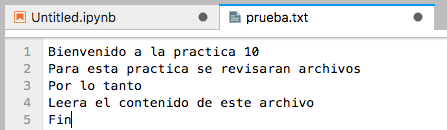
**Prepráctica(Alejandro)**

1. Ejercicios de la guía: En la guía se describen los archivos, empezando por las características básicas con las que cuentan, como pueden ser su tamaño, fecha de creación, nombre, etc. La característica más importante es el nombre, pues, gracias a este podremos identificar el archivo con el que querremos trabajar.

Después de los atributos, se describen las operaciones sobre archivos, las cuales se realizan en la implementación presentada de Python.

Por último, se habla de cosas más específicas de archivos, como métodos de acceso a ellos, los tipos que existen, su estructura interna, entre otras cuestiones.

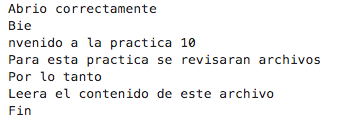
Respecto a la implementación en Python, para poder comprobar el funcionamiento de las operaciones realizadas, se creó un archivo cuyo nombre es prueba.txt, el cual contiene los siguientes datos iniciales:



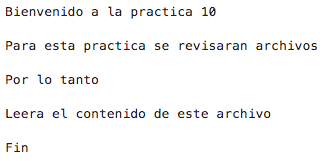
Se empieza por abrir abrir el archivo, utilizando un bloque tipo try-catch en Python, el cual maneja una excpeción en caso de que exista un error al abrir el archivo. Si el archivo se abrió correctamente, se le indica al usuario mediante una impresión en pantalla lo anterior.

Existen diversas formas de apertura que se pueden indicar al momento de trabajar con archivos, pues se pueden abrir en modo de solo lectura, lectura y escritura, etc.

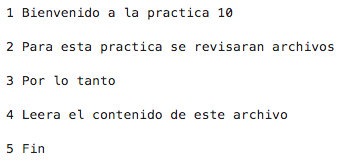
Una vez abierto el archivo, se utiliza el método read(), al cual se le pasa como parámetro la cantidad de bytes a leer del archivo y, si no se especifica, se leen todos los bytes del archivo. Primero se leen tres bytes y luego el resto.



Con el método readline() podemos ir leyendo toda una línea completa, como se muestra a continuación:

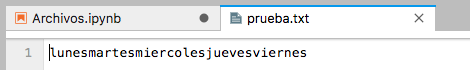


Con el método readlines() se leen todas las líneas en el archivo, y se guardan en una variable de tipo lista. Se recorren con un ciclo for y se va mostrando en pantalla también el número de línea correspondiente:

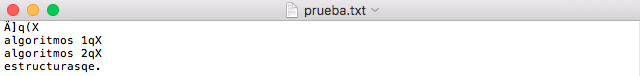


Con la estructura with-as, lo anterior se simplifica, pues se cierran los archivos automáticamente después de ser utilizados, además de que su lecutra se facilita.

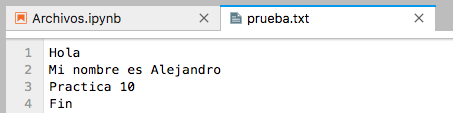
Por otro lado, con respecto a la escritura, tenemos los método write() y writelines(), en donde, writelines() se utiliza para escribir una lista de elementos y write() escribe una cadena o carácter. Utilizando writelines() se encuentra lo siguientes escrito en el archivo:

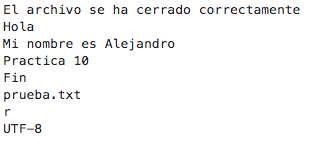


Los métodos seek() y read() nos ayudarán a ir moviendo el apuntador utilizado para leer la información del archivo. A seek() se le indica la posición en bytes hasta la cual mover el apuntador a partir de su posición actual y a read() se le indica cuantos bytes leer a paritr de la posición actual del apuntador. Por último, tell() nos indica el byte actual del apuntador.

Para la escritura y lectura de objetos, se importa pickle, la cual nos permite leer y escribir objetos en un archivo. Para este caso, así se observa un objeto de tipo lista en un archivo de texto plano:

Al igual que con archivos, también se pueden crear directorios y recorrerlos. Para crearlos y recorrerlos se importa os, en donde, el método makedirs() nos permite crear un directorio y, chdir() ubicarnos en dicho direcotrio. Es importante remarcar que, cuando se utiliza chdir(), la ruta actual se mueve hacía el direcotorio especificado, por lo que, es importante utilizar rutas absolutas para evitar confusiones. En este caso, se crea un directorio llamado como nuevo y se recorre. Debido a que acaba de ser creado, no contiene nada, por lo que, al recorrerlo, no aparece nada en pantalla.

Por último, el método read() nos permite leer todo el contenido de un archivo. Los atributos name, mode y encoding de un objeto de tipo archivo, nos ofrecen el nombre, modo en que se barió el archuvo y tipo de codificamiento de un archivo respectivamente. Para este caso, se modificó el archivo original de la siguiente manera:

En pantalla se muestra lo siguiente del archivo: