How to Customize your Gallery

- Overview
- Example 1: Set the thumbnail size for all galleries
- Example 2: Set the thumbnail size for an individual gallery
- Example 3: Set the gallery dimensions for the “combined” gallery
- Example 4: Set the spacing between tiled images
- Example 5: Override the component partial
- Example 6: Add a "Download" Button to the Lightbox
- Customize your Gallery with CSS!

Overview

You have several options to customize how your gallery looks and behaves, depending on how deep you are willing to go down the rabbit hole:

1. You can customize the gallery to some extent through the inspector; for example, you can set the gallery layout
2. You can also add some custom options that will be passed onto the UniteGallery script via the inspector
3. You can also add some CSS to control how the thumbnails are displayed
4. You can adjust additional settings in the plugin back-end page, you can even completely override the generated script there.
5. You can override the component partial

Regarding #2 - Setting Script Options

In each demo, you were also able to read about some possible additional options you can set on your galleries. It is important to understand which underlying JS library is used for each gallery. Currently, all galleries except for Swiper use UniteGallery. Swiper uses...Swiper.

So, for example, if you wish to make sure autoplay resumes even if there is any user interaction, then you can read up on the autoplay options in the Swiper API documentation. You can then set this in OctoberCMS in the component inspector for the Slider component, in the "Additional Swiper Options" parameter: `autoplay: {disableOnInteraction=false}`

Similarly, if you wish to disable the "full-screen" option on the UniteGallery slider, you can find the relevant option on the UniteGallery Slider docs. You can then set this in the component inspector for the Embedded Gallery component in "Script options" parameter:

```
slider_enable_fullscreen_button: false
```

Example 1: Set the thumbnail size for all galleries

Why?

If you know that you will be using a bunch of galleries throughout your site, and you also know that on most of these galleries, you will want to show thumbnails of a specific dimension, then it's easier to set it in one place instead of in every individual gallery component.

How?

You can set the thumbnail size on the plugin backend settings page: **Settings → November Gallery → Thumbnails**. If you want more control, go to **Settings → Image Resizer Settings**

Image Gallery | Popup Lightbox | Video Gallery | Blog Integration | Thumbnails | Advanced | Help!

Image Resizer Settings

ON

Use Image Resizer for image galleries

Use the October Image Resizer plugin, which automatically creates a resized and compressed thumbnail of your original image.

Width

150

Leave empty or set to 0 to only constrain the image by height; you can override this in the component inspector

Height

100

Leave empty or set to 0 to only constrain the image by width; you can override this in the component inspector

Mode

Default (as set in ImageResizer)

Quality

90

The quality of compression *requires cache clear

Please set other options, such as sharpening and compression, on the Image Resizer plugin settings page!

