Paths

Natalie Weaver

October 13, 2021

Contents

1 Path definition **1**

2 Actions **1**
2.1 Draw 2
2.2 Fill 2
2.3 Shade 3
2.4 Clip 4
2.5 Compound commands 4

3 Self-intersecting paths **5**

1 Path definition

A path is a series of coordinates; paths are the most basic building blocks of tikz pictures. We can define a path using the `\path` command within a `tikzpicture` environment. By default, nothing is drawn.

```
\begin{tikzpicture}
    \path (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}
```

2 Actions

To see the path, we can use the optional action argument, which takes one or more of the following values:

- `draw`
- `fill`
- `shade`
- `clip`

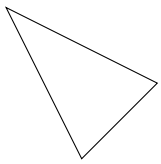
Tikz comes with built-in abbreviations for commonly-used combinations of these values.

2.1 Draw

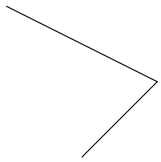
`draw` simply draws the line segments of the path, and does not require the path to be closed.

```
% Long form
\begin{tikzpicture}
    \path[draw] (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}

% Abbreviated form
\begin{tikzpicture}
    \draw (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}
```



```
\begin{tikzpicture}
    \draw (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```

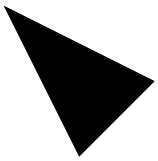


2.2 Fill

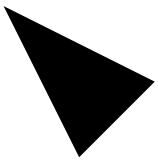
`fill` fills in the region enclosed by the path (which is required to be closed). If the path is not closed, tikz will automatically close it for us.

```
% Long form
\begin{tikzpicture}
    \path[fill] (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}

% Abbreviated form
\begin{tikzpicture}
    \fill (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}
```



```
\begin{tikzpicture}
    \fill (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```



2.3 Shade

`shade` shades the region enclosed by the path (which is required to be closed). If the path is not closed, tikz will automatically close it for us.

```
% Long form
\begin{tikzpicture}
    \path[shade] (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}

% Abbreviated form
\begin{tikzpicture}
    \shade (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```

2.4 Clip

`clip` defines the region where graphics are permitted to appear. Only graphics located inside of the (closed) path are drawn. Note that tikz statements are executed sequentially, so the `clip` command will only apply to statements below it.

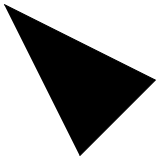
```
% Long form
\begin{tikzpicture}
    \path[clip] (-1, 0) -- (1, 0) -- (1, 1.5) -- (-1, 1.5) -- cycle;
    \path[fill] (0, 0) -- (1, 1) -- (-1, 2) -- cycle;
\end{tikzpicture}

% Abbreviated form
\begin{tikzpicture}
    \clip (-1, 0) -- (1, 0) -- (1, 1.5) -- (-1, 1.5);
    \shade (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```

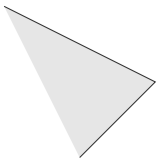
2.5 Compound commands

The `draw` action can be combined with `fill` or `shade` to produce the area and perimeter of a shape simultaneously. In the case of `\filldraw` we have to pass in some optional arguments to be able to visually distinguish the perimeter from the area. We'll revisit these graphics parameters in a future section. Notice that for an open path, the path is implicitly closed for the area portion but not for the perimeter portion of these compound commands.

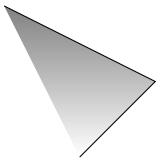
```
\begin{tikzpicture}
    \filldraw (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```



```
\begin{tikzpicture}
    \filldraw[color=black!80, fill=black!10] (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```



```
\begin{tikzpicture}
    \shadedraw (0, 0) -- (1, 1) -- (-1, 2);
\end{tikzpicture}
```



3 Self-intersecting paths

What happens if we draw to draw and/or fill a path that crosses over itself?