*#Andrés Bendaña*

*#Módulo 1, Semana 8, “File”, Ejercicio 2*

* close(): Cierra un archivo abierto previamente, liberando recursos del sistema, asegura terminar operaciones en curso y evita pérdidas de datos o corrupción de datos.
* detach(): Se utiliza para desprender el flujo binario subyacente a objetos creados por ejemplo con TextIOWrapper, que convierten datos binarios en texto, de esta manera pueden ser manipulados directamente.
* fileno(): Este método devuelve un número de descriptor de archivo. Este número identifica un archivo abierto lo cual es útil para manipular este archivo por APIs o ciertas bibliotecas, o incluso para interactuar con otros lenguajes.
* flush(): Vacia el búfer de salida de un archivo, asegurando que todo el contenido que ha sido escrito en memoria se escriba de inmediato en el archivo o flujo subyacente.
* isatty(): Se utiliza para determinar si un archivo está conectado a un terminal interactivo (TTY); devuelve True si está conectado.
* read(): Lee y retorna todo el contenido del archivo; puede leer tanto archivos de texto como archivos binarios.
* readable(): Retorna si un archivo ha sido abierto en un modo que permite su lectura.
* readline(): Lee y retorna una sola línea de un archivo.
* readlines(): Lee todo el contenido del archivo y retorna una lista con cada línea.
* seek(): Sirve para mover el puntero de lectura/escritura dentro de un archivo a una posición específica.
* seekable(): Retorna True si un archivo puede moverse por medio de seek().
* tell(): Devuelve la posición de un archivo abierto.
* truncate(): Permite cambiar el tamaño del archiv; recorta los bytes para disminuirlo o rellena con bytes para agrandarlo.
* writable(): Verifica si un archivo ha sido abierto en un modo que permite la escritura. Devuelve True o False
* write(): Escribe contenidos en un archivo.
* writelines(): Se usa para escribir múltiples líneas en un archivo de una sola vez.