

0. Shopify

- Noël - Neige sur le site
- Changer background sur le thème Brooklyn
- Comment ajouter une section Hero (ou autre) sur une page
- Changer texte variantes de produits par bulle de couleur
- How to add product description tabs on product page

Noël - Neige sur le site

Le script va simuler un effet de neige qui tombe en continu, ainsi vos clients se sentiront joyeux et enclin à effectuer un achat de Noël.

Ce code est léger et développé spécialement pour décorer son site Shopify pour Noël afin d'apporter un peu de joie sur votre boutique.

Comment l'installer ?

Rendez-vous dans **online store** (boutique en ligne) - cliquez sur le bouton **Actions - Edit code** (modifier code).

Trouvez le dossier **Layout** - Ouvrez le fichier **theme.liquid** et repérez la zone de fermeture du head le **</head>** - Il vous suffit de coller le code suivant juste avant cette balise donc une ligne avant !

```
<script type="text/javascript" src="//dgeodev.com/codes/snoweffect.js"></script>
```

Changer background sur le thème Brooklyn

Mettre un arrière plan sur le thème Brooklyn

<https://www.youtube.com/watch?v=GXuz4aqO41I>

1. dupliquez votre thème et utilisez la copie
2. ouvrez edit code et aller dans Asset
3. juste en haut cliquez sur add a new skipper et cliquez sur choisir un fichier
4. allez chercher l'image que vous voulez mettre en arrière plan après avoir téléchargé l'image sur le dossier asset
5. allez dans votre dossier thème.scss.liquid et collez le code suivant

```
body {  
  
background-image: url({{ 'NOM DE VOTRE IMAGE.jpg' | asset_url }});  
  
background-repeat: no-repeat;  
  
background-position: center center;  
  
background-attachment: fixed;
```

```
-webkit-background-size: cover;
```

```
-moz-background-size: cover;
```

```
-o-background-size: cover;
```

```
background-size: cover
```

```
}
```

6. remplacez NOM DE VOTRE IMAGE dans le code par le nom de l'image

7. Sauvez et c'est terminé

Add background image to header Brooklyn theme on no-homepage pages

<https://community.shopify.com/c/Shopify-Design/Add-background-image-to-header-Brooklyn-theme-on-non-homepage/m-p/382597>

Setting a background image for an individual page with Brooklyn theme

<https://community.shopify.com/c/Shopify-Design/Setting-a-background-image-for-an-individual-page-with-Brooklyn/m-p/227453>

How To Add A Slideshow To Any Page In Your Store

<https://support.weareunderground.com/article/96-how-to-add-a-look-book-to-your-store>

How to create two columns of text on pages in Shopify

<https://www.envision.io/blogs/ecommerce-pulse/28829761-how-to-create-two-columns-of-text-on-pages-in-shopify>

Comment ajouter une section Hero (ou autre) sur une page

<https://help.invisiblethemes.com/article/46-how-to-add-a-hero-section-to-a-page>

In this article we will demonstrate how to bring a homepage section like the hero banner onto a page template

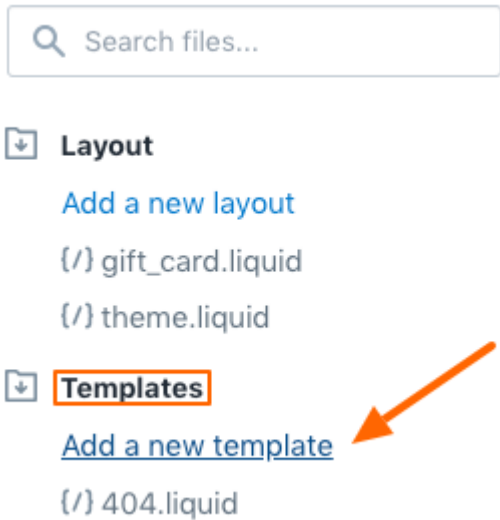
Homepage sections are located in the Sections folder in your code editor. To use a homepage section on a standard page, the best method is to create a new page template and call in the section you'd like to include. I'll breakdown the steps below.

1

Create a new page template

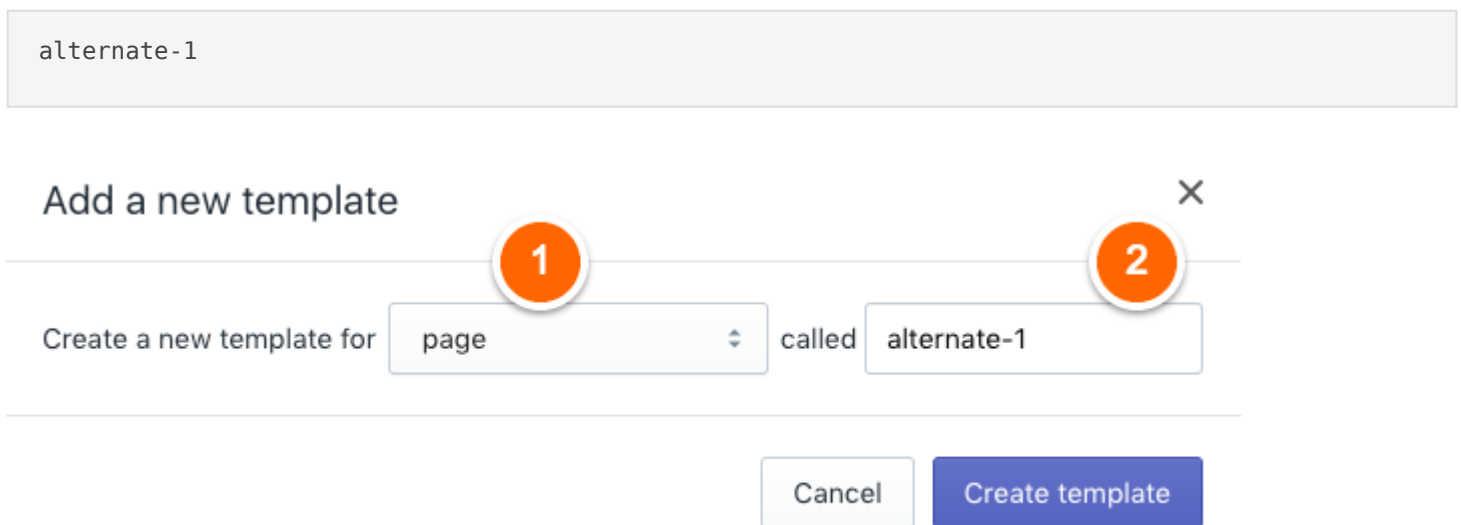
The reason for creating a new page template is to assure that the homepage section you're adding isn't applied to all your pages. Unless you need a certain homepage section to appear on all your pages with the exact same content, then creating a new page template is the best approach.

