CSS Color Module Level 3

W3C Recommendation 19 June 2018

This version:

https://www.w3.org/TR/2018/REC-css-color-3-20180619/

Latest version:

https://www.w3.org/TR/css-color-3/

Previous version:

https://www.w3.org/TR/2018/PR-css-color-3-20180315/

Editor's Draft:

https://drafts.csswg.org/css-color-3/

Feedback:

GitHub Issues are preferred for discussion of this specification. When filing an issue, please put the text “css-color-3” in the title, preferably like this: “[css-color-3] *...summary of comment...*”. All issues and comments are archived, and there is also a historical archive.

Editors:

Tantek Çelik (Mozilla Corporation, and before at Microsoft Corporation) <tantek@cs.stanford.edu>

Chris Lilley (W3C) <chris@w3.org>

L. David Baron (Mozilla Corporation) <dbaron@dbaron.org>

Additional Authors:

Steven Pemberton (CWI) <steven.pemberton@cwi.nl>

Brad Pettit (Microsoft Corporation) <bradp@microsoft.com>

Test Suite:

http://test.csswg.org/suites/css3-color/nightly-unstable/

Please check the **errata** for any errors or issues reported since publication.

Abstract

CSS (Cascading Style Sheets) is a language for describing the rendering of HTML and XML documents on screen, on paper, in speech, etc. It uses color-related properties and values to color the text, backgrounds, borders, and other parts of elements in a document. This specification describes color values and properties for foreground color and group opacity. These include properties and values from CSS level 2 and new values.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at https://www.w3.org/TR/.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

This Edited Recommendation is identical to the 15 March 2018 Proposed Recommendation except for minor editorial corrections, as noted in the Changes section.

This document was produced by the CSS Working Group as a Proposed Recommendation. This document is intended to become a W3C Edited Recommendation.

A W3C Recommendation is a document that has been widely reviewed and is ready for implementation. W3C encourages everybody to implement this specification and return comments to GitHub issues.

This document was produced by a group operating under the W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

This document is governed by the 1 February 2018 W3C Process Document.

A separate implementation report shows that each test in the test suite was passed by at least two independent implementations.

A complete list of changes to this document is available.

Table of Contents

✧ ✧ ✧

1. Introduction

This section is non-normative.

CSS beyond level 2 is a set of modules, divided up to allow the specifications to develop incrementally, along with their implementations. This specification is one of those modules.

This module describes CSS properties which allow authors to specify the foreground color and opacity of an element. This module also describes in detail the CSS <color> value type.

It not only defines the color-related properties and values that already exist in CSS1 and CSS2, but also defines new properties and values.

The Working Group doesn't expect that all implementations of CSS3 will implement all properties or values. Instead, there will probably be a small number of variants of CSS3, so-called "profiles". For example, it may be that only the profile for 32-bit color user agents will include all of the proposed color-related properties and values.

The specification is the result of the merging of relevant parts of the following Recommendations and Working Drafts, and the addition of some new features.

- HTML 4.01 [HTML401]
- CSS 2.0 [CSS2]
- SVG 1.0 [SVG10]
- User Interface for CSS3 (16 February 2000) [CSS-UI-3]

2. Dependencies

Additional terminology is defined in the Definitions section of [CSS21]. Examples of document

source code and fragments are given in XML [XML10] or HTML [HTML401] syntax.

3. Color properties

3.1. Foreground color: the 'color' property

Name:	***color***
Value:	<color> | inherit
Initial:	depends on user agent
Applies to:	all elements
Inherited:	yes
Percentages:	N/A
Media:	visual
Computed value:	• The computed value for basic color keywords, RGB hex values and extended color keywords is the equivalent triplet of numerical RGB values, e.g. six digit hex value or rgb(...) functional value , with an alpha value of 1. • The computed value of the keyword 'transparent' is the quadruplet of all zero numerical RGBA values, e.g. rgba(0,0,0,0) . • For all other values, the computed value is the specified value.

This property describes the foreground color of an element's text content. In addition it is used to provide a potential indirect value (currentColor) for any *other* properties that accept color values. If the 'currentColor' keyword is set on the 'color' property itself, it is treated as 'color: inherit'.

There are different ways to specify lime green:

```
em { color: lime }              /* color keyword */
em { color: rgb(0,255,0) }      /* RGB range 0-255   */
```

<color>
Color units are defined in a following section.

3.2. Transparency: the 'opacity' property

Opacity can be thought of as a postprocessing operation. Conceptually, after the element (including its descendants) is rendered into an RGBA offscreen image, the opacity setting specifies how to blend the offscreen rendering into the current composite rendering. See simple alpha compositing for details.

Name:	***opacity***
Value:	<alphavalue> | inherit
Initial:	1
Applies to:	all elements
Inherited:	no
Percentages:	N/A
Media:	visual
Computed value:	The same as the specified value after clipping the <alphavalue> to the range [0.0,1.0].

