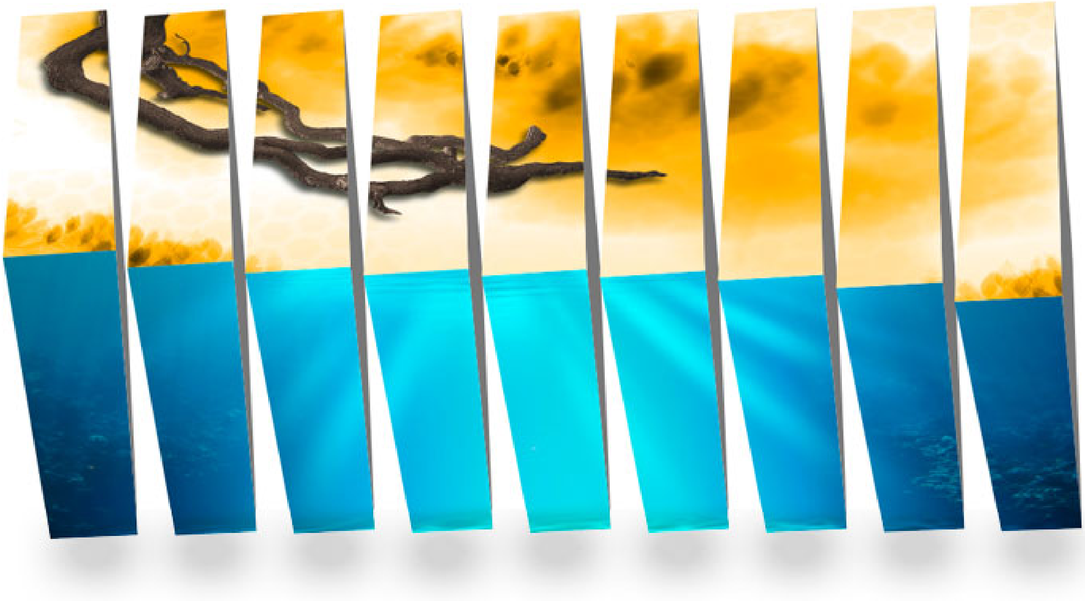

LayerSlider

3D Responsive jQuery Slider Plugin

Version 4.5.0

Documentation

Overview

LayerSlider is the most advanced Responsive jQuery Slider Plugin with the famous Parallax Effect and 3D Transitions! You can use LayerSlider to create a breathtaking Image Slideshow, Banner Rotator or Content Slider to your site!

Thank you

Thank you for purchasing this product. If you have any questions that are beyond the scope of this documentation, please feel free to email us.

Features

- Over 200 breathtaking 3D & 2D hardware accelerated transitions with transition gallery!
- Transition gallery and option to create your own custom transitions!
- 5 displaying modes (normal, responsive, full-width, full-width + responsive & full-width + responsiveUnder. Smart resizing and repositioning image, video and also text sublayers in responsive modes including font-sizes, line-heights, etc.)
- Tons of (mostly unique) features (auto-play and auto-preview images for YouTube and Vimeo videos, auto-pause slideshow if videos are playing, pause on hover, image preload, loops, linking sublayers to another layer or to any url, yourLogo feature with link, deep-linking, random start, random-slideshow, etc.)
- Unlimited types of usage (image slider, image slider with text, content slider, video gallery slider, mixed content slider, etc...)
- 3 types of navigation controls (thumbnail mode, hover thumbnails with buttons & buttons only) with auto hide feature
- Unlimited variations of hardware accelerated animations (sliding from / to 4 directions or fading in / out (sublayers only) with countless easing types, optional rotation and scaling, and sublayers can slide or fade out before other sublayers appear)
- Use with ANY HTML content (including images, embedded videos, text, forms, flash content and anything you need!)
- Supports all major browsers (including IE7)
- Skin editor
- Sample sliders included

- Touch-control for mobile devices
- Keyboard navigation
- Built-in powerful API
- Multiple sliders on one page
- Very detailed documentation
- SEO friendly
- Skin support with 12 skins
- Layered skin files included in .PSD format
- Free support and future updates

Changelog

4.5.0

- New features
  - Added 4 new skins
  - Added 41 additional 2D transitions with semi-3D effects
  - Added bar and circle timers
  - Added scale and rotate options for sublayers animations
- Improvements
  - Major performance improvements
  - The new transitions now can animate to the opposite direction depending on the user interaction and the "two way slideshow" feature
  - Responsiveness now can handle images with custom dimensions
  - A full-width slider now will enlarge the background image if needed
  - In older versions of Internet Explorer we are now allow fade effects
  - Improvements in the LayerSlider Debug Mode
- Bug fixes
  - Fixed the "link the whole slide" issue under older versions of Internet Explorer
  - Fixed an issue which resulted the loading indicator to appear during slide changes
  - Several fixes with the navigation area
  - Several fixes with the old and new transitions

