método en cada clase que visualice sus atributos
 - c) crear un método en la clase vector que normalice el vector
 - d) crear un método que invierta un vector ([1, 2, 3] -> [-1, -2, -3])

2. Entregables obligatorios

- 1) Crea un programa que dados dos parámetros r y h, calcule el perímetro P de una circunferencia de radio r, el área A de un círculo del mismo radio y el volumen V de un cilindro de radio r y altura h. El programa debe mostrar los resultados por pantalla.
- 2) Crea un programa que descubra si un número es par:
 - a) Utilizando cualquier operador
 - b) Sin usar ninguno de los operadores de división ('/', '//', '%', etc.)
- 3) Crea un programa que divida un número entre 2 sin usar ninguno de los operadores de división.
- 4) Crea un programa que genere una lista de 1 a 100 y después encuentre los números primos que se encuentran ahí, incluirlos en una lista y mostrarlos por pantalla.

- 5) Crea un programa que limpie los espacios innecesarios en la siguiente string:

```
"  Hola Mundo,      necesito  una limpieza  .  "
```

- 6) Crea un programa que genere una lista de 100 números aleatorios y los ordene de menor a mayor. Usando el módulo random

```
import random
```

- 7) Dada una lista de números enteros (int_lst), escriba los siguientes bucles en una sola línea:

```
for i, elem in enumerate(int_lst):
    if elem > 20:
        int_lst[i] = 'mayor'
```

```
for i, elem in enumerate(int_lst):
    if elem > 20:
        int_lst[i] = 'mayor'
    else:
        int_lst[i] = 'menor'
```

- 8) Corrija el siguiente código con la lista anterior:

```
i = 0
for i in range(int_lst):
    if int_lst[i]:
        if int_lst[i] > 20:
            int_lst[i] = 'mayor'
        if int_lst[i] <= 20:
            int_lst[i] = 'menor'
        else:
            break
```

- 9) Una pequeña empresa tiene 6 clientes, los cuales vienen identificados por el nombre, los apellidos, un código de identificación o ID aleatorio de 6 números y es de tipo string, el total de dinero gastado, el número de pedidos realizados en total y una lista de los pedidos. Por otro lado, cada elemento de una lista de pedidos contiene un código identificador del pedido, la fecha en la que se realizó, el código de cliente que realizó el pedido y el importe total que se pagó. Sabiendo esto:

- Genere un tipo de dato que contenga todos elementos necesarios para un pedido y otro para un cliente.
- Cree una lista de clientes (sin rellenar el campo de pedidos) y otra de pedidos (con al menos 3 clientes y 2 pedidos por cliente)
- Cree un programa que rellene el campo de pedidos de cada cliente con los elementos correspondientes de la lista de pedidos.
- Cree un programa que pueda ordenar los clientes por los siguientes campos:
 - ID
 - Nombre
 - Apellidos
 - Número de pedidos

v) Total de dinero gastado

10) Dado el ejercicio anterior, cree las siguientes funciones:

- a) Una función que rellene el campo de pedidos de todos los clientes
- b) Una función de ordenación de los clientes por los campos dados en el ejercicio anterior
- c) Una función que calcule el total de pedidos y el dinero gastado por un cliente si estos campos no estuviesen rellenos
- d) Con la función anterior, crear una función que añada un nuevo pedido a la lista de pedidos y al cliente correspondiente y actualice los campos necesarios de dicho cliente.
- e) Una función que añada un nuevo cliente pasándole solamente el nombre y los apellidos.
- f) Una función que visualice un cliente con el siguiente formato

Nombre: Pepe

Apellidos: García González

ID: 000345

Número de pedidos: 4

Dinero gastado: 94.57€

Fecha del último pedido: 4.8.2022

11) Escriba un programa que cree un archivo .txt y escriba sobre él una serie de hipervínculos a youtube. Genere además una funcionalidad que permita leer dicho archivo y mostrar su contenido por pantalla.

12) cree un programa que en el cual haya una clase animal, que contenga los siguientes atributos y métodos:

- Reino
 - Regiones en las que habita
 - Población
 - Un constructor que rellene los campos anteriores
 - Un método por el cual se imprima por pantalla el sonido del animal
- a) Cree la clase animal y su constructor
 - b) Cree la clase mamífero que herede de animal, por la cual se añaden los atributos:
 - Filo
 - c) Cree dos clases, perro y gato, que hereden de mamífero y que sobrescriban el sonido realizado por el animal. Y que además añada
 - Nombre
 - Edad
 - Género
 - d) Escriba una funcionalidad por la cual se genere una jauría, que contiene varios perros, y otra que muestre por pantalla los nombres de los animales de la jauría, su edad y su género
 - e) Una funcionalidad que guarde los datos de los miembros de la jauría en un archivo .txt y otro que, dado un .txt con el mismo formato, cree una jauría nueva