Example 2: Set the thumbnail size for an individual gallery

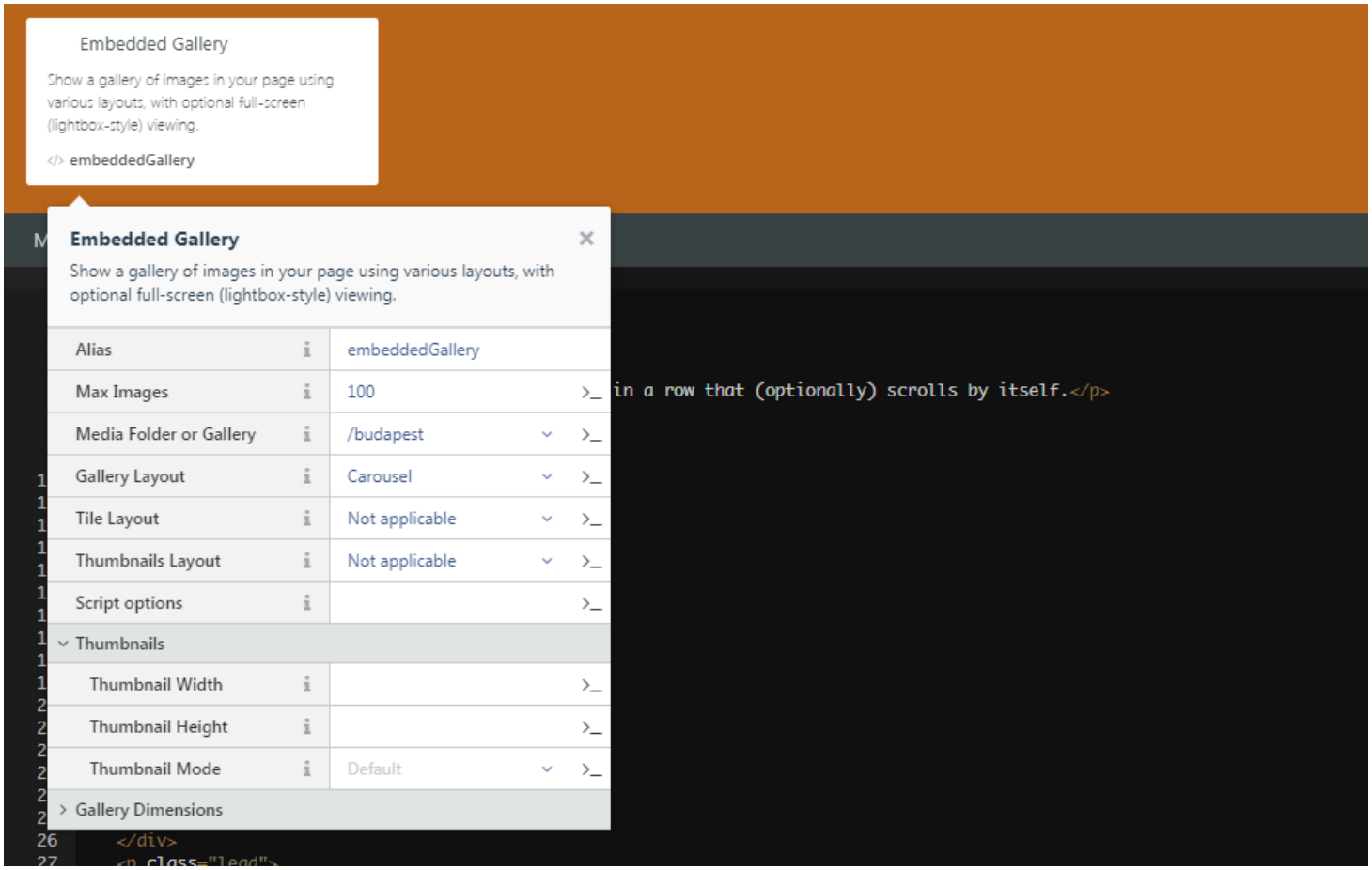
Why?

Because you may not want to use the default thumbnail size set for all galleries.

How?

You can set the thumbnail size for a specific gallery on the component properties page (a.k.a. “Inspector”).

You can also override many other settings (like gallery layout, size, etc.).


Embedded Gallery
Show a gallery of images in your page using various layouts, with optional full-screen (lightbox-style) viewing.
embeddedGallery
Embedded Gallery
Show a gallery of images in your page using various layouts, with optional full-screen (lightbox-style) viewing.
Alias
embeddedGallery
Max Images
100
Media Folder or Gallery
/budapest
Gallery Layout
Carousel
Tile Layout
Not applicable
Thumbnails Layout
Not applicable
Script options
Thumbnails
Thumbnail Width
Thumbnail Height
Thumbnail Mode
Default
Gallery Dimensions
in a row that (optionally) scrolls by itself.</p>

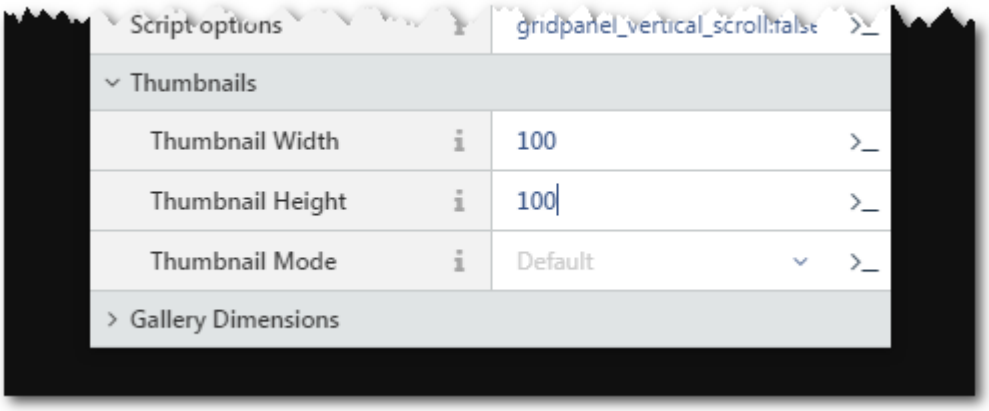
Example 3: Set the gallery dimensions for the “combined” gallery

Why?

