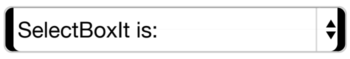
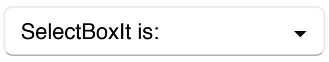
(https://github.com/gfranko/jquery.selectBoxIt.js)

# SelectBoxIt

Turns this


SelectBoxIt is:

Into this


SelectBoxIt is:

## Description

SelectBoxIt allows you to replace ugly and hard to use HTML select boxes with gorgeous and feature rich drop downs. **Twitter Bootstrap**, **jQueryUI**, and **jQuery Mobile** themes are supported right out of the box. If you don't want to use these a library theme, then you can use the SelectBoxIt **default theme**, which closely resembles the Twitter Bootstrap theme. Feel free to customize the default theme to your liking.

To use SelectBoxIt, you do not have to rewrite any of your existing form validation logic or event handling. **SelectBoxIt just works**. SelectBoxIt also provides first-class support for **mobile**, **tablet**, and **desktop** browsers, triggering the native "wheel" interface for mobile and tablet devices.

The project is hosted on Github (https://github.com/gfranko/jQuery.selectBoxIt.js), the annotated source code (http://www.gregfranko.com/jquery.selectBoxIt.js/docs/jQuery.selectBoxIt.html) is available, and an online test suite (https://travis-ci.org/gfranko/jquery.selectBoxIt.js) is also available via Travis CI. SelectBoxIt is available for use under the MIT software license (https://github.com/gfranko/jQuery.selectBoxIt.js/blob/master/MIT-LICENSE.txt). You can report bugs and discuss features on the GitHub issues page (https://github.com/gfranko/jQuery.selectBoxIt.js/issues?sort=created&direction=desc&state=open), or send tweets to @gregfranko (http://www.twitter.com/gregfranko).

Fork on Github » (https://github.com/gfranko/jquery.selectBoxIt.js)

Also, if you would like to receive updates about new SelectBoxIt releases, you can subscribe to the **SelectBoxIt Mailing List**

**Subscribe to the SelectBoxIt mailing list**

email address

**Subscribe**

## Examples

**Note:** You can try all SelectBoxIt options, events, and methods inside of the HTML editor at the top of the page. Just click the Show Editor button.

## Theming Support

### Default Theme

**Note:** The default theme closely resembles the Twitter Bootstrap theme, but does not require Twitter Bootstrap as a dependency.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box and uses the default theme
$("select").selectBoxIt();
```

### Bootstrap Theming Support

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Uses the Twitter Bootstrap theme for the drop down
  theme: "bootstrap"

});
```

### jQueryUI Theming Support

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Uses the jQueryUI theme for the drop down
  theme: "jqueryui"

});
```

### jQuery Mobile Theming Support

**Note:** SelectBoxIt supports the jQuery Mobile **data-theme** HTML5 data attribute.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Uses the jQueryUI theme for the drop down
  theme: "jquerymobile"

});
```

## Show/Hide Animations

### jQuery Show/Hide Support

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Uses the jQuery 'fadeIn' effect when opening the drop down
  showEffect: "fadeIn",

  // Sets the jQuery 'fadeIn' effect speed to 400 milleseconds
  showEffectSpeed: 400,

  // Uses the jQuery 'fadeOut' effect when closing the drop down
  hideEffect: "fadeOut",

  // Sets the jQuery 'fadeOut' effect speed to 400 milleseconds
  hideEffectSpeed: 400

});
```

### jQueryUI Show/Hide Support

Turns this
SelectBoxIt is:

Into this
SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Uses the jQueryUI 'shake' effect when opening the drop down
  showEffect: "shake",

  // Sets the animation speed to 'slow'
  showEffectSpeed: 'slow',

  // Sets jQueryUI options to shake 1 time when opening the drop down
  showEffectOptions: { times: 1 },

  // Uses the jQueryUI 'explode' effect when closing the drop down
  hideEffect: "explode"

});
```

## Hide the First Option

Turns this
SelectBoxIt is:

Into this
SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Hides the first select box option from appearing when the drop down is opened
  showFirstOption: false

});
```

## Hide the Currently Selected Option

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Hides the currently selected option from appearing when the drop down is opened
  hideCurrent: true

});
```

## Long Lists

Turns this

Pick a Country

Into this

Pick a Country

**Example CSS Code**

```
.selectboxit-container .selectboxit-options {

  /* Set's the drop down options width to the same width as the drop down button */
  width: 210px;

  /* Set's the max-height property to only show a subset of the drop down items.
     If you do not set a max-height property, SelectBoxIt will dynamically
     position the dropdown (when opened) to make sure the drop down items are not
     displayed outside of the current window viewport.
  */
  max-height: 240px;

}
```

## Trigger the Native Select Box

Turns this

Into this

SelectBoxIt is:

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Triggers the native select box when a user interacts with the drop down
  native: true

});
```

## Set Default Text

Turns this

SelectBoxIt is:

Into this

Sample text here

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Sets default text to appear for the drop down
  defaultText: "Sample text here"

});
```

## Set Size

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example HTML Code**

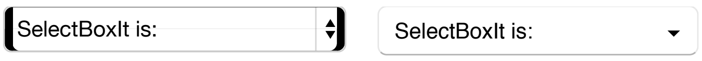
```
<select id="test" name="test" data-size="3">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <option value="a jQuery Plugin">a jQuery Plugin</option>
  <option value="a Select Box Replacement">a Select Box Replacement</option>
  <option value="a Stateful UI Widget">a Stateful UI Widget</option>
</select>
```

