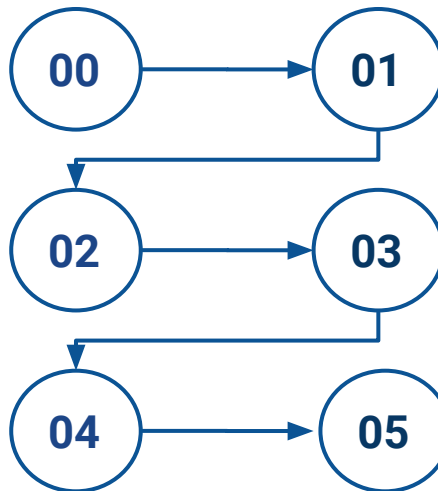


Diccionarios

Programación y Laboratorio I

Diccionarios Avanzado

¿Qué es un diccionario?



Recorrer un diccionario

get

keys / values / items

pop

clear

¿Qué es un Diccionario?

Un diccionario en Python funciona de manera similar al diccionario en el mundo real.

Las claves de un diccionario deben ser únicas y de un tipo de datos inmutable como cadenas, enteros y tuplas, pero los valores pueden repetirse y ser de cualquier tipo.

¿Qué es un Diccionario?

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

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
print(diccionario['nombre'])    # Marty  
print(diccionario['edad'])      # 18
```

Recorrer diccionarios

Al recorrer un diccionario utilizando un **for**, el valor devuelto son las claves del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
for clave in diccionario:
    print(clave,diccionario[clave])

# name Marty
# edad 18
```

El método `get()` permite consultar el valor para una clave determinada. El segundo parámetro es el valor a devolver si no se encuentra la clave (es opcional).

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
print(diccionario.get('name', 'NO NAME')) # Marty  
print(diccionario.get('nombre', 'NO NAME')) # NO NAME
```

El método `keys()` devuelve una lista con todas las claves del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
print(list(diccionario.keys()))  
# ['name', 'edad']
```

values()

El método `values()` devuelve una lista con todas los valores del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
print(list(diccionario.values()))  
# ['Marty', 18]
```


items()

El método `items()` devuelve una lista con las claves y valores del diccionario. Si se convierte en lista se puede acceder utilizando el índice.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
print(list(diccionario.items()))  
# [('name', 'Marty'), ('edad', 18)]
```

El método pop() busca y elimina la key que se pasa como parámetro y devuelve su valor asociado. Dará un error si se intenta eliminar una key que no existe.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
diccionario.pop('edad')  
print(diccionario)  
# {'name': 'Marty'}
```

También se puede pasar un segundo parámetro que es el valor a devolver si la key no se ha encontrado. En este caso si no se encuentra no habría error.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
valor_eliminado = diccionario.pop('year', 'none')  
print(valor_eliminado)  
# none
```

El método `clear()` elimina todo el contenido del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}  
diccionario.clear()  
print(diccionario) # {}
```