Pionpill's TikZ Notebook

Pionpill [1]

学习 TikZ 的笔记，正在学习中，有待更新
本文主要根据 TikZ 官方手册 [2] 学习

2021 年 8 月 27 日

[1]笔名：北岸，电子邮件: 673486387@qq.com，Github：https://github.com/Pionpill
[2]TikZ Github：https://github.com/pgf-tikz/pgf

目录

I 案例

次章节源于 TikZ 手册[1] 的学习

[1] TikZ Github 仓库：TikZGithub https://github.com/pgf-tikz/pgf

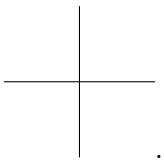
一、三角函数：A Picture for Karl's Students

1.1 环境：坐标轴

启用 tikz 环境：

```
\begin{tikzpicture}
    ......
\end{tikzpicture}
```

画坐标轴：



```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \end{tikzpicture}.
```

图 1.1.1 直线

两种 TikZ 画法：

1. 通过 TikZ 环境，如上所述
2. 行内 TikZ，如 _____

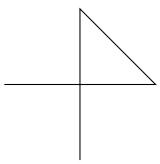
在不同的环境下 TikZ 有多种写法，这里不做说明

1.2 基本曲线

1.2.1 直线

直线通过 `\draw` 指令绘制：

```
\draw (x,y) -- (x,y) -- (x,y) ... ;
```



```
\tikz \draw (-1,0) -- (1,0) -- (0,1) -- (0,-1);
```

图 1.1.2 直线

1.2.2 曲线

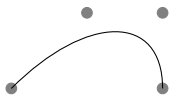
TikZ 的曲线原理在官方文档中没有详细解释，类似于贝塞尔曲线，我的理解如下：

- 和直线类似，曲线也由点组成，和分为起止点，过程点
- 起止点为曲线的起点和终点，必须经过
- 过程点比必须经过，曲线有趋近的趋势

通过 `\.. controls` 关键字控制曲线

```
Official: <start point> .. controls <1st control point> and <2nd control point> .. <end point>;
Simple: \draw (x,y) .. controls (x,y) and (x,y) .. (x,y);
```

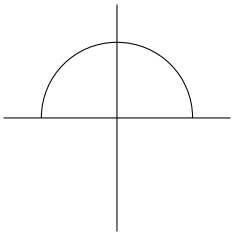
可以取消.. 这样会让第一个点使用两次



```
\begin{tikzpicture}
   \filldraw [gray] (0,0) circle [radius=2pt]
                    (1,1) circle [radius=2pt]
                    (2,1) circle [radius=2pt]
                    (2,0) circle [radius=2pt];
   \draw (0,0) .. controls (1,1) and (2,1) .. (2,0);
\end{tikzpicture}
```

图 1.1.3 曲线

继续坐标系的例子，可以画出伪半圆，注意下面的例子中，(0,1) 点既作为终点又作为起点



```
\begin{tikzpicture}
   \draw (-1.5,0) -- (1.5,0);
   \draw (0,-1.5) -- (0,1.5);
   \draw (-1,0) .. controls (-1,0.555) and (-0.555,1) .. (0,1)
   .. controls (0.555,1) and (1,0.555) .. (1,0);
\end{tikzpicture}
```

图 1.1.4 带半圆曲线的坐标轴

1.2.3 椭圆

可以通过如下方式绘制椭圆

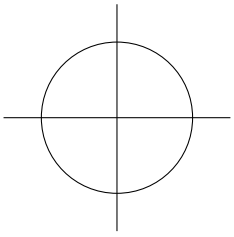
```
\draw <point> <category> [args];
```

其中：<point>: 圆心 <category>: 椭圆类型 [args]: 对应的参数例如下面绘制的圆和椭圆

```
\draw (0,0) circle [radius=10pt];
\draw (0,0) ellipse [x radius=20pt, y radius=10pt];
```

图 1.1.5 椭圆

继续在坐标轴上加上圆



```
\begin{tikzpicture}
   \draw (-1.5,0) -- (1.5,0);
   \draw (0,-1.5) -- (0,1.5);
   \draw (0,0) circle [radius = 1cm];
\end{tikzpicture}
```

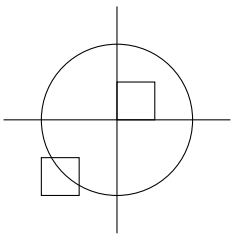
图 1.1.6 带单位圆的坐标轴

1.2.4 矩形

矩形的绘制方式和椭圆几乎一致，只需要将下面 category 写为 rectangle

```
\draw <point> <category> [args];
```