## Custom Selected Text

**Hint:** Click the third option.

Turns this

Into this



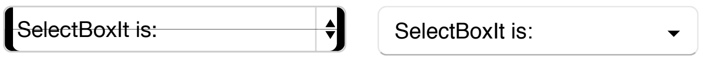
**Example HTML Code**

```html
  <select id="test" name="test">
    <option value="SelectBoxIt is:">SelectBoxIt is:</option>
    <option value="a jQuery Plugin">a jQuery Plugin</option>
    <option data-selectedtext="This is custom text" value="a Select Box Replacement">a Select Box Replacement</option>
    <option value="a Stateful UI Widget">a Stateful UI Widget</option>
  </select>
```

## Prevent Closing

**Hint:** Clicking the last drop down option will not close the options list. This is helpful when you want a second action to be triggered when clicking an option and you do not want the list to close.

Turns this Into this



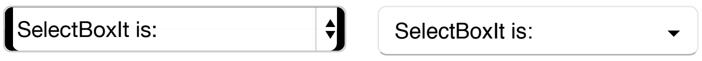
**Example HTML Code**

```html
  <select id="test" name="test">
    <option value="SelectBoxIt is:">SelectBoxIt is:</option>
    <option value="a jQuery Plugin">a jQuery Plugin</option>
    <option value="a Select Box Replacement">a Select Box Replacement</option>
    <option value="a Stateful UI Widget" data-preventclose="true">a Stateful UI Widget</option>
  </select>
```

## Aggressive Change Mode

**Note:** Aggressive Change Mode will select a drop down option (and trigger the change event on the original select box) when a user navigates to an option using the up and down arrow keys via the keyboard, or searches for an option using the keyboard.

Turns this Into this



**Example Code**

```javascript
  // Calls the selectBoxIt method on your HTML select box
  $("select").selectBoxIt({

    // Sets default text to appear for the drop down
    aggressiveChange: true

  });
```

## Native Mousedown Mode

**Hint:** Click and hold down your mouse, hover over a drop down option, and then release the mouse to select that option.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Sets default text to appear for the drop down
  nativeMousedown: true

});
```

## Populate Options

**Note:** Accepts a jQuery **Deferred/Promise Object**, an array of objects, an array of strings, a single object, a JSON array, or a valid HTML string to add options to the drop down list. The populate option can also be a function returns an accepted data format.

### jQuery Deferred Object

**Hint:** You must call the **resolve()** method and return an accepted data format.

Turns this

Into this

Greg Franko Github Repos

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({
  defaultText: "Greg Franko Github Repos",
  // Populates the drop down using a jQuery deferred object
  populate: function() {
      var deferred = $.Deferred(),
          arr = [],
          x = -1;
      $.ajax({
          url: 'https://api.github.com/users/gfranko/repos'
      }).done(function(data) {
          while(++x < data.length) {
              arr.push(data[x].name);
          }
          deferred.resolve(arr);
      });
      return deferred;
  }

});
```

### Array of Objects

**Hint:** Each object property can use jQuery shortcuts (i.e. val instead of value)

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Populates the drop down using an array of objects
  populate: [
    { value: "SelectBoxIt is:", text: "SelectBoxIt is:" },
    { value: "a jQuery Plugin", text: "a jQuery Plugin" },
    { value: "a Select Box Replacement", text: "a Select Box Replacement" },
    { value: "a Stateful UI Widget", text: "a Stateful UI Widget" }
  ]

});
```

### Array of Strings

**Hint:** Each string is used for both the value and text option properties

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Populates the drop down using an array of strings
  populate: [
    "SelectBoxIt is:",
    "a jQuery Plugin",
    "a Select Box Replacement",
    "a Stateful UI Widget"
  ]

});
```

### Single Object

**Hint:** This is helpful when only populating a drop down with one option.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Populates the drop down using an array of strings
  populate: {
    value: "SelectBoxIt is:",
    text: "SelectBoxIt is:"
  }

});
```

### JSON Array

**Hint:** This is helpful when populating a drop down with external JSON data from an ajax/jsonp/cors request.

**IMPORTANT**: You must use a property called **data**, since this is what SelectBoxIt expects.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Populates the drop down using a JSON array
  populate: {"data":[
    {"text":"SelectBoxIt is:","value":"SelectBoxIt is:"},
    {"text":"a jQuery Plugin","value":"a jQuery Plugin"},
    {"text":"a Select Box Replacement","value":"a Select Box Replacement"},
    {"text":"a Stateful UI Widget","value":"a Stateful UI Widget"}
  ]}

});
```

### HTML String

**Hint:** This is the fastest way to dynamically build a drop down list.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example Code**

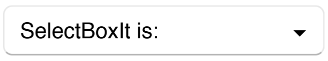
```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Populates the drop down using a JSON array
  populate: '<option value="SelectBoxIt is:">SelectBoxIt is:</option>' +
  '<option value="a jQuery Plugin">a jQuery Plugin</option>' +
  '<option value="a Select Box Replacement">a Select Box Replacement</option>' +
  '<option value="a Stateful UI Widget">a Stateful UI Widget</option>'

});
```

## Dynamically Add Options

**Hint:** The `populate` option uses the `add()` method internally, which means that the `add()` method supports all of the same data formats (arrays, objects, JSON data, HTML string)

