

# Tài liệu ngắn gọn về TikZ

Phan Văn Phương

Ngày 6 tháng 11 năm 2016

## Tóm tắt nội dung

Đây là một tài liệu ngắn gọn, giới thiệu và hướng dẫn sử dụng TikZ một cách cơ bản nhất. Vì thời gian và năng lực có hạn, nên tài liệu này chắc chắn còn nhiều sơ suất, cách viết đôi chỗ còn khó hiểu. Rất mong nhận được góp ý của các bạn.

Phan Văn Phương - THPT Xuân Trường B - Nam Định.

## Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>3</b>
<b>2</b>	<b>Cài đặt</b>	<b>3</b>
2.1	Tải về và cài đặt . . . . .	3
2.2	Môi trường vẽ hình của TikZ . . . . .	3
2.3	Tự tạo các style cho riêng mình . . . . .	4
<b>3</b>	<b>Các thành phần cơ bản</b>	<b>5</b>
3.1	Điểm . . . . .	5
3.2	Đường . . . . .	6
3.3	Nốt . . . . .	7
<b>4</b>	<b>Vẽ đường và các thuộc tính</b>	<b>8</b>
4.1	Vẽ đoạn thẳng . . . . .	8
4.1.1	Độ rộng của đường . . . . .	9
4.1.2	Nét đứt . . . . .	10
4.1.3	Màu sắc . . . . .	10
4.1.4	Mối nối giữa các đường . . . . .	11
4.2	Vẽ mũi tên . . . . .	11
4.3	Đánh dấu góc . . . . .	12
4.4	Vẽ hình chữ nhật . . . . .	12
4.5	Vẽ đường tròn . . . . .	13
4.6	Vẽ cung tròn . . . . .	13
4.7	Vẽ ellipse . . . . .	13
4.8	Vẽ parabol . . . . .	14
4.9	Vẽ đường cong đi qua các điểm . . . . .	14
4.10	Vẽ đường lưới . . . . .	14
4.11	Vẽ đồ thị, biểu đồ . . . . .	15
<b>5</b>	<b>Tô màu</b>	<b>17</b>
<b>6</b>	<b>Đổ bóng</b>	<b>18</b>
<b>7</b>	<b>Cắt xén hình</b>	<b>18</b>
<b>8</b>	<b>Linh tinh</b>	<b>18</b>
8.1	Hệ tọa độ cực . . . . .	18
8.2	Lệnh vòng lặp . . . . .	18
8.3	Vẽ tiếp tuyến . . . . .	21
8.4	Tìm giao điểm của hai đường . . . . .	21

<b>9</b>	<b>Vẽ bảng biến thiên</b>	<b>21</b>
9.1	Vẽ hình bằng GeoGebra . . . . .	21
9.2	Sự khác nhau giữa path và draw . . . . .	23

# 1 Giới thiệu

Mục đích của tài liệu này là để giới thiệu và hướng dẫn về gói TikZ của tác giả Till Tantau, một gói giúp bạn vẽ hình vector<sup>1</sup> rất mạnh và đẹp trong L<sup>A</sup>T<sub>E</sub>X. Tôi hi vọng rằng tài liệu này sẽ giúp cho các giáo viên và những người làm trong ngành xuất bản vẽ được những hình ảnh, biểu đồ, đồ thị... của các sách khoa học, kĩ thuật một cách nhanh chóng và đẹp mắt. Đây chỉ là tài liệu vỡ lòng, các lệnh trong tài liệu này hết sức cơ bản, nếu bạn muốn một tài liệu đầy đủ nhất, có thể tham khảo hướng dẫn sử dụng đi kèm gói TikZ.

Tài liệu này được hoàn thành từ việc tham khảo tài liệu *PGF and TikZ manual*, lấy từ địa chỉ <https://sourceforge.net/projects/pgf/>; *A very minimal introduction to TikZ* của tác giả Jacques Cr'emer, bản tiếng Anh có thể được tải về từ địa chỉ <http://cremeronline.com/LaTeX/minimaltikz.pdf> và một số ví dụ được lấy từ trang <http://www.texample.net/tikz/examples/>

Chắc chắn rằng tài liệu này không tránh khỏi sai sót, hãy sử dụng nó một cách thông minh, và nếu bạn phát hiện ra bất cứ sai sót nào, hãy gửi email cho tôi theo địa chỉ [gachduoi@hotmail.com](mailto:gachduoi@hotmail.com). Hãy theo dõi các bài giảng về hướng dẫn sử dụng TikZ, trong thời gian tới, tại kênh youtube của tôi <https://www.youtube.com/c/hocbaicungconchannel>.

## 2 Cài đặt

### 2.1 Tải về và cài đặt

Để sử dụng gói TikZ, bạn chỉ cần thêm dòng `\usepackage{tikz}` vào phần preamble tài liệu của bạn, khi đó các phần mềm L<sup>A</sup>T<sub>E</sub>X, như MiKTeX chẳng hạn, sẽ tự động tải và cài đặt<sup>2</sup> gói TikZ. Nếu muốn sử dụng các thư viện của TikZ thì thêm dòng `\usetikzlibrary{<danhsách các thư viện>}`. Gói TikZ có chứa một số thư viện thường sử dụng sau:

- `arrows` dùng để vẽ (trang trí) nhiều loại mũi tên khác nhau.
- `shapes` dùng để vẽ các hình hình học (shape) như các đa giác, các kí hiệu (sign) như kí hiệu cấm hút thuốc chẳng hạn.
- `calc` để tính toán các trên tọa độ của các điểm trong mặt phẳng tọa độ.
- `intersections` dùng để tính toán tọa độ giao điểm giữa các đường.
- `mindmap` dùng để vẽ sơ đồ tư duy.
- `folding` dùng để vẽ các hình gấp giấy, như kiểu origami (!)

Và còn một số thư viện nữa, bạn có thể tham khảo thêm trong hướng dẫn đi kèm gói TikZ. Nếu sử dụng nhiều hơn một thư viện, thì các thư viện được khai báo cách nhau bởi dấu phẩy.

### 2.2 Môi trường vẽ hình của TikZ

Bây giờ, khi bạn muốn vẽ một hình bằng TikZ, chỉ cần đặt các lệnh của TikZ vào trong môi trường `tikzpicture`, như sau:

```
\begin{tikzpicture}[<các tùy chọn>]
<các lệnh vẽ hình, mỗi lệnh kết thúc bằng dấu chấm phẩy (;)>
\end{tikzpicture}
```

Ở đây, `<các tùy chọn>` thường là thuộc tính của hình vẽ, chẳng hạn độ đậm nhạt, nét đứt, nét liền, màu sắc... Các lệnh vẽ hình gồm có các lệnh vẽ đường, vẽ điểm, tô màu... mà chúng ta sẽ tìm hiểu sau. Tất cả các lệnh của TikZ đều phải đặt trong môi trường `tikzpicture`, ngoại trừ lệnh `\tikzstyle`. Bạn cũng có thể sử dụng các cách viết sau:

- `\tikz[<các tùy chọn>]{<lệnh vẽ hình>}`
- `\tikz[<các tùy chọn>] <lệnh vẽ hình>;`

<sup>1</sup>Bạn có thể tìm hiểu thêm về ảnh dạng vector bằng google, ưu điểm của ảnh vector là khi ta phóng to, hình ảnh không bị vỡ.

<sup>2</sup>Bạn cần kết nối mạng internet để làm điều này, nếu không, xin hãy tự tải về và cài đặt thủ công.

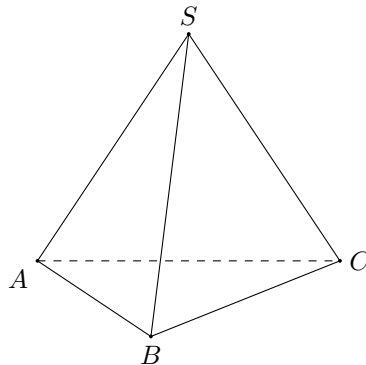
đối với các hình đơn giản ở trên cùng dòng với văn bản. Chú ý rằng trong cách thứ nhất không cần kết thúc bằng dấu chấm phẩy. Ví dụ, đoạn mã `\tikz \draw (0pt,0pt) -- (20pt,6pt);` sẽ thu được  $\text{---}$  và `\tikz \fill[orange] (1ex,1ex) circle (1ex);` sẽ vẽ hình  $\bullet$ . Nếu bạn muốn hình vẽ có nhãn, thì đặt trong môi trường `figure` như sau:

```
\begin{figure}
\begin{tikzpicture}[<các tùy chọn>]
<các lệnh vẽ hình, mỗi lệnh kết thúc bằng dấu chấm phẩy (;)>
\end{tikzpicture}
\caption{<Tên (nhãn), hoặc chú thích cho bức ảnh.>}
\end{figure}
```

Sau đó, bạn biên dịch tài liệu sử dụng pdfTEX hoặc XeTEX, và tận hưởng thành quả. Một số tùy chọn của môi trường `tikzpicture` mà tôi thường sử dụng:

- `scale=1.5` co giãn hình vẽ theo cả hai chiều ngang và dọc gấp 1.5 lần, bạn có thể thay số 1.5 bởi số khác tùy thích, chú ý rằng, số thập phân ở đây sử dụng dấu chấm để ngăn cách phần nguyên và phần thập phân.
- `xscale=2.5, yscale=0.5` kéo giãn chiều ngang 2.5 lần, chiều dọc 0.5 lần.
- `color=<tên màu>` màu sắc mặc định của các nét vẽ, bạn có thể sử dụng các màu sau
- `opacity` độ đậm nhạt của hình vẽ

Sau đây là một ví dụ, vẽ hình chóp  $S.ABC$ , sử dụng các lệnh vẽ đoạn thẳng và lệnh tô màu hình tròn, lệnh vẽ tên các nốt.



