

CSC309 Winter 2016

Lecture 5

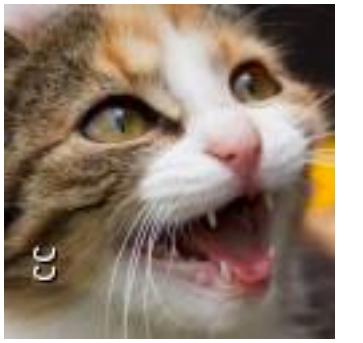
Larry Zhang

Announcements

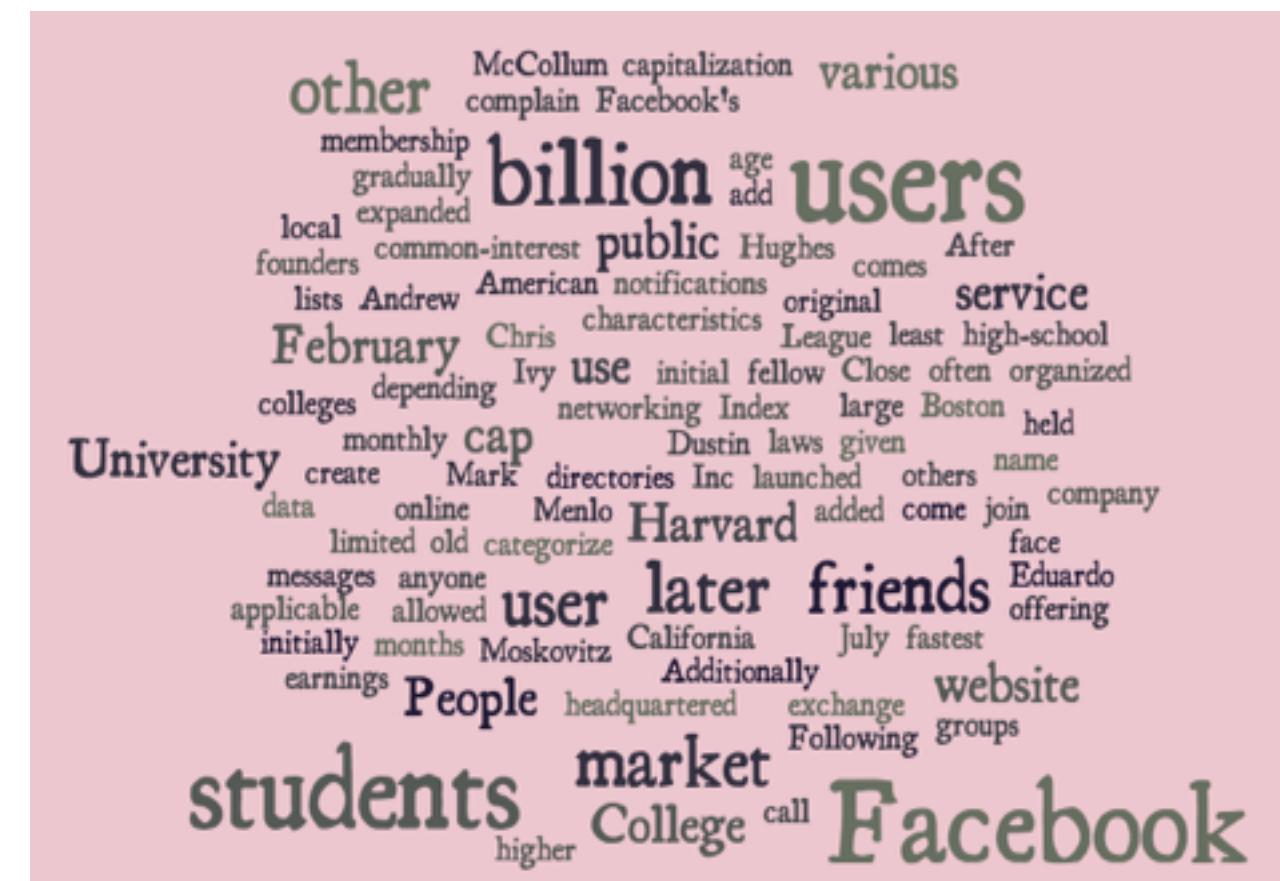
- A2 will be out later today
- ask questions (early) for clarification!



I just created a word cloud with recent tweets about Canada!



I just created a word cloud with recent tweets about Facebook!



This is just a quick mock-up, not required style.

PHP demo continued...

