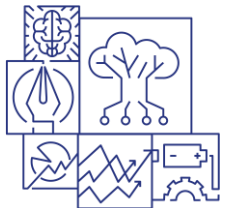


Programación Básica Clase 11

Agenda

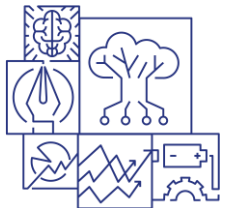
Administración de archivos planos

- Introducción
- Concepto
- Rutas de almacenamiento de los archivos
- Archivos, nombres y contenido
- Creación, lectura y escritura de un archivo
- Ejercicios



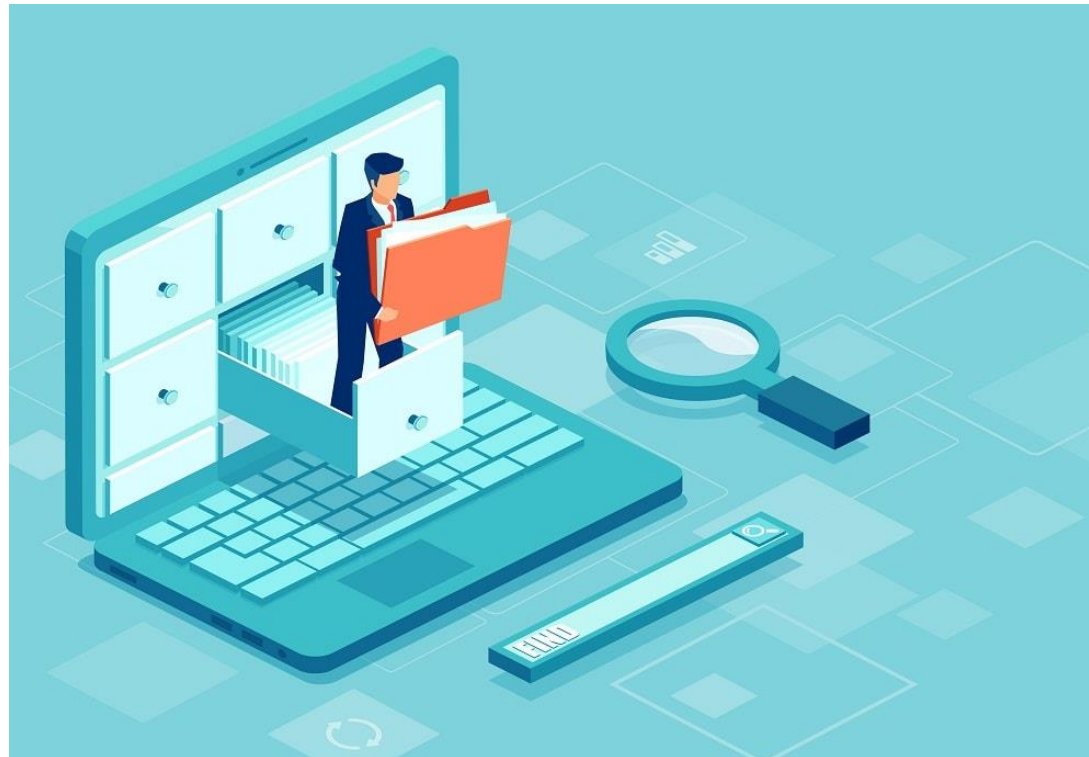
Archivos planos versus variables

¿Quizás hemos considerado que sería de utilidad que nuestros programas almacenaran información posterior a su finalización?
¿ En qué ocasiones?



Concepto de archivo

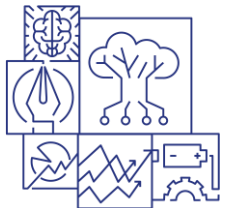
Un archivo es “un conjunto de datos que se encuentran almacenados en un medio de almacenamiento físico.” (Trespaderne, 2020)



Características de los archivos

Los archivos presentan las siguientes características:

- 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.

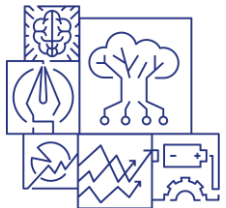


Ruta de almacenamiento de los archivos en el disco

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**

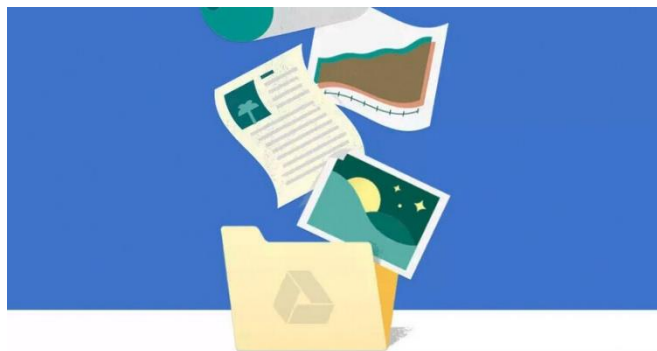


Archivos, nombres y contenido

Los archivos cuentan con un nombre, extensión y contenido.