Mã nguồn hoàn chỉnh của hình trên:

```
\documentclass{standalone}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[scale=.7]
\draw (2,-1)-- (5,-3)-- (10,-1)-- (6,5)-- (2,-1);
\draw (6,5)-- (5,-3);
\draw [dashed] (2,-1)-- (10,-1);
\fill (2,-1) circle (.05) node [below left] {$A$};
\fill (5,-3) circle (.05) node [below] {$B$};
\fill (10,-1) circle (.05) node [right] {$C$};
\fill (6,5) circle (.05) node [above] {$S$};
\end{tikzpicture}
\end{document}
```

### 2.3 Tự tạo các style cho riêng mình

Bạn có thể tự tạo các phong cách, `style`, của hình vẽ cho riêng mình, bằng cách sử dụng lệnh

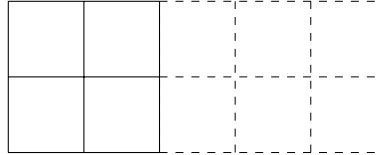
```
\tikzstyle<tên style của bạn>=[<các tùy chọn>]
```

Sau đó, mỗi khi vẽ hình, bạn chỉ cần nạp lại style của mình bằng cách thêm tên style của bạn vào phần <các thuộc tính và tùy chọn> của môi trường `tikzpicture` hoặc của lệnh `TikZ`. Ví dụ, bạn tự tạo một style tên là `help lines` bằng dòng lệnh

```
\tikzstyle{help lines}=[dashed,very thin]
```

Sau đó, khi cần vẽ hình với phong cách này, bạn chỉ cần gọi lại style này, chẳng hạn

```
\begin{tikzpicture}
\draw (0,0) grid +(2,2);
\draw[style=help lines] (2,0) grid +(2,2);
\end{tikzpicture}
```



### 3 Các thành phần cơ bản

Để vẽ hình, `TikZ` sử dụng hệ tọa độ Descart hoặc hệ tọa độ cực, đơn vị mặc định là `cm`. Mỗi điểm có hai tọa độ, cách nhau bởi dấu phẩy, ví dụ trong hệ tọa độ Descart, gốc tọa độ là  $(0,0)$ .

Các yếu tố cơ bản của `TikZ` là điểm, đường `path` và nốt<sup>3</sup> `node`. Đối với các điểm, có các thao tác tính toán trên tọa độ của chúng. Đối với các đường, có bốn thao tác chính là vẽ `\draw`, tô màu `\fill`, thay đổi sắc thái `\shade` và cắt xén `\clip`. Chúng ta sẽ lần lượt đi tìm hiểu các vấn đề này.

#### 3.1 Điểm

Để *khai báo*<sup>4</sup> một điểm, ta sử dụng lệnh `\coordinate (<tên điểm> at (<tọa độ của điểm>)` hoặc `\coordinate [<thuộc tính>](<tên điểm> at (<tọa độ của điểm>)`. Ví dụ, để khai báo điểm `A` có tọa độ  $(3,1)$  ta sử dụng lệnh `\coordinate (A) at (3,1);`. Sau khi đã định nghĩa một điểm, bạn có thể sử dụng chúng để thêm vào một đường, định nghĩa một nốt, hoặc tính toán trên tọa độ của nó.

Trong mặt phẳng tọa độ vuông góc *Oxy*, <tọa độ của điểm> gồm hai yếu tố là hoành độ và tung độ, cách nhau bởi dấu phẩy, đặt trong cặp ngoặc tròn, giống như thế này  $(1\text{cm},2\text{pt})$ . Nếu bạn không chỉ ra đơn vị, chẳng hạn  $(2,1)$ , thì đơn vị mặc định là centimet. Chú ý rằng,  $\text{\LaTeX}$  chỉ làm việc với các đối tượng nhỏ hơn 575 `cm`. Một số đơn vị độ dài được phép sử dụng: `cm`, `mm`, `pt`, `ex`... Bạn xem bảng<sup>5</sup> sau để rõ hơn về các đơn vị được sử dụng trong  $\text{\LaTeX}$

1pt	1mm	1cm	1ex	1em	1bp	1dd	1pc	1in	
1	2.84526	28.45274	4.3045	9.99756	1.00374	1.07	12	72.26999	pt
0.35146	1	10.00005	1.51286	3.51376	0.35277	0.37607	4.21754	25.40013	mm
0.03514	0.1	1	0.15129	0.35138	0.03528	0.03761	0.42175	2.54	cm
0.23232	0.661	6.60999	1	2.32259	0.23318	0.24858	2.78778	16.78938	ex
0.10002	0.28459	2.84596	0.43056	1	0.1004	0.10703	1.20029	7.22876	em
0.99628	2.83467	28.34677	4.28847	9.96033	1	1.06602	11.9553	72.00082	bp
0.93457	2.6591	26.59117	4.02287	9.34344	0.93806	1	11.21487	67.54158	dd
0.08333	0.2371	2.37106	0.3587	0.83313	0.08365	0.08917	1	6.0225	pc
0.01384	0.03937	0.3937	0.05956	0.13834	0.01389	0.0148	0.16605	1	in

Bạn cũng có thể chỉ rõ một điểm trong mặt phẳng tọa độ cực, bằng cách sử dụng dấu hai chấm thay cho dấu phẩy, như kiểu  $(30:1\text{cm})$ , điều này có nghĩa là điểm ở vị trí cách gốc tọa độ "1cm theo chiều dương góc 30°".

<sup>3</sup>Đôi khi tôi còn gọi là *điểm nút*.

<sup>4</sup>Tôi thường dùng từ *định nghĩa* thay cho từ *khai báo*

<sup>5</sup>Bảng này được lấy từ đường link

[tex.stackexchange.com/questions/8260/what-are-the-various-units-ex-em-in-pt-bp-dd-pc-expressed-in-mm](http://tex.stackexchange.com/questions/8260/what-are-the-various-units-ex-em-in-pt-bp-dd-pc-expressed-in-mm)

Nếu bạn sử dụng hệ tọa độ *Oxyz* trong không gian, thì một điểm sẽ có ba thành tố là hoành độ, tung độ và cao độ, chẳng hạn  $(1, -2, 2)$ .



Nếu đã có sẵn những điểm được định nghĩa, bạn có thể định nghĩa những điểm mới dựa vào vị trí tương đối của nó so với những điểm có sẵn này. Chẳng hạn như tạo ra điểm mới ở phía nam so với điểm đã có bằng cách sử dụng mã lệnh (`<điểm đã có>.south`). Hoặc cụ thể hơn, bạn thêm hai dấu cộng trước một tọa độ như kiểu `++(1cm,0pt)`, điều này có nghĩa là điểm mới sẽ ở vị trí "1cm sang phía đông so với điểm đã có". Điều này cho phép bạn dễ dàng mô tả sự di chuyển. Ví dụ,  $(1,0) ++(1,0) ++(0,1)$  sẽ chỉ ra ba tọa độ lần lượt là  $(1,0)$  rồi  $(2,0)$ , và cuối cùng là  $(2,1)$ .

Cuối cùng, thay vì sử dụng hai dấu cộng, bạn có thể sử dụng một dấu cộng. Điều này cũng dùng để chỉ tọa độ một điểm so với điểm ban đầu, nhưng không làm thay đổi tọa độ của điểm cuối cùng được sử dụng để dùng tiếp. Chẳng hạn,  $(1,0) +(1,0) +(0,1)$  sẽ chỉ ra ba tọa độ lần lượt là  $(1,0)$  rồi  $(2,0)$ , và cuối cùng là  $(1,1)$ , tức là điểm cuối cùng sẽ tăng hoành độ x lên 0 đơn vị và tăng tung độ y lên 1 đơn vị, nhưng là so với điểm đầu tiên, chứ không phải điểm thứ hai, điều này khác với ví dụ trước.