Turns this

Into this



**Example Code**

```
 // Calls selectBoxIt on your select box
 $("select").selectBoxIt();

 // Appends a drop down option to your drop down
 $("select").data("selectBox-selectBoxIt").add({ value: "This is a new option", text: "This
is a new option" });
```

## Dynamically Remove Options

### Removing a Single Option

Turns this

Into this

**Example Code**

```
// Calls selectBoxIt on your select box
$("select").selectBoxIt();

// Removes the first drop down option from the list
$("select").data("selectBox-selectBoxIt").remove(0);
```

### Removing Multiple Options

Turns this

Into this

SelectBoxIt is: a Select Box Replacement

**Example Code**

```
// Calls selectBoxIt on your select box
$("select").selectBoxIt();

// Removes the first and second drop down options from the list
$("select").data("selectBox-selectBoxIt").remove([0,1]);
```

### Removing All Options

Turns this Into this

SelectBoxIt is:

**Example Code**

```
// Calls selectBoxIt on your select box
$("select").selectBoxIt();

// Removes all of the drop down options from the list
$("select").data("selectBox-selectBoxIt").remove();
```

## Down Arrow Customization

### Bootstrap Up/Down Arrow Support

Turns this Into this

SelectBoxIt is: SelectBoxIt is:

**Example Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  theme: "bootstrap"

});

$("select").bind({

  // Binds to the 'open' event on the original select box
  "open": function() {

    // Adds the Twitter Bootstrap 'dropup' class to the drop down
    $(this).data("selectBox-selectBoxIt").dropdown.addClass("dropup");

  },

  // Binds to the 'close' event on the original select box
  "close": function() {

    // Removes the Twitter Bootstrap 'dropup' class from the drop down
    $(this).data("selectBox-selectBoxIt").dropdown.removeClass("dropup");

  }

});
```

## Custom Down Arrow Support

Turns this

Into this

**Example JavaScript Code**

```
// Calls the selectBoxIt method on your HTML select box
$("select").selectBoxIt({

  // Set a custom down arrow icon by adding new CSS class(s)
  downArrowIcon: "icon-hand-down"

});
```

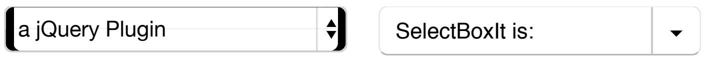
**Example CSS Code**

```
.selectboxit .selectboxit-arrow-container .selectboxit-arrow {
  top: 30%;
}
```

## Arrow Box Support

Turns this

Into this



**Example CSS Code**

```
.selectboxit-arrow-container {

  /* Encloses the down arrow in a box */
  border-left: 1px solid #ccc;

}
```

## Selected Styling Support

Turns this Into this

**Example CSS Code**

```
.selectboxit-selected {

  background: #ADD8E6;

}
```

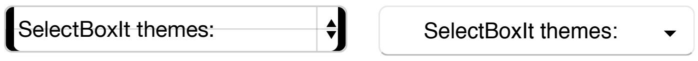
## Icons and Images

### Custom Icon Support

**Note:** This example uses absolute URL's, but you can also use relative URL's. If you are using relative URL's, make sure the image path is relative to your HTML file, since the image is being included via the *style* HTML attribute.

**Browser Support:** Modern Browsers (IE9+)

Turns this Into this



**Example HTML Code**

```
  <select id="test" name="test">
    <option value="SelectBoxIt themes:" data-iconurl="http://icons.iconarchive.com/icons/custom-icon-
design/pretty-office-5/256/themes-icon.png">
    SelectBoxIt themes:
    </option>
    <option value="Twitter Bootstrap" data-iconurl="http://blog.getbootstrap.com/public/ico/apple-tou
ch-icon-144-precomposed.png">
    Twitter Bootstrap
    </option>
    <option value="jQuery UI" data-iconurl="http://c747925.r25.cf2.rackcdn.com/blog/wp-content/upload
s/2010/09/jquery-ui-logo.png">
    jQuery UI
    </option>
    <option value="jQuery Mobile" data-iconurl="https://twimg0-a.akamaihd.net/profile_images/26339787
89/80508321d8ce3ba8aa264380bb7eba33.png">
    jQuery Mobile
    </option>
  </select>
```

### Bootstrap Icon Support

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example HTML Code**

```
  <select id="test" name="test">
    <option value="SelectBoxIt is:" data-icon="icon-search">SelectBoxIt is:</option>
    <option value="a jQuery Plugin" data-icon="icon-ok">a jQuery Plugin</option>
    <option value="a Select Box Replacement" data-icon="icon-ok">a Select Box Replacement</o
ption>
    <option value="a Stateful UI Widget" data-icon="icon-ok">a Stateful UI Widget</option>
  </select>
```

## Bootstrap Option Popovers

**Note:** This example requires the Twitter Bootstrap Tooltip and Popover plugins.

Turns this

SelectBoxIt is:

Into this

SelectBoxIt is:

**Example HTML Code**

