

Installation Guide for Vintage Themes

Bold Discounts requires a Liquid code install on vintage themes in order to run correctly on your store. This ensures the discounts appear correctly on your product pages, cart page, and in the checkout.

Before adding the Liquid code, it's best to create and work on a [backup copy of your theme](#). This ensures you can test the install without affecting your live store. Once you've ensured everything is working as it should, you can publish your backup copy.

Step 1: Add Snippet Files

1. From the Shopify admin, click **Online Store > Themes**.
2. **Duplicate** the theme you would like the installation completed on.
3. Click **Edit code** on the duplicated theme.
4. Under **Snippets**, find the **bold-pr.liquid** file.
5. If this file doesn't exist:
 - a. Click **Add a new snippet**.
 - b. Enter the correct **snippet name (bold-pr.liquid)**.
 - c. Click **Create snippet**.
 - d. Copy and paste the following code into the **bold-pr.liquid** file:

```
<script id="bold-platform-data"
type="application/json">
{
  "shop": {
    "domain": "{{ shop.domain }}",
    "permanent_domain": "{{ shop.permanent_domain }}",
    "url": "{{ shop.url }}",
    "secure_url": "{{ shop.secure_url }}",
    "money_format": {{ shop.money_format | json }},
    "currency": {{ shop.currency | json }}
  },
  "customer": {
    "id": {{ customer.id | json }},
    "tags": {{ customer.tags | json }}
  }
}
```

```

    },
    "cart": {{ cart | json }},
    "line_item_products": {{ cart | map: "items" | map:
"product" | json }},
    "template": "{{ template | split: "." | first }}",
    "product": {{ product | json }},
    "collection": {{ collection.products | json }}
}
</script>
{{
'https://static.boldcommerce.com/bold-platform/sf/pr.js
' | script_tag }}
<style>
    .money[data-product-id], .money[data-product-handle],
.money[data-variant-id], .money[data-line-index],
.money[data-cart-total] {
        animation: moneyAnimation 0s 2s forwards;
visibility: hidden;
    }
    @keyframes moneyAnimation { to { visibility: visible;
} }
    .shappify_qb_grid {
        width: auto;
        background: #fff;
        margin: 8px 0;
    }
    .shappify_qb_grid, .shappify_qb_grid td,
.shappify_qb_grid th {
        text-align: left;
        padding: 5px;
        border: 1px solid #ccc;
        border-collapse: collapse;
    }

```

</style>

```
bold-pr.liquid •
bold-pr.liquid Older versions
1 <script id="bold-platform-data" type="application/json">
2 {
3   "shop": {
4     "domain": "{{ shop.domain }}",
5     "permanent_domain": "{{ shop.permanent_domain }}",
6     "url": "{{ shop.url }}",
7     "secure_url": "{{ shop.secure_url }}",
8     "money_format": {{ shop.money_format | json }},
9     "currency": {{ shop.currency | json }}
10  },
11  "customer": {
12    "id": {{ customer.id | json }},
13    "tags": {{ customer.tags | json }}
14  },
15  "cart": {{ cart | json }},
16  "line_item_products": {{ cart | map: "items" | map: "product" | json }},
17  "template": "{{ template | split: "." | first }}",
18  "product": {{ product | json }},
19  "collection": {{ collection.products | json }}
20 }
21 </script>
22
23 {{ 'https://static.boldcommerce.com/bold-platform/sf/pr.js' | script_tag }}
24
25 <style>
26   .money[data-product-id], .money[data-product-handle], .money[data-variant-id], .money[data-line-index], .money[d
27     animation: moneyAnimation 0s 2s forwards; visibility: hidden;
```

e. Click **Save**.

Step 2: Edit theme.liquid

1. Under **Layout**, click **theme.liquid**.
2. Copy and paste the following code below the `{{ content_for_header }}` line:

Unset

```
{%- render 'bold-pr' -%}
```

```
theme.liquid •
theme.liquid Older versions
49  {% if template contains 'customers' %}
50  <!--[if (gt IE 9)!(IE)]><!--><script src="{{ 'shopify_common.js' | shopify_asset_url }}" defer="defer"></scr.
51  <!--[if lte IE 9]><script src="{{ 'shopify_common.js' | shopify_asset_url }}"></script><![endif]-->
52  {% endif %}
53
54  <!--[if (gt IE 9)!(IE)]><!--><script src="{{ 'vendor.js' | asset_url }}" defer="defer"></script><!--![endif]--
55  <!--[if lt IE 9]><script src="{{ 'vendor.js' | asset_url }}"></script><![endif]-->
56
57  {% if template contains 'product' or template contains 'cart' %}
58  <script>
59    window.theme = window.theme || {};
60    theme.moneyFormat = {{ shop.money_format | json }};
61  </script>
62  {% endif %}
63
64  <!--[if (gt IE 9)!(IE)]><!--><script src="{{ 'theme.js' | asset_url }}" defer="defer"></script><!--![endif]--
65  <!--[if lte IE 9]><script src="{{ 'theme.js' | asset_url }}"></script><![endif]-->
66
67  {{ content_for_header }}{% include 'bold-checkout-loader' %}
68
69  {%- render 'bold-pr' -%}
70
71  {%- include 'bold-ro-init' -%}
72
73  {%- include 'bold-product' with product, hide_action: 'header' -%}
74  {{ 'bold-brain-widget.css' | asset_url | stylesheet_tag }}
75
76  {%- render 'bold-common' -%}
77
```

3. Click **Save**.

Step 3: Add the Product Loop Code

Updates need to be done on **all** theme files that pull product information to the storefront. The number and name of these files vary from theme to theme. Generally, three to five files need to be updated.