<alphavalue>
Syntactically a <number>. The uniform opacity setting to be applied across an entire object. Any values outside the range 0.0 (fully transparent) to 1.0 (fully opaque) will be clamped to this range. If the object has children, then the effect is as if the children were blended against the current background using a mask where the value of each pixel of the mask is <alphavalue>. For SVG, 'has children' is equivalent to being a container element [SVG11].

Since an element with opacity less than 1 is composited from a single offscreen image, content outside of it cannot be layered in z-order between pieces of content inside of it. For the same reason, implementations must create a new stacking context for any element with opacity less than 1. If an

element with opacity less than 1 is not positioned, then it is painted on the same layer, within its parent stacking context, as positioned elements with stack level 0. If an element with opacity less than 1 is positioned, the '`z-index`' property applies as described in [CSS21], except that if the used value is '`auto`' then the element behaves exactly as if it were '`0`'. See section 9.9 and Appendix E of [CSS21] for more information on stacking contexts. The rules in this paragraph do not apply to SVG elements, since SVG has its own rendering model ([SVG11], Chapter 3).

4. Color units

A ***<color>*** is either a keyword or a numerical specification.

4.1. Basic color keywords

The list of basic color keywords is: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. The color names are case-insensitive.

Color names and sRGB values

Named	Numeric	Color name	Hex rgb	Decimal
		black	#000000	0,0,0
		silver	#C0C0C0	192,192,192
		gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
		maroon	#800000	128,0,0
		red	#FF0000	255,0,0
		purple	#800080	128,0,128
		fuchsia	#FF00FF	255,0,255
		green	#008000	0,128,0
		lime	#00FF00	0,255,0
		olive	#808000	128,128,0
		yellow	#FFFF00	255,255,0
		navy	#000080	0,0,128
		blue	#0000FF	0,0,255
		teal	#008080	0,128,128
		aqua	#00FFFF	0,255,255

```
body {color: black; background: white }
h1 { color: maroon }
h2 { color: olive }
```

4.2. Numerical color values

4.2.1. RGB color values

The RGB color model is used in numerical color specifications. These examples all specify the same color:

```
em { color: #f00 }              /* #rgb */
em { color: #ff0000 }           /* #rrggbb */
em { color: rgb(255,0,0) }
em { color: rgb(100%, 0%, 0%) }
```

The format of an RGB value in hexadecimal notation is a '`#`' immediately followed by either three or six hexadecimal characters. The three-digit RGB notation (#rgb) is converted into six-digit form (#rrggbb) by replicating digits, not by adding zeros. For example, #fb0 expands to #ffbb00. This ensures that white (#ffffff) can be specified with the short notation (#fff) and removes any dependencies on the color depth of the display.

The format of an RGB value in the functional notation is '`rgb(`' followed by a comma-separated list of three numerical values (either three integer values or three percentage values) followed by '`)`'. The integer value 255 corresponds to 100%, and to F or FF in the hexadecimal notation: rgb(255,255,255) = rgb(100%,100%,100%) = #FFF. White space characters are allowed around the numerical values.

All RGB colors are specified in the sRGB color space (see [SRGB]). User agents may vary in the fidelity with which they represent these colors, but using sRGB provides an unambiguous and objectively measurable definition of what the color should be, which can be related to international standards (see [COLORIMETRY]).

Values outside the device gamut should be clipped or mapped into the gamut when the gamut is known: the red, green, and blue values must be changed to fall within the range supported by the device. User agents may perform higher quality mapping of colors from one gamut to another. This specification does not define precise clipping behavior. For a typical CRT monitor, whose device gamut is the same as sRGB, the four rules below are equivalent:

```
em { color: rgb(255,0,0) }       /* integer range 0 - 255 */
em { color: rgb(300,0,0) }       /* clipped to rgb(255,0,0) */
em { color: rgb(255,-10,0) }     /* clipped to rgb(255,0,0) */
em { color: rgb(110%, 0%, 0%) }  /* clipped to rgb(100%,0%,0%) */
```

Other devices, such as printers, have different gamuts than sRGB; some colors outside the 0..255 sRGB range will be representable (inside the device gamut), while other colors inside the 0..255 sRGB range will be outside the device gamut and will thus be mapped.

4.2.2. RGBA color values

The RGB color model is extended in this specification to include “alpha” to allow specification of the opacity of a color. See simple alpha compositing for details. These examples all specify the same color:

```
em { color: rgb(255,0,0) }      /* integer range 0 - 255 */
em { color: rgba(255,0,0,1)     /* the same, with explicit opacity of
em { color: rgb(100%,0%,0%) }   /* float range 0.0% - 100.0% */
em { color: rgba(100%,0%,0%,1) } /* the same, with explicit opacity o
```

Unlike RGB values, there is no hexadecimal notation for an RGBA value.

