Assign

7 min

We now have everything we need to find a place in the hash map array to store a value. The only thing left to do is assign the value to the index we generated. A method, .assign() will handle the logic needed to take in a key-value pair and store the value at a particular index.

A general outline of how .assign() will work is this:

store the hashed key in a variable arrayIndex

assign the value to the element at arrayIndex in the hash map

Instructions

1. Checkpoint 1 Passed

1.

Declare a HashMap method called .assign() with the parameters key and value.

2. Checkpoint 2 Passed

2.

Declare a constant called arrayIndex with the value of the hashed and compressed key.

Hint

Use the HashMap class method that hashes and compresses keys.

3. Checkpoint 3 Passed

3.

Assign the value to the element at the index you derived from hashing, arrayIndex.

Hint

The hash map array can be accessed through the .hashmap property.

4. Checkpoint 4 Passed

4.

Check your work. At the bottom of the **HashMap.js** file store a new instance of HashMap with a size of 3 in a constant named employees. Assign employees the key-value pair '34-567' and 'Mara', then log the hash map.

Hint

The hash map's array can be accessed through the .hashmap property.

HashMap.js

```
class HashMap {
  constructor(size = 0) {
    this.hashmap = new Array(size)
      .fill(null);
  }

  hash(key) {
    let hashCode = 0;
    for (let i = 0; i < key.length; i++) {
      hashCode += hashCode + key.charCodeAt(i);
    }
    return hashCode % this.hashmap.length;
  }

  assign(key, value) {
    const arrayIndex = this.hash(key);
    this.hashmap[arrayIndex] = value;
  }
}

module.exports = HashMap;

const employees = new HashMap(3);
employees.assign('34-567', 'Mara');

console.log(employees.hashmap);
```