



Semana 6: Listas y tuplas

Programación

2020 - 2021

- Ejercicio 1.** Crea una lista inicializando sus elementos en la declaración. Asigna a uno de sus elementos otro de ellos (ej. `a[5] = a[3]`). Muestra por pantalla ambos elementos. A continuación añade una instrucción para cambiar el valor del segundo elemento y muestra por pantalla ambos de nuevo. ¿Ha cambiado también el primero? ¿Por qué? ¿Qué sucede si se cambia el valor del primero (`a[5]`)?
- Ejercicio 2.** Crea una lista de longitud mayor que 20 e inicializa aleatoriamente sus valores. Crea una segunda lista y hazla igual a la primera (`a = b`). Cambia el valor de un elemento de la primera lista. ¿Cambia también el correspondiente de la segunda lista? ¿Por qué? Crea otras dos listas y repite el paso previo pero en vez de utilizar el operador de asignación (`=`) copia los elementos de la primera lista en la segunda uno a uno. ¿Hay alguna diferencia?
- Ejercicio 3.** Escribe un programa que haga lo siguiente:
1. Pregunta al usuario por el tamaño de la lista a crear que debe ser mayor que 0. En caso contrario continuará preguntando por el tamaño hasta obtener uno correcto.
 2. Rellena la lista aleatoriamente con números flotantes entre 1 y 49
 3. Crea una variable llamada 'total'
 4. Suma la parte entera de todos los elementos de la lista y almacena el valor en la variable 'total'
 5. Muestra por pantalla la lista y el valor de 'total'
- Ejercicio 4.** Escribe un programa que cree una lista de 20 números enteros, completándola aleatoriamente con números entre 1 y 9. Pide al usuario un número (el programa debe comprobar que el número está entre 1 y 9), finalmente muestra por pantalla si el número está en la lista y en qué posición aparece en caso afirmativo. Si el número aparece más de una vez, muestra por pantalla todas las posiciones.
- Ejercicio 5.** Escribe un programa que adivine el número que el usuario ha pensado (Entre 1 y 100). En cada intento el programa generará números aleatorios y deberá controlar que no se repite. Deberá ser posible detectar si el usuario está mintiendo (si ha intentado decir todos los números y el usuario dice que no es ninguno de ellos). El programa contará además el número de intentos que ha necesitado para adivinarlo.
- Ejercicio 6.** Escribe un programa que cree una tupla con los nombres de los meses del año. El programa pedirá al usuario un número. Si el número está entre 1 y la longitud de la tupla, mostrará el correspondiente mes del año. En otro caso mostrará un mensaje de error y pedirá otro número. El programa estará funcionando hasta que el usuario introduzca 0.
- Ejercicio 7.** Escribe un programa que pida al usuario introducir una frase. Se introducirá cada carácter en una tupla ignorando los espacios. Finalmente se mostrará por pantalla la tupla. Por ejemplo `Hola, ¿cómo estás?` ('H', 'o', 'l', 'a', ',', '¿', 'c', 'ó', 'm', 'o', ' ', 'é', 's', 't', 'á', 's', '?')
- Ejercicio 8.** Escribe un programa que le pida al usuario introducir una frase y cree una tupla ignorando los caracteres repetidos. Los caracteres introducidos en la tupla serán convertidos en mayúsculas.

Finalmente mostrará la tupla por pantalla. Por ejemplo: Si el usuario escribe la frase `Hi, how are you?` Imprimirá `('H', 'I', ',', ',', ' ', 'O', 'W', 'A', 'R', 'E', 'Y', 'U', '?')`
Nota: El método `.upper()` en un string convierte todos los caracteres a mayúscula y el método `.lower()` convierte todo a minúsculas.

Ejercicio 9. Un gimnasta puede obtener de cada juez una puntuación entre 1 y 10 (valores enteros). Hay 5 jueces. Escribe un programa que almacene las posibles puntuaciones que un gimnasta puede recibir de un jurado en una tupla (generada aleatoriamente) y que muestre por pantalla las siguientes frases usando los valores de la tupla:

El Juez `z` da al gimnasta `t` puntos (Si obtiene 1 punto se debe decir 1 punto, no 1 puntos)

La menor puntuación obtenida es `x` y la mayor puntuación es `y`

Nota: Aunque la función `max()` devuelve el máximo valor de una tupla y `min()` devuelve el mínimo, **no está permitido utilizarlas en este ejercicio**.

Ejemplo: Suponiendo que la tupla de puntuaciones es `(8,6,8,4,1)`, la salida del programa sería:

El juez 1 da al gimnasta 8 puntos

El juez 2 da al gimnasta 6 puntos

El juez 3 da al gimnasta 8 puntos

El juez 4 da al gimnasta 4 puntos

El juez 5 da al gimnasta 1 punto

La menor puntuación obtenida es 1 y la mayor puntuación es 8

Ejercicio 10. Para comprobar el tiempo que tarda en ejecutarse un programa podemos usar la librería `time` y guardar en sendas variables el tiempo de inicio de nuestro programa (`inicio = time.time()` en la primera línea del programa) y el de final (`final = time.time()` en la última línea). Imprimiendo la diferencia entre ambas, tendremos el tiempo de ejecución en segundos. Crear un programa que rellene tres listas con 100.000 valores aleatorios entre 1 y 100 de tres formas distintas:

1. Para la primera usar el método `append()`
2. Para la segunda usar el operador `+=`
3. Para la tercera usar el operador `+`

Imprimir el tiempo que tarda en rellenarse cada una de las listas. ¿Hay alguna diferencia?

Normas de entrega

Los ejercicios se deben **subir a Aula Global** hasta las 7:00 de la mañana del lunes 26 de Octubre de 2020 (Leganés) o el viernes 30 de Octubre de 2020 (Colmenarejo). Se deberá subir un fichero comprimido **zip** con un fichero para cada programa, nombrados `ejercicio1.py`, `ejercicio2.py`, etc. El nombre del fichero será "s6-iniciales-del-alumno.zip" (por ejemplo Lucía Pérez Gómez subirá un archivo llamado `s6-lpg.zip`)