```
  <select id="test" name="test">
    <option value="SelectBoxIt is:" rel="popover" title="SelectBoxIt" data-content="A jQuery
Select Box Plugin for Mobile, Tablet, and Desktop">
    SelectBoxIt is:
    </option>
    <option value="a jQuery Plugin" rel="popover" title="SelectBoxIt" data-content="a jQuery
plugin">
    a jQuery Plugin
    </option>
    <option value="a Select Box Replacement" rel="popover" title="SelectBoxIt" data-content=
"a SelectBox Replacement">
    a Select Box Replacement
    </option>
    <option value="a Stateful UI Widget" rel="popover" title="SelectBoxIt" data-content="a S
tateful UI Widget">
    a Stateful UI Widget
    </option>
  </select>
```

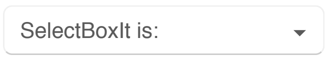
**Example JavaScript Code**

```
  // Calls the Twitter Bootstrap popover method
  $("[rel='popover']").popover({ trigger: "hover", container: "body" });
```

## Disabled Support

Turns this

Into this



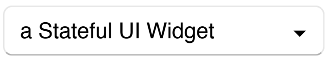
**Example HTML Code**

```
  <select id="test" name="test" disabled>
    <option value="SelectBoxIt is:">SelectBoxIt is:</option>
    <option value="a jQuery Plugin">a jQuery Plugin</option>
    <option value="a Select Box Replacement">a Select Box Replacement</option>
    <option value="a Stateful UI Widget">a Stateful UI Widget</option>
  </select>
```

## Selected and Disabled Option Support

Turns this

Into this



**Example HTML Code**

```
<select id="test" name="test">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <option value="a jQuery Plugin" disabled>a jQuery Plugin</option>
  <option value="a Select Box Replacement" disabled>a Select Box Replacement</option>
  <option value="a Stateful UI Widget" selected>a Stateful UI Widget</option>
</select>
```

## Optgroup Support

Turns this

a jQuery Plugin

Into this

a jQuery Plugin

**Example HTML Code**

```
<select id="optgroups" name="optgroups">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <optgroup label="Optgroup 1">
  <option value="a jQuery Plugin">a jQuery Plugin</option>
  </optgroup>
  <optgroup label="Optgroup 2">
  <option value="a Select Box Replacement">a Select Box Replacement</option>
  </optgroup>
  <optgroup label="Optgroup 3">
  <option value="a Stateful UI Widget">a Stateful UI Widget</option>
  </optgroup>
</select>
```

## HTML Option Support

**Important:** HTML is supported by using the **data-text** HTML5 data attributes for each option. Make sure to use single quotes to surround your text and double quotes for class names, attributes, etc, that are used within your HTML text.

Turns this

a jQuery Plugin

Into this

SelectBoxIt is:

**Example HTML Code**

```
  <select id="optgroups" name="optgroups">
    <option value="SelectBoxIt is:">SelectBoxIt is:</option>
    <option value="a jQuery Plugin" data-text='a <strong style="font-weight:bold;">jQuery</s
trong> Plugin'></option>
    <option value="a jQuery Plugin" data-text='a <strong style="font-weight:bold;">Select Bo
x </strong> Replacement'></option>
    <option value="a jQuery Plugin" data-text='a Stateful UI <strong style="font-weight:bold
;">Widget</strong>'></option>
  </select>
```

# Requirements

jQuery 1.8.3+ (http://jquery.com) (It is always recommended to use the latest version of jQuery)

jQueryUI Widget Factory 1.10.0+ (http://jqueryui.com/download) (It is always recommended to use the latest version of the jQueryUI Widget Factory)

## Optional Dependencies

Twitter Bootstrap (http://twitter.github.com/bootstrap/), jQueryUI CSS Theme (http://jqueryui.com/themeroller/), or jQuery Mobile Theme (http://jquerymobile.com/themeroller/index.php)

jQueryUI effects (http://jqueryui.com/download) (only the custom Effects you use are required)

## Notable Features

Supports Mobile, Tablet, and Desktop browsers

Themeable with Twitter Bootstrap, jQueryUI, and jQuery Mobile

Built-in ARIA support (Accessible Rich Internet Applications)

Full keyboard search and navigation support

An event API triggered on the original select box element that calls the plugin

A method API providing developers with methods to interact with the dropdown list (i.e. Search, Open, Disable, Set Options).

Selected, disabled, and optgroup support

Easily extendable to allow developers to create new widgets

## Desktop Browser Support

Tested in IE8+, Firefox 4+, Chrome, Safari 4+, and Opera 11+

## Mobile/Tablet Browser Support

Tested in iOs 3+ and Android 2.1+

## Getting Started

SelectBoxIt is proudly hosted by CloudFare CDN servers via the open source project, cdn.js (http://cdnjs.com/).

**Note:** If a new SelectBoxIt version is not on the CDN yet (it usually takes within 1-3 days), you can get the latest JavaScript and CSS from Github here (https://github.com/gfranko/jquery.selectBoxIt.js/tree/master/src). The files you will most likely want are jquery.selectBoxIt.min.js (https://github.com/gfranko/jquery.selectBoxIt.js/blob/master/src/javascripts/jquery.selectBoxIt.min.js) and jquery.selectBoxIt.css (https://github.com/gfranko/jquery.selectBoxIt.js/blob/master/src/stylesheets/jquery.selectBoxIt.css). You can also create a custom SelectBoxIt build (http://gregfranko.com/jquery.selectBoxIt.js/customDownload.html).

## Include CSS files

**Note:** Pick **one** of the themes below

### SelectBoxIt CSS with Default Theme

```
<link type="text/css" rel="stylesheet" href="http://gregfranko.com/jquery.selectBoxIt.js/css/
jquery.selectBoxIt.css" />
```

### SelectBoxIt with Twitter Bootstrap