Chú ý rằng, một điểm ngoài tên, ta có thể gán nhãn cho chúng, bằng cách thêm thuộc tính `label`, khi đã có nhãn, điểm đó sẽ được vẽ vào hình vẽ. Ví dụ `\coordinate [label=$A$] (A) at (3,1);` sẽ thu được hình vẽ:

A

## 3.2 Đường

Khi tạo ra các hình vẽ bằng TikZ, công việc chủ yếu của bạn là làm việc với các đường nét `\path`. Một đường là một tập gồm nhiều các đoạn thẳng hoặc các đường (đoạn) cong, có thể không cần phải liên kết với nhau. Để chỉ ra một đường, bạn chỉ ra tọa độ các điểm mà đường đi qua, trực tiếp bằng cách cung cấp tọa độ của các điểm này hoặc thông qua các điểm đã được định nghĩa sẵn, và hình dạng của đường. Ví dụ, `(8pt,0pt) -- (0pt,0pt) -- (0pt,8pt) -- cycle` là một đường gấp khúc khép kín gồm ba đoạn thẳng, bắt đầu từ điểm  $(8pt,0pt)$ , qua điểm  $(0pt,0pt)$  rồi đến điểm  $(0pt,8pt)$  và cuối cùng kết thúc chính tại điểm ban đầu  $(8pt,0pt)$ , kí hiệu `--` chỉ hình dạng của đường là đường thẳng. Khi bạn dùng thao tác `\draw` để vẽ đường này sẽ thu được một tam giác như này , còn nếu bạn dùng thao tác tô màu `\fill` với màu cam, thì sẽ thu được như này .




Sau khi định nghĩa một đường, bạn có thể sử dụng chúng để tính toán, như để tìm giao điểm giữa các đường chẳng hạn, hoặc sử dụng để vẽ hình với các thao tác `\draw`, `\fill` hoặc `\clip...` Cấu trúc chung của lệnh làm việc với đường là

```
\path [<thao tác, thuộc tính>] (<tọa độ các điểm, hình dạng của đường>);
```

Ví dụ, vẽ tam giác ở trên là `\path [draw] (8pt,0pt) -- (0pt,0pt) -- (0pt,8pt) -- cycle;`, tô màu tam giác trên là `\path [fill, orange] (8pt,0pt) -- (0pt,0pt) -- (0pt,8pt) -- cycle;`. Có bốn thao tác chính đối với đường là vẽ `\draw`, tô màu `\fill`, đổ bóng `\shade` và cắt xén `\clip`.

- `\draw` có thể hiểu là bạn vẽ theo một đường đã định nghĩa.
- `\fill` có thể hiểu là bạn tô màu toàn bộ miền trong của một đường đã định nghĩa, tất nhiên đường này phải khép kín, nếu không TikZ sẽ tự tạo nên một miền khép kín cho bạn bằng cách nối điểm bắt đầu và điểm kết thúc của đường bằng một đoạn thẳng.
- `\shade` gần giống như tô màu, nhưng màu sắc sẽ thay đổi dần dần sắc thái từ đậm sang nhạt.
- `\clip` có thể hiểu là bạn cắt xén lấy một phần của đường.


Bạn cũng có thể sử dụng nhiều thao tác trên cùng một đường. Ta sẽ xem xét một đường không phải đường thẳng, chẳng hạn là một đường hình chữ nhật với hai đỉnh của một đường chéo là  $(0,0)$  và  $(2ex,1ex)$ , bạn định nghĩa đường này bằng hai tọa độ và hình dạng của đường là hình chữ nhật `rectangle`, như thế này `\path (0,0) rectangle (2ex,1ex);`.

Bạn vẽ nó bằng cách thêm thao tác `draw`, như này `\path[draw] (0,0) rectangle (2ex,1ex);`, thì được , hoặc tô màu xanh cho nó, như này `\path[fill, blue] (0,0) rectangle (2ex,1ex);`, sẽ thu được . Bạn có thấy hình chữ nhật màu xanh bé hơn hình chữ nhật ban đầu không? Tất nhiên, vì thao tác `fill` chỉ tô màu miền trong của hình chữ nhật mà không vẽ hình chữ nhật đó, để vừa vẽ vừa tô màu bạn có thể thêm cả thao tác `draw` vào nữa, như này chẳng hạn `\path[draw, fill, blue] (0,0) rectangle (2ex,1ex);`, sẽ được .

Lệnh `\path[draw]` có thể được viết gọn lại thành `\draw`. Để tô màu một đường, thêm thao tác `fill` vào lệnh `\path` để được `\path[fill]` hoặc viết gọn thành `\fill`. Tương tự, ta cũng có thể viết lệnh `\filldraw` thay cho `\path[fill,draw]`.

Khi TikZ vẽ hoặc tô màu một đường, cần có các tham số như màu sắc, độ dày mảnh của đường nét, hoặc kiểu vẽ nét liền hay nét đứt... Ví dụ, đoạn mã sau:

```
\tikz \draw[line width=2pt,color=red] (1,0) -- (0,0) -- (1,0) -- cycle;
```

sẽ thu được một đoạn thẳng rộng 2pt, màu đỏ 

### 3.3 Nốt

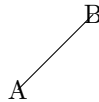
Ở đây tôi sẽ dùng từ *nốt* hoặc đôi khi *điểm nút* để thay cho *node*. Nốt cũng gần giống như điểm, nhưng ngoài yếu tố tọa độ, một nốt bắt buộc còn kèm theo nhãn và có thể có các hình dạng `shape` khác nhau, và có thể được vẽ lên trong hình. Khi bạn định nghĩa một nốt, thì nhãn của nó sẽ luôn luôn hiện ra trên hình vẽ, còn điểm thì không. Bạn có thể hiểu nốt chính là hiển thị một điểm lên hình vẽ bằng cách vẽ nhãn của nó, hoặc nếu cần thì vẽ cả hình dạng của nó, bạn có thể chọn hình dạng của nốt là một hình tròn, hình chữ nhật... Để định nghĩa một nốt, bạn sử dụng mã lệnh

```
\node (<tọa độ của nốt>) [<các tùy chọn>]{<nhãn của nốt>;
```

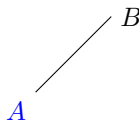
hoặc bạn có thể đặt tên cho nốt đó, để sử dụng lại sau này

```
\node (<tên nốt>) at (<tọa độ của nốt>) [<các tùy chọn>]{<nhãn của nốt>;
```

Trong lệnh trên, `<tọa độ của nốt>` cũng có cú pháp như tọa độ của điểm, `<các tùy chọn>` gồm có màu sắc `color`, vị trí tương đối của nhãn so với nốt đó, có các vị trí `left`, `right`, `above`, `below` hoặc kết hợp hai trong bốn vị trí này, chẳng hạn `below left`, chú ý rằng khi kết hợp `below` và `above` phải được viết trước. Ngoài ra còn có hình dạng của nốt, sử dụng thuộc tính `shape=<tên của hình dạng>`, mặc định, có các hình dạng `rectangle`, `circle`, `coordinate`. Khi bạn định nghĩa một đường, bạn có thể thêm các nốt vào các điểm của đường đó, chẳng hạn, `\draw (1,1) node {A} --(2,2) node {B};` sẽ cho ta

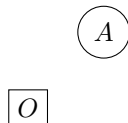


Như thế nhìn chưa đẹp, bạn có thể thêm thuộc tính cho nốt để đẹp hơn, chẳng hạn, đoạn mã sau `\draw (1,1) node [blue, below left] {$A$} -- (2,2) node [right] {$B$};`, sẽ cho ta:



Hoặc bạn có thể vẽ hình dạng của nốt, như ví dụ sau chẳng hạn:

```
\begin{tikzpicture}
\node (A) at (1,1) [circle, draw] {$ A $};
\node (O) at (0,0) [rectangle ,draw] {$ 0 $};
\end{tikzpicture}
```



Sau đây, chúng ta sẽ lần lượt đi tìm hiểu chi tiết các thao tác vẽ hình trong TikZ.

