About HoverAlls

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Thank you for purchasing HoverAlls, the one jQuery Plugin for all your mouse adventures. If you enjoy this plugin, please mouse on back to the marketplace to rate it and provide feedback and ideas for the next version.

Read this documentation before using HoverAlls to get the most out of the plugin.

Table of Contents

HOVERALLS

INTRO

TUTORIALS

About HoverAlls

Captions. Lightboxes. Tooltips. Panels. Scrollers.

HoverAlls was designed with one goal in mind: Create everyday jQuery animations using the exact same method. Not only does this dramatically increase productivity, but it reduces the number of commonplace jQuery plugins needed to build an attractive and interactive website. After you've mastered the basics of HoverAlls, it will ultimately save countless hours of referencing websites for appropriate usage settings, digging through plugin collections, or searching the web for a plugin that may not work. And, just as a sidenote, fewer plugins means fewer HTTP requests (and less Javascript being loaded).

This is the first release of HoverAlls, and it's mission is quite expansive. It is meant to be the solution to everyday situations - not everything. Everyone has their favorite lightbox plugin, so if you don't feel like using the HoverAlls lightbox feature, don't. HoverAlls was built to work alongside other jQuery plugins, so feel free to load your favs.

When leaving comments, please be constructive. With over 40 settings and a million browsers, odds are that there are some bugs that haven't been worked out yet. If you find a bug, please notify me immediately (info@crusader12.com) with all of the event details, browser version, and your code. If needed I will update the plugin as soon as I can. Also, if you have any suggestions, comments or ideas, HoverAlls v2 is on the drawing board.

Most importantly, HoverAlls was designed to be open-ended. With most effects there really isn't a right or wrong way of creating something, and HoverAlls is not limited to only the items on the demo page. Have fun, experiment and share your results.

It is highly recommended to get Firebug so you can see exactly where your objects are and what they're doing.

Installation

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Installing HoverAlls is similar to any other jQuery plugin.

1. Download the latest version of jQuery from jQuery.com.

2. Place the jquery file on your server or in your website folder. Link to it in the HEAD section of your webpage as shown below.

```
<HEAD> ... <script type=”text/javascript” src=”path/to/jquery.js”></script>         ... </HEAD>
```

3. Place jquery.hoveralls.min.js in the same location and link to it AFTER the library.

```
<HEAD> ...
<script type=”text/javascript” src=”path/to/jquery.js”></script>
<script type=”text/javascript” src=”path/to/jquery.hoveralls.min.js”></script>
... </HEAD>
```

4. Place the HoverAlls CSS on your server and link to it in the HEAD section.

```
<HEAD> ...
<link rel=”stylesheet” type=”text/css” media=”screen” href=”path/to/hoveralls.css”/>
... </HEAD>
```

OPTIONAL

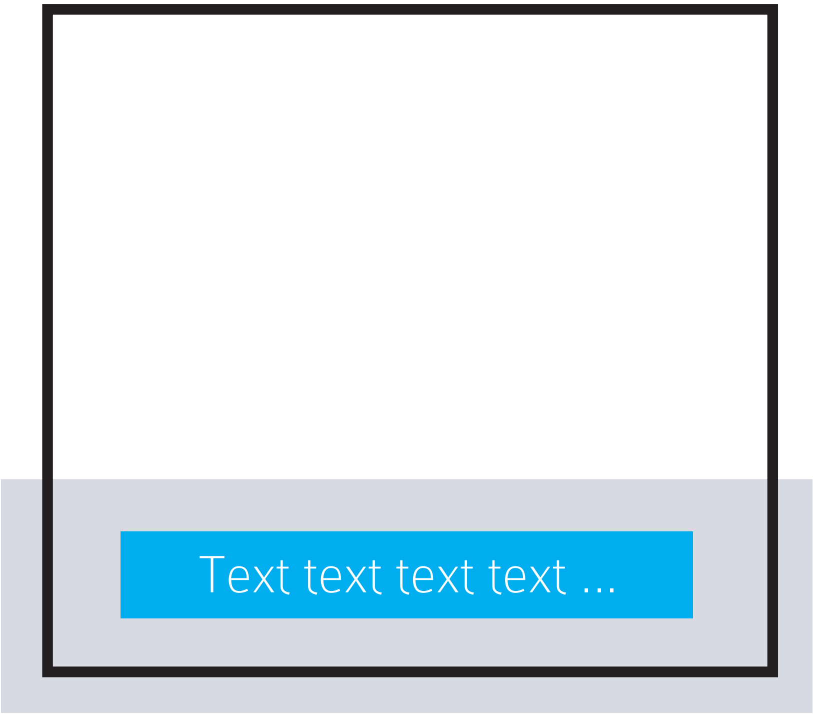
5. If you wish to use Easing transitions, link to the Easing Plugin after the jQuery library.

```
<HEAD> .....
<script type=”text/javascript” src=”path/to/jquery.js”></script>
<script type=”text/javascript” src=”js/jquery.easing.1.3.js”></script>
... </HEAD>
```

Containers

Captions. Lightboxes. Tooltips. Panels. Scrollers.

HoverAlls build an animation environment, automatically creating elements for the animation. Below is a diagram of what elements are made and how they interact with each other. Fully understanding this diagram will help when setting up your effects.



Container

Your element is wrapped inside of the HoverAlls Container (outlined in black).

Background

The background element (shown in gray) is placed inside of the HoverAlls container. This example shows the background positioned like a caption. Remember that the position of the background object is relative to the container. So, starts:”0px,0px” will place the background in the top left corner of the Container.

Text Container

The text container is inside of the background container.

Use the following settings to apply your own CSS to these objects:

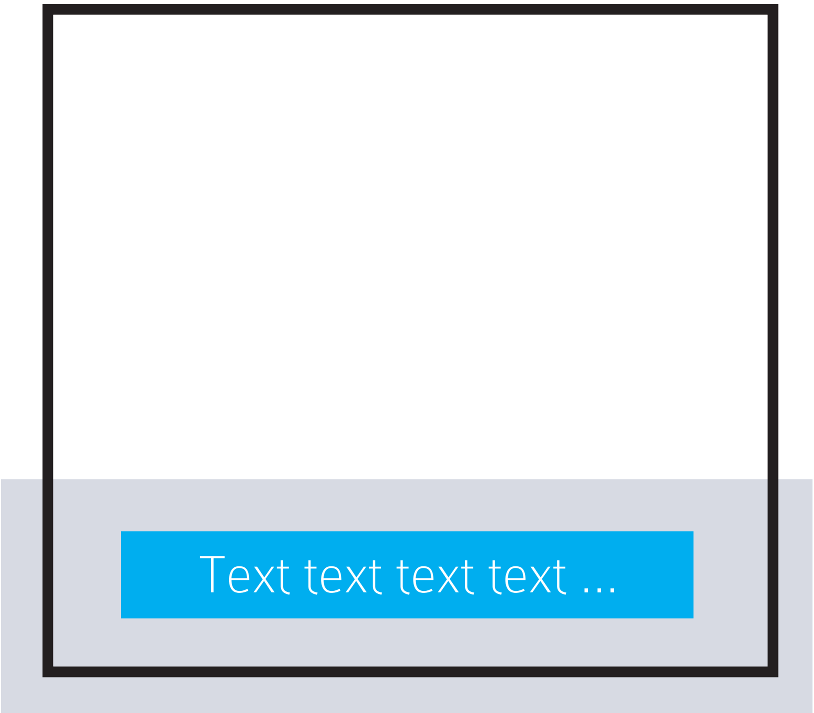
container_class:”myclass”
bg_class:”myclass”
text_class:”myclass”

How It Works

Captions. Lightboxes. Tooltips. Panels. Scrollers.

