



[Home](#) > [Legacy Articles](#) > [Shopify](#)

Changing Shopify theme

If you want to change your Shopify theme, you'll have to migrate the Customily integration into your new theme. Each theme behaves differently, so the instructions on how to migrate to a new theme are a guide for you to follow.

Keep in mind that editing the theme requires some knowledge on Javascript, CSS and JQuery, so if you're not familiar with them, we encourage you to work with someone who is!

Very important!

If you are not a developer, or don't have knowledge on Shopify themes and liquid, please contact our team and we'll be happy to migrate themes for a small cost.

Tip:

Before starting, we recommend doing this on a cloned version of your new theme, so you don't risk losing any information while editing.

Step 1 - Copy theme snippets

The first step is to copy some snippets from your current theme into the new one.

1. Create the following snippets on your new Shopify theme:

- customily-canvas.liquid
- customily-cart-image.liquid
- customily-conditions.liquid
- customily-options.liquid
- customily-sources.liquid



3. Copy the content of each snippet into the newly created files, to transfer all the code.

Step 2 - Edit the file that contains your Shopify product picture

You will need to edit the file that contains your Shopify product picture. This could be a section, a template or a snippet, depending on your Shopify theme. So first, you will need to locate where the information on your Shopify product picture is contained.

- **To locate in which file your Shopify product picture is located**

1. Open your new theme's product page (as if you were a customer making a purchase).
2. Right-click on your product picture and click inspect the element
3. Look at the class the product picture element has
4. Use that class to look for the product picture element on the theme files

Once you have located the product picture element on your files, you'll need to add some snippets in that file. The exact location of where the snippets need to be added might change from theme to theme, so this is a reference of where they should be located.

- **To add snippets to your file**

1. Open to edit the file containing Shopify's product picture
2. Add the snippet for Customily-sources.liquid at the beginning of the file. To add the snippet, copy and paste the following line:

```
{% include 'customily-sources' %}
```

3. Add the snippet for Customily-canvas.liquid inside the element containing the product picture. To add the snippet copy and paste the following line:

```
{% include 'customily-canvas' %}
```

The files structure should now look something like this:

```
1  <!-- include customily sources here -->
2
3  <div> <!-- image container -->
```

Html



```

8
9      <!-- include customily canvas here -->
10
11     <div><!-- thumbnails parent -->
12         .....
13         .....
14     </div>
15 </div>

```

Step 3 - Edit the file that contains your Shopify add to cart form

You will need to edit the file that contains your products add to cart form. This could be a section, a template or a snippet, depending on your Shopify theme. So first, you will need to locate where the information on your Shopify product add to cart form is contained.

- **To locate in which file your Shopify add to cart form is located**

1. Open your new theme's product page (as if you were a customer making a purchase).
2. Right-click on your product add to cart button and click inspect the element
3. Look at the form where the button is contained
4. Use the class of the form to look for the add to cart form on the theme files

Once you have located the add to cart form on your files, you'll need to add a snippet in that file. The exact location of where the snippet needs to be added might change from theme to theme, so this is a reference of where they should be located.

- **To add the snippet to your file**

1. Open to edit the file containing Shopify's add to cart form
2. Add the snippet for Customily-options.liquid inside the add to cart form, after the variants, and before the quantity selector. To add the snippet, copy and paste the following line:

```
{% include 'customily-options' %}
```

The files structure should now look something like this:



```

4      .....
5    </div>
6    <!-- include customily options here -->
7    <div><!-- quantity selector -->
8      .....
9      .....
10   </div>
11 </form>

```

Step 4 - Edit customily-options.liquid snippet

To make sure the hiding and showing of the preview, the selection of variants, product thumbnails, sticky preview and the styles of the options work properly, you need to edit the customily-options.liquid snippet and make sure it is connected with the proper elements from your product page.

- To make sure the preview shows correctly

1. Search for the function "window.displayCanvas". This is the function that makes the preview appear only when customers start interacting with the options. The code from this function varies depending on your theme.
2. Make sure the function is pointing to the right elements from your product page, for your new theme.

Here's an example of how that function could look like:

```

1  window.displayCanvas = function(){
2      $('.canvas-wrapper').siblings().filter(':visible').data('customily-h
3      $('.canvas-wrapper').data('customily-visible', true);
4      $('.canvas-wrapper').show();
5      };

```

Note:

Depending on the complexity of your new theme, this function could be a little more complicated.

- To make sure the preview hides correctly



2. Make sure the function is pointing to the right elements from your product page, for your new theme.

Here's an example of how that function could look like:

```

1  window.hideCanvas = function(){
2      $('.canvas-wrapper').siblings().filter(function(i,e){ return $(e).da
3          .data('customily-hidden', false).show();
4      $('.canvas-wrapper').data('customily-visible', false);
5      $('.canvas-wrapper').hide();
6  }

```

Note:

Depending on the complexity of your new theme, this function could be a little more complicated.

3. Make sure the clicking of the thumbnails of the product gallery calls the window.hideCanvas function. For example:

```

1  $(document).on('click', '.product_gallery_nav .gallery-cell', hideCanvas);
2
3  $(document).on('tap', '.product_gallery_nav .gallery-cell', hideCanvas);

```

- To make sure sticky preview is working correctly

1. Find at the beginning of the document the if:

`if (!window.mobileAndTabletcheck())`

2. Edit it to make sure it is pointing to the product picture image container (usually located on the left of your product page) and the product options container (usually located on the right of the product page).

3. The resulting code should look something like this:



```
4 }
```

Make sure you replace the selectors `'.customily-image-container'` and `'form.shopify-product-form'` with the product picture container (including the thumbnails), and the product options form from your product page.

Step 5 - Add the preview thumbnail to the cart

In order for customers to see the thumbnail of their preview when they add a product to the cart, you need to make sure it is properly configured on your new theme files.

1. Locate in which file the cart image is configured. For that, add a product to the cart, and when on the cart page right-click the thumbnail and inspect element to know the thumbnails class. Look for that class in the files, and open the file containing it.
2. To add the **custom thumbnail** add this code structure inside the thumbnail cart image:

```
1  {% comment %}
2    Adding customily cart image
3  {% endcomment %}
4  {% include 'customily-cart-image' %}
5  {% if custom_img.last == '' %}
6    {% comment %}
7      CART IMAGE
8    {% endcomment %}
9    <!-- HTML OF CART IMAGE -->
10   {% else %}
11     {% comment %}
12       CUSTOMILY UPLOADED IMAGE
13     {% endcomment %}
14     
15   {% endif %}
```

3. To add in the cart a link to a bigger version of the preview, look for `{% for p in item.properties %}` and inside it paste the following code:

```
1  {% if p.first contains '_custom-thumb' %}
2      Preview: <a class="lightbox" href="{{ p.last }}" target="_
3      {% endif %}
```



On the cart the link will be displayed after the details of the personalization, reading preview: link so customers can go and see a bigger version of their purchased preview.

Step 6 - Edit settings_schema.json

1. Look in your old theme for the file **settings_schema.json**
2. Copy the contents inside the following block

```
1  {
2      "name": "Customily Settings",
3      "settings": [
4          ...
5          ...
6          ...
7          ...
8          "info": "Set this setting to save device info (os and browser) in order no
9      ]
10 ]
11 }
```

3. Open the file **settings_schema.json** from your new theme
4. Browse to the bottom of the file, and before the closing] paste the code copied above.

Still have questions?

Our support team is ready to help you out

[Contact Us](#)