4.1.0

- Fixed video issues in Firefox
- Fixed an issue that caused slowness in the transition gallery
- Fixed an issue with the "animateFirstLayer" that caused the slideshow to stop
- Fixed an issue with the cbAnimStop if there is only one layer
- Added jQuery version number check
- Fixed an issue that caused the LayerSlider notices to embed each other
- Fixed some style settings
- Improved documentation

4.0.1

- Improved CSS overrides to prevent the theme applying global CSS style settings on the slider and its elements
- Fixed an issue which caused the slider to freeze if the navigation area was set to hidden

4.0

- New features
  - We added new transition feature with more than 200 predefined 2D and 3D transitions
  - We are now using hardware acceleration to give you killer performance on mobile devices
  - Added loading indicator
- Improvements
  - Improved "Animate first layer" feature, it doesn't animate the background now
  - We are now support the value "-1" for distance level to position your sublayers exactly outside of the layer (class="ls-s-1")
  - You can add percentage values of the width and height of your sublayers

    - Full jQuery 1.9 compatibility
    - Improved various style settings

- Bug fixes

    - Fixed an issue in linked sublayers with fade effect which resulted the sublayer disappearance
    - Fixed an issue on mobile devices regarding the thumbnail images touch detection
    - Fixed an issue in the LayerSlider API when the "data" object wasn't passed for some callback functions
    - Fixed an issue in the cbAnimStop() callback function of the LayerSlider API which contained some invalid values regarding the slides

3.6

- Improvements

    - Same skin for more slider will be loaded only once
    - Improved LayerSlider init code
    - Improved style settings in layerslider.css
    - Improved Developer Mode

- Bug fix

    - Fixing a bug in always thumbnail mode

3.5

- New features

    - Added two types of fully customizable thumbnail navigation
    - Ability to set custom thumbnail image for thumbnail navigation
    - Added three new skins
    - Thumbnail navigation added to all (new and old) skins
    - Added responsiveUnder feature for full width sliders (slider will switch into responsive mode when you reduce the screen size under a given size)
    - Added sublayerContainer feature for full width sliders (sublayers are positioned inside a centered container)
    - Added randomSlideshow feature
    - Added hoverPrevNext feature (prev and next buttons will shown only on hover)
    - Added hoverBottomNav feature (bottom navigation will shown only on hover)
    - Added debug mode

- Improvements

    - Changes in the sample sliders and added more examples
    - Huge performance improvements
    - Better handling of deep linking, the page won't scroll away now

- Bug fixes

    - Fixed slideDelay when animateFirstLayer is enabled
    - Fixed slideIn and slideOut animations when there is only one layer in the slider
    - Fixed yourLogo percentage positioning
    - The slider won't fail to load when the skin isn't available now
    - More minor changes & bugfixes

3.0

- Added smart resizing in responsive modes: resizing and repositioning image, video and also text(!) sublayers in responsive mode including font-sizes, line-heights, etc.)
- showUntil feature (sublayers can now fade / slide out before other sublayers appear)
- Deep linking feature
- Option to start with a random layer
- Flickering when the slider loading is now fixed
- Modified LayerSlider API make easier your job if you want to create something unique

2.0

- added loops feature
- added autoPlay feature for embedded YouTube and Vimeo videos
- added autoPauseSlideshow feature while videos are playing
- added auto-preview image feature to YouTube and Vimeo videos

- added linkTo feature
- included two new skins
- touch control now can be disabled
- changed one layer mode (not duplicating it now)
- fixed a wrong variable
- the bottom side of the slider (next to the controls) is now clickable
- a lot of minor bugfixes and improvements

1.8

- fixed some animations
- added background-images of layers to preload
- modified some CSS styles
- some parts of the source code rewritten
- added yourLogoLink feature
- added yourLogoTarget feature
- added noskin feature

1.7

- fixed fading bug (sublayers now appears if sliding in after fading out)
- fixed preload bug with sublayers with gradient backgrounds
- fixed working delayout of layers (From now, if you add a delayout to a layer, sublayers of that layer will be animating out immediately, but the layer (and the next layer) will be delayed by the delayout time of the current layer. So, you can animate out all sublayers BEFORE the parent layer will begin the animation.)
- fixed imagepreload bug
- added an option to animating the first slide instead of fading in
- the script is now fully responsive! (You can add % instead of pixels to left and top properties of sublayers and width or height properties of LayerSlider container. IMPORTANT: you have to use style tags to add these values! For more information, please read the documentation about this!)
- added yourLogo feature (You can now display your fixed logo above LayerSlider container.)
- added a new API function called cbInit()
- some minor bugfixes

1.6

- now sublayers supports fading transition
- added slideoutdirection property to sublayers
- added global background color / background image support
- fixed '1px gap' bug
- fixed one layer bug
- fixed swing, linear and easeInOutQuint animations
- fixed hover on pause / stop bug
- fixed wrong directions of animations if navigating by slidebuttons
- fixed durationout property for sublayers

1.5

- added touch control for mobile devices
- now supports rel tags for properties
- added slidedirection property for sublayers too
- added pauseOnHover feature
- added a powerful API with callback functions

1.0

- first release

Requirements

jQuery (changed in version 4.0!)

