Javascript useful console methods()

Part (2)

Table of contents
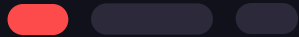
Error method()

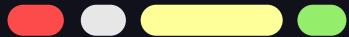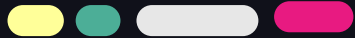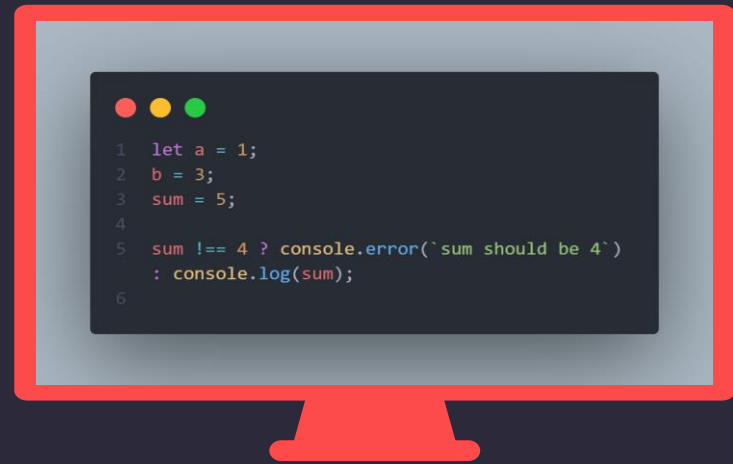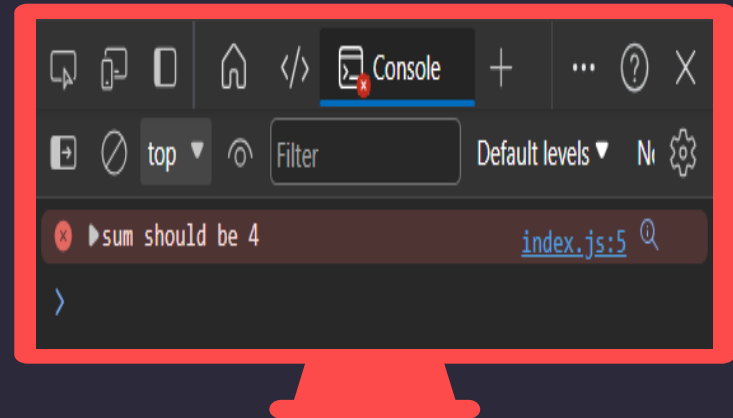In JavaScript, **console.error()** is a method used to log error messages to the console. The console object is part of the browser's debugging tools and provides methods for logging information, warnings, and errors. The **console.error()** method specifically is used to output error messages to the console, and it is typically formatted in a way that makes it stand out from regular log messages.
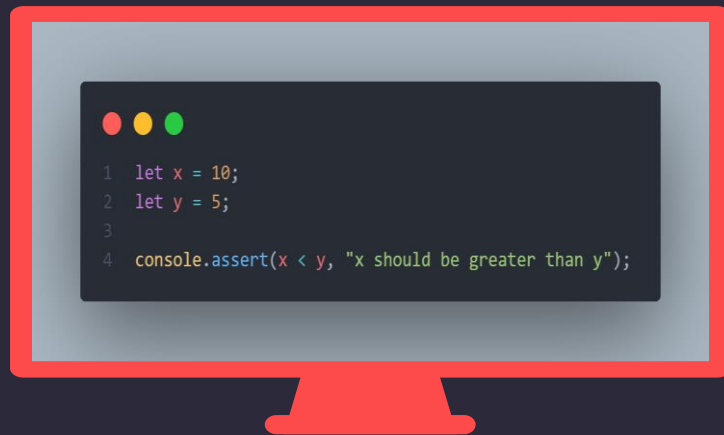
```
1   let a = 1;
2   b = 3;
3   sum = 5;
4
5   sum !== 4 ? console.error(`sum should be 4`)
      : console.log(sum);
6
```

Output :

```
Console
top ▼   Filter   Default levels ▼

⊗  ▶ sum should be 4                    index.js:5

>
```

Assert method()

The **console.assert()** method is another function provided by the console object in JavaScript, and it is used for debugging purposes. This method takes an expression as its first argument, and if the expression evaluates to false, it logs an error message to the console.

```javascript
1  let x = 10;
2  let y = 5;
3
4  console.assert(x < y, "x should be greater than y");
```

Output :

Console

top ▼ Filter Default levels ▼ No Issues

Assertion failed: x should be greater than y   index.js:10

>
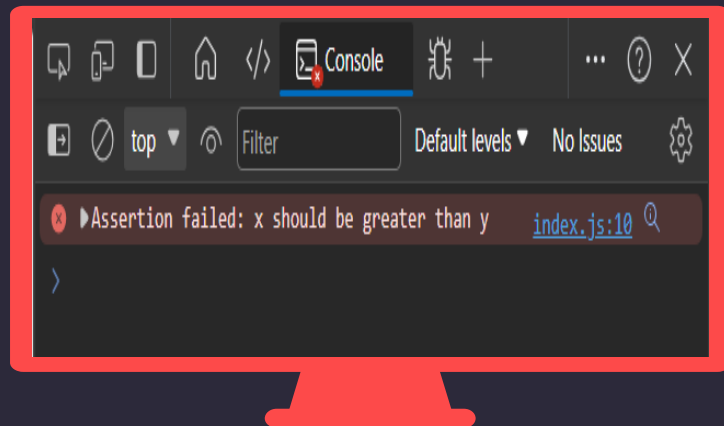
Group method()

