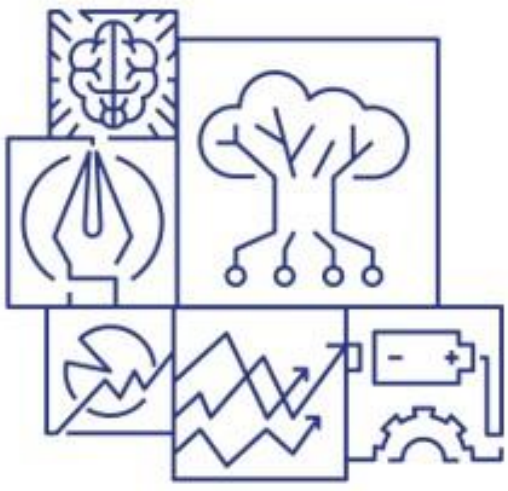
fidélitas
INGENIERÍA EN SISTEMAS DE COMPUTACIÓN

APRENDAMOS A PROGRAMAR

Programación Básica.



Clase 11
Python

Contenido

Introducción

Hasta este momento, en nuestros programas hemos utilizado variables en todas sus formas para almacenar y gestionar la información requerida. Estos elementos han sido suficientes para los ejercicios realizados, pero quizás hemos considerado que sería de utilidad que nuestros programas almacenaran información posterior a su finalización, de tal manera que los datos permanezcan o queden almacenados en algún medio.

La solución al planteamiento anterior es la utilización de archivos.

Concepto

Un archivo es "un conjunto de datos que se encuentran almacenados en un medio de almacenamiento físico." (Trespaderne, 2020)

Características de las variables y los archivos

Las variables tienen las siguientes características:

- Se almacenan en memoria.
- No son accesibles para otras aplicaciones.
- Desaparecen al finalizar el programa.

Los archivos presentan las siguientes facilidades:

- Los datos se conservan aun finalizado el programa.
- Nos permiten compartir datos entre aplicaciones.
- Podemos almacenar datos de distinto tipo.
- Se pueden replicar y mover entre distintos dispositivos de almacenamiento.

Las alternativas de almacenamiento descritas anteriormente se pueden utilizar en conjunto, es decir las **variables** se utilizan en el programa para resolver situaciones y los **archivos** para guardar la información de forma permanente.

Ruta de almacenamiento de los archivos en el disco

El almacenamiento de los archivos en el disco duro del computador es gestionado por el sistema operativo y almacenados en disco, no en la memoria como sucede con las variables.

La ubicación de los archivos se debe indicar según la ruta en la cual estará ubicado el archivo en el disco duro del computador.

Por ejemplo: **c:\misprogramas\miarchivo.txt**

Archivos, nombres y contenido

Los archivos cuentan con un nombre, extensión y contenido.

Es conveniente mencionar que se debe utilizar extensiones adecuadas, de acuerdo con el tipo de datos que almacenará el archivo. Por ejemplo, si se desea que el archivo almacene texto, entonces la extensión a utilizar es .txt. Utilizar extensiones designadas para otro tipo de archivo, confundirá al sistema operativo sobre la aplicación con la que se gestionará el mismo.

Creando archivos en Python

Para el manejo de archivos en Python se requiere una serie de funciones, las cuales se encuentran en una biblioteca llamada os, que debe ser importada al inicio del programa.

Cada vez que se requiere utilizar un archivo debe abrirse mediante la instrucción open, cuya sintaxis es la siguiente:

nombreArchivo = open(“nombreArchivo.ext”,”mododeapertura”)

nombreArchivo se refiere al nombre con el que Python conocerá al archivo dentro del programa. **open** es la instrucción que permite abrir el archivo físico.

nombreArchivo.ext se refiere al nombre real del archivo ubicado en el disco duro u otro dispositivo. **mododeapertura** es una letra que indica para qué se va a utilizar el archivo.

A continuación, se explicará los modos más utilizados para la apertura de un archivo. **a:** Abre el archivo para agregar datos al final. Si el archivo no existe, lo crea. **r:** Abre el archivo para leer los datos que contiene.

W: Abre el archivo para escribir datos. Si el archivo no existe, lo crea. Si existe sobreescribe la información que contenga. **a+:** Abre el archivo para leer y escribir datos en él.

