SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [Color](http://docs.google.com/classsf_1_1Color.htm)

[Public Member Functions](#_gjdgxs) | [Public Attributes](#_30j0zll) | [Static Public Attributes](#_1fob9te) | [Related Functions](#_3znysh7) | [List of all members](http://docs.google.com/classsf_1_1Color-members.htm)

sf::Color Class Reference

[Graphics module](http://docs.google.com/group__graphics.htm)

Utility class for manpulating RGBA colors. [More...](http://docs.google.com/classsf_1_1Color.htm#details)

#include <[Color.hpp](http://docs.google.com/Color_8hpp_source.htm)>

| Public Member Functions | |
| --- | --- |
|  | [Color](http://docs.google.com/classsf_1_1Color.htm#ac2eb4393fb11ad3fa3ccf34e92fe08e4) () |
|  | Default constructor. |
|  | |
|  | [Color](http://docs.google.com/classsf_1_1Color.htm#ac791dc61be4c60baac50fe700f1c9850) (Uint8 red, Uint8 green, Uint8 blue, Uint8 alpha=255) |
|  | Construct the color from its 4 RGBA components. |
|  | |

| Public Attributes | |
| --- | --- |
| Uint8 | [r](http://docs.google.com/classsf_1_1Color.htm#a6a5256ca24a4f9f0e0808f6fc23e01e1) |
|  | Red component. |
|  | |
| Uint8 | [g](http://docs.google.com/classsf_1_1Color.htm#a591daf9c3c55dea830c76c962d6ba1a5) |
|  | Green component. |
|  | |
| Uint8 | [b](http://docs.google.com/classsf_1_1Color.htm#a6707aedd0609c8920e12df5d7abc53cb) |
|  | Blue component. |
|  | |
| Uint8 | [a](http://docs.google.com/classsf_1_1Color.htm#a56dbdb47d5f040d9b78ac6a0b8b3a831) |
|  | Alpha (opacity) component. |
|  | |

| Static Public Attributes | |
| --- | --- |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Black](http://docs.google.com/classsf_1_1Color.htm#a77c688197b981338f0b19dc58bd2facd) |
|  | Black predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [White](http://docs.google.com/classsf_1_1Color.htm#a4fd874712178d9e206f53226002aa4ca) |
|  | White predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Red](http://docs.google.com/classsf_1_1Color.htm#a127dbf55db9c07d0fa8f4bfcbb97594a) |
|  | Red predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Green](http://docs.google.com/classsf_1_1Color.htm#a95629b30de8c6856aa7d3afed12eb865) |
|  | Green predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Blue](http://docs.google.com/classsf_1_1Color.htm#ab03770d4817426b2614cfc33cf0e245c) |
|  | Blue predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Yellow](http://docs.google.com/classsf_1_1Color.htm#af8896b5f56650935f5b9d72d528802c7) |
|  | Yellow predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Magenta](http://docs.google.com/classsf_1_1Color.htm#a6fe70d90b65b2163dd066a84ac00426c) |
|  | Magenta predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Cyan](http://docs.google.com/classsf_1_1Color.htm#a64ae9beb0b9a5865dd811cda4bb18340) |
|  | Cyan predefined color. |
|  | |
| static const [Color](http://docs.google.com/classsf_1_1Color.htm) | [Transparent](http://docs.google.com/classsf_1_1Color.htm#a569b45471737f770656f50ae7bbac292) |
|  | Transparent (black) predefined color. |
|  | |