Because you wish to show images on your website in a gallery of a specific size, and you also want to control the size of the thumbnails in the navigator strip.

How?

In the combined gallery, a large image is displayed along with a row of thumbnails. First, set your thumbnail size using the inspector let’s say to 100.



Then check the relevant options available for the combined gallery on the UniteGallery plugin page.

```
gallery_min_width: 400,                    //gallery minimal width when resizing
gallery_min_height: 300,                   //gallery minimal height when resizing
```

You can see that the “gallery_width” and “gallery_height” options control the overall size of the gallery. So enter something like the following into the *Script options* component option: `gallery_width:900,gallery_height:700,thumb_fixed_size:false`

The thumbnail size you defined using the inspector will control the size of the generated thumbnails, it will also automatically add a “thumb_height” option to the gallery. You can override this if you wish by manually adding a “thumb_height” option under *Script options*, but this should not be necessary. Additionally, the `thumb_fixed_size:false` setting enables dynamically sized thumbnails.

Example 4: Set the spacing between tiled images

Why?

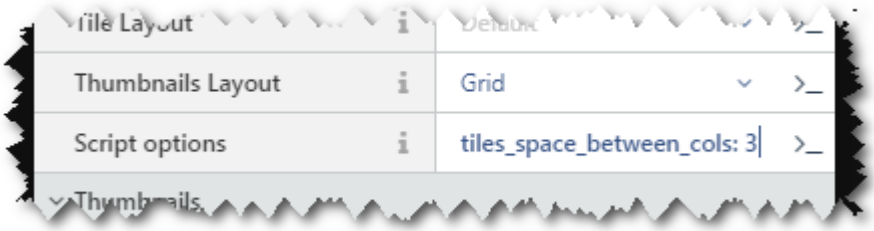
You wish to control exactly how your embedded "Tiles" gallery looks, including the space between thumbnails.

How?

First, review what options you have for the various tiled galleries on the UniteGallery website.

```
    tiles_col_width: 250,                   //column width - exact or base according the settings
    tiles_align:"center",                   //align of the tiles in the space
    tiles_space_between_cols: 3,            //space between images
    tiles_exact_width: false,               //exact width of column - disables the min and max columns
    tiles_space_between_cols: 3,            //space between images
    tiles_space_between_cols_mobile: 3,     //space between cols for mobile type
    tiles_include_padding: true,            //include padding at the sides of the columns, equal to current space between cols
    tiles_min_columns: 2,                   //min columns
    tiles_max_columns: 0,                   //max columns (0 for unlimited)
```

You can see that for the Tiles - Columns layout, you can control the spacing between the columns with the `tiles_space_between_cols: 3` option – so add it to the *Script options* NovemberGallery component option!


Tile Layout
Default
Thumbnails Layout
Grid
Script options
tiles_space_between_cols: 3
Thumbnails

Example 5: Override the component partial

See the live demo for this recipe in action!

The official OctoberCMS docs provide an in-depth explanation on how you can override component partials. Here we will describe why you would want to do so with November Gallery, as well as how.

Why?

If you are well-versed in October, then skip this section and go to "How?"!

November Gallery uses templates to generate code into your OctoberCMS page. If you wish to significantly alter the code that is generated, then you will be forced to override the component partial. Doing so may also be easier than trying to add long and complicated settings using the component inspector.

