



Education
@topdev_media



Javascript

Complete Map

CHEATSHEET

Save it for later !

WHAT IS A MAP IN JAVASCRIPT ?

A Map is a built-in object in JavaScript that allows you to store key-value pairs, where:

- Keys can be of any data type (including objects, functions, and primitives).
- Maps maintain the insertion order of their entries.
- It is designed to provide better performance and flexibility than regular JavaScript objects for key-value storage.

Key features:

1. Keys are unique: A key can only have one associated value in the Map.
2. Order of entries: Keys are iterated in the order they were added.
3. Flexible keys: Unlike objects, keys in a Map can be any type, not just strings or symbols

CREATION AND INITIALIZATION JAVASCRIPT

```
// Create an empty Map
const map = new Map();

// Create a Map with key-value pairs
const map = new Map([
  ['key1', 'value1'],
  ['key2', 'value2'],
  ['key3', 'value3']
]);
```

MAIN METHODS

Add an entry

```
map.set('key1', 'value1'); // Adds or updates a key with a value
```

Get a value

```
map.get('key1'); // Returns 'value1', or undefined if  
the key does not exist
```

Check if a key exists

```
map.has('key1'); // Returns true if 'key1' exists,  
otherwise false
```

Delete an entry

```
map.delete('key1'); // Removes 'key1' from the Map,  
returns true if deleted
```

Clear all entries

```
map.clear(); // Empties the Map completely
```


PROPERTIES

Size of the Map

```
map.size; // Returns the number of entries in the Map
```

ITERATION

Iterate over keys

```
for (const key of map.keys()) {  
  console.log(key); // Logs each key  
}
```

Iterate over values

```
for (const value of map.values()) {  
  console.log(value); // Logs each value  
}
```

Iterate over key-value pairs

```
for (const [key, value] of map.entries()) {  
  console.log(`${key}: ${value}`); // Logs 'key: value'  
}
```

Using forEach

```
map.forEach((value, key) => {  
  console.log(`${key}: ${value}`);  
});
```

CONVERSIONS

Convert a Map to an array

```
const array = Array.from(map); // [[key1, value1],  
[key2, value2], ...]
```

Convert an array to a Map

```
const array = [['key1', 'value1'], ['key2', 'value2']];  
const map = new Map(array);
```

Convert a Map to an object

```
const obj = Object.fromEntries(map); // { key1: 'value1',  
key2: 'value2' }
```

Convert an object to a Map

```
const obj = { key1: 'value1', key2: 'value2' };  
const map = new Map(Object.entries(obj));
```


COMPARISON WITH AN OBJECT

Feature	Map	Object
Key types	Can be any type	Only strings or symbols
Key order	Insertion order guaranteed	Not guaranteed
Size	size to get the number of entries	Requires iterating through all keys
Performance	Optimized for additions/deletions	Less optimized for these operations

USES CASES

- Managing collections of unique key-value pairs where keys are not strings.
- Scenarios requiring guaranteed insertion order.
- Optimized searching and manipulation.

BEST PRACTICES

- Use Map if you need to store key-value pairs with non-string keys.
- Prefer Map when the collection is large or frequent additions and deletions are required.
- Use Object for simple data structures or static data.