| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Color.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Choice.AccessibleAWTChoice.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Component.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Color.html)    [**NO FRAMES**](http://docs.google.com/Color.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#32hioqz) | [METHOD](#2u6wntf) |

## **java.awt**

Class Color

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.Color**

**All Implemented Interfaces:** [Paint](http://docs.google.com/java/awt/Paint.html), [Transparency](http://docs.google.com/java/awt/Transparency.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [ColorUIResource](http://docs.google.com/javax/swing/plaf/ColorUIResource.html), [SystemColor](http://docs.google.com/java/awt/SystemColor.html)

public class **Color**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Paint](http://docs.google.com/java/awt/Paint.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

The Color class is used to encapsulate colors in the default sRGB color space or colors in arbitrary color spaces identified by a [ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html). Every color has an implicit alpha value of 1.0 or an explicit one provided in the constructor. The alpha value defines the transparency of a color and can be represented by a float value in the range 0.0 - 1.0 or 0 - 255. An alpha value of 1.0 or 255 means that the color is completely opaque and an alpha value of 0 or 0.0 means that the color is completely transparent. When constructing a Color with an explicit alpha or getting the color/alpha components of a Color, the color components are never premultiplied by the alpha component.

The default color space for the Java 2D(tm) API is sRGB, a proposed standard RGB color space. For further information on sRGB, see <http://www.w3.org/pub/WWW/Graphics/Color/sRGB.html>.

**See Also:**[ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html), [AlphaComposite](http://docs.google.com/java/awt/AlphaComposite.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Color)

| **Field Summary** | |
| --- | --- |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**black**](http://docs.google.com/java/awt/Color.html#black)            The color black. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**BLACK**](http://docs.google.com/java/awt/Color.html#BLACK)            The color black. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**blue**](http://docs.google.com/java/awt/Color.html#blue)            The color blue. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**BLUE**](http://docs.google.com/java/awt/Color.html#BLUE)            The color blue. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**cyan**](http://docs.google.com/java/awt/Color.html#cyan)            The color cyan. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**CYAN**](http://docs.google.com/java/awt/Color.html#CYAN)            The color cyan. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**DARK\_GRAY**](http://docs.google.com/java/awt/Color.html#DARK_GRAY)            The color dark gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**darkGray**](http://docs.google.com/java/awt/Color.html#darkGray)            The color dark gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**gray**](http://docs.google.com/java/awt/Color.html#gray)            The color gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**GRAY**](http://docs.google.com/java/awt/Color.html#GRAY)            The color gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**green**](http://docs.google.com/java/awt/Color.html#green)            The color green. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**GREEN**](http://docs.google.com/java/awt/Color.html#GREEN)            The color green. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**LIGHT\_GRAY**](http://docs.google.com/java/awt/Color.html#LIGHT_GRAY)            The color light gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**lightGray**](http://docs.google.com/java/awt/Color.html#lightGray)            The color light gray. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**magenta**](http://docs.google.com/java/awt/Color.html#magenta)            The color magenta. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**MAGENTA**](http://docs.google.com/java/awt/Color.html#MAGENTA)            The color magenta. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**orange**](http://docs.google.com/java/awt/Color.html#orange)            The color orange. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**ORANGE**](http://docs.google.com/java/awt/Color.html#ORANGE)            The color orange. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**pink**](http://docs.google.com/java/awt/Color.html#pink)            The color pink. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**PINK**](http://docs.google.com/java/awt/Color.html#PINK)            The color pink. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**red**](http://docs.google.com/java/awt/Color.html#red)            The color red. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**RED**](http://docs.google.com/java/awt/Color.html#RED)            The color red. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**white**](http://docs.google.com/java/awt/Color.html#white)            The color white. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**WHITE**](http://docs.google.com/java/awt/Color.html#WHITE)            The color white. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**yellow**](http://docs.google.com/java/awt/Color.html#yellow)            The color yellow. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**YELLOW**](http://docs.google.com/java/awt/Color.html#YELLOW)            The color yellow. |

