ZINCKSOFT

Give The Dream New Life.

JSU.Color

The complete Module API Reference

Quick Introduction

The JSU Color Module (**JSU.Color / JSUColor**) provides methods for converting between color spaces such as RGB, CMYK, HSL, and so on. There are currently 12 color spaces supported.

Including the module in your page

Before including this module in your page, first include the core module:

```
<script type="text/javascript" src="jsu.core.js"></script>

<script type="text/javascript" src="jsu.color.js"></script>
```

JSU.Color API Reference

Methods

1. **dg2rad** and **rad2dg**

Syntax:

```
dg2rad(dg)
rad2dg(rad)
```

Arguments:

- ***dg***: the degrees that will be converted to radians;
- ***rad***: the radians that will be converted to degrees.

Returns: A float number.

Example:

```
alert(JSU.Color.dg2rad(270)); // 4.7123889
alert(JSU.Color.rad2dg(4.7123889)); // 269.999999
```

The reason for these methods being inside the color module is that they're used for converting between some color spaces.

2. **getRed**, **getGreen** and **getBlue**

Syntax:

```
getRed(c)
getGreen(c)
getBlue(c)
```

Arguments:

- ***c***: the color combination from which to extract the specified color part - either a hex value, a css rgb value (with or without percentages), or individual rgb parts separated by comma; these formats are allowed:

```
hex: “#xxxxxx”, “#xxx”, “xxxxxx”, “xxx”
css: “rgb(x[%],x[%],x[%])”
rgb: “(x[%],x[%],x[%])”, “x[%],x[%],x[%]”
```

Returns: A number representing the needed color part.

Example:

```
alert(JSU.Color.getRed("rgb(12, 14,15)")); // 12
alert(JSU.Color.getRed("121,15,17")); // 121
alert(JSU.Color.getRed("#ffaaff")); // 255
alert(JSU.Color.getRed("#ffa")); // 255
alert(JSU.Color.getRed("0fa0fa")); // 15
alert(JSU.Color.getRed("12%, 14, 15")); // 30.6
```

The other two methods work the same way.

3. hex2rgb / rgb2hex

Converts between HEX and RGB colors.

Syntax:

```
hex2rgb(c1 [, css [, percent [, decimals]]])
rgb2hex(c2 [, trim])
rgb2hex(r, g, b [, trim])
```

Arguments:

- ***c1***: a string containing the hex color to convert; formats allowed:

  ```
  hex: “#xxxxxx”, “#xxx”, “xxxxxx”, “xxx”
  ```

- ***c2***: the rgb color to convert; formats allowed:

  ```
  css: “rgb(x[%],x[%],x[%])”
  rgb: “(x[%],x[%],x[%])”, “x[%],x[%],x[%]”
  array: [x,x,x]
  ```

- ***r, g, b***: the individual color values;
- ***css:*** a boolean indicating whether to format the result in the css rgb format; default is false;

- ***percent***: a boolean indicating whether the result contains percentage values for the individual color parts; default is false;
- ***decimals***: the number of decimals for every color value; default is 0;
- ***trim***: a boolean indicating whether to trim the hex result to 3 characters only where possible (ex: #ffffff can be trimmed to #fff); default is false;

Returns: A string if any formatting is applied, or an array with each color value. As a note, it's best if you choose the array format for the color value argument so that you can chain multiple conversion methods to get more conversion possibilities (ex: there's no rgb2clab method, but you could use the rgb2xyz and xyz2clab methods chained together, but you'll have to work only with array values).

Example:

```
alert(JSU.Color.hex2rgb("#ff00ff")); // [255, 0, 255]
alert(JSU.Color.hex2rgb("f0f")); // [240, 255, 15]
alert(JSU.Color.hex2rgb("#f0f", true, true, 2)); // rgb(94.12%, 100%, 5.88%)

alert(JSU.Color.rgb2hex(10, 10, 10)); // #0a0a0a
alert(JSU.Color.rgb2hex([10, 10, 10])); // #0a0a0a
alert(JSU.Color.rgb2hex("rgb(30%, 20%, 45%)")); // #4c3372
alert(JSU.Color.rgb2hex("255,255,100%", true)); // #fff
alert(JSU.Color.rgb2hex(255, 255, 255, false)); // #ffffff
```

4. rgb2hsl / hsl2rgb

Converts between RGB and HSL/HSI colors.

Syntax:

```
rgb2hsl(rgb [, decimals [, css]])
hsl2rgb(hsl [, decimals [, css]])
```

Aliases: **rgb2hsi**, **hsi2rgb**

Arguments:

- ***rgb***: an array with each color part value;
- ***hsl***: either an array with each color part value, or a CSS 3 formatted hsl value:

```
hsl(x,x%,x%)
(x,x%,x%)
x,x%,x%
```

- ***decimals***: the number of decimals for the result;
- ***css***: a boolean indicating whether to format the result as a CSS 3 color value;

Returns: A string if css is true, or an array with each color value if css is false.

