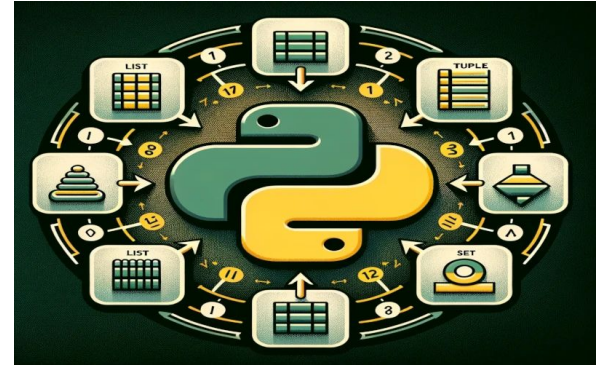


Tipo de datos Avanzados

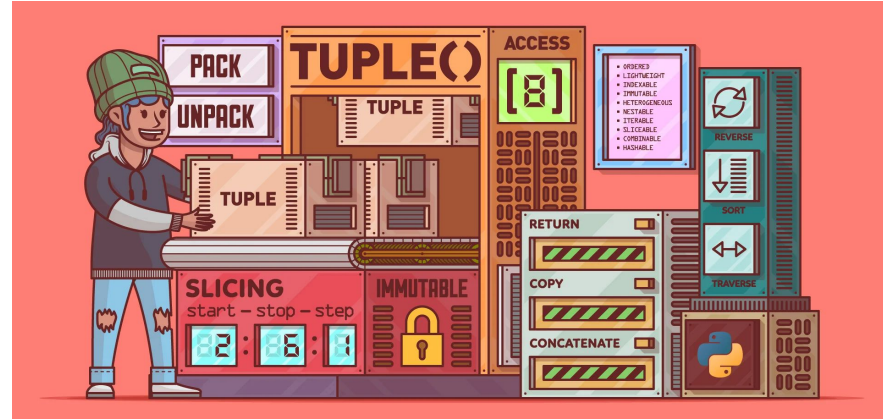
Tuplas, sets y diccionarios



Programación I

Tuplas: tuple

Las tuplas son muy similares a las listas, pero son inmutables, lo que significa que no pueden ser modificadas una vez declaradas. Las tuplas son útiles para almacenar datos que no deben cambiar a lo largo del ciclo de vida del programa.

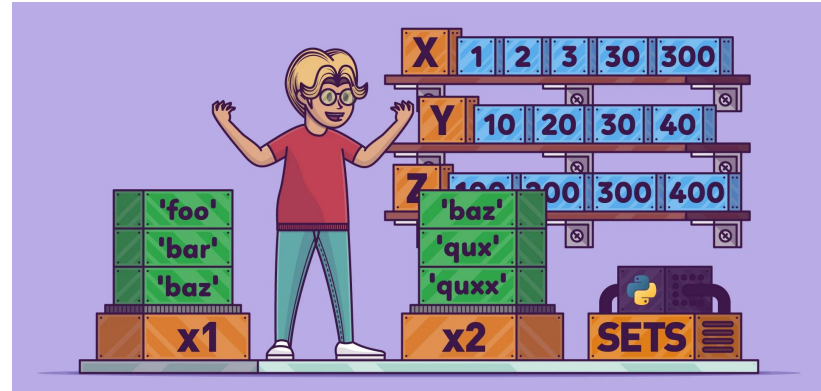


Tuplas: tuple

```
lista = tuple([1, "Hola", 3.67])  
print(type(lista))      # <class 'tuple'>  
print(lista[1])         # Hola  
lista[1] = "Chau"       # object does not support item assignment
```

Conjuntos: set

- Los elementos de un set son únicos, no contiene elementos duplicados.
- Los set no respetan el orden que tenían al ser declarados.
- Sus elementos son inmutables.