## 4 Vẽ đường và các thuộc tính

Để vẽ đường, dùng lệnh `\draw [<thuộc tính>] <tọa độ 1> <loại đường> <tọa độ 2>`; đây chính là cách viết tắt của `\path [draw,<thuộc tính>] <tọa độ 1> <loại đường> <tọa độ 2>`; như trong phần trước đã nói. Trong đó `<tọa độ 1>` là tọa độ điểm bắt đầu, `<tọa độ 2>` là tọa độ điểm kết thúc, còn `<loại đường>` cho biết lệnh `\draw` phải vẽ gì, vẽ đoạn thẳng `--`, vẽ hình chữ nhật `rectangle`, vẽ lưới `grid` hay vẽ đường cong... Riêng trường hợp vẽ đường tròn `circle` và vẽ cung tròn `arc` thì `<tọa độ 1>` là tọa độ tâm còn `<tọa độ 2>` thay bằng bán kính. Các `<thuộc tính>` thì có độ dày hay mảnh của nét vẽ, màu sắc, nét liền hay nét đứt...

Để cho biết tọa độ của một điểm, bạn có thể nhập trực tiếp tọa độ của điểm đó, hoặc sử dụng lệnh `\coordinate` để khai báo<sup>6</sup> tọa độ của một điểm, và sử dụng tên của điểm đã khai báo để dễ làm việc hơn.

Ví dụ, để vẽ đoạn thẳng từ điểm có tọa độ (1,1) đến điểm có tọa độ (3,2), thì `<tọa độ 1>` là (1,1), `<tọa độ 2>` là (3,2), còn `<loại đường>` là `--`. Đoạn mã hoàn chỉnh như sau:

```
\begin{tikzpicture}
\draw (1,1)--(3,2);
\end{tikzpicture}
```

và ta thu được kết quả như thế này:



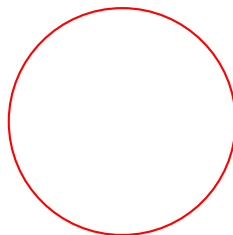
Một ví dụ khác, đoạn mã sau, bạn chỉ cần thêm `<thuộc tính>` là `dashed` thì sẽ vẽ được đoạn thẳng bằng nét đứt. Ta có một hình chữ nhật bằng nét đứt như sau:

```
\begin{tikzpicture}
\draw [dashed] (0,0) rectangle (3,1.5);
\end{tikzpicture}
```



hoặc vẽ một đường tròn màu đỏ, nét đậm:

```
\begin{tikzpicture}
\draw [thick, red] (2,2) circle (1.5cm);
\end{tikzpicture}
```



TikZ còn vẽ được các đường phức tạp hơn, như đường `parabol`, đường `ellipse`, hoặc một đường cong bất kì. Sau đây, chúng ta sẽ lần lượt đi tìm hiểu về các lệnh vẽ đường (nét) này!

### 4.1 Vẽ đoạn thẳng

Để vẽ một đoạn thẳng đơn giản từ điểm có tọa độ (a,b) đến điểm có tọa độ (c,d) bạn sử dụng lệnh `\draw (a,b)--(c,d)`; . Nếu muốn tùy biến thêm cho đoạn thẳng, bạn dùng lệnh:

```
\draw [<các thuộc tính>] (a,b)--(c,d);
```

---

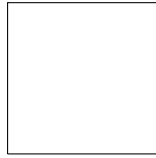
<sup>6</sup>Xin xem phần 3.1 để rõ hơn.



Ví dụ, lệnh `\draw (0,0)--(3,1)`; sẽ vẽ một đoạn thẳng từ điểm  $(0,0)$  đến điểm có tọa độ  $(3,1)$ .

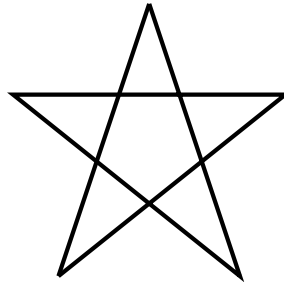


Bạn cũng có thể vẽ nhiều đoạn thẳng liên tiếp nhau, chẳng hạn, để vẽ hình vuông có các đỉnh lần lượt là  $(0,0)$ ;  $(0,2)$ ;  $(2,2)$  và  $(2,0)$  chỉ cần viết `\draw (0,0)--(0,2)--(2,2)--(2,0)--(0,0)`;



Dĩ nhiên, bạn có thể vẽ nhiều đoạn thẳng trên cùng một hình vẽ, hoặc bổ sung thêm các thuộc tính về màu sắc, độ dày mảnh... cho đoạn thẳng. Chẳng hạn, vẽ hình ngôi sao bằng nét rất đậm `ultra thick`:

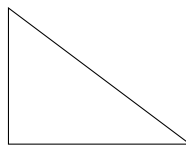
```
\begin{tikzpicture}[scale=.6]
\draw [ultra thick] (9,5)--(11,-1)--(6,3)--(12,3)--(7,-1);
\draw [ultra thick] (7,-1)--(9,5);
\end{tikzpicture}
```



hoặc bạn có thể xem hình chóp ở trang 4 của tài liệu này.

Ta xem xét một ví dụ nữa, sử dụng các lệnh khai báo điểm. Ta khai báo các điểm  $A(1,0)$ ,  $B(5,0)$ ,  $C(1,3)$  và vẽ tam giác  $ABC$  vuông tại  $A$ .

```
\begin{tikzpicture}[scale=.6]
\coordinate (A) at (1,0);
\coordinate (B) at (5,0);
\coordinate (C) at (1,3);
\draw (A)--(B)--(C)--cycle;
\end{tikzpicture}
```



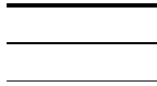
Sau đây, chúng ta sẽ tìm hiểu chi tiết hơn về <các thuộc tính> mà lệnh `\draw` có thể có.

#### 4.1.1 Độ rộng của đường

Để thay đổi độ rộng (đậm hay mảnh) của đường, ta bổ sung cho lệnh `\draw` thuộc tính về độ rộng của đường nét. Ví dụ, đoạn mã sau:

```
\begin{tikzpicture}
\draw [ultra thick] (0,1) -- (2,1);
\draw [thick] (0,0.5) -- (2,0.5);
\draw [thin] (0,0) -- (2,0);
\end{tikzpicture}
```

sẽ cho ta hình vẽ:



Có các độ rộng mặc định sau: `ultra thin`, `very thin`, `thin`, `semithick`, `thick`, `very thick` và `ultra thick`. Bạn cũng có thể quy định độ rộng cho nét vẽ bằng một con số cụ thể, sử dụng thuộc tính `line width`, chẳng hạn:

```
\begin{tikzpicture}
\draw [line width=12] (0,0) -- (2,0);
\draw [line width=0.2cm] (4,.75) -- (5,.25);
\end{tikzpicture}
```

sẽ thu được các đoạn thẳng có độ rộng 12 pt (đơn vị mặc định là pt) và 0.2 cm như hình vẽ sau:



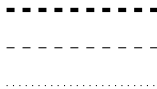
Về các đơn vị đo chiều dài, xin xem chi tiết trong bảng ở trang 5 để rõ hơn.

#### 4.1.2 Nét đứt

Bạn cũng có thể vẽ các đường nét đứt sử dụng thuộc tính `dashed`, hoặc chấm chấm sử dụng thuộc tính `dotted`, ví dụ:

```
\begin{tikzpicture}
\draw [dashed, ultra thick] (0,1) -- (2,1);
\draw [dashed] (0, 0.5) -- (2,0.5);
\draw [dotted] (0,0) -- (2,0);
\end{tikzpicture}
```

sẽ thu được:



Giá trị mặc định về đường nét đứt hay liền: `solid`, `dotted`, `densely dotted`, `loosely dotted`, `dashed`, `densely dashed`, `loosely dashed`, `dash dot`, `densely dash dot`, `loosely dash dot`, `dash dot dot`, `densely dash dot dot`, `loosely dash dot dot`. Hoặc bạn có thể tự quyết định kiểu dáng cho nét đứt của mình, chẳng hạn:

```
\draw[ultra thick,dash pattern=on 1cm off .25cm on .25cm off .5cm](0,0)--(8,0);
```

sẽ thu được

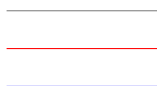





















#### 4.1.3 Màu sắc

Để thay đổi màu sắc cho các đường, bạn thêm thuộc tính `color`, chẳng hạn:

```
\begin{tikzpicture}
\draw [gray] (0,1) -- (2,1);
\draw [red] (0, 0.5) -- (2,0.5);
\draw [blue] (0,0) -- (2,0);
\end{tikzpicture}
```

sẽ thu được:



Các màu sắc mặc định gồm có: red , green , blue , cyan , magenta , yellow , black , gray , darkgray , lightgray , brown , lime , olive , orange , pink , purple , teal , violet  và white .

