



The Javascriptor's Dream

DESCRIPTION

- JavaScript Library
- Interaction between JavaScript and HTML
- Open Source
- Separates “behavior” characteristics from HTML

USE

- jquery.js
- `<script type="text/javascript" src="jquery.js"></script>`

AJAX

- Asynchronous data retrieval
- Interactive web applications
- XMLHttpRequest object
- Ajax and jQuery

ADVANTAGES OF AJAX

- Pages with similar content would have to be loaded every time a new request is sent
- Sections of pages reloaded
- Reduce connections to server
- States can be maintained throughout a website

DISADVANTAGES OF AJAX

- Back button usually does not work
- Browser bookmarks have problems
- Web crawlers can't find individual states of an app
- Problems with mobile devices
- Hard to test Ajax apps

DAVE - JQUERY

SELECTORS

`$(" ")`

<code>#id</code>	Matches a single element with the given id attribute.
<code>element</code>	Matches all elements with the given name.
<code>.class</code>	Matches all elements with the given class.
<code>sel1, sel2, selN</code>	Matches the combined results of all the specified selectors.
<code>anc descend</code>	Matches all descendant elements specified by "descendant" of elements specified by "ancestor".
<code>:first</code>	Matches the first selected element.
<code>:last</code>	Matches the last selected element.
<code>:not(selector)</code>	Filters out all elements matching the given selector.
<code>:even</code>	Matches even elements, zero-indexed.
<code>:odd</code>	Matches odd elements, zero-indexed.
<code>:contains(text)</code>	Matches elements which contain the given text.
<code>attribute=value</code>	Matches elements that have the specified attribute with a certain value.

COMMON FUNCTIONS

<code>attr(name)</code>	Access a property on the first matched element.
<code>attr(key, value)</code>	Set a single property to a value, on all matched elements.
<code>text()</code>	Get the combined text contents of all matched elements.
<code>text(val)</code>	Set the text contents of all matched elements.
<code>css(name)</code>	Return a style property on the first matched element.
<code>css(name, value)</code>	Set a single style property to a value on all matched elements.
<code>show(speed, callback)</code>	Show all matched elements using a graceful animation and firing an optional callback after completion.
<code>hide(speed, callback)</code>	Hide all matched elements using a graceful animation and firing an optional callback after completion.
Also:	<code>fadeIn, fadeOut, slideDown, slideUp, toggle</code>

MORE COMMON FUNCTIONS

<code>append(content)</code>	Append content to the inside of every matched element.
<code>prepend(content)</code>	Prepend content to the inside of every matched element.
<code>after(content)</code>	Insert content after each of the matched elements.
<code>before(content)</code>	Insert content before each of the matched elements.
<code>remove()</code>	Removes all matched elements from the DOM.
<code>clone()</code>	Clone matched DOM Elements and select the clones.
	Many, many more...

EXAMPLES

- Replacing contents by class:
- `$(".CIA").text("Nothing to see here!");`
- Adding checked to a checkbox:
- `$("input type=checkbox").attr("checked", "checked");`
- Filtering visible rows by search text:
- `$("div").show();`
- `$("div:not(:contains('filterText'))").hide();`

TABLE MANIPULATION

- Difficult process in regular Javascript – cross-browser issues
- Insert row after “selected class” column:

```
$("#table tr.selected").after("<tr><td></td></tr>");
```
- Duplicate a row and add at the end:

```
$("#table tbody").append($("#table tr.selected").clone());
```
- Striping table rows (light green):

```
$(table tr:even).css("background-color", "#CCFFCC");
```

JQUERY ARRAY TOOLS

<code>\$.each(object, callback)</code>	Perform the specified function for each of the items in the array (also a version for selectors)
<code>\$.isArray(value, array)</code>	Determine the index of the first parameter in the Array (-1 if not found).
<code>\$.unique(array)</code>	Removes duplicates from array.

EVENTS

- Event Binding is made much easier.
 - Simply specify the function off of a selector:
 - ```
$("#myItem").click(function () {
 alert("you clicked me!");
});
```
- Some common events:
  - `hover( over, out )`
  - `click( fn )`
  - `keypress( fn )`
  - `mousedown( fn )`
  - `mouseover( fn )`
  - `blur( fn )`
  - `keydown( fn )`
  - `keyup( fn )`
  - `mouseout( fn )`
  - `mouseup( fn )`

# \$.READY(FUNCTION () { });

- Ready function one of the most useful parts of jQuery
- Runs function the moment the DOM is loaded, instead of waiting for the rest of the page to load.
- Speeds up javascript execution at load
- Prevents flickering from initial javascript DOM manipulation.

