Code On My Mind - LaTeX Cheat Sheet

For beginners I recommend using templates and pasting your content in. Experiment with various formatting styles, maybe create a table and see where it goes from there. For a basic cheat sheet see: https://wch.github.io/latexsheet/. This is actually more of a collection of useful stuff with no claim of being complete.

Structure

Organising your document is important. These commands will be useful friends in doing so:

```
\part{Part Title}
\section{Section Title}
\subsection{Small Section Title}
\subsubsection{Even Smaller Section Title}

Some Text.

Some Text on a new line.
```

Part I

Part Title

1 Section Title

1.1 Small Section Title

1.1.1 Even Smaller Section Title

Some Text.
Some Text on a new line.

I recommend using a blank line to achieve a line break to make your code easier to read. Some prefer the double-backslash-method \\
All commands are available with a $*$, which stops the automatic numbering. For example `\section{Title}` turns into `\section*{Title}`. Using the titlesec package will enable customisation of these structure elements.

Lists and Tables

```
\begin{itemize}
    \item Unordered
    \item List
\end{itemize}
```

- Unordered
- List

```
\begin{enumerate}
    \item Ordered
    \item List
\end{enumerate}
```

1. Ordered
2. List

```
\begin{tabular}{l | c | r}
    Table & with & text-alignment \\
    \hline
    left & center & right
\end{tabular}
```

Table	with	text-alignment
left	center	right

Lists are implemented using the `itemize` or `enumerate` environments. The parameter `{lcr}` for the `tabular` environment determines the alignment of the columns. Pipe symbols create a vertical line between columns - `\hline` creates a horizontal line between rows. In case tables are too complicated, one can always use a table generator.

Figures

Most images are inserted using a figure. A short caption describes the image. A label will make referencing the image effortless `\ref{label_name}`.

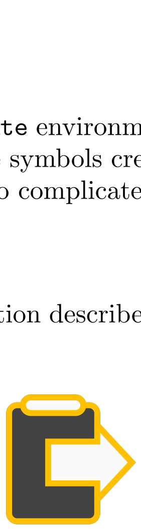


Figure 1: This text describes the image

```
\begin{figure}[ht]
    \begin{center}
        \includegraphics[width=.1\textwidth]{codeonmymind.png}
    \end{center}
    \caption{This text describes the image}
    \label{label_name}
\end{figure}
```

The optional parameter `[ht]` determines the figures float property. For a complete guide see this wiki.

Formatting Code

The listings package is the one I found most helpful. First

```
\usepackage{listings}
\lstdefinestyle{codeblock} {
    breaklines=true,
    basicstyle=\small\tt,
    keywordstyle=\color{RubineRed},
    showspaces=false,
    showstringspaces=false,
    stringstyle=\color{blue},
    commentstyle=\color{ForestGreen}
}
```

And for the actual code snippet:

```
\begin{lstlisting}[style=codeblock, language=Python]
   ...
\end{lstlisting}
```

For inline code I recommend `\verb|...|`, since it also escapes any LaTeX special characters.

New Command: TO DO

New commands are always defined before the actual document begins. Define name, number of arguments and the actual macro.

```
\newcommand{\todo}[1]{
    \textcolor{red}{$\bigstar$TODO: #1}
}
```

Usage:

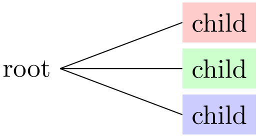
```
\todo{Include to-do command in cheat sheet}
```

★TODO: Include to-do command in cheat sheet

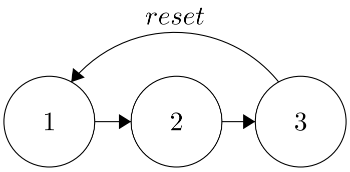
Graphs

For tree graphs, the forest package provides a compact syntax.

```
\begin{forest}
    [root[child,fill=red!20][child,fill=
       green!20][child,fill=blue!20]]
\end{forest}
```



A graphical method of creating any sort of graph is the fsm tool:



Detexify

A really awesome tool. Draw any symbol to search for the corresponding latex command:
`http://detexify.kirelabs.org/classify.html`

German

`\usepackage{epsf,german}` will convert the language to german and will take care of any Umlaute issues.