Glossary

color value: #RGB

A shorthand form of the hexadecimal color value format **#RRGGBB**, in which each digit is duplicated. For example, `#06C` is equivalent to `#0066CC`, and `#FFF` is the same as `#FFFFFF`.

color value: #RRGGBB

The hexadecimal color format that was used in traditional HTML styling and that persists to this day. Each pair of digits is a hexadecimal number in the range `00` through `FF`, which corresponds to the range `0` through `255` in decimal notation. Examples of this notation include `#000000` (black), `#808080` (medium gray), `#900080` (medium purple), and `#FF9933` (light orange). Any such value that is made of three "twins" (that is, identical digits for each pair) can be expressed with the more compact notation **#RGB**.

color value: rgb(rrr%,ggg%,bbb%)

This color value format enables the author to express colors as percentages of the red, green, and blue channels. For example, black would be `rgb(0%,0%,0%)` and white would be `rgb(100%,100%,100%)`, while medium gray is `rgb(50%,50%,50%)` and a tan color might be `rgb(95%,92%,81%)`. Fractional values are permitted, so `rgb(57.5%,100%,77.3%)` is a valid value (although older browsers might have trouble understanding it).

color value: rgb(rrr,ggg,bbb)

This expresses colors as three numbers ranging from `0` to `255` in each of the three-color channels. Thus, white would be `rgb(255,255,255)` and medium gray would be `rgb(127,127,127)`. This is the decimal version of the color format **#RRGGBB**.

color value: named

One of a predefined list of keywords that correspond to a certain color. The mapping between a name and an RGB value can vary by browser—although there is agreement on a core set of sixteen color names. These are shown, along with some equivalent RGB values, in the following table.

Color	Number	RGB (%) Value	RGB (number) Value
Aqua	#00FFFF	rgb(0%,100%,100%)	rgb(0,255,255)
Black	#000000	rgb(0%,0%,0%)	rgb(0,0,0)
Blue	#0000FF	rgb(0%,0%,100%)	rgb(0,0,255)
Fuchsia	#FF00FF	rgb(100%,0%,100%)	rgb(255,0,255)
Gray	#808080	rgb(50%,50%,50%)	rgb(128,128,128)
Green	#008000	rgb(0%,50%,0%)	rgb(0,128,0)
Lime	#00FF00	rgb(0%,100%,0%)	rgb(0,255,0)
Maroon	#800000	rgb(50%,0%,0%)	rgb(128,0,0)
Navy	#000080	rgb(0%,0%,50%)	rgb(0,0,128)
Olive	#808000	rgb5(0%,50%,0%)	rgb(128,128,0)
Purple	#800080	rgb(50%,0%,50%)	rgb(128,0,128)
Red	#FF0000	rgb(100%,0%,0%)	rgb(255,0,0)
Silver	#C0C0C0	rgb(75%,75%,75%)	rgb(192,192,192)
Teal	#008080	rgb(0%,50%,50%)	rgb(0,128,128)
White	#FFFFFF	rgb(100%,100%,100%)	rgb(255,255,255)
Yellow	#FFFF00	rgb(100%,100%,0%)	rgb(255,255,0)

See `http://www.w3.org/TR/css3-color/` *for more details on colors and color value equivalents.*

declaration

Any property-value pair, such as `color: red;` or `position: absolute;`. The property is separated from its value by a colon, and the declaration is terminated with a semicolon. One or more declarations form a **declaration block**.

declaration block

A set of one or more declarations bounded by curly braces. For example, `{color: purple;}` and `{color: yellow; position: absolute;}` are both declaration blocks. A declaration block can contain any number of declarations but cannot contain another declaration block. A declaration block is combined with a **selector** to form a **rule**.

descendant selector

A **selector** that chooses elements based on their being descended from other elements. For example, to select any `h1` element that is a descendant of a `div` element, you would write the descendant selector `div h1` and follow it with a **declaration block** that contains the styles to be applied to such elements. Any number of descendant relations can be chained together, so selectors like `html body div ul li ol li a em` are perfectly valid.

DOCTYPE

Found at the top of an HTML document, this directive indicates the document type used in the markup of the document itself. A Document Type Definition (DTD) describes the markup that can be used in a document that uses the DTD. The `DOCTYPE` is used when validating a document and is used by some browsers to determine which **rendering mode** to use in the display of the document (see **DOCTYPE switching**).

DOCTYPE switching

The mechanism by which a browser analyzes the **DOCTYPE** found at the beginning of an HTML document and uses it to decide how to render the page. Not all browsers do this, and even those that do might not always behave in the same ways. See "Picking a Rendering Mode," which is also on this Web site, for more details.

em unit

In CSS, a unit of measure based on the value of the property `font-size`. For example, if an element's `font-size` is `15px`, then `1em` is equal to `15px` for that element alone. Another element with a `font-size` of `1.5cm` would make `1em` equivalent to `1.5cm` for that element. This unit is useful for defining padding and margins that are sized relative to the text size of the element, all without having to know the text size ahead of time. For example, `h1, h2, h3 {margin-top: 2em;}` will give all three element types a top margin twice as big as the text size of the elements themselves.

keyword

A component of a **value**. This can be whatever is permitted for a given value, including length and color units, as well as named keywords like `center` or `underline`. One or more keywords are combined to form a value. Therefore, `white url(bg.gif) top left repeat-x` is a single value composed of five keywords.

offset properties

The positioning properties `top`, `right`, `bottom`, and `left`. Each of these defines the distance between the outer margin edge of a positioned element and the corresponding edge of its containing block. Thus, `top: 50px;` defines a 50-pixel distance between a positioned element's top outer margin edge and the top of the containing block. This distance is sometimes called an "offset," and therefore these are called **offset properties**.

property

One aspect of the presentation of an element that can be altered or affected. For example, `color`, `background`, `font-weight`, `text-decoration`, and `position` are all properties. A property is associated with a **value** to form a **declaration**.

rendering mode

The way in which a page is laid out. As of this writing, browsers that support **DOCTYPE** **switching** have two rendering modes, generally referred to as "quirks" and "standard." See "Picking a Rendering Mode," which is also on this Web site, for more details.

rule

The combination of a **selector** and a **declaration block**. A very basic example is `h1 {color: green;}`, but rules can be arbitrarily long.

selector

The portion of a **rule** that determines which elements (if any) will be styled by the associated **declaration block**. Selectors can be very simple or highly complicated; a simple example is `h1` or `div#help p`. (This last one is an example of a **descendant selector**.) Multiple selectors can be grouped together in the same rule by separating them with commas, as in `h1, h2, h3`.

Uniform Resource Identifier (URI)

This describes a unique address on the Internet, although it might not correspond to a physical document or resource. One such example is the optional URI in a **DOCTYPE**, which does not necessarily point to an actual DTD on the Web. Thus URIs are a superset of **URL**s.

Uniform Resource Locator (URL)

Every time you type a Web site address into a Web browser, you're typing in a URL. The Web is not the sole province of URLs, however; other communication protocols, such as newsgroups and FTP sites, can also be described by means of a URL. A URL is always supposed to point to an actual resource that is available on the Internet; **URI**s do not have this same restriction.

value

One or more **keywords**. A value is associated with a **property**, and together they form a **declaration**. Examples of values include `13px`, `yellow`, `underline overline`, and `black url(stars.gif) center repeat fixed`. Each of these examples is a single value; the latter two are composed of multiple keywords.