These control the product information on the product, collection, search, related products, and home pages. It's important that each theme file that pulls this information be edited. This is to ensure the correct information always shows for your customers.

Note: This requires you to go back and edit multiple files within your theme files.

Go through the list of files, one at a time, and then reference your theme's files to see if that particular file exists in your theme. Not all of these files exist in a given theme.

List of theme files to update

- **Templates:** Themes should contain most of these files. If your theme is using "Sections", these files are located within the "Sections" folder as ***-template.liquid** files.

- collection.liquid
- list-collections.liquid
- search.liquid
- **Sections & Snippets:** Any given theme should typically have a few of these files.
 - collection-template.liquid
 - featured-collection.liquid
 - product-full.liquid
 - product-loop.liquid
 - product-recommendations
 - product-slider.liquid
 - related-products.liquid

Not all of these appear in a given file, but the ones that do may appear several times.

Code adjustments required

Your theme's product loop code should look similar to this:

Unset

```
{% for product in collection.products %}
```

```
collection.liquid •
collection.liquid Older versions
10      &ndash; {% assign title_tags = current_tags | join: ', ' %}
11      {{ title_tags }}
12      {% endif %}
13    </hl>
14    {% if collection.description != blank %}
15    <div class="section-header_subtext rte">
16      {{ collection.description }}
17    </div>
18    {% endif %}
19  </div>
20  <div class="grid_item medium-up--one-half medium-up--text-right section-header_item">
21    {% section 'collection-filters' %}
22  </div>
23 </header>
24
25 <div class="grid grid--no-gutters grid--uniform">
26   {% for product in collection.products %}
27
28
29   <div class="grid_item small--one-half medium-up--one-fifth">
30     {% include 'product-card', product: product %}
31   </div>
32   {% else %}
33     {% comment %}
34       Add default products to help with onboarding for collections/all only.
35
36       The onboarding styles and products are only loaded if the
37       store has no products.
38     {% endcomment %}
```

The following code should be inserted directly after each instance:

Unset

```
<script type="application/json"
class="bold-product-json">{{ product | json }}</script>
```

```
collection.liquid •
collection.liquid Older versions
17   {% if collection.description != blank %}
18     <div class="section-header_subtext rte">
19       {{ collection.description }}
20     </div>
21   {% endif %}
22 </div>
23 <div class="grid_item medium-up--one-half medium-up--text-right section-header_item">
24   {% section 'collection-filters' %}
25 </div>
26 </header>
27
28 <div class="grid grid--no-gutters grid--uniform">
29
30   {% for product in collection.products %}
31
32     <script type="application/json" class="bold-product-json">{{ product | json }}</script>
33
34     <div class="grid_item small--one-half medium-up--one-fifth">
35       {% include 'product-card', product: product %}
36     </div>
37   {% else %}
38     {% comment %}
39       Add default products to help with onboarding for collections/all only.
40
```

In some cases, the product loop code may look slightly different:

Unset

```
{% for item in search.results %}
```

search.liquid •

search.liquid [Older versions](#)

```
15
16 <div class="input-group btn">
17   <button type="submit" class="btn btn--narrow">
18     {% include 'icon-arrow-right' %}
19     <span class="icon_fallback-text">{{ 'general.search.submit' | t }}</span>
20   </button>
21 </div>
22 </form>
23 </div>
24 </header>
25
26 {% if search.performed %}
27
28   {% comment %}
29   Find the number of products and pages in the search results
30   {% endcomment %}
31   {% assign product_count = 0 %}
32   {% assign page_count = 0 %}
33
34   {% for item in search.results %}
35
36     {% if item.object_type == 'product' %}
37       {% assign product_count = product_count | plus: 1 %}
38     {% else %}
39       {% assign page_count = page_count | plus: 1 %}
40     {% endif %}
41   {% endfor %}
42
43   {% if product_count > 0 %}
```

We then need to adjust the end of our Bold price code to include the loop's product reference:

Unset

```
<script type="application/json"
class="bold-product-json">{{ item | json }}</script>
```

search.liquid •

search.liquid [Older versions](#)

```
16 <div class="input-group btn">
17   <button type="submit" class="btn btn--narrow">
18     {% include 'icon-arrow-right' %}
19     <span class="icon_fallback-text">{{ 'general.search.submit' | t }}</span>
20   </button>
21 </div>
22 </form>
23 </div>
24 </header>
25
26 {% if search.performed %}
27
28   {% comment %}
29   Find the number of products and pages in the search results
30   {% endcomment %}
31   {% assign product_count = 0 %}
32   {% assign page_count = 0 %}
33
34   {% for item in search.results %}
35
36     <script type="application/json" class="bold-product-json">{{ item | json }}</script>
37
38     {% if item.object_type == 'product' %}
39       {% assign product_count = product_count | plus: 1 %}
40     {% else %}
41       {% assign page_count = page_count | plus: 1 %}
42     {% endif %}
43   {% endfor %}
44
```

Step 4: Update the Product Price & Compare At Price Code

Updates need to be done on **all** theme files that output product prices on the site. The number and name of these files will vary from theme to theme. Generally, three to five files will need to be updated.

