Abstract

The purpose of this document is to give a terse outline of the most useful TikZ macros as a quick reference. At the time of writing it is nowhere near complete, but still may be useful for a quick reference for some basics.

1. Rough idea of how TikZ works

When drawing a picture on a piece of paper, we may place our pen down and write in some places, while we may lift up our pen without drawing to move it to another location to draw there. "pen" while drawing, or they may move to another location on the page without drawing anything. It is with this idea that we write TikZ. TikZ is a language we use to instruct the computer how to move its "pen" around to draw diagrams. A *path* is the basic unit of these instructions. Paths take the following form.

`\path`⟨*specification*⟩`;`

A *specification* of a path is a sequence whose elements are separated by spaces consisting of graphics options and path operations. A *graphics option* is a list of options in brackets, such as `[rounded corners]`, which may contain key-value pairs. Graphics options come in two forms.

graphics option	description	examples
local	applies only to the current path; can be scoped	`[rounded corners]`, `[draw]`, `[fill]`
global	applied to all path operations on the path; cannot be scoped	`[red]`, `[draw]`, `[fill]`, `[pattern]`, `[shade]`, `[clip]`, `[use as bounding box]`

A *path operation* tells TikZ a command that describes how the path is built. Path operations tell TikZ how the path is built. There are several types of path operations.

Coordinate	Description	Library Needed
`(x,y)`	Coordinate specification (Cartesian)	
`(theta:r)`	Coordinate specification (polar)	
`++coordinate`	Coordinate relative to previous position	
`+coordinate`	Coordinate relative to previous position without saving the new position	
`(A-|B)`	(x-coord. of (A), y-coord. of (B))	
`(A|-B)`	(x-coord. of (B), y-coord. of (A))	
`($(A)+{sin(60)}*(B)$)`	Coordinate calculations	`calc`
`($(A)!.25!(B)$)`	0.25 the distance from (A) to (B)	`calc`
`($(A)!3cm!(B)$)`	3 cm from (A) toward (B)	`calc`
`($(A)!1.2!30:(B)$)`	Stretch by 1.2, then rotate by 30°	`calc`
`($(A)!(B)!(C)$)`	Projection of point B onto $\overline{AC}$	`calc`
`($(A)!(B)!30:(C)$)`	Project B onto $\overline{AC}$, then rotate 30°	`calc`

Thickness	Result	Dashing	Result
`ultra thin`		`solid`	
`very thin`		`dotted`	
`thin`		`densely dotted`	
`semithick`		`loosely dotted`	
`thick`		`dashed`	
`very thick`		`densely dashed`	
`ultra thick`		`loosely dashed`	
`line width=5pt`		`dash pattern=on 2pt off 4pt`	

Table 1: Styling path thickness and dashing. The code is `\tikz\draw[`⟨*style*⟩`] (0,0) -- (2,0);`.

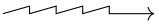
`\path ...` *path operation* `... ;`	Description
`(B)`	set current position to (B)
`-- (B)`	straight line to (B)
`rectangle (B);`	rectangle
`-- (B);`	line
`circle (4);`	circle of radius 4
`ellipse(3 and 2);`	ellipse of width 6 and height 4
`arc(`θ_1`:`θ_2`:`r`) (B);`	circular arc of radius r from angle θ_1 to θ_2
`arc(`θ_1`:`θ_2`:`r_1` and `r_2`)`	elliptical arc
`.. controls (C1) and (C2) .. (B);`	Bézier curve starting with tangent through C1 and entering B with tangent through C2
`grid (B);`	a grid
`parabola (B);`	parabola (several options for bending)
`sin (B);`	sine curve from (0,0) to (π/2,1) stretched from the current position to (B)
`cos (B);`	cosine curve from (0,0) to (π/2,1) stretched from the current position to (B)
`-- cycle;`	return to start and join up nicely

1.1. **Path decorations.** The following examples demonstrate various path decorations available in the `decorations.pathmorphing` library of Ti*k*Z. Each example shows the Ti*k*Z code used to create a decorated path from (0,0) to (2,0), alongside its rendered output.

```
% Standard arrow
\tikz\draw[->] (0,0) -- (2,0);
```


```
% Decoration: zigzag, post length=1mm
\tikz\draw[->,decorate,decoration={zigzag, post
    length=1mm}] (0,0) -- (2,0);
```


```
% Decoration: random steps, amplitude=5mm
\tikz\draw[->,decorate,decoration={random steps,
    amplitude=5mm}] (0,0) -- (2,0);
```


```
% Decoration: saw, amplitude=1mm, post length=3mm
\tikz\draw[->,decorate,decoration={saw, amplitude=1mm
    , post length=3mm}] (0,0) -- (2,0);
```


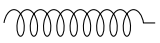
```
% Decoration: bent, amplitude=1mm
\tikz\draw[->,decorate,decoration={bent, amplitude=1
    mm}] (0,0) -- (2,0);
```


```
% Decoration: bumps, amplitude=1mm
\tikz\draw[->,decorate,decoration={bumps, amplitude=1
    mm}] (0,0) -- (2,0);
```


```
% Decoration: snake, amplitude=1mm, post length=1.5mm
\tikz\draw[->,decorate,decoration={snake, amplitude=1
    mm, post length=1.5mm}] (0,0) -- (2,0);
```


```
% Decoration: brace, amplitude=5pt, aspect=0.5
\tikz\draw[decorate,decoration={brace, amplitude=5pt,
     aspect=0.5}] (0,0) -- (2,0);
```


```
% Decoration: triangles
\tikz\draw[decorate,decoration=triangles] (0,0) --
    (2,0);
```


```
% Decoration: coil, segment length=5pt, amplitude=5pt,
     aspect=0.5
\tikz\draw[decorate,decoration={coil, segment length
    =5pt, amplitude=5pt, aspect=0.5}] (0,0) --
    (2,0);
```


```
% Decoration: expanding waves, angle=7, amplitude=5pt
\tikz\draw[decorate,decoration={expanding waves,
    angle=7, amplitude=5pt}] (0,0) -- (2,0);
```


1.2. **Arrow tips.** See here.

```
\tikz\draw[->] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[->>] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[><->>] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[->>>] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[->>>>] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[-{stealth}] (0,0.5) -- (2,0.5);
```

```
\tikz\draw[-{latex}] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[-{to}] (0,0.5) -- (2,0.5);
```


```
\tikz\draw[-{Stealth}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Arc Barb}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Bracket}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Hooks}] (0,0.5) -- (2,0.5);
```

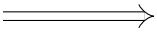

```
% Requires library: arrows.meta
\tikz\draw[-{Parenthesis}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Straight Barb}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Tee Barb}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Classical TikZ Rightarrow}] (0,0.5) --
      (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Computer Modern Rightarrow}] (0,0.5) --
      (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Implies},double,double distance=1mm
      ](0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{To}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Latex[round]}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Rectangle[]}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Square[]}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Stealth[]}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Stealth[round]}] (0,0.5) -- (2,0.5);
```