In the code editor under the **Templates** folder, choose **Add a new template**:



(1) Choose **page** for the template type

(2) Call the template :



Use the Create template button to complete.

The default code will look like:

page.alternate-1.liquid [Older versions](#)

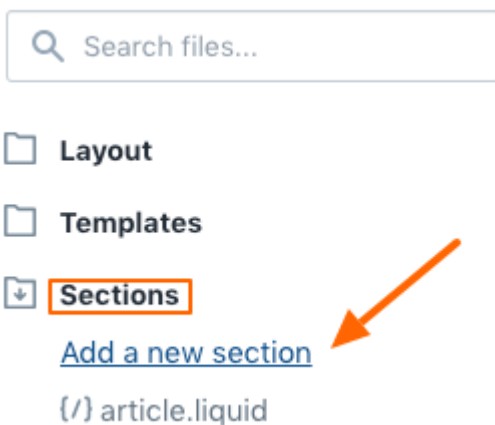
```
1 <!-- /templates/page.fullwidth.liquid -->
2
3 <section class="page">
4   <h2 class="theme">{{ page.title }}</h2>
5   {{ page.content }}
6 </section>
7
```

2

Create a new section

When a section file is configured, the content and settings are attached to that section file. In order for our page to have unique content, we need to create a new file. Without creating a new section file, the exact same content and settings from the homepage would be appear on our page. If that's your goal, you won't need to create this section file. If you need unique or different testimonials from your homepage, then this step is required.

Under the **Sections** folder, choose **Add a new section**:



Call the new section:

page-hero-1

Add a new section



Create a new section called

Cancel

Create section

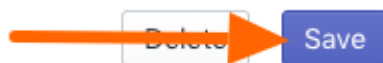
Use the Create section button to complete.

Delete all the default code that's generated:

page-hero-1.liquid [Older versions](#)

```
1  {% schema %}
2  {
3    "name": "Section name",
4    "settings": []
5  }
6  {% endschema %}
7
8  {% stylesheet %}
9  {% endstylesheet %}
10
11 {% javascript %}
12 {% endjavascript %}
13
```

Save the file:

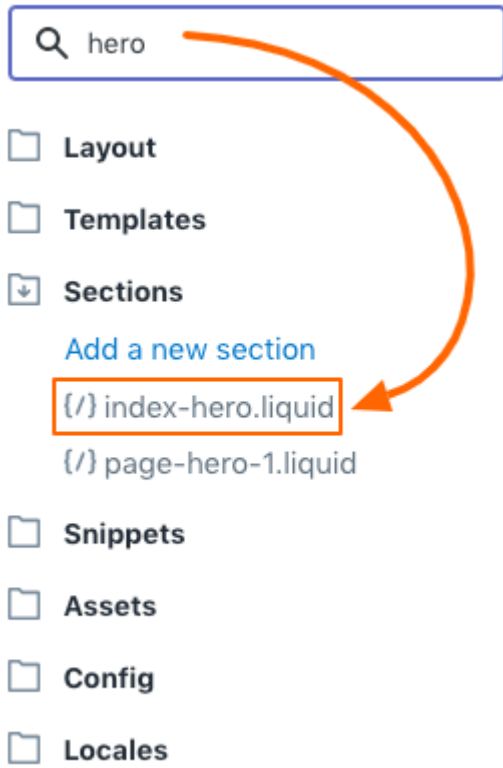


We will replace that code with code from the section file we want to use. For the testimonials, the section file is called reviews.liquid.

Refresh the code editor in your browser after the new section is created. Use the filter box to find files starting with review and you'll see the standard reviews.liquid file and the new one we created called

reviews-page-1.liquid.

We want to open the standard **reviews.liquid** file:



Copy all the code from that file:

index-hero.liquid Older versions

```
1 <!-- /sections/index-hero.liquid -->
2
3 {% if section.settings.image %}
4   {% assign img_object = section.settings.image %}
5   {% assign img_small = section.settings.image | img_url: '18x' %}
6   {% assign img_api = section.settings.image | img_url: '1x1' | replace
7 {% else %}
8   {% assign img_object = 'blank.svg' | asset_url %}
9   {% assign img_small = 'blank.svg' | asset_url %}
10  {% assign img_api = 'blank.svg' | asset_url %}
11 {% endif %}
12 {% assign transparency_class = '' %}
13 {% if section.settings.transparent or section.settings.button_image %}
14   {% assign transparency_class = 'hero_content--transparent' %}
15 {% endif %}
16
17 <div class="index-hero wide-image {{ section.settings.text_color }}" {{ s
18   data-section-id="{{ section.id }}"
19   data-section-type="hero"
20   id="FeaturedImage-{{ section.id }}"
21   {% if section.settings.parallax %}
22     data-parallax-src="{{ img_small }}"
23     data-src="{{ img_api }}"
24     data-widths="[180, 360, 540, 720, 900, 1080, 1296, 1512, 1728, 2048,
25     data-aspectratio="{{ section.settings.image.aspect_ratio }}"
26     data-sizes="auto"
27   {% endif %}>
28   {% if section.settings.link_text == '' and section.settings.link and s
29     <a href="{{ section.settings.link }}">
30   {% endif %}
31   <div class="hero_content_wrapper {{ section.settings.flex_align }}" {
32     <div class="hero_content" data-aos="fade-up">
33     {% if section.settings.button_image %}
```

Open the blank **page-hero-1.liquid** file:

🔍 hero

📁 Layout

📁 Templates

📁 Sections

[Add a new section](#)

