# Four Ways to Customize a Cart Item

- by Tim Bezhashvyly
- Openstream Internet Solutions

#### **About Me**

# Tim Bezhashvyly

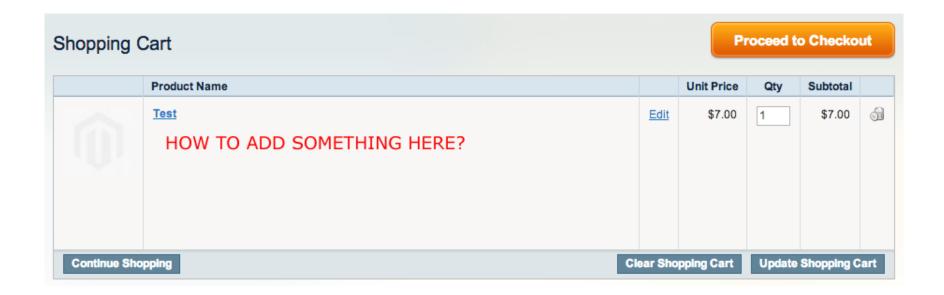
Certified Magento Developer

12 years of PHP3 years of Magento





# What the hell is he talking about?



# Method 1

# Method 1

Mage\_Sales\_Model\_Quote\_Item::addMessage()

# Register an Observer

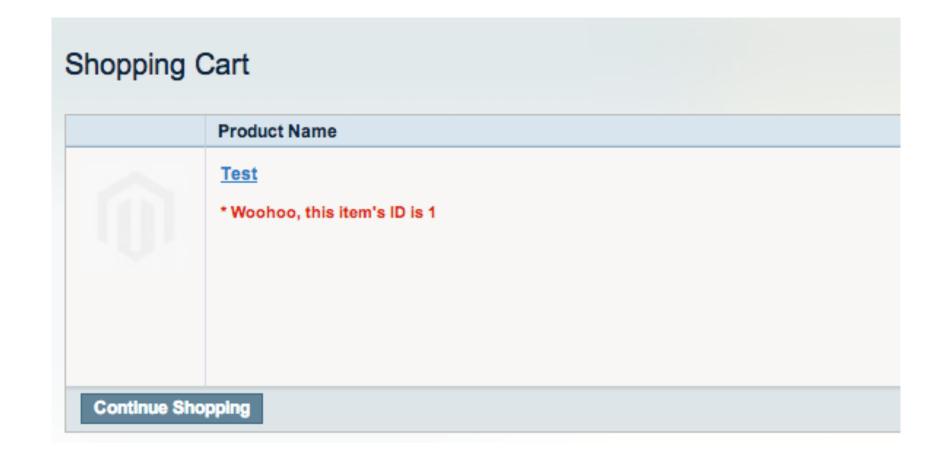
```
<frontend>
  <events>
    <sales_quote_collect_totals_before>
      <observers>
         <hackathon_presentation>
           <type>singleton</type>
           <class>Hackathon_Presentation_Model_Observer</class>
           <method>salesQuoteAddressCollectTotalsBefore</method>
         </hackathon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



+ can be used with observer

- + 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 Test \* Woohoo, this item's ID is 1 \* and the SKU is "test". Continue Shopping

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

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

- + 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**

# Test This text is displayed through additional options ID is 1 and SKU is test. Continue Shopping

+ can be used with observer

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

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

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

- + 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/lcolumn.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**

# Test Oops, we've overwritten the complete template (ID: 1, SKU: test) Continue Shopping

+ full control of styling

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

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

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

- + 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

# 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**

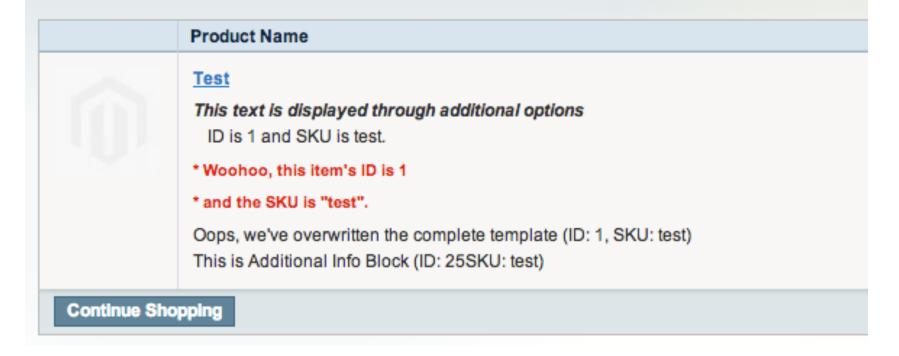
# Test This is Additional Info Block (ID: 25SKU: test) Continue Shopping

+ full control of styling

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

#### Result

#### **Shopping Cart**



# Thank You!

# Questions?