| Related Functions | |
| --- | --- |
| (Note that these are not member functions.) | |
| bool | [operator==](http://docs.google.com/classsf_1_1Color.htm#a2adc3f68860f7aa5e4d7c79dcbb31d30) (const [Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the == operator. |
|  | |
| bool | [operator!=](http://docs.google.com/classsf_1_1Color.htm#a394c3495753c4b17f9cd45556ef00b8c) (const [Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the != operator. |
|  | |
| [Color](http://docs.google.com/classsf_1_1Color.htm) | [operator+](http://docs.google.com/classsf_1_1Color.htm#a0355ba6bfd2f83ffd8f8fafdca26cdd0) (const [Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the binary + operator. |
|  | |
| [Color](http://docs.google.com/classsf_1_1Color.htm) | [operator\*](http://docs.google.com/classsf_1_1Color.htm#a1bae779fb49bb92dbf820a65e45a6602) (const [Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the binary \* operator. |
|  | |
| [Color](http://docs.google.com/classsf_1_1Color.htm) & | [operator+=](http://docs.google.com/classsf_1_1Color.htm#af39790b2e677c9ab418787f5ff4583ef) ([Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the binary += operator. |
|  | |
| [Color](http://docs.google.com/classsf_1_1Color.htm) & | [operator\*=](http://docs.google.com/classsf_1_1Color.htm#a7d1ea2b9bd5dbe29bb2e54feba9b4b38) ([Color](http://docs.google.com/classsf_1_1Color.htm) &left, const [Color](http://docs.google.com/classsf_1_1Color.htm) &right) |
|  | Overload of the binary \*= operator. |
|  | |

## Detailed Description

Utility class for manpulating RGBA colors.

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) is a simple color class composed of 4 components:

* Red
* Green
* Blue
* Alpha (opacity)

Each component is a public member, an unsigned integer in the range [0, 255]. Thus, colors can be constructed and manipulated very easily:

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) color(255, 0, 0); // red

color.r = 0; // make it black

color.b = 128; // make it dark blue

The fourth component of colors, named "alpha", represents the opacity of the color. A color with an alpha value of 255 will be fully opaque, while an alpha value of 0 will make a color fully transparent, whatever the value of the other components is.

The most common colors are already defined as static variables:

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) black = [sf::Color::Black](http://docs.google.com/classsf_1_1Color.htm#a77c688197b981338f0b19dc58bd2facd);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) white = [sf::Color::White](http://docs.google.com/classsf_1_1Color.htm#a4fd874712178d9e206f53226002aa4ca);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) red = [sf::Color::Red](http://docs.google.com/classsf_1_1Color.htm#a127dbf55db9c07d0fa8f4bfcbb97594a);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) green = [sf::Color::Green](http://docs.google.com/classsf_1_1Color.htm#a95629b30de8c6856aa7d3afed12eb865);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) blue = [sf::Color::Blue](http://docs.google.com/classsf_1_1Color.htm#ab03770d4817426b2614cfc33cf0e245c);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) yellow = [sf::Color::Yellow](http://docs.google.com/classsf_1_1Color.htm#af8896b5f56650935f5b9d72d528802c7);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) magenta = [sf::Color::Magenta](http://docs.google.com/classsf_1_1Color.htm#a6fe70d90b65b2163dd066a84ac00426c);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) cyan = [sf::Color::Cyan](http://docs.google.com/classsf_1_1Color.htm#a64ae9beb0b9a5865dd811cda4bb18340);

[sf::Color](http://docs.google.com/classsf_1_1Color.htm) transparent = [sf::Color::Transparent](http://docs.google.com/classsf_1_1Color.htm#a569b45471737f770656f50ae7bbac292);

Colors can also be added and modulated (multiplied) using the overloaded operators + and \*.

Definition at line [40](http://docs.google.com/Color_8hpp_source.htm#l00040) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::Color::Color | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

Constructs an opaque black color. It is equivalent to sf::Color(0, 0, 0, 255).

| sf::Color::Color | ( | Uint8 | *red*, |
| --- | --- | --- | --- |
|  |  | Uint8 | *green*, |
|  |  | Uint8 | *blue*, |
|  |  | Uint8 | *alpha* = 255 |
|  | ) |  |  |

Construct the color from its 4 RGBA components.

Parameters

| red | Red component (in the range [0, 255]) |
| --- | --- |
| green | Green component (in the range [0, 255]) |
| blue | Blue component (in the range [0, 255]) |
| alpha | Alpha (opacity) component (in the range [0, 255]) |

## Friends And Related Function Documentation

| | bool operator!= | ( | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the != operator.