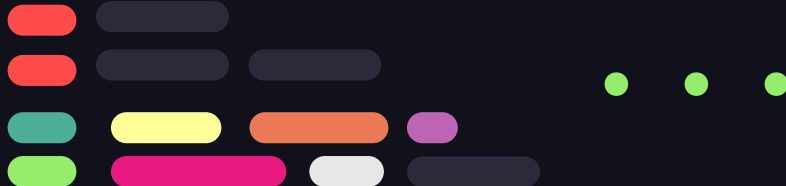**console.group()** method in JavaScript, it is used to group together related log messages in the console, providing a more organized and structured output. This is often helpful when dealing with complex code or debugging large applications.

```javascript
console.group("Group 1");
console.log("Message 1");
console.log("Message 2");
console.groupEnd();

console.group("Group 2");
console.log("Message 3");
console.log("Message 4");
console.groupEnd();
```

Output :
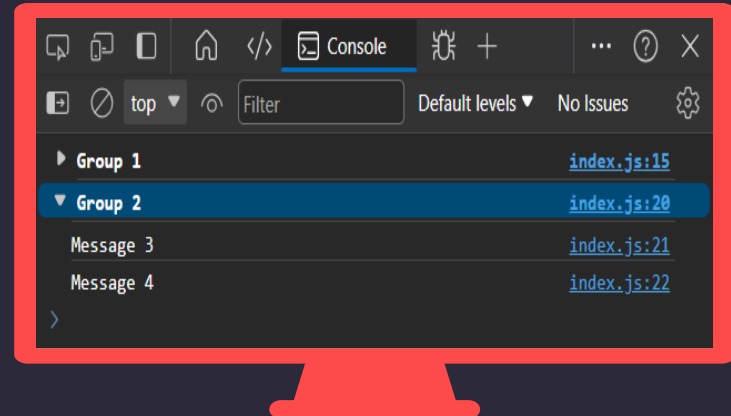
Table method()

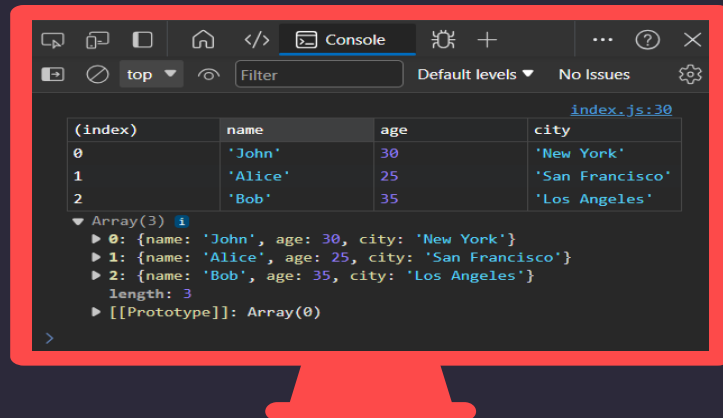The **console.table()** method in JavaScript is used to display tabular data in a tabular format within the console. It is particularly useful for visualizing arrays or objects with structured data. When you pass an array or an object to **console.table()**, it presents the data in a table, making it easier to interpret.

```javascript
const data = [
  { name: "John", age: 30, city: "New York" },
  { name: "Alice", age: 25, city: "San Francisco" },
  { name: "Bob", age: 35, city: "Los Angeles" },
];

console.table(data);
```

Output :

(index)	name	age	city
0	'John'	30	'New York'
1	'Alice'	25	'San Francisco'
2	'Bob'	35	'Los Angeles'

```
▼ Array(3) ⓘ
  ▶ 0: {name: 'John', age: 30, city: 'New York'}
  ▶ 1: {name: 'Alice', age: 25, city: 'San Francisco'}
  ▶ 2: {name: 'Bob', age: 35, city: 'Los Angeles'}
    length: 3
  ▶ [[Prototype]]: Array(0)
```
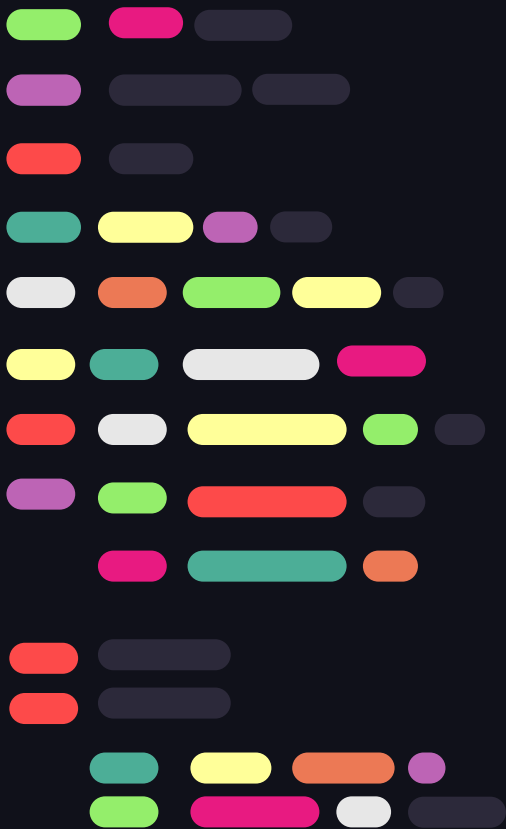
Abdelrahman hassan

Front-end (React)
Developer

Thanks!

< Do you have any questions? >