**Iterating over jQuery and non-jQuery Objects**

jQuery provides an object iterator utility called $.each() as well as a jQuery collection iterator: .each(). These are not interchangeable. In addition, there are a couple of helpful methods called $.map() and .map() that can shortcut one of our common iteration use cases.

**$.each()**

[$.each()](http://api.jquery.com/jQuery.each/) is a generic iterator function for looping over object, arrays, and array-like objects. Plain objects are iterated via their named properties while arrays and array-like objects are iterated via their indices.

$.each() is essentially a drop-in replacement of a traditional for or for-in loop. Given:

|  |  |
| --- | --- |
| 1  2  3 | **var sum = 0;**    **var arr = [ 1, 2, 3, 4, 5 ];** |

Then this:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **for ( var i = 0, l = arr.length; i < l; i++ ) {**  **sum += arr[ i ];**  **}**    **console.log( sum ); *// 15*** |

Can be replaced with this:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **$.each( arr, function( index, value ){**  **sum += value;**  **});**    **console.log( sum ); *// 15*** |

Notice that we don't have to access arr[ index ] as the value is conveniently passed to the callback in $.each().

#### signature ---[jQuery.each( collection, callback(indexInArray, valueOfElement) )](http://api.jquery.com/jQuery.each/" \l "jQuery-each-collection-callbackindexInArray--valueOfElement)

In addition, given:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var sum = 0;**  **var obj = {**  **foo: 1,**  **bar: 2**  **}** |

Then this:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **for (var item in obj) {**  **sum += obj[ item ];**  **}**    **console.log( sum ); *// 3*** |

Can be replaced with this:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **$.each( obj, function( key, value ) {**  **sum += value;**  **});**    **console.log( sum ); *// 3*** |

Again, we don't have to directly access obj[ key ] as the value is passed directly to the callback.

Note that $.each() is for plain objects, arrays, array-like objects *that are not jQuery collections*.

***This would be considered incorrect:***

|  |  |
| --- | --- |
| 1  2  3  4 | ***// Incorrect:***  **$.each( $( "p" ), function() {**  ***// Do something***  **});** |

For jQuery collections, use .each().

**.each()**

[.each()](http://api.jquery.com/each/) is used directly on a jQuery collection. It iterates over each matched element in the collection and performs a callback on that object. The index of the current element within the collection is passed as an argument to the callback. The value (the DOM element in this case) is also passed, but the callback is fired within the context of the current matched element so the this keyword points to the current element as expected in other jQuery callbacks.

For example, given the following markup:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <ul>  <li><a href="#">**Link 1**</a></li>  <li><a href="#">**Link 2**</a></li>  <li><a href="#">**Link 3**</a></li>  </ul> |

.each() may be used like so:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **$( "li" ).each( function( index, element ){**  **console.log( $( this ).text() );**  **});**    ***// Logs the following:***  ***// Link 1***  ***// Link 2***  ***// Link 3*** |