

---

# Four Ways to Customize a Cart Item

---

by Tim Bezhashvyly

 Openstream Internet Solutions

---

# About Me

---

## Tim Bezhashvyly

Certified Magento Developer

12 years of PHP

3 years of Magento



boy-named-sue



tim-openstream





# What the hell is he talking about?

---

Shopping Cart

Proceed to Checkout

	Product Name		Unit Price	Qty	Subtotal	
	<a href="#">Test</a>  HOW TO ADD SOMETHING HERE?	<a href="#">Edit</a>	\$7.00	<input type="text" value="1"/>	\$7.00	

Continue Shopping

Clear Shopping Cart

Update Shopping Cart

---

# Method 1

---

---

# Method 1

`Mage_Sales_Model_Quote_Item::addMessage()`

---

# Register an Observer

---

```
<frontend>
  <events>
    <sales_quote_collect_totals_before>
      <observers>
        <hackathon_presentation>
          <type>singleton</type>
          <class>Hackathon_Presentation_Model_Observer</class>
          <method>salesQuoteAddressCollectTotalsBefore</method>
        </hackathon_presentation>
      </observers>
    </sales_quote_collect_totals_before>
  </events>
</frontend>
```

---

# Model / Observer.php

---

```
$quote = $observer->getEvent()->getQuote();
```

```
$quote_items = $quote->getItemsCollection();
```

```
foreach ($quote_items as $item) {
```

```
    $item->addMessage('Woohoo, this item\'s ID is ' . $item->getProductId());
```


```
}
```

---

# Result

---

## Shopping Cart

	Product Name
	<a href="#">Test</a> <b>* Woohoo, this item's ID is 1</b>
<a href="#">Continue Shopping</a>	



# Pros'n'cons

---

+ can be used with observer

---

# Pros'n'cons

---

- + can be used with observer
- + add as much messages as you want

# Model / Observer.php

---

```
$quote = $observer->getEvent()->getQuote();
```

```
$quote_items = $quote->getItemsCollection();
```

```
foreach ($quote_items as $item) {
```

```
    $item->addMessage('Woohoo, this item\'s ID is ' . $item->getProductId());
```

```
    $item->addMessage('and the SKU is ' . $item->getSku() . '');
```


```
}
```

---

# Result

---

## Shopping Cart

	Product Name
	<p><a href="#">Test</a></p> <p>* Woohoo, this item's ID is 1</p> <p>* and the SKU is "test".</p>
<a href="#">Continue Shopping</a>	

# Pros'n'cons

---

- + can be used with observer
  - + add as much messages as you want
  - styling
-

# Pros'n'cons

---

- + can be used with observer
  - + add as much messages as you want
  - styling
  - ± not saved into database
-

# Pros'n'cons

---

- + can be used with observer
  - + add as much messages as you want
  - styling
  - ± not saved into database
  - ± nor even in a session
-

---

# Method 2

---



---

# Method 2

Custom Option

---

# Model / Observer.php

---

```
$quote = $observer->getEvent()->getQuote();
$quote_items = $quote->getItemsCollection();


foreach ($quote_items as $item) {
    $additionalOptions = array(array(
        'code' => 'my_code',
        'label' => 'This text is displayed through additional options',
        'value' => 'ID is ' . $item->getProductId() . ' and SKU is ' . $item->getSku()
    ));
    $item->addOption(array(
        'code' => 'additional_options',
        'value' => serialize($additionalOptions),
    ));
}
```

---

# Result

---

## Shopping Cart

	Product Name
	<p><u>Test</u></p> <p><i>This text is displayed through additional options</i></p> <p>ID is 1 and SKU is test.</p>
<a href="#">Continue Shopping</a>	

# Pros'n'cons

---

+ can be used with observer

---

# Pros'n'cons

---

- + can be used with observer
  - + can be easily saved to order
-

# Pros'n'cons

---

- + can be used with observer
  - + can be easily saved to order
  - + styling
-

# Pros'n'cons

---

- + can be used with observer
  - + can be easily saved to order
  - + styling
  - styling
-

# Pros'n'cons

---

- + can be used with observer
  - + can be easily saved to order
  - + styling
  - styling
  - option can already hold some information
-