The format of an RGBA value in the functional notation is ‘`rgba(`’ followed by a comma-separated list of three numerical values (either three integer values or three percentage values), followed by an <alphavalue>, followed by ‘`)`’. The integer value 255 corresponds to 100%, rgba(255,255,255,0.8) = rgba(100%,100%,100%,0.8). White space characters are allowed around the numerical values.

Implementations must clip the red, green, and blue components of RGBA color values to the device gamut according to the rules for the RGB color value composed of those components.

These examples specify effects that are possible with the rgba() notation:

```
p { color: rgba(0,0,255,0.5) }        /* semi-transparent solid blue
p { color: rgba(100%, 50%, 0%, 0.1) } /* very transparent solid orang
```

> **Note.** If RGBA values are not supported by a user agent, they should be treated like unrecognized values per the CSS forward compatibility parsing rules ([CSS21], Chapter 4). RGBA values must *not* be treated as simply an RGB value with the opacity ignored.

4.2.3. '`transparent`' color keyword

CSS1 introduced the '`transparent`' value for the background-color property. CSS2 allowed border-color to also accept the '`transparent`' value. The Open eBook(tm) Publication Structure 1.0.1 [OEB101] extended the '`color`' property to also accept the '`transparent`' keyword. CSS3 extends the color value to include the '`transparent`' keyword to allow its use with all properties that accept a <color> value. This simplifies the definition of those properties in CSS3.

transparent
: Fully transparent. This keyword can be considered a shorthand for transparent black, rgba(0,0,0,0), which is its computed value.

4.2.4. HSL color values

CSS3 adds numerical hue-saturation-lightness (HSL) colors as a complement to numerical RGB colors. It has been observed that RGB colors have the following limitations:

- RGB is hardware-oriented: it reflects the use of CRTs.
- RGB is non-intuitive. People can learn how to use RGB, but actually by internalizing how to translate hue, saturation and lightness, or something similar, to RGB.

There are several other color schemes possible. Some advantages of HSL are that it is symmetrical to lightness and darkness (which is not the case with HSV for example), and it is trivial to convert HSL to RGB.

HSL colors are encoding as a triple (hue, saturation, lightness). Hue is represented as an angle of the color circle (i.e. the rainbow represented in a circle). This angle is so typically measured in degrees that the unit is implicit in CSS; syntactically, only a <number> is given. By definition red=0=360, and the other colors are spread around the circle, so green=120, blue=240, etc. As an angle, it implicitly wraps around such that -120=240 and 480=120. One way an implementation could normalize such an angle x to the range [0,360) (*i.e.* zero degrees, inclusive, to 360 degrees, exclusive) is to compute (((x mod 360) + 360) mod 360). Saturation and lightness are represented as percentages. 100% is full saturation, and 0% is a shade of gray. 0% lightness is black, 100% lightness is white, and 50% lightness is "normal".

So for instance:

```
* { color: hsl(0, 100%, 50%) }   /* red */
* { color: hsl(120, 100%, 50%) } /* lime */
* { color: hsl(120, 100%, 25%) } /* dark green */
* { color: hsl(120, 100%, 75%) } /* light green */
* { color: hsl(120, 75%, 75%) }  /* pastel green, and so on */
```

The advantage of HSL over RGB is that it is far more intuitive: you can guess at the colors you want, and then tweak. It is also easier to create sets of matching colors (by keeping the hue the same and varying the lightness/darkness, and saturation)

If saturation is less than 0%, implementations must clip it to 0%. If the resulting value is outside the device gamut, implementations must clip it to the device gamut. This clipping should preserve the hue when possible, but is otherwise undefined. (In other words, the clipping is different from applying the rules for clipping of RGB colors after applying the algorithm below for converting HSL to RGB.)

The algorithm to translate HSL to RGB is simple (here expressed in ABC [ABC] which was used to generate the tables.) In these algorithms, all three values (H, S and L) have been normalized to fractions 0..1:

```
HOW TO RETURN hsl.to.rgb(h, s, l):
   SELECT:
      l<=0.5: PUT l*(s+1) IN m2
      ELSE: PUT l+s-l*s IN m2
   PUT l*2-m2 IN m1
   PUT hue.to.rgb(m1, m2, h+1/3) IN r
   PUT hue.to.rgb(m1, m2, h    ) IN g
   PUT hue.to.rgb(m1, m2, h-1/3) IN b
   RETURN (r, g, b)

HOW TO RETURN hue.to.rgb(m1, m2, h):
   IF h<0: PUT h+1 IN h
   IF h>1: PUT h-1 IN h
   IF h*6<1: RETURN m1+(m2-m1)*h*6
   IF h*2<1: RETURN m2
   IF h*3<2: RETURN m1+(m2-m1)*(2/3-h)*6
   RETURN m1
```

4.2.4.1. HSL examples

Each table below represents one hue. Twelve equally spaced colors (i.e. at 30° intervals) have been chosen from the color circle: red, yellow, green, cyan, blue, magenta, with all the intermediate colors

(the last is the color between magenta and red).