You probably realized from the previous page that HoverAlls is powered by 3 objects: the Container, Background, and Text - all of which you have styling control over. Don't worry if everything isn't making sense yet, it will. So what exactly is HoverAlls doing?

When you call HoverAlls using the $('#example').HoverAlls(); method the plugin constructs the 3 containers. HoverAlls goes out, retrieves your text and places it inside of the Text Container. It then retreives your link and places it wherever your HoverAlls settings say so.



After calling HoverAlls, your element is wrapped in the HoverAlls Container. A Background Container and Text Container have been added inside the HoverAlls Container. Any text and/or link found has been applied to your object. The animation environment is now set up. All you have to do from here is use the HoverAlls settings to tell it how to animate.

HoverAlls does NOT come with any pre-styling. These are usually wasted and unnecessary. Instead: Build your object. Write CSS for the 3 containers. Call HoverAlls and assign your styles using the settings on the previous page.

Movement

Coordinates

Once you understand HoverAlls coordinates you can move your objects around within your container. Later on we'll get to the specifics about how to call HoverAlls and pass these coordinate settings into the HoverAlls call (just like any other jQuery plugin). For now we'll just be using the settings for demonstrative purposes. The Coordinates System uses a standard X,Y format.

Moving Objects

To move the Background Container (remember it's the gray one), we first need to setup starting coordinates. This setting does just that:

```
starts : “0px, 200px”
```

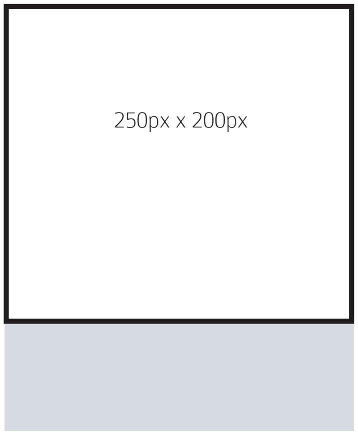
The above sets up the starting coordinates and positions the Background Container (in CSS terms) - left: 0px; top:0px;. Because it's relative to the HoverAlls container, the Background Container is positioned as pictured in Figure A. Next we need to supply ending coordinates which the Background Container will move to. This setting does that:

```
ends : “0px, 150px”
```

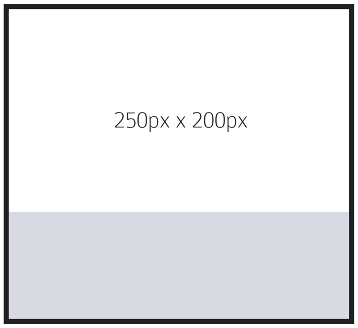
This will animate your Background Container to the position in Figure B. There's one last set of coordinates: Return Coordinates. Usually these will be exactly the same as the Start Coordinates, unless you want your Background Container to move to a different location on animation exit (mouseout). The Return Coordinates below will move the Background Container to the position in Figure C.

```
returns : “25px, 50px”
```

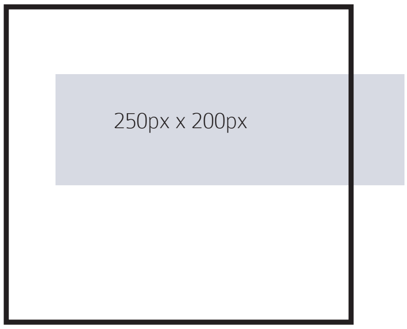
A.



B.



C.

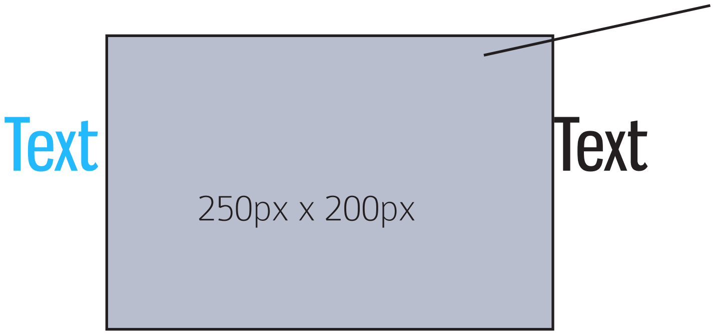


Controlling Text

Controlling text is as easy as controlling the Background Container... it's actually done the exact same way (and for the most part with the same exact settings - just with "text_" in front of them). The first thing to do is setup your Background Container.

Use the bg_width and bg_height settings to tell HoverAlls the dimensions of your Background Container. Every HoverAlls Call should include these 2 settings along with your 3 Coordinates settings.

We start by positioning the text. This is done using the text_starts, text_ends and text_returns settings. They function the same as the coordinates settings discussed on the previous page. Below is a diagram to assist with visualizing how HoverAlls controls movement for text.



We want the Background Container to stay in place while text animates across the top. We'll set the dimensions: bg_width : "250px", bg_height : "200px". We'll also set the start, end and return coordinates for the Background Container all to 0px,0px. This doesn't move the Background Container.

Now that the Background Container is locked in place and won't be animated, we can setup the start, end and return coordinates for the text. We want the text to move right to left - so the start settings are: text_starts : "250px,100px". This places the Text Container to the right-outside of the Background Container (as shown in black). The end coordinates are text_ends: "-100px,100px" (shown in blue). You can optionally set Return Coordinates using the same method.

Links and Text

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Now that you know what containers are available to you and how to position and move these containers, it's time to cover links and text. HoverAlls handles links and text from images differently than from other elements.

