AJAX and Custom Dialogs

Here we will go through the steps to create custom lightbox popups for your site. A lightbox is like a traditional popup box, but it doesn't appear in a separate frame or window. These are used throughout the website builder editor, such as when you edit the settings of any component. Fortunately, this is easy to do!

Before you Start

We recommend you save a backup of your theme before starting in case you don't like the changes made. Alternately, you can remove the code added, but saving a backup provides extra security and is a good practice in general. It is also likely a quicker way to revert to a previous state.

Getting Started

This will require the addition of small amounts of 4 different code types to various sections within the theme's source code - HTML, CSS, AJAX, and JavasScript. We recommend you save a backup of your theme, make a copy of the backup, then make these changes to it. When complete, you will want to upload the modified theme.

Step 1 - HTML

When ready, first we need to add some HTML code **to our theme's *default.html* file**.

You will want to add this to the **<body>** tag area specifically:

```
<div class="window" id="content-dialog">
  <div id="dialog-content"></div>
  <div id="dialog-controls">
    <a href="/#" class="close">Close</a>
  </div>
</div>
<div id="mask"></div>
```

Step 2 - CSS

This CSS code is how we hide the dialog from being seen at all times, and how we style it to look nice. For the most part, you can leave this how it is, but you'll probably want to modify the colours to match your site. **This code will need to be added to your theme's *style.css* file.**

```
#mask {
  position: absolute;
  z-index: 9000;
  background-color: #000;
  display: none;
}
#content-dialog {
  position: absolute;
  width: 550px;
  height: 500px;
  z-index: 9999;
  padding: 10px 13px;
  display: none;
  background-color: #23201B;
  font-size: 90%;
}
#content-dialog a {
  color: #FE8D2F;
}
#content-dialog #dialog-controls {
  text-align: right;
  padding: 10px;
}
#dialog-content {
  overflow: auto;
  height: 455px;
}
```

Step 3 - AJAX

The AJAX part is actually quite simple thanks to this article. We use jQuery to load a page, then grab the content we want using a selector. We are using **$j** instead of **$** because the jQuery we provide is in a no-conflict mode. **This code will need to be added to a Javascript file that is not currently part of your theme, but one that you will need to create and add to your list of theme files - for example, a created file named*site.js* added to your theme's /assets/ folder.** The code you will need to add is:

```
// Load a new page and place it in the dialog
function loadPage( page ) {
  $j('#dialog-content').load( page + ' #location_0 > *', function() { $j('#location_0');});
}
```

Step 4 - JavaScrip

The last step is to add this JavaScript code. All of it will need to be added, however we have broken it into segments here to explain what each part does. **This code will need to be added to a Javascript file that is not currently part of your theme, but one that you will need to create and add to your list of theme files - for example, a created file named *site.js* added to your theme's /assets/ folder.**

This first piece of code is going to execute once the page has finished loading. It'll call the other two functions to setup the links and dialog so everything will work:

```
$j( document ).ready( function() {
  setupLinks();
  setupDialog();
});
```

This next bit of code will only need one small change, and that's the selector. In this example, we created a custom code block which was just a list of links to the pages we wanted to load. You'll need to do the same. The reason we use this example is so that if JavaScript is disabled, the visitor can still get to those other pages. For the most part, all this code does is find out where to position the dialog and show it. You can also tweak the fade-in and fade-out times if you'd like:

```
function setupLinks() {
  $j('#linkHolder ul li a').click( function(e) {
    // prevent default link action
    e.preventDefault();

    // Hide the scrollbars
    $j('body').css('overflow', 'hidden');

    // clear the dialog content
    $j('#dialog-content').empty();
    // start loading the new content
    loadPage( $j(this).attr('href') );
    // open the new dialog
    var id = '#content-dialog';
    //Get the screen height and width
    var maskHeight = $j(document).height();
    var maskWidth = $j(window).width();

    //Set height and width to mask to fill up the whole screen
    $j('#mask').css({'width':maskWidth,'height':maskHeight,'top':0,'left':0});

    //transition effect
    $j('#mask').fadeIn(300);
    $j('#mask').fadeTo("fast",0.8);

    //Get the window height and width
    var winH = $j(window).height();
    var winW = $j(window).width();

    //Set the popup window to center
    $j(id).css('top',  parseInt((winH/2-$j(id).height()/2) + $j(document).scrollTop()));
    $j(id).css('left', winW/2-$j(id).width()/2);

    //transition effect
    $j(id).fadeIn(500);
  });
}
```