```
<link type="text/css" rel="stylesheet" href="http://netdna.bootstrapcdn.com/twitter-bootstrap
/2.2.2/css/bootstrap-combined.min.css" />
<link type="text/css" rel="stylesheet" href="http://http://gregfranko.com/jquery.selectBoxIt.
js/css/jquery.selectBoxIt.css" />
```

### SelectBoxIt with jQueryUI

```
<link type="text/css" rel="stylesheet" href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.
9.2/themes/base/jquery-ui.css" />
<link type="text/css" rel="stylesheet" href="http://gregfranko.com/jquery.selectBoxIt.js/css/
jquery.selectBoxIt.css" />
```

### SelectBoxIt with jQuery Mobile

```
<link type="text/css" rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobil
e-1.2.0.min.css" />
<link type="text/css" rel="stylesheet" href="http://gregfranko.com/jquery.selectBoxIt.js/css/
jquery.selectBoxIt.css" />
```

## Include JavaScript files

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.2/jquery-ui.min.js"></script
>
<script src="http://gregfranko.com/jquery.selectBoxIt.js/js/jquery.selectBoxIt.min.js"></scr
ipt>
```

# HTML

Create an **HTML select box** with **id**, **class**, or **name** attributes. If you include an **id** attribute, SelectBoxIt will create and copy a "unique" **id** attribute to the dropdown list. If you include a **class** and/or **name** attributes, the attributes will be copied to the new dropdown list that the plugin creates (this allows you to easily interact with the new dropdown list without having to know a new **id** attribute). It is also best practice for each select box option to have a **value** attribute, but it is not required.

**Note:** All inline styles will also be copied to the new dropdown list.

```
<select id="test" name="test">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <option value="a jQuery Plugin">a jQuery Plugin</option>
  <option value="a Select Box Replacement">a Select Box Replacement</option>
  <option value="a Stateful UI Widget">a Stateful UI Widget</option>
</select>
```

**Note:** Your select box option **value** attributes and **text** do not have to be the same.

## Selected and Disabled Support

SelectBoxIt supports the **selected** and **disabled** HTML properties. Keep in mind that the last select box option to contain the **selected** property will be the select box option that the dropdown list uses as it's default option. Also, the **disabled** property can be used to disable the entire dropdown or specific dropdown options.

Here is an example of setting a select box option using the **selected** attribute:

```
<select id="test" name="test">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <option value="a jQuery Plugin">a jQuery Plugin</option>
  <option value="a Select Box Replacement">a Select Box Replacement</option>
  <option value="a Stateful UI Widget" selected>a Stateful UI Widget</option>
</select>
```

Here is an example of setting the **disabled** property for multiple individual select box options:

```
<select id="test" name="test">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <option value="a jQuery Plugin" disabled>a jQuery Plugin</option>
  <option value="a Select Box Replacement">a Select Box Replacement</option>
  <option value="a Stateful UI Widget" disabled>a Stateful UI Widget</option>
</select>
```

## Optgroup Support

SelectBoxIt supports **optgroups**. You have full control to style the optgroups by using the *optgroupHeader* and *optgroupOption* CSS classes. There is no special syntax to use optgroups, just use them like you normally do.

Here is an example of a dropdown that uses **optgroups**:

```
<select id="optgroups" name="optgroups">
  <option value="SelectBoxIt is:">SelectBoxIt is:</option>
  <optgroup label="Optgroup 1">
    <option value="a jQuery Plugin">a jQuery Plugin</option>
  </optgroup>
  <optgroup label="Optgroup 2">
    <option value="a Select Box Replacement">a Select Box Replacement</option>
  </optgroup>
  <optgroup label="Optgroup 3">
    <option value="a Stateful UI Widget">a Stateful UI Widget</option>
  </optgroup>
</select>
```

### CSS

**Note:** If you have multiple select boxes on the page and want each select box to have unique styles, then you can add an **id** attribute to your select box, since SelectBoxIt uses the **id** attribute on your select box and appends the word "SelectBoxIt" to the end.

For example, if your **id** attribute is *test*, then SelectBoxIt will create a *testSelectBoxIt* **id** attribute on your new drop down and a *testSelectBoxItOptions* **id** attribute on your new drop down list.

Also, if you are interested in just using class names to style your drop downs, check out this jsfiddle (http://jsfiddle.net/PgzDX/4/).

You may use **class names** or **name** attributes, since SelectBoxIt copies both over to the new drop down.

### JavaScript

**Call the plugin:** In your JavaScript code, call the **selectBoxIt()** method on your HTML select box.

```
//Executes your code when the DOM is ready.  Acts the same as $(document).ready().
  $(function() {
    //Calls the selectBoxIt method on your HTML select box.
    $("select").selectBoxIt();
  });
```

## Options API

The Options API allows you to customize the dropdown list by **setting** custom options and **getting** all current options.

