**Hash Map:**

Hash maps are indexed data structures. A hash map makes use of a [hash function](https://www.geeksforgeeks.org/what-are-hash-functions-and-how-to-choose-a-good-hash-function/) to compute an index with a key into an array of buckets or slots. Its value is mapped to the bucket with the corresponding index. The key is unique and immutable. Think of a hash map as a cabinet having drawers with labels for the things stored in them. For example, storing user information- consider email as the key, and we can map values corresponding to that user such as the first name, last name etc to a bucket.

Hash function is the core of implementing a hash map. It takes in the key and translates it to the index of a bucket in the bucket list. Ideal hashing should produce a different index for each key. However, collisions can occur. When hashing gives an existing index, we can simply use a bucket for multiple values by appending a list or by rehashing.

In Python, dictionaries are examples of hash maps. We’ll see the implementation of hash map from scratch in order to learn how to build and customize such data structures for optimizing search.

The hash map design will include the following functions:

* **set\_val(key, value):** Inserts a key-value pair into the hash map. If the value already exists in the hash map, update the value.
* **get\_val(key):** Returns the value to which the specified key is mapped, or “No record found” if this map contains no mapping for the key.
* **delete\_val(key):** Removes the mapping for the specific key if the hash map contains the mapping for the key.

Time Complexity:

Memory index access takes constant time and hashing takes constant time. Hence, the search complexity of a hash map is also constant time, that is, O(1).

Advantages of HashMaps

* Fast random memory access through hash functions
* Can use negative and non-integral values to access the values.
* Keys can be stored in sorted order hence can iterate over the maps easily.

Disadvantages of HashMaps

* Collisions can cause large penalties and can blow up the time complexity to linear.
* When the number of keys is large, a single hash function often causes collisions.

Applications of HashMaps

* These have applications in implementations of Cache where memory locations are mapped to small sets.
* They are used to index tuples in Database management systems.
* They are also used in algorithms like the Rabin Karp pattern matching