```
    <td colspan="2">Total</td>
    <td class="total-cell"><%= number_to_currency(cart.total_price) %></td>
  </tr>
</table>

<%= button_to "Empty cart", :action => :empty_cart %>
<% end %>
```

Although this works, the user interface is somewhat brutal: the whole sidebar redraws on the transition between a cart that's empty and a cart with something in it. So let's not use this code. Instead, let's smooth it out a little.

The Script.aculo.us effects library contains a number of nice transitions that make elements appear. Let's use *blind_down*, which will smoothly reveal the cart, sliding the rest of the sidebar down to make room.

Not surprisingly, we'll use our existing .rjs template to call the effect. Because the add_to_cart template is invoked only when we add something to the cart, then we know that we have to reveal the cart in the sidebar whenever there is exactly one item in the cart (because that means that previously the cart was empty and hence hidden). And, because the cart should be visible before we start the highlight effect, we'll add the code to reveal the cart before the code that triggers the highlight.

The template now looks like this.

Download depot_n/app/views/store/add_to_cart.rjs

```
page.replace_html("cart", :partial => "cart", :object => @cart)

page[:cart].visual_effect :blind_down if @cart.total_items == 1

page[:current_item].visual_effect :highlight,
                                  :startcolor => "#88ff88",
                                  :endcolor => "#114411"
```

This won't yet work, because we don't have a total_items method in our cart model.

Download depot_n/app/models/cart.rb

```
def total_items
  @items.sum { |item| item.quantity }
end
```

We have to arrange to hide the cart when it's empty. There are two basic ways of doing this. One, illustrated by the code at the start of this section, is not to generate any HTML at all. Unfortunately, if we do that, then when we add something to the cart and suddenly create the cart HTML, we see a flicker in the browser as the cart is first displayed and then hidden and slowly revealed by the blind_down effect.