{/} index-hero.liquid

{/} page-hero-1.liquid

📁 Snippets

📁 Assets

📁 Config

📁 Locales



Paste all the code into the blank file:

page-hero-1.liquid [Older versions](#)

```
1
```



After pasting, towards the bottom - Find this line in the **presets** section:

page-hero-1.liquid [Older versions](#)

```
201     "id": "transparent",
202     "label": "Transparent background",
203     "default": true
204   },
205   {
206     "type": "header",
207     "content": "Button"
208   },
209   {
210     "type": "text",
211     "id": "link_text",
212     "label": "Text",
213     "default": "View products",
214     "info": "Leave blank to link entire image"
215   },
216   {
217     "type": "url",
218     "id": "link",
219     "label": "Link"
220   },
221   {
222     "type": "header",
223     "content": "Button image (optional)"
224   },
225   {
226     "type": "image_picker",
227     "id": "button_image",
228     "label": "Button image"
229     "info": "Transparent .png recommended "
230   }
231 ],
232 "presets": [
233   {
234     "name": "Image with text overlay",
235     "category": "Image"
236   }
237 ]
238 }
239 {% endschema %}
240
```

Change the name to identify this section - Example "Page Hero 1":

```
201     "id": "transparent",
202     "label": "Transparent background",
203     "default": true
204   },
205   {
206     "type": "header",
207     "content": "Button"
208   },
209   {
210     "type": "text",
211     "id": "link_text",
212     "label": "Text",
213     "default": "View products",
214     "info": "Leave blank to link entire image"
215   },
216   {
217     "type": "url",
218     "id": "link",
219     "label": "Link"
220   },
221   {
222     "type": "header",
223     "content": "Button image (optional)"
224   },
225   {
226     "type": "image_picker",
227     "id": "button_image",
228     "label": "Button image",
229     "info": "Transparent .png recommended "
230   }
231 ],
232 "presets": [
233   {
234     "name": "Page Hero 1",
235     "category": "Pages"
236   }
237 ]
238 }
239 {% endschema %}
240
```

Also change the category to "Pages".

Save the file:



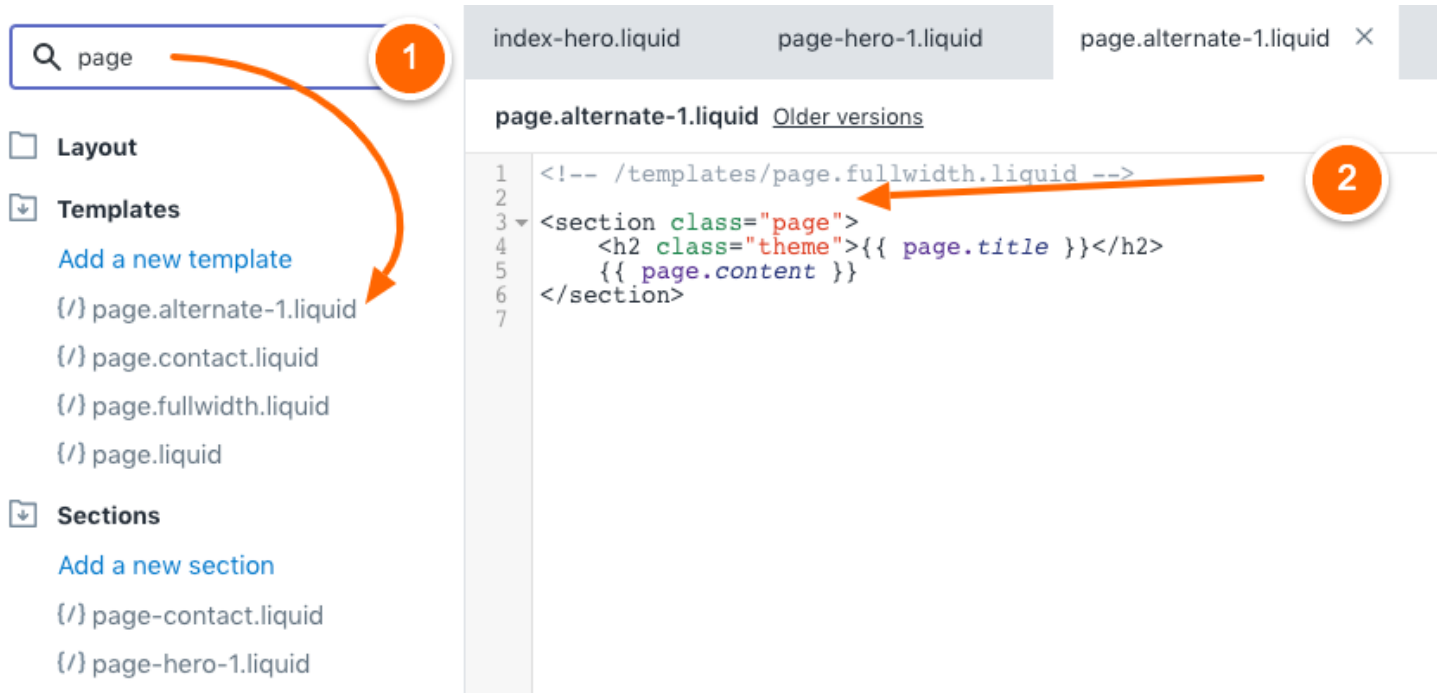
3

Assign the new section file in the page

template

Now that we have our new page template and section file created, we can connect them together.

1 - Open our new page template which we created in Step 1 - page.alternate-1.liquid:



The screenshot shows a file manager interface with a search bar containing the text "page". A circled "1" is next to the search bar. Below the search bar, the "Templates" section is expanded, showing a list of files including "page.alternate-1.liquid". An orange arrow points from the search bar to "page.alternate-1.liquid". To the right, the code editor for "page.alternate-1.liquid" is open, showing the following code:

```
1 <!-- /templates/page.fullwidth.liquid -->
2
3 <section class="page">
4   <h2 class="theme">{{ page.title }}</h2>
5   {{ page.content }}
6 </section>
7
```

A circled "2" is next to the code editor, with an orange arrow pointing to the main section code (lines 3-6).

2 - Create a few blank lines before the main section code (line 3):



The screenshot shows the code editor for "page.alternate-1.liquid" with the following code:

```
1 <!-- /templates/page.fullwidth.liquid -->
2
3
4
5
6
7 <section class="page">
8   <h2 class="theme">{{ page.title }}</h2>
9   {{ page.content }}
10 </section>
11
```

Three blank lines (lines 3, 4, and 5) have been added before the main section code (line 7).

