

jQuery: what we can do without plugins

As a jQuery developer I always face the same question over and over again: do I need a plugin for a specific project? Well, it depends a lot on the requirements of the project in question. To put it simple: how much code and how many routines should I use to create a particular effect? This is a matter of time: since we're always running out of time when we have to deal with project deadlines, sometimes is preferable to use a plugin to achieve a certain effect instead of doing all the heavy-lifting by ourselves. Other times, when the project requires something that cannot be easily found in an existing plugin, we have to write our own implementation. Let's see the latter case.

Project requirements

A client wants to turn a set of images scattered through the content of a page into a lightbox gallery. He wants something really simple: a single modal box with an underlying semitransparent overlay. The gallery must have a pagination system. Once a pagination link is clicked, it shows the related image with a fading effect.

When you click on an image in the document, the gallery and the overlay appears. The first image in the gallery must always be the current image, followed by the other images which must be initially hidden.

The markup

The HTML code used in the document is as follows:

```
<a href="path/to/full-image.jpg" class="gallery">
    
</a>
```

With jQuery we need to create the following markup to be appended to the body element:

```
<div id="overlay"></div>
<div id="gallery">
    <div id="gallery-wrapper">
        <img id="first-image" />
    </div>
    <div id="gallery-nav"></div>
</div>
```

The CSS code

We'll use absolute positioning to display the gallery and its overlay. The total height of the overlay will be later set by jQuery:

```
#overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    background: rgba(0, 0, 0, 0.7);
```

```
        display: none;
    }

    #gallery {
        width: 500px;
        height: 400px;
        border: 5px solid #eee;
        position: absolute;
        top: 50%;
        left: 50%;
        margin: -210px 0 0 -260px;
        background: #fff;
        display: none;
    }

    #gallery-wrapper {
        width: 400px;
        height: 300px;
        position: absolute;
        top: 50%;
        left: 50%;
        margin: -150px 0 0 -200px;
    }

    #gallery-nav {
        height: 2em;
        width: 100%;
        position: absolute;
        bottom: 0;
        left: 0;
        background: #f6f6f6;
        border-top: 1px solid #eee;
        text-align: center;
    }
```

```
#gallery-nav a {
    display: inline-block;
    height: 1.6em;
    width: 1.6em;
    line-height: 1.6;
    margin: 0.2em 0.5em 0.2em 0;
    text-align: center;
    text-decoration: none;
    color: #333;
    background: #ddd;
}

#gallery-wrapper img {
    max-width: 100%;
    display: none;
}

#gallery-wrapper > img:first-child {
    display: block;
}
```

The jQuery code

With jQuery we need to:

1. create an helper method to get the path and the alternate text of the current image
2. create an helper method to insert the other images into the gallery when a user opens the lightbox
3. create a method to generate the base HTML structure of the gallery
4. create a method to generate the HTML links for the gallery navigation with a custom data attribute which points to a specific image's index in

the DOM

5. create a method to populate and show the gallery
6. create a method that handles the gallery's navigation
7. create a method that closes the gallery and restores the initial DOM structure of the lightbox.

As you can imagine, we'll use an object for our purposes:

```
var Gallery = {
  utils: {
    getCurrentImage: function(element) {
      var $img = $('img', element),
          src = $img.attr('src'),
          alt = $img.attr('alt');
      return {
        src: src,
        alt: alt
      }
    },

    insertOtherImages: function(element) {
      var parent = element.parents('#content');
      var links =
parent.find('a.gallery').not(element);
      links.each(function() {
        var $link = $(this);
        var image =
Gallery.utils.getCurrentImage($link);
        $('<img/>').attr({
          src: image.src,
          alt: image.alt
        }).appendTo('#gallery-wrapper');
```

```

        });

    }

},

fn: {
    createGallery: function() {
        $('<div id="overlay"/>').
            css('height', $(document).height()).
            appendTo('body');
        $('<div
id="gallery"/>').insertAfter('#overlay');
        $('#gallery').html('<div id="gallery-
wrapper"></div><div id="gallery-nav"></div>');
        $('<img id="first-
image"/>').appendTo('#gallery-wrapper');
    },
    populateGallery: function() {
        $('a.gallery').each(function(i) {
            $('<a/>').attr({
                href: '#',
                'data-image': i
            }).text(i + 1).appendTo('#gallery-
nav');

        });
    },
    showGallery: function() {

```

```
        $('a.gallery').on('click', function(e) {
            e.preventDefault();
            var $a = $(this);
            var image =
Gallery.utils.getCurrentImage($a);

            $('#first-image').attr({
                src: image.src,
                alt: image.alt
            });

            Gallery.utils.insertOtherImages($a);

            $('#overlay').fadeIn(400, function()
{
                $('#gallery').fadeIn(400);

            });

        });

    },

    navigate: function() {
        $('a', '#gallery-nav').live('click',
function(e) {
            e.preventDefault();
            var index = $(this).data('image');
            $('img', '#gallery-
wrapper').eq(index).fadeIn(600).siblings('img').hide(
);

        });
    }
});
```

```

    },

    closeGallery: function() {
        $('#overlay').on('click', function() {
            var overlay = $(this);
            $('#gallery').fadeOut(400, function()
{
                overlay.fadeOut(400);
                $('#first-image').show().attr({
                    src: '',
                    alt: ''
                }).siblings('img').remove();

            });

        });

    }

},

init: function() {
    this.fn.createGallery();
    this.fn.populateGallery();
    this.fn.showGallery();
    this.fn.navigate();
    this.fn.closeGallery();

}

};

```

Then you can initialize our object as follows:

```
$(function() {  
    Gallery.init();  
});
```

[view-example url="http://jsfiddle.net/gabrieleromanato/Bum3z/"]