LaTeXing Documentation

Release 0.9.10

LaTeXing.com

October 04, 2013
# CONTENTS

1 Introduction
   1.1 Installing LaTeXing ........................................... 3
   1.2 LaTeXing Configuration ...................................... 6

2 First Steps with LaTeXing
   2.1 Create a new TeX File ....................................... 11
   2.2 Save Document.tex ........................................... 11
   2.3 Bibliography File ............................................ 12
   2.4 Fill a Reference .............................................. 12
   2.5 Build ......................................................... 13
   2.6 The Working Directory ...................................... 14

3 User Guide
   3.1 Jumping to sections and labels .............................. 15
   3.2 Multi-Document Support .................................... 15
   3.3 Key Bindings ................................................ 15
   3.4 Commands ..................................................... 20
   3.5 Sublime Settings ............................................. 22
   3.6 Internal Options ............................................. 26

4 Tutorials
   4.1 Set up a Project ............................................. 29
   4.2 Cwl Support of LaTeXing ................................... 32
   4.3 Entering TeX Symbols ........................................ 34
   4.4 Phrases in LaTeXing .......................................... 37
   4.5 Use Online Lookup for Definition or Translation ....... 39
   4.6 Using knitr with LaTeXing ................................. 41
   4.7 Cache Support of LaTeXing ................................. 43
   4.8 Setup Bibsonomy with LaTeXing ............................ 43
   4.9 Setup Citeulike with LaTeXing ............................. 47
   4.10 Setup Mendeley with LaTeXing ............................. 49
   4.11 Setup Zotero with LaTeXing ................................ 53
LaTeXing is a complex plugin with a lot of functions, so please take time to read throughout the manuals and tutorials if you are not absolute familiar with the content of Sublime Text or LaTeX itself.
1.1 Installing LaTeXing

**Warning:** Since Package Control 2.0 is available the Sublime Text 2 and Sublime Text 3 version are both available under the same name. The features are different but we are looking forward and supporting the Sublime Text 3 version with new features. This will lead to great plugin completely written in Python3 with all advantages. If you have difficulties while updating it please just remove LaTeXing3 and reinstall the new version. If this still didn’t solved your problems, just drop an email at support@latexing.com.

1.1.1 Automatic Control Package Installation

The installation of LaTeXing via the popular Sublime Text plugin “package control” is very easy and fundamental. If you do not have that plugin installed, please see here for details on how to install it. If you do not wish to use package control, please follow the simple Manual Installation.

**Start Package Control: Install Package**

After adding the LaTeXing repository, please start the command palette (click Tools > Command Palette...) inside Sublime Text 3 and select Package Control: Install Package. The package control plugin will now fetch all necessary meta-details of the available plugins and list all extensions for the current system.
Install LaTeXing Package

After fetching the data, you need to select the LaTeXing package from that list at this point; the version number of the plugin may vary.

A moment later, the status message Package LaTeXing successfully installed should appear on the status bar and the initial message of LaTeXing show.
Note: The version for Sublime Text 2 is marked as deprecated, the version will still receive bug fix if required but the active development is focused on Sublime Text 3.

Warning: To work with Mendeley and Zotero you need the package `oauthlib` available from the package control.

### 1.1.2 Manual Installation

If you do not wish to use package control, please follow the following steps to install LaTeXing manually:

**Sublime Text 2 (Deprecated)**

1. Click the Preferences > Browse Packages... menu entry
2. Go up one directory and then open the `Installed Packages` folder
3. Download ST2 version and copy it into the `Installed Packages` directory
4. Restart Sublime Text 2

**Sublime Text 3**

1. Click the Preferences > Browse Packages... menu entry
2. Go up one directory and then open the Packages folder
3. Download ST3 version and copy it into the Installed Packages directory
4. Restart Sublime Text 3

1.1.3 Testing Version

If you would like to try out the latest version of LaTeXing you can use the testing repository, http://testing.latexing.com/packages.json, this version is under active development and is not recommended for the use on productive systems. If you found a bug in this version please drop a mail with a problem report at support@latexing.com.

The documentation for the testing version is earliest available via http://docs.latexing.com/en/development/, but it could also be that a new function is not supported prior release on the testing channel.

1.2 LaTeXing Configuration

1.2.1 Requirements

The first thing you need is Sublime Text, the plugin is compatible with both Sublime Text 2 and 3. The Sublime Text 3 version is being actively developed and will keep receiving new features; the last compatible version for Sublime Text 2 is any version below 0.6. So make sure that you are on a supported version before reporting any problems or bugs on LaTeXing.com.

Second, get the LaTeXing plugin, as explained (in two different ways) here see Installing LaTeXing. The easiest way is to complete the installation via the Package Control, but a manual installation is also simple and explained in the document.

OS X

On OS X you need to be running the MacTeX distribution (which is the most popular distribution on a Mac system anyway). If you encounter a problem while e.g. compiling the LaTeX file please update the distribution to the latest version before reporting any bugs. Normally the plugin should work on both 32 and 64 bits without limitations and in the exact same way.

Skim

The Skim PDF reader is recommended. Just download and install these two the usual way. Please setup the path in LaTeXing.sublime-settings, see Check your System to check your system and be sure that Skim can be found by LaTeXing. To configure inverse search, open the preferences dialog of the Skim app, select the sync tab, select Sublime Text 2 under the present; deselect the “Check for file changes” option. If you use an old version without an existing profile for Sublime Text 2 or using Sublime Text 3 with a different path do following:
1. Deselect the “Check for file changes” option
2. Preset: Custom
3. Command: /Applications/Sublime Text.app/Contents/SharedSupport/bin/subl
4. Arguments: "%file":%line

In case you have created a symbolic link to Sublime Text somewhere in your path, you can also use that in the command field. The above instructions will work in any case though, and do not require that you create a symbolic link. After setting up the path in preference, Skim should look similar to this one:

![Skim configuration screenshot]

**Preview**

To use Preview you just have to setup the path in `LaTeXing.sublime-settings`. Please see *Check your System* to check your system and be sure that Preview can be found by LaTeXing.

**Note:** Preview will ignore the following settings: forward_sync, reverse_sync

**Windows**

On Windows you can either use *MikTeX* or *TeX*. Both distributions are supported, but support for *TeX* is **better**. If you encounter a problem while e.g. compiling the LaTeX file please update the distribution to the latest version before reporting any bugs. Normally the plugin should work on both 32 and 64 bits without limitations and in the exact same way.

**Note:** Tip for Windows users when using MikTeX: Install MikTeX first, then install Perl manually. Run the basic commands like latexmk and texcount from command prompt to force MikTeX to install these Perl scripts automatically!

**SumatraPDF**

Be sure that you are running a current version of the *Sumatra PDF reader*. Just download and install these the usual way. Please setup the path in `LaTeXing.sublime-settings`, see *Check your System* to check your system and be sure that SumatraPDF can be found by LaTeXing. To configure the inverse search, open Sumatra PDF and do the following:

1. Click Settings > Advanced Options
2. Find the option *InverseSearchCmdLine* and change the value to: "C:\Program Files\Sublime Text 3\sublime_text.exe" "%f:%l". If you do not have the option just append a new line at the end of the file.

The path of the executable file may vary if you changed it during installation or for different Sublime Text versions, but for a standard installation of Sublime Text 3 this should work. Just check the path within explorer if you are not sure about the location.

