**Traversing**

Once you have a jQuery selection, you can find other elements using your selection as a starting point.

For complete documentation of jQuery traversal methods, visit <http://api.jquery.com/category/traversing/>.

**Note**

Be cautious with traversing long distances in your documents -- complex traversal makes it imperative that your document's structure remain the same, something that's difficult to guarantee even if you're the one creating the whole application from server to client. One- or two-step traversal is fine, but you generally want to avoid traversals that take you from one container to another.

**Example 3.24. Moving around the DOM using traversal methods**

$('h1').next('p');

$('div:visible').parent();

$('input[name=first\_name]').closest('form');

$('#myList').children();

$('li.selected').siblings();

You can also iterate over a selection using $.fn.each. This method iterates over all of the elements in a selection, and runs a function for each one. The function receives the index of the current element and the DOM element itself as arguments. Inside the function, the DOM element is also available as this by default.

**Example 3.25. Iterating over a selection**

$('#myList li').each(function(idx, el) {

console.log(

'Element ' + idx +

'has the following html: ' +

$(el).html()

);

});