Add:

```
{% section 'page-hero-1' %}
```

Example:

page.alternate-1.liquid [Older versions](#)

```
1 <!-- /templates/page.fullwidth.liquid -->
2
3 {% section 'page-hero-1' %}
4
5
6
7 <section class="page">
8   <h2 class="theme">{{ page.title }}</h2>
9   {{ page.content }}
10 </section>
11
```

That's the name of the section file we created in Step 2.

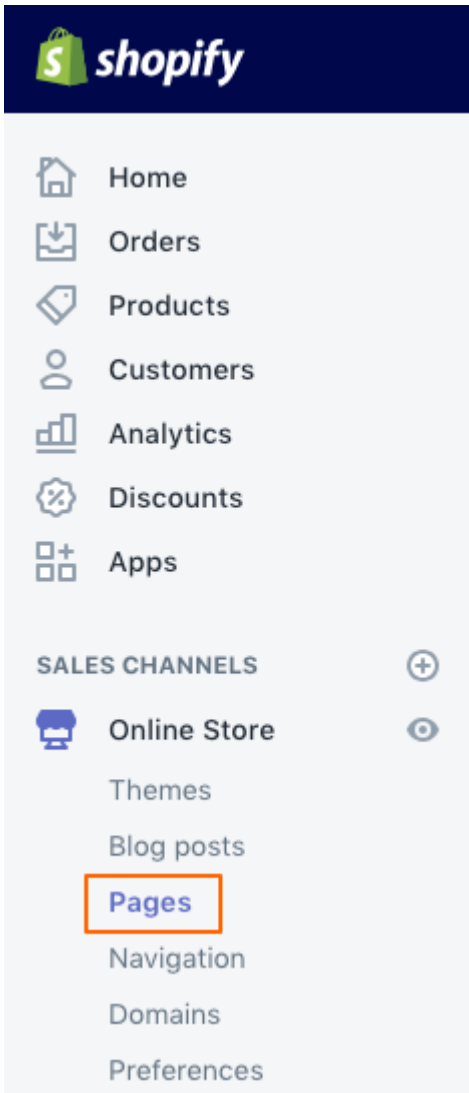
Save the file:



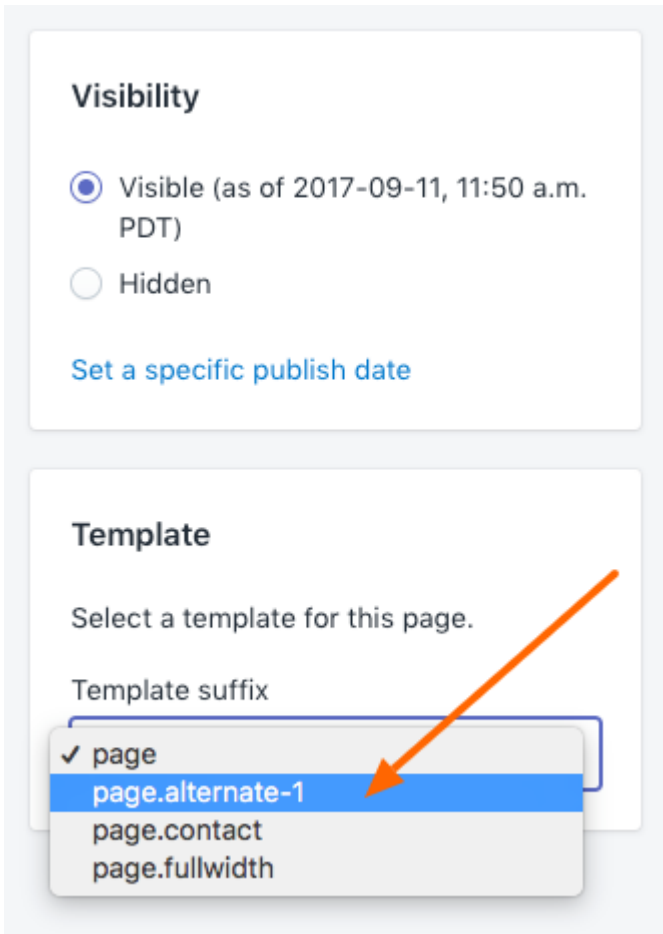
4

Choose the new page template in the page editor

Now that we have a new page template which includes testimonials, we can assign that template to a page in the Shopify page editor.

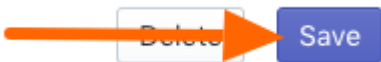


Open the page you want to use the new template with or create a new page. On the right column under templates, choose the new template:



Important Note: If you're working from a draft theme, the templates will not appear. This is a Shopify restriction. You'll only be able to assign the new template if the theme is live/published.

Save the file:



5

View the page in the Theme Editor

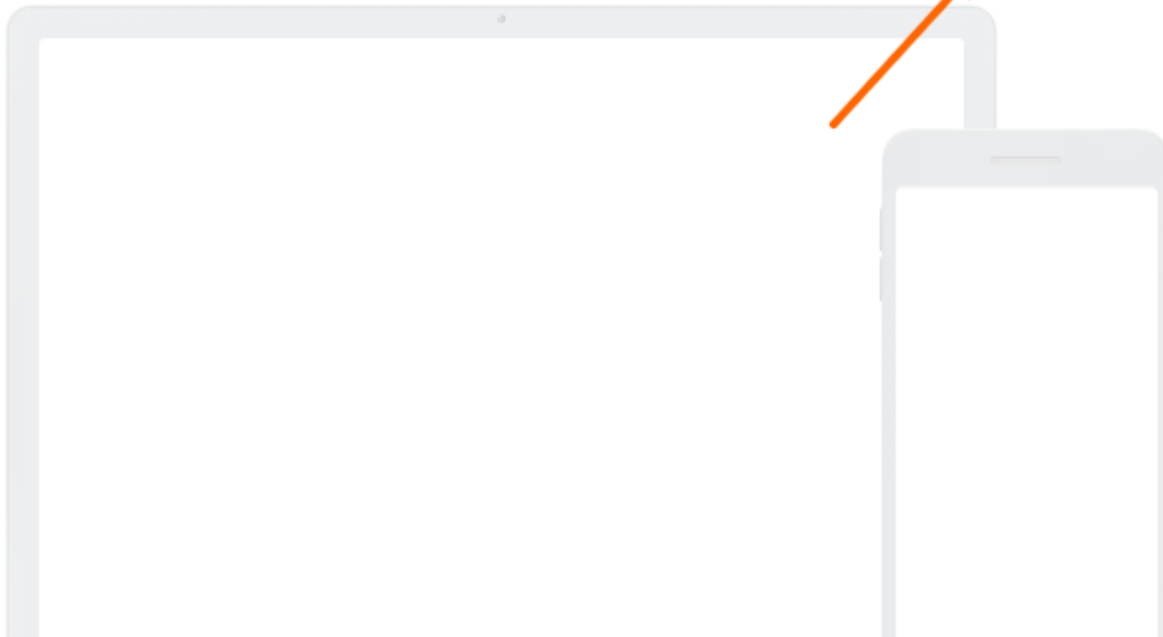
Open the Theme Editor - Customize:

Broadcast

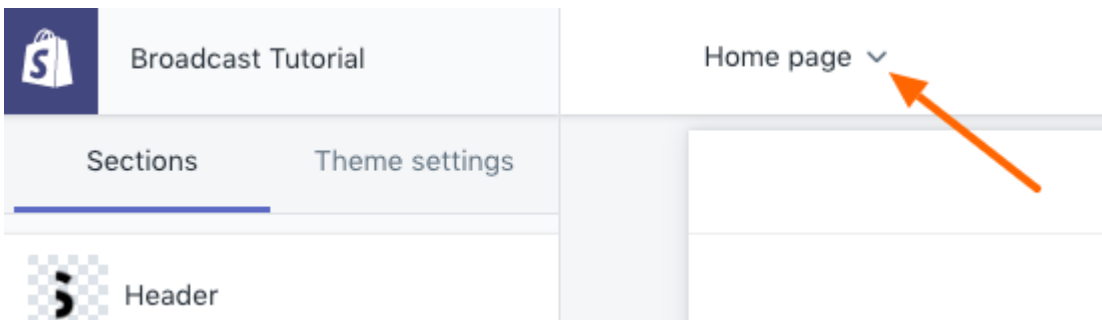
Last saved yesterday at 1:01 pm

Actions ▾

Customize

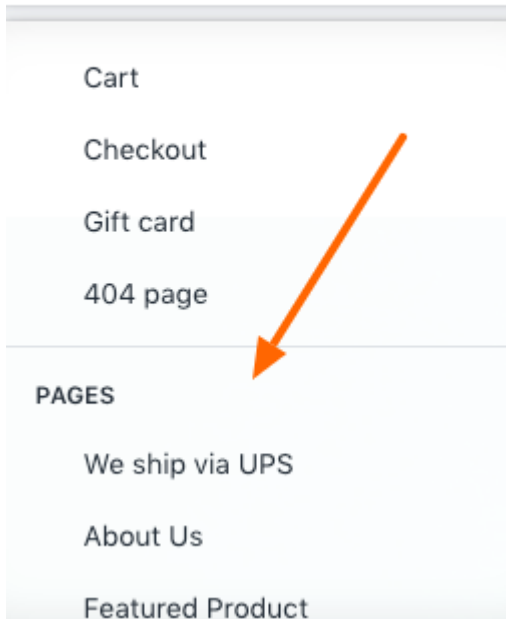


In the Theme Editor, you will need to load the page you assigned the new template to in Step 4. Use the top menu drop-down:




Scroll down to the Pages section:

Home page ▾





Choose your page.


After you've loaded the page in the Theme Editor, you'll see the new testimonial section setting on the left sections column:

 Broadcast Tutorial

Sections Theme settings

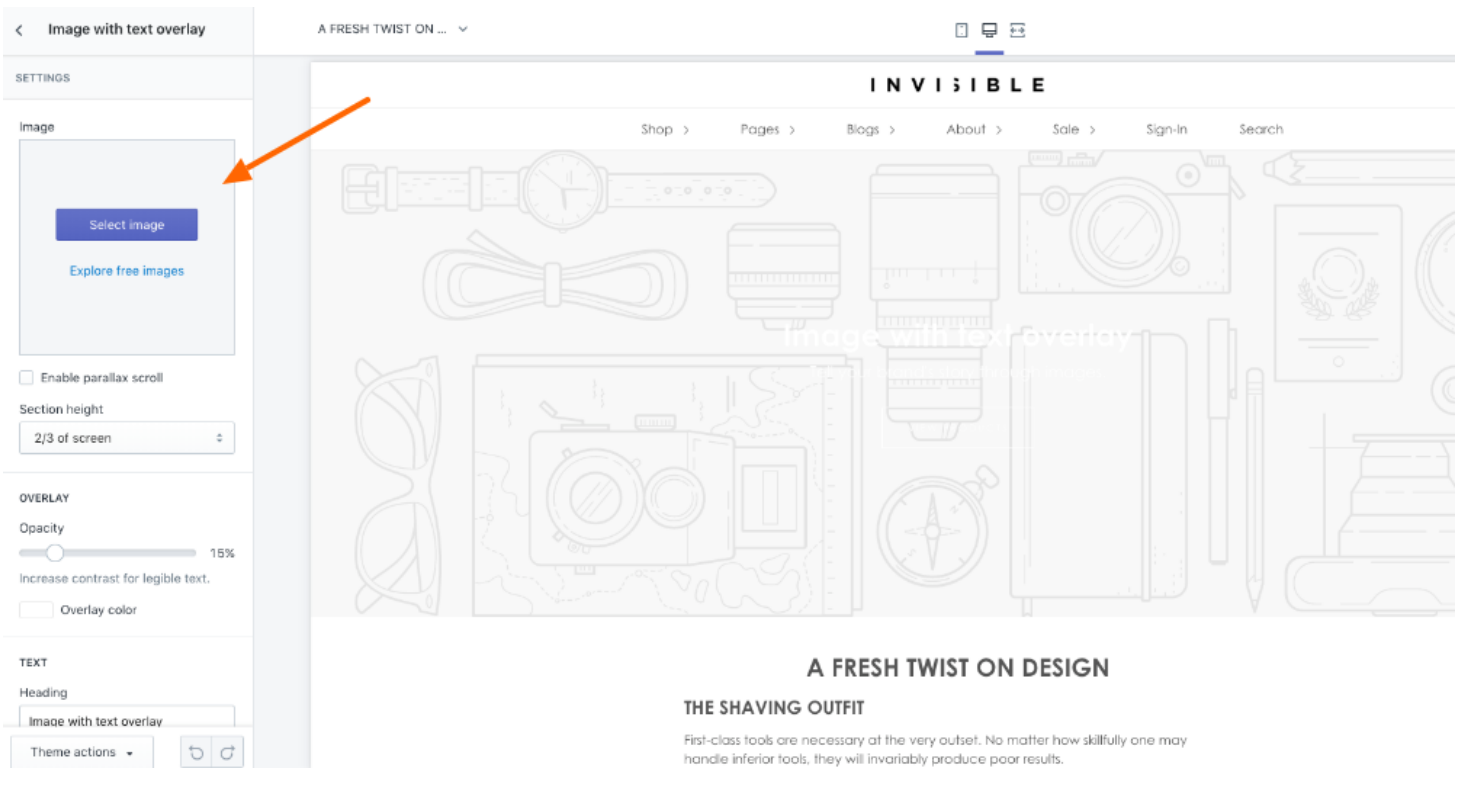
 Header

 Image with text overlay

 Footer

[Edit this page ↗](#)

You can now configure your hero banner section.