---

1.2. **LaTeXing Configuration**
As an alternative to the graphical way, you can use the command-line to save the inverse search settings. In this case, open the command-line console and execute the following command:

```
"C:\Program Files\SumatraPDF\SumatraPDF.exe" -inverse-search ""C:\Program Files\Sublime Text 3\sublime_text.exe" "%f:%l"
```

The absolute path to SumatraPDF can vary, please adjust the path or use the GUI method. After setting up the path in the preference, SumatraPDF should look similar to this one:

![SumatraPDF settings](image)

**Foxit Reader**

To use Foxit Reader you just have to setup the path in `LaTeXing.sublime-settings`. Please see [Check your System](#) to check your system and be sure that Foxit Reader can be found by LaTeXing.

**Note:** Foxit Reader will ignore the following settings: forward_sync, reverse_sync

**PDF XChange Viewer**

To use PDF XChange Viewer you just have to setup the path in `LaTeXing.sublime-settings`. Please see [Check your System](#) to check your system and be sure that PDF XChange Viewer can be found by LaTeXing.

**Note:** PDF XChange Viewer will ignore the following settings: forward_sync, reverse_sync

**Adobe Reader**

To use Adobe Reader you just have to setup the path in `LaTeXing.sublime-settings`. Please see [Check your System](#) to check your system and be sure that Adobe Reader can be found by LaTeXing.

**Note:** Adobe Reader will ignore the following settings: forward_sync, reverse_sync

**Linux**

On Linux your can use TeXLive which is available for the most Linux distributions. LaTeXing is supporting Evince and Okular as pdf viewer. If you encounter a problem while e.g. compiling the LaTeX file please update the distribution to the latest version before reporting any bugs. Normally the plugin should work on both 32 and 64 bits without limitations and in the exact same way.

**Evince**

To use Evince you just have to setup the path in `LaTeXing.sublime-settings`. Please see [Check your System](#) to check your system and be sure that Evince can be found by LaTeXing.

**Okular**

To use Okular with LaTeXing you just have to setup the path in `LaTeXing.sublime-settings`. Please see [Check your System](#) to check your system and be sure that Okular can be found by LaTeXing. To configure the inverse search, open Okular and do the following:

1. Click Settings > Configure Okular
2. Select the Editor tab
3. Editor: Custom Text Editor
4. Command: /opt/sublime_text/sublime_text %f:%l

1.2.2 Check your sublime-build

Before you can start your work, you need to check the “LaTeX.sublime-build” file within the plugin directory of LaTeXing to make sure that the configuration reflects your preferred TeX distribution. Open the file by selecting Preferences > Package Settings > LaTeXing > Settings - Extended and check the corresponding settings of the block for the keyword “path”. By default MacTeX is configured for OS X and in the figure below MikTeX is used for Windows. Make sure that you do not change anything else apart from the path, if you are not sure what are you doing.

If you think you configured everything correctly, please run the supplied LaTeXing system check tool (explained below) to make sure that LaTeXing will work without issues.
1.2.3 Check your System

LaTeXing requires different command-line tools to support all functions. The most important command-line tool (and mandatory for compilation) is “latexmk”. The “latexmk” tool is shipped with all big LaTeX distributions, such as TeXlive, MacTeX, and MikTeX; for others it may be required to install it manually.

LaTeXing offers a tool to check if all required tools (latexmk, perl, texcount, kpswhich, mthelp for MikTeX, texdoc for TeXlive or MacTeX) are available in your system; to run it, click on Tools > Packages > LaTeXing > Check System. Some of these commands, like “texdoc”, are not mandatory but are still helpful during your daily work. Attached to the test, LaTeXing opens a window with a list of all the required tools and the total status of your system.

OS X

![OS X window showing LaTeXing check results]

Linux

![Linux window showing LaTeXing check results]
2.1 Create a new TeX File

LaTeXing provides several templates for creating a new LaTeX file; the templates are very elementary and nothing fancy. You can easily add your own snippets, please check the ST website for more details. For this tutorial we choose the article template.

2.2 Save Document.tex

The first thing we have to do is save the file as Document.tex; the extension must be typically, .tex. After saving, syntax highlighting will also become available, if enabled in the settings. Now we add some content to the file, which also includes a bibliography file.
2.3 Bibliography File

Everything here is LaTeX standard and not really related to LaTeXing, so if you are not comfortable with this step please check any well-known LaTeX information website, such as wikibooks.org/wiki/LaTeX.

2.4 Fill a Reference

After saving the bibliography file as References.bib at the same location as the Document.tex LaTeX file, you can use the LaTeXing fill command to choose your reference and select an item from the list.
2.5 Build

After we have finished adding content and hitting `cmd + b` on OS X (or the equivalent key combination on other platforms), the LaTeX document will compile. A log panel, located at the bottom of the screen, will appear and show information such as compile time, errors, warnings, or badboxes, if present. In this document everything is fine, so less than a second after, the PDF document will be available.
2.6 The Working Directory

The file structure is fairly simple, but to avoid any misunderstanding (shown below), the output folder is set to `Output` (which can be customized), relative to the current working directory.
3.1 Jumping to sections and labels

**Keybinding:** cmd + r

LaTeXing integrates the sophisticated Sublime Text “Goto Anything” facility. After hitting the key combination, a list of all heading commands (part, chapter, section, subsection, subsubsection, paragraph, subparagraph), label, include, input, subfile, bibsonomy, and internal options will be displayed. You can filter the items in that list by typing in a few initial letters. Selecting any entry in the list will take you to the corresponding place in the TeX file.

3.2 Multi-Document Support

As LaTeXing makes full use of ST’s project functionality, it also features multi-document support. Using it is quick and simple; you just have to connect every subfile with the root document by adding one line. For example, for a subfile named `Introduction.tex`, located in a directory `Chapter1` relative to the root file `Document.tex`, you need to insert the LaTeXing option `{\texttt{\% root: ../Document.tex \texttt{-**}}} at the very first line of your subfile. ST’s project functionality makes it possible for LaTeXing to recursively search through all further included files for corresponding items like references, abbreviations, etc. See *Set up a Project*.

3.3 Key Bindings

LaTeXing offers a wide range of key bindings to make your work more efficient and comfortable; these commands are listed and explained following:

3.3.1 Fill Command

**Keybinding:** cmd + l, cmd + l

The basic idea of this command is to help you out with inserting values in commands. The support of the commands is wide and the action depends on the selected command. For example it is possible to fill the argument of an `\includegraphics` command with the path to a figure just by selecting the item out of a list or the path of a subfile for the `\include` command. The supported commands are:

\texttt{\textbackslash input}

Provides a list of the installed packages, it is possible to search and select an individual package. The list just contains the local available packages of your LaTeX distribution and will be refreshed every 48 hours, so if you would like to deleted the cached file, please rebuild the cache

\texttt{\textbackslash documentclass}
Provides a list of the installed document classes, it is possible to search and select an individual class. The list just contains the local available classes of your LaTeX distribution and will be refreshed every 48 hours, so if you would like to delete the cached file, please rebuild the cache.

\include, \input, \subfile

Provides a list of TeX files related to the current TeX file, the filter for the available files can be adjusted in the settings, the default configuration includes *.tex and *.ltx files. The command subfile is just available with the popular subfiles package.