This last bit of code just tells the dialog to close itself. It registers listeners on the close button and the semi-transparent mask to see when they're clicked. As soon as they're clicked, it hides both the mask and the dialog.

```
function setupDialog() {
  //if close button is clicked
  $j('.window .close').click(function (e) {
      //Cancel the link behavior
      e.preventDefault();
      $j('#mask, .window').hide();
      $j('body').css('overflow', 'auto');
  });
  //if mask is clicked
  $j('#mask').click(function () {
      $j(this).hide();
      $j('.window').hide();
      $j('body').css('overflow', 'auto');
  });
}
```

To link to the JavaScript file in your theme, you'll need to add this to the **<head>** tag of your theme file:

```
  <script src="/assets/site.js" type="text/javascript"></script
```

AJAXify your Website

You can AJAXify your website builder site, or make all of your pages load via AJAX by adding some simple JavaScript code utilizing the built injQuery.

Since the main content area of website builder is inside a **div#location_0**, all you have to do is hook into the Primary Navigation links and use thejQuery load function to replace the contents of **#location_0** on the current page with the contents of **#location_0** on the target page. Adding aslideUp/slideDown with some callbacks makes it have a nice transition as well.

Adding to your Site

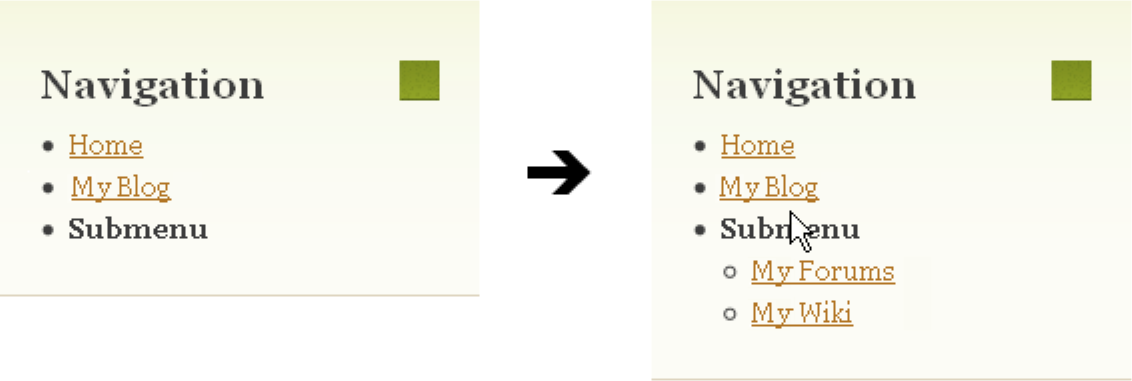
In order to add this code to your website's source code, you'll need to follow the directions below:

1. Go to the My Website toolbar menu, and click on Edit Theme Source Code.
2. Click on the *default.html* in the table that lists your theme files.
3. Add the following code below code inside of the **<head>** element:

```
<script type="text/javascript">

$j('#primary_navigation li a').click(function() { loadPage( $j(this).attr('href')); return false; });



function loadPage( page ) {

 $j('#location_0').slideUp(300, function() {

  $j('#location_0').load( page + ' #location_0 > *', function() { $j('#location_0').slideDown(300);});

 });

}

</script>
```

Creating Navigation Component Hover Menus

To further customize the content within a Navigation Component, you can create Navigation Component Hover Menus!



Note

This works only with nested sub-menus created within the Navigation Component.

Configuring for all Navigation Components:

In order to activate the hover menus for *all* of your navigation components you'll need to follow the directions below:

1. Go to the My Website toolbar menu, and click on Edit Theme Source Code.
2. Click on the *default.html* in the table that lists your theme files.
3. Add the following code below code inside of the **<head>** element:

```
<script type="text/javascript" src="/extras/js/superfish.js"></script>

<script type="text/javascript"> $j(function(){ $j("ul.navigation").superfish() });</script>
```

4. Click the Save button at the bottom of the dialog.

The page should reload and any nested menus in your Navigation Component will now be activated when the mouse hovers overtop of the label.

