

Conjuntos

Ejercicio: Gestión de los participantes en 2 eventos deportivos

Realizar lo siguiente:

- Definir un conjunto para los participantes que asistieron al evento A y otro para el evento B
- Registrar nuevos participantes a cada evento
- Eliminar Participantes de cada evento
- Encontrar Participantes Comunes en los 2 eventos
- Encontrar Participantes Únicos en los 2 eventos

En este código, se gestionan los participantes de dos eventos utilizando conjuntos y se realizan varias operaciones para actualizar y analizar los participantes.

1. Definir Conjuntos:

- Se crean dos conjuntos, `evento_A` y `evento_B`, que contienen los nombres de los participantes iniciales en cada evento

```
# Definir conjuntos
evento_A = {"Ana", "Luis", "Carlos", "María"}
evento_B = {"Luis", "Jorge", "Ana", "Sofía"}
print(evento_A)
print(evento_B)
```

```
{'Ana', 'María', 'Luis', 'Carlos'}
{'Ana', 'Luis', 'Sofía', 'Jorge'}
```

2. Registrar Nuevos Participantes a Cada Evento:

- Se añaden nuevos participantes a los eventos utilizando la función `add()`. Los nuevos participantes se registran en cada evento de acuerdo con las listas `nuevos_A` y `nuevos_B`. El método `zip` se usa para iterar sobre las listas simultáneamente

```
# Registrar nuevos participantes a cada evento
nuevos_A = ["Pedro", "Julia", "Sofía"]
nuevos_B = ["Pedro", "Carlos", "Juan"]

for nuevo_A, nuevo_B in zip(nuevos_A, nuevos_B):
    evento_A.add(nuevo_A)
    evento_B.add(nuevo_B)
print(evento_A)
print(evento_B)
```

```
{'Ana', 'María', 'Luis', 'Sofía', 'Pedro', 'Julia', 'Carlos'}
{'Juan', 'Ana', 'Luis', 'Sofía', 'Pedro', 'Jorge', 'Carlos'}
```

3. Eliminar Participantes de Cada Evento:

- Se eliminan participantes de los eventos utilizando el método `discard()`. Los participantes a eliminar se especifican en las listas `eliminar_A` y `eliminar_B`. Al igual que en el paso anterior, `zip` se utiliza para eliminar simultáneamente los elementos de ambas listas

```
# Eliminar Participantes de cada evento
eliminar_A = ["Ana", "Pedro"]
eliminar_B = ["Sofía", "Pedro"]

for elim_A, elim_B in zip(eliminar_A, eliminar_B):
    evento_A.discard(elim_A)
    evento_B.discard(elim_B)
print(evento_A)
print(evento_B)
```

```
{'María', 'Luis', 'Sofía', 'Julia', 'Carlos'}
{'Juan', 'Ana', 'Luis', 'Jorge', 'Carlos'}
```

4. Encontrar Participantes Comunes en los Dos Eventos:

- Se determina la intersección de los dos conjuntos para encontrar los participantes que están en ambos eventos

```
# Encontrar Participantes Comunes en los 2 eventos
comunes = evento_A & evento_B
print(comunes)
```

```
{'Luis', 'Carlos'}
```

5. Encontrar Participantes Únicos en los Dos Eventos:

- Se identifican los participantes que están solo en uno de los eventos pero no en el otro:

- Participantes en el evento A que no están en el evento B:

```
solo_A = evento_A - evento_B
```

- Participantes en el evento B que no están en el evento A:

```
solo_B = evento_B - evento_A
```

```
# Encontrar Participantes Únicos en los 2 eventos
# Encontrar participantes en el evento A que no están en el evento B
solo_A = evento_A - evento_B
print(solo_A)

# Encontrar participantes en el evento B que no están en el evento A
solo_B = evento_B - evento_A
print(solo_B)
```

```
{'Julia', 'María', 'Sofía'}
{'Ana', 'Jorge', 'Juan'}
```