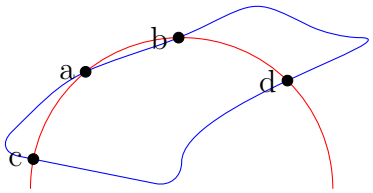

```
% Requires library: arrows.meta
\tikz\draw[-{Triangle[]}] (0,0.5) -- (2,0.5);
```


```
% Requires library: arrows.meta
\tikz\draw[-{Turned Square[]}] (0,0.5) -- (2,0.5);
```

```
% Requires library: arrows.meta
\tikz\draw[-{Circle[open]}] (0,0.5) -- (2,0.5);
```

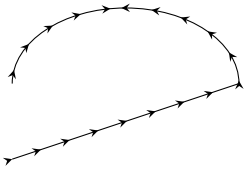


```
\usetikzlibrary{intersections}
\begin{tikzpicture}
    \path[draw,red,name path=c1]
      (0,0) to[out=90,in=180]
      (2,2) to[out=0,in=90] (4,0);
    \path[draw,rounded corners=10pt,
      blue,name path=c2] (-0.5,0.5)
      .. controls +(1,1) .. ++(2.5,1.5)
      .. controls +(1,0.5) .. (4,2)
      to[out=0,in=90] (2,0)--cycle;
    \path[name intersections=
      {of=c1 and c2,by={a,b,c,d}}];
    \foreach \p in {a,b,c,d}
      {\draw [fill=black] (\p)
      circle (2pt) node[left] {\p};}
\end{tikzpicture}
```



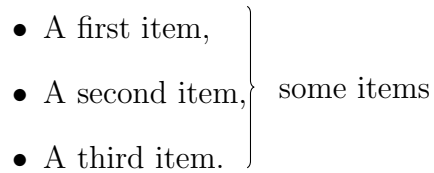

```
\begin{tikzpicture}
    \path[save path=\pathA,name path=A] (0,1) to [bend
          left] (1,0);
    \path[save path=\pathB, name path=B]
       (0,0) .. controls (.33,.1) and (.66,.9) ..
            (1,1);
    \fill[name intersections={of=A and B}] (
          intersection-1) circle (1pt);
    \draw[blue][use path=\pathA];
    \draw[red] [use path=\pathB];
\end{tikzpicture}
```

```
% Requires \usetikzlibrary{decorations.markings}
\begin{tikzpicture}
  \draw[postaction={decorate,
    decoration={
       markings,
       mark=between positions 0 and .75 step 4mm with
              {\arrow{stealth}},
       mark=between positions .75 and 1 step 4mm with
             {\arrowreversed{stealth}}
    }}] (0,0) -- (3,1) arc (0:180:1.5 and 1);
\end{tikzpicture}
```

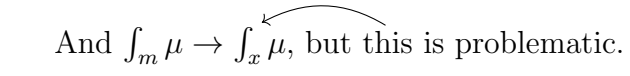


1.3. **TikZmark.** See here.

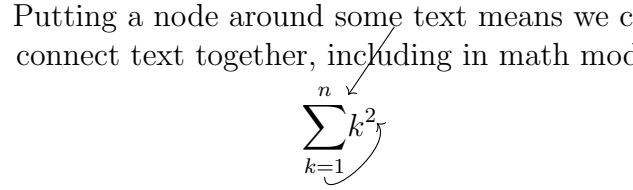
```
\tikzset{tikzmark prefix=ex2-}
\begin{itemize}
    \item A first item,\tikzmark{b}
    \item A second item,\tikzmark{c}
    \item A third item.\tikzmark{d}
\end{itemize}
\begin{tikzpicture}[remember picture,overlay]
\draw[decorate,decoration={brace}] ({pic cs:c} |- {
      pic cs:b})
+(0,1em) -- node[right,inner sep=1em] {some items} ({
      pic cs:c}
|- {pic cs:d});
\end{tikzpicture}
```




```
And $\int_{m}\mu\to\int_{\tikzmark{picname}x}\mu$,
      but this%
\begin{tikzpicture}[remember picture,overlay]
    \draw[->] (-2ex,2ex) to[bend right] ([shift={(1ex
          ,3ex)}]pic cs:picname);
\end{tikzpicture}
is problematic.
```

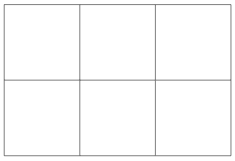



```
Putting a node around \tikzmarknode{txt}{some text}
      means we can
connect text together, including in math mode:
\[
\tikzmarknode{a}{\sum_{k=1}^n} k^{\tikzmarknode{b
      }{2}}
\]
\begin{tikzpicture}[remember picture,overlay]
\draw[->] (txt) -- (a);
\draw[->] (a.south) to[out=-90,in=-45] (b.south east)
      ;
\end{tikzpicture}
```

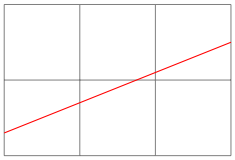


1.4. **From *Unlocking LaTeX Graphics*.** The following examples are from the book Unlocking LaTeX Graphics by Tamara G. Kolda (I encourage you to check it out!)

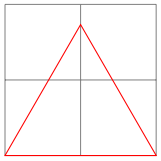
```
% Example 2.1 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid (3,2);
\end{tikzpicture}
```



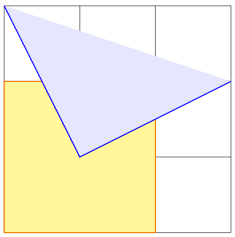
```
% Example 2.2 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid(3,2);
  \draw[red] (0,0.3) -- (3,1.5);
\end{tikzpicture}
```



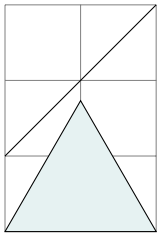
```
% Example 2.3 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid(2,2);
  \draw[red] (0,0) -- (2,0) -- (60:2) -- cycle;
\end{tikzpicture}
```



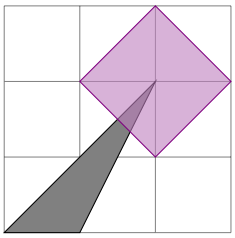
```
% Example 2.4 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid(3,3);
  \draw[orange,fill=yellow!50] (0,2)--(0,0)--(2,0)
        --(2,2)--cycle;
  \draw[blue,fill=.!10] (0,3)--(1,1)--(3,2);
\end{tikzpicture}
```



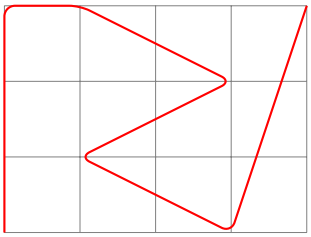
```
% Example 2.5 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid(2,3);
  \draw[fill=teal!10] (0,1) -- (2,3) (0,0) -- (0:2) --
        (60:2) -- cycle;
\end{tikzpicture}
```



