

ejercicios

1. Hacer una lista de diccionarios que guarden información sobre alumnos, con su nombre, apellidos y número de matrícula. A partir de eso: a) ordenar la lista según sus campos. b) añadir a cada elemento de la lista un campo booleano de repetidor, que indica si han repetido curso. Si ninguno lo ha hecho, que salga por pantalla un mensaje indicándolo. Si alguno sí, que salgan por pantalla los repetidores y si todos son repetidores, que salga por pantalla un mensaje de “curso de repetición” c) Añadir un campo a cada alumno con la fecha de matriculación en el centro d) Crear un nuevo alumno, comprobar que no existe por su nombre y apellidos y añadirlo a la lista en tal caso e) Sacar un alumno de la lista si este se desmatricula buscándolo por su número de matrícula 4.
2. Crear una clase vector que tenga atributos (x, y, z) y otra clase triángulo que contenga 3 elementos de tipo vector (p1, p2, p3). a) En el constructor de triángulo, debemos generar un atributo más que sea la normal del plano del triángulo y normalizarla (que su módulo sea 1) b) Generar un método en cada clase que visualice sus atributos c) crear un método en la clase vector que normalice el vector d) crear un método que invierta un vector ([1, 2, 3] -> [-1, -2, -3])

```
In [ ]: import random
# a)
# Creamos La lista de diccionarios con Los alumnos
lst = [
    {'nombre': 'Pepe', 'apellidos': 'Martín Romero', 'Numero de matrícula': 1},
    {'nombre': 'Manolo', 'apellidos': 'Julián Vázquez', 'Numero de matrícula': 2},
    {'nombre': 'Ana', 'apellidos': 'Fernandez Narváez', 'Numero de matrícula': 3}
]
# b)
for alumno in lst:
    rep = [True, False]
    alumno['repetidor'] = random.choice(rep)

repetidores = [alumno for alumno in lst if alumno['repetidor']]

if len(repetidores) == 0:
    print('No hay repetidores')
for alumno in repetidores:
    print(alumno)

if len(lst) == len(repetidores):
    print('Este es un curso de repetición')
```

No hay repetidores

d) Crear un nuevo alumno, comprobar que no existe por su nombre y apellidos y añadirlo a la lista en tal caso

e) Sacar un alumno de la lista si este se desmatricula buscándolo por su número de matrícula

```
In [ ]: # david añadió una función para añadir campos.

def add_field(std_lst, field_name, field_lst):
    for alumno in zip(std_lst, field_lst):
        alumno[str(field_name)] = field_lst[i]
    return std_lst
```

```
In [ ]: # c)
import datetime

fecha = [datetime.datetime(2020, 9, 4), datetime.datetime(2019, 9, 17),
          datetime.datetime(2021, 9, 9), datetime.datetime(2018, 9, 1)]

for alumno in lst:
    alumno['fecha de matriculación en el centro'] = random.choice(fecha)

# d)
luis = {'nombre': 'Luis', 'apellidos': 'Martín Romero', 'Numero de matrícula': 4,
        'repetidor': False, 'fecha de matriculación en el centro': 12}
existe = False
for alumno in lst:
    if alumno['nombre'] == luis['nombre'] and alumno['apellidos'] == luis['apellidos']:
        existe = True
        break

if not existe:
    lst.append(luis)
for alumno in lst:
    print(alumno)

# e) Si se des matricula uno lo quitamos.
num_mat = 2
borrado = {}
for alumno in lst:
    if alumno['Numero de matrícula'] != num_mat:
        borrado = alumno
        break

else:
    print('No se encontró el alumno')

lst = borrado
```

{'nombre': 'Pepe', 'apellidos': 'Martín Romero', 'Numero de matrícula': 1, 'repetidor': False, 'fecha de matriculación en el centro': datetime.datetime(2019, 9, 17, 0, 0)}
{'nombre': 'Manolo', 'apellidos': 'Julián Vázquez', 'Numero de matrícula': 2, 'repetidor': False, 'fecha de matriculación en el centro': datetime.datetime(2018, 9, 1, 0, 0)}
{'nombre': 'Ana', 'apellidos': 'Fernandez Narváez', 'Numero de matrícula': 3, 'repetidor': False, 'fecha de matriculación en el centro': datetime.datetime(2019, 9, 17, 0, 0)}
{'nombre': 'Luis', 'apellidos': 'Martín Romero', 'Numero de matrícula': 4, 'repetidor': False, 'fecha de matriculación en el centro': 12}

```
In [ ]: def add_student(std_lst, alumno):
        if alumno not in std_lst:
            std_lst.append(alumno)
        return (std_lst)

def remove_student(std_lst, alumno):
    if alumno in std_lst:
        std_lst.remove(alumno)
    return (std_lst)

lst = [
    {'nombre': 'Pepe', 'apellidos': 'Martín Romero', 'Numero de matrícula': 1},
    {'nombre': 'Manolo', 'apellidos': 'Julián Vázquez', 'Numero de matrícula': 2},
    {'nombre': 'Ana', 'apellidos': 'Fernandez Narváez', 'Numero de matrícula': 3}
]

luis = {'nombre': 'Luis', 'apellidos': 'Martín Romero', 'Numero de matrícula': 4,
        'repetidor': False, 'fecha de matriculación en el centro': 12}

print(add_student(lst, luis))

print(remove_student(lst, lst[0]))
```

[{'nombre': 'Pepe', 'apellidos': 'Martín Romero', 'Numero de matrícula': 1}, {'nombre': 'Manolo', 'apellidos': 'Julián Vázquez', 'Numero de matrícula': 2}, {'nombre': 'Ana', 'apellidos': 'Fernandez Narváez', 'Numero de matrícula': 3}, {'nombre': 'Luis', 'apellidos': 'Martín Romero', 'Numero de matrícula': 4, 'repetidor': False, 'fecha de matriculación en el centro': 12}]
[{'nombre': 'Manolo', 'apellidos': 'Julián Vázquez', 'Numero de matrícula': 2}, {'nombre': 'Ana', 'apellidos': 'Fernandez Narváez', 'Numero de matrícula': 3}, {'nombre': 'Luis', 'apellidos': 'Martín Romero', 'Numero de matrícula': 4, 'repetidor': False, 'fecha de matriculación en el centro': 12}]