

User Interaction: jQuery

Assoc. Professor Donald J. Patterson
INF 133 Fall 2012



- jQuery
 - A JavaScript Library
 - Cross-browser
 - Free (beer & speech)
 - It supports
 - manipulating HTML elements (DOM)
 - animations
 - event handling
 - AJAX

- Getting access to jQuery in our code
 - You can't just write JavaScript that calls jQuery
 - You must load it
 - From where?
 - From your own machine
 - <http://jquery.com/>
 - From someone else's machine
 - From Google
 - Why would you do this?
 - <http://code.google.com/apis/libraries/devguide.html>
 - Shortcut



- Loading jQuery is just like loading a remote JavaScript

```
<html>
  <head>
    <script type="text/javascript" src="Step5.js"></script>
  </head>

  <body onload="message()">
    Hello World!
  </body>
</html>
```

- Loading jQuery is just like loading a remote JavaScript

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
  </head>

  <body>
    
  </body>
</html>
```

- With jQuery you select HTML elements in the DOM and perform actions on them
 - Basic syntax is: `$(selector).action()`
 - A dollar sign (\$) is used to define jQuery
 - `$(selector).action() = jQuery(selector).action()`
 - A **selector** to "query (or find)" HTML elements
 - A jQuery **action** to apply to the element(s)

```
<p class="foo">words</p>
```

```
<p id="bar">more words</p>
```

```
<p class="foo">even more words</p>
```

```
<img class="foo" />
```

- Selection examples:
 - multiple results
 - \$("p").hide()
 - \$(".foo").hide()
 - \$("p.foo").hide()
 - single result
 - \$("#bar").hide()
 - context dependent
 - \$(this).hide()

Possible Actions:

- **append** content to a DOM element
- **wrap** content around a DOM element
- **toggle** the display of a DOM element
- attach a function that is run when you **hover** over a DOM element
- you can request data via the **ajax** pattern

- Loading jQuery is just like loading a remote JavaScript

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
    <script>
      $(document).ready(
        function() {
          $("img").click(
            function () {
              $(this).hide("blind", { direction: "horizontal" }, 5000);
            }
          );
        }
      );
    </script>
  </head>

  <body>
    
  </body>
</html>
```

Step 7

- Loading jQuery is just like loading a remote JavaScript

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
    <script>
      $(document).ready(
        function() {
          $("img").click(
            function () {
              $(this).hide("explode", 5000);
            }
          );
        }
      );
    </script>
  </head>

  <body>
    
  </body>
</html>
```

- Step 8

- Loading jQuery is just like loading a remote JavaScript

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></scr
    <script>
      $(document).ready(
        function() {
          $("img").click(
            function () {
              $(this).hide("blind", { direction: "horizontal" }, 5000);
            }
          );
        }
      );
    </script>
  </head>

  <body>
    
    <br/>
    
  </body>
</html>
```

Step 07_02: Is this an example of an AJAX pattern?

The screenshot shows the jQuery website homepage. At the top, there is a navigation bar with links for jQuery, Plugins, UI, Meetups, Forum, Blog, About, and Donate. Below this is a secondary navigation bar with links for Download, Documentation, Tutorials, Bug Tracker, and Discussion. The main content area features the jQuery logo and tagline "write less, do more." followed by a headline "jQuery is a new kind of JavaScript Library." and a paragraph describing the library's capabilities. A "GRAB THE LATEST VERSION!" section offers two compression levels: PRODUCTION (31KB, Minified and Gzipped) and DEVELOPMENT (229KB, Uncompressed Code). A large "Download (jQuery);" button is present, with the current release version v1.6.4 noted below it. A "WHO'S USING JQUERY?" section displays logos for various companies and organizations. The footer contains two columns of links: "LEARN JQUERY NOW!" with a code snippet and a "RUN CODE" button, and "JQUERY RESOURCES" with links for getting started and developer resources.

jQuery Plugins UI Meetups Forum Blog About Donate

jQuery
write less, do more.

Download Documentation Tutorials Bug Tracker Discussion

jQuery is a new kind of JavaScript Library.

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.

✓ [Lightweight Footprint](#) ✓ [CSS3 Compliant](#) ✓ [Cross-browser](#)

GRAB THE LATEST VERSION!

