HashMap in Python

Hashmaps are data structures that are used to store data in a key-value format. They are commonly used because they provide fast lookup times, which can be important in applications where you need to quickly retrieve data. Additionally, they can be used to implement other data structures, such as sets and caches.

In Python, a hashmap is implemented as a dictionary. The keys in a dictionary are hashed, which means that they are transformed into a fixed-size integer value, called a hash code. The hashmap uses this hashcode to store value on the specific index of the hash table and uses a key to search for the value on that specific index. When you look up a value in a dictionary by its key, the key is first passed through a hash function.

Python hashmap vs dictionary

In Python, “hashmap” and “dictionary” refer to the same data structure. The dictionary data type is implemented as a hash table, which allows for fast access to values using keys. Dictionaries are used to store key-value pairs, where each key must be unique and immutable.
It’s worth noting that while in Python, the terms “hashmap” and “dictionary” refer to the same built-in data structure, in other programming languages the two terms may not be interchangeable.