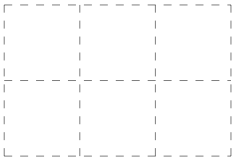
```
% Example 2.6 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid(3,3);
  \draw[fill=.!50] (0,0)--(1,0)--(2,2)--cycle;
  \draw[violet,fill=.!50,fill opacity=0.6] (2,3)
        --(1,2)--(2,1)--(3,2)--cycle;
\end{tikzpicture}
```



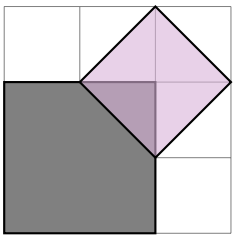
```
% Example 2.7 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \draw[help lines] grid (4,3);
  \draw[thick,rounded corners,red] (0,0) -- (0,3) --
        (1,3) -- (3,2) -- (1,1) -- (3,0) -- (4,3);
\end{tikzpicture}
```



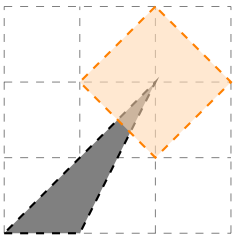
```
% Example 2.8 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    help lines/.append style={dashed}
  ]
  \draw[help lines] grid(3,2);
\end{tikzpicture}
```



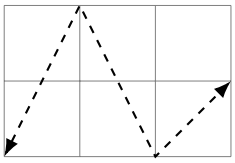
```
% Example 2.9 from Unlocking LaTeX Graphics
\tikzset{
  box/.style={thick,fill=.!50},
  diamond/.style={thick,fill=violet!30, fill opacity
        =.6}
}
\begin{tikzpicture}
  \draw[help lines] grid(3,3);
  \draw[box] (0,2)--(0,0)--(2,0)--(2,2)--cycle;
  \draw[diamond] (2,3)--(1,2)--(2,1)--(3,2)--cycle;
\end{tikzpicture}
```



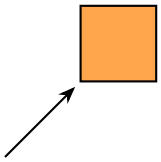
```
% Example 2.10 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    every path/.style={thick,dashed} ]
  \draw[help lines] grid(3,3);
  \draw[fill=.!50] (0,0)--(1,0)--(2,2)--cycle;
  \begin{scope}[every path/.append style={orange,
        fill=.!30, fill opacity=0.6}]
    \draw (2,3)--(1,2)--(2,1)--(3,2)--cycle;
  \end{scope}
\end{tikzpicture}
```



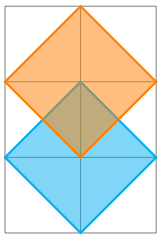
```
% Example 2.11 from Unlocking LaTeX Graphics
\begin{tikzpicture}[>=Latex]
  \draw[help lines] grid (3,2);
  \draw[thick,dashed,<->] (0,0) -- (1,2) -- (2,0) --
        (3,1);
\end{tikzpicture}
```



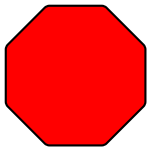
```
% Example 2.12 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \filldraw[fill=orange!70] (1,1) -- (1,2) -- (2,2) --
        (2,1) -- cycle;
  \draw[-Stealth,shorten >=1mm] (0,0) -- (1,1);
\end{tikzpicture}
```



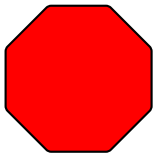
```
% Example 2.13 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[help lines] grid(2,3);
  \filldraw[cyan,fill opacity=0.5] (0,1) -- ++(1,1) --
    ++(1,-1) -- ++(-1,-1) -- cycle;
  \filldraw[orange,fill opacity=0.5] (0,2) -- ++(1,1)
        --
    ++(1,-1) -- ++(-1,-1) -- cycle;
\end{tikzpicture}
```



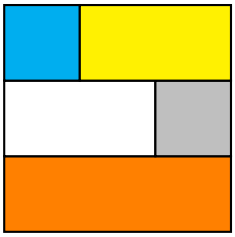
```
% Example 2.14 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[fill=red,rounded corners=1pt] (0,0)
    +(22.5:1) -- +(67.5:1) -- +(112.5:1) --
    +(157.5:1) -- +(202.5:1) -- +(247.5:1) --
    +(292.5:1) -- +(337.5:1) -- cycle;
\end{tikzpicture}
```



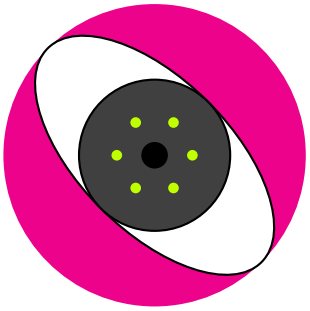
```
% Example 2.15 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[fill=red,rounded corners=1pt] (0,0) --
        +(0:0.8) -- ([turn]45:0.8) -- ([turn]45:0.8)
        -- ([turn]45:0.8) -- ([turn]45:0.8) -- ([turn
        ]45:0.8) -- ([turn]45:0.8) -- cycle;
\end{tikzpicture}
```



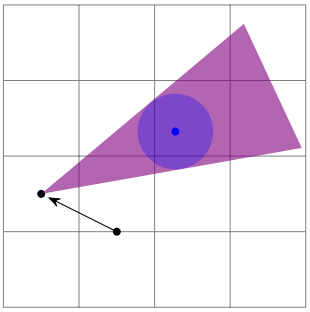
```
% Example 2.16 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[fill=orange] rectangle (3,1);
  \draw[fill=white] (2,1) rectangle +(-2,1);
  \draw[fill=lightgray] (3,2) rectangle (2,1);
  \draw[fill=cyan] (0,2) rectangle ++(1,1);
  \draw[fill=yellow] (3,3) rectangle ++(-2,-1);
\end{tikzpicture}
```



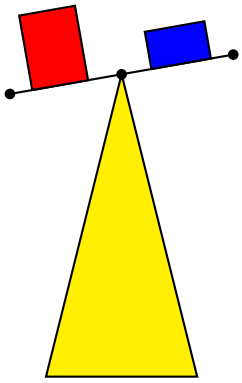
```
% Example 2.17 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \fill[magenta] (2,2) circle[radius=2];
  \draw[fill=white] (2,2) circle[x radius=1, y radius
        =2, rotate=45];
  \draw[fill=darkgray] (2,2) circle(1);
  \fill (2,2) circle(5pt);
  \fill[lime,radius=2pt] (2,2)
    +(0:0.5) circle +(60:0.5) circle +(120:0.5) circle
          +(180:0.5) circle +(240:0.5) circle
         +(300:0.5) circle;
\end{tikzpicture}
```



