A Guide to LaTeX & TikZ for High School Students

Introduction

This document is dedicated to creating a collection of my knowledge of LaTeX and TikZ for both my personal use and as a reference guide to those wanting or trying to learn LaTeX . As a helpful note, a table of contents has been included for any specific references necessary. Appendices at the end contain a list of specific TikZ code to help in the creation of diagrams. Otherwise, the guide is laid out in a logical order to learn LaTeX .

Contents

1 Basics of LaTeX

1.1 What is LaTeX?

LaTeX is a mathematical typesetting language that mathematicians, engineers, and even writers use to create and format documents. The language has a particularly excellent ability to produce mathematical diagrams and equations with elegance.

1.2 Starting your first document

Before you begin, I suggest using the LaTeX editor '**Overleaf**', which is a free, online editor that shortcuts a lot of the development for you, as well as contains a built-in PDF viewer.

Every document in LaTeX has a class. There are many classes you may use, but for this guide, we will refer to mainly three: **article**, **standalone**, and **beamer**. The article class will be the main class you use, as it works with most documents. The standalone class will be used mainly for the creation of standalone images that can be saved for later use. Finally, the beamer class will be used for presentations. You express the class of document first in any LaTeX document. It is expressed with `\documentclass{type}`. The 'type' is a placeholder where 'article', 'standalone', 'beamer', or another class will appear.

Now that you have established your document type, you should give your document a title. To do this, there are a few steps involved. The command for a title is `\title{}`, where your title is inside the braces. For example, a document with a title of "Eggs" may have a command that looks similar to `\title{Eggs}`. After your title, you should clarify the author's name (that's you) and the date your document was created. To establish your author's name, the command `\author{}` can be used, again with your name in the braces. To establish the date, you can either type `\date{}` with the day or month of document creation inside the braces or `\date{\today}`, which will output the day at the time of compilation (this day will change every different day you compile your document).

Okay, now let's actually start making your document. In order to define what is contained **inside** your document, you use the command `\begin{document}` to indicate the beginning of your document and `\end{document}` to indicate where your document ends. The preamble is not included inside these commands. The first thing you should do inside your document is write `\maketitle`. This will organize those title components you wrote earlier into a specific style. From here, the rest of the document is at your liberty. We will discuss other components to your document in later sections, but for now, you are ready to begin writing your first LaTeX document.

1.2.1 Compiling your document

When you have finished creating your document, or periodically throughout creation of your document, you may want to compile your work into a PDF that makes viewing your potential errors extremely easy. If you are using Overleaf, simply clicking the green 'Recompile' button above your PDF will compile/recompile your document. In other editors, such as TeXmaker, the compile button appears in the form of an arrow. Simply click the button, and the document will compile. *Note: Errors may occur with compiling outside over Overleaf. If you receive these errors, you likely have not installed MiKTeX.*

1.3 Math Mode

Have you ever seen a math textbook say something like this?

$$\sum_{n=0}^{\infty} \frac{1}{n}$$

That is made using LaTeX! This is known as math mode. Specifically, that form is called 'display math mode'. Another form is 'inline math mode', which will appear like $x = 0$. We will get into detailed math mode commands later, but first we should know some of the basics. Inline math is wrapped in a pair of single $'s. For example, $5 + 5 = 10$ is written as `$5 + 5 = 10$`. As you can see, inline math appears as it sounds - in the line with the text around it. Display math mode, however, goes into a space **between** lines as seen above. Rather than single $'s, display math mode uses double $'s. For example, to write the same equation as we wrote earlier in inline but in display mode, we can write `$$ 5 + 5 = 10 $$`. This would appear as follows.

$$5 + 5 = 10$$

We will expand on math mode later, but for now, know that this is how you enter math mode.

1.4 Packages

Other things that go in the preamble of your document are 'packages'. Defined with the `\usepackage{}` command, these packages do a variety of things. For example, one package we will refer to often in this guide is **TikZ** , defined with `\usepackage{tikz}`. This imports the TikZ commands into your document so that you can create diagrams (we will get to what TikZ is later). Another package that I tend to use often is the **geometry** package. This package is capable of many things, but the main reason I use this package is for margins. An optional parameter to this package, the margin setting changes the look of your LaTeX document to whatever margins you have set.

