

25.1.7 CairoTextExtents

In den Eigenschaften der zwei Klassen *CairoTextExtents* und → 25.1.8 *CairoFontExtents* werden die Maße einer einzelnen Glyphen oder einer Aneinanderreihung von mehreren Glyphen gespeichert. Die zahlreichen, recht unterschiedlichen Schrift-Maße werden in der so genannten *typographischen Metrik* beschrieben.

Es wird in diesem Kapitel und im → Kapitel *CairoFontExtents* nur die typographische Metrik für ein *horizontales* Layout einer Schrift besprochen. Dazu gehört auch die Schrift des Textes, den Sie gerade lesen. Kennzeichen für ein *horizontales* Layout ist eine (virtuelle) horizontale (Text-)BaseLine → Abbildung 25.1.7.1.1.

In Gambas werden bestimmte Schrift-Maße *gruppiert* und jeweils in den Eigenschaften der zwei Klassen *CairoTextExtents* und → 25.1.8 *CairoFontExtents* beschrieben. Die eine Gruppe bezieht sich nur auf den verwendeten *Schrift-Font*, während die andere sich auf die *gezeichneten* Glyphen des vorgegebenen Textes bezieht. Für welche Eigenschaften der beiden Klassen Sie sich beim Zeichnen von Text entscheiden, wird wesentlich von Ihren Designvorstellungen für die Zeichnung bestimmt.

25.1.7.1 Eigenschaften

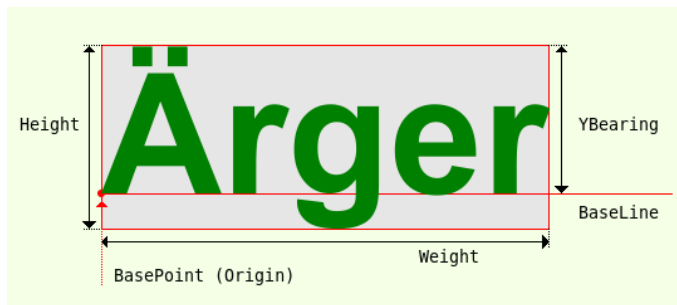


Abbildung 25.1.7.1.1: Fiktiver Text-Begrenzungsrahmen (bounding box)

Die Klasse *CairoTextExtents* beschreibt Eigenschaften der typographischen Metrik vom Typ Float:

Eigenschaft	Beschreibung
Height	Gibt die Höhe der Glyphen an - so wie sie gezeichnet werden.
Width	Gibt die Weite der Glyphen an - so wie sie gezeichnet werden.
YBearing	Gibt die vertikale Entfernung von der BaseLine zu dem obersten Teil der Glyphen - wie gezeichnet - an. Der Abstand ist nur dann positiv, wenn die Glyphen vollständig unterhalb des Ursprungs liegen. <i>In der Regel ist dieser Wert negativ.</i>
XAdvance	Gibt den Abstand in der x-Richtung nach dem Zeichnen der Glyphen an.
XBearing	Gibt die horizontale Entfernung vom BasePoint (Origin - Ursprung) zu dem am weitesten links stehenden Teil der Glyphen an – so wie gezeichnet. Der Abstand ist positiv, wenn die Glyphen vollständig auf der rechten Seite vom BasePoint liegen.
YAdvance	<i>Gibt den Abstand in der y-Richtung nach dem Zeichnen dieser Glyphen an. Der Wert ist Null für ein horizontales Layout.</i>

Tabelle 25.1.7.1.1 : Eigenschaften der Klasse CairoTextExtents

Kommentar:

- Das Zeichnen von Text sollte vornehmlich für Bezeichnungen von ausgewählten Details einer Zeichnung eingesetzt werden. Ausnahmen könnten Texte mit einer klaren Struktur in Form von Zeilen und Spalten bilden – so wie sie zum Beispiel für Tabellen typisch sind. In den Projekten zu Cairo finden Sie deshalb auch ein Projekt für einen einfachen Datenbank-Report, der mit Methoden von Klassen der Komponente Cairo gezeichnet wurde.
- Für das Zeichnen von Text kommen Sie mit den Eigenschaften Weite, Höhe und YBearing aus, da die Werte für XBearing für Font-Größen bis 100 hinreichend Null sind und YAdvance für das hier betrachtete horizontale Schrift-Layout generell Null ist.

25.1.7.2 Hinweise

Unter dem → Link <http://de.wikipedia.org/wiki/Glyphe> finden Sie Informationen zum Begriff 'Glyph', der immer wieder im Zusammenhang mit Eigenschaften und Methoden unterschiedlicher Klassen der Komponente Cairo (gb.cairo) verwendet wird. Dort wird u.a. sinngemäß herausgestellt, dass *in der Typografie eine Glyphie die grafische Darstellung eines Schriftzeichens, zum Beispiel eines Buchstabens, eines Silbenzeichens, einer Ligatur oder eines Buchstabenteils ist. Die Glyphie bildet dabei in sich eine grafische Einheit, wobei das Zeichen die abstrakte Idee eines Buchstabens ist und die Glyphie deren konkrete grafische Darstellung.* Viele Informationen zur typographische Metrik finden Sie auf dieser Website → <http://www.freetype.org/freetype2/docs/glyphs/glyphs-3.html>. Es kann empfohlen werden, sich diese Seite auf jeden Fall anzusehen, weil der Inhalt vor allem das Verständnis für die vielen Bezeichnungen der typographische Metrik unterstützt.

