CSE 300 Practice on tikZ

Your Roll

July 20, 2019

1 picture environment

```
\begin{picture}(width, height)
     \put(starting point){object}
\end{picture}
```

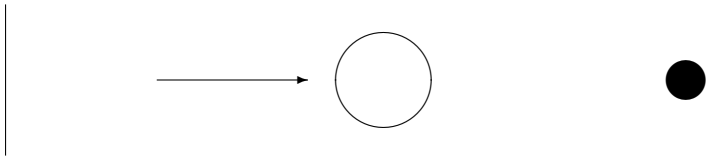
1.1 Different Objects

Line `\put(starting point){\line(direction){length}}`
Such as; `\put(0,0){\line(0,1){2}}`

Vector `\put(starting point){\vector(direction){length}}`
Such as; `\put(2,1){\vector(1,0){2}}`

Circle `\put(center){\circle{radius}}`
Such as; `\put(5,1){\circle{1.2}}`

Filled Circle `\put(center){\circle*{radius}}`
Such as; `\put(9,1){\circle*{1}}`



2 tikz package

There are mainly two ways for using tikz package

- ```
  \begin{tikzpicture}[options]
      <tikz code>
  \end{tikzpicture}
  ```

- `\tikz[options]{tikz codes}`
  ```

2.1 tikz codes

`\path[option1][option2] <specification>;`
This is the most basic command inside tikzpicture environment. We will first learn about the usage of option1.

2.1.1 Option1

draw, fill, pattern, clip, shade, use as bounding box.
Example1: `\path[draw], \path[fill], & so on`
Example2: `\path[draw, fill], \path[draw, clip], & so on`

Note: `\path` with these options can be combined to `option1`,

`\path[option1]`≡`\option1`

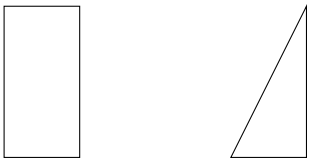
Example: `\draw, \fill, \drawclip, etc.`

Using \draw

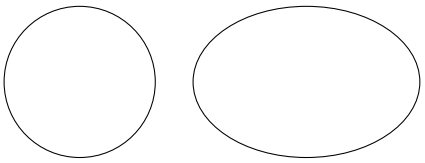
```
\draw (1,0) -- (0,0) -- (0,1);
```

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

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

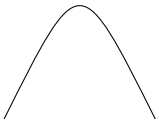


```
\draw (0,0) rectangle (1,2);
\draw (3,0)--(4,0)--(4,2)--cycle;
```



```
\draw (0,0) circle[radius=1cm];
\draw (3,0) circle[x radius = 1.5cm, y radius=1cm];
```

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

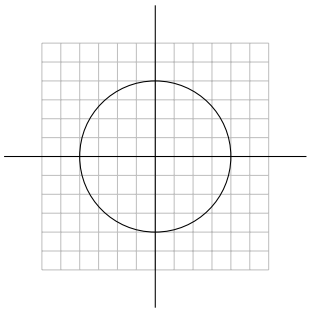


For drawing a curved line along with the start and end co-ordinate, you must provide the control point.

```
\draw (1,0) .. controls (2,2) .. (3,0);
```

We can also draw arcs.

```
\draw (0,0) arc[start angle=30, end angle=90, radius=1cm];
\draw (3,0) arc(30:90:3cm);
```



We need to use the grid command for drawing grids. We can add some additional features for grid in option2

```
\draw[step=0.25cm, opacity=0.25] (-1.5,-1.5) grid (1.5,1.5);
```

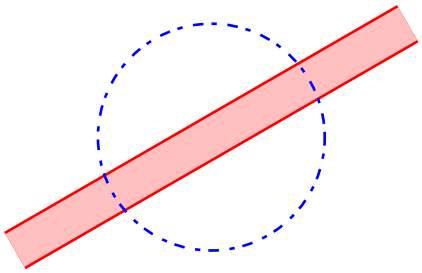
By default step is set at 1cm & opacity at 1.

There are numerous options available under option2. Everything under option2 can also be used as option for the environment as a whole.
Everything drawn under the following environment will be red:

```
\begin{tikzpicture}[color=red]
    \draw (-2,0)--(2,0);
    \draw (0,0) circle[radius=1cm];
\end{tikzpicture}
```

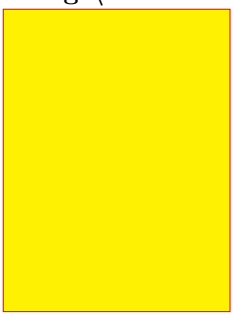
Some example options:

- line width=dim, ultra thin, very thin, thin(default), semithick, thick, very thick, ultra thick
- dash pattern=on dim. off dim. ..., solid, dotted, densely dotted, loosely dotted, dash dot, dash dot dot, double, double=color name, double distance=dim
- line cap= < rect, round, butt >, arrows= < start arrow kind - end arrow kind >



Using other commands

Using \fill



```
   \fill[color=yellow] (0,0) rectangle (3,4);
\draw[color=red] (0,0) rectangle (3,4);
```

Using \shading

