When an example needs a special package, I’ve put a `\usepackage` statement in code for the example. The one exception is `\usepackage{amsmath}`. I always use this, and so should you, if only to get the `align` environment.

**XeLaTeX**

Here’s the example of an XeLaTeX document from my talk:

```
\documentclass[12pt]{article}
\font\zapfino="Zapfino" at 12pt
\begin{document}
  Normal font.
  \zapfino Unusual font.
\end{document}
```

Just remember to run it with the command `xelatex file.tex` rather than `latex file.tex` or `pdflatex file.tex`.

You can set different font options like this:

```
\font\garamonditalics="Garamond/I" at 12pt
\font\garamondbold="Garamond/B" at 12pt
```

There are also a lot of options specific to the font. See the XeLaTeX reference for more details.

**Package microtype**

Just add `\usepackage{microtype}`. If you’re feeling conservative and want to reduce font stretching, try `\usepackage[stretch=10,shrink=10]{microtype}`.

**Spacing after sentence**

If you have a lower-case letter followed by a period followed by a space, LaTeX assumes that it’s the end of a sentence. If it’s not, replace the space by `\` or by `~`.

If you use a capital letter followed by a period followed by a space, LaTeX assumes that it’s an abbreviation instead of treating it as the end of a sentence. If it is the end of a sentence, use `\` instead of just a period.

Here’s an example of all of these done right:

Prove that $f$ is well-defined (i.e. that this construction makes sense).
We computed these with SAGE. Results are shown below.

There really should be a comma after ‘‘i.e.’’ in the first example.

You can also invoke the nuclear option: just write \frenchspacing, and you get the same spacing after sentences as everywhere else. This is actually how most non-technical writing is typeset, I think.

Non-breaking spaces

Use a ~ to make a non-breaking space:

We apply Lemma 2.

Find this on pp. 13--17.

Suppose that the graph $G$ is simple.

Address me as Dr. Johnson, please.

Dashes

Use --- for parenthetical statements, and don’t put spaces around them unless you write for the New York Times.

The controversy---if you could call it that---fizzled out quickly.

Use -- for ranges of numbers or dates.

I’ll be around from January 3--February 7.

Use a single dash for most other things:

Let $M$ be a 3-manifold.

When you’re connecting things with a dash as in the above example, and one of them is more than one word, use --:

New Yorkers are known for adopting a New--York--centric view of the world.

Some people also use -- for connecting words that don’t modify each other, like love–hate relationship. I think this looks a little bit nicer than love-hate relationship.
Spacing in math mode

To add space in math mode, use the following commands:

\begin{align*}
& a \! b \\
& a b \\
& a \ b \\
& a \quad b \\
& a \quad b \\
& a \quad b \\
& a \quad b \\
& a \quad b \\
& a \quad b \\
\end{align*}

(The ampersands are just to left-align each line.) Here’s the result:

\[ ab \]
\[ ab \]
\[ a b \]
\[ a b \]
\[ a b \]
\[ a b \]
\[ a b \]
\[ a b \]
\[ a b \]

The \( \! \) is a negative space, which can be useful sometimes. (By the way, when you typeset displayed mathematics, you should use \begin{align*}, and never \begin{eqnarray*}, if you don’t want weird spacing around your equal signs.)

Separate \( dx \) from the rest of an integral with a \( \, \). You can write it as \( d x \) by writing \texttt{mathrm}(d)x, but only if you’re British.

To get the right spacing when you use \( | \) to separate things in a formula, use \( \mid \). If you need a large \( | \), you’ll have to add spaces on your own:

\begin{align*}
\E[f(X_t) \mid X_0=1]
\end{align*}

\begin{align*}
\E \left[ f(X_t) \mid \sum_{i=1}^n X_i^2 \! \right. \\
\left. \sum_{i=1}^n X_i=0 \right]
\end{align*}

This produces:

\[ \E[f(X_t) \mid X_0 = 1] \]
\[
E \left[ \sum_{i=1}^{n} X_i^2 \right] \quad \left| \sum_{i=1}^{n} X_i = 0 \right.
\]

\textbf{colon}

Always use \texttt{colon} instead of : when you use colons in math. For example,

Let $f: X \to Y$ be defined as follows:

Consider the set $\{ x : f(x)=0 \}$.

\textbf{Absolute values}

To typeset absolute values and norms, make macros like this:

\begin{verbatim}
\newcommand{\abs}[1]{\left\lvert #1 \right\rvert}
\newcommand{\norm}[1]{\left\lVert #1 \right\rVert}
\end{verbatim}

Just using | usually is fine, but every once in a while it messes up the spacing. Also, really don’t write || instead of \texttt{lVert} and \texttt{rVert}. Otherwise you’ll get ||x|| instead of $\|x\|$. 

\textbf{Don’t use eqnarray}

The \texttt{eqnarray} environment has different spacing from the usual equation environments. For example,

\begin{verbatim}
\[ 2+3 = 4 \]
\begin{align*}
2 + 3 &= 4 \\
3 + 4 &= 7
\end{align*}
\end{verbatim}

looks like this:

\[
2 + 3 = 4 \\
2 + 3 = 4 \\
3 + 4 = 7
\]

It works perfectly with an \texttt{align} environment, though:

\begin{verbatim}
\begin{align*}
2 + 3 &= 4 \\
3 + 4 &= 7
\end{align*}
\end{verbatim}
\[2 + 3 = 4\]
\[3 + 4 = 7\]

There are other reasons not to use `eqnarray`. See the article *Avoid eqnarray!* at http://tug.org/pracjourn/2006-4/madsen/madsen.pdf.

