

## 17.14 ColorButton

Farb-Werte als valide Daten in einem Programm erfordern sichere Eingaben. Für die direkte, aber dezentere Eingabe von Farb-Werten steht die Komponente *ColorButton* zur Verfügung. Mit dem Color-Button (gb.form) kann der Benutzer eine Farbe auswählen. Die Komponente hat zwei bemerkenswerte Eigenschaften und verfügt nur über ein spezielles Ereignis.

### 17.14.1 Eigenschaften und Ereignis

- Ein ColorButton hat *ColorButton.Color* (Typ Integer) als dominante Eigenschaft, dessen Wert Sie auslesen oder setzen können. Die Eigenschaft *ColorButton.Value* ist ein Synonym für die Eigenschaft *ColorButton.Color*.
- Das Ereignis *ColorButton\_Change()* wird ausgelöst, wenn Sie die Farbe im Farb-Dialog ändern.

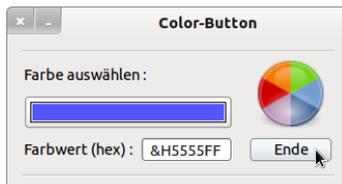


Abbildung 17.14.1.1: Sichere Eingabe von Farben → ColorButton

Nach einem Klick auf den ColorButton öffnet sich ein Farbauswahl-Dialog. Sie können die Farbe auswählen sowie den Farbwert anschließend mit OK übernehmen.

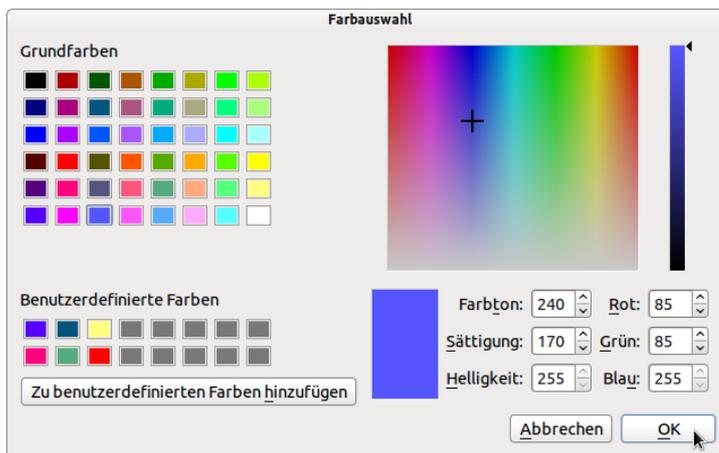


Abbildung 17.14.1.2: ColorButton-Dialog

Im folgenden Quelltext wird der Farbwert hexadezimal angezeigt (→ Abbildung 23.1.3.1.1) und in einer (globalen) Variablen *iFarbWert* gespeichert. Damit kann der Farbwert im Projekt weiter verwendet werden:

```
' Gambas class file

Public iFarbWert As Integer

Public Sub Form_Open()
  FMain.Center
  FMain.Resizable = False
  PictureBox1.Stretch = True
  PictureBox1.Picture = Picture["Symbols/color.png"]
  ColorButton.Color = Color.Orange
  txtColor.Text = "&H" & Hex(ColorButton.Color, 6)
End ' Form_Open()

Public Sub btnClose_Click()
  FMain.Close
End ' btnClose_Click()

*****
```

```
Public Sub ColorButton_Change()  
    iFarbWert = ColorButton.Color  
    txtColor.Text = "&H" & Hex(ColorButton.Color, 6)  
End ' ColorButton_Change()
```

### 17.14.1.1 Einsatz ColorButton

Im Gambas-Beispiel *Editor* werden zwei ColorButton zur Festlegung der Text-Farbe und zur Festlegung der Text-Hintergrundfarbe eingesetzt.

In der Prozedur *Form\_Open()* erfolgt die Initialisierung der Text-Farbe und der Text-Hintergrundfarbe:

```
Object.Lock(ColorButton1)  
Object.Lock(ColorButton2)  
    ColorButton1.Color = TextEdit1.Format.Color  
    ColorButton2.Color = TextEdit1.Format.Background  
Object.Unlock(ColorButton1)  
Object.Unlock(ColorButton2)
```

Änderungen der Text-Farbe und der Text-Hintergrundfarbe werden im Editor-Quelltext so realisiert:

```
Public Sub ColorButton1_Change()  
    TextEdit1.Format.Color = ColorButton1.Color  
End  
  
Public Sub ColorButton2_Change()  
    TextEdit1.Format.Background = ColorButton2.Color  
End
```

### 17.14.1.2 Alternative ColorButtonBox

Bei einem ColorButton sieht man im Original nur die ausgewählte Farbe. Nutzt man dagegen eine ButtonBox, so können Sie den (hexadezimalen) Farbwert in der TextBox lesen und die Farbe im umgefärbten Icon sehen:



Abbildung 17.14.1.2.1: ColorButtonBox in Aktion

Der Quelltext ist recht einfach:

```
Public Sub ColorButtonBox_Click()  
    Dim picColor As Picture  
  
    If Dialog.SelectColor() Then  
        ColorButtonBox.Text = "Farbwert"  
        ColorButtonBox.Picture = Picture["Symbols/color.png"]  
    Else  
        ' ColorButtonBox.Text = Dialog.Color ' Anzeige dezimal  
        ColorButtonBox.Text = "&H" & Hex(Dialog.Color, 6)  
        picColor = New Picture(16, 12, False)  
        picColor.Fill(Dialog.Color)  
        ColorButtonBox.Picture = picColor  
    Endif ' Dialog.SelectColor() ?  
End ' ColorButtonBox_Click()
```

Es wird der originale Farbauswahl-Dialog (→ Dialog.SelectColor) des ColorButton eingesetzt.