

# Añadir elementos a una HashTable y ordenar los elementos en Python

En Python, puedes usar un diccionario (dict) para simular una HashTable. A continuación, te muestro cómo puedes añadir elementos a un diccionario, ordenarlos según las claves o los valores, y luego imprimirlos.

## 1. Crear y añadir elementos a la HashTable (diccionario)

```
# Crear una HashTable (diccionario en Python)
hash_table = {}
```

```
# Añadir elementos a la HashTable
hash_table['clave1'] = 50
hash_table['clave2'] = 20
hash_table['clave3'] = 30
hash_table['clave4'] = 40
```

## 2. Ordenar la HashTable por claves

```
# Ordenar por claves
ordenado_por_clave = dict(sorted(hash_table.items()))
print("Ordenado por clave:")
print(ordenado_por_clave)
```

## 3. Ordenar la HashTable por valores

```
# Ordenar por valores
ordenado_por_valor = dict(sorted(hash_table.items(),
key=lambda item: item[1]))
print("Ordenado por valor:")
```

```
print(ordenado_por_valor)
```

## 4. Código completo

```
# Crear una HashTable (diccionario en Python)
hash_table = {
    'clave1': 50,
    'clave2': 20,
    'clave3': 30,
    'clave4': 40
}

# Ordenar por claves
ordenado_por_clave = dict(sorted(hash_table.items()))
print("Ordenado por clave:")
print(ordenado_por_clave)

# Ordenar por valores
ordenado_por_valor = dict(sorted(hash_table.items(),
key=lambda item: item[1]))
print("Ordenado por valor:")
print(ordenado_por_valor)
```

## Salida esperada

```
# Salida esperada
Ordenado por clave:
{'clave1': 50, 'clave2': 20, 'clave3': 30, 'clave4': 40}

Ordenado por valor:
{'clave2': 20, 'clave3': 30, 'clave4': 40, 'clave1': 50}
```