Because this is a jQuery plugin, you'll need jQuery and jQuery Easing JS or jQuery's UI's Effects Core to run it.

- jQuery 1.7+ (important: due to a jQuery bug, required version 1.7 or newer), compatible with jQuery 1.9
- jQuery Easing JS (included) or jQuery UI Effects Core
- a modified version of jQuery Transit JS (included) to the new transitions
- recommended: valid xHTML or HTML 5 markup with DOCTYPE

Browser Support

- Chrome 10+
- Firefox 3+
- Safari 3+
- Opera 10+
- Internet Explorer 7+ (some rendering problems may appear in IE7)
- iOS Safari
- Android Browser
- Other mobile browsers

Browser Support for the 3D Transitions *

- Chrome 12+
- Chrome for iOS
- Chrome for Android 18+
- Firefox 10+
- Firefox for Android 18+
- Safari 4+
- iOS Safari 3.2+

* Browsers which don't support 3D transitions will use 2D transitions instead as fallback mode.

Please note, that Internet Explorer 10 and some versions of Android browsers has CSS3 3D support but they don't support the 'preserve-3d' CSS3 property well, which is required for the correct 3D transitions - so they will be using 2D transitions. In old browsers some animations (for example fading the transparent PNGs) are missing, but the plugin is fully functional.

Usage

Please note, that all the source codes in this documentation are in HTML 5 markup.
If you are using xHTML, please be aware of the differences.

Changes in version 4

The most important changes in LayerSlider version 4.0 are the new 2D & 3D transitions.

These transitions requires a special modified version of the jQuery Transit Javascript library that we included in our plugin package. If you are using version 3.x or older, you need to include it into your document to get the new transitions working. You will also have to include the layerslider.transitions.js file which contains the configuration settings for our more than 200 added transitions.

Differences in the WordPress version

In the WordPress version we are talking about slides and layers and here, in the standalone version we are talking about layers and sublayers. Of course the cores of the WordPress and standalone versions are the same and they work with the same HTML markup, only the naming is different. We are calling the layers to slides, and the sublayers to layers in the WordPess version because we are thinking that those names are more logical, but we had to stay compatible with the previous standalone versions, so we couldn't change for example the ls-layer class name, etc.

Setup

Include the following lines into the **head** section of your page. Please be aware of the order of these files. Of course the path can be different, depending where is layerslider folder on your server.

```
<!-- LayerSlider styles -->
<link rel="stylesheet" href="/layerslider/css/layerslider.css" type="text/css">

<!-- jQuery with jQuery Easing, and jQuery Transit JS -->
<script src="/layerslider/jQuery/jquery.js" type="text/javascript"></script>
<script src="/layerslider/jQuery/jquery-easing-1.3.js" type="text/javascript"></script>
<script src="/layerslider/jQuery/jquery-transit-modified.js" type="text/javascript"></script>

<!-- LayerSlider from Kreatura Media with Transitions -->
<script src="/layerslider/js/layerslider.transitions.js" type="text/javascript"></script>
<script src="/layerslider/js/layerslider.kreaturamedia.jquery.js" type="text/javascript"></script>
```

Loading the Script

You have to run this plugin like any other jQuery plugins, please paste the following code before the **</body>** tag in your HTML source:

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

    // jQuery $('document').ready(); function
    $('document').ready(function(){

        // Calling LayerSlider on your selected element after the document loaded
        $(selector).layerSlider();
    });

</script>
```

For example: if your LayerSlider container has an id named layerslider, the correct call method will be:

```
$('#layerslider').layerSlider();
```

HTML Markup

Create HTML markup for layers into the **body** section of your page. A typical HTML markup for the script looks like this:

```
<!--LayerSlider begin-->
<div id="layerslider" style="width: 1000px; height: 500px;">

    <!--LayerSlider layer-->
    <div class="ls-layer">

        <!--LayerSlider background-->
        <img class="ls-bg" src=".." alt="layer1-background">

        <!--LayerSlider sublayers-->
        <img class="ls-s-1" src=".." alt="layer1-sublayer1">
        <img class="ls-s-1" src=".." alt="layer1-sublayer2">
        <img class="ls-s-1 src=".." alt="layer1-sublayer3">
    </div>

    <!--LayerSlider layer-->
    <div class="ls-layer">

        <!--LayerSlider background-->
        <img class="ls-bg" src=".." alt="layer2-background">

        <!--LayerSlider sublayers-->
        <img class="ls-s1" src=".." alt="layer2-sublayer1">
        <img class="ls-s2" src=".." alt="layer2-sublayer2">
        <img class="ls-s3" src=".." alt="layer2-sublayer3">
    </div>