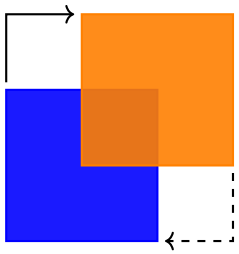
```
% Example 2.18 from Unlocking LaTeX Graphics
\begin{tikzpicture}[ >=Stealth,
    wedge/.style={violet,fill opacity=.60},
    bigcirc/.style={blue,fill opacity=0.3}]
  \draw[help lines] grid(4,4);
  \fill (0.5,1.5) coordinate (A) circle(1.5pt);
  \fill[wedge] ($(A)+(40:3.5)$) -- (A) --
    ($(A)+(10:3.5)$);
  \fill (1.5,1) coordinate (B) circle(1.5pt);
  \draw[->,shorten >=1mm] (B) -- (A);
  \coordinate (C) at (2.275,2.325);
  \fill[bigcirc] (C) circle(0.5cm);
  \fill[blue] (C) circle(1.5pt);
\end{tikzpicture}
```



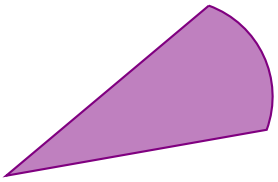
```
% Example 2.19 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[fill=yellow] (0,0) -- (0:2) -- (1,4)
        coordinate (A) -- cycle;
  \fill[radius=2pt] (A) circle
    +(10:-1.5) circle coordinate (B)
    +(10:1.5) circle coordinate (C);
  \draw[black] (B) -- (C) coordinate[pos=0.1] (L)
        coordinate[pos=0.9] (R);
  \draw[fill=red] (L) -- ++(10:0.75) -- ([turn]90:1)
        -- ([turn]90:0.75) -- cycle;
  \draw[fill=blue] (R) -- ++(10:-0.8) -- ([turn
        ]270:0.5) -- ([turn]270:0.8) -- cycle;
\end{tikzpicture}
```



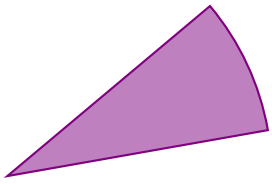
```
% Example 2.20 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick, fill opacity=0.9,
    arw/.style={->,shorten <=1mm, shorten >=1mm}]
  \filldraw[blue] (0,0) coordinate (BSW) rectangle
        ++(2,2) coordinate (BNE);
  \filldraw[orange] (1,1) coordinate (OSW) rectangle
        ++(2,2) coordinate (ONE);
  \draw[arw] (BSW |- BNE) -- (BSW |- ONE) -- (OSW |-
        ONE);
  \draw[arw,dashed] (ONE |- OSW) -- (ONE |- BSW) -- (
        BNE |- BSW);
\end{tikzpicture}
```



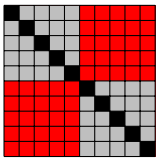
```
% Example 2.21 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \path (0.5,0.5) coordinate (A) +(40:3.5) coordinate
        (B) +(10:3.5) coordinate (C);
  \filldraw[violet, thick, fill=.!50] (B)--(A)--(C) to
        [bend right=45] (B);
\end{tikzpicture}
```



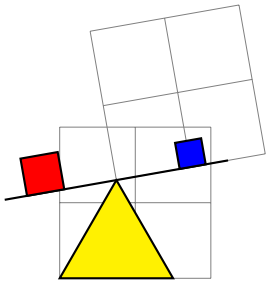
```
% Example 2.22 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \filldraw[violet,thick,fill=.!50] (0.5,0.5) --
        ++(10:3.5) arc(10:40:3.5) -- cycle;
\end{tikzpicture}
```



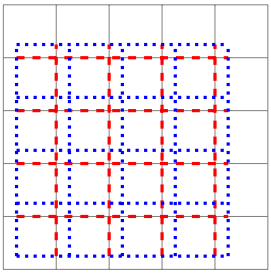
```
% Example 2.23 from Unlocking LaTeX Graphics
\begin{tikzpicture}[scale=0.2]
  \fill[red] (0,0) rectangle (10,10);
  \fill[lightgray] (0,10) rectangle ++(5,-5) rectangle
        ++(5,-5);
  \fill[black] (0,10) rectangle ++(1,-1) rectangle
    ++(1,-1) rectangle ++(1,-1) rectangle ++(1,-1)
    rectangle ++(1,-1) rectangle ++(1,-1) rectangle
    ++(1,-1) rectangle ++(1,-1) rectangle ++(1,-1)
    rectangle ++(1,-1);
  \draw (0,0) grid (10,10);
\end{tikzpicture}
```



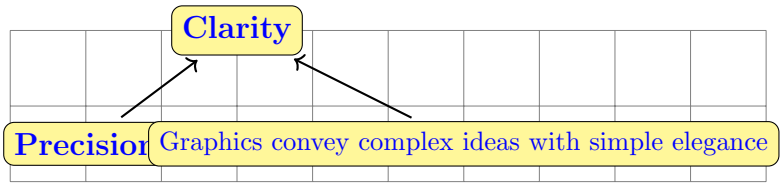
```
% Example 2.24 from Unlocking LaTeX Graphics
\begin{tikzpicture}[thick]
  \draw[help lines] grid(2,2);
  \draw[scale=0.75, fill=yellow] (0,0) -- (0:2) --
        (60:2) coordinate (tip) -- cycle;
  \begin{scope}[shift=(tip),rotate=10]
    \draw[help lines] grid(2,2);
    \draw (-1.5,0) -- coordinate[pos=0.1] (A)
      coordinate[pos=0.9] (B) (1.5,0);
    \draw[fill=red] (A) rectangle ++(0.5,0.5);
    \draw[fill=blue,scale=0.7] (B) rectangle
      ++(-0.5,0.5);
  \end{scope}
\end{tikzpicture}
```



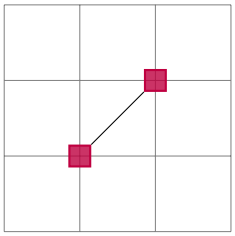
```
% Example 2.25 from Unlocking LaTeX Graphics
\begin{tikzpicture}[very thick,scale=0.7]
  \draw[help lines] grid (5,5);
  \draw[red,dashed] (0.25, 0.25) grid +(4,4);
  \draw[blue,dotted,shift={(0.25,0.25)}]
    grid +(4,4);
\end{tikzpicture}
```



```
% Example 3.1 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    nd/.style={draw,fill=yellow!50,text=blue,font=\
          bfseries,
      rounded corners},
    arw/.style={thick,shorten >=1mm,shorten <=1mm,->}
          ]
  \draw[help lines] grid(10,2);
  \node[nd] (Precision) at (1,0.5) {Precision};
  \node[nd] (Clarity) at (3,2) {Clarity};
  \node[nd,font=\footnotesize] (Graphics) at (6,0.5) {
        Graphics convey complex ideas with simple
        elegance};
  \draw[arw] (Precision) -- (Clarity);
  \draw[arw] (Graphics) -- (Clarity);
\end{tikzpicture}
```




```
% Example 3.2 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.append style= {purple
      ,draw,thick,fill,fill opacity=0.8}]
  \draw[help lines] grid(3,3);
  \node (A) at (1,1) {}; % empty node
  \node (B) at (2,2) {}; % empty node
  \draw (A) -- (B);
\end{tikzpicture}
```



