AJAX

Notes:

- AJAX: Asynchronous JavaScript and XML.
- It is a development technique for creating interactive web apps.
- It is not a programming language.
- It uses JavaScript: To send and recieve data betwween web browser and server.
- It makes web pages more responsive (Instead of reloading the page each time a user makes a change, it exchange the data with a server behind the scenes.)
- It was made popular in 2005 by Google. (Google Suggest)
- Google Suggest: is using AJAX to create a very dynamic web interface: When you start typing in Google's search box, a JavaScript sends the letters off to a server and the server returns a list of suggestions.
- AJAX uses an XMLHttpRequest (Object) to send data to server.
- Plain text/XML is used as the format for receiving server data.
- NOTE: you don’t have to understand XML to use AJAX.
- Web applications that benefit from AJAX are those who have frequent interactions between client and server.
- AJAX Goals achieved by:

1. Client requests are handled asynchronously.
2. Only small parts of the current document are updated.

- The keystone of AJAX is the XMLHttpRequest object

HTTP response status codes

Status Codes
[200]: Ok (The Request Succeeded)
[404]: Not Found

Ready State: is the status of the request.

Ready Status Codes
[0] Request Not Initialized
[1] Server Connections Established

[2] Request Recieved

[3] Processing Request

[4] Request is Finished and response is Ready

Codes

- All of these are written inside `<script> </script>`

```
var xhr = new XMLHttpRequest();
```

Assigning a variable for the request of object (XHR)

```
xhr.onreadystatechange = function() {
}
```

onreadystatechange --> the function is called when the ready state of request changes

---

```
if (xhr.readyState == 4 && xhr.status == 200) {
   }
```

when the ready state is 4 and responce status is 200 which means request is finished and status is OK

```
document.getElementById("myDiv").innerHTML = xhr.responseText;
```

document → **your web page**

.getElementById(””) → **method returns an element with a specified value (ID)**

*Any **id** should be unique, but:*

*If two or more elements with the same **id** exist, `getElementById()` returns the first.*

.innerHTML → **property sets or returns the HTML content (inner HTML) of an element.**

.responseText → **property returns the text received from a server following a request being sent.**

```
xhr.open("GET", "ajax.txt", true);
```

`open(method, url, async, user, password)`

method:

The HTTP request method
to use, such as `"GET"`
, `"POST"`
, `"PUT"`
, `"DELETE"`
, etc.

async:

If this value is `false`
, the `send()`
method does not return until the response is received.

```
xhr.send();
```

sends the request to the server

`send(string)`, string: only used for POST requests

```
xhr.setRequestHeader();
```

The `XMLHttpRequest` method `setRequestHeader()` sets the value of an HTTP request header. When using `setRequestHeader()`, you must call it after calling `open()`, but before calling `send()`

```
<input type="text" onblur="myFunction()">
```

Call a function when a user leaves an input field:Call a function when a user leaves an input field