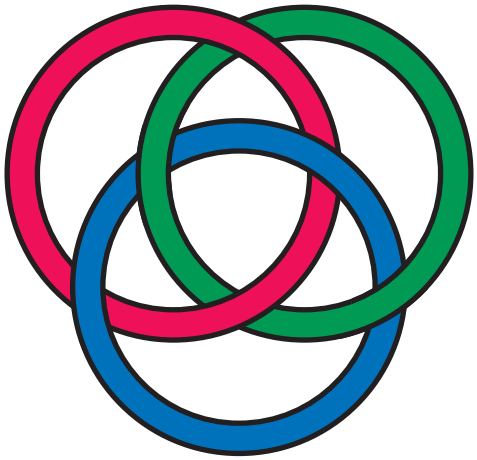
Zeichnen mit LaTeX



28. Oktober 2017

Inhaltsverzeichnis

1 Pakete, Optionen und Bibliotheken

```
% Vorlage und globale Optionen
\documentclass
[
  draft      = true,
  fontsize   = 11pt,
  parskip    = half-,
  BCOR       = 0pt,
  DIV        = 10,
  dvipsnames % vermeidet 'option clash' mit xcolor
]
{scrartcl}

% Standardpakete
\usepackage{fixltx2e}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[ngerman]{babel}
% Zusatzpakete
\usepackage{fp}
\usepackage{graphicx}
\usepackage{ifthen}
\usepackage{tikz}
\usepackage{xcolor}
% TikZ-Bibliotheken (alphabetisch)
\usetikzlibrary{arrows, calc, decorations.pathmorphing,
                decorations.pathreplacing, decorations.shapes,
                decorations.text, intersections, patterns, shapes}
```

2 Umgebungen und Zeichenbefehle

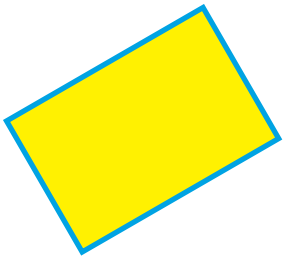
2.1 Die `tikzpicture`-Umgebung

```
% die tikzpicture-Umgebung enthaelt die Zeichenbefehle
\begin{tikzpicture}
  % hier wird der Rand eines Rechtecks gezeichnet:
  \draw[line width=1pt] (0, 0) rectangle (3, 2);
\end{tikzpicture}
```

```
% die tikzpicture-Umgebung enthaelt die Zeichenbefehle
\begin{tikzpicture}
  % hier wird die Flaeche eines Rechtecks ausgefuellt:
  \fill[fill=LimeGreen] (0, 0) rectangle (3, 2);
\end{tikzpicture}
```

2.2 Die scope-Umgebung

```
% die tikzpicture-Umgebung enthaelt die Zeichenbefehle
\begin{tikzpicture}
  % die scope-Umgebung kann z.B. fuer Transformationen
  % genutzt werden
  \begin{scope}[rotate=30]
    % hier wird gleichzeitig gezeichnet und gefuellt:
    \filldraw[fill=Yellow, draw=Cerulean, line width=2pt]
             (0, 0) rectangle (3, 2);
  \end{scope}
\end{tikzpicture}
```



2.3 Das tikz-Makro

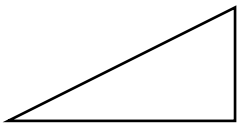
```
% wenn die Zeichnung nur aus einem einzigen Pfad besteht,
% kann man sie mit \tikz auch direkt in den Text einfuegen
direkt \tikz \fill (0, 0) rectangle (1em, 1ex); im Text
```

direkt ■ im Text

3 Koordinaten

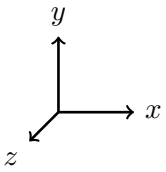
3.1 Zweidimensionale kartesische Koordinaten

```
\begin{tikzpicture}
  % Kartesische Koordinaten in der gewohnten Form: (x, y)
  \draw (0, 0) -- (3, 0) -- (3, 1.5) -- cycle;
\end{tikzpicture}
```



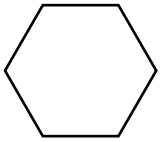
3.2 Dreidimensionale kartesische Koordinaten

```
\begin{tikzpicture}
  % Kartesische Koordinaten in der Form: (x, y, z)
  \draw[->] (0, 0, 0) -- (1, 0, 0) node[right]      {$x$};
  \draw[->] (0, 0, 0) -- (0, 1, 0) node[above]      {$y$};
  \draw[->] (0, 0, 0) -- (0, 0, 1) node[below left]{$z$};
\end{tikzpicture}
```



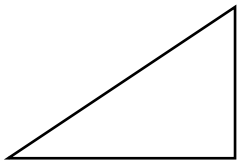
3.3 Polarkoordinaten

```
\begin{tikzpicture}
  % Polarkoordinaten in der Form: (Winkel:Radius)
  \draw   (0:1) --  (60:1) -- (120:1) --
        (180:1) -- (240:1) -- (300:1) -- cycle;
\end{tikzpicture}
```



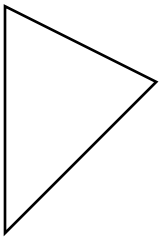
3.4 Benannte Punkte

```
\begin{tikzpicture}
  % mit '\coordinate' koennen Punkte benannt werden
  \coordinate (A) at (0, 0);
  \coordinate (B) at (3, 0);
  \coordinate (C) at (3, 2);
  % die Namen ersetzen dann die Koordinaten
  \draw (A) -- (B) -- (C) -- cycle;
\end{tikzpicture}
```

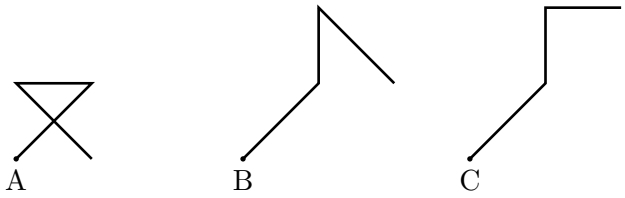


3.5 Verschiebungen

```
\begin{tikzpicture}
  % mit '\coordinate' koennen Punkte benannt werden
  \coordinate (A) at (1, 2);
  % Verschiebungen koennen mit kartesischen und mit
  % Polarkoordinaten definiert werden
  \coordinate (B) at ([shift={(2, -1)}]A);
  \coordinate (C) at ([shift={(270:3)}]A);
  % die Namen ersetzen dann die Koordinaten
  \draw (A) -- (B) -- (C) -- cycle;
\end{tikzpicture}
```