The X axis of each table represents the saturation (100%, 75%, 50%, 25%, 0%).

The Y axis represents the lightness. 50% is “normal”.

	0° Reds				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					
88					
75					
63					
50					
38					
25					
13					
0					

	30° Red-Yellows (=Oranges)				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					
88					
75					
63					
50					
38					
25					
13					
0					

	60° Yellows				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					
88					
75					
63					
50					
38					
25					
13					
0					

	90° Yellow-Greens				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					
88					
75					
63					
50					
38					
25					
13					
0					

	120° Greens				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					

	150° Green-Cyans				
	Saturation				
	100%	**75%**	**50%**	**25%**	**0%**
100					

88
75
63
50
38
25
13
0

88
75
63
50
38
25
13
0

180° Cyans

Saturation

100% 75% 50% 25% 0%

100
88
75
63
50
38
25
13
0

210° Cyan-Blues

Saturation

100% 75% 50% 25% 0%

100
88
75
63
50
38
25
13
0

240° Blues

Saturation

100% 75% 50% 25% 0%

100
88
75
63
50
38
25
13
0

270° Blue-Magentas

Saturation

100% 75% 50% 25% 0%

100
88
75
63
50
38
25
13
0

	300° Magentas					330° Magenta-Reds				
	Saturation					Saturation				
	100%	75%	50%	25%	0%	100%	75%	50%	25%	0%
100						100				
88						88				
75						75				
63						63				
50						50				
38						38				
25						25				
13						13				
0						0				

4.2.5. HSLA color values

Just as the '`rgb()`' functional notation has the '`rgba()`' alpha counterpart, the '`hsl()`' functional notation has the '`hsla()`' alpha counterpart. See simple alpha compositing for details. These examples specify the same color:

```
em { color: hsl(120, 100%, 50%) }     /* green */
em { color: hsla(120, 100%, 50%, 1) } /* the same, with explicit opac
```

The format of an HSLA color value in the functional notation is '`hsla(`' followed by the hue in degrees, saturation and lightness as a percentage, and an <alphavalue>, followed by '`)`'. White space characters are allowed around the numerical values.

Implementations must clip the hue, saturation, and lightness components of HSLA color values to the device gamut according to the rules for the HSL color value composed of those components.

These examples specify effects that are possible with the hsla() notation:

```
p { color: hsla(240, 100%, 50%, 0.5) } /* semi-transparent solid blue
p { color: hsla(30, 100%, 50%, 0.1) }  /* very transparent solid oran
```

4.3. Extended color keywords

The table below provides a list of the X11 colors [X11COLORS] supported by popular browsers with the addition of gray/grey variants from SVG 1.0. The resulting list is precisely the same as the SVG 1.0 color keyword names. This specification extends their definition beyond SVG. The two color swatches on the left illustrate setting the background color of a table cell in two ways: The first column uses the named color value, and the second column uses the respective numeric color value.

Named	Numeric	Color name	Hex rgb	Decimal
		aliceblue	#F0F8FF	240,248,255
		antiquewhite	#FAEBD7	250,235,215
		aqua	#00FFFF	0,255,255
		aquamarine	#7FFFD4	127,255,212
		azure	#F0FFFF	240,255,255
		beige	#F5F5DC	245,245,220
		bisque	#FFE4C4	255,228,196
		black	#000000	0,0,0
		blanchedalmond	#FFEBCD	255,235,205
		blue	#0000FF	0,0,255
		blueviolet	#8A2BE2	138,43,226
		brown	#A52A2A	165,42,42
		burlywood	#DEB887	222,184,135
		cadetblue	#5F9EA0	95,158,160
		chartreuse	#7FFF00	127,255,0
		chocolate	#D2691E	210,105,30
		coral	#FF7F50	255,127,80
		cornflowerblue	#6495ED	100,149,237
		cornsilk	#FFF8DC	255,248,220
		crimson	#DC143C	220,20,60
		cyan	#00FFFF	0,255,255
		darkblue	#00008B	0,0,139
		darkcyan	#008B8B	0,139,139
		darkgoldenrod	#B8860B	184,134,11
		darkgray	#A9A9A9	169,169,169
		darkgreen	#006400	0,100,0
		darkgrey	#A9A9A9	169,169,169