- sessions
- store data in a file

PostgreSQL
server-side database store

PostgreSQL: quick intro

- It is an object relational database management system (ORDBMS)
- ACID-compliance, transactional, concurrency control
- free and open-source
- it is just one of many choice for database: SQLite, MySQL, MongoDB, ...

PostgreSQL demo
Learn by examples...

Reference (PostgreSQL)

<http://www.postgresql.org/docs/8.2/static/index.html>

PHP & PostgreSQL demo
learn by examples...

Reference (PHP & PostgreSQL):

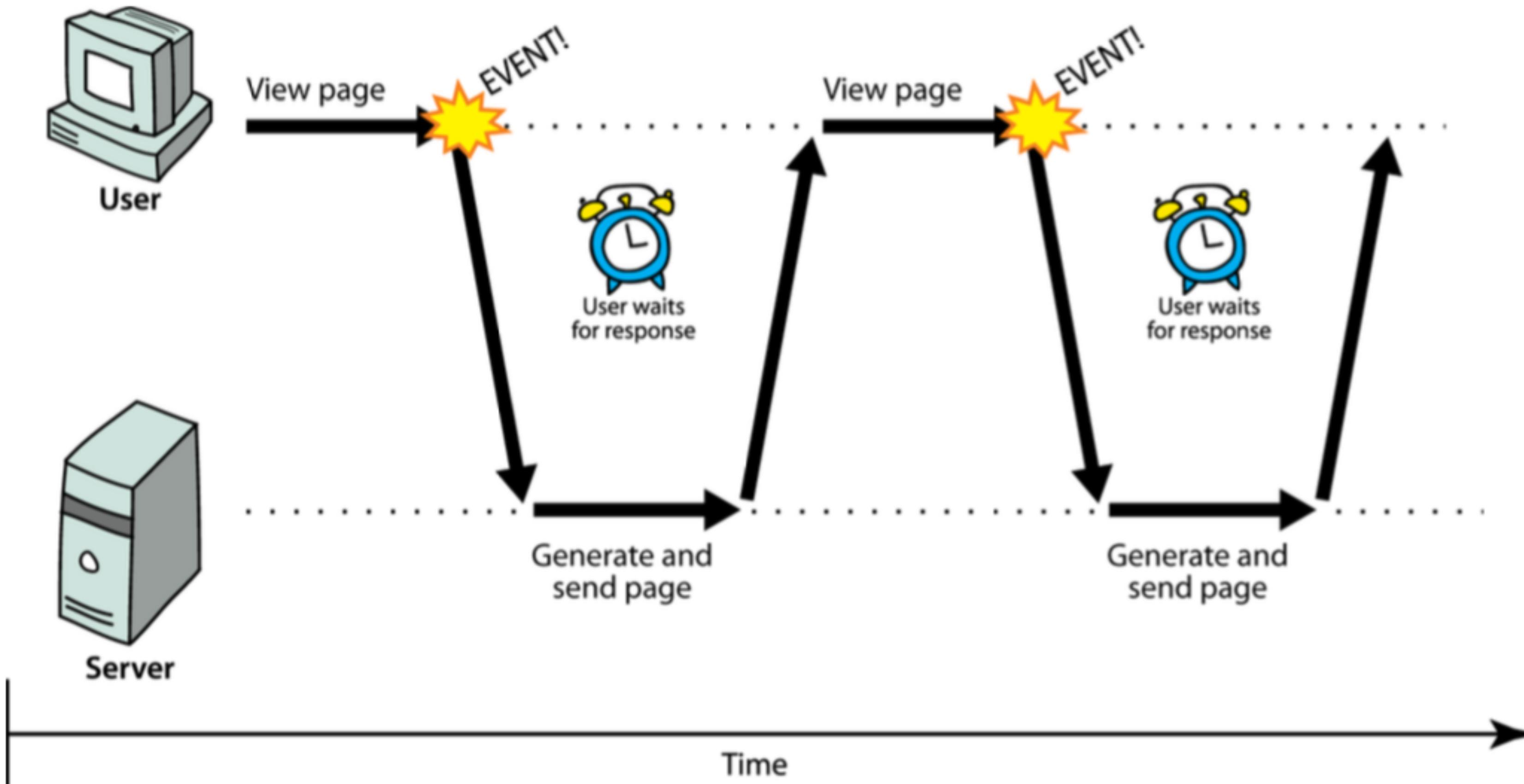
<http://php.net/manual/en/bookpgsql.php>

