

RedCrab

The Calculator

Neuheiten in der Bedienungsanleitung Version 4.40

copyright © by Redchillicrab, Singapore 2009 - 2013

Neu ab Version 4.40

13.9 Plot Box

Das Menü Plotbox öffnet auf dem Arbeitsblatt eine Grafikbox, in der Ergebnisse von Berechnungen grafisch dargestellt werden können. Die Handhabung der Initialisierung ist ähnlich der von Resultat- oder Chartboxen. Durch Anklicken der Box mit der rechten Maus Taste öffnen Sie das Popup Menü der Plotbox. Mit dem Menü References öffnen Sie ein Dialogfenster in dem der Name der Variable eingegeben wird, mit der die Plotbox verbunden wird.

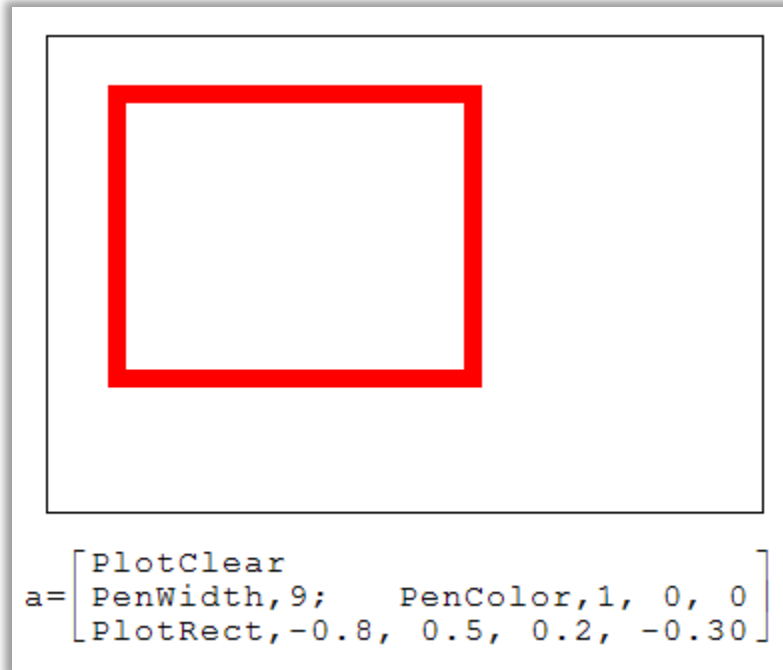
Im Unterschied zur Chartbox verarbeitet die Plotbox Grafikbefehle auf niedrigerer Ebene. Einer Chartbox wird eine Referenzvariable mit einem Datenfeld übergeben, dessen Daten in der Chartbox grafisch angezeigt wird. Die Handhabung ist einfach, aber die Möglichkeiten sind auf die von der Chartbox vorgegebenen Serien und Optionen begrenzt.

Die Plotbox erwartet als Referenz ein Datenfeld das aus einer Serie von Grafikbefehlen und Parametern besteht. Das bedeutet das Resultate, anders als bei der Chartbox, nicht direkt angezeigt werden können. Die Daten müssen zuerst in eine Folge von Grafikbefehlen umgesetzt werden. Die grafische Gestaltung unterliegt dabei keinen Einschränkungen.

Datenformat der Plotbox

Die Plotbox erwartet als Referenz ein zweidimensionales Datenfeld. Die erste Zeile einer Zeile muß den Grafikbefehl enthalten. Die folgenden Zeilen enthalten eine unterschiedliche Anzahl von Parametern.

Beispiel:



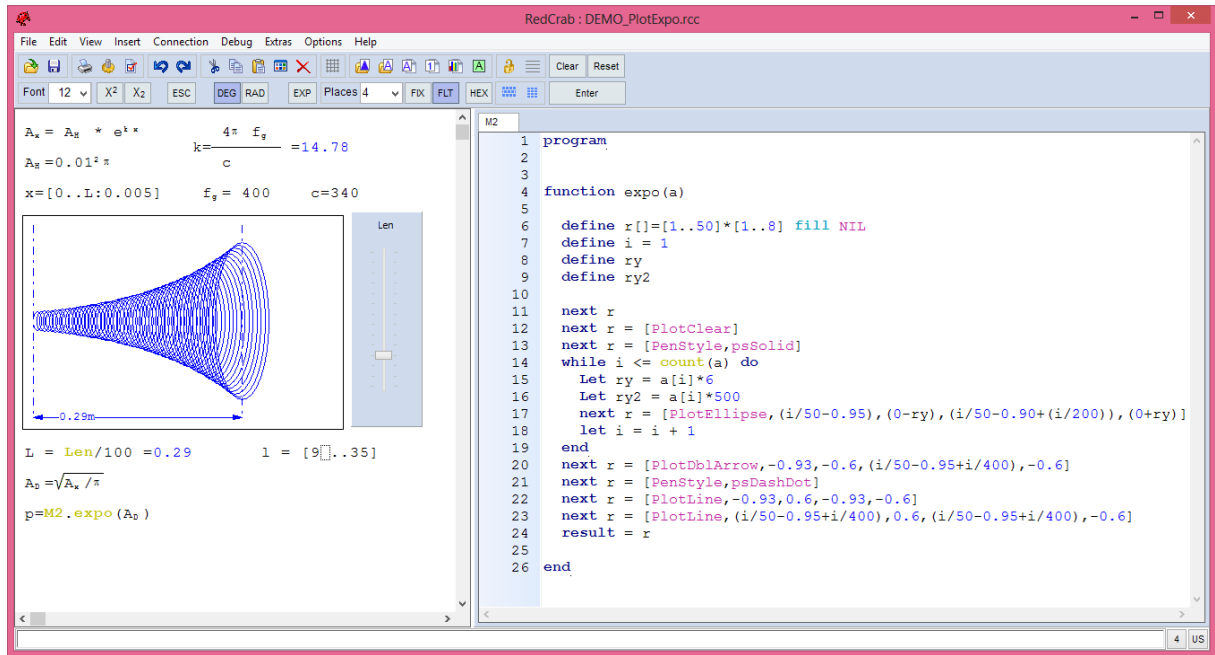
Die

Referenzvariable *a* in dem Beispiel oben, enthält eine vierzeilige Befehlssequenz, die ein rotes Rechteck zeichnet.

1. ***PlotClear*** löscht die Plot Box mit der eingestellten Hintergrundfarbe .
2. ***PenWidth*** stellt die Breite des Zeichenstifts auf 9 Pixel ein. Das Semikolon hinter der neun leitet eine neue Zeile ein.
3. ***PenColor*** setzt die Farbe des Zeichenstifts auf rot.
4. ***PlotRect*** zeichnet ein Rechteck.

Die Einbindung der Grafikbefehle in ein Datenfeld ermöglicht das eine Grafik, oder Teile davon, als Parameter in Funktionen importiert, oder als Resultat exportiert werden können.

In dem Beispiel unten wird das Exponetionalhorn eines Hochtonlautsprechers berechnet. Die Daten des Horns werden als Parameter an die externe Funktion ***M2.expo()*** übergeben. ***M2.expo()*** liefert als Resultat ein Datenfeld an die Variable ***p***, in dem das Resultat grafisch dargestellt wird. ***p*** wird als Referenzvariable der Plotbox zugewiesen.



13.9.1 Farb Komponenten

In RedCrab wird eine Farbe durch die Rot-, Grün und Blau-Komponente, in dieser Reihenfolge, bestimmt. Der Wert der einzelnen Komponenten wird durch eine Fließkommazahl zwischen 0.0 und 1.0 bestimmt.

Syntax: [SetPenColor, red, green, blue]

Beispiel: [SetPenColor, 0.98, 0.625, 0.12]

In dem Beispiel oben, wird die Farbe des Zeichenstifts auf orange eingestellt.

Die Tabelle unten zeigt die Komponenten einiger Farbtöne. Diese Werte können in alle Ausdrücke eingesetzt werden, die die Farbe betreffen.

Farbton	Rot	Grün	Blau
Weiß	1.0	1.0	1.0
Schwarz	0.0	0.0	0.0
Rot	1.0	0.0	0.0
Grün	0.0	1.0	0.0
Blau	0.0	0.0	1.0

Gelb	1.0	1.0	0.0
Magenta	1.0	0.0	1.0
Cyan	0.0	1.0	1.0
Dunkel Grau	0.25	0.25	0.25
Hell Grau	0.75	0.75	0.75
Braun	0.60	0.40	0.12
Orange	0.98	0.625	0.12
Pink	0.98	0.04	0.70
Violett	0.60	0.40	0.70

Die Verwendung der Rot-, Grün- und Blau Komponenten ist identisch mit der Windows Spezifikation, mit dem Unterschied, daß bei Windows jede Komponente einen Wert-Bereich von 1 bis 255 hat. Wenn Sie diesen Bereich bevorzugen, können Sie ihn mit der Anweisung [AbsColor, TRUE] auf diesen Bereich umschalten.

13.9.2 Plot Koordinaten

Ähnlich der Farbkomponenten können auch die X,Y-Koordinaten als Fließkommazahl oder in absoluten Pixel Positionen eingegeben werden. Voreingestellt ist der Fließkomma Modus. In diesem Modus kann jede gültige positive oder negative Fließkommazahl zur Positionsbestimmung verwendet werden. Die niedrigsten Werte repräsentieren den linken bzw. den unteren Rand der Plotbox. Die Höhe und Breite der Grafik wird maximiert an die Größe der Box angepaßt.

Wenn Sie keine automatische Anpassung der Größe wünschen, können Sie die Minimum- und Maximumwerte der X,Y-Achsen mit ***PlotRange*** festlegen. Hier kann auch jede gültige positive oder negative Fließkommazahl eingesetzt werden.

Anstelle der Fließkomma Koordinaten können Sie die Positionen auch in absoluten Pixelkoordinaten bestimmen. Bevor Sie Pixelkoordinaten verwenden können müssen Sie die Anweisung [AbsPosition,TRUE] eintragen. Entsprechend der Windows Spezifikation repräsentieren dann die niedrigsten Werte (x=1) den linken Rand, (y=1) den oberen Rand der Plotbox. Die Grafik hat in diesem Modus eine feste Größe und wird nicht an die Größe der Plotbox angepaßt.

13.9.3 Plot Komandos

Die folgende Tabelle zeigt eine Übersicht der Grafikbefehle, die im Folgenden ausführlich beschrieben sind.

PlotClear	löscht den Inhalt der Plotbox.
ClearColor	bestimmt die Hintergrundfarbe für <i>PlotClear</i> .
PlotSize	bestimmt die Breite und die Höhe der Plotbox in Pixel.
AbsPosition	bestimmt den Datentyp der Argumente zur Positionsbestimmung.
AbsColor	bestimmt der Bereich der Farbkomponenten.
PlotRange	bestimmt den Bereich der X und Y Achse.
PlotBorder	schaltet die Umrandung der Plotbox ein oder aus.
BorderColor	bestimmt die Farbe der Umrandung.
MoveTo	den Zeichenstift auf eine bestimmte X/Y Position.
LineTo	zeichnet eine Linie ab der aktuellen Zeichenposition.
PlotLine	zeichnet eine Linie zwischen zwei angegebenen Positionen.
PlotArrow	zeichnet eine Linie mit einer Pfeilspitze am Kopfende.
PlotDblArrow	zeichnet eine Linie mit je einer Pfeilspitze an den Enden.
FrameRect	zeichnet ein Rechteck.
FillRect	füllt eine rechteckige Fläche.
PlotRect	zeichnet ein Rechteck.
RoundRect	zeichnet ein Rechteck mit abgerundeten Ecken.
PlotEllipse	zeichnet eine Ellipse.
PlotArc	zeichnet eine Kreisbogen.
PlotChord	zeichnet einen geschlossenen Kreisbogen.
PlotPie	zeichnet ein Tortendiagramm.
PolyBezier	zeichnet eine Bezier Kurve.
PlotPolygon	zeichnet ein geschlossenes Vieleck.
KeepPolygon	erweitert <i>PlotPolygon</i> .
PenColor	bestimmt die Farbe des Zeichenstifts.
PenWidth	bestimmt die Breite des Zeichenstifts.
PenStyle	bestimmt das Muster, in dem Linien gezeichnet werden.
PenMode	bestimmt die Interaktion der Vordergrund- mit der Hintergrundfarbe.
BrushColor	bestimmt die Vordergrund Farbe beim Füllen von Flächen.
BrushStyle	spezifiziert das Muster mit dem eine Fläche gefüllt wird.
PlotText	schreibt einen Text an die aktuelle Pen Position.
FontSize	bestimmt Größe der Zeichen bei der Textausgabe.
FontStyle	spezifiziert die Attribute der Textausgabe.
FontColor	bestimmt die Farbe in der Text geschrieben wird.
FontName	bestimmt den Zeichensatz für Textausgabe.

13.9.4.1 PlotClear

PlotClear löscht den Inhalt der Plotbox.

Syntax: [PlotClear]

PlotClear löscht den Inhalt der Plotbox. Der Hintergrund wird mit der Farbe gefüllt, die mit ClearColor eingestellt wurde.

13.9.4.2 ClearColor

ClearColor bestimmt die Hintergrundfarbe für **PlotClear**.

Syntax: [ClearColor, red, green, blue]

Der gültige Bereich der Werte ist eine Fließkomma Zahl zwischen 0 und 1.0, oder zwischen 0 und 255 wenn **AbsColor** auf **TRUE** gesetzt ist.

13.9.4.3 PlotSize

PlotSize bestimmt die Breite und die Höhe der Plotbox in Pixel.

Syntax: [PlotSize, width, height]

PlotSize ändert die Breite und Höhe der Plotbox entsprechend den Argumenten width und height. Außerdem wird automatisch PlotClear ausgeführt und. PlotSize muß als immer aufgerufen werden bevor eine Grafik gezeichnet wird.

13.9.4.4 AbsPosition

AbsPosition bestimmt den Datentyp der Argumente zur Positionsbestimmung.

Syntax: [AbsPosition, {TRUE | FALSE}]

Nach der Anweisung [AbsPosition, TRUE] werden die Angaben von Positionen in der Plotbox in Pixelkoordinaten, entsprechend der Windows Spezifikation erwartet. Mit [AbsPosition, FALSE] wird auf fließkomma Eingabe zurückgeschaltet. Voreingestellt ist die Eingabe von Fließkommazahlen. Weitere Informationen dazu finden Sie oben in dem Absatz **Plot Koordinaten**.

13.9.4.5 AbsColor

AbsColor bestimmt der Bereich der Farbkomponenten.

Syntax: [AbsColor, {TRUE | FALSE}]

Zur Eingabe der Farbkomponenten ist der Fließkommabereich 0.0 bis 1.0 voreingestellt. Mit der Anweisung [AbsColor, TRUE] wird der Bereich der Argumente auf 0 bis 255, entsprechend der Windows Spezifikation umgeschaltet. [AbsColor, FALSE] schaltet zurück in den Fließkomma Modus.

13.9.4.6 PlotRange

PlotRange bestimmt den Bereich der X und Y Achse.

Syntax: [PlotRange, minX, maxX, minY, maxY]

Für die X und Y Achsen ist ein variabler Bereich voreingestellt der an die Größe der eingegebenen Koordinaten angepaßt wird. Mit PlotRange wird der Bereich der Achsen festgelegt. Als Argument ist jeder gültige positive oder negative Fließkommanwert zulässig.

13.9.4.7 PlotBorder

PlotBorder schaltet die Umrandung der Plotbox ein oder aus.

Syntax: [PlotBorder, {TRUE | FALSE}]

In der Voreinstellung wird die Plotbox mit einer 1 Pixel breiten Umrandung angezeigt. Mit der Anweisung [Plotborder, FALSE] wird die Umrandung ausgeschaltet.

13.9.4.8 BorderColor

BorderColor bestimmt die Farbe der Umrandung.

Syntax: [BorderColor, red, green, blue]

Der gültige Bereich der Werte ist eine Fließkomma Zahl zwischen 0 und 1.0, oder zwischen 0 und 255 wenn **AbsColor** auf **TRUE** gesetzt ist.

13.9.5.1 MoveTo

MoveTo setzt den Zeichenstift (Pen) auf eine bestimmt **X,Y** Position.

Syntax: [MoveTo, X, Y]

Verwenden Sie **MoveTo** bevor Sie mit **LineTo** eine Line zeichnen.

13.9.5.2 LineTo

LineTo zeichnet eine Line ab der aktuellen Zeichenposition.

Syntax: [LineTo, X, Y]

LineTo zeichnet eine Line von der aktuellen Zeichenposition bis zu einer mit **X** und **Y** spezifizierten Position; ausschließlich der **X,Y** Position. Anschließend setzt **LineTo** die aktuelle Zeichenposition auf **X,Y**. Zum Zeichnen werden die **Pen** Einstellungen verwendet.

Hinweis: Wenn per **PenStyle** ein anderes Muster als **psSolid** eingestellt wurde, werden die Lücken der Linie mit den aktuellen Einstellungen von **Brush** gezeichnet.

13.9.5.3 PlotLine

PlotLine zeichnet eine Linie zwischen zwei angegebenen Positionen.

