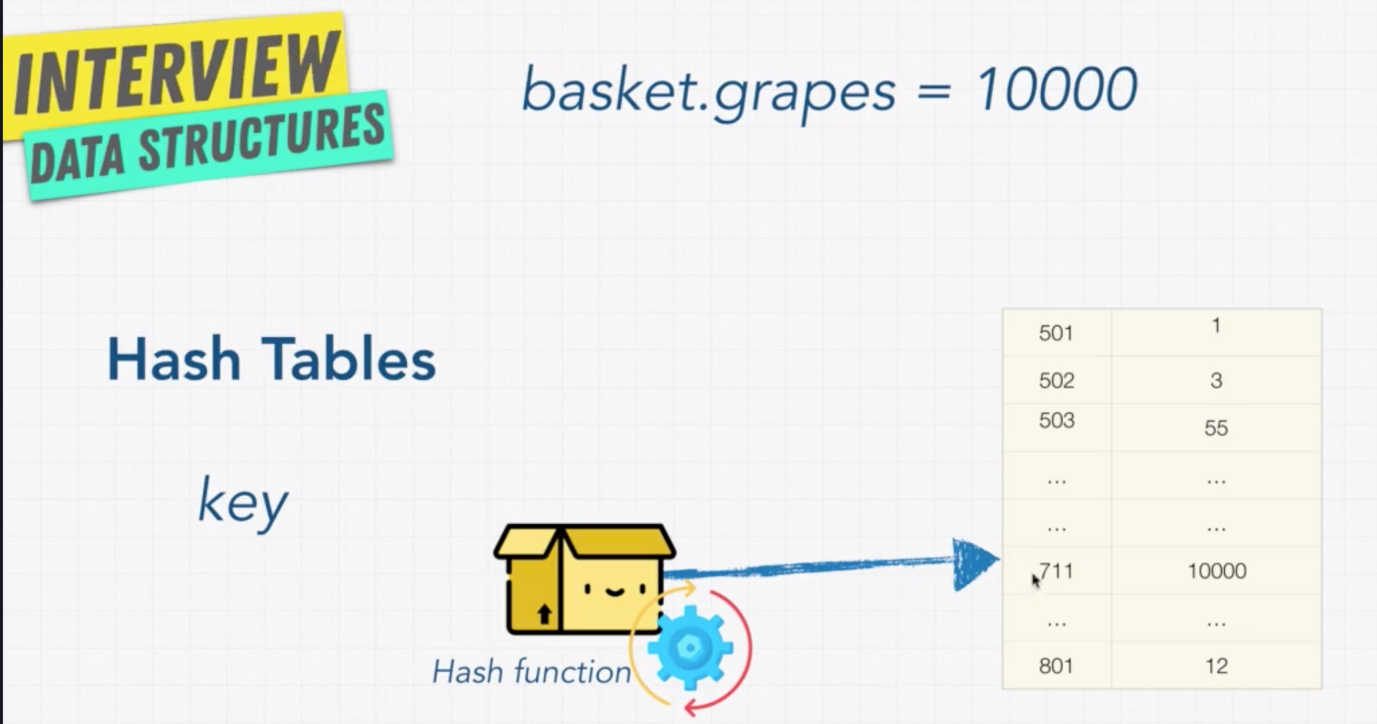
**Hash Tables**

* **Hash Tables Introduction:**

****

****

Note: Hash table stores the key value pairs after computing the hash value.

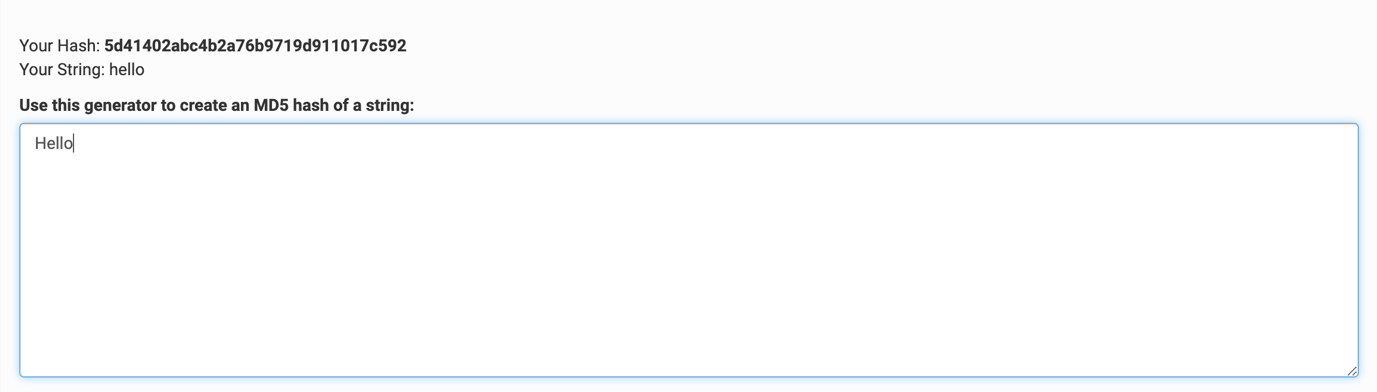
It decides where to store the given hash in the table.

