IBTISSAM OULAHCHOUM

JavaScript for Beginners Series

(مع شرح بالدارجة)

Part12

IBTISSAM OULAHCHOUM

Async/Await & The Fetch API

IBTISSAM OULAHCHOUM

Technical Explanation

While .then() and .catch() are powerful, they can sometimes lead to messy, nested code.
Async/Await is modern syntax built on top of Promises that lets you write asynchronous code that looks and feels synchronous, making it much cleaner and easier to read.

- async: This keyword is placed before a function to signify that it will always return a Promise.
- await: This keyword can only be used inside an async function. It pauses the function's execution until the Promise is settled (resolved or rejected).

واخا .then() و .catch() خدامين مزيان، الكود ديالهم يقدر يترون.
Async/Await هي طريقة كتابة جديدة وعصرية مبنية على الـ Promises،
كتخلي الكود اللامتزامن (asynchronous) يبان بحال إلا كان متزامن (synchronous)، وهادشي كيخليه نقي وساهل فالقراءة.

- async: كتحطها قبل من شي دالة (function) باش تعلم بلي ديك الدالة ديما غترجع Promise.
- await: كتقدر تستعملها غير وسط شي دالة async. كتحبس التنفيذ ديال الدالة و "كتسنا" الـ Promise حتى تكمل.

IBTISSAM OULAHCHOUM

The Fetch API

- The fetch() function is a modern browser API for making network requests (like getting data from another server or an API). It is the perfect real-world use case for async/await.

الدالة fetch() هي الطريقة الحديثة فالمتصفح باش تجيب بيانات من شي سيرفور آخر (API). هي أحسن مثال واقعي فين كنحتاجو async/await.

- fetch() itself returns a Promise that resolves to the Response object. You then need to use another method (like .json(), which also returns a Promise) to get the actual data.

fetch كترجع Promise لي كيعطيك واحد الـ Response object. من بعد خاصك تستعمل Promise آخر (بحال .json()) باش تجبد البيانات الحقيقية لي بغيتي.

IBTISSAM OULAHCHOUM

```
// This is an async function
async function getGitHubUser(username) {
  const apiUrl = `https://api.github.com/users/${username}`;

  // We use try...catch for error handling with async/await
  try {
    console.log("Fetching user data...");

    // 1. 'await' pauses the function until fetch completes
    const response = await fetch(apiUrl);

    // 2. 'await' pauses again until the .json() method completes
    const userData = await response.json();

    console.log(`Name: ${userData.name}`);
    console.log(`Followers: ${userData.followers}`);

  } catch (error) {
    console.error("Oops, something went wrong:", error.message);
  }
}

// Call the async function
getGitHubUser("torvalds"); // Try with a real username
```

IBTISSAM OULAHCHOUM

QUESTION FOR YOU:

- In the example above, what do you think would happen if you removed the await keyword before fetch(apiUrl)? Would the code still work? Why or why not?

IBTISSAM OULAHCHOUM

See you in other parts :)