Example:

```
alert(JSU.Color.rgb2hsl([10, 15, 250])); // [239, 96, 51]
alert(JSU.Color.rgb2hsl([10, 15, 250], 2)); // [238.75, 96, 50.98]
alert(JSU.Color.rgb2hsl([10, 15, 250], 0, true)); // hsl(239,96%,51%)

alert(JSU.Color.hsl2rgb([239, 96, 51])); // [10, 14, 250]
alert(JSU.Color.hsl2rgb([239, 96, 51], 2)); // [10.1, 14.1, 250]
alert(JSU.Color.hsl2rgb([239, 96, 51], 0, true)); // rgb(10,14,250)
alert(JSU.Color.hsl2rgb("hsl(239,96%,51%)", 0, true)); // rgb(10,14,250)
```

5. rgb2hsv / hsv2rgb

Converts between RGB and HSV/HSB colors.

Syntax:

```
rgb2hsv(rgb [, decimals])
hsv2rgb(hsv [, decimals])
```

Aliases: **rgb2hsb**, **hsb2rgb**

Arguments:

- ***rgb, hsv***: an array with each color part value;
- ***decimals***: the number of decimals for the result;

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.rgb2hsv([10, 15, 20])); // [210,50,8]
alert(JSU.Color.hsv2rgb([210, 50, 8])); // [10, 15, 20]
```

6. rgb2yuv / yuv2rgb

Converts between RGB and YUV colors.

Syntax:

```
rgb2yuv(rgb [, decimals [, hdtv]])
yuv2rgb(yuv [, decimals])
```

Arguments:

- ***rgb, yuv***: an array with each color part value;
- ***decimals***: the number of decimals for the result;
- ***hdtv***: a boolean indicating whether to convert to an HDTV YUV color; default is false.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.rgb2yuv([10, 15, 250])); // [16,40,-10]
alert(JSU.Color.rgb2yuv([10, 15, 250], 0, true)); // [11,40,-10]

alert(JSU.Color.yuv2rgb([16, 40, -10])); // [12, 15, 248]
```

7. rgb2xyz / xyz2rgb

Converts between RGB and XYZ colors.

Syntax:

```
rgb2xyz(rgb [, decimals])
xyz2rgb(xyz [, decimals])
```

Arguments:

- ***rgb, xyz***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.rgb2xyz([10, 15, 250], 2)); // [17.55, 7.31, 90.93]
alert(JSU.Color.xyz2rgb([17.55, 7.31, 90.93])); // [10, 15, 250]
```

As a note, having more decimals in a color will produce finer results.

8. xyz2Yxy / Yxy2xyz

Converts between XYZ and Yxy colors.

Syntax:

```
xyz2Yxy(xyz [, decimals])
Yxy2xyz(Yxy [, decimals])
```

Arguments:

- ***xyz, Yxy***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.xyz2Yxy([17.55, 7.31, 90.93], 2)); // [7.31, 0.15, 0.06]
alert(JSU.Color.Yxy2xyz([7.31, 0.15, 0.06], 2)); // [18.27, 7.31, 96.25]
```

9. xyz2hlab / hlab2xyz

Converts between XYZ and Hunter-Lab colors.

Syntax:

```
xyz2hlab(xyz [, decimals])
hlab2xyz(hlab [, decimals])
```

Arguments:

- ***xyz, hlab***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.xyz2hlab([17.55, 7.31, 90.93], 2)); // [27.04,68.55,-180.48]
alert(JSU.Color.hlab2xyz([27.04, 68.55, -180.48], 2)); // [17.55,7.31,90.94]
```

10. xyz2clab / clab2xyz

Converts between XYZ and CIE-L*ab colors.

Syntax:

```
xyz2clab(xyz [, decimals])
clab2xyz(clab [, decimals])
```

Arguments:

- ***xyz, clab***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.xyz2clab([17.55, 7.31, 90.93], 2)); // [32.5,75.66,-104.72]
alert(JSU.Color.clab2xyz([32.5, 75.66, -104.72], 2)); // [17.55, 7.31, 90.93]
```

11. rgb2cmy / cmy2rgb

Converts between RGB and CMY colors.

Syntax:

```
rgb2cmy(rgb [, decimals])
cmy2rgb(cmy [, decimals])
```

Arguments:

- ***rgb, cmy***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.rgb2cmy([10, 0, 250])); // [96, 100, 2]
alert(JSU.Color.cmy2rgb([96, 100, 2])); // [10, 0, 250]
```

12. cmy2cmyk / cmyk2cmy

Converts between CMY and CMYK colors.

Syntax:

```
cmy2cmyk(cmy [, decimals])
cmyk2cmy(cmyk [, decimals])
```

Arguments:

- ***cmy, cmyk***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.cmy2cmyk([96, 100, 2])); // [96, 100, 0, 2]
alert(JSU.Color.cmyk2cmy([96, 100, 0, 2])); // [96, 100, 2]
```

13. rgb2cmyk / cmyk2rgb

Converts between RGB and CMYK colors.

Syntax:

```
rgb2cmyk(rgb [, decimals])
cmyk2rgb(cmyk [, decimals])
```

Arguments:

- ***rgb, cmyk***: an array with each color part value;
- ***decimals***: the number of decimals for the result.

Returns: An array with the converted color parts.

Example:

```
alert(JSU.Color.cmyk2rgb([96, 100, 0, 2])); // [10, 0, 250]
alert(JSU.Color.rgb2cmyk([10, 0, 250])); // [96, 100, 0, 2]
```