# Shadowbox.js

Home Download Support Usage Options API FAQ Forum

Shadowbox was designed with both the web hobbyist and the experienced programmer in mind. As such, it can easily be added to any web page with very little time and effort using the default settings. At the same time, Shadowbox offers a great deal of flexibility for those users who need to fine-tune the details of its appearance and operation for a given project.

This page contains detailed instructions on how to set up and run Shadowbox from both perspectives.

## Installation

After you download a copy of the code with all of the options you need, decompress the archive (unzip it) and upload the entire folder to your web server. The exact location is not important since Shadowbox will automatically detect where it is installed when it runs. However, it is important that you keep all of the files together. After you have uploaded the code to your server, simply link to the JavaScript and CSS files as described below.

## Setup

The simplest way to set up Shadowbox is to include the JavaScript and CSS files in the `<head>` of your document (web page) and then call `Shadowbox.init`, like this:

```
<link rel="stylesheet" type="text/css" href="shadowbox.css">
<script type="text/javascript" src="shadowbox.js"></script>
<script type="text/javascript">
Shadowbox.init();
</script>
```

The following example is a bit more complex and uses an object literal to specify some options. Options passed to this method will become the default for all instances of Shadowbox on the page.

```
<link rel="stylesheet" type="text/css" href="shadowbox.css">
<script type="text/javascript" src="shadowbox.js"></script>
<script type="text/javascript">
Shadowbox.init({
    handleOversize: "drag",
    modal: true
});
</script>
```

## Markup

Next you need to tell Shadowbox which links you want it to open. The simplest way to do this is through your HTML markup. If you're going to take this route, at the very least you must add a `rel="shadowbox"` attribute to each link. For example, say you have this link to an image on your page:

```
<a href="myimage.jpg">My Image</a>
```

In order to set up this link for use with Shadowbox, simply change it to this:

```
<a href="myimage.jpg" rel="shadowbox">My Image</a>
```

That's it! Clicking on this link should now open up the image in Shadowbox.

> **NOTE:** The word `lightbox` will also work here. This feature was added for compatibility with the original Lightbox script. Also note that because HTML area tags do not support the `rel` attribute, you cannot use this method to set them up for use with Shadowbox. Instead, use `Shadowbox.setup` as described below.

If you would like to display a title for your image, simply add a `title` attribute to the link.

```
<a href="myimage.jpg" rel="shadowbox" title="My Image">My Image</a>
```

You must explicitly tell Shadowbox the dimensions to use to display content other than images. This is done by adding a few parameters to the end of the `rel` attribute, separated by semi-colons. To specify a movie's height and width (in pixels), use the `height` and `width` parameters.

```
<a href="mymovie.swf" rel="shadowbox;height=140;width=120">My Movie</a>
```

In addition to displaying single images and movies, Shadowbox is also capable of displaying galleries of content. In order to designate a link as part of a gallery, you must add the gallery name to the `rel` attribute between square brackets immediately following the word `shadowbox`. The following markup creates a gallery called "Vacation" with two pictures.

```
<a href="beach.jpg" rel="shadowbox[Vacation]">The Beach</a>
<a href="pier.jpg" rel="shadowbox[Vacation]">The Pier</a>
```

Galleries may be composed of content of many different types. The following markup demonstrates how various media can be combined into a single gallery.

```
<a rel="shadowbox[Mixed];" href="myimage.jpg">jpg</a>
<a rel="shadowbox[Mixed];width=520;height=390" href="myswf.swf">swf</a>
<a rel="shadowbox[Mixed];width=292;height=218" href="mymovie.mp4">movie</a>
<a rel="shadowbox[Mixed]" href="mywebsite.html">iframe</a>
```

## Advanced

If you don't want to add to your markup, you don't have to. Shadowbox may be manipulated and controlled with only JavaScript. This use is slightly more complex, but it has several benefits including the fact that your HTML markup will be cleaner and you can more easily integrate Shadowbox into an existing project.

If you were paying attention in the section about markup, you'll notice that there are several properties that are commonly present on a Shadowbox content object. They are listed in the table below.

| Property | Description |
|---|---|
| content | The actual content of the object (e.g. URL, HTML code, etc.) |
| player | The abbreviated name of the player to use for the object (optional, can be automatically determined in most cases) |
| title | The title to use for the object (optional) |
| gallery | The name of the gallery the object belongs to (optional) |
| height | The height of the object (in pixels, only required for movies and Flash) |
| width | The width of the object (in pixels, only required for movies and Flash) |
| options | A set of options to use for this object |

When using markup, each of these properties is automatically derived from the link element itself using the `Shadowbox.buildObject` method. But now that you know what's really going on behind the scenes, you can just pass objects (or arrays of objects) that contain these properties to `Shadowbox.open` as in the following example:

```
<link rel="stylesheet" type="text/css" href="shadowbox.css">
<script type="text/javascript" src="shadowbox.js"></script>
<script type="text/javascript">
Shadowbox.init({
    // let's skip the automatic setup because we don't have any
    // properly configured link elements on the page
    skipSetup: true
});

window.onload = function() {

    // open a welcome message as soon as the window loads
    Shadowbox.open({
        content:    '<div id="welcome-msg">Welcome to my website!</div>',
        player:     "html",
        title:      "Welcome",
        height:     350,
        width:      350
    });
};
</script>
```

You can also use `Shadowbox.setup` to manually set up link elements for use with Shadowbox. This can be useful, for example, if you have a page that is updated via Ajax with links being created and destroyed dynamically.

```
<link rel="stylesheet" type="text/css" href="shadowbox.css">
<script type="text/javascript" src="shadowbox.js"></script>
<script type="text/javascript">
Shadowbox.init({
    // skip the automatic setup again, we do this later manually
    skipSetup: true
});

window.onload = function() {

    // set up all anchor elements with a "movie" class to work with Shadowbox
    Shadowbox.setup("a.movie", {
        gallery:            "My Movies",
        autoplayMovies:     true
    });

};
</script>
```

When using this method to set up links with Shadowbox, you may mix in the link parameters with the options argument as the second parameter. However, the same parameters will apply to all links set up in the same call, so you may want to make separate calls to `Shadowbox.setup` for each link.

**NOTE:** The first parameter to `Shadowbox.setup` may be a single link element, an array of elements, a CSS selector (assuming you included CSS support when you downloaded the code), or an array in which the first item is a CSS selector and the second item is the context element under which the selection should take place.