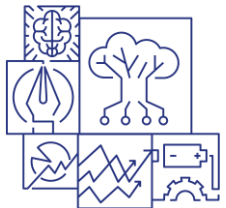
Es conveniente mencionar que se debe utilizar extensiones adecuadas, de acuerdo con el tipo de datos que almacenará el archivo. Por ejemplo, si se desea que el archivo almacene texto, entonces la extensión a utilizar es .txt.

Utilizar extensiones designadas para otro tipo de archivo, confundirá al sistema operativo sobre la aplicación con la que se gestionará el mismo.



Operaciones con archivos

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.



Apertura de un archivo

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

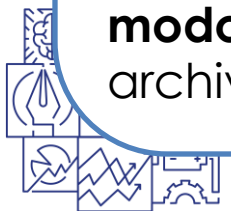
```
nombreArchivo = open("nombreArchivo.ext","mododeapertura")
```

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.

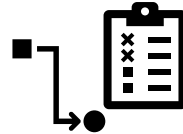


Modos de apertura de un archivo



Los modos más utilizados para la apertura de un archivo son:

- “a”: Abre el archivo para agregar datos al final. Si el archivo no existe, lo crea.

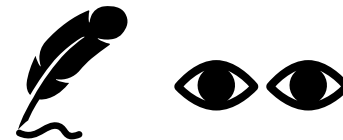


- “r”: Abre el archivo para leer los datos que contiene. 

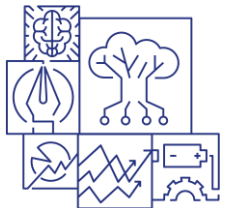
- “w”: Abre el archivo para escribir datos. Si el archivo no existe, lo crea. Si existe sobrescribe la información que contenga.



- “a+”: Abre el archivo para leer y escribir datos en él.



Ahora vamos a crear
un ejemplo con archivos...



Creación y escritura de archivo en Python

La instrucción write nos permite almacenar o escribir información en el archivo.

