jQuery MiniColors

Now with Bootstrap 3 support!

A project by [A Beautiful Site](http://www.abeautifulsite.net/), originally developed for [Surreal CMS](http://www.surrealcms.com/).

## Contents

* [Download](#gjdgxs)
* [Demos](#30j0zll)
* [API](#1fob9te)
  + [Instantiation](#3znysh7)
  + [Settings](#2et92p0)
  + [Methods](#tyjcwt)
  + [Events](#3dy6vkm)
* [License](#1t3h5sf)

## Download

This project is on GitHub. Feel free to post bug reports, feature requests, and code improvements on the official project page.

[Download on GitHub](https://github.com/claviska/jquery-minicolors)

## Demos

This is the main demo page, which uses [Bootstrap 3](http://getbootstrap.com/), but this plugin works without Bootstrap as well.

[View Demo Without Bootstrap](http://docs.google.com/without-bootstrap.html)

### Control Types

Hue (default)

Saturation

Brightness

Wheel

### Input Modes

Text field

Hidden Input

Inline

### Positions

Valid positions include bottom left, bottom right, top left, and top right.

bottom left (default)

top left

bottom right

top right

### RGB(A)

RGB

RGB input can be assigned by setting the format option to rgb.

RGBA

RGBA input can be assigned by setting the format option to rgb and opacity option to true.

### …and more!

Opacity

Opacity can be assigned by including the data-opacity attribute or by setting the opacity option to true.

Keywords

CSS-wide keywords can be assigned by setting the keywords option to a comma-separated list of valid keywords: transparent, initial, inherit.

Default Value

This field has a default value assigned to it, so it will never be empty.

Letter Case

This field will always be uppercase.

Input Groups

Button

Example using Bootstrap's input groups.

More Input Groups

Color Button

Input group example with addon.

Swatches

Example with swatches.

Swatches and opacity

Example with swatches and opacity.

## API

### Instantiation

Instantiate like any other jQuery plugin:

$('INPUT.minicolors').minicolors(settings);

### Settings

Default settings are as follows:

$.minicolors = {  
 defaults: {  
 animationSpeed: 50,  
 animationEasing: 'swing',  
 change: null,  
 changeDelay: 0,  
 control: 'hue',  
 defaultValue: '',  
 format: 'hex',  
 hide: null,  
 hideSpeed: 100,  
 inline: false,  
 keywords: '',  
 letterCase: 'lowercase',  
 opacity: false,  
 position: 'bottom left',  
 show: null,  
 showSpeed: 100,  
 theme: 'default',  
 swatches: []  
 }  
};

For convenience, you can change default settings globally by assigning new values:

$.minicolors.defaults.changeDelay = 200;

To change multiple properties at once, use $.extend():

$.minicolors.defaults = $.extend($.minicolors.defaults, {  
 changeDelay: 200,  
 letterCase: 'uppercase',  
 theme: 'bootstrap'  
});

**Note:** Changing default settings will *not* affect controls that are already initialized.

animationSpeed

The animation speed of the sliders when the user taps or clicks a new color. Set to 0 for no animation.

animationEasing

The easing to use when animating the sliders.

changeDelay

The time, in milliseconds, to defer the change event from firing while the user makes a selection. This is useful for preventing the change event from firing frequently as the user drags the color picker around.

The default value is 0 (no delay). If your change callback features something resource-intensive (such as an AJAX request), you’ll probably want to set this to at least 200.

control

Determines the type of control. Valid options are hue, brightness, saturation, and wheel.

defaultValue

To force a default color, set this to a valid hex string. When the user clears the control, it will revert to this color.

format

The format miniColors should use. Valid options are hex and rgb.

hideSpeed & showSpeed

The speed at which to hide and show the color picker.

inline

Set to true to force the color picker to appear inline.

keywords

A comma-separated list of keywords that the control should accept (e.g. inherit, transparent, initial). By default, no keywords are allowed.

letterCase

Determines the letter case of the hex code value. Valid options are uppercase or lowercase.

opacity

Set to true to enable the opacity slider. (Use the input element's data-opacity attribute to set a preset value.)

position

Sets the position of the dropdown. Valid options are bottom left, bottom right, top left, and top right.

The swatchPosition setting has been removed in version 2.1. The position of the swatch is now determined by position.

theme

A string containing the name of the custom theme to be applied. In your CSS, prefix your selectors like this:

.minicolors-theme-yourThemeName { ... }

If you are using the default theme, you will probably need to adjust the swatch styles depending on your existing stylesheet rules. Version 2.1 removes as much styling on the input element as possible, which means it’s up to you to adjust your CSS to make sure the swatch aligns properly.

To adjust the swatch, override these styles:

.minicolors-theme-default .minicolors-swatch {  
 top: 5px;  
 left: 5px;  
 width: 18px;  
 height: 18px;  
}  
.minicolors-theme-default.minicolors-position-right .minicolors-swatch {  
 left: auto;  
 right: 5px;  
}

swatches

An array containing some colors, in either rgb(a) or hex format, that will show up under the main color grid.

### Methods

Use this syntax for calling methods:

$(*selector*).minicolors('method', *[data]*);

create

Initializes the control for all items matching your selector. This is the default method, so data may be passed in as the only argument.

To set a preset color value, populate the value attribute of the original input element.

destroy

Returns the *input* element to its original, uninitialized state.

opacity

Gets or sets a control's opacity level. To use this method as a setter, pass data in as a value between 0 and 1. (You can also obtain this value by checking the input element's data-opacity attribute.)

To set a preset opacity value, populate the data-opacity attribute of the original input element.

rgbObject

Returns an object containing red, green, blue, and alpha properties that correspond to the control's current value. Example:

{ r: 0, g: 82, b: 148, a: 0.75 }

rgbString & rgbaString

Returns an RGB or RGBA string suitable for use in your CSS. Examples:

rgb(0, 82, 148)  
rgba(0, 82, 148, .75)

settings

Gets or sets a control's settings. If new settings are passed in, the control will destroy and re-initialize itself with any new settings overriding the old ones.

value

Gets or sets a control's color value. To use this method as a setter, pass in a color string or an object (ex: {color: '#fc0', opacity: .5}).

### Events

change

Fires when the value of the color picker changes. The this keyword will reference the original input element.

$(*selector*).minicolors({  
 change: function(value, opacity) {  
 console.log(value + ' - ' + opacity);  
 }  
});

**Warning!** This event will fire a lot when the user drags the color picker around. Use the changeDelay setting to reduce its frequency.

hide

Fires when the color picker is hidden. The this keyword will reference the original input element.

$(*selector*).minicolors({  
 hide: function() {  
 console.log('Hide event triggered!');  
 }  
});

show

Fires when the color picker is shown. The this keyword will reference the original input element.

$(*selector*).minicolors({  
 show: function() {  
 console.log('Show event triggered!');  
 }  
});

## License

Licensed under the [MIT license](http://opensource.org/licenses/MIT), same as [jQuery](https://jquery.org/license/).

©2013 [A Beautiful Site, LLC.](http://www.abeautifulsite.net/)