在上述坐标轴基础上继续绘制矩形



```
\begin{tikzpicture}
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \draw (0,0) rectangle (0.5,0.5);
    \draw (-0.5,-0.5) rectangle (-1,-1);
\end{tikzpicture}
```

图 1.1.7 带矩形的单位圆

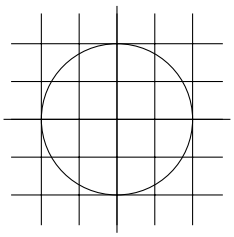
1.3 网格

1.3.1 网格的绘制

网格的绘制形式：

```
\draw[step=2pt] (0,0) grid (10pt,10pt);
```

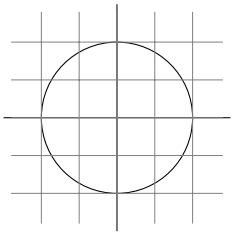
与曲线不同的是，第一个坐标点表示网格一个角，第二个坐标点则表示另一个不相邻的角，由此，我们可以在坐标轴基础上绘制网格背景了



```
\begin{tikzpicture}
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \draw[step=.5cm] (-1.4,-1.4) grid (1.4,1.4);
    \end{tikzpicture}
```

图 1.1.8 带网格的坐标轴

网格的颜色与单位圆相同，不易区别，这里使用更多的参数来控制



```
\begin{tikzpicture}
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \end{tikzpicture}
```

图 1.1.9 修改颜色的带网格的坐标轴

1.3.2 样式设置

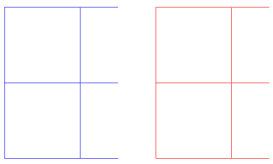
图像中的样式对于进场重复使用的样式，我们可以设置一个统一样式，如网格虚线，我们定义为辅助线 help lines。

```
help lines/. style={color=blue!50,very thin}
```

全局样式除了在图像中定义样式供单个图像使用，还可以定义全局样式。全局样式允许嵌套

```
\tikzset{help lines/.style=very thin}
\tikzset{Pionpill gird/.style={help lines,color = blue!50}}
```

样式同时可以设置参数



```
\begin{tikzpicture}
    [Karl's grid/.style ={help lines,color=#1!50},
    Karl's grid/.default=blue]
    \draw[Karl's grid] (0,0) grid (1.5,2);
    \draw[Karl's grid=red] (2,0) grid (3.5,2);
    \end{tikzpicture}
```

图 1.1.10 带参数的样式

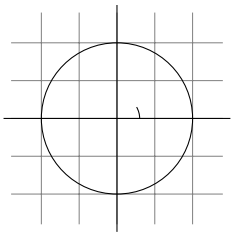
网格除了以上重要参数，还有虚线参数可以设置 dash pattern

1.4 弧度

绘制弧度的形式：

```
\draw (x,y) arc [args]
```

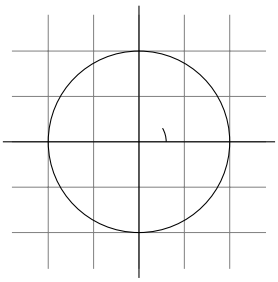
和网格类使，第一个坐标表示弧线起点，arc 为关键词，之后为样式参数，默认为逆时针旋转



```
\begin{tikzpicture}
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw(-1.5,0) -- (1.5,0);
    \draw(0,-1.5) -- (0,1.5);
    \draw(0,0) circle [radius=1cm];
    \draw(3mm,0mm) arc [start angle=0,end angle=30,radius=3mm];
\end{tikzpicture}
```

图 1.1.11 带弧度的单位圆

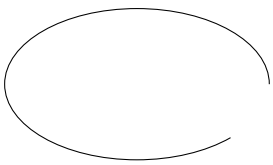
如果对图像大小不满意，可以使用 tikzpicture 的 scale 参数进行放大



```
\begin{tikzpicture}[scale=1.2]
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \draw (3mm,0mm) arc [start angle=0, end angle=30, radius=3
        mm];
    \end{tikzpicture}
```

图 1.1.12 放大后的图像

除了圆弧，还能绘制椭圆弧



```
\tikz \draw (0,0) arc [start angle=0, end angle=315, x radius
    =1.75cm, y radius=1cm];
```