Conjuntos: set

```
set = {3, 5, 9, 5, 3, 4, 3}
print(type(set))      # <class
set>
print(set) # 9, 3, 4, 5
```

Conjuntos: **set**

Un set también se puede crear haciendo uso de la palabra reservada **set** la cual permite transformar cualquier objeto iterable en un **set**

```
lista = [1, 3, 6, 3, 2, 1]
s = set(lista)
print(s)           # {1, 2, 3, 6}
print(type(s))     # <class 'set'>
```

Método: **add**

El método **add** añade un elemento al conjunto. Si el elemento ya está en el conjunto, no hace nada.

```
set = {1, 2, 3}
set.add(4)
print(set)  #{1, 2, 3, 4}
```

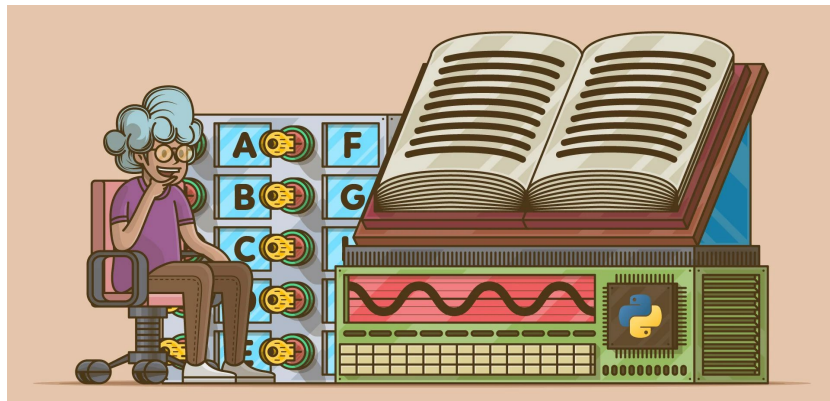
Método: **discard**

El método **discard** elimina un elemento del conjunto si está presente. Si el elemento no está presente, no hace nada.

```
set = {1, 2, 3}  
set.discard(2)  
print(set)  #{1, 3}
```

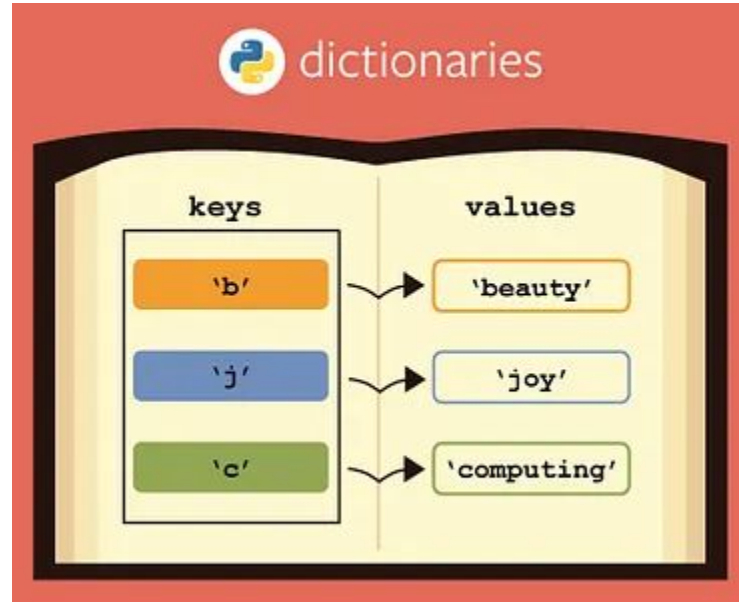

Diccionarios: **dict**

Un diccionario es una colección de elementos, donde cada uno tiene una clave (**key**) que es única y un valor (**value**).



Diccionarios: dict

```
diccionario = {'nombre': 'Juan',  
              'edad': 21,  
              'Ciudad': 'Buenos Aires'}  
print(diccionario['nombre'])  
print(diccionario['edad'])  
print(diccionario[ciudad])
```



Método: **get**

El método **get** devuelve el valor asociado a una clave. Si la clave no existe, devuelve un valor por defecto (que por defecto es `None`).

```
diccionario = {'a': 1, 'b': 2}
print(diccionario.get('a')) #1
print(diccionario.get('c')) #None
```

Método: **keys**

El método **keys** devuelve un objeto de tipo **dict_keys** de todas las claves del diccionario.

```
diccionario = {'a': 1, 'b': 2}
print(diccionario.keys()) #dict_keys(['a', 'b'])
print(list(diccionario.keys())) #['a', 'b']
```

Método: **values**

El método **values** devuelve un objeto de tipo **dict_values** de todos los valores del diccionario.

```
diccionario = {'a': 1, 'b': 2}
print(diccionario.values()) #dict_values([1, 2])
print(list(diccionario.values())) #[1, 2]
```

Método: **pop**

El método **pop** elimina y devuelve el valor asociado a una clave. Si la clave no existe, puede devolver un valor por defecto o lanzar una excepción.

```
diccionario = {'a': 1, 'b': 2}  
print(diccionario.pop('a')) #1  
print(diccionario) #{'b': 2}
```