Bạn có thể pha trộn nhiều màu với nhau theo một tỉ lệ cho trước, chẳng hạn `black!60!green` có nghĩa là màu được pha bởi màu đen và màu xanh lá, với tỉ lệ 60% đen và 40% xanh lá. Hoặc có thể tự định nghĩa màu sắc cho mình, sử dụng cú pháp:

```
\definecolor{<tên màu>}{<hệ màu>}{<mô tả màu>}
```

Ở đây, <tên màu> là tên màu do bạn tự đặt ra, <hệ màu> có các hệ `gray`, `rgb`, `RGB`, `HTML` và `cmymk`. Còn <mô tả màu> là thông tin (tỉ lệ các màu cơ bản) để tạo thành màu của bạn. Ví dụ, bạn có thể sử dụng mã lệnh sau:

```
\definecolor{mauxanh}{rgb}{0.36, 0.54, 0.66}
```

với các giá trị nằm trong khoảng (0,1), hoặc sử dụng

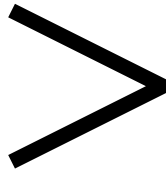
```
\definecolor{mautim}{RGB}{255,51,76}
```

với các giá trị là số nguyên nằm trong khoảng (0,255). Sau này, để dùng màu đó, bạn chỉ cần sử dụng các tên màu `mauxanh`, `mautim` đã được định nghĩa.

Một mẹo nhỏ, bạn có thể vào trang <http://latexcolor.com/> hoặc <http://cloford.com/resources/colours/500col.htm> để chọn màu sắc và lấy mã màu phù hợp.

#### 4.1.4 Mối nối giữa các đường

Khi hai đường giao nhau, để thay đổi hình dạng của chỗ nối, ta thêm thuộc tính `line join` cho lệnh `\draw`. Bạn xem ví dụ sau để rõ hơn:



Ở trên ta thấy chỗ mối nối giữa hai đoạn thẳng có hình tròn, do ta đã cho thuộc tính `line join=round`. Bạn có thể thay `round` bằng các từ `bevel`, hoặc `miter`. Bạn hãy tự khám phá để xem sự thay đổi của mối nối giữa các đường khi thay đổi thuộc tính này nhé.

## 4.2 Vẽ mũi tên

Bạn có thể vẽ (trang trí) thêm mũi tên cho các đoạn thẳng bằng cách thêm thuộc tính cho lệnh `\draw`, ví dụ, sử dụng các lệnh

```
\begin{tikzpicture}
\draw [->] (0,0) -- (2,0);
\draw [<-] (0,-0.5) -- (2,-0.5);
\draw [|->] (0,-1) -- (2,-1);
\draw[<->] (0,-1.5)--(5,-2);
\end{tikzpicture}
```

sẽ thu được như hình vẽ sau:



Khi bạn vẽ liên tiếp nhiều đoạn thẳng, các mũi tên sẽ được đặt vào điểm đầu và điểm cuối của đường gấp khúc này. Ví dụ, để vẽ hệ trục tọa độ, bạn chỉ cần dùng lệnh

```
\begin{tikzpicture}
\draw [<->] (0,2) -- (0,0) -- (3,0);
\end{tikzpicture}
```

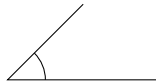
sẽ thu được:



### 4.3 Đánh dấu góc

Để đánh dấu một góc trong hình, cần nạp thư viện `angles` bằng lệnh `\usetikzlibrary{angles}`. Sau đây là một số ví dụ:

```
\draw (2,0)coordinate(A)--(0,0)coordinate(B)--(1,1)coordinate(C)pic[draw]{angle};
```



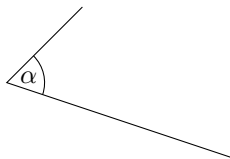
Một ví dụ khác, ta thay `draw` bằng `fill` sẽ được:

```
\draw (2,0)coordinate(A)--(0,0)coordinate(B)--(1,1)coordinate(C)pic[fill]{angle};
```



Để thêm tên góc, hoặc số đo của góc, ta sử dụng thư viện `quotes` của TikZ, `\usetikzlibrary{quotes}`

```
\draw (3,0) coordinate (A) -- (0,1) coordinate (B) -- (1,2)
coordinate (C) pic [draw,"$\alpha$"] {angle};
```



### 4.4 Vẽ hình chữ nhật

Để vẽ một hình chữ nhật, sử dụng lệnh sau để vẽ:

```
\draw [<các thuộc tính>] <tọa độ 1> rectangle <tọa độ 2>;
```

trong đó `<tọa độ 1>` là tọa độ đỉnh bên trái, phía dưới của hình chữ nhật, còn `<tọa độ 2>` là tọa độ đỉnh phía trên, bên phải. Ví dụ, vẽ hình chữ nhật bằng nét rất đậm có đỉnh dưới cùng bên trái là  $(0,0)$ , và đỉnh trên cùng bên phải là  $(5,3)$  ta dùng lệnh

```
\draw [ultra thick] (0,0) rectangle (5,3);
```

sẽ thu được:



## 4.5 Vẽ đường tròn

Để vẽ một đường tròn, sử dụng lệnh sau để vẽ:

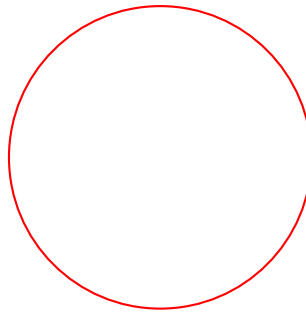
```
\draw [<các thuộc tính>] <tọa độ tâm> circle [radius=<độ dài bán kính>];
```

hoặc đơn giản hơn

```
\draw [<các thuộc tính>] <tọa độ tâm> circle (<độ dài bán kính>);
```

Ví dụ, vẽ một đường tròn bằng nét đậm, màu đỏ, có tâm ở (3,0.5) và bán kính bằng 2cm ta sử dụng lệnh:

```
\draw [red, thick] (3,0.5) circle [radius=2cm];
```



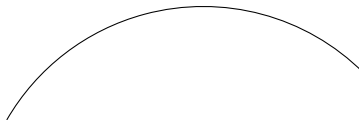
## 4.6 Vẽ cung tròn

Để vẽ một cung tròn, arc, bán kính bằng <R>, kéo dài từ <góc 1> đến <góc 2> ta sử dụng mã lệnh:

```
\draw [<thuộc tính>] <tâm> arc [radius=<R>,start angle=<góc 1>,end angle= <góc 2>];
```

Ví dụ, vẽ một cung tròn bán kính bằng 3cm, kéo dài từ góc 45° đến góc 120° ta sử dụng mã lệnh:

```
\draw (6,0) arc [radius=3cm, start angle=45, end angle= 150];
```

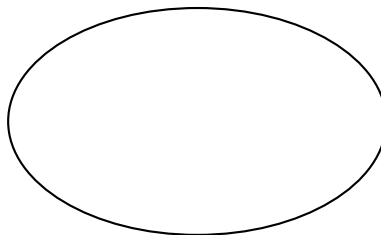


## 4.7 Vẽ ellipse

Để vẽ một ellipse với tâm có tọa độ là <tọa độ> và độ dài trục nằm ngang là <a>, độ dài trục thẳng đứng là <b> ta sử dụng lệnh sau để vẽ:

```
\draw [<các thuộc tính>] <tọa độ> ellipse (<a> and <b>);
```

Ví dụ, lệnh `\draw [thick] (0,0) ellipse (2.5cm and 1.5cm);` sẽ cho ta:



Chú ý rằng nếu bạn thay từ `ellipse` bởi từ `circle` trong câu lệnh trên thì vẫn được kết quả tương tự.