| **Fields inherited from interface java.awt.**[**Transparency**](http://docs.google.com/java/awt/Transparency.html) |
| --- |
| [BITMASK](http://docs.google.com/java/awt/Transparency.html#BITMASK), [OPAQUE](http://docs.google.com/java/awt/Transparency.html#OPAQUE), [TRANSLUCENT](http://docs.google.com/java/awt/Transparency.html#TRANSLUCENT) |

| **Constructor Summary** | |
| --- | --- |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(java.awt.color.ColorSpace,%20float%5B%5D,%20float))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace, float[] components, float alpha)            Creates a color in the specified ColorSpace with the color components specified in the float array and the specified alpha. |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(float,%20float,%20float))(float r, float g, float b)            Creates an opaque sRGB color with the specified red, green, and blue values in the range (0.0 - 1.0). |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(float,%20float,%20float,%20float))(float r, float g, float b, float a)            Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0.0 - 1.0). |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(int))(int rgb)            Creates an opaque sRGB color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7. |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(int,%20boolean))(int rgba, boolean hasalpha)            Creates an sRGB color with the specified combined RGBA value consisting of the alpha component in bits 24-31, the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7. |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(int,%20int,%20int))(int r, int g, int b)            Creates an opaque sRGB color with the specified red, green, and blue values in the range (0 - 255). |
| [**Color**](http://docs.google.com/java/awt/Color.html#Color(int,%20int,%20int,%20int))(int r, int g, int b, int a)            Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0 - 255). |

