**Manipulating Elements: Part 3**

**Creating New Elements**

jQuery offers a trivial and elegant way to create new elements using the same $() method you use to make selections.

**Example 3.29. Creating new elements**

$('<p>This is a new paragraph</p>');

$('<li class="new">new list item</li>');

**Example 3.30. Creating a new element with an attribute object**

$('<a/>', {

html : 'This is a <strong>new</strong> link',

'class' : 'new',

href : 'foo.html'

});

*Note that in the attributes object we included as the second argument, the property name class is quoted, while the property names text and href are not. Property names generally do not need to be quoted unless they are reserved words (as class is in this case).*

When you create a new element, it is not immediately added to the page. There are several ways to add an element to the page once it's been created.

**Example 3.31. Getting a new element on to the page**

var $myNewElement = $('<p>New element</p>');

$myNewElement.appendTo('#content');

$myNewElement.insertAfter('ul:last'); // this will remove the p from #content!

$('ul').last().after($myNewElement.clone()); // clone the p so now we have 2

*Strictly speaking, you don't have to store the created element in a variable -- you could just call the method to add the element to the page directly after the $(). However, most of the time you will want a reference to the element you added, so you don't need to select it later.*

You can even create an element as you're adding it to the page, but note that in this case you don't get a reference to the newly created element.

**Example 3.32. Creating and adding an element to the page at the same time**

$('ul').append('<li>list item</li>');

**Note**

The syntax for adding new elements to the page is so easy, it's tempting to forget that there's a huge performance cost for adding to the DOM repeatedly. If you are adding many elements to the same container, you'll want to concatenate all the html into a single string, and then append that string to the container instead of appending the elements one at a time. You can use an array to gather all the pieces together, then join them into a single string for appending.

var myItems = [], $myList = $('#myList');

for (var i=0; i<100; i++) {

myItems.push('<li>item ' + i + '</li>');

}

$myList.append(myItems.join(''));