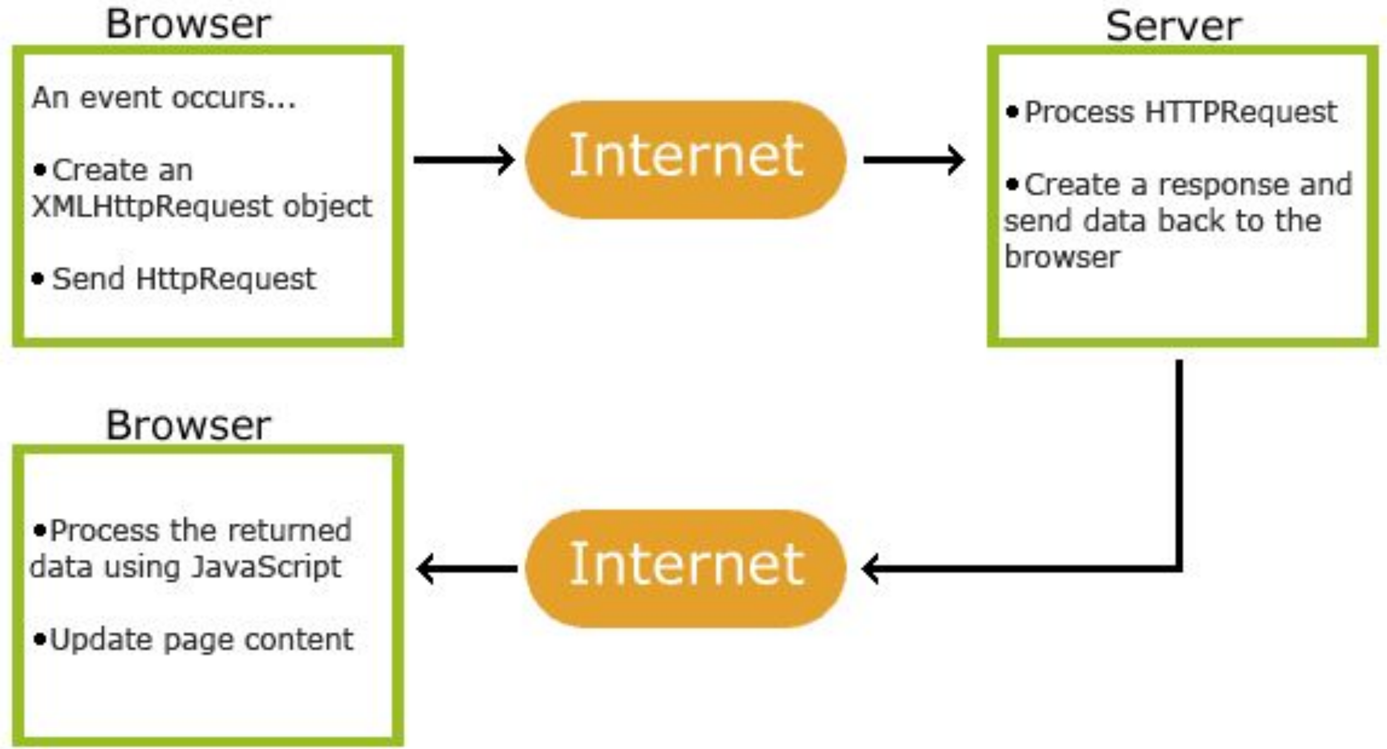
Lecture 11.1

Fetch API

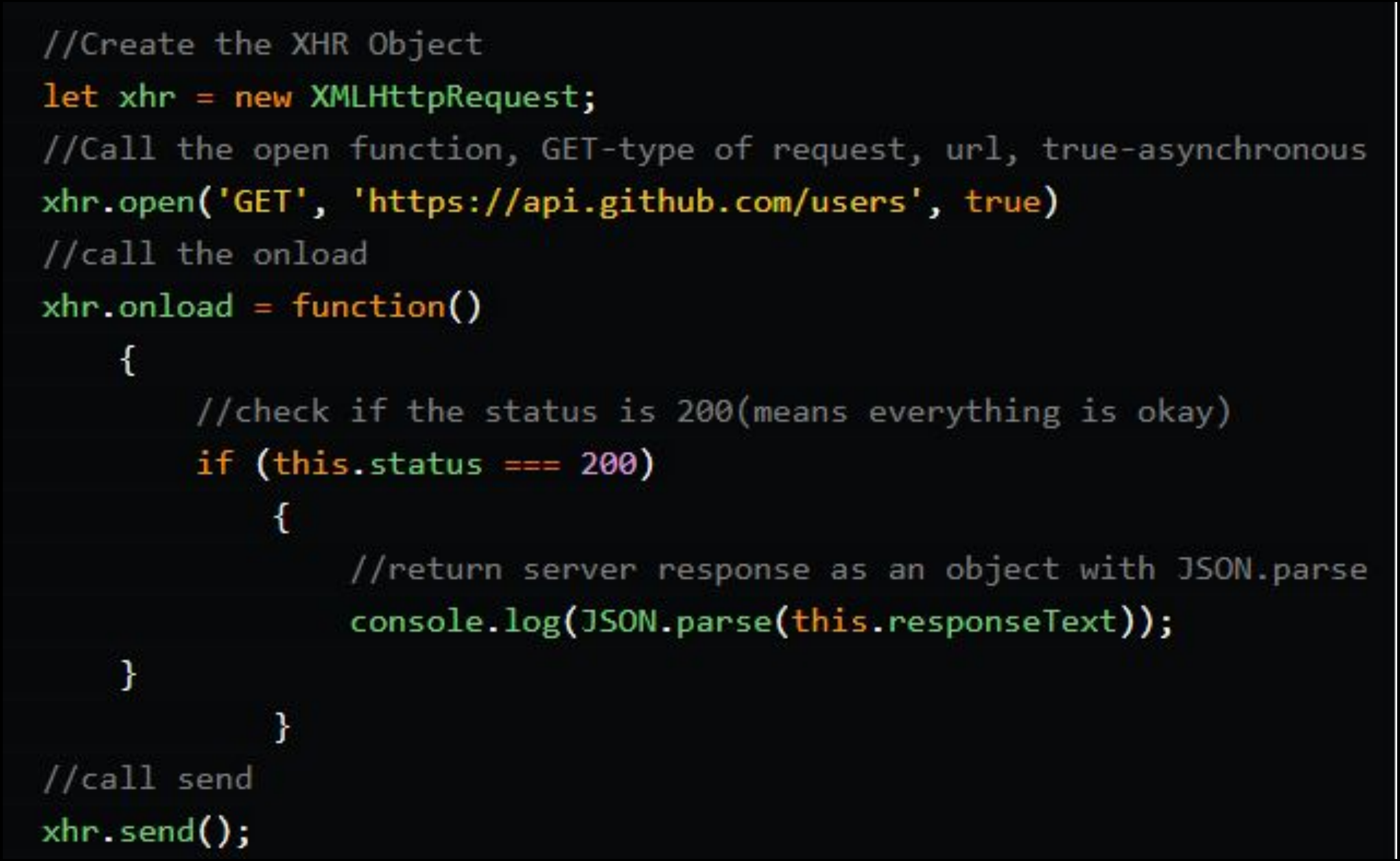
Ajax

AJAX stands for Asynchronous Javascript and XML, it is a set of web technology to send and receive data asynchronously from a client or server,

It is done behind the scene and you don't need to reload the webpage, JSON(Javascript Object Notation) have actually replaced XML(eXtensible Markup Language), most of the API's returns JSON data, AJAX can also return plain text.


