

Simple List Reorderer Tutorial - v0.4

This documentation refers to the API documents for the v0.4 released version of the Fluid Infusion code. For documentation specific to the trunk version, please see [List Reorderer Tutorial](#)

This page will walk you through an example of using the Fluid Reorderer's `reorderList()` function to reorder a list in an HTML file.

This tutorial assumes that:

- you are already familiar with HTML, Javascript and CSS
- you are familiar with what the Reorderer is and does
- now you just want to know how to add it to your file.

For more general information about the Reorderer, see [Reorderer](#). For technical API documentation, see [Simple List Reorderer API - v0.4](#) and [Reorderer API - v0.4](#).

Tutorial: How to Use the List Reorderer

Scenario

You are organizing a small conference, and there is lots to do. Being the organized person you are, you create a to-do list. You'd like to use the Reorderer to allow the order of items to be changed. This tutorial will show you how to use the Fluid List Reorderer for this.

There are four basic steps to using the List Reorderer in your application:

- Step 0: Download and install the Fluid Infusion library
- Step 1: Prepare your markup
- Step 2: Write the script
- Step 3: Add the script to your HTML
- Step 4: Apply styles

The rest of this tutorial will explain each of these steps in detail.

On This Page

- [Scenario](#)
- [Step 0: Download and install the Fluid Infusion library](#)
- [Step 1: Prepare your markup](#)
- [Step 2: Write the script](#)
- [Step 3: Add the script to your HTML](#)
- [Step 4: Apply styles](#)

See Also

- [List Reorderer API](#)
- [Reorderer](#)
- [Reorderer API - v0.4](#)
- [Integrating the Reorderer](#)

Still need help?

Join the [fluid-talk mailing list](#) and ask your questions there.

Step 0: Download and install the Fluid Infusion library

1. Download a copy of the Fluid Infusion component library from:
 - <http://fluidproject.org/index.php/downloads>
You only really need the "Minified deployment package," but if you want to actually look at the code, you should download the "Source package."

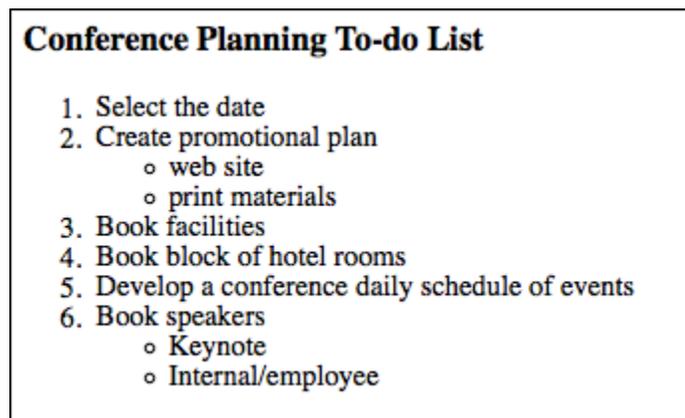
- Unpack the zip file you just downloaded, and place the resulting folder somewhere convenient for your development purposes. The folder will have the release number in its name (e.g. `fluid-0.4/`). The rest of this tutorial will use `fluid-0.4` in its examples, but if you downloaded a different version, you'll have to adjust.

Step 1: Prepare your markup

Let's assume that you're starting with an HTML file, called `todo-list.html`, that contains an ordered list:

```
<ol>
  <li>Select the date</li>
  <li>Create promotional plan
    <ul>
      <li>web site</li>
      <li>print materials</li>
    </ul>
  </li>
  <li>Book facilities</li>
  <li>Book block of hotel rooms</li>
  <li>Develop a conference daily schedule of events</li>
  <li>Book speakers
    <ul>
      <li>Keynote</li>
      <li>Internal/employee</li>
    </ul>
  </li>
  ...
</ol>
```

In a browser window, this might look something like this:



The List Reorderer needs to know about the 'container' of your list. In this case, that would be the `` element. The Reorderer accepts a jQuery selector, so you can choose any method that will uniquely identify the `` element. We'll attach a unique ID to it:

```
<ol id="todo-list">
  <li>Select the date</li>
  ...
```

Note

This example uses an ID, but you might, for example, use a CSS class, or the element hierarchy - whatever works, so long as it uniquely identifies the right element.