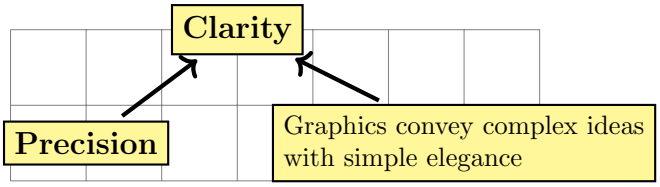
```
% Example 3.3 from Unlocking LaTeX Graphics
\begin{tikzpicture}[nd/.style={draw=gray,very thin},
    ltr/.style={font=\large\bfseries,text=gray}]
  \node[nd] {What a remarkable \\ statement}; % "\\"
        ignored!
  \node[nd,align=left] at (4,0) {What a\\ remarkable\\
        statement};
  \node[nd,text width=4cm] at (8,0) {What a\\
        remarkable\\ statement};
\end{tikzpicture}
```

What a remarkablestatement

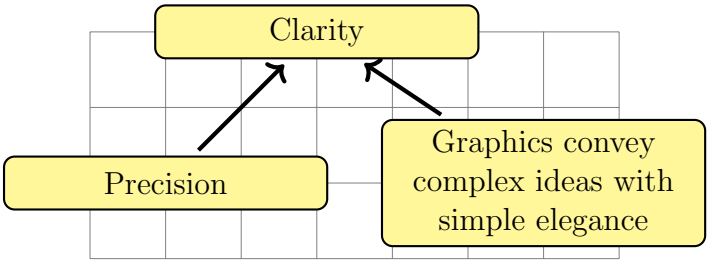
What a remarkable statement

What a remarkable statement

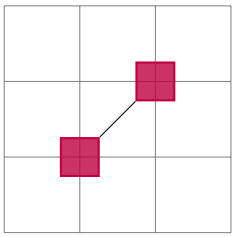
```
% Example 3.4 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    every node/.style={draw,thick,fill=yellow!50,font
          =\bfseries,align=left},
    arw/.style={ultra thick,shorten >=1mm,shorten <=1
          mm,->}]
  \draw[help lines] grid(7,2);
  \node (Precision) at (1,0.5) {Precision};
  \node at (3,2) (Clarity) {Clarity};
  \node[font=\footnotesize] at (6,0.5) (Graphics) {
        Graphics convey complex ideas\\ with simple
        elegance};
  \draw[arw] (Precision) -- (Clarity);
  \draw[arw] (Graphics) -- (Clarity);
\end{tikzpicture}
```



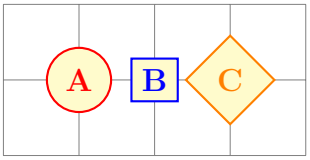

```
% Example 3.5 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    nd/.style={draw,thick,fill=yellow!50,text width=4
          cm,
      align=flush center, minimum height=0.7cm,
            rounded corners},
    arw/.style={ultra thick,shorten >=1mm,shorten <=1
          mm,->}]
  \draw[help lines] grid(7,3);
  \node[nd] (Precision) at (1,1) {Precision};
  \node[nd] at (3,3) (Clarity) {Clarity};
  \node[nd] at (6,1) (Graphics) {Graphics convey
        complex ideas with simple elegance};
  \draw[arw] (Precision) -- (Clarity);
  \draw[arw] (Graphics) -- (Clarity);
\end{tikzpicture}
```



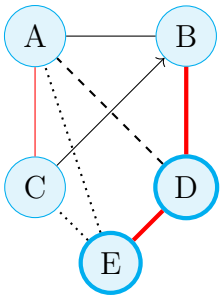

```
% Example 3.6 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.append style=
    {purple, draw, thick, fill, fill opacity=0.8,
      minimum size=0.5cm}]
  \draw[help lines] grid(3,3);
  \path (1,1) node (A) {} (2,2) node (B) {};
  \draw (A) -- (B);
\end{tikzpicture}
```



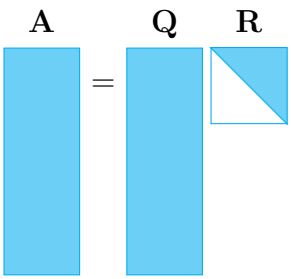
```
% Example 3.7 from Unlocking LaTeX Graphics
\begin{tikzpicture}[nd/.style={draw, thick, fill=
      yellow!25, font=\bfseries}]
  \draw[help lines] grid (4,2);
  \node[red,circle,nd] at (1,1) {A};
  \node[blue,nd] at (2,1) {B};
  \node[orange,diamond,nd] at (3,1) {C};
\end{tikzpicture}
```



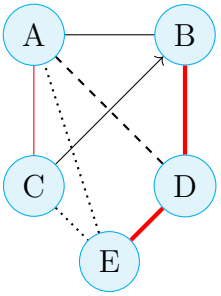

```
% Example 3.8 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={circle, draw=
      cyan, fill=cyan!10}]
  \path (0,3) node (A){A};
  \path (2,3) node (B){B} edge(A);
  \path (0,1) node (C){C} edge[->](B) edge[red](A);
  \path[draw=red,ultra thick] (2,1) node (D){D} edge(B
        ) (1,0) node (E){E} edge(D);
  \path[thick] (E) edge[dotted](C) edge[dotted](A)
      (D) edge[dashed](A);
\end{tikzpicture}
```



```
% Example 3.9 from Unlocking LaTeX Graphics
\begin{tikzpicture}[c/.style={cyan,draw,fill=.!50},
   matrix/.style={c,anchor=north west,
     minimum width=1cm, minimum height=3cm},
   matname/.style={anchor=base,yshift=2mm},]
 \node[matrix] (A) {};
 \path (A.north east) node[anchor=west, yshift=-5mm]
       (eq) {=};
 \path (A.north -| eq.east) node[matrix] (Q) {};
 \path (Q.north east) node[matrix, minimum height=1cm
       , fill=none, xshift=1mm] (R) {};
 \filldraw[c] (R.north west) -- (R.north east) --(R.
       south east) -- cycle;
 \node[matname] at (A.north) {$\mathbf{A}$};
 \node[matname] at (Q.north) {$\mathbf{Q}$};
 \node[matname] at (R.north) {$\mathbf{R}$};
\end{tikzpicture}
```