These control the price fields on the product, collection, search, related products, and home pages. It's important that each theme file that outputs a price be modified. This is to ensure the correct pricing always shows for your customers.

Note: This will require you to go back and edit multiple files within your theme coding.

Go through the list of files, one at a time, and then reference your theme's files to see if that particular file exists in your theme. Not all of these files will exist in a given theme.

List of theme files to update

- **Templates:** Themes should contain most of these files. If your theme is using **Sections**, these files will be located within the **Sections** folder as ***-template.liquid** files.
 - product.liquid
 - index.liquid
 - collection.liquid
 - search.liquid
- **Sections & Snippets:** Any given theme should typically have a few of these files.
 - featured-product.liquid
 - product-card.liquid
 - product-grid-item.liquid
 - product-form.liquid
 - product-template.liquid
 - related-products.liquid

Code adjustments required:

Your theme's product price code should look similar to this:

Unset

```
{{ product.price | money }}
```

```
product-card.liquid •
product-card.liquid Older versions
25     {% endcomment %}
26     {% if product.price_varies %}
27         {% assign sale_price = product.price | money_without_trailing_zeros %}
28         {{ 'products.product.on_sale_from_html' | t: price: sale_price }}
29     {% else %}
30         <span class="visually-hidden">{{ 'products.product.regular_price' | t }}</span>
31         <s class="product-card_regular-price">{{ product.compare_at_price | money_without_trailing_zeros }}</
32
33         <span class="visually-hidden">{{ 'products.product.sale_price' | t }}</span>
34
35         {{ product.price | money }}
36     {% endif %}
37
38
39     {% else %}
40     {% comment %}
41     Not on sale, but could still have varying prices
42     {% endcomment %}
43     {% if product.price_varies %}
44         {% assign price = product.price | money_without_trailing_zeros %}
45         {{ 'products.product.from_text_html' | t: price: price }}
46     {% else %}
47         <span class="visually-hidden">{{ 'products.product.regular_price' | t }}</span>
48         <span class="money" data-product-id="{{ product.id }}">{{ product.price | money_without_trailing_zeros
49     </span>{% endif %}
50
```

The following code should be placed around each of these instances with the `` tags wrapping this price code:

Unset

```
<span class="money" data-product-id="{{ product.id }}">{{
product.price | money }}</span>
```

product-card.liquid •

product-card.liquid Older versions

```
25     {% endcomment %}
26     {% if product.price_varies %}
27     {% assign sale_price = product.price | money_without_trailing_zeros %}
28     {{ 'products.product.on_sale_from_html' | t: price: sale_price }}
29     {% else %}
30     <span class="visually-hidden">{{ 'products.product.regular_price' | t }}</span>
31     <s class="product-card_regular-price">{{ product.compare_at_price | money_without_trailing_zeros }}</
32
33     <span class="visually-hidden">{{ 'products.product.sale_price' | t }}</span>
34
35     <span class="money" data-product-id="{{ product.id }}">{{ product.price | money }}</span>
36
37     {% endif %}
38
39     {% else %}
40     {% comment %}
41     Not on sale, but could still have varying prices
42     {% endcomment %}
43     {% if product.price_varies %}
44     {% assign price = product.price | money_without_trailing_zeros %}
45     {{ 'products.product.from_text_html' | t: price: price }}
46     {% else %}
47     <span class="visually-hidden">{{ 'products.product.regular_price' | t }}</span>
48     <span class="money" data-product-id="{{ product.id }}">{{ product.price | money_without_trailing_zeros
49
50 </span>{% endif %}
```

Your theme's end compare at price code should look similar to this:

Unset

```
{{ compare_at_price | money }}
```

product-price.liquid Older versions

Wrap

```
61 </div>
62 <div class="price_sale">
63 <dt>
64 <span class="visually-hidden visually-hidden--inline">{{ 'products.product.sale_price' | t }}</span>
65 </dt>
66 <dd>
67 <span class="price-item price-item--sale" data-sale-price>
68 {{ money_price }}
69 </span>
70 </dd>
71 <dt>
72 <span class="visually-hidden visually-hidden--inline">{{ 'products.product.regular_price' | t }}</span>
73 </dt>
74 <dd>
75 <s class="price-item price-item--regular" data-regular-price>
76 {{ compare_at_price | money }}
77 </s>
78 </dd>
79 </div>
80 <div class="price_badges">
81 <span class="price_badge price_badge--sale" aria-hidden="true">
82 <span>{{ 'products.product.on_sale' | t }}</span>
83 </span>
84 <span class="price_badge price_badge--sold-out">
85 <span>{{ 'products.product.sold_out' | t }}</span>
86 </span>
87 </div>
88 </div>
```

The following code should still be placed around each of these instances with the `` tags wrapping this price code:

Unset

```
<span class="bold-compare-at-money" data-product-id="{{ product.id }}">{{ compare_at_price | money }}</span>
```

```

product-price.liquid Older versions Wrap
61 </div>
62 <div class="price_sale">
63 <dt>
64 <span class="visually-hidden visually-hidden--inline">{{ 'products.product.sale_price' | t }}</span>
65 </dt>
66 <dd>
67 <span class="price-item price-item--sale" data-sale-price>
68 {{ money_price }}
69 </span>
70 </dd>
71 <dt>
72 <span class="visually-hidden visually-hidden--inline">{{ 'products.product.regular_price' | t }}</span>
73 </dt>
74 <dd>
75 <s class="price-item price-item--regular" data-regular-price>
76 <span class="bold-compare-at-money" data-product-id="{{ product.id }}">{{ compare_at_price | money }}</span>
77 </s>
78 </dd>
79 </div>
80 <div class="price_badges">
81 <span class="price_badge price_badge--sale" aria-hidden="true">
82 <span>{{ 'products.product.on_sale' | t }}</span>
83 </span>
84 <span class="price_badge price_badge--sold-out">
85 <span>{{ 'products.product.sold_out' | t }}</span>
86 </span>
87 </div>
88 </div>

