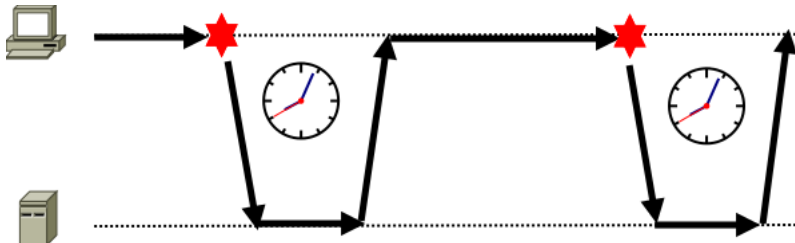


AJAX

Class 26

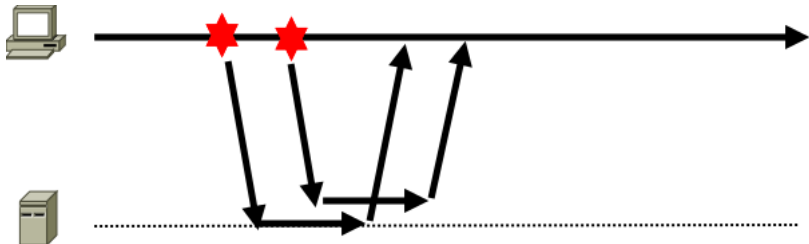
The Current Situation



HTTP requests happen **synchronously**

★ = GET or POST request

AJAX



AJAX allows

- **asynchronous** HTTP requests
- page updates without reloading

History

- technology invented at Microsoft 1996 – 1998 (as ActiveX)
- Jesse James Garrett coined the term AJAX in 2005
- Microsoft did not participate in the standardization
- Microsoft refused to follow the standards until IE 7

Misnomer

Asynchronous JavaScript and XML

- does not require XML
- does not have to be asynchronous
- most requests are for (in order)
 - JSON data
 - plain text data
 - XML data
 - HTML code
- requests
 - GET (most common) or POST
 - asynchronous (synchronous is still legal but deprecated)

AJAX Components

- AJAX depends on
 - the same HTML, CSS, and JS technologies we've been using for document content and display
 - refresh data, formatted as JSON, XML, text, etc.
 - a way for JS to communicate with the server asynchronously without a page reload
- there are two communication technologies, both in JS
 - the XMLHttpRequest object (XHR) — most commonly used
 - the promise mechanism — new in ES6, gaining popularity

Simple Example

poem in ajax.html

XHR Methods and Properties

- **open**(method, url)
- **onload**: the callback function for when the pull succeeds
- **reponstext**: the data that's received (JSON, text, etc.)
- **send**(data or null)

Asynchronous Request

- an asynchronous request does not block the browser from performing other functions while waiting for the request
- a synchronous request is like a function call — nothing else can happen until the function returns
- a slow network or a loaded server can take a long time to respond
- making the user wait is considered poor design
- we will only use asynchronous requests

Promise-Based Request

- `fetch` is a function that returns a `promise`
- a promise is an object that will eventually resolve, either with a result or with an error
- you don't manually register callback functions with a promise
- instead, you chain desired actions together with `.then` methods
- each `.then()` passes its return value to the next `.then()`

A Series of Requests

- some requests are singletons or idempotent
- but sometimes requests can interfere and cause race conditions
- an XHR request can be **anceled** via the `abort()` message
- this example returns plain text as **formatted HTML**

`selectword.html` and `.js` and `get_words.php`

Uses of AJAX

the primary uses of AJAX include:

real-time form data validation form data can be validated before submission

auto-completion input control completions can be suggested as the user types

load on demand a web page can load more content after an initial load is completed and rendered

polling and refreshing data a web page can periodically poll a server to see if more or newer data is available