Syntax: [PlotLine, X1, Y1, X2, Y2]

PlotLine zeichnet eine Line von der Position **X1,Y1** bis zu der mit **X2** und **Y2** spezifizierten Position; ausschließlich der Position **X2,Y2**. Anschließend setzt **PlotLine** die aktuelle Zeichenposition auf **X2,Y2**. Zum Zeichnen werden die **Pen** Einstellungen verwendet.

Hinweis: Wenn per **PenStyle** ein anderes Muster als **psSolid** eingestellt wurde, werden die Lücken der Linie mit den aktuellen Einstellungen von **Brush** gezeichnet.

13.9.5.4 PlotArrow

PlotArrow zeichnet eine Line mit einer Pfeilspitze am Kopfende.

Syntax: [PlotArrow, topX, tipY, tailX, tailY]

PlotArrow zeichnet eine Linie, beginnend an der Position **topX** und **topY**, bis zu der Position **tailX** und **tailY**, ausschließlich der Position **tailX**, **tailY**; mit einer Pfeilspitze an **topX**, **topY**. Anschließend setzt **PlotArrow** die aktuelle Zeichenposition auf **tailX**, **tailY**. Zum Zeichnen werden die **Pen** Einstellungen verwendet.

Hinweis: Wenn per **PenStyle** ein anderes Muster als **psSolid** eingestellt wurde, werden die Lücken der Linie mit den aktuellen Einstellungen von **Brush** gezeichnet.

13.9.5.5 PlotDblArrow

PlotDblArrow zeichnet eine Linie mit je einer Pfeilspitze an den Enden.

Syntax: [PlotDblArrow, X1, Y1, X2, Y2]

PlotDblArrow zeichnet eine Linie, beginnend an der Position **X1** und **Y1**, bis zu der Position **X2** und **Y2**, ausschließlich der Position **X2**, **Y2**, mit je einer Pfeilspitze an den Enden. Anschließend setzt **PlotDblArrow** die aktuelle Zeichenposition auf **X2**, **Y2**. Zum Zeichnen werden die **Pen** Einstellungen verwendet.

Hinweis: Wenn per **PenStyle** ein anderes Muster als **psSolid** eingestellt wurde, werden die Lücken der Linie mit den aktuellen Einstellungen von **Brush** gezeichnet.

13.9.5.6 FrameRect

FrameRect zeichnet ein Rechteck.

Syntax: [FrameRect, X1, Y1, X2, Y2]

Verwenden Sie **FrameRect** um einen 1 Pixel breiten Rahmen um eine rechteckige Region zu zeichnen, deren Eckpunkte mit **X1**, **Y1** (obere, linke Ecke) und **X2**, **Y2** (untere, rechte Ecke) spezifiziert sind. **FrameRect** verwendet den mit **BrushColor** eingestellten Farbton. Wenn Sie statt **BrushColor** die Einstellungen von **Pen** verwenden wollen, benutzen Sie das Kommando **PlotRect** oder **PlotPolygon**.

13.9.5.7 FillRect

FillRect füllt eine rechteckige Fläche.

Syntax: [FillRect, X1, Y1, X2, Y2]

FillRect füllt eine rechteckige Fläche, deren Eckpunkte mit **X1**, **Y1** (obere, linke Ecke) und **X2**, **Y2** (untere, rechte Ecke) spezifiziert sind, unter Verwendung der Einstellungen von **Brush**.

13.9.5.8 PlotRect

PlotRect zeichnet ein Rechteck.

Syntax: [PlotRect, X1, Y1, X2, Y2]

Syntax: [PlotRect, X1, Y1, X2, Y2, TRUE]

PlotRect zeichnet einen Rahmen um eine rechteckige Region, deren Eckpunkte mit **X1**, **Y1** (obere, linke Ecke) und **X2**, **Y2** (untere, rechte Ecke) spezifiziert sind. Zum Zeichnen des Rahmens verwendet **PlotRect** die **Pen** Einstellungen. Optional kann das Rechteck ausgefüllt werden. Dazu tragen Sie in das sechste Feld der Zeile den Wert **TRUE** (eins) ein. Zum Ausfüllen des Rechtecks werden die **Brush** Einstellungen verwendet.

Hinweis: Wenn eine rechteckige Fläche ohne Umrandung gezeichnet werden soll, verwenden Sie das Kommando **FillRect**. Ein Rechteck mit abgerundeten Ecken kann mit **RoundRect** gezeichnet werden.

13.9.5.9 RoundRect

RoundRect zeichnet ein Rechteck mit abgerundeten Ecken.

Syntax: [RoundRect, X1, Y1, X2, Y2, X3, Y3]

Syntax: [RoundRect, X1, Y1, X2, Y2, X3, Y3, 1]

RoundRect zeichnet einen Rahmen um eine rechteckige Region, deren Eckpunkte mit **X1, Y1** (obere, linke Ecke) und **X2, Y2** (untere, rechte Ecke) spezifiziert sind. Die Ecken werden abgerundet mit dem Bogen einer Ellipse. **X3** bestimmt die Breite, **Y3** bestimmt die Höhe der Ellipse.

Zum Zeichnen des Rahmens verwendet **RoundRect** die **Pen** Einstellungen. Optional kann das Rechteck ausgefüllt werden. Dazu tragen Sie in das achte Feld der Zeile den Wert **TRUE** (eins) ein. Zum Ausfüllen des Rechtecks werden die **Brush** Einstellungen verwendet.

13.9.5.10 PlotEllipse

PlotEllipse zeichnet eine Ellipse.

Syntax: [PlotEllipse, X1, Y1, X2, Y2]

Syntax: [PlotEllipse, X1, Y1, X2, Y2, 1]

PlotEllipse zeichnet eine Ellipse deren Begrenzung durch ein Rechteck definiert wird. Die Eckpunkte der rechteckigen Region, werden mit **X1, Y1** (obere, linke Ecke) und **X2, Y2** (untere, rechte Ecke) spezifiziert. Wenn die Begrenzung ein Quadrat ist, wird ein Kreis gezeichnet.

Zum Zeichnen des Rahmens verwendet **PlotEllipse** die **Pen** Einstellungen. Optional kann die Ellipse ausgefüllt werden. Dazu tragen Sie in das sechste Feld der Zeile den Wert **TRUE** (eins) ein. Zum Ausfüllen der Ellipse werden die **Brush** Einstellungen verwendet.

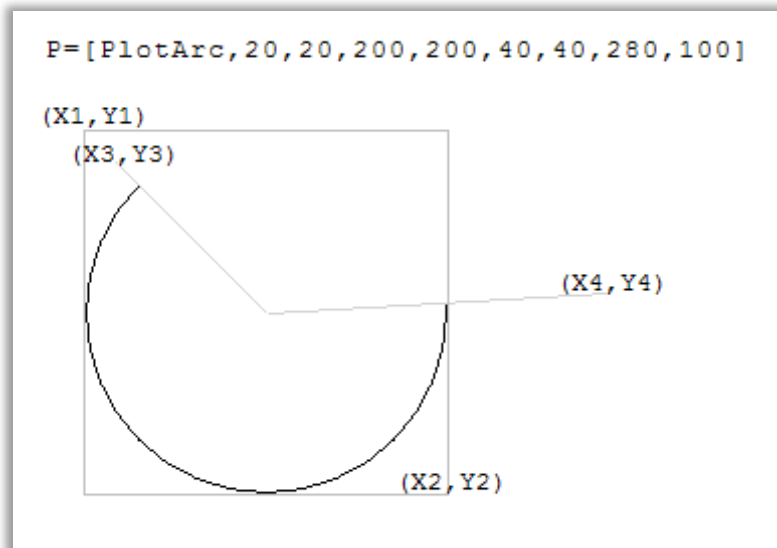
13.9.5.11 PlotArc

PlotArc zeichnet eine Kreisbogen.

Syntax: `[PlotArc, X1, Y1, X2, Y2, X3, Y3, X4, Y4]`

PlotArc zeichnet einen Kreisbogen der ein Teil einer Ellipse ist, deren Begrenzung durch ein Rechteck definiert wird. Die Eckpunkte der rechteckige Region, werden mit **X1, Y1** (obere, linke Ecke) und **X2, Y2** (untere, rechte Ecke) spezifiziert.

Der Kreisbogen wird im Uhrzeigersinn als Ausschnitt der Ellipse vom Ausgangs- zum Endpunkt gezeichnet. Der Ausgangspunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X3, Y3** führt. Der Endpunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X4, Y4** führt. Zum Zeichnen verwendet **PlotArc** die **Pen** Einstellungen.



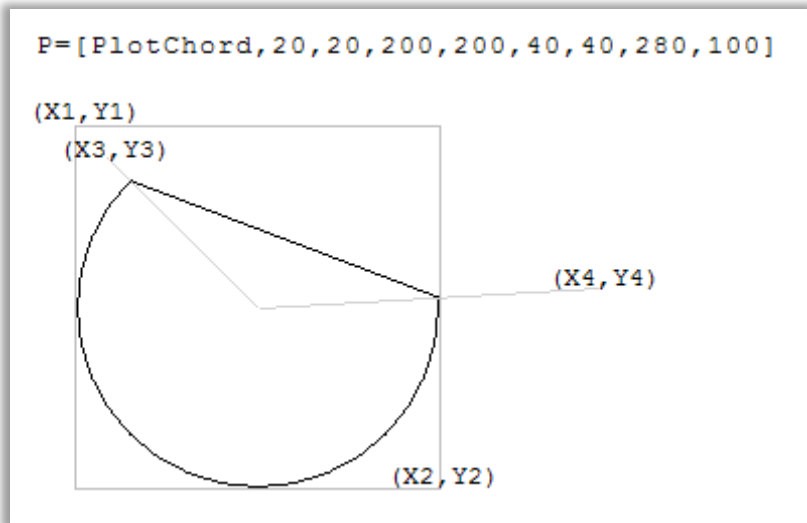
13.9.5.12 PlotChord

PlotChord zeichnet einen geschlossenen Kreisbogen.

Syntax: `[PlotChord, X1, Y1, X2, Y2, X3, Y3, X4, Y4]`

Syntax: `[PlotChord, X1, Y1, X2, Y2, X3, Y3, X4, Y4, 1]`

PlotChord zeichnet einen Kreisbogen dessen Endpunkte mit einer Linie geschlossen sind. Der Kreisbogen ist der Teil einer Ellipse deren Begrenzung durch ein Rechteck definiert wird. Die Eckpunkte der rechteckige Region, werden mit **X1, Y1** (obere, linke Ecke) und **X2, Y2** (untere, rechte Ecke) spezifiziert.



Der Kreisbogen wird im Uhrzeigersinn als Ausschnitt der Ellipse vom Ausgangs- zum Endpunkt gezeichnet. Der Ausgangspunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X3, Y3** führt. Der Endpunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X4, Y4** führt. Zum Zeichnen verwendet **PlotChord** die **Pen** Einstellungen.

Optional kann der Kreisbogen ausgefüllt werden. Dazu tragen Sie in das zehnte Feld der Zeile den Wert **TRUE** (eins) ein. Zum Ausfüllen des Kreisbogens werden die **Brush** Einstellungen verwendet.