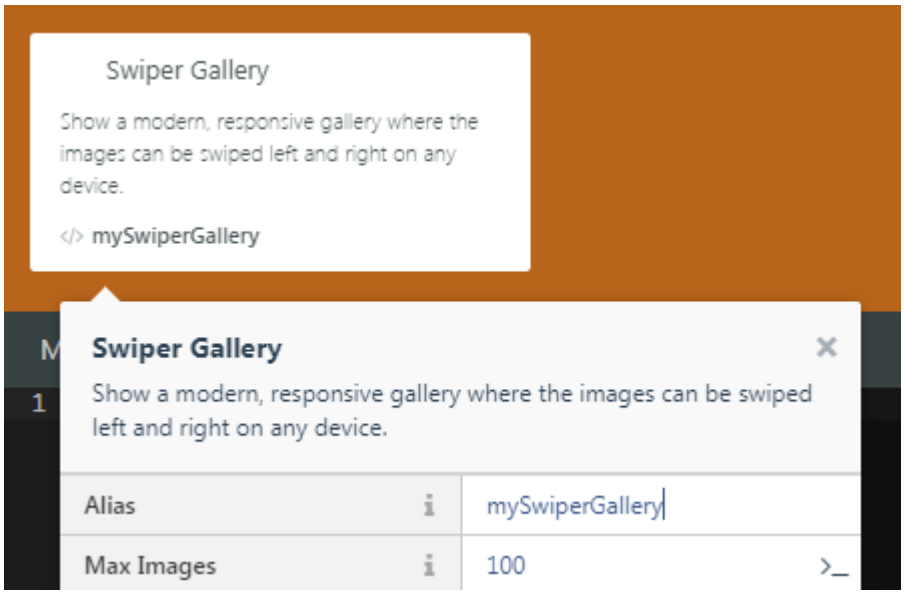
How?

These code-generation templates are called "Partials" in October lingo, and they can be found in your filesystem after you install November Gallery under the following directory: `plugins/zenware/novembergallery`. Alternatively, you can find the source code on GitHub (go to components, then check out any of the subfolders).

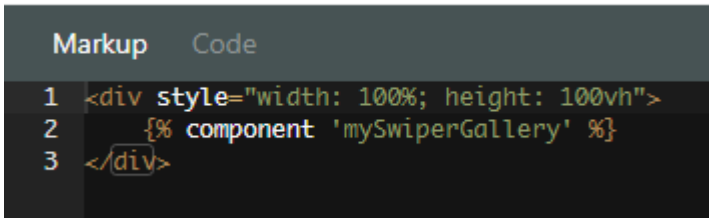
You then need to create a file in your OctoberCMS backend, in the "CMS" area, on the "Partials" page,

with the same name as the partial. The file must be "placed" in a directory that has the same name as your component alias.

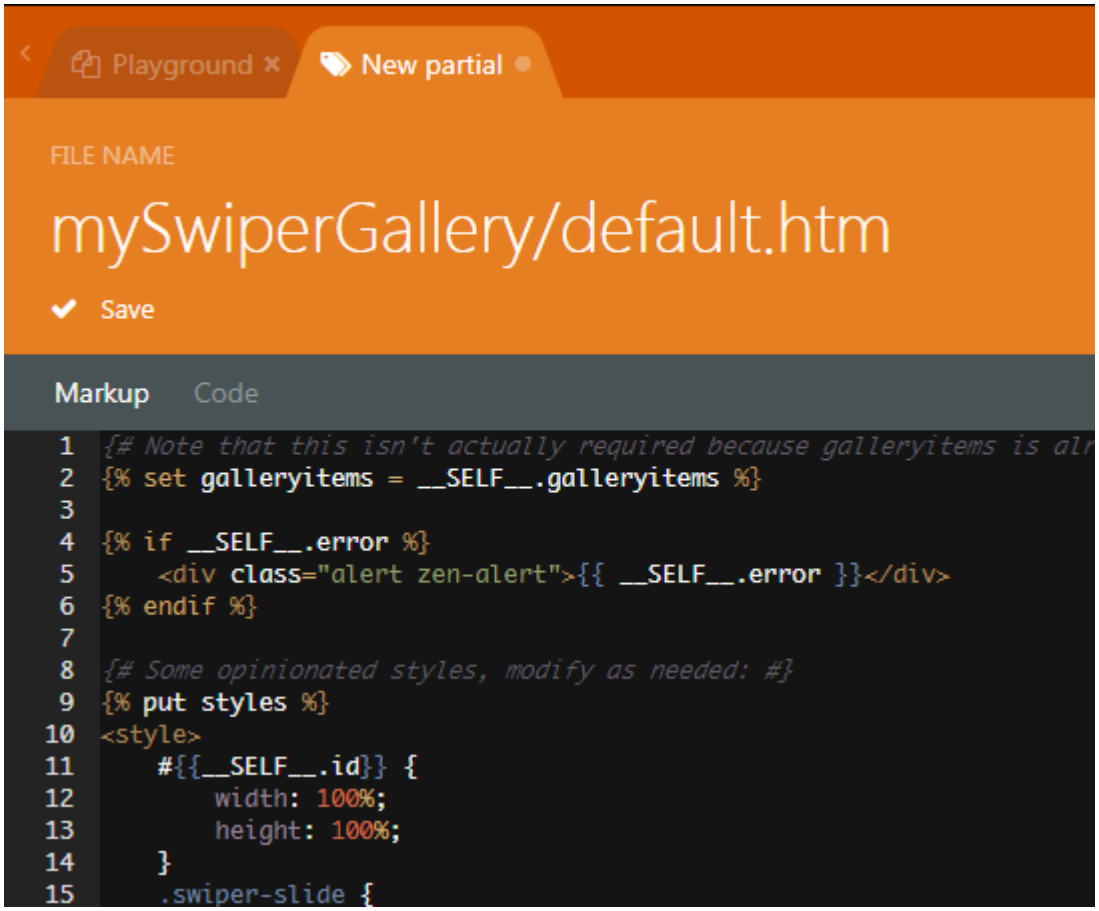
Let's go through this step-by-step. Let's say we wish to override the template for generating a swiper component. We've already dropped the component onto our page. *Although this isn't necessary*, for the sake of this tutorial let's change the component alias to "mySwiperGallery":



