

## 23.4.2.1 Klassen Color (gb.qt4 und gb.image)

Diese statische Klasse Color (gb.qt4) hat Eigenschaften, welche die System-Farben zurückgeben. Beachten Sie, dass in Gambas neben der Klasse Color (gb.qt4) auch eine Klasse Color (gb.image) existiert. Diese statische Klasse Color (gb.image) definiert Konstanten vordefinierter Farben und enthält einige nützliche Methoden, um mit Farben umzugehen. Es ist so, dass in Color (gb.image) die angesprochenen Farb-Konstanten und die nützlichen Methoden sind und in Color (gb.qt4) die Systemfarben. Die Klasse Color (gb.image) enthält Konstanten und Methoden, die nicht von bestimmten Bibliotheken abhängen. Die Konstanten liegen im RGB-Format vor. Die Methoden nutzen von Minisini geschriebene oder adaptierte Algorithmen. Die Systemfarben hingegen sind an das aktuelle System gebunden und werden vom aktuellen Toolkit bereitgestellt.

## 23.4.2.1.1 Klasse Color (gb.qt4) – Eigenschaften

Es werden die (statischen) Eigenschaften in der folgenden Tabelle beschrieben:

Farbname	dezimal	hexadezimal
Background	15724527	&HEFEFEF
ButtonBackground	14540772	&HDDDFE4
ButtonForeground	0	&H0
Foreground	0	&H0
LightBackground	13885419	&HD3DFEB
LightForeground	10921381	&HA6A5A5
SelectedBackground	6786482	&H678DB2
SelectedForeground	16777215	&HFFFFFF
TextBackground	16777215	&HFFFFFF
TextForeground	0	&H0
TooltipBackground	16777180	&HFFFDC
TooltipForeground	0	&H0

Tabelle 23.4.2.1.1: Farbkonstanten Klasse Color (gb.qt4)

## 23.4.2.1.2 Klasse Color (gb.image) – Konstanten

Diese Klasse Color (gb.image) ist statisch und definiert diese Farb-Konstanten:

Farbname	dezimal	hexadezimal	RGB	HSV
Default - Transparent	-1	&HFFFFFFF	-	-
Black	0	&H000000	0,0,0	0,0,0
White	16777217	&HFFFFFF	255,255,255	0,0,255
Gray	8421504	&H808080	128,128,128	0,0,128
DarkGray	4210752	&H404040	64,64,64	0,0,64
LightGray	12632256	&HC0C0C0	192,192,192	0,0,192
Red	16711680	&HFF0000	255,0,0	0,255,255
DarkRed	8388608	&H800000	128,0,0	0,255,128
Green	65280	&H00FF00	0,255,0	120,255,255
DarkGreen	32768	&H008000	0,128,0	120,255,128
Blue	255	&H0000FF	0,0,255	240,255,255
DarkBlue	128	&H000080	0,0,128	240,255,128
Yellow	16776960	&HFFFF00	255,255,0	60,255,255
DarkYellow	8421376	&H808000	128,128,0	60,255,128

Orange	16744448	&HFF8000	255,128,0	30,255,255
Magenta	16711935	&HFF00FF	255,0,255	300,255,255
DarkMagenta	8388736	&H800080	128,0,128	300,255,128
Cyan	65535	&H00FFFF	0,255,255	180,255,255
DarkCyan	32896	&H008080	0,128,128	180,255,128
Pink	16744703	&HFF80FF	255,128,255	300,127,255
Violet	8388863	&H8000FF	128,0,255	270,255,255

Tabelle 23.4.2.1 : Farbkonstanten der statischen Klasse Color (gb.image)

### 23.4.2.1.3 Klasse Color (gb.image) – Methoden

Die zahlreichen Methoden der Klasse Color (gb.image) ermöglichen dem Benutzer eine qualifizierte Arbeit mit Farben in Gambas.

Color	Beschreibung
RGB ( Red AS Integer, Green AS Integer, Blue AS Integer [ , Alpha AS Integer ] ) AS Integer	Gibt einen Farbwert aus dem RGB-Farbraum mit seinem roten, grünen und blauen Farb-Anteil zurück.
HSV ( Hue AS Integer, Saturation AS Integer, Value AS Integer [ , Alpha AS Integer ] ) AS Integer	Gibt einen Farbwert aus dem HSV-Farbraum mit seinen Farb-Anteilen Farbton, Sättigung und Helligkeit zurück.
SetRGB ( Color As Integer [ , Red As Integer, Green As Integer, Blue As Integer, Alpha As Integer ] ) As Integer	Die gesetzte Farbe wird optional durch die vier neuen RGBA-Farbanteile geändert. Wenn mindestens einer dieser vier Farbanteile angegeben ist, dann wird der entsprechende Farb-Anteil der gesetzten Farbe durch den angegebenen Parameter ersetzt. Wenn kein Farb-Anteil angegeben wurde, dann bleibt die gesetzte Farbe unverändert.
SetHSV ( Color As Integer [ , Hue As Integer, Saturation As Integer, Value As Integer, Alpha As Integer ] ) As Integer	Die gesetzte Farbe wird optional durch die vier neuen HSV-A-Farbanteile geändert. Wenn mindestens einer dieser vier Farbanteile angegeben ist, dann wird der entsprechende Farb-Anteil der gesetzten Farbe durch den angegebenen Parameter ersetzt. Wenn kein Farb-Anteil angegeben wurde, dann bleibt die gesetzte Farbe unverändert.
SetAlpha ( Color As Integer, Alpha As Integer ) As Integer	Ändert den Alpha-Wert einer Farbe und gibt die neue Farbe zurück.
GetAlpha ( Color As Integer ) As Integer	Gibt den Alpha-Wert der angegebenen Farbe zurück.
Darker ( Color As Integer ) As Integer	Gibt einen dunkleren Farbton der bestehenden Farbe zurück.
Lighter ( Color As Integer ) As Integer	Gibt einen helleren Farbton der bestehenden Farbe zurück.
Blend ( Source As Integer, Destination As Integer ) As Integer	Mischt je nach Alphakanal der beiden Farben die Quell-Farbe und die Ziel-Farbe und liefert die resultierende Farbe. Der Alpha-Kanal der entstehenden Farbe ist der <i>transparenteste</i> Alphakanal von Quelle und Ziel.
Merge ( Color1 As Integer, Color2 As Integer [ , Weight As Float ] ) As Integer	Gibt eine Mischung aus Color1 und Color2 zurück. Gewicht ist das relative Gewicht der ersten Farbe, das zwischen 0 und 1 liegt. Wenn das Gewicht nicht angegeben ist, wird es auf 0,5 gesetzt.
Desaturate ( Color As Integer ) As Integer	Gibt die angegebene Farbe (entsättigt) als Grauton zurück.
Gradient ( Color1 As Integer, Color2 As Integer [ , Weight As Float ] ) As Integer	Es wird – je nach Gewicht – eine Farbe zwischen Color1 und Color2 zurückgegeben. Ist das Gewicht 0, dann wird die Farbe Color1 zurückgegeben. Ist das Gewicht 1, dann wird die Farbe Color2 zurückgegeben. Wird kein Gewicht angegeben, dann wird das Gewicht auf 0,5 gesetzt.
Distance (Color1 As Integer, Color2 As Integer) As Float	Gibt den RGB-Abstand zwischen zwei Farben als Gleitkommazahl zwischen 0.0 und 1.0 zurück.

