

jQuery: expandable portfolio

Most of my spare time after work is spent helping other developers to solve their coding problems. An interesting problem was brought to my attention a couple of days ago. Basically, you have a portfolio laid out as a multiline grid. When you click on an item, a row below the current one slides down to reveal the item's details. The HTML structure behind such a portfolio is quite simple, as well as the CSS code. The problems lie in the jQuery code.

In the following HTML structure, we have two rows with three items on each row. After the end of each row there's an empty element which will be used to contain the item's details taken from an hidden unordered list:

```
<div id="portfolio">
  <div class="row">
    <div class="item" data-rel="0">1</div>
    <div class="item" data-rel="1">2</div>
    <div class="item" data-rel="2">3</div>
  </div>
  <div class="details"></div>
  <div class="row">
    <div class="item" data-rel="3">4</div>
    <div class="item" data-rel="4">5</div>
    <div class="item" data-rel="5">6</div>
  </div>
  <div class="details"></div>
  <ul id="portfolio-details">
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
  </ul>
</div>
```

```
        <li>...</li>
    </ul>
</div>
```

The CSS code is pretty simple:

```
#portfolio {
    margin: 2em 0;
    width: 100%;
}

#portfolio div.row {
    margin-bottom: 1em;
    height: 15em;
}

#portfolio div.details {
    display: none;
    padding: 5%;
    margin: 0.5em;
    background: #f5f5f4;
}

#portfolio div.row div.item {
    width: 30%;
    float: left;
    height: 100%;
    margin: 0 1.5%;
    background: #eee;
    text-align: center;
    font-size: 4em;
    line-height: 4;
```

```
        cursor: pointer;
    }

    #portfolio #portfolio-details {
        position: absolute;
        top: -9999px;
    }
```

We've set a `data-rel` attribute on each portfolio item which contains the index of each list item within the unordered list. When we click on an item, we hide all the containing blocks and we empty the contents of the wrapper which follows the current row.

Done this, we copy the HTML contents of the target list item into the current containing block and we show it. At the same time we make the entire page scroll to properly show the newly inserted contents:

```
var Portfolio = {
    Elements: {

        items: $('div.item', '#portfolio'),
        details: $('#portfolio-details',
            '#portfolio')

    },

    fn: {

        select: function() {

            Portfolio.Elements.items.each(function()
```

```

        var $item = $(this);
        var $row = $item.parent();
        var $details =
$row.next('div.details');
        var $li = $('li',
Portfolio.Elements.details);

        $item.click(function() {

            $('div.details').hide();
            $details.empty();
            var $html =
$li.eq($item.data('rel')).html();

            $details.html($html).slideDown(600);
            $('html, body').animate({
                scrollTop: 0
            }, 0).animate({
                scrollTop:
$details.offset().top
            }, 300);

        });

    });
}

```

```
    }  
  
};  
  
Portfolio.fn.select();
```

You can see the demo on [this page](#).