### Options

| Name | Type | Default | Options |
|---|---|---|---|
| showEffect | String | "none" | "none", "fadeIn", "show", "slideDown", or any of the jQ show effects (http://jqueryui.com/demos/s (i.e. "bounce") |
| showEffectOptions | Object Literal | { } | All of the available properties based on the jqueryUI effect (http://docs.jquery.com/UI/Ef {direction: "down"} ) |
| | | | |

| | | | |
|---|---|---|---|
| showEffectSpeed | String or Number | "medium" | "slow", "medium", "fast", or a numeric number (millisecond |
| hideEffect | String | "none" | "none", "fadeOut", "hide", "s or any of the jQueryUI hide ef (http://jqueryui.com/demos/h "bounce") |
| hideEffectOptions | Object Literal | { } | All of the available properties based on the jqueryUI effect (http://docs.jquery.com/UI/Ef {direction: "up"} ) |
| hideEffectSpeed | String or Number | "medium" | "slow", "medium", "fast", or a numeric number (millisecond |
| showFirstOption | Boolean | true | true or false |
| defaultText | String | " " | Any text may be used |
| defaultIcon | String | " " | Any valid string classname m used |
| downArrowIcon | String | " " | Any valid string classname m used |
| theme | String | "default" | "default", bootstrap", "jquery "jquerymobile" |
| keydownOpen | Boolean | true | true or false |
| isMobile | Function | function() { return (/iPhone\|iPod\|iPad\|Android\|BlackBerry\|Opera Mini\|IEMobile/).test(navigator.userAgent);} | Any function that returns true |
| copyAttributes | Array | ["title", "rel"] | An array of HTML attributes |
| copyClasses | String | "button" | "none", "button", or "contain |
| native | Boolean | false | true or false |
| aggressiveChange | Boolean | false | true or false |
| selectWhenHidden | Boolean | true | true or false |
| viewport | jQuery selector | $(window) | Any jQuery selector |
| similarSearch | Boolean | false | true or false |
| nativeMousdown | Boolean | false | true or false |
| customShowHideEvent | Boolean | false | true or false |
| autoWidth | Boolean | true | true or false |
| html | Boolean | true | true or false |
| populate | JSON, Array, Object, or String | "" | JSON, array, single object, or HTML string |
| dynamicPositioning | Boolean | true | true or false |
| hideCurrent | Boolean | false | true or false |

## HTML5 Data Attribute Options

These options can be set on the select box or individual options directly within the HTML code

| Name | Type | Default | Options |
|---|---|---|---|

| | | | |
|---|---|---|---|
| data-icon | String | " " | Any valid string classname may be used |
| data-iconurl | String | " " | Any valid absolute or relative string URL |
| data-downarrow | String | " " | Any valid string classname may be used |
| data-text | String | " " | Any text may be used |
| data-preventclose | String | " " | "true" or "false" |
| data-selectedtext | String | " " | Any text may be used |

## Setting Options

All options can be **set** when the plugin is called, or after the plugin is called, using the **setOption()** or **setOptions()** methods.

Here is an example of setting a **single option** when the plugin is first invoked:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  //Calls the selectBoxIt method on your HTML select box and updates the showEffect option
  $("select#test").selectBoxIt({ showEffect: "fadeIn" });

});
```

Here is an example of setting **multiple options** after the plugin is called using the **setOptions()** method:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  //Calls the selectBoxIt method on your HTML select box
  var selectBox = $("select#test").selectBoxIt().data("selectBox-selectBoxIt");

  // Updates both the showEffect and showEffectSpeed options
  selectBox.setOptions({ showEffect: "fadeIn", showEffectSpeed: "medium" });

});
```

Here is an example of setting a **single option** after the plugin is called using the **setOption()** method:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  //Calls the selectBoxIt method on your HTML select box
  var selectBox = $("select#test").selectBoxIt().data("selectBox-selectBoxIt");

  // Updates the showEffect option
  selectBox.setOption("showEffect", "fadeIn");

});
```

## Getting Options

A single option can be **retrieved** by using the **option()** method. All of the current options can be retrieved by referencing the **options** property.

Here is an example of retrieving a **single option** after the plugin is called using the **option()** method:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  //Calls the selectBoxIt method on your HTML select box
  var selectBox = $("select#test").selectBoxIt().data("selectBox-selectBoxIt");

  // Writes the showFirstOption option to the console
  console.log(selectBoxIt.option("showFirstOption"));

});
```

Here is an example of retrieving **all current options** after the plugin is called using the **options** property:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  // Calls the selectBoxIt method on your HTML select box
  var selectBox = $("select#test").selectBoxIt().data("selectBox-selectBoxIt");

  // Writes all of the current plugin options to the console
  console.log(selectBoxIt.options);

