Drag-and-Drop Shopping Cart with jQuery UI

In this article, I`ll introduce jQuery UI Drag-and-Drop and show you how with just a few lines of code, you can make a richly interactive shopping cart that is functional and fun to use. To get right into it, I'm going to assume that you're at least familiar with jQuery (the base of jQuery UI). You should have a working (latest) copy of jQuery installed and be comfortable with the jQuery syntax and basic selectors.

Introducing jQuery UI

jQuery UI is a library of jQuery User Interface plugins that makes it really easy to create RIAs (Rich Interactive Applications). Or if you're not building an application, they can just add a bit of richness and interactivity to your site. These components and behaviours are ones we're all familiar with in desktop applications, but they are not (yet) the norm on the web. Some examples: drag-and-drop, selecting, sorting, grids, treeviews, panels, dialogs.

Note: jQuery UI makes use of the latest (as of this writing) major version of jQuery: 1.2. Also, most UI plugins require the dimensions plugin.

jQuery UI Drag-and-Drop

Drag-and-Drop in jQuery UI is achieved by two separate plugins: Draggables, and Droppables. Draggables are things you can drag around the page; droppables are things that draggables can be dropped on, like targets.

Note: Nearly any web page element can be promoted to be a UI draggable and/or droppable, including an image, a div, a span, or a list item.

Draggables

In true jQuery fashion, making an item draggable is as simple as selecting the item, and calling the **draggable** method on it. Like this:

```
$("#dragMe").draggable();
```

Since I've used the ID selector (#{id}), only one item will be selected and then made draggable. But like any good jQuery plugin, draggable works on any number of results at once. For example, the following code will select all list items on the page and make them draggable:

```
$("li").draggable();
```

The **draggable** method has a number of initialization options, any of which can be passed in an option hash object (collection of name-value pairs) as an argument to the draggable call. Here's an example:

```
$("#dragMe").draggable({
    revert: true
});
```

By default a draggable will stay where you drop it. With the revert option set to true (as above), it will return to where it was when you started dragging it, with a nice smooth animation.

Note: jQuery UI plugins adhere to a simple (and elegant) standard: **All options are optional**. Other names for this include **Sane Defaults** or **Convention over Configuration**. The defaults should work in most cases which will minimize the amount you have to look up in the documentation and then specify.