3.6 Absolute und relative Koordinaten

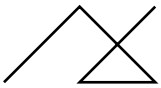



```
\begin{tikzpicture}
  \fill (0, 0) circle[radius=1pt] node[below]{A};
  % absolute Koordinaten
  \draw (0, 0) -- (1, 1) -- (0, 1) -- (1, 0);
  \begin{scope}[xshift=3cm]
    \fill (0, 0) circle[radius=1pt] node[below]{B};
    % relative Koordinaten ohne Verschiebung des Bezugspunktes
    \draw (0, 0) -- (1, 1) -- +(0, 1) -- +(1, 0);
  \end{scope}
  \begin{scope}[xshift=6cm]
    \fill (0, 0) circle[radius=1pt] node[below]{C};
    % relative Koordinaten mit Verschiebung des Bezugspunktes
    \draw (0, 0) -- (1, 1) -- ++(0, 1) -- ++(1, 0);
  \end{scope}
\end{tikzpicture}
```

4 Pfade

4.1 Polygonzüge

```
\begin{tikzpicture}
  % gradlinige Verbindung der gegebenen Koordinaten
  \draw (0, 0) -- (1, 1) -- (2, 0) -- (1, 0) -- (2, 1);
\end{tikzpicture}
```

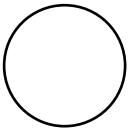


```
\begin{tikzpicture}
  % gradlinige Verbindung der gegebenen Koordinaten, aber
  % parallel zu den Koordinatenachsen:
  % erst vertikal dann horizontal
  \draw (0, 0) |- (3, 2);
\end{tikzpicture}
```

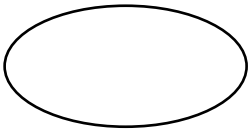
```
\begin{tikzpicture}
  % gradlinige Verbindung der gegebenen Koordinaten, aber
  % parallel zu den Koordinatenachsen:
  % erst horizontal dann vertikal
  \draw (0, 0) -| (3, 2);
\end{tikzpicture}
```

4.2 Geschlossene Pfade

```
\begin{tikzpicture}
  % Kreis mit Radius 8mm um den Mittelpunkt (0, 0)
  \draw (0, 0) circle[radius=8mm];
\end{tikzpicture}
```

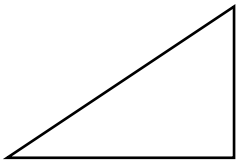


```
\begin{tikzpicture}
  % Ellipse mit dem Mittelpunkt (0, 0)
  % grosse Halbachse: 16mm (in x-Richtung)
  % kleine Halbachse:  8mm (in y-Richtung)
  \draw (0, 0) circle[x radius=16mm, y radius=8mm];
\end{tikzpicture}
```



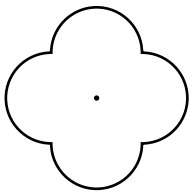
```
\begin{tikzpicture}
  % zwei Eckpunkte definieren ein Rechteck
  % links unten: (0, 0)
  % rechts oben: (3, 2)
  \draw (0, 0) rectangle (3, 2);
\end{tikzpicture}
```

```
\begin{tikzpicture}
  % cycle erzeugt einen geschlossenen Pfad, indem
  % eine Verbindung mit dem Anfang hergestellt wird
  \draw (0, 0) -| (3, 2) -- cycle;
\end{tikzpicture}
```

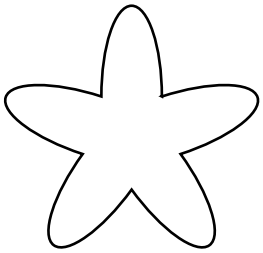


4.3 Kreisbögen

```
\begin{tikzpicture}
  \fill (0, 0) circle[radius=1pt];
  % vier aneinadergesetzte Halbkreise
  \draw (6mm, 6mm)
        arc[start angle=0,   end angle=180, radius=6mm]
        arc[start angle=90,  end angle=270, radius=6mm]
        arc[start angle=180, end angle=360, radius=6mm]
        arc[start angle=270, end angle=450, radius=6mm]
        -- cycle;
\end{tikzpicture}
```

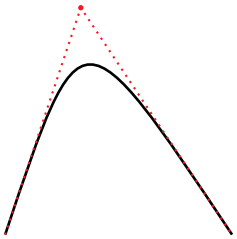


```
\begin{tikzpicture}
  % eine Liste von Optionen
  \tikzstyle{myopts}=[start angle=0,   x radius=4mm,
                      delta angle=180, y radius=12mm];
  % fuenf identische aneinadergesetzte Ellipsen
  \draw (0, 0) arc[myopts] [rotate=72]
               arc[myopts] [rotate=72]
               arc[myopts] [rotate=72]
               arc[myopts] [rotate=72]
               arc[myopts] -- cycle;
\end{tikzpicture}
```

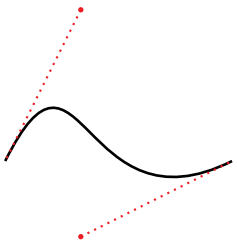


4.4 Bézierkurven

```
\begin{tikzpicture}[line width=1pt]
  % Bezierkurve mit einem Kontrollpunkt
  \draw (0, 0) .. controls (1, 3) .. (3, 0);
  % Hilfslinien
  \draw[line width=0.8pt, draw=Red, style=dotted]
       (0, 0) -- (1, 3) -- (3, 0);
  % der Kontrollpunkt
  \fill[fill=Red] (1, 3) circle[radius=1pt];
\end{tikzpicture}
```



```
\begin{tikzpicture}[line width=1pt]
  % Bezierkurve mit zwei Kontrollpunkten
  \draw (0, 0) .. controls (1, 2) and (1, -1) .. (3, 0);
  % Hilfslinien
  \draw[line width=0.8pt, draw=Red, style=dotted]
       (1,  2) -- (0, 0)
       (1, -1) -- (3, 0);
  % Kontrollpunkte
  \fill[fill=Red] (1,  2) circle[radius=1pt];
  \fill[fill=Red] (1, -1) circle[radius=1pt];
\end{tikzpicture}
```

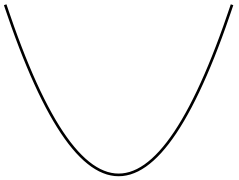


4.5 Verbindungen mit `to[out, in]`

```
\begin{tikzpicture}
  % Verbindungen mit vorgegebener Richtung
  \draw (0, 0) to[out=45,  in=225] (1, 2)
               to[out=270, in=90]  (3, 0);
\end{tikzpicture}
```

4.6 Verbindungen mit `plot`

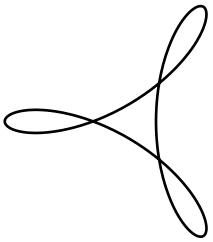
```
% coordinates {...} legt die zu verbindenden Punkte fest
\draw plot[smooth] coordinates
{
  (-1.5, 2.25)  (-1.4, 1.96)  (-1.3, 1.69)  (-1.2, 1.44)
  (-1.1, 1.21)  (-1.0, 1.00)  (-0.9, 0.81)  (-0.8, 0.64)
  ...
  ( 0.5, 0.25)  ( 0.6, 0.36)  ( 0.7, 0.49)  ( 0.8, 0.64)
  ( 0.9, 0.81)  ( 1.0, 1.00)  ( 1.1, 1.21)  ( 1.2, 1.44)
  ( 1.3, 1.69)  ( 1.4, 1.96)  ( 1.5, 2.25)
};
```



```
\begin{tikzpicture}
  % Koordinaten aus einer Datei laden
  \draw plot[smooth] file{spirograph.xy};
\end{tikzpicture}
```

