PROGRAMACIÓN 1

Gestión de datos de paises

Franco Rios - Fabrizio Simon

Conceptos aplicados

01 Listas

02 Diccionarios

03 Funciones

04 Condicionales

05 Ordenamientos

06 Archivos CSV

01 LISTAS

¿Que son las listas?

Una lista es la estructura de datos más fundamental en Python, permite almacenar una colección ordenada y mutable de elementos. Los elementos pueden ser de diferentes tipos de datos, pueden ser tanto como números, cadenas, o incluso otras listas y diccionarios.

Cada elemento dentro de una lista es accesible mediante posiciones llamadas indice, que siempre comienzan desde cero.

Ejemplo de listas:

Ejemplo práctico:

En el caso de nuestro código “datos_paises” es de la estructura principal que almacena los datasets.
Otro ejemplo de la listas es el .append para agregar paises a la lista principal.

```python
def main():
    """Función principal: Controla el flujo de la aplicación mediante el menú de opciones."""
    nombre_archivo_csv = "paises.csv"
    datos_paises = []
```

```python
# 3. Conversión de tipos
pais = {
    'nombre': datos[0],
    'poblacion': int(datos[1]),
    'superficie': float(datos[2]), # Vuelve a ser float
    'continente': datos[3]
}
paises.append(pais)
```

02 DICCIONARIOS

¿Que son los diccionarios?

Un diccionario es una colección de elementos que almacena en pares clave-valor. Se trata de una estructura no ordenada y mutable. Su principal ventaja es que permite acceder a los datos de forma rápida y por su clave, no por su posición numérica.

Ejemplo de diccionarios:

Ejemplo practico:

De la imagen anterior también se puede llegar a ver un diccionario, donde se agregan los paises con su clave y valor.

```python
# 3. Conversión de tipos
pais = {
    'nombre': datos[0],
    'poblacion': int(datos[1]),
    'superficie': float(datos[2]), # Vuelve a ser float
    'continente': datos[3]
}
```

03 FUNCIONES

¿Que son las funciones?

Las funciones son bloques de código encapsulados y reutilizables diseñados para realizar una tarea específica. Su uso permite la modularización del código, mejorando la legibilidad y facilitando la depuración.

Ejemplo de funciones:

def validar_string()

Un ejemplo del código para validar si se encuentran strings y no números o este vacio

def validar_entero()

Si el programa pide que se ingrese un número entero valida que no se haya ingresado un caracter, etc.

Funciones

```python
def validar_string(mensaje):
    """
    Pide un string, valida que NO esté vacío y que NO contenga números.
    Permite espacios y otros caracteres (ej: "Corea del Sur").
    """
    while True:
        entrada = input(mensaje).strip()

        if not entrada:
            print("❌ La entrada no puede estar vacía.")
        # any(c.isdigit() for c in entrada) revisa si algún caracter es un dígito
        elif any(char.isdigit() for char in entrada):
            print("❌ La entrada no puede contener números.")
        else:
            return entrada # Entrada válida
```

```python
def validar_entero(mensaje):
    """Pide un entero al usuario y lo valida."""
    while True:
        try:
            entrada = input(mensaje)
            return int(entrada)
        # Captura error al ingresar un valor que no sea entero
        except ValueError:
            print("❌ Entrada inválida. Se esperaba un valor numérico entero ! ")
```

04 CONDICIONALES

¿Que son las estructuras condicionales?
Las estructuras condicionales son estructuras de control de flujo (if, elif, else) que permiten que el programa tome decisiones y ejecute diferentes bloques de código según si una o varias condiciones lógicas son verdaderas o falsas.

Ejemplo de condicionales:

```
if opcion == 0:
    print("🖐 Saliendo del sistema de Gestión de Datos de Países... ¡Hasta pronto!")
    break
```

```
elif opcion == 3:
    resultados = fn.filtrar_por_continente(datos_paises)
    if resultados:
        fn.mostrar_paises(resultados)
```

```
else:
    print("ℹ No se encontraron países para ese continente.")
```

El uso de condicionales en este trabajo es fundamental y aca esta el porque:

if

Es la primera opción del código principal.

elif

El usuario al seleccionar la opción 3, el programa ignora todo lo demás y se queda con esa opción.

else

En la misma opción, si no se cumple el elif, el código ejecuta el else.

05 ORDENAMIENTOS

¿Que son los ordenamientos?
Es el proceso de reestructurar una colección de datos en una secuencia lógica (ascendente o descendente) basándose en el valor de una clave específica.

Ejemplo de ordenamientos:

Opción 8

Al usar fn.mostrar_paises, el programa los ordena alfabeticamente:

```
elif opcion == 8:
    print("\n--- LISTADO COMPLETO DE PAÍSES ---")
    fn.mostrar_paises(datos_paises)
```

06 ESTADÍSTICAS

¿Que son las estadísticas básicas?
Las estadísticas básicas son un conjunto de métodos sencillos que describen las características principales de un conjunto de datos.

