

Introduction

The Color class states colors in the default sRGB color space or colors in arbitrary color spaces identified by a ColorSpace.

Class declaration

Following is the declaration for **java.awt.Color** class:

```
public class Color
    extends Object
    implements Paint, Serializable
```

Field

Following are the fields for **java.awt.geom.Arc2D** class:

- **static Color black** -- The color black.
- **static Color BLACK** -- The color black.
- **static Color blue** -- The color blue.
- **static Color BLUE** -- The color blue.
- **static Color cyan** -- The color cyan.
- **static Color CYAN** -- The color cyan.
- **static Color DARK_GRAY** -- The color dark gray.
- **static Color darkGray** -- The color dark gray.
- **static Color gray** -- The color gray.
- **static Color GRAY** -- The color gray.
- **static Color green** -- The color green.
- **static Color GREEN** -- The color green.
- **static Color LIGHT_GRAY** -- The color light gray.
- **static Color lightGray** -- The color light gray.
- **static Color magenta** -- The color magenta.
- **static Color MAGENTA** -- The color magenta.
- **static Color orange** -- The color orange.
- **static Color ORANGE** -- The color orange.
- **static Color pink** -- The color pink.
- **static Color PINK** -- The color pink.
- **static Color red** -- The color red.
- **static Color RED** -- The color red.
- **static Color white** -- The color white.

- **static Color WHITE** -- The color white.
- **static Color yellow** -- The color yellow.
- **static Color YELLOW** -- The color yellow.

Class constructors

S.N.	Constructor & Description
1	Color<i>ColorSpace</i><i>space</i>, <i>float[]</i><i>components</i>, <i>float</i><i>alpha</i> Creates a color in the specified ColorSpace with the color components specified in the float array and the specified alpha.
2	Color<i>floatr</i>, <i>floatg</i>, <i>floatb</i> Creates an opaque sRGB color with the specified red, green, and blue values in the range 0.0 – 1.0.
3	Color<i>floatr</i>, <i>floatg</i>, <i>floatb</i>, <i>floata</i> Creates an sRGB color with the specified red, green, blue, and alpha values in the range 0.0 – 1.0.
4	Color<i>intrgb</i> 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.
5	Color<i>intrgba</i>, <i>boolean</i><i>hasalpha</i> 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.
6	Color<i>intr</i>, <i>intg</i>, <i>intb</i> Creates an opaque sRGB color with the specified red, green, and blue values in the range 0 – 255.
7	Color<i>intr</i>, <i>intg</i>, <i>intb</i>, <i>inta</i> Creates an sRGB color with the specified red, green, blue, and alpha values in the range 0 – 255.

Class methods

S.N.	Method & Description
------	----------------------

1	Color brighter Creates a new Color that is a brighter version of this Color.
2	PaintContext createContext <i>ColorModelcm, Rectangler, Rectangle2Dr2d, AffineTransformxform, RenderingHintshints</i> Creates and returns a PaintContext used to generate a solid color pattern.
3	Color darker Creates a new Color that is a darker version of this Color.
4	static Color decodeStringnm Converts a String to an integer and returns the specified opaque Color.
5	boolean equalsObjectobj Determines whether another object is equal to this Color.
6	int getAlpha Returns the alpha component in the range 0-255.
7	int getBlue Returns the blue component in the range 0-255 in the default sRGB space.
8	static Color getColorStringnm Finds a color in the system properties.
9	static Color getColorStringnm, Colorv Finds a color in the system properties.
10	static Color getColorStringnm, intv Finds a color in the system properties.
11	float[] getColorComponentsColorSpaceospace, float[]compArray Returns a float array containing only the color components of the Color in the ColorSpace specified by the cspace parameter.
12	float[] getColorComponentsfloat[]compArray

Returns a float array containing only the color components of the Color, in the ColorSpace of the Color.

13

ColorSpace getColorSpace

Returns the ColorSpace of this Color.

14

float[] getComponentsColorSpace*space, float[]compArray*

Returns a float array containing the color and alpha components of the Color, in the ColorSpace specified by the cspace parameter.

15

float[] getComponents*float[]compArray*

Returns a float array containing the color and alpha components of the Color, in the ColorSpace of the Color.

16

int getGreen

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

17

static Color getHSBColor*float, float, float*

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

18

int getRed

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

19

int getRGB

Returns the RGB value representing the color in the default sRGB ColorModel.

20

float[] getRGBColorComponents*float[]compArray*

Returns a float array containing only the color components of the Color, in the default sRGB color space.

21

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.

22

int getTransparency

Returns the transparency mode for this Color.

23

int hashCode

Computes the hash code for this Color.

24

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.

25

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.

26

String toString

Returns a string representation of this Color.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object

Color Example

Create the following java program using any editor of your choice in say **D:/ > AWT > com > tutorialspoint > gui >**

AWTGraphicsDemo.java

```
package com.tutorialspoint.gui;

import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class AWTGraphicsDemo extends Frame {

    public AWTGraphicsDemo(){
        super("Java AWT Examples");
        prepareGUI();
    }

    public static void main(String[] args){
        AWTGraphicsDemo awtGraphicsDemo = new AWTGraphicsDemo();
        awtGraphicsDemo.setVisible(true);
    }

    private void prepareGUI(){
        setSize(400,400);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
    }

    @Override
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;
        Font plainFont = new Font("Serif", Font.PLAIN, 24);
```

```
g2.setFont(plainFont);  
g2.setColor(Color.red);  
g2.drawString("Welcome to Tutorialspoint", 50, 70);  
g2.setColor(Color.GRAY);  
g2.drawString("Welcome to Tutorialspoint", 50, 120);  
}  
}
```

Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AWTGraphicsDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\AWT>java com.tutorialspoint.gui.AWTGraphicsDemo
```

Verify the following output



Loading [MathJax]/jax/output/HTML-CSS/jax.js