		darkkhaki	#BDB76B	189,183,107
		darkmagenta	#8B008B	139,0,139
		darkolivegreen	#556B2F	85,107,47
		darkorange	#FF8C00	255,140,0
		darkorchid	#9932CC	153,50,204
		darkred	#8B0000	139,0,0
		darksalmon	#E9967A	233,150,122
		darkseagreen	#8FBC8F	143,188,143
		darkslateblue	#483D8B	72,61,139
		darkslategray	#2F4F4F	47,79,79
		darkslategrey	#2F4F4F	47,79,79
		darkturquoise	#00CED1	0,206,209
		darkviolet	#9400D3	148,0,211
		deeppink	#FF1493	255,20,147
		deepskyblue	#00BFFF	0,191,255
		dimgray	#696969	105,105,105
		dimgrey	#696969	105,105,105
		dodgerblue	#1E90FF	30,144,255
		firebrick	#B22222	178,34,34
		floralwhite	#FFFAF0	255,250,240
		forestgreen	#228B22	34,139,34
		fuchsia	#FF00FF	255,0,255
		gainsboro	#DCDCDC	220,220,220
		ghostwhite	#F8F8FF	248,248,255
		gold	#FFD700	255,215,0
		goldenrod	#DAA520	218,165,32
		gray	#808080	128,128,128
		green	#008000	0,128,0
		greenyellow	#ADFF2F	173,255,47
		grey	#808080	128,128,128
		honeydew	#F0FFF0	240,255,240
		hotpink	#FF69B4	255,105,180
		indianred	#CD5C5C	205,92,92

		indigo	#4B0082	75,0,130
		ivory	#FFFFF0	255,255,240
		khaki	#F0E68C	240,230,140
		lavender	#E6E6FA	230,230,250
		lavenderblush	#FFF0F5	255,240,245
		lawngreen	#7CFC00	124,252,0
		lemonchiffon	#FFFACD	255,250,205
		lightblue	#ADD8E6	173,216,230
		lightcoral	#F08080	240,128,128
		lightcyan	#E0FFFF	224,255,255
		lightgoldenrodyellow	#FAFAD2	250,250,210
		lightgray	#D3D3D3	211,211,211
		lightgreen	#90EE90	144,238,144
		lightgrey	#D3D3D3	211,211,211
		lightpink	#FFB6C1	255,182,193
		lightsalmon	#FFA07A	255,160,122
		lightseagreen	#20B2AA	32,178,170
		lightskyblue	#87CEFA	135,206,250
		lightslategray	#778899	119,136,153
		lightslategrey	#778899	119,136,153
		lightsteelblue	#B0C4DE	176,196,222
		lightyellow	#FFFFE0	255,255,224
		lime	#00FF00	0,255,0
		limegreen	#32CD32	50,205,50
		linen	#FAF0E6	250,240,230
		magenta	#FF00FF	255,0,255
		maroon	#800000	128,0,0
		mediumaquamarine	#66CDAA	102,205,170
		mediumblue	#0000CD	0,0,205
		mediumorchid	#BA55D3	186,85,211
		mediumpurple	#9370DB	147,112,219
		mediumseagreen	#3CB371	60,179,113
		mediumslateblue	#7B68EE	123,104,238

		mediumspringgreen	#00FA9A	0,250,154
		mediumturquoise	#48D1CC	72,209,204
		mediumvioletred	#C71585	199,21,133
		midnightblue	#191970	25,25,112
		mintcream	#F5FFFA	245,255,250
		mistyrose	#FFE4E1	255,228,225
		moccasin	#FFE4B5	255,228,181
		navajowhite	#FFDEAD	255,222,173
		navy	#000080	0,0,128
		oldlace	#FDF5E6	253,245,230
		olive	#808000	128,128,0
		olivedrab	#6B8E23	107,142,35
		orange	#FFA500	255,165,0
		orangered	#FF4500	255,69,0
		orchid	#DA70D6	218,112,214
		palegoldenrod	#EEE8AA	238,232,170
		palegreen	#98FB98	152,251,152
		paleturquoise	#AFEEEE	175,238,238
		palevioletred	#DB7093	219,112,147
		papayawhip	#FFEFD5	255,239,213
		peachpuff	#FFDAB9	255,218,185
		peru	#CD853F	205,133,63
		pink	#FFC0CB	255,192,203
		plum	#DDA0DD	221,160,221
		powderblue	#B0E0E6	176,224,230
		purple	#800080	128,0,128
		red	#FF0000	255,0,0
		rosybrown	#BC8F8F	188,143,143
		royalblue	#4169E1	65,105,225
		saddlebrown	#8B4513	139,69,19
		salmon	#FA8072	250,128,114
		sandybrown	#F4A460	244,164,96
		seagreen	#2E8B57	46,139,87

		seashell	#FFF5EE	255,245,238
		sienna	#A0522D	160,82,45
		silver	#C0C0C0	192,192,192
		skyblue	#87CEEB	135,206,235
		slateblue	#6A5ACD	106,90,205
		slategray	#708090	112,128,144
		slategrey	#708090	112,128,144
		snow	#FFFAFA	255,250,250
		springgreen	#00FF7F	0,255,127
		steelblue	#4682B4	70,130,180
		tan	#D2B48C	210,180,140
		teal	#008080	0,128,128
		thistle	#D8BFD8	216,191,216
		tomato	#FF6347	255,99,71
		turquoise	#40E0D0	64,224,208
		violet	#EE82EE	238,130,238
		wheat	#F5DEB3	245,222,179
		white	#FFFFFF	255,255,255
		whitesmoke	#F5F5F5	245,245,245
		yellow	#FFFF00	255,255,0
		yellowgreen	#9ACD32	154,205,50