Optional - How to use the fullwidth page format for this custom template

In Step 1, we used a standard page template. If you need to use the page.fullwidth style template with your hero, then change the following code in page-alternate-1.liquid:

From the templates folder, open page-alternate-1.liquid:

🔍 page

- Layout
- Templates
 - [Add a new template](#)
 - `{/}_page.alternate-1.liquid`
 - `{/} page.contact.liquid`
 - `{/} page.fullwidth.liquid`
 - `{/} page.liquid`

Change this line:

page.alternate-1.liquid [Older versions](#)

```
1 <!-- /templates/page.fullwidth.liquid -->
2
3 {% section 'page-hero-1' %}
4
5
6
7 <section class="page">
8   <h2 class="theme">{{ page.title }}</h2>
9   {{ page.content }}
10 </section>
11
```

To:

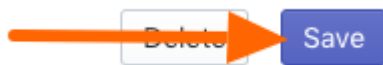
```
<section class="page page-fullwidth">
```

Example:

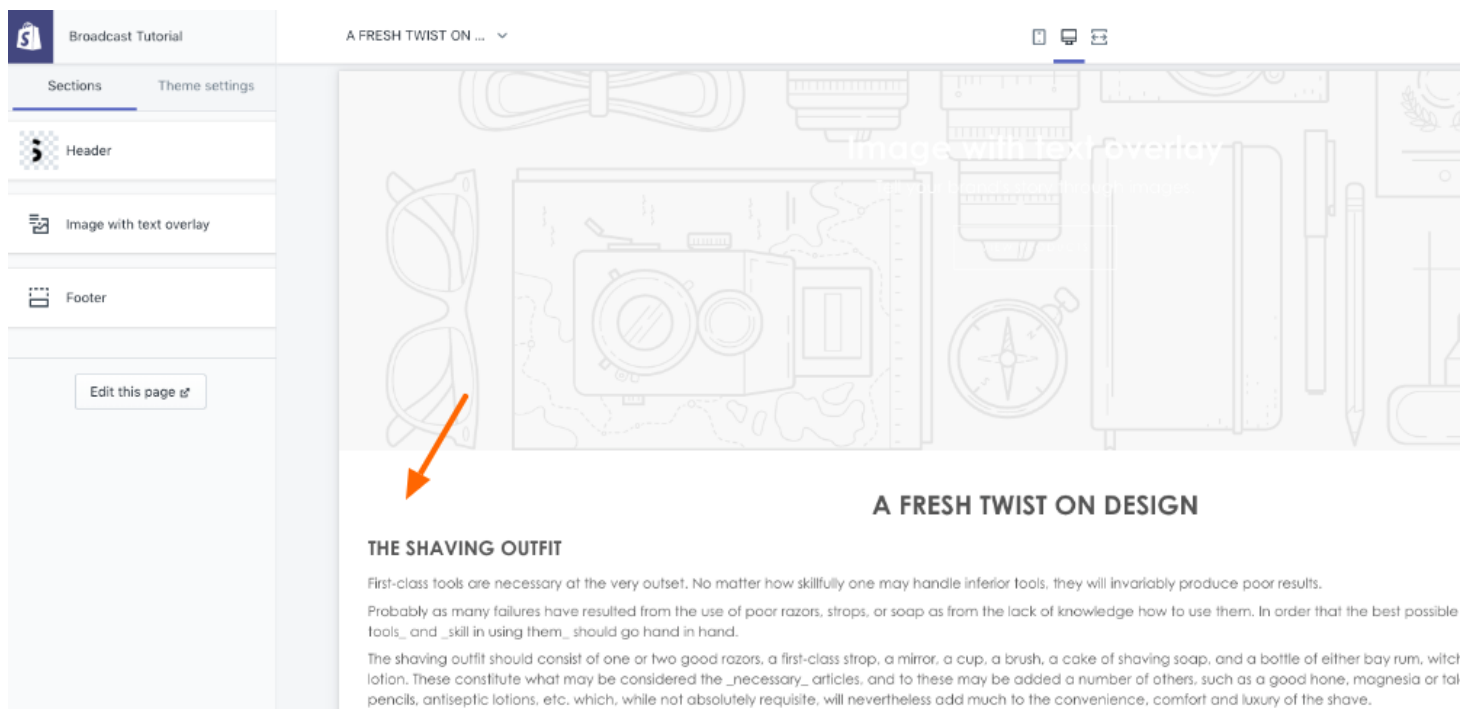
page.alternate-1.liquid Older versions

```
1 <!-- /templates/page.fullwidth.liquid -->
2
3 {% section 'page-hero-1' %}
4
5
6
7 <section class="page page-fullwidth">
8   <h2 class="theme">{{ page.title }}</h2>
9   {{ page.content }}
10 </section>
11
```

Save the file:



Example result:

A screenshot of a web editor interface. On the left is a sidebar with a 'Sections' tab selected, showing a list of sections: 'Header', 'Image with text overlay', and 'Footer'. Below the list is an 'Edit this page' button. The main area shows a preview of a page with a hero section featuring a background image of various tools (glasses, camera, compass, etc.) and the text 'Image with text overlay'. Below the hero section is a text block with the heading 'A FRESH TWIST ON DESIGN' and the sub-heading 'THE SHAVING OUTFIT'. An orange arrow points from the 'Image with text overlay' section in the sidebar to the hero section in the preview. The text block contains the following content:

A FRESH TWIST ON DESIGN

THE SHAVING OUTFIT

First-class tools are necessary at the very outset. No matter how skillfully one may handle inferior tools, they will invariably produce poor results. Probably as many failures have resulted from the use of poor razors, strops, or soap as from the lack of knowledge how to use them. In order that the best possible tools_ and _skill in using them_ should go hand in hand.

