

Master's Thesis / Doctoral Dissertation

JAIST  $\text{\LaTeX}$  模板使用教程  
DOCUMENT FOR JAIST THESIS  
 $\text{\LaTeX}$  TEMPLATE  
V1.0

Author: Wensheng Lin  
Email: linwest@jaist.ac.jp

*Graduate School of Advanced Science and Technology*  
*Japan Advanced Institute of Science and Technology*  
(Information Science)

March 2019



# Abstract

This document introduces the usage of JAIST L<sup>A</sup>T<sub>E</sub>X Thesis Template [1]. This project is revised from official thesis template [2]. Since it is relatively simple, this project divides the template into many files, and adds some functions, such as appendices, glossaries, algorithms, and codes.

本文档介绍了 JAIST L<sup>A</sup>T<sub>E</sub>X 模板 [1] 的使用方法。该模板由学校的官方模板 [2] 改进而来。这个模板的改动也没有什么别的，大概三件事：

- 一个，将模板分割为若干独立文件，便于管理和撰写；
- 第二个，修改了原模板的一些 BUG；
- 第三个，就是我们知道的一些常用功能，也被加进了这个模板中。

如果说还有一点什么成绩，就是再附加了一份教程。但这些都是次要的，这个模板主要的改动就是三件事情，很惭愧，就做了一点微小的工作，谢谢大家。

**Keywords:** JAIST, L<sup>A</sup>T<sub>E</sub>X, Thesis, Template, +1s.



## Acknowledgment

Thanks Mr. Qisen Wang for introducing Overleave and other convenient tools. We are also grateful to those who support this project, including the developers of the macro packages used in this project.

感谢王麒森介绍了 Overleave 和其它实用工具。同样感谢所有支持、使用和提供帮助的人们，包括但不限于本模板中所引用的宏包的开发者们。



# List of Abbreviations

BER	Bit error rate
CEO	Chief executive officer
i.i.d.	Independent and identically distributed
JPEG	Joint photographic experts group
MPEG	Moving picture experts group
$pmf$	Probability mass function
SNR	Signal-to-noise ratio
WSN	Wireless sensor network





# List of Symbols

$C(\cdot)$	the Shannon capacity using Gaussian codebook
$t$	time index
$\delta(\epsilon)$	a function of $\epsilon$ that tends to zero as $\epsilon \rightarrow 0$



# List of Figures

1.1	Configure Texmaker. . . . .	3
1.2	Revise LaTeX command. . . . .	4
1.3	Revise quick build. . . . .	5
1.4	Overleaf menu. . . . .	6
1.5	Overleaf configuration for Japanese. . . . .	6
1.6	Overleaf configuration for English. . . . .	8
2.1	JPG figure title. . . . .	16
2.2	Subfigure Examples. . . . .	18
2.3	Edit User Commands. . . . .	28
2.4	Make glossaries. . . . .	28



# List of Tables

2.1	Caption of the table . . . . .	20
A.1	Table in Appendix . . . . .	37



# Contents

<b>Abstract</b>	<b>I</b>
<b>Acknowledgment</b>	<b>III</b>
<b>List of Abbreviations</b>	<b>V</b>
<b>List of Symbols</b>	<b>VII</b>
<b>List of Figures</b>	<b>IX</b>
<b>List of Tables</b>	<b>XI</b>
<b>Contents</b>	<b>XIII</b>
<b>Chapter 1 Introduction of L<sup>A</sup>T<sub>E</sub>X/ L<sup>A</sup>T<sub>E</sub>X 入门</b>	<b>1</b>
1.1 Required Software/所需软件 . . . . .	1
1.1.1 Offline Writing/离线撰写 . . . . .	1
1.1.2 Online Writing/在线撰写 . . . . .	2
1.2 Writing in Chinese/中文编译环境 . . . . .	2
1.3 Writing in Japanese/日文编译环境 . . . . .	2
1.3.1 Configuration for Texmaker/配置 Texmaker . . . . .	3
1.3.2 Configuration for Overleaf/配置 Overleaf . . . . .	5
1.4 Command, Argument, Macro and Environment/命令、参数、 宏、环境 . . . . .	8
1.4.1 Command/命令 . . . . .	8
1.4.2 Argument/参数 . . . . .	9