Remember, swiper needs to be inside of an element that has a width and a height - so we put it inside of a div that covers the whole viewport. Also, make sure to also rename your component in your page:



We then go to "Partials" and create a file called "mySwiperGallery/default.htm" and set copy-paste the original source code for this partial from GitHub.



We are going to completely override the javascript that is used to initialize Swiper. We will implement the "Multiple Slides Per View" demo on the Swiper demo website. The source code for that demo is available here.

You may in fact find that it's easier to take the demo and copy-paste that into our partial, and replace the needed bits only. You will have to:

- Make sure any `<script>...</script>` is inside of a `{% put scripts %}...{% endput %}` twig tag
- Similarly, surround any `<style>...</style>` content inside of `{% put styles %}...{% endput %}`
- Replace that actual list of images in the demo with the following:

```
{% for galleryitem in gallery.items.sortBy('fileName') %}<div class="swiper-slide"
style="background-image:url({{ galleryitem.url }})"></div>{% endfor %}
```

Our final complete page looks like this:

```
{% set galleryitems = __SELF__.gallery.items %}
{% if __SELF__.error %}
<div class="alert zen-alert">{{ __SELF__.error }}</div>
{% endif %}
{% put styles %}
<style>
    html, body {
      position: relative;
      height: 100%;
    }
    body {
      background: #eee;
      font-family: Helvetica Neue, Helvetica, Arial, sans-serif;
      font-size: 14px;
      color:#000;
      margin: 0;
      padding: 0;
    }
.swiper-container {
      width: 100%;
      height: 100%;
    }
    .swiper-slide {
      text-align: center;
      font-size: 18px;
      background: #fff;
      /* Center slide text vertically */
      display: -webkit-box;
      display: -ms-flexbox;
      display: -webkit-flex;
      display: flex;
      -webkit-box-pack: center;
      -ms-flex-pack: center;
      -webkit-justify-content: center;
      justify-content: center;
      -webkit-box-align: center;
      -ms-flex-align: center;
```

```
      -webkit-align-items: center;
      align-items: center;
    }
</style>
{% endput %}
<!-- Swiper -->
<div class="swiper-container">
    <div class="swiper-wrapper">
        {% for galleryitem in galleryitems.sortBy('fileName') %}<div class="swiper-slide"
style="background-image:url({{ galleryitem.url }})"></div>
    {% endfor %}
    </div>
    <!-- Add Pagination -->
    <div class="swiper-pagination"></div>
</div>
<!-- Initialize Swiper -->
{% put scripts %}
<script>
var swiper = new Swiper('.swiper-container', {
  slidesPerView: 3,
  spaceBetween: 30,
  pagination: {
    el: '.swiper-pagination',
    clickable: true,
  },
});
</script>{% endput %}
```

And voila, you can see that we've gotten the demo up and running using our gallery of images! You can see the demo in action at https://novembergallery.zenware.hu/cookbook/overriding-partials

Default Partials Source Code

You can find the source code for the partials used by the various November Gallery components below:

Gallery Type	Default Component Partial Path
Embedded Gallery	/plugins/zenware/novembergallery/components/embeddedgallery/default.htm
Swiper	/plugins/zenware/novembergallery/components/swipergallery/default.htm
Pop-up Lightbox	/plugins/zenware/novembergallery/components/popupgallery/default.htm
Video Gallery	/plugins/zenware/novembergallery/components/videogallery/default.htm
Image List Only	/plugins/zenware/novembergallery/components/customgallery/default.htm

Example 6: Add a "Download" Button to the Lightbox

Why?

