

## Estructuras de Datos (II) - set (Conjunto)

Un **conjunto** en Python es una **colección desordenada de elementos únicos**. Los conjuntos **son mutables** y se pueden modificar después de su creación. Los conjuntos se utilizan para realizar **operaciones** matemáticas como la **unión**, **intersección y diferencia**.

Para **crear un conjunto** vacío en Python, puedes usar el método `set()` sin pasarle ningún argumento. Aquí tienes un ejemplo:

```
my_set = set()

print(my_set) # set()
```

Es importante tener en cuenta que **no puedes crear un conjunto vacío usando {}** ya que **esto creará un diccionario** vacío en su lugar.

**Los conjuntos** en Python son **colecciones desordenadas de elementos únicos y no tienen índices**. Por lo tanto, no puedes acceder a los elementos de un conjunto utilizando índices como lo harías con una lista o una tupla.

Sin embargo, puedes recorrer los elementos de un conjunto utilizando un bucle `for`. Aquí tienes un ejemplo:

```
# Crear un conjunto
my_set = {1, 2, 3}

# Recorrer los elementos del conjunto
for element in my_set:
    print(element)
```

También puedes verificar si un elemento está presente (o no) en un conjunto utilizando el operador `in` y `not in`. Aquí tienes un ejemplo:

```
# Crear el conjunto
mi_conjunto = {'rojo', 'verde', 'azul'}

# Verificar si un elemento está presente en el conjunto
print('rojo' in mi_conjunto) # True
print('amarillo' in mi_conjunto) # False

# Verificar si un elemento NO está presente en el conjunto
print('amarillo' not in mi_conjunto) # True
print('rojo' not in mi_conjunto) # False
```

Algunos de los métodos más comunes para trabajar con conjuntos en Python son:

- **.add()**: Agrega un elemento al conjunto.

```
# Crear un conjunto
one = {1, 2, 3}
# Agregar un elemento
one.add(4)

print(one) # {1, 2, 3, 4}
```

- **.remove()**: Elimina un elemento del conjunto.

```
# Eliminar un elemento
one.remove(4)

print(one) # {1, 2, 3}
```

- **.clear()**: Elimina todos los elementos del conjunto.

```
# Eliminar todos los elementos
one.clear()

print(one) # set()
```

- **.discard()**: Elimina un elemento del conjunto si está presente.

```
# Descartar un elemento
one = {1, 2, 3}
one.discard(3)

print(one) # {1, 2}
```

- **.pop()**: Elimina y devuelve un elemento aleatorio del conjunto.

```
# Eliminar y devolver un elemento aleatorio
element = one.pop()

print(element) # 1 (o podría ser 2 ya que es aleatorio)
print(one) # {2} (o podría ser {1} ya que es aleatorio)
```

- **.union()**: Devuelve la unión de dos conjuntos.

```
# Crear dos conjuntos
one = {1, 2, 3}
two = {3, 4, 5}

# Unión
print(one.union(two)) # {1, 2, 3, 4, 5}
```

- **.intersection()**: Devuelve la intersección de dos conjuntos.

```
# Intersección  
print(one.intersection(two)) # {3}
```

- **.difference()**: Devuelve la diferencia entre dos conjuntos.

```
# Diferencia  
print(one.difference(two)) # {1, 2}
```

- **.symmetric\_difference()**: Devuelve la diferencia simétrica entre dos conjuntos.

```
# Diferencia simétrica  
print(one.symmetric_difference(two)) # {1, 2, 4, 5}
```