1.4.1 Optional Parameters

When importing different packages into your LaTeX document, you may want to specify specific options for that document. These options are specified in brackets before the braces in a command. For example, the margins of this document are indicated by `\usepackage[a4paper, margin=0.8in]{geometry}`. *Note: The 'a4paper' argument changes the type of paper you write on. A list of possible paper types can be found online.* We will get into some more optional parameters later.

2 Commands for Normal Writing

In a LaTeX document, the way you format is not the same as in Microsoft Word, LibreOffice, or other writing software. Instead, LaTeX uses commands to perform specific tasks. In the following table, a few of these commands are listed with their function.

Command	Use of Command
`\\`	establishes a line break.
`\LaTeX`	writes LaTeX in the official format.
`\noindent`	removes the indent before a new paragraph.
`\verb!some_stuff!`	allows LaTeX code to appear as text.
`\color{}`	changes the color of text
`\fontfamily{font code}`	changes the font of text.
`\tableofcontents`	generates a table of contents.
`\textbf{some_stuff}`	generates bold text.
`\textit{some_stuff}`	generates italicized text.
`\section{name}`	creates a new section.
`\section*{name}`	creates a non-numbered section that is not added to the TOC.
`\subsection{name}`	creates a subsection
`\subsection*{name}`	creates a non-numbered subsection and does not add it to the TOC
`\subsubsection{name}`	creates a supersub section
`\subsubsection*{name}`	creates a non-numbered supersub section and does not add it to the TOC.
`\chapter{name}`	creates a new chapter.
`\chapter*{name}`	creates a non-numbered chapter and does not add it to the TOC.
`\begin{tabular}{c|c}` → `\end{tabular}`	creates a table.
`\hline`	adds a horizontal line on a table.

Command	Use of Command
`\include{stuff}`	appends contents of files into your document.
`\includegraphics[]{stuff}`	appends images into your document.
`\tiny`	very small text.
`\large`	slightly bigger than normal text.
`\Large`	decently large text.
`\LARGE`	big text.
`\huge`	bigger text.
`\Huge`	really big text.

Of course, there are many more commands, but I find myself using these the most often.

3 Math Mode

3.1 Common Commands

To create equations in math mode, there are many different symbols and commands to use. The following is a list of the most common commands for math equations, expressions, and functions.

Command	Use of Command
`\sum`	$\sum$
`\int`	$\int$
`{some_stuff}_{stuff}`	$\textbf{stuff}_{\textbf{stuff}}$
`\mathbf{stuff}`	creates words in math mode.
`\frac{x}{y}`	$\frac{x}{y}$
`\displaystyle`	sets the math to the display math mode style.
`\everymath{stuff}`	sets properties for all math in your document.
`{stuff}^{stuff}`	$\textbf{stuff}^{\textbf{stuff}}$
`\int_{a}^{b}`	$\int_a^b$
`\sum_{n=1}^{5}`	$\sum_{n=1}^{5}$
`\begin{array}` → `\end{array}`	creates an array of math (discussed later).
`\lim_{x\to 0}`	$\lim_{x\to 0}$
`\infty`	∞
`\frac{dy}{dx}`	$\frac{dy}{dx}$
`\pi`	π

Command	Use of Command
`\epsilon`	ϵ
`\delta or \Delta`	δ **or** Δ
`\varepsilon`	ε
`\alpha`	α
`\textrm{units}`	puts normal text into math mode.
`\sin`	puts sin in non italic text.
`\cos`	puts cos in non italic text.
`\tan`	puts tan in non italic text.
`\leq`	puts a $\leq$.

Again, of course, there are many, many more.

3.2 Arrays

Before we begin discussing arrays, we should clarify what an array actually is.

Arrays organize mathematical equations and ideas into a readable format.