\includegraphics

Provides a list of image files related to the current TeX file, the filter for the available files can be adjusted in the settings, the default configuration includes *.jpeg, *.jpg, *.png, *.eps, and *.pdf files. To support the currfile packages LaTeXing offers an option that the path of the image file can be either relative to the root file within a project or relative to the current working directory.

\caption

This command is only available within a table and figure environment, it normalises the caption and adds a label to the next line. For example executing it within \caption{Text Caption Figure 1} inside a figure environment will normalise the caption and insert the following to the next line: \label{fig:text_caption_figure_1}. The used fig prefix is for a figure, for a table tbl will be used, this makes it comfortable to create quick readable labels for references.

\part(*), \chapter(*), \section(*), \subsection(*), \subsubsection(*), \paragraph(*), \subparagraph(*)

Similar to captions it is possible to create normalised labels of the selected part, chapter, section, or paragraph for your citations. The used prefix varies for the selected command, for example a part uses prt, a chapter cha, or a section sec.

\bibliography

Provides a list of bibliography files related to the current TeX file, the filter for the available files can be adjusted in the settings, the default configuration includes *.bib files.

\cite(*), \nocite(*)

Provides a list of the available citations, LaTeXing searched through all the included bibliography files and lists the related items. Thanks to the project support, multi-documents are fully supported, see the related section in the documentation for more information. Additional to the local citations it is very easy to import citations from the online reference service Bibsonomy.org, please see the related section within the documentation for further information on how to configure and use it properly. LaTeXing makes it possible to use this function also within mutations of the actual \cite command like \citeyear or \citet.

\ref, \cref

Provides a list of the available references, LaTeXing also supports multi-documents for cross references and list labels from included files as well as the current file. The cleveref package is fully supported and therefore also possible to use this function for commands like \ref, \namecref, or \crefrange.

\ac

Provides a list of the available abbreviations, this is just available with the acronym package. All the acronym commands are supported please check the package documentation for more information and a complete list of commands regarding this section.

\label

This function is not really a fill command but it could be counted as this since there are no possible values for a label expect the name. By pressing the key combination LaTeXing offers an input box and you can rename the current label thought your whole project.
3.3.2 Fill Anywhere Command

Keybinding: \(\text{shift + cmd + l, shift + cmd + l}\)

The fill anywhere command is providing similar functions like the Fill Command with the addition that a drop down menu will let you choose what you would like to do at the current selection. For captions and headings is this function useless and so not available but for other commands is this might be useful. For example it could be required to fill a image path somewhere else then just the normal includegraphics command. The fill anywhere command is available for \texttt{\include}, \texttt{\input}, \texttt{\subfile}. \texttt{\includegraphics}, \texttt{\ref}, \texttt{\cite}, and \texttt{\ac} commands and there mutations like usual.

3.3.3 Open Command

Keybinding: \(\text{cmd + l, cmd + k}\)

This command can open or follow a couple of commands, for example it is possible to open a subfile, open a citation for fast editing, or follow a ref cross reference to the corresponding label. The supported commands are:

\texttt{\documentclass}

Provides a list of the local documentations for the selected class, especially during the learning process of using a new class is this feature very handy and can save time to find the individual class documentation. The command line tool “\texttt{texdoc}” or “\texttt{mthelp}” is used and therefore required, the provided list also offers a search on ctan.org and google.com to find an unavailable class documentation online.

\texttt{\usepackage}

Provides a list of the local documentations for the selected package, especially during the learning process of using a new package is this feature very handy and can save time to find the individual package documentation. The command line tool “\texttt{texdoc}” or “\texttt{mthelp}” is used and therefore required, the provided list also offers a search on ctan.org and google.com to find an unavailable package documentation online.

\texttt{\include, \input, \subfile}

Just opens the used included LaTeX or bibliography file, very useful for big projects and makes every action outside of Sublime Text unnecessary.

\texttt{\cite(*)}, \texttt{\nocite(*)}

Follows the selected citation by locating the item and open the bibliography file if available. The supported commands are the same like for the fill command of \texttt{\cite} commands, so check the correlated section for more informations.

\texttt{\ref, \cref}

Follows the selected cross reference by location the item and open the LaTeX file if available. The supported commands are the same like for the fill command of \texttt{\ref} commands, so check the correlated section for more informations.

\texttt{\ac}

Follows the selected abbreviation by location the abbreviation and open the LaTeX file if available. The supported commands are the same like for the fill command of \texttt{\ac} commands, so check the correlated section for more informations.

\texttt{\label}

Follows the use of the current label throughout the whole project. You can select a item from the list of just check the used locations and return to the origin by pressing escape.

3.3.4 Open Anywhere Command

Keybinding: \(\text{shift + cmd + l, shift + cmd + k}\)
The open anywhere command is providing similar functions like the Open Command with the addition that you can try to open a file from the current selection. This is very useful if you use user defined commands and you would like to open the file which is defined as argument.

### 3.3.5 Open PDF

**Keybinding:** `cmd + l`, `cmd + o`

LaTeXing makes it very easy to handle your work without leaving the text editor, with this command it is possible to open the generated PDF file if available. The option `keep_focus` in the settings determines if the PDF viewer will gain focus or Sublime Text will stay in the foreground.

### 3.3.6 Jump to PDF

**Keybinding:** `cmd + l`, `cmd + j`

The function of this command is similar to the “Open PDF” command with the small difference that the PDF viewer will always be focused after open the file unrelated to the `keep_focus` setting and jump to the correlated position in the PDF file.

### 3.3.7 Import Citation(s)

**Keybinding:** `cmd + l`, `cmd + r`

LaTeXing offers a wide support for the online reference system Bibsonomy.org. It is possible to import single post, multiple post by tags, all available post, or even import post by searching for missing citations in your TeX file. The import command is available for TEX files as well as for bibliography files.

Apart from using Bibsonomy.org you can also configure a global bibliography files which will be used for further as reference. This is very handy if you use a global library exported by e.g. Mendeley.

**Note:** This support will be extended in the future by supporting the api of Mendeley or/and Zotero for example.

### 3.3.8 Toggle Preferences

**Keybinding:** `cmd + l`, `cmd + t`

Provides a list of the simple preference options of LaTeXing with the possibility of toggle them. This command can access boolean options, to these options which can be either true or false. With this command it is possible to quick disable or enable the use of the forward search for the generated file for example.

### 3.3.9 Online Lookup

**Keybinding:** `cmd + l`, `cmd + d`

The online lookup function is very useful for the daily work and fully customizable from the settings and so adjustable for your own preferences. For example it is possible to check for a translation, a definition, or a thesaurus of a selected word or a custom phrase. If the cursor is located in the middle of a word, LaTeXing will find the boundary and offers a list of search options for the selected word or ask for a custom phrase. If the cursor is already located at a word boundary or within spaces LaTeXing will ask for for a phrase straight away.
3.3.10 Insert TeX Symbol

Keybinding: § (OS X) or ‘ (Windows, Linux)

LaTeXing helps you to fill TeX symbols, these are for example simple Greek letters or more complex arrows or symbols. After hitting the trigger, a menu will appear and you can choose the symbol out of that list. The behaviour is a wee bit more complex and need some extra explanation, please check this tutorial Entering TeX Symbols.

