

COMANDOS

→ Modos de acceso al archivo:

1. Read Only ('r'): Abre el archivo en modo LECTURA.
 - a. Read and Write ('r+'): Abre el archivo en modo LECTURA Y ESCRITURA. Coloca el puntero al inicio del archivo (*Escribe nuevos registros al inicio*).
2. Write Only ('w'): Abre el archivo en modo ESCRITURA, que permite modificar y sobrecribir los datos de los archivos. **Si el archivo no existe** al momento de abrirlo, éste **es creado**. Coloca el **puntero al inicio** del archivo.
 - a. Write and Read ('w+'): Abre el archivo en modo LECTURA Y ESCRITURA.
3. Append Only ('a'): Abre el archivo en modo ESCRITURA. **Si el archivo no existe** al momento de abrirlo, éste **es creado**. Coloca el puntero al **final del archivo**. **Los datos** aquí escritos **se añadirán en el último registro** del archivo (*antes de EOF*), después de los datos ya contenidos en el archivo. **USAR PARA CSV.**
 - a. Append and Read ('a+'): Abre el archivo en modo LECTURA Y ESCRITURA.

Cómo CREAR y ABRIR archivos

1. "x" - Create: Este comando va a crear un nuevo archivo si no hay otro con ese nombre (*sino va a dar error*).

→ Ejemplo: `archivo = open("datos.txt", "x")`

2. "w" - Write: Este comando crea un archivo. Si existe otro con ese nombre, lo **sobreescribe**.

→ Ejemplo: `archivo = open("datos.txt", "w")`

!) Si el programa y el archivo están en la misma carpeta, sólo hace falta especificar el nombre del archivo (*como se muestra en los ejemplos*). Caso contrario, tendremos que escribir la ruta completa (*por ejemplo c:/Downloads/datos.txt*)

3. "r" - Read: Para almacenar todo el archivo de texto dentro de una variable (*lo que nos permitirá trabajar con ella más adelante*), utilizaremos el modo r descrito antes.
4. "a+" - Append: El modo a+ nos permite añadir nuevos datos automáticamente en el último registro disponible antes de EOF ya que coloca el puntero al final del archivo.

Cómo ESCRIBIR en un archivo

1. write(): Esta función añade un registro nuevo al archivo que tengamos abierto

→ Ejemplo: `archivo.write("Hello World")`
`archivo.write("\n")` → Este *enter* nos permite trabajar con el formato CSV.

2. writelines(): Esta función añade múltiples registros al mismo tiempo. Hay que seguir añadiendo “\n” para saltar a un nuevo registro.

→ Ejemplo: `archivo.writelines([“ Hello World ”, “ The car is red \n ”])`

¡ATENCIÓN!

Cuando se escribe “\n”, si bien el espacio aparece en el siguiente registro, éste carácter se almacena en el registro donde está escrito.

Cómo LEER un archivo

1. read(n): Devuelve el contenido leído dentro del archivo como una sola string. n indica el largo de la string a devolver: si n = 2, se mostrarán sólo 2 caracteres.

→ Ejemplo: `archivo = open(“datos.txt”, “r”) // Abre el archivo en modo lectura
print (archivo.read()) // Imprime el archivo completo
print (archivo.read(10)) // Imprime los primeros 10 caracts.`

2. readline(n): Lee un registro del archivo. n indica la cantidad de caracteres del registro que se leerán. Si n es más grande que el tamaño del registro, leerá el registro completo. Según la cantidad de veces que el comando esté escrito, será hasta qué registro se leerá (*recordar que no se puede leer directamente un registro porque es un archivo secuencial, hay que leerlos uno por uno*).

→ Ejemplo: `archivo = open(“datos.txt”, “r”) // Abre el archivo en modo lectura
print (archivo.readline()) // Imprime el primer registro completo
print (archivo.readline(3)) // Imprime los primeros 10 caracts. del 2º reg.`

3. readlines(): Almacena cada uno de los registros del archivo en un elemento de un **vector**, los cuales después pueden ser accedidos a través de un índice.

→ Ejemplo 1: `archivo = open(“datos.txt”, “r”) // Abre el archivo en modo lectura
vector = archivo.readlines() // Almacena todos los registros en elementos
de un vector
print (vector[2]) // Imprime el segundo registro del archivo`

→ Ejemplo 2: `archivo = open(“datos.txt”, “r”) // Abre el archivo en modo lectura.
for linea in archivo.readlines(): // Imprime todos los
registros.
print(linea)`

Cómo CERRAR un archivo

Una vez que terminamos de trabajar con un archivo, lo cerramos

→ [Ejemplo](#):

```
archivo = open("datos.txt", "r") // Abre el archivo en modo lectura
contenido = archivo.readline() // Almacena el primer registro completo.
archivo.close() // Cierra el archivo.
print contenido // Imprime el registro previamente almacenado.
```

Un motivo para querer cerrar un archivo es porque ya lo leímos, es decir llegamos al EOF:

→ [Ejemplo 2](#):

```
bandera = 0
while bandera == 0:
    registro = archivoLocalidad.readline()
    print(registro) // Imprimimos registros

    if not registro: //Si llegamos a EOF, terminamos.
        bandera = 1
```

Otros comandos útiles

1. seek(byte): Mueve el puntero hacia el byte indicado

→ [Ejemplo](#):

```
archivo = open("datos.txt", "r") // Abre el archivo en modo lectura
contenido = archivo.read() // Lee el archivo completo: el puntero queda al
                           final del documento
archivo.seek(0) // Reposiciona el puntero al inicio del archivo.
```

2. tell(byte): Retorna la posición actual del puntero

→ [Ejemplo](#):

```
archivo = open("datos.txt", "r") // Abre el archivo en modo lectura
linea1 = archivo.read() // Lee el archivo completo: el puntero queda al
                           final del documento
archivo.tell() // Indica la posición del puntero.
```