

Práctica 3

Seminario de Python - Facultad de Informática - Universidad Nacional de La Plata

2020

1. Subí el ejercicio resuelto de la práctica 2 que más te costó a tu repositorio de git y compartí el link en la tarea “Ejercicio práctica2” creada en catedras.info. Sino pudiste resolverlo, indicá en qué parte tuviste dificultades como comentarios dentro del programa (aunque no funcione y/o esté incompleto). Esta actividad suma 10 Python plus.
2. Dado un texto generar una estructura que permita acceder directamente a una lista de palabras según la cantidad de caracteres de cada una. Es decir que cada palabra esté asociada al número de caracteres que tiene. **Nota:** las palabras no tienen que estar repetidas, y se debe tener en cuenta mayúsculas y minúsculas, espacios, comas y puntos.

Ejemplo, dado este texto:

```
frase = '''Si trabajás mucho CON computadoras, eventualmente encontrar
ás
que te gustaría automatizar alguna tarea. Por ejemplo, podrías desear
realizar una búsqueda y reemplazo en un gran número DE archivos de
texto,
o renombrar y reorganizar un montón de archivos con fotos de una
manera
compleja. Tal vez quieras escribir alguna pequeña base de datos
personalizada, o una aplicación especializada con interfaz gráfica,
o UN juego simple.'''
```

Se debe informar

```
{9: ['renombrar', 'reemplazo'], 8: ['gustaría', 'escribir', 'trabajás',
, 'archivos', 'búsqueda', 'gráfica,', 'compleja', 'interfaz', '
ejemplo,'], 1: ['o', 'y'], 3: ['una', 'por', 'tal', 'que', 'con', '
vez'], 11: ['encontrarás', 'automatizar', 'reorganizar'], 2: ['si',
'en', 'un', 'de', 'te'], 5: ['datos', 'juego', 'mucho', 'tarea', '
fotos'], 6: ['manera', 'alguna', 'montón', 'simple', 'número', '
texto,'], 7: ['pequeña', 'quieras', 'podrías'], 13: ['especializada',
, 'computadoras,', 'eventualmente'], 14: ['desearrealizar', '
personalizada,'], 4: ['gran', 'base'], 10: ['aplicación']}
```

3. Para registrar las actividades de un usuario en un juego dado se requiere guardar los siguientes datos: nombre del jugador, nivel alcanzado en el juego, puntaje máximo y tiempo de juego. Realizar los siguientes items con esta estructura:
 - Seleccione la estructura que considere más adecuada para almacenar la información de varios usuarios ingresados desde teclado. Tener en cuenta que se necesita acceder a un usuario determinado en forma directa sin recorrerla.
 - Con la estructura generada, imprimir los datos de un usuario dado sin recorrer la estructura. ¿La elección sobre la estructura fue adecuada? ¿Cuál usó?

- Con la estructura generada en el ejercicio, imprimir sólo los nombres de los usuarios que jugaron sin recorrer la estructura.
 - Dada la estructura generada en el ejercicio imprimir el usuario que obtuvo mayor puntaje.
 - Dada la estructura del ejercicio, ingresar los datos correspondientes a la jugada de un usuario. Si el usuario existe previamente, guardar los datos de mayor puntaje. Luego imprimir el ranking de los 10 mejores puntajes. **Nota:** utilizar una expresión **lambda** para ordenar el diccionario.
4. Dado el siguiente diccionario donde las claves son nombres de animales y los valores representan posiciones:

```
anim={'Gato Montes':2,'Yacare overo':4,'Boa acuática':5}
```

Imprimir un string por cada animal de la estructura, reemplazando sus caracteres por el string '_' (inclusive los espacios en blanco) salvo el carácter correspondiente al valor del mismo.

Ejemplo: Gato Montes tiene asociado el *valor* 2:

```
_ _ t _ _ _ _ _ _ _
0 1 2 3 4 5 6 7 8 9 10
```

5. Dada una lista con nombres de colores

```
colores = ['azul','amarillo','rojo','blanco','negro']
```

y una lista con coordenadas

```
coordenadas = [(2,3),(5,6),(8,8),(10,2),(-5,-8)]
```

- Generar una estructura que contenga coordenadas y un color asociado. La forma de asociar las coordenadas con el color debe ser aleatoria sin importar que se repitan los colores elegidos.
 - Generar una estructura que contenga coordenadas y un color asociado. La forma de asociar las coordenadas con el color debe ser aleatoria **sin que se repitan los colores**.
6. Usando el diccionario del Ejercicio 5, acceder a las coordenadas (x,y) y, según el color asociado, ejecutar una función asociada a la misma. Las funciones pueden plantear la resolución de ejercicios simples como ser:

- Suma de dos números que se generen en forma aleatoria cada vez que se llama a la función, reciba el resultado por teclado y verifique el resultado.
- Dada la estructura que contiene palabras clasificadas según su acentuación:

```
palabras = [('grave',['molesto']), ('aguda',['ratón']),
('esdrújula',['murciélago'])]
```

Cada vez que se ejecuta la función se elige una palabra en forma aleatoria, se consulta al usuario sobre el tipo de palabra que cree que es por su acentuación y se verifica la respuesta.

7. Utilizar como estructura de datos de referencia la generada en el ejercicio 3 y generar funciones que ejecuten lo siguiente:
- Imprimir los 10 primeros puntajes de la estructura.
 - Imprimir los datos de los usuarios ordenados alfabéticamente por apellido.
 - Imprimir los datos de los usuarios ordenados por nivel alcanzado.

Nota: utilice expresiones lambda para resolver los incisos.

8. Definir dos funciones que reciban una cantidad variable de argumentos: a) una función que puede llegar a recibir hasta 30 números como parámetros y debe devolver la suma total de los mismos; b) otra función que reciba un número variable de parámetros nombrados (usar `**kwargs`), e imprima dichos parámetros. De antemano no se sabe cuáles de los siguientes tres **posibles** parámetros se reciben:

- nombre
- apellido
- sexo

9. Escribir una función que reciba una cantidad variable de argumentos correspondientes a nombres de personas e imprima por pantalla los nombres enumerándolos.

```
0. Nombre 1
1. Nombre 2
etc
```

Nota: consultar el uso de *enumerate*.

10. Escribir una función que reciba al menos un argumento y opcionalmente una lista de argumentos variables y una lista de argumentos con nombre. El argumento fijo deberá ser la operación que se desea hacer con lista de números que se reciba como variable y los argumentos con nombre corresponden a los datos de la persona que solicitó la operación. Las operaciones posibles son: “+” y “*”. Los datos con nombre variables pueden ser: nombre, apellido y dirección.

Nota: investigar la función *reduce* del módulo *functools*.