CHOOSE YOUR COMPRESSION LEVEL:

- PRODUCTION (31KB, Minified and Gzipped)
- DEVELOPMENT (229KB, Uncompressed Code)

Download (jQuery);

Current Release: **v1.6.4**

WHO'S USING JQUERY? Google DELL IBM NBC CBS NETFLIX Technorati mozilla.org

LEARN JQUERY NOW!

What does jQuery code look like? Here's the quick and dirty:

```
$("#p.neat").addClass("ohmy").show("slow");
```

RUN CODE

JQUERY RESOURCES

Getting Started With jQuery

- ◆ [How jQuery Works](#)
- ◆ [Tutorials](#)
- ◆ [Using jQuery with other libraries](#)
- ◆ [jQuery Documentation](#)

Developer Resources

- ◆ [Mailing List](#)
- ◆ [Source code / Git](#)
- ◆ [Plugin Authoring](#)
- ◆ [Submit a New Bug Report](#)

- Assignment 4
 - Present the data as a table
 - http://www.w3schools.com/html/html_tables.asp
 - HTML tables overview
 - 3 primary tags
 - `<table>`
 - `<tr>`
 - `<td>`

- Basic table
 - <table>
 - <tr>
 - <td>

1	2
3	4

```
<html>
  <body>
    <table border="1">
      <tr>
        <td>
          1
        </td>
        <td>
          2
        </td>
      </tr>
      <tr>
        <td>
          3
        </td>
        <td>
          4
        </td>
      </tr>
    </table>
  </body>
</html>
```

- Assignment #4
 - Geocoded Feeds
 - http://api.flickr.com/services/feeds/geo/QDd_2PObCZ4ZsRM6Sw&format=json
 - Use dynamic data, not static data
 - jQuery
 - HTML Table
 - AJAX request

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
    <script src="Step10.js"></script>
  </head>

  <body>
    <div class="dataHere">Replace this text!</div>
  </body>
</html>
```



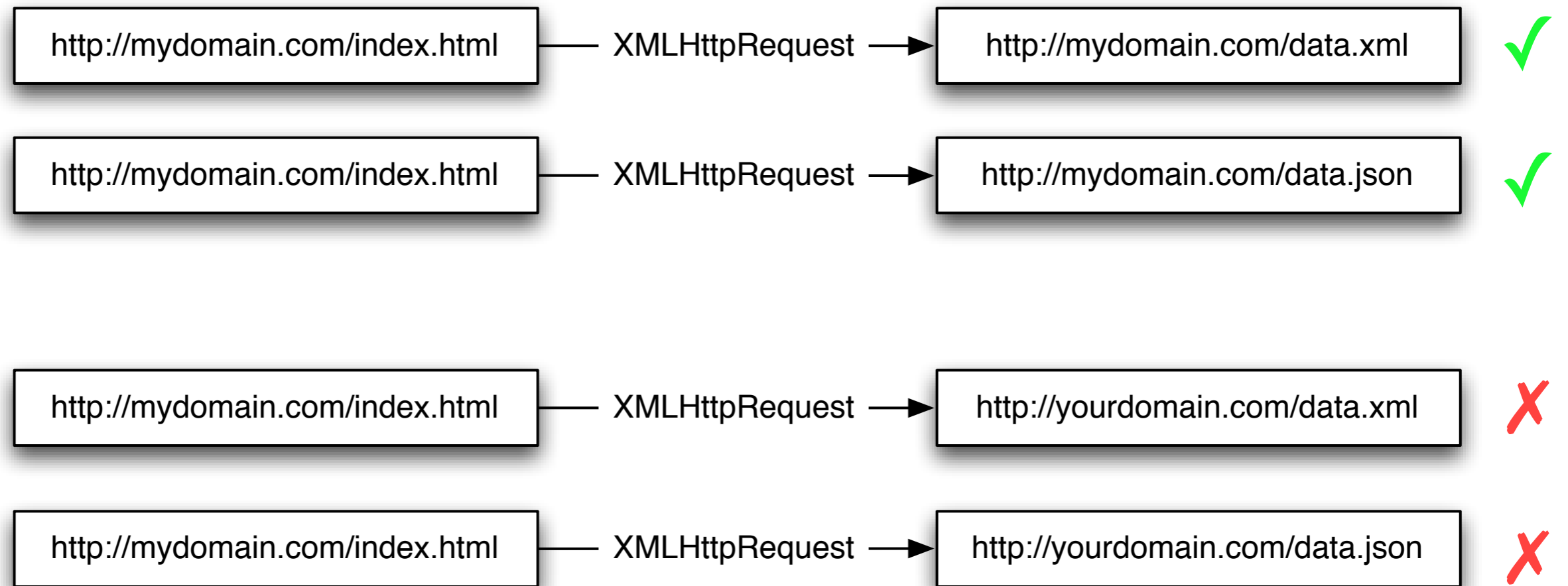
```
{  
    "HELLO": "WORLD",  
    "This": "is",  
    "my": "data"  
}
```