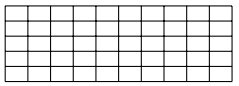
spirograph.xy

```
  0.250  -0.000
  0.250  -0.022
  0.250  -0.044
  ...
```



4.7 Gitternetze

```
\begin{tikzpicture}
  % ein Gitternetz mit verschiedenen Schrittweiten
  \draw (0, 0) grid[xstep=3mm, ystep=2mm] (3, 1);
\end{tikzpicture}
```



4.8 Flächen mit `even odd rule`

```
\begin{tikzpicture}
  % ein (einziger) Pfad, der aus zwei Kreisen besteht
  \filldraw[fill=Cerulean, draw=Black, even odd rule]
            (0, 0) circle[radius=6mm]
            (0, 0) circle[radius=9mm];
\end{tikzpicture}
```

5 Pfadmodifikationen

5.1 line width

```
% ein ziemlich dicker Strich
\draw[line width=5mm] (0, 0) -- (3, 0);
```

5.2 draw

```
% ein dicker, roter Strich
\draw[line width=5mm, draw=RubineRed] (0, 0) -- (3, 0);
```

5.3 line cap

```
% abgerundete Enden
\draw[line width=5mm, line cap=round] (0, 0) -- (3, 0);
```

```
% ueberstehende rechteckige Enden (default)
\draw[line width=5mm, line cap=rect] (0, 0) -- (3, 0);
```

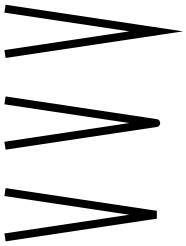
```
% buendige rechteckige Enden
\draw[line width=5mm, line cap=butt] (0, 0) -- (3, 0);
```

5.4 line join

```
% spitze Ecken (default)
\draw[line join=miter] (0, 0) -- (2, 0.3) -- (0, 0.6);
```

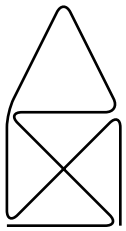
```
% abgerundete Ecken
\draw[line join=round] (0, 0) -- (2, 0.3) -- (0, 0.6);
```

```
% abgeflachte Ecken
\draw[line join=bevel] (0, 0) -- (2, 0.3) -- (0, 0.6);
```



5.5 rounded corners

```
\begin{tikzpicture}
  % abgerundete Ecken
  \draw[rounded corners=2mm] plot coordinates
  {
    (0, 0) (2, 0) (0, 2)
    (2, 2) (1, 4) (0, 2)
    (0, 0) (2, 2) (2, 0)
  };
\end{tikzpicture}
```



5.6 shorten

```
% verkuerzte Strecken
\draw[shorten <=1mm, shorten >=5mm] (0, 0) -- (3, 0);
```


5.7 style

```
% eine gepunktete Linie
\draw[style=dotted] (0, 0) -- (3, 0);
```



```
% gepunktet mit groesseren Abstaenden
\draw[style=loosely dotted] (0, 0) -- (3, 0);
```



```
% gepunktet mit kleineren Abstaenden
\draw[style=densely dotted] (0, 0) -- (3, 0);
```



```
% eine gestrichelte Linie
\draw[style=dashed] (0, 0) -- (3, 0);
```



```
% gestrichelt mit groesseren Abstaenden
\draw[style=loosely dashed] (0, 0) -- (3, 0);
```



```
% gestrichelt mit kleineren Abstaenden
\draw[style=densely dashed] (0, 0) -- (3, 0);
```


5.8 dash pattern

```
% ein eigenes Muster
\draw[dash pattern=on 3pt off 2pt on 1pt off 2pt]
     (0, 0) -- (3, 0);
```


5.9 Pfeilspitzen

```
\draw[->] (0, 0) -- (3, 0);
```



```
\draw[<-] (0, 0) -- (3, 0);
```



```
\draw[<->] (0, 0) -- (3, 0);
```



```
\draw[<<->>] (0, 0) -- (3, 0);
```



```
\draw[|-|] (0, 0) -- (3, 0);
```



```
\draw[|<->|, >=latex] (0, 0) -- (3, 0);
```



```
\draw[<->, >=latex] (0, 0) -- (3, 0);
```

```
\draw[<->, >=stealth] (0, 0) -- (3, 0);
```



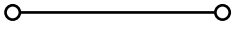
```
\draw[<<->>, >=to reversed] (0, 0) -- (3, 0);
```



```
\draw[<<->>, >=stealth reversed] (0, 0) -- (3, 0);
```

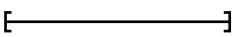


```
% benoetigt \usetikzlibrary{arrows}
\draw[o-o] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{arrows}
\draw[(-)] (0, 0) -- (3, 0);
```

```
% benoetigt \usetikzlibrary{arrows}
\draw[[-{]}] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{arrows}
\draw[<->, >=diamond] (0, 0) -- (3, 0);
```

5.10 Dekorationen

```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=bent] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=bumps] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=coil] (0, 0) -- (3, 0);
```

```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=random steps] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=saw] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=snake] (0, 0) -- (3, 0);
```

```
% benoetigt \usetikzlibrary{decorations.pathmorphing}
\draw[decorate, decoration=zigzag] (0, 0) -- (3, 0);
```

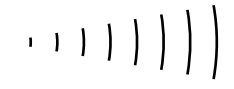
```
% benoetigt \usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration=border] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration=brace] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration={expanding waves, angle=10}]
     (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration=ticks] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration=waves] (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.shapes}
\draw[decorate, decoration=crosses] (0, 0) -- (3, 0);
```



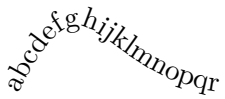
```
% benoetigt \usetikzlibrary{decorations.shapes}
\draw[decorate, decoration=shape backgrounds]
     (0, 0) -- (3, 0);
```



```
% benoetigt \usetikzlibrary{decorations.shapes}
\draw[decorate, decoration=triangles] (0, 0) -- (3, 0);
```



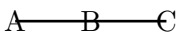
```
% benoetigt \usetikzlibrary{decorations.text}
\draw[decorate, decoration={text along path,
                            text={abcdefghijklmnopqr}}]
     (0, 0) .. controls (1, 2) and (1, 0) .. (3, 0);
```



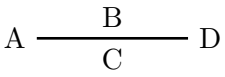
6 Beschriftung

6.1 Text innerhalb eines Pfades

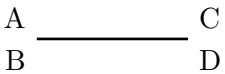
```
\begin{tikzpicture}
  % 'node' fuegt einen Text in einen Pfad ein
  \draw (0, 0) node{A} -- node{B} (2, 0) node{C};
\end{tikzpicture}
```