4.4. ‘currentColor’ color keyword

CSS1 and CSS2 defined the initial value of the ‘border-color’ property to be “the value of the ‘color’ property” but did not define a corresponding keyword. This omission was recognized by SVG, and thus SVG 1.0 introduced the ‘currentColor’ value for the ‘fill’, ‘stroke’, ‘stop-color’, ‘flood-color’, and ‘lighting-color’ properties. CSS3 extends the color value to include the ‘currentColor’ keyword to allow its use with all properties that accept a <color> value. This simplifies the definition of those properties in CSS3.

currentColor

The value of the ‘color’ property. The used value of the ‘currentColor’ keyword is the computed value of the ‘color’ property. If the ‘currentColor’ keyword is set on the ‘color’ property itself, it is treated as ‘color: inherit’.

4.5. CSS system colors

4.5.1. CSS2 system colors

Deprecated. In addition to being able to assign color keyword values to text, backgrounds, etc., CSS2 allowed authors to specify colors in a manner that integrated them into the user's graphic environment.

For systems that do not have a corresponding value, the specified value should be mapped to the nearest system color value, or to a default color. Note that some profiles of CSS may not support System Colors at all.

The following lists additional values for color-related CSS values and their general meaning. Any color property can take one of the following names. Although these are case-insensitive, it is recommended that the mixed capitalization shown below be used, to make the names more legible.

ActiveBorder
: Active window border.

ActiveCaption
: Active window caption.

AppWorkspace
: Background color of multiple document interface.

Background
: Desktop background.

ButtonFace
: The face background color for 3-D elements that appear 3-D due to one layer of surrounding border.

ButtonHighlight
: The color of the border facing the light source for 3-D elements that appear 3-D due to one layer of surrounding border.

ButtonShadow
: The color of the border away from the light source for 3-D elements that appear 3-D due to one layer of surrounding border.

ButtonText
: Text on push buttons.

CaptionText
: Text in caption, size box, and scrollbar arrow box.

GrayText
: Grayed (disabled) text. This color is set to #000 if the current display driver does not support a solid gray color.

Highlight
Item(s) selected in a control.

HighlightText
Text of item(s) selected in a control.

InactiveBorder
Inactive window border.

InactiveCaption
Inactive window caption.

InactiveCaptionText
Color of text in an inactive caption.

InfoBackground
Background color for tooltip controls.

InfoText
Text color for tooltip controls.

Menu
Menu background.

MenuText
Text in menus.

Scrollbar
Scroll bar gray area.

ThreeDDarkShadow
The color of the darker (generally outer) of the two borders away from the light source for 3-D elements that appear 3-D due to two concentric layers of surrounding border.

ThreeDFace
The face background color for 3-D elements that appear 3-D due to two concentric layers of surrounding border.

ThreeDHighlight
The color of the lighter (generally outer) of the two borders facing the light source for 3-D elements that appear 3-D due to two concentric layers of surrounding border.

ThreeDLightShadow
The color of the darker (generally inner) of the two borders facing the light source for 3-D elements that appear 3-D due to two concentric layers of surrounding border.

ThreeDShadow
The color of the lighter (generally inner) of the two borders away from the light source for 3-D elements that appear 3-D due to two concentric layers of surrounding border.

Window
Window background.

WindowFrame
Window frame.

WindowText
Text in windows.

> For example, to set the foreground and background colors of a paragraph to the same foreground and background colors of the user's window, write the following:
>
> ```
> p { color: WindowText; background-color: Window }
> ```

4.6. Notes on using colors

Although colors can add significant amounts of information to document and make them more readable, please consider the W3C Web Content Accessibility Guidelines [WCAG20] when including color in your documents.

- *1.4.1 Use of Color:* Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element

5. Simple alpha compositing

When drawing, implementations must handle alpha according to the rules in Section 14.2 Simple alpha compositing of [SVG11]. (If the '`color-interpolation`' or '`color-rendering`' properties mentioned in that section are not implemented or do not apply, implementations must act as though they have their initial values.)

6. Sample style sheet for (X)HTML

This appendix is informative, not normative. This style sheet could be used by an implementation as part of its default styling of HTML4, XHTML1, XHTML1.1, XHTML Basic, and other XHTML Family documents.

```
html {
        color: black;
        background: white;
}
```

```
/* traditional desktop user agent colors for hyperlinks */
:link    { color: blue; }
:visited { color: purple; }

/* default focus outline */
:focus {
        outline: 1px dotted;  /* or 1px dotted invert */
}
```

7. Profiles

Each specification using CSS3 Color must define the subset of CSS3 Color features it allows and excludes, and describe the local meaning of all the components of that subset.

