



Seminario de Lenguajes

Opción Python

Práctica 3

Manejo de archivos

Parte I

- 1.- Dado un conjunto de números (que se tomarán de la entrada estándar), generar dos archivos: uno con los números pares y otro con los impares.
- 2.- Escriba una función que permita realizar un merge entre los dos archivos generados en el ejercicio 1. Los nombres de los archivos vienen dados como argumentos en la línea de comando.
- 3.- Agregue al ejercicio 2, un parámetro opcional adicional que permita ordenar, en forma ascendente o descendente, los archivos.
- 4.- Escriba un módulo que provea la funcionalidad de almacenar en un archivo nombres de personas y sus edades. Además realice una función que busque en el archivo e imprima un listado de todos los nombres cuya edad sea mayor a un número determinado.
- 5.- Realice el ejercicio anterior utilizando la librería pickle.

Parte II - Ejercicio a entregar: manejo de calendario de cumpleaños

Se necesita manejar un calendario de cumpleaños donde se pueda almacenar la siguiente información:

- **Fecha de cumpleaños**
- **Nombre del cumpleaños**
- **Medio de contacto** (1 por cumpleaños): compuesto por:
 - **Tipo de contacto:** por ejemplo "email", "teléfono", "celular", "dirección"
 - **Medio de contacto:** por ejemplo "jorge@example.com", "(0221) 423-2030", "(0221) 15 567-8910", "50 y 120" respectivamente.

Esta información debe ser persistida en un archivo utilizando alguna estructura que permita realizar las siguientes operaciones:

- Búsqueda por fecha de cumpleaños o por nombre del cumpleaños
- Cargar un nuevo cumpleaños
- Borrar un cumpleaños

Al persistir información en archivos de texto plano, estos pueden ser leídos y manipulados con cualquier editor (editPlus, vi, vim, gedit, bloc de notas, etc.). Es por eso que se deberá encriptar la información que se guarda. Para ello proponemos diferentes técnicas sencillas para poder implementarlo:

1. **Utilizando las funciones `ord()` y `chr()`:** se pueden manipular los valores de cada uno de los caracteres de forma que lo que se vea no es realmente lo que es. Pongamos un ejemplo; si tenemos la palabra “python” podríamos efectuar un tratamiento carácter por carácter convirtiendo (incrementando o decrementando) su valor ASCII, si a cada carácter se le resta 30 del código ASCII quedaría así “R[VJQP” y de esta forma sería como podríamos almacenarlo dificultando la tarea de comprensión del archivo. Luego para desencriptarlo haríamos la función inversa: toma caracter por caracter, se lo convierte a valores ASCII y se le suma 30 quedando nuevamente la palabra “python”.
2. **Usando la función `maketrans()`:** dicha función realiza una conversión simultánea de los caracteres basados en una tabla de conversión que es provista por el programador. Se necesita la tabla para encriptar y una tabla para desencriptar. Para que funcione correctamente una debe ser la inversa de la otra.
3. **Pasando cada dígito a binario y separarlo por un caracter especial:** primero se debería convertir cada carácter a su correspondiente código ASCII, y luego se podría convertir dicho valor a binario para guardarlo, separando cada caracter, por ejemplo, con un guión (-). De esta manera, para desencriptarlo se podría guardar en una lista cada número separado por guión, convertirlo a decimal y obtener su correspondiente carácter ASCII.
4. **Usando un módulo de compresión:** importando algún módulo de compresión como el zlib, gzip o el bz2 se puede comprimir un string mediante la función `compress()`, lo que permite reducir su espacio físico una vez almacenado y que además su código es ilegible. Luego se puede descomprimir cada string utilizando la función `decompress()`.
(<http://docs.python.org/tutorial/stdlib.html#data-compression>)

Adicionalmente se requiere compartir la información entre distintas aplicaciones, para ello se pide que se desarrolle la exportación del calendario en formato **iCalendar**. Una vez que se obtenga el archivo con la exportación de los eventos creados (cumpleaños) se debe probar en alguna aplicación que soporte este formato.

Información sobre iCalendar <http://es.wikipedia.org/wiki/ICalendar>

Nota 1: Cada grupo va a tener asignado un método de encriptación. La asignación será en base a la siguiente función que deberán correr en un módulo para averiguar qué método deben implementar:

```
x = int(raw_input('Ingrese su numero de grupo: '))  
print (x%4) if (x%4!=0) else 4
```

Nota 2: Recomendamos la utilización de **Pickle** para guardar las estructuras completas en el archivo (<http://docs.python.org/library/pickle.html>).

Fecha tope de entrega: **Lunes 13 de Mayo a las 10:00hs**
Fecha de coloquios: **Lunes 13 y Martes 14 de Mayo en los horarios de práctica**