});
```

## Events API

The **Events API** allows your application to listen to user events triggered on the dropdown list. All custom/default **events are triggered on the original select box element** (not the new dropdown list).

The original select box **value** attribute is also synced with the new dropdown list, so if a user selects a new value from the dropdown list, **the original select box value will also be updated**. *This allows your existing code to continue working inside of forms.*

You can catch **Default Events** by using the jQuery **bind()** or **on()** methods, or by using jQuery convenience methods such as **click()**, **change()**, etc. *You must use the jQuery **bind()** or **on()** methods to catch **Custom Events**.*

### Default Events

| Name | Description |
|---|---|
| focus | A focus event will be triggered when a user either clicks or tabs to the dropdown list. |
| focusin | A focusin event will be triggered when a user either clicks or tabs to the dropdown list. Focus and focusin events are closely related, but the focusin event bubbles up the DOM tree and the focus event does not bubble. If you are using a library such as Backbone.js (http://www.backbonejs.org), which uses event delegation, use the focusin event to determine when the select box element gains focus. |
| click | A click event will be triggered when a user clicks on the dropdown list. |
| blur | A blur event will be triggered when the dropdown list loses focus. |
| focusout | A focusout event will be triggered when the dropdown list loses focus. Blur and focusout events are closely related, but the focusout event bubbles up the DOM tree and the blur event does not bubble. If you are using a library such as Backbone.js (http://www.backbonejs.org), which uses event delegation, use the focusout event to determine when the select box element loses focus. |

| | |
|---|---|
| change | A change event will be triggered when a user selects a new dropdown list option. |
| mouseenter | A mouseenter event will be triggered when a user's mouse enters an element. jQuery uses both mouseenter and mouseleave to simulate a hover event. |
| mouseleave | A mouseleave event will be triggered when a user's mouse leaves an element. jQuery uses both mouseenter and mouseleave to simulate a hover event. |
| mousedown | A mousedown event will be triggered when a user clicks on the drop down |
| mouseup | A mouseup event will be triggered when a user clicks on the drop down |

## Custom Events

| **Name** | **Description** |
|---|---|
| open | An open event will be triggered when a user opens the dropdown list. |
| close | A close event will be triggered when a user closes the dropdown list. |
| moveDown | A moveDown event will be triggered when a user presses the down arrow key to select a dropdown list option directly beneath the currently selected option. |
| moveUp | A moveUp event will be triggered when a user presses the up arrow key to select a dropdown list option directly above the currently selected option. |
| search | A search event will be triggered when a user does a text search that matches a dropdown list option. Keep in mind that this event will be fired only when a search match is found. |
| enter | An enter event will be triggered when a user presses the enter key while the dropdown list is focused. |
| tab-focus | A tab-focus event will be triggered when a user presses the tab key to focus the dropdown list. |
| tab-blur | A tab-blur event will be triggered when a user presses the tab key to blur the dropdown list. |
| option-click | An option-click event will be triggered when a user clicks a dropdown list option. |
| backspace | A backspace event will be triggered when a user presses the backspace key while the dropdown list is focused. |
| disable | A disable event will be triggered if a dropdown list is disabled. |
| disable-option | A disable-option event will be triggered if a single dropdown list option is disabled. |
| enable | An enable event will be triggered if a dropdown list becomes enabled, or in other words, no longer disabled. |
| enable-option | A enable-option event will be triggered if a single dropdown list option is disabled. |
| destroy | A destroy event will be triggered if a dropdown list is destroyed |
| create | A create event will be triggered if a dropdown list is created. |
| changed | A changed event will be triggered after the original select box change event is fired and the dropdown text is changed. |
| refresh | A refresh event will be triggered after the refresh method is called to recreate the SelectBoxIt dropdown. |

Here is an example of catching a **Default Event** by using the jQuery **bind()** method:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {
  // Uses the jQuery bind method to bind to the focus event on the dropdown list
  $("select#test").bind({
    "focus": function(ev, obj) {
      // Do something when the focus event is triggered
    }
  });

});
```

Here is an example of catching a **Default Event** by using the jQuery **focus()** convenience method:

```
//Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {
  // Uses the jQuery focus convenience method to bind to the focus event on the dropdown li
st
  $("select#test").focus(function() {
    // Do something when the focus event is triggered
  });
});
```

Here is an example of catching a **Custom Event** by using the jQuery **bind()** convenience method:

```
//Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {
  // Uses the jQuery bind method to bind to the custom open event on the dropdown list
  $("select#test").bind({
    "open": function() {
      //Do something when the open event is triggered
    }
  });
});
```

**Note:** If you don't want to have to explicitly list the **id** or **classname** attributes of the select box you are listening to events on, you can use the pseudo selector that SelectBoxIt provides you.

SelectBoxIt provides **$(":selectBox-selectBoxIt")**, a **custom jQuery pseudo selector**, that returns all select box elements on the page that are using the SelectBoxIt plugin.

# Method API

## Methods

| Name | Parameters | Description |
|---|---|---|
| open | Function callback | Opens the dropdown options list. |
| close | Function callback | Closes the dropdown options list. |
| moveDown | Function callback | Selects the dropdown option directly beneath the currently selected option. |
| moveUp | Function | Selects the dropdown option directly above the currently selected |

| | | |
|---|---|---|
| | callback | option. |
| search | String searchTerm | Selects the dropdown option that most closely matches the text passed into the method. If a pattern match is found, the dropdown text value changes. If a pattern match is not found, then the dropdown text value does not change. |
| setOption | String key, String value | Sets a single plugin option. |
| setOptions | Object newOptions | Sets or adds new plugin option settings. |
| disable | Function callback | Disables the dropdown/select box. |
| disableOption | Number index, Function callback | Disables the dropdown/select box option |
| enable | Function callback | Enables the dropdown/select box. |
| enableOption | Number index, Function callback | Enables the dropdown/select box option |
| destroy | Function callback | Removes the dropdown from the DOM and makes the original select box element visible. |
| wait | Number time, Function callback | Delays execution of the callback function by the amount of time (milleseconds) specified by the time parameter. |
| refresh | Function callback | Rebuilds the dropdown. Useful for dynamic content. |
| selectOption | (Number index or String value), Function callback | The passed value can either be a number or a string. If the value is a number, then SelectBoxIt will select the select box option by index. If the value is a string, then SelectBoxIt will select the select box option that contains the string value. |
| add | data (JSON, Array, Object, or HTML string), Function callback | Adds drop down options using JSON data, an array, a single object, or valid HTML string |
| remove | Number or Array indexes, Function callback | Removes drop down list options using an index or array of indexes |

The **Method API** allows you to programmatically interact with the dropdown list after it is created. All methods can be called individually or chained.

