



Ajax tutorial in hindi | JavaScript Tutorial In Hindi #35

[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Ajax tutorial in hindi | JavaScript Tutorial In Hindi #35

In this tutorial, we will learn what Ajax is and how it works. If you want to learn **AJAX** quickly, then you are at right place. In this tutorial, we will not only learn AJAX theoretically but also learn how to implement the AJAX concept in programming.

Ajax stands for **A**synchronous **J**avaScript **A**nd **X**ML Ajax loads the data from the server and updating the parts of a web page selectively without reloading the whole page.

Introduction of AJAX:-

AJAX is a technique for creating faster, and more interactive web applications with the help of XML, HTML, CSS, and JavaScript. It is a web browser technology which is independent of web server software.

AJAX use of the built-in browser **XMLHttpRequest** (XHR) objects to send and receive information to and from a web server asynchronously, in the background, without blocking the page or interfering with the user's experience.

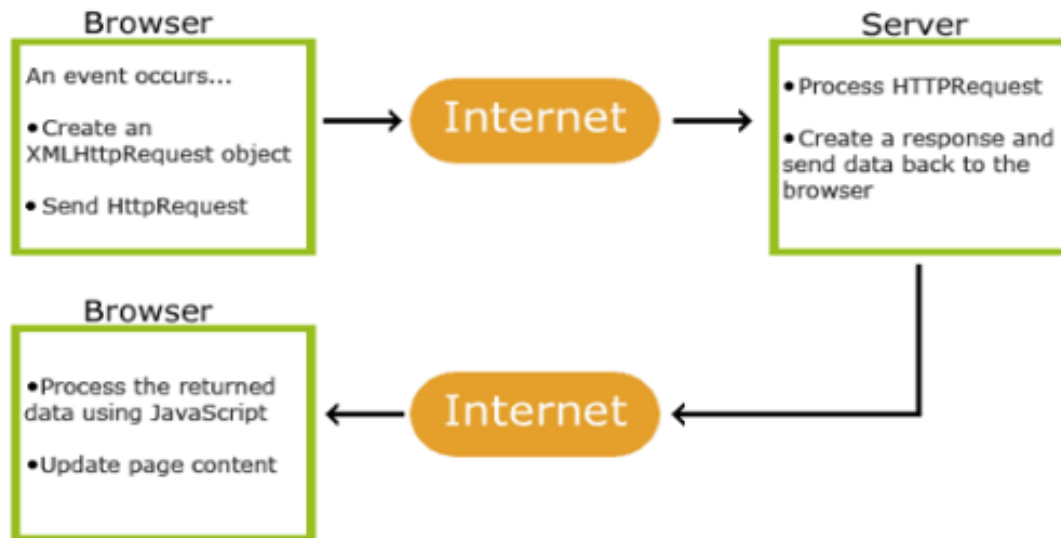
Ajax uses XHTML for the content, CSS for designing, along with Document Object Model and JavaScript for dynamic content display. Before AJAX technology, the web applications transmit information to and from the server using synchronous requests. It this we fill out a form, hit submit, and get directed to a new page with new information from the server. But with AJAX, when we hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen.

We can guess the popularity of AJAX, such that we hardly find an application that doesn't use Ajax to some extent. The example of Ajax-driven online applications are Gmail, Google Maps, YouTube, Facebook, and so many other applications.

How Does AJAX Work?

JavaScript and XML combine to make asynchronous updating happen through the use of something called an XMLHttpRequest (XHR) object. When the user visits a web page which is designed using AJAX technology, and a prescribed event occurs (a button, or fills out a form) the JavaScript creates an XMLHttpRequest (XHR) object, which then transfers data in an XML format between a web browser and a web. The XMLHttpRequest(XHR) object sends a request for updated page data to

the web server, the server process the request, a response is created at server-side and sent back to the browser, which then uses JavaScript to process the response and display it on the screen as updated content.



Summary:-

The JavaScript automates the updating process, the request for updated content is formatted in XML to make it understandable, and JavaScript again refreshes the relevant content for the user viewing the page. Whereas the AJAX technique ignores extraneous page data and only handles requests for updated information and the updated information itself. This shows the AJAX's effectiveness, as it makes the websites and applications that use AJAX faster and more responsive to users.

Code ajax.html as described/written in the video

```
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

```
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<title>Ajax in one video - CodeWithHarry</title>
</head>

<body>
  <h1>Ajax tutorial</h1>
  <button type="button" id="fetchBtn" class="btn btn-primary">Fetch Data</button>
  <button type="button" id="popBtn" class="btn btn-secondary">Populate</button>

  <div class="container">
    <h1>Employee list</h1>
    <ul id="list">

    </ul>
  </div>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
    crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
    integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
    integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60OrQ6VrjIEeAfF/nJGzIxFDs4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
  <script src="js/ajax.js"></script>
</body>

</html>
```

Code ajax.js as described/written in the video

```
console.log("Ajax tutorial in one video");
```

[Copy](#)

```
let fetchBtn = document.getElementById('fetchBtn');
fetchBtn.addEventListener('click', buttonClickHandler)

function buttonClickHandler() {
    console.log('You have clicked the fetchBtn');

    // Instantiate an xhr object
    const xhr = new XMLHttpRequest();

    // Open the object
    // xhr.open('GET', 'https://jsonplaceholder.typicode.com/todos/1', true);

    // USE THIS FOR POST REQUEST
    xhr.open('POST', 'http://dummy.restapiexample.com/api/v1/create', true);
    xhr.setRequestHeader('Content-type', 'application/json');

    // What to do on progress (optional)
    xhr.onprogress = function(){
        console.log('On progress');
    }

    // xhr.onreadystatechange = function () {
    //     console.log('ready state is ', xhr.readyState);

    // }

    // What to do when response is ready
    xhr.onload = function () {
        if(this.status === 200){

            console.log(this.responseText)
        }
        else{
            console.log("Some error occurred")
        }
    }
}
```

```
}

// send the request
params = `{ "name": "test34sad545", "salary": "123", "age": "23" }`;
xhr.send(params);

console.log("We are done!");

}

let popBtn = document.getElementById('popBtn');
popBtn.addEventListener('click', popHandler);

function popHandler() {
    console.log('You have clicked the pop handler');

    // Instantiate an xhr object
    const xhr = new XMLHttpRequest();

    // Open the object
    xhr.open('GET', 'http://dummy.restapiexample.com/api/v1/employees', true);

    // What to do when response is ready
    xhr.onload = function () {
        if(this.status === 200){
            let obj = JSON.parse(this.responseText);
            console.log(obj);
            let list = document.getElementById('list');
            str = "";
            for (key in obj)
            {
                str += `<li>${obj[key].employee_name} </li>`;
            }
            list.innerHTML = str;
        }
        else{
            console.log("Some error occured")
        }
    }
}
```

```
    }  
  }  
  
  // send the request  
  xhr.send();  
  console.log("We are done fetching employees!");  
}
```

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