| **Method Summary** | |
| --- | --- |
| [Color](http://docs.google.com/java/awt/Color.html) | [**brighter**](http://docs.google.com/java/awt/Color.html#brighter())()            Creates a new Color that is a brighter version of this Color. |
| [PaintContext](http://docs.google.com/java/awt/PaintContext.html) | [**createContext**](http://docs.google.com/java/awt/Color.html#createContext(java.awt.image.ColorModel,%20java.awt.Rectangle,%20java.awt.geom.Rectangle2D,%20java.awt.geom.AffineTransform,%20java.awt.RenderingHints))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r, [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r2d, [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) xform, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Creates and returns a [PaintContext](http://docs.google.com/java/awt/PaintContext.html) used to generate a solid color pattern. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**darker**](http://docs.google.com/java/awt/Color.html#darker())()            Creates a new Color that is a darker version of this Color. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**decode**](http://docs.google.com/java/awt/Color.html#decode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Converts a String to an integer and returns the specified opaque Color. |
| boolean | [**equals**](http://docs.google.com/java/awt/Color.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Determines whether another object is equal to this Color. |
| int | [**getAlpha**](http://docs.google.com/java/awt/Color.html#getAlpha())()            Returns the alpha component in the range 0-255. |
| int | [**getBlue**](http://docs.google.com/java/awt/Color.html#getBlue())()            Returns the blue component in the range 0-255 in the default sRGB space. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**getColor**](http://docs.google.com/java/awt/Color.html#getColor(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Finds a color in the system properties. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**getColor**](http://docs.google.com/java/awt/Color.html#getColor(java.lang.String,%20java.awt.Color))([String](http://docs.google.com/java/lang/String.html) nm, [Color](http://docs.google.com/java/awt/Color.html) v)            Finds a color in the system properties. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**getColor**](http://docs.google.com/java/awt/Color.html#getColor(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) nm, int v)            Finds a color in the system properties. |
| float[] | [**getColorComponents**](http://docs.google.com/java/awt/Color.html#getColorComponents(java.awt.color.ColorSpace,%20float%5B%5D))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace, float[] compArray)            Returns a float array containing only the color components of the Color in the ColorSpace specified by the cspace parameter. |
| float[] | [**getColorComponents**](http://docs.google.com/java/awt/Color.html#getColorComponents(float%5B%5D))(float[] compArray)            Returns a float array containing only the color components of the Color, in the ColorSpace of the Color. |
| [ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) | [**getColorSpace**](http://docs.google.com/java/awt/Color.html#getColorSpace())()            Returns the ColorSpace of this Color. |
| float[] | [**getComponents**](http://docs.google.com/java/awt/Color.html#getComponents(java.awt.color.ColorSpace,%20float%5B%5D))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace, float[] compArray)            Returns a float array containing the color and alpha components of the Color, in the ColorSpace specified by the cspace parameter. |
| float[] | [**getComponents**](http://docs.google.com/java/awt/Color.html#getComponents(float%5B%5D))(float[] compArray)            Returns a float array containing the color and alpha components of the Color, in the ColorSpace of the Color. |
| int | [**getGreen**](http://docs.google.com/java/awt/Color.html#getGreen())()            Returns the green component in the range 0-255 in the default sRGB space. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**getHSBColor**](http://docs.google.com/java/awt/Color.html#getHSBColor(float,%20float,%20float))(float h, float s, float b)            Creates a Color object based on the specified values for the HSB color model. |
| int | [**getRed**](http://docs.google.com/java/awt/Color.html#getRed())()            Returns the red component in the range 0-255 in the default sRGB space. |
| int | [**getRGB**](http://docs.google.com/java/awt/Color.html#getRGB())()            Returns the RGB value representing the color in the default sRGB [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html). |
| float[] | [**getRGBColorComponents**](http://docs.google.com/java/awt/Color.html#getRGBColorComponents(float%5B%5D))(float[] compArray)            Returns a float array containing only the color components of the Color, in the default sRGB color space. |
| float[] | [**getRGBComponents**](http://docs.google.com/java/awt/Color.html#getRGBComponents(float%5B%5D))(float[] compArray)            Returns a float array containing the color and alpha components of the Color, as represented in the default sRGB color space. |
| int | [**getTransparency**](http://docs.google.com/java/awt/Color.html#getTransparency())()            Returns the transparency mode for this Color. |
| int | [**hashCode**](http://docs.google.com/java/awt/Color.html#hashCode())()            Computes the hash code for this Color. |
| static int | [**HSBtoRGB**](http://docs.google.com/java/awt/Color.html#HSBtoRGB(float,%20float,%20float))(float hue, float saturation, float brightness)            Converts the components of a color, as specified by the HSB model, to an equivalent set of values for the default RGB model. |
| static float[] | [**RGBtoHSB**](http://docs.google.com/java/awt/Color.html#RGBtoHSB(int,%20int,%20int,%20float%5B%5D))(int r, int g, int b, float[] hsbvals)            Converts the components of a color, as specified by the default RGB model, to an equivalent set of values for hue, saturation, and brightness that are the three components of the HSB model. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/Color.html#toString())()            Returns a string representation of this Color. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### white

public static final [Color](http://docs.google.com/java/awt/Color.html) **white**

The color white. In the default sRGB space.

### WHITE

public static final [Color](http://docs.google.com/java/awt/Color.html) **WHITE**

The color white. In the default sRGB space.

**Since:** 1.4

### lightGray

public static final [Color](http://docs.google.com/java/awt/Color.html) **lightGray**

The color light gray. In the default sRGB space.

### LIGHT\_GRAY

public static final [Color](http://docs.google.com/java/awt/Color.html) **LIGHT\_GRAY**

The color light gray. In the default sRGB space.

**Since:** 1.4

### gray

public static final [Color](http://docs.google.com/java/awt/Color.html) **gray**

The color gray. In the default sRGB space.

### GRAY

public static final [Color](http://docs.google.com/java/awt/Color.html) **GRAY**

The color gray. In the default sRGB space.

**Since:** 1.4

### darkGray

public static final [Color](http://docs.google.com/java/awt/Color.html) **darkGray**

The color dark gray. In the default sRGB space.