Configuring for a specific Navigation Component:

If you just want the hover menu activated on *one specific* Navigation Component, you'll first need to get the CSS ID of the component. To do this, you'll need to:

1. Click the collapse component button in the website builder status bar, so that the components ID are displayed.
2. Select and copy the ID into your clipboard (it should look similar to #component_0000)
3. Go to the My Website toolbar menu, and click on Edit Theme Source Code.
4. Click on the *default.html* in the table that lists your theme files.
5. You'll now need to add the following code inside of the **<head>** element:

```
<script type="text/javascript" src="/extras/js/superfish.js"></script>

<script type="text/javascript"> $j(function(){ $j("#component_0000 .navigation").superfish() });</script>
```

6. The second line of the above code needs to have #component_0000 replaced with whatever you copied as the components CSS ID.
7. Click the Save button at the bottom of the dialog.

Superfish also has a variety of options that can be tweaked. A list of these options are available on the Superfish documentation page.

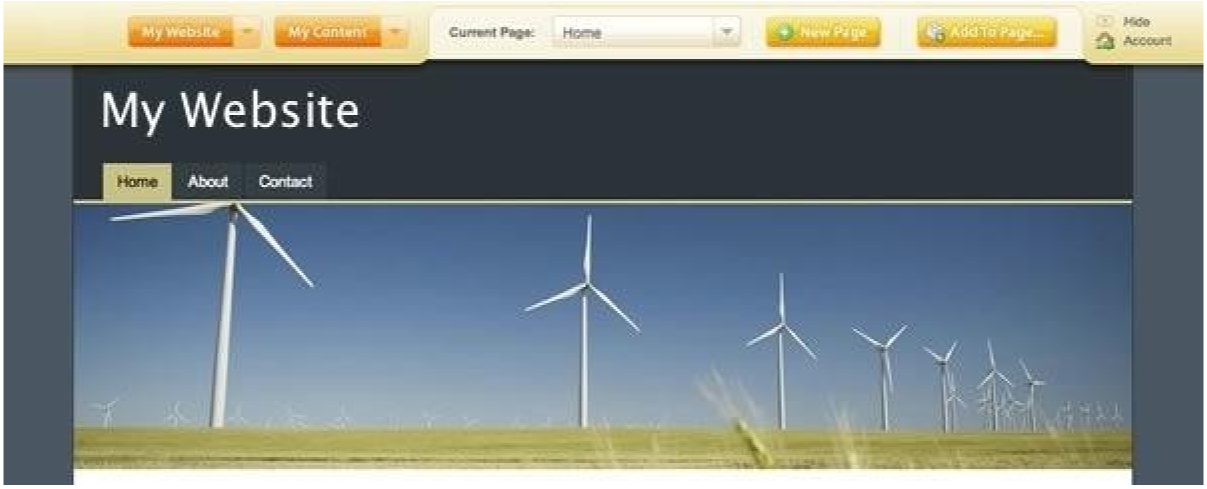
Note

Group labels can also be edited to link to internal pages.

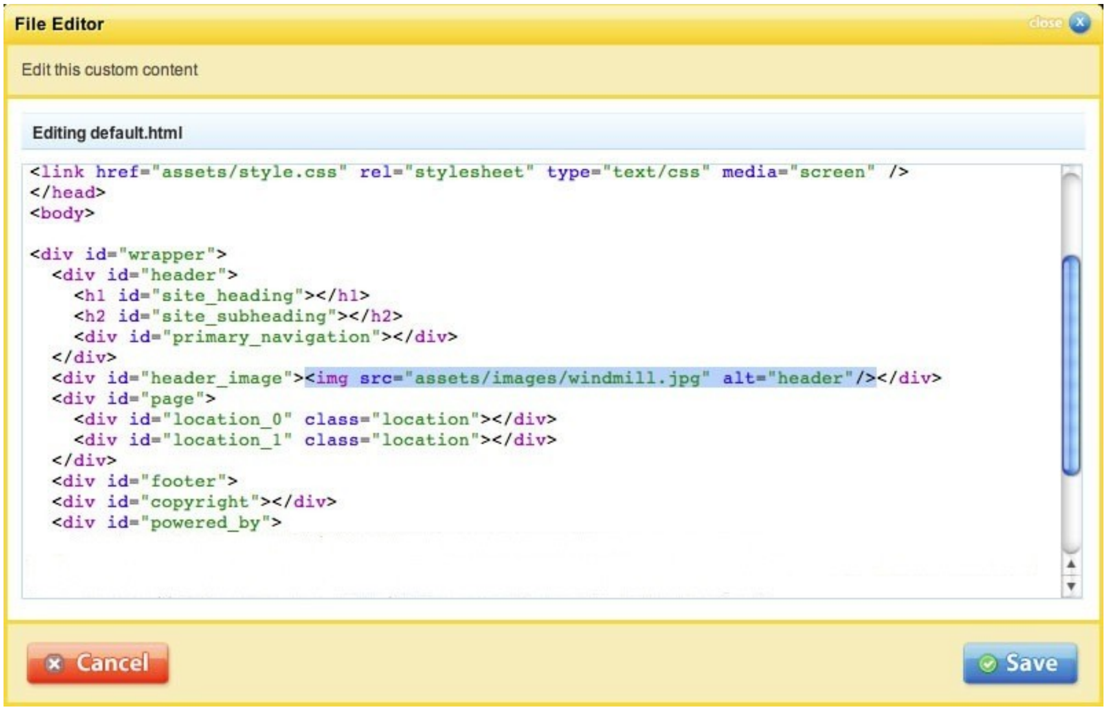
Here we'll help you through the process of using an Image Gallery Component as your theme header image. This can be useful in the carousel view setting with an auto-rotate value set.