---

# Method 3

---

---

# Method 3

Override Default Renderer

# checkout.xml

---

```
<checkout_cart_index translate="label">
  <label>Shopping Cart</label>
  <remove name="right"/>
  <remove name="left"/>
  <!-- Mage_Checkout -->
  <reference name="root">
    <action method="setTemplate"><template>page/1column.phtml</template></action>
  </reference>
  <reference name="content">
    <block type="checkout/cart" name="checkout.cart">
      <action method="setCartTemplate"><value>checkout/cart.phtml</value></action>
      <action method="setEmptyTemplate"><value>checkout/cart/noItems.phtml</value></action>
      <action method="chooseTemplate"/>
      <action method="addItemRender"><type>simple</type><block>checkout/cart_item_renderer</block><template>checkout/cart
/item/default.phtml</template></action>
      <action method="addItemRender"><type>grouped</type><block>checkout/cart_item_renderer_grouped</block><template
>checkout/cart/item/default.phtml</template></action>
      <action method="addItemRender"><type>configurable</type><block>checkout/cart_item_renderer_configurable</block
><template>checkout/cart/item/default.phtml</template></action>
```

# presentation.xml

---


```
<?xml version="1.0"?>
<layout>
  <checkout_cart_index>
    <reference name="checkout.cart">
      <action method="addItemRender">
        <type>simple</type>
        <block>checkout/cart_item_renderer</block>
        <template>hackathon/presentation/checkout/cart/item/default.phtml
        </template>
      </action>
    </reference>
  </checkout_cart_index>
</layout>
```

---

# Result

---

## Shopping Cart

	Product Name
	<a href="#">Test</a> Oops, we've overwritten the complete template (ID: 1, SKU: test)
<a href="#">Continue Shopping</a>	

# Pros'n'cons

---

+ full control of styling

---

# Pros'n'cons

---

- + full control of styling
- + you can do within a theme

# Pros'n'cons

---

- + full control of styling
  - + you can do within a theme
  - you have to overwrite a template
-



# Pros'n'cons

---

- + full control of styling
  - + you can do within a theme
  - you have to overwrite a template
  - hard to save a value to db
-

# Pros'n'cons

---

- + full control of styling
  - + you can do within a theme
  - you have to overwrite a template
  - hard to save a value to db
  - ± different renderers for each product type
-

---

# Method 4

---

---

# Method 4

`additional.product.info`

---

# checkout.xml

---

```
        <block type="checkout/cart_totals" name="checkout.cart_totals" as="totals" t
    </block>
</reference>
<block type="core/text_list" name="additional.product.info" translate="label">
    <label>Additional Product Info</label>
</block>
</checkout_cart_index>
```

# presentation.xml

---

```
<reference name="additional.product.info">  
  <block type="hackathon_presentation/additional_product_info" name="..."/>  
</reference>
```

# Info.php

---

```
class Hackathon_Presentation_Block_Additional_Product_Info extends
Mage_Core_Block_Abstract
{
    protected function _toHtml()
    {
        $item = $this->getParentBlock()->getItem();


        return 'This is Additional Info Block (ID: ' . $item->getId() . 'SKU: ' . $item-
>getSku() . ')';
    }
}
```

---

# Result

---

## Shopping Cart

	Product Name
	<a href="#">Test</a> This is Additional Info Block (ID: 25SKU: test)
<a href="#">Continue Shopping</a>	



# Pros'n'cons

---

+ full control of styling

---

# Pros'n'cons


---

- + full control of styling
- hard to save value to database

# Result

---

## Shopping Cart

	Product Name
	<p><u><a href="#">Test</a></u></p> <p><i>This text is displayed through additional options</i></p> <p>ID is 1 and SKU is test.</p> <p><b>* Woohoo, this item's ID is 1</b></p> <p><b>* and the SKU is "test".</b></p> <p>Oops, we've overwritten the complete template (ID: 1, SKU: test)</p> <p>This is Additional Info Block (ID: 25SKU: test)</p>
<a href="#">Continue Shopping</a>	

---

# Thank You!

<https://github.com/tim-bezhashvyly/magento-customize-cart-item>

---

---

# Questions?

---