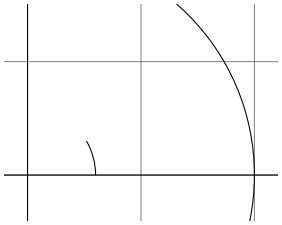
图 1.1.13 椭圆弧

1.5 截图

截图的用法与网格十分相识，可以指定截图方式，以及截图的对角来划定区域

```
\clip (x,y) graph (x,y)
```

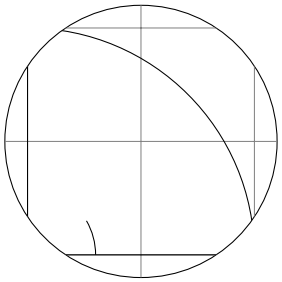
需要注意的是，clip 指令需要写在前面的位置，下面使用矩形截图：



```
\begin{tikzpicture}[scale = 3]
    \clip (-0.1,-0.2) rectangle (1.1,0.75);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius = 1cm];
    \draw (3mm,0mm) arc [start angle=0, end angle=30, radius=3
        mm];
\end{tikzpicture}
```

图 1.1.14 矩形截图

下面结合 draw 绘制圆形截图，技术细节不在此讨论



```
\begin{tikzpicture}[scale=3]
    \clip[draw] (0.5,0.5) circle (.6cm);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \draw (3mm,0mm) arc [start angle=0, end angle=30, radius=3
        mm];
\end{tikzpicture}
```

图 1.1.15 结合 draw 绘制圆形截图

1.6 抛物线与三角函数

1.6.1 抛物线

抛物线同样使用 draw 命令绘制

```
\tikz \draw (0,0) parabola (1,1)
```

图 1.1.16 抛物线

其他形式的抛物线

```
\tikz \draw[x=1pt,y=1pt] (0,0) parabola bend (4,16) (6,12)
```

图 1.1.17 其他形式抛物线

1.6.2 三角函数

三角函数同样使用 draw 指令绘制

```
\draw[x=,y=] (x,y) sin (x,y)
```

其中，可选项 [x=,y=] 用于控制图像比例，第一个点为起点，随后指定三角函数类型，最后一个点为终点


```
\tikz \draw[x=1ex,y=1ex] (0,0) sin (1.57,1);
```

图 1.1.18 部分三角函数图像

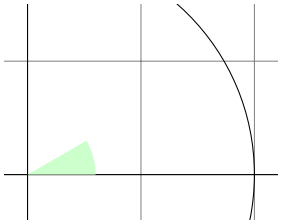
```
\tikz \draw [x=1.57ex,y=1ex] (0,0) sin(1,1) cos(2,0) sin(3,-1)
    cos(4,0) (0,1) cos(1,0) sin(2,-1) cos(3,0) sin(4,1);
```

图 1.1.19 三角函数图像

1.7 填充与边线

1.7.1 基本填充与边线

使用 fill 代替 draw 关键字可绘制填充图形



```
\begin{tikzpicture}[scale = 3]
   \clip (-0.1,-0.2) rectangle (1.1,0.75);
   \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
   \draw (-1.5,0) -- (1.5,0);
   \draw (0,-1.5) -- (0,1.5);
   \draw (0,0) circle [radius=1cm];
   \fill[green!20!white] (0,0) -- (3mm,0mm)
   arc [start angle=0, end angle=30, radius=3mm] -- (0,0);
\end{tikzpicture}
```

图 1.1.20 填充图形

边框线：useasboundingbox

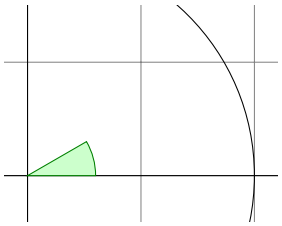
```
\useasboundingbox (low,high)
```

low 和 high 分别代表原边界向内，向外扩展，可使用 cycle 来指明曲线闭合

```
\begin{tikzpicture}[line width=5pt]
   \draw (0,0) -- (1,0) -- (1,1) -- (0,0);
   \draw (2,0) -- (3,0) -- (3,1) -- cycle;
   \useasboundingbox (0,1.5); % make bounding box higher
\end{tikzpicture}
```

