

Manejo de archivos



Un archivo es una **colección de datos almacenados** en un sistema de almacenamiento (DH, SSD, USB, etc). Permiten a los usuario y a los programas **guardar y recuperar** información de manera organizada y persistente. Los archivos pueden contener diversos tipos de datos, como texto, imágenes, audio, video, código fuente, etc



Los **archivos de texto** contienen caracteres legibles, es posible no solo leer dicho contenido sino también modificarlo usando un editor de texto.



Un **archivo binario** es cualquier archivo que no se pueda interpretar en forma de texto: una imagen, un sonido o incluso un archivo comprimido.



1. **Apertura:** Acceder a un archivo para leer, escribir o modificar su contenido.
2. **Lectura:** Extrae datos de un archivo.
3. **Escritura:** Guardar datos en un archivo, ya sea creando contenido nuevo o modificando el existente.
4. **Cierre:** terminar el acceso a un archivo y liberar cualquier recurso asociado.



Modo de apertura



utilizar el modo
correspondiente para
cada caso

MODO	DESCRIPCIÓN
r	Abre un archivo de texto sólo para lectura
rb	Abre un archivo binario sólo para lectura
r+	Abre un archivo de texto para escritura y lectura
w	Abre un archivo de texto sólo para escritura (si existe lo sobrescribe)
wb	Abre un archivo sólo para escritura binaria (si existe lo sobrescribe)
w+	Abre un archivo de texto para escritura y lectura (si existe lo sobrescribe)
a	Abre un archivo para anexar información

r -> abre un archivo sólo para lectura. El puntero al archivo está localizado al comienzo del archivo. Este es el modo predeterminado de la función.

r+ -> abre un archivo para escritura y lectura. El puntero del archivo está localizado al comienzo del archivo.

w -> abre un archivo solo para escritura.
Sobreescribe el archivo si este ya existe. Si el
archivo no existe lo crea.

w+ -> abre un archivo para escritura y lectura.
Sobreescribe el archivo si este ya existe. Si el
archivo no existe lo crea.

a -> abre un archivo para anexo. El puntero del archivo está al final del archivo si este existe. Es decir, el archivo está en modo anexo. Si el archivo no existe, crea un nuevo archivo para escritura.

Para abrir un archivo con Python, podemos usar la función **open**.

```
archivo = open(nombre_archivo, modo)
```

archivo objeto file, el cual será utilizado para llamar a otros métodos asociados con el archivo.

nombre_archivo string que contiene el nombre del archivo al que queremos acceder.

modo string que contiene el modo de apertura del archivo.

Modo de cierre



Para cerrar un archivo con Python, podemos usar la función **close**.

```
archivo.close()
```

archivo objeto file que fue obtenido con el método open.

Podemos usar la sentencia **with** para abrir archivos en Python y dejar que el intérprete se encargue de cerrar el mismo.

```
with open('archivo.txt', 'r+') as archivo:  
    for linea in archivo:  
        print(linea, end="")
```

Modo de escritura



Write : Escribe una cadena de caracteres dentro del archivo.

```
archivo = open('archivo.txt', 'w')  
archivo.write('Primer linea de texto\n')  
archivo.write('segunda linea\n')  
archivo.write('tercera linea\n')  
archivo.close()
```


Writelines : Escribe una lista de cadena de caracteres dentro del archivo. El parámetro que recibe el método podrá ser cualquier iterable.

```
archivo = open('archivo.txt', 'w')  
lineas_texto = ['Primer linea de texto\n',  
                'segunda linea\n',  
                'tercera linea\n']  
archivo.writelines(lineas_texto)  
archivo.close()
```

Modo de lectura



read: permite extraer un string que contenga todos los caracteres del archivo(Es posible limitar al método read a que lea cierta cantidad de bytes **read**(cantidad)).

```
# Abrimos el archivo en modo lectura y escritura
archivo = open('archivo.txt', 'r+')
texto = archivo.read()
print('El contenido del archivo es: ' + texto)
# Cerramos el archivo
archivo.close()
```

readlines: Permite obtener una lista con todas las líneas que contiene el archivo.

```
archivo = open('archivo.txt', 'r+')  
lista_lineas = archivo.readlines()  
for linea in lista_lineas:  
    print(linea, end="")  
# Cerramos el archivo  
archivo.close()
```