LaTeX- Cheat Sheet

A collection of useful latex codes for Computer Science

Version: 1.0

Last updated: October 29, 2024

Joakim Iversen

Contents

1 Introduction

The creation of this document is for students studying Computer Science or interested in the depth of LaTeX and the possibilities therein.
The document will have different section with commands and techniques to make the layout and styling of the paper easier and more pleasant to write.

If you have suggestions for things to add, or stumble across anything that will be useful to have in the sheet. Please contact: au761308@post.au.dk

Hope you enjoy using the Cheat Sheet ☕

2 Layout

This section will have focus on some basic styling and layout in latex, such as styling text, inserting images and how to make different types of lists.

2.1 Text

Writing text in LaTeX in the most basic form is just as easy as one might expect. You just have to write in the editor. But there are some basic functions we can add to this, to make things in the text stand more out.

It is these function we will learn about in the following section.

The following section covers these usepackages and commands (Click on any part to get to the choosen subject):

Usepackage mentioned in this section:

Usepackage implementation	What the usepackage does
\usepackage{*FONT-PACKAGE-NAME*}	Used to import font for your document

2.1.1 Size

To set the font-size for your document you have to change the \documentclass{...} to include the following \documentclass[*FONT-SIZE*]{...}. When declaring the font-size you have to use the unit "pt" which stands for points (A typography unit).

Besides setting the document font-size LaTeX has 10 different size modifiers to use inside the basic text. These modifiers are the following:

\tiny	Lorem Ipsum
\scriptsize	Lorem Ipsum
\footnotesize	Lorem Ipsum
\small	Lorem Ipsum
\normalsize	Lorem Ipsum
\large	Lorem Ipsum
\Large	Lorem Ipsum
\LARGE	Lorem Ipsum
\huge	Lorem Ipsum
\Huge	Lorem Ipsum

The sizing is gonna be active for the duration of the line you are typing. This text is therefor gonna be foot note size until i change it back using \normalsize.

It is therefor not enough to make a line change by pressing enter or using the \\line break.

Examples:

Setup:

\documentclass[18pt]{article}

Output:

This command sets up a document with a font size of 12pt.

Setup:

\tiny

Output:

This text is using the tiny command

2.1.2 Family

The font-family is the type of font you use in your LaTeX document. The standard font-family for a LaTeX document is *Computer Modern Typeface Family*. This includes different options for serif, sans serif and monospaced (Typewriter). You can switch between these throughout your document by using the following commands

\textrm	Lorem Ipsum
\textsf	Lorem Ipsum
\texttt	Lorem Ipsum

If you want to see more fonts you can check under the section Miscellaneous and font-family

When you have to use another font, as one shown under the section Miscellaneous, you have to import the "font package name" by implementing \usepackage{*Font-Package-Name*}. This allows you to use this font throughout your document.

If your otherwise just want to use a font in your text and not through out the whole document, you can use the code \fontfamily{*Font-Code*}\selectfont The first part of this code is to choose the font you want, where you have to insert the Font-Code as seen in the table in section 3.1.1. The second part is then sets the font, this means that whitout the \selectfont the font will not be set. If you only want the font change for a specific part of your text. you can do that the following way: {\usepackage{*Font-Package-Name*}*The Text you want to be affected by the font*}

Examples:

Setup:

\usepackage{*lmodern*}

Output:

This Commands makes your document use the Latin Modern font-family

2.1.3 Styles

There are some basic styles that we are used to having available to us in the text editors we are using on a normal basis like word and notepad. The most basic of these are *Italic*, **Bold**, and <u>Underline</u>.

To use these you have the following three commands

\textit	*This is Italic*
\textbf	**This is Bold**
\underline	<u>This is underlined</u>

If you want to find more examples of way to style your text in LaTeXyou can check under Miscellaneous and font-style

2.2 Images

The following section covers these usepackages and commands (Click on any part to get to the choosen subject):

Usepackage mentioned in this section:

Usepackage implementation	What the usepackage does
\usepackage{*graphicx*}	Used to add and manipulate images
\usepackage{wrapfig}	Used to allow text wrapping around images

When working with images there are a few basic usepackages we want to have imported at the beginning of our document.