1.4.3	Macro/宏	10
1.4.4	Environment/环境	10
1.4.5	Including Files/引用文件	11
<b>Chapter 2</b>	<b>Introduction of the Template/模板介绍</b>	<b>13</b>
2.1	Files and Folders/文件介绍	13
2.2	Insert Chapters and Sections/插入章节	14
2.3	Appendices/附录	15
2.4	Figures/图片	15
2.4.1	Inserting Figures/插入图片	15
2.4.2	Inserting Sub-Figures/插入子图	18
2.4.3	Referring to Figures/引用图片	19
2.4.4	List of Figures/插入图片列表	19
2.5	Tables/表格	19
2.5.1	Inserting Tables/插入表格	19
2.5.2	Referring to Tables/引用表格	22
2.5.3	List of Tables/插入表格列表	22
2.6	Equations/公式	23
2.7	Inline Mode/行内公式	23
2.8	Display Mode/独行公式	23
2.9	Aligning Equations/多行公式	23
2.9.1	Referring to Equations/引用公式	24
2.10	Abbreviations and Symbols/缩略语和符号	25
2.10.1	Abbreviations Management/管理缩略语	25
2.10.2	Symbols/符号	26
2.10.3	List of Glossaries/术语列表	27
2.11	Bibliography and Citations/引用参考文献	29
2.11.1	Compiling Order/编译顺序	30
2.12	Algorithms/算法	31
2.13	Codes/代码	32
2.13.1	Inserting Code/插入代码	32
2.13.2	Importing Codes from Files/从文件中导入代码	33



2.13.3 Referring to Codes/交叉引用代码 . . . . .	34
2.14 Definition, Lemma and Proposition/定义、引理和命题 . . . .	34
<b>Appendices</b>	<b>37</b>
<b>Appendix A Example</b>	<b>37</b>
<b>Appendix B Code Example</b>	<b>39</b>
<b>References</b>	<b>41</b>
<b>Publications</b>	<b>43</b>



# Chapter 1

## Introduction of L<sup>A</sup>T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X 入门

L<sup>A</sup>T<sub>E</sub>X divides text and format in a document, and therefore it is easy to separately control the text and format without influence on each other. It is equivalent to programming when you use L<sup>A</sup>T<sub>E</sub>X. Hence, you are the product manager and the developer of your thesis.

与所见即所得的 Word 不同, L<sup>A</sup>T<sub>E</sub>X 将样式与文字分开, 需要将文档编译后才能排版生成 PDF 文件。所以 Word 学起来简单用起来难, 而 L<sup>A</sup>T<sub>E</sub>X 学起来难用起来简单。L<sup>A</sup>T<sub>E</sub>X 相当于在写代码, 因此可以高效地控制样式, 易于交叉引用图表、章节、文献等。在 L<sup>A</sup>T<sub>E</sub>X 中, 你的文章就是一个工程项目, 你则是该项目的产品经理和开发者。

### 1.1 Required Software/所需软件

#### 1.1.1 Offline Writing/离线撰写

You need an editor to write your L<sup>A</sup>T<sub>E</sub>X codes and a compiler to generate the document from L<sup>A</sup>T<sub>E</sub>X codes. Some useful softwares are listed as follows

1. Integrated L<sup>A</sup>T<sub>E</sub>X system, e.g., T<sub>E</sub>X Live, which includes compilers and editor。
2. Separate L<sup>A</sup>T<sub>E</sub>X editor, e.g., Texmaker。
3. Reference manager, e.g., JabRef, Mendeley。

你需要一个文本编辑器来编写 L<sup>A</sup>T<sub>E</sub>X 代码, 还需要一个编译器将代码

编译成 PDF 文档。必备工具如下：

1. L<sup>A</sup>T<sub>E</sub>X 集成环境。如：T<sub>E</sub>X Live，它包含了编译器和文本编辑器。
2. 文本编辑器。如：Texmaker。
3. 参考文献管理工具。如：JabRef, Mendeley。

### 1.1.2 Online Writing/在线撰写

Overleaf is a popular online L<sup>A</sup>T<sub>E</sub>X editor. You can use L<sup>A</sup>T<sub>E</sub>X without download huge software, and store your projects in cloud.