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

Importa las funcionalidades relacionadas con el sistema operativo

Abre el archivo para escritura

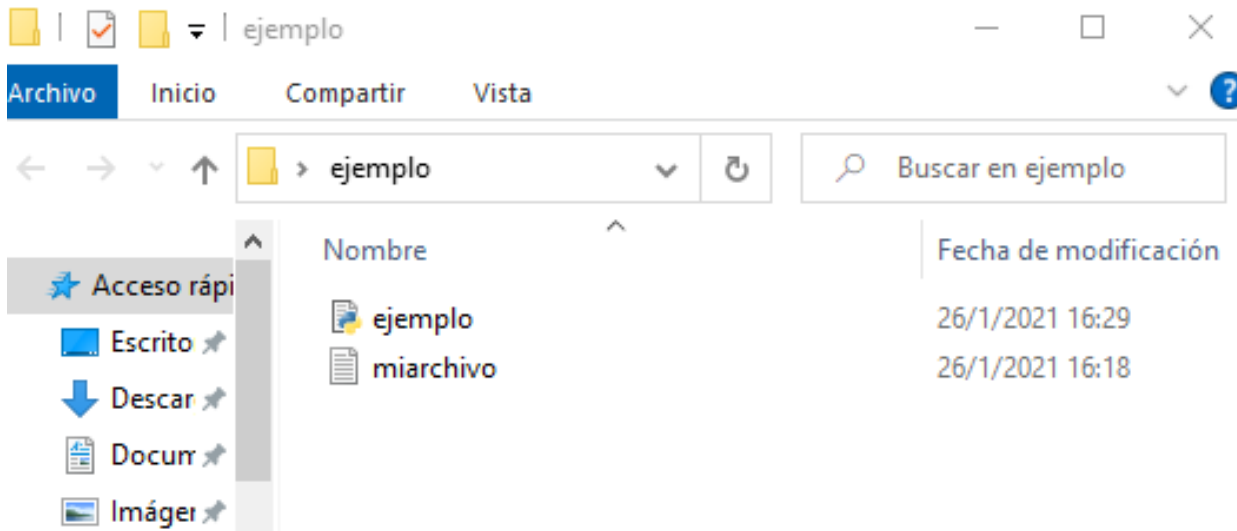
Escribe en el archivo

Escribe en el archivo

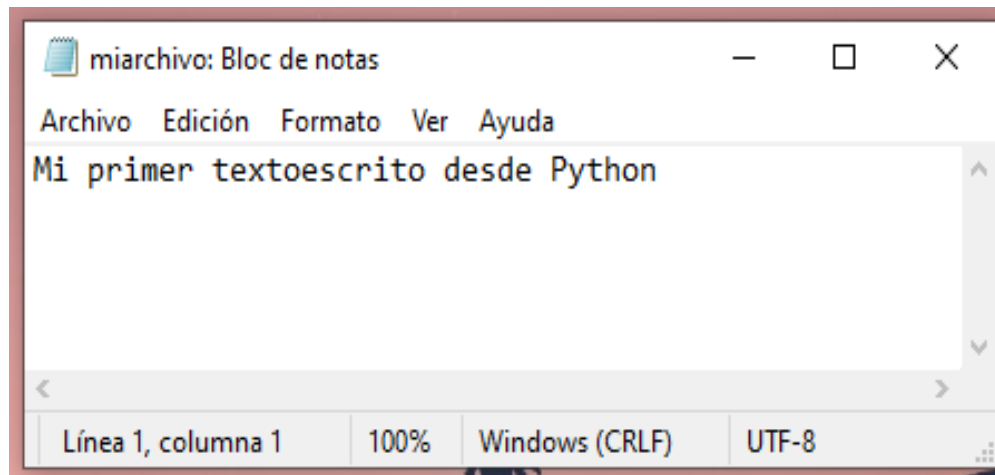
Cierra el archivo y lo libera



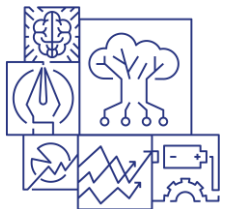
Se crea un archivo en la ubicación indicada



Esto creará el archivo en la misma ubicación del programa.

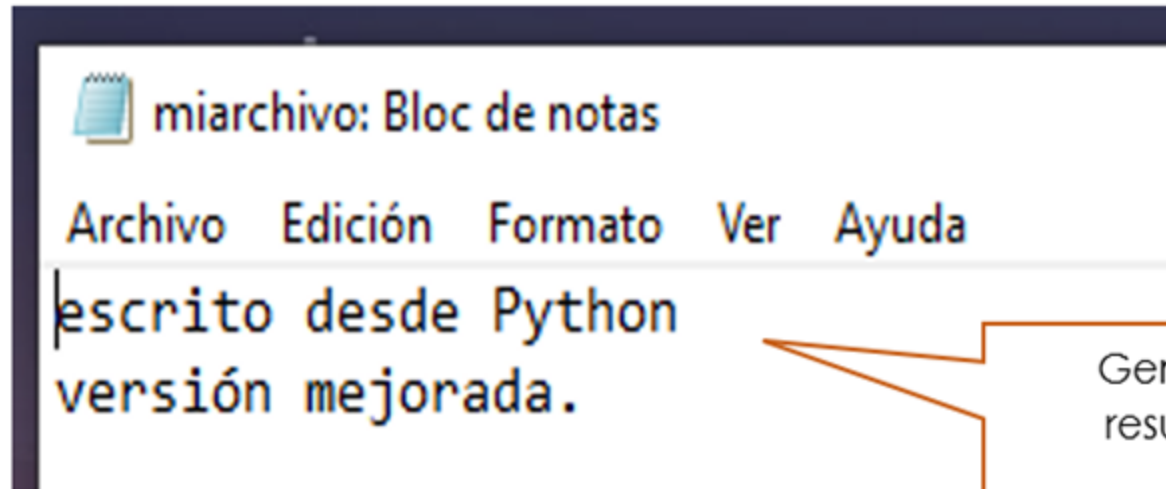


Este será el contenido del archivo.