The following list covers the basic usepackages and commands:

\usepackage{graphicx}	Allows us to insert and manipulate images in our document
\graphicspath{*PATH-FOR-IMAGE-FOLDER*}	Informs of the folder where your images is stored
\includegraphics{*NAME-OF-IMAGE*}	Inserts an image into your document

With these three commands can we insert and modify images in our document. We use the \includegraphics to import the picture:

This is the most basic way of inserting an image. And there are a few different things we can do to make the image look better.

2.2.1 Resizing and Rotating

If we want to change the size or orientation of an image we can pass in different settings in the format as shown bellow

$$\text{includegraphics}[FORMAT\text{-}SETTINGS]\{\dots\}$$

This means that you have to import the image into the document as normal but add the [...] with your modifiers before the image you're importing.

There are "three" settings we can use with our image:

rotate=XXX	Rotates the image XXX degrees
width=XXX	Sets the width of the image to XXX
height=XXX	Sets the height of the image to XXX
scale=XXX	Scales the image XXX-times

The gray box is not counted, since it has the same effect as the width, and height setting. You can combine all these settings on the same image, by separating them with a ','. The values we give to the settings can be given with a selection of different units. To see the complete list you can look under Miscellaneous and Image Units

2.2.2 Positioning

Now that we know how to insert images into our document using the \includegraphics{} command, we will look into the positioning of the image. To do this we will have to use an environment called *figure*. (We will cover environments in a later section - so just ignore that for now)

Setup:

Example
\begin{figure}[*float-value*]
\includegraphics[*setting=value*]{*IMAGE-NAME*}
\end{figure}

Output:

Entered
\begin{figure}[H]
\includegraphics[scale=.1]{Images/Programming.png}
\end{figure}

Here we use the *figure* environment to control the positioning of the image. The positioning is controlled by the *float-value*. There are six different parameters that we can use to control the positioning:

Parameter	Position
h	Place the float *here* - approximately where it is placed in the source text (However, not exactly at the spot)
t	Position at the *top* of the page
b	Position at the *bottom* of the page
p	Put on a special *page* for float only
!	Override internal parameters LaTeXuses for determining "good" float position
H	Places the float at precisely the location in the LaTeXsource code. This requires the *float* package, and though may cause problems occasionally. This is in a way the same as using h!

Note that the '!' can be used in combination with the rest of the parameters, but not H.

2.2.3   Wrapping of text

It's also possible to wrap text around a picture. To do this we will use the "wrapfigure" environment. To be able to use this you have to implement the usepackage "wrapfig".

Setup:
\begin{wrapfigure}[LINES]{*PLACEMENT-PARRAMETER*}[OVERHANG]{*WIDTH*}
\centering
\includegraphic[*SETTING-PARAMETERS*]{*IMAGE-NAME*} \end{wrapfigure}

Output:

This is a blank text to show to possibilities of wrapping a text around an image. This text has zero actual meaning, and is just trying to fill as much space as possible.

Note that the wrapfig environment takes 4 different input parameters. The LINE parameter refers to the number of lines there should be narrowed besides the picture. In our example above we have set this to be 3, and as you can see, we have three lines that has been narrowed.

The next on is the placement of the image. Here we have 4 different values:

Position parameter	Placement of image
r	Right side of the text
l	Left side of the text
i	Inside edge-near the binding (in a *twoside* document)
o	Outside edge-far from the binding

Each of these does also have the equivalent letter in the uppercase version. When using the uppercase you allow the image to float, thus the lowercase version means to place the image *exactly* here.

The next parameter is the overhang. This is a fine-tuning of the image placement and can be used to give an image some "overhang" in the text area.

The last parameter is the width of the width of the "box" given to the image inside the environment. Because this only refers to the box given to the image, and not the image itself. It's good practice to let the image inside the wrapfig just be a tiny bit smaller than the wrapfig environment. That is how you make sure the text and image are not overlapping, and that you get a small white space between image and text.

2.2.4   Captioning, labelling and referencing

When you have inserted a figure, image or anything else into our document we want to give the reader a small explanation to what it is this figure resembles. To do this be use the code \caption{*IMAGE-CAPTION*}. We normally place the caption inside an environment with the figure we want to caption.

Setup:
\begin{figure}
\includegraphic{*IMAGE-NAME*}
\caption{Image of Computer}
\end{figure}

Output:

Figure 1: Image of Computer

As you can see we have the caption written out below the image along with *Figure 1*. The reason of the figure being numbered as number one, is because it is the first figure being captioned.