One of our users asked:

> Endi Hariadi @EndiHariadi43 Sep 28 06:40
> I want to add a download button to each image. How to do it? I have tried and failed.

How?

Endi is using the *Embedded Gallery* component, so that is what we will focus on. There are various ways to approach this problem.

1. You could dive into the source code of the underlying component, UniteGallery and customize it to your needs. However, this would require quite an effort.
2. Or, you could attempt to use the UniteGallery API. However, in our experience, the API is not

very mature and in this case not usable (the "item_change" event could work, but unfortunately it isn't triggered when the first image is clicked on and the lightbox first shown, only when the user changes slides).

3. You could create your own UniteGallery skin- but again, this would require several hours of research and programming.
4. Or you cold just run with a quick-and-dirty jQuery hack

We'll go with #4 :-)

If you inspect the html behind a lightbox in your browser, you can see that the structure is as follows:

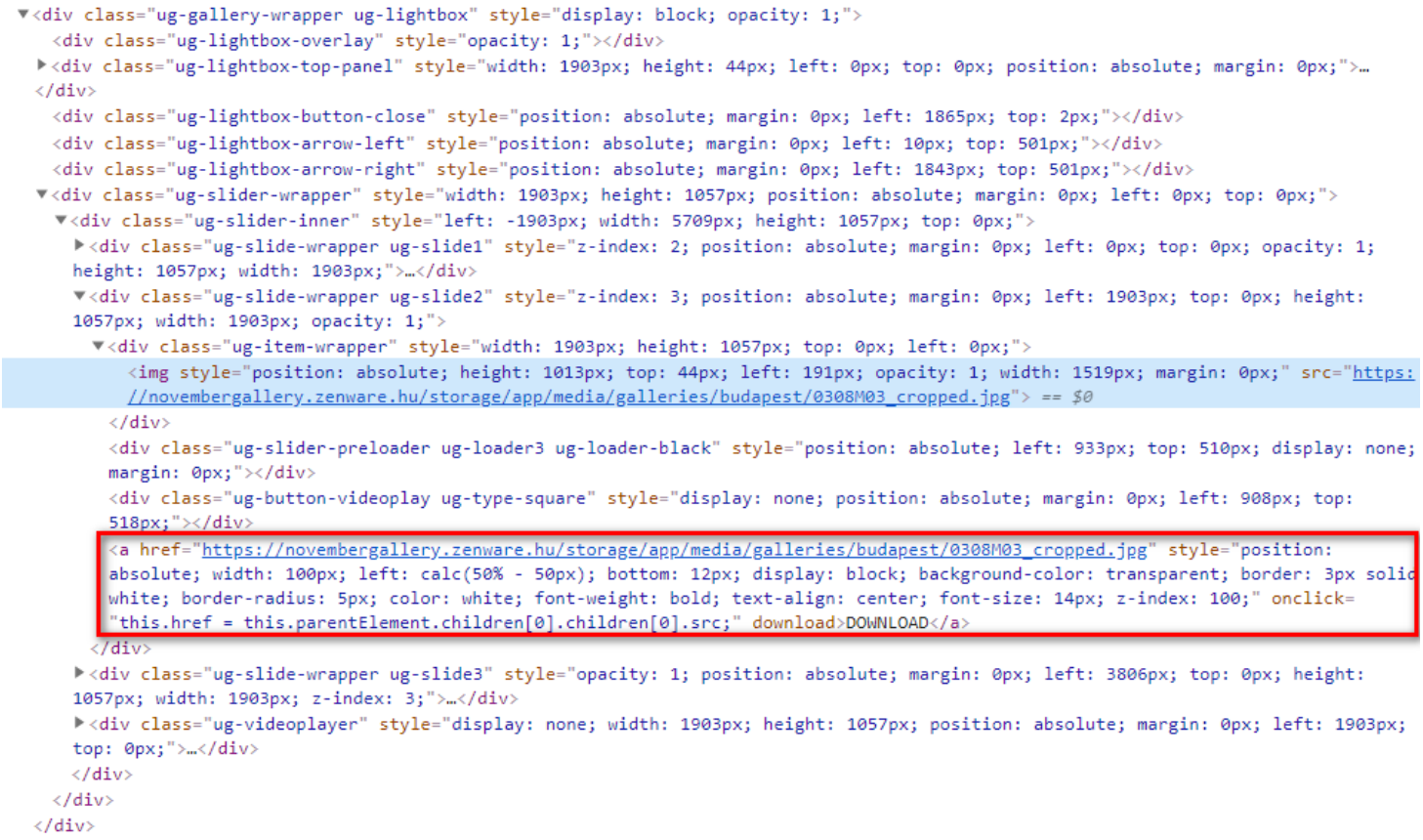
div.ug-gallery-wrapper
→ div.ug-slider-wrapper
  → dig.ug-slider-inner
    → div.ug-slide-wrapper ug-slide-1
      → div.ug-item-wrapper
        → img
      → div.ug-slider-preloader
    → div.ug-slide-wrapper ug-slide-2
    → div.ug-slide-wrapper ug-slide-3

