

# jQuery Plugins

Expanding the functionality of jQuery

---

**Telerik Software Academy**

Learning & Development Team

<http://academy.telerik.com>

- ◆ jQuery Plugins
  - ◆ jQueryUI
- ◆ Creating Custom Plugins
- ◆ Implementing chaining
- ◆ Plugins Options

# jQuery Plugins

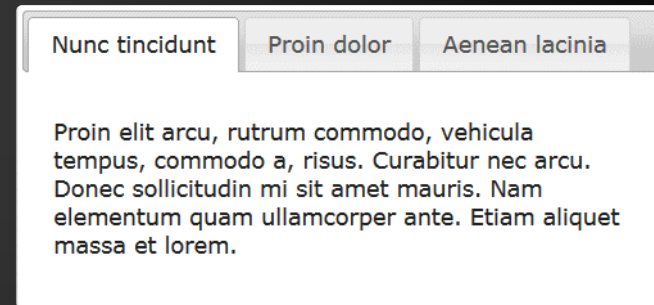
- ◆ A Plugin is just a method that extends the jQuery objects prototype
  - ◆ Enabling all jQuery objects to use this method
- ◆ Once a plugin is imported, it is used as regular jQuery method
  - ◆ Like `addClass()`, `fadeOut()` and `hide()`

- ◆ jQuery has many ready-to-use plugins
  - ◆ A library jQueryUI for UI controls

- ◆ Plugins for UI

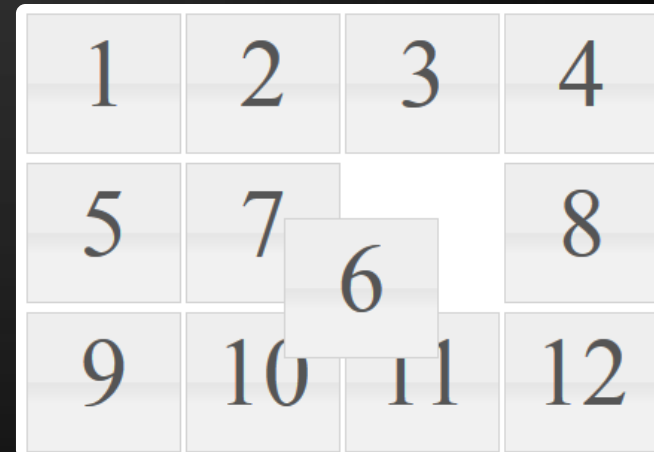
- ◆ Tabs

```
$("#tabs-holder").tabs();
```



- ◆ Arrangeable elements

```
$("#grid" ).sortable();
```



# jQuery Plugins

## Live Demo

# Creating Custom jQuery Plugins

# Creating jQuery Plugins

- ◆ jQuery has an easy way to extend the initial functionality with plugins
  - ◆ Just create a regular JavaScript method and attach it to `jQuery.fn` object

```
(function($){  
    $.fn.zoom = function(){  
        var $this = $(this);  
        $this.on("mouseover", function() {  
            //zoom in element  
        });  
        $this.on("mouseout", function() {  
            //zoom out element  
        });  
    }  
})(jQuery);
```

```
$(".image").zoom();
```



# Zoom Plugin

## Live Demo

- ◆ Chaining is pretty good and useful feature
  - ◆ Better to be implemented in our plugins
  - ◆ Done easy, just return `this` at the end of the plugin function

```
(function($){  
  $.fn.zoom = function(){  
    var $this = $(this);  
    $this.on('mouseover', function() {...});  
    $this.on('mouseout', function() {...});  
    return $this;  
  }  
}(jQuery));  
  
$('.image')  
  .zoom()  
  .addClass('zoom');
```

# Chaining in Plugins

Live Demo

# Plugins with Options

Creating plugins more customizable

- ◆ Plugins can also be provided with options
  - ◆ Just pass regular function parameters

```
(function($){  
    //zoom with size percents  
    $.fn.zoom = function(size){  
        var $this = $(this);  
        $this.on('mouseover', function() {...});  
        $this.on('mouseout', function() {...});  
        return $this;  
    }  
})(jQuery));
```

# Plugins with Options

Live Demo

- ◆ Yet the options sometimes become too many
  - ◆ When too many arguments, just use a options object
  - ◆ Give the options as a JSON-like object

# Plugins with Options Object

Live Demo



Questions?



# 1. Create a jQuery plugin for creating dropdown list

- ♦ By given the following:

```
<select id="dropdown">
  <option value="1">One</option>
  <option value="2">Two</option>
</select>
```

```
$('#dropdown').dropdown()
```

- ♦ Produces the following HTML:

```
<select id="dropdown" style="display: none">...</select>
<div class="dropdown-list-container">
  <ul class="dropdown-list-options">
    <li class="dropdown-list-option" data-value="0">One</li>
    ...
  </ul>
</div>
```

- ♦ And make it work as SELECT node
  - ♦ Selecting an one of the generated LI nodes, selects the corresponding OPTION node
    - ♦ So \$('#dropdown:selected') works

## 2. \*Create a jQuery plugin for fading in/fading out message box

- ♦ Creates a message box

```
var msgBox = $('#message-box').messageBox();
```

- ♦ Show a success/error message in the box
  - ♦ Showing is done by setting the opacity of the message from 0 to 1 in an interval of 1 second
  - ♦ The message disappears after 3 seconds

```
msgBox.success('Success message');  
msgBox.error('Error message');
```