Class java.awt.Color

```
java.lang.Object
   |
   +----java.awt.Color
```

---

public final class **Color**
extends Object

A class to encapsulate RGB Colors.

Version:
1.12, 09/08/95
Author:
Sami Shaio, Arthur van Hoff

Variable Index

o **black**
The color black.
o **blue**
The color blue.
o **cyan**
The color cyan.
o **darkGray**
The color dark gray.
o **gray**
The color gray.
o **green**
The color green.
o **lightGray**
The color light gray.
o **magenta**
The color magneta.
o **orange**
The color orange.
o **pink**
The color pink.

o **red**
The color red.
o **white**
The color white.
o **yellow**
The color yellow.

Constructor Index

o **Color**(int, int, int)
Creates a color with the specified red, green, and blue values in the range (0 – 255).
o **Color**(int)
Creates a 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.
o **Color**(float, float, float)
Creates a color with the specified red, green, and blue values in the range (0.0 – 1.0).

Method Index

o **HSBtoRGB**(float, float, float)
Returns the RGB value defined by the default RGB ColorModel, of the color corresponding to the given HSB color components.
o **RGBtoHSB**(int, int, int, float[])
Returns the HSB values corresponding to the color defined by the red, green, and blue components.
o **brighter**()
Returns a brighter version of this color.
o **darker**()
Returns a darker version of this color.
o **equals**(Object)
Compares this object against the specified object.
o **getBlue**()
Gets the blue component.
o **getColor**(String)
Gets the specified Color property.
o **getColor**(String, Color)
Gets the specified Color property of the specified Color.
o **getColor**(String, int)
Gets the specified Color property of the color value.
o **getGreen**()
Gets the green component.
o **getHSBColor**(float, float, float)
A static Color factory for generating a Color object from HSB values.
o **getRGB**()
Gets the RGB value representing the color in the default RGB ColorModel.
o **getRed**()
Gets the red component.

- **hashCode**()
  Computes the hash code.
- **toString**()
  Returns the String representation of this Color's values.

Variables

- **white**

```
public final static Color white
```

  The color white.

- **lightGray**

```
public final static Color lightGray
```

  The color light gray.

- **gray**

```
public final static Color gray
```

  The color gray.

- **darkGray**

```
public final static Color darkGray
```

  The color dark gray.

- **black**

```
public final static Color black
```

  The color black.

- **red**

```
public final static Color red
```

  The color red.

- **pink**

```
public final static Color pink
```

  The color pink.

o **orange**

```
public final static Color orange
```

The color orange.

o **yellow**

```
public final static Color yellow
```

The color yellow.

o **green**

```
public final static Color green
```

The color green.

o **magenta**

```
public final static Color magenta
```

The color magneta.

o **cyan**

```
public final static Color cyan
```

The color cyan.

o **blue**

```
public final static Color blue
```

The color blue.

Constructors

o **Color**

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

Creates a color with the specified red, green, and blue values in the range (0 – 255). The actual color used in rendering will depend on finding the best match given the color space available for a given output device.
Parameters:

r – the red component
g – the green component
b – the blue component

See Also:

getRed, getGreen, getBlue, getRGB

o **Color**

```
public Color(int rgb)
```

Creates a 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 will depend on finding the best match given the color space available for a given output device.

Parameters:

rgb – the combined RGB components

See Also:

getRGBdefault, getRed, getGreen, getBlue, getRGB

o **Color**

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

Creates a color with the specified red, green, and blue values in the range (0.0 – 1.0). The actual color used in rendering will depend on finding the best match given the color space available for a given output device.

Parameters:

r – the red component
g – the red component
b – the red component

See Also:

getRed, getGreen, getBlue, getRGB

Methods

o **getRed**

```
public int getRed()
```

Gets the red component.

See Also:

getRGB

o **getGreen**

```
public int getGreen()
```

Gets the green component.
See Also:
getRGB

o **getBlue**

```
public int getBlue()
```

Gets the blue component.
See Also:
getRGB

o **getRGB**

```
public int getRGB()
```

Gets the RGB value representing the color in the default RGB ColorModel. (bits 24–31 are 0xff, 16–23 are red, 8–15 are green, 0–7 are blue).
See Also:
getRGBdefault, getRed, getGreen, getBlue

o **brighter**

```
public Color brighter()
```

Returns a brighter version of this color.

o **darker**

```
public Color darker()
```

Returns a darker version of this color.

o **hashCode**

```
public int hashCode()
```

Computes the hash code.
Overrides:
hashCode in class Object

o **equals**

```
public boolean equals(Object obj)
```

Compares this object against the specified object.
Parameters:
obj – the object to compare with.
Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

o **toString**

```
public String toString()
```

Returns the String representation of this Color's values.

Overrides:

toString in class Object

o **getColor**

```
public static Color getColor(String nm)
```

Gets the specified Color property.

Parameters:

nm – the name of the color property

o **getColor**

```
public static Color getColor(String nm,
                             Color v)
```

Gets the specified Color property of the specified Color.

Parameters:

nm – the name of the color property

v – the specified color

Returns:

the new color.

o **getColor**

```
public static Color getColor(String nm,
                             int v)
```

Gets the specified Color property of the color value.

Parameters:

nm – the name of the color property

v – the color value

Returns:

the new color.

o **HSBtoRGB**

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

Returns the RGB value defined by the default RGB ColorModel, of the color corresponding to the given HSB color components.

Parameters:

hue – the hue component of the color
saturation – the saturation of the color
brightness – the brightness of the color

See Also:

getRGBdefault, getRGB

o **RGBtoHSB**

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

Returns the HSB values corresponding to the color defined by the red, green, and blue components.

Parameters:

r – the red component of the color
g – the green component of the color
b – the blue component of the color
hsbvals – the array to be used to return the 3 HSB values, or null

Returns:

the array used to store the results [hue, saturation, brightness]

See Also:

getRGBdefault, getRGB

o **getHSBColor**

```
public static Color getHSBColor(float h,
                                float s,
                                float b)
```

A static Color factory for generating a Color object from HSB values.

Parameters:

h – the hue component
s – the saturation of the color
b – the brightness of the color

Returns:

the Color object for the corresponding RGB color