**Sums and smashing**

To make \LaTeX{} ignore the width of the subscript of a sum:

\begin{verbatim}
\usepackage{mathtools}
\begin{align*}
  S_n &= \smashoperator{ \sum_{j = V + W + 2i}^n } a_j x_j^2 \\
\end{align*}
\end{verbatim}

This gives you

\[ S_n = \sum_{j=V+W+2i}^n a_j x_j^2 \]

instead of

\[ S_n = \sum_{j=V+W+2i}^n a_j x_j^2 \]

To make a multiline subscript for a sum, use `\substack`:

\begin{verbatim}
\begin{align*}
  \sum_{\substack{ i \in X, \\ j \in Y \}} x_{ij}
\end{align*}
\end{verbatim}

\[ \sum_{i \in X, j \in Y} x_{ij} \]

I don’t know how to get the subscript aligned right. The audience at my talk had some suggestions, but I couldn’t get any of them to work perfectly (but email me if you can!). You can adjust by hand using the spacing commands, and you can also try using “phantom” letters, like this:

\begin{verbatim}
\begin{align*}
  \sum_{\substack{ i \in X, \\ \phantom{i} j \in Y \}} x_{ij}
\end{align*}
\end{verbatim}

\[ \sum_{i \in X, \phantom{X}, j \in Y} x_{ij} \]
\[ \sum_{i \in X, j \in Y} x_{ij} \]

What’s happening here is that \texttt{\phantom{i}} and \texttt{\phantom{X,}} leave space for \texttt{i} and \texttt{X}, but don’t actually typeset them. The \texttt{\mathllap{j}} and \texttt{\mathrlap{Y}} do the opposite, typesetting \texttt{j} and \texttt{Y} but not having them take up any space.

**The enumerate package**

If you want to number your \texttt{enumerate} environment with something other than the default, do it like this:

\begin{verbatim}
\usepackage{enumerate}

\begin{enumerate}[i)]
  \item first
  \item second
\end{enumerate}

i) first
ii) second
\end{verbatim}

**The exam document class**

The \texttt{exam} document class is a wonderful thing, and I’ll just refer you to its documentation, which you can find by typing \texttt{texdoc exam} into any terminal.

**TikZ**

Here are all the examples from my talk. You need a \texttt{\usepackage{tikz}} at the top of your document for them.

\begin{verbatim}
\begin{tikzpicture}[scale=2]
  \fill[blue] (0, 0) -- (0, 1) -- (1, 1) -- (0,0);
  \draw[thick, rounded corners] (2, 0) -- +(0,1) -- ++(-30 : 1);
\end{tikzpicture}
\end{verbatim}
\begin{tikzpicture}
  \node (toby) at (0, 0) {Toby};
  \node[shape=circle, draw] (lindsay) at (6, 0) {Lindsay};
  \node[shape=circle, fill, inner sep=2pt] (point) at (3, -1) {};
  \draw (toby) -- (lindsay) -- (point);
  \draw[->] (point) to[bend left] node[auto] {$\varphi$} (toby);
\end{tikzpicture}

\begin{tikzpicture}
  \node (toby) at (0, 0) {Toby};
  \node[draw] at (6, 0) {Lindsay};
  \node[circle, fill, inner sep=2pt] at (3, -1) {};
  \draw (toby) -- (6, 0) -- (3, -1);
  \draw[->] (3, -1) to[bend left] node[auto] {$\varphi$} (toby);
\end{tikzpicture}

You can also use TikZ to make diagrams that just sit in the text. Here's the example from my abstract:
\usetikzlibrary{decoration.text}

TikZ lets you do
\begin{tikzpicture}[overlay]
  \draw decorate[decoration={
    text along path,text={ridiculous things like this.}}] {(0,0)--++(1,0) ..controls +(2,0) and +(1,1).. ++(2,-2)};
\end{tikzpicture}

To see how this works, let's look at it without the “text decoration”:
\usetikzlibrary{decoration.text}

TikZ lets you do
\begin{tikzpicture}[overlay]
  \draw (0,0) -- ++(1,0) ..controls +(2,0) and +(1,1).. ++(2,-2);
\end{tikzpicture}

The overlay command tells TikZ not to make space for the picture as it usually does. (Try typesetting it without this to see what goes wrong.)

Here’s a more useful example. I wanted to have a labeled arrow within the text of a paper, and when I tried to typeset it with \texttt{xypic} or anything else I tried, I couldn’t make it look exactly right. For instance, the label was too high, and \LaTeX had to give more space than usual to the line. Here’s what I came up with:
\usetikzlibrary{calc}

\newcommand{\labeledarrow}[3]{
  \tikz[baseline]{
    \node[anchor=base west] (n1) at (0,0) {$#1$};
    \path (n1.base east)+(1.1,0) node[anchor=base west] (n2) {$#2$};
    \draw[->] (n1.mid east) -- node[auto,font=\scriptsize,yshift=-0.07cm] (lb) {$#3$} (n2.mid west);
  }

\labeledarrow{L}{A}{TEX}
If \text{labeledarrow}{a}{b}{f}, then it holds that \text{labeledarrow}{b}{a}{f^{-1}}.

If $a \xrightarrow{f} b$, then it holds that $b \xrightarrow{f^{-1}} a$. 