

Contents

Chapter 1

基础知识

1.1 字符输入

T_EX 系统中有些字符用于控制系列，不可以直接输入。

输入	<code>\#</code>	<code>\\$</code>	<code>\%</code>	<code>\&</code>	<code>\{</code>	<code>\}</code>	<code>_</code>	<code>\~{}</code>	<code>\~{}</code>
显示	#	\$	%	&	{	}	_	^	~
输入	<code>\textless</code>		<code>\textgreater</code>		<code>\textbar</code>		<code>\textbackslash</code>		
输出	<		>				\		

1.2 字体

1.2.1 字型编码：encoding

字型编码即各个个别的字在一个字型里头的排列顺序以及安排方式。原始的 T_EX 字型编码我们就称为 OT1(Old T_EX text encoding), 这是预设的, 如果都不指定字型编码, 那所使用的就是 OT1 编码。

1.2.2 字族：family

命令式	环境式	描述
<code>\textrm{text}</code>	<code>{\rmfamily text}</code>	roman 字族
<code>\textsf{text}</code>	<code>{\sffamily text}</code>	sans-serif 字族
<code>\texttt{text}</code>	<code>{\ttfamily text}</code>	monospace 字族

serif

罗马字体，又称衬线字体，字符笔画的起始处有装饰，利于阅读，为印刷专用字体。

Serif 字体著名的有：Times New Roman、DejaVu Serif、宋体、明体等。

sans-serif

无衬线字体，又称等线字体，字符笔画的起始处无装饰，专用于荧幕、简报、艺术字体、展示等，较美观，但不适于长时间阅读。

Sans-Serif 字体著名的有：DejaVu Sans、Helvetica、Verdana、圆体、黑体等。

monospace

打字机字体，又称等宽字体，每个英文字母皆设计为同一宽度，以便于排版；现代为节省不必要空间的浪费，依字母形状分配其宽度。现多用于终端机显示、程序码表示等。

Monospace 字体著名的有：Andale Mono、DejaVu SansMono、Courier、AR PL New Sung Mono。

1.2.3 字型系列：series

命令式	环境式	描述
<code>\textmd{text}</code>	<code>{\mdseries text}</code>	正常字体粗细
<code>\textbf{text}</code>	<code>{\bfseries text}</code>	bold 粗体

正常用的是 medium, m。粗体是 bold, b。然后还有 Bold extended, bx。还有 Semi-bold, sb。。和 Condensed, c

1.2.4 字形：shap

字形有正常的 normal, n。还有意大利斜体 Italic, it。斜体 Slanted, sl。和 Small Caps, sc。

命令式	环境式	描述
<code>\textup{text}</code>	<code>{\upshape text}</code>	正常字形
<code>\textit{text}</code>	<code>{\itshape text}</code>	意大利斜体
<code>\textsl{text}</code>	<code>{\slshape text}</code>	斜体
<code>\textsc{TEXT}</code>	<code>{\scshape TEXT}</code>	small caps

1.2.5 字号：size

`\tiny \scriptsize \footnotesize \small \normalsize \large \Large \LARGE \huge \Huge`

1.2.6 中文字体

用 xelatex 调用系统字体的方法

```
%以定义华文细黑字体为例
\setCJKfamilyfont{huawen}{STXihei}
\newcommand\xihei{\CJKfamily{huawen}}
```

1.3 长度

1.3.1 单位长度

单位	说明	单位	说明
pt	point()	bp	big point(1in=72bp)
pc	pica(1pc=12pt)	in	inch(1in=72.27pt)
cm	centimeter(1in=2.54cm)	mm	millimeter(1mm=2.84pt)
em	当前字体中 M 的宽度	ex	当前字体中 x 的高度
mu	math unit, 1/18 em	sp	scaled poind(65536sp=1pt)

1.3.2 长度命令

`\newlength\lengthname`: 新建一个长度名
`\setlength\lengthname{length}`: 给长度量赋值
`\addtolength\lengthname{length}`: 在原有的长度上加上一个量
 长度命令定义后, 可以使用 `\the\lengthname` 显示。

1.4 段落

1.4.1 换行与分段

`\par` 命令表分段, 一个空行当作 `\par` 处理。一个换行符当作空格处理, 多个空格当一个空格处理。另外在数学模式中空格全忽略。

换行而不分段, 用命令 `\\`。此命令后面可以跟上 `[10pt]`, 加上额外的间距, 参数可为负。

`\` 后接一个空格, 表示一个空格, 在数学模式中也有效。

由于换行被处理成一个空格, 如果想去掉它, 只需在行末加上%

1.4.2 段落对齐与间距

\TeX 中的段落缺省两端对齐 (fully justified), 下面的三个环境可以让段落分别居左、居右或居中对齐, `flushleft`, `flushright`, `center`。另有三个命令 (`\raggedleft`, `\raggedright`, `\centering`) 可以完成同样功能, 这些命令还可以控制表格图片 (也许是一切盒子?) 的位置。

段落首行缩进的距离可以用 `\parindent` 变量来控制。

段落之间的距离可以用 `\parskip` 变量来控制。比如

```
\setlength{\parskip}{1.6ex plus 0.2ex minus 0.2ex}
```

行间距是段落中相邻两行基线之间的距离, \TeX 缺省使用单倍行距。我们可以用 `\linespread` 命令来控制行距。

`\linespread{1.3}` 表示一倍半行距 `\linespread{1.6}` 表示双倍行距。另外行间距是一个 glue, 我们知道 glue 有基本的 space 和伸缩量。行距的基本 space 由命令 `baselineskip` 控制, 伸缩量有 `baselinestretch` 命令。分别用 `\setlength` 和 `\renewcommand` 来修改。如果要在文档中改变行间距必须采用如下的形式:

```
{\linespread{伸缩量}\selectfont sometext \par}
```

1.4.3 特殊段落

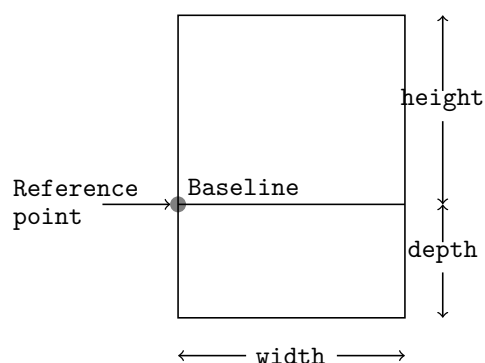
\TeX 中有三种摘录环境: `quote`, `quotation`, `verse`。`quote` 两端都缩进, `quotation` 在 `quote` 的基础上增加了首行缩进, `verse` 比 `quote` 多了第二行起的缩进。