### DARK\_GRAY

public static final [Color](http://docs.google.com/java/awt/Color.html) **DARK\_GRAY**

The color dark gray. In the default sRGB space.

**Since:** 1.4

### black

public static final [Color](http://docs.google.com/java/awt/Color.html) **black**

The color black. In the default sRGB space.

### BLACK

public static final [Color](http://docs.google.com/java/awt/Color.html) **BLACK**

The color black. In the default sRGB space.

**Since:** 1.4

### red

public static final [Color](http://docs.google.com/java/awt/Color.html) **red**

The color red. In the default sRGB space.

### RED

public static final [Color](http://docs.google.com/java/awt/Color.html) **RED**

The color red. In the default sRGB space.

**Since:** 1.4

### pink

public static final [Color](http://docs.google.com/java/awt/Color.html) **pink**

The color pink. In the default sRGB space.

### PINK

public static final [Color](http://docs.google.com/java/awt/Color.html) **PINK**

The color pink. In the default sRGB space.

**Since:** 1.4

### orange

public static final [Color](http://docs.google.com/java/awt/Color.html) **orange**

The color orange. In the default sRGB space.

### ORANGE

public static final [Color](http://docs.google.com/java/awt/Color.html) **ORANGE**

The color orange. In the default sRGB space.

**Since:** 1.4

### yellow

public static final [Color](http://docs.google.com/java/awt/Color.html) **yellow**

The color yellow. In the default sRGB space.

### YELLOW

public static final [Color](http://docs.google.com/java/awt/Color.html) **YELLOW**

The color yellow. In the default sRGB space.

**Since:** 1.4

### green

public static final [Color](http://docs.google.com/java/awt/Color.html) **green**

The color green. In the default sRGB space.

### GREEN

public static final [Color](http://docs.google.com/java/awt/Color.html) **GREEN**

The color green. In the default sRGB space.

**Since:** 1.4

### magenta

public static final [Color](http://docs.google.com/java/awt/Color.html) **magenta**

The color magenta. In the default sRGB space.

### MAGENTA

public static final [Color](http://docs.google.com/java/awt/Color.html) **MAGENTA**

The color magenta. In the default sRGB space.

**Since:** 1.4

### cyan

public static final [Color](http://docs.google.com/java/awt/Color.html) **cyan**

The color cyan. In the default sRGB space.

### CYAN

public static final [Color](http://docs.google.com/java/awt/Color.html) **CYAN**

The color cyan. In the default sRGB space.

**Since:** 1.4

### blue

public static final [Color](http://docs.google.com/java/awt/Color.html) **blue**

The color blue. In the default sRGB space.

### BLUE

public static final [Color](http://docs.google.com/java/awt/Color.html) **BLUE**

The color blue. In the default sRGB space.

**Since:** 1.4

| **Constructor Detail** |
| --- |

### Color

public **Color**(int r,  
 int g,  
 int b)

Creates an opaque sRGB color with the specified red, green, and blue values in the range (0 - 255). The actual color used in rendering depends on finding the best match given the color space available for a given output device. Alpha is defaulted to 255.

**Parameters:**r - the red componentg - the green componentb - the blue component **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if r, g or b are outside of the range 0 to 255, inclusive**See Also:**[getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**(int r,  
 int g,  
 int b,  
 int a)

Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0 - 255).

**Parameters:**r - the red componentg - the green componentb - the blue componenta - the alpha component **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if r, g, b or a are outside of the range 0 to 255, inclusive**See Also:**[getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getAlpha()](http://docs.google.com/java/awt/Color.html#getAlpha()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**(int rgb)

Creates an opaque sRGB color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7. The actual color used in rendering depends on finding the best match given the color space available for a particular output device. Alpha is defaulted to 255.

**Parameters:**rgb - the combined RGB components**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**(int rgba,  
 boolean hasalpha)

Creates an sRGB color with the specified combined RGBA value consisting of the alpha component in bits 24-31, the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7. If the hasalpha argument is false, alpha is defaulted to 255.