## 4.8 Vẽ parabol

## 4.9 Vẽ đường cong đi qua các điểm

Bạn có thể vẽ một đường cong điều khiển, `controls`, bởi các điểm

## 4.10 Vẽ đường lưới

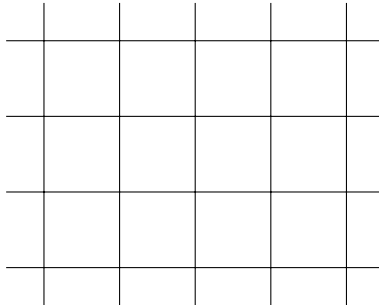
Để vẽ các đường lưới, `grid`, cho một hình chữ nhật ta sử dụng lệnh sau để vẽ:

```
\draw [<các thuộc tính>] <tọa độ 1> grid <tọa độ 2>;
```

trong đó `<tọa độ 1>` là tọa độ đỉnh bên trái, phía dưới của hình chữ nhật, còn `<tọa độ 2>` là tọa độ đỉnh phía trên, bên phải. Còn `<các thuộc tính>` thì có `step,xstep,ystep` là khoảng cách giữa các đường kẻ của lưới, màu sắc, độ dày mảnh của đường kẻ, `rotate` là đường lưới sẽ bị quay đi một góc cho trước. Ví dụ,

```
\draw[step=1,black,thin] (0.5,0.5) grid (5.5,4.5);
```

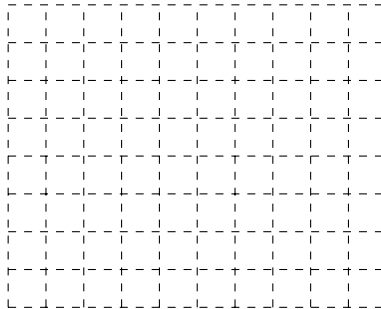
sẽ thu được



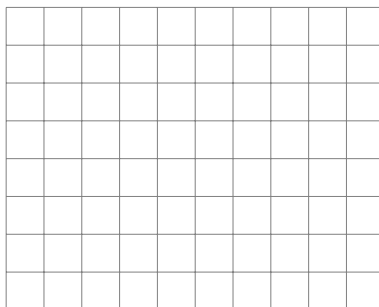
Muốn các đường kẻ của lưới là nét đứt, ta thêm thuộc tính `dashed`, chẳng hạn lệnh

```
\draw[step=0.5,black,thin,dashed] (0,0) grid (5,4);
```

sẽ thu được:



Ngoài ra, còn một thuộc tính là `style=help lines`, thì các đường kẻ của lưới sẽ tự động nhạt và mảnh hơn. Ví dụ `\draw[step=0.5,help lines] (0,0) grid (5,4);`



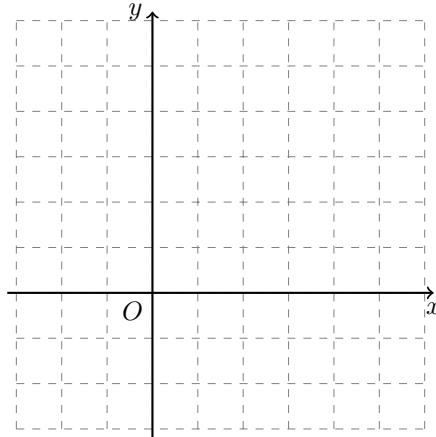
Bạn có thể kết hợp các lệnh vẽ đường lưới, vẽ mũi tên và vẽ nốt để được hệ trục tọa độ. Chẳng hạn đoạn mã sau

```

\begin{tikzpicture}[scale=.6]
\draw [help lines, dashed] (-3,-3) grid (6,6);
\draw [->, thick] (-3.2,0) -- (6.2,0) node [below]{$ x $};
\draw [->, thick] (0,-3.2) -- (0,6.2) node [left]{$ y $};
\node (0,0) [below left] (0){$ 0 $};
\end{tikzpicture}

```

sẽ cho ta một hệ trục tọa độ với các đường lưới:

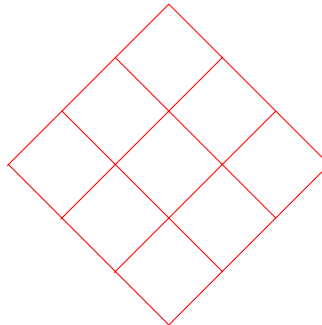


Bạn cũng có thể xoay các đường lưới đi một góc tùy ý, chẳng hạn một góc  $45^\circ$ , như ví dụ sau:

```

\draw[red] (0,0) grid [rotate=45] (3,3);

```



#### 4.11 Vẽ đồ thị, biểu đồ

*TikZ* có thể vẽ đồ thị trực tiếp, sử dụng lệnh:

```

\draw [<các thuộc tính>] plot <(biến, {hàm})>;

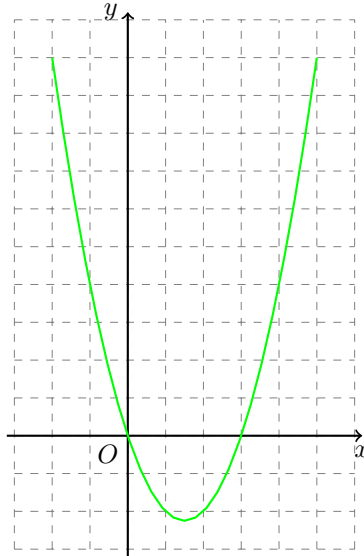
```

Các thuộc tính ở đây, ngoài màu sắc, độ dày mảnh của đường còn có *domain* là miền mà ta sẽ vẽ đồ thị, *samples* là số điểm lấy mẫu để vẽ, ta tưởng tượng rằng *TikZ* sẽ lấy một số điểm trên đồ thị hàm số và nối chúng lại bằng các đoạn thẳng, nên ta lấy mẫu càng nhiều thì đồ thị càng *mịn*. Tên biến chẳng hạn như  $x$ , hàm số là một biểu thức tính toán dựa trên biến ta vừa cung cấp, ví dụ hàm bậc hai  $y = x^2 - 3x$  thì ta viết là  $(x)^2-3*x$ . Hãy xem ví dụ sau để hiểu rõ.

```

\begin{tikzpicture}[scale=.5]
\draw [help lines, dashed] (-3,-3) grid (6,11);
\draw [->, thick] (-3.2,0) -- (6.2,0) node [below]{$ x $};
\draw [->, thick] (0,-3.2) -- (0,11.2) node [left]{$ y $};
\node (0,0) [below left] {$ 0 $};
\draw [green, thick, domain=-2:5] plot (\x, {(\x)^2-3*\x});
\end{tikzpicture}

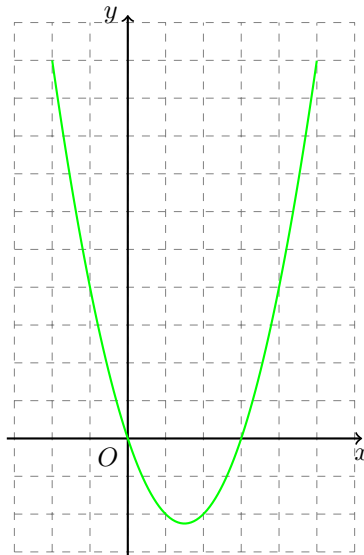
```



Nếu bạn phóng to hình trên, sẽ thấy đồ thị của ta chưa đẹp, chưa *mịn*, ta thêm thuộc tính sau `sample=100` vào nhìn đồ thị sẽ đẹp hơn.

```
\begin{tikzpicture}[scale=.5]
\draw[help lines, dashed] (-3,-3) grid (6,11);
\draw[->, thick] (-3.2,0) -- (6.2,0) node[below]{$ x $};
\draw[->, thick] (0,-3.2) -- (0,11.2) node[left]{$ y $};
\node(0,0)[below left] {$ 0 $};
\draw[green, thick, domain=-2:5, samples=100] plot (\x, {\x*\x-3*\x});
\end{tikzpicture}
```

Lúc này, để vẽ đồ thị hàm số trên, TikZ sẽ lấy mẫu là 100 điểm thuộc đồ thị và nối chúng lại, nhìn đường cong của chúng ta sẽ mịn hơn rất nhiều:



Một ví dụ khác, vẽ đồ thị hàm số  $y = x^3 - 3x^2 + 1$ , được lấy từ blog của tác giả Nguyễn Thanh Triều, tại địa chỉ <https://nttrieu.wordpress.com/2013/05/14/ve-do-thi-cua-ham-so-voi-goi-tikz/>