```

In some cases, the product price code may look slightly different:

Unset

```

{{ item.price | money }}

```

```

search-template.liquid •
search-template.liquid Older versions
65 <span class="visually-hidden">{{ 'products.general.regular_price' | t }}</span>
66 <s class="product-item_price--reg">{{ item.compare_at_price | money_without_trailing_zeros }}</s>
67 <span class="product-item_price--sale">{{ 'products.general.now_price_html' | t: price: item }}
68 {% else %}
69 {% if item.price_varies %}
70 {{ 'products.general.from_text_html' | t: price: itemPrice }}
71 {% else %}
72 {% if item.compare_at_price > item.price %}
73 <span class="visually-hidden">{{ 'products.general.sale_price' | t }}</span>
74 {% else %}
75 <span class="visually-hidden">{{ 'products.general.regular_price' | t }}</span>
76 {% endif %}
77
78 {{ item.price | money }}
79
80 {% endif %}
81 {% endif %}
82 </p>
83
84 {% unless item.available %}
85 <p class="product-item_sold-out">{{ 'products.product.sold_out' | t }}</p>
86 {% endunless %}
87 </div>
88 </div>
89 </a>
90 {% else %}

```

The following code should still be placed around each of these instances with the `` tags wrapping this price code:

Unset

```

<span class="money" data-product-id="{{ item.id }}">{{
item.price | money }}</span>

```

```
search-template.liquid •
search-template.liquid Older versions
65 <span class="visually-hidden">{{ 'products.general.regular_price' | t }}</span>
66 <s class="product-item_price--reg">{{ item.compare_at_price | money_without_trailing_zeros
67 <span class="product-item_price--sale">{{ 'products.general.now_price_html' | t: price: ite
68 {% else %}
69   {% if item.price_varies %}
70     {{ 'products.general.from_text_html' | t: price: itemPrice }}
71   {% else %}
72     {% if item.compare_at_price > item.price %}
73       <span class="visually-hidden">{{ 'products.general.sale_price' | t }}</span>
74     {% else %}
75       <span class="visually-hidden">{{ 'products.general.regular_price' | t }}</span>
76     {% endif %}
77   <span class="money" data-product-id="{{ item.id }}">{{ item.price | money }}</span>
78
79   {% endif %}
80   {% endif %}
81 </p>
82
83
84   {% unless item.available %}
85     <p class="product-item_sold-out">{{ 'products.product.sold_out' | t }}</p>
86   {% endunless %}
87 </div>
```

Step 5: Update the Cart Price Code

Updates **need** to be done on the cart files that pull your product price information to the cart. The number and name of these files vary from theme to theme.

It's important that each theme file that pulls this price information be edited. This is to ensure the correct information always shows for your customers.

Note: This may require you to go back and edit multiple files within your theme coding.

Go through the list of files, one at a time, and then reference your theme's files to see if that particular file exists in your theme. Not all of these files exist in a given theme.

List of theme files to update

- **Templates:** If your theme is using **Sections**, these files are located within the **Sections** folder as ***-template.liquid** files.
 - cart.liquid
- **Sections & Snippets:**
 - cart-ajax.liquid
 - cart-content.liquid
 - cart-template.liquid

Code adjustments required

First, you need to find each instance of coding that looks similar to this:

Unset

```
{{ item.price | money }}
```

```
cart-template.liquid •
cart-template.liquid Older versions
60         {% endif %}
61     {% endunless %}
62     {% endfor %}
63     {% endif %}
64
65     <p class="txt--minor">
66     <a href="/cart/change?line={{ forloop.index }}&quantity=0" class="cart_remove">{{ 'cart.remove' | t }}
67     </p>
68 </td>
69 </td>
70 <td data-label="{{ 'cart.label.price' | t }}">
71 <span>
72     {{ item.price | money }}
73 </span>
74 </td>
75 </td>
76 </td>
77 <td data-label="{{ 'cart.label.quantity' | t }}" class="large-up--text-center">
78 <input type="number" name="updates[]" id="updates_{{ item.key }}" value="{{ item.quantity }}" min="1" max="999" />
79 </td>
80 <td data-label="{{ 'cart.label.total' | t }}" class="text-right">
81     {% if item.original_line_price != item.line_price %}
82     <p class="cart-item__original-price"><small><s>{{ item.original_line_price | money }}</s></small>
83     </p>
84     {% endif %}
85 </td>
86 </tr>
87 </tbody>
88 </table>
89 </div>
90 </div>
91 </div>
```

The following code should be placed around each of these instances with the **** tags wrapping this code:

Unset

```
<span class="money" data-line-index="{{ forloop.index0 }}">{{ item.price | money }}</span>
```

cart-template.liquid •

cart-template.liquid [Older versions](#)

```
60         {% endif %}
61     {% endunless %}
62     {% endfor %}
63     {% endif %}
64
65
66     <p class="txt--minor">
67         <a href="/cart/change?line={{ forloop.index }}&amp;quantity=0" class="cart__remove">{{ 'cart.get'
68     </p>
69 </td>
70 <td data-label="{{ 'cart.label.price' | t }}">
71     <span>
72
73     <span class="money" data-line-index="{{ forloop.index0 }}">{{ item.price | money }}</span>
74
75     </span>
76 </td>
77 <td data-label="{{ 'cart.label.quantity' | t }}" class="large-up--text-center">
78     <input type="number" name="updates[]" id="updates_{{ item.key }}" value="{{ item.quantity }}" min=
79 </td>
80 <td data-label="{{ 'cart.label.total' | t }}" class="text-right">
81     {% if item.original_line_price != item.line_price %}
82     <p class="cart-item__original-price"><small><s>{{ item.original_line_price | money }}</s></small>
83     {% endif %}
```

Next you need to find each instance of coding that looks similar to this:

Unset

```
{{ item.line_price | money }}
```

cart-template.liquid •

cart-template.liquid [Older versions](#)

```
75     </span>
76 </td>
77 <td data-label="{{ 'cart.label.quantity' | t }}" class="large-up--text-center">
78     <input type="number" name="updates[]" id="updates_{{ item.key }}" value="{{ item.quantity }}" min=
79 </td>
80 <td data-label="{{ 'cart.label.total' | t }}" class="text-right">
81     {% if item.original_line_price != item.line_price %}
82     <p class="cart-item__original-price"><small><s>{{ item.original_line_price | money }}</s></small>
83     {% endif %}
84     <span>
85
86     {{ item.line_price | money }}
87
88     </span>
89     {% for discount in item.discounts %}
90     <p class="cart-item__discount"><small>{{ discount.title }}</small></p>
91     {% endfor %}
92 </td>
93 </tr>
94 {% endfor %}
95 </tbody>
96 </table>
97
98 <footer class="cart__footer">
```

The following code should be placed around each of these instances with the **** tags wrapping this code:

Unset

```
<span class="money" data-line-total data-line-index="{{ forloop.index0 }}">{{ item.line_price | money }}</span>
```

```
cart-template.liquid •
cart-template.liquid Older versions
75     </span>
76   </td>
77   <td data-label="{{ 'cart.label.quantity' | t }}" class="large-up--text-center">
78     <input type="number" name="updates[]" id="updates_{{ item.key }}" value="{{ item.quantity }}" min=
79   </td>
80   <td data-label="{{ 'cart.label.total' | t }}" class="text-right">
81     {% if item.original_line_price != item.line_price %}
82     <p class="cart-item__original-price"><small><s>{{ item.original_line_price | money }}</s></small>
83     {% endif %}
84   <span>
85     <span class="money" data-line-total data-line-index="{{ forloop.index0 }}">{{ item.line_price | money }}</span>
86   </span>
87   </td>
88   </tr>
89   {% for discount in item.discounts %}
90   <p class="cart-item__discount"><small>{{ discount.title }}</small></p>
91   {% endfor %}
92 </td>
93 </tr>
94 {% endfor %}
95 </tbody>
96 </table>
97 <footer class="cart_footer">
```

Lastly, you need to find each instance of coding that looks similar to this:

Unset

```
{{ cart.total_price | money }}
```

```
cart-template.liquid •
cart-template.liquid Older versions
95 </tbody>
96 </table>
97
98 <footer class="cart_footer">
99   <div class="grid">
100     {% if settings.cart_notes_enable %}
101     <div class="grid_item medium-up--one-half">
102     <label for="CartSpecialInstructions" class="label--block">{{ 'cart.general.note' | t }}</label>
103     <textarea name="note" id="CartSpecialInstructions" class="input--block">{{ cart.note }}</textarea>
104     </div>
105     {% endif %}
106   <div class="grid_item text-center {% if settings.cart_notes_enable %} medium-up--one-half medium-up--te
107   <p>
108     <span class="cart_subtotal-title h4">{{ 'cart.general.subtotal' | t }}</span>
109     <span class="cart_subtotal">
110       {{ cart.total_price | money }}
111     </span>
112   </p>
113   </div>
114   <p>
115     {% if cart.total_discounts > 0 %}
116     {% assign savings = cart.total_discounts | money %}
117     <p>
118     <span class="cart-subtotal savings"><em>{{ 'cart.general.savings_html' | t: price: savings }}</em>
```

The following code should be placed around each of these instances with the `` tags wrapping this code:

Unset

```
<span class="money" data-cart-total>{{ cart.total_price | money }}</span>
```

```
cart-template.liquid •
cart-template.liquid Older versions
95 </tbody>
96 </table>
97
98 <footer class="cart_footer">
99 <div class="grid">
100   {% if settings.cart_notes_enable %}
101   <div class="grid_item medium-up--one-half">
102     <label for="CartSpecialInstructions" class="label--block">{{ 'cart.general.note' | t }}</label>
103     <textarea name="note" id="CartSpecialInstructions" class="input--block">{{ cart.note }}</textarea>
104   </div>
105   {% endif %}
106 <div class="grid_item text-center {% if settings.cart_notes_enable %} medium-up--one-half medium-up--te
107 <p>
108   <span class="cart_subtotal-title h4">{{ 'cart.general.subtotal' | t }}</span>
109   <span class="cart__subtotal">
110     <span class="money" data-cart-total>{{ cart.total_price | money }}</span>
111   </span>
112 </p>
113 </div>
114 </p>
115   {% if cart.total_discounts > 0 %}
116   {% assign savings = cart.total_discounts | money %}
117 <p>
118   <span class="cart-subtotal savings"><em>{{ 'cart.general.savings_html' | t: price: savings }}</em>
```

Step 6: Install the AJAX Scripts

Note: Not all of the following coding snippets need to be inserted into your theme. You need to determine the function your theme uses to build your AJAX modals and install the appropriate coding from there. This section requires advanced knowledge of JavaScript and web coding to complete. If you are not comfortable with this, please [contact our Customer Success team](#) for further assistance.