Arrays are entered using the command `\begin{array}` that must be in math mode. I use them most often for writing out long equations in multiple, easy to read lines or for showing steps to reach a solution. Take for example the following array.

$$\begin{array}{rcl}
3x + 5 & = & 2x + 7 \\
x + 5 & = & 7 \\
x & = & 2
\end{array}$$

This array is written with the following code.

```
$$\begin{array}{rcl}
  3x + 5 & = &  2x + 7\\
  x + 5 & = &  7\\
  x & = & 2
\end{array}$$
```

Each part of this code is crucial in creating the array shown above. For example, the `{rcl}` tells LaTeX to place the items in each column to the right, center, and left, respectively. The & symbol denotes the end of the content in a column for that row. Finally, the line break `\\` at the end of each line tells LaTeX that the line has ended, and it can move onto the next line. Each row does not have to have content in every column. For example, the following array shows a series of equations where only one side changes.

$$\begin{array}{rcl}
\int_0^a \sqrt{x^2\frac{a-x}{a+x}}dx & = & \left(\frac{(a-x)^\frac{1}{2}(a+x)^\frac{1}{2}(x-2a)}{2} + a^2\arcsin\sqrt{\frac{1}{2} - \frac{x}{2a}}\right)\bigg\rvert_0^a \\
& = & (0 + a^2\arcsin 0) - (-a^2 + a^2\arcsin\sqrt{\frac{1}{2}}) \\
& = & 0 + (a^2 - a^2\frac{\pi}{4}) \\
& = & a^2(1 - \frac{\pi}{4})
\end{array}$$

This array is produced by the following code (ignore the commands you don't know - we will get to these later).

```
$$\begin{array}{rcl}
  \int_0^a \sqrt{x^2\frac{a-x}{a+x}}dx & = & \left(\frac{
  (a-x)^\frac{1}{2}(a+x)^\frac{1}{2}(x-2a)}{2}
  + a^2\arcsin{\sqrt{\frac{1}{2} - \frac{x}{2a}}} \right)\bigg\rvert_0^a\\\\
  & = & (0 + a^2\arcsin{0}) -
  (-a^2+a^2\arcsin{\sqrt{\frac{1}{2}}})\\& = & 0 + (a^2 -
  a^2\frac{\pi}{4})\\\\& = & a^2(1 -
  \frac{\pi}{4})\\\\\\
end{array}$$
```

As you can see, this array follows the same structure as the previous array, but incorporates different commands, such as integrals, arcsin, fractions, and square roots (Notable is the large bar in the first row, created by `\bigg\rvert_0^a` , which could also be expressed as `\right|_0^a`). Such are the basics of arrays - they are just ways to organize your equations into a readable format.

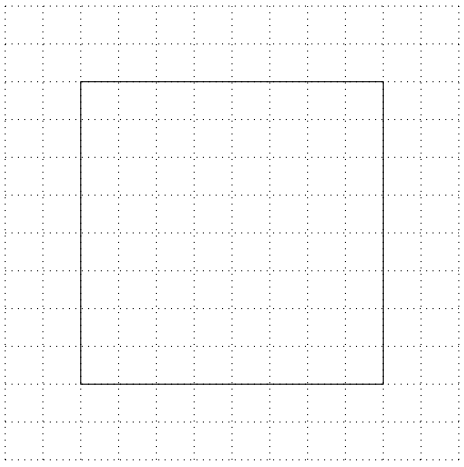
3.3 Math Mode Conclusions

While I could continue on the subject of math mode for many more pages, much of the joy of learning LaTeX comes from discovering these things on your own. Therefore, to finish this section, I will only leave you with the following reference for your future LaTeX endeavours (don't worry - this isn't the end of the guide!).

A Not So Short Introduction to LaTeX

4 Introduction to TikZ

Let us begin by defining what TikZ is. TikZ is a tool that can be used to create images and diagrams in LaTeX. We will walk through the creation of several diagrams in order to properly introduce this tool. The first command to know is `\begin{tikzpicture}` and `\end{tikzpicture}`. These two commands wrap every diagram that you create with TikZ - they tell LaTeX that you are creating a TikZ picture. Now, let us build a simple square.



