

# AJAX

JAVASCRIPT



JS

## What is AJAX?

AJAX stands for **Asynchronous JavaScript and XML**.

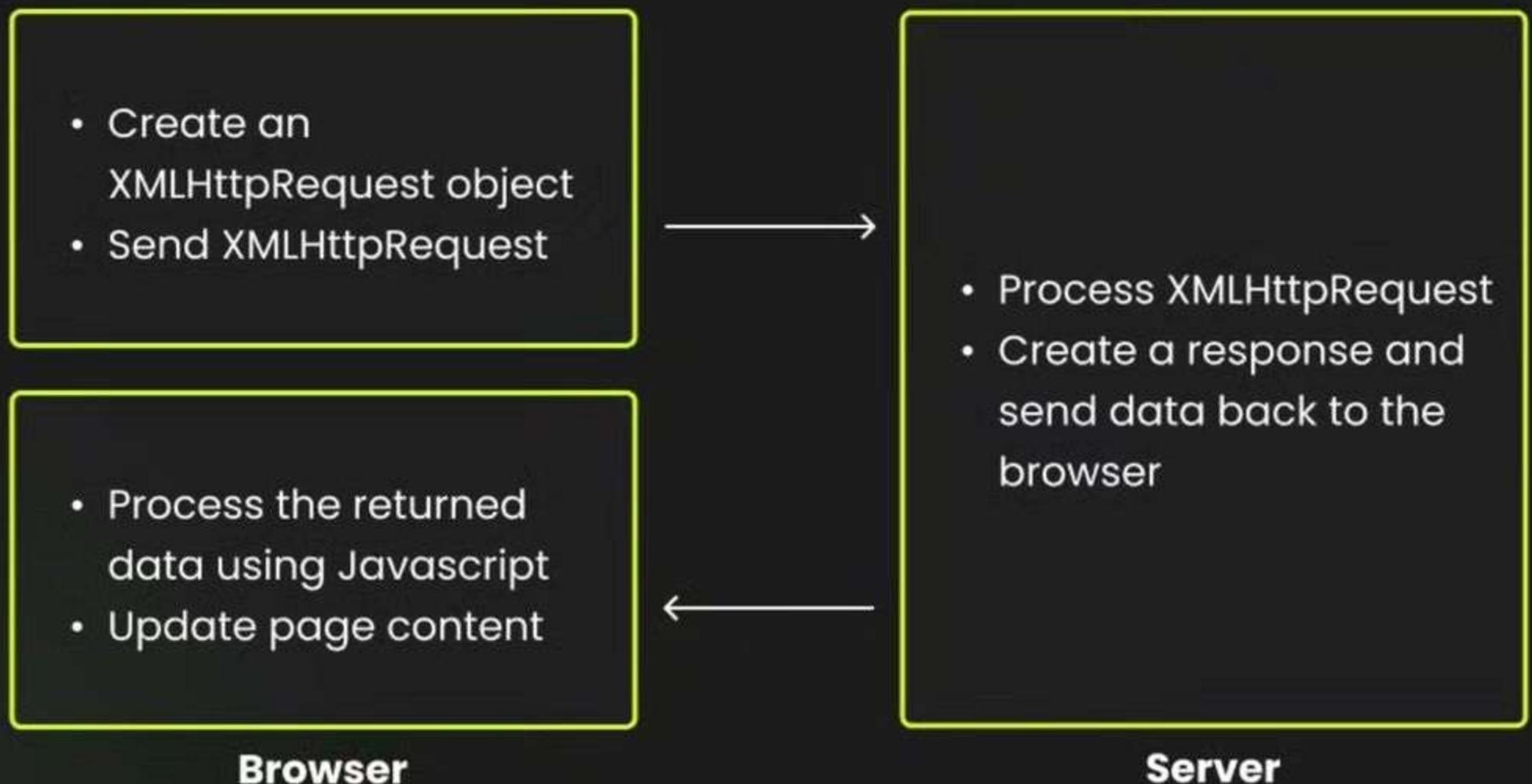
It's not a programming language **but a technique used to create more dynamic** and responsive web applications.

By using AJAX, web **pages can be updated asynchronously** by exchanging small amounts of data with the server behind the scenes.

This means that parts of a **web page can be updated without** needing to reload the entire page.

## How Does AJAX Work?

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.



## Getting started with AJAX

To get started with **AJAX** in JavaScript, we'll use the **XMLHttpRequest** object.

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 **created a new XMLHttpRequest object** and use the **open method** to specify the HTTP method and URL for the request.

Then we **call the send method** to send the request to the server.

## Handling AJAX responses

We can handle response 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.

## 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();
```

## 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.

# AJAX Vs. Fetch

## AJAX:

- Use for compatibility with **older browsers**.
- It has a more **complex event-driven API** and supports **synchronous requests**.

## Fetch:

- Use for modern development due to its **simplicity and integration with promises**.
- Offers cleaner syntax and **better handling of JSON** and streaming data.

While **understanding AJAX** is valuable for historical context, **fetch** is generally **preferred for new projects** due to its modern and simpler approach to making HTTP requests.

## Follow us for a daily dose of tech insights

If you have any suggestions or topics you'd like to see covered in future posts, please leave a comment below!

As always, I hope you enjoyed the post and learned something new.

**Was it  
helpful?  
follow for more!**



**Muhammad Zubair**  
@iamzubairarain



Did you know? I'm a freelancer too! 📁 Visit my Upwork profile:  
[<https://www.upwork.com/freelancers/~014ffd655fc0791835>]