For Images:

```
<img src="path/to/image.jpg" width="250px" height="200px" alt="This is my HoverAlls text that gets placed inside the Text Container" rel="http://www.mylink.com" id="myimage"/>
```

Images are simple. Place your HoverAlls text inside the ALT attribute and the link inside the REL attribute, as shown above. The text will automatically be moved into your HoverAlls Text Container so you can manipulate it using HoverAlls settings. By default, the link will wrap around your Background Container. You can set the **big_link** setting to true to have the link wrap around your entire HoverAlls Container. You can also use the **link_target** setting to specify "_parent" or _blank". (set to "_blank" by default)

For Other Elements:

```
<div id="aDiv">
        <p class="hoveralls_text">This my my HoverAlls text that gets placed inside the
Text Container.</p>
        <a href="http://www.mylink.com" class="hoveralls_link"></a>
</div>
```

For other elements, assign the '**hoveralls_text**' class to a < p > tag inside your element. HoverAlls will treat your text the same as it does for images. Add the '**hoveralls_link**' class to an < a > inside your element to have HoverAlls use that link. Link settings function the same for other elements as they do for images.

Calling HoverAlls

Captions. Lightboxes. Tooltips. Panels. Scrollers.

The beauty of HoverAlls is you build your HTML and CSS the way you’re used to. Below is a basic example of how to call HoverAlls. (see the tutorials for full examples). The image has an ID of myImage - so we will be passing HoverAlls settings to the #myImage object as shown.

HTML

```
<img src=”path/to/image.jpg” width=”250px” height=”200px” alt=”This is my HoverAlls text
that gets placed inside the Text Container” rel=”http://www.mylink.com” id=”myimage”/>
```

HoverAlls

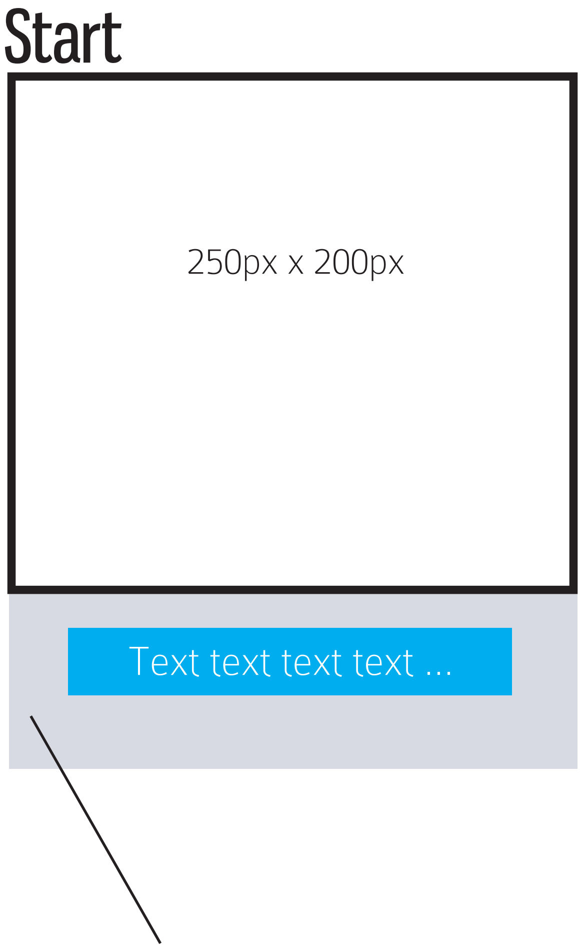
```
$(‘#myimage’).HoverAlls({
        bg_width : “250px”,          Set the width of the Background Container.
        bg_height : “70px”,          Set the height of the Background Container.
        speed_in : 500,              Speed on Enter
        speed_out : 1200,            Speed on Exit
        starts : “0px,200px”,        Starting position (from left of HoverAlls Container)
        ends : “0px,130px”,          Ending position (from left of HoverAlls Container)
        returns : “0px,250px”,       Return position on exit (from left of HoverAlls Container)
});
```

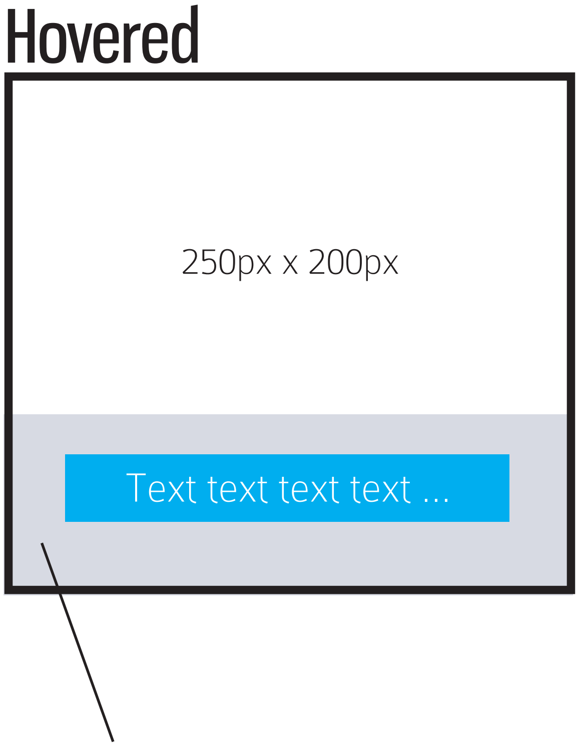
You always need to use the 3 Coordinates settings, bg_width, bg_height and assign your classes.

View the next page to see the results.

If you’re unfamiliar with using jQuery plugins, pay close attention to apostrophes, commas and curly brackets. If the setting value is true/false or a number (like speed_in), it is NOT surrounded by quotes. The last setting does not receive a comma.

The example from the previous page would appear as below:





You've told HoverAlls to make the Background Container 250px wide and 70px tall. You've also positioned it (relative to the HoverAlls Container) at **starts:"0px,250px"** - (left: 0px, top: 250px) .. placing it out of view.

The **ends setting** controls where the Background Container is moved to (relative to the HoverAlls Container). You've specified **ends:"0px,130px"**. We don't want the Background Container to move left or right, so the X value is left at 0px. HoverAlls will animate the Background Container to this position in 500 milliseconds... on mouse exit it will animate back to it's original position in 1200 milliseconds.

Part 2

Captions. Lightboxes. Tooltips. Panels. Scrollers.

If you don't understand how to use HoverAlls Coordinates to move containers...

If you don't know what 3 containers HoverAlls builds for you...