You also have a function in latex to label your figures and images. When doing this, you use the \label{*CAPTION-TITLE*} When we label our images and figures, it is not possible to see what you write in the label. But the label allows you to reference a figure later in your document. You can choose your own system for labeling, but if you don't know what to do I have inserted a table down below here, for how I in generally label my figures:

How I label	What figure
img:*IMAGE_DESCRIPTION*	Image
fig:*FIGURE_DESCRIPTION*	Figure
secc:*SECTION_DESCRIPTION*	Section
tab:*TABLE_DESCRIPTION*	Table
gra:*GRAPH_DESCRIPTION*	Graph

Table 1: Labeling Guide for Figures and Tables

By using this system I know what it is my label is referring to, whether it is a figure, image, section or table. And this is useful when you want the reference a thing from your document. I always split the label up with '_' if I use more words. This is just personal preference, and you can make a label with spaces in it if you would like.

If you want to reference a figure you have a few different options to use, each having its own unique advantages

Command	Usage
\ref{*FIG-NAME*}	Used to reference the figure number as shown below in the caption
\pageref{*FIG-NAME*}	Used to reference the page number a figure is on
\hyperref{*FIG-NAME*}{*TEXT-TO-BE-SHOWN*}	Used to create a hyperlink between placement of the reference and the figure

2.3 Lists

3 Miscellaneous

3.1 Font

3.1.1 Font-Family

Font	"Font package name"	"Font code"	Example
Computer Modern Roman		cmr	The quick Brown Fox jumps over the lazy dog
Latin Modern Roman	lmodern	lmr	The quick Brown Fox jumps over the lazy dog
Latin Modern dunhill	lmodern	lmdh	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Termes	tgtermes	qtm	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Pagella	tgpagella	qpl	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Bonum	tgbonum	qbk	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Scholar	tgschola	qcs	The quick Brown Fox jumps over the lazy dog
Times	mathptmx	ptm	The quick Brown Fox jumps over the lazy dog
Utopia / Fourier	utopia/fourier	put	The quick Brown Fox jumps over the lazy dog
Palatino	palatino	ppl	The quick Brown Fox jumps over the lazy dog
Bookman	bookman	pbk	The quick Brown Fox jumps over the lazy dog
Charter	charter	bch	The quick Brown Fox jumps over the lazy dog
Computer Modern Sans Serif		cmss	The quick Brown Fox jumps over the lazy dog
Latin Modern Sans Serif	lmodern	lmss	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Adventor	tgadventor	qag	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Heros	tgheros	qhv	The quick Brown Fox jumps over the lazy dog
Helvetica	helvet	phv	The quick Brown Fox jumps over the lazy dog
Computer Modern Typewriter		cmtt	The quick Brown Fox jumps over the lazy dog
Latin Modern Sans Typewriter	lmodern	lmtt	The quick Brown Fox jumps over the lazy dog
T$_E$XGyre Cursor	tgcursor	qcr	The quick Brown Fox jumps over the lazy dog
Courier	courier	pcr	The quick Brown Fox jumps over the lazy dog

3.1.2   Font-Styles

Style	Command	Switch command	Output
medium	\textmd{*INPUT*}	\mdseries	*The quick brown fox jumps over the lazy dog*
bold	\textbf{*INPUT*}	\bfseries	**The quick brown fox jumps over the lazy dog**
upright	\textup{*INPUT*}	\upshape	The quick brown fox jumps over the lazy dog
italic	\textit{*INPUT*}	\itshape	**The quick brown fox jumps over the lazy dog**
slanted	\textsl{*INPUT*}	\slshape	*The quick brown fox jumps over the lazy dog*
small caps	\textsc{*INPUT*}	\scshape	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

3.2   Image

3.3   Image Units

pt	A point, is the default length unit approximately 0.315mm
mm	A millimeter
cm	A centimeter
in	An inch
ex	The height of an **x** in the current font
em	The height of an **m** in the current font
\columnsep	The distance between columns
\columnwidth	The Width of the column
\linewidth	The width of the line in the current environment
\paperwidth	The width of the paper
\paperheight	The height of the paper
\textwidth	The width of the text
\textheight	The height og the text
\unitlength	Units of length in the *picture* environment