13.9.5.13 PlotPie

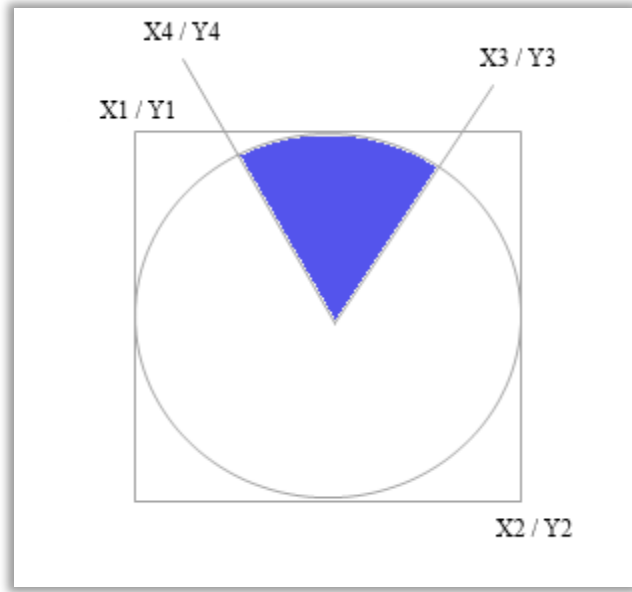
PlotPie zeichnet ein Tortendiagramm.

Syntax: [PlotPie, X1, Y1, X2, Y2, X3, Y3, X4, Y4]

Syntax: [PlotPie, X1, Y1, X2, Y2, X3, Y3, X4, Y4, 1]

PlotPie zeichnet einen tortenförmigen Ausschnitt einer Ellipse. Die Begrenzung der Ellipse ist durch ein Rechteck definiert. Die Eckpunkte des Rechtecks werden mit **X1, Y1** (obere, linke Ecke) und **X2, Y2** (untere, rechte Ecke) spezifiziert.

Der tortenförmige Ausschnitt wird im Uhrzeigersinn als Ausschnitt der Ellipse vom Ausgangs- zum Endpunkt gezeichnet. Der Ausgangspunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X3, Y3** führt. Der Endpunkt ist definiert durch den Schnittpunkt der Ellipse und einer Geraden, die vom Mittelpunkt der Ellipse zur Position **X4, Y4** führt. Zum Zeichnen verwendet **PlotPie** die **Pen** Einstellungen.



Optional kann der Ausschnitt ausgefüllt werden. Dazu tragen Sie in das zehnte Feld der Zeile den Wert **TRUE** (eins) ein. Zum Ausfüllen des Ausschnitts werden die **Brush** Einstellungen verwendet.

13.9.5.14 PolyBezier

PolyBezier zeichnet eine Bezier Kurve.

Syntax: [PolyBezier, X1, Y1, X2, Y2, X3, Y3]

PolyBezier zeichnet eine Bezier Kurve von der aktuellen **Pen** Position zu der Endposition **X3, Y3**. Die Parameter **X1, Y1** und **X2, Y2** bestimmen die Kontrollpunkte.

PolyBezier verwendet die **Pen** Einstellungen.

13.9.5.15 PlotPolygon

PlotPolygon zeichnet ein geschlossenes Vieleck.

Syntax: [PlotPolygon, X1, Y1, X2, Y2, ... Xn, Yn]

PlotPolygon zeichnet ein geschlossenes Vieleck. Die Anzahl der Ecken ist beliebig. Der Rand beginnt an Position **X1, Y1**, verläuft dann über die folgenden Positionen bis **Xn, Yn**. Zum Abschluß wird das Vieleck geschlossen indem **Xn, Yn** mit **X1, Y1** verbunden wird, anschließend wird der Innenraum ausgefüllt. Zum Zeichnen des Rands wird die Einstellung von **Pen** verwendet; der Innenraum wird entsprechend den **Brush** Einstellungen gefüllt. Wenn der Innenraum nicht ausgefüllt werden soll, setzen Sie **BrushStyle** auf **bsClear**.

Lesen Sie auch die Beschreibung unter **KeepPolygon**.

13.9.5.16 KeepPolygon

KeepPolygon erweitert **PlotPolygon**.

Syntax: [KeepPolygon, X1, Y1, X2, Y2, ... Xn, Yn]

Wenn die Zeile des Datenfeldes nicht genug Felder hat um alle Eckpunkte des Polygons aufzunehmen, können Sie weitere Positionen mit **KeepPolygon** anfügen. Alle Positionen einer **PlotPolygon** und den unmittelbar folgenden **KeepPolygon** Anweisung werden zu einem Polygon zusammengefaßt.

13.9.6.1 PenColor

PenColor bestimmt die Farbe des Zeichenstifts (**Pen**), der zum Zeichnen von Linien verwendet wird.

Syntax: [PenColor, red, green, blue]

Der gültige Bereich der Werte ist eine Fließkomma Zahl zwischen 0 und 1.0, oder zwischen 0 und 255 wenn *AbsColor* auf *TRUE* gesetzt ist.

13.9.6.2 PenWidth

PenWidth bestimmt die Breite des Zeichenstifts (Pen) in Pixel.

Syntax: [PenWidth, w]

Beispiel: Next r = [PenWidth, 3]

Verwenden Sie *PenWidth* um die Breite des Zeichenstifts zu vergrößern. Voreingestellt ist eine Breite von einem Pixel.

Hinweis: Die Breite von Pen ist auch abhängig von der *PenStyle* Einstellung.

13.9.6.3 PenStyle

PenStyle bestimmt das Muster, in dem Linien gezeichnet werden.

Syntax: [PenStyle, ps]

Beispiel: Next r = [PenStyle, psSolid]

Verwenden Sie *PenStyle* zum Zeichnen von gepunkteten oder gestrichelten Linien, oder um Rahmen um gefüllte Flächen zu unterdrücken.

Hinweis: Gepunkteten oder gestrichelte Linien können nur gezeichnet werden, wenn *SetWidth* auf 1 gesetzt ist.

Die folgende Liste zeigt die möglichen Werte für den Parameter *ps*. Voreingestellt ist *psSolid*.

psSolid	Zeichnet eine solide Linie.
psDash	Zeichnet ein gestrichelte Linie.
psDot	Zeichnet eine gepunktete Linie
psDashDot	Zeichnet eine Strich – Punkt Linie.
psDashDotDot	Zeichnet eine Strich – Punkt – Punk Line.
psClear	Zeichnet keine Linie (kann verwendet werden um Rahmen um gefüllte Flächen zu unterdrücken)
psInsideFrame	Zeichnet eine solide Linie, kann aber abweichende Farbe verwenden wenn die Breite größer 1 ist.