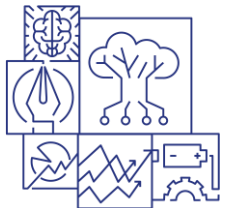


```
import os
file = open("miarchivo.txt", "w")
file.write("Mi primer archivo\n")
file.write("versión mejorada.")
file.close()
```

El carácter `\n` realiza un cambio de línea

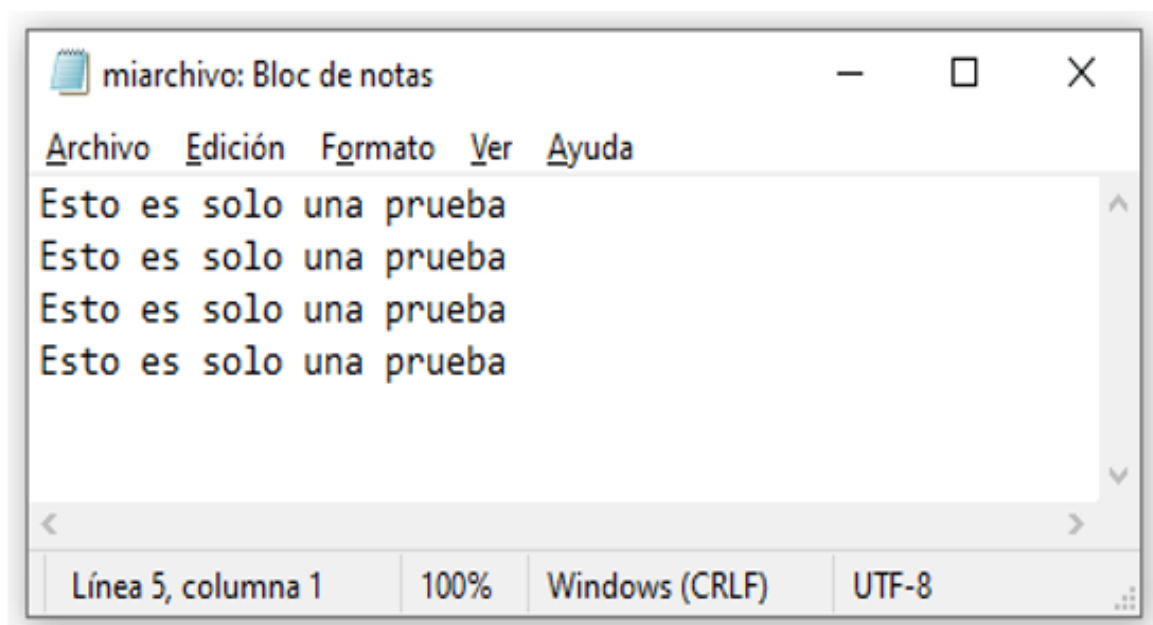


Generando el siguiente resultado al ejecutarse



```
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



Elimine el archivo creado
anteriormente, ejecute varias
veces el programa y obtendrá
un contenido como el siguiente



Lectura de Archivos en Python

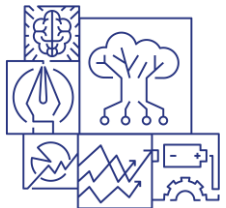
La instrucción read nos permite leer los datos almacenados en el archivo.

```
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

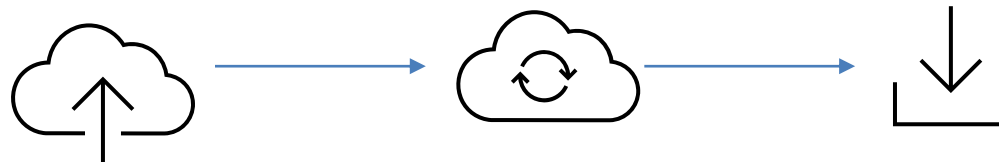


En algunas ocasiones se requiere guardar los datos conforme el programa los va procesando como se muestra el siguiente ejemplo.

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.

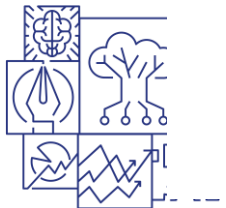


Ejemplo de archivos

```
#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()
```



Ejercicio

1



Desarrolle un programa que controla las notas obtenidas por los estudiantes del curso de Programación Básica.

En un archivo debe registrar la siguiente información: nombre, número de grupo y calificación.

Cree las funciones para agregar y mostrar información.

Es conveniente que el acceso a las funciones se realice a través de un menú.

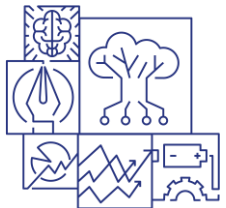
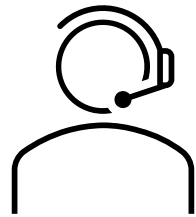


Tiempo aproximado 40 minutos



Con la gestión de archivos hemos logrado que nuestros datos trasciendan a la ejecución de nuestros programas.

Con este tema hemos finalizado los contenidos del curso, ahora invertiremos un poco más de tiempo en afinar estas habilidades en las próximas clases.





¡Nos vemos la próxima semana!