Note:
For information about managing your site header settings, see the Header Settings page.

The first thing you need to do is create a location in your header, if one doesn't already exist. The location id should be unique from the others. If you're using a two-column layout, it's likely that you'll be creating a layout with an id of "location_2". For these example purposes, we'll be using the theme "Hiller Windmill" as our base as it already has a header image that we can replace.

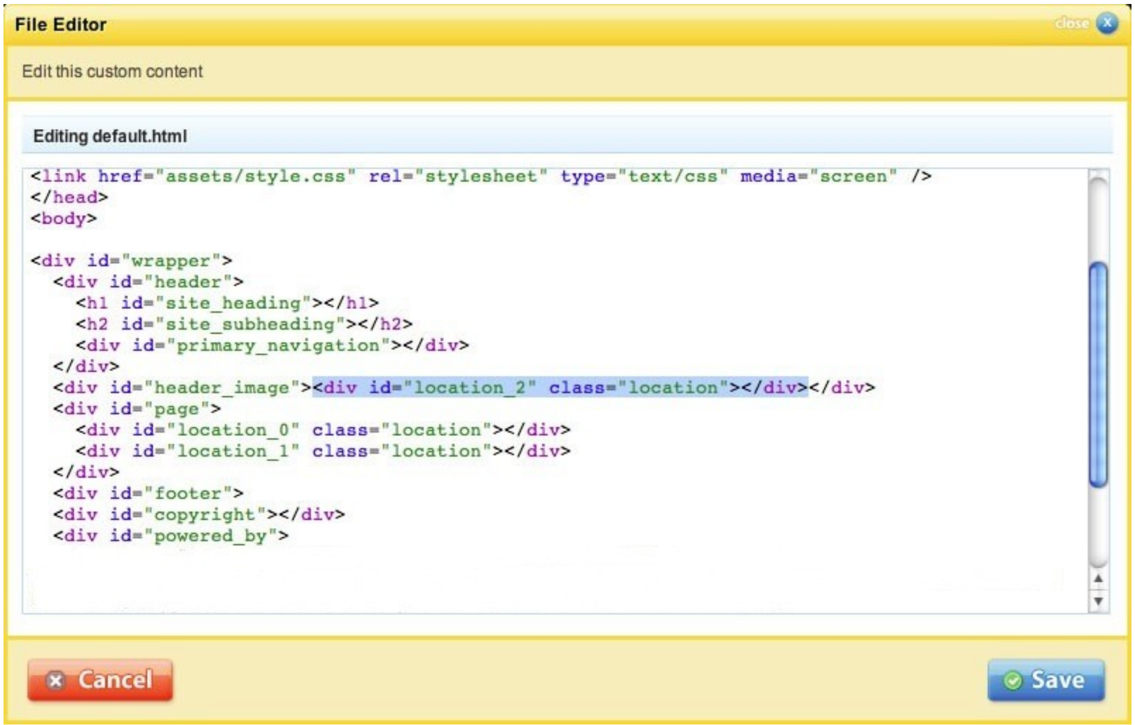


Now we want to modify the HTML in our default.html file (see: Editing Theme Source Code). It will look like this when we start:



```
<link href="assets/style.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>

<div id="wrapper">
  <div id="header">
    <h1 id="site_heading"></h1>
    <h2 id="site_subheading"></h2>
    <div id="primary_navigation"></div>
  </div>
  <div id="header_image"><img src="assets/images/windmill.jpg" alt="header"/></div>
  <div id="page">
    <div id="location_0" class="location"></div>
    <div id="location_1" class="location"></div>
  </div>
  <div id="footer">
  <div id="copyright"></div>
  <div id="powered_by">
```

