

Module 37.5-Bonus API and JS Recap Concepts

মাইলস্টোন ৬ টেকএণ্ডয়ে

এই মাইলস্টোন থেকে তুমি যদি আটটা জিনিস শিখতে চাও তাহলে নিচের এই আটটি জিনিস আরেকবার ভালো করে দেখে নাও-

১. fetch বা async await ইউজ করে API থেকে কিভাবে ডাটা লোড করতে হয়। ডাটা অনেক সময় অনেকভাবে থাকে। সেই ডাটা কোনটা কখন array কখন অবজেক্ট এর ভিতরে আছে। সেটা বুঝে সেই অনুসারে ডাটা দেখানোর সিস্টেম

২. arrow ফাংশন কিভাবে ইউজ করা হয়

৩. template string এ ডাইনামিকভাবে কিভাবে ডাটা যোগ করতে হয়

৪. map, forEach, filter, find এইগুলো কখন কোনটা ব্যবহার করতে হয়, এদের মধ্যে পার্থক্য কি

৫. let, const, var এদের মধ্যে ডিফারেন্স কি, কোন কোনটা ইউজ করতে হয়।

৬. কোনটা দিয়ে array এর মধ্যে লুপ করতে হয়, কোনটা দিয়ে অবজেক্ট এর মধ্যে লুপ করতে হয়

৭. spread কিভাবে ইউজ করা হয়, স্প্রেড অপারেটর দিয়ে কিভাবে array কপি করে ফেলে।

৮. ES6 এর মধ্যে কিভাবে অবজেক্ট বা array এর destructure করে সেটা থেকে ভেরিয়েবল ডিক্লেয়ার করতে হয়।

এই মাইলস্টোন থেকে তুমি যদি আরো দশটা জিনিস এ খেয়াল রাখতে চাও তাহলে সেগুলো হবে-

১. JSON কি জিনিস, এইটা দিয়ে কি করে ?

২. == আর === এর ডিফারেন্স কি ?

৩. Block scope এবং Hoisting সম্পর্কে হালকা হলেও ধারণা

৪. closure স্কোপ সম্পর্কে ধারণা

৫. Truthy আর Falsy সম্পর্কে হালকা একটু ধারণা

৬. কোন কোন ক্ষেত্রে undefined হয় এবং null আর undefined এর মধ্যে পার্থক্য কি ?

৭. GET আর POST এর মধ্যে পার্থক্য কি ?

৮. class আর অবজেক্ট কি জিনিস

৯. bind, call, apply এর পার্থক্য কি ?

১০. জাভাস্ক্রিপ্ট এর this সম্পর্কে ধারণা

37_5-1 Get API key and load temperature data by city name

37_5-2 Display temperature, city, dynamic weather icon

```
const API_KEY = `e47206a03983fbf8a7804bf340282e1c`;
const searchTemperature = () => {
  const city = document.getElementById("city-name").value;

  const url =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=metric`;

  fetch(url)
    .then((res) => res.json())
    .then((res) => displayTemperature(res));
};

const setInnerText = (id, text) => {
  document.getElementById(id).innerText = text;
};

const displayTemperature = (temperature) => {
  setInnerText("city", temperature.name);
  setInnerText("temperature", temperature.main.temp);
  setInnerText("condition", temperature.weather[0].main);

  // set weather icon
  const url =
`http://openweathermap.org/img/wn/${temperature.weather[0].icon}@2x.png`;
  const imgIcon = document.getElementById("weather-icon");
  imgIcon.setAttribute("src", url);
};
```

37_5-3 Understanding closure and return multiples from function

In JavaScript, a closure is **a function that references variables in the outer scope from its inner scope**. The closure preserves the outer scope inside its inner scope. To understand the closures, you need to know how the lexical scoping works first.

37_5-4 (advanced) Hoisting, block scope, scope chain

JavaScript Hoisting refers to **the process whereby the interpreter appears to move the declaration of functions, variables or classes to the top of their scope, prior to execution of the code**.

Block scoped variables: A block scoped variable means that **the variable defined within a block will not be accessible from outside the block**. A block can reside inside a function, and a block scoped variable will not be available outside the block even if the block is inside a function.

Scope chain: Whenever our code tries to access a variable during the function call, it starts the searching from local variables. And if the variable is not found, it'll continue searching in its outer scope or parent functions' scope until it reaches the global scope and completes searching for the variable there.

```
const baperTaka=12000;
function hamid(age,earning){
    console.log(trueAge);

    var decision=canMarry(10000);
    console.log(decision);

    console.log(age);
    console.log(earning);

    var trueAge=age+5;

    function canMarry(expanse){
        const remaining=earning-expanse;
        // console.log(showTaka);
        if(remaining>1000)
        {
            const showTaka=12000;
            console.log(showTaka);
            console.log(baperTaka);
            return true;
        }
    }
}
```

```
    }  
    return false;  
  }  
}  
  
hamid(21,50000);
```

37_5-5 Transfer assignment private repo to your github private repo