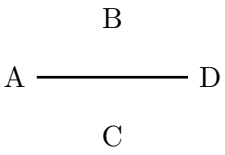

```
\begin{tikzpicture}
  % der Text laesst sich orthogonal verschieben
  \draw (0, 0) node[left] {A}
        --     node[above]{B}
               node[below]{C}
        (2, 0) node[right]{D};
\end{tikzpicture}
```



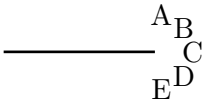

```
\begin{tikzpicture}
  % die Kombination verschiebt diagonal
  \draw (0, 0) node[above left]  {A}
               node[below left]  {B}
        --
        (2, 0) node[above right] {C}
               node[below right] {D};
\end{tikzpicture}
```



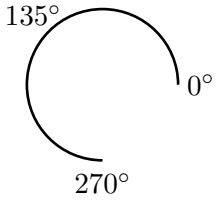
```
\begin{tikzpicture}
  % die Distanz kann auch angegeben werden
  \draw (0, 0) node[left]     {A}
        --     node[above=5mm]{B}
               node[below=5mm]{C}
        (2, 0) node[right]    {D};
\end{tikzpicture}
```




```
\begin{tikzpicture}
  % die Option 'shift' mit Polarkoordinaten
  \draw (0, 0) -- (2, 0) node[shift={( 80:5mm)}] {A}
                         node[shift={( 40:5mm)}] {B}
                         node[shift={(  0:5mm)}] {C}
                         node[shift={(320:5mm)}] {D}
                         node[shift={(280:5mm)}] {E};
\end{tikzpicture}
```

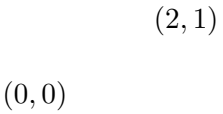



```
\begin{tikzpicture}
  % die Option 'pos' verschiebt die Position der
  % Beschriftung entlang des gegebenen Pfades
  \draw (0, 0) arc[start angle=0, end angle=270, radius=1]
        node[pos=0.0, shift={(  0:3mm)}] {$0^{\circ}$}
        node[pos=0.5, shift={(135:3mm)}] {$135^{\circ}$}
        node[pos=1.0, shift={(270:3mm)}] {$270^{\circ}$};
\end{tikzpicture}
```

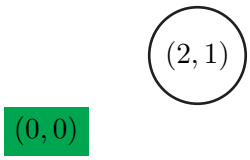


6.2 Text als eigenständiges Objekt

```
\begin{tikzpicture}
  % mit '\node' kann man Text an beliebige Stellen setzen
  \node at (0, 0) {$(0,0)$};
  \node at (2, 1) {$(2,1)$};
\end{tikzpicture}
```




```
\begin{tikzpicture}
  % mit 'shape' kann man 'Knoten' grafisch gestalten
  \node[shape=rectangle, fill=Green] at (0, 0) {$(0,0)$};
  \node[shape=circle,    draw=Black] at (2, 1) {$(2,1)$};
\end{tikzpicture}
```



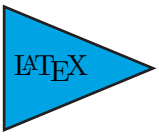

```
\begin{tikzpicture}
  % ellipse benoetigt \usetikzlibrary{shapes}
  \node[shape=ellipse, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```


```
\begin{tikzpicture}
  % trapezium benoetigt \usetikzlibrary{shapes}
  \node[shape=trapezium, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```

```
\begin{tikzpicture}
  % diamond benoetigt \usetikzlibrary{shapes}
  \node[shape=diamond,  fill=Cerulean,
        shape aspect=2, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```

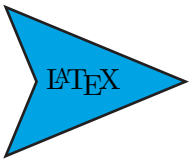

```
\begin{tikzpicture}
  % semicircle benoetigt \usetikzlibrary{shapes}
  \node[shape=semicircle, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```


```
\begin{tikzpicture}
  % isosceles triangle benoetigt \usetikzlibrary{shapes}
  \node[shape=isosceles triangle, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```




```
\begin{tikzpicture}
  % kite benoetigt \usetikzlibrary{shapes}
  % \rotatebox benoetigt \usepackage{graphicx}
  \node[shape=kite, fill=Cerulean, draw=Black]
       at (0, 0) {\rotatebox{90}{\LaTeX}};
\end{tikzpicture}
```

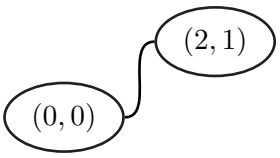

```
\begin{tikzpicture}
  % dart benoetigt \usetikzlibrary{shapes}
  \node[shape=dart, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```




```
\begin{tikzpicture}[line width=1pt]
  % circular sector benoetigt \usetikzlibrary{shapes}
  \node[shape=circular sector, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```


```
\begin{tikzpicture}
  % cylinder benoetigt \usetikzlibrary{shapes}
  \node[shape=cylinder, fill=Cerulean, draw=Black]
       at (0, 0) {\LaTeX};
\end{tikzpicture}
```


```
\begin{tikzpicture}
  % Knoten koennen benannt werden
  \node[shape=ellipse, draw=Black] (A) at (0, 0) {$(0,0)$};
  \node[shape=ellipse, draw=Black] (B) at (2, 1) {$(2,1)$};
  % aus den Namen lassen sich Koordinaten bilden
  \draw (A.0) to[out=0, in=180] (B.180);
\end{tikzpicture}
```

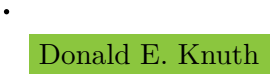


```
% die Abstaende zwischen Text und Rechteck
\node[inner sep=3mm] at (0, 0) {Donald E. Knuth};
```


```
% die horizontalen Abstaende zwischen Text und Rechteck
\node[inner xsep=3mm] at (0, 0) {Donald E. Knuth};
```

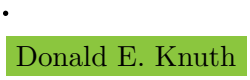

```
% die vertikalen Abstaende zwischen Text und Rechteck
\node[inner ysep=3mm] at (0, 0) {Donald E. Knuth};
```


```
% aeussere Abstaende beeinflussen die 'anchor'-Positionen
\node[outer sep=3mm] (A) at (0, 0) {Donald E. Knuth};
% Referenzpunkt markieren
\fill (A.north west) circle[radius=1pt];
```




```
% aeussere Abstaende beeinflussen die 'anchor'-Positionen
\node[outer xsep=3mm] (A) at (0, 0) {Donald E. Knuth};
% Referenzpunkt markieren
\fill (A.north west) circle[radius=1pt];
```

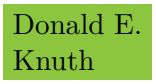

```
% aeussere Abstaende beeinflussen die 'anchor'-Positionen
\node[outer ysep=3mm] (A) at (0, 0) {Donald E. Knuth};
% Referenzpunkt markieren
\fill (A.north west) circle[radius=1pt];
```



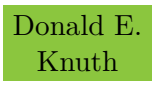

```
% Mindestbreite der 'shape'
\node[minimum width=4cm] at (0, 0) {Donald E. Knuth};
```


```
% Mindesthoehe der 'shape'
\node[minimum height=1cm] at (0, 0) {Donald E. Knuth};
```


```
% Ausrichtung von mehrzeiligem Text
\node[align=left] at (0, 0) {Donald E.\\Knuth};
```

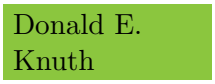