3.3.11 Word to Phrases Dictionary

Keybinding: cmd + l, cmd + u

LaTeXing helps you to fill often used phrases in your report. With this command you can add one ore more selections to you defined phrases dictionary, please check Phrases in LaTeXing.

3.3.12 Build Options

Keybinding: cmd + l, cmd + b

Shows an overlay with the prefix Build: which will display all available build commands for the selected mode.

3.3.13 Clean Build

Keybinding: shift + cmd + b

Cleans the temporary files for the current project.

3.3.14 Show Commands

Keybinding: cmd + l, cmd + p

Shows an overlay with the prefix LaTeXing: which will display all available LaTeXing commands for the selected mode.

3.3.15 Show Snippets

Keybinding: cmd + l, cmd + s

Shows an overlay with the prefix Snippet: which will display all available snippets for the selected mode.

3.3.16 Text Decoration

Keybinding: cmd + l, cmd + c, b

Wrap the selected text in the command \textbf and so bold the text.

Keybinding: cmd + l, cmd + c, e

Wrap the selected text in the command \emph and so emphasise the text.

Keybinding: cmd + l, cmd + c, u

Wrap the selected text in the command \underline and so underlines the text.
3.3.17 LaTeX Commands

**Keybinding:** `shift + enter`

This will insert a double backslash and a new line at the current location, useful within a table environment.

**Keybinding:** `shift + enter`

This command is available within a list environment and insert a new line and prepares a new list item (also available within the beamer class).

3.3.18 BibTeX Commands

**Keybinding** @

Shows an overlay with the prefix Snippet: Bibtex which will display all available snippets inside a BibTeX.

**Keybinding:** `cmd + l, cmd + r`

See `Import Citation(s)`.

3.3.19 Special Characters

**Keybinding** `tab`

Pressing tab right behind the following symbols with escape the symbol to work with LaTeX like following:

`%, #, &, {, }, _` will change to `\%, \#, \&, \{, \}, \_`

3.3.20 Other Useful Stuff

**Keybinding:** ‘

This will auto pair single quotes.

**Keybinding:** ”

This will auto pair double quotes.

**Keybinding:** `§`

This will auto pair in-line equations.

**Keybinding:** `{`

This will auto pair parentheses.

**Keybinding:** `}`

This will auto pair curly brackets.

**Keybinding:** `[`

This will auto pair square brackets.

**Keybinding:** `<`

This will auto pair angle brackets.

3.4 Commands

These commands are available from the command palette. Some of these commands in this section are also available via a key binding, therefore you have to check the corresponding section in the key bindings for these commands.
3.4.1 Open PDF

Keybinding: cmd + l, cmd + o
See Open PDF.

3.4.2 Jump to PDF

Keybinding: cmd + l, cmd + j
See Jump to PDF.

3.4.3 Check System

See Check your System.

3.4.4 Import Citation(s)

Keybinding: cmd + l, cmd + r
See Import Citation(s).

3.4.5 Insert TeX Symbol

Keybinding: $ (OS X) or ‘ (Windows, Linux)
See Insert TeX Symbol.

3.4.6 Lookup TeX Symbol

Opens the browser at http://detexify.kirelabs.org/classify.html, on this website you can sketch a symbol and see the possible latex commands to generate it.

3.4.7 Phrases Dictionary

You can open any phrase dictionary with this command.

3.4.8 Word to Phrases Dictionary

Keybinding: cmd + l, cmd + u
See Word to Phrases Dictionary.

3.4.9 Synchronise Remote Data

If you changed any remote data like on Bibsonomy, Citeulike, Mendeley or Zotero you can synchronise the data by using this command. There is a different between rebuilding the cache and this command, this command will just refetching updated items rather than fetching all available items.
3.4.10 Update Bibliography File(s)

After you already synchronised your remote data, the content of a bibliography could be different, with this command the cached data and the bibliography are going to compared and if required the bibliography file filled with the required informations.

3.4.11 Rebuild Cache

One new feature of LaTeXing 0.8 is a cache function to improve the working speed with big projects. With this command you can rebuild the cache on the soft way which means that just the timestamps are going to be adjusted and you are not cleaning the cache rather than just re-fetching all informations.

3.4.12 Rebuild Cache (Hard)

This command is rebuilding the cache on a hard way, deleting all files and be ready to build the cache new while working from scratch. Normally this shouldn’t be required because LaTeXing is also checking the cache every startup for unavailable cached options to keep the cached data small. If you have problems, please rebuild the cache anyway.

3.4.13 Texcount

The word count of LaTeX files isn’t always an easy task but with LaTeXing this is very easy. This feature uses the command line tool texcount and gives a message after executing with the total words, the words in headers as well as the word in captions.

3.4.14 Toggle Preferences

**Keybinding:** cmd + l, cmd + t

See [Toggle Preferences](#).

3.5 Sublime Settings

LaTeXing allows you adjust many settings, all this options are located in the LaTeXing.sublime-settings file inside the package directory. Please copy the file into your user directory and change the settings there. This will keep your settings even if you update LaTeXing.

3.5.1 username, license

username and license code of your license from LaTeXing.com
3.5.2 debug

Debug informations inside the python console, just for test purposes

3.5.3 fallback_encoding

Fallback encoding, adjust this if you have problems with the encoding

3.5.4 path

Adjust the path where executable programs are located. For example for accessing curl on Linux for using the https connection with Zotero.

3.5.5 show_log_panel_on

Open the log panel after every build for the followings...

3.5.6 keep_focus

Sublime Text retains focus after a successful build, if false, the PDF viewer will gain focus

3.5.7 sublime_bin

In some situation LaTeXing needs to know where the Sublime Text binary (executable) file is located. If the keep focus setting to not work be sure Sublime Text is available on your path or adjust the path below.

3.5.8 forward_sync, reverse_sync

LaTeXing offers a full support for SyncTeX:

- forward_sync: Support to jump from the TeX file to the corresponding position inside the PDF
- reverse_sync: Support to jump back from the PDF file into the TeX file at the selected position

3.5.9 pdf_viewer_osx, pdf_viewer_windows, pdf_viewer_linux

Set your favorite PDF viewer for the different OS’s, the order defines the priority. For example on OS X is Skim the first choice but if this program is not available Preview will be used instead.

- Supported on OS X: Skim and Preview
- Supported on Windows: SumatraPDF, Foxit Reader, and Adobe Reader
- Supported on Linux: Evince and Okular

**Warning:** forward_sync and reverse_sync is just supported by Skim, SumatraPDF, Evince and Okular
3.5.10 phrases, -mode, -min_count, -min_length, -max_length, -bounding_words

With LaTeXing the auto completion is extended to use multi word phrases. This words are often used phrases and LaTeXing helps to complete this phrases in no time.

- phrases: enable the analyses of phrases
- phrases_mode: just phrases dictionary files (0), including current file (1) or the whole project (2)
- phrases_min_count: minimum count of phrases to be visible
- phrases_min_length: minimum length of the phrase
- phrases_max_length: maximum length of the phrase
- phrases_bounding_words: stop the phrase at these word

3.5.11 partial_build

If partial build is enabled mode, the document that is currently being edited is built. All the preambles of root document will be included as well, so you are not losing any settings while a partial build. This can be useful if you for example just would like to typeset one single chapter of a high project. Apart from the preambles, the bibliography will also be included.

