

Manejo de archivos

La función clave para trabajar con archivos en Python es la `open()` función.

La `open()` función toma dos parámetros; nombre de archivo, y el modo .

Hay cuatro métodos diferentes (modos) para abrir un archivo:

`"r"`- Lectura - Valor por defecto. Abre un archivo para leer, error si el archivo no existe

`"a"` - Anexar: abre un archivo para anexar, crea el archivo si no existe.

`"w"` - Escribir: abre un archivo para escribir, crea el archivo si no existe.

`"x"` - Crear: crea el archivo especificado, devuelve un error si el archivo existe

Además, puede especificar si el archivo debe manejarse como modo binario o de texto.

`"t"`- Texto - Valor por defecto. Modo de texto

`"b"` - Binario - Modo binario (por ejemplo, imágenes)

Sintaxis

Para abrir un archivo para leerlo basta con especificar el nombre del archivo:

```
f = open("demofile.txt")
```

El código de arriba es el mismo que:

```
f = open("demofile.txt", "rt")
```

Dado que `"r"`para lectura y `"t"`para texto son los valores predeterminados, no es necesario especificarlos.

Abra un archivo en el servidor

Asume que tenemos el siguiente archivo, ubicado en la misma carpeta que Python:

```
demofile.txt
```

```
Hello! Welcome to demofile.txt  
This file is for testing purposes.  
Good Luck!
```

Para abrir el archivo, use la `open()` función incorporada.

La `open()` función devuelve un objeto de archivo, que tiene un `read()` método para leer el contenido del archivo:

```
f = open("demofile.txt", "r")  
print(f.read())
```

Sólo leer partes del archivo

De forma predeterminada, el `read()` método devuelve todo el texto, pero también puede especificar cuántos caracteres desea devolver:

Ejemplo

Devuelve los 5 primeros caracteres del archivo:

```
f = open("demofile.txt", "r")  
print(f.read(5))
```

Leer Líneas

Puedes devolver una línea usando el `readline()` método:

Ejemplo

Lea una línea del archivo:

```
f = open("demofile.txt", "r")  
print(f.readline())
```

Al llamar `readline()` dos veces, puedes leer las dos primeras líneas:

Ejemplo

Lee dos líneas del archivo:

```
f = open("demofile.txt", "r")
print(f.readline())
print(f.readline())
```

Al recorrer las líneas del archivo, puede leer el archivo completo, línea por línea:

Ejemplo

Recorrer el archivo línea por línea:

```
f = open("demofile.txt", "r")
for x in f:
    print(x)
```

Escribir en un archivo existente

Para escribir en un archivo existente, debe agregar un parámetro a la `open()` función:

`"a"` - Adjuntar - se adjuntará al final del archivo

`"w"` - Escribir - sobrescribirá cualquier contenido existente

Ejemplo

Abra el archivo "demofile.txt" y agregue el contenido al archivo:

```
f = open("demofile.txt", "a")
f.write("Now the file has one more line!")
```

Ejemplo

Abra el archivo "demofile.txt" y sobrescriba el contenido:

```
f = open("demofile.txt", "w")
f.write("Woops! I have deleted the content!")
```

Nota: el método "w" sobrescribirá todo el archivo.

Crear un nuevo archivo

Para crear un nuevo archivo en Python, use el `open()` método, con uno de los siguientes parámetros:

`"x"` - Crear: creará un archivo, devuelve un error si el archivo existe

`"a"` - Anexar: creará un archivo si el archivo especificado no existe

`"w"` - Escribir: creará un archivo si el archivo especificado no existe

Ejemplo

Crea un archivo llamado "myfile.txt":

```
f = open("myfile.txt", "x")
```

Resultado: se crea un nuevo archivo vacío!

Ejemplo

Crea un nuevo archivo si no existe:

```
f = open("myfile.txt", "w")
```

Borrar un archivo

Para eliminar un archivo, debe importar el módulo del sistema operativo y ejecutar su `os.remove()` función:

Ejemplo

Eliminar el archivo "demofile.txt":

```
import os  
os.remove("demofile.txt")
```

Compruebe si el archivo existe:

Para evitar obtener un error, es posible que desee comprobar si el archivo existe antes de intentar eliminarlo:

Ejemplo

Compruebe si el archivo existe, luego elimínelo:

```
import os
if os.path.exists("demofile.txt"):
    os.remove("demofile.txt")
else:
    print("The file does not exist")
```

Eliminar carpeta

Para borrar una carpeta completa, usa el `os.rmdir()` método:

Ejemplo

Eliminar la carpeta "myfolder":

```
import os
os.rmdir("myfolder")
```

Nota: Sólo puedes eliminar carpetas vacías .