# color-convert

[Build Status](https://travis-ci.org/Qix-/color-convert)

Color-convert is a color conversion library for JavaScript and node. It converts all ways between rgb, hsl, hsv, hwb, cmyk, ansi, ansi16, hex strings, and CSS keywords (will round to closest):

var convert = require('color-convert');

convert.rgb.hsl(140, 200, 100); // [96, 48, 59]

convert.keyword.rgb('blue'); // [0, 0, 255]

var rgbChannels = convert.rgb.channels; // 3

var cmykChannels = convert.cmyk.channels; // 4

var ansiChannels = convert.ansi16.channels; // 1

# Install

$ npm install color-convert

# API

Simply get the property of the *from* and *to* conversion that you're looking for.

All functions have a rounded and unrounded variant. By default, return values are rounded. To get the unrounded (raw) results, simply tack on .raw to the function.

All 'from' functions have a hidden property called .channels that indicates the number of channels the function expects (not including alpha).

var convert = require('color-convert');

// Hex to LAB

convert.hex.lab('DEADBF'); // [ 76, 21, -2 ]

convert.hex.lab.raw('DEADBF'); // [ 75.56213190997677, 20.653827952644754, -2.290532499330533 ]

// RGB to CMYK

convert.rgb.cmyk(167, 255, 4); // [ 35, 0, 98, 0 ]

convert.rgb.cmyk.raw(167, 255, 4); // [ 34.509803921568626, 0, 98.43137254901961, 0 ]

### Arrays

All functions that accept multiple arguments also support passing an array.

Note that this does **not** apply to functions that convert from a color that only requires one value (e.g. keyword, ansi256, hex, etc.)

var convert = require('color-convert');

convert.rgb.hex(123, 45, 67); // '7B2D43'

convert.rgb.hex([123, 45, 67]); // '7B2D43'

## Routing

Conversions that don't have an *explicitly* defined conversion (in [conversions.js](http://conversions.js)), but can be converted by means of sub-conversions (e.g. XYZ -> **RGB** -> CMYK), are automatically routed together. This allows just about any color model supported by color-convert to be converted to any other model, so long as a sub-conversion path exists. This is also true for conversions requiring more than one step in between (e.g. LCH -> **LAB** -> **XYZ** -> **RGB** -> Hex).

Keep in mind that extensive conversions *may* result in a loss of precision, and exist only to be complete. For a list of "direct" (single-step) conversions, see [conversions.js](http://conversions.js).

# Contribute

If there is a new model you would like to support, or want to add a direct conversion between two existing models, please send us a pull request.

# License

Copyright © 2011-2016, Heather Arthur and Josh Junon. Licensed under the [MIT License](http://license).