

Lesson 03 Demo 06

Declaring and Initializing a HashMap

Objective: To demonstrate HashMap operations in JavaScript by declaring, initializing, and using key-value pairs to store and retrieve data efficiently for dynamic data access

Tools required: Visual Studio Code and Node.js

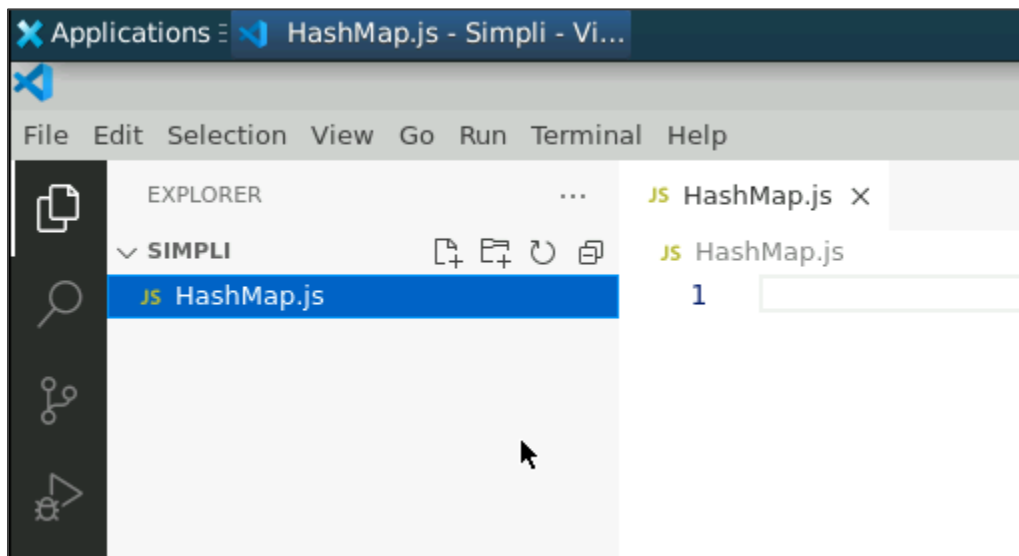
Prerequisites: A basic understanding of data structures and JavaScript

Steps to be followed:

1. Create a JavaScript file and execute it

Step 1: Create a JavaScript file and execute it

- 1.1 Open the Visual Studio Code editor and create a JavaScript file named **HashMap.js**