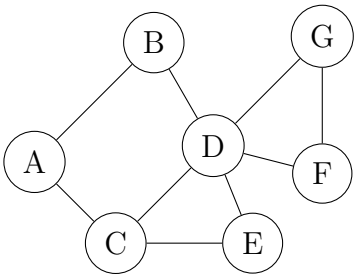

```
% Example 3.10 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={circle, draw=
      cyan, fill=cyan!10}]
 \path (0,3) node (A){A} (2,3) node (B){B} (0,1) node
       (C){C} (2,1) node (D){D} (1,0) node (E){E};
 \draw (A.east) -- (B.west);
 \draw[->] (C.45) -- (B.225);
 \draw[red] (C.north) -- (A.south);
 \draw[thick,dotted] (C.315) -- (E.135)
   (E.110) -- (A.290);
 \draw[ultra thick,red] (D.north) -- (B.south) (E.
       north east) -- (D.south west);
 \draw[thick,dashed] (D.135) -- (A.south east);
\end{tikzpicture}
```




```
% Example 3.11 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw}]
 \node (A) {A};
 \node[above] at (A.north east) (C) {C};
 \node[right] at (A.east) (B) {B};
\end{tikzpicture}
```


```
% Example 3.12 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw}]
 \node (A) {A};
 \node[right=0mm of A] (B) {B};
 \node[above=0mm of A.north east] (C) {C};
\end{tikzpicture}
```


```
% Example 3.13 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
     every node/.style={circle,draw}]
 \node (A) {A};
 \node[above right=of A] (B) {B} edge (A);
 \node[below right=7mm of A] (C) {C} edge (A);
 \node[above right=10mm of C] (D){D} edge(B) edge(C);
 \node[right=of C] (E) {E} edge (C) edge (D);
 \node[above right=5mm of E] (F) {F} edge (D);
 \node[above=of F] (G) {G} edge (D) edge (F);
\end{tikzpicture}
```

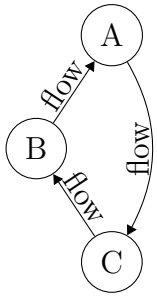



```
% Example 3.14 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw,font=\
      footnotesize}]
 \draw[thick] (0,0) -- ++(8,0) node[pos=0.03,above
       right] {Compression} node[pos=0.97,above left
       ] {Accuracy};
\end{tikzpicture}
```


```
% Example 3.15 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw,font=\
      footnotesize}]
 \draw[thick] (0,0) -- ++(8,0.5) node[pos=0.03,above
       right] {Compression} node[pos=0.97,above left
       ] {Accuracy};
\end{tikzpicture}
```

```
% Example 3.16 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw,font=\
      footnotesize, sloped}]
  \draw[thick] (0,0) -- ++(8,0.5) node[pos=0.03,above
       right] {Compression} node[pos=0.97,above left
       ] {Accuracy};
\end{tikzpicture}
```


```
% Example 3.17 from Unlocking LaTeX Graphics
\begin{tikzpicture}[>=Triangle,
   circ/.style={circle, draw},
   connect/.style={auto, sloped, inner sep=0.1em}]
 \node[circ] at (0,0) (A) {A};
 \node[circ] at (-1,-1.5) (B) {B}
   edge[->] node[connect]{flow} (A) ;
 \node[circ] at (0,-3) (C) {C}
   edge[->] node[connect]{flow} (B);
 \path (A) edge[->,bend left] node[connect]{flow} (C)
       ;
\end{tikzpicture}
```




```
% Example 3.18 from Unlocking LaTeX Graphics
\begin{tikzpicture}[scale=0.75, every node/.style= {
      draw}]
 \draw[rotate=3,thick] (0,0) -- ++(8,0)
   node[pos=0.03,above right,transform shape] {
        Compression}
   node[pos=0.97,above left,sloped] {Accuracy};
\end{tikzpicture}
```

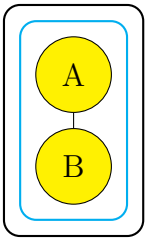

```
% Example 4.1 from Unlocking LaTeX Graphics
Compare the default baseline
\tikz{\filldraw (0,0) -- (0.5,0) circle(1pt) -- (1,0)
      ;}
with a negative shift of -0.5ex
\tikz[baseline=-0.5ex]{\filldraw (0,0) -- (0.5,0)
      circle(1pt) -- (1,0);}.
```

Compare the default baseline —•— with a negative shift of -0.5ex —•—.

```
% Example 4.2 from Unlocking LaTeX Graphics
Compare \tikz{\node[fill=yellow]{no baseline};} to
      setting the baseline to
the \tikz[baseline=(T.base)]{\node[fill=yellow](T){
      node baseline};}.
```

Compare no baseline to setting the baseline to the node baseline.

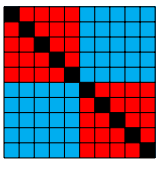
```
% Example 4.3 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw, fill=
      yellow, circle, minimum size=1cm}]
 \path node (A) {A} node[below=2mm of A] (B) {B} edge
       (A);
 \draw[thick, rounded corners=6pt,cyan]
   ($(current bounding box.south west)+(-0.2,-0.2)$)
   rectangle ($(current bounding box.north east)
       +(0.2,0.2)$);
 \draw[thick, rounded corners=6pt]
   ($(current bounding box.south west)+(-0.2,-0.2)$)
   rectangle ($(current bounding box.north east)
       +(0.2,0.2)$);
\end{tikzpicture}
```




```
% Example 4.5 from Unlocking LaTeX Graphics
\begin{tikzpicture}
 \pgfmathsetmacro{\t}{10^(-1)}
 \node[rounded corners,draw] (A) {TikZ math says
       $10^{-1}=\t$};
\end{tikzpicture}
```

TikZ math says $10^{-1} = 0.09999$

```
% Example 4.6 from Unlocking LaTeX Graphics
\begin{tikzpicture}[scale=0.2]
\fill[cyan] (0,0) rectangle (10,10);
\fill[red] (0,10) foreach \i in {1,2} {rectangle
      ++(5,-5)};
\fill (0,10) foreach \i in {1,2,...,10} {rectangle
      ++(1,-1)};
\draw (0,0) grid (10,10);
\end{tikzpicture}
```



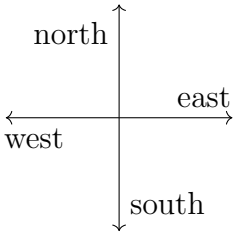
```
% Example 4.7 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \node[minimum height=1.5cm,minimum width=1cm,draw] (
        H) {$H^{\otimes 3}$};
  \foreach \val in {0.1,0.5,0.9} {
    \draw ($(H.north west)!\val!(H.south west)$) --
      ++(-0.4,0) node[left] {$\ket{0}$};
    \draw ($(H.north east)!\val!(H.south east)$) --
      ++(0.4,0) node[right] {$\frac{1}{\sqrt{2}} (\ket
            {0} + \ket {1})$}; }
\end{tikzpicture}
```

$$|0\rangle \;\; |0\rangle \;\; |0\rangle \quad H^{\otimes 3} \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \;\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \;\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

```
% Example 4.8 from Unlocking LaTeX Graphics
\begin{tikzpicture}[scale=0.9,
    every node/.style={anchor=base, transform shape}]
  \draw[help lines] grid (13,1);
  \path foreach[count=\cnt] \ltr in {a,b,...,z} {
    (0.5*\cnt-0.25,0.4) node {\ltr} };
\end{tikzpicture}
```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

```
% Example 4.9 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
    every node/.style={near end, auto}
  ]
  \foreach \dir/\deg in
    {east/0,north/90,west/180,south/270}
  \draw[->] (0,0) -- (\deg:1.5cm) node {\dir};
\end{tikzpicture}
```




```
% Example 4.10 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw,align=
      center}]
  \pgfmathsetmacro{\nw}{width("$2^6=64$")}
  \foreach[count=\i, evaluate=\X as \Xeval using int(\
        X)] \X in {2^0,2^...,2^6} {
    \node[minimum width=\nw+1em] at (1.75*\i,0) {$\X=\
          Xeval$};
  }
\end{tikzpicture}
```