The `img` block is replaced dynamically, and the three `ug-slider-wrapper` components are manipulated so that it looks like images slide into and out of the user's viewport. We are going to insert a link button inside each `div.ug-slider-wrapper` element, which will dynamically tell the browser to download the image that is currently being shown. We will make use of the new HTML5 "download" attribute. For some background, read this great tutorial. The generated link will then use javascript to dynamically find the `div.ug-item-wrapper` element that is before it, and the `img` element inside of the `div.ug-item-wrapper` element. So, our link button will look like this:

```
<a
href="https://novembergallery.zenware.hu/storage/app/media/galleries/budapest/0308M03_cropped.jpg"
   style="position: absolute; width: 100px; left: calc(50% - 50px); bottom: 12px; display:
block; background-color: transparent; border: 3px solid white; border-radius: 5px; color: white;
```

```
font-weight: bold; text-align: center; font-size: 14px; z-index: 100;"    onclick="this.href =
this.parentElement.children[0].children[0].src;"    download="">DOWNLOAD</a>
```

And it will be inserted into our page as follows:



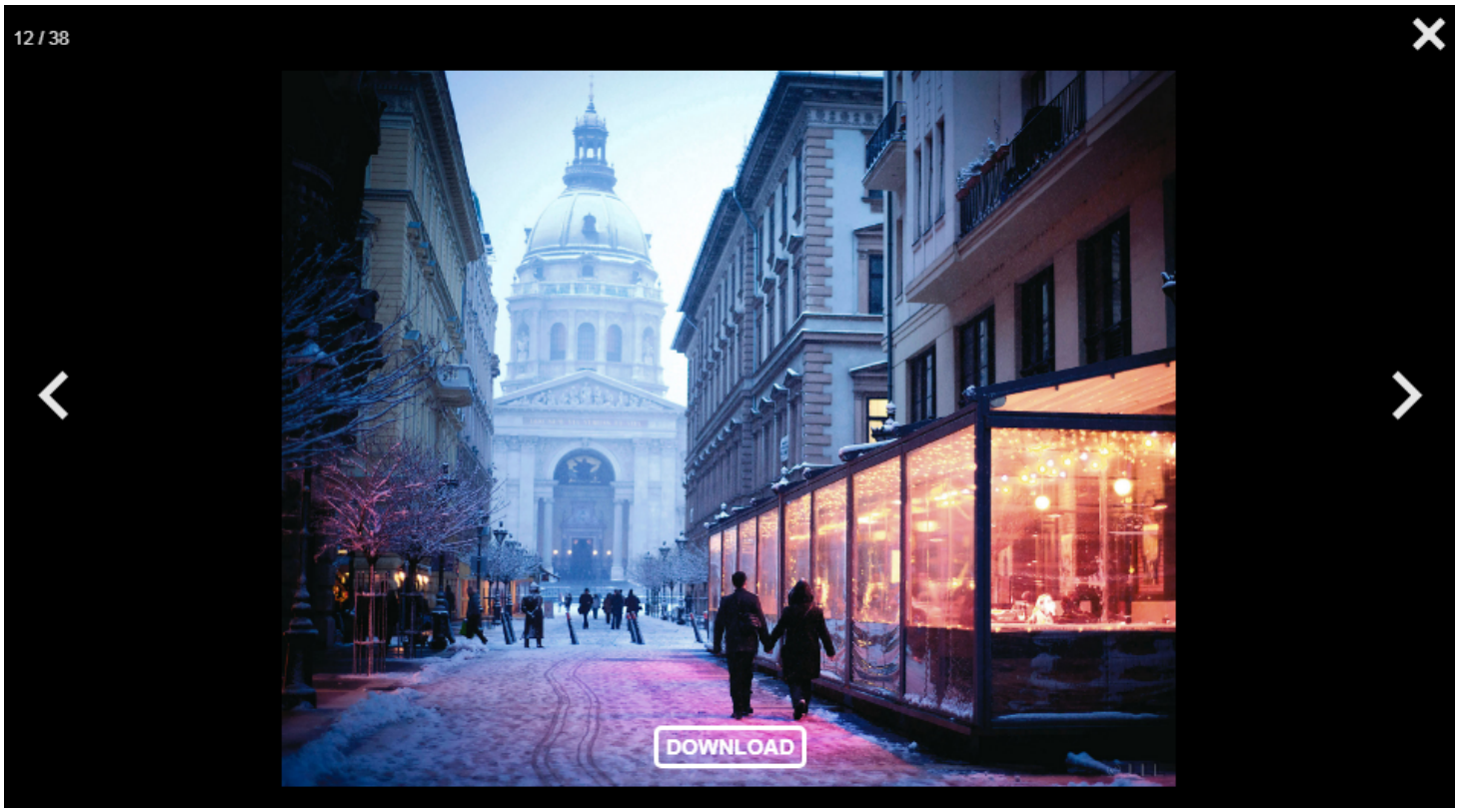
Get to the point!