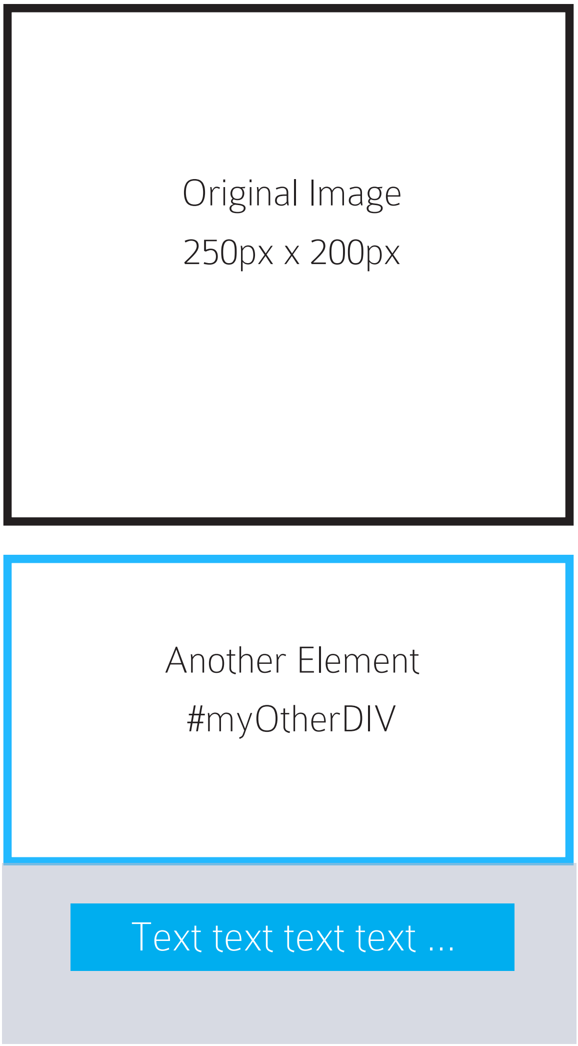
Reread the previous sections.

Target Containers

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Target Containers are a powerful feature of HoverAlls (and one which many other settings rely upon). A target container is exactly what it sounds like: a place where you want the animation to occur. This means you can set up other DIVs on your page that the animation will occur inside. How about hovering an image to slide in a panel? Or clicking an image and opening a tooltip on another container? It's all done with the **target_container** setting.

Using Target Containers



Using the example from part 1, we'll add a target container. All of the code is exactly the same, except one extra setting inside the HoverAlls call:

```
target_container : “#myOtherDIV”
```

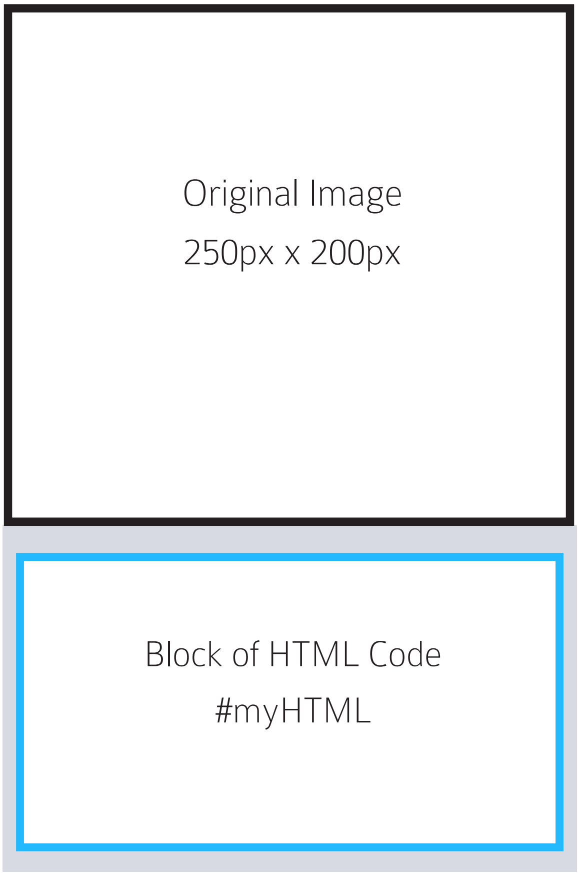
Placing this extra setting into the HoverAlls call will automatically seek out an element with the ID of myOtherDIV and move the animation containers into that element. The user will now hover over the original image and trigger the slideup/caption example inside of the #myOtherDIV container. Keep in mind that the **bg_width** and the **bg_height** may need to be adjusted if the width and height of your target container is different than the original image.

HTML Mode

Captions. Lightboxes. Tooltips. Panels. Scrollers.

HTML Mode is another powerful feature of HoverAlls, and is slightly similar to the **target_container** setting. HTML Mode will take your written block of HTML (anywhere on the page) and move it inside of your Background Container (even if it's in a Target Container).

Using HTML Mode



Using the same example, we'll add the HTML Mode setting and supply it with the ID of a block of HTML we want moved into the Background Container. All of the code is exactly the same, except one extra setting inside the HoverAlls call:

```
html_mode : "#myHTML"
```

This tells HoverAlls to find the element with the ID of myHTML and move it inside of your Background Container. If you have a **target_container** set as well, it will move #myHTML into the Target Container's Background Container. HoverAlls then treats your entire block of HTML as the Background Container and you're free to use the settings to animate it.

You can add the '**hoveralls_text**' class to a < p > tag inside #myHTML and use the text settings to animate your text. You can also add the '**hoveralls_link**' class to an < a > inside #myHTML to use it with the HoverAlls link settings. **NOTE: Moving your block of HTML into a different container may change your #myHTML styling depending upon how you wrote your CSS. Some CSS adjustments may need to be done. If you use the big_link setting with HTML_mode on, your original link is automatically removed.**

Part 3

Captions. Lightboxes. Tooltips. Panels. Scrollers.

If you don’t understand target containers ...

If you don’t understand HTML Mode ...

Reread the previous section.

You now understand how to install HoverAlls, how it works, the 3 containers that are being animated and how to call HoverAlls. The next section is a set of mini-tutorials. Keep in mind that the 3 containers are used throughout all of these situations, and that everything is built upon the information provided in sections 1 and 2. You’ll soon be able to experiment and play with your own HoverAlls techniques.

Captions Tut

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Finally, time to start working with some animation! Captions are the easiest element to build, so we’ll start with those. This tutorial will show you how to build a slide-up caption using HoverAlls. You can slide the caption in any way you want by using the coordinates (discussed on page 6). For simplicity of the tutorial, we’ll be using an image. If you want to use a different element, the HoverAlls settings remain the same. The only difference is how you supply the text and link (covered on page 8).

Build a Slide-Up Caption on an Image

Step 1.

First start by creating an image in HTML. Make sure you place your caption text inside the ALT and your link inside the REL.

```
<img src=”path/to/image.jpg” width=”250px” height=”200px” id=”myCaption” alt=”This is
my Caption Text”  rel=”http://www.codecanyon.net” />
```

Step 2.

If you want your image to have any styling, add it to the CSS now. Also, create a style for the caption background and the caption text.

```
#myCaption { width:250px; height:200px; ... }

.captionbackground { width:250px; height:50px; background-color: #000; }

.captiontext { color:#FFF; font-size: 16px; ... }
```

NOTE Using COPY and PASTE from this document can result in syntax errors.

