

In the `_cart_item.rhtml` partial, we then check to see whether the item we're rendering is the one that just changed. If so, we tag it with an id of `current_item`.

[Download](#) `depot_m/app/views/store/_cart_item.rhtml`

```
<% if cart_item == @current_item %>
  <tr id="current_item">
<% else %>
  <tr>
<% end %>
  <td><%= cart_item.quantity %>&times;</td>
  <td><%= h(cart_item.title) %></td>
  <td class="item-price"><%= number_to_currency(cart_item.price) %></td>
</tr>
```

As a result of these three minor changes, the `<tr>` element of the most recently changed item in the cart will be tagged with `id="current_item"`. Now we just need to tell the JavaScript to invoke the highlight effect on that item. We do this in the existing `add_to_cart.js` template, adding a call to the `visual_effect` method.

[Download](#) `depot_m/app/views/store/add_to_cart.js`

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

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

See how we identified the browser element that we wanted to apply the effect to by passing `:current_item` to the page? We then asked for the *highlight* visual effect and overrode the default yellow/white transition with colors that work better with our design. Click to add an item to the cart, and you'll see the changed item in the cart glow a light green before fading back to merge with the background.

9.4 Iteration D4: Hide an Empty Cart

One last request from the customer: right now, even carts with nothing in them are still displayed in the sidebar. Can we arrange for the cart to appear only when it has some content? But of course!

In fact, we have a number of options. The simplest is probably to include the HTML for the cart only if the cart has something in it. We can do this totally within the `_cart` partial.

```
<% unless cart.items.empty? %>
<div class="cart-title">Your Cart</div>
<table>
  <%= render(:partial => "cart_item", :collection => cart.items) %>

  <tr class="total-line">
```