The shaving outfit should consist of one or two good razors, a first-class strop, a mirror, a cup, a brush, a cake of shaving soap, and a bottle of either bay rum, witch lotion. These constitute what may be considered the _necessary_ articles, and to these may be added a number of others, such as a good hone, magnesia or tal pencils, antiseptic lotions, etc. which, while not absolutely requisite, will nevertheless add much to the convenience, comfort and luxury of the shave.

Using this technique, you can add any of the homepage sections on to a page template. The key is to always create a custom section file. Remember, each file's content and settings are attached to that file. If you don't create custom section files, the same content and settings are shown when you call that section file.

Unique content on other pages

Now that you've created a custom page template which you can use on a page, that content will always be associated with `page-alternate-1.liquid`. If you have a second page and use that same template, the exact same hero banner will be shown. If you change the hero image, it will also change for any other page using that template. For that reason, you will need to create additional templates if you need unique content. It's an important concept to understand as it's very different from the homepage.

To create a second template which allows you to use a different hero banner, follow the same steps again but change the filenames.

Example, in Step 1, create a new template called:

```
alternate-2
```

In Step 2, create a new section called:

```
page-hero-2
```

The rest of the steps are the same, copy the code from `page-hero.liquid`. Change the presets to be "Page Hero 2".

In Step 3, change the section reference to be the same name as the section file from Step 2:

```
{% section 'page-hero-2' %}
```

In Step 4, use the new template that you've created.

Follow the same concepts for each page that needs unique hero content. If you are using the same hero on multiple pages, you can choose that page template in the Page Editor.

Using other homepage sections like Slideshow or Video

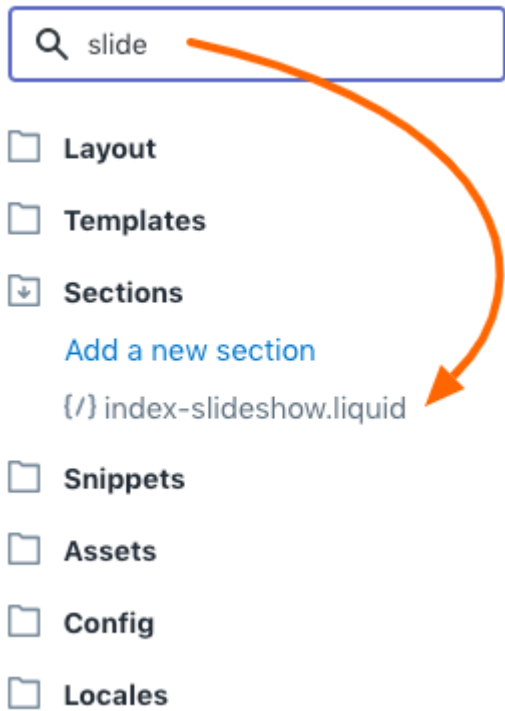
Now that you know how to use a hero banner on the top of a standard page, you can easily follow the same steps and use a different homepage section on the top. If you'd like to use the Slideshow section or Background Video section instead of the Hero section, you can use the code from those section files in Step 2 and modify your file naming.

Example for Slideshow:

In Step 2, create a new section called:

```
page-slideshow-1
```

Copy the code from:



In Step 3, change the section reference to be the same name as the section file from Step 2:

```
{% section 'page-slideshow-1' %}
```

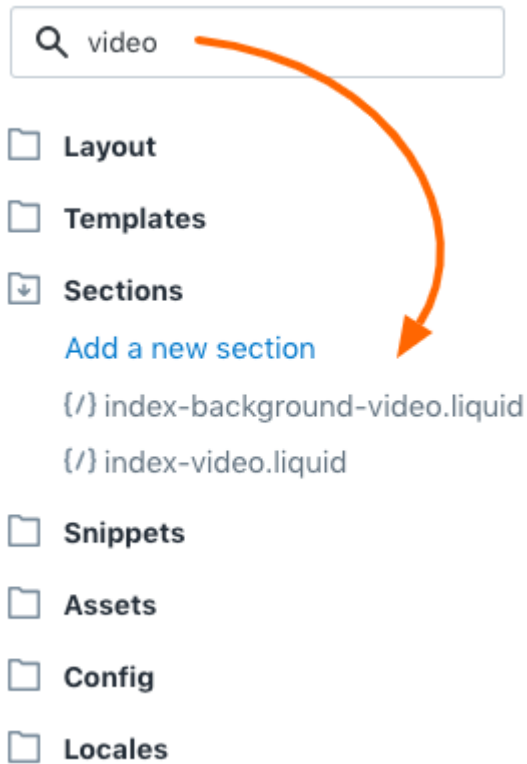
The only difference is you're using the code from `index-slideshow.liquid` instead of `index-hero.liquid`.

Example for Video:

In Step 2, create a new section called:

```
page-video-1
```

Copy the code from one of the video files:



index-background-video.liquid is the version that plays the video in the background.

index-video.liquid is the version which uses a still image and the user clicks on the play button for the popup video.

In Step 3, change the section reference to be the same name as the section file from Step 2:

```
{% section 'page-video-1' %}
```

Changer texte variantes de produits par bulle de couleur

<https://gist.github.com/carolineschnapp/ed47c42d48ae4100f7be>

COULEURS

Add Color swatches to #Brooklyn theme. Use hexadecimal values or images.

Raw

Add Color Swatches using hex values.md

What we want

We want to go from this:

Alt text

Image not found or type unknown

... to that:

Alt text

Image not found or type unknown

How to get there

1. Start compiling all the names of your colors store-wide. Add to a text file. By color names, we mean these:

Alt text

Image not found or type unknown

Note

The name of the product option does not need to be Color. This tutorial can work with any product option.

