

# Befehlsübersicht für METAPOST

Uwe Siart

Version 0.98 vom 25. Mai 2025

## 1 Wichtige Strukturen

Dateirumpf

```
beginfig(1);  
...  
endfig;  
end
```

Verwendung von  $\LaTeX$  für Beschriftungen

```
verbatimtex  
\documentclass{class}  
\begin{document}  
etex
```

Das `\end{document}` entfällt.

Variablentypen

```
pair      path      string  
numeric   boolean   transform  
(rgb)color  cmykcolor  pen  
picture
```

Zuweisung eines Wertes

```
a := a + 1;
```

Lineare Bedingungsleichungen

```
a + b = 3; 2a = b + 3;
```

Wiederholung

```
for i = a step b until c:  
...  
endfor  
  
for i = a upto b:  
...  
endfor  
  
for i = a downto b:  
...  
endfor
```

Bedingte Ausführung

```
if boolean:  
...  
elseif boolean:  
...  
else:  
...  
fi
```

Der `elseif`-Teil kann entfallen.

Operatoren

```
pair dotprod pair  
angle pair  
sind numeric  
cosd numeric
```

Eigenschaften der Ausgabe

```
outputtemplate := "template";  
outputformat := "[eps|svg|png]";
```

## 2 Punkte

Zugriff auf Koordinaten

```
xpart pair  
ypart pair
```

Vordeclarierte Variablen

```
z[n]  
x[n] = xpart z[n]  
y[n] = ypart z[n]
```

Mediation

```
numeric[pair,pair]  
whatever[pair,pair]
```

Vordefinierte Punkte

```
up    down    left    right
```

### 3 Pfade

#### Verbinden von Punkten

```
pair--pair    pair---pair
pair..pair    pair...pair
```

#### Richtungsangaben

```
pair{up}..{right}pair
pair{dir 50}..{dir 120}pair
```

#### Bézier-Punkte

```
pair..controls pair..pair
pair..controls pair and pair..pair
```

#### Spannung und Krümmung

```
pair..tension numeric..pair
pair..
    tension numeric and numeric
..pair
pair{curl c}..{curl d}pair
```

Ein tension-Parameter muss mindestens den Wert 0,75 haben. Für Krümmungen sind Werte zwischen 0 und  $\infty$  erlaubt.

#### Umkehrung

```
reverse path
```

#### Bounding-Box

```
path = bbox picture;
pair = llcorner picture;
pair = lrcorner picture;
pair = ulcorner picture;
pair = urcorner picture;
pair = center picture;
```

#### Vordefinierte Pfade

```
fullcircle    unitsquare
halfcircle    quartercircle
```

#### Teilpfade

```
path = path cutbefore path;
path = path cutafter path;
path = subpath (a,b) of path;
```

#### Geschlossene Pfade erzeugen

```
path = pair--pair ... cycle;
path = buildcycle(pathlist);
```

#### Superellipse

```
path = superellipse(pair_e,pair_n,
                    pair_w,pair_s,
                    superness);
```

Der Parameter *superness* darf zwischen 0,5 und 1 liegen. Für  $1/\sqrt{2}$  ergibt sich eine Ellipse.

#### Pfadparameter

```
pair    = direction numeric    of path;
numeric = directiontime pair of path;
pair    = directionpoint pair of path;
pair    = path intersectionpoint path;
pair    = path intersectiontimes path;
pair    = point numeric        of path;
numeric = length                path;
numeric = arclength             path;
numeric = arctime length        of path;
```

### 4 Affine Transformationen

#### Grundbausteine

```
object shifted pair
object rotated numeric
object slanted numeric
object scaled numeric
object xscaled numeric
object yscaled numeric
object zscaled pair
object reflectedabout (pair,pair)
object rotatedaround (pair,numeric)
```

#### Definition aus Grundbausteinen

```
transform = identity transforms
```

#### Definition durch die Bilder dreier Punkte

```
pair1 transformed transform = pair2;
pair3 transformed transform = pair4;
pair5 transformed transform = pair6;
```

### 5 Ausgabe von Pfaden und Bildern

#### Vordefinierte Stifte

```
pencircle    pensquare
```

Mit dem Befehl `pickup pen` wird die Stiftform *pen* für alle folgenden Zeichenbefehle aktiviert. Sie bleibt bis zu einem erneuten `pickup`-Befehl aktiv und kann lokal durch die Anweisung `withpen pen` übersteuert werden.