# JQUERY

- Makes DOM manipulation easy and fast
- Easy way to change many items at once
- Very flexible
- Many improvements in AJAX as well...

KYLE - JQUERY AND AJAX

# JQUERY AJAX - ADVANTAGES

- Messy details are abstracted, but still very functional.
- One function call for all browsers - IE, Firefox, Safari, Opera, and Chrome (according to [jquery.com](http://jquery.com))

# JQUERY AJAX – CODE COMPARISON

```
function ajaxFunction() {
 var xmlhttp;

 try {
 // Firefox, Opera 8.0+, Safari
 xmlhttp=new XMLHttpRequest();
 } catch (e) {
 // Internet Explorer
 try {
 xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
 } catch (e) {
 try {
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 } catch (e) {
 alert("Your browser does not support AJAX!");
 return false;
 }
 }
 }
}
```

# JQUERY AJAX – CODE COMPARISON

(Ugliness Continued)

```
xmlHttp.onreadystatechange=function() {

 if(xmlHttp.readyState==4) {
 document.myForm.field.value+xmlHttp.responseText;

 }

}

xmlHttp.open("GET","example.aspx",true);

xmlHttp.send(null);
```

# JQUERY AJAX – CODE COMPARISON

How to do the exact same operation with jQuery

```
$.get("example.aspx", function(data) {
 $('#field').val(data);
});
```

# JQUERY AJAX - METHODS

- \$.ajax (main jQuery AJAX method; all other methods are simpler subsets of this one)
- \$.get (simple way to do GET requests)
- \$.post (simple way to do POST requests)
- \$.load (AJAX virtual load of content from another file into the DOM)
- \$.getScript (same as above, but with .js file)
- \$.getJSON (specialized method for JSON that allows cross-domain AJAX requests)

# JQUERY AJAX - CUSTOMIZATION

Some things you can customize:

- Synchronous or Asynchronous communication
- Pre-callback handler, Callback handler, error handler
- Data and Data Type
- Request Type (GET, POST, PUT, DELETE)
- Username and Password for authentication

# JQUERY AJAX - EXAMPLE

```
jQuery.post("peopleInfoSearch.cfm", {fn: fname,
 ln: lname, bn: boxnum, dp: deptgroup, ca:
 curAjaxNum, ShowRelatives: additionalresults},
 handleAjaxPeopleInfoSearch);
```

DRAKE - PEOPLE SEARCH

# PEOPLE INFO SEARCH

- People Search
  - Event Handlers
  - Ajax requests
- People Results
  - DOM Manipulation
  - Resulting actions

## People Search

Last Name:  First Name:  Box #:  Dept. / Group:

## People Results

| First Name | Last Name | Middle Name | Box # | Classification | Dept. / Group   |
|------------|-----------|-------------|-------|----------------|-----------------|
| Drake      | Angle     | Robert      | 2109  | Student        |                 |
| John       | Anglea    | C.          |       | Facc/Staff     | Business Office |

# PEOPLE SEARCH

## EVENT HANDLERS

- JavaScript

“document.getElementById("test").onKeyUp = function { }”

- jQuery

```
//Set event handlers for the search fields
//First Name Event Handler
$("#fname").keyup(function (e) {
 search(e);
});
//Last Name Event Handler
$("#lname").keyup(function (e) {
 search(e);
});
//Box Num Event Handler
$("#boxnum").keyup(function (e) {
 search(e);
});
//Dept. / Group Event Handler
$("#deptgroup").keyup(function (e) {
 search(e);
});
```

# PEOPLE SEARCH

## AJAX REQUESTS

- JavaScript

```
function getXmlHttpRequestObject() {
 if (window.XMLHttpRequest) {
 return new XMLHttpRequest(); //Not IE
 } else if(window.ActiveXObject) {
 return new ActiveXObject("Microsoft.XMLHTTP"); //IE
 } else {
 alert("Your browser doesn't support Ajax.");
 }
}

function test()
{
 if (ajax.readyState == 4 || ajax.readyState == 0) {
 ajax.open("GET",URL, true);
 ajax.onreadystatechange = callBack;
 ajax.send(null);
 }
}
```

- jQuery

“\$.post(url, data, callback);”

```
//Get search values
var fname = jQuery('#fname').val();
var lname = jQuery('#lname').val();
var boxnum = jQuery('#boxnum').val();
var deptgroup = jQuery('#deptgroup').val();

clearInfoTable();
//send ajax searchrequest
curAjaxNum++;
jQuery.post("peopleInfoSearch.cfm", {fn: fname, ln: lname, bn: boxnum, dp: deptgroup, ca: curAjaxNum}, handleAjaxPeopleInfoSearch);
return false;
```