13) Cree un programa que compruebe la existencia de una carpeta, si no existe la crea. Después, con la carpeta creada, debe guardar el archivo del ejercicio 11) y, a su

- vez, generar una carpeta donde descargar los vídeos. Si esa carpeta tiene archivos en su interior, borrar la carpeta y su contenido.
- 14) Con la funcionalidad anterior, comprimir el contenido de la carpeta de vídeos en un archivo formato .zip, guardando dicho archivo en otra carpeta creada para guardar los archivos comprimidos
 - 15) Utilizando el módulo YouTube, instalándolo en el entorno si es necesario, descargar los vídeos del archivo de links.
 - 16) Utilizando la librería Numpy, cree un programa que calcule el volumen y la superficie de una esfera y al que se le puede pasar el radio o el diámetro.
 - 17) Cree un decorador que de el nombre por pantalla de una función antes de ejecutarla. Luego, cree un timer (utilizando el módulo functools), que calcule el tiempo de ejecución de una función. Finalmente cree una función cualquiera y decórela con los decoradores anteriores.
 - 18) Cree un programa en JupyterLab que llame a una librería que contenga las siguientes funciones:
 - a) Abra una Imagen de periódico y la visualice en JupyterLab
 - b) Cree una lista de unos elementos llamados 'textbox' que contengan, de momento, un campo 'type' y un campo llamado 'bbox', que a su vez contenga los siguientes elementos: 'xmin', 'ymin', 'xmax' e 'ymax', que son coordenadas ('x' es ancho e 'y' es alto).
 - c) Una función que dadas 4 coordenadas (xmin, ymin, xmax e ymax), cree un textbox y los añada a la lista. Si uno de los parámetros no se pasa, se pone automáticamente a 0. A su vez, genere las siguientes funciones auxiliares:
 - i) Una función que cambien las coordenadas mínima y máxima si estas están invertidas (ej.: xmin = 300., xmax = 0. están invertidas)
 - ii) Una función que cambie las coordenadas negativas a 0. y las que son mayores que el ancho o alto de la imagen
 - d) Una función que visualice los elementos de la lista de textboxes sobre la imagen como output de una celda
 - e) Una función decorador que calcule el tiempo de ejecución de una función y muestre el nombre de la función y el tiempo de ejecución.
 - 19) Dada una carpeta (in_path) que contiene varios archivos (de texto .txt e imágenes .png) y subcarpetas, cree varias funciones que:
 - a) Compruebe la existencia de una carpeta y, si no existe, la crea.
 - b) Copie todos los elementos y subcarpetas de in_path en otra carpeta (tmp_path)
 - c) Cree una lista con los nombres de los archivos de texto.
 - d) Abra uno por uno los archivos, lea las líneas y cambie el primer elemento de cada línea al número 3 y los guarde en la misma carpeta (tmp_path). Si no encuentra el archivo, debe continuar con el proceso, mostrando una por pantalla:

```
ERROR: el archivo no existe o no está en la carpeta especificada
```
 - e) Cree una carpeta, si no existe, llamada "total_datos" donde se guarda un archivo .txt que contiene:

```
Txt: <número de .txt>
Img: <número de .png>
```

- f) Comprima toda la carpeta tmp_path en un archivo .zip y guarde el archivo en otra carpeta llamada out_path.
- 20) Cree un .pdf que guarde un título (ej.: arial 36) y debajo una imagen que abarque buena parte de la página. Después:
- Añada una página en blanco
 - Añada una página en la que se añadan, al menos, 3 líneas de texto normal
 - guarde el pdf
- 21) Cree un lector que lea el texto de la última página del pdf anterior.
- 22) Cree una clase que herede de FPDF con dos métodos que añadan siempre un encabezado con una imagen y un pie de página con el número de la página.
- 23) Dada una base de datos de MongoDB, con dos colecciones, una de clientes y otra de pedidos, hacer lo siguiente:
- crear la base de datos y las colecciones
 - con los mismos pedidos y clientes que en los ejercicios 9) y 10), generar los datos localmente.
 - Inserte uno por uno los clientes y luego todos los pedidos a la vez
 - Lea los pedidos y los clientes de la base de datos, rellene el campo de pedidos correspondiente a cada cliente, así como el dinero gastado en total y el número total de pedidos realizados de cada uno.
 - Actualice los datos del cliente con el apartado anterior
- Los datos de cada cliente deben tener el siguiente formato:
- ```
client = { 'ID': string de 6 caracteres numéricos
 'name':
 'surname':
 'total_orders': int
 'total_amount_spent': float
 'orders': list}

order = {'ID':
 'date': str
 'ID_client': str
 'total_amount': float}
```
- 24) Dado un dataset de imágenes, repartirse entre todos un porcentaje equivalente del mismo, hacer las labels con LabelImg, pasar cada parte al resto y luego en colab, generar las carpetas de train y valid y el archivo data.yaml
- 25)

### 3. Entregables opcionales