BU Center for Digital Imaging Arts

WDV - Introduction to AJAX

jQuery $.each()

Remember the PHP "foreach" structure? This is jQuery's adaptation of it. The each() function allows a set of values to be iterated and a callback function to run on each of those values. The syntax of the function is as follows:

```
$.each( collection, callback(indexInArray, valueOfElement)
```

In the example below, we see that a JSON map object can be used for the collection (similar to our AJAX request responses), and can then be sent into the callback key/value pair.

```
var grades = {
  'WDV101': 'HP',
  'WDV102: 'F'
};
$.each(map, function(key, value) {
  if(value == 'F') {
     alert('You failed: ' + key);
  }
});
```

After the server delivers the set of grades to our AJAX application, our application then notifies the user that he or she failed a particular course (if any).

Stopping the each() function

If we are in the middle of an iterating each() function, and have found the information for which we were looking, we can end the iteration by simply making the callback function return false. Returning any value other than "false" is the same as a continue statement, which means it will continue on to the next iteration of the loop.

Project

Create a JSON object that gets sent to your AJAX application upon request which delivers information about library books that are in a particular section (there are 5 books in the section). Iterate through each value of the JSON object and alert if the book is checked out and unavailable for your review.

Sean Mahoney - 02102010