

## 16.2 TextLabel

Die Komponente Text-Label kann einfachen HTML-Text – auch bekannt als RichText – anzeigen. Ein Text-Label wird immer dann zum Einsatz kommen, wenn Sie den anzuzeigenden Text nur in einfacher Form formatiert ausgeben wollen:



Abbildung 16.2.1: Anzeige von umfangreichem RichText in einem TextLabel

### 16.2.1 Eigenschaften Text-Label

In einem Text-Label können Sie Text anzeigen, den Sie über die Eigenschaft `TextLabel.Text` oder `TextLabel.Caption` zuweisen. Weitere Eigenschaften eines Text-Labels werden in der Tabelle beschrieben:

TextLabel	Datentyp	Default	Beschreibung
.Alignment	Integer	0	Setzt die horizontale Ausrichtung des Textes im Text-Label oder gibt dessen Ausrichtung zurück → Klasse <i>Align</i>
.AutoSize	Boolean	False	Gibt an, ob sich die Größe des Text-Labels automatisch an den Text anpasst
.Border	Integer	None	Setzt den Rand des Text-Labels oder bestimmt die Randform oder gibt an, ob ein Rand vorhanden ist und welche Randform verwendet wird → Klasse <i>Border</i>
.Caption	String		Synonym für <code>.Text</code>
.Text	String		Setzt den anzuzeigenden Text des Text-Labels oder gibt diesen zurück
.Padding	Integer	0	Setzt einen Abstand zwischen Text und Text-Label-Grenzen (Rand)
.Transparent	Boolean	False	Legt die Transparenz des Text-Labels fest oder gibt an, ob das Text-Label transparent ist

Tabelle 16.2.1.1: Ausgewählte Text-Label-Eigenschaften

### 16.2.2 Syntax von RichText in Gambas

RichText in Gambas nutzt eine Untermenge von HTML. Die folgenden HTML-Tags sind erlaubt, um Textelemente auszuzeichnen:

```
<h1>, <h2>, <h3>, <h4>, <h5>, <h6> → Überschriften
<b> → Fettschrift, <i> → kursiv, <s> → durchgestrichen, <u> → unterstrichen
<sub> → tiefgestellt, <sup> → hochgestellt, <small> → schmal, <p> → Absatz
<br> → Zeilenwechsel, <a> → Link, <font> → Schrift-Font
```

Informationen zum `<font>`-Tag werden in der folgenden Tabelle angegeben. Bitte beachten Sie, dass der Tag `<font>` unter HTML5 nicht mehr existiert.

Font-Attribut	Werte	Hinweise
face	Schriftfamilie	Verschiedene Schriften können in einer komma-getrennten Liste angegeben werden
color	Farbwerte über Farbnamen oder als hexadezimale Zahl oder im RGB-Format	color = Red color = #C3DDFF → hellblau color = RGB(255,175,255,95) → orange
size	Schriftgröße (Default = 3)	Absolute Werte von 1 bis 7 oder die zwei relativen Werte -1 für kleinere Schrift oder +1 für größere Schrift

Tabelle 16.2.2.1: Informationen zum `<font>`-Tag

Mit `<a>...</a>` fügen Sie einen Link in den RichText ein. Er wird unterstrichen und in der Farbe blau formatiert ausgegeben – so, wie es für einen Hyperlink erwartet wird – er bleibt jedoch ohne Wirkung! Nach einem Klick mit der rechten Maustaste auf den Link können Sie sich aber im Kontextmenü die Link-Adresse kopieren.

Erprobt sind auch diese Tags, um ein- oder mehrzeiligen Quelltext auszuzeichnen oder um eine horizontale Linie einzufügen oder eine unsortierte Liste mit n Einträgen anzuzeigen:

```
<tt> → einzeiliger Quelltext, <pre> → mehrzeiliger Quelltext, <hr> → horizontale Linie
<ul> → unsortierte Liste, <li> → Listeneintrag
```

Hinweise:

- Ändern Sie die Font-Eigenschaften eines Text-Labels über das `<font>`-Tag. Wenn Sie nur eine Schrift mit einem Stil und in einer Größe verwenden wollen, können Sie darauf verzichten und setzen die Font-Eigenschaften im Eigenschaften-Fenster (IDE) oder im Quelltext.
- Das Arbeiten mit CSS wird nicht unterstützt, so dass dieser Text in einem Text-Label nicht so angezeigt wird, wie das in jedem Browser möglich wäre.
- Es hat sich als günstig erwiesen, die Größe des Text-Labels an den anzuzeigenden Text anzupassen.
- Sie können jede HTML-Entität verwenden → <http://unicode.e-workers.de/entities.php>
- Um schnell und effizient einen längeren Text mit den o.a. Tags (RichText) auszuzeichnen reicht ein Texteditor wie *BlueFish* aus. Legen Sie eine HTML-Datei mit diesem Editor an und speichern Sie diese im Projektverzeichnis des Gambas-Projekts, das den RichText verwenden soll. In der IDE können Sie den Inhalt der HTML-Datei mit dem Texteditor *BlueFish* öffnen und bearbeiten.

