

# TỰ HỌC *TikZ* CĂN BẢN

Phùng Gia Luân

07.June.2014

# Mục lục

<b>1</b>	<b>Những điều cơ bản</b>	<b>2</b>
1.1	Hình vẽ cho sinh viên của thầy Karl	2
1.1.1	Vấn đề cần giải quyết	2
1.1.2	Cài đặt môi trường TikZ trong L <sup>A</sup> T <sub>E</sub> X	2
1.1.3	Vẽ đường thẳng	3
1.1.4	Vẽ đường cong	3
1.1.5	Vẽ đường tròn và elip	4
1.1.6	Vẽ hình chữ nhật	5
1.1.7	Vẽ hình lưới	6
1.1.8	Sử dụng <i>Style</i> quản lý các thông số điều chỉnh	7
1.1.9	Các thông số tùy chọn vẽ	8
1.1.10	Vẽ cung	8
1.1.11	Biểu diễn một phần của hình vẽ	9
1.1.12	Vẽ parabol, đường cong sin và cosin	10
1.1.13	Tô màu và vẽ liền nét	11
1.1.14	Tạo hiệu ứng shading	12
1.1.15	Xác định tọa độ	13
1.1.16	Vẽ các đường giao nhau	14
1.1.17	Vẽ mũi tên	15
1.1.18	Môi trường Scope	15
1.1.19	Sự thay đổi vị trí	16
1.1.20	Vòng lặp	17
1.1.21	Thêm chữ trong hình vẽ	18

# Lời nói đầu

Để vẽ hình, người dùng L<sup>A</sup>T<sub>E</sub>X có thể sử dụng PSTrick, MetaPost ... nhưng từ khi gói PGF được viết bởi Till Tantau (2006), một phần lớn đã chuyển sang dùng TikZ vì quá trình gõ nhanh chóng với cú pháp lệnh đơn giản. Trong quá trình tự học, tôi biên soạn tài liệu này từ *pgfmanual*, và dùng kiến thức vừa học để vẽ một số hình liên quan đến công việc. Do đó, tài liệu này có thể sẽ hữu ích với những ai mới tìm hiểu TikZ, học theo cách *Learning by doing*, hoặc học qua việc phân tích mã lệnh. Khi đã quen, chúng ta tìm hiểu TikZ sẽ dễ dàng và nhanh chóng hơn.

Mọi ý kiến đóng góp về tài liệu này vui lòng gửi về địa chỉ email: [gialuannb@gmail.com](mailto:gialuannb@gmail.com).

Xin trân trọng cảm ơn sự góp ý của bạn!

***Nào, mình cùng chơi với TikZ...***

# Chương 1

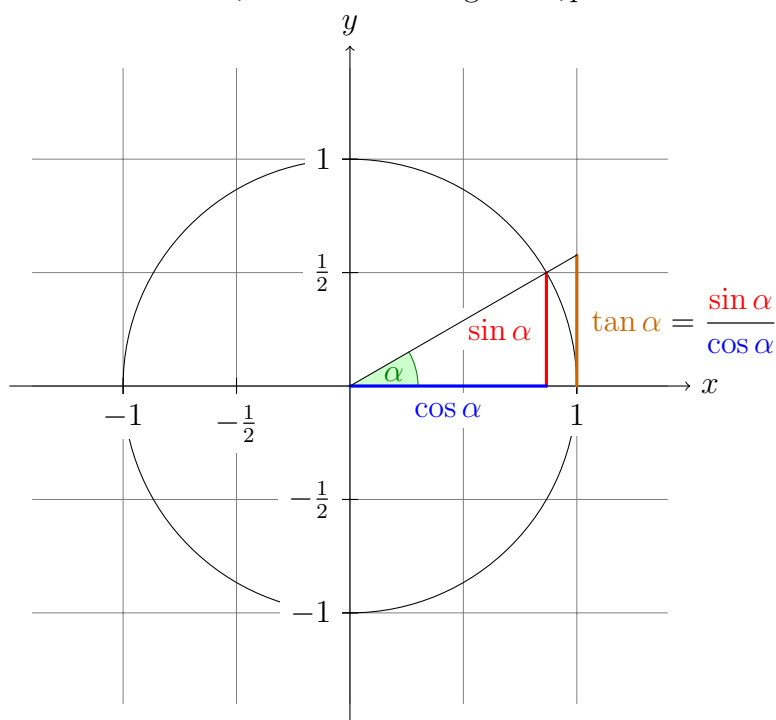
## Những điều cơ bản

Phần này đề cập đến những điều cơ bản nhất, không đi sâu vào chi tiết, chủ yếu là các hướng dẫn cách vẽ một hình cho trước thông qua các dự án đơn giản. Qua các dự án này, chúng ta sẽ quen dần và tự vẽ được những hình vẽ đầu tiên với *TikZ*.

### 1.1 Hình vẽ cho sinh viên của thầy Karl

#### 1.1.1 Vấn đề cần giải quyết

Karl muốn có một hình vẽ ở trong bài tập cho sinh viên như sau:



Góc  $\alpha$  là  $30^\circ$  trong ví dụ ( $\pi/6$  nếu đơn vị đo là radian). Giá trị  $\sin \alpha$  là:

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

Theo định lý Pitago, ta có  $\cos^2 \alpha + \sin^2 \alpha = 1$ . Do đó độ dài của đường màu xanh là giá trị của  $\cos \alpha$ , phải là

$$\cos \alpha = \sqrt{1 - 1/4} = \frac{1}{2}\sqrt{3}.$$

Điều này cho ta độ dài đường màu cam  $\tan \alpha$  có giá trị là

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = 1/\sqrt{3}.$$

#### 1.1.2 Cài đặt môi trường *TikZ* trong $\LaTeX$

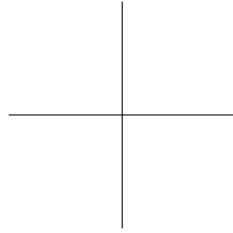
Karl dùng  $\LaTeX$  tạo một file *.tex* có nội dung như sau:

```

\documentclass{article}
\usepackage{tikz}
\begin{document}
We are working on
\begin{tikzpicture}
\draw (-1.5,0) -- (1.5,0);
\draw (0,-1.5) -- (0,1.5);
\end{tikzpicture}
\end{document}

```

Khi biên dịch bởi pdflatex hay latex, ta sẽ được kết quả như sau:



We are working on

Tất nhiên đây chưa phải là hình vẽ mà Karl cần nhưng chúng ta đã vẽ được hai đường thẳng vuông góc với nhau cần thiết để tạo hệ trục tọa độ.

