

Archivos de texto: creación, escritura y lectura

Una actividad muy común en un programa es el almacenamiento y recuperación de datos almacenado en un dispositivo secundario (disco duro).

Existen muchos modos de almacenar datos como son los archivos de texto, archivos binarios, bases de datos etc.

Archivo de texto

Un archivo de texto contiene un conjunto de caracteres estructurados en distintas líneas. Es un formato de archivo ampliamente utilizado como pueden ser:

El código fuente de un script en Python se almacena en un archivo de texto (igual que cualquier otro lenguaje de programación)

Archivos HTML, CSS, XML se almacenan en archivos de texto.

Creación de un archivo de texto y almacenamiento de datos.

Como es una actividad tan común en todo programa el lenguaje Python incluye por defecto todas las funcionalidades para trabajar con archivos de texto.

Para crear un archivo de texto debemos llamar a la función **open** y pasar dos parámetros, el primero indica el nombre del archivo a crear y el segundo un string con el caracter "**w**" (la "**w**" **write** indica crear el archivo de texto). Ejemplo:

```
arch1=open("datos.txt","w")
arch1.write("Primer línea.\n")
arch1.write("Segunda línea.\n")
arch1.write("Tercer línea.\n")
arch1.write("Línea final.")
arch1.close()
print('-'*80)
```

Luego de ejecutar este programa (no muestra nada por pantalla) debemos abrir con un editor de texto el archivo de texto que se acaba de crear llamado 'datos.txt': Y veremos:

Primer línea.

Segunda línea.

Tercer línea.

Final de línea.

Si el archivo 'datos.txt' ya existe luego se crea uno nuevo y se borra el actual.

El archivo se crea en la misma carpeta donde se está ejecutando el script de Python, si necesitamos que se cree en otra carpeta podemos indicar la ruta del mismo.

Si indicamos una ruta inexistente se genera un error.

Para grabar caracteres en el archivo de texto utilizamos el **método 'write'** y le pasamos un string a grabar:

```
archi1.write("Primer línea.\n")
```

Mediante la sintaxis `\n` indicamos que debe almacenarse un salto de línea en el archivo de texto. Si no incluimos los respectivos `\n` veremos:

```
arch1.write("Primer línea.")
arch1.write("Segunda línea.")
arch1.write("Tercer línea.")
```

El archivo de texto tiene una sola línea: *Primer línea.Segunda línea.Tercer línea.*

Luego de trabajar con el archivo de texto debemos liberarlo para que otro programa pueda acceder a su contenido:

```
arch1.close()
```

Lectura de un archivo de texto.

Para leer un archivo de texto debemos llamar a la función **'open'** y le pasamos el nombre del archivo a leer y un string con el caracter **"r"** (read):

```
arch1=open("datos.txt","r")
```

El método **'read'** recupera todas las líneas del archivo de texto (recupera el archivo completo):

```
contenido=arch1.read()
```

Luego de trabajar con el archivo de texto debemos liberarlo para que otro programa pueda acceder a su contenido:

```
arch1.close()
```

El método **'read'** sin parámetros retorna todos los caracteres almacenados en el archivo, opcionalmente podemos pasar un entero que represente la cantidad de caracteres a leer:

```
arch1=open("datos.txt","r")
contenido=arch1.read()
print(contenido)
arch1.close()
```

Lectura de un archivo de texto línea a línea.

En algunas situaciones podemos necesitar leer el contenido de un archivo de texto línea a línea. Disponemos de un método llamado **'readline'** que lee una línea completa del archivo, inclusive retorna el caracter `\n` de fin de línea.

```
arch1=open("datos.txt","r")
linea=arch1.readline()
while linea!="":
    print(linea, end="")
    linea=arch1.readline()
arch1.close()
```

Antes de la estructura repetitiva 'while' leemos la primera línea del archivo de texto:

```
linea=arch1.readline()
```

Luego, mientras la variable '**linea**' sea distinta a un **string vacío** imprimimos el contenido de la variable 'linea' evitando que genere un salto de línea la función print mediante la asignación del parámetro **end** con el valor de "":

```
while linea!="":  
    print(linea, end="")  
    linea=arch1.readline()
```

También dentro de la estructura while leemos las siguientes líneas.

Podemos recorrer el archivo leyendo línea a línea utilizando la estructura repetitiva for:

```
arch1=open("datos.txt","r")  
for linea in arch1:  
    print(linea, end="")  
arch1.close()
```

Almacenar un archivo de texto en una lista

Mediante el método '**readlines**' podemos recuperar cada una de las líneas del archivo de texto y almacenarlas en una lista.

```
arch1=open("datos.txt","r")  
lineas=arch1.readlines()  
print('El archivo tiene', len(lineas), 'líneas')  
print('El contenido del archivo')  
for linea in lineas:  
    print(linea, end="")  
arch1.close()
```

Mediante la llamada al **método 'readlines'** obtenemos una lista con todas las líneas del archivo 'datos.txt':

```
lineas=arch1.readlines()
```

Imprimimos seguidamente la cantidad de elementos de la lista que coincide con la cantidad de líneas que tiene el archivo de texto:

```
print('El archivo tiene', len(lineas), 'líneas')
```

Para imprimir el contenido del archivo de texto procedemos a imprimir la lista:

```
print('El contenido del archivo')  
for linea in lineas:  
    print(linea, end="")
```

Una variante del método **'readlines'** es pasar como un parámetro entero que represente que línea queremos recuperar:

```
linea=arch1.readlines(2)

print(linea, end=")    # imprime ['Primer línea.\n']
```

Abrir un archivo de texto para añadir líneas.

Si queremos que el archivo se abra para añadir sin borrar las líneas actuales del archivo debemos hacerlo con el parámetro **"a" (append)**

```
arch1=open("datos.txt","a")

arch1.write("nueva línea 1\n")

arch1.write("nueva línea 2\n")

archi1.close()

arch1=open("datos.txt","r")

contenido=arch1.read()

print(contenido)

arch1.close()
```

Abrimos el archivo 'datos.txt' en modo agregar:

```
archi1=open("datos.txt","a")
```

Almacenamos dos líneas en el archivo:

```
archi1.write("nueva línea 1\n")

archi1.write("nueva línea 2\n")
```

Cerramos el archivo y lo volvemos a abrir para leer su contenido:

```
archi1.close()

archi1=open("datos.txt","r")

contenido=archi1.read()

print(contenido)

archi1.close()
```

Abrir un archivo para leer y agregar datos.

Hay una cuarta forma de abrir un archivo indicando en el segundo parámetro de la función open el string "r+", con dicha opción podemos leer y escribir

```
arch1=open("datos.txt","r+")  
contenido=arch1.read()  
print(contenido)  
arch1.write("Otra línea 1\n")  
arch1.write("Otra línea 2\n")  
arch1.close()
```