1.5 盒子

1.5.1 盒子简述

在 \TeX 中每一个排版对象都是一个盒子。排版就是要将小盒子用空白间距粘在一起放到大盒子里, 然后再依次嵌套到更大的盒子里。怎样优化这些大大小小的盒子是一门很深的学问。

最小的盒子就是一个字符, 这些字符然后组成更大的盒子 --- 单词, 然后单词组成更大的盒子 --- 行等等。行是一个盒子, 段落也是一个盒子, 图片是一个盒子, 表格也是一个盒子。而这些盒子按照 Knuth 的描述都是用 glue 胶水粘合起来的。下面就是一个盒子的详细参数:



介绍两个 \TeX 命令。`\hbox` 让所有的盒子水平对齐，而 `\vbox` 把一些 `\hbox` 竖直对齐。

1.5.2 简单盒子

最简单的盒子命令是 `\mbox` 和 `\fbox`。前者把一组对象组合起来，后者在此基础上加上了个边框。

黑色盒子可以用于画线条：`\rule[depth]{width}{height}`

1.5.3 中等盒子

稍复杂的 `\makebox` 和 `\framebox` 命令提供了宽度和对齐方式控制的选项。其对齐方式有居中（缺省）、居左、居右和分散对齐，分别用 `c`, `l`, `r`, `s` 来表示。

语法：[宽度][对齐方式]{内容}

`raisebox` 命令一般的用法就是：`\raisebox{高度}{内容}`

表示把某个内容放进一个盒子里然后抬高多少高度，高度值可以是负值则是降低。

带颜色的盒子：`\colorbox{yellow}{test}` `\fcolorbox{red}{pink}{test}`

1.5.4 高级盒子

盒子中的文本不能换行，大一些的对象比如整个段落可以用 `\parbox` 命令或 `minipage` 环境，两者语法类似，有宽度、高度、外部对齐、内部对齐等选项。这里的外部对齐是指该盒子与周围对象的纵向关系，有三种方式：居顶、居中和居底对齐，分别用 `t`, `c`, `b` 来表示。内部对齐是指该盒子内部内容的纵向排列方式，也是同样三种。

语法：[外部对齐][高度][内部对齐] 宽度内容

1.6 学习帮助

学习 \LaTeX 开始时可以看入门的书籍，然后就需要看宏包的说明文档了。比如我要查看 `xeCJK` 文档，就在终端中输入：`texdoc xecjk`

Chapter 2

数学公式

2.1 基本知识

2.1.1 数学模式

数学命令必须在数学模式中才能使用。数学模式有正文公式，如 $(a+b)^2 = a^2 + 2ab + b^2$ ，和行间公式，如

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

实现方式如下：

	\TeX 命令	\LaTeX 命令	\LaTeX 环境	amsmath 环境
行间公式	$\$...\$$	$\backslash(...\backslash)$	math	
行间公式	$\$...\$$	$\backslash[...\backslash]$	displaymath	equation*
编号行间公式			equation	equation

有时希望在正文中实现行间公式的效果，可用 $\backslash\text{displaystyle}$ 实现。另有同类命令 $\backslash\text{textstyle}\backslash\text{scriptstyle}\backslash\text{scriptscriptstyle}$ 分别实现文中、下标、下下标的效果。

2.1.2 上下标

下标用 $_$ 表示，如 a_1 输出 a_1 ，当下标多于一个字符时用大括号，如 a_{ij} 输出 a_{ij} ，而 a_{ij} 输出 a_{ij} 。上标用 $^$ 表示，可与下标一起使用，如 a_1^2 或 a^2_1 输出 a_1^2 。

在行间公式中上下标遇到 $\backslash\text{sum}$ 之类的命令会调整为 $\sum_{k=1}^n (\backslash\text{sum}_{k=1}^n)$ 。

2.1.3 根号

$$\sqrt{2} \quad \sqrt{a} \quad \sqrt[3]{a+b} \quad \backslash\mathrm{sqrt}2 \quad \backslash\mathrm{sqrt}\{a\} \quad \backslash\mathrm{sqrt}[3]\{a+b\}$$

2.1.4 分数

$\frac{1}{2} \quad \frac{a}{b+c} \quad \frac{a}{b} \quad \backslash\mathrm{frac}12 \quad \backslash\mathrm{frac}\{a\}\{b+c\} \quad \backslash\mathrm{dfrac} \quad ab \quad \backslash\mathrm{dfrac}$ 可以在文中公式中显示行间公式的效果。

$$\binom{m}{k} \quad \backslash\mathrm{binom}\{m\}\{k\}$$

2.1.5 求和等

$$\sum_{k=1}^n k \quad \sum_{k=1}^n k \quad \backslash\mathrm{sum}_{k=1}^n k \quad \backslash\mathrm{sum}\limits_{k=1}^n k$$

$$\sum_{k=1}^n k \quad \sum_{k=1}^n k \quad \backslash\mathrm{sum}_{k=1}^n k \quad \backslash\mathrm{sum}\nolimits_{k=1}^n k$$

类似的还有 $\backslash\mathrm{prod} \quad \backslash\mathrm{lim} \quad \backslash\mathrm{int}$

2.1.6 向量

$$\vec{a} \quad \backslash\mathrm{vec}\{a\} \quad \overrightarrow{AB} \quad \backslash\mathrm{overrightarrow}\{AB\}$$