Wie bei einem einfachen Label reagieren Sie mit der Methode `TextLabel.Adjust()`, wenn sich der Text für ein Text-Label zur Laufzeit geändert hat und sich die Text-Label-Größe automatisch an den geänderten Text anpassen soll:

```
Public Sub btnAdjust_Click()
    TextLabel.Wrap = False ' Bei längeren Textpassagen TRUE
    TextLabel.AutoResize = False
    TextLabel.Padding = 8
    TextLabel.Text = "<font size=4 face=Ubuntu color=#C3DDFF>IP-Adresse &rarr; IPv4</font>"
    TextLabel.Text &= "<hr>Informationen auf <a href=http://de.wikipedia.org/wiki/IPv4>"
    TextLabel.Text &= "www.wikipedia.de</a>!"
    TextLabel.Adjust()
End ' btnAdjust_Click()

Public Sub btnAutoResize_Click()
    TextLabel.AutoResize = True
    TextLabel.Padding = 8
    TextLabel.Text = "<font size+=1 face=Arial color=Red>Neue IP-Adresse &rarr; IPv6</font>"
    TextLabel.Text &= "<hr>Informationen auf <a href=http://de.wikipedia.org/wiki/IPv6>"
    TextLabel.Text &= "www.wikipedia.de</a>!"
End ' btnAutoResize_Click()
```

Alternativ setzen Sie die Eigenschaft `TextLabel.AutoResize` auf True, um den gleichen Effekt zu erzielen.

### 16.2.3 Projekt

Eine Grafik mit mehreren sensitiven, anklickbaren Bereichen nennt man *ImageMap*. Diese Bereiche definieren so genannte *HotSpots* innerhalb eines Bildes → Abbildung 16.2.3.1, die Sie mit ausgewählten Aktionen verlinken. Das können Aufrufe von HTML-Seiten, von Dateien oder von Programmen sein. Die Website des Gambas-Projekts zum Beispiel nutzt solche HotSpots in einer ImageMap um im Abschnitt *Gambas-Architektur* Erläuterungen zur Architektur der Sprache Gambas anzuzeigen. Um eine solche Anzeige in einem Gambas-Projekt nachzubilden, werden für die Hotspots auf einem Bild *transparente Label* über ausgewählten Bereichen eingefügt. Die Anzeige des mit einem HotSpot korrespondierenden Textes wird in einem *Text-Label* realisiert. Der Text für alle Hotspots liegt in einer Datei im RichText-Format. Jeder Textabschnitt wird durch ein Pipe-Zeichen abgeschlossen. Das Zeichen darf natürlich im Text nicht vorkommen. Außerdem wird die Group-Eigenschaft `allert` transparenten (HotSpot-) Label auf den Wert 'HotSpots' gesetzt und die Eigenschaft `(HotSpot-)Label.Tag` vom Typ String wird dazu verwendet, den anzuzeigenden Text aufzunehmen.

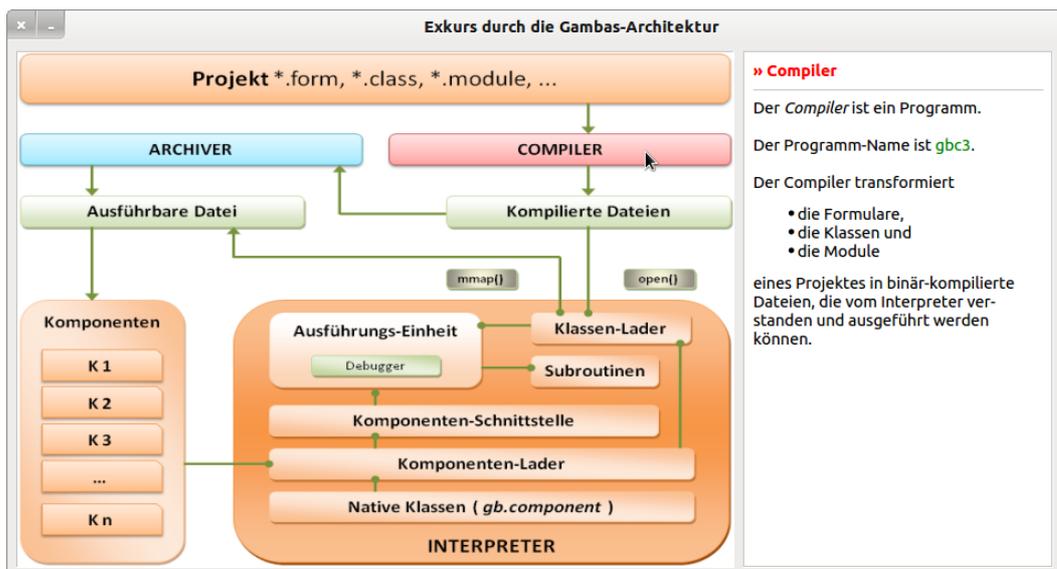


Abbildung 16.2.3.1: Anzeige von RichText für jeden HotSpot in einem Text-Label

Wie Sie sehen ist der Quelltext für das Projekt sehr überschaubar:

```
' Gambas class file

Private mText As String[]

Public Sub Form_Open()

    FMain.Center
    FMain.Resizable = False

    mText = New String[]
    mText = Split(File.Load("Text/architecture.txt"), "|")

    lblProject.Tag = mText[0]
    lblCompiler.Tag = mText[1]
    lblArchiver.Tag = mText[2]
    lblCompiledFiles.Tag = mText[3]
    lblExecutableFile.Tag = mText[4]
    lblMMAP.Tag = mText[5]
    lblClassLoader.Tag = mText[6]
    lblExecutionUnit.Tag = mText[7]
    lblSubRoutines.Tag = mText[8]
    lblComponentInterface.Tag = mText[9]
    lblComponentLoader.Tag = mText[10]
    lblInterpreter.Tag = mText[11]
    lblComponents.Tag = mText[12]
    lblNativeClasses.Tag = mText[13]

End ' Form_Open()

Public Sub HotSpots_MouseDown()
    TextLabel1.Text = Last.Tag
End ' HotSpots_MouseDown()
```

Die Prozedur *HotSpots\_MouseDown()* ersetzt, in Verbindung mit dem speziellen Objekt Last - das eine Referenz auf das Objekt zurück gibt, das als letztes ein Ereignis ausgelöst hat – hier mindestens 14 einzelne Prozeduren!