WunderSlider

Responsive Embedded & Fullscreen jQuery Slider

www.wunderslider.com

by www.itthinx.com

WunderSlider

Copyright (c) www.itthinx.com

LICENSE RESTRICTIONS

This software is provided subject to the license granted.

Unauthorized use and distribution is prohibited.

See COPYRIGHT.txt and LICENSE.txt.

This software relies on code that is NOT licensed under the GNU General Public License. Files licensed under the GNU General Public License state so explicitly in their header.

You MUST be granted a license by the copyright holder for those parts that are not provided under the GNU Public License.

If you have not been granted a license DO NOT USE this software until you have BEEN GRANTED A LICENSE.

Use of this software without a granted license constitutes an act of COPYRIGHT INFRINGEMENT and LICENSE VIOLATION and may result in legal action taken against the offending party.

Being granted a license is GOOD because you will get support and contribute to the development of useful free and premium software, themes and plugins that you will be able to enjoy.

Thank you!

Visit www.itthinx.com for more information.

Table of Contents

1 Setup

Files

1. Unzip the file wunderslider-x.y.z.zip[1].

   You will obtain the following folder structure:

   ○ wunderslider-x.y.z

      • build : these files are used to deploy WunderSlider on your web site

      • demo : these are used in section 2 Tutorials

      • documentation

      • src

      • wordpress : contains the WunderSlider WordPress plugin

2. To get started, try the demos that are included in the demo folder. You can copy them to your local or to your remote web server. Please refer to section 2 Tutorials on details about these.

3. To use WunderSlider on your web site and use it with **HTML** or **PHP** code, upload the build folder to your web server and apply the instructions given in the corresponding tutorials for HTML or PHP. To use WunderSlider on your **WordPress**[2] site, upload and activate the WunderSlider plugin and apply the instructions as given in the WordPress tutorial section.

WunderSlider pages

   • WunderSlider page   : http://www.itthinx.com/wunderslider

   • Demo : http://www.wunderslider.com

---

1   x, y and z represent the version numbers
2   requires WordPress 3.3 or above

Support

Please make sure to review the complete information contained in this document before asking for support. Much care has been taken to guide you through the steps necessary to deploy this software on your web site.

If you still have questions or would like to post a suggestion, please first review the WunderSlider page and its comments http://www.itthinx.com/wunderslider/ for similar problems or suggestions and proposed solutions.

Ultimately, you can ask for support on that page as well by posting a comment.

2 Tutorials

The following three sections cover different ways to include WunderSlider on your web site:

- HTML – using HTML and Javascript

- PHP – using PHP and XML

- WordPress – using the included WordPress plugin and shortcodes

For the tutorials we assume that you have already extracted the contents of the file `wunderslider-x.y.z.zip` and will be using the folder structure as outlined in Files.

Shortcuts

- For **WordPress** please proceed directly to the WordPress section.

- **HTML** and **PHP** tutorials : Note that you can either follow these examples or simply use the contents of the `demo` subfolder to have a ready set of examples to work on.

HTML

To follow the tutorial, copy the content of the `build` folder to a folder on your web server. In that folder you should have:
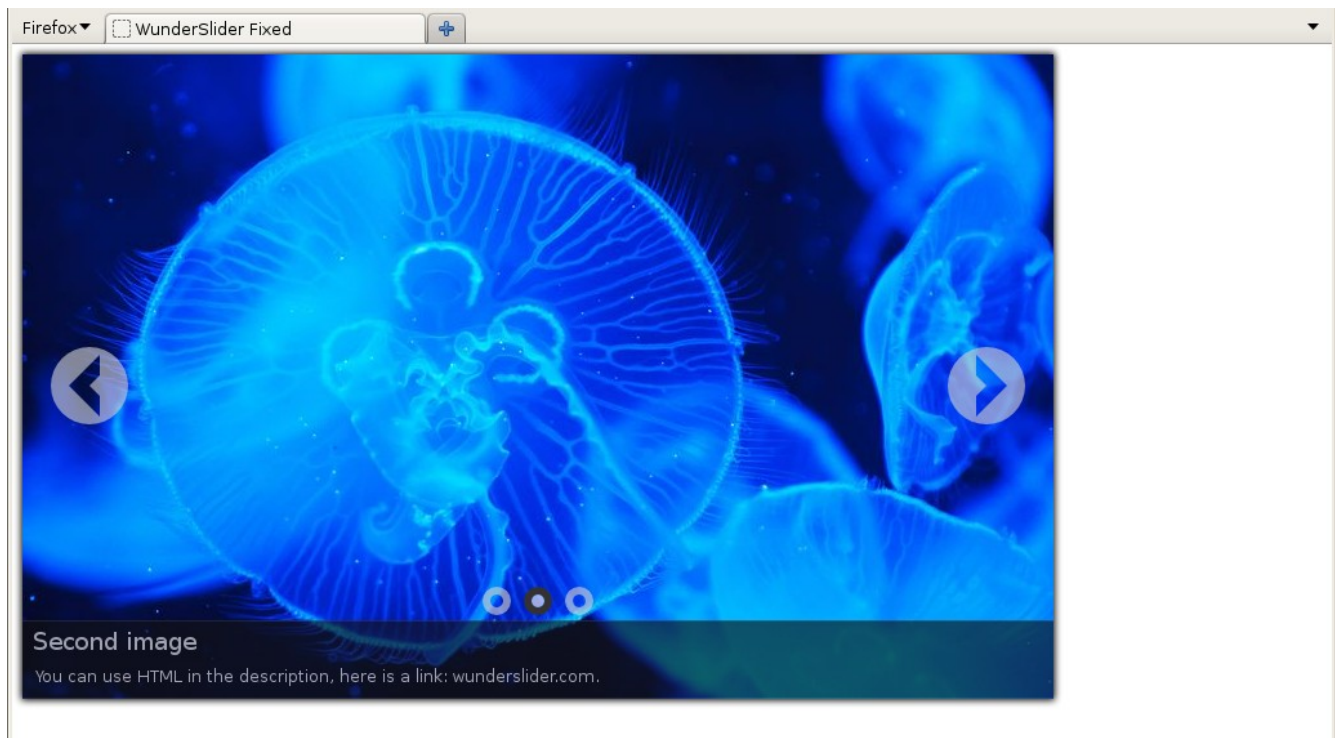