Tabelle 23.4.2.1.2: Methoden der Klasse Color (gb.image)

## 23.4.2.1.4 Beispiele für den Einsatz von Methoden der Klasse Color (gb.image)

Es werden in den ausgewählten Beispielen die Originalfarben und die geänderten Farben in zwei ColorButton oder drei ColorChooser angezeigt sowie der verwendete Quelltext angegeben.

## Beispiel 1 – Color.Darker(..)



```
Public Sub btnDarker_Click()
    Dim iSourceColor, iResultColor As Integer

    iSourceColor = ColorButton1.Color
    iResultColor = Color.Darker(iSourceColor)
    ColorButton2.Color = iResultColor
    '-- ColorButton2.Color = Color.Darker(ColorButton1.Color) '-- Kurzversion
End
```

## Beispiel 2 – Color.Lighter(..)



```
Public Sub btnLighter_Click()
    ColorButton2.Color = Color.Lighter(ColorButton1.Color)
End
```

## Beispiel 3 – Color.Desaturate(..)



```
Public Sub btnDesaturate_Click()
    ColorButton2.Color = Color.Desaturate(ColorButton1.Color)
End
```

## Beispiel 4 – Color.Distance(..)



```
Public Sub btnDistance_Click()
    Dim iColor1, iColor2 As Integer
    Dim fDistance As Float

    iColor1 = ColorButton1.Color
    iColor2 = ColorButton2.Color

    fDistance = Color.Distance(iColor1, iColor2)
    Print fDistance
End
```

Mit den beiden Farben Rot (Color.RGB(255,0,0)) und Grün (Color.HSV(120,255,255)) ergibt sich ein Distance-Wert von 0,70710678118655. Ist das nicht schön! Der Autor jedenfalls ist begeistert – kann aber mit diesem Wert nichts anfangen.

## Beispiel 5 – Color.Merge(..)



```
Public Sub btnMerge_Click()
    Dim iColor1, iColor2, iResultColor As Integer
```

```

Dim fWeight As Float

'-- Weight is the relative weight of the first color, between 0 and 1.
iColor1 = ColorButton1.Color
fWeight = 0.3
iColor2 = ColorButton2.Color

iResultColor = Color.Merge(iColor1, iColor2, fWeight)
'-- iResultColor = Color.Merge(iColor1, iColor2) ' ---> fWeight = 0.5 (Default)
ColorButton3.Color = iResultColor

End

```

Beispiel 6 – Color.Gradient(..)



```

Public Sub btnGradient_Click()

    Dim iColor1, iColor2, iResultColor As Integer
    Dim fWeight As Float

    iColor1 = ColorButton1.Color
    iColor2 = ColorButton2.Color
    '-- fWeight = 0.0 ' Sonderfall 1
    '-- fWeight = 1.0 ' Sonderfall 2
    fWeight = 0.4

    iResultColor = Color.Gradient(iColor1, iColor2, fWeight)
    '-- iResultColor = Color.Gradient(iColor1, iColor2) ' ---> fWeight = 0.5 (Default)
    ColorButton3.Color = iResultColor

End

```

Beispiel 7 – Color.Blend(..)



```

Public Sub btnBlend_Click()

    Dim iSourceColor, iDestinationColor, iResultColor As Integer

    iSourceColor = ColorButton1.Color
    iSourceColor = Color.SetAlpha(iSourceColor, 190)
    ColorChooser1.SelectedColor = iSourceColor

    iDestinationColor = ColorButton2.Color
    iDestinationColor = Color.SetAlpha(iDestinationColor, 90)
    ColorChooser2.SelectedColor = iDestinationColor

    iResultColor = Color.Blend(iSourceColor, iDestinationColor)
    Print Color.GetAlpha(iResultColor)
    '-- Alternative: Print Color[iResultColor].Alpha
    ColorChooser3.SelectedColor = iResultColor

End

```