Step 3.

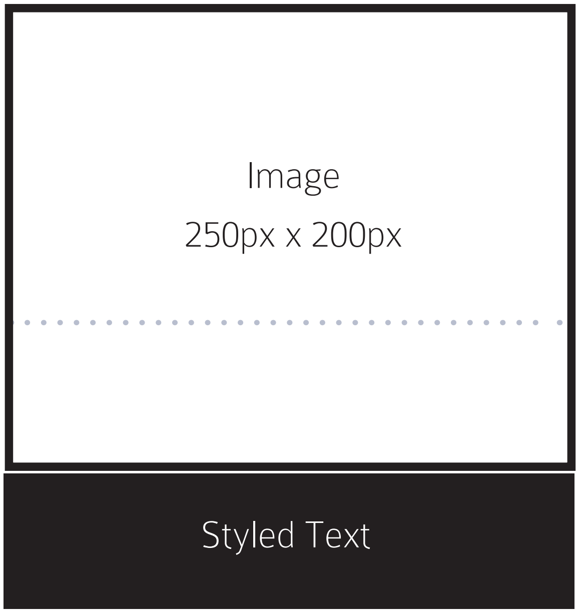
On your webpage (preferrably at the bottom) you will want to create a script tag to place the HoverAlls calls inside. We're using the window.load method rather than the document.ready method because it waits until all images are completely loaded before executing the script.

```
<script type=”text/javascript”>
        $(window).load(function () {
                ... all your HoverAlls calls ...
        }; </script>
```

Then we call HoverAlls and configure the animation inside the window.load function.

```
<script type=”text/javascript”>
        $(window).load(function () {
$(‘#myCaption’).HoverAlls({
        speed_in : 500,
        bg_width : “250px”,
        bg_height : “50px”,
        starts : “0px, 200px”,
        ends : “0px, 150px”,
        returns : “0px, 200px”,
        bg_class : “captionbackground”,
        text_class : “captiontext”
});
}); </script>
```

HoverAlls Call



The styles have been assigned to the Background Container and Text Container. The position is setup (hidden) below the image. The End position is setup to end the caption where the dotted gray line is. If you can’t see your container, refer to the Movement page for positioning.

Creating Tooltips with HoverAlls is so simple you'll be wondering why you've been leaving these helpful bad boys out of your previous projects. The steps are similar to a caption. Again, refer to page 6 (Movement) for different positioning and to page 8 (Links and Text) for elements other than an image.

Build a ToolTip on an Image

Step 1.

First start by creating an image. Make sure you place your tooltip text inside the ALT and your link inside the REL. If you use the **big_link** setting on a tooltip, your image as well as the tooltip will be wrapped in your link.

```
<img src="path/to/image.jpg" width="250px" height="200px" id="myToolTipImage"
alt="This is my ToolTip Text"  rel="http://www.codecanyon.net" />
```

Step 2.

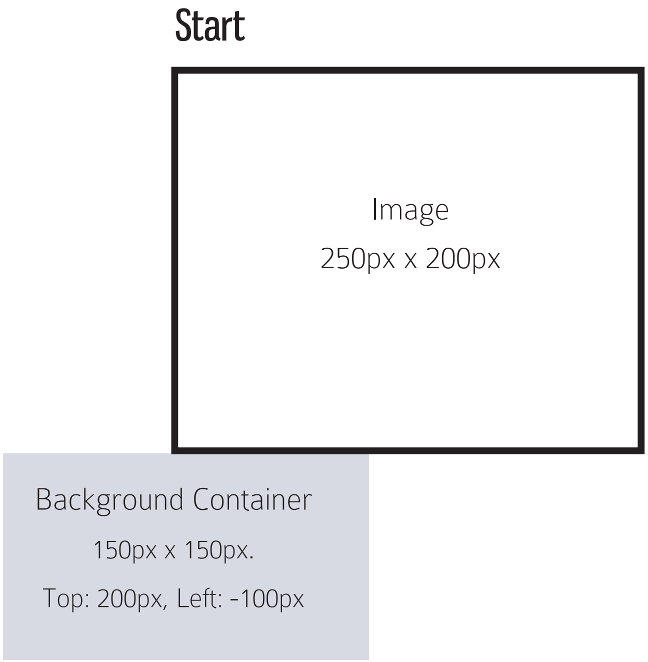
If you want your image to have any styling, add it to the CSS now. Also, create a style for the ToolTip background and the ToolTip text. (starting to sound familiar yet?)

```
#myToolTipImage { width:250px; height:200px; ... }

.tooltipbackground { width:250px; height:50px; background: #fff; border: 2px #000; }

.tooltiptext { color:#000; font-size: 16px; ... }
```

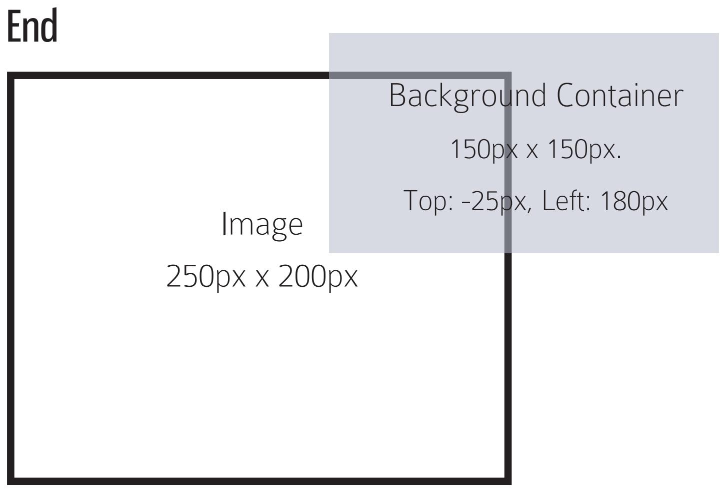
Step 3.

Last, call HoverAlls to configure your animation (don’t forget to use the same script tag setup as shown in the Captions tutorial). Notice we set tooltip : true

```
$(‘#myToolTipImage’).HoverAlls({
        bg_width : “150px”,
        bg_height : “150px”,
        starts : “-100px, 200px”,
        ends : “180px, -25px”,
        returns : “-100px, 200px”,
        bg_class : “tooltipbackground”,
        text_class : “tooltiptext”,
        container_class : “tooltipcontainer”,
        tooltip : true
});
```



NOTE: You can also optionally use HTML inside of the tooltip using the HTML_mode setting. Just write a block of HTML, give it an ID, style it and call it by passing the ID into the HTML_Mode setting. Some settings may not be compatible with Tooltips.



Lightboxes / Modals

Captions. Lightboxes. Tooltips. Panels. Scrollers.