So in the end this is all you need to add into your OctoberCMS page to add a download button to each image:

```
<div class="panel-body">
{% component 'embeddedGallery' %}
{% put scripts %}
<script type="text/javascript">
$(document).ready(function(){$(".ug-slide-wrapper").append('<a href="#" style="position:
absolute; width: 100px; left: calc(50% - 50px); bottom: 12px; display: block; background-color:
transparent; border: 3px solid white; border-radius: 5px; color: white; font-weight: bold; text-
```

```
align: center; font-size: 14px; z-index: 100;" onclick="this.href =
this.parentElement.children[0].children[0].src;" download>DOWNLOAD</a>');
})
</script>
{% endput %}</div>
```

and this is how it looks:



You can check out the first gallery on the November Gallery Demo site for a working example!

> Of course this is not the perfect solution to this problem but it works. You can improve upon it by removing the inline-css and perhaps adjust it so that the button doesn't become invisible on a white background. The button only works with browsers that support the `download` attribute - if you wanted to support older browsers, you could add some checks to the button OnClick handler and handle such cases as well. You could replace the button with an image, or move it to some other part of the page.

Good luck and have fun coding!

Customize your Gallery with CSS!

You can easily add CSS to your page or layout to affect how your gallery looks. How to do so is really beyond the scope of this manual. Instead we will share some hidden tricks to add style to your gallery.

Trick #1 for Swiper Gallery: style one specific image

November Gallery adds each image's file name as one of the CSS classes of that image when rendered using the Swiper Gallery component. So, if your image filename is "cute-cat-pic-1.jpg", the corresponding image will have the class "cute-cat-pic-1", which you can then style as in the following example (which sets the image to be black-and-white provided that you are using the slider component):

```
{% put styles %}
<style>
.cute-cat-pic-1 {
  background-blend-mode: luminosity;
}
</style>{% endput %}
```

Trick #2 for Swiper Gallery: Style all horizontal images

November Gallery also adds the class `swiper-slide-horizontal` to each horizontal image when rendered using the Swiper Gallery component, and `swiper-slide-vertical` to each vertical image. So it's easy to, for example, to make sure that vertical images "cover" the screen by default, but are

"contained" and centered and not repeated on landscape devices:

```
{% put styles %}
<style>
    .swiper-slide {
        background-size: cover;
        background-position: center center;
        background-repeat: no-repeat;
    }
@media screen and (min-width: 766px) and (orientation: landscape) {       .swiper-slide-
vertical {
            background-size: contain;
            background-position: center center;
            background-repeat: no-repeat;
        }
    }
</style>{% endput %}
```