

8. Sesión nº 8

Objetivos: usar estructuras de datos simples como listas, tuplas, conjuntos y diccionarios en python; refuerzo de bucles y funciones; aumento entrada/salida con ficheros

1. Crea un fichero de texto con una lista de asignaciones de pasajeros de asientos donde en cada línea aparece primero nombre y luego identificador del asiento separados por coma, por ejemplo:

```
María, 2A
Manuel, 21D
Dorotea, 14F
Samuel, 10C
```

Si tenemos un fichero como el anterior con nombre `datos.txt`, el siguiente programa lo leerá y transformará ligeramente el formato, creando otro fichero llamado `datos2.txt`:

```
fin = open('datos.txt', 'r')
lines = fin.read().splitlines()
fin.close()

fout = open('datos2.txt', 'w')
fout.write('Reservations list:\n')
fout.write('-----\n')
for line in lines:
    fout.write(' '+line+'\n')
fout.close()
```

Escribe un programa que lea tal fichero y lo convierta automáticamente a un fichero donde en la primera línea aparecen todos los asientos ocupados y en la segunda línea los nombres en el orden según sus asientos.

Opcional: intenta ordenar ambas líneas según los identificadores de los asientos.

2. Realiza un pequeño sistema de reservas para un avión [A320](#). El programa debe actuar con un menú principal donde la entrada de una letra selecciona un módulo de actuación, por ejemplo, `r` activa el módulo de reserva, `l` activa el módulo de generar lista, `q` sale del programa.

El menú debe contener por lo menos los siguientes módulos:

- leer un fichero (con formato como en el apartado arriba),
- listar todos los asientos ocupados,
- añadir una reserva con un nombre y un asiento (ojo, solo un pasajero por asiento, y solo asientos posibles),
- guardar las reservas en un fichero,
- borrar una reserva (o bien con nombre o bien con asiento),
- mostrar asientos todavía libres,
- aumenta con funcionalidades que te resultan útiles, (algunas pueden ayudarte

también para la depuración del programa).

Para no complicarte la vida al principio, puedes asumir que las filas de los asientos tengan números consecutivos y que siempre haya la misma cantidad de asientos por fila.

Emplea bien y con sentido las estructuras de datos ya conocidas, por ejemplo, tuplas para unir pasajero con asiento, diccionarios para la ocupación del avión, conjuntos para los caracteres del menú, etc.

Notarás que cuando más modular desarrollas el programa, más fácil es completar la tarea.

Podeis trabajar en grupos pequeños y que cada participante contribuya con uno o varios módulos.