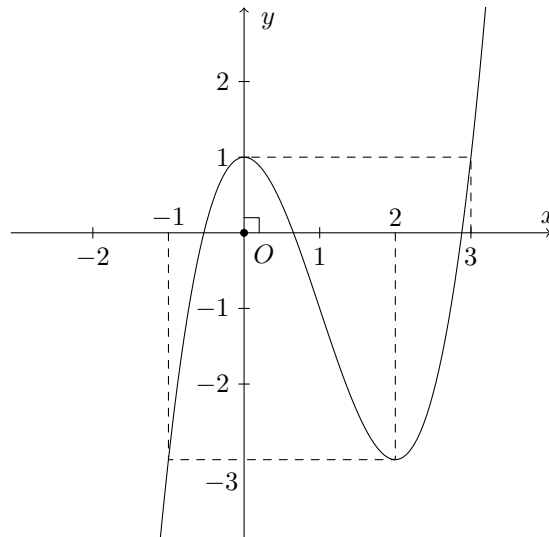
```
\begin{tikzpicture}[line cap=round,line join=round,x=1cm,y=1cm]
% Vẽ 2 trục, điền các số lên trục
\draw[->] (-3.08,0) -- (4.06,0);
\foreach \x in {-2,1,3}
\draw[shift={(\x,0)},color=black] (0pt,2pt) -- (0pt,-2pt) node[below] { $\x$};
\draw[->,color=black] (0,-4.06) -- (0,2.98);
```



```

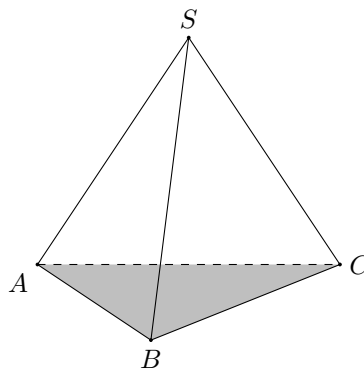
\foreach \y in {-2,-1,1,2}
\draw[shift={(0,\y)},color=black] (2pt,0pt) -- (-2pt,0pt) node[left] {\normalsize $\y$};
\draw[color=black] (3.8,.2) node[right] {$x$};
\draw[color=black] (.1,2.8) node[right] {\normalsize $y$};
\draw[color=black] (0pt,-8pt) node[right] {\normalsize $0$};
\clip(-4.08,-4.06) rectangle (4.06,2.98);
%Vẽ đồ thị
\draw[smooth,samples=100,domain=-4.08:4.06] plot(\x,{(\x)^3-3*(\x)^2+1});
% Vẽ thêm mấy cái râu ria
\draw[dashed] (2,0) -- (2,-3) -- (0,-3);
\draw[dashed] (3,0) -- (3,1) -- (0,1);
\draw[dashed] (-1,0) -- (-1,-3) -- (0,-3);
\node at (2,0.2) {2};
\node at (-1,0.2) {$-1$};
\node at (-0.3,-3.3) {$-3$};
\draw (0,0.2) -- (0.2,0.2) -- (0.2,0);
%Vẽ dấu chấm tròn
\fill (0cm,0cm) circle (1.5pt);
\end{tikzpicture}

```



## 5 Tô màu

Để tô màu một vùng (miền) nào đó, ta sử dụng lệnh `\fill`, tương tự như lệnh `\draw`, lệnh tô màu cũng có các tùy chọn và thuộc tính tương tự. Bạn chỉ cần thay `\draw` bằng `\fill` là sẽ có các kết quả tương tự như phần 4. Để minh họa, ta sẽ vẽ lại hình chóp ở phần đầu tài liệu, thêm màu sắc cho các mặt của hình chóp.



## 6 Đổ bóng

Đang viết...

## 7 Cắt xén hình

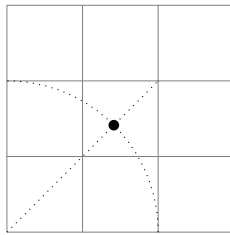
Đang viết...

## 8 Linh tinh

### 8.1 Hệ tọa độ cực

Để sử dụng tọa độ cực, polar coordinates, bạn sử dụng cú pháp dạng (`<angle:radius>`), ví dụ<sup>7</sup>

```
\fill (45:2cm) circle (2pt);
```



hoặc viết đầy đủ: `\fill (canvas polar cs:angle=45,radius=2cm) circle (2pt);`

### 8.2 Lệnh vòng lặp

Khi bạn muốn làm lặp đi lặp lại một thao tác nào đó, cách hay nhất là dùng vòng lặp `\foreach`, cú pháp như sau:

```
\foreach <biến> in {<danh sách>} <lệnh>
```

Câu lệnh `\foreach` sẽ thực hiện `<lệnh>` lặp đi lặp lại, mỗi lần tương ứng với một giá trị của `<biến>` lấy từ `<danh sách>` cho trước. Một `<biến>` có thể là một số, một tọa độ... và được đặt tên bắt đầu bằng dấu `\`, ví dụ biến `\x`, biến `\y`, biến `\toado...` Các `<lệnh>` ở đây chính là các lệnh vẽ hình của TikZ.

Ví dụ, lệnh `\tikz \foreach \x in {0,1,2,3} \draw (\x,0) circle (0.2cm);` sẽ vẽ bốn đường tròn có bán kính bằng 0.2 cm với tọa độ tâm là  $(\x, 0)$ , trong đó giá trị của `\x` được lấy lần lượt từ danh sách `{0,1,2,3}`. Kết quả thu được như hình sau:



Trong một vòng lặp, nếu bạn muốn thực hiện nhiều hơn một lệnh, thì các lệnh được nhóm lại với nhau, đặt trong cặp ngoặc móc. Hãy xem ví dụ sau để rõ.

```
\begin{tikzpicture}
\foreach \y in {0,1,2,3}
{
\draw (0,\y) circle (0.2cm);
\fill (0,\y) circle (0.1cm);
}
\end{tikzpicture}
```

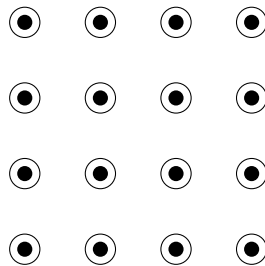
Hình vẽ thu được sẽ như thế này:

<sup>7</sup>Trong hình vẽ này, tôi đã thêm các đường lưới để bạn dễ hình dung vị trí của điểm cần vẽ.



Bạn có thể lồng nhiều vòng lặp lại với nhau, chẳng hạn:

```
\begin{tikzpicture}
\foreach \x in {0,1,2,3}
\foreach \y in {0,1,2,3}
{
\draw (\x,\y) circle (0.2cm);
\fill (\x,\y) circle (0.1cm);
}
\end{tikzpicture}
```



Nếu < danh sách > của bạn là một cấp số cộng, tức là phần tử đứng sau bằng phần tử đứng trước cộng thêm một số không đổi, bạn có thể không cần liệt kê hết các phần tử của danh sách, mà chỉ cần liệt kê vài phần tử và sử dụng dấu ba chấm `\cdots`.

- `\foreach \x in {1,2,...,6} {\x, }` sẽ cho ta danh sách 1, 2, 3, 4, 5, 6.
- `\foreach \x in {1,2,3,...,6} {\x, }` sẽ cho ta danh sách 1, 2, 3, 4, 5, 6.
- `\foreach \x in {1,3,...,11} {\x, }` sẽ cho ta danh sách 1, 3, 5, 7, 9, 11.
- `\foreach \x in {1,3,...,10} {\x, }` sẽ cho ta danh sách 1, 3, 5, 7, 9.
- `\foreach \x in {0,0.1,...,0.5} {\x, }` sẽ cho ta danh sách 0, 0.1, 0.20001, 0.30002, 0.40002.
- `\foreach \x in {a,b,9,8,...,1,2,2.125,...,2.5} {\x, }` sẽ cho ta danh sách a, b, 9, 8, 7, 6, 5, 4, 3, 2, 1, 2, 2.125, 2.25, 2.375, 2.5

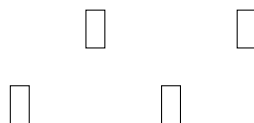
Một < danh sách > có thể không cần là một số, mà là một chữ cái, hoặc một cặp số, ví dụ:

- `\foreach \position in {(0,0), (1,1), (2,0), (3,1)}`
- `\foreach \x in {a,b,...,e}`

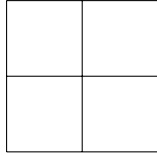
Chẳng hạn, câu lệnh

```
\tikz \foreach \toado in {(0,0),(1,1),(2,0),(3,1)}\draw \toado rectangle +(.25,.5);
```

sẽ cho ta:



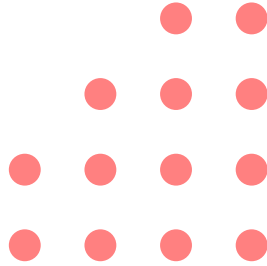
Một câu lệnh vòng lặp `\foreach` có thể được sử dụng bên trong các lệnh của đường `\path`, chẳng hạn `\tikz \draw \foreach \p in {1,...,3} {(\p,1)--(\p,3) (1,\p)--(3,\p)}`; sẽ cho ta:



Trong câu lệnh vòng lặp `\foreach` có thể sử dụng nhiều biến, các biến ngăn cách nhau bởi dấu `/`, chẳng hạn, lệnh `\foreach \x / \y in {1/2,a/b}` {“`\x`” and “`\y`”} sẽ thu được “1 and 2” và “a and b”.

