

Estructura de Datos y POO

Semana 12





Contenido

- Accediendo a archivos
 - Lectura de archivos
 - Escritura de archivos

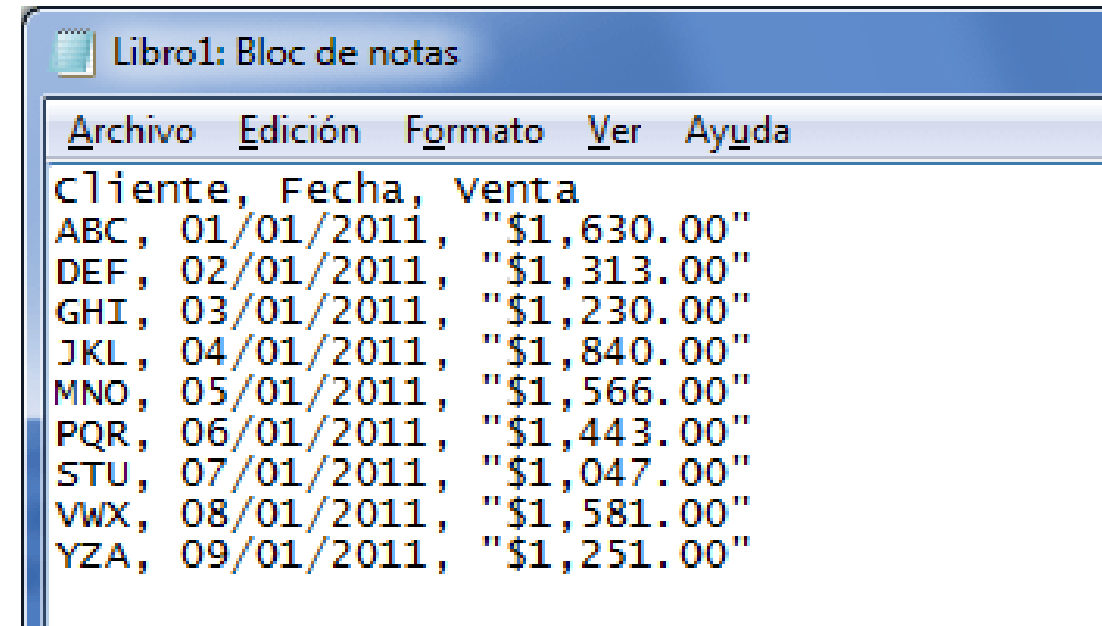


PYTHON



Accediendo a archivos

- ✚ Una actividad muy común en un programa es el almacenamiento y recuperación de datos almacenados.
- ✚ Existen muchos modos de almacenar datos como son los archivos de texto, bases de datos, etc.



Libro1: Bloc de notas				
Archivo	Edición	Formato	Ver	Ayuda
Cliente, Fecha, Venta				
ABC,	01/01/2011,	"\$1,630.00"		
DEF,	02/01/2011,	"\$1,313.00"		
GHI,	03/01/2011,	"\$1,230.00"		
JKL,	04/01/2011,	"\$1,840.00"		
MNO,	05/01/2011,	"\$1,566.00"		
PQR,	06/01/2011,	"\$1,443.00"		
STU,	07/01/2011,	"\$1,047.00"		
VWX,	08/01/2011,	"\$1,581.00"		
YZA,	09/01/2011,	"\$1,251.00"		



Accediendo a archivos

Hay dos operaciones básicas a realizar:

- ✚ **Lectura:** Las porciones de los datos se recuperan del archivo y se colocan en un área de memoria administrada por el programa.
- ✚ **Escritura:** Las porciones de los datos de la memoria se transfieren al archivo.





Accediendo a archivos

Hay tres modos básicos utilizados para abrir un archivo:

✚ **Modo Lectura (r ó rt):** Permite solo lectura. El archivo debe existir.

```
fo = open("Archivo.txt", "rt")
print(fo.read())
fo.close()
```

✚ **Modo Escritura (w ó wt):** Permite solo escritura. Si el archivo no existe, se creará; si existe, borrará el contenido previo.

```
fo = open('Archivo2.txt', 'wt')
fo.write("Estructura de Datos y POO")
fo.close()
```

✚ **Modo Actualizar (a ó at):** Permite solo escritura. Si no existe, se creará; si existe, agregará el contenido al final del archivo.

```
fo = open('Archivo2.txt', 'at')
fo.write("\nSemana 06: Procesamiento de archivos")
fo.close()
```



Lectura de archivos: Método read()

- Lee todo el contenido del archivo y lo devuelve como una cadena.
- Cuando no hay que leer la función devuelve una cadena vacía.

```
try:
    cnt = 0
    s = open('text.txt', "rt")
    ch = s.read(1)
    while ch != '':
        print(ch, end='')
        cnt += 1
        ch = s.read(1)
    s.close()
    print("\n\nCaracteres en el archivo: ", cnt)
except IOError:
    print("Se produjo un error de E/S")
```



Lectura de archivos: Método readline()

- Maneja el contenido del archivo como un conjunto de líneas.
- Lee una línea completa del archivo y la devuelve como una cadena de caracteres.
- Cuando no hay nada que leer la función devuelve una cadena vacía.

```
try:
    cnt = lcnt = 0
    s = open('text.txt', 'rt')
    line = s.readline()
    while line != '':
        lcnt += 1
        for ch in line:
            print(ch, end='')
            cnt += 1
        line = s.readline()
    s.close()
    print("\n\nCaracteres en el archivo: ", cnt)
    print("Lineas en el archivo:      ", lcnt)
except IOError:
    print("Se produjo un error de E/S")
```



Lectura de archivos: Método readlines()

- + Sin argumentos, lee todo el contenido del archivo y devuelve una lista de cadenas, un elemento por línea del archivo.
- + Cuando se le pasa un argumento, se indica el tamaño máximo de la memoria de almacenamiento temporal aceptado.

```
try:
    cnt = lcnt = 0
    s = open('text.txt', 'rt')
    lines = s.readlines(20)
    while len(lines) != 0:
        for line in lines:
            lcnt += 1
            for ch in line:
                print(ch, end='')
                cnt += 1
        lines = s.readlines(10)
    s.close()
    print("\n\nCaracteres en el archivo: ", cnt)
    print("Lineas en el archivo:      ", lcnt)
except IOError:
    print("Se produjo un error de E/S")
```




Escritura de archivos: Método write()

- ✚ Este método recibe una cadena que se transferirá a un archivo abierto, el modo de apertura debe reflejar la forma en que se transfieren los datos.
- ✚ No se agrega carácter de nueva línea al argumento de **write()**, se debe agregar de ser el caso.

```
try:
    fo = open('newtext.txt', 'wt')
    for i in range(10):
        s = "línea #" + str(i+1) + "\n"
        for ch in s:
            fo.write(ch)
    fo.close()
except IOError:
    print("Se produjo un error de E/S")
```



Gracias