2.1.7 希腊字母

α	$\backslash\mathrm{alpha}$	β	$\backslash\mathrm{beta}$	γ	$\backslash\mathrm{gamma}$	δ	$\backslash\mathrm{delta}$	ϵ	$\backslash\mathrm{epsilon}$
ε	$\backslash\mathrm{varepsilon}$	ζ	$\backslash\mathrm{zeta}$	η	$\backslash\mathrm{eta}$	θ	$\backslash\mathrm{theta}$	ϑ	$\backslash\mathrm{vartheta}$
ι	$\backslash\mathrm{iota}$	κ	$\backslash\mathrm{kappa}$	λ	$\backslash\mathrm{lambda}$	μ	$\backslash\mathrm{mu}$	ν	$\backslash\mathrm{nu}$
ξ	$\backslash\mathrm{xi}$	\omicron	$\backslash\mathrm{o}$	π	$\backslash\mathrm{pi}$	ϖ	$\backslash\mathrm{varpi}$	ρ	$\backslash\mathrm{rho}$
ϱ	$\backslash\mathrm{varrho}$	σ	$\backslash\mathrm{sigma}$	ς	$\backslash\mathrm{varsigma}$	τ	$\backslash\mathrm{tau}$	υ	$\backslash\mathrm{upsilon}$
ϕ	$\backslash\mathrm{phi}$	φ	$\backslash\mathrm{varphi}$	χ	$\backslash\mathrm{chi}$	ψ	$\backslash\mathrm{psi}$	ω	$\backslash\mathrm{omega}$
Γ	$\backslash\mathrm{Gamma}$	Δ	$\backslash\mathrm{Delta}$	Θ	$\backslash\mathrm{Theta}$	Λ	$\backslash\mathrm{Lambda}$	Ξ	$\backslash\mathrm{Xi}$
Π	$\backslash\mathrm{Pi}$	Σ	$\backslash\mathrm{Sigma}$	Υ	$\backslash\mathrm{Upsilon}$	Φ	$\backslash\mathrm{Phi}$	Ψ	$\backslash\mathrm{Psi}$
Ω	$\backslash\mathrm{Omega}$	$\$$	$\backslash\mathrm{\$}$						

2.1.8 常用符号

\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>	\div	<code>\div</code>
$*$	<code>\ast</code>	\leq	<code>\le</code>	\geq	<code>\ge</code>	\equiv	<code>\equiv</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\sim	<code>\sim</code>	\approx	<code>\approx</code>
\cong	<code>\cong</code>	\neq	<code>\neq</code>	\perp	<code>\perp</code>	\in	<code>\in</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\angle	<code>\angle</code>	∞	<code>\infty</code>	∂	<code>\partial</code>	\triangle	<code>\triangle</code>
$'$	<code>\prime</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>	\neg	<code>\neg</code>
$\sqrt{}$	<code>\sqrt</code>	\checkmark	<code>\checkmark</code>	\cdot	<code>\cdot</code>	\cdots	<code>\cdots</code>
\ldots	<code>\ldots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>	\leftarrow	<code>\leftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\iff	<code>\iff</code>	\rightarrow	<code>\rightarrow</code>
\square	<code>\square</code>	\therefore	<code>\therefore</code>	\because	<code>\because</code>	\leqslant	<code>\leqslant</code>
\geqslant	<code>\geqslant</code>	\varnothing	<code>\varnothing</code>	\circ	<code>\circ</code>		

在符号命令前加一个 `\not` 会画一个斜线在上面, 如 `\not\in` `\not\equiv`

\notin \neq ,

2.1.9 函数

数学模式中字母是斜体, 而函数名应该用正体, 为此 \TeX 定义了一些函数名。如 `\sin x` 输出 $\sin x$ 。

`\sin` `\cos` `\tan` `\cot` `\sec` `\csc` `\arcsin` `\arccos` `\arctan` `\sinh` `\cosh`
`\exp` `\lg` `\ln` `\log` `\lim` `\inf` `\liminf` `\sup` `\limsup` `\Pr` `\tanh`
`\mod` `\gcd` `\min` `\max` `\arg` `\dim` `\det` `\deg` `\ker` `\hom`

这些函数命令可以把下标转化为下限:

`\det` `\gcd` `\inf` `\sup` `\lim` `\liminf` `\limsup` `\max` `\min`
 $10 \equiv 1 \pmod{3}$ $a \bmod b$ $y \pmod{a+b}$
`$10\equiv 1\mod 3$` `$a\bmod b$` `$y\pmod{a+b}$`

2.1.10 分隔符

我们可以在上述分隔符前面加 `\big` `\Big` `\bigg` `\Bigg` 等命令来调整其大小。也可以在分隔符前面加 `\left` `\right` 来自动调整大小, 但效果欠佳。

$$\left(\left(\left(\left(x\right)\right)\right)\right) \left[\left[\left[\left[x\right]\right]\right] \left\{\left\{\left\{\left\{x\right\}\right\}\right\}\right\}$$

$$\left\langle\left\langle\left\langle\left\langle x\right\rangle\right\rangle\right\rangle\right\rangle \left\|x\right\| \left\|x\right\|$$

2.1.11 空白间距

<code>\,</code>	<code>3mu</code>	<code>\quad</code>	<code>1em</code>
<code>\:</code>	<code>4mu</code>	<code>\qqquad</code>	<code>2em</code>
<code>\;</code>	<code>5mu</code>	<code>\!</code>	<code>-3mu</code>

2.2 跨行公式

2.2.1 矩阵

```
\[ \left(
\begin{array}{cccc}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{array}
\right)
```

`\right)\]`

`amsmath` 的 `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix` 等环境可以在矩阵两边加上各种分隔符, 但是它们没有对齐方式参数。`smallmatrix` 环境可以生成行间矩阵。

```
\[ \begin{pmatrix} a & b \end{pmatrix} \quad \begin{bmatrix} a & b \end{bmatrix} \quad \begin{Bmatrix} a & b \end{Bmatrix} \quad \begin{vmatrix} a & b \end{vmatrix} \quad \begin{Vmatrix} a & b \end{Vmatrix} ]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

2.2.2 长公式

无须对齐的长公式可以使用 `multline` 环境。需要对齐的长公式可以使用 `split` 环境, 它本身不能独立使用, 必须包含在其它数学环境内, 因此也称作次环境。

```
\begin{multline}
x = a+b+c+{} \\
d+e+f+g
\end{multline}
```

$$x = a + b + c + d + e + f + g \quad (2.1)$$

```

\[\begin{split}
x =\{ \} & \&a+b+c+\{ \} \\\
& \&d+e+f+g \\
\end{split}
\]

```

$$x = a + b + c + d + e + f + g$$

2.2.3 公式组

不需要对齐的公式组可以使用 `gather` 环境，需要对齐的公式组用 `align` 环境。

```

\begin{gather}
a = b+c+d \\\
x = y+z \\
\end{gather}

```

$$a = b + c + d \quad (2.2)$$

$$x = y + z \quad (2.3)$$