13.9.6.4 PenMode

PenMode bestimmt die Interaktion der Vordergrund- mit der Hintergrundfarbe der Plotbox.

Syntax: [PenMode, pm]

Beispiel: Next r = [PenMode, pmBlack]

Die folgende Liste zeigt die vordefinierten Konstanten für *PenMode*. Der voreingestellte Wert ist *pmCopy*.

pmBlack	Immer schwarz
pmWhite	Immer weiß
pmNop	Keine Änderung
pmNot	Hintergrundfarbe wird invertiert
pmCopy	Verwendet die Farbe, die mit <i>PenColor</i> spezifiziert ist
pmNotCopy	Inverse of pen color
pmMergePenNot	Combination of pen color and inverse of plot box background
pmMaskPenNot	Combination of colors common to both pen and inverse of plot box background
pmMergeNotPen	Combination of plot box background color and inverse of pen color
pmMaskNotPen	Combination of colors common to both plot box background and inverse of pen color
pmMerge	Combination of pen color and plot box background color
pmNotMerge	Inverse of pmMerge: combination of pen color and plot box background color
pmMask	Combination of colors common to both pen and plot box background

pmNotMask	Inverse of pmMask: combination of colors common to both pen and plot box background
pmXor	Combination of colors in either pen or plot box background, but not both
pmNotXor	Inverse of pmXor: combination of colors in either pen or plot box background, but not both

13.9.7.1 BrushColor

BrushColor bestimmt die Vordergrund Farbe die beim Füllen von Flächen verwendet wird.

Syntax: [BrushColor, red, green, blue]

BrushColor bestimmt die Farbe in ein mit **BrushStyle** gewähltes Muster gezeichnet wird (nicht den Hintergrund, außer **BrushStyle** = **bsSolid**). Der gültige Bereich der Werte ist eine Fließkomma Zahl zwischen 0 und 1.0, oder zwischen 0 und 255 wenn **AbsColor** auf **TRUE** gesetzt ist.

Hinweis: Bei der Einstellung **BrushStyle** = **bsClear** wird das **BrushColor** Kommando ignoriert. Außerdem wird der bei **BrushColor** eingestellte Wert gelöscht, wenn **BrushStyle** auf **bsClear** gesetzt wird.

13.9.7.2 BrushStyle

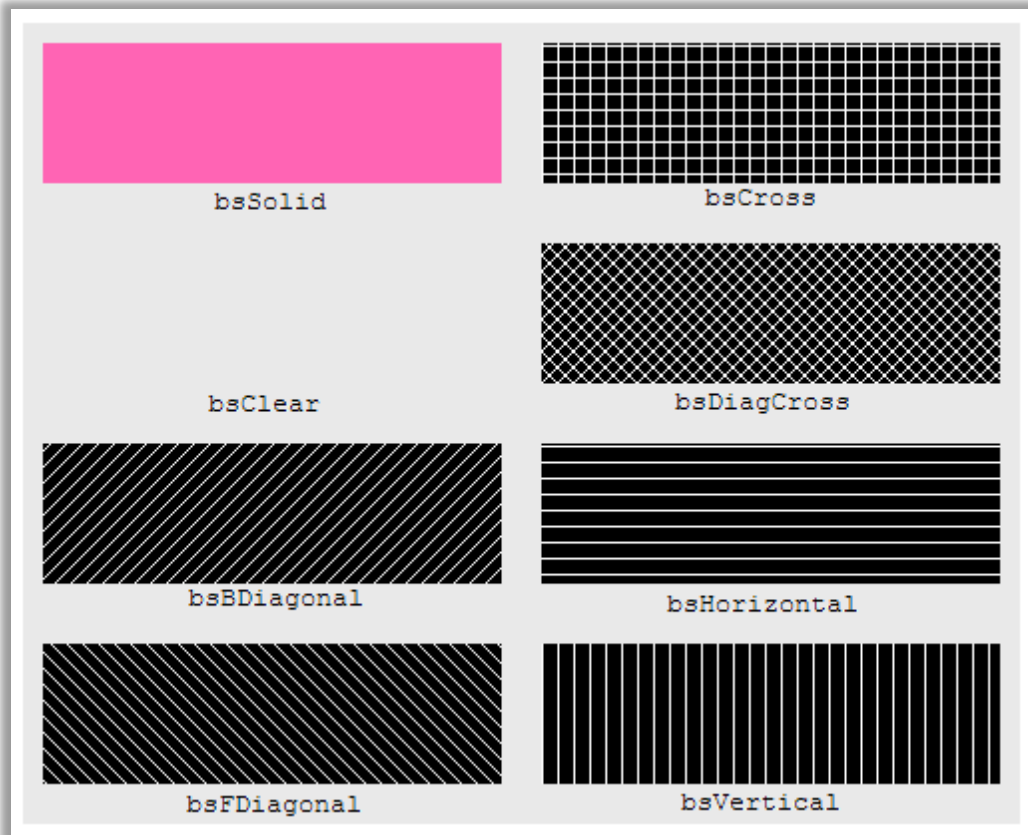
BrushStyle spezifiziert das Muster mit dem eine Fläche gefüllt wird.

Syntax: [BrushStyle, bs]

Beispiel: Next r = [BrushStyle, bsSolid]

Die folgende Liste zeigt die definierten Werte für **BrushStyle**. Voreingestellt ist der Wert **bsSolid**. Das folgende Bild zeigt die den Werten zugeordneten Muster.

bsSolid	bsClear	bsBDiagonal	bsFDiagonal
bsCross	bsDiagCross	bsHorizontal	bsVertical



13.9.8.1 PlotText

PlotText schreibt einen Text an die aktuelle Pen Position.

