**Recurso útiles:**

* **phyton para todos.pdf** (páginas 24-28)
* <http://es.wikipedia.org/wiki/Python#Listas_y_Tuplas>
* **introduccion a phyton.pdf** (páginas 40-50): incluye operaciones sobre listas y Formateo de cadenas

lista.**append**(elemento) # añade un elemento al final de la lista

lista.**extend**([elemento]) # añade un elemento al final de la lista

lista.**extend**(elemento) # añade un elemento por cada caracter del string

lista.**insert**(índice,elemento) # inserta el elemento en la posición de antes del índice

**len**(lista) # tamaño de una lista

lista.**count**(elemento) # cuenta el número de veces del elemento

lista.**index**(elemento,[start[,stop]])) # primera posición de elemento entre start y stop

**del** lista[index] # borra el elemento con ese índice (si el índice es negativo se empezará a contar por el final de la lista)

lista.**remove**(elemento) # igual que poner:

del lista[lista.index(elemento)] #igual que la línea anterior

lista.**reverse**() # ordena la lista (no devuelve una lista)

lista.**sort**() # ordena la lista

**Ejercicios**:

1. El programa tiene que pedir un número y luego solicitar tantos nombres como ha indicado en el número, crear una la lista con dichos nombres. Por último, el programa tiene que escribir la lista, print(**"La lista creada es:"**, lista)

Cuántos nombres tiene la lista: 3

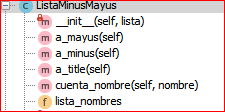
Nombre 1: Iker

Nombre 2: Pablo

Nombre 3: Josetxo

('La lista creada es:', ['Iker', 'Pablo', 'Josetxo'])

1. Crea una clase con un atributo de una lista y tenga 4 métodos que convierten a mayúsculas, minúsculas, title cada uno de los elementos y cuente cuántas veces aparece un elemento en la lista, respectivamente:



Realiza un programa utilizando la clase anterior que escribir la lista original y convertida a minúsculas, mayúsculas y title.

Pedir un nombre y escribir cuántas veces aparece en la lista.

Lista original ['Beñat', 'XaBi', 'Xabi', 'MARIA', 'Alexander', 'Carlos', 'JUAN', 'Imanol', 'PEDRO', 'Uxue', 'javier', 'iker', 'carlos', 'Xabi', 'ALEJANDRA', 'cAROLINA', 'iÑaKI', 'Asier', 'Maria']

Lista en MAYÚSCULAS ['BEÑAT', 'XABI', 'XABI', 'MARIA', 'ALEXANDER', 'CARLOS', 'JUAN', 'IMANOL', 'PEDRO', 'UXUE', 'JAVIER', 'IKER', 'CARLOS', 'XABI', 'ALEJANDRA', 'CAROLINA', 'IÑAKI', 'ASIER', 'MARIA']

Lista en minúsculas ['beñat', 'xabi', 'xabi', 'maria', 'alexander', 'carlos', 'juan', 'imanol', 'pedro', 'uxue', 'javier', 'iker', 'carlos', 'xabi', 'alejandra', 'carolina', 'iñaki', 'asier', 'maria']

Lista en Title ['Beñat', 'Xabi', 'Xabi', 'Maria', 'Alexander', 'Carlos', 'Juan', 'Imanol', 'Pedro', 'Uxue', 'Javier', 'Iker', 'Carlos', 'Xabi', 'Alejandra', 'Carolina', 'Iñaki', 'Asier', 'Maria']

Nombre a buscar: xabi

xabi aparece 3 veces

1. OPCIONAL. Tenemos una lista con los nombres de la clase. Hacer un programa que pida un número de alumnos a añadir y posteriormente pida tantos nombre como ha indicado el número y los añada a la lista(pueden repetirse). Mostrar lista.

ALTAS ELEMENTOS - PUEDEN REPETIRSE

Nº a alumnos a añadir:1

Nombre a añadir: PEPE

Beñat - Xabi - Tiago- María - Alexander - Carlos - Tiago - Imanol - Alexandrov - Uxue - Arturo - Uxue - Juan - Xabi - Alexander - Carlos - Iñaki - Asier - Maria - PEPE -

1. Añade a la clase del ejercicio02 el método nombre\_posiciones(self,nombre): el cual devolverá las posiciones en las que se encuentra el *nombre* en la lista o *False* si no está.

Hacer un programa (importando la clase del ejercicio02) pida un número de nombres a consultar en la lista y por cada uno pedir un nombre y decir cuántas veces aparece y escribir la posición que ocupa en la lista. Puede haber nombres repetidos en la lista.

CONSULTAS

['SERGIO', 'XaBi', 'Xabi', 'MARIA', 'Alexander', 'Carlos', 'JUAN', 'Imanol', 'PEDRO', 'Uxue', 'javier', 'iker', 'carlos', 'Xabi', 'ALEJANDRA', 'cAROLINA', 'iÑaKI', 'Asier', 'Maria']

Nº a alumnos a consultar:2

Nombre a buscar: aLexander

Está en las posiciones [4]

Nombre a buscar: no está

Está en las posiciones False

1. Tenemos una lista con nombres de personas. Hacer un programa que pida cuántos nombres quiere eliminar, pida nombre y elimine de la lista, hasta hacer todos los solicitados. En caso de haber dos repetidos, ¿cual borraría si utilizas cualquier método de las listas?.