AJAX and JQuery

AJAX

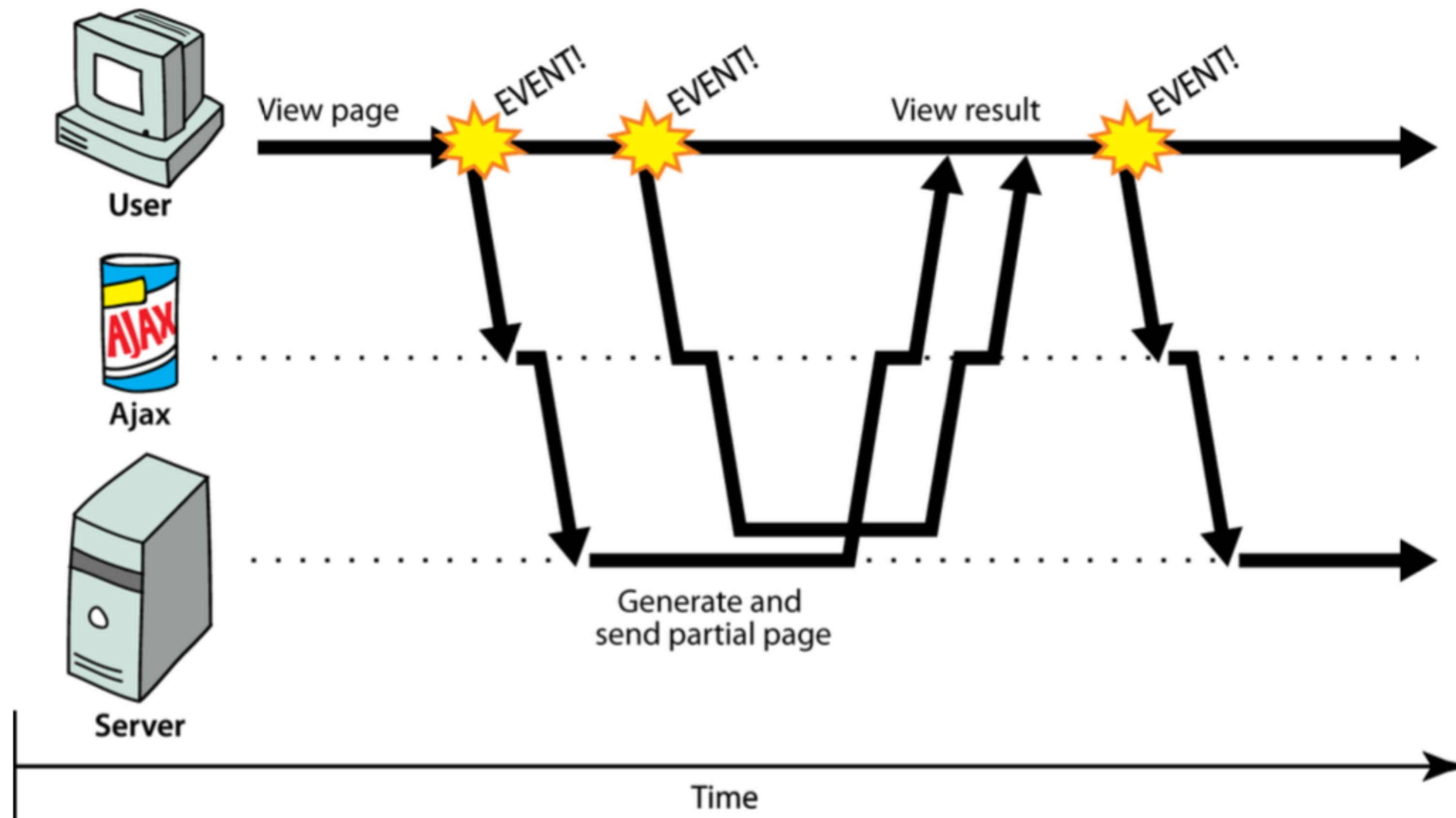
- Stands for **Asynchronous Javascript And XML**
- not a programming language; a particular way of using JavaScript
- downloads data from a server in the background
- allows dynamically updating a page without making the user wait
- avoids the "click-wait-refresh" pattern

