

Lazy Load Plugin for jQuery

Lazy Load is delays loading of images in long web pages. Images outside of viewport are not loaded until user scrolls to them. This is opposite of [image preloading](#).

license MIT

Using Lazy Load on long web pages will make the page load faster. In some cases it can also help to reduce server load.

release v1.9.5

Plugin is inspired by [YUI ImageLoader](#) Utility by Matt Mlinac.

stars 4275

For those in hurry there are several demo pages: [basic options](#), [with fadein effect](#), [noscript fallback](#), [horizontal scrolling](#), [horizontal scrolling inside container](#), [vertical scrolling inside container](#), [page with gazillion images](#), [load images using timeout](#) and [load images using AJAX\(H\)](#).

forks 1103

When checking the demos clear browser cache between each request. You can check what is actually loaded with developers console (Chrome, Safari and IE) or FireBug (Firefox).

How to Use?

Lazy Load depends on [jQuery](#). Include them both in end of your HTML code:

```
<script src="jquery.js"></script>
<script src="jquery.lazyload.js"></script>
```

You must alter your image tags. Address of the image must be put into `data-original` attribute. Give lazy loaded images a specific class. This way you can easily control which images plugin is binded to.

```
<img class="lazy" data-original="img/example.jpg" width="640" height="480">
```

```
$(function() {
    $("img.lazy").lazyload();
});
```

This causes all images of class `lazy` to be lazy loaded. See the [basic options](#) demo.

PRO TIP! You must set image dimensions either as `width` and `height` attributes or in CSS. Otherwise plugin might not work properly.

Setting Threshold

By default images are loaded when they appear on the screen. If you want images to load earlier use `threshold` parameter. Setting threshold to 200 causes image to load 200 pixels before it appears on viewport.

```
$("img.lazy").lazyload({
    threshold : 200
});
```

Event to Trigger Loading

You can use jQuery event such as `click` or `mouseover`. You can also use custom events such as `sparty` or `foobar`. Default is to wait until user scrolls down and image appears on the viewport. To load images only when user clicks them you could do:

```
$("#img.lazy").lazyload({
    event : "click"
});
```

PRO TIP! You can use this for tricks like delayed loading of images. Following code waits five seconds after rest of page has finished loading before it loads images. See it working at [delayed loading demo](#).

```
$(function() {
    $("#img.lazy").lazyload({
        event : "sporty"
    });
});

$(window).bind("load", function() {
    var timeout = setTimeout(function() {
        $("#img.lazy").trigger("sporty")
    }, 5000);
});
```

Using Effects

By default plugin waits for image to fully load and calls `show()`. You can use any effect you want. Following code uses `fadeIn` effect. Check how it works at [effect demo page](#).

```
$("#img.lazy").lazyload({
    effect : "fadeIn"
});
```

Fallback for Non JavaScript Browsers

Practically everyone has JavaScript enabled. However if you still want to support non JavaScript users you can include the real image tag inside

```
<img class="lazy" data-original="img/example.jpg" width="640" height="480">
<noscript>
    
</noscript>
```

To prevent both placeholder and the real image showing at the same time hide the placeholder with css.

```
.lazy {
    display: none;
}
```

For JavaScript enabled browser you must enable displaying the placeholders when documents loads. This can be done at the same time when initializing the plugin.

```
$("#img.lazy").show().lazyload();
```

HEADS UP! All this is optional can should be done only if you want to support non JavaScript users.

Images Inside Container

You can also use plugin for images inside scrolling container, such as div with scrollbar. Just pass the container as jQuery object. There is a demo for [horizontal](#) and [vertical](#) container.

```
#container {
```

```
height: 600px;
overflow: scroll;
}

$("img.lazy").lazyload({
    container: $("#container")
});
```

When Images Are Not Sequential

After scrolling page plugin loops through unloaded images. Loop checks if image has become visible. By default loop is stopped when first image outside viewport is found. This is based on following assumption. Order of images on page is same as order of images in HTML code. With some layouts assumption this might be wrong. You can control loading behaviour with `failure_limit` setting.

```
$("img.lazy").lazyload({
    failure_limit : 10
});
```

Setting `failure_limit` to 10 causes plugin to stop searching for images to load after finding 10 images below the fold. If you have a funky layout set this number to something high. Worst case being the actual number of images.

Dealing With Invisible Images

There are cases when you have images which are in viewport but not `:visible`. To improve performance you can ignore `.not(":visible")` images.

```
$("img.lazy").lazyload({
    skip_invisible : true
});
```

HEADS UP! Webkit browsers will report images with without `width` and `height` as not `.not(":visible")`. This causes images to appear only when you scroll a bit. Either fix your image tags or keep `skip_invisible` as `false`. Use this feature only if you know what you are doing.