In the following, we assume that the HTML files in which the WunderSlider is embedded, will be placed in that folder as well.

The following three examples cover:

- creating a fixed size slideshow,

- a proportional slideshow,

- and a fullscreen slideshow.

Note that paths and URLs have to be adjusted if you are using a different folder structure.

Creating a WunderSlider with fixed size

In the HTML file you want to place the WunderSlider in, add these lines in the head section:

```
<link rel="stylesheet" type="text/css" href="css/default/wunderslider-
min.css" />
<!--[if lt IE 9]>
<link rel="stylesheet" type="text/css" href="css/default/wunderslider-ie-
min.css" />
<![endif]-->
<script type="text/javascript" src="js/jquery-1.7.2.min.js"></script>
<script type="text/javascript" src="js/wunderslider-min.js"></script>
```

Note that we are using the default skin here, if you want to choose another theme, substitute `default` with the name of the skin's subfolder. The skin subfolders are located in the `css` folder.

In the body section add a container `div` for the WunderSlider, we will give it the id `wunderslider` to easily identify it:

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

To create the WunderSlider we are going to add some Javascript that provides the images and options we want.

To create a WunderSlider, we need to create an instance:

```
wunderslider = new itthinx.WunderSlider( parent, images, args );
```

The constructor's signature uses:

- `parent` is the container of the WunderSlider which needs to be an empty div that can be appropriately identified.

- **images** is an array of objects which must at least provide the url to each image and optionally additional parameters as described in <u>Image Parameters</u>.

- **args** is an object that provides selected options from those described in <u>General Options</u>.
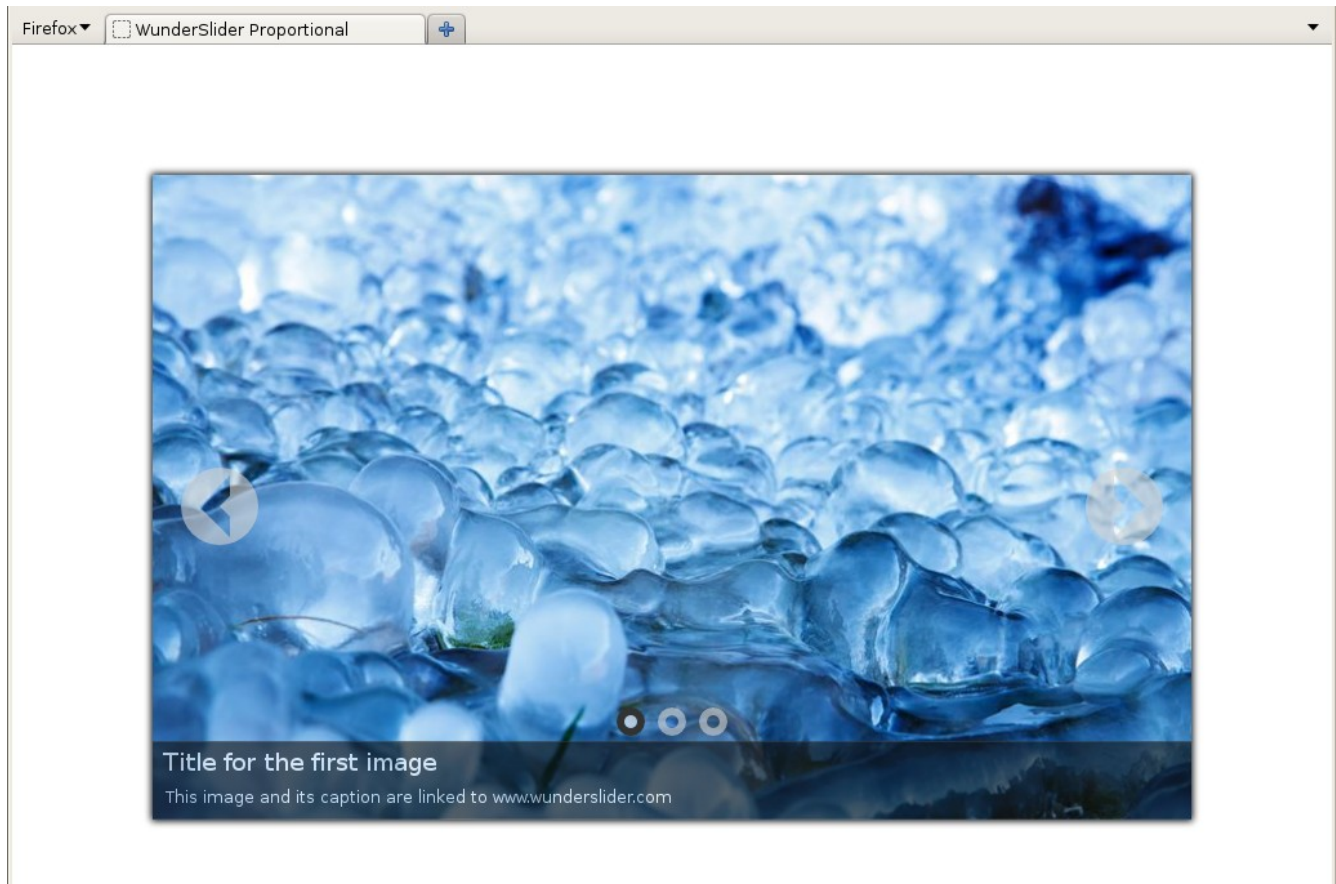
Add this code after the container `div`:

```
<script type="text/javascript">
(function($) {
      $(window).load(function(){
            var parent = document.getElementById('wunderslider');
            var wunderSlider = new itthinx.WunderSlider(
                  parent,
                  [
                  {
                        url:'images-demo/image-0.jpg',
                        title:'Title for the first image',
                        description:'A nice image...',
                        linkUrl:'http://www.wunderslider.com/'
                  },
                  {
                        url:'images-demo/image-1.jpg',
                        title:'Second image',
                        description:'Another image.'
                  },
                  {
                        url:'images-demo/image-2.jpg',
                        title:'Here is another image',
                        description:'One more image.'
                  }
                  ],
                  {mode:'proportional', mouseOverPause:true}
            );
});
})(jQuery);
</script>
```