```

\begin{align}
a \&= b+c+d \\\
x \&= y+z \\
\end{align}

```

$$a = b + c + d \quad (2.4)$$

$$x = y + z \quad (2.5)$$

`multline`, `gather`, `align` 等环境都有带 `*` 的版本，不生成公式编号。

2.2.4 分支公式

```

\[\begin{cases}
-x, \text{quad } x \leq 0 \\\
x, \text{quad } x > 0 \\
\end{cases}
\]

```

$$y = \begin{cases} -x, & x \leq 0 \\ x, & x > 0 \end{cases}$$

2.3 定理和证明

下面的代码定制了四个环境：定义、定理、引理和推论，它们都在一个 `section` 内统一编号，而引理和推论会延续定理的编号。

```

\newtheorem{definition}{定义}[section]
\newtheorem{theorem}{定理}[section]
\newtheorem{lemma}{引理}[theorem]
\newtheorem{corollary}{推论}[theorem]

```

`amsthm` 宏包提供的 `proof` 环境可以用来输入证明，它会在证明结尾加一个 QED 符号。

2.4 数学字体

Chapter 3

表格处理

Chapter 4

编译

Chapter 5

宏包介绍

5.1 xcolor

定义颜色\definecolor{bgcolor-co}{RGB}{255,255,255}

红色 一种激奋的色彩。刺激效果，能使人产生冲动，愤怒，热情，活力的感觉。

绿色 介于冷暖两中色彩的中间，显得和睦，宁静，健康，安全的感觉。它和金黄，淡白搭配，可以产生优雅，舒适的气氛。

橙色 也是一种激奋的色彩，具有轻快，欢欣，热烈，温馨，时尚的效果。

黄色 具有快乐，希望，智慧和轻快的个性，它的明度最高。




















蓝色 是最具凉爽，清新，专业的色彩。它和白色混合，能体现柔顺，淡雅，浪漫的气氛（像天空的色彩:）

白色 具有洁白，明快，纯真，清洁的感受。

黑色 具有深沉，神秘，寂静，悲哀，压抑的感受。

灰色 具有中庸，平凡，温和，谦让，中立和高雅的感觉。

基本颜色

 black	 darkgray	 lime	 pink	 violet
 blue	 gray	 magenta	 purple	 white
 brown	 green	 olive	 red	 yellow
 cyan	 lightgray	 orange	 teal	

green!60!white

 green	 green	 green
 green!80!white	 green!80!gray	 green!80!black
 green!60!white	 green!60!gray	 green!60!black
 green!40!white	 green!40!gray	 green!40!black
 green!20!white	 green!20!gray	 green!20!black
 white	 gray	 black
 green	 green	 green
 green!80!red	 green!80!blue	 green!80!yellow
 green!60!red	 green!60!blue	 green!60!yellow
 green!40!red	 green!40!blue	 green!40!yellow
 green!20!red	 green!20!blue	 green!20!yellow
 red	 blue	 yellow

多色混合

 red	 -red
 red!75	 -red!75
 red!75!green	 -red!75!green
 red!75!green!50	 -red!75!green!50
 red!75!green!50!blue	 -red!75!green!50!blue
 red!75!green!50!blue!25	 -red!75!green!50!blue!25
 red!75!green!50!blue!25!gray	 -red!75!green!50!blue!25!gray

5.1.1 PGF

宏包调用的名称是 `tikz`






















PGF 的缺省长度单位的 `1cm`

```
\pgfsetxvec{\pgfpoint{10pt}{0}}
\pgfsetyvec{\pgfpoint{0pt}{10pt}}
```

5.2 Tikz

5.2.1 快速参考

线型

	solid: 默认		ultra thin 0.1pt		to
	dotted		very thin 0.2pt		latex
	densely dotted		thin 0.4pt		stealth
	loosely dotted		semithick 0.6pt		
	dashed		thick 0.8pt		o
	densely dashed		very thick 1.2pt	)
	loosely dashed		ultra thick 1.6pt		
	double	也可以通过 line width 直接定义			

文本的距离

at star=0 very near start=0.123 near start=0.25 near end=0.75
 very near end=0.875 at end=1

标签

label=above label=below label=left label=right
 above left above right below left below right

calc library

$(\$ (A) + \sin(60) * (B) \$)$	coordinate calculations
$(\$ (A) ! 0.25 ! (B) \$)$	partway calculations
$(\$ (A) ! 3cm ! (B) \$)$	3cm from (A) in direction of (B)
$(\$ (A) ! 1.2 ! 30:(B) \$)$	stretch by 1.2, then rotate by 30°
$(\$ (A) ! (B) ! (C) \$)$	projection of point B onto line AC
$(\$ (A) ! (B) ! 30:(C) \$)$	project B onto line AC, then rotate by 30°

Let-operations

<code>\path ... let \p1 = (\$ (B)-(A) \$) in ...</code>	save a point' s coordinates
<code>... \x1</code>	x-coordinate of point \p1
<code>... \y1</code>	y-coordinate of point \p1
<code>... \p1</code>	string containing coordinates of \p1
<code>... {veclen(\x1,\y1)}</code>	length of vector (x,y)
<code>\path ... let \n1 ={\sin(60)} in ...</code>	save a number

Layers

```

\pgfdeclarelayer{background}
\pgfdeclarelayer{foreground}
\pgfsetlayers{background,main,foreground}

\begin{pgfonlayer}{background}
\node {This node will appear on the background layer.};
\end{pgfonlayer}

```

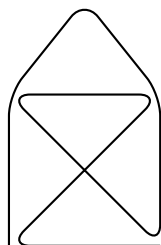
等价命令

```

\draw = \path[draw], \fill = \path[fill], \clip = \path[clip]
\filldraw = \path[fill,draw], \shade = \path[shade], ...

```

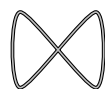
5.2.2 基本图形



```

\tikz \draw[thick,rounded corners=8pt]% 此处默认是
4pt
(0,0)--(0,2)--(1,3.25)--(2,2)--(2,0)--(0,2)--(2,2)--(0,0)--(2,0)

```



```

\tikz \draw[double,draw=black,double=lightgray]
plot[smooth cycle] coordinates{(0,0) (1,1) (1,0)
(0,1)};

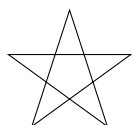
```



```

\tikz\draw(0,0)rectangle(1,1);