```
% Ausrichtung von mehrzeiligem Text
\node[align=center] at (0, 0) {Donald E.\\Knuth};
```



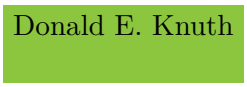

```
% Ausrichtung von mehrzeiligem Text
\node[align=right] at (0, 0) {Donald E.\\Knuth};
```


```
% Breite des Textbereichs
\node[text width=25mm] at (0, 0) {Donald E. Knuth};
```




```
% Hoehe der Textzeile
\node[text height=3ex] at (0, 0) {Donald E. Knuth};
```

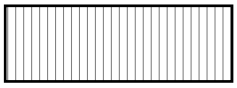

```
% Tiefe der Textzeile
\node[text depth=3ex] at (0, 0) {Donald E. Knuth};
```



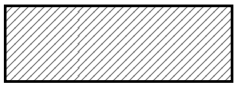
7 Füllmuster

```
\draw[pattern=horizontal lines,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```

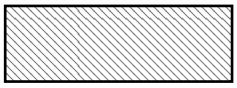
```
\draw[pattern=vertical lines,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



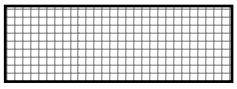
```
\draw[pattern=north east lines,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



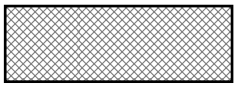
```
\draw[pattern=north west lines,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



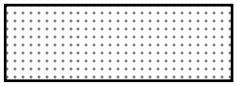
```
\draw[pattern=grid,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



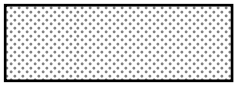
```
\draw[pattern=crosshatch,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



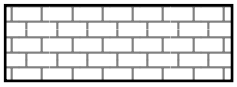
```
\draw[pattern=dots,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



```
\draw[pattern=crosshatch dots,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



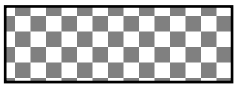
```
\draw[pattern=bricks,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



```
\draw[pattern=fivepointed stars,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```

```
\draw[pattern=sixpointed stars,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```

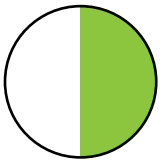
```
\draw[pattern=checkerboard,
      pattern color=Black!50!White]
      (0, 0) rectangle (3, 1);
```



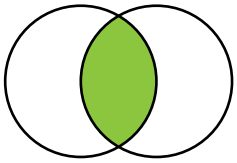
8 Clipping

8.1 Clipping mit einfachen Pfaden

```
\begin{tikzpicture}
  \begin{scope}
    % Rechteck ueber der rechten Haelfte des Kreises
    \clip (0, -1) rectangle (1, 1);
    % den Teil des Kreises ausfuellen, der im
    % Clipping-Bereich liegt
    \fill[fill=LimeGreen] (0, 0) circle[radius=1];
  \end{scope}
  % den Rand des Kreises zeichnen
  \draw (0, 0) circle[radius=1];
\end{tikzpicture}
```

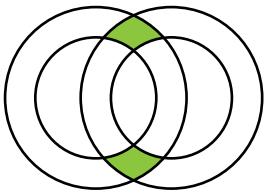


```
\begin{tikzpicture}
  \begin{scope}
    % Durchschnitt zweier Clipping-Bereiche
    \clip (-5mm, 0) circle[radius=1cm];
    \clip ( 5mm, 0) circle[radius=1cm];
    % Flaeche im Clipping-Bereich ausfuellen
    \fill[fill=LimeGreen] (0, 0) circle[radius=1cm];
  \end{scope}
  % Kreisraender zeichnen
  \draw (-5mm, 0) circle[radius=1cm];
  \draw ( 5mm, 0) circle[radius=1cm];
\end{tikzpicture}
```



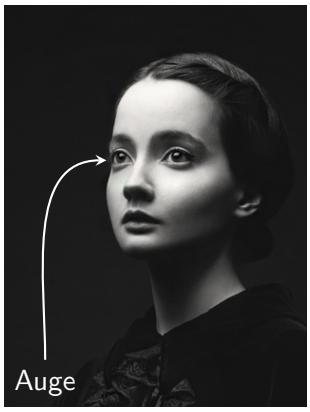
8.2 Clipping mit `even odd rule`

```
\begin{tikzpicture}
  % die Optionen 'even odd rule' kann dem clip-Befehl
  % nicht direkt uebergeben werden
  \begin{scope}[even odd rule]
    % linker Kreisring als Clipping-Bereich
    \clip (-5mm, 0) circle[radius=8mm]
          (-5mm, 0) circle[radius=12mm];
    % rechten Kreisring ausfuellen
    \fill[fill=LimeGreen]
         (5mm, 0) circle[radius=8mm]
         (5mm, 0) circle[radius=12mm];
  \end{scope}
  % Raender der Kreisringe zeichnen
  \draw (-5mm, 0) circle[radius=8mm]
        (-5mm, 0) circle[radius=12mm];
  \draw ( 5mm, 0) circle[radius=8mm]
        ( 5mm, 0) circle[radius=12mm];
\end{tikzpicture}
```



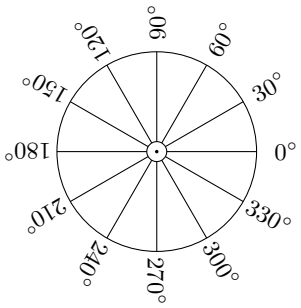
9 Rastergrafik

```
\begin{tikzpicture}
  % die Datei 'rastergrafik.png' laden und unter dem Namen
  % 'einbild' fuer spaetere Verwendung verfuegbar machen
  \pgfdeclareimage[interpolate=true, width=4cm]
                  {einbild}{rastergrafik.png};
  % Bild mit der unteren linken Ecke an der Position
  % (0, 0) einfuegen
  \pgftext[bottom, left, at=\pgfpoint{0cm}{0cm}]
          {\pgfuseimage{einbild}};
  % Text einfuegen
  \node[above right, text=White, font=\sffamily]
       (A) at (0, 0) {Auge};
  % weissen Pfeil zeichnen
  \draw[line width=0.8pt, ->, >=stealth, White]
       (A) to[out=90, in=180] (1.35, 3.25);
\end{tikzpicture}
```

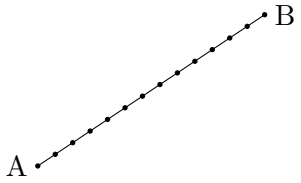


10 Schleifen

```
\begin{tikzpicture}
  \fill (0, 0) circle (1pt);
  \draw (0, 0) circle (2mm);
  \draw (0, 0) circle (2cm);
  % alle Werte von 0 bis 330 in 30er Schritten
  \foreach \x in {0,30,...,330}
  {
    \draw (\x:2mm) -- (\x:2cm);
    \node[rotate=\x] at (\x:26mm)
         {{\footnotesize$\x^{\circ}$}};
  }
\end{tikzpicture}
```