I always like to begin by first drawing a grid with `\draw (x1, y1) grid (x2, y2)`. In this case, the grid begins at (0,0) and ends at (6,6). There are extra parameters I have also added to the grid here. First, I made

the 'step' 0.5. This means that each square has a side length of 0.5 rather than 1. Then, I used thin, dotted lines through the arguments of [thin, dotted]. This allows the grid to only be a guide. Usually, I delete the grid at the end, but for now, we will leave it. You will notice that every command ends with a ";". This is a crucial step. The semicolon indicates to LaTeX that the command has ended. The code for the square is as follows:

```
\begin{tikzpicture}
\draw[step=0.5, thin, dotted] (0,0) grid (6,6);
\draw (1,1) rectangle (5,5);
\end{tikzpicture}
```

Note that this is relatively simple code. In fact, much of TikZ is, in its core, simple code. For example, to draw a line segment, we write `\draw (a,b) -- (c,d)`, where a, b, c, and d are the coordinates of the line segment's endpoints. Similarly, an arc can be drawn with `\draw (a,b) arc (start:stop:radius)`. Many shapes can be drawn in much the same way as the square above, and irregular shapes can be created by connecting individual lines [it is important to note that the endpoint and starting point should be connected as well, that is, (a,b) – (c,d) – (e,f) – (a,b) should be written in the draw command.]

We will continue to learn by recreating some well-known logos. Take, for example, the following 2 images.

Both of these images were created using TiKz. We will begin by analyzing the code for the Target logo. The code used to produce this logo is:

```
\draw[red, fill=red] (0,1.35) circle (5.4/2);
\draw[white, fill=white] (0,1.35) circle (5.4*2/2/3));
\draw[red, fill=red] (0,1.35) circle (5.4/2/3);
```