```
function myGoodLoadFunction(data) {  
    $("div.dataHere").text(data.HELLO);  
}  
  
function myBadLoadFunction(XMLHttpRequest, errorMessage, errorThrown) {  
    alert("Load failed: "+errorMessage+": "+errorThrown);  
}  
  
function myReadyFunction(){  
    $.ajax({  
        url: "http://www.ics.uci.edu/~djp3/classes/2012_09_INF133/Lectures/Lecture07/test.json",  
        dataType: "json",  
        success: myGoodLoadFunction,  
        error: myBadLoadFunction,  
    });  
}  
  
$(document).ready(  
    function(){  
        myReadyFunction();  
    }  
);
```

Assignment 4

- Two problems:
 - Javascript does not allow you to make an XMLHttpRequest from an external domain
 - XMLHttpRequest is the javascript function which retrieves data asynchronously
 - Not just XML, but any data
 - For security reasons it was restricted
 - An http URL cannot request from an https URL and vice versa

Assignment 4



Assignment 4

- Solutions to the cross-domain security restriction
 - Run a “proxy”
 - a php proxy
 - You request
 - <http://mydomain.com/proxy.php?http://yourdomain.com/data.xml>
 - Javascript thinks its coming from local server
 - proxy.php contains something like this code:
- php doesn't run on ics.uci.edu
 - You need to use “students.ics.uci.edu”
 - <http://www.ics.uci.edu/computing/web/faqs.php#students>

Assignment 4

- This is the contents of a file on your server
- It is running a language called php
- the language is interpreted by your web server code

```
<?php
//Get the requested url from our proxy url
$url = $_SERVER['QUERY_STRING'];

// initialize curl with given url
$ch = curl_init($url);

//Tell curl to write the response to a variable
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

//Tell curl to follow any redirects
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);

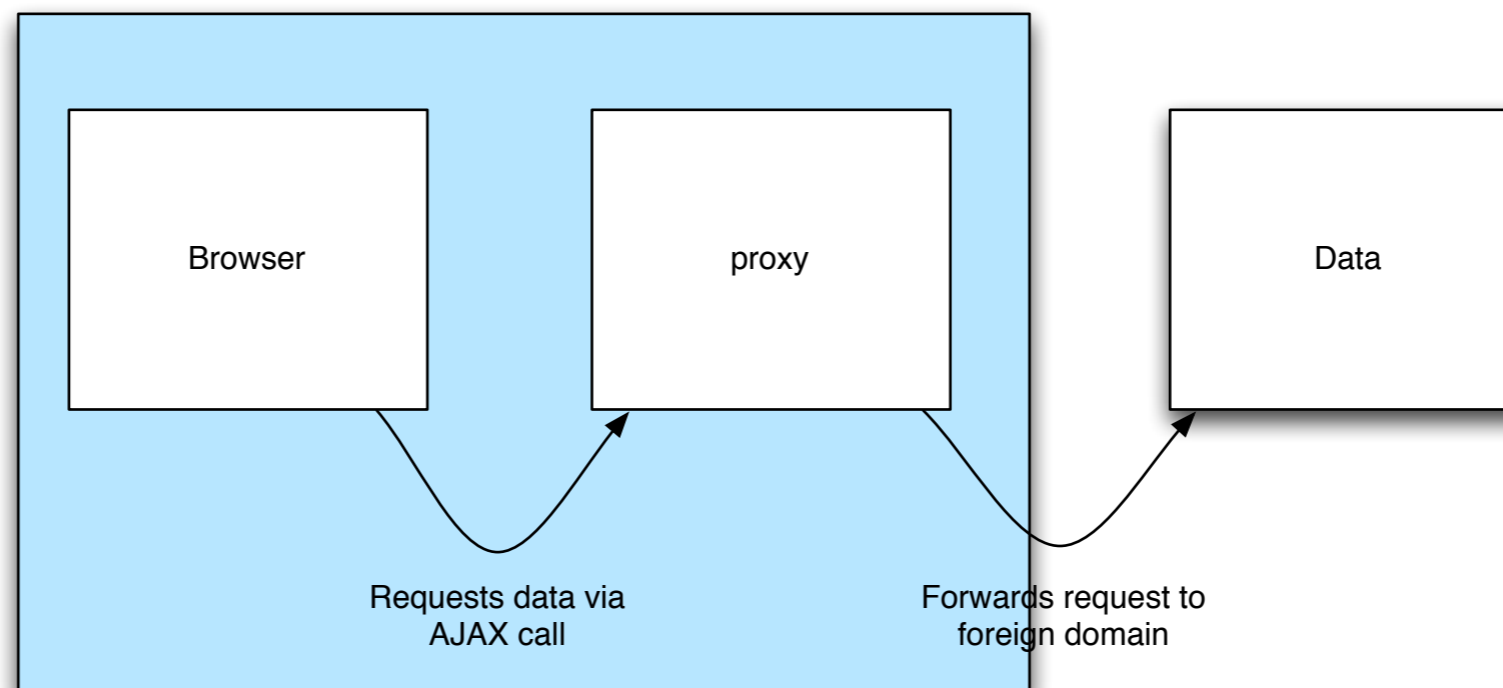
$response = curl_exec($ch);