Veamos el siguiente ejemplo, donde se crea un archivo con el nombre "miarchivo.txt", en el cual se almacena un mensaje. Note que al final se debe cerrar el archivo.

```
import os
file = open("miarchivo.txt","w")
file.write("Mi primer texto")
file.write("escrito desde Python")
file.close()
```

Importa las funciones relacionadas con el manejo de archivos

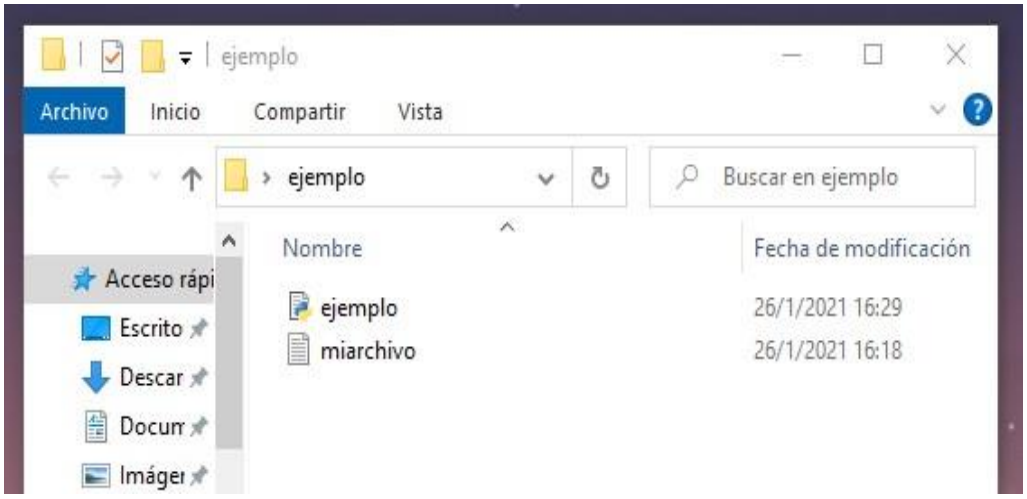
Abre el archivo para escritura

Escribe en el archivo

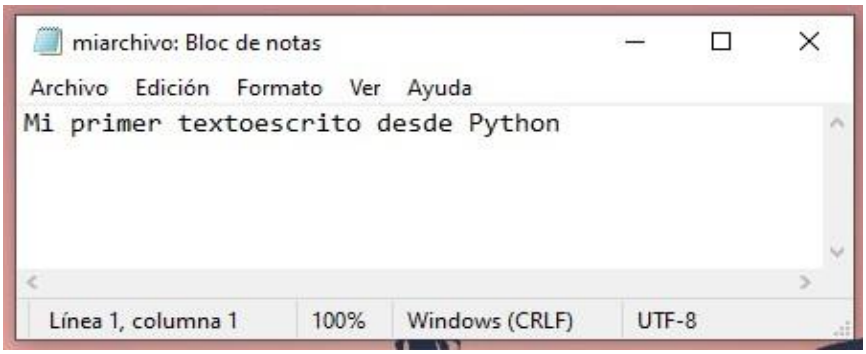
Escribe en el archivo

Cierra el archivo

Esto creará el archivo en la misma ubicación del programa



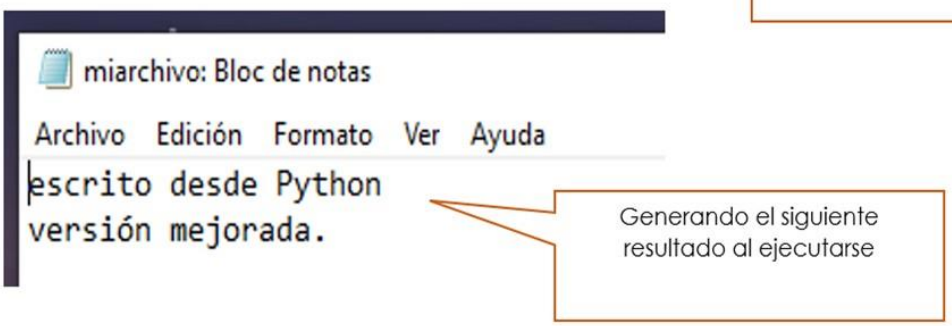
Si abrimos el archivo con el bloc de notas podemos ver que tiene el texto que se indicó en las instrucciones write. Note que, aunque fueron dos instrucciones, el texto queda en una sola línea, lo cual nos indica que cuando se escribe en el archivo el punto de inserción queda ubicado después del último carácter.



