

## Retrieve

5 min

To be a fully functional hash map, we have to be able to retrieve the values we are storing. To implement retrieval for our hash map we'll create a new HashMap [method](#), `.retrieve()`.

This method will make use of `.hash()`'s deterministic nature to find the value we're looking for in the hash map.

### Instructions

1. Checkpoint 1 Passed

#### 1.

Define a method `.retrieve()` for HashMap. It should have one parameter, `key`, the key of the value we want to retrieve.

2. Checkpoint 2 Passed

#### 2.

`.retrieve()` should calculate the array index in the same way `.assign()` does and then retrieve the value at that index.

Inside of `.retrieve()` declare a constant `arrayIndex` with the value of the hashed key. Use the HashMap method that takes a key and returns an index in the hash map's array.

Hint

The HashMap method `.hash()` takes a key and returns a valid index in the hash map's array.

3. Checkpoint 3 Passed

#### 3.

Return the value stored at `arrayIndex`.

Hint

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

4. Checkpoint 4 Passed

#### 4.

Check your work. At the bottom of the **HashMap.js** file declare a new constant `glossary` that stores a hashmap with a size of 3.

- Add a new key of: 'semordnilap'
- With a value of: 'Words that form different words when reversed'

Log the result of retrieving 'semordnilap' from your glossary.

## 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;  
  }  
  
  retrieve(key) {  
    const arrayIndex = this.hash(key);  
    return this.hashmap[arrayIndex];  
  }  
}  
  
module.exports = HashMap;  
  
const glossary = new HashMap(3);  
glossary.assign('semordnilap', 'Words that form different words when reversed');  
console.log(glossary.retrieve('semordnilap'));
```