Syntax: `[PlotText, X, Y, TextString]`

Mit **PlotText** schreiben Sie einen Text zur Plot Box, beginnend an der aktuellen X/Y-Position (Pen Position). Zum Schreiben werden die aktuellen Werte der Font Einstellung verwendet. Nach dem Aufruf der Funktion ist die aktuelle Pen Position die obere / rechte Ecke des Textes.

13.9.8.2 FontSize

Mit *FontSize* wird die Größe der Zeichen bei der Textausgabe eingestellt.

Syntax: `[FontSize, fs]`

13.9.8.3 FontStyle

FontStyle spezifiziert die Attribute der Textausgabe.

Syntax: `[FontStyle, fs]`

Beispiel: `Next r = [FontStyle, (fsBold or fsItalic)]`

Wenn mehr als ein Attribut gesetzt werden soll, können diese mit dem *or* Operator verknüpft werden. Die folgende Liste zeigt die möglichen Werte.

<code>fsBold</code>	Text wird fett gedruckt.
<code>fsItalic</code>	Text wird kursiv
<code>fsUnderline</code>	Text wird unterstrichen.
<code>fsStrikeOut</code>	Text wird durchgestrichen.

13.9.8.4 FontColor

FontColor bestimmt die Farbe in der Text geschrieben wird.

Syntax: `[FontColor, red, green, blue]`

Der gültige Bereich der Werte ist eine Fließkomma Zahl zwischen 0 und 1.0, oder zwischen 0 und 255 wenn *AbsColor* auf *TRUE* gesetzt ist.

13.9.8.5 FontName

FontName bestimmt den Zeichensatz der bei der Textausgabe verwendet wird.

Syntax: [FontName, NameString]

Hinweis: Wenn ein Zeichensatz mit bestimmten Attributen kombiniert wird (z.B. *fsBold* oder *fsItalic*), die auf dem System nicht unterstützt werden, kann vom System ein abweichender Zeichensatz verwendet werden

Neu ab Version 4.34

5.9 Sqr

Sqr liefert als Resultat den Quadratwert des Parameters. *Sqr(x)* ist identisch mit x^2 .

Beispiel: **Sqr** (4) = 16

5.10 Sqrt

Sqrt liefert als Resultat die Quadratwurzel des Parameters. Auf dem Arbeitsblatt kann statt *Sqrt(x)* auch das Wurzelsymbol verwendet werden.

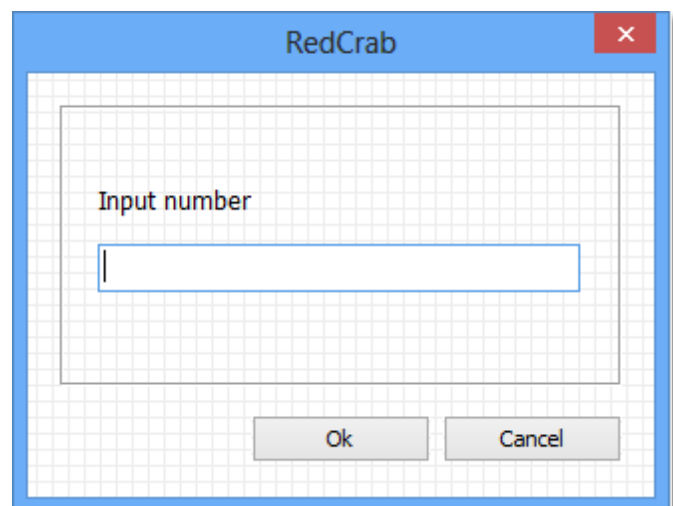
Beispiel: **Sqrt** (4) = 2

Neu ab Version 4.34 (Programers Manual)

3.1 Input

Das Kommando *input* öffnet ein Fenster zur Daten-Eingabe. Dem *input* Statement folgen zwei, oder drei Parameter:

1. Einen Text-String der den Text enthält, der im Fenster über der Eingabezeile angezeigt wird.



2. Den Namen der Variable, der die Eingabe zugewiesen wird.
3. Optional kann als dritter Parameter in einem Text-String ein Wert vorgegeben werden, der in der Eingabezeile angezeigt wird.

Beispiel 1: `input "Input number", x, "123"`

Beispiel 2: `let a = "Input number"
let b = "123"
Input a, x, b`

3.2 Display

Das Kommando ***display*** öffnet ein Fenster zur Anzeige eines numerischen Resultats. Dem ***display*** Statement folgen zwei Parameter:

1. Der Name der Variable deren Wert angezeigt werden soll.
2. Einen Text-String mit optionaler Formatierungsanweisung.

Beispiel 1: `Let x = 471.2
Display x, "Das Resultat lautet: "`

Anzeige im Fenster: „Das Resultat lautet: 471,2“

Beispiel 2: `Let x = 471.2
Display x, "Das Resultat lautet: #.##"`

Anzeige im Fenster: „Das Resultat lautet: 471,20“

Die Formartierung ist identisch mit der von Result Boxen. Lesen Sie dazu die Beschreibung im User Manual: ***Result Box / Formatierung***.

3.3 Menu

Das Kommando **menu** öffnet ein Fenster zur Anzeige eines Menüs. Dem **menu** Statement folgen eine unterschiedliche Anzahl String Parameter.

Der erste String ist der Titel des Menüs, dann folgt der Name des ersten Menü Item und der Name der Funktion, die von diesem Item aufgerufen wird. Es folgt der Name des zweiten Menü Item und der Name der aufzurufenden Funktion, u.s.w. Alle Parameter werden durch Komma getrennt und in Anführungszeichen geschrieben.

Beispiel: Menu "Present Value",
 "PRESENT VALUE", "PresentVal",
 "PAYMENT", "PresentPayment",
 "PERIODS", "PresentPeriods",
 "RATE", "PresentRate",
 "QUIT", "Quit"