Overleaf 网站可以在线撰写 L<sup>A</sup>T<sub>E</sub>X。你无需下载安装其它软件，并且可以将文档项目存在云端。

## 1.2 Writing in Chinese/中文编译环境

Just use the “ctex” macro package by:

```
1 \usepackage{ctex}
```

引用 ctex 宏包即可用 pdfLaTeX 编译，无需额外配置编译环境：

```
1 \usepackage{ctex}
```

## 1.3 Writing in Japanese/日文编译环境

You need additional configurations for writing in Japanese. T<sub>E</sub>XLive and Overleaf have already contained the compiler “platex” for Japanese documents. You can use the document class for Japanese, such as “jbook”, “jreport”, “jarticle”, and the corresponding revised version as “jsbook”, “jsreport”, “jsarticle”. For more details, please check the wiki and github of jsclasses:

- <https://texwiki.texjp.org/?jsclasses>
- <https://github.com/texjporg/jsclasses>

The jsclasses requires pLaTeX2e for compiling. The steps are as follows:

1. First compile .text file into .dvi file by “platex”, as:

```
platex filename.tex;
```

2. Then, generate .pdf file from .dvi file by “dvipdfm”.

TeXLive 自带日文环境, 可以选择的文档样式有 jbook, jreport, jarticle, 以及优化过后的 jsbook, jsreport, jsarticle。可以参考 jsclasses 的 wiki 和 github:

- <https://texwiki.texjp.org/?jsclasses>
- <https://github.com/texjporg/jsclasses>

以上样式需要使用 pLaTeX2e 进行编译, 编译顺序为:

1. 首先用 platex 把 tex 文件编译成 dvi 文件, 命令为:

```
platex 文件名.tex;
```

2. 然后用 dvipdfm 把 dvi 文件编译为 pdf 文件。

### 1.3.1 Configuration for Texmaker/配置 Texmaker

We introduce the configuration for Texmaker as an example.

下面以 Texmaker 为例介绍配置。

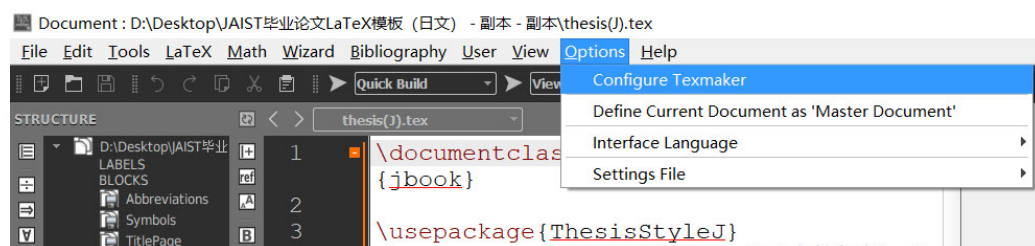
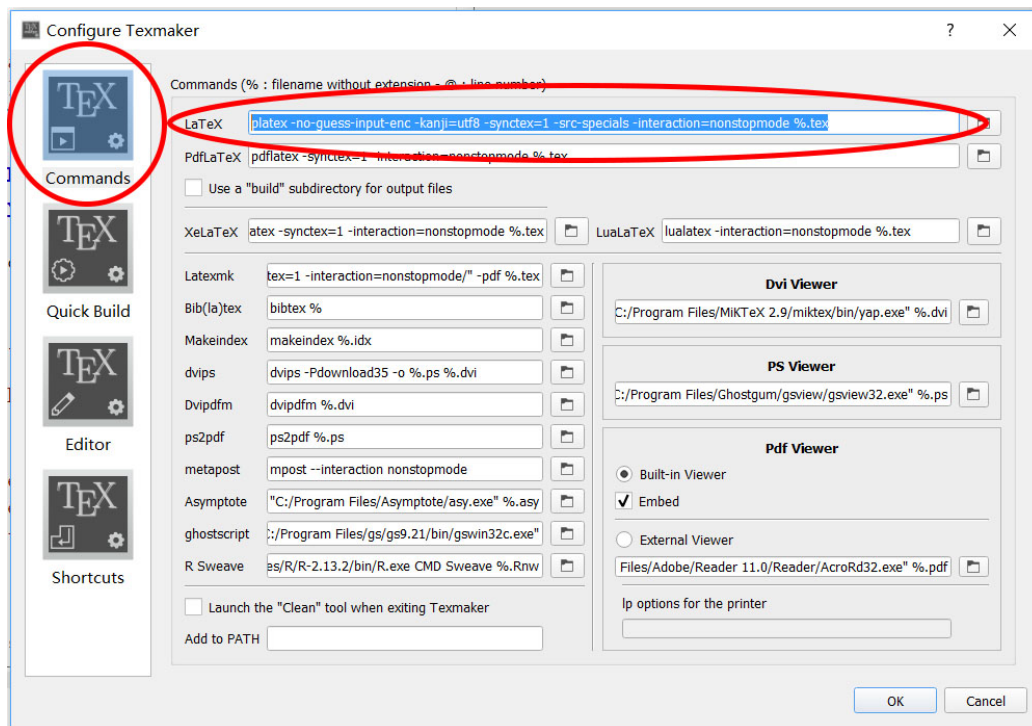


