JavaScript and Related Concepts[FSRNL 38]

Taught By Anshika Mam

October 14, 2023

1 Event Loop Revision

In asynchronous JavaScript, if there is any step that takes time (e.g., setTimeout), JavaScript will skip that step and move to the next one. In the macrotask queue, all promises go. In the callback queue, you find web APIs and setTimeout functions. Promises have higher priority than setTimeout. The fetch method is used to make API calls and always returns a promise. To convert it to JSON format, you use Response.json(), which returns another promise. You can then resolve it with .then() and display the data. If there's an error, catch it and log it to the console.

2 Async/Await

The async/await syntax in JavaScript makes working with promises easier. The `await` keyword waits for promises to be either fulfilled or rejected. You can't use `await` without the `async` keyword. Axios is another popular library for making HTTP requests in a more user-friendly way than the fetch method.

3 Objects in JavaScript

```
let user = {
  firstname: "Syed",
  lastname: "Israr",
  printname() {
    console.log(`${this.firstname} ${this.lastname}`);
  },
  getname: () => {
    // Arrow function in an object doesn't have its "this
    ," it will point to the global context.
    console.log(this.firstname + " " + this.lastname);
    console.log(`${this.firstname} ${this.lastname}`);
  },
```

```
}
```

Listing 1: JavaScript Object

In JavaScript, the `this` keyword refers to the object that calls the function. When a function is declared inside an object, `this` refers to the object itself.

4 Call, Apply, and Bind

```
var logname = function(snack, hobby) {
  console.log(this.printname() + " is Cool");
  console.log("snack", snack);
  console.log("hobby", hobby);
}

var name1 = logname.bind(user); // Now "name1" is a
   version of "logname" with "this" bound to "user."
name1("Chips", "Coding");

logname.call(user, "Chips", "Coding"); // Call is similar
    to bind but it calls the function immediately.

logname.apply(user, ["Chips", "Coding"]); // Apply is
   also similar to call but it takes an array of
   arguments.
```

Listing 2: Call