@CODE.CLASH

AJAX

JAVASCRIPT

1

@CODE.CLASH

What is AJAX?

As a beginner in web development, you may have heard of AJAX(Asynchronous JavaScript And XML) but don't know exactly what it is or how to use it.

- AJAX is a web development technique that allows web pages to be updated asynchronously, without having to reload the entire page.
- It's a combination of technologies that includes JavaScript, XML, and other technologies like HTML, CSS, JSON, etc.
- This enables developers to create dynamic web applications that can update content in real-time without the need for a page refresh.

@CODE.CLASH

What is AJAX?

When an AJAX request is made, the server returns data in a lightweight format such as JSON or XML, which is then processed by JavaScript and used to update the web page dynamically.

An event occurs

- Create an XMLHttpRequest object
- Send XMLHttpRequest

- Process the returned data using Javascript
- Update page content

Browser

- Process XMLHttpRequest
- Create a response and send data back to the browser

Server

3

@CODE.CLASH

Getting Started With AJAX

To get started with AJAX in JavaScript, we'll use the XMLHttpRequest object, which is built into most modern browsers.

Here's an example of how to make an AJAX request using the XMLHttpRequest object:

```
const xhr = new XMLHttpRequest();
xhr.open('GET', 'https://example.com/my-api-endpoint');
xhr.send();
```

In this ex, we create a new XMLHttpRequest object and use the open method to specify the HTTP method and URL for the request.

We then call the send method to send the request to the server.

4

@CODE.CLASH

Handling AJAX Responses

After sending an AJAX request, we need to handle the response from the server. We can do this by setting up a callback function using the onload property of the XMLHttpRequest object.

```
const xhr = new XMLHttpRequest();
xhr.open('GET', 'https://example.com/my-api-endpoint');
xhr.onload = function() {
  console.log(xhr.response);
};
xhr.send();
```

In this example, we set the onload property to a callback function that logs the response from the server to the console.

The response property of the XMLHttpRequest object contains the response data.

5

Error Handling In AJAX

When making AJAX requests, it's important to handle errors that may occur.

This can be done by setting up a callback function for the onerror property of the XMLHttpRequest object.

```
const xhr = new XMLHttpRequest();
xhr.open('GET', 'https://example.com/my-api-endpoint');
xhr.onload = function() {
  console.log(xhr.response);
};
xhr.onerror = function() {
  console.error('An error occurred');
};
xhr.send();
```

In this example, we set the onerror property to a callback function that logs an error message to the console if the request fails.

@CODE.CLASH

Real-World Applications

AJAX is an essential technique for building modern web applications that provide a better user experience.

Social media feeds are a great example of how AJAX can be used to update content on a web page without requiring a page refresh.

Weather apps are another example of how AJAX can be used to update content in real-time.

E-commerce websites often use AJAX to allow users to add items to their cart without leaving the product page.

7

@CODE.CLASH

Conclusion

- As you can see, AJAX is a powerful technique for building modern web applications that provide a better user experience.
- Modern Browsers can use Fetch API instead of the XMLHttpRequest Object.
- If you use the XMLHttpRequest Object, Fetch can do the same in a simpler way.
- As always, I hope you enjoyed the post and learned something new.
- If you have any queries then let me know in the comment box.

8