CURSOS INTERSEMESTRALES

PROTECO

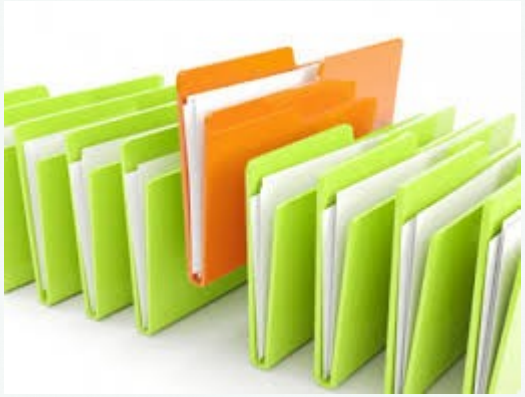
Manejo de Archivos

Archivos

Python Intermedio
Junio 2019

Archivos

- Los objetos del tipo File en Python, son la principal manera de interactuar con archivos presentes en nuestra computadora.
- Estos objetos pueden ser usados para leer o escribir cualquier tipo de archivo almacenado en nuestro disco duro: notas, archivos de sonido, documentos de Excel, etcétera.



PROTECO

Archivos

- Para crear un nuevo archivo de texto, podemos llamar la función ‘open’ y como argumentos usamos el nombre del archivo que queremos crear y una ‘w’ como segundo parámetro. A continuación un ejemplo de ello:
-
- >> d = open(‘miPrimerArchivo.txt’,’w’)
- >>d.write(‘Hola ‘)
- >>d.write(‘todos!’)
- >>d.close()
-

PROTECO

Modos de apertura de archivos

Por el tipo de archivo:
- ‘t’: para archivos de texto.
-’b’: para permitir escritura en modo binario.

Por el tipo de acceso:
-’r’: indica modo de lectura (read).
-’w’: indica el modo de escritura (write). En caso de existir el archivo, este se sobreescribe.
-’a’: es otro modo de escritura (append). En caso de existir el archivo, comienza a escribir al final de este.
- ‘x’: es otro modo de escritura para crear un nuevo archivo. Si dicho archivo ya existe, se emitirá el error ‘FileExistsError’.

PROTECO

Tabla tomada de https://uniwebsidad.com/libros/python/capitulo-9/sobre-el-objeto-file

Indicador	Modo de apertura	Ubicación del puntero
`r`	Solo lectura	Al inicio del archivo
`rb`	Solo lectura en modo binario	Al inicio del archivo
`r+`	Lectura y escritura	Al inicio del archivo
`rb+`	Lectura y escritura en modo binario	Al inicio del archivo
`w`	Solo escritura. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
`wb`	Solo escritura en modo binario. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
`w+`	Escritura y lectura. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
`wb+`	Escritura y lectura en modo binario. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
`a`	Añadido (agregar contenido). Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
`ab`	Añadido en modo binario (agregar contenido). Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
`a+`	Añadido (agregar contenido) y lectura. Crea el archivo si éste no existe.	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
`ab+`	Añadido (agregar contenido) y lectura en modo binario. Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo

PROTECO

Métodos más utilizados

- close(): Una vez que se hayan realizado todas las operaciones de entrada y de salida de archivos, este debe ser cerrado de forma adecuada. Si no lo hacemos, es muy probable que nuestra información se corrompa o destruya.
- writable() : Este método devuelve ‘True’ si el archivo está en modo de escritura.
- readable(): Devuelve True si el archivo está en modo de lectura.
- seekable(): Devuelve ‘True’ si es posible desplazarse dentro del archivo.

PROTECO

Métodos más utilizados

- read(): Leerá y regresará el contenido del archivo desde la posición en la que se encuentre hasta el final del archivo. Si se ingresa como argumento algún número, leerá ese número de posiciones.
-
- write(): Añade al archivo el contenido ingresado como argumento a partir de la posición actual. Terminada la operación, retorna la nueva posición del puntero.
-
- tell(): Regresa la posición en la que se encuentra el puntero dentro del archivo.
- .

PROTECO

Métodos exclusivos de archivos de texto

- readline(): Leerá el texto desde la posición donde se encuentre hasta que encuentre el carácter de escape retorno de linea (\n).
-
- readlines(): Leerá el texto desde la posición del puntero hasta el final del archivo, y creará una lista que contenga todas las líneas del mismo.
-
- writelines(): Escribirá el texto contenido dentro de un elemento de tipo list o tuple.
-
- seek(): Mueve el puntero a la posición indicada.

PROTECO

CURSOS INTERSEMESTRALES

PROTECO

Manejo de Archivos

Manejadores de contexto

Python Intermedio
Enero 2019

Manejadores de contexto

- Mediante una estructura de manejo de contexto, podemos realizar operaciones seguras con nuestros archivos, ya que nos permite ejecutar un bloque de código y automáticamente cerrar el archivo. Su sintaxis es la siguiente:
-
- >>with open(<nombre del archivo>,<modo>) as <nombre>:
- >> ...
- >> ...

PROTECO