#### Vordefinierte Farben

```
black    white    red    green    blue
```

#### Vordefinierte *picture*-Variablen

```
currentpicture    nullpicture
```

Jeder Zeichenbefehl fügt ein Objekt zur Variable `currentpicture` hinzu (`addto currentpicture`). In `nullpicture` ist das leere Bild vordefiniert. Durch die Zuweisung `currentpicture := nullpicture` kann das aktuelle Bild gelöscht werden.

Bilder begrenzen (clipping)

```
clip picture to cyclic_path;
```

Zeichenbefehle

```
draw object;
drawdot pair;
undraw object;
fill cyclic_path;
unfill cyclic_path;
filldraw cyclic_path;
unfilldraw cyclic_path;
```

Zeichenoptionen

```
drawcommand withpen pen withcolor color
dashed pattern;
```

Mögliche Werte für `pattern` (Linientypen)

```
evenly
withdots
dashpattern(on dim off dim ...)
```

Linientypen können auch transformiert werden.

Setzen und Löschen von Zeichenoptionen

```
drawoptions(dashed evenly);
drawoptions();
```

Ecken und Liniendenen

```
linecap := rounded;
linecap := butt;
linecap := squared;
```

```
linejoin := rounded;
linejoin := mitered;
linejoin := beveled;
```

Pfeile

```
drawarrow path;
drawblarrow path;
```

Bei großen Strichstärken sollte `ahlength` auf einen höheren Wert gesetzt werden. Die Voreinstellung ist 4 bp.

Beschriftung von Punkten

```
draw thelabel.suffix(picture,pair);
label.suffix(picture,pair);
```

Mögliche Suffixe

```
top    bot    urt    ulft
rt     lft    lrt    llft    leer
```

## 6 Boxen

Laden des Makropaketes

```
input boxes
```

Erzeugung

```
boxit.name(picture);
circleit.name(picture);
```

Ausgabe

```
drawboxed (namelist);
drawunboxed (namelist);
```

Der Ausdruck `namelist` steht für eine kommaseparierte Liste von Boxnamen.

Parameter einer Box<sup>1</sup>

```
name.n  name.e  name.s  name.w
name.ne name.nw name.se name.sw
name.c  name.dx name.dy bpath name
```

Relative Lage aufeinanderfolgender Boxen

```
boxjoin(equations);
boxjoin(a.se=b.sw;a.ne=b.nw);
boxjoin();
```

## 7 Makrodefinitionen

Makro ohne Argument

```
def name = replacement enddef
```

Makro mit Argument

```
def name(expr var_list) =
replacement
enddef
```

In `var_list` stehen die Übergabevariablen durch Kommata getrennt.

Aufruf

```
name(var_list)
```

Gruppierung

```
begingroup statement_list endgroup
```

Lokale Variablen in Makros

```
save variable_list;
```

<sup>1</sup>.ne, .nw, .se und .sw nur bei `boxit`-Boxen.

## 8 Sonderfunktionen

Zufallszahl mit der Verteilung  $\mathcal{N}(0, 1)$

```
numeric = normaldeviate;
```

Zufallszahl mit Gleichverteilung zwischen 0 und  $a$

```
numeric = uniformdeviate a;
```

Vektor auf Länge 1 skalieren

```
pair = unitvector pair;
```

## 9 Nomenklatur

<i>pair</i>	Variable von Typ <code>pair</code>
<i>path</i>	Variable von Typ <code>path</code>
<i>numeric</i>	Variable von Typ <code>numeric</code>
<i>color</i>	Variable von Typ <code>color</code>
<i>picture</i>	Variable von Typ <code>picture</code>
<i>object</i>	Grafikobjekt ( <code>pair</code> , <code>path</code> , <code>picture</code> )

---

Copyright © 2003–2025 by Uwe Siart <uwe@siart.de>

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Diese Befehlsübersicht enthält eine Auswahl häufig verwendeter MetaPost-Befehle zur schnellen Referenz. Sie umfasst *nicht* den gesamten Befehlsvorrat von MetaPost und ersetzt vor allem nicht die Lektüre der ausführlichen MetaPost-Dokumentation. Für weitere Informationen siehe CTAN:/graphics/metapost/base/manual/mpman.pdf und CTAN:/graphics/metapost/base/manual/mpboxes.pdf.