**Parameters:**rgba - the combined RGBA componentshasalpha - true if the alpha bits are valid; false otherwise**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getAlpha()](http://docs.google.com/java/awt/Color.html#getAlpha()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**(float r,  
 float g,  
 float b)

Creates an opaque sRGB color with the specified red, green, and blue values in the range (0.0 - 1.0). Alpha is defaulted to 1.0. The actual color used in rendering depends on finding the best match given the color space available for a particular output device.

**Parameters:**r - the red componentg - the green componentb - the blue component **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if r, g or b are outside of the range 0.0 to 1.0, inclusive**See Also:**[getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**(float r,  
 float g,  
 float b,  
 float a)

Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0.0 - 1.0). The actual color used in rendering depends on finding the best match given the color space available for a particular output device.

**Parameters:**r - the red componentg - the green componentb - the blue componenta - the alpha component **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if r, g b or a are outside of the range 0.0 to 1.0, inclusive**See Also:**[getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue()), [getAlpha()](http://docs.google.com/java/awt/Color.html#getAlpha()), [getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### Color

public **Color**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace,  
 float[] components,  
 float alpha)

Creates a color in the specified ColorSpace with the color components specified in the float array and the specified alpha. The number of components is determined by the type of the ColorSpace. For example, RGB requires 3 components, but CMYK requires 4 components.

**Parameters:**cspace - the ColorSpace to be used to interpret the componentscomponents - an arbitrary number of color components that is compatible with the ColorSpacealpha - alpha value **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the values in the components array or alpha is outside of the range 0.0 to 1.0**See Also:**[getComponents(float[])](http://docs.google.com/java/awt/Color.html#getComponents(float%5B%5D)), [getColorComponents(float[])](http://docs.google.com/java/awt/Color.html#getColorComponents(float%5B%5D))

| **Method Detail** |
| --- |

### getRed

public int **getRed**()

Returns the red component in the range 0-255 in the default sRGB space.

**Returns:**the red component.**See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### getGreen

public int **getGreen**()

Returns the green component in the range 0-255 in the default sRGB space.

**Returns:**the green component.**See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### getBlue

public int **getBlue**()

Returns the blue component in the range 0-255 in the default sRGB space.

**Returns:**the blue component.**See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### getAlpha

public int **getAlpha**()

Returns the alpha component in the range 0-255.

**Returns:**the alpha component.**See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB())

### getRGB

public int **getRGB**()

Returns the RGB value representing the color in the default sRGB [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html). (Bits 24-31 are alpha, 16-23 are red, 8-15 are green, 0-7 are blue).

**Returns:**the RGB value of the color in the default sRGB ColorModel.**Since:** JDK1.0 **See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [getRed()](http://docs.google.com/java/awt/Color.html#getRed()), [getGreen()](http://docs.google.com/java/awt/Color.html#getGreen()), [getBlue()](http://docs.google.com/java/awt/Color.html#getBlue())

### brighter

public [Color](http://docs.google.com/java/awt/Color.html) **brighter**()

Creates a new Color that is a brighter version of this Color.

This method applies an arbitrary scale factor to each of the three RGB components of this Color to create a brighter version of this Color. Although brighter and darker are inverse operations, the results of a series of invocations of these two methods might be inconsistent because of rounding errors.

**Returns:**a new Color object that is a brighter version of this Color.**Since:** JDK1.0 **See Also:**[darker()](http://docs.google.com/java/awt/Color.html#darker())

### darker

public [Color](http://docs.google.com/java/awt/Color.html) **darker**()

Creates a new Color that is a darker version of this Color.

This method applies an arbitrary scale factor to each of the three RGB components of this Color to create a darker version of this Color. Although brighter and darker are inverse operations, the results of a series of invocations of these two methods might be inconsistent because of rounding errors.