There are three "popup" settings: **lightbox**, **modal** and **alertbox**. **Modals** are simple popup boxes with text, logins, etc. Using the **modal : true** setting, HoverAlls retrieves your text and link, and places them inside a popup box. **Lightboxes will do the same, but automatically load your link as the background image. With Lightboxes, the REL attribute is a link to the large version of your image.** Alert boxes are exactly the same as Modals, but are automatically centered on the screen. All 3 popups are compatible with the **HTML_Mode** setting and run on Target Containers.

Build a Lightbox on an Image

Step 1.

First start by creating an image. Make sure you place your lightbox text inside the ALT and your link inside the REL. If you use the **big_link** setting on a lightbox your entire lightbox container is wrapped in your link.

```
<img src="path/to/image.jpg" width="250px" height="200px" id="myLightboxImage"
alt="This is my Lightbox Text"  rel="/link/to/large_image.jpg"/>
```

Step 2.

Create your actual lightbox container. This is the Target Container that HoverAlls will place the animation inside. Place this anywhere on the page. **You do not need to apply any CSS styles to this container now.**

```
<div id="lightboxtarget"></div>
```

... Lightboxes

Captions. Lightboxes. Tooltips. Panels. Scrollers.

Step 3.

Style your image in the CSS as you normally would. Create a class for your lightbox overlay (background), lightbox container, lightbox background, and lightbox text, as shown below.

```
#myLightboxImage { width:250px; height:200px; ... }
.lightboxcontainer { width:450px; height:650px; background: #fff; border: 2px #000; }
.lightboxoverlay { background-color: #000; ... }
.lightboxtext { color:#000; font-size: 16px; ... }
.lightboxbackground { background-color: #CCC; border:1px dotted #333; }
```

Step 3.

Call HoverAlls the same way. Notice we set lightbox : true.

```
$('#myLightboxImage').HoverAlls({
        bg_width : "450px",
        bg_height : "650px",
        starts : "-450px, -650px",
        ends : "350px, 100px",
        returns : "-450px, -650px",
        overlay_class : "lightboxoverlay",
        container_class : "lightboxcontainer",
        bg_class : "lightboxbackground",
        text_class : "lightboxtext",
        target_container : "#lightboxtarget",
        lightbox : true
});
```

By default, the HoverAlls coordinates will precisely control the start, end and return points of the lightbox itself. You can use the lightbox_center setting to automatically center the lightbox. Use the alertbox setting to convert your lightbox into an alert message, or the modal setting for easy popups.

Add the .closelightbox class to any HTML element to have it function as a button to close a HoverAlls popup. By default, clicking the overlay closes HoverAlls popups.

Below is an illustration of the lightbox (gray) we just setup. You can see that the **bg_width** and **bg_height** act as they normally do (controlling the width and height of the Background Container. The **starts** setting: X value is -450px (since the lightbox is 450px wide - and 0px is the left side of the screen - we want to place it entirely offscreen). The return position tells us it's returning to it's starting coordinates. The overlay (pink) received our style.

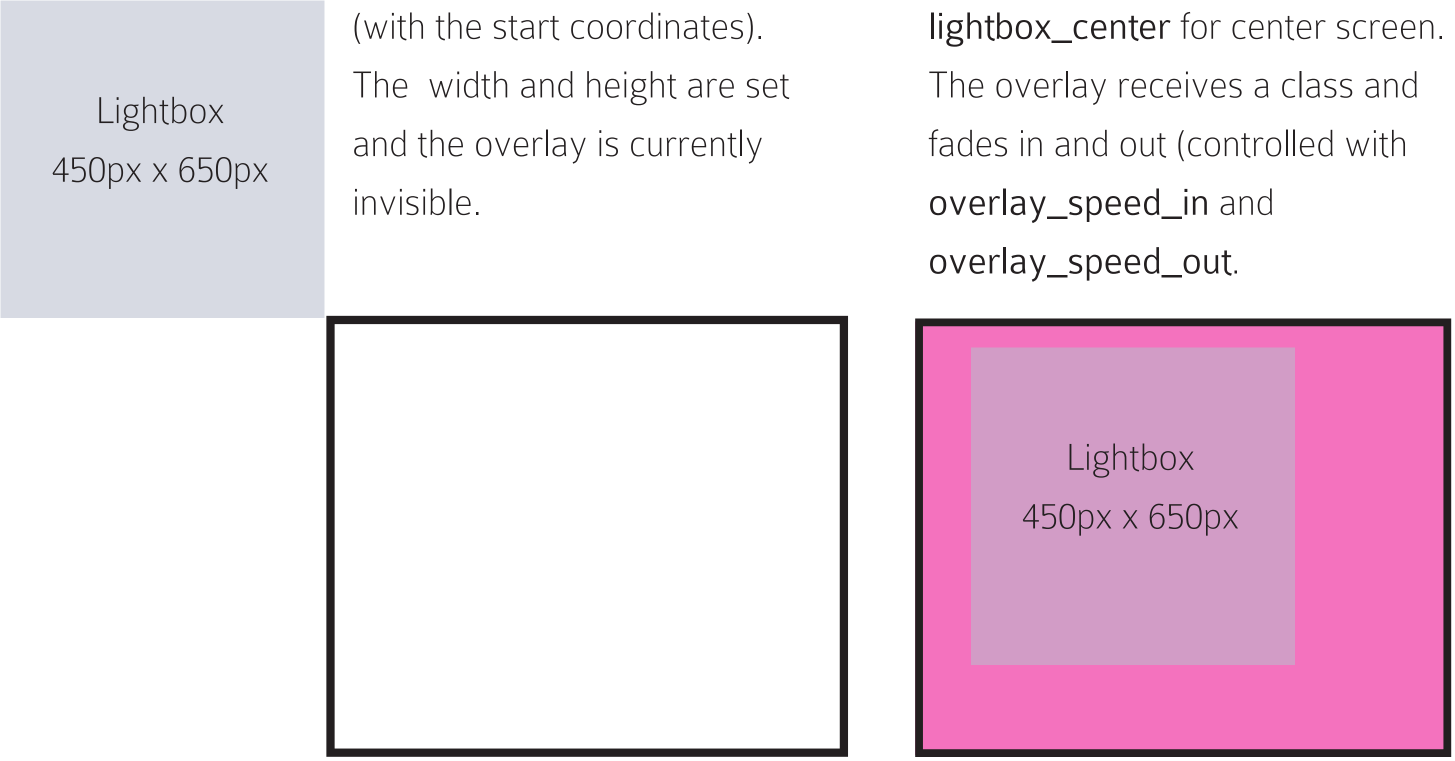
If you don't see the overlay you haven't assigned a class.

Start

The lightbox is setup to be positioned offscreen at start (with the start coordinates). The width and height are set and the overlay is currently invisible.