Browser
An event occurs...
•Create an XMLHttpRequest object
•Send HttpRequest
Internet
Server
•Process HTTPRequest
•Create a response and send data back to the browser
Internet
Browser
•Process the returned data using JavaScript
•Update page content

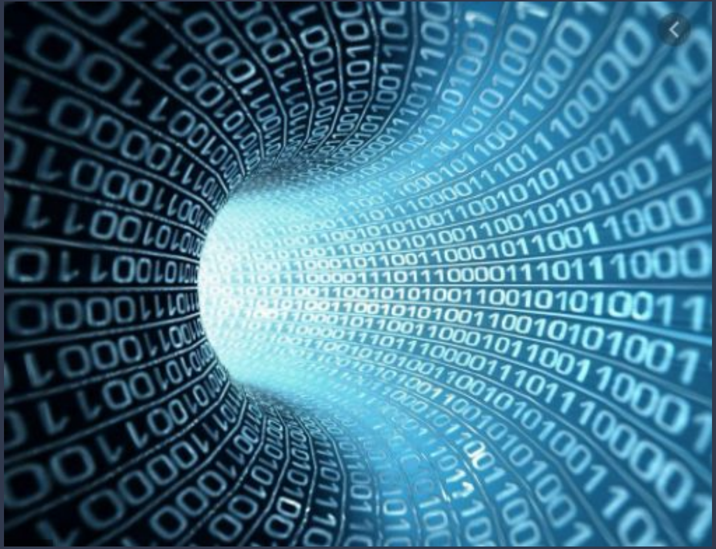
How Ajax works..



```
//Create the XHR Object
let xhr = new XMLHttpRequest;
//Call the open function, GET-type of request, url, true-asynchronous
xhr.open('GET', 'https://api.github.com/users', true)
//call the onload
xhr.onload = function()
    {
        //check if the status is 200(means everything is okay)
        if (this.status === 200)
            {
                //return server response as an object with JSON.parse
                console.log(JSON.parse(this.responseText));
    }
            }
//call send
xhr.send();
```

Fetch API

Fetch API



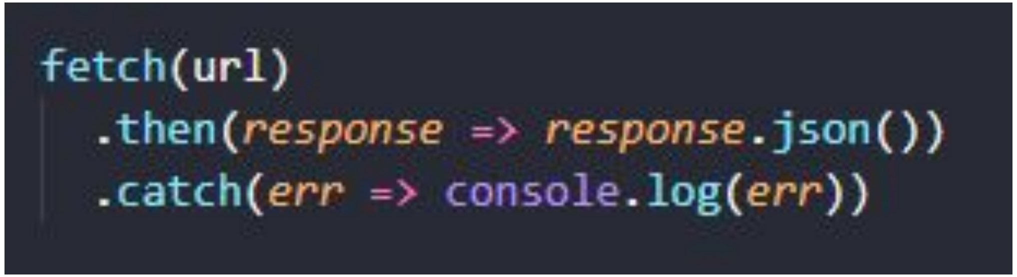
Fetch is the newest standard for dealing with HTTPRequest, it is part of the window object, and we can easily fetch data from an external API as well.Fetch returns Promises

The Fetch API provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as requests and responses. It also provides a global fetch() method that provides an easy, logical way to fetch resources asynchronously across the network.

Fetch

Calling fetch returns a promise, with a Response object. The promise is rejected if there is a network error, and it's resolved if there is no problem connecting to the server and the server responded a status code.

This status code could be 200s, 400s or 500s.



```
fetch(url)
  .then(response => response.json())
  .catch(err => console.log(err))
```

Fetch & ES Array Methods

A common pattern in today’s current web/mobile applications is to request or show some sort of data from the server (such as users, posts, comments, subscriptions, payments and so forth) and then manipulate it by using CRUD (create, read, update or delete) operations.

To further manipulate a resource, we often use these JS methods (recommended) such as .map(), .filter() and .reduce().

Video - Fetch API