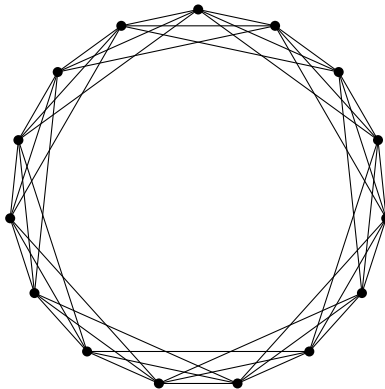
Trong khi thực hiện, ta có thể thoát khỏi vòng lặp bằng cách sử dụng lệnh `\breakforeach` nếu một <điều kiện> nào đó được thỏa mãn. Hãy xem ví dụ sau để rõ hơn.

```
\begin{tikzpicture}
\foreach \x in {1,...,4}
\foreach \y in {1,...,4}
{
\fill[red!50] (\x,\y) circle (6pt);
\ifnum \x<\y
\breakforeach
\fi
}
\end{tikzpicture}
```

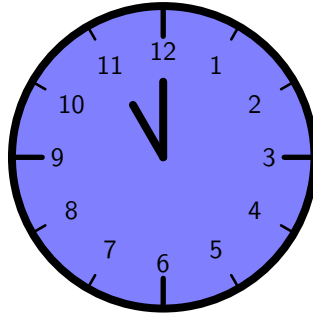


Một ví dụ nữa, kết hợp với cách sử dụng tọa độ cực:

```
\begin{tikzpicture}[scale=2.5]
\foreach \x in {18, 42,..., 360}
\node at (\x: 1)[circle, fill=black][scale=0.4]{};
\foreach \x in {114, 138,...,450} \foreach \y in {24, 48, 72}
\draw (\x:1)--(\x+\y:1);
\end{tikzpicture}
```



Chúng ta kết thúc phần này bằng một ví dụ, vẽ một cái đồng hồ:



Đoạn mã vẽ hình trên như sau:

```

\begin{tikzpicture}[cap=round,line width=3pt]
\filldraw [fill=blue!50] (0,0 ) circle (2cm);
\foreach \angle / \label in
{0/3,30/2,60/1,90/12,120/11,150/10,180/9,210/8,240/7,270/6,300/5,330/4}
{
\draw[line width= 1pt] (\angle:1.8 cm) -- (\angle:2cm) ;
\draw (\angle:1.4cm) node {\textsf {\label}};
}
\foreach \angle in {0,90,180,270}
\draw[line width=2pt] (\angle :1.6 cm) -- (\angle: 2cm);
\draw (0,0 ) -- (12 0:0 .8cm ); % hour
\draw (0,0 ) -- (90 :1c m); % minute
\end{tikzpicture}

```

### 8.3 Vẽ tiếp tuyến

Để vẽ tiếp tuyến với một đường cong, bạn cần sử dụng thư viện `calc` của TikZ.

### 8.4 Tìm giao điểm của hai đường

## 9 Vẽ bảng biến thiên

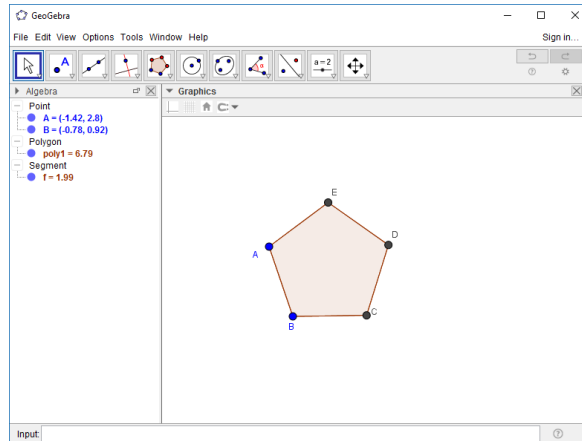
Bạn có thể sử dụng TikZ để vẽ bảng biến thiên, sử dụng thêm gói `tkz-tab`, nạp vào bằng lệnh `\usepackage{tkz-tab}`. Để tạo một bảng biến thiên, với hai dòng  $x, y$ , bạn sử dụng lệnh:

$x$	a	h	b	c
$y$	1	3	4	5

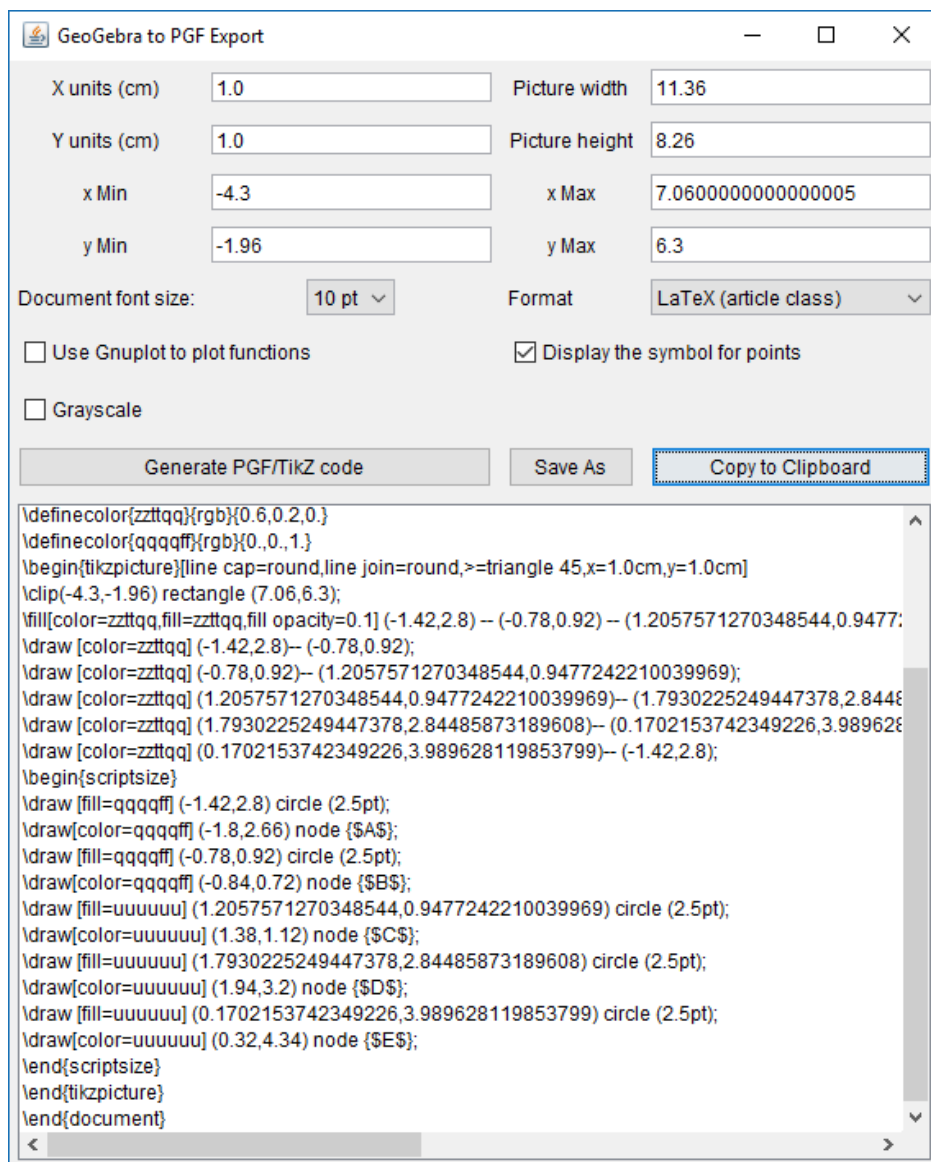
### 9.1 Vẽ hình bằng GeoGebra

Có một cách vẽ hình bằng TikZ mà không cần biết mã lệnh, đó là sử dụng phần mềm GeoGebra. Đây là một phần mềm khá đa năng, không chỉ đơn thuần là vẽ hình, GeoGebra còn dùng để giải toán Hình học phẳng và Hình học không gian, giải toán Đại số, Xác suất... Bạn có thể tải về ở đây <https://www.geogebra.org/download>

Sau khi cài đặt, bạn chạy chương trình GeoGebra, dùng các công cụ rất trực quan của nó để vẽ hình. Chẳng hạn ở đây tôi vẽ một hình ngũ giác đều:



Bạn chọn File--Export--Graphic View as PGF/TikZ sẽ hiện ra một cửa sổ như sau:



Bạn chọn Generate PGF/TikZ code, sau đó copy vào clipboard và dán vào trong tài liệu  $\text{\TeX}$  của bạn.

## 9.2 Sự khác nhau giữa path và draw

Sometimes we only need a virtual path, or a totally transparent path just to compute some coordinates, intersections, and so on. To do an invisible path we use `\path` and if you want to put some ink on it you use `\draw`.

Example: Here is an example where I used some paths to compute the intersections.