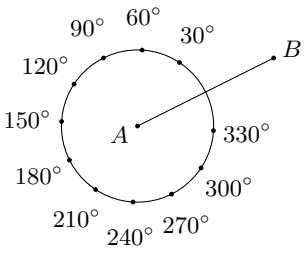

```
\begin{tikzpicture}
  \coordinate (A) at (0, 0);
  \coordinate (B) at (3, 2);
  % Punkte verbinden und bezeichnen
  \draw (A) node[left]{A} -- (B) node[right]{B};
  % die Schleifenvariable durchlaeuft nur ganzzahlige Werte
  \foreach \i in {0,...,13}
  {
    % rationale Werte sollten aus der Schleifenvariablen
    % berechnet werden: hier liegt \x im Intervall [0,1]
    \pgfmathsetmacro{\x}{\i/13}
    % die Strecke AB wird mit 14 Punkten in 13 gleich lange
    % Teilstrecken unterteilt
    \fill ($(A)!\x!(B)$) circle[radius=1pt];
  }
\end{tikzpicture}
```



11 Einfache Berechnung von Koordinaten

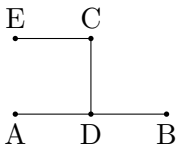
11.1 Der partway und distance modifier

```
\begin{tikzpicture}
  % zwei Punkte definieren
  \coordinate (A) at (1, 1);
  \coordinate (B) at (3, 2);
  % Punkte markieren
  \fill (A) circle[radius=1pt];
  \fill (B) circle[radius=1pt];
  % Punkte beschriften (distance modifier)
  \node at ($(A)!3mm!180:(B)$) {{\footnotesize$A$}};
  \node at ($(B)!3mm!180:(A)$) {{\footnotesize$B$}};
  % Punkte verbinden
  \draw (A) -- (B);
  % Kreis um A zeichnen mit der halben Strecke AB als Radius
  \draw (A) circle[radius={sqrt(5)/2}];
  % alle Winkel aus [30,330] in 30er Schritten
  \foreach \angle in {30,60,...,330}
  {
    % Punkt auf dem Kreis berechnen (partway modifier)
    \coordinate (X) at ($(A)!0.5!\angle:(B)$);
    % Punkt X markieren
    \fill (X) circle[radius=1pt];
    % Punkt X beschriften (distance modifier)
    \node at ($(X)!5mm!180:(A)$)
             {{\footnotesize$\angle^{\circ}$}};
  }
\end{tikzpicture}
```



11.2 Projektionen

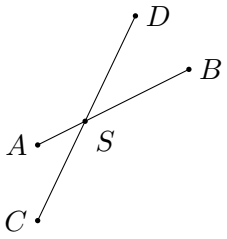
```
\begin{tikzpicture}
  % Punkte definieren
  \coordinate (A) at (1, 1);
  \coordinate (B) at (3, 1);
  \coordinate (C) at (2, 2);
  % Punkt C auf AB projizieren
  \coordinate (D) at ($(A)!(C)!(B)$);
  % AB erst 90 Grad um A drehen und dann C projizieren
  \coordinate (E) at ($(A)!(C)!90:(B)$);
  % Punkte markieren und beschriften
  \fill (A) circle[radius=1pt] node[below] {A}
        (B) circle[radius=1pt] node[below] {B}
        (C) circle[radius=1pt] node[above] {C}
        (D) circle[radius=1pt] node[below] {D}
        (E) circle[radius=1pt] node[above] {E};
  % Punkte verbinden
  \draw (A) -- (B) (C) -- (D) (C) -- (E);
\end{tikzpicture}
```



12 Schnittpunkte bestimmen

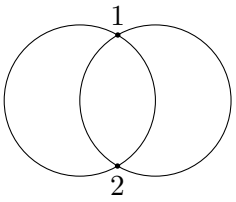
12.1 Schnittpunkt zweier Geraden bestimmen

```
\begin{tikzpicture}
  \coordinate (A) at (0, 0);
  \coordinate (B) at (2, 1);
  \coordinate (C) at ([shift={(270:1cm)}]A);
  \coordinate (D) at ([shift={(135:1cm)}]B);
  % Schnittpunkt berechnen
  \coordinate (S) at (intersection of A--B and C--D);
  % Punkte verbinden
  \draw (A) node[left]{$A$} -- (B) node[right]{$B$}
        (C) node[left]{$C$} -- (D) node[right]{$D$};
  % Punkte markieren
  \fill (A) circle[radius=1pt] (B) circle[radius=1pt]
        (C) circle[radius=1pt] (D) circle[radius=1pt]
        (S) circle[radius=1pt] node[below right] {$S$};
\end{tikzpicture}
```

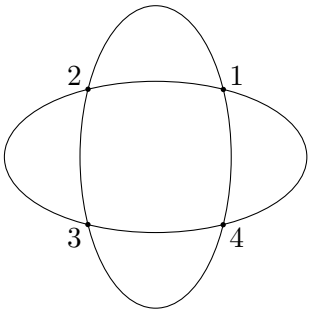


12.2 Schnittpunkte beliebiger Pfade bestimmen

```
\begin{tikzpicture}
  % zuerst muessen den zu schneidenden Pfaden
  % Namen zugewiesen werden
  \draw[name path=kreisA] (0, 0) circle[radius=1];
  \draw[name path=kreisB] (1, 0) circle[radius=1];
  % mit \path findet die Berechnung statt, aber
  % es wird nichts gezeichnet
  \path[name intersections={of=kreisA and kreisB}];
  % 'name intersections' benennt die Schnittpunkte alle
  % nach dem Schema 'intersection-i'
  \fill (intersection-1) circle[radius=1pt] node[above]{1};
  \fill (intersection-2) circle[radius=1pt] node[below]{2};
\end{tikzpicture}
```



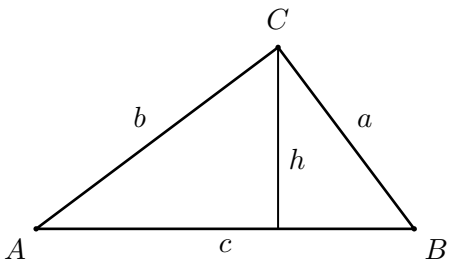

```
\begin{tikzpicture}
  % zwei Ellipsen mit vier Schnittpunkten
  \draw[name path=A] (0, 0) circle[x radius=1, y radius=2];
  \draw[name path=B] (0, 0) circle[x radius=2, y radius=1];
  % 'total' liefert die Anzahl der Schnittpunkte, wobei
  % das Makro \n nur innerhalb des Pfades definiert ist
  \fill[name intersections={of=A and B, total=\n}]
       \foreach \i in {1,...,\n}
       {
         (intersection-\i) circle[radius=1pt]
         (intersection-\i) [scale=1.2] node {\i}
       };
\end{tikzpicture}
```