# PEOPLE RESULTS

## DOM MANIPULATION

- XML Parsing

```
//list each person in xml data
jQuery(data).find("person").each(function() {

 //get individual person data from xml
 var recipientid = jQuery(this).find("recipientid").text();
 var idtype = jQuery(this).find("idtype").text();
 var firstname = jQuery(this).find("firstname").text();
 var lastname = jQuery(this).find("lastname").text();
 var middlename = jQuery(this).find("middlename").text();
 var boxn = jQuery(this).find("box").text();
 var email = jQuery(this).find("email").text();
 var classname = jQuery(this).find("classname").text();
 var deptgroup = jQuery(this).find("deptgroup").text();
```

# PEOPLE RESULTS

## DOM MANIPULATION

- Object Creation

```
//create new table DOM object for each person
var newTable = jQuery('<table onclick="highlightIndex('+maxIndex+');" onDb1Click="selectIndex();" id="rt'+maxIndex+
 '" border="0" cellspacing="0" cellpadding="0" ><tr align="center">'+
 '<td width="130px">'+
 '<div id="rtrecipientid'+maxIndex+' " style="display:none">'+recipientid+'</div>'+
 '<div id="rtidtype'+maxIndex+' " style="display:none">'+idtype+'</div>'+
 '<div id="rtemail'+maxIndex+' " style="display:none">'+email+'</div>'+
 '<div id="rtfn'+maxIndex+' ">'+firstname+'</div>'+
 '</td>'+
 '<td width="130px"><div id="rtlfn'+maxIndex+' ">'+lastname+'</div></td>'+
 '<td width="120px"><div id="rtmn'+maxIndex+' ">'+middlename+'</div></td>'+
 '<td width="60px"><div id="rtbox'+maxIndex+' ">'+boxn+'</div></td>'+
 '<td width="105px"><div id="rtcfn'+maxIndex+' ">'+classname+'</div></td>'+
 '<td width="150px"><div id="rtgrp'+maxIndex+' ">'+deptgroup+'</div></td></tr></table>');

//add table object to the resultsDiv object
newTable.appendTo('#resultsDiv');
```

# PEOPLE RESULTS

## DOM MANIPULATION

- DOM Manipulation

```
function highlightIndex(index)
{
 //unhighlight curIndex
 jQuery("#rt"+curIndex).css("background-color", "white");
 jQuery("#rt"+curIndex).css("color", "black");

 //highlight NewIndex
 curIndex=index;
 jQuery("#rt"+curIndex).css("background-color", "blue");
 jQuery("#rt"+curIndex).css("color", "white");
}

function moveDown()
{
 var tempIndex=curIndex;
 tempIndex++;
 if(tempIndex<=maxIndex)//>
 {
 //Test to see if the new record still fits in the display window
 curHeight+=jQuery("#rt"+tempIndex).height();
 if(curHeight>=135)
 {
 //The new result would not fit into the results window
 curHeight-=jQuery("#rt"+tempIndex).height();
 //move the scrollbar
 jQuery("#resultsDiv").scrollTop(jQuery("#resultsDiv").scrollTop()+
 ((jQuery("#rt"+tempIndex).height())-(135-curHeight)));
 curHeight=135;
 }
 highlightIndex(tempIndex);
 }
}
```

# PEOPLE RESULTS

## DOM MANIPULATION

- Resulting Actions

```
function selectIndex()
{
 var recipientid = jQuery("#rtrecipientid"+curIndex).text();
 var idtype = jQuery("#rtidtype"+curIndex).text();

 var email = jQuery("#rtemail"+curIndex).text();
 var fname = jQuery("#rtfn"+curIndex).text();
 var lname = jQuery("#rtlfn"+curIndex).text();
 var mname = jQuery("#rtmn"+curIndex).text();
 var boxn = jQuery("#rtbox"+curIndex).text();
 var cname = jQuery("#rtcname"+curIndex).text();
 var deptgroup = jQuery("#rtdept"+curIndex).text();

 jQuery("#formRecipientid").val(recipientid);
 jQuery("#formIdtype").val(idtype);

 jQuery("#infoFirstName").text(fname);
 jQuery("#infoMiddleName").text(mname);
 jQuery("#infoLastName").text(lname);
 jQuery("#infoBoxNum").text(boxn);
 jQuery("#infoEmail").text(email);
 jQuery("#infoClassName").text(cname);
 jQuery("#infoDeptGroup").text(deptgroup);
}
```

# REFERENCES

- [jquery.com](http://jquery.com)
- [visualjquery.com](http://visualjquery.com)