

[HTML](#)[CSS](#)[MORE ▼](#)

# HTML Colors

[< Previous](#)[Next >](#)

HTML colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

## Color Names

In HTML, a color can be specified by using a color name:

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

LightGray

Try it Yourself »

HTML supports 140 standard color names.

---

## Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

### Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

[Try it Yourself »](#)

## Text Color

You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

### Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

[Try it Yourself »](#)

## Border Color

You can set the color of borders:

Hello World

Hello World

Hello World

### Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>  
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Try it Yourself »

## Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

Same as color name "Tomato", but 50% transparent:

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

### Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>  
<h1 style="background-color:#ff6347;">...</h1>  
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>  
  
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>  
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

[Try it Yourself »](#)

## RGB Value

In HTML, a color can be specified as an RGB value, using this formula:

***rgb(red, green, blue)***

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

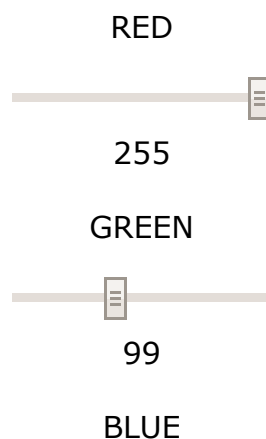
For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display the color black, all color parameters must be set to 0, like this: `rgb(0, 0, 0)`.

To display the color white, all color parameters must be set to 255, like this: `rgb(255, 255, 255)`.

Experiment by mixing the RGB values below:

**rgb(255, 99, 71)**





71

## Example

`rgb(255, 0, 0)``rgb(0, 0, 255)``rgb(60, 179, 113)``rgb(238, 130, 238)``rgb(255, 165, 0)``rgb(106, 90, 205)`

Try it Yourself »

Shades of gray are often defined using equal values for all the 3 light sources:

## Example

`rgb(0, 0, 0)`

`rgb(60, 60, 60)`

`rgb(120, 120, 120)`

`rgb(180, 180, 180)`

`rgb(240, 240, 240)`

`rgb(255, 255, 255)`

Try it Yourself »

## HEX Value

In HTML, a color can be specified using a hexadecimal value in the form:

**`#rrggbb`**

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, `#ff0000` is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

### Example

`#ff0000`

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

Try it Yourself »

Shades of gray are often defined using equal values for all the 3 light sources:

## Example

#000000

#3c3c3c

#787878



#b4b4b4

#f0f0f0

#ffffff

Try it Yourself »

## HSL Value

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

***hsl(hue, saturation, lightness)***

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

### Example

***hsl(0, 100%, 50%)***

***hsl(240, 100%, 50%)***

***hsl(147, 50%, 47%)***

`hsl(300, 76%, 72%)`

`hsl(39, 100%, 50%)`

`hsl(248, 53%, 58%)`

[Try it Yourself »](#)

## Saturation

Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

0% is completely gray, you can no longer see the color.

### Example

`hsl(0, 100%, 50%)`

`hsl(0, 80%, 50%)`

`hsl(0, 60%, 50%)`

`hsl(0, 40%, 50%)`

`hsl(0, 20%, 50%)`

`hsl(0, 0%, 50%)`

[Try it Yourself »](#)

## Lightness

The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).

### Example

`hsl(0, 100%, 0%)`

`hsl(0, 100%, 25%)`

`hsl(0, 100%, 50%)`

`hsl(0, 100%, 75%)`

`hsl(0, 100%, 90%)`

**hsl(0, 100%, 100%)**

[Try it Yourself »](#)

Shades of gray are often defined by setting the hue and saturation to 0, and adjust the lightness from 0% to 100% to get darker/lighter shades:

## Example

**hsl(0, 0%, 0%)**

**hsl(0, 0%, 24%)**

**hsl(0, 0%, 47%)**

**hsl(0, 0%, 71%)**

**hsl(0, 0%, 94%)**

**hsl(0, 0%, 100%)**

[Try it Yourself »](#)

# RGBA Value

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

**`rgba(red, green, blue, alpha)`**

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

## Example

**`rgba(255, 99, 71, 0)`**

**`rgba(255, 99, 71, 0.2)`**

**`rgba(255, 99, 71, 0.4)`**

**`rgba(255, 99, 71, 0.6)`**

**`rgba(255, 99, 71, 0.8)`**

**`rgba(255, 99, 71, 1)`**

Try it Yourself »

# HSLA Value

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

**`hsla(hue, saturation, lightness, alpha)`**

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

## Example

**`hsla(9, 100%, 64%, 0)`**

**`hsla(9, 100%, 64%, 0.2)`**

**`hsla(9, 100%, 64%, 0.4)`**

**`hsla(9, 100%, 64%, 0.6)`**

**`hsla(9, 100%, 64%, 0.8)`**

**`hsla(9, 100%, 64%, 1)`**

Try it Yourself »

[< Previous](#)

[Next >](#)

