

# Tutorial - Dropdown Inline Edit

This page will walk you through an example of adding the Fluid [Dropdown Inline Edit](#) component to an HTML file. For more general information about the Dropdown Inline Edit API, see [Dropdown Inline Edit API](#).

This tutorial assumes that:

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

## Tutorial: How to Use the Dropdown Inline Edit

### Scenario

You've created a database to keep track of your vast collection of CDs, and you're working on a web interface for it. When viewing the details of a CD, you would like to very easily categorize its genres. This tutorial will show you how to use the Fluid Dropdown Inline Edit for this.

There are four basic steps to adding the Inline Edit to your application:

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

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

#### Status

This component is in [Sneak Peek status](#)

#### On This Page

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

#### See Also

- [Inline Edit](#)
- [Dropdown Inline Edit API](#)

#### Still need help?

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

## Setup: Download and install the Fluid Infusion library

1. Download a copy of the Fluid Infusion component library from:
  - <http://fluidproject.org/products/fluid-infusion/download-infusion/>  
You only really need the "Minified deployment package," but if you want to actually look at the code, you should download the "Source package."
2. 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. infusion-1.4/). The rest of this tutorial will use infusion-1.4 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 working with some HTML that displays the detailed information about a CD in your collection - something simple like this:

```
<dl>
<dt>Artist</dt>
<dd>Portishead</dd>

<dt>Album</dt>
<dd>Third</dd>

<dt>Year</dt>
<dd>2008</dd>

<dt>Genre</dt>
<dd>Triphop</dd>

<dt>Play Length</dt>
<dd>49 min 13 s</dd>

<dt>Label</dt>
<dd>Universal Music</dd>
</dl>
```

### The Changes

The simplest way to make the *Genre* appear as a dropdown selection is to do these three things:

1. add a container for the options you want to make available
2. specify your options and wrap it inside a `<select>`
3. specify which fragment of text on the screen will inherit the dropdown (in this example, the word "Triphop" will gain the dropdown)
4. tell the Inline Edit about the container of these elements.

More specifically, we would:

1. create a `selection` element using the default class of `flc-inlineEdit-edit` and list the options you want to appear on the screen by using `option` for each.
2. wrap the `select` inside a container using the default class of `flc-inlineEdit-editContainer dropdown`.
3. wrap the text that will gain the dropdown in a container using the default class `flc-inlineEdit-text`.
4. finally, place a unique ID on the `dd`

This might look like the HTML sample to the right.

```

<dl>
<dt>Artist</dt>
<dd>Portishead</dd>

<dt>Album</dt>
<dd>Third</dd>

<dt>Year</dt>
<dd>2008</dd>

<dt>Genre</dt>
<dd id="albumGenre">
  <span class="flc-inlineEdit-text">Triphop</span>
  <span class="flc-inlineEdit-editContainer dropdown">
    <select class="flc-inlineEdit-edit" name="myselectbox" tabindex="1">
      <option value="Triphop" selected="selected">Triphop</option>
      <option value="Funk">Funk</option>
      <option value="Refuse to be Categorized">Refuse to be Categorized</option>
    </select>
  </span>
</dd>
<dt>Play Length</dt>
<dd>49 min 13 s</dd>

<dt>Label</dt>
<dd>Universal Music</dd>
</dl>

```

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

## Step 2: Write the script

The script needed to instantiate the Inline Edit Dropdown component will look like this:

```

jQuery(document).ready(function () {
  fluid.inlineEdit.dropdown("#albumGenre");
});

```

This function will look inside the element with the "albumGenre" ID (in this case, your <dd> element) for anything with the flc-inlineEditable class, and convert everything it finds into a Dropdown Inline Edit field with the appropriate options.

By putting the initialization function inside the jQuery(document).ready() call, you ensure that all of your markup is ready before you try to initialize the Inline Edit components.

This script can also be placed in a <script> block at the end of your document.