3.5.12 open_pdf_on_load

LaTeXing can open the corresponding PDF file together with the current loaded TEX file

3.5.13 typeset_on_save

Typeset the document on saving it

3.5.14 type_scrolling

Set the line you work on automatically as vertical center of the screen

3.5.15 output_directory

LaTeXing can set an output_directory, this directory can be relative to the root document or can be an absolute file anywhere on your disk (be sure that the directory is writable). In case you set an directory outside of your TEX file dictionary LaTeXing will add the PDF name to the path

3.5.16 default_tex_extension, default_bib_extension

Default TEX and BIB file extension, this is used if no extension is defined

3.5.17 tex_pattern

TEX pattern, the search pattern of the provided list for the fill command of input, include, subfile commands
3.5.18 graphics_pattern, currfile_graphicspath

Graphics pattern, the search pattern of the provided list for the fill command of “includegraphics”. The option “currfile_graphicspath” enables the use of the package currfile and a changed graphicspath like this in your TeX file: \graphicspath{{\currfiledir}}

3.5.19 auto_trigger_fill

Auto trigger the fill command of TeX commands with the starting curly brake or the comma between two items. For example typing a { after cite will automatically open the list with the available citations

3.5.20 label_format, label_type_prefix

Customise the format of the label, the available keys are type, file, prefix, name; the value for type is defined with label_type_prefix. The list order is important and not changeable. Please note that the file key will be just used if you are not in your root document.

3.5.21 cite_panel_format

Customise the citation format of the quick panel for filling the cite command. The available variables are every property of the cite element, {stitle}, {sauthor} as well as the {origin} which is the location of the item

3.5.22 label_format, label_type

Customise the format of the label, the available keys are type, prefix, name; the value for type is defined by label_type below. Please note that a normalised version of the filename will be used if no prefix is defined and you are not in your root document.

3.5.23 static_cwl, dynamic_cwl

LaTeXing offers full support of the open source plugin LaTeX-cwl which can be found here. In addition to the provided cwl files by the plugin you have the ability to save your own files under User/cwl within the Sublime Text package directory. By default LaTeXing is loading the files in “static_cwl” and the user files depending on your used packages if “dynamic_cwl” is enabled.

3.5.24 build_arguments

Provide additional build arguments for latexmk, e.g. -shell-escape, -xelatex

3.5.25 symbols_in_category

Sort the symbols of the command “Insert TeX Symbols” in categories

3.5.26 cache_timeout

Defines the time interval (in minutes) when LaTeXing is refetching the information at a startup. You can also manually rebuild the cache or disable the cache by setting 0

3.5.27 remote_bibliography_in_category

Display the remote bibliography items in a own category instead of showing them together with the local items
### 3.5.28 bibname

Default bibliography file name for imported items from Bibsonomy.org, the global bibliography or any other available service.

### 3.5.29 bibsonomy, bibsonomy_username, bibsonomy_apikey

Enables the Bibsonomy.org support, you have to provide your username and apikey to use this feature, please check *Setup Bibsonomy with LaTeXing* for details how to obtain this information.

### 3.5.30 citeulike, citeulike_username

Enables the Citeulike.org support, you simple has to provide your particular username and then you can access the references.

### 3.5.31 mendeley, -_oauth_token, -_oauth_token_secret, -_cite_key_pattern

Enables the Mendeley.com support, just enable mendley, LaTeXing will get the other necessary data automatically, please check *Setup Mendeley with LaTeXing* for details. You can adjust the pattern to create a citation key if not available, the available keys here are {Author}, {author}, {year}, {Title}, {title}, you have to rebuild (hard) the cache after changing this settings.

### 3.5.32 zotero, _user_id, _user_key, _cite_key_pattern

Enables the Zotero.com support, just enable mendley, LaTeXing will get the other necessary data automatically, please check *Setup Zotero with LaTeXing* for details. You can adjust the pattern to create a citation key, the available keys here are {Author}, {author}, {year}, {Title}, {title}, you have to rebuild (hard) the cache after changing this settings.

### 3.5.33 global_bib_file, global_bib_file_path

Set a global bibliography file, you can import items from this file, similar to bibsonomy this can be useful if you exported all your references in one file (e.g. from Mendeley).

### 3.5.34 online_lookup

LaTeXing offers the support of running a command in the web browser with the selected word as argument. For example it is possible to look up for a definition or translation of a word very quick and easily.

### 3.5.35 knitr, -_command

knitr support, here you can enable the support, please be sure Rscript is available on your path, you can also set an absolute location for the Rscript executable.

### 3.6 Internal Options

LaTeXing offers some options which needs to be filled in the TeX files, these options need to be inserted at the top of a file, before any other command, with the following syntax: \verb|\% -- key: value --|. The option key can be one of the following values.
3.6.1 root

This option is for the multi-document support to connect subfiles with a root document. Please see the particular section for multi-document support for more information about the project support, see Set up a Project for an example to build up a project.

3.6.2 pdf

This option is just affecting the root document and defines the PDF file name. Normally the PDF name is a normalised version of the file name of the root TeX file but with this option you can freely customise the output PDF name.

3.6.3 phrases

You have the chance to define a dictionary with often used phrases, the content if this dictionary will be included in the normal autocompletion, see Phrases in LaTeXing for more details.

3.6.4 program

You can change the program used for the typeset, the possible values are xelatex and lualatex.

3.6.5 prefix

Set a unique prefix for the current file, e.g. for creating labels

**Note:** You can also place any of this options in a Sublime Text project file, see Set up a Project.
4.1 Set up a Project

Normally if you writing a document in LaTeX you always want to have multi files. This tutorial gives you an overview about the options within LaTeXing to hand that situation. The support is wide, LaTeXing offers completing citations, references, abbreviations (through the package acronym), and all other available completions throughout the whole project.

You have to different ways of building up a project in LaTeXing. Both methods do have there advantages and disadvantages depending on your use.

4.1.1 First Method (Options)

In every file you can fill some options, see Internal Options for the complete list. The important options for building up a multi files project is root. Please follow the next steps to understand this is working.

Create the Root Document

Within the root document you don’t have to change anything; just create a file as normal.
Create a Sub Document

After creating your main document, you need to create individual sub-files. In this screenshot the package subfiles is used, but the default commands `\include` and `\input` are surely also supported. If you would like to connect a sub-file with a main document, you have two different options. One option is to create a Sublime Text project and the other one is just to add the following line at the start of your tex file. The path can be absolute or a relative path to the current tex file.

\% -*- root: Project_Example.tex -*-

Note: Using the subfiles class is a special case. LaTeXing will also recognise the project due to the option of the document class, therefore you could just leave the first line out for the subfile class.

Build the Project

The screenshot below shows the success build, started from the subfile. It can be seen in the log panel that the compiled file is the main file instead of the currently selected sub-document.
4.1.2 Second Method (Sublime Text project file)

Rather than adding an extra line to every sub file you can use a Sublime Text project file, this is especially for huge projects with many files a nice help. Just create a project file in the same directory like your main files and save the root as option.

The path of the root file must be relative to the project file and will be applied to all views in this window, please check the Sublime Text website if you are not confident how to handle projects.