$cType = curl_getinfo($ch,CURLINFO_CONTENT_TYPE);

//Output the right mime/type
header('Content-type: '.$cType);

//Pass through the data
print $response;

curl_close($ch);
?>
```



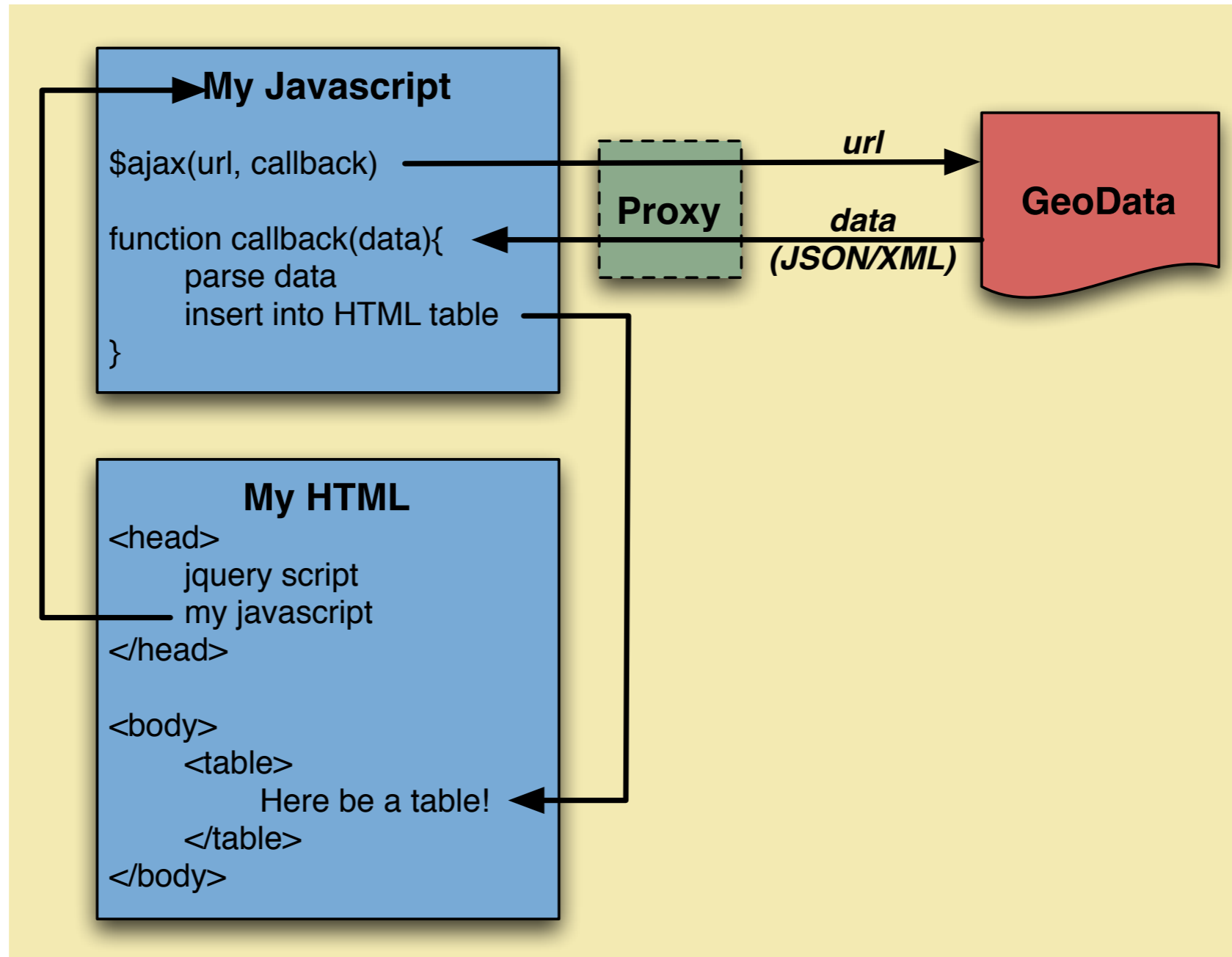
Assignment 4

- Solutions to the cross-domain security restriction
 - Use JSONP
 - Requires server support
 - Yahoo APIs support this
 - Leverages Javascript loophole
 - XMLHttpRequests are restricted
 - Remote Javascript is not

Assignment 4

- JSONP adds a `<script src=>` element to your web page with the external URL as the src target
- Getting JSON looks like this:
 - Request: `http://yourdomain.com/data.json`
 - Return: `{"hello":"world"}`
- Getting JSONP looks like this:
 - Request `http://yourdomain.com/data.jsonp?callback=myCallback`
 - Return: `myCallback("{\"hello\":\"world\"}");`
- Your webpage writes a function called `myCallback` to deal with the data

Assignment 4



Assignment 4

- Security issue
 - You are running server generated code on your machine
 - !