For themes that use AJAX functionalities, updates must be made to the JavaScript files that build these AJAX pages and/or modals. The file name and location of this code vary from theme to theme.

Note: Only required for themes that have AJAX functions.

These updates are done to ensure the correct product price displays every time it's shown on your storefront. This may require you to go back and edit multiple files in your theme coding.

List of theme files to update

- **Assets:** Your theme may contain one or a few of these files. The required coding adjustment(s) are explained in the next section.
 - ajax-cart.js.liquid
 - ajaxify-cart.js.liquid
 - ajaxify-shop.js.liquid
 - ajaxify.js.liquid
 - app.js
 - apps.js
 - atlantic.js* - **Atlantic by Pixel Union**
 - colors-ajax-cart.js.liquid* - **Colors by Small Victories**
 - ella.js* - **Ella by Halo Themes**
 - empire.js* - **Empire by Pixel Union**
 - functions.min.js
 - handy.js* - **Handy by Pixel Union**
 - jas_theme.min.js
 - js_main.js
 - main.js.liquid
 - script.js.liquid
 - script.min.js
 - sections.js.liquid
 - shop.js.liquid
 - site.js.liquid
 - superstore.js* - **Superstore by Pixel Union**
 - theme.js
 - theme.js.liquid
 - theme.min.js
 - timber.js.liquid
 - vendor.min.js

* = Theme-specific file. The theme name and developer are noted beside the file name.

Code adjustments required

Themes that pass cart objects between your regular cart and the AJAX modals need to use our **Cart Doctor** function.

Most themes use the function **buildCart** to show your products in their AJAX modals and pass this information between both areas.

You need to find where this code is in one of the above files, and paste the following code within this function:

Unset

```
// Bold:PRE

if(typeof window.BOLD !== 'undefined'

    && typeof window.BOLD.common !== 'undefined'

    && typeof window.BOLD.common.cartDoctor !== 'undefined')
{

    // NOTE: "cart" should be the variable containing the
    cart JSON data

    cart = window.BOLD.common.cartDoctor.fix(cart);

}

// Bold:PRE
```

Sample A: Standard themes use the **buildCart** function to rebuild the cart.

▼ [Click here to expand...](#)

```
1 buildCart = function (cart) {
2   // Bold:PRE
3   if(typeof window.BOLD !== 'undefined'
4     && typeof window.BOLD.common !== 'undefined'
5     && typeof window.BOLD.common.cartDoctor !== 'undefined') {
6     // NOTE: "cart" should be the variable containing the cart json data
7     cart = window.BOLD.common.cartDoctor.fix(cart);
8   }
9   // Bold:PRE
10
11   // Empty cart if using default layout or not using the .load method to get /cart
12   if (!settings.useCartTemplate || cart.item_count === 0) {
13     $cartContainer.empty();
14   }
15   // Show empty cart
16   if (cart.item_count === 0) {
```

Some themes may use the function **refreshCart** to show your items in these modals.

You need to find where this code is in one of the above files, and paste the following code within this function:

Unset

```
// Bold:PRE

if(typeof window.BOLD !== 'undefined'

  && typeof window.BOLD.common !== 'undefined'

  && typeof window.BOLD.common.cartDoctor !== 'undefined')
{

  // NOTE: "cart" should be the variable containing the
cart JSON data

  cart = window.BOLD.common.cartDoctor.fix(cart);

}

// Bold:PRE
```

Sample B: Other popular themes use the following format to rebuild the cart.

▼ [Click here to expand...](#)

```
1 function refreshCart(cart) {
2   // Bold:PRE
3   if(typeof window.BOLD !== 'undefined'
4     && typeof window.BOLD.common !== 'undefined'
5     && typeof window.BOLD.common.cartDoctor !== 'undefined') {
6     // NOTE: "cart" should be the variable containing the cart json data
7     cart = window.BOLD.common.cartDoctor.fix(cart);
8   }
9   // Bold:PRE
10
11   $(".cart_count").empty();
12   $cartBtn = $(".cart_count");
```

Themes that simply display the information from your cart in the AJAX modals need to use our **Event Emitter** function.

Most themes pull this information by using the **getCartData** function.

Note: If Bold Upsell's AJAX code is already installed on your theme, you do not need to install this code. It should already be contained in your theme files.

You need to find where this code is in one of the above files, and paste the following code within this function:

Unset

```
// Bold:PRE

if (window.BOLD && BOLD.common && BOLD.common.eventEmitter
&&

    typeof BOLD.common.eventEmitter.emit === 'function'){
BOLD.common.eventEmitter.emit('BOLD_COMMON_cart_loaded');
}

// Bold:PRE
```

Sample A: Here is a standard way to use this emitter to redraw the cart price.