1.2 Add the following code to the file:

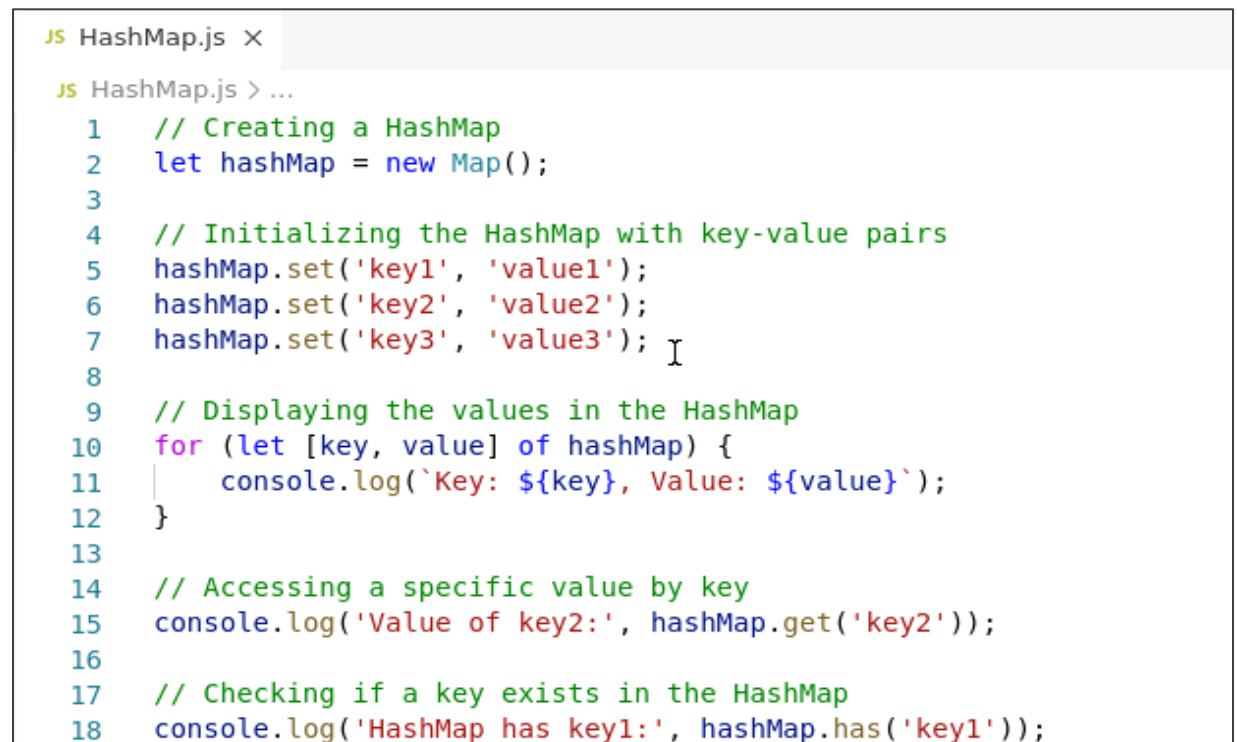
```
// Creating a HashMap
let hashMap = new Map();

// Initializing the HashMap with key-value pairs
hashMap.set('key1', 'value1');
hashMap.set('key2', 'value2');
hashMap.set('key3', 'value3');

// Displaying the values in the HashMap
for (let [key, value] of hashMap) {
  console.log(`Key: ${key}, Value: ${value}`);
}

// Accessing a specific value by key
console.log('Value of key2:', hashMap.get('key2'));

// Checking if a key exists in the HashMap
console.log('HashMap has key1:', hashMap.has('key1'));
```

A screenshot of a code editor window. The title bar at the top says "JS HashMap.js X". The editor content shows the same JavaScript code as the previous block, with line numbers 1 through 18 on the left. The code is color-coded: comments are green, keywords like 'let', 'new', 'for', 'of', 'console.log' are blue, and string literals are red. The cursor is positioned at the end of line 7, which is 'hashMap.set('key3', 'value3');'.

```
JS HashMap.js X
JS HashMap.js > ...
1  // Creating a HashMap
2  let hashMap = new Map();
3
4  // Initializing the HashMap with key-value pairs
5  hashMap.set('key1', 'value1');
6  hashMap.set('key2', 'value2');
7  hashMap.set('key3', 'value3');
8
9  // Displaying the values in the HashMap
10 for (let [key, value] of hashMap) {
11   console.log(`Key: ${key}, Value: ${value}`);
12 }
13
14 // Accessing a specific value by key
15 console.log('Value of key2:', hashMap.get('key2'));
16
17 // Checking if a key exists in the HashMap
18 console.log('HashMap has key1:', hashMap.has('key1'));
```

1.3 Press **Ctrl + S** to save the file and execute it in the **TERMINAL** using the commands given below:

ls

node HashMap.js

```
9 // Displaying the values in the HashMap
10 for (let [key, value] of hashMap) {
11     console.log(`Key: ${key}, Value: ${value}`);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
priyanshurajsim@ip-172-31-39-132:~/Downloads/Simpli$ ls
HashMap.js
priyanshurajsim@ip-172-31-39-132:~/Downloads/Simpli$ node HashMap.js
Key: key1, Value: value1
Key: key2, Value: value2
Key: key3, Value: value3
Value of key2: value2
HashMap has key1: true
priyanshurajsim@ip-172-31-39-132:~/Downloads/Simpli$ █
```

By following these steps, you have effectively learned to create and manipulate a HashMap in JavaScript to store and retrieve data efficiently for dynamic data access, improving your ability to efficiently manage key-value pair data structures.