Nghiên cứu chi tiết hơn về code. Trước hết, gói tikz được loaded. Sau đó là hai lệnh vẽ `\draw` nhằm vẽ một đường thẳng từ tọa độ  $(-1.5,0)$  tới tọa độ  $(1.5,0)$  và vẽ một đường thẳng khác từ tọa độ  $(0,-1.5)$  tới tọa độ  $(0,1.5)$ . Kết thúc mỗi lệnh luôn là dấu chấm phẩy (;)

### 1.1.3 Vẽ đường thẳng

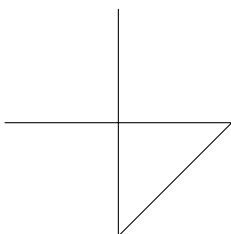
Yếu tố cơ bản trong mọi hình ảnh được vẽ bởi TikZ là đường. Đường ở đây được hiểu là đường thẳng hoặc đường cong. Bạn bắt đầu một đường tại một điểm được chỉ ra bởi tọa độ của nó. Ví dụ  $(0,0)$ .

Theo sau tọa độ là các kí hiệu chỉ loại đường. Tiếp sau kí hiệu chỉ loại đường là một tọa độ khác để chỉ điểm cuối của đường đó. Để vẽ đường thẳng chúng ta dùng kí hiệu - -

Cú pháp lệnh vẽ đường thẳng:

```
\draw (tọa độ 1) - - (tọa độ 2);
```

Một ví dụ đơn giản khác về vẽ đường thẳng:



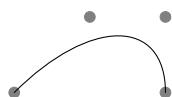
```
\tikz \draw (-1.5,0) -- (1.5,0) -- (0,-1.5) -- (0,1.5);
```

### 1.1.4 Vẽ đường cong

TikZ cung cấp một cú pháp lệnh đặc biệt để vẽ đường cong: một hoặc hai "control point" (điểm điều chỉnh sự cong) cần phải có để xác định đường cong đó. Thuật toán đơn giản để vẽ đường cong: giả sử bạn bắt đầu ở điểm  $x$  và điểm điều chỉnh control point  $y$  đầu tiên.

Điều này có nghĩa tiếp tuyến của đường cong tại x sẽ đi qua điểm y. Tiếp theo, giả sử đường cong kết thúc tại điểm z với điểm điều chỉnh control point w. Khi đó, tiếp tuyến của đường cong tại điểm z sẽ đi qua w.

Dưới đây là một ví dụ trong đó điểm điều chỉnh control point được thêm vào để minh họa rõ hơn:



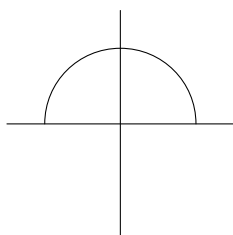
```
\begin{tikzpicture}
\filldraw [gray] (0,0) circle (2pt)
(1,1) circle (2pt)
(2,1) circle (2pt)
(2,0) circle (2pt);
\draw (0,0) .. controls (1,1) and (2,1) .. (2,0);
\end{tikzpicture}
```

Cú pháp lệnh vẽ đường cong:

`\draw (tọa độ đầu) .. controls (control point1) and (control point2) ..(tọa độ cuối);`

Lưu ý: có thể bỏ phần: **and (control point2)** trong cấu trúc lệnh nếu control point1 chính là control point2.

Karl đã có thể vẽ nửa vòng tròn bằng lệnh vẽ đường cong vào hệ trục tọa độ như sau:



```
\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}
```

Kết quả có thể làm Karl hài lòng nhưng vẽ đường tròn theo cách này tương đối dài dòng. Có cách khác ngắn gọn hơn là dùng lệnh vẽ đường tròn riêng.

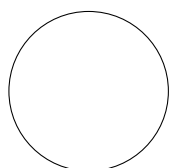
### 1.1.5 Vẽ đường tròn và elip

Một đường tròn xác định khi ta biết tọa độ tâm (x,y) và bán kính r, do vậy cú pháp lệnh vẽ đường tròn như sau:

`\draw (x,y) circle (r);`

Với **x**, **y**, **r** là các số, sau số **r** có thể để trống hoặc thêm đơn vị của **r** như cm, pt ...

Ví dụ:



```
\begin{tikzpicture}
\draw (0,0) circle (30pt);
\end{tikzpicture}
```

Tương tự đường tròn, để vẽ một hình elip ta cần biết tọa độ tâm (x,y) và hai bán kính a, b. Do vậy cú pháp lệnh vẽ elip là:

---

```
\draw (x,y) ellipse (a and b);
```

Ví dụ:



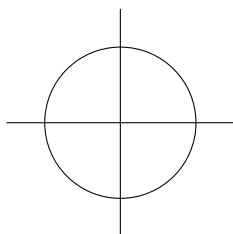
```
\begin{tikzpicture}
\draw (0,0) ellipse (30pt and 10pt);
\end{tikzpicture}
```

Để vẽ một hình elip lệch một góc so với trục tọa độ, ta dùng thêm tùy chọn *rotate* như sau:



```
\begin{tikzpicture}
\draw[rotate=30] (0,0) ellipse (30pt and 15pt);
\end{tikzpicture}
```

Quay trở lại với hình vẽ của Karl, ta dùng lệnh vẽ đường tròn có bán kính 1cm thay vì dùng lệnh vẽ đường cong. Mã lệnh TikZ là:



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

Để có hình vẽ cuối cùng với độ lớn phù hợp, Karl cần dùng thêm một tham số tùy chọn đơn giản nhưng chúng ta sẽ không sử dụng ở đây nhằm trình bày hình vẽ cùng với mã TikZ ngắn gọn, hợp với khổ giấy hơn.

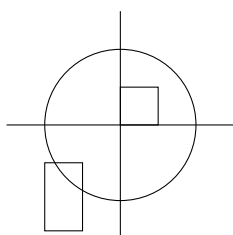
### 1.1.6 Vẽ hình chữ nhật

Có một vài cách để vẽ hình lưới nền trong hình vẽ của Karl. Theo cách nhìn đơn giản hình lưới nền là tập hợp nhiều hình chữ nhật. Hình chữ nhật là một trong những hình phổ biến trong hình học sơ cấp nên có một lệnh riêng. Lệnh này bắt đầu từ vị trí góc 1 có tọa độ  $(x_1, y_1)$  của hình chữ nhật và vị trí góc có tọa độ  $(x_2, y_2)$  đối diện với góc 1.