With this we are now ready to test our slider.

If you want to try the fixed size version without having to create all the above, copy the contents of the `demo` folder to your web server and visit `demo/wunderslider-fixed.html`.

Creating a proportionally sized WunderSlider

The steps to create a WunderSlider that adapts its size in proportion to the viewport are the same as for the fixed size version with only two modifications:

The container div must have a width set as a percentage:

```
<div id="wunderslider" style="width:80%;height:500px;"></div>
```

To make it a little more attractive we can center it:

```
<div id="wunderslider" style="width:80%;height:500px;margin-left:auto;margin-right:auto;left:0;right:0;"></div>
```

We also need to set the display option: `display:'proportional'`

In this case we are also going to specify what effects we want to apply using the `effects` option and in addition we set the `period` to 3 seconds to change the default interval from one slide to the next.

All else maintained, we create a new WunderSlider thus:

```
var wunderSlider = new itthinx.WunderSlider(
        parent,
        [
            {
                url:'images-demo/image-0.jpg',
                title:'Title for the first image',
                description:'This image and its caption are linked to
```

```
www.wunderslider.com',
                linkUrl:'http://www.wunderslider.com/'
        },
        {
                url:'images-demo/image-1.jpg',
                title:'Second image',
                description:'You can use HTML in the description, here is a
link: <a href="http://www.wunderslider.com/">wunderslider.com</a>.'
        },
        {
                url:'images-demo/image-2.jpg',
                title:'Here is another image',
                description:'This image is not linked.'
        }
    ],
    {display:'proportional', mode:'proportional', mouseOverPause:true,
effects:['random','checkers'], period:3000}
);
```
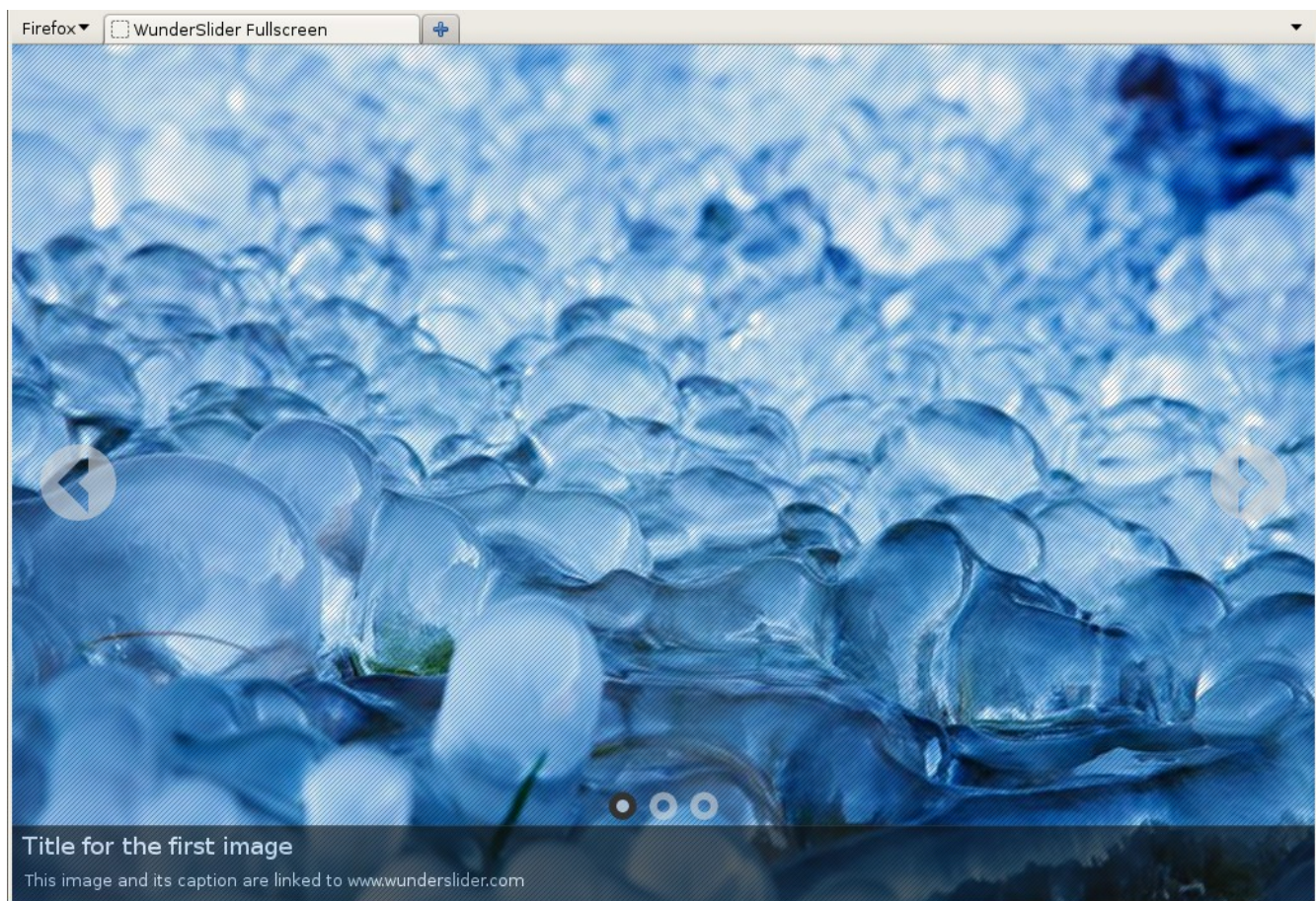
The equivalent for this example can be found in the `demo` folder: `demo/wunderslider-proportional.html`

Creating a fullscreen WunderSlider

For our fullscreen version we are going to use a simple div again:

```
<div id='wunderslider'></div>
```

We also need to set the `display` option to `fill` when creating the WunderSlider instance.