```
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></scr
    <script src="Step11.js"></script>
  </head>

  <body>
    <div class="dataHere">Replace this text!</div>
    <div class="textHere">Don't replace this text!</div>
    <div class="textAlsoHere">Don't replace this text either</div>
  </body>
</html>
```

```
function escapeText(t){
    return document.createTextNode(t).textContent;
}

function jsonFlickrFeed(data) {
    myGoodLoadFunction(data);
}

function myGoodLoadFunction(data) {
    var newHTML= "<div style=\"padding:20px;border:solid 1px black\">"+escapeText(data.title)+"<br/>";
    newHTML += escapeText(data.link)+"<br/>";
    newHTML += escapeText(data.items[0].link)+"</div>";
    $("div.dataHere").html(newHTML);
}

function myBadLoadFunction(XMLHttpRequest,errorMessage,errorThrown) {
    alert("Load failed:"+errorMessage+": "+errorThrown);
}

function myReadyFunction(){
    $.ajax({
        url: "http://api.flickr.com/services/feeds/geo/QDd_2P0bCZ4ZsRM6Sw&format=json",
        dataType: "jsonp",
        jsonp: false,
        jsonpCallback: "jsonFlickrFeed",
        error: myBadLoadFunction,
    });
}

$(document).ready(
    function(){
        myReadyFunction();
    }
);
```

```
function escapeText(t){
    return document.createTextNode(t).textContent;
}

function jsonFlickrFeed(data) {
    myGoodLoadFunction(data);
}

function myGoodLoadFunction(data) {
    var newHTML= "<div style=\"padding:20px;border:solid 1px black\">"+escapeText(data.title)+"<br/>";
    newHTML += escapeText(data.link)+"<br/>";
    newHTML += escapeText(data.items[0].link)+"</div>";
    $("div.dataHere").html(newHTML);
}

function myBadLoadFunction(XMLHttpRequest,errorMessage,errorThrown) {
    alert("Load failed:"+errorMessage+": "+errorThrown);
}

function myReadyFunction(){
    $.ajax({
        url: "http://api.flickr.com/services/feeds/geo/QDd_2P0bCZ4ZsRM6Sw&format=json",
        dataType: "jsonp",
        jsonp: false,
        jsonpCallback: "jsonFlickrFeed",
        error: myBadLoadFunction,
    });
}