Cú pháp lệnh vẽ hình chữ nhật là:

```
\draw (x1,y1) rectangle (x2,y2);
```

Ví dụ:



```
\begin{tikzpicture}
\draw (-1.5,0) -- (1.5,0);
\draw (0,-1.5) -- (0,1.5);
\draw (0,0) circle (1cm);
\draw (0,0) rectangle (0.5, 0.5);
\draw (-0.5,-0.5) rectangle (-1,-1.4);
\end{tikzpicture}
```

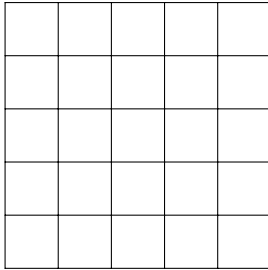
Trong nhiều trường hợp lệnh vẽ hình chữ nhật rất hữu dụng, nhưng không nên dùng lệnh

này để vẽ hình lưới nền trong hình vẽ của Karl vì phải vẽ một số lượng lớn hình chữ nhật và hơn nữa một số hình chữ nhật còn thiếu một cạnh hoặc cạnh bị hở. Giải pháp tốt hơn là dùng lệnh `grid` sau đây để vẽ hình lưới.

### 1.1.7 Vẽ hình lưới

Lệnh `grid` là lệnh vẽ hình lưới. Có thể dùng lệnh này để vẽ một hình chữ nhật lưới theo cách bắt đầu từ tọa độ một góc và góc còn lại của hình chữ nhật lưới được xác định theo cú pháp của lệnh `grid`.

Ví dụ:



```
\tikz \draw[step=20pt] (0,0) grid (100pt,100pt);
```

Cú pháp lệnh vẽ hình lưới là:

```
\draw[xstep=x, ystep=y, c, w] (x1,y1) grid (a,b);
```

Trong đó:

$x$  là số chỉ chiều dài của ô cơ sở hình lưới theo trục hoành.

$y$  là số chỉ chiều cao của ô cơ sở hình lưới theo trục tung. Sau  $x$ ,  $y$  có thể bỏ trống (mặc định đơn vị là cm) hoặc có đơn vị (cm, pt ...). Nếu  $x=y$  thì chỉ cần dùng tùy chọn **step** giống như ví dụ trên.

$c$ : (color) màu sắc của đường kẻ (gray, blue, lightgray, red, ...).

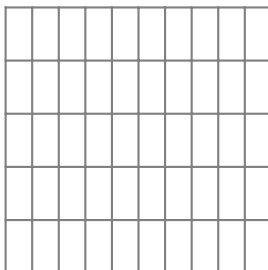
$w$ : (width) tùy chọn độ rộng của đường kẻ gồm: *ultra thin* (0,1pt), *very thin* (0,2pt), *thin* (0,4pt), *semithick* (0,6pt), *thick* (0,8pt), *very thick* (1,2pt), *ultra thick* (1,6pt). Nếu không sử dụng tùy chọn này,  $w$  sẽ tự mặc định là *thin* (0,4pt) giống ví dụ trên.

$(x1,y1)$ : điểm góc ban đầu của lưới.

$(a,b)$ : điểm góc cuối của lưới.

Lệnh `grid` tạo hình lưới bằng cách vẽ các đường thẳng (ngang và dọc) trong đó điểm góc đầu tiên của hình chữ nhật đầu tiên tạo bởi các đường đó, là điểm được xác định bởi cặp số  $(x1,y1)$ .

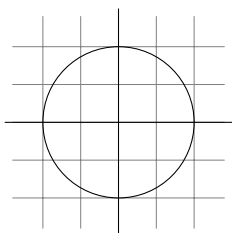
Ví dụ:



```
\draw[xstep=10pt, ystep=20pt, gray, thick] (0,0) grid (100pt,100pt);
```



Với hình vẽ của Karl, ta thêm hình lưới và làm mờ hình lưới bằng tùy chọn *w* là *very thin* như sau:

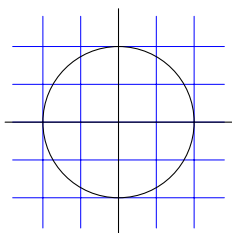


```
\begin{tikzpicture}
\draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
\draw (-1.5,0) -- (1.5,0);
\draw (0,-1.5) -- (0,1.5);
\draw (0,0) circle (1cm);
\end{tikzpicture}
```

### 1.1.8 Sử dụng *Style* quản lý các thông số điều chỉnh

Thay vì sử dụng các tùy chọn điều chỉnh trong lệnh `\draw` ở phần 1.1.7 như: *xstep*, *ystep*, *step*, *color*, *width* với các tùy chọn như *gray*, *very thin* ..., chúng ta có thể đặt tất cả các điều chỉnh này vào một *Style*. Một *Style* là một tập hợp những tùy chọn điều chỉnh được sử dụng để quy định chi tiết cách vẽ trước khi chúng ta vẽ hình.

Ví dụ: mã lệnh với *Styles* [help lines](#):



```
\begin{tikzpicture}
% Styles
[help lines/.style={step=0.5cm,color=blue,very thin}]
\draw[help lines] (-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 (1cm);
\end{tikzpicture}
```

Việc sử dụng *Style* có tác dụng làm cho mã lệnh vẽ của bạn ngắn gọn, linh hoạt. Hơn nữa, từ việc chỉnh sửa thông số trong *Style* có thể tạo ra sự thay đổi đồng loạt với các lệnh vẽ có sử dụng *Style* đó. Thông thường, các *Style* được đặt ở phần đầu của mỗi hình vẽ nhưng nếu bạn muốn tất cả mọi hình vẽ trong văn bản của bạn sử dụng chung một *Style* thì cần đặt lệnh `\tikzset` ở phần đầu của văn bản tương tự như ví dụ sau:

```
\begin{document}
\tikzset{help lines/.style={step=0.5cm,color=blue,very thin}}
\begin{tikzpicture}
\draw[help lines] (0,0) -- (2,2);
\draw[help lines] (0,0) circle (1cm)
\end{tikzpicture}

\begin{tikzpicture}
\draw[help lines] (0,0) rectangle (1,2);
\draw[help lines] (0,0) grid (10pt,10pt)
\end{tikzpicture}
\end{document}
```

Ở mức độ cao hơn nữa, bạn có thể tạo hệ thống các *Style*. Các *Style* sau có thể bao gồm các *Style* trước.

