

ACCESO INTELIGENTE A LA INFORMACIÓN

PRÁCTICA 1 - PROGRAMACIÓN PYTHON I

Cadenas de Caracteres

Ejercicio 1. Escribir funciones que dada una cadena y un caracter:

- a) Inserte el caracter entre cada letra de la cadena. Ej: 'separar' y ',' debería devolver 's,e,p,a,r,a,r'
- b) Reemplace todos los espacios por el caracter. Ej: 'mi archivo de texto.txt' y '_' debería devolver 'mi_archivo_de_texto.txt'
- c) Reemplace todos los dígitos en la cadena por el caracter. Ej: 'su clave es: 1540' y 'X' debería devolver 'su clave es: XXXX'
- d) Inserte el caracter cada 3 dígitos en la cadena. Ej. '2552552550' y '.' debería devolver '255.255.255.0'

Ejercicio 2. Escribir funciones que dadas dos cadenas de caracteres:

- a) Indique si la segunda cadena es una subcadena de la primera. Por ejemplo, 'cadena' es una subcadena de 'subcadena'.
- b) Devuelva la que sea anterior en orden alfabético. Por ejemplo, si recibe 'kde' y 'gnome' debe devolver 'gnome'.

Tuplas y Listas

Ejercicio 1. Campaña electoral

- a) Escribir una función que reciba una tupla con nombres, y para cada nombre imprima el mensaje Estimado <nombre>, vote por mí.
- b) Escribir una función que reciba una tupla con nombres, una posición de origen p y una cantidad n, e imprima el mensaje anterior para los n nombres que se encuentran a partir de la posición p.
- c) Modificar las funciones anteriores para que tengan en cuenta el género del destinatario, para ello, deberán recibir una tupla de tuplas, conteniendo el nombre y el género.

Ejercicio 2. Escribir una función que reciba una lista de tuplas (Apellido, Nombre, Inicial_segundo_nombre) y devuelva una lista de cadenas donde cada una contenga primero el nombre, luego la inicial con un punto, y luego el apellido.

Búsqueda

Ejercicio 1. Agenda simplificada

Escribir una función que reciba una cadena a buscar y una lista de tuplas (nombre_completo, telefono), y busque dentro de la lista, todas las entradas que contengan en el nombre completo la cadena recibida (puede ser el nombre, el apellido o sólo una parte de cualquiera de ellos). Debe devolver una lista con todas las tuplas encontradas.

Diccionarios

Ejercicio 1. Continuación de la agenda.

Escribir un programa que vaya solicitando al usuario que ingrese nombres.

- a) Si el nombre se encuentra en la agenda (implementada con un diccionario), debe mostrar el teléfono y, opcionalmente, permitir modificarlo si no es correcto.
 - b) Si el nombre no se encuentra, debe permitir ingresar el teléfono correspondiente.
- El usuario puede utilizar la cadena "**", para salir del programa.

Objetos

Ejercicio 1. Botella y Sacacorchos

- a) Escribir una clase Corcho, que contenga un atributo bodega (cadena con el nombre de la bodega).
- b) Escribir una clase Botella que contenga un atributo corcho con una referencia al corcho que la tapa, o None si está destapada.
- c) Escribir una clase Sacacorchos que tenga un método destapar que le reciba una botella, le saque el corcho y se guarde una referencia al corcho sacado.
- d) Agregar un método limpiar, que saque el corcho del sacacorchos.

Herencia y Polimorfismo

Ejercicio 1. Juego de Rol

- a) Escribir una clase Personaje que contenga los atributos vida, posicion y velocidad, y los métodos recibir_ataque, que reduzca la vida según una cantidad recibida y lance un mensaje si la vida pasa a ser menor o igual que cero, y mover que reciba una dirección y se mueva en esa dirección la cantidad indicada por velocidad.
- b) Escribir una clase Soldado que herede de Personaje, y agregue el atributo ataque y el método atacar, que reciba otro personaje, al que le debe hacer el daño indicado por el atributo ataque.
- c) Escribir una clase Campesino que herede de Personaje, y agregue el atributo cosecha y el método cosechar, que devuelva la cantidad cosechada.