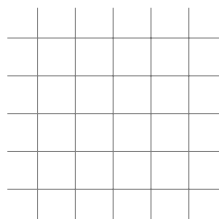
```



```

\tikz\draw(0,0)--(72:1.618)--(1,0)--(108:1)--(36:1.618)--cycle;

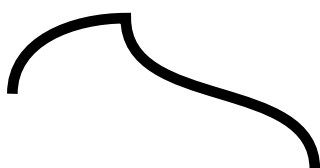
```



```
\tikz \draw[step=.5cm,gray,very thin] (-1.4,-1.4)
grid (1.4,1.4);
```



```
\begin{tikzpicture}
\draw (0,0)..controls(1,1)and(2,1)..(2,0);
\foreach\x in(0,0),(1,1),(2,1),(2,0)
\fill[black,opacity=0.5]\x circle(2pt);
\end{tikzpicture}
```



```
\begin{tikzpicture}[scale=.5]
\draw[line width=4pt] (0,0) to [out=90,
in=180] (3,2)
to [out=-90, in=90] (8,-2);
```



```
\tikz \draw (0,0) ellipse (2pc and 1pc);
```



```
\tikz \draw(0,0)arc(0:270:1.75pc and 1pc);
```

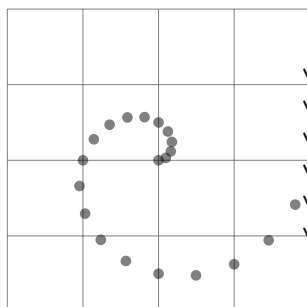


```
\tikz \draw[x=2pt,y=2pt] (0,0) parabola bend
(4,16) (6,12);
```



```
\tikz \draw[x=3.14ex,y=2ex] (0,0) sin (1,1) cos
(2,0) sin (3,-1) cos (4,0)
(0,1) cos (1,0) sin (2,-1) cos (3,0) sin (4,1);
```

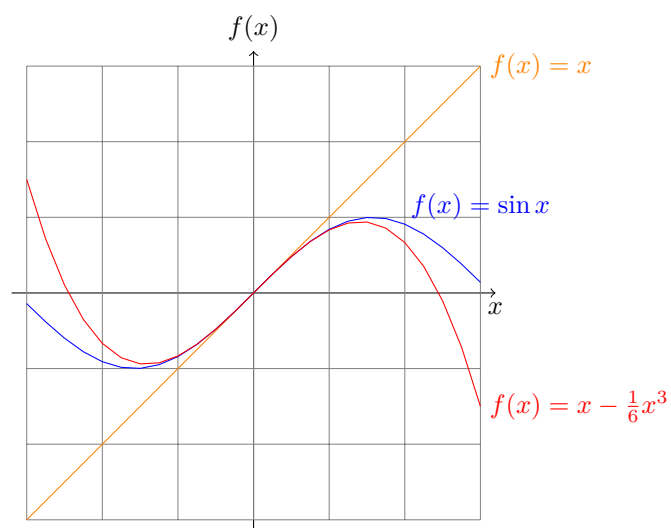
5.2.3 特殊图形



```
\begin{tikzpicture}
\draw [help lines] (0,0) grid (4,4);
\foreach \i in 0,0.1,...,2
\fill [black,opacity=0.5] ($ (2,2)
• \i*180:(3,2)$) circle (2pt);
\end{tikzpicture}
```

5.2.4 命令示例

Plot

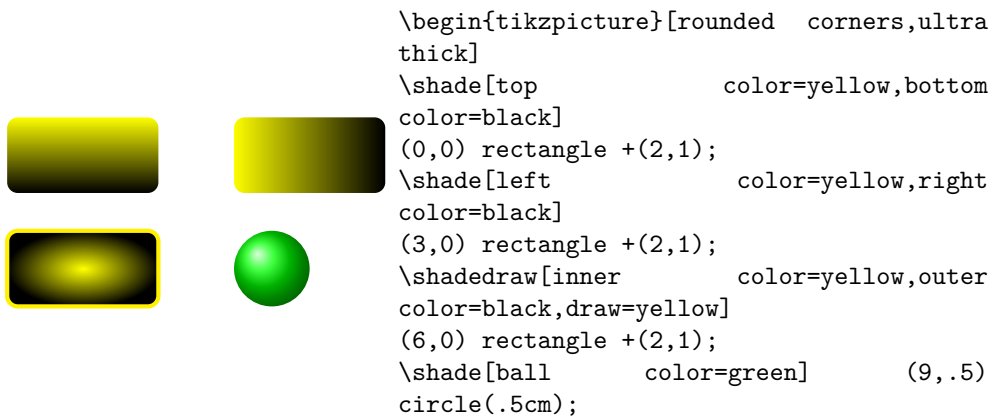


```
\begin{tikzpicture}[domain=-3:3]
\draw[->] (-3.2,0) -- (3.2,0) node[below] {$x$};
\draw[->] (0,-3.2) -- (0,3.2) node[above] {$f(x)$};
\draw[very thin,color=gray] (-3,-3) grid (3,3);
\draw[color=orange] plot (\x,\x) node[right] {$f(x)=x$};
\draw[color=blue] plot (\x,{sin(\x r)})
node[above=7mm] {$f(x)=\sin x$};
\draw[color=red] plot(\x,{\x-(1/6)*(\x)^3})
node[right]{$f(x)=x-\frac{1}{6}x^3$};
```

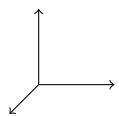
Shading



```
\tikz \shade (0,0) rectangle (2,1) (3,0.5) circle
(0.5);
```

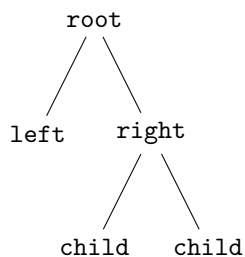
XYZ



```

\begin{tikzpicture}[->]
\draw (0,0) -- (xyz cs:x=1);
\draw (0,0) -- (xyz cs:y=1);
\draw (0,0) -- (xyz cs:z=1);

```

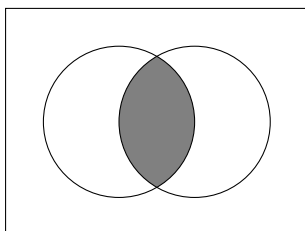


```

\tikz\noderoot
child node left
child node right
child node child
child node child
;

```

scope



