JavaScript Maps | JavaScript Tutorial In Hindi #57

Today we will study **maps in JavaScript**, which was introduced in ES2015. Before ES6, JavaScript did not have a data structure for dictionaries. Programmers use objects as dictionaries from strings to arbitrary values. ES6 introduces Maps, which are dictionaries from arbitrary values to arbitrary values. In this tutorial, we will learn how to work with the JavaScript Map object and its useful methods like get(), set(), clear() etc., to manipulate entries in the map.

What are Maps in JavaScript ?

Maps are a new object type that allows us to store collections of key-value pairs. Map keys can be of any type, even objects or functions. In the case of maps, it is also easy to get the size of a map, whereas, it is not as straightforward in case of objects. Along with that, in maps, we can iterate in the order in which the values were added. Here is the syntax of maps in JavaScript:

Syntax:

```
let mymap= new Map([iterable])
```

Parameter:

iterable - It is any iterable object whose values are stored as key, value pair. Providing parameters is optional. If the parameter is not specified, then a new map is created Empty.

Returns: A new Map object

Methods and Properties of Maps:-

size: It returns the number of elements or the key-value pairs in the map.

set(): This method adds the key and value to the Map Object. Here is the syntax:

```
mymap1.set(x, y);
```

"x" is the key of the element to be added to the Map and "y" is the value of the element to be added to the Map. It returns a Map object

has(): This method returns a boolean value depending on whether the specified *key* is present or not. Here is the syntax:

```
mymap1.has(x);
```

Here "x" is the key of the element to checked. It will return true if the element with the specified key is present or else returns false.

get(): This method returns the *value* of the corresponding *key. Here is the s*yntax:

```
map1.get(x);
```

Here "x" is the key, whose value is to be returned. It will return the value associated with the key if it is present in Map, otherwise returns undefined

delete(): This method deletes both the *key* as well as a *value* from the map. Here is the syntax:

```
map1.delete(k);
```

Here "k" is the key which is to be deleted from the map. It will return true if the value is found and deleted from the map; otherwise, it returns false

clear(): This method will remove all the elements from the Map object. Here is the syntax:

```
map1.clear();
```

It requires no parameters and return undefined.

Example:-

In this example, we initialize a map from an array that contains arrays of three values:

```
const students = [
  ['1', 'Harry'],
  ['2', 'Joe'],
  ['3', 'Peter'],
];
let myMap = new Map(students);
myMap.get('2'); //Output: "Joe"
```

Code file tut57.js as described in the video

```
console.log("This is tutorial 57");

// Maps in JavaScript: We can use any type of key or value
const myMap = new Map();

const key1 = 'myStr', key2 = {}, key3 = function () { };

// Setting map values
myMap.set(key1, 'This is a string');
myMap.set(key2, 'This is a blank obj');
myMap.set(key3, 'This is an empty function');
console.log(myMap);

// Getting the values from a Map
let value1 = myMap.get(key3);
console.log(value1);

// Get the size of the map
console.log(myMap.size);

// You can loop using for..of to get keys and values
for (let [key, value] of myMap) {
    console.log(key, value);
}

// Get only keys
for (let key of myMap.keys()) {
    console.log('key is ', key);
}
// Get only values
for (let value of myMap.values()) {
    console.log('value is ', value);
}

// You can loop through a map using for each loop
myMap.forEach((value, key)=>{
    console.log('Key is ', key);
```

```
    console.log('Value is ', value);
})

// Converting map to an array
let myArray = Array.from(myMap);
console.log('Map to array is ', myArray);

// Converting map keys to an array
let myKeysArray = Array.from(myMap.keys());
console.log('Map to keys array is ', myKeysArray);

// Converting map values to an array
let myValuesArray = Array.from(myMap.values());
console.log('Map to values array is ', myValuesArray);
```

Previous Next