---

Ví dụ:

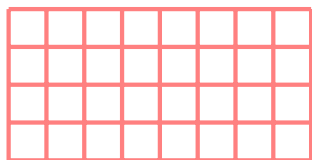
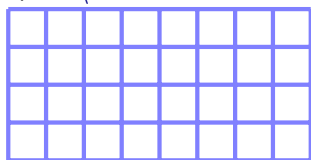
```
\tikzset{help lines/.style=very thin}
\tikzset{Karl's grid/.style={help lines, color=blue}}
...
\tikz \draw[Karl's grid] (0,0) grid (5,5);
```

Với sự mềm dẻo trong việc chỉnh sửa các thông số nên các *Style* rất hữu dụng. Điều này thể hiện ở chỗ có thể dùng *Style* để điều chỉnh một thông số trong một hình vẽ riêng, trong khi các hình vẽ khác vẫn tuân theo mặc định chung của *Style* đó.

Ví dụ:

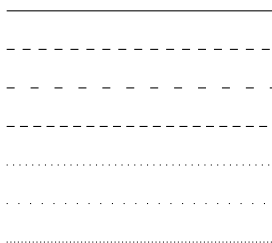
```
\tikzset{help lines/.style=ultra thick,step=0.5cm}
\tikzset{Karl's grid/.style={help lines, color=#1!50}, Karl's grid/.default=blue}
\tikz \draw[Karl's grid] (0,0) grid (4,2);\hspace{2cm}
\tikz \draw[Karl's grid=red] (0,0) grid (4,2);
```

Trong ví dụ này, Karl đặt thông số màu cho hình lưới nền là **blue** với mọi hình vẽ nhưng có một hình vẽ Karl muốn màu của hình lưới nền là **red** do đó điều chỉnh thông số màu **red** ở lệnh `\draw` thứ hai.



### 1.1.9 Các thông số tùy chọn vẽ

Chúng ta đã biết các tùy chọn về màu sắc (tùy chọn *c*), độ dày của đường kẻ (tùy chọn *width*: *ultra thin* (0,1pt), *very thin* (0,2pt), *thin* (0,4pt), *semithick* (0,6pt), *thick* (0,8pt), *very thick* (1.2pt), *ultra thick* (1.6pt)), ngoài ra còn có một thông số quan trọng nữa là loại đường kẻ bao gồm: nét liền (giống như chúng ta đã vẽ), nét đứt (dashed -----), nét chấm (dotted .....). Riêng với nét đứt (dashed) và nét chấm (dotted) chúng ta cũng có hai tùy chọn khác là *loosely dashed*, *densely dashed* và *loosely dotted*, *densely dotted*. Hãy so sánh các kiểu đường theo thứ tự: nét liền, nét đứt (dashed, loosely dashed, densely dashed), nét chấm (dotted, loosely dotted, densely dotted) trong ví dụ sau:

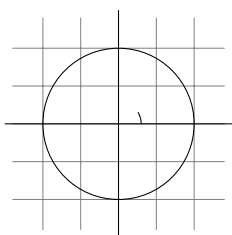


### 1.1.10 Vẽ cung

Một cung (**arc**) là một phần của đường tròn hoặc elip. Lệnh vẽ cung tròn như sau:

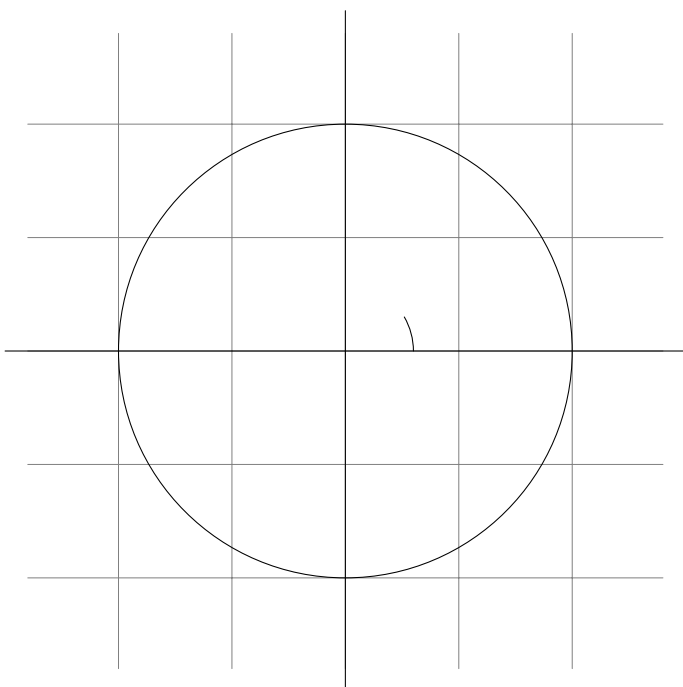
```
\draw (x,y) arc (a:b:r);
```

Trong đó:  $(x,y)$  là vị trí bắt đầu vẽ;  $a, b$  là giá trị các góc (đơn vị là độ);  $r$  là bán kính. Lệnh này có thể diễn giải như sau: tại vị trí  $(x,y)$  vẽ một cung tròn có bán kính  $r$  từ  $a$  độ tới  $b$  độ. Ví dụ:



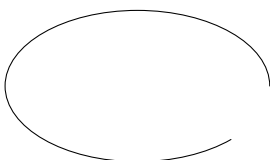
```
\begin{tikzpicture}
\draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
\draw (-1.5,0) -- (1.5,0);
\draw (0,-1.5) -- (0,1.5);
\draw (0,0) circle (1cm);
\draw (3mm,0mm) arc (0:30:3mm);
\end{tikzpicture}
```

Để hình vẽ lớn hơn, Karl cần sử dụng tùy chọn **scale=3**. Có thể sử dụng tùy chọn này cho mỗi lệnh `\draw` nhưng nếu muốn tất cả đều lớn gấp 3 lần thì tốt nhất là sử dụng tùy chọn này cho cả môi trường `tikzpicture`.



```
\begin{tikzpicture}[scale=3]
\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 (1cm);
\draw (3mm,0mm) arc (0:30:3mm);
\end{tikzpicture}
```

Tương tự cung tròn, để vẽ cung elip cần hai giá trị bán kính. Ví dụ:



```
\begin{tikzpicture}
\draw (0,0) arc (0:315:1.75cm and 1cm);
\end{tikzpicture}
```

Cú pháp lệnh vẽ cung elip như sau:

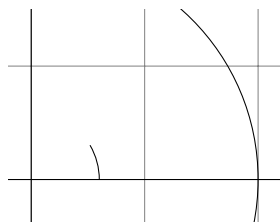
```
\draw (x,y) arc (a:b:r1:r2);
```

### 1.1.11 Biểu diễn một phần của hình vẽ

Để tiết kiệm giấy sẽ là tốt hơn nếu chúng ta chỉ tập trung vào một phần hình cần vẽ quan tâm. Biểu diễn một phần hình vẽ rất đơn giản với TikZ bằng cách dùng lệnh `\clip`. Lệnh này tương tự như lệnh `\draw` nhưng khác biệt ở chỗ lệnh `\clip` vẽ một hình bao và hệ thống sẽ chỉ hiển thị những gì ở trong hình bao mà thôi.