```

\begin{tikzpicture} \draw (-2, 1.5)
rectangle (2, -1.5); \begin{scope} \clip
(-0.5, 0) circle (1); \clip ( 0.5, 0) circle
(1); \fill[color=gray] (-2,1.5) rectangle
(2,-1.5); \end{scope} \draw (-0.5, 0)
circle (1); \draw ( 0.5, 0) circle (1);
\end{tikzpicture}

```

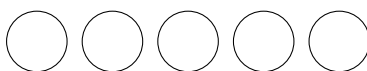
even odd rule



```
\begin{tikzpicture}[
  even odd rule,rounded corners=2pt,x=10pt,y=10pt]
\filldraw[fill=yellow!80!black] (0,0) rectangle (1,1)
[xshift=5pt,yshift=5pt] (0,0) rectangle (1,1)
[rotate=30] (-1,-1) rectangle (2,2);
\end{tikzpicture}
```

Foreach

$x = 1, x = 2, x = 3,$ `\foreach \x in {1,2,3} { $\$x=\x$, }`



```
\tikz \foreach \x in {1,...,10} \draw (\x,0) circle (0.4);
```

1,5	2,5	3,5	4,5	5,5
1,4	2,4	3,4	4,4	5,4
1,3	2,3	3,3	4,3	5,3
1,2	2,2	3,2	4,2	5,2
1,1	2,1	3,1	4,1	5,1

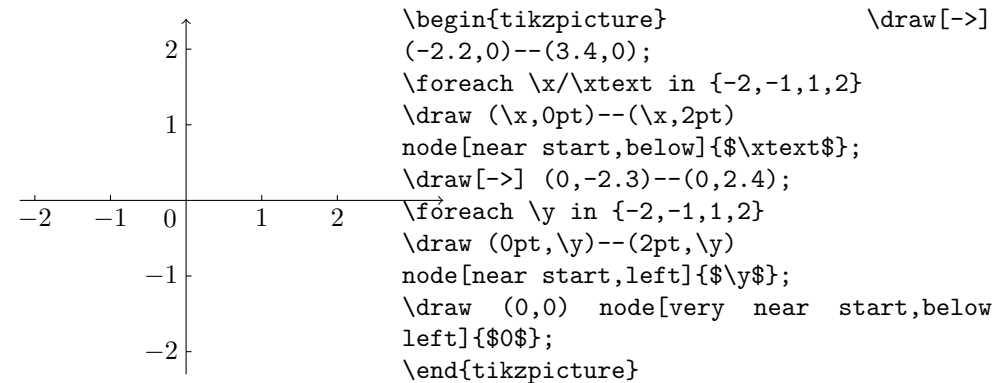
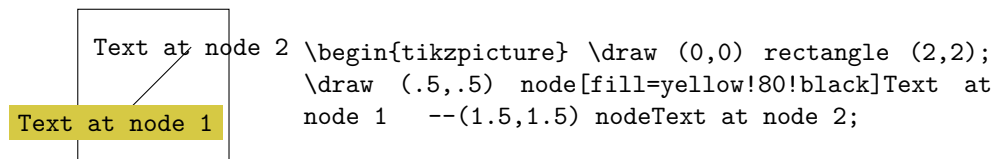
7,5	8,5	9,5	10,5	11,5
7,4	8,4	9,4	10,4	11,4
7,3	8,3	9,3	10,3	11,3
7,2	8,2	9,2	10,2	11,2
7,1	8,1	9,1	10,1	11,1

```
\begin{tikzpicture}
\foreach \x in {1,2,...,5,7,8,...,12}
\foreach \y in {1,...,5}
{ \draw (\x,\y) +(-.5,-.5)
  rectangle +(.5,.5);
\draw (\x,\y) node{\x,\y}; }
\end{tikzpicture}
```

Node

`node` 命令的可选项 `left` , `right` , `above` , `below` 用于控制插入文本的位置。此外还有 `above right` , `below right` , `above left` , `below left` 等。文本对齐用 `align=left` , `align=right` , `align=center` 来控制在 `node` 里面用

`includegraphics` 命令可以插入图片



Style

```
\tikzset{help lines/.style= {step=0.5cm,color=gray!40,very thin}}
```

```
\tikzset{style001/.style={color=red,fill=red!20}}
```

```
\tikzset{help lines/.append style=blue!50}%原样式修改
```

```
\tikz \path (top-left) ++(1,-2) coordinate (name-point);%定义相对点
```

这种形式 (p1 |- xline) 表示取第一个点的 x 和第二个点的 y 组成一个新的点。
如果是 (p1 -| xline) 表示取第二个点的 x 和第一个点的 y 组成一个新的点。

```
shorten >=-0.4cm,shorten <=-0.4cm
```

可以通过类似上面的选项让两个点确定的线条延长,

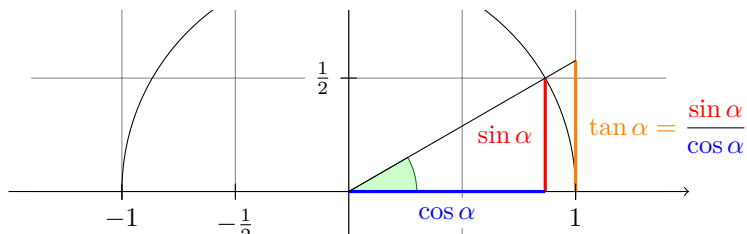
不过这种延长是不能用 `intersection` 方法处理的。

其中 `>=` 表示到第二个点超过

的部分, 负值表示超过; 然后 `<=` 表示到第一个点超过的部分, 正值则缩回去了。

5.2.5 实例

三角函数

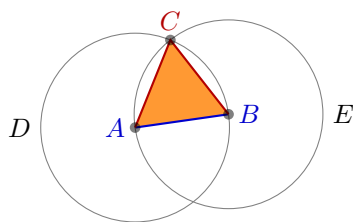