In this example we are also going to:

- enable flicking by disabling the `clickable` option and enabling the `useFlick` option

- use an `overlay`

All else maintained, the code to create our WunderSlider instance now looks like this:

```
var wunderSlider = new itthinx.WunderSlider(
    parent,
    [
    {
            url:'images-demo/image-0.jpg',
            title:'Title for the first image',
            description:'...',
            linkUrl:'http://www.wunderslider.com/'
    },
    {
            url:'images-demo/image-1.jpg',
            title:'Second image',
            description:'...'
    },
    {
            url:'images-demo/image-2.jpg',
            title:'Here is another image',
            description:'...'
    }
    ],
    {display:'fill', mode:'proportional', overlay:'stripes',
clickable:'false', useFlick:'true'}
);
```

The equivalent for this example can be found in the `demo` folder: `demo/wunderslider-fullscreen.html`

Creating a fixed size WunderSlider

WunderSlider for PHP code provides an additional tool that can be used to easily deploy instances of the slider where needed, based on embedded XML or XML files that determine the slider's images and appearance.

In all cases you need to include the `class-wunderslider-xml-parser.php` script located along with the other resources in the build folder.

To create our slider we are going to use this PHP code:

```php
<?php
    require_once( 'class-wunderslider-xml-parser.php' );
    $p = new WunderSlider_XML_Parser( 'wunderslider-fixed.xml' );
    echo $p->code;
?>
```

This assumes that the WunderSlider XML specification is in the file wunderslider-fixed.xml just like in the example given in the `demo` folder.

The WunderSlider XML specification consists of the `<wunderslider>` root element and an `<image>` element for each image:

```
<wunderslider ...>
    <image … />
    <image … />
    ...
</wunderslider>
```

The General Options are given as attributes of the `<wunderslider>` element while Image Parameters are passed as attributes of the `<image>` element.

The XML can also be passed in the argument directly instead of providing its filename:

```php
<?php
    require_once( 'class-wunderslider-xml-parser.php' );
    $p = new WunderSlider_XML_Parser( 'XML goes here' );
    echo $p->code;
?>
```

Please refer to the file `wunderslider-fixed.php` for a full code example.