</div>
<!--LayerSlider end-->
```

It is important do define the width and height of your slider in the style tag (please don't use your css file for this)!

As you can see, we created two layers with adding **class** named **ls-layer** to the divs - in this case the slideshow will have only two complete screens. We also put sublayers into layers - these are images, but you can use any type of content you want, including HTML code, or Flash movies too.

There are a special sublayer, it is called background and it has a **class** named **ls-bg**. The parallax effect does not apply to background sublayers - so they are always staying in the background of the parent layer. (Otherwise you can add background-images to layers too.)

Other sublayers have a **class** named **ls-s***number*. This number is important because of the parallax effect: sublayers with higher number will animate faster and therefore it seems that you see the whole animation in 3D. So the class name of a sublayer is one of the most important setting of the parallax effect. Sublayers with all the same properties BUT different number in their class names will animate with different speeds. For example a sublayer with class="ls-s4" will animate faster than other sublayer with class="ls-s2". From version 4.0 you can use the value -1. Sublayers with **class="ls-s-1"** will be positioned exactly outside of the layer. We recommend to use this setting for your sublayers if you want to slide them in from various directions.

Obviously you can create unlimited sublayers and they can have the same **class** too.

Positioning sublayers

There are some important things that you have to keep if you want your slider to work properly.

- All sublayers are absolute positioned inside the layer
- You can use relative or static positioning INSIDE a sublayer
- You HAVE TO add the positions of your sublayers with a style tag. Please DO NOT use CSS file!
- You can use only left and top properties for positioning. Please DO NOT use right and bottom properties!
- You don't have to add left or top properties if you don't want. The default values are 0px;
- You can use percentage values if you want. Note that percentage positioning works differently in LayerSlider, the plugin will position the sublayer into the center point to the specified position, so for example 50% means that the center of the sublayer will be exactly in the center of the layer.
- You cannot use margins on sublayers, but you can use borders or paddings.
- Please DO NOT add any style properties to ls-bg classed background sublayers.

Example:

```
<div id="layerslider" style="width: 100px; height: 500px;">

    <div class="ls-layer">

        <img class="ls-s1" style="left: 20px; top: 200px;" src=".." alt="layer1-sublayer1">
        <img class="ls-s2" style="left: 100px; top: 30px;" src=".." alt="layer1-sublayer2">
        <img class="ls-s3" style="left: 50%; top:50%;"src=".." alt="layer1-sublayer3">

    </div>

    ... more layers ...

</div>
```

Global Settings

You can use the following - global - settings which are applied to the container of the slider. If you run multiple sliders on the same page, you can add different settings to each sliders. Note, that these are case sensitive.

```
$('#slider').layerSlider({

    autoStart               : true,
    responsive              : true,
    responsiveUnder         : 0,
    sublayerContainer       : 0,
    firstLayer              : 1,
    twoWaySlideshow         : false,
    randomSlideshow         : false,
    keybNav                 : true,
    touchNav                : true,
    imgPreload              : true,
    navPrevNext             : true,
    navStartStop            : true,
    navButtons              : true,
    thumbnailNavigation     : 'hover',
    tnWidth                 : 100,
    tnHeight                : 60,
    tnContainerWidth        : '60%',
    tnActiveOpacity         : 35,
    tnInactiveOpacity       : 100,
    hoverPrevNext           : true,
    hoverBottomNav          : false,
    skin                    : 'default',
    skinsPath               : '/layerslider/skins/',
    pauseOnHover            : true,
    globalBGColor           : 'transparent',
    globalBGImage           : false,
    animateFirstLayer       : false,
    yourLogo                : false,
    yourLogoStyle           : 'position: absolute; z-index: 1001; left: 10px; top: 10px;',
    yourLogoLink            : false,
    yourLogoTarget          : '_blank',
    loops                   : 0,
    forceLoopNum            : true,
    autoPlayVideos          : true,
    autoPauseSlideshow      : 'auto',
    youtubePreview          : 'maxresdefault.jpg',
    showBarTimer        : false,
    showCircleTimer     : true,

    // you can change this settings separately by layers or sublayers with using html style attribute

    slideDirection          : 'right',
    slideDelay              : 4000,
    durationIn              : 1000,
    durationOut             : 1000,
    easingIn                : 'easeInOutQuint',
    easingOut               : 'easeInOutQuint',
    delayIn                 : 0,
    delayOut                : 0