You also need to tell the Reorderer which of your list items should be reorderable. Perhaps the first item on your list, "Select the date," really needs to *stay* the first item, since everything else depends on the date. You don't want that item to be moveable, so you can't just make *every* list item orderable.

You'll tell the Reorderer which items are to be orderable with another jQuery selector. Let's add a CSS class for that, say `movable`:

```
<ol id="todo-list">
```

```

<li>Select the date</li>
<li class="movable">Create promotional plan
  <ul>
    <li>web site</li>
    <li>print materials</li>
  </ul>
</li>
<li class="movable">Book facilities</li>
<li class="movable">Book block of hotel rooms</li>
<li class="movable">Develop a conference daily schedule of events</li>
<li class="movable">Book speakers
  <ul>
    <li>Keynote</li>
    <li>Internal/employee</li>
  </ul>
</li>
...
</ol>

```

Note

As with the ID on the ``, we can use any jQuery selector. For example, we could attach a unique ID to each movable `` with a unique prefix, maybe `mylist-movable1`, `mylist-movable2`, etc. Then we could use the jQuery selector `[fluid:id^=mylist-movable]`

That's all - these are the only changes you *need* to make to your HTML.

Step 2: Write the script

You'll need to create a file, say `todo-list.js`, to contain your initialization script - the script you write to apply the Reorderer to your list.

In this file, write a function that calls the :

```

jQuery(document).ready(function () {
  return fluid.reorderList("#todo-list", ".movable");
});

```

In this function call, the first parameter, `"#todo-list"`, is a jQuery selector identifying the element with the ID "todo-list". The second parameter is a jQuery selector identifying elements with the CSS class "movable". That's all the information required by the `fluid.reorderList()` function.

By enclosing the function call inside `jQuery(document).ready()`, we ensure that the list is fully rendered before we apply the Reorderer to it.

Step 3: Add the script to your HTML

You'll need to add your initialization script, along with the Fluid library, to you HTML file. In the header of the file, link to the Javascript files with `<script>` tags:

```

<script type="text/javascript" src="fluid-0.4/fluid-components/js/Fluid-all.js"></script>
<script type="text/javascript" src="todo-list.js"></script>

```

NOTE that the `Fluid-all.js` file is minified - all of the whitespace has been removed, so it isn't really human-readable. If you're using the source distribution and you want to be able to debug the code, you'll want to include each of the required files individually. This would look like this:

```

<script type="text/javascript" src="fluid-0.4/fluid-components/js/jquery/jquery-1.2.6.js"></script>
<script type="text/javascript" src="fluid-0.4/fluid-components/js/jquery/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="fluid-0.4/fluid-components/js/jquery/jARIA.js"></script>
<script type="text/javascript" src="fluid-0.4/fluid-components/js/fluid/Fluid.js"></script>
<script type="text/javascript" src="fluid-0.4/fluid-components/js/fluid/Reorderer.js"></script>
<script type="text/javascript" src="todo-list.js"></script>

```

But all of these individual files are not necessary to make it work - the `Fluid-all.js` file has everything you need.

That's it! That's all you need to do to make your list reorderable!

Step 4: Apply styles

BUT: If you look at your file in a browser now, it doesn't look any different than it looked before - there's no way to tell that your list items are reorderable. That's what the styles are for.

There are a number of "interesting moments" that happen when dealing with a reorderable list. These include, for example, when the mouse hovers over a reorderable item, or when an item is selected using the keyboard. The Reorderer automatically applies CSS classes to the list items to mark these interesting moments. You can use these classes to define styles that inform users of the moments.

The CSS classes that are applied by the Reorderer, and the 'interesting moments' they are used for, are described below.

Note

The class names given below are the defaults. It is possible to override these, if you desire. For information on how to do this, see the [List Reorderer API](#).