13 Rechnen mit dem Paket `fp`

```
% Konstanten
\FPe                    % 2.718281828459045235
\FPpi                   % 3.141592653589793238
% Zuweisungen
\FPset    {\x}{2}       % x := 2
\FPset    {\y}{2.5}     % y := 2.5
% unaere Operationen
\FPabs    {\a}{\x}      % a := abs(x)
\FPneg    {\a}{\x}      % a := -x
% binaere Operationen
\FPadd    {\a}{\x}{\y}  % a := x + y
\FPsub    {\a}{\x}{\y}  % a := x - y
\FPmul    {\a}{\x}{\y}  % a := x * y
\FPdiv    {\a}{\x}{\y}  % a := x / y
\FPmin    {\a}{\x}{\y}  % a := min(x,y)
\FPmax    {\a}{\x}{\y}  % a := max(x,y)
% Nachkommastellen
\FPround {\a}{\x}{\y}  % a := x auf y Nachkommastellen gerundet
\FPtrunc {\a}{\x}{\y}  % a := x nach y Nachkommastellen abgeschnitten
\FPclip  {\a}{\x}      % a := x nur mit signifikanten Nachkommastellen
% Potenzen und Wurzeln
\FPpow    {\a}{\x}{\y}  % a := x^y
\FProot   {\a}{\x}{\y}  % a := x^(1/y)
% Trigonometrische Funktionen
\FPsin    {\a}{\x}      % a := sin(x)
\FPcos    {\a}{\x}      % a := cos(x)
\FPtan    {\a}{\x}      % a := tan(x)
\FPcot    {\a}{\x}      % a := cot(x)
\FParcsin{\a}{\x}       % a := arcsin(x)
\FParccos{\a}{\x}       % a := arccos(x)
\FParctan{\a}{\x}       % a := arctan(x)
\FParccot{\a}{\x}       % a := arccot(x)
% Exponential- und Logarithmusfunktion
\FPexp    {\a}{\x}      % a := exp(x)
\FPln     {\a}{\x}      % a := ln(x)
% Fallunterscheidungen
\FPiflt {\x}{\y} ... \else ... \fi   % ist (x < y) ?
\FPifeq {\x}{\y} ... \else ... \fi   % ist (x = y) ?
\FPifgt {\x}{\y} ... \else ... \fi   % ist (x > y) ?
\FPifint{\x}     ... \else ... \fi   % ist x eine ganze Zahl?
```

13.1 Beispiel: Höhen- und Kathetensatz




```
\begin{tikzpicture}
  % Seitenlaengen
  \FPset{\a}{3}           % a = 3
  \FPset{\b}{4}           % b = 4
  \FPset{\c}{5}           % c = 5
  % Kathetensatz: p
  \FPmul{\p}{\a}{\a}      % p = a * a
  \FPdiv{\p}{\p}{\c}      % p = p / c
  % Kathetensatz: q
  \FPmul{\q}{\b}{\b}      % q = b * b
  \FPdiv{\q}{\q}{\c}      % q = q / c
  % Hoehensatz: h
  \FPmul{\h}{\q}{\p}      % h = p * q
  \FProot{\h}{\h}{2}      % h = 2-te wurzel aus h
  % Koordinaten
  \coordinate (A) at ( 0,  0);
  \coordinate (B) at (\c,  0);
  \coordinate (C) at (\q, \h);
  % Dreieck zeichnen
  \draw[line width=1pt]
       (A) -- node[below]       {$c$}
       (B) -- node[above right] {$a$}
       (C) -- node[above left]  {$b$}
       (A);
  % Hoehe zeichnen
  \draw[line width=0.75pt]
       (C) -- node[below right] {$h$} (\q, 0);
  % Punkte zeichnen
  \fill (A) circle[radius=1pt] node[below left]  {$A$};
  \fill (B) circle[radius=1pt] node[below right] {$B$};
  \fill (C) circle[radius=1pt] node[above=3pt]   {$C$};
\end{tikzpicture}
```

14 Eigene Makros

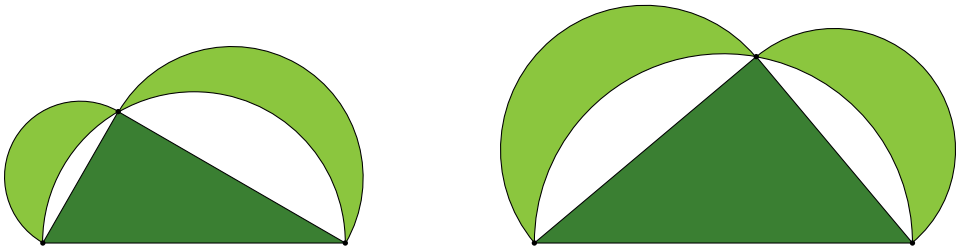
14.1 Abstand zwischen zwei Punkten berechnen

```
% --------
% shapedst
% --------
%
%  \shapedst{A}{B}{\mydistance}
%
\newcommand{\shapedst}[3]
{%
  % define new macro if missing
  \ifthenelse{\isundefined{#3}}{\def#3{\relax}}{\relax}%
  % get x-coordinate from vector
  \pgfextractx{\dimen0}{\pgfpointdiff{\pgfpointanchor{#1}{center}}%
                                     {\pgfpointanchor{#2}{center}}}%
  % get y-coordinate from vector
  \pgfextracty{\dimen1}{\pgfpointdiff{\pgfpointanchor{#1}{center}}%
                                     {\pgfpointanchor{#2}{center}}}%
  % calculate length
  \pgfmathsetmacro{#3}{veclen(\the\dimen0,\the\dimen1)}%
  % add unit 'pt'
  \edef#3{#3pt}%
}
```

14.2 Richtung von einem zu einem anderen Punkt berechnen

```
% --------
% shapedir
% --------
%
% \shapedir{A}{B}{\mydirection}
%
\newcommand{\shapedir}[3]
{%
  % define new macro if missing
  \ifthenelse{\isundefined{#3}}{\def#3{\relax}}{\relax}%
  % get x-coordinate from vector
  \pgfextractx{\dimen0}{\pgfpointdiff{\pgfpointanchor{#1}{center}}%
                                     {\pgfpointanchor{#2}{center}}}%
  % get y-coordinate from vector
  \pgfextracty{\dimen1}{\pgfpointdiff{\pgfpointanchor{#1}{center}}%
                                     {\pgfpointanchor{#2}{center}}}%
  % arctangent of y/x in degrees
  % this also takes into account the quadrants
  \pgfmathsetmacro{#3}{atan2(\the\dimen1,\the\dimen0)}%
}
```