BAJAS

Sergio Xabi Xabi Maria Alexander Carlos Juan Imanol Pedro Uxue Javier Iker Carlos Xabi Alejandra Carolina Iñaki Asier Maria

Nº alumnos a borrar:3

Nombre a buscar: noesta

Sergio Xabi Xabi Maria Alexander Carlos Juan Imanol Pedro Uxue Javier Iker Carlos Xabi Alejandra Carolina Iñaki Asier Maria

Nombre a buscar: xabi

Sergio Maria Alexander Carlos Juan Imanol Pedro Uxue Javier Iker Carlos Alejandra Carolina Iñaki Asier Maria

Nombre a buscar: ikEr

Sergio Maria Alexander Carlos Juan Imanol Pedro Uxue Javier Carlos Alejandra Carolina Iñaki Asier Maria

1. Importa la clase del ejercicio02 y añade un método al cual pasándole un nombre lo busque en la lista y si está, que pida un NUEVO nombre y modifique el primero (solo la primera coincidencia con el nombre). Si no está no hará nada.

MODIFICACIONES

SERGIO XaBi Xabi MARIA Alexander Carlos JUAN Imanol PEDRO Uxue javier iker carlos Xabi ALEJANDRA cAROLINA iÑaKI Asier Maria

Nombre a MODIFICAR: pedro

NUEVO nombre: ppppp

SERGIO XaBi Xabi MARIA Alexander Carlos JUAN Imanol ppppp Uxue javier iker carlos Xabi ALEJANDRA cAROLINA iÑaKI Asier Maria

---

MODIFICACIONES

SERGIO XaBi Xabi MARIA Alexander Carlos JUAN Imanol PEDRO Uxue javier iker carlos Xabi ALEJANDRA cAROLINA iÑaKI Asier Maria

Nombre a MODIFICAR: noesta

SERGIO XaBi Xabi MARIA Alexander Carlos JUAN Imanol PEDRO Uxue javier iker carlos Xabi ALEJANDRA cAROLINA iÑaKI Asier Maria

1. OPCIONAL. Tenemos una lista con los nombres de la clase. Pedir cuántos alumnos a añadir. Por cada uno pedir un nombre, si ya existe (en mayúsculas o minúsculas), mensaje “YA existe”. Si no existe añadirlo a la lista, pero poniendo la primera letra en mayúsculas. Mostrar lista.

ALTAS ELEMENTOS-sin repetirse

Ana - Pablo - Iker - Elsa - Javier - José - Francisco - Javier - Pedro - Javier -

Nº a alumnos a añadir:**2**

Nombre a añadir: **pedro**

Nombre YA existe

Nombre a añadir: **carmen**

Ana - Pablo - Iker - Elsa - Javier - José - Francisco - Javier - Pedro - Javier - **Carmen** -

1. OPCIONAL. Tenemos una lista con los nombre de la clase, pedir un nombre y borrar el nombre de la lista. Si es que está repetido muestra cada una de las repeticiones, pregunta si desea borrar. Borrar en caso afirmativo.

BAJA DE TODOS LOS ELEMENTOS QUE COINCIDAN CON UN NOMBRE

Ana Pablo Iker Elsa Javier José Francisco Javier Pedro Javier

Nombre a eliminar: javier

Javier se repite 3 veces

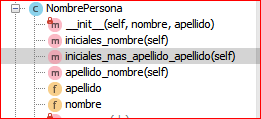
Javier Posición 5 Javier -- ¿Quieres borrarlo(S/N)? s

Javier Posición 7 Javier -- ¿Quieres borrarlo(S/N)? n

Javier Posición 9 Javier -- ¿Quieres borrarlo(S/N)? s

Ana Pablo Iker Elsa José Francisco Javier Pedro

1. Leer nombre, leer apellidos. Mostrar apellidos, nombre y mostrar iniciales del nombre y el apellido/s. Realiza el ejercicio con una clase NombrePersona, con dos atributos



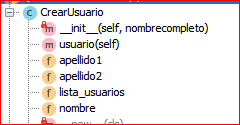
Nombre: pEdro joSé

Apellido: del aMor hermoso

Iniciales: P.J.

P.J., Del Amor Hermoso

Del Amor Hermoso, Pedro José

1. Tenemos una lista compuesta por nombre, apellidos, de empleados. A partir de esa lista generar nombres de usuario con la inicial del nombre más apellido o apellidos (8 caracteres) . OPCIONAL: en caso de que haber nombres de usuario iguales, el octavo carácter será 1, 2, etc. 

Crea una clase Crea Usuario, con el atributo nombrecompleto, que es un elemento de la lista. El método usuario devuelve el usuario de 8 caracteres. Al mismo tiempo que se crea dicho usuario, añadirlo a una lista de nombres de usuario (atributo de clase).

Lista de nombres:

['Pedro', ['Rodriguez', 'Saavedra']] ['Iker', ['Fernandez', 'Ochoa']] ['Carlos', ['Izu', 'Rodriguez']] ['Paco', ['Rodriguez', 'Ochoa']]

Lista de usuarios:

['prodrig', 'ifernan', 'cizurod', 'prodrig']

1. OPCIONAL. Tenemos dos lista con nombres y su dirección de correo electrónico. Fusiona ambas listas de manera que los mismos nombres estén en un único elemento con sus direcciones correspondientes. Crea y utiliza las clases que tú creas más conveniente.