End

The end coordinates position the lightbox wherever you like. Use **lightbox_center** for center screen. The overlay receives a class and fades in and out (controlled with **overlay_speed_in** and **overlay_speed_out**.



For Lightboxes, Modals, Alerts and Sliding Panels, the coordinates settings are relative to the page.

Use HTML Mode to insert all sorts of fun things into your HoverAlls lightbox.

Scrollers

Scrollers can provide a ton of functionality, and with HoverAlls you can quickly place them practically anywhere (using the **target_container** setting). If you're planning on skipping this section, remember that the 3 containers are all being affected by the HoverAlls settings in different ways... coincidentally, creating text animations is quite similar.

Build a Scroller on an Image

Step 1.
Setup the image as usual.

```
<img src="path/to/image.jpg" width="250px" height="200px" id="myScroller" alt="This is
my Scroller Text"  rel="http://www.codecanyon.net" />
```

Step 2.
Setup any CSS you want for the image. The Text Container holds the scroller, so any styles applied to the Text Container will affect the scrolling text.

Step 3.
Setup the HoverAlls settings as shown on the next page. Notice we've initiated the scroller by using the **ticker_mode** setting. Sometimes tickers will appear jerky. If your ticker does this, make sure that you do not have an easing effect, or competing speeds for the Background Container and Text Container.

... Scrollers

```
$('#myScroller').HoverAlls({
        text_speed_in : 5000,
        bg_width : "250px",
        bg_height : "250px",
        starts : "0px, 0px",
        ends : "0px, 0px",
        returns : "0px, 0px",
        text_starts : "250px, 15px",
        text_ends : "-500px, 15px",
        text_returns : "250px, 15px",
        bg_class : "scrollerbackground",
        text_class : "captiontext",
        ticker_mode : true,
        ticker_repeat : 5
});
```

Animations are controlled with the **speed_in** and **speed_out** settings, while text is controlled with the **text_speed_in** and **text_speed_out** settings. You may need to use very large numbers in the **text_speed_in** setting to force it to act as a crawler. Text is positioned using coordinate settings similar to positioning the Background Container. All that is happening is we are moving the Text Container from the right side of the Background Container to the left side (250px to -500px). The Xpx value of the **text_ends** setting - will be dependent upon how much text you have in your container. If you have a lot of text, the Text Container will be wider **(so you will have to move it farther and increase the text_speed_in number to slow down the animation). Ticker_mode** is set to true to activate it as a scroller. This example uses **ticker_repeat** to repeat the ticker text (highlighted in yellow on the previous page) 5 times.

Make sure to include a *word-space : nowrap* in your CSS in order to keep the text on 1 line.

Use the .ticker_spacer class to style the space between repeated text. It should be possible to use ticker_mode with a CMS loop as long as you accomodate for the width of the Text Container. Refer to page 7 (Text Movement) for an illustration of exactly what is happening.

Sliding Panels

Sometimes Sliding Panels can be a pain in the mouse to create, but they're useful for all types of sites. HoverAlls makes creating sliding panels so easy, you'll be abusing them in no time. Sliding panels use the top_drop setting, but remember that they're just Target Containers.

Build a Sliding Panel on an Image

Step 1.

First, we will create our image as usual.

```
<img src="path/to/image.jpg" width="250px" height="200px" id="myPanel" alt="This is my Panel Text" rel="http://www.codecanyon.net" />
```

Step 2.

Next, as usual, apply any CSS you want to your image. Then create an empty DIV anywhere on your page. This will ultimately be the Target Container for the sliding panel. We'll make a social media icon bar sliding panel (from the left of the screen).

```
<div id="social"></div>
```

Step 3.

Apply some CSS to the #social div. A good trick is to set the background color to black, then position it exactly where you want it, then remove the color. Below is the CSS to position the container at the left of the screen. You can position a sliding panel wherever you like, just make sure your HoverAlls call is starting, ending and returning to the correct coordinates and it has the correct bg_width and bg_height.

```
#social { position:fixed; z-index:99999; width:100px ; left: -100px; height:600px; top:250px; }
```

Sliding Panels

```
$(‘#myPanel’).HoverAlls({
        bg_width : “100px”,
        bg_height : “600px”,
        starts : “-100px, 250px”,
        ends : “0px, 250px”,
        returns : “-100px, 250px”,
        container_class : “panelcontainer”,
        text_class : “paneltext”,
        top_drop : true,
        target_container : “#social”,
        on_click : true
});
```

If you want to place more than 1 sliding panel on a single page, take a look at the HoverAlls CSS file. Duplicate the **#top_drop** style and rename it to your new container. Make sure you include <div id=”____”></div> somewere on your page.

Notice that **top_drop** is set to true, indicating it’s a sliding panel. We’re also activating this panel when the user clicks the #myPanel image by using the **on_click** setting. Compare the #social CSS to the HoverAlls settings to see the similarities.

```
#social { position:fixed; z-index:99999; width:100px ; left: -100px; height:600px; top:250px; }
```

This will slide a panel in from the left of the screen (similar to the demo page).

BONUS: Mobile Devices

A BONUS! feature is included inside of the jquery.hoveralls.js file. If you want your HoverAlls effect to happen a specific way on mobile devices (like using clicks instead of hovers) then you can write specific settings. This requires some knowledge of Javascript.

Special Settings for Mobile Devices

Open jquery.hoveralls.js and search for the BONUS Mobile Device section toward the top of the document. You'll notice a commented section inside that block that says

```
/* YOUR CODE HERE */
```

Write your HoverAlls settings inside this section in the following format:

o.setting = "value"; o.setting = false; o.setting = 500;

Currently, HoverAlls will automatically detect and apply these settings on the following:

Android, webOS, iPhone, iPod, BlackBerry and Windows Phone OS 7

BONUS: Presets

If you find yourself repeating a set of HoverAlls settings to achieve the same effect again and again, or you'd like to save your effect for a future website, HoverAlls comes with 3 presets. Open the jquery.hoveralls.js file and search for the BONUS Presets section. Pass in your settings in the same format that you would for the mobile device section. If you've saved your preset in slot 1, then all you have to do in the HoverAlls call is:

```
$('#myDIV').HoverAlls({ preset1 : true });
```

Keep in mind that effects will appear differently when you apply them to objects with different dimensions, and you will most likely need to set the bg_width and bg_height.

Experimentation

Captions. Lightboxes. Tooltips. Panels. Scrollers.

So, we've walked you through everything and it all comes back to those 3 containers. That's what HoverAlls is all about, and that's what makes it easy. If you're having any difficulties hang in there, and realize that once you figure out what HoverAlls is doing when it's making your Captions, you'll understand how it's creating everything else. The main settings to understand are the bg_width, bg_height and the 3 sets of coordinates. You'll use those settings on every call for precise animation movement.