**Returns:**a new Color object that is a darker version of this Color.**Since:** JDK1.0 **See Also:**[brighter()](http://docs.google.com/java/awt/Color.html#brighter())

### hashCode

public int **hashCode**()

Computes the hash code for this Color.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**Since:** JDK1.0 **See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Determines whether another object is equal to this Color.

The result is true if and only if the argument is not null and is a Color object that has the same red, green, blue, and alpha values as this object.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object to test for equality with this Color **Returns:**true if the objects are the same; false otherwise.**Since:** JDK1.0 **See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string representation of this Color. This method is intended to be used only for debugging purposes. The content and format of the returned string might vary between implementations. The returned string might be empty but cannot be null.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this Color.

### decode

public static [Color](http://docs.google.com/java/awt/Color.html) **decode**([String](http://docs.google.com/java/lang/String.html) nm)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Converts a String to an integer and returns the specified opaque Color. This method handles string formats that are used to represent octal and hexidecimal numbers.

**Parameters:**nm - a String that represents an opaque color as a 24-bit integer **Returns:**the new Color object. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - if the specified string cannot be interpreted as a decimal, octal, or hexidecimal integer.**Since:** JDK1.1 **See Also:**[Integer.decode(java.lang.String)](http://docs.google.com/java/lang/Integer.html#decode(java.lang.String))

### getColor

public static [Color](http://docs.google.com/java/awt/Color.html) **getColor**([String](http://docs.google.com/java/lang/String.html) nm)

Finds a color in the system properties.

The argument is treated as the name of a system property to be obtained. The string value of this property is then interpreted as an integer which is then converted to a Color object.

If the specified property is not found or could not be parsed as an integer then null is returned.

**Parameters:**nm - the name of the color property **Returns:**the Color converted from the system property.**Since:** JDK1.0 **See Also:**[System.getProperty(java.lang.String)](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)), [Integer.getInteger(java.lang.String)](http://docs.google.com/java/lang/Integer.html#getInteger(java.lang.String)), [Color(int)](http://docs.google.com/java/awt/Color.html#Color(int))

### getColor

public static [Color](http://docs.google.com/java/awt/Color.html) **getColor**([String](http://docs.google.com/java/lang/String.html) nm,  
 [Color](http://docs.google.com/java/awt/Color.html) v)

Finds a color in the system properties.

The first argument is treated as the name of a system property to be obtained. The string value of this property is then interpreted as an integer which is then converted to a Color object.

If the specified property is not found or cannot be parsed as an integer then the Color specified by the second argument is returned instead.

**Parameters:**nm - the name of the color propertyv - the default Color **Returns:**the Color converted from the system property, or the specified Color.**Since:** JDK1.0 **See Also:**[System.getProperty(java.lang.String)](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)), [Integer.getInteger(java.lang.String)](http://docs.google.com/java/lang/Integer.html#getInteger(java.lang.String)), [Color(int)](http://docs.google.com/java/awt/Color.html#Color(int))

### getColor

public static [Color](http://docs.google.com/java/awt/Color.html) **getColor**([String](http://docs.google.com/java/lang/String.html) nm,  
 int v)

Finds a color in the system properties.

The first argument is treated as the name of a system property to be obtained. The string value of this property is then interpreted as an integer which is then converted to a Color object.

If the specified property is not found or could not be parsed as an integer then the integer value v is used instead, and is converted to a Color object.

**Parameters:**nm - the name of the color propertyv - the default color value, as an integer **Returns:**the Color converted from the system property or the Color converted from the specified integer.**Since:** JDK1.0 **See Also:**[System.getProperty(java.lang.String)](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)), [Integer.getInteger(java.lang.String)](http://docs.google.com/java/lang/Integer.html#getInteger(java.lang.String)), [Color(int)](http://docs.google.com/java/awt/Color.html#Color(int))

### HSBtoRGB

public static int **HSBtoRGB**(float hue,  
 float saturation,  
 float brightness)

Converts the components of a color, as specified by the HSB model, to an equivalent set of values for the default RGB model.