$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$

```
% Example 4.11 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={anchor=base},
      xscale=0.9]
  \draw[help lines] grid (13,1);
  \foreach[count=\cnt,
    evaluate=\cnt as \xx using {mod(\cnt-1,13)+0.5},
    evaluate=\cnt as \yy using {0.4*(1-div(\cnt-1,13))
          +0.2}
  ] \ltr in {a,b,...,z} { \node at (\xx,\yy) {\ltr}; }
\end{tikzpicture}
```

a n	b o	c p	d q	e r	f s	g t	h u	i v	j w	k x	l y	m z

```
% Example 4.12 from Unlocking LaTeX Graphics
\begin{tikzpicture}
  \node[draw,thick,inner sep=1em] (A) {$f(x)=xe^x$};
  \path let \p1=(A.south west),\p2=(A.south east) in
        (\p1) -- (\p2)
  node[text width=0.8*(\x2-\x1), blue, midway, below,
    font=\footnotesize\itshape,align=flush center]
  {Integrate this equation.};
\end{tikzpicture}
```

$$f(x) = xe^x$$

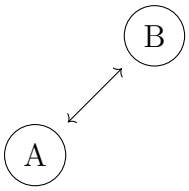
Integrate this equation.

```
% Example 4.13 from Unlocking LaTeX Graphics
\begin{tikzpicture}[every node/.style={draw,align=
      center}]
  \path let \n1={width("$2^6=64$")} in foreach [count
        =\i, evaluate=\X as \Xeval using int(\X)] \X
        in {2^0,2^...,2^6} { (1.75*\i,0) node[minimum
         width=\n1+1em] {$\X=\Xeval$} };
\end{tikzpicture}
```

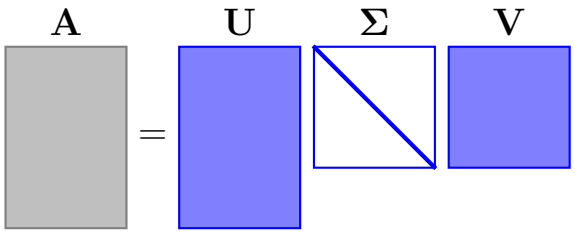
$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$

```
% Example 4.14 from Unlocking LaTeX Graphics
\begin{tikzpicture}[
   topsy/.style={draw,circle,rotate=#1},
   topsy/.default=180
 ]
 \path node[topsy] {A} ++(1,0) node[topsy=60] {B}
   ++(1,0) node[topsy=-45] {C};
\end{tikzpicture}
```


```
% Example 4.15 from Unlocking LaTeX Graphics
\tikzset{shorten/.style={shorten >=#1,shorten <=#1}}
\begin{tikzpicture}[every node/.style={circle,draw}]
 \node (A) {A};
 \node[above right=of A] (B) {B};
 \path[draw,shorten=2mm,<->] (A) -- (B);
\end{tikzpicture}
```




```
% Example 4.16 from Unlocking LaTeX Graphics
\tikzset{label/.style={above,font=\largeabove,font=\
      large},
 matrix/.style n args={2}{transform shape, thick,
       draw, fill=.!50,
   minimum height=#1 cm, minimum width=#2 cm,anchor=
        north west}}
\begin{tikzpicture}[scale=0.8]
 \node[gray,matrix={3}{2}] (A) {};
 \node[right,font=\large] at (A.east) (eq) {=};
 \node[blue,matrix={3}{2}] at (eq.east |- A.north) (U
       ) {};
 \node[blue,matrix={2}{2},fill=none,xshift=2mm]
   at (U.north east) (S) {};
 \draw[blue,ultra thick] (S.north west) -- (S.south
       east);
 \node[blue,matrix={2}{2},xshift=2mm] at (S.north
       east) (V) {};
 \node[label] at (A.north) {$\mathbf{A}$};
 \node[label] at (U.north) {$\mathbf{U}$};
 \node[label] at (S.north) {$\mathbf{\Sigma}$};
 \node[label] at (V.north) {$\mathbf{V}$};
\end{tikzpicture}
```



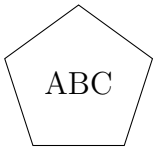

```
\tikz\node[draw, circle] {ABC};
```


```
\tikz\node[draw, rectangle] {ABC};
```

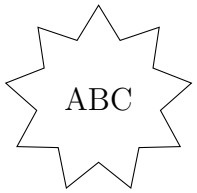

```
% Requires library: shapes.geometric
\tikz\node[draw, diamond] {ABC};
```


```
% Requires library: shapes.geometric
\tikz\node[draw, ellipse] {ABC};
```

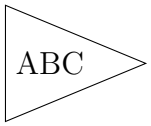

```
% Requires library: shapes.geometric
\tikz\node[draw, regular polygon, regular polygon
      sides=5] {ABC};
```



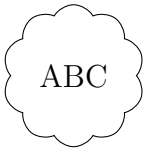
```
% Requires library: shapes.geometric
\tikz\node[draw, star, star points=9] {ABC};
```




```
% Requires library: shapes.geometric
\tikz\node[draw, isosceles triangle] {ABC};
```




```
% Requires library: shapes.symbols
\tikz\node[draw, cloud, cloud puffs=10, cloud puff
        arc=120] {ABC};
```




```
% Requires library: shapes.symbols
\tikz\node[draw, circle, path picture={\draw[thick] (
        path picture bounding box.south west) -- (path
        picture bounding box.north east);}] {ABC};
```


```
% Requires library: shapes.geometric
\tikz\node[draw, cylinder, shape border rotate=90,
        aspect=0.25] {ABC};
```


```
% Requires library: shapes.geometric
\tikz\node[draw, semicircle, shape border rotate=30]
        {ABC};
```


```
% Requires library: shapes.symbols
\tikz\node[draw, starburst, starburst points=30,
        starburst point height=0.2cm] {ABC};
```


```
% Requires library: shapes.symbols
\tikz\node[draw, tape, font=\fontsize{18}{18}\
        sffamily\bfseries] {ABC};
```

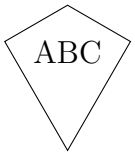

```
% Requires library: shapes.misc
\tikz\node[draw, cross out] {ABC};
```


```
% Requires library: shapes.misc
\tikz\node[draw, strike out] {ABC};
```