});
```

autoStart : ***true*** or ***false***
If true, slideshow will automatically start after loading the page.

firstLayer : ***number*** *(positive integer)* or **'*random*'**
LayerSlider will begin with this layer. Use the word 'random' to start with a random layer.

twoWaySlideshow : ***true*** or ***false***
If true, slideshow will go backwards if you click the prev button.

randomSlideshow : ***true*** or ***false***
If true, LayerSlider will change to a random layer instead of changing to the next / prev layer. Note that 'loops' feature won't work with randomSlideshow!

keybNav : ***true*** or ***false***
Keyboard navigation. You can navigate with the left and right arrow keys.

touchNav : ***true*** or ***false***
Touch-control (on mobile devices)

imgPreload : ***true*** or ***false***
Image preload. Preloads all images and background-images of the next layer.

navPrevNext : ***true*** or ***false***
If false, Prev and Next buttons will be invisible.

navStartStop : ***true*** or ***false***
If false, Start and Stop buttons will be invisible.

navButtons : ***true*** or ***false***
If false, slide buttons will be invisible.

thumbnailNavigation : **'*hover*'**, **'*always*'** or **'*disabled*'**
Thumbnail navigation mode. Can be 'disabled', 'hover', 'always'. Note, that 'hover' setting needs navButtons true!

tnWidth : ***number*** *(in pixels, a positive integer)*
Width of the thumbnails (in pixels).

tnHeight : ***number*** *(in pixels, a positive integer)*
Height of the thumbnails (in pixels).

tnContainerWidth : **'*width*'**
Default width of the thumbnail container, for example: '100' (pixels) or '60%' You must put the value in apostrophes!

tnActiveOpacity : ***number*** *(0 or a positive integer lower, than 100)*
Opacity of the active thumbnail (0-100).

tnInactiveOpacity : ***number*** *(0 or a positive integer lower, than 100)*
Opacity of the active thumbnail (0-100).

hoverPrevNext : ***true*** or ***false***
If true, the prev and next buttons will be shown only if you move your mouse over the slider.

hoverBottomNav : ***true*** or ***false***
If true, the bottom navigation controls (with also thumbnails) will be shown only if you move your mouse over the slider.

skin : **'*name_of_the_skin*'**
You can change the skin of the Slider. You can read more abut this at Skin support.

skinsPath : *'path_of_the_skins_folder/'*
You can change the default path of the skins folder. Note, that you must use the slash at the end of the path.

pauseOnHover : ***true*** or ***false***
If ture, SlideShow will pause when you move the mouse pointer over the LayerSlider container.

globalBGColor : ***CSS color methods***
Background color of LayerSlider. You can use all CSS methods, like hexa colors, rgb(r,g,b) method, color names, etc. Note, that background sublayers are covering the background.

globalBGImage : ***URL of the background image***
Background image of LayerSlider. This will be a fixed background image of LayerSlider by default. Note, that background sublayers are covering the global background image.

animateFirstLayer : ***false*** or ***true***
If true, first layer will animate (slide in) instead of fading.

yourLogo : ***'path_of_your_image'***
This is a fixed image that will be shown above of LayerSlider container. For example if you want to display your own logo, etc. You have to add the correct path to your image file.

yourLogoStyle : ***'css_properties'***
You can style your logo. You are allowed to use any CSS properties, for example add left and top properties to place the image inside the LayerSlider container anywhere you want.

yourLogoLink : ***'domain_or_page'*** or ***false***
You can add a link to your logo. Set false is you want to display only an image without a link.

yourLogoTarget : ***'target'***
If '_blank', the clicked url will open in a new window.

loops : ***number*** *(0 or a positive integer)*
Number of loops if autoStart set true (0 means infinite).

forceLoopNum : ***true*** or ***false***
If true, the slider will always stop at the given number of loops even if the user restarts the slideshow.

autoPlayVideos : ***true*** or ***false***
If true, slider will autoplay youtube / vimeo videos - you can use it with autoPauseSlideshow.

autoPauseSlideshow : ***'auto'***, ***true*** or ***false***
'auto', true or false. 'auto' means, if autoPlayVideos is set to true, slideshow will stop UNTIL the video is playing and after that it continues. True means slideshow will stop and it won't continue after video is played.

youtubePreview : ***'videopreviewname.jpg'***
Default thumbnail picture of YouTube videos. Can be 'maxresdefault.jpg', 'hqdefault.jpg', 'mqdefault.jpg' or 'default.jpg'. Note, that 'maxresdefault.jpg' os not available to all (not HD) videos.

showBarTimer : ***true*** or ***false***
You can show or hide the bar timer.

showCircleTimer : ***true*** or ***false***
You can show or hide the circle timer.

The following global settings can be override separately by each layers or sublayers local settings.

slideDirection : ***'left'***, ***'right'***, ***'top'***, ***'bottom'*** or ***'fade'* ***
Slide direction. New layers sliding FROM(!) this direction.
* From version 1.6 you can use the fade parameter if you want to fade in / out a sublayer instead of sliding. Note, that fade is working ONLY on sublayers but not on layers so if you want to fade in / out a sublayer (instead of sliding), you have to add the slidedirection or the slideoutdirection property to that sublayer and you cannot use fade as a global property! See the Local Settings below.

slideDelay : ***number*** *(millisecs)*
Time before the next slide will be loading.

durationIn : ***number*** *(millisecs)*
Duration of the slide-in animation.

durationOut : ***number*** *(millisecs)*

Duration of the slide-out animation.

easingIn : ***easing*** *(all easing types of jQuery UI Effects Core can be used)*
Easing (type of transition) of the slide-in animation.

easingOut : ***easing*** or *(all easing types of jQuery UI Effects Core can be used)*
Easing (type of transition) of the slide-out animation.

delayIn : ***number*** *(millisecs)*
Delay time of the slide-in animation.

delayOut : ***number*** *(millisecs)*
Delay time of the slide-out animation.

Local Settings

Some of the above settings you can apply uniquely to a layer or a sublayer. For example, if you want to add an extra slide direction and slide delay property to a layer, use the following code:

```
<div class="ls-layer" style="transition2d: all; slidedelay: 6000">

    <!--sublayers-->

</div>
```

As you can see, we used the style tag to add properties.* Of course you can mix the normal style properties with the properties of LayerSlider. Another example, we are adding some other properties to a sublayer:

```
<img class="ls-s10" src="../gallery/l29.png" style="left: 20px; top: 100px; durationin: 1000; easingin: ea
```

* From version 1.5, you can use both rel and style tags to add properties. Some older browser (like Firefox 2) doesn't allow to add custom properties into style tag, so in that case you can use rel tag.

You can add this properties to layers if you are using OLD transitions:

- slidedirection
- durationin
- durationout
- easingin
- easingout
- delayin
- delayout
- slidedelay

You can add this properties to layers if you are using NEW 2D or 3D transitions (all the other properties listed above will be skipped):

- slidedelay
- transition2d
- transition3d
- timeshift

You can add this properties to sublayers:

- slidedirection *
- slideoutdirection **
- durationin
- durationout
- easingin
- easingout
- delayin
- delayout
- showuntil
- rotatein ***
- rotateout ***
- scalein ***
- scaleout ***

* From version 1.5, you can add the slidedirection property to the sublayers too. So you can animate some sublayers for example from top as the main layer is sliding from left. You can use all four directions at the same time with different

sublayers inside a main layer!

** From version 1.6, you can add the slideoutdirection property to the sublayers. So you can set the slide out animation independent from the slide in animation. You can also use the fade parameter if you want to fade out a sublayer.

*** From version 4.5.0 you can rotate and / or scale sublayers as well!

Using the new 2D & 3D transitions

In the older versions of LayerSlider layers animated with simple slide transitions. In LayerSlider 4.0 you can use more breathtaking 2D & 3D transitions instead. Please note that not all browsers are supporting 3D transitions. You can read more about this in the Browser Support section of this documentation.

Because these are only layer transitions, you will need to put special properties on your layers. Let's see an example:

```
<div class="ls-layer" style="transition3d: 1,4,5,11; transition2d: 2,8,30; slidedelay: 6000;">

    <!--background sublayer-->
    <!--other sublayers-->

</div>
```

As you can see, we added some transition IDs to both 2D and 3D transitions. You can choose the IDs of the transitions from the transition gallery. We created more, than 200 2D & 3D transitions for you, the configuration settings of these transitions are in the layerslider.transitions.js file. Of course you can change the properties of the transitions or you can also create new transitions as well if you open this file in a text editor. Modifying or creating new transitions requires some Javascript knowledge, please do not attempt to do it if you aren't familiar with Javascript.

Please note, that if you add 2D and / or 3D transitions to the layer, the slider will use that transition when changing TO that layer (but not, when changing FROM that layer) and you don't need to add the following layer properties, because these are used for animating with an old transition:

- slidedirection
- durationin
- durationout
- easingin
- easingout
- delayin
- delayout

Of course you will need to add the slidedelay property, and there is a new layer property which you can use with the new transitions:

- timeshift

With timeshift feature, you can control the timing of the sublayer animations when the slider changes TO the current layer with a new 2D or 3D transition. Zero means that the sublayers of the current layer will animate in when the slide transition ends. You can time-shift the starting times of the sublayer animations with positive or negative values. For example:

```
<div class="ls-layer" style="transition3d: 1,4,5,11; transition2d: 2,8,30; timeshift: -1000;">

    <!--background sublayer-->
    <!--other sublayers-->

</div>
```

In this case, the sublayers of this layer will start to animate in BEFORE 1 sec (1000 ms) of the end of the layer transition.

Using the transition3d and transition2d properties

Here are some examples of using these properties in the right way:

transition2d: 2,3,10,12; - the slider will choose one of the specified 2D transitions.

transition3d: 1,3,6,9; - the slider will choose one of the specified 3D transitions, if your browser doesn't support 3D transitions, the slider will choose one 2D transition from all the available 2D transitions.

transition3d: 1,3,6,9; transition2d: 2,4,8; - the slider will choose one of the specified 3D transitions, if your browser doesn't support 3D transitions, the slider will choose one 2D transition from the specified 2D transitions.

transition3d: all; transition2d: all; - the slider will choose one of all the available 3D transitions, if your browser doesn't support 3D transitions, the slider will choose one 2D transition from all the available 2D transitions.

Making LayerSlider Responsive

From version 3.0 it is very easy: just put the responsive : true in the LayerSlider init code:

```
$('#slider').layerSlider({

    responsive : true,

    <!--other global settings-->

});

```

If you do this, your slider will get a starting width / height position (which you have to set with style tag). If the width of the container element of your slider changes, your LayerSlider inside will also change its width / height, and also all the images, text, and video sublayers will change their sizes and positions. You won't need to position the sublayers with percentage values (but of course you can do it if you want) because LayerSlider will reposition every sublayer - even if they have their positions in pixels.

responsiveUnder and sublayerContainer features

If you create a full width slider you may need these two smart features.

responsiveUnder

You can force the slider to change automatically into responsive mode but only if the slider width is smaller than responsiveUnder pixels. It can be used if you need a full-width slider with fixed height but you also need it to be responsive if the browser is smaller... Important! If you enter a value higher than 0, the normal responsive mode will be switched off automatically!

You will need to specify the value in the init code:

```
$('#slider').layerSlider({

    responsiveUnder : 960,

    <!--other global settings-->

});

```

sublayerContainer

This feature is needed if you are using a full-width slider and you need that your sublayers forced to positioning inside a centered custom width container. Just specify the width of this container in pixels! Note, that this feature is working only with pixel-positioned sublayers, but of course if you add left: 50% position to a sublayer it will be positioned horizontally to the center, as before!

You will need to specify the value in the init code:

```
$('#slider').layerSlider({

    sublayerContainer : 900,

    <!--other global settings-->

});

```

Thumbnail navigation

From version 3.5 you can use thumbnails instead of buttons. We added two types of thumbnail navigation (as you can read in Global Settings chapter). If you enable thumbnail navigation the slider will use your background sublayers (images with ls-bg class) and your video preview images to create a thumbnail to your layers by default. Of course you can add your own thumbnail images to all of your layers one by one. To do this first you will need to generate your own images. After that you have to add an image element inside each of your layers and add a special class for them.

```
<div class="ls-layer" style="slidedirection: top; slidedelay: 6000">

    <img class="ls-tn" src="yourthumbnailimage" alt="dontneed">

    <!--background sublayer-->
    <!--other sublayers-->

</div>
```

As you can see, we used an image with ls-tn class to add our own thumbnail to the selected layer.

How to change sublayers before the layer will slide out

From version 3.0 you can slide / fade out a sublayer BEFORE the layer will slide out (or before other sublayers will slide / fade in).

To get this work, you just need to add a parameter to the style tag of the sublayer:

```
<img class="ls-s10" src="../gallery/l29.png" style="left: 20px; top: 100px; showuntil: 2000;">
```

In this case the sublayer will slide / fade out after 2000ms of sliding / fading in and before the layer will slide out. Of course you can create another sublayer for example with 3000ms of delayin time, so the sublayer above will disappear and the other will appear after. Please check layers 1 (texts), 3 (fishes) and 5 (cars) in our demo slider to see this feature in action.

How to add a link which jumps to another layer (linkTo)

From version 2.0 you can easily add a special link to any sublayer which jumps to the selected layer of the slider. You don't have to write any JS code, just add a special class to the sublayer by the following way:

```
<img class="ls-s2 ls-linkto-3" src="../gallery/l29.png" style="durationin: 1000; easingin: easeOutExpo">
```

As you can see, we used a class name "ls-linkto-" and a *number* which is of course the ordinal number of the layer where you want to jump to.

How to add a link to the whole layer (class="ls-link")

If you need to link the whole layer, you will need to create a special sublayer (an empty <a> element with the ls-link class) by the following way:

```
<div class="ls-layer" style="slidedirection: top; slidedelay: 6000">

    <!--background sublayer-->
    <!--other sublayers-->

    // linking the whole layer

    <a href="yoururl" class="ls-link"></a>

</div>
```

Deep linking

In LayerSlider 3.5 you have an option to deep link a layer. This is means that you can give a special URL to your users that will open LayerSlider with the desired layer on start. To do that, you have to add a special style property to your layer and reference in the URL with a hashmark sign followed this value.

```
<div class="ls-layer" style="deeplink: layername; slidedirection: top; slidedelay: 6000">

    <!--background sublayer-->
    <!--other sublayers-->

</div>
```

For example if you added deeplink: layername to your layer, you can reference to this layer with the URL: yourdomain.com/yourpage/#layername

LayerSlider API

We added an API to make more comfortable the usage of this script.
The API can be divided into three parts:

1. The LayerSlider Data Object - Important: it's changed in v.3.0!

Sometimes you may need to know the current layer index or for example if the script is even animating or not. Now you can get the whole LayerSlider object with realtime values of variables at any time. Just call the $(your-layerslider-container).layerSlider('data'); function and you can get back this object. Try this:

```
var lsdata = $(your-layerslider-container).layerSlider('data');
alert( lsdata['curLayerIndex'] );
```

This will show you the current layer index.
You can get all the events and variables in real time, but the most important are the following:

- lsdata**['curLayer']** (jQuery object of the current layer)
- lsdata**['curLayerIndex']** (current layer index)
- lsdata**['nextLayer']** (jQuery object of the next layer)
- lsdata**['nextLayerIndex']** (next layer index)
- lsdata**['isAnimating']** (true or false)
- lsdata**['layersNum']** (number of layers)
- lsdata**['autoSlideshow']** (true or false, slideshow is running if true)
- lsdata**['paused']** (true or false)

2. Callback functions

You can use some callback functions too. You have to define these functions with the global settings:

```
$('#slider').layerSlider({

    ...

    global settings

    ...