## Step 3: Add the Fluid library to your HTML

You'll need to add the Fluid library and the jQuery Selectbox Plugin to you HTML file. Note that the jQuery selectbox plugin is currently found in the tests folder of the Fluid Infusion bundle. In the header of the file, link to the Javascript files with <script> tags:

```

<script type="text/javascript" src="infusion/InfusionAll.js"></script>
<script type="text/javascript" src="infusion/tests/manual-tests/lib/jquery/plugins/selectbox/jquery.selectbox-0.5.js"></script>

```

**NOTE** that the InfusionAll.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="infusion/lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="infusion/lib/jquery/ui/js/ui.core.js"></script>
<script type="text/javascript" src="infusion/lib/jquery/plugins/tooltip/js/jquery.tooltip.js"></script>
<script type="text/javascript" src="infusion/framework/core/js/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="infusion/framework/core/js/Fluid.js"></script>
<script type="text/javascript" src="infusion/components/inlineEdit/js/InlineEdit.js"></script>
<script type="text/javascript" src="infusion/components/inlineEdit/InlineEditIntegrations.js"></script>

<script type="text/javascript" src="infusion/tests/manual-tests/lib/jquery/plugins/selectbox/jquery.selectbox-0.5.js"></script>
```

But all of these individual files are not necessary to make it work - the `InfusionAll.js` and `jquery.selectbox.js` files are everything you need.

So that's it! That's all you need to do to add the Dropdown Inline Edit functionality to your table cells.

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

## Step 4: Apply styles

The Dropdown Inline Edit component uses the jQuery selectbox plugin and currently requires the CSS that is shipped with the plugin. Note that right now this CSS is available in the `tests` directory of the Fluid Infusion bundle. We can use the [Fluid Skinning System \(FSS\)](#) CSS skin files alongside the selectbox plugin CSS files.

```
<link rel="stylesheet" type="text/css" href="infusion/framework/fss/css/fss-layout.css" />
<link rel="stylesheet" type="text/css" href="infusion/framework/fss/css/fss-theme-mist.css" />
<link rel="stylesheet" type="text/css" href="infusion/tests/manual-tests/lib/jquery/plugins/selectbox/selectbox.css" />
<link rel="stylesheet" type="text/css" href="infusion/components/inlineEdit/css/InlineEdit.css" />
```

and to attach a class attribute that represents the skin you want to the components container, such as:

```
<dl class="fl-theme-mist">
  ...
</dl>
```

The above code would make use of the `fss-theme.mist.css` skin file.

If you wish to define your own styles instead, use the following information:

The Inline Edit adds classes to the display text element that can be used to style the element to let users know that the text is editable - that is, to make the functionality 'discoverable.' These classes are applied at various 'interesting moments,' for example when the cursor hovers over the editable text, or when Tab key is used to move focus to the text.



### Important note

The visual appearance of the styles in the example below are just that: examples. You are free to create whatever styles you like. The important thing to understand is

- what the interesting moments are, and
- what the names of the styles for those moments are

The styles that are applied by the Inline Edit, and the 'interesting moments' they are used for, are:

**Style:** "fl-inlineEdit-invitation"

**When:** mouse hover and keyboard focus

**Why:** so that users can know that the text is editable

Sample style:

```
.fl-inlineEdit-invitation {
  background-color: #FFECB3 !important;
  border: 1px solid #CCC !important;
}
```

**Style:** "fl-inlineEdit-focus"

**When:** keyboard focus on the text

**Why:** so that users know that focus is on the text, and that they can press

Enter to start editing

Sample style:

```
.fl-inlineEdit-focus {
  border: 2px solid #777777;
}
```

If we add a stylesheet with these styles, the page will look like this when the mouse hovers over the album genre:

Artist  
Portishead

Album  
Third

Year  
2008

Genre  
Triphop

Play Length  
49 min 13 s

Label  
Universal Music