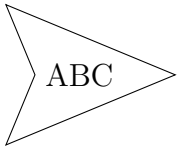

```
% Requires library: shapes.misc
\tikz\node[draw, rounded rectangle, rounded rectangle
         west arc=none] {ABC};
```


```
% Requires library: shapes.misc
\tikz\node[draw, chamfered rectangle] {ABC};
```

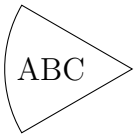
```
1 % Requires library: shapes.geometric
2 \tikz\node[draw, kite] {ABC};
```



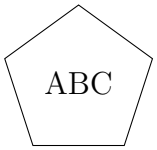

```
1 % Requires library: shapes.geometric
2 \tikz\node[draw, dart] {ABC};
```



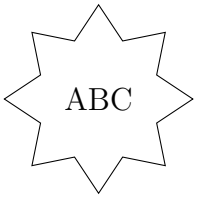

```
1 % Requires library: shapes.geometric
2 \tikz\node[draw, circular sector] {ABC};
```




```
1 % Requires library: shapes.geometric
2 \tikz\node[draw, regular polygon, regular polygon
        sides=5] {ABC};
```

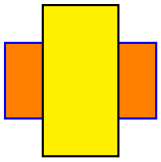



```
1 % Requires library: shapes.geometric
2 \tikz\node[draw, star, star points=8] {ABC};
```



1.6. **Layers.**

```
\pgfdeclarelayer{back}
\pgfdeclarelayer{front}
\pgfsetlayers{back,main,front} % (orders the layers)
\begin{tikzpicture}
    \draw[thick,fill=yellow] (0,0) rectangle (1,2);
    \begin{pgfonlayer}{back}
        \draw[thick,blue,fill=orange] (-.5,.5)
              rectangle (1.5,1.5);
    \end{pgfonlayer}
\end{tikzpicture}
```



1.7. **More detailed path operations.** The following is a more comprehensive list of path operations. (Note however that this is not even close to being an exhaustive list.)

`\path ...` ⟨*path operation*⟩ `...;`	**Description**
`\path ... path operation ...;` `[`⟨*options for* `path operation`⟩`]`	description
⟨*coordinate*⟩	move-to; sets `(current subpath start)` to ⟨*coordinate*⟩
`--`⟨*coordinate or cycle*⟩ `rounded corners=`⟨*corner radius*⟩ `sharp corners`	line-to
`-|`⟨*cordinate or cycle*⟩ `rounded corners=`⟨*corner radius*⟩ `sharp corners`	horizontal-then-vertical-to
`|-`⟨*coordinate or cycle*⟩ `rounded corners=`⟨*corner radius*⟩ `sharp corners`	vertical-then-horizontal-to
`..controls` ⟨*c*⟩ `and` ⟨*d*⟩`..`⟨*y or cycle*⟩ `rounded corners=`⟨*corner radius*⟩ `sharp corners`	curve-to; starts (resp. ends) with tangent through ⟨*c*⟩ (resp. ⟨*d*⟩); if `and` ⟨*d*⟩ is omitted, it is assumed ⟨*d*⟩ = ⟨*c*⟩
`rectangle`⟨*corner*⟩ `rounded corners=`⟨*corner radius*⟩ `sharp corners`	rectangle; one corner at the current point and another corner at ⟨*corner*⟩
`circle[`⟨*options*⟩`]` `radius=`⟨*value*⟩ `x radius=`⟨*value*⟩ `y radius=`⟨*value*⟩ `rotate=`⟨*angle*⟩ `scale=`⟨*scale factor*⟩	circle; may also write circ (⟨*value*⟩) for a circle of radius ⟨*value*⟩

Continued on next page

\path ... ⟨path operation⟩ ... ;	Description
arc[⟨options⟩] start angle=⟨degrees⟩ end angle=⟨degrees⟩ delta angle=⟨degrees⟩ radius=⟨value⟩ x radius=⟨value⟩ y radius=⟨value⟩	arc; may also write arc (⟨value⟩) for an arc of radius ⟨value⟩
grid[⟨options⟩]⟨corner or cycle⟩ step=⟨number or dimension or coordinate⟩ xstep=⟨dimension or number⟩ ystep=⟨dimension or number⟩ help lines	grid
parabola[⟨options⟩]⟨bend coordinate⟩⟨coordinate or cycle⟩ bend pos=⟨fraction⟩ parabola height=⟨dimension⟩ bend at start bend at end	bend specifies where the bend should go. bend defaults to the old current point. For bend +(0,0), saying bend pos=0 (resp. bend pos=1) puts the bend at the old (resp. new) current point, or just put bend at start (resp. bend at end)
sin⟨coordinate or cycle⟩	draws a (scaled and shifted) sine curve over $[0, \pi/2]$ from the old current point to the new current point
cos⟨coordinate or cycle⟩	similar to sin
to	to path
foreach⟨variables⟩[⟨options⟩] in ⟨list⟩ {⟨path commands⟩}	You can also write \foreach instead of foreach. E.g. \foreach \x in {1,2,...,6} {\x, } yields 1, 2, 3, 4, 5, 6, .
let \p1 = ⟨coordinate⟩ in	save a point's coordinates
\x1	x-coordinate of point \p1
\y1	y-coordinate of point \p1
\p1	string containing coordinates of \p1
{\veclen(\x1,\y1)}	length of vector (x,y)
\path ... let \n1 = {sin(60)} in ...	save a number

Continued on next page

\path ... ⟨*path operation*⟩ ... ;	**Description**
scoping	When TikZ encounters and opening or a closing brace ({ or }) at some point where a path operation should come, it will open or close a *scope*. A scope is an area where local graphics options are applied only to that area. All options that can be applied "locally" will be scoped inside the scope. For example, if you apply a transformation like `[xshift=1cm]` inside the scoped area, the shifting only applies to the scope. On the other hand, an option like `color=red` does not have any effect inside a scope since it can only be applied to the path as a whole.

`node`⟨*foreachs*⟩`[`⟨*options*⟩`]`(⟨*name*⟩) `at` (⟨*coordinate*⟩){⟨*contents*⟩}
`node contents=`⟨*contents*⟩
`behind path`
`in front of path`
`name=`⟨*node name*⟩
	Except for the described shifting, the coordinate system of a pic is reset prior to executing the pic type's code. This can be changed using the `transform shape` option, which applies "outer" transformation to the node. For `at` you should enclose ⟨*coordinate*⟩ in curly braces so that any comma inside ⟨*coordinate*⟩ does not confuse TEX.

`pic`⟨*foreachs*⟩`[`⟨*options*⟩`]`(⟨*prefix*⟩) `at` (⟨*coordinate*⟩){⟨*type*⟩} `pic type=`⟨*type*⟩ `transform shape`	The options and meanings are essentially identical to that of `node`. Inside of the pic type's code any mentioning of the origin refers to the last position used on the path or to the specified at.

Continued on next page