Note: As you can see it is also possible to define every internal option within a project file. These values are overwriting the in file options.
4.2 Cwl Support of LaTeXing

LaTeX has many commands; perhaps too many to remember them all; Especially if you are just starting to use LaTeX, it is difficult to remember even the most important of them. With cwl files it is possible to autocomplete many LaTeX commands while writing.

By default LaTeXing loads the following cwl files for every file (this list can be adjusted in the LaTeXing settings):
tex.cwl, latex-209.cwl, latex-dev.cwl, latex-document.cwl, latex-l2tabu.cwl, latex-mathsymbols.cwl.

LaTeXing offers two different ways to include cwl files:

1. Using the LaTeX-cwl plugin
2. Save your own cwl files in the Sublime Text package directory under: Packages/User/cwl (files in this folder are of higher priority and will override files from the LaTeX-cwl plugin)

Perhaps this way of using a different plugin for the cwl files looks more complicated right now, but it will make sense in the long run because there are A LOT of packages and this way it is possible to update LaTeX-cwl more often and independently of the LaTeXing event cycle.

If you create your own cwl files for missing packages please consider to submit your work to http://github.com/Chris—/LaTeX-cwl, if you have difficulties with github.com just drop an email at support@latexing.com. On this way it could be possible to have one day a complete collection of cwl files which could be also used also from different project and not just LaTeXing.

4.2.1 Typical cwl file

A typical cwl file, here of the cleveref package, is shown below. This document gives an overview on how to enable cwl support or add custom customs.

4.2.2 Include Package and use Autocomplete

LaTeXing covers the dynamic load of cwl files by detecting the use packages. After the package was included on line 4 the package commands are available in the autocomplete drop down menu. On the right side of the list is the source file of the command shown, here cleveref.
4.2.3 Create Custom cwl Files

If you created some user commands which are difficult to remember, you can create a custom cwl file like this one here, which is useful to write LaTeXing documentations. Just save the created cwl file in the Sublime Text package directory under: Packages/User/cwl and include the file in the LaTeXing static_cwl sublime-settings.

4.2.4 Use Custom cwl Files

After including the custom cwl file in the static_cwl sublime-settings you can use the commands like every other normal command. These files have a higher priority and will override files from the LaTeX-cwl plugin, if you have files with the same name.
LaTeXing includes a range of completion snippets, which make entering math symbols very easy. There are two triggers, $ and \text{tab}$, which follow a clear logic: $ expands the character(s) around the cursor for example into a corresponding Greek symbol, while \text{tab} expands the character(s) left of the cursor into math symbols by using the default auto completion of Sublime Text. In this tutorial will just focus on the $ trigger key.

\textbf{Note:} On Windows and Linux the trigger is ‘ instead of $, so just use your right trigger key. If you which to use a different key, for example to adapt a different keyboard layout, you can change the key in the key bindings setting file.

Say you are in an equation environment and want to enter \alpha.

\begin{equation}
a\end{equation}

You can expand a into \alpha by either hitting \text{tab} or $:

\begin{equation}
\alpha
\end{equation}

If you now hit $ again, nothing happens because no 2nd expansion is available for this Greek symbol and $ already expanded to the Greek character alpha. However, if you hit \text{tab} behind a single a it would result in the same because it is the best match of the Sublime Text auto completion as well.

The \text{tab} trigger lets you choose from a large number of symbols provided from the *.sublime-completions files, while $ gives you direct access to the frequently used Greek math characters and symbols. Finally, $ lets you quickly enter Greek symbols surrounded by opening and closing $ while writing text outside of math environments. If you hit $ in this situation LaTeXing will expand a to the following:

$a[$] will change to $\alpha$
### 4.3.1 Greek characters

Below is a list of currently implemented completions for Greek symbols. When more than one symbol is listed, you can reach it by pressing § repeatedly.

<table>
<thead>
<tr>
<th>character</th>
<th>1st expansion</th>
<th>2nd exp.</th>
<th>3rd exp.</th>
<th>4th exp.</th>
<th>5th exp.</th>
<th>6th exp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>\alpha</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>\beta</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>\delta</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>\epsilon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>\phi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>\gamma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>\eta</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>\iota</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>\jmath</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>\kappa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>\lambda</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>\mu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>\nu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>\pi</td>
<td>\psi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>\rho</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>\sigma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>\tau</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u</td>
<td>\upsilon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>\varepsilon</td>
<td>\varphi</td>
<td>\varpi</td>
<td>\varrho</td>
<td>\varsigma</td>
<td>\vartheta</td>
</tr>
<tr>
<td>w</td>
<td>\omega</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>\xi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>\zeta</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Capital versions of Greek symbols are also available.

<table>
<thead>
<tr>
<th>character</th>
<th>1st expansion</th>
<th>2nd expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>\Delta</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>\Gamma</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>\Theta</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>\Lambda</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>\Omega</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>\Pi</td>
<td>\Psi</td>
</tr>
<tr>
<td>S</td>
<td>\Sigma</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>\Xi</td>
<td></td>
</tr>
</tbody>
</table>

### 4.3.2 Arrows

Arrows are very useful but difficult to remember and long to type, with the § trigger this is very easy and quick to complete.

<table>
<thead>
<tr>
<th>character</th>
<th>1st expansion</th>
<th>2nd expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-</td>
<td>\leftarrow</td>
<td>\longleftarrow</td>
</tr>
<tr>
<td>&lt;=</td>
<td>\Leftarrow</td>
<td>\Longleftarrow</td>
</tr>
<tr>
<td>-&gt;</td>
<td>\rightarrow</td>
<td>\rightarrow</td>
</tr>
<tr>
<td>=&gt;</td>
<td>\Rightarrow</td>
<td>\Rrightarrow</td>
</tr>
<tr>
<td>&lt;&lt;</td>
<td>\leftrightharpoons</td>
<td>\longleftrightharpoons</td>
</tr>
<tr>
<td>&lt;-&gt;</td>
<td>\leftrightharpoons</td>
<td>\longleftrightharpoons</td>
</tr>
</tbody>
</table>

You can expand -> into \rightarrow by either hitting §:

```latex
\begin{equation}
\rightarrow
\end{equation}
```
If you now hit $ again, nothing happens, if no 2nd expansion is available. However, this is also working outside of math environments and LaTeXing surround the command by opening and closing $.

\begin{equation}
\rightarrow
\end{equation}

### 4.3.3 Other quick symbol shortcuts

Some commonly used math symbols (such as dots) are accessible via $ trigger as well (as always, further expansions can be reached via $ trigger).

<table>
<thead>
<tr>
<th>character</th>
<th>1st expansion</th>
<th>2nd expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>\left(</td>
<td></td>
</tr>
<tr>
<td>)</td>
<td>\right)</td>
<td></td>
</tr>
<tr>
<td>[</td>
<td>\left[</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>\right]</td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>\left/</td>
<td>\right/</td>
</tr>
<tr>
<td>.</td>
<td>\cdot</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>\times</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>\leq</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td>\geq</td>
<td></td>
</tr>
</tbody>
</table>

Note: Outside of math environments < and > will expand to \textless and \textgreater

### 4.3.4 Simultaneously enter open and closing parentheses

Using tab or $ trigger you can enter open and closing parentheses more quickly. Say you want to enter parentheses that adapt to the size of their content, pressing one trigger when the cursor is within parentheses

\begin{equation}
()\end{equation}

expands to

\begin{equation}
\left(\right)\end{equation}

This works for square brackets and curly braces as well.