And look like this with our modification:



```
<link href="assets/style.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>

<div id="wrapper">
  <div id="header">
    <h1 id="site_heading"></h1>
    <h2 id="site_subheading"></h2>
    <div id="primary_navigation"></div>
  </div>
  <div id="header_image"><div id="location_2" class="location"></div></div>
  <div id="page">
    <div id="location_0" class="location"></div>
    <div id="location_1" class="location"></div>
  </div>
  <div id="footer">
  <div id="copyright"></div>
  <div id="powered_by">
```

We've added a new location in-place of the existing header image. Now we can add an image gallery to our new location (see: Adding to a Page). Make sure you've added the images you want to use to a collection. That will make it easier for displaying on the the images you want to display in your header (see: File Collections).

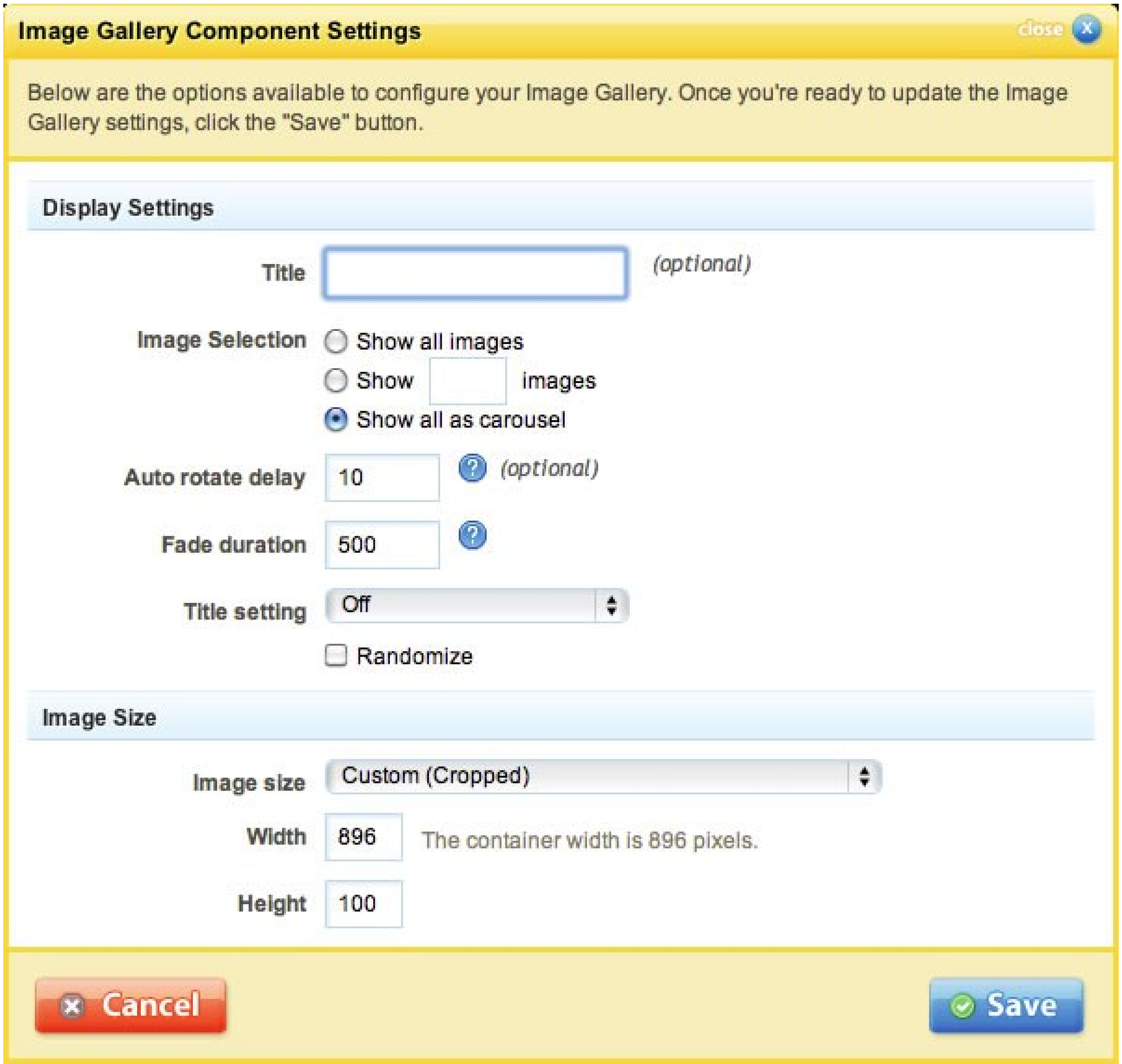


Image Gallery Component Settings

Below are the options available to configure your Image Gallery. Once you're ready to update the Image Gallery settings, click the "Save" button.