PassThrough Mode

PassThrough Mode can be confusing, and somewhat tricky. It affects the Text Container, and resets the text_returns setting back to the original coordinates of text_starts. Why? Well, if you want the text to exit on the other side, then reappear on the original... use the passthrough setting.

I sincerely hope you enjoy HoverAlls and find it to be easy to use. Apologies for the lengthy documentation, I just wanted to make sure that nobody was left out. This is a plugin designed for beginners and advanced designers, and once mastered, you can trim hours off of your production time by plugging in HoverAlls to take care of all the little stuff. Most importantly, EXPERIMENT with HoverAlls. It's open-ended, screw with it as much as you can and don't forget to suggest ideas for version 2.

You're almost ready to go, but before you do, you should check out the full list of settings at the end.

FAQ / Troubleshooting

Captions. Lightboxes. Tooltips. Panels. Scrollers.

FAQ

How can I delay an animation?

Use the starting coordinates to position your container further away.

I can’t see my lightbox overlay!

By default the overlay container is empty. Apply a background image or color with your class using the overlay_class setting.

My animation is jerky, why?

If your animation appears jerky the first thing to check is the easing. If you’ve applied effect_in or effect_out then remove these settings and test. If it still appears jerky, leave the effects out and try adjust the speed slightly.

My scroller is speeding up and slowing down?

If your scroller slows down at the end or has inconsistent speeds, it’s most likely because easing effects are applied. Remove the effects and make sure the text_ends is set far enough to the right.

I keep getting syntax errors from copying text from the manual!

Formatting issues are affecting the apostrophes. You can manually type in the code, or retype the apostrophes and quotes.

It’s not working!

If nothing appears to be happening, check to make sure your javascript files are being loaded. A simple trick for this is viewing the source code and clicking the link to your javascript file. If it’s linked correctly it should show you the file text.

Intro to jQuery

Captions. Lightboxes. Tooltips. Panels. Scrollers.

jQuery

This page is an extra for anyone interested in a little jQuery.

Selectors

Similar to CSS, jQuery uses selectors to first target an element to work with.

$('h1')	Selects all H1 tags on the page.
$('#myDiv')	Selects div with ID of myDiv.
$('.myClass')	Selects objects with class of myClass.

Commands

Once an object is selected you can run commands on it.

$('h1').remove();	Removes all h1 tags.
$('#myDiv').fadeIn(500);	Fades in #myDiv in 500 milliseconds.

There's obviously a lot more to it than the above. However, if the above makes any sense you shouldn't hesistate diving into some jQuery.

Settings

Captions. Lightboxes. Tooltips. Panels. Scrollers.

We've covered some of the HoverAlls settings, but I thought I'd leave the rest to those that aren't afraid to learn. Everything covered to this point has been the barebone necessities, the rest is all fun.

General Settings

Setting	Default Value	Description
starts	" Xpx, Ypx "	Starting Position for the BG Container. (relative to HoverAlls Container).
ends	" Xpx, Ypx "	Ending Position for BG Container. (relative to HoverAlls Container).
returns	" Xpx, Ypx "	Return Position for BG Container. (sets new point to move to on exit).
big_link	false	Set to true to wrap your entire element inside of your link.
full_frame	false	Set to true to make the Background Container to 100% wide x100% high.
center_text	false	Centers text vertically in Full Frame mode.
speed_in	380	Speed of Background Container on enter. (in milliseconds)
speed_out	500	Speed of Background Container on exit. (in milliseconds)
effect_in	"swing"	Transition on enter. If Easing is installed you can use Easing Effects here.
effect_out	"swing"	Transition on exit. If Easing is installed you can use Easing Effects here.
start_opacity	0	Starting opacity of the Background Container.
end_opacity	1	Ending opacity of the Background Container.
bg_width	" 0px "	Width of Background Container (important for some features to work).
bg_height	" 0px "	Height of Background Container (important for some features to work).
on_click	false	Set to true to trigger your effect on click.
call_on_start	false	Set to true to trigger your effect on page load.
link_target	"_blank"	Decide whether a link opens in a new tab, window or itself.

Special Settings

Setting	Default Value	Description
text_starts	“ Xpx, Ypx ”	Starting Position for the Text Container. (relative to Background Container).
text_ends	“ Xpx, Ypx ”	Ending Position for the Text Container. (relative to Background Container).
text_ returns	“ Xpx, Ypx ”	Return Position for the Text Container. (sets new point to move to on exit).
text_ speed_in	1000	Speed of Text Container on enter. (in milliseconds)
text_ speed_out	1000	Speed of Text Container on exit. (in milliseconds)
text_ effect_in	“swing”	Text Transition on enter. If Easing is installed you can use Easing Effects here.
text_ effect_out	“swing”	Text Transition on exit. If Easing is installed you can use Easing Effects here.
text_ start_opacity	1	Starting opacity of the Text Container.
text_ end_opacity	1	Ending opacity of the Text Container.
passthrough	false	Resets the Text Container’s Return Points to the Start Points.
target_container	false	Supply an HTML ID to place the HoverAlls animation inside of that element.
html_mode	false	Supply an HTML ID to place that code inside the HoverAlls animation.
container_class	false	Add your class to the HoverAlls Container.
bg_class	false	Add your class to the Background Container.
text_class	false	Add your class to the Text Container.
overlay_class	false	Add your class to the Lightbox Overlay.
ticker_mode	false	Convert the Text Container into a scroller.
ticker_repeat	1	Number of times to repeat the HoverAlls text inside a scroller.
top_drop	false	Create a Sliding Panel from the top of the screen.
tooltip	false	Convert your animation into a ToolTip.
mobile	false	Use special settings for mobile devices (JS knowledge required).
lightbox	false	Use your animation as a HoverAlls Lightbox.
lightbox_center	false	Keep the Lightbox centered on screen.
alertbox	false	Use your animation as a HoverAlls Alertbox.
overlay_speed_in	800	Control the time to fade in the lightbox overlay (in milliseconds).
overlay_speed_out	500	Control the time to fade out the lightbox overlay (in milliseconds).
modal	false	Create a simple popup box

Notes

Notes

When applying classes with container_class, you may need to use the !important; hack in order to force the style on the container.

The full_frame setting does not work on target containers.

The center_text setting only works with the full_frame setting.

HoverAlls cannot inject HTML into multiple containers or select multiple target containers.

Make sure you set the bg_width, bg_height and 3 sets of coordinates for all HoverAlls calls.

When applying classes with container_class, you may need to use the !important; hack in order to force the style on the container.

v2

Version 2 is underway and will primarily focus on extending the features of the lightbox. The scroller is set to have some revisions (namely auto detecting the new Text Container width when ticker_repeat is on). If you have any ideas for version 2 please let me know and I will do my best to work them in.