# Archivos

Hasta ahora hemos visto como interactuar con un programa a través del teclado (entrada de datos) y la terminal (salida), pero en la mayor parte de las aplicaciones reales tendremos que leer y escribir datos en archivos.

Al utilizar archivos para guardar los datos estos perdurarán tras la ejecución del programa, pudiendo ser consultados o utilizados más tarde.

Las operaciones más habituales con archivos son:

- Crear un archivo.
- Escribir datos en un archivo.
- Leer datos de un archivo.
- Borrar un archivo.

## 1. Creación y escritura de archivos

Para crear un archivo nuevo se utiliza la instrucción

- `open(ruta, 'w')`: Crea el archivo con la ruta `ruta`, lo abre en modo escritura (el argumento `'w'` significa `write`) y devuelve un objeto que lo referencia.

*Si el archivo indicado por la ruta ya existe en el sistema, se reemplazará por el nuevo.*

Una vez creado el archivo, para escribir datos en él se utiliza el método

- `archivo.write(c)`: Escribe la cadena `c` en el archivo referenciado por `archivo`.

---

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> f = open('bienvenida.txt', 'w')
... f.write('¡Bienvenido a Python!')
```

---

## 2. Añadir datos a un archivo

Si en lugar de crear un archivo nuevo queremos añadir datos a un archivo existente se debe utilizar la instrucción

- `open(ruta, á')`: Abre el archivo con la ruta `ruta` en modo añadir (el argumento `'a'` significa append) y devuelve un objeto que lo referencia.

Una vez abierto el archivo, se utiliza el método de escritura anterior y los datos se añaden al final del archivo.

---

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> f = open('bienvenida.txt', 'a')
... f.write('\n¡Hasta pronto!')
```

---

## 3. Leer datos de un archivo

Para abrir un archivo en modo lectura se utiliza la instrucción

- `open(ruta, 'r')`: Abre el archivo con la ruta `ruta` en modo lectura (el argumento `'r'` significa `read`) y devuelve un objeto que lo referencia.

Una vez abierto el archivo, se puede leer todo el contenido del archivo o se puede leer línea a línea.

## 4. Leer datos de un archivo

- `archivo.read()`: Devuelve todos los datos contenidos en archivo como una cadena de caracteres.
- `archivo.readlines()`: Devuelve una lista de cadenas de caracteres donde cada cadena es una linea del archivo referenciado por archivo.

---

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> f = open('bienvenida.txt', 'r')
... print(f.read())
¡Bienvenido a Python!
¡Hasta pronto!
```

---

---

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> f = open('bienvenida.txt', 'r')
... lineas = print(f.readlines())
>>> print(lineas)
['Bienvenido a Python!\n', '¡Hasta pronto!']
```

---

## 5. Cerrar un archivo

Para cerrar un archivo se utiliza el método

- `archivo.close()`: Cierra el archivo referenciado por el objeto archivo.

Cuando se termina de trabajar con un archivo conviene cerrarlo, sobre todo si se abre en modo escritura, ya que mientras está abierto en este modo no se puede abrir por otra aplicación. Si no se cierra explícitamente un archivo, Python intentará cerrarlo cuando estime que ya no se va a usar más.

---

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> f = open('bienvenida.txt'):
... print(f.read())
... f.close()  # Cierre del archivo
...
¡Bienvenido a Python!
¡Hasta pronto!
```

---

## 6. Renombrado y borrado de un archivo

Para renombra o borrar un archivo se utilizan funciones del módulo os.

- `os.rename(ruta1, ruta2)`: Renombra y mueve el archivo de la ruta `ruta1` a la ruta `ruta2`.
- `os.remove(ruta)`: Borra el archivo de la ruta `ruta`.

Antes de borrar o renombra un directorio conviene comprobar que existe para que no se produzca un error. Para ello se utiliza la función

- `os.path.isfile(ruta)`: Devuelve True si existe un archivo en la ruta `ruta` y `False` en caso contrario.

## 7. Renombrado y borrado de un archivo o directorio

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> import os
>>> f = 'bienvenida.txt'
>>> if os.path.isfile(f):
...     os.rename(f, 'saludo.txt') # renombrado
... else:
...     print('¡El archivo', f, 'no existe!')
...
>>> f = 'saludo.txt'
>>> if os.path.isfile(f):
...     os.remove(f) # borrado
... else:
...     print('¡El archivo', f, 'no existe!')
...
```

## 8. Creación, cambio y eliminación de directorios

Para trabajar con directorios también se utilizan funciones del módulo os.

- `os.listdir(ruta)`: Devuelve una lista con los archivos y directiorios contenidos en la ruta ruta.
- `os.mkdir(ruta)`: Crea un nuevo directorio en la ruta ruta.
- `os.chdir(ruta)`: Cambia el directorio actual al indicado por la ruta ruta.
- `os.getcwd()`: Devuelve una cadena con la ruta del directorio actual.
- `os.rmdir(ruta)`: Borra el directorio de la ruta ruta, siempre y cuando esté vacío.

## 9. Leer un archivo de Internet

Para leer un archivo de internet hay que utilizar la función urlopen del módulo urllib.request.

- `urlopen(url)`: Abre el archivo con la url especificada y devuelve un objeto del tipo archivo al que se puede acceder con los métodos de lectura de archivos anteriores.

***Ejercicio de Inducción:*** *Pruebe las siguientes líneas de código y verifique los resultados presentados:*

```
>>> from urllib import request
>>> f = request.urlopen('https://mindhubweb.com/')
>>> datos = f.read()
>>> print(datos.decode('utf-8'))
MindHub
===============

Este es el repositorio del sitio web de Mindhub http://mindhubweb.com
```