▼ [Click here to expand...](#)

```
1  getCartData: function() {
2    var self = this;
3    this.doAjax('/cart.js', null, function(cart) {
4      $('body').trigger(self.eventStrs.getCartSuccess, [cart]);
5    }, function(cart) {
6      $('body').trigger(self.eventStrs.getCartError, [cart]);
7    });
8    // Bold:PRE
9    if (window.BOLD && BOLD.common && BOLD.common.eventEmitter &&
10       typeof BOLD.common.eventEmitter.emit === 'function'){
11       BOLD.common.eventEmitter.emit('BOLD_COMMON_cart_loaded');
12     }
13     // Bold:PRE
14  },
```

Some themes grab this information through the **updateView** function.

You need to find where this code is in one of the above files, and paste the following code within this function:

Unset

```
// Bold:PRE

if (window.BOLD && BOLD.common && BOLD.common.eventEmitter
&&

    typeof BOLD.common.eventEmitter.emit === 'function'){

BOLD.common.eventEmitter.emit('BOLD_COMMON_cart_loaded');

}


// Bold:PRE
```

Sample B: Another way to use this emitter to redraw the cart price.

▼ [Click here to expand...](#)

```
1 updateView: function updateView() {
2   Shopify.theme.asyncView.load('/cart', // template name
3   'ajax' // view name (suffix)
4   ).done(function (_ref3) {
5     var html = _ref3.html,
6         options = _ref3.options;
7     if (options.item_count > 0) {
8       var itemList = $(html.content).find('.ajax-cart__list');
9       var savings = $(html.content).find('.ajax-cart__savings');
10      var subtotal = $(html.content).find('.ajax-cart__subtotal');
11      $('.ajax-cart__list').replaceWith(itemList);
12      $('.ajax-cart__savings').replaceWith(savings);
13      $('.ajax-cart__subtotal').replaceWith(subtotal);
14      ...
15    } else {
16      ...
17      $('[data-ajax-cart-trigger]').removeClass('has-cart-count');
18      $('[data-bind="itemCount"]').text('0');
19    }
20    // Bold:PRE
21    if (window.BOLD && BOLD.common && BOLD.common.eventEmitter &&
22        typeof BOLD.common.eventEmitter.emit === 'function'){
23      BOLD.common.eventEmitter.emit('BOLD_COMMON_cart_loaded');
24    }
25    // Bold:PRE
26    if (Shopify.theme_settings.show_multiple_currencies) {
27      convertCurrencies();
28    }
29  }).fail(function () { // some error handling
30  });
31 },
```


Step 7: Install Sales Icon (Optional)



Red Yoga Mat
\$24.99
\$5 Off!

[ADD TO CART](#)

- Extra thick for comfort and balance: firm 1-Inch extra thick mat strap alleviates stress on pressure points such as joints and knees. 72" X 24" X 1"
- Non-slip surface: non-slip material keeps you steady and staying securely on the floor, preventing injury and allowing on your practice or workout
- Durability: thick, durable material allows for everyday use the intensity of your exercise. Ideal for your home or studio
- Bpa free: Our products are free of Phthalate, heavy metals. Your mat may initially release a harmless odor. If so, unroll and air out.



1. Under **Templates**, click **product.liquid**.

Note: If your theme is using **Sections**, the code for this may be contained in the **product-template.liquid** file.

2. Copy and paste the following code where you want the sale icon to appear:

Unset

```
<div class="sale-icon-container" data-product-id="{  
product.id }" ></div>
```

Note: You can place this code anywhere within the theme file. We generally recommend placing it above the product image.



```
product-template.liquid •  
  
product-template.liquid Older versions  
13  {% comment %}  
14    Get first variant, or deep linked one  
15  {% endcomment %}  
16  {% assign current_variant = bold_selected_or_first_available_variant %}  
17  {%- include 'bold-variant' with current_variant, output: 'none' -%}  
18  
19  <div class="grid product-single">  
20    <div class="grid_item {% if product.images.size > 1 %}medium-up--three-fifths{% else %}medium-up--one-half{%  
21      <div class="photos">  
22        <div class="photos_item photos_item--main">  
23          <div class="product-single_photo" id="ProductPhoto">  
24  
25            <div class="sale-icon-container" data-product-id="{ product.id }" ></div>  
26  
27            {% assign featured_image = current_variant.featured_image | default: product.featured_image %}  
28            <a href="{ featured_image | img_url: '1024x1024' }" class="js-modal-open-product-modal" id="ProductI  
29                
31          </div>  
32        </div>  
33      {% if product.images.size > 1 %}  
34        <div class="photos_item photos_item--thumbs">
```

3. Click **Save**.

Step 8: Position the Sales Clock (Optional)

Note: When a sale has an end date applied to the sale, a clock appears on the product page beneath the product title by default. The steps below are only required if you want to change the placement of the sales clock.