图 1.1.21 边框粗细与闭合

可以使用 filldraw 绘制有边界线和填充的图形



```
\begin{tikzpicture}[scale = 3]
    \clip (-0.1,-0.2) rectangle (1.1,0.75);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \filldraw[fill=green!20!white, draw=green!50!black] (0,0)
        -- (3mm,0mm) arc [start angle=0, end angle=30, radius
        =3mm] -- cycle;
\end{tikzpicture}
```

图 1.1.22 filldraw 效果

1.7.2 渐变

与填充类使，渐变也有两种画法 shade 与 shadedraw

```
\tikz \shade (0,0) rectangle (2,1) (3,0.5) circle (.5cm)
```

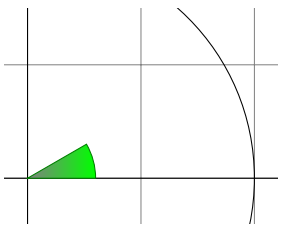
图 1.1.23 shade 效果

更详细的渐变设置

```
\begin{tikzpicture}[scale = 1]
    \shade [top color=yellow,bottom color=black] (0,0) rectangle + (2,1);
    \shade[left color=yellow,right color=black] (3,0) rectangle +(2,1);
    \shadedraw[inner color=yellow,outer color=black,draw=yellow] (6,0) rectangle +(2,1);
    \shade[ball color=green] (9,.5) circle (.5cm);
\end{tikzpicture}
```

图 1.1.24 几种渐变样式

尝试绘制新的圆弧



```
\begin{tikzpicture}[scale = 3]
    \clip (-0.1,-0.2) rectangle (1.1,0.75);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \shadedraw[left color=gray,right color=green, draw=green
        !50!black] (0,0) -- (3mm,0mm)arc [start angle=0, end
        angle=30, radius=3mm] -- cycle;
\end{tikzpicture}
```

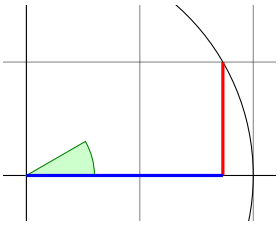
图 1.1.25 带渐变的圆弧

1.8 指定坐标

TikZ 指定坐标需要依靠数学计算，主要有两种方式

1. 平面直角坐标系：例如：(10pt,2cm) 代表 x 方向偏移 10pt, y 方向偏移 2cm
2. 极坐标系：例如：(30:1cm) 代表，逆时针选择 30°, 长度为 1cm

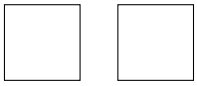
在确定坐标之后便可以依照新的点绘制图像，例如 +(0cm,1cm) 代表在坐标上方绘制 1cm 长线，++(2cm,0cm) 则再次绘制线段



```
\begin{tikzpicture}[scale = 3]
    \clip (-0.1,-0.2) rectangle (1.1,0.75);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \draw (-1.5,0) -- (1.5,0);
    \draw (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm)
        arc [start angle=0, end angle=30, radius=3mm] -- cycle;
    \draw[red,very thick] (30:1cm) -- +(0,-0.5);
    \draw[blue,very thick] (30:1cm) ++(0,-0.5) -- (0,0);
\end{tikzpicture}
```

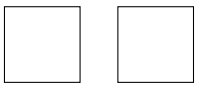
图 1.1.26 指定坐标绘制线段

从上述的例子可以发现，– 承担了指定线段类型的责任，若没有 – 则不会绘制线段，可以将这种用法用在定位点上，下面用几个例子详细说明 + 与 ++ 的作用



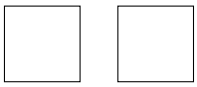
```
\begin{tikzpicture}[scale = 1]
    \def\rectanglepath{-- ++(1cm,0cm) -- ++(0cm,1cm) -- ++(-1cm
        ,0cm) -- cycle}
    \draw (0,0) \rectanglepath;
    \draw (1.5,0) \rectanglepath;
\end{tikzpicture}
```

图 1.1.27 ++ 绘制矩形



```
\begin{tikzpicture}[scale = 1]
    \def\rectanglepath{-- +(1cm,0cm) -- +(0cm,1cm) -- +(-1cm,0
        cm) -- cycle}
    \draw (0,0) \rectanglepath;
    \draw (1.5,0) \rectanglepath;
\end{tikzpicture}
```