### 4.3.5 Menu of symbols and commands

There are too many math symbols and commands to memorize them all. Thus LaTeXing can provide you with a menu to choose from. When in an empty math environment pressing $ on its own (i.e., without a preceding character) opens a menu of available categories:
Entering a categories (press [enter]) allows you to choose from a range of symbols and commands. You can also disable the list in categories with disable the setting `symbols_in_category`.

### 4.4 Phrases in LaTeXing

If you are writing a big project you do often have phrases which you always need, the default auto completion of Sublime Text do not complete multiple words. For this purpose LaTeXing can supply this often used phrases, there are two modes, the automatic mode of detecting often used phrases and the opportunity to define a dictionary of phrases.

#### 4.4.1 Automatic Mode

With the automatic mode you don’t have to do anything expect adjusting the settings, the `phrase_mode` must be 1 (for parsing the current file) or 2 (for parsing the whole project). Both options are also including the manual mode as well, if specified.

**Warning:** Prior version 0.8 this function can slow down the auto completion significant if you are working on large projects.

#### 4.4.2 Manual Mode

The manual mode is simple, you just have to add the option `phrases` to your file. Now you can start filling the dictionary by selecting a phrase and adding these phrase to your dictionary. The key combination for this can be found here *Word to Phrases Dictionary* or this is available from the command palette as well.
A status message will confirm this import with a message, if the successful.

**Note:** If you defined multiple dictionaries you have to choose where you would like to save the current selection prior import.

If you are opening the auto completion now you can also see the just added phrase.
You can also check the content of the phrases dictionary from within LaTeXing they are saved directly in your user dictionary. Just select the command Phrases Dictionary and select the right dictionary to open.

As you can see, the phrases dictionaries are just ordinary JSON list structures. You can also add words straight rather than using the import command.

### 4.5 Use Online Lookup for Definition or Translation

For that you just have to select a word and press `cmd + l`, `cmd + d`, a quick panel will appear and let you choose between different actions. If you are writing a text in LaTeX with LaTeXing it is easy to find a translation, definition, or just a different word for a word in your text.
If you select an action in that menu it will open the url, shown in the second row in the item. If you wish to search for a different word you can choose “Enter Different Lookup Phrase” and enter an individual word. The same will happen if you run the command without selecting a word or being at the border of a word.

Now LaTeXing offers a quick panel for the new entered word.
You can fully customise the search by adjusting the “online_lookup” setting for your preferences. The default settings are, while %s represents the word in every url:

```json
[
  {
    "title": "Dict.cc",
    "url": "http://www.dict.cc/?s=%s"
  },
  {
    "title": "Dict.leo.org",
    "url": "http://dict.leo.org/ende?search=%s"
  },
  {
    "title": "Dictionary.com",
    "url": "http://dictionary.reference.com/browse/%s"
  },
  {
    "title": "Linguee.com EN-DE",
    "url": "http://www.linguee.com/english-german/search?query=%s"
  },
  {
    "title": "Linguee.de DE-EN",
    "url": "http://www.linguee.de/deutsch-englisch/search?query=%s"
  },
  {
    "title": "Pons.eu",
    "url": "http://en.pons.eu/dict/search/results/?q=%s&l=deen"
  },
  {
    "title": "Thesaurus.com",
    "url": "http://thesaurus.com/browse/%s"
  }
]
```

## 4.6 Using knitr with LaTeXing

knitr is a syntax and an R package that allow you to put R code right in your LaTeX files. This is great for writing scientific articles or reports that use data, because you never lose track of how exactly you arrived at your results.
LaTeXing makes it dead simple to use knitr in Sublime Text 3. In this post I would like to introduce you to my knitr setup.

Let’s consider an example before we talk about the setup in detail.

\documentclass{article}
\begin{document}

R can be used as a calculator.

\begin{verbatim}
<<calc>>=
5 + 5
@
\end{verbatim}

Knitr integrates R plots into your document.

\begin{verbatim}
<<plot_something, fig.width=3.5, fig.height=3.5>>=
plot(1:100, rnorm(100), ylab = "A random value")
@
\end{verbatim}

Knitr integrates R tables into your document.

\begin{verbatim}
<<some_table, results='asis'>>=
library(xtable)
xtable(data.frame(a = 1:3, b = 11:13, c = 21:23))
@
\end{verbatim}

\end{document}

knitr files use the suffix ‘.Rnw’ and contain a combination of regular LaTeX and chunks of R code. These chunks begin with \texttt{<<name_of_the_chunk, options>>=} and end in a line containing a single @. When you compile your document, knitr evaluates the code chunks and produces a .tex file which can be compiled using pdflatex. knitr will try to present the results of the R code as nicely as it can, but there are plenty of options to customize its behavior. For example we use the \texttt{fig.width} and \texttt{fig.height} options to control the size of the resulting plot in our second chunk. We use \texttt{results='asis'} for the table, because \texttt{xtable()} produces LaTeX code that we want to insert ‘as it is’. Here is the pdf produced by the above code. It looks quite nice already.

### 4.6.1 How to knit in Sublime Text 3

There are three packages for Sublime Text 3 that make knitting a walk in the park:

1. SublimeKnitr provides syntax highlighting and some editing capabilities for knitr files.
2. Our very own LaTeXing handles knitr files just like regular LaTeX files. Simply hit Ctrl+b (or Super+b on a Mac) to build.
3. Enhanced-R allows you to send code from your knitr document to R.

The easiest way to install these plugins is to get Package Control, bring up the command palette (Strg+Shift+p on Windows and Linux, Super+Shift+p on the Mac), select Package Control: Install Package, select the package you want, and hit enter. You can easily get things up and running in three minutes or less.

To enable knitr support in LaTeXing you only have to add a single option to your LaTeXing settings: (In the menu select Preferences/Package Settings/LaTeXing/Settings - User and insert the following line.):

"knitr": true

Now, copy the sample code above, paste it into a new file, and save it as \texttt{something.Rnw} in a folder of your liking. Hit Ctrl+b (or Super+b on a Mac) to create the pdf.

While I am working on my R code I prefer not to build the whole document all the time. Instead I have an instance of R running and use Ctrl+enter to send the current line or selection to R. Ctrl+Shift+Alt+r (Super+Shift+Alt+r...
on a Mac) sends the current chunk instead. This capability is provided by Enhanced-R. It should work out of the box if you use the R GUI on Windows or Mac or run R inside tmux on Linux. screen is also supported on Linux.

4.6.2 Summary and where to go next

knitr is a great way to keep your LaTeX and R code together. LaTeXing has dead simple support for knitr and together with SublimeKnitr and Enhanced-R it provides all the tools you need to use knitr in Sublime Text 3.

knitr’s website is a good place to find out more about knitr’s features such as its caching and graphics capabilities. It provides a number of demos that will get you started on the basics.

A big thanks to Severin for writing up this tutorial!!