Ví dụ:

Lệnh `\clip` dưới đây tạo ra một hình bao là hình chữ nhật với góc đầu tiên là  $(-0.1, -0.2)$ , góc đối diện là  $(1.1, 0.75)$ . Hình vẽ hiển thị sau khi biên dịch bởi  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  là một phần đã được cắt bởi hình bao ấy.



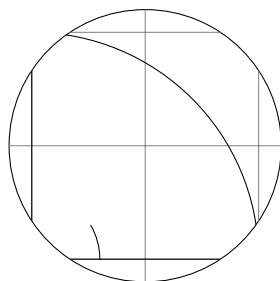
```
\begin{tikzpicture}[scale=3]
\clip (-0.1,-0.2) rectangle (1.1,0.75);
\draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid (1.4,1.4);
\draw (-1.5,0) -- (1.5,0);
\draw (0,-1.5) -- (0,1.5);
\draw (0,0) circle (1cm);
\draw (3mm,0mm) arc (0:30:3mm);
\end{tikzpicture}
```

Điều thú vị là có thể coi `clip` là thông số tùy chọn của lệnh `draw` hoặc ngược lại. Đồng thời cả `clip` và `draw` đều là thông số tùy chọn của lệnh `\path`.

Ví dụ:

```
\clip[draw] ...
\draw[clip] ...
\path[clip] ...
\path[draw] ...
\path[clip,draw] ...
```

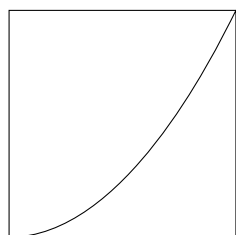
Chúng ta dùng `draw` làm thông số tùy chọn cho lệnh `\clip` như sau:



```
\begin{tikzpicture}[scale=3]
\clip[draw] (0.5,0.5) circle (0.6cm);
\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 (1cm);
\draw (3mm,0mm) arc (0:30:3mm);
\end{tikzpicture}
```

### 1.1.12 Vẽ parabol, đường cong sin và cosin

Mặc dù hình của Karl không cần vẽ toàn bộ đường cong sin, cosin hoặc parabol nhưng có thể dùng lệnh vẽ một phần các loại đường cong này.



```
\draw (0,0) rectangle (1,1) (0,0) parabola (1,1);
```

Chúng ta cũng có thể vẽ một phần đường cong tại vị trí bất kì:



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

Mã lệnh `sin` và `cos` vẽ một đường cong sin và cos tương ứng trong khoảng  $(0, \pi/2)$  từ tọa độ điểm đầu tới điểm kết thúc đường cong. Dưới đây là hai ví dụ:

Một đường cong sin



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

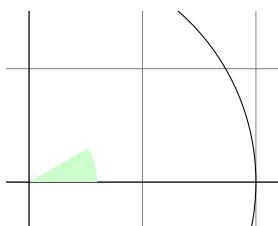
và:



```
\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);
```

### 1.1.13 Tô màu và vẽ liền nét

Để tô màu xanh nhạt phần góc ở tâm hình vẽ của Karl, ta dùng lệnh `\fill` thay cho lệnh `\draw`. Mã lệnh như sau:



```
\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 (1cm);
\fill[green!20!white] (0,0) -- (3mm,0mm)
arc (0:30:3mm) -- (0,0);
\end{tikzpicture}
```

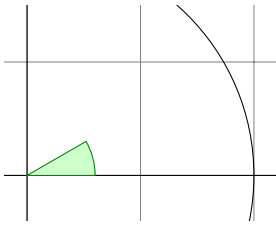
Trong đó `green!20!white` có nghĩa *20% green + 80% white*. Xem thêm cách pha màu ở góc `xcolor` của Uwe Kern.

Trong mã lệnh trên, tại dòng lệnh cuối ta sử dụng `|-(0,0)|` để kết thúc đường vẽ, sẽ tốt hơn nếu ta thay thế bằng `|-cycle|` vì `|-cycle|` có tác dụng vẽ liền nét khép kín từ điểm đầu đến điểm cuối. Hãy so sánh sự khác biệt sau:



```
\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}
```

Bạn cũng có thể vừa vẽ đường vừa tô màu với chỉ một lệnh là `\filldraw`, ban đầu sẽ vẽ đường bao sau đó là tô màu. Với lệnh này chúng ta có thể dùng màu tô khác với màu của đường vẽ như sau:



```
\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 (1cm);
\filldraw[fill=green!20!white, draw=green!50!black]
(0,0) -- (3mm,0mm) arc (0:30:3mm) -- cycle;
\end{tikzpicture}
```

### 1.1.14 Tạo hiệu ứng shading

Để tạo hiệu ứng shading (hiệu ứng bóng mờ) có thể dùng 1 trong 2 lệnh `\shade` hoặc `\shadedraw` với mặc định bóng mờ chuyển từ màu gray sang màu white.



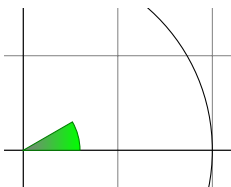
```
\begin{tikzpicture}[scale=1]
\tikz \shade (0,0) rectangle (2,1) (3,0.5) circle (.5cm);
\end{tikzpicture}
```

Nếu muốn dùng màu khác thì sử dụng tùy chọn màu:



```
\begin{tikzpicture}[rounded corners,ultra thick]
\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}
```

Tạo hiệu ứng shading tại phần cung màu xanh trong hình vẽ của Karl như sau:



```
\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 (1cm);
\shadedraw[left color=gray,right color=green, draw=green!50!black]
(0,0) -- (3mm,0mm) arc (0:30:3mm) -- cycle;
\end{tikzpicture}
```

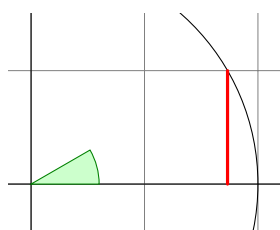
### 1.1.15 Xác định tọa độ

Để vẽ thêm các đường biểu thị sin và cos (2 đường thẳng màu xanh và đỏ) trong hình vẽ của thầy Karl tất nhiên phải sử dụng tùy chọn `color` ở trong thiết lập tùy chọn cho màu của đường thẳng nhưng cách tốt nhất để xác định chính xác tọa độ (vị trí) của đường thẳng này như thế nào?