The saturation and brightness components should be floating-point values between zero and one (numbers in the range 0.0-1.0). The hue component can be any floating-point number. The floor of this number is subtracted from it to create a fraction between 0 and 1. This fractional number is then multiplied by 360 to produce the hue angle in the HSB color model.

The integer that is returned by HSBtoRGB encodes the value of a color in bits 0-23 of an integer value that is the same format used by the method [getRGB](http://docs.google.com/java/awt/Color.html#getRGB()). This integer can be supplied as an argument to the Color constructor that takes a single integer argument.

**Parameters:**hue - the hue component of the colorsaturation - the saturation of the colorbrightness - the brightness of the color **Returns:**the RGB value of the color with the indicated hue, saturation, and brightness.**Since:** JDK1.0 **See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB()), [Color(int)](http://docs.google.com/java/awt/Color.html#Color(int)), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

### RGBtoHSB

public static float[] **RGBtoHSB**(int r,  
 int g,  
 int b,  
 float[] hsbvals)

Converts the components of a color, as specified by the default RGB model, to an equivalent set of values for hue, saturation, and brightness that are the three components of the HSB model.

If the hsbvals argument is null, then a new array is allocated to return the result. Otherwise, the method returns the array hsbvals, with the values put into that array.

**Parameters:**r - the red component of the colorg - the green component of the colorb - the blue component of the colorhsbvals - the array used to return the three HSB values, or null **Returns:**an array of three elements containing the hue, saturation, and brightness (in that order), of the color with the indicated red, green, and blue components.**Since:** JDK1.0 **See Also:**[getRGB()](http://docs.google.com/java/awt/Color.html#getRGB()), [Color(int)](http://docs.google.com/java/awt/Color.html#Color(int)), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

### getHSBColor

public static [Color](http://docs.google.com/java/awt/Color.html) **getHSBColor**(float h,  
 float s,  
 float b)

Creates a Color object based on the specified values for the HSB color model.

The s and b components should be floating-point values between zero and one (numbers in the range 0.0-1.0). The h component can be any floating-point number. The floor of this number is subtracted from it to create a fraction between 0 and 1. This fractional number is then multiplied by 360 to produce the hue angle in the HSB color model.

**Parameters:**h - the hue components - the saturation of the colorb - the brightness of the color **Returns:**a Color object with the specified hue, saturation, and brightness.**Since:** JDK1.0

### getRGBComponents

public float[] **getRGBComponents**(float[] compArray)

Returns a float array containing the color and alpha components of the Color, as represented in the default sRGB color space. If compArray is null, an array of length 4 is created for the return value. Otherwise, compArray must have length 4 or greater, and it is filled in with the components and returned.

**Parameters:**compArray - an array that this method fills with color and alpha components and returns **Returns:**the RGBA components in a float array.

### getRGBColorComponents

public float[] **getRGBColorComponents**(float[] compArray)

Returns a float array containing only the color components of the Color, in the default sRGB color space. If compArray is null, an array of length 3 is created for the return value. Otherwise, compArray must have length 3 or greater, and it is filled in with the components and returned.

**Parameters:**compArray - an array that this method fills with color components and returns **Returns:**the RGB components in a float array.

### getComponents

public float[] **getComponents**(float[] compArray)

Returns a float array containing the color and alpha components of the Color, in the ColorSpace of the Color. If compArray is null, an array with length equal to the number of components in the associated ColorSpace plus one is created for the return value. Otherwise, compArray must have at least this length and it is filled in with the components and returned.

**Parameters:**compArray - an array that this method fills with the color and alpha components of this Color in its ColorSpace and returns **Returns:**the color and alpha components in a float array.

### getColorComponents

public float[] **getColorComponents**(float[] compArray)

Returns a float array containing only the color components of the Color, in the ColorSpace of the Color. If compArray is null, an array with length equal to the number of components in the associated ColorSpace is created for the return value. Otherwise, compArray must have at least this length and it is filled in with the components and returned.