Si deseamos que la información se almacene en varias líneas dentro del archivo, debemos utilizar el salto de línea dentro de la instrucción write(), según se muestra a continuación.

```
import os
file = open("miarchivo.txt","w")
file.write("Mi primer archivo\n")
file.write("versión mejorada.")
file.close()
```

El carácter **\n** realiza un cambio de linea



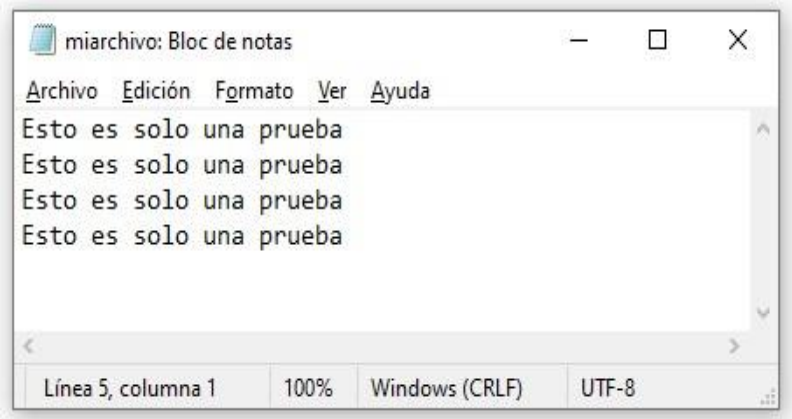
Generando el siguiente resultado al ejecutarse

El siguiente es un ejemplo en el que archivo se abre con el modo append.

```
import os
file = open("miarchivo.txt","a")
file.write("Esto es solo una prueba\n")
file.close()
```

Con este carácter no se sobre escribe el archivo, sino que se adjuntan los datos al final

Para verificar el funcionamiento del append, elimine el archivo creado anteriormente (esto se hace de forma manual desde el explorador de archivos, presionando la tecla Suprimir). Ejecute varias veces el programa y obtendrá un archivo con el siguiente contenido:



Lectura de datos de un archivo

La información almacenada en un archivo se lee mediante la instrucción read ().

```
import os
file = open("miarchivo.txt","r")
mensaje = file.read()
print(mensaje)
```

Establece que el archivo ha sido abierto para ser leído.

Carga todo el contenido en una variable

Mostramos el contenido de esa variable

En algunas ocasiones se requiere guardar los datos conforme el programa los va procesando como se muestra el siguiente ejemplo.

En el siguiente ejemplo, puedes observar tres funciones que realizan operaciones con un archivo. La primera función permite crear un archivo vacío.

La función agregarInformacion almacena la información leída en el archivo indicado en la instrucción open.

La función mostrarInformacion lee e imprime el contenido del archivo.

Ejemplo de operaciones con archivos

```
#se crea el archivo
def crearArchivo():
    file=open("datosCurso.txt","w")
    print("El archivo está listo para grabar informacion!")
    file.close()


#se leen los datos de una persona y se almacenan en el archivo
def agregarInformacion():
    nombreContacto=input("Digite el nombre del contacto:")
    telefono=input("Digite su número telefónico:")
    file=open("datosCurso.txt","a")
    file.write(nombreContacto)
    file.write("\n")
    file.write(telefono)
    file.write("\n\n")
    print("\nLa información fue grabada correctamente!")
    file.close()


#se muestra la información almacenada en el archivo
def mostrarInformacion():
    file=open("datosCurso.txt","r")
    mensaje=file.read()
    print(mensaje)
    file.close()
```

```
#para acceder a cada una de las opciones creamos un menú
def mostrarMenu():
  opc=0
  while(opc!=4):
    opc=int(input("***MENÚ PRINCIPAL***\n\n"+
    "1.Crear archivo\n"+
    "2.Agregar informacion\n"+
    "3.Mostrar información\n"+
    "4.Salir del sistema\n\n"+
    "Digite su opción:"))
    if(opc==1):
       crearArchivo()
    elif(opc==2):
       agregarInformacion()
    elif(opc==3):
       mostrarInformacion()
    elif(opc==4):
       break
    else:
       print("Opción incorrecta!")


  #llamado al menú en la función principal
  if __name__=="__main__":
    mostrarMenu()
```

Ejercicio # 1

La fábrica de Camisetas Deporte y Más ha decidido lanzar una nueva colección de camisetas conmemorativas de la Copa Mundial 2022.

Por esta razón se requiere un programa, que permita controlar el inventario de dichos productos. La información a registrar es: código, detalle, talla, cantidad y precio unitario.

Para la solución debe utilizar archivos, para almacenar los datos. Debe una proveer una forma de mostrar el contenido del archivo.

Tiempo aproximado 30 minutos

Ejercicio # 2

La Escuela El Sol Naciente le ha solicitado un programa que muestre las tablas de multiplicar del 0 al 12, con el fin de que los estudiantes puedan estudiarlas.

El programa debe realizar los cálculos de las tablas utilizando estructuras de repetición y almacenar los resultados en un archivo.

Nota: Es conveniente que cree un menú para controlar el acceso a las opciones de agregar y mostrar.

Tiempo aproximado 40 minutos

Referencias

Joyannes Aguilar, L. (2010). *Fundamentos de Programación*. Madrid, España: Mc. Graw-Hill.