Here is an example of **chaining** (calling multiple **SelectBoxIt** methods one after another):

```
  //Executes your code when the DOM is ready.  Acts the same as $(document).ready().
  $(function() {
    // Note: This code assumes you have already called the selectBoxIt() method somewhere els
e in your code
    // Retrieves all of the SelectBoxIt methods
    var selectBox = $("select#test").data("selectBox-selectBoxIt");

    // Chaining
    selectBox.open().close().moveDown().disable();
  });
```

Here is an example of **individual method calls** :

```
  //Executes your code when the DOM is ready.  Acts the same as $(document).ready().
  $(function() {
    // Note: This code assumes you have already called the selectBoxIt() method somewhere els
e in your code
    // Retrieves all of the SelectBoxIt methods
    var selectBox = $("select#test").data("selectBox-selectBoxIt");

    // Individual calls
    selectBox.open();

    selectBox.close();

    selectBox.moveDown();

    selectBox.disable();

  });
```

If you want to provide a delay (in milleseconds) before your methods are called, use the **wait()** method. Here is an example of the **wait()** method.

```
  //Executes your code when the DOM is ready.  Acts the same as $(document).ready().
  $(function() {

    // Note: This code assumes you have already called the selectBoxIt() method somewhere els
e in your code
    // Retrieves all of the SelectBoxIt methods
    var select = $("select#test").data("selectBox-selectBoxIt");

    // Opens the dropdown list for one second before closing the dropdown list
    select.wait(1000, select.open).wait(1000, select.close);

  });
```

**Note:** You can pass a callback function to all of the methods. Inside of the callback function, the **this** keyword refers to the plugin object, which allows you to call another plugin method like so:

```
// Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  // Note: This code assumes you have already called the selectBoxIt() method somewhere els
e in your code
  // Retrieves all of the SelectBoxIt methods
  var selectBox = $("select#test").data("selectBox-selectBoxIt");

  // Calls the selectBoxIt open() method
  selectBox.open(function() {

    // The 'this' keyword references the selectBoxIt object
    this.moveDown();
  });
});
```

## Mobile Browsers

Selecting the dropdown will trigger the default **wheel** interface for all mobile devices. This behavior improves usability for users.

By default, SelectBoxIt uses a small mobile detection script that will catch most popular mobile browsers.

If you would like to customize which mobile browsers you would like to support, feel free to create your own mobile detection function by setting the **isMobile** option. Below is an example of setting the **isMobile** option to only support iPhones, iPods, and iPads:

```
//Executes your code when the DOM is ready.  Acts the same as $(document).ready().
$(function() {

  // Calls the selectBoxIt method on your HTML select box
  $("select#test").selectBoxIt({ isMobile: function() {

    // Adapted from http://www.detectmobilebrowsers.com
    var ua = navigator.userAgent || navigator.vendor || window.opera;

    // Checks for iOs mobile devices
    return (/iPhone|iPod|iPad/).test(ua);

  }});

});
```

## Contributing

Take care to maintain the existing coding style. Add **Jasmine** unit tests for any new or changed functionality. Lint and test your code using **Grunt**.

To set up the SelectBoxIt grunt/node.js dependencies, first make sure you have nodejs (http://nodejs.org/) and PhantomJS (http://phantomjs.org/) installed.

Next, navigate to within the **jquery.selectBoxIt.js** folder and type `npm install` (this should install grunt and a few other node.js libraries).

**Note:** If you are on Windows, remember you need to run the grunt command using `grunt.cmd`. Also, if you have trouble getting the Jasmine Unit Tests to work with the latest version of

PhantomJS, install PhantomJS 1.3 (I have found that this works).

After you have verified your code, send a pull request to the **SelectBoxIt dev branch**. After you send a pull request, you will hear back from me shortly after I review your code.

You'll find source code in the **src** subdirectory!

## Extending

If you find that you need a feature that SelectBoxIt does not currently support, either let me know via the Github issue tracker (https://github.com/gfranko/jquery.selectBoxIt.js/issues), or fork the project (https://github.com/gfranko/jquery.selectBoxIt.js) and and easily extend SelectBoxIt to create your own widget!

**Note:** Remember that you need to include **jQuery**, the **jQueryUI Widget Factory**, and **SelectBoxIt** before you include your new plugin file, since your plugin will depend on SelectBoxIt and all of its dependencies.

Here is an example of **extending** SelectBoxIt

```
// Plugin setup
(function ($) {

  // Declaring a new jQueryUI Widget that extends SelectBoxIt
  $.widget('a.newPlugin', $.selectBox.selectBoxIt, {

    // Changing SelectBoxIt's default showEffect from 'none' to 'slide'
    options: {

      showEffect: "slide",

    },

      // Overwriting the SelectBoxIt open method
      open: function() {

        // Calling the default SelectBoxIt open method
        $.selectBox.selectBoxIt.prototype.open.call(this);

        // Adding new logic
        console.log("Just opened my new plugin!");

      }

    });

    // Then call your new plugin like this
    var selectBox = $("select#test").newPlugin().data("newPlugin");

  }(jQuery));
```

## Donation

If you would like to support the SelectBoxIt project, please consider sending a donation to Greg Franko (the project maintainer). All donations (small or large) are appreciated and help the continued development of the project. Gratipa