Synchronous Web Communication



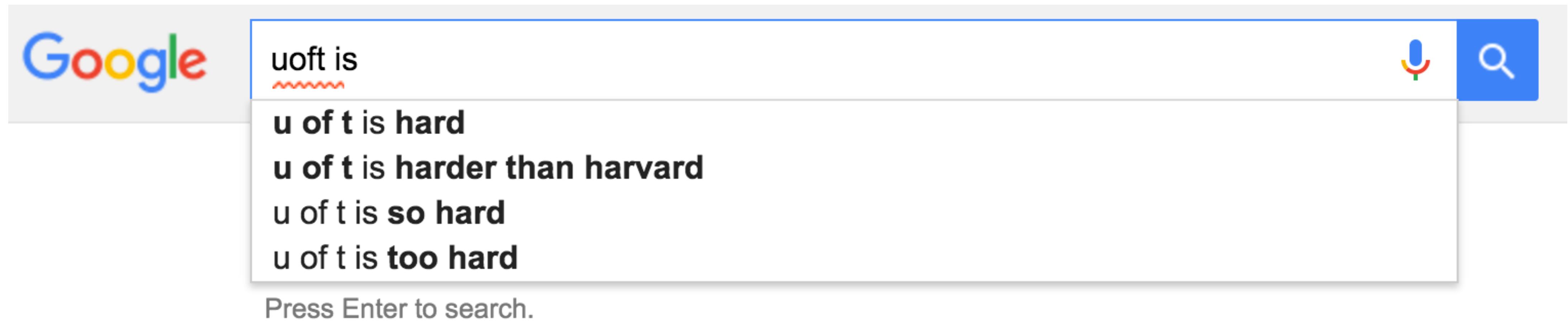
Synchronous: User must wait while the new pages load (click, wait, refresh)