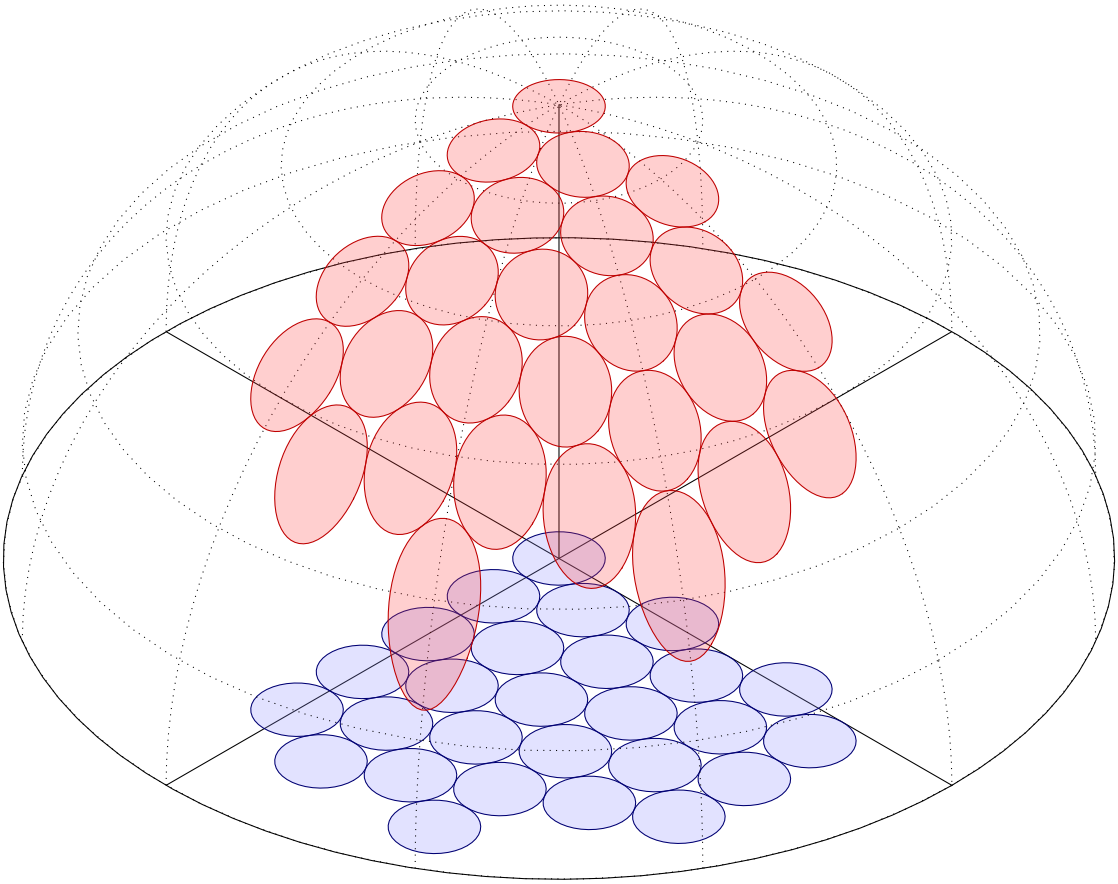
TiKz draws on top of anything that has already been drawn. We can think of this like an image in Microsoft Word. The image can be behind or in front of any other object in the document. Tikz chooses any new drawings to be in front of the previous drawings for us. As such, we create our circles starting from the biggest and going to the smallest. The first command clarifies that we want a red outline and fill (the fill is specified with `fill=red`). Then, we choose a point to center the circle at. (0, 1.35) is used here only to keep the logo in line with the other logo. The point is usually (0, 0) for just the one diagram. Next, we tell TiKz we want a circle (given by the `circle`) and its radius (5.4/2/3). Again, the radius does not really matter, as long as the other circles are scaled appropriately. This was done to match the height of the other logo. The ext two commands construct the other two circles in the exact same manner, with different radii and colors (white and red). Now, we move onto the code for the McDonald's™ logo.

```
\clip (-3, -1.3) -- (-1, -1.3) --
    (-1, -0.95) -- (1, -0.95) -- (1, -1.3) -- (3.7, -1.3) -- (3.7, 4.1) -- (-3, 4.1);
\draw[Goldenrod, fill=Goldenrod] (-1.3, -1) ellipse (1.7 and 5);
\draw[Goldenrod, fill=Goldenrod] (1.3, -1) ellipse (1.7 and 5);
\draw[Goldenrod, fill=white] (-1.3, -1) ellipse (1 and 4.7);
\draw[Goldenrod, fill=white]  (1.3, -1) ellipse (1 and 4.7);
\node[shape=circle, fill=white] at (3.4, -1) {\color{Goldenrod}{R}};
\draw[thick, Goldenrod] (3.4, -1) circle (0.2);
```

As you can see, we begin with the statement 'clip', which simply crops out a portion of the image we create and only shows that portion. We then asked TikZ to draw an ellipse, specifing its center and radii, which were (-1.3, -1), 1.7, and 5 respectively. Then we repeated this process two more times, drawing this logo in a very similar

manner to how we drew the Target logo, but changing our position each time. Note that because we wanted the white ellipses to be cutouts, we drew them last (so that they are on top).

This is, of course, nowhere near all that the TikZ package can do. It is great for creating beatiful graphs and diagrams in LaTeX. I encourage you to research this topic more so that you can create some beatiful TikZ diagrams similar to the one below, showing the projection of circles onto a spherical surface, which was retreived from TeXample.net and authored by Mark Wibrow.



5 Conclusions

To conclude, I would like to again encourage you to pursue learning LaTeXon your own. There is no better method for writing beatiful, complex mathematical documents. Upon exploration, you may even find that there is an entire document class of LaTeXcalled beamer, which allows you to create incredible math presentations (or even just presentations without the math part). This is far from a comprehensive guide. There is much in LaTeX, and especially in TikZ, that this guide did not cover. I challenge you to become an expert at LaTeXand discover its magic, as I have.