```

\begin{tikzpicture}[scale=3]
\clip (-2,-0.2) rectangle (2,0.8); %设定画图范围
\draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
\filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm)
arc [start angle=0, end angle=30, radius=3mm] -- cycle;
\draw[->] (-1.5,0) -- (1.5,0) coordinate (x axis);
\draw[->] (0,-1.5) -- (0,1.5) coordinate (y axis);
\draw (0,0) circle [radius=1cm];
\draw[very thick,red]
(30:1cm) -- node[left=1pt,fill=white] {\sin \alpha}
(30:1cm |- x axis); %|-表示画垂线
\draw[very thick,blue]
(30:1cm |- x axis) -- node[below=2pt,fill=white] {\cos \alpha} (0,0);
\path [name path=upward line] (1,0) -- (1,1); % 命名路径
\path [name path=sloped line] (0,0) -- (30:1.5cm);
\draw [name intersections={of=upward line and sloped line, by=t}] %交点
[very thick,orange] (1,0) --
node [right=1pt,fill=white]
{\displaystyle\tan\alpha\color{black}=
\frac{\color{red}\sin\alpha}{\color{blue}\cos\alpha}(t)};
\draw (0,0) -- (t);
\foreach \x/\xtext in {-1, -0.5/-\frac{1}{2}, 1}%注意1/2的表示法
\draw (\x cm,1pt) -- (\x cm,-1pt) node[anchor=north,fill=white] {\xtext};
\foreach \y/\ytext in {-1, -0.5/-\frac{1}{2}, 0.5/\frac{1}{2}, 1}
\draw (1pt,\y cm) -- (-1pt,\y cm) node[anchor=west,fill=white] {\ytext};
\end{tikzpicture}

```

等边三角形



Proposition I

To construct an *equilateral triangle* on a given *finite straight line*.

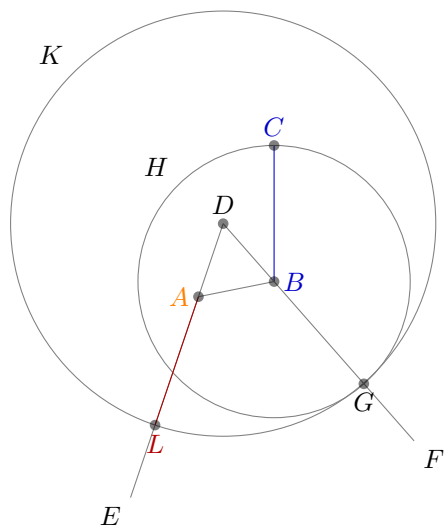
Let *AB* be the given *finite straight line*. ...

```
\begin{tikzpicture}[thick]
\tikzset{help lines/.style={thin,draw=black!50}}
\def\A{\textcolor{input}{$A$}} \def\B{\textcolor{input}{$B$}}
\def\C{\textcolor{output}{$C$}} \def\D{$D$} \def\E{$E$}
\colorlet{input}{blue!80!black} \colorlet{output}{red!70!black}
\colorlet{triangle}{orange}

\coordinate[label=left:\A] (A) at ($(0,0)+.1*(rand,rand)$);
\coordinate[label=right:\B] (B) at ($(1.25,0.25)+.1*(rand,rand)$);
\draw [input] (A)--(B);
\node[name path=D,help lines,draw,label=left:\D] (D) at (A) [circle through=(B)] {};
\node[name path=E,help lines,draw,label=right:\E] (E) at (B) [circle through=(A)] {};
\path[name intersections={of=D and E,by={\C}}];
\draw[output] (A)--(C)--(B);
\foreach \point in {A,B,C}\fill[black,opacity=.5](\point)circle(2pt);

\begin{pgfonlayer}{background}
\fill[triangle!80] (A) -- (C) -- (B) -- cycle;
\end{pgfonlayer}

\node [below right, text width=10cm,align=justify] at (4,2) {
\small\textbf{Proposition I}\par
\emph{To construct an \textcolor{triangle}{equilateral triangle}
on a given \textcolor{input}{finite straight line}.}
\par\vskip1em
Let \A\B be the given \textcolor{input}{finite straight line}. \dots
};
\end{tikzpicture}
```



```

\begin{tikzpicture}
\tikzset{help lines/.style={thin,draw=black!50}}
\def\A{\textcolor{orange}{$A$}} \def\B{\textcolor{input}{$B$}}
\def\C{\textcolor{input}{$C$}} \def\L{\textcolor{output}{$L$}}
\colorlet{input}{blue!80!black} \colorlet{output}{red!70!black}

\coordinate[label=left:\A](A)at(0,0);
\coordinate[label=right:\B](B)at(1,0.2);
\coordinate[label=above:\C](C)at(1,2);
\draw[input](B)--(C);
\draw[help lines](A)--(B);
\coordinate[label=above:\D](D)at($(A)!.5!(B)!\sin(60)*2)!90:(B)$);

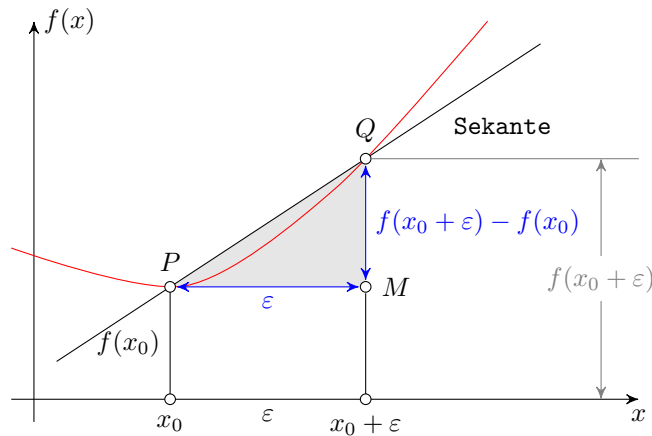
\draw[help lines](D)--($(D)!3.75!(A)$) coordinate[label=-135:\E](E);
\draw[help lines](D)--($(D)!3.75!(B)$) coordinate[label=-45:\F](F);
\node[name path=H,help lines,circle through=(C),draw,label=135:\H](H)at(B){};

\path[name path=B--F](B)--(F);
\path[name intersections={of=H and B--F,by={ [label=below:\G]G}}];
\node(K)at(D)[name path=K,help lines,circle through=(G),draw,label=135:\K]{};

\path[name path=A--E](A)--(E);
\path[name intersections={of=K and A--E,by={ [label=below:\L]L}}];
\draw[output](A)--(L);

```

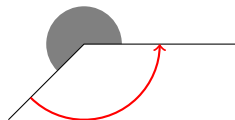
```
\foreach \point in {A,B,C,D,G,L}
\fill[black,opacity=.5](\point)circle(2pt);
\end{tikzpicture}
```