Ejemplo de estadísticas:

Opción 7

Al usar la opción 7 muestra las estadísticas de cada país y cálcula.

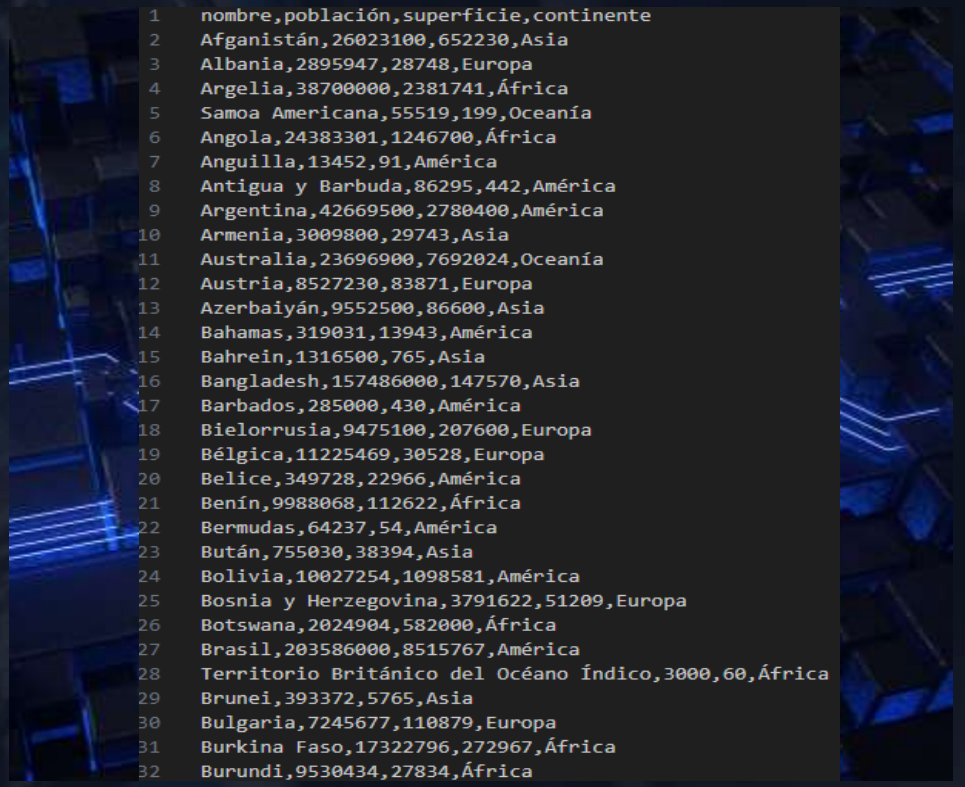
```
elif opcion == 7:
    fn.mostrar_estadisticas(datos_paises)
```

07 ARCHIVOS CSV

¿Que son los archivos CSV?
Los archivos CSV “Comma Separated Values” (Valores separados por coma) son archivos de texto plano que se utiliza para almacenar datos tabulares (filas y columnas)

Ejemplo de CSV

El archivo CSV es fundamental para este proyecto, ya que ahi se encuentran todos los datos guardados de cada país.



```
nombre,población,superficie,continente
Afganistán,26023100,652230,Asia
Albania,2895947,28748,Europa
Argelia,38700000,2381741,África
Samoa Americana,55519,199,Oceanía
Angola,24383301,1246700,África
Anguilla,13452,91,América
Antigua y Barbuda,86295,442,América
Argentina,42669500,2780400,América
Armenia,3009800,29743,Asia
Australia,23696900,7692024,Oceanía
Austria,8527230,83871,Europa
Azerbaiyán,9552500,86600,Asia
Bahamas,319031,13943,América
Bahrein,1316500,765,Asia
Bangladesh,157486000,147570,Asia
Barbados,285000,430,América
Bielorrusia,9475100,207600,Europa
Bélgica,11225469,30528,Europa
Belice,349728,22966,América
Benín,9988068,112622,África
Bermudas,64237,54,América
Bután,755030,38394,Asia
Bolivia,10027254,1098581,América
Bosnia y Herzegovina,3791622,51209,Europa
Botswana,2024904,582000,África
Brasil,203586000,8515767,América
Territorio Británico del Océano Índico,3000,60,África
Brunei,393372,5765,Asia
Bulgaria,7245677,110879,Europa
Burkina Faso,17322796,272967,África
Burundi,9530434,27834,África
```

Fuentes Bibliográficas

Documentación Oficial de Python (La fuente principal):

https://docs.python.org/3/

W3Schools (Tutoriales de Python):

https://www.w3schools.com/python/

Stack Overflow (Para resolución de errores específicos):

https://stackoverflow.com/

Real Python (Tutoriales y artículos en profundidad):

https://realpython.com/

GeeksforGeeks (Ejemplos de código y conceptos):

https://www.geeksforgeeks.org/python-programming-language/

¡GRACIAS POR VER!

Hecho por Franco Rios y
Fabrizio Simon