    cbInit      : function(element){},  // Calling when LayerSlider loads, returns the LayerSlider jQuery
    cbStart     : function(data){},     // Calling when you click the slideshow start button, returns the
    cbStop      : function(data){},     // Calling when click the slideshow stop / pause button, returns
    cbPause     : function(data){},     // Calling when slideshow pauses (if pauseOnHover is true), retur
    cbAnimStart : function(data){},     // Calling when animation starts, returns the LayerSlider Data ob
    cbAnimStop  : function(data){},     // Calling when the animation of current layer ends, but the subl
    cbPrev      : function(data){},     // Calling when you click the previous button (or if you use keyb
    cbNext      : function(data){},     // Calling when you click the next button (or if you use keyboar

});
```

3. Controller functions

You can call these functions to control the slider:

- $('#slider').layerSlider('**prev**'); if you want to change to the previous layer
- $('#slider').layerSlider('**next**'); if you want to change to the next layer
- $('#slider').layerSlider(**2**); if you want to jump to a specified layer
- $('#slider').layerSlider('**start**'); if you want to start the slideshow
- $('#slider').layerSlider('**stop**'); if you want to stop the slideshow

Important: if you want to create an element OUTSIDE of the slider which will force the slider to change / jump to a specified slide, please use the following code if you have an anchor element:

```
<a href="javascript:void(0);" onclick="$('#slider').layerSlider(2);">
    Some text
</a>
```

or if you have a div or other element:

```
<div onclick="$('#slider').layerSlider(2);">
    Some content
</div>
```

Embedding Videos (YouTube and Vimeo)

We want to show an example how to create a layer with an embedded video sublayer. LayerSlider works with embed codes of many video sharing websites, but we created some features to improve the work with two most popular sites: YouTube and Vimeo.

So, how to put your video into a layer?

It is very simple: you will need to create a DIV sublayer and put your video embed code into. That's it! You can see in the following example, that we created a slider with two layers and we put a YouTube video into a sublayer of the first layer, and put a Vimeo video into a sublayer of the second layer.

```
<div id="layerslider">

    <div class="ls-layer">

```

```
        <img class="ls-bg" src=".." alt="layer1-background">
        <div class="ls-s2">
            <iframe width="640" height="360" src="http://www.youtube.com/embed/MT_r7Nw14EA" frameborder="
        </div>
    </div>

    <div class="ls-layer">

        <img class="ls-bg" src=".." alt="layer2-background">
        <div class="ls-s2">
            <iframe src="http://player.vimeo.com/video/24535181?color=ff9933" width="873" height="491" fr
        </div>
    </div>

</div>
```

Important things about embedding:

- You can position the video by adding style tag to the container DIV sublayer
- The dimensions of the video will be always the same as you define in the embed code with width and height properties
- You can create a full size video by adding the same width and height property to the iframe embed code as your slider's width and height

YouTube and Vimeo video features

As we said, we included some nice features if you are using YouTube and / or Vimeo video services.

Video Preview Image

LayerSlider adds a video preview image (from the video) with a play button, so when the layer slides in with a video sublayer, you don't see those video controls until the video will begin to play. If you are using YouTube, you can choose from different sizes of preview images (see youtubePreview option in Global Settings).

autoPlayVideos

If you set it true, videos will begin to play immediately when the layer slides in.

autoPauseSlideshow

If a slideshow is running and autoPlayVideos is set to true, the slideshow can be auto-paused while videos are playing. Just set autoPauseSlideshow to true or 'auto' and the slideshow will stop. Of course in 'auto' mode it will continue after the video has played.

Skin support

You can easily add skins to LayerSlider.

Actually there are eight pre-defined skins that you can use. These are:

- borderlessdark
- borderlesslight
- darkskin
- defaultskin
- fullwidth
- glass
- lightskin
- minimal

and if you want to forbid LayerSlider to load any skin, use: skin : 'noskin'.

Each of them have a separated folder in **/layerslider/skins/**. The name of the folder is the name of the skin too. If you open skin.css from any skin folder, you realize, that I used the name of the skin as a css class by the following way:

.ls-*name_of_the_skin*

If you run LayerSlider with the skin property, it will trying to load skin.css from the folder **/layerslider/skin/name_of_the_skin/** and it will add **ls-name_of_the_skin** css class to LayerSlider container.

So, you can easily load your skin, but don't forget that the path of the skins folder.

```
$('#layerslider').layerSlider({
    skin : 'darkskin'
});
```

We attached a layered PSD file with the skins.

Examples

Example 1: LayerSlider with 2D transitions and hover thumbnail navigation

Example 2: LayerSlider with 3D transitions and thumbnail navigation

Example 3. Full-width slider with smart features and with old and new transitions

Example 4. Fixed size and responsive content sliders

Example 5. Video gallery slider

Example 6. LayerSlider API demo

Example 7. 3D & 2D transition gallery

Contact us!

If you like our works.
If you need a help in using one of our CodeCanyon items.
If you have a great idea which we can work together.

Visit our website!

Find us on Facebook!

Our Apps & Codecanyon