Non normative examples:

CSS3 Color profile	
Specification	HTML4
Accepts	Basic color keywords RGB six digit hex color values
Excludes	‘`color`’ property ‘`opacity`’ property RGB three digit hex color values and RGB functional notation color value RGBA color values HSL and HSLA color values Extended color keywords ‘`currentColor`’ color value CSS2 UI Colors ‘`transparent`’ color value
Extra constraints	none.

CSS3 Color profile	
Specification	CSS level 1

Accepts	‘color’ property Basic color keywords RGB color values
Excludes	‘opacity’ property RGBA color values HSL and HSLA color values Extended color keywords ‘currentColor’ color value CSS2 UI Colors ‘transparent’ color value
Extra constraints	none.

CSS3 Color profile	
Specification	CSS level 2
Accepts	‘color’ property Basic color keywords RGB color values CSS2 UI Colors ‘transparent’ color value
Excludes	‘opacity’ property RGBA color values HSL and HSLA color values Extended color keywords ‘currentColor’ color value
Extra constraints	‘transparent’ color value not valid for ‘color’ property. ‘orange’ color value (part of Extended color keywords) is accepted in CSS level 2 revision 1

CSS3 Color profile	
Specification	SVG 1.0 and 1.1
Accepts	‘color’ property ‘opacity’ property Basic color keywords RGB color values CSS2 UI Colors

	Extended color keywords ‘currentColor’ color value
Excludes	RGBA color values HSL and HSLA color values ‘transparent’ color value
Extra constraints	‘currentColor’ color value not valid for ‘color’ property.

8. Test suite

A CSS Color Module Test Suite has been developed, although further tests may be added. This test suite is intended to allow user agents to verify their basic conformance to the specification. This test suite does not pretend to be exhaustive and does not cover all possible numerical color values. These tests are available at https://test.csswg.org/harness/suite/css-color-3_dev/.

9. Call for Implementations of dropped features

A number of features that were present in the 14 May 2003 Candidate Recommendation are no longer present in this specification. However, the call for implementations for these features remains, and they may be included in a future level of this specification given sufficient implementations and a test suite to demonstrate interoperability. These features are:

- ICC Color Profile: the ‘color-profile’ property
- The ‘rendering-intent’ property
- The ‘@color-profile’ at-rule
- ‘flavor’ system color

10. Acknowledgments

Thanks to Brad Pettit both for writing up color-profiles, and for implementing it. Thanks to Steven Pemberton for his write up on HSL colors. Thanks especially to the feedback from Marc Attinasi, Bert Bos, Joe Clark, fantasai, Patrick Garies, Tony Graham, Ian Hickson, Susan Lesch, Alex LeDonne, Cameron McCormack, Krzysztof Maczyński, Chris Moschini, Chris Murphy, Christoph Päper, David Perrell, Jacob Refstrup, Dave Singer, Jonathan Stanley, Andrew Thompson, Russ Weakley, Etan

Wexler, David Woolley, Boris Zbarsky, Steve Zilles, the XSL FO subgroup of the XSL working group, and all the rest of the www-style community. And thanks to Chris Lilley for being the resident CSS Color expert.

11. Changes

Changes since the 15 March 2018 Proposed Recommendation

- Added a link to the current Editor's Draft
- Corrected markup around a link anchor
- Updated this changes section
- Date and boilerplate updates for W3C Recommendation

Changes since the 5 December 2017 Candidate Recommendation

- A non-substantive typo was corrected (stray semicolon)
- Removed trailing link to the no longer present `appearance` property
- Updated this changes section
- Date and boilerplate updates for Proposed Recommendation

Changes since the 2011 June 07 Recommendation

This document differs from the 2011 June 07 Recommendation in that it contains the following changes:

- Mark Introduction section as non-normative.
- Don't use SVG-specific terminology in the definition of opacity.
- Describe the effect of opacity on z-ordering more clearly and make it clear that "treated as" does not imply a change in the computed value.
- Make '`currentColor`' be a computed value and resolve at used value time, instead of being resolved in the computed value. (test, results).
- Updated references to current versions

For changes in earlier drafts, see changes in previous Recommendation and previous versions linked

therefrom.

Prior dispositions of comments:

- Disposition of comments on the 2008 Last Call.
- Disposition of comments on the 2003 Candidate Recommendation.
- Disposition of comments on the 2003 Last Call.