14.3 Beispiel: Die Möndchen des Hippokrates (von Chios)



```
\newcommand{\moendchen}[2]
{
  \begin{scope}
    % Koordinaten der Eckpunkte
    \coordinate (A) at (-#1, 0);
    \coordinate (B) at ( #1, 0);
    \coordinate (C) at ($(0, 0)!#1!#2:(B)$);
    % Abstand und Richtung von B nach C
    \shapedir{B}{C}{\dirBC}
    \shapedst{B}{C}{\dstBC}
    % Abstand und Richtung von C nach A
    \shapedir{C}{A}{\dirCA}
    \shapedst{C}{A}{\dstCA}
    % Dreieck
    \filldraw[fill=OliveGreen] (A) -- (B) -- (C) -- cycle;
    % Moendchen ueber a
    \filldraw[fill=LimeGreen]
             (B) arc[start angle=0, end angle=#2, radius=#1]
                 arc[start angle=\dirBC, delta angle=-180, radius=\dstBC/2];
    % Moendchen ueber b
    \filldraw[fill=LimeGreen]
             (C) arc[start angle=#2, end angle=180, radius=#1]
                 arc[start angle=\dirCA, delta angle=-180, radius=\dstCA/2];
    % Eckpunkte
    \fill (A) circle[radius=1pt];
    \fill (B) circle[radius=1pt];
    \fill (C) circle[radius=1pt];
  \end{scope}
}

\begin{tikzpicture}
  \moendchen{20mm}{120}
  \begin{scope}[xshift=7cm]
    \moendchen{25mm}{80}
  \end{scope}
\end{tikzpicture}
```

14.4 Beschriftung einer Strecke auf der Mittelsenkrechten

```
% ------
% clnode
% ------
%
% \clnode{B}{C}{2mm}{$a$}
%
\newcommand{\clnode}[4]
{
  \node at ($(#1)!0.5!(#2)!#3!270:(#2)$) {#4};
}
```

14.5 Beschriftung eines Winkels auf der Winkelhalbierenden

```
% ------
% canode
% ------
%
% \canode{B}{A}{C}{4mm}{7mm}{$\alpha$}
%
\newcommand{\canode}[6]
{
  \begin{scope}
    \begin{scope}
      % der Kreisbogen
      \clip (#1) -- (#2) -- (#3) -- cycle;
      \draw (#2) circle[radius=#5];
    \end{scope}
    \coordinate (tempnodeA) at ($(#2)!#4!(#1)$);
    \coordinate (tempnodeB) at ($(#2)!#4!(#3)$);
    \coordinate (tempnodeC) at ($(tempnodeA)!0.5!(tempnodeB)$);
    \node at ($(#2)!#4!(tempnodeC)$) {#6};
  \end{scope}
}
```

14.6 Das Paket `geometry`

```
% genaue Kontrolle ueber die Groesse der Seite und die Breite der Raender
\usepackage
[
  paperwidth  = 76.3pt, % Hoehe der Seite
  paperheight = 87.2pt, % Breite der Seite
  top         = 0pt,    % Rand oben
  left        = 0pt,    % Rand links
  right       = 0pt,    % Rand rechts
  bottom      = 0pt     % Rand unten
]
{geometry}
```

14.7 Breite und Höhe einer Zeichnung ermitteln

```
% ------
% sizeof
% ------
%
% \sizeof{%
% \begin{tikzpicture}
%   ...
% \end{tikzpicture}}
%
\newcommand{\sizeof}[1]
{%
  \begingroup
    \setbox0=\hbox{#1}%
    \setlength  {\dimen0}{\wd0}%
    \setlength  {\dimen1}{\ht0}%
    \addtolength{\dimen1}{\dp0}%
    \makebox[3em][r]{$w=$\,}\the\dimen0\\
    \makebox[3em][r]{$h=$\,}\the\dimen1\par
  \endgroup
}
```

15 Maßeinheiten

Alle Koordinaten müssen im Intervall $[-16\,383, 16\,383]$ pt liegen ($\approx \pm 5{,}73$ m).

sp	scaled point	65 536 sp =	1 pt
pt	point	1 pt ≈	0,35 mm
bp	big point	1 bp ≈	1,004 pt
dd	didot point	1 dd ≈	1,07 pt
mm	millimeter	1 mm ≈	2,85 pt
pc	pica	1 pc =	12 pt
cc	cicero	1 cc ≈	12,84 pt
cm	centimeter	1 cm ≈	28,45 pt
in	inch	1 in ≈	72,27 pt
em	Breite eines großen M		
ex	Höhe eines kleinen x		

16 Farben

16.1 Die `dvipsnames` vordefinierter Farben

Apricot	Aquamarine	Bittersweet	Black
Blue	BlueGreen	BlueViolet	BrickRed
Brown	BurntOrange	CadetBlue	CarnationPink
Cerulean	CornflowerBlue	Cyan	Dandelion
DarkOrchid	Emerald	ForestGreen	Fuchsia
Goldenrod	Gray	Green	GreenYellow
JungleGreen	Lavender	LimeGreen	Magenta
Mahogany	Maroon	Melon	MidnightBlue
Mulberry	NavyBlue	OliveGreen	Orange
OrangeRed	Orchid	Peach	Periwinkle
PineGreen	Plum	ProcessBlue	Purple
RawSienna	Red	RedOrange	RedViolet
Rhodamine	RoyalBlue	RoyalPurple	RubineRed
Salmon	SeaGreen	Sepia	SkyBlue
SpringGreen	Tan	TealBlue	Thistle
Turquoise	Violet	VioletRed	White
WildStrawberry	Yellow	YellowGreen	YellowOrange

16.2 Eigene Farben definieren

```
% Farbe im RGB-System definieren [0, 255]
\definecolor{UniBlau}{RGB}{3, 3, 133}

% Mischung aus 80% LimeGreen und 20% Cyan
\colorlet{LimeGreenCyan}{LimeGreen!80!Cyan}
```

17 Octave

Octave dient zur numerischen Berechnung von Funktionswerten. Sobald man den Octave-Code in *<Dateiname.m>* gespeichert hat, kann man ihn wie folgt ausführen:

```
octave -q <Dateiname.m>
```

17.1 Die Wertepaare für `plot` direkt ausgeben

```
% LaTeX-Zeichenbefehl beginnen
printf("\\draw plot[smooth] coordinates {");
% Anzahl der berechneten Punkte
n = 0;
% Intervall und Schrittweite der x-Werte
for x = -1.5:0.1:1.5
  % Zeile nach 4 Punkten umbrechen und einruecken
  if (mod(n++, 4) == 0)
    printf("\n     ");
  endif
  % Funktionswert berechnen: y = -x^2 + 2x + 1
  y = -x**2 + 2*x + 1;
  % Wertepaar ' (x, y)' ausgeben
  printf(" (%4.1f, %5.2f)", x, y);
end
% LaTeX-Zeichenbefehl beenden
printf(" };\n");
```

17.2 Die Wertepaare für `plot` in eine Datei schreiben

```
% Datei 'f.xy' zum Schreiben oeffnen
FID = fopen("f.xy", "w");
% Intervall und Schrittweite der x-Werte
for x = -5:0.1:5
  % Funktionswert berechnen: y = -x^2 + 2x + 1
  y = -x**2 + 2*x + 1;
  % Wertepaar 'x y' ausgeben
  fprintf(FID, "%6.2f\t%6.2f\n", x, y);
end
% Datei schliessen
fclose(FID);
```