图 1.1.28 + 绘制矩形



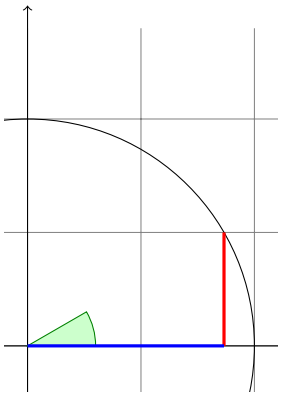
```
\tikz \draw (0,0) rectangle +(1,1) (1.5,0) rectangle +(1,1);
```

图 1.1.29 更快速的矩形绘制方式

1.9 线段与箭头

1.9.1 箭头样式

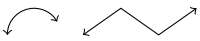
使用 -> 代替 – 绘制有箭头的坐标轴



```
\begin{tikzpicture}[scale = 3]
   \clip (-0.1,-0.2) rectangle (1.1,1.51);
   \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
   \draw[->] (-1.5,0) -- (1.5,0);
   \draw[->] (0,-1.5) -- (0,1.5);
   \draw (0,0) circle [radius=1cm];
   \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm
        ,0mm) arc [start angle=0, end angle=30, radius=3mm] --
         cycle;
   \draw[red,very thick] (30:1cm) -- +(0,-0.5);
   \draw[blue,very thick] (30:1cm) ++(0,-0.5) -- (0,0);
\end{tikzpicture}
```

图 1.1.30 带箭头的坐标轴

可以在 draw 可选参数中使用箭头来指明箭头



```
\begin{tikzpicture}[scale = 1]
   \draw [<->] (0,0) arc [start angle = 180,end angle = 30,
        radius=10pt];
   \draw [<->] (1,0) -- (1.5cm,10pt) -- (2cm,0pt) -- (2.5cm,10
        pt);
\end{tikzpicture}
```

图 1.1.31 箭头示例

使用不一样的箭头样式

```
\usetikzlibrary {arrows.meta}
\begin{tikzpicture}[scale = 1,>=Stealth]
   \draw [->] (0,0) arc [start angle=180, end angle=30,radius
        =10pt];
   \draw [<<-,very thick] (1,0) -- (1.5cm,10pt) -- (2cm,0pt)
        -- (2.5cm,10py);
\end{tikzpicture}
```

图 1.1.32 箭头样式

1.9.2 范围 scope

对于经常出现的同类样式，可以用 scope 设为统一样式

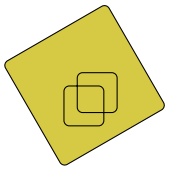
```
\begin{tikzpicture}[scale = 1,ultra thick]
   \draw (0,0) -- (0,1);
   \begin{scope}[thin]
      \draw (1,0) -- (1,1);
      \draw (2,0) -- (2,1);
   \end{scope}
   \draw (3,0) -- (3,1);
\end{tikzpicture}
```

图 1.1.33 scope 的作用

scope 还有一个重要作用，在 scope 内的操作只会影响内部范围

1.10 位移

可以使用 shift 关键字调整位置



```
\begin{tikzpicture}[scale = 1,rounded corners=2pt,x=15pt,y=15pt]
   \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}
```

图 1.1.34 位移示范

1.11 循环

1.11.1 数字循环

使用 foreach 指令，进行数字循环

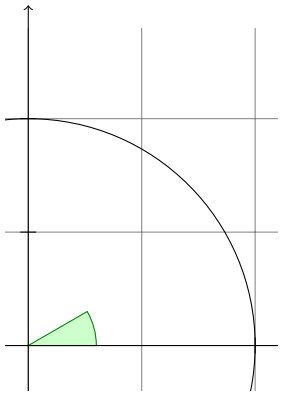
```
\foreach <variable> in <values> <commands>
```

$x = 1$，$x = 2$，$x = 3$，

```
\begin{tikzpicture}[scale = 1]
   \foreach \x in {1,2,3} {$x = \x$,}
\end{tikzpicture}
```

图 1.1.35 foreach 数字循环

同样，我们可以将其应用到绘图中



```
\begin{tikzpicture}[scale = 3]
   \clip (-0.1,-0.2) rectangle (1.1,1.51);
   \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);
   \draw[->] (0,-1.5) -- (0,1.5);
   \draw (0,0) circle [radius=1cm];
   \foreach \x in {-1cm,-0.5cm,1cm} \draw (\x,-1pt) -- (\x,1pt);
   \foreach \y in {-1cm,-0.5cm,0.5cm,1cm} \draw (-1pt,\y) -- (1pt,\y);
\end{tikzpicture}
```