To install the sales clock in your theme code, please choose your preferred placement method and follow the steps:

Product page

1. Under **Templates**, click **product.liquid**.

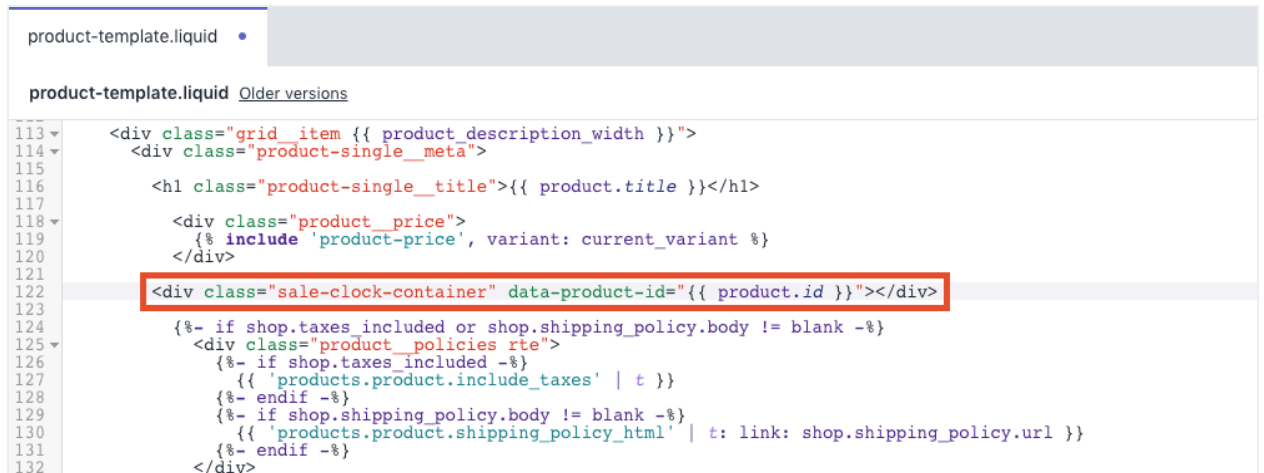
Note: If your theme is using **Sections**, the code for this may be contained in the **product-template.liquid** file.

2. Copy and paste the following code where you want the sales clock to appear:

Unset

```
<div class="sale-clock-container" data-product-id="{{ product.id }}"></div>
```

Note: You can place this code anywhere within this theme file.



The screenshot shows a code editor with the file name 'product-template.liquid' at the top. Below the file name, there is a tab labeled 'product-template.liquid' and a link for 'Older versions'. The code is displayed with line numbers on the left side, ranging from 113 to 132. The code is as follows:

```
113 <div class="grid_item {{ product_description_width }}">
114 <div class="product-single_meta">
115
116 <h1 class="product-single_title">{{ product.title }}</h1>
117
118 <div class="product_price">
119 {% include 'product-price', variant: current_variant %}
120 </div>
121
122 <div class="sale-clock-container" data-product-id="{{ product.id }}"></div>
123
124 {%- if shop.taxes included or shop.shipping_policy.body != blank -%}
125 <div class="product_policies rte">
126   {%- if shop.taxes_included -%}
127     {{ 'products.product.include_taxes' | t }}
128   {%- endif -%}
129   {%- if shop.shipping_policy.body != blank -%}
130     {{ 'products.product.shipping_policy_html' | t: link: shop.shipping_policy.url }}
131   {%- endif -%}
132 </div>
```

The line containing the sales clock code (line 122) is highlighted with a red border in the original image.

3. Click **Save**.

Collection page

1. Under **Templates**, click **collection.liquid**.

Note: If your theme is using **Sections**, the code for this may be contained in the **collection-template.liquid** file.

2. Copy and paste the following code where you want the sales clock to appear:

Unset

```
<div class="sale-clock-container" data-product-id="{{ collections['all'].products.first.id }}"></div>
```


Note: This code only displays the clock if you are running a site-wide sale. You need to specify the **ID** of an on-sale product, in this code, if your sale is not a global discount. You can also place this code anywhere within this theme file. We generally recommend placing it near the top of the file.

```
collection.liquid •
collection.liquid Older versions
1 <!-- Bold: Memberships (collection) -->
2 {% include 'bold_mem_helper' with 'collection' %}
3 <!-- end Bold code -->
4 {% paginate collection.products by 20 %}
5
6 <div class="page-width">
7
8 <header class="grid medium-up--grid--table section-header small--text-center">
9 <div class="grid_item medium-up--one-half section-header__item">
10 <h1 class="section-header__title">
11   {{ collection.title }}
12   {% if current_tags %}
13     &ndash; {% assign title_tags = current_tags | join: ', ' %}
14     {{ title_tags }}
15   {% endif %}
16 </h1>
17
18 <div class="sale-clock-container" data-product-id="{{ collections['all'].products.first.id }}"></div>
19
20   {% if collection.description != blank %}
21 <div class="section-header__subtext rte">
22   {{ collection.description }}
23 </div>
24   {% endif %}
25 </div>
```

3. Click **Save**.

Home page

1. Under **Templates**, click **index.liquid**
2. Copy and paste the following code at the top of this file:

Unset

```
<div class="sale-clock-container" data-product-id="{{
collections['all'].products.first.id }}"></div>
```

Note: This code only displays the clock if you are running a site-wide sale. You need to specify the **ID** of an on-sale product, in this code, if your sale is not a

global discount.

```
index.liquid •  
index.liquid Older versions  
1 <div class="sale-clock-container" data-product-id="{{ collections['all'].products.first.id }}"></div>  
2  
3 {% section 'slideshow' %}  
4  
5 {{ content_for_index }}  
6
```

3. Click **Save**.

Verify the Discounts installation

When the installation has been completed, we recommend creating a discount group and placing a test order on your store.

If the discount is not appearing correctly on the storefront or in the checkout, please [contact our Customer Success team](#) for further assistance.

Code removal

If you would like to remove the code from your theme, the code can be removed by deleting the snippet files, asset files, and code includes mentioned in the instruction steps above.

You can otherwise request a code removal by [contacting our Customer Success team](#).