```
\begin{tikzpicture}
\tikzset{thick,>=stealth',dot/.style={draw,fill=white,circle,inner sep=0pt,minimum size=4pt}}
\coordinate (0) at (0,0);
\draw[->] (-0.3,0)--(8,0) coordinate[label=below:$x$] (xmax);
\draw[->] (0,-0.3)--(0,5) coordinate[label=right:$f(x)$] (ymax);
\path[name path=x] (0.3,0.5) -- (6.7,4.7);
\path[name path=y] plot[smooth] coordinates {(-0.3,2) (2,1.5) (4,2.8) (6,5)};
\begin{scope}[name intersections = {of= x and y,name = i}]
\fill[gray!20] (i-1) -- (i-2 |- i-1) -- (i-2) -- cycle;
\draw (0.3,0.5) -- (6.7,4.7) node[pos=0.8,below right] {Sekante};
\draw[red] plot[smooth] coordinates {(-0.3,2) (2,1.5) (4,2.8) (6,5)};
\draw (i-1) node[dot,label={above:$P$}] (i-1) {} -- node[left] {$f(x_0)$} (i-1 |- 0) node[dot,label={below:$f(x_0)$}] (i-1) {};
\path (i-2) node[dot,label={above:$Q$}] (i-2) {} -- (i-2 |- i-1) node[dot,label={right:$M$}] (i-12) {};
\draw (i-12) -- (i-12 |- 0) node[dot,label={below:$x_0 + \varepsilon$}] (i-12) {};
\draw[blue,<->] (i-2) -- node[right] {$f(x_0 + \varepsilon) - f(x_0)$} (i-12);
\draw[blue,<->] (i-1) -- node[below] {$\varepsilon$} (i-12);
\path (i-1 |- 0) -- node[below] {$\varepsilon$} (i-2 |- 0);
\draw[gray] (i-2) -- (i-2 -| xmax);
\draw[gray,<->] ([xshift=-0.5cm]i-2 -| xmax) -- node[fill=white] {$f(x_0 + \varepsilon)$} ([xshift=-0.5cm]i-2 -| xmax);
\end{scope}
\end{tikzpicture}
```

5.2.6 Library

Angels

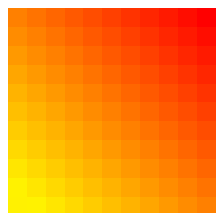


```
\tikz \draw (2,0) coordinate(A)--(0,0)coordinate(B)--(-1,-1)coordinate(C)
pic [fill=black!50] {angle = A--B--C}
pic [draw,->,red,thick,angle radius=1cm] {angle = C--B--A};
```

Math

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
1597, 2584, 4181, 6765,

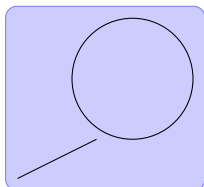
```
\tikzmath{function fibonacci(\n){
if \n == 0 then {return 0;}
else {return fibonacci2(\n, 0, 1);}
};
function fibonacci2(\n, \p, \q){
if \n == 1 then {return \q;}
else {return fibonacci2(\n-1, \q, \p+\q);}
};
int \f, \i;
for \i in {0,1,...,20}{\f= fibonacci(\i); print {\f,};}
};}
```



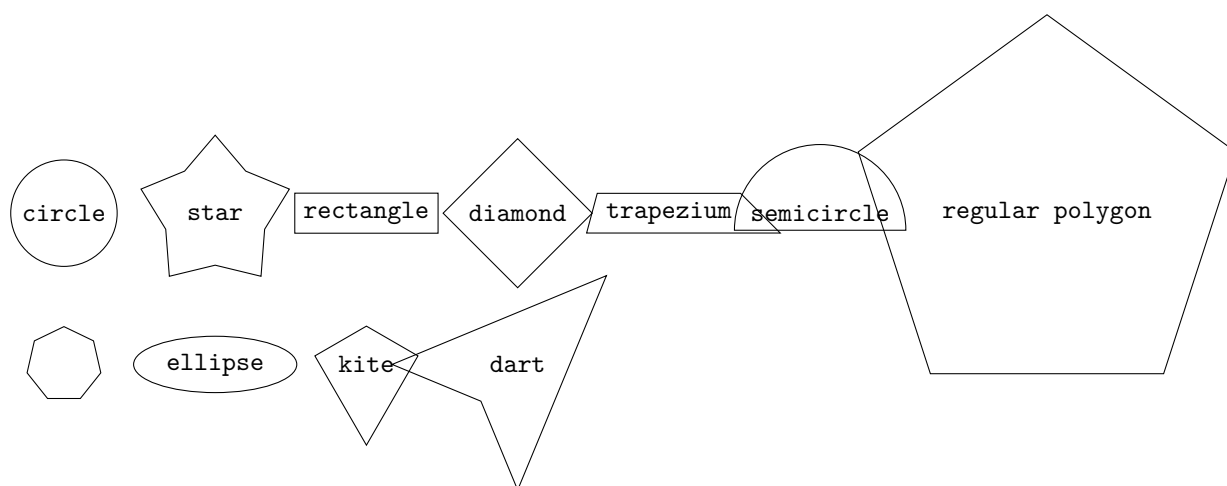
```
\tikz[x=0.25cm,y=0.25cm,evaluate={
int \i, \j;
for \i in {0,...,10}{
for \j in {0,...,10}{
\{a{\i,\j} = (\i+\j)*5;\};\};}
\foreach \i in {0,...,10}
\foreach \j in {0,...,10}
\fill [red!\a{\i,\j}!yellow] (\i,\j) rectangle ++(1, 1);}
```

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

```
\tikzmath{ int \x, \y, \z;
\x=random(2,5);
for \y in {0,...,6}{\z=\x^\y; print{\$ \x^\y=\z$,};}}
```


backgrounds

```
\begin{tikzpicture}[ scale=.8,background rectangle/.style=
{draw=blue!50,fill=blue!20,rounded corners=1ex},show background rectangle]
\draw (2,2) circle (1);
\draw (1 mm, 10 pt) -- (4 em, 1);
```

shapes**5.2.7 变换**

`xscale=-1` 或者 `yscale=-1` 就刚好相对 y 轴或 x 轴反对称。

5.3 Beamer**5.4 paralist**

\LaTeX 的列表缺省行间距较大，如要节省空间，可以考虑 Bernd Schandl 的 `paralist` 宏包，它提供了一系列压缩列表和行间列表环境。

它提供了类似的环境 `compactitem`, `compactenum`, `compactdesc`. 另外还提供了行间列表环境 `inparaitem`, `inparaenum`, `inparadesc`.