Da Text-Maße in User-Space-Koordinaten angegeben werden, sind diese in der Regel – aber nicht immer – unabhängig von der aktuellen Transformationsmatrix. Wenn Sie zum Beispiel die Methode `Cairo.Scale(2.0, 2.0)` aufrufen, so wird der Text doppelt so groß aufgezogen – aber das ermittelte Text-Maß wird nicht verdoppelt sein.

25.1.7.3 Beispiel

Das Beispiel demonstriert den Einsatz aller Eigenschaften der Klasse `CairoTextExtents` und das Auslesen der Werte für diese Eigenschaften:

```
Private Sub GeneratePDF()
    Dim PDFSurface As CairoPdfSurface
    Dim fXOffset As Float = 20
    Dim fYOffset As Float = 20

    PDFSurface = New CairoPdfSurface(sPfadPDFDatei, 210, 297) ' → A4-Format der Oberfläche

    Cairo.Begin(PDFSurface)
    Cairo.Matrix = Cairo.Matrix.Translate(MMToPoints(fXOffset), MMToPoints(fYOffset))
    Cairo.Matrix = Cairo.Matrix.Scale(1, 1) ' Zoom-Faktor = 1

    Cairo.AntiAlias = False
    Line(MMToPoints(0), MMToPoints(0), MMToPoints(0), MMToPoints(297 - fYOffset), 0.1, [1, 1], Color.Red)
    Line(MMToPoints(0), MMToPoints(0), MMToPoints(210 - fXOffset), MMToPoints(0), 0.1, [1, 1], Color.Red)
    Cairo.Stroke

    ' TEXTE
    Cairo.Source = Cairo.ColorPattern(Color.DarkBlue)
    Cairo.Font.Name = "Arial"
    Cairo.Font.Size = 100
    Cairo.Font.Bold = True

    Print
    Print "TEXT"
    Print "-----"
    Print "Height = ", PointsToMM(Cairo.TextExtents("Ärger").Height)
    Print "Width = ", PointsToMM(Cairo.TextExtents("Ärger").Width)
    Print "XAdvance = ", PointsToMM(Cairo.TextExtents("Ärger").XAdvance)
    Print "YAdvance = ", PointsToMM(Cairo.TextExtents("Ärger").YAdvance)
    Print "XBearing = ", PointsToMM(Cairo.TextExtents("Ärger").XBearing)
    Print "YBearing = ", PointsToMM(Cairo.TextExtents("Ärger").YBearing)

    Cairo.MoveTo(MMToPoints(0), MMToPoints(50))

    Cairo.DrawText("Ärger")

    Cairo.End

    PDFSurface.Finish()
    Desktop.Open(sPfadPDFDatei)
End ' GeneratePDF

Private Function MMToPoints(Value As Float) As Float
    Return Value * 2.83527
End ' MMToPoints(..)

Private Sub Line(Xa As Float, Ya As Float, Xe As Float, Ye As Float, fWidth As Float, Optional fDash As Float[], Optional cColor As Integer)
    ' Farbige Linie von Punkt A nach Punkte E - A(xa|ya), E(xe|ye) - Angaben in Millimetern!
    Cairo.MoveTo(MMToPoints(Xa), MMToPoints(Ya))
    Cairo.LineTo(MMToPoints(Xe), MMToPoints(Ye))
End Sub
```

```
Cairo.Source = Cairo.ColorPattern(cColor)
Cairo.LineWidth = fWidth
Cairo.Dash = fDash
Cairo.Stroke()
Cairo.Dash = Null
End ' Line(..)

Private Function PointsToMM(Value As Float) As Float
    Return Value * 0.3527
End ' MMToPoints(..)
```

Mit dem o.a. Quelltext-Ausschnitt ergeben sich diese Ausgaben – gerundet und mit der Einheit Millimeter versehen – in der Konsole der IDE:

```
TEXT
-----
Height =      38,39 mm
Width =       94,53 mm
XAdvance =    94,08 mm
YAdvance =     0 mm
XBearing =     0 mm
YBearing =    -30,97
```

Kommentar:

- Über die beiden Funktionen *MMToPoints(Value As Float)* und *PointsToMM(Value As Float)* können Sie User-Koordinaten bequem in Millimeter und Koordinaten in Millimetern in User-Koordinaten konvertieren.
- Wenn Sie die generierte PDF-Datei ausdrucken und für das Wort 'Ärger' den in der → Abbildung 25.1.7.1.1 angegebenen (fiktiven) Begrenzungsrahmen einzeichnen und die Höhe, Weite sowie das Maß |YBearing| über der BaseLine messen, so erhalten Sie mit hinreichender Genauigkeit die berechneten Werte in der o.a. Übersicht.
- Beachten Sie, dass der Wert yBearing negativ ist!