Có nhiều cách xác định vị trí chính xác để vẽ đường thẳng. Cách dễ nhất là dùng tọa độ kiểu như (10pt, 2cm). Điều này có nghĩa 10pt ở hướng x và 2cm ở hướng y. Có thể bỏ đơn vị, ví dụ chỉ cần viết là (1,2) có nghĩa một lần giá trị véc tơ x và 2 lần giá trị véc tơ y. Giá trị mặc định của các véc tơ x và y đều là 1 cm.

Để chỉ ra điểm đặc biệt trong tọa độ cực, sử dụng kí hiệu (30:1cm) có nghĩa vị trí này cách 1cm với  $30^\circ$  tính từ phương ngang. Điều này rất hữu ích khi chỉ ra tọa độ một điểm ( $\cos 30^\circ, \sin 30^\circ$ ) trên vòng tròn.

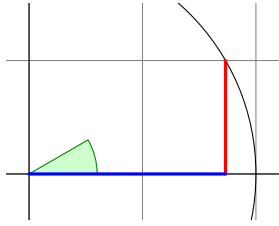
Khi thêm dấu + ở trước tọa độ kiểu như +(1cm,0cm) hoặc ++(0cm,2cm) có nghĩa là vị trí này cách 1cm về phía trước tại vị trí đã xác định tọa độ và cách 2cm về bên phải của vị trí tọa độ đã xác định. Ví dụ:



```
\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 (1cm);
  \filldraw[fill=green!20,draw=green!50!black]
    (0,0) -- (3mm,0mm) arc (0:30:3mm) -- cycle;
  \draw[red,very thick] (30:1cm) -- +(0,-0.5);
\end{tikzpicture}
```

Một cách để xác định vị trí giao của đường sin với trục x là sử dụng cú pháp (30:1cm |- 0,0). Nghĩa của cú pháp (p |- q) là tọa độ của điểm giao của đường thẳng qua p và một đường ngang qua q.

Để vẽ đường cos ta có thể dùng 1 trong 3 cách sau: cách 1 đơn giản nhất vẽ một đường thẳng từ vị trí tâm hình tròn tọa độ (0,0) với chiều dài bằng chiều dài của giá trị  $\cos 30^\circ = 0.866$ ; cách 2 là từ vị trí giao của đường sin với trục x, ta vẽ đường thẳng nối với tâm (0,0) bằng cách sử dụng cú pháp (30:1cm |- 0,0) - - (0,0); cách 3 là tiếp tục vẽ từ vị trí cuối của đường sin (là vị trí giao điểm trên) theo cú pháp (30:1cm)++(0,-0.5) - - (0,0). Theo cách thứ 3, trước tiên giá trị (30:1cm) cho phép ta đưa đầu bút vẽ tới vị trí cách 1cm ở cạnh góc  $30^\circ$  sau đó di chuyển bút vẽ tới vị trí đặc biệt ( $\cos 30^\circ$ ) mà không vẽ gì cả. Cuối cùng là từ điểm cuối của đường sin này vẽ một đường thẳng quay về vị trí tâm (0,0).

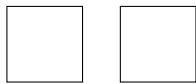


```

\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 (1cm);
  \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm)
  arc (0:30:3mm) -- cycle;
  \draw[red,very thick] (30:1cm) -- +(0,-0.5);
  \draw[blue,very thick] (30:1cm) ++(0,-0.5) -- (0,0);
\end{tikzpicture}

```

Lưu ý sự khác biệt giữa các kí hiệu + và ++.  
 Khi sử dụng kí hiệu ++

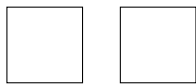


```

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

```

Và sử dụng kí hiệu +

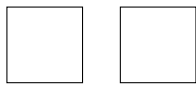


```

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

```

Hoặc mã lệnh tốt hơn nên viết như sau:



```

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

```

### 1.1.16 Vẽ các đường giao nhau

Để vẽ đường màu nâu  $\tan\alpha$  bắt đầu tại vị trí (1,0) và hướng lên trên tới điểm giao của nó với đường thẳng từ trục toạ độ qua điểm (30:1cm) trước hết cần tạo 2 *name path* khác nhau sao cho 2 *path* này giao nhau. Khi 2 *path* đã được tạo, ta sử dụng lệnh `\draw` hoặc lệnh `\fill` với thông số *name intersections* như sau:



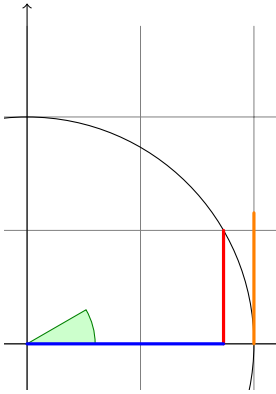
```

\path [name path=upward line] (1,0) -- (1,1);
\path [name path=sloped line] (0,0) -- (30:1.5cm);
% dài hơn một chút để chắc chắn có điểm giao
\draw [name intersections={of=upward line and sloped line, by=x}]
[very thick,orange] (1,0) -- (x);

```

### 1.1.17 Vẽ mũi tên

Để vẽ mũi tên ở hai trục tọa độ, chỉ cần thêm tùy chọn `->` trong lệnh `\draw`



```

\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 (1cm);
\filldraw[fill=green!20,draw=green!50!black]
(0,0) -- (3mm,0mm) arc (0:30:3mm) -- cycle;
\draw[red,very thick] (30:1cm) -- +(0,-0.5);
\draw[blue,very thick] (30:1cm) ++(0,-0.5) -- (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=x}]
[very thick,orange] (1,0) -- (x);
\end{tikzpicture}

```

Vẽ mũi tên với các tùy chọn có thể có `->`, `<-` hoặc `<->`. Hơn thế nữa, để vẽ các đầu mũi tên có hình dạng khác nhau, có thể sử dụng tùy chọn `>=right arrow tip kind` của môi trường `tikzpicture`. Các *right arrow tip kind* là một trong các tùy chọn `stealth`, `to`, `latex`, `space` (tham khảo thêm ở phần **Arrow Tip Library** trong *pgf manual*). Tuyệt vời hơn nữa, bạn có thể định nghĩa kiểu mũi tên của riêng mình.



```

\begin{tikzpicture}[>=stealth]
\draw [->] (0,0) arc (180:30:10pt);
\draw [<<- ,very thick] (1,0) -- (1.5cm,10pt)
-- (2cm,0pt) -- (2.5cm,10pt);
\end{tikzpicture}

```

### 1.1.18 Môi trường Scope

Môi trường `Scope` cho phép dùng các tùy chọn riêng chỉ áp dụng để vẽ các đối tượng nằm trong môi trường đó.



```
\begin{tikzpicture}[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}
```

Môi trường `scope` còn có tính chất đặc biệt: tùy chọn thay đổi được đặt bất kì chỗ nào trong môi trường `scope`. Bởi vậy thay vì viết `\draw[thin] (0,0) -- (1,1)` có thể viết là `\draw (0,0) [thin] -- (1,1)` hoặc `\draw (0,0) -- (1,1) [thin]`. Nếu dùng hai tùy chọn khác nhau là `[thin]` và `[thick]` thì tùy chọn nào đứng cuối cùng sẽ được áp dụng.