2. Find the proper hexadecimal value for each color. Use this tool: <http://www.color-hex.com/>
Add the hex value next to each color in your text file.
3. Time to write some CSS! Open your theme.scss.liquid stylesheet in the online code editor. At the very bottom of the file, define your colors, following this very strict format:

```
$colors
: ( 'Color-Hot-Pink'      #FE4365
,   'Color-Vieux-Rose'   #FC9D9A
,   'Color-Old-Paper'    #F9CDAD
,   'Color-Faded-Green'  #C8C8A9
,
'Color-Green'           #83AF9B);
```

Notes

- We are writing Sassy CSS here. We are defining a comma-separated list of space-separated lists. A list of lists. Alas, the version of Sass used by Shopify is too old for us to use a `map` object, which would have made our code more succinct.
- The name of each color must match this pattern: `<Option Name>-<Option Value>`. Use the same case as in your Admin. **Except replace white space with a dash.** So if you have a 'Color' option, and one of its values is 'Faded Green', use 'Color-Faded-Green', not 'Color Faded Green' nor 'color-Faded-green'. Accents are allowed.
- There ought to be no comma between the color name and the hex value.

- There ought to be a comma at the end of each color definition except the last one.

4. Right below that, add this CSS:

```
@each $color in $colors
{ $colorName: nth($color, 1);
  $bgColor: nth($color, 2);
  #ProductSelect-option-#{$colorName} + label
{   background-color: $bgColor;
    color: $bgColor;
    width: 50px;
    height: 50px;
    overflow: hidden;
    border-radius: 25px;
    margin: 5px;
    text-indent: 100%;
    white-space: nowrap;
  }
  #ProductSelect-option-#{$colorName}:checked + label
{   border-color: #555;
    border-width: 3px;
  }}}
```

TEXTURES

What we want

We want to go from this:

Alt text

Image not found or type unknown

... to that:

Alt text

Image not found or type unknown

How to get there

1. Start compiling the names of all our options store-wide. Add to a text file. By “options”, we mean these things:

Alt text

Image not found or type unknown

Note: The name of the product option label does not need to be Color nor Finish. This tutorial can work with any product option.

2. **Upload to your theme assets an image that will illustrate the option.** Write down the filename next to the option name in your text file. Repeat for all options.
3. Time to write some CSS! Open your theme.scss.liquid stylesheet in the online code editor. At the very bottom of the file, define your options, following this very strict format:

```
$options
: ( 'Finish-Walnut'      'swatch_walnut.png
', 'Finish-Gold'       'swatch_gold.png
', 'Finish-Graphite'   'swatch_graphite.png
');
```

Notes

- We are writing Sassy CSS here. We are defining a comma-separated list of space-separated lists. A list of lists. Alas, the version of Sass used by Shopify is too old for us to use a `map` object, which would have made our code more succinct.
- The name of each option must match this pattern: `<Option Name>-<Option Value>`. Use the same case as in your Admin. *Except* **replace white space with a dash**. So if you have a 'Finish' option, and one of its values is 'Marbled Green', use 'Finish-Marbled-Green', not 'Finish Marbled Green' nor 'finish-Marbled-green'. Accents are allowed.

- There ought to be no comma between the option name and the filename.
- **The filename can be anything, there's no naming convention to follow here.**
- There ought to be a comma at the end of each option definition except the last one.

4. Right below that, add this CSS:

```

$url_prefix: '{{ '*' | asset_url | split: '*' | first }}';
$url_suffix: '{{ '*' | asset_url | split: '*' | last }}';
@each $option in $options
{ $optionName: nth($option, 1);
  $bgImage: nth($option, 2);
  #ProductSelect-option-#{$optionName} + label
{   background-image: url( $url_prefix + $bgImage + $url_suffix );
    background-size: cover;
    color: transparent;
    width: 50px;
    height: 50px;
    overflow: hidden;
    border-radius: 25px;
    margin: 5px;
    text-indent: 100%;
    white-space: nowrap;
  }
  #ProductSelect-option-#{$optionName}:checked + label
{   border-color: #555;
    border-width: 3px;
  }}

```

That's it!

How to add product description tabs on product page

Copy the following code in the bottom of the html part of the product page (directly in the product page's editor)

Then add the informations under the related section of the code

CODE

```
<div class="tab">
```

```
  <button class="tablinks" onclick="openTab(event, 'Description')"  
  id="defaultOpen">Description</button>
```

```
  <button class="tablinks" onclick="openTab(event, 'Size')">Size Chart</button>
```

```
  <button class="tablinks" onclick="openTab(event, 'Shipping')">Shipping</button>
```

```
</div>
```

```
<div id="Description" class="tabcontent">
```

```
<h3>Description</h3>
```

```
<!-- ENTER YOUR DESCRIPTION BELOW -->
```

```
</div>
```

```
<div id="Size" class="tabcontent">
```

```
<h3>Size Chart</h3>
```

```
<!-- ENTER YOUR SIZE CHART BELOW -->
```

```
<img src=""/> <!-- IF YOUR SIZE CHART IS AN IMAGE, ADD YOUR IMAGE URL INSIDE THE QUOTES -->
```

```
</div>
```

```
<div id="Shipping" class="tabcontent">
```

```
<h3>Shipping</h3>
```

```
<!-- ENTER YOUR SHIPPING INFORMATION BELOW -->
```

```
</div>
```

```
<!-- PLEASE DO NOT EDIT THE FOLLOWING CODE -->
```

```
<style>
```

```
.tab {
```

```
  overflow: hidden;
```

```
  border-bottom: 1px solid #ccc;
```

```
}
```

```
.tab button {
```

```
  float: left;
```

```
  border: none;
```

```
  outline: none;
```

```
  cursor: pointer;
```

```
  padding: 14px 16px;
```

```
  transition: 0.3s;
```

```
  font-size: 17px;
```

```
}
```

```
.tab button:hover {
```

```
background-color: #ddd;

}

.tab button.active {

background-color: #ccc;

}

.tabcontent {

display: none;

padding: 6px 12px;

border-top: none;

}

h3 {

padding-top: 5px;

}

</style>

<script>

function openTab(evt, descName) {

var i, tabcontent, tablinks;

tabcontent = document.getElementsByClassName("tabcontent");

for (i = 0; i < tabcontent.length; i++) {
```

```
    tabcontent[i].style.display = "none";

}

tablinks = document.getElementsByClassName("tablinks");

for (i = 0; i < tablinks.length; i++) {

    tablinks[i].className = tablinks[i].className.replace(" active", "");

}

document.getElementById(descName).style.display = "block";

evt.currentTarget.className += " active";

}

document.getElementById("defaultOpen").click();

</script>
```