---

## COLOR PICKER



## HOW TO

- Tabs
- Dropdowns
- Accordions
- Convert Weights
- Animated Buttons
- Side Navigation
- Top Navigation

Modal Boxes  
Progress Bars  
Parallax  
Login Form  
HTML Includes  
Google Maps  
Range Sliders  
Tooltips  
Slideshow  
Filter List  
Sort List

## SHARE

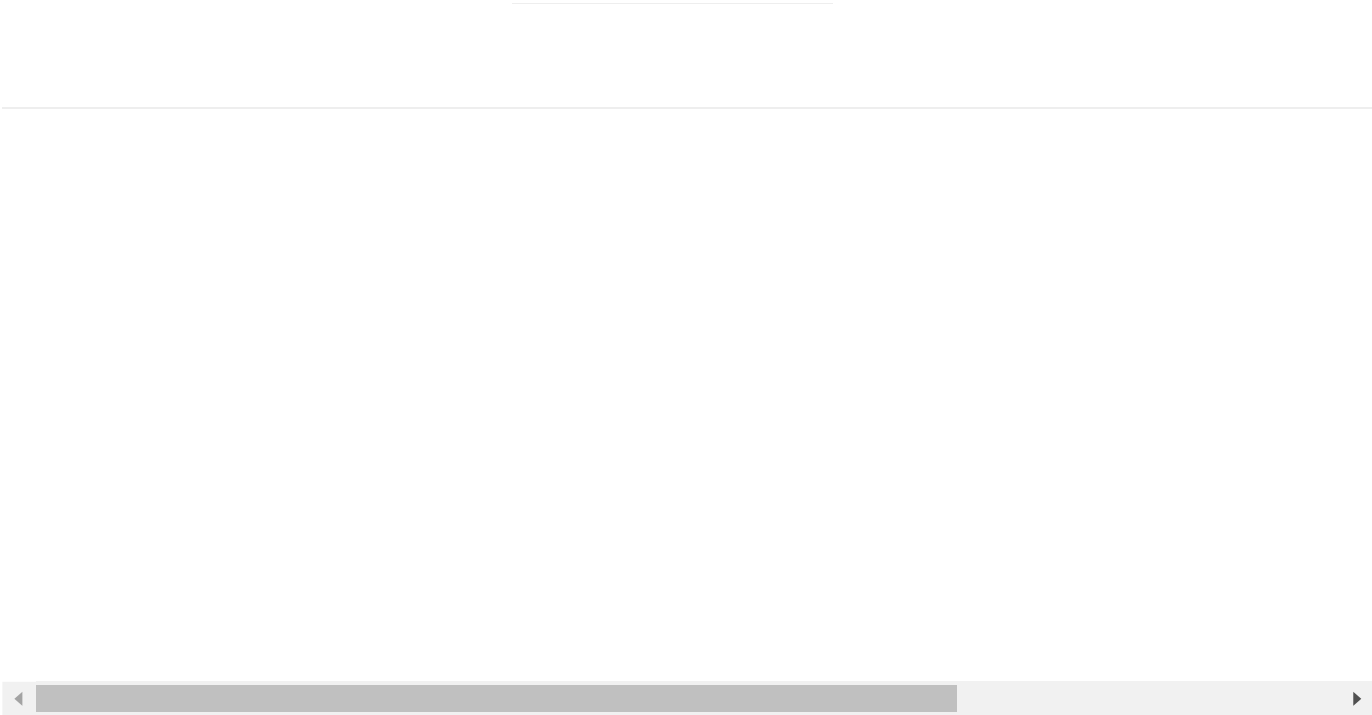


## CERTIFICATES

HTML, CSS, JavaScript, PHP, jQuery, Bootstrap and XML.

[Read More »](#)





FORUM  
ABOUT

---

## Top 10 Tutorials

HTML Tutorial  
CSS Tutorial  
JavaScript Tutorial  
W3.CSS Tutorial  
Bootstrap Tutorial  
SQL Tutorial  
PHP Tutorial  
jQuery Tutorial  
Angular Tutorial  
How To Tutorial

## Top 10 References

HTML Reference  
CSS Reference  
JavaScript Reference  
W3.CSS Reference  
Bootstrap Reference  
SQL Reference  
PHP Reference  
HTML Colors  
jQuery Reference  
AngularJS Reference

## Top 10 Examples

HTML Examples  
CSS Examples  
JavaScript Examples  
W3.CSS Examples  
Bootstrap Examples  
HTML DOM Examples  
PHP Examples  
jQuery Examples  
Angular Examples  
XML Examples

## Web Certificates

HTML Certificate  
CSS Certificate  
JavaScript Certificate  
jQuery Certificate  
PHP Certificate  
Bootstrap Certificate  
XML Certificate

---

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of

use, cookie and privacy policy. Copyright 1999-2018 by Refsnes Data. All Rights Reserved.  
Powered by W3.CSS.