```
\shade[shading=radial] (0,0) rectangle (3,4);
```

Different kinds of shading are there: axis, radial, ball

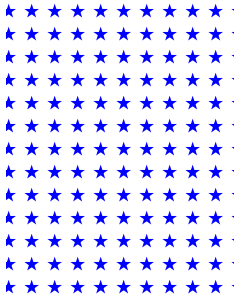
Besides, shading angle can be defined as `shading angle=angle value`

Shading color can be defined too:
left color= value, right color = value
top color=value, bottom color=value, middle color=value
inner color=value, outer color=value

For additional shading options use tikzlibrary shadings by writing the following command in your preamble: `\usetikzlibrary{shadings}`
Different options available under the shadings library are shading=color wheel, upper right=color name, etc.

Using \pattern
For patterns, we have to use another tikzlibrary called patterns. Some examples

are as follows:



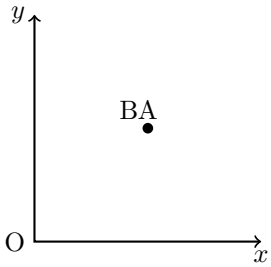
There are so many patterns. Some are: dots, fivepointed stars, vertical lines, grid horizontal lines, bricks, checkboard, checkboard light gray, etc.

We can also set pattern color=value.

```
\begin{tikzpicture}
  \pattern[pattern=fivepointed stars, pattern color=blue] (0,0) rectangle (3,4);
\end{tikzpicture}
```

Working with \node

```
\node[options](label){<text>};
\node[options](label) at (coordinate){<text>};
```



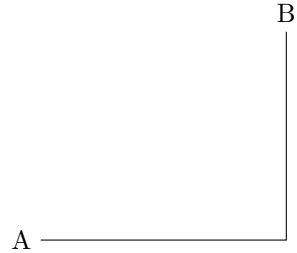
For getting the above figure the required code is as follows:

```
\begin{tikzpicture}
    \draw[<->,thick] (3,0)--(0,0)--(0,3);
    \fill (1.5,1.5) circle[radius=2pt];
    \node[below] at (3,0){$x$};
    \node[left] at (0,3){$y$};
    \node[left] at (0,0){O};
```

```
    \node[above](n1) at (1.5,1.5){A};
    \node[left] at (n1){B};
\end{tikzpicture}
```

2.2 Flow Chart

At first we will learn how to connect two points with two labelled nodes.

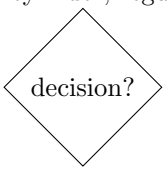


The corresponding code:

```
\begin{tikzpicture}
    \node[left](n1) at (2,0){A};
    \node[right] (n2) at (5,3){B};
    \draw (n1) -| (n2);
\end{tikzpicture}
```

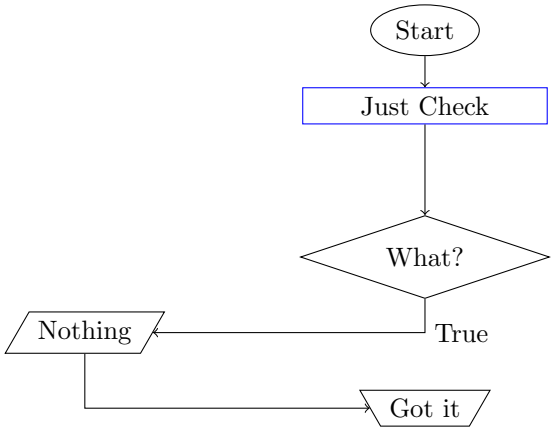
For generating different geometric shapes, we have to use tikzlibrary called 'shapes'. [`\usetikzlibrary{shapes.geometric}`]

Different shapes under the library are: diamond, ellipse, trapezium, semicircle, cylinder, regular polygon, kite, etc.




```
\node[draw, diamond]{decision?};
```

Example



- Ellipse: `\node[draw, ellipse] (label){text};`
- Diamond:

```
\node[draw, diamond, align=center|left|right, text width=dim,
aspect=x:y, xshift=dim, yshift=dim, <below of|left of|right of>=node,
inner sep=dim] (label){text};
```

- Trapezium:

```
\node[draw, trapezium, trapezium left angle=val,
trapezium right angle=val, xshift=dim, yshift=dim,
<below of|left of|right of>=node,] (label){text};
```

You can write your own object using the command `\tikzstyle{objectName}=[options]`. For example the rectangular box above within blue bounding box is drawn as custom object:

```
\tikzstyle{myBox} = [rectangle, text centered, text width=3cm, draw=blue]
```

The command used to draw the aforementioned object:

```
\node[myBox, below of=start] (mB){Just Check};
```

Now, let's focus on connecting the shapes: We already know how to draw a line with directional arrowheads between two nodes. We use the same command here:

```
\draw[->] (start)--(mB);
```

Here, the tricky part is if we want to draw connection from any definite side, we have to mention (east‖west‖north‖south) side of the particular object/node. Such as,

```
\draw[->] (l1.south) |- (l2.west);
```

If we want to add tag to any connection, we use the following code:

```
\draw[->] (decision.south) |- node[right|left|above|below]
{tag} (l1.east);
```