4.7 Cache Support of LaTeXing

One new feature of LaTeXing 0.8 is a cache function to improve the working speed with big projects. Here is a short summary of the major informations:

### 4.7.1 Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc.cache</td>
<td>Result of texdoc command</td>
</tr>
<tr>
<td>pkg.cache</td>
<td>LaTeX packages and classes</td>
</tr>
<tr>
<td>tex.cache</td>
<td>TeX files</td>
</tr>
<tr>
<td>bib.cache</td>
<td>Bibliography files</td>
</tr>
<tr>
<td>bibsonomy.cache</td>
<td>Bibsonomy data</td>
</tr>
<tr>
<td>citeulike.cache</td>
<td>CiteuLike data</td>
</tr>
<tr>
<td>mendedely.cache</td>
<td>Mendeley data</td>
</tr>
<tr>
<td>zotero.cache</td>
<td>Zotero data</td>
</tr>
</tbody>
</table>

### 4.7.2 Workflow

For you as user there is no change while writing LaTeX files but under the hood there are some changes. When you start Sublime Text, LaTeXing is fetching the existing cache data, this may take a moment depending on your cache size. The cache will be updated after every build but no more than every 5 minutes to keep the performance of the whole system high with huge cache files.

### 4.7.3 Commands

You can rebuild the cache informations, there are two different ways: a normal rebuild and a hard rebuild. The hard rebuild will delete all cache files and you can start from scratch. On the other hand the normal rebuild will just adjust the timestamps and execute a normal rebuild after a cache time out.

On a rebuild, for example the doc.cache, LaTeXing will check the documentation for every cached package again and if not available delete the item. For the tex.cache LaTeXing will check the cached files and if a file is not available any more this cache item will be also cleaned.

### 4.8 Setup Bibsonomy with LaTeXing

LaTeXing offers a great support for the work together with Bibsonomy. This feature was introduced with LaTeXing version 0.8. The steps to setting up Bibsonomy are very simple and you just have to follow these steps systematically.
Activate the **bibsonomy** option inside the *LaTeXing.sublime-settings*. At the first moment LaTeXing tries to access some data from Bibsonomy and you haven’t set up the access the authorisation process will be initiated.

### 4.8.1 Obtain the Api Key

For a successful communication between LaTeXing and Bibsonomy a private api key is required. You have to login into your webaccount and generate this key.

At the bottom an input panel was created with a link, normally your standard browser should automatically open this link and you don’t have to copy the link from there. The Bibsonomy website will open and you need to follow the instructions there. If you are not reaching the shown website, please check Bibsonomy.org for more details.

Log in with your default Bibsonomy login details.
Now you have to generate and/or copy the api key for LaTeXing, scroll down and copy the api key and go back to LaTeXing.

4.8.2 Enter the Username and Api key

With the api key code from Bibsonomy.org you can finalize the authorisation process. Just type your username and paste the code in the input box and press enter. Separate the username and api key by a space.

If the authorisation was successful a status message will confirm this, shown on the next picture.
4.8.3 Using your Bibsonomy Database

Note: Just be aware that you have to force LaTeXing to synchronise the data every time when you change something and you wouldn’t like to wait for the cache refresh. The synchronisation command will be a particle update and just the changed citations are updated.
4.9 Setup Citeulike with LaTeXing

LaTeXing offers a great support for the work together with Citeulike. This feature was introduced with LaTeXing version 0.8. The steps to setting up Citeulike are very simple and you just have to follow these steps systematically.

Activate the `citeulike` option inside the `LaTeXing.sublime-settings`. At the first moment LaTeXing tries to access some data from Citeulike and you haven’t set up the access the authorisation process will be initiated.

4.9.1 Enter the Username

For a successful communication between LaTeXing and Citeulike you just have to provide an username. At the bottom an input panel will appear and you just have to fill your username in that panel.

If the authorisation was successful a status message will confirm this, shown on the next picture.
4.9.2 Using your Citeulike Database

Note: Just be aware that you have to force LaTeXing to synchronise the data every time when you changes something and you wouldn’t like to wait for the cache refresh. The synchronisation command will be a particle update and just the changed citations are updated.
4.10 Setup Mendeley with LaTeXing

LaTeXing offers a great support for the work together with Mendeley. This feature was introduced with LaTeXing version 0.8. The steps to setting up Mendeley are very simple and you just have to follow these steps systematically.

**Warning:** To work with Mendeley and Zotero you need the package `oauthlib` available from the package control.

Activate the `mendeley` option inside the `LaTeXing.sublime-settings`. At the first moment LaTeXing tries to access some data from Mendeley and you haven’t set up the access the authorisation process will be initiated.

4.10.1 Obtain the Verification Code

For a successful communication between LaTeXing and Mendeley a private key is required. Normally you don’t have to configure much instead LaTeXing is handle all the handshakes for you.

At the bottom an input panel was created with a link, normally your standard browser should automatically open this link and you don’t have to copy the link from there. The Mendeley website will open and you need to follow the instructions there.
Log in with your default Mendeley login details.

Now you have to accept the access for LaTeXing.
After accepting the permissions the verification will be displayed and you just need to copy this code and go back to LaTeXing.

### 4.10.2 Enter the Verification Code

With the verification code from Mendeley.com you can finalize the authorisation process. Just paste the code in the input box and press enter.

If the authorisation was successful a status message will confirm this, shown on the next picture.
Now you are ready to use your Mendeley database without leaving LaTeXing; like it is saved on your local machine for citations or import multiple items to your local bibliography file.

4.10.3 Using your Mendeley Database

Here the original database in Mendeley Desktop.
Note: Just be aware that you have to force LaTeXing to synchronise the data every time when you change something and you wouldn’t like to wait for the cache refresh. The synchronisation command will be a particle update and just the changed citations are updated.

4.11 Setup Zotero with LaTeXing

LaTeXing offers a great support for the work together with Zotero. This feature was introduced with LaTeXing version 0.8. The steps to setting up Zotero are very simple and you just have to follow these steps systematically.

Warning: To work with Mendeley and Zotero you need the package oauthlib available from the package control.

Activate the zotero option inside the LaTeXing.sublime-settings. At the first moment LaTeXing tries to access some data from Zotero and you haven’t set up the access the authorisation process will be initiated.

4.11.1 Obtain the Verification Code

For a successful communication between LaTeXing and Zotero a private key is required. It is possible to generate and fill this code by hand but here we are going to use the automatic way with LaTeXing.
At the bottom an input panel was created with a link, normally your standard browser should automatically open this link and you don’t have to copy the link from there. The Zotero website will open and you need to follow the instructions there.

Log in with your default Zotero login details.
Now you have to accept the key for LaTeXing. You can check the permissions in here but it is also a read request required.

After accepting the permissions the verification will be displayed and you just need to copy this code and go back to LaTeXing.

### 4.11.2 Enter the Verification Code

With the verification code from Zotero.org you can finalize the authorisation process. Just paste the code in the input box and press enter.
If the authorisation was successful a status message will confirm this, shown on the next picture.

Now you are ready to use your Zotero database without leaving LaTeXing; like it is saved on your local machine for citations or import multiple items to your local bibliography file.
4.11.3 Using your Zotero Database

Here the original database in Zotero.

Note: Just be aware that you have to force LaTeXing to synchronise the data every time when you changes something and you wouldn’t like to wait for the cache refresh. The synchronisation command will be a particle update and just the changed citations are updated.

Note: The documentation is also available as PDF, HTMLZip, or Epub.