Dictionary vs Set

This lesson will discuss the key difference between Dictionary and Set in python.

We'll cover the following

Introduction

Before solving any challenges regarding Hash Tables, it is necessary to look at the implementations of **`dict`**, and **`set`** and see how they are different. Both are implemented in Python. It is also a common misconception that these two structures are the same, but they are very different from each other.

🔍 `dict`

`dict` or dictionary is a **Mapping Type** object which maps hashable values to arbitrary objects. It stores an element in the form of `key-value` pairs.

It provides the basic functionality of hashing along with some helper functions that help in the process of insertion, deletion, and search.

Some of the key features of `dict` are given below:

- An `dict` stores `key-value` pairs (examples given below) to map a key to the value:

$$abc -> 123$$

$$xyz -> 456$$

- `dict` cannot contain duplicate keys. It can, however, have duplicate values.
- `dict` does not store elements in any order either by the `key` or the `value`.
- `dict` uses a hash table for its implementation. It takes the key and then maps it into the range of hash table using the hash function.
- On average, the complexity of the basic operation is $O(1)$. It will go up to $O(n)$ in the worst-case.

🔍 `set`

`set` is a container in Python which has no duplicates. It consists of elements in no specific order. It is also built in the same way as `dict`, i.e., using the Hash Table, but it is still quite different from the `dict`.

Some of the key features of `set` are listed below:

- `set` is a container that implements the `Set` interface, and this interface only stores values, not a `key-value` pair. The `value` of an element will be its `key` at the same time.

$$1 -> 1$$

$$abc -> abc$$

- `set` does not allow storing duplicate elements as a **set** can only contain unique elements.
- On average, the complexity of the basic operation is $O(1)$. It will go up to $O(n)$ in the worst-case.

Member Functions

Some of the commonly used member functions of `set` are given below:

Function	Definition
`set1 .add (element)`	Adds `element` to the set `set1`
`set1 .remove (element)`	Removes the `element` from the set `set1`. If the element is not found then it throws an error.
`set1 - set2`	Returns difference between `set1` and `set2`
`set1 | set2`	Returns union of `set1` and `set2`
`set1 & set2`	Returns intersection of `set1` and `set2`
`key in container`	Search element with the given value `key`. If the element is present, it will return True.

Some of the commonly used member functions of `dict` are given below:

Function	Definition

Function	Definition
`dict1 [key] = value`	Adds `value` to the dictionary `dict` mapped to `key`
`del dict1[key]`	Removes the corresponding key-value pair from `dict1` with the key `key` .
`key in dict1`	Search element with the given key. If the element is present, it will return True.

In the following lessons, we will use the in-built Python hash table to solve popular interview questions.

← Back

Trees vs Hash Table

Next →

Challenge 1: A List as a Subset of Ano...

☑ Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/dictionary-vs-set__introduction-to-hashing__data-structures-for-coding-interviews-in-python)