Asynchronous Web Communication



Asynchronous: User can keep interacting with the page while data loads

Example of asynchronous communication

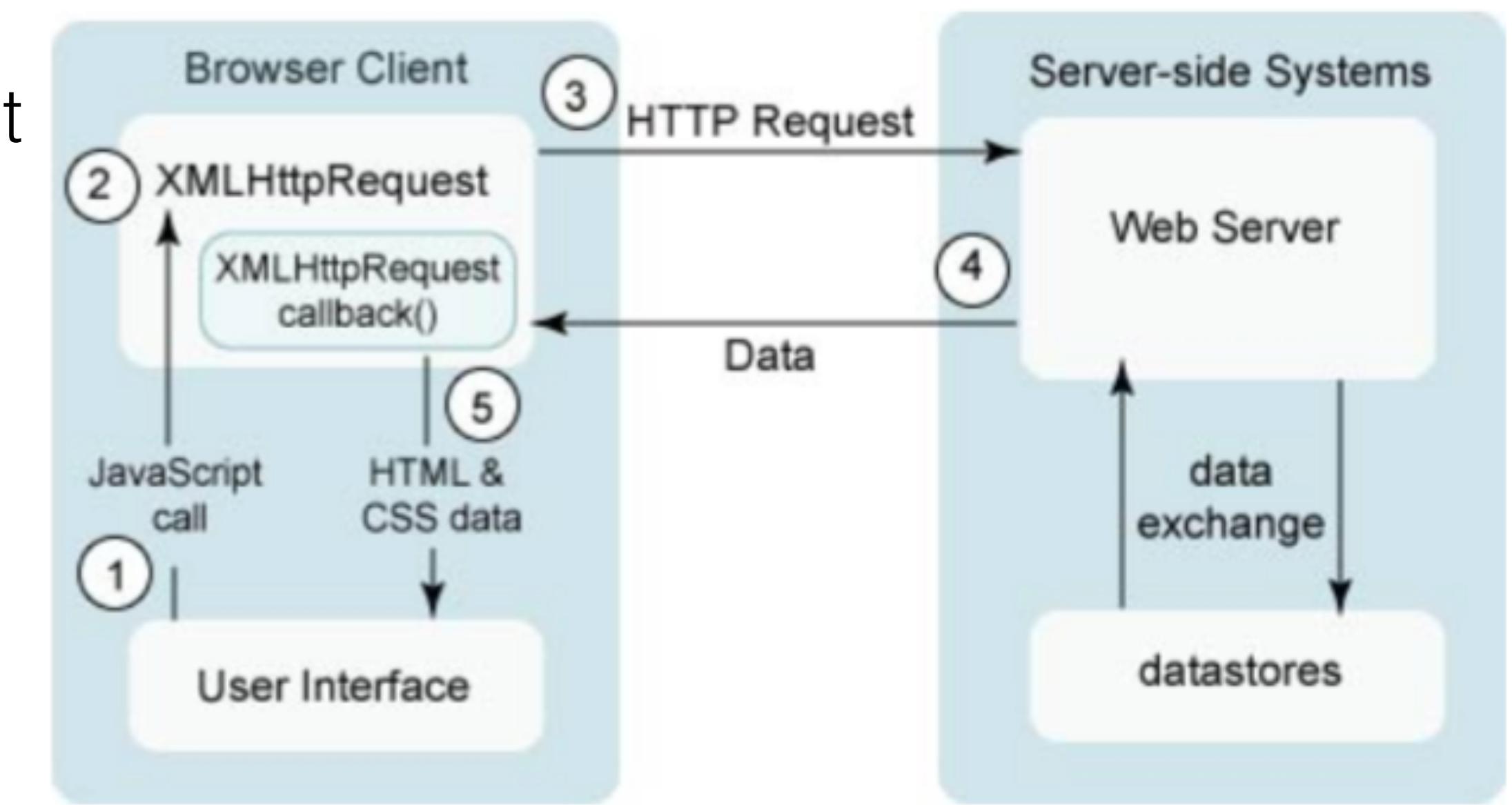


XMLHttpRequest

- The JavaScript object that makes the asynchronous web communication possible
- Despite its name, it is NOT required to use XML (JSON is often used)
- Supported by all major browsers (IE5+, Safari, Firefox, Chrome, ...)
- Raw low-level XMLHttpRequest is clunky to use, we typically use tools such JQuery to make our lives easier.

A typical AJAX request

1. User clicks invoking an event handler
2. Handler's code creates an XMLHttpRequest object
3. XMLHttpRequest object sends request to server
4. Server retrieves appropriate data, sends it back
5. XMLHttpRequest fires an event when data arrives — this is called a callback. The callback function is a handler function attached to the event.
6. The callback event hander processes the data and displays it (using the DOM).



demo:
raw low-level XMLHttpRequest

JQuery

- a JavaScript library that makes JavaScript easier to use (or when it comes to AJAX, a lot easier)
- One of many JavaScript libraries available (Dojo, AngularJS, ReactJS, Prototype, YUI, MooTools, Midori, Bootstrap, ...)

Basic Syntax: the “\$” selector

- Instead of saying **document.getElementById(“myDivID”)**, you can just do **\$ (“#myDivId”)** in JQuery
- Similarly, you can do **\$ (“.myDivClass”)** to select class
- **var myHTML = \$("#myDivId").html();**: **get** the innerHTML code of the div.
- **var myHTML = \$("#myDivId").html(someHTML);**: to **set** the innerHTML of the div to someHTML.

When document is ready, attached a function to all elements' click handler.

```
1 $( document ).ready(function() {  
2     $( "a" ).click(function( event ) {  
3         alert( "Thanks for visiting!" );  
4     });  
5 } );
```

Add/Remove class “test” for all <a> elements

```
1 | $( "a" ).addClass( "test" );
```

```
1 | $( "a" ).removeClass( "test" );
```

Callback without/with arguments

```
1 | $.get( "myhtmlpage.html", myCallBack );
```

```
1 | $.get( "myhtmlpage.html", function() {
2 |
3 |     myCallBack( param1, param2 );
4 |
5 |});
```

JQuery's AJAX methods

```
1 // Using the core $.ajax() method
2 $.ajax({
3
4     // The URL for the request
5     url: "post.php",
6
7     // The data to send (will be converted to a query string)
8     data: {
9         id: 123
10    },
11
12    // Whether this is a POST or GET request
13    type: "GET",
14
15    // The type of data we expect back
16    dataType : "json",
17
18    // Code to run if the request succeeds;
19    // the response is passed to the function
20    success: function( json ) {
21        $( "<h1>" ).text( json.title ).appendTo( "body" );
22        $( "<div class='content'>" ).html( json.html ).appendTo( "body" );
23    },
24
25    // Code to run if the request fails; the raw request and
26    // status codes are passed to the function
27    error: function( xhr, status, errorThrown ) {
28        alert( "Sorry, there was a problem!" );
29        console.log( "Error: " + errorThrown );
30        console.log( "Status: " + status );
31        console.dir( xhr );
32    },
33
34    // Code to run regardless of success or failure
35    complete: function( xhr, status ) {
36        alert( "The request is complete!" );
37    }
38});
```

Convenience AJAX methods

```
1 // Using jQuery's Ajax convenience methods
2
3 // Get plain text or HTML
4 $.get( "/users.php", {
5     userId: 1234
6 }, function( resp ) {
7     console.log( resp ); // server response
8 });
9
10 // Add a script to the page, then run a function defined in it
11 $.getScript( "/static/js/myScript.js", function() {
12     functionFromMyScript();
13 });
14
15 // Get JSON-formatted data from the server
16 $.getJSON( "/details.php", function( resp ) {
17
18     // Log each key in the response data
19     $.each( resp, function( key, value ) {
20         console.log( key + " : " + value );
21     });
22 });
```

JQuery/AJAX demo

Reference for JQuery

<https://learn.jquery.com/>

<http://api.jquery.com/>