Display Settings

Title: (optional)

Image Selection: ○ Show all images ○ Show images ● Show all as carousel

Auto rotate delay: 10 (optional)

Fade duration: 500

Title setting: Off

□ Randomize

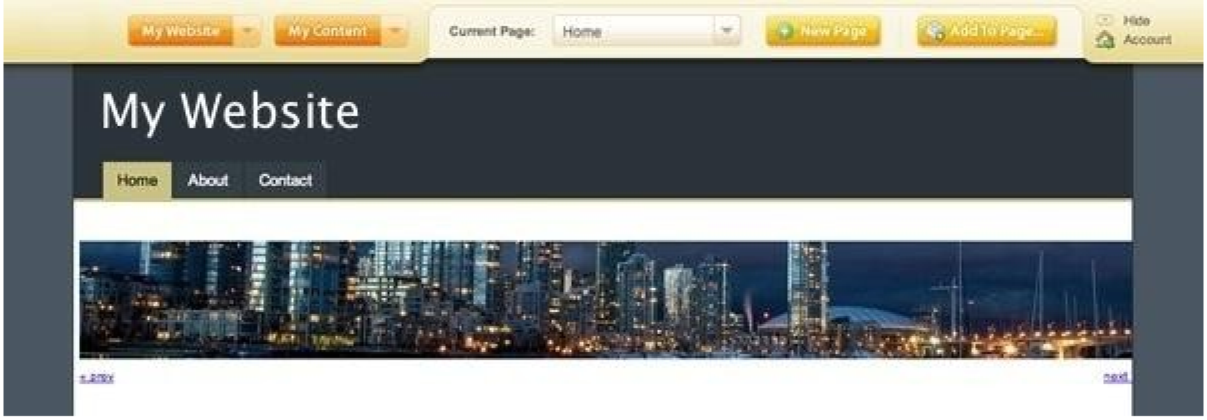
Image Size

Image size: Custom (Cropped)

Width: 896 The container width is 896 pixels.

Height: 100

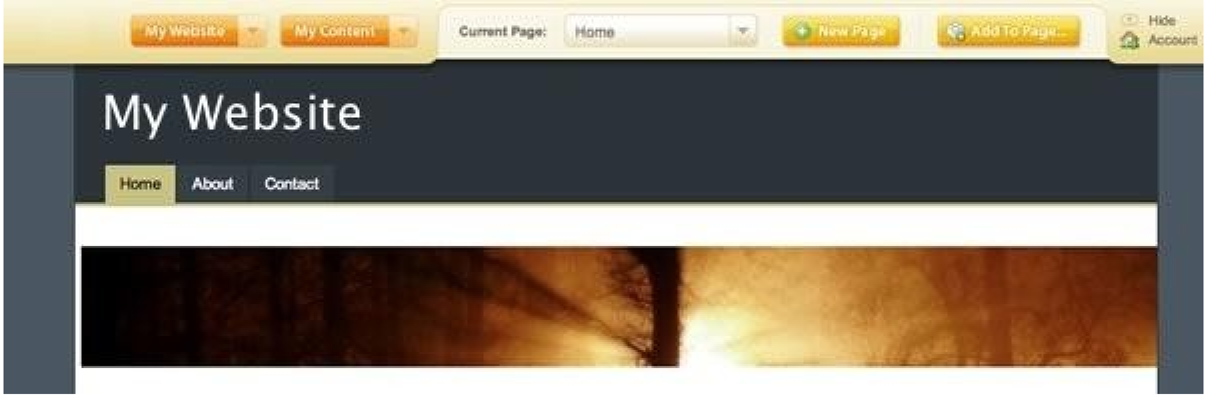
Now you have an image rotator in your header! You may need to tweak the styling to have it align and display how you'd like. This will vary from theme-to-theme and by your preference.



Extra Steps

If you want to hide the previous and next buttons, you can very easily hide them with this code. You'll need to adjust the selector to use the correct id for your layout. You'll need to add this to your main CSS file. See: Editing Theme Source Code.

```
#location_2 .image_gallery_component .carousel_controls { display: none; }
```


My Website
My Content
Current Page:
Home
New Page
Add To Page...
Hide
Account
My Website
Home
About
Contact