### 1.1.19 Sự thay đổi vị trí

Có một số cách để thay đổi vị trí của đối tượng vẽ trong TikZ. Ví dụ: tùy chọn `xshift` cho phép di chuyển tất cả đối tượng vẽ sau đó một khoảng cách cho trước.



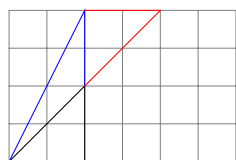
```
\tikz \draw (0,0) -- (0,0.5) [xshift=2pt] (0,0) -- (0,0.5);
```

Chú ý rằng bạn có thể thực hiện sự thay đổi vị trí ở giữa dòng lệnh như dưới đây:



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

Hai tùy chọn `xshift` và `yshift` rất hữu ích trong việc xác định sự thay đổi vị trí. Tùy chọn `rotate` để quay đối tượng vẽ với một góc cho trước. Tùy chọn `scale` cho phép phóng to hay thu nhỏ đối tượng, có thể thực hiện theo hướng x hoặc y (`xscale = -1` sẽ làm đối tượng lộn ngược). Tùy chọn `xslan`, `yslan` để tạo độ nghiêng cho đối tượng vẽ. Nếu các tùy chọn này chưa đủ hiệu quả bạn hãy thử với tùy chọn `cm` phức tạp (đọc thêm trong [PGF manual](#) tùy chọn `cm`)



```
\begin{tikzpicture}
\draw[help lines,step=.5cm,gray,very thin] (0,0) grid (3,2);
\draw (0,0) -- (1,1) -- (1,0);
\draw[cm={1,1,0,1,(0,0)},blue] (0,0) -- (1,1) -- (1,0);
\draw[cm={0,1,1,0,(1cm,1cm)},red] (0,0) -- (1,1) -- (1,0);
\end{tikzpicture}
```

## 1.1.20 Vòng lặp

Để tạo vòng lặp, dùng lệnh `\foreach`

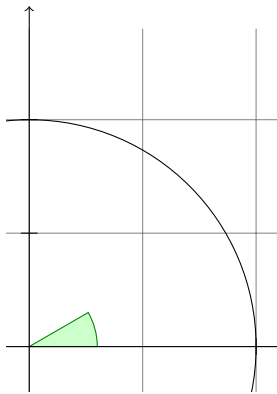
```
x = 1, x = 2, x = 3, \begin{tikzpicture}
\foreach \x in {1,2,3} {$x = \x, \; $ }
\end{tikzpicture}
```

Cú pháp của lệnh `\foreach` là:

`\foreach <biến> in {danh sách giá trị} <các lệnh có chứa biến>`

Khi thực thi vòng lặp, các lệnh sẽ được truyền danh sách giá trị qua biến. Chú ý: nếu các lệnh không bắt đầu bằng dấu ngoặc { thì tất cả những gì trước dấu chấm phẩy ; đều được sử dụng như là các lệnh.

Để vẽ các khoảng đơn vị trong hệ trục tọa độ ở hình vẽ của Karl ta dùng vòng lặp như sau:



```
\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 (0:30:3mm) -- cycle;
\draw[->] (-1.5,0) -- (1.5,0);
\draw[->] (0,-1.5) -- (0,1.5);
\draw (0,0) circle (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}
```

Có nhiều mã lệnh để vẽ các khoảng đơn vị tại hai trục tọa độ. Một cách khác là:

```
\foreach \x in {-1,-0.5,1}
\draw[xshift=\x cm] (0pt,-1pt) -- (0pt,1pt);
```

Có thể sử dụng lệnh `\foreach` theo cách sau khi có danh sách giá trị dài:



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

Nếu bạn cung cấp hai giá trị đầu tiên, lệnh lặp này sẽ phân chia danh sách giá trị theo khoảng của hai giá trị đầu đó:

.....

```
\foreach \x in {-1,-0.5,...,1}
\draw (\x cm,-1pt) -- (\x cm,1pt);
```

Sử dụng lệnh tạo vòng lặp có thể làm được nhiều việc phức tạp, ví dụ sau đây chỉ ra cách sử dụng vòng lặp trong vòng lặp:

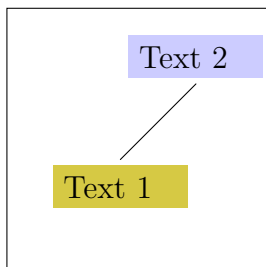
1,5	2,5	3,5	4,5	5,5
1,4	2,4	3,4	4,4	5,4
1,3	2,3	3,3	4,3	5,3
1,2	2,2	3,2	4,2	5,2
1,1	2,1	3,1	4,1	5,1

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

```
\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};
  }
```

### 1.1.21 Thêm chữ trong hình vẽ

Dùng `node` để thêm chữ trong hình vẽ.

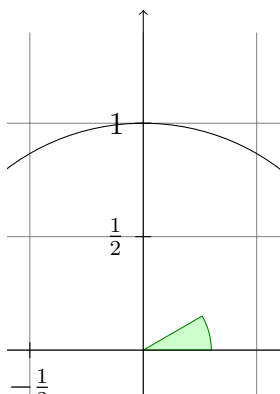


```
\begin{tikzpicture}
  \draw (-2,-2) rectangle (2.5,2.5);
  \draw (0.5,0.5)
  node [fill=examplefill, below=2pt]
  {Text 1} -- (1.5,1.5)
  node [fill=blue!20, above=2pt]{Text 2};
\end{tikzpicture}
```

Tất cả các node đều có thể dùng trong các lệnh như `draw`, `filled`, `shared`, `clipped`, ... Dùng thêm các tùy chọn cho node như `anchor = north`, `south`, `south east`. Ngoài ra còn có các tùy chọn như `below = x pt`; `above = x pt` để xác định vị trí đặt node một khoảng cách tùy chọn là `x pt`.