* **Hash Function:**

Hash function always generate the ***fixed length of output*** for given input.

Hash function is ***one-way***. Means you can only generate the Hashes, once it is generated you can’t decode it to get what is the input used to get the hash.

No matter how many times you generate the hash for the same input, the output will be the same always.



Graphical user interface, text, application

Description automatically generated

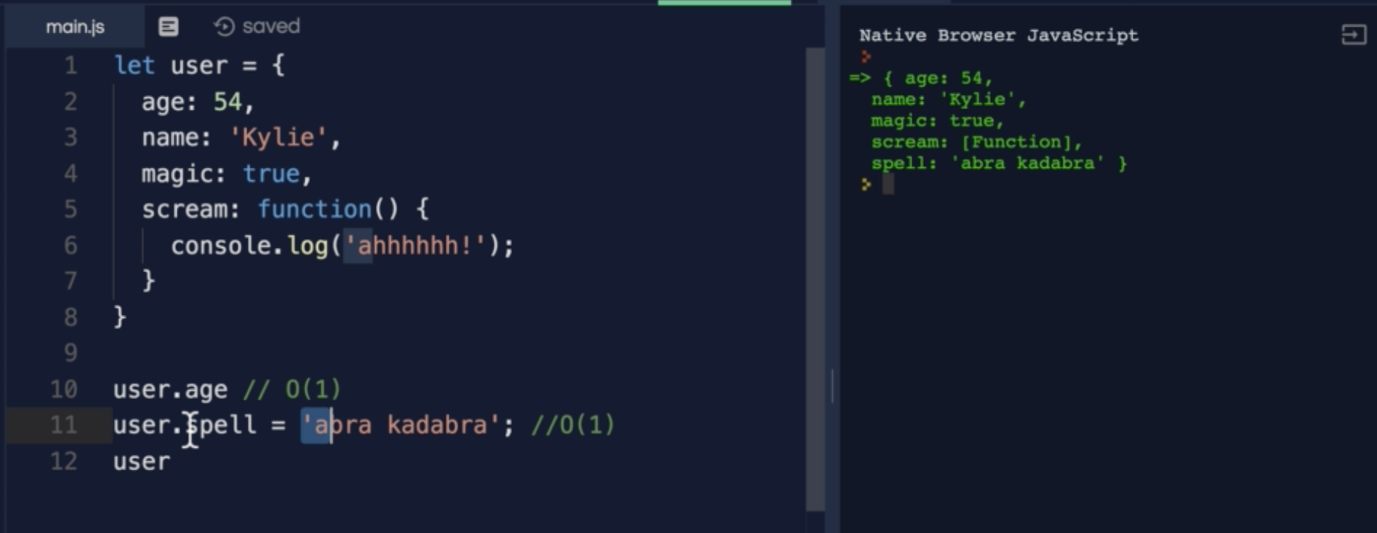
This is mainly used in DS for fast data access.

Whenever we do an operation with Hash Tables, there will be a time associated as it has to call the hash function to get the hash. But every language has a default hash mechanism, which works pretty fast.

* **Hash Collisions:**

**Table

Description automatically generated**

****

****

As we know, all operations on Hash Table is O(1), then why don’t we use it everywhere?

Well, all data structures will have pros and cons, even Hash Table has a con, let’s see that in detail.

Use this URL to see the hashing example:

<https://www.cs.usfca.edu/~galles/visualization/OpenHash.html>

Diagram

Description automatically generated

Note: Number 55 and 3 are placed in the same location.

A picture containing diagram

Description automatically generated

Whatever happened in above example (placing multiple values at same location) is called collision.

Diagram

Description automatically generated

Text

Description automatically generated

Whenever the collision happens, it slows down reading and writing.

O(n/k) k – size of Hash table 🡺 Simplified to O(n) operation.

There are two ways to deal with collision, one is using the linked-list.

To find more ways check resolution types explained in Wikipedia.

* **Hash Tables vs Arrays:**

**Table

Description automatically generated**