If you would like to separate loading scripts and stylesheets from the Javascript and container code, please refer to the file `wunderslider-fixed-separate.php`.

The WunderSlider XML for this example looks like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wunderslider mode="proportional" mouseOverPause="true">
    <image url='images-demo/image-0.jpg' title='Title for the first image'
description='This image and its caption are linked to www.wunderslider.com'
```

```
linkUrl='http://www.wunderslider.com/' />
      <image url='images-demo/image-1.jpg' title="Second image" description="You
can use HTML in the description, here is a link: &lt;a
href='http://www.wunderslider.com/'&gt;wunderslider.com&lt;/a&gt;." />
      <image url='images-demo/image-2.jpg' title='Here is another image'
description='This image is not linked.' />
</wunderslider>
```

This is the content of the file wunderslider-fixed.xml in the `demo` folder.

Creating a proportionally sized WunderSlider

For this example we are simply going to change the WunderSlider XML while maintaining the rest. The important modification here is done on the opening `wunderslider` tag, where we add the `display="proportional"` attribute and specify the proportional container dimension:

```
<?xml version="1.0" encoding="UTF-8"?>
<wunderslider display="proportional" mode="proportional" container-width="80%"
container-height="500px" container-style="margin:0 auto;left:0;right:0;"
mouseOverPause="true" effects="random,checkers" period="3000">
      <image url='images-demo/image-0.jpg' title='Title for the first image'
description='This image and its caption are linked to www.wunderslider.com'
linkUrl='http://www.wunderslider.com/' />
      <image url='images-demo/image-1.jpg' title="Second image" description="You
can use HTML in the description, here is a link: &lt;a
href='http://www.wunderslider.com/'&gt;wunderslider.com&lt;/a&gt;." />
      <image url='images-demo/image-2.jpg' title='Here is another image'
description='This image is not linked.' />
</wunderslider>
```

Please refer to the files:

— `demo/wunderslider-proportional.php`

— `demo/wunderslider.proportional.xml`

for the full code example.

Creating a fullscreen WunderSlider

For the fullscreen example we need to provide the `display="fill"` and `mode="proportional"` options in the `<wunderslider>` element:

```
<wunderslider display="fill" mode="proportional" overlay="stripes"
clickable="false" useFlick="true">
…
</wunderslider>
```

Of course specifying `useFlick` is again optional.

Please refer to these files for the full code:

— `demo/wunderslider-fullscreen.php`

- demo/wunderslider-fullscreen.xml

WordPress

To use WunderSlider within WordPress posts and pages, please install the WunderSlider WordPress plugin found in the `wordpress` folder.

The plugin requires **WordPress 3.3 or above**.

This plugin provides the `[wunderslider]` **shortcode**. The shortcode is used to surround WunderSlider XML on any post or page to embed a WunderSlider:

```
[wunderslider]
<wunderslider ...>
        <image … />
        <image … />
        ...
</wunderslider>
[/wunderslider]
```

Note the surrounding `[wunderslider]` … `[/wunderslider]` opening and closing shortcodes and that the specification does start directly with the `<wunderslider>` tag (do **not** include `<?xml version="1.0" encoding="UTF-8"?>` as in the PHP examples).

Creating a fixed size WunderSlider

To embed a WunderSlider on a post or page, you will use the `[wunderslider]` shortcode enclosing WunderSlider XML.

Here's a skeleton example for three images:

```
[wunderslider]
<wunderslider mode="proportional" width="560px" height="400px">
<image url="..." />
<image url="..." />
<image url="..." />
</wunderslider>
[/wunderslider]
```

Let's see how this looks like with uploaded images' URLs, titles and descriptions:

Create a new page or post and upload some images.

Switch the editor to HTML and use the following code – **substitute** the image URLs by those of your uploaded images:

```
[wunderslider]
<wunderslider mode="proportional" mouseOverPause="true" width="560px"
height="400px">
<image url="http://www.example.com/wp-content/uploads/2012/07/image-0.png"
title="Title for the first image" description="This image and its caption are
linked to www.wunderslider.com" linkUrl="http://www.wunderslider.com/" />
<image url="http://www.example.com/wp-content/uploads/2012/07/image-1.png"
title="Second image" description="You can use HTML in the description, here is a
link: &lt;a
```

```
href='http://www.wunderslider.com/'&gt;wunderslider.com&lt;/a&gt;." />
<image url="http://www.example.com/wp-content/uploads/2012/07/image-2.png"
title="Here is another image" description="This image is not linked." />
</wunderslider>
[/wunderslider]
```

Any single or double quotes must be plain, that is **no left or right quotes**:

- These quotes are ok: `'` and `"`

- These quotes are **not** ok: `"` or `"`

Creating a proportionally sized WunderSlider

To create a proportional version do as in in the previous example but modify it by adding the `display` attribute to the `wunderslider` tag and specify the `container-width` as a percentage and the `container-height` in pixels, for example:

```
<wunderslider display="proportional" mode="proportional" container-width="80%"
container-height="500px" container-style="margin:0 auto;left:0;right:0;"
mouseOverPause="true" effects="random,checkers" period="3000">
```

This will result in a WunderSlider that will fill 80% of the content space.

Creating a fullscreen WunderSlider

For our fullscreen version we are going to change the `display` option to fill and add an overlay. We also make the slider flickable:

```
<wunderslider display="fill" mode="proportional" overlay="stripes"
clickable="false" useFlick="true" appendTo="body">
```

As before, just create a new post, copy the code from the fixed size example and substitute the opening `<wunderslider ...>` tag with the above.

Note the `appendTo="body"` option added here, this allows to add the WunderSlider shortcode and its XML specification anywhere in the content but have it placed on a specific element in the document.

Also note that the navigation buttons on a page in fullscreen mode can be unintentionally covered by other elements such as the `body` tag and are thus not clickable unless the size of the covering element is restricted. This happens for example with the *Twenty Eleven* theme and the default skin, where the `body` covers these controls as it takes up the entire width of the viewport. When placing a WunderSlider in fullscreen mode, this can be remedied by applying for example a `max-width` and `margin` to maintain the content centered:

```
body {
     margin: 0 auto;
     max-width: 1000px;
}
```

When editing the page in HTML mode you can apply these CSS rules and embed the fullscreen WunderSlider like this:

```
<style type="text/css">
body {
     margin: 0 auto;
     max-width: 1000px;
}
</style>
[wunderslider]
<wunderslider display="fill" mode="proportional" overlay="stripes"
clickable="false" useFlick="true" appendTo="body">
     <image … />
     …
</wunderslider>
[/wunderslider]
```

This will restrict the size of the body element to a maximum width of 1000 pixels and make the previous/next navigation buttons accessible.

Depending on the theme, the skin used and the location of the navigation buttons, this or a similar approach can be used. Optionally the navigation buttons can be disabled by adding useNav="false" to the <wunderslider> element.

3 Documentation

Please also refer to the WunderSlider pages:

- WunderSlider page : http://www.itthinx.com/wunderslider

- Demo : http://www.wunderslider.com

Please make sure to review the complete information contained in this document before asking for support. Much care has been taken to guide you through the steps necessary to deploy this software on your web site. If you still have questions or would like to post a suggestion, please first review the WunderSlider page and its comments http://www.itthinx.com/wunderslider/ for similar problems or suggestions and proposed solutions. Ultimately, you can ask for support on that page as well by posting a comment.

Noteworthy ...

- All options or parameters are usually optional, except when marked as required.

- Options are to be provided as strings unless in some cases where numbers can be used.

- **Upper and lower case** is important when specifying the parameters and options.

- Parameters and options are set in a Monospace type like this: `someParameter`

- Options that can be enabled or disabled will show `true` and `false` as possible values, where `true` enables the option and `false` disables it.

Image Parameters

These options can be given for each individual image.

- `url` – (required) the URL of the image to show

- `title` – the title of the image shown in its caption

- `description` – the description shown it the image's caption

- `linkUrl` – where the caption and/or image are linked to

- `caption` – allows to change the default location and dimensions of the caption

  ○ `left` – distance with respect to the left margin

- ○ `right` – distance with respect to the right margin

- ○ `top` – distance with respect to the top margin

- ○ `bottom` – distance with respect to the bottom margin

These parameters determine the position of the caption with respect to the WunderSlider container, values can be given in pixels (px) or as a percentage (%). When none is specified, pixels is assumed. Percentages are relative to the container's dimensions.

Sensible combinations are specifying *left* and *top*, *right* and *top*, *left* and *bottom* or *right* and *bottom*.

- ○ `width` – width of the caption

- ○ `height` – height of the caption

These dimensions are used to set the size of the caption. Values can be given in pixels (px) or as a percentage (%). When none is specified, pixels is assumed. Percentages are relative to the container's dimensions.

General Options

These options are applicable to an instance of the WunderSlider.

Skins / Themes

WunderSlider comes with different skins that are selected by using the appropriate stylesheets. Each skin is provided in a subfolder of the build folder:

- • `default` – The default skin.

- • `box` – The Box skin.

- • `dotsy` – The Dotsy skin.

- • `dotsy-dark` – The Dotsy Dark skin.

- • `dotsy-light` – Use The Dotsy Light skin.

- • `glass` – The Glass skin.

- • `glass-light` – The Glass Light skin.

- • `zenspace` – The Zen Space skin.

Example – To use the box skin, add these lines in the head section of your page:

```
<link rel="stylesheet" type="text/css" href="box/default/wunderslider-min.css"
/>
<!--[if lt IE 9]>
<link rel="stylesheet" type="text/css" href="box/default/wunderslider-ie-
min.css" />
<![endif]-->
```

Display modes

- mode – (required) Sets the image display mode .

    - fixed – Images are used "as is", at their original size.

    - proportional – Images are resized and cropped to fit the container .

- display - Determines how the WunderSlider will be displayed or adjusted to the available space .

    - fixed – (default) The container has a fixed size, the dimensions are determined by the width and height attributes which default to 800x500 pixels .

    - fill – The WunderSlider will fill the available space .

    - proportional – The WunderSlider will adopt a size proportional to the available space .

- autoAdjust – When display is fill, will set slide properties to take up 'all space' around it (e.g. in body, fullscreen). When display is fixed, the container's dimensions are set to those determined by the width and height attributes.

    - true – (default)

    - false

Size

- width – Width of the WunderSlider in pixels, defaults to 800 .

- height – Height of the WunderSlider in pixels, defaults to 500 .

Controls

- `useCaption` – When enabled, image captions with title and description will be shown when set.

  - `true` – (default)

  - `false`

- `useSelectors` – Image selectors that allow to switch to any of the attached images will be shown when this is enabled.

  - `true` – (default)

  - `false`

- `useNav` – The next and previous image navigation buttons are shown when enabled.

  - `true` – (default)

  - `false`

- `useThrobber` – A throbber is shown while the slider loads the first image.

  - `true` – (default)

  - `false`

- `useFlick` – Enables flicking (swiping) to flip to the next or previous image by dragging over the WunderSlider. Works with touch-enabled devices as well.

  - `true` – (default) Note that this option will only take effect when the `clickable` option is disabled.

  - `False`

- `flickDistanceFactor` – Determines the distance relative to the WunderSlider's width so that a flick is recognized as valid. Defaults to 0.25 .

- `clickable` – If this option is enabled and a link has been specified for an image by providing the linkUrl parameter, clicking the image will direct the visitor to the given URL.

  - `true` – (default)

  - `false`

- `mouseOverPause` – The slideshow will pause when the pointer is placed over it .

- ○ `true` – (default)

- ○ `false`

Captions

- `caption` – allows to change the default location and dimensions of the caption in general. The same options as for individual images apply. Please refer to [Image Parameters](#) for details.

- `captionContentElement` – This determines the HTML element that encloses the caption content, title and description.

  - ○ `div` – (default)

  - ○ `span`

Effects

- `effect` or `effects` – Determines the effect(s) that are used to transition from one image to the next in turn. A single value can be given by using the singular form `effect`, which will be applied to all images, or an array of values can be given by using the plural form `effects` which will be applied in sequence. When fewer entries than images are supplied, they are rotated through.

  These effects are available:

  `random` – On each transition, an effect is chosen randomly.

  `none` – No effect is applied.

  `fadeIn` – Fade in .

  `checkers` – Checkers .

  `slideDown` – Slides the new image down into view.

  `slideUp` – Slide up .

  `slideLeft` – Slide left .

  `slideRight` – Slide right .

`slideLeftDown` – Slide left down .

`slideRightDown` – Slide right down .

`slideLeftUp` – Slide left up .

`slideRightUp` – Slide right up .

`joinUp` – Horizontal stripes fly in upwards composing the image into view.

`joinDown` – Join down .

`joinLeft` – Join left .

`joinRight` – Join right .

`joinLeftDown` – Blocks fly in downwards from the left to compose the image.

`joinRightDown` – Join right down .

`joinLeftUp` – Join left up .

`joinRightUp` – Join right up .

`hBlinds` – Horizontal blinds .

`vBlinds` – Vertical blinds .

`hStripes` – Horizontal stripes alternating from both sides.

`vStripes` – Vertical stripes .

`stripes` – Flying blocks … if you were to slice up and combine vertical and horizontal stripes, you would end up with this effect.

`hJoinStripes` – Horizontal composition with blocks building up the image.

`vJoinStripes` – Vertical composition.

`unplode` *aka* `joinStripes` – Unplode : the reverse of an implosion.

`hCurtain` – Horizontal curtain .

`vCurtain` – Vertical curtain .

`curtain` – Magic curtain.

`runRightDown` – Running right down, block by block, building up the image from left to right and top-down .

`runLeftDown` – Run left down .

`runRightUp` – Run right up .

`runLeftUp` – Run left up .

`runDownRight` - Run down right, top-down left to right .

`runDownLeft` - Run down left .

`runUpRight` - Run up right .

`runUpLeft` - Run up left .

`scanDown` – Scan down block by block, top-down zig-zagging from left to right, from right to left .

`scanUp` – Scan up .

`scanLeft` – Scan left .

`scanRight` – Scan right .

`snakeIn` – Snake in , spiraling block by block.

`snakeOut` – Snake out.

- `animateInterval` – If not null or zero, sets the duration in miliseconds of some animation effects, default value is null, maximum 5000 ms (calculated according to effect duration) .

- `morph` – Defaults to 'opacity', no other values are currently accepted .

- `hzones` – Number of horizontal blocks used in effects . Possible values:

  - A number between 1 and 16, defaults to 8

  - `random` – Will choose an arbitrary number .

- `vzones` – Number of vertical blocks used in effects . Options:

  - When not set (default), will automatically adjust the number of vertical blocks used so that their shape is square, according to the number of horizontal blocks used and the size of the WunderSlider.

  - A number between 1 and 16.

  - `random` – Will choose an arbitrary number .

Overlay

- `overlayOpacity` – Sets the opacity of the overlay mesh .

  - A decimal number between 0 and 1, defaults to 0.3 .

- `overlay` – The name of the overlay to use, if none is given, no overlay is used (this is the default setting) . Any of these overlays can be chosen:

  Black meshes:

  - `chain`

  - `gilgamesh`

  - `mesh`

  - `polka`

  - `stripes`

  - `thickmesh`

  White meshes :

- ○ chain-white

  - ○ gilgamesh-white

  - ○ mesh-white

  - ○ polka-white

  - ○ stripes-white

  - ○ thickmesh-white

Transitions

- randomize – If enabled, shows images in random order.

  - ○ true

  - ○ false – (default)

- period – How long an image will be displayed until showing the next one. The value given is in milliseconds.

  - ○ A number greater than zero, defaults to 5000 milliseconds (for 5 seconds) .

- duration – How long the transition between images should last. The value given is in milliseconds .

  - ○ A number greater than zero, defaults to 600 (for 0.6 seconds) – the actual duration can be longer depending on the performance of the browser.

- fps – Attempted framerate for effects . The value given is in frames per second.

  - ○ A number between 1 and 24, the default is 16 .

Other

- zIndex – determines the base z-index for the WunderSlider.

- `useShadow` – If enabled, adds a `div` with class `shadow` to the WunderSlider. This is currently not used but is provided as a convenience if needed.

    - `true`

    - `false` – (default)

- `preloadImages` – If greater than 0, preloads as many images before and next to the current image. The default value is 1.

- `basepath` – must be given when the location of the resources (overlay images) is different than the default

- `appendTo` – appends the WunderSlider container to the given element; uses jQuery selector syntax, i.e. prepend `#` for IDs and `.` for classes.

- `buttonEffects` – If enabled, applies effects when one of the controls is used to change the image.

    - `true`

    - `false` – (default)

WunderSlider XML Specification

- All options in the General Options section are provided as attributes to the `<wunderslider>` element as in the examples.

- To specify general caption options, add `<caption>` element as a child of the `<wunderslider>` element:

```
<wunderslider ...>
    <caption … />
    <image … />
    ...
</wunderslider>
```

- All options in the Image Parameters section are provided as attributes to the `<image>` tag.

- To specify image caption options, add a `<caption>` element as a child of an `<image>` element:

```
<image ...>
    <caption … />
</image>
```

Creating an instance

Creating an instance to read the WunderSlider XML Specification from a file:

```php
<?php
    require_once( 'class-wunderslider-xml-parser.php' );
    $p = new WunderSlider_XML_Parser( 'wunderslider-fixed.xml' );
?>
```

Creating an instance passing the specification as a parameter:

```php
<?php
    require_once( 'class-wunderslider-xml-parser.php' );
    $xml = '<wunderslider mode="proportional" mouseOverPause="true">
<image url="images-demo/image-0.jpg" title="..." description="..." />
<image url='images-demo/image-1.jpg' title="..." description="..." />
</wunderslider>';
    $p = new WunderSlider_XML_Parser( $xml );
?>
```

Rendering

Once an instance has been created, to embed the full WunderSlider code, including scripts and stylesheets:

```php
<?php
    echo $p->code;
?>
```

Alternatively print the required parts where appropriate using:

- `$p->links` to produce the link tags that load the skin's CSS files

- `$p->container` to render the div tag that will accommodate the WunderSlider on the page

- `$p->scripts` to render the script tags that load the required WudnerSlider and jQuery Javascript – to avoid loading WunderSlider's included jQuery see below.

- `$p->javascript` to print the Javascript that builds the WunderSlider

A full example is provided in the file `wunderslider-fixed-separate.php` located in the `demo` folder.

jQuery and Javascript files loaded

WunderSlider will automatically produce a `script` tag to load both its included version of jQuery as well as its own script, when rendering either the full code through `echo $p->code;` or only the scripts using `echo $p->scripts;`

If you are already loading jQuery elsewhere, set `$p->jquery = false;` before invoking either of

the above so that jQuery is not referenced twice.

Note that in any case jQuery has to be included **before** WunderSlider's own script.

Paths

By default, the paths for all resources including stylesheets and scripts used are relative to the location of the `class-wunderslider-xml-parser.php` script.

If the location of the resources is different, the `$p->basepath` property is used.

Assuming the following folder structure at the web root of the domain www.example.com

```
resources
  css
    … (WunderSlider skin folders)
  images
    … (WunderSlider mesh images etc.)
  js
    jquery.x.y.z.min.js
    wunderslider-min.js
includes
  class-wunderslider-xml-parser.php
index.php
```

With index.php located at the web root of the domain, if we want to include the WunderSlider script we need to specify the basepath property:

```php
require_once( 'class-wunderslider-xml-parser.php' );
$p = new WunderSlider_XML_Parser( $content );
$p->basepath = 'http://www.example.com/resources/';
```

Multiple WunderSliders and the Container ID

By default, `$p->container` will produce a `div` tag with the id `wunderslider`. If you want to place several instances on one page, you need to specify different ids explicitly for each instance:

```php
require_once( 'class-wunderslider-xml-parser.php' );
$p = new WunderSlider_XML_Parser( $content );
$p->container_id = 'slider-5';
```