What is AJAX?

- AJAX = Asynchronous JavaScript And XML.
- AJAX is not a programming language.

  AJAX just uses a combination of:
- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

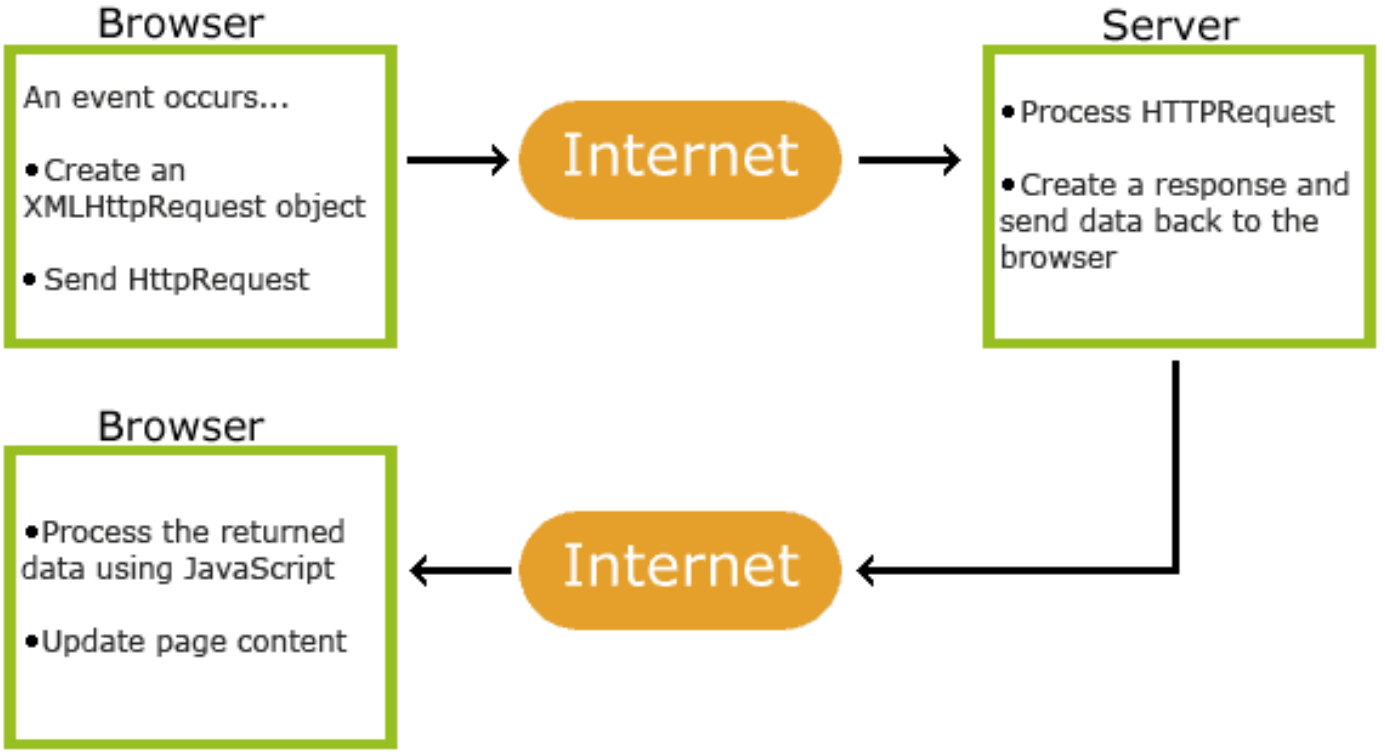
What is AJAX?

- AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

What is AJAX?

- AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

How Ajax Works



How Ajax works

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

The XMLHttpRequest Object

- All modern browsers support the XMLHttpRequest object.
- The XMLHttpRequest object can be used to exchange data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Create an XMLHttpRequest Object

- Syntax for creating an XMLHttpRequest object:
  var xhttp = new XMLHttpRequest();

Access Across Domains

- For security reasons, modern browsers do not allow access across domains.
- This means that both the web page and the XML file it tries to load, must be located on the same server.

Older Browsers (IE5 and IE6)

- Old versions of IE (5/6) use an ActiveX object instead of the XMLHttpRequest object:

```
  if (window.XMLHttpRequest) {
  // code for modern browsers
  xmlhttp = new XMLHttpRequest();
 } else {
  // code for old IE browsers
  xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
```

XMLHttpRequest Object Methods

Method	Description
new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information

XMLHttpRequest Object Methods

open(*method, url, async, user, psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

AJAX - Send a Request To a Server

- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();


xhttp.open("GET", "/api/courses", true);
xhttp.send();
```

GET Requests

```
xhttp.open("GET", "/api/courses", true);
xhttp.send();
```

- To avoid caching

```
xhttp.open("GET", "/api/courses?t=" + Math.random(), true);
xhttp.send();
```

POST Requests

```
xhttp.open("POST", "demo_post.php", true);
xhttp.send();

xhttp.open("POST", "/api/courses", true);
xhttp.send();
```

POST Requests

- To POST data like an HTML form, add an HTTP header with setRequestHeader().
- Specify the data you want to send in the send() method:

```
xhttp.open("POST", "/api/courses", true);
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname=Henry&lname=Ford");
```

Set Header

Method	Description
setRequestHeader(*header, value*)	Adds HTTP headers to the request *header*: specifies the header name *value*: specifies the header value

Asynchronous - True or False?

- Server requests should be sent asynchronously.
- The async parameter of the open() method should be set to true:

  xhttp.open("GET", "/api/courses", true);

  By sending asynchronously, the JavaScript does not have to wait for the server response, but can instead:

- execute other scripts while waiting for server response
- deal with the response after the response is ready

The onreadystatechange Property

- With the XMLHttpRequest object you can define a function to be executed when the request receives an answer.

```
xhttp.onreadystatechange = function() {
 if (this.readyState == 4 && this.status == 200) {
   document.getElementById("demo").innerHTML = this.responseText;
 }
};
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

The onreadystatechange Property

- The **readyState** property holds the status of the XMLHttpRequest.
- The **onreadystatechange** property defines a function to be executed when the readyState changes.
- The **status** property and the **statusText** property holds the status of the **XMLHttpRequest** object.

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")