**Parameters:**compArray - an array that this method fills with the color components of this Color in its ColorSpace and returns **Returns:**the color components in a float array.

### getComponents

public float[] **getComponents**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace,  
 float[] compArray)

Returns a float array containing the color and alpha components of the Color, in the ColorSpace specified by the cspace parameter. If compArray is null, an array with length equal to the number of components in cspace plus one is created for the return value. Otherwise, compArray must have at least this length, and it is filled in with the components and returned.

**Parameters:**cspace - a specified ColorSpacecompArray - an array that this method fills with the color and alpha components of this Color in the specified ColorSpace and returns **Returns:**the color and alpha components in a float array.

### getColorComponents

public float[] **getColorComponents**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) cspace,  
 float[] compArray)

Returns a float array containing only the color components of the Color in the ColorSpace specified by the cspace parameter. If compArray is null, an array with length equal to the number of components in cspace is created for the return value. Otherwise, compArray must have at least this length, and it is filled in with the components and returned.

**Parameters:**cspace - a specified ColorSpacecompArray - an array that this method fills with the color components of this Color in the specified ColorSpace **Returns:**the color components in a float array.

### getColorSpace

public [ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) **getColorSpace**()

Returns the ColorSpace of this Color.

**Returns:**this Color object's ColorSpace.

### createContext

public [PaintContext](http://docs.google.com/java/awt/PaintContext.html) **createContext**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r,  
 [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r2d,  
 [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) xform,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Creates and returns a [PaintContext](http://docs.google.com/java/awt/PaintContext.html) used to generate a solid color pattern. This enables a Color object to be used as an argument to any method requiring an object implementing the [Paint](http://docs.google.com/java/awt/Paint.html) interface. The same PaintContext is returned, regardless of whether or not r, r2d, xform, or hints are null.

**Specified by:**[createContext](http://docs.google.com/java/awt/Paint.html#createContext(java.awt.image.ColorModel,%20java.awt.Rectangle,%20java.awt.geom.Rectangle2D,%20java.awt.geom.AffineTransform,%20java.awt.RenderingHints)) in interface [Paint](http://docs.google.com/java/awt/Paint.html) **Parameters:**cm - the specified ColorModelr - the specified [Rectangle](http://docs.google.com/java/awt/Rectangle.html)r2d - the specified [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html)xform - the specified [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html)hints - the specified [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) **Returns:**a PaintContext that is used to generate a solid color pattern.**See Also:**[Paint](http://docs.google.com/java/awt/Paint.html), [PaintContext](http://docs.google.com/java/awt/PaintContext.html), [Graphics2D.setPaint(java.awt.Paint)](http://docs.google.com/java/awt/Graphics2D.html#setPaint(java.awt.Paint))

### getTransparency

public int **getTransparency**()

Returns the transparency mode for this Color. This is required to implement the Paint interface.

**Specified by:**[getTransparency](http://docs.google.com/java/awt/Transparency.html#getTransparency()) in interface [Transparency](http://docs.google.com/java/awt/Transparency.html) **Returns:**this Color object's transparency mode.**See Also:**[Paint](http://docs.google.com/java/awt/Paint.html), [Transparency](http://docs.google.com/java/awt/Transparency.html), [createContext(java.awt.image.ColorModel, java.awt.Rectangle, java.awt.geom.Rectangle2D, java.awt.geom.AffineTransform, java.awt.RenderingHints)](http://docs.google.com/java/awt/Color.html#createContext(java.awt.image.ColorModel,%20java.awt.Rectangle,%20java.awt.geom.Rectangle2D,%20java.awt.geom.AffineTransform,%20java.awt.RenderingHints))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Color.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Choice.AccessibleAWTChoice.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Component.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Color.html)    [**NO FRAMES**](http://docs.google.com/Color.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#32hioqz) | [METHOD](#2u6wntf) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).