Class: "orderable-default"

What: any item that is orderable

When: default state

Why: so that users can know that the item is orderable

Sample style:

```
.orderable-default{
  background-color: #eee;
}
```

Class: "orderable-selected"

Applied to the element that has been selected. The selected item can then be moved using keystrokes

Sample style:

```
.orderable-selected{
  background-color: #ddd;
}
```

Class: "orderable-dragging"

Applied to an element while it is in the middle of being dragged

Sample style:

```
.orderable-dragging {
  background-color: lightyellow;
}
```

Class: "orderable-hover"

Applied to an element while the cursor is hovering over top of it

Sample style:

```
.orderable-hover{
  background-color: lightyellow;
  cursor: move;
}
```

Class: "orderable-avatar"

Applied to a clone of the dragged element that is carried around with the cursor while being dragged

Sample style:

```
.orderable-avatar {
  opacity: 0.55;
  width: 300px;
  border: 0;
}
```

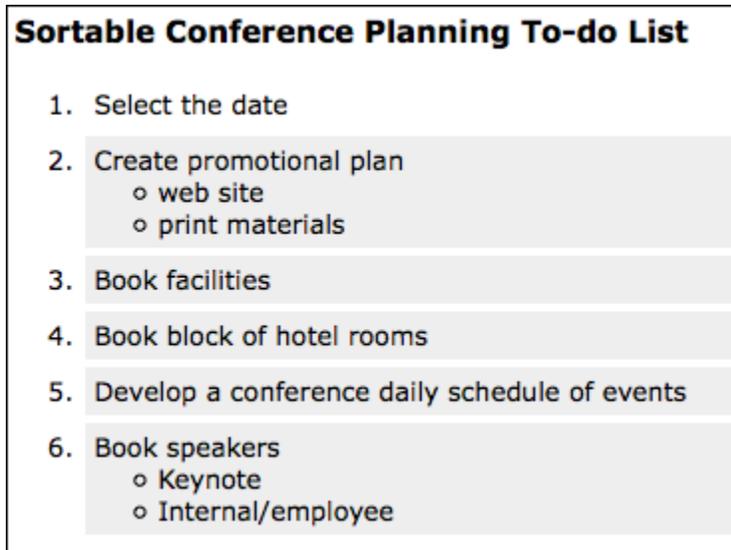
Class: "orderable-drop-marker"

Applied to an element that is created and used to indicate where the item will end up when it is dropped

Sample style:

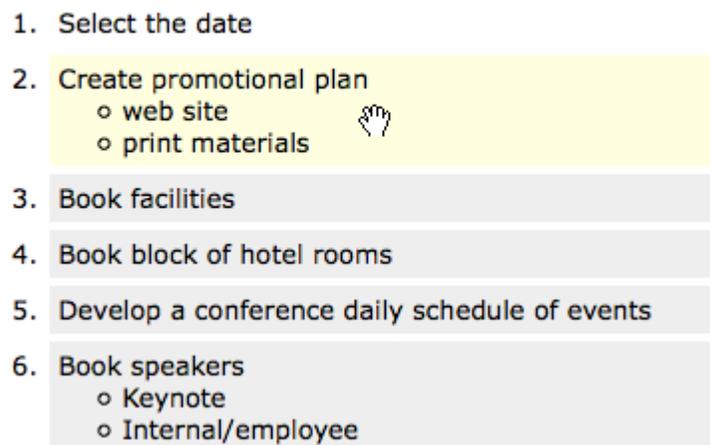
```
.orderable-drop-marker{
  background-color: red;
  height: 2px;
  margin: -4px;
  padding: 0px;
}
```

If we add a stylesheet with these styles (along with some global styles to make the font look a bit nicer, etc.), your list will look like this:



When you hover over an item, it will be apparent:

Sortable Conference Planning To-do List



If you use the tab and arrow keys to select an item, you'll be able to tell:

Sortable Conference Planning To-do List

1. Select the date
2. Create promotional plan
 - o web site
 - o print materials
3. Book facilities
4. Book block of hotel rooms
5. Develop a conference daily schedule of events
6. Book speakers
 - o Keynote
 - o Internal/employee

And while you're dragging, you'll be able to tell where the drop will happen:

Sortable Conference Planning To-do List

1. Select the date
2. Create promotional plan
 - o web site
 - o print materials
3. Book facilities
4. Book block of hotel rooms
5. Develop a conference daily schedule of events
6. Book speakers
 - o Keynote
 - o Internal/employee