$(document).ready(
    function(){
        myReadyFunction();
    }
);
```

- What if you have something other than JSON to load?
 - XML
- Where can I find info about jQuery actions?
 - jQuery API Reference
 - for example, `html()` vs `text()` vs `append()`?

jQuery - Final example Step14

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
    <script src="Step14.js"></script>
  </head>

  <body>
    <div class="dataXML">
      <br/>
      Replace this with XML
    </div>
    <hr/>
    <div class="dataJSON">
      <br/>
      Replace this with JSON
    </div>
    <hr/>
    <div class="dataJSONP">
      <br/>
      Replace this with JSONP
    </div>
  </body>
</html>
```

```
function myBadLoadFunction(myXMLHttpRequest,myErrorMessage,myErrorThrown) {
    alert('status: ' + myErrorMessage + '\n' + myXMLHttpRequest.responseText);
}

function myReadyFunction(){
    $.ajax({
        url: "https://students.ics.uci.edu/~djp3/myProxy.php?http://api.flickr.com/service",
        dataType: "xml",
        success: myGoodLoadXML,
        error: myBadLoadFunction
    });
    $("div.dataXML").html("AJAX XML call initiated:<br/>");

    $.ajax({
        url: "https://students.ics.uci.edu/~djp3/myProxy.php?https://graph.facebook.com/c",
        dataType: "json",
        success: myGoodLoadJSON,
        error: myBadLoadFunction
    });
    $("div.dataJSON").html("AJAX JSON call initiated:<br/>");

    $.ajax({
        url: "https://gdata.youtube.com/feeds/base/users/djp3/uploads?alt=json-in-script&",
        dataType: "jsonp",
        success: myGoodLoadJSONP,
        error: myBadLoadFunction
    });
    $("div.dataJSONP").html("AJAX JSONP call initiated:<br/>");
}

$(document).ready(
    myReadyFunction
);
```



```
function escapeText(t){
    return document.createTextNode(t).textContent;
}

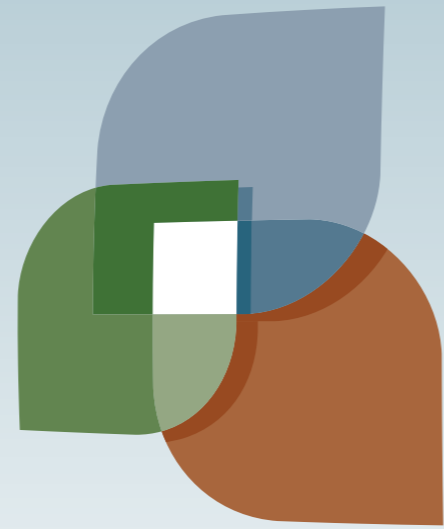
function myGoodLoadXMLHelper()
{
    var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px red\">";
    newHTML += escapeText($(this).text());
    newHTML += "</div>";
    $("div.dataXML").append(newHTML);
}

function myGoodLoadXML(data) {
    $("div.dataXML").html("<h1>AJAX XML call returned:</h1>");
    $(data).find("item title").each(myGoodLoadXMLHelper);
}
```


jQuery - JSON

```
function myGoodLoadJSON(data) {  
    $("div.dataJSON").html("<h1>AJAX JSON call returned:</h1>");  
  
    var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px green\">";  
    newHTML += escapeText(data.name);  
    newHTML += "</div>";  
    $("div.dataJSON").append(newHTML);  
  
}
```

```
function myGoodLoadJSONP(data) {  
    $("div.dataJSONP").html("<h1>AJAX JSONP call returned:</h1>");  
  
    for(var i = 0; i < data.feed.entry.length; i++){  
        var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px blue\">";  
        newHTML += escapeText(data.feed.entry[i].title.$t);  
        newHTML += "</div>";  
        $("div.dataJSONP").append(newHTML);  
    }  
}
```



L U C I