图 1.1.36 循环绘制刻度线

再进一步，我们可以设计两个变量，绘制二维矩阵

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	12,5
7,4	8,4	9,4	10,4	11,4	12,4
7,3	8,3	9,3	10,3	11,3	12,3
7,2	8,2	9,2	10,2	11,2	12,2
7,1	8,1	9,1	10,1	11,1	12,1

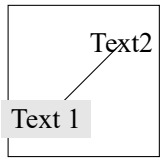
```
\begin{tikzpicture}[scale = 1]
   \foreach \x in {1,2,...,5,7,8,...,12}
      \foreach \y in {1,...,5} {
      \draw (\x,\y) +(-.5,-.5) rectangle ++(.5,.5);
      \draw (\x,\y) node{\x,\y};
      }
\end{tikzpicture}
```

图 1.1.37 二维矩阵

1.12 增加文字

1.12.1 node 关键字

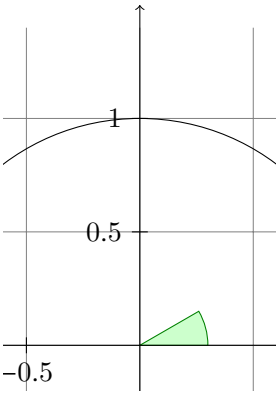
TikZ 可以通过 node 关键字指定节点并绘制区局，以达到添加文字的效果




```
\begin{tikzpicture}[scale = 1]
   \draw (0,0) rectangle (2,2);
   \draw (0.5,0.5) node [fill=grey!20] {Text 1} -- (1.5,1.5)
        node {Text2}
\end{tikzpicture}
```

图 1.1.38 node 关键字的作用

默认将以 node 为几何中心构建方形区域，但也可自定义文字位置

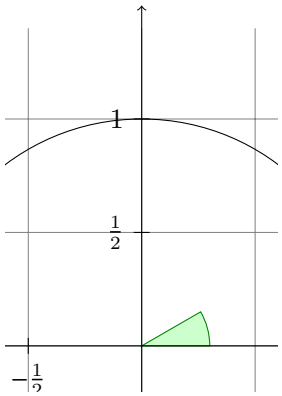



```
\begin{tikzpicture}[scale = 3]
   \clip (-0.6,-0.2) rectangle (0.6,1.51);
   \draw[step=.5cm,help lines] (-1.4,-1.4) grid (1.4,1.4);
   \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm)
        arc [start angle=0, end angle=30, radius=3mm] -- cycle;
   \draw[->] (-1.5,0) -- (1.5,0); \draw[->] (0,-1.5) -- (0,1.5);
   \draw (0,0) circle [radius=1cm];
   \foreach \x in {-1,-0.5,1}
      \draw (\x cm,1pt) -- (\x cm,-1pt) node[anchor=north] {$\x$};
   \foreach \y in {-1,-0.5,0.5,1}
      \draw (1pt,\y cm) -- (-1pt,\y cm) node[anchor=east] {$\y$};
\end{tikzpicture}
```

图 1.1.39 绘制坐标系制度

如果需要使用分数形式的文本，可以使用如下形式

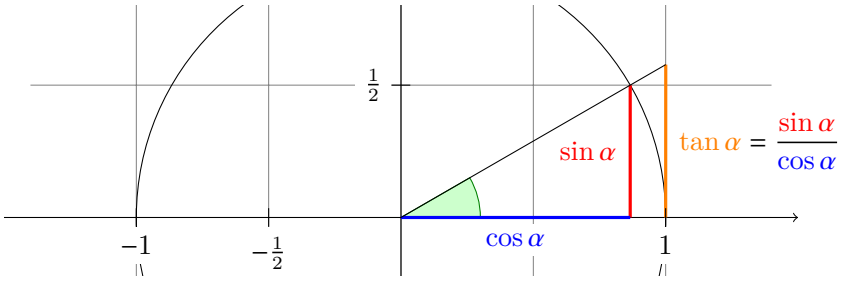
```
\x / \xtex
```




```
\begin{tikzpicture}[scale = 3]
    \clip (-0.6,-0.2) rectangle (0.6,1.51);
    \draw[step=.5cm,help lines] (-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); \draw[->] (0,-1.5) -- (0,1.5);
    \draw (0,0) circle [radius=1cm];
    \foreach \x/\xtext in {-1, -0.5/-\frac{1}{2}, 1}
       \draw (\x cm,1pt) -- (\x cm,-1pt) node[anchor=north] {$\xtext$};
    \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=east] {$\ytext$};
\end{tikzpicture}
```