This operator compares two colors and check if they are different.

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsTrue if colors are different, false if they are equal

| | [Color](http://docs.google.com/classsf_1_1Color.htm) operator\* | ( | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the binary \* operator.

This operator returns the component-wise multiplication (also called "modulation") of two colors. Components are then divided by 255 so that the result is still in the range [0, 255].

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsResult of *left* \* *right*

| | [Color](http://docs.google.com/classsf_1_1Color.htm) & operator\*= | ( | [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the binary \*= operator.

This operator returns the component-wise multiplication (also called "modulation") of two colors, and assigns the result to the left operand. Components are then divided by 255 so that the result is still in the range [0, 255].

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsReference to *left*

| | [Color](http://docs.google.com/classsf_1_1Color.htm) operator+ | ( | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the binary + operator.

This operator returns the component-wise sum of two colors. Components that exceed 255 are clamped to 255.

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsResult of *left* + *right*

| | [Color](http://docs.google.com/classsf_1_1Color.htm) & operator+= | ( | [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the binary += operator.

This operator computes the component-wise sum of two colors, and assigns the result to the left operand. Components that exceed 255 are clamped to 255.

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsReference to *left*

| | bool operator== | ( | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *left*, | | --- | --- | --- | --- | |  |  | const [Color](http://docs.google.com/classsf_1_1Color.htm) & | *right* | |  | ) |  |  | | related |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Overload of the == operator.

This operator compares two colors and check if they are equal.

Parameters

| left | Left operand |
| --- | --- |
| right | Right operand |

ReturnsTrue if colors are equal, false if they are different

## Member Data Documentation

| Uint8 sf::Color::a |
| --- |

Alpha (opacity) component.

Definition at line [83](http://docs.google.com/Color_8hpp_source.htm#l00083) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| Uint8 sf::Color::b |
| --- |

Blue component.

Definition at line [82](http://docs.google.com/Color_8hpp_source.htm#l00082) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Black | | --- | | static |
| --- | --- | --- |

Black predefined color.

Definition at line [67](http://docs.google.com/Color_8hpp_source.htm#l00067) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Blue | | --- | | static |
| --- | --- | --- |

Blue predefined color.

Definition at line [71](http://docs.google.com/Color_8hpp_source.htm#l00071) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Cyan | | --- | | static |
| --- | --- | --- |

Cyan predefined color.

Definition at line [74](http://docs.google.com/Color_8hpp_source.htm#l00074) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| Uint8 sf::Color::g |
| --- |

Green component.

Definition at line [81](http://docs.google.com/Color_8hpp_source.htm#l00081) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Green | | --- | | static |
| --- | --- | --- |

Green predefined color.

Definition at line [70](http://docs.google.com/Color_8hpp_source.htm#l00070) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Magenta | | --- | | static |
| --- | --- | --- |

Magenta predefined color.

Definition at line [73](http://docs.google.com/Color_8hpp_source.htm#l00073) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| Uint8 sf::Color::r |
| --- |

Red component.

Definition at line [80](http://docs.google.com/Color_8hpp_source.htm#l00080) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Red | | --- | | static |
| --- | --- | --- |

Red predefined color.

Definition at line [69](http://docs.google.com/Color_8hpp_source.htm#l00069) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Transparent | | --- | | static |
| --- | --- | --- |

Transparent (black) predefined color.

Definition at line [75](http://docs.google.com/Color_8hpp_source.htm#l00075) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::White | | --- | | static |
| --- | --- | --- |

White predefined color.

Definition at line [68](http://docs.google.com/Color_8hpp_source.htm#l00068) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

| | const [Color](http://docs.google.com/classsf_1_1Color.htm) sf::Color::Yellow | | --- | | static |
| --- | --- | --- |

Yellow predefined color.

Definition at line [72](http://docs.google.com/Color_8hpp_source.htm#l00072) of file [Color.hpp](http://docs.google.com/Color_8hpp_source.htm).

The documentation for this class was generated from the following file:

* [Color.hpp](http://docs.google.com/Color_8hpp_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::