12. References

12.1. Normative

[COLORIMETRY]
Colorimetry, Third Edition. CIE 15:2004. ISBN 978-3-901906-33-6

[CSS21]
Bert Bos; et al. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification.* 7 June 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-CSS2-20110607

[SRGB]
Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB. IEC 61966-2-1 (1999-10) ISBN: 2-8318-4989-6 - ICS codes: 33.160.60, 37.080 - TC 100 - 51 pp. as amended by Amendment A1:2003. URL: https://webstore.iec.ch/publication/6168

[SVG11]
Erik Dahlström; et al. *Scalable Vector Graphics (SVG) 1.1 (Second Edition).* 16 August 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-SVG11-20110816/

12.2. Informative

[ABC]
Leo Geurts; Lambert Meertens; Steven Pemberton. *The ABC Programmer's Handbook.* Prentice-Hall. ISBN: 0-13-000027-2. URL: http://www.cwi.nl/~steven/abc

[CSS-UI-3]
Tantek Çelik; Florian Rivoal. *CSS Basic User Interface Module Level 3 (CSS3 UI).* 2 March 2017. W3C Candidate Recommendation. (Work in progress.) URL: https://www.w3.org/TR/2017/CR-css-ui-3-20170302/

[CSS2]
Ian Jacobs; et al. *Cascading Style Sheets, level 2 (CSS2) Specification.* 11 April 2008. W3C

Recommendation. URL: http://www.w3.org/TR/2008/REC-CSS2-20080411

[HTML401]

Dave Raggett; Arnaud Le Hors; Ian Jacobs. *HTML 4.01 Specification.* 24 December 1999. W3C Recommendation. URL: http://www.w3.org/TR/1999/REC-html401-19991224

[OEB101]

Open eBook(tm) Publication Structure 1.0.1. Open eBook Forum(tm). 02 July 2001. URL: https://www.loc.gov/preservation/digital/formats/fdd/fdd000054.shtml

[SVG10]

Jon Ferraiolo. *Scalable Vector Graphics (SVG) 1.0 Specification.* 4 September 2001. W3C Recommendation. URL: http://www.w3.org/TR/2001/REC-SVG-20010904

[WCAG20]

Ben Caldwell; et al. *Web Content Accessibility Guidelines (WCAG) 2.0.* 11 December 2008. W3C Recommendation. URL: http://www.w3.org/TR/2008/REC-WCAG20-20081211/

[X11COLORS]

Wikipedia *X11 color names.* URL: https://en.wikipedia.org/wiki/X11_color_names

[XML10]

C. M. Sperberg-McQueen; et al. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* 26 November 2008. W3C Recommendation. URL: http://www.w3.org/TR/2008/REC-xml-20081126/

Index

- darkgoldenrod, **4.3.**
- darkgray, **4.3.**
- darkgreen, **4.3.**
- darkgrey, **4.3.**
- darkkhaki, **4.3.**
- darkmagenta, **4.3.**
- darkolivegreen, **4.3.**
- darkorange, **4.3.**
- darkorchid, **4.3.**
- darkred, **4.3.**
- darksalmon, **4.3.**
- darkseagreen, **4.3.**
- darkslateblue, **4.3.**
- darkslategray, **4.3.**
- darkslategrey, **4.3.**
- darkturquoise, **4.3.**
- darkviolet, **4.3.**
- deeppink, **4.3.**
- deepskyblue, **4.3.**
- dimgray, **4.3.**
- dimgrey, **4.3.**
- dodgerblue, **4.3.**
- firebrick, **4.3.**
- floralwhite, **4.3.**
- forestgreen, **4.3.**
- fuchsia, **4.1.**, **4.3.**
- gainsboro, **4.3.**
- ghostwhite, **4.3.**
- gold, **4.3.**

- lightgray, **4.3.**
- lightgreen, **4.3.**
- lightgrey, **4.3.**
- lightpink, **4.3.**
- lightsalmon, **4.3.**
- lightseagreen, **4.3.**
- lightskyblue, **4.3.**
- lightslategray, **4.3.**
- lightslategrey, **4.3.**
- lightsteelblue, **4.3.**
- lightyellow, **4.3.**
- lime, **4.1.**, **4.3.**
- limegreen, **4.3.**
- linen, **4.3.**
- magenta, **4.3.**
- maroon, **4.1.**, **4.3.**
- mediumaquamarine, **4.3.**
- mediumblue, **4.3.**
- mediumorchid, **4.3.**
- mediumpurple, **4.3.**
- mediumseagreen, **4.3.**
- mediumslateblue, **4.3.**
- mediumspringgreen, **4.3.**
- mediumturquoise, **4.3.**
- mediumvioletred, **4.3.**
- Menu, **4.5.1.**
- MenuText, **4.5.1.**
- midnightblue, **4.3.**
- mintcream, **4.3.**

- wheat, 4.3.
- white, 4.1., 4.3.
- whitesmoke, 4.3.
- Window, 4.5.1.
- WindowFrame, 4.5.1.
- WindowText, 4.5.1.
- yellow, 4.1., 4.3.
- yellowgreen, 4.3.

Property index

Property	Values	Initial	Applies to	Inh.	Percentages	Media
color	<color> | inherit	depends on user agent		yes	N/A	visual
opacity	<alphavalue> | inherit	1		no	N/A	visual