图 1.1.40 绘制坐标系制度 (分数形式)

接着之前的直角坐标系，我们可以标出函数 (以下为官方代码，出现了部分前面未说明的内容)

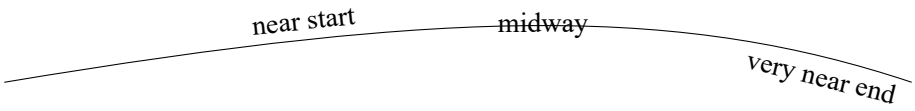



```
\begin{tikzpicture}[scale=3.5]
    \clip (-2,-0.22) rectangle (2,0.8);
    \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
    \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm) arc [start angle=0, end angle=30,
         radius=3mm] -- cycle;
    \draw[->] (-1.5,0) -- (1.5,0) 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);
    \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);
    \path [name path=sloped line] (0,0) -- (30:1.5cm);
    \draw [name intersections={of=upward line and sloped line, by=t}] [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}
       \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=east,fill=white] {$\ytext$};
\end{tikzpicture}
```

图 1.1.41 带文字的单位圆

1.12.2 线段文字

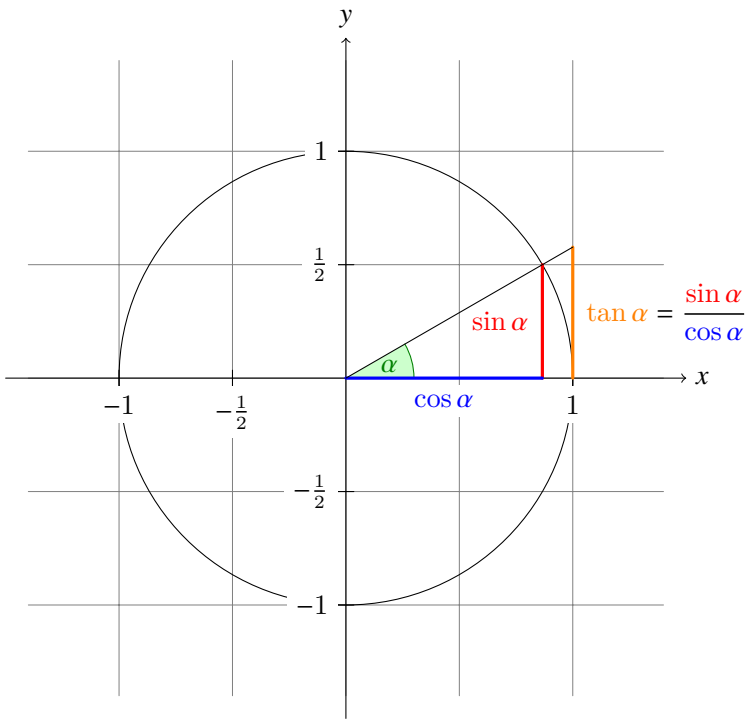
可以通过 sloped 关键字指定在线段的某一区域添加文字




```
\begin{tikzpicture}[scale = 1]
    \draw (0,0) .. controls (6,1) and (9,1) ..
        node [near start,sloped,above] {near start}
        node {midway}
        node [very near end,sloped,below] {very near end}(12,0);
\end{tikzpicture}
```

图 1.1.42 sloped 控制段落文字位置

1.13 最终结果



The angle α is $30°$ in the example ($\pi/6$ in radians). The sine of α, which is the height of the red line, is

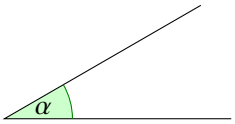
$$\sin\alpha = 1/2.$$

By the Theorem of Pythagoras ...

图 1.1.43

1.14 其他技巧

可以使用 coordinate 指定点位置并赋予别名，并通过 pic 来进行快速绘图，pic 这里只做演示，详细内容后续章节会讲解




```
\usetikzlibrary {angles,quotes}
\begin{tikzpicture}[scale = 3]
    \coordinate (A) at (1,0);
    \coordinate (B) at (0,0);
    \coordinate (C) at (30:1cm);
    \draw (A) -- (B) -- (C)
        pic [draw=green!50!black, fill=green!20, angle radius=9mm,
            "$\alpha$"] {angle = A--B--C};
\end{tikzpicture}
```

图 1.1.44 使用 pic 快速绘制

参考文献