Trong hình vẽ của Karl, kí tự cần hiển thị là  $\frac{1}{2}$  nhưng lệnh `\foreach` chỉ hiểu giá trị là 0.5, không thể truyền theo tham trị  $\frac{1}{2}$ . Để giải quyết vấn đề này, dùng thêm biến khác (`xtext`) đi kèm biến `x` trong lệnh `\foreach`. Hai biến (`x` và `xtext`) cách nhau bởi một gạch chéo (/).

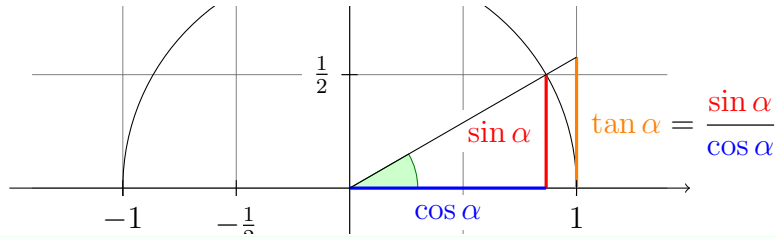
Trong mỗi vòng lặp, lệnh thứ nhất được thực hiện sau đó đến lệnh thứ hai. Nếu không có lệnh thứ 2 thì lệnh thứ nhất sẽ được lặp lại ngay.



```
\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 (0:30:3mm) -- cycle;
\draw[->] (-1.5,0) -- (1.5,0); \draw[->] (0,-1.5) -- (0,1.5);
\draw (0,0) circle (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}
```

Do node luôn được vẽ trong một ô chữ nhật nên dùng tùy chọn `[fill = white]` để tô màu nền trắng cho node và hình chữ nhật màu trắng sẽ đè lên các đường đã vẽ trước đó. Để đặt dòng text  $\sin \alpha$  ở giữa của đường thẳng sin, phải viết **node**  $\sin \alpha$  tại vị trí ngay sau lệnh vẽ - - trước toạ độ thứ 2 thay vì tại vị trí cuối cùng của dòng lệnh **draw**. Theo mặc định của gói lệnh PGF, **node**  $\sin \alpha$  được đặt ở giữa của đường thẳng sin nhưng nếu dùng thêm các tùy chọn **pos**, **near start**, **near end** sẽ điều chỉnh được vị trí của node theo ý muốn.

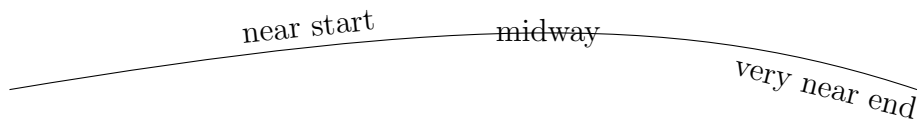


```

\begin{tikzpicture}[scale=3]
  \clip (-2,-0.2) 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
  (0:30:3mm) -- 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 (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}

```

Bạn cũng có thể đặt vị trí các node text dọc theo đường cong và điều chỉnh độ cong của node với tùy chọn `sloped`, `near start`, `very near end` như ví dụ dưới đây:



```

\begin{tikzpicture}
  \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}

```

Phần cuối cùng là vẽ một node text màu hồng bên phải đi kèm hình vẽ của Karl. Để đạt được độ rộng cần thiết của node, sử dụng tùy chọn `text with=6cm`. Toàn bộ mã lệnh sẽ là:

```

\begin{tikzpicture}
  [scale=3,line cap=round,
  % Styles
  axes/.style=,
  important line/.style={very thick},
  information text/.style={rounded corners,fill=red!10,inner sep=1ex}]

% Local definitions
\def\costhirty{0.8660256}

% Colors
\colorlet{anglecolor}{green!50!black}
\colorlet{sincolor}{red}
\colorlet{tancolor}{orange!80!black}
\colorlet{coscolor}{blue}

% The graphic
\draw[help lines,step=0.5cm] (-1.4,-1.4) grid (1.4,1.4);

\draw (0,0) circle (1cm);

\begin{scope}[axes]
  \draw[->] (-1.5,0) -- (1.5,0) node[right] {$x$};
  \draw[->] (0,-1.5) -- (0,1.5) node[above] {$y$};
  \foreach \x/\xtext in {-1, -.5/-\frac{1}{2}, 1}
    \draw[xshift=\x cm] (0pt,1pt) -- (0pt,-1pt) node[below,fill=white] {$\xtext$};
  \foreach \y/\ytext in {-1, -.5/-\frac{1}{2}, .5/\frac{1}{2}, 1}
    \draw[yshift=\y cm] (1pt,0pt) -- (-1pt,0pt) node[left,fill=white] {$\ytext$};
\end{scope}

\filldraw[fill=green!20,draw=anglecolor] (0,0) -- (3mm,0pt) arc(0:30:3mm);
\draw (15:2mm) node[anglecolor] {$\alpha$};
\draw[important line,sincolor]
  (30:1cm) -- node[left=1pt,fill=white] {$\sin \alpha$} +(0,-.5);
\draw[important line,coscolor]
  (0,0) -- node[below=2pt,fill=white] {$\cos \alpha$} (\costhirty,0);

\draw[important line,tancolor] (1,0) --
  node [right=1pt,fill=white]
  {
    $\displaystyle \tan \alpha \color{black}=$
    $\frac{\color{sincolor}\sin \alpha}{\color{coscolor}\cos \alpha}$
  } (intersection of 0,0--30:1cm and 1,0--1,1) coordinate (t);

\draw (0,0) -- (t);

\draw[xshift=2cm] node [right,text width=6cm,information text]
  {Góc {\color{anglecolor} $\alpha$} là $30^\circ$ trong ví dụ ($\pi/6$ nếu đơn vị đo là radian).
  Giá trị {\color{sincolor} $\sin \alpha$} là:
  $\[\color{sincolor} \sin \alpha = 1/2.\]$
  Theo định lý Pitago, ta có {\color{coscolor} $\cos^2 \alpha$} +
  {\color{sincolor} $\sin^2 \alpha$} = 1$.
  Do đó độ dài của đường màu xanh là giá trị của {\color{coscolor} $\cos \alpha$}, phải là
  $\[\color{coscolor} \cos \alpha = \sqrt{1 - 1/4} = \textstyle \frac{1}{2} \sqrt{3}.\]$
  Điều này cho ta độ dài đường màu cam {\color{tancolor} $\tan \alpha$} có giá trị là
  $\[\color{tancolor} \tan \alpha = \frac{\color{sincolor} \sin \alpha}{\color{coscolor} \cos \alpha}$
  {\color{coscolor} $\cos \alpha = 1/\sqrt{3}.\]$};
\end{tikzpicture}

```