Figure 1.1. Configure Texmaker.

First, as shown in Figure 1.1, click “Options → Configure Texmaker”.

首先, 如 Figure 1.1所示, 点击“选项 → 配置 Texmaker”。



**Figure 1.2.** Revise LaTeX command.

Then, as shown in Figure 1.2, revise the LaTeX command as:

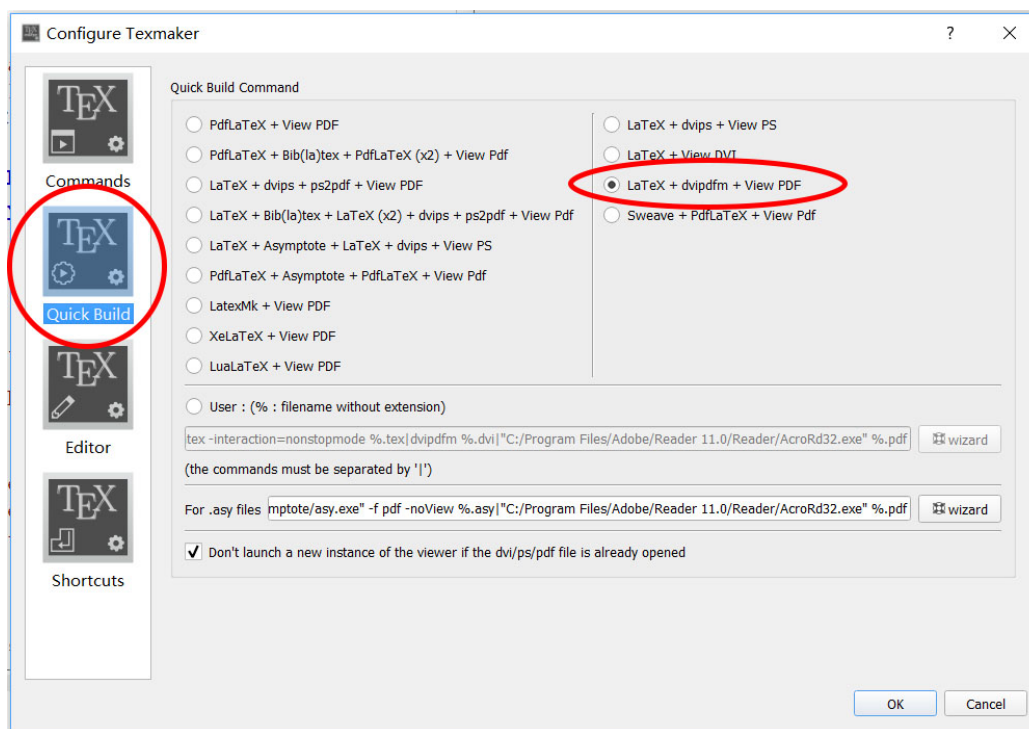
```
platex -no-guess-input-enc -kanji=utf8 -synctex=1 -src-specials
-interaction=nonstopmode %.tex
```

Now, we have already changed the  $\text{\LaTeX}$  compiler into platex. In order to build more conveniently, we can further configure the “Quick Build” command.

然后，如 Figure 1.2所示，将 LaTeX 命令改为以下代码：

```
platex -no-guess-input-enc -kanji=utf8 -synctex=1 -src-specials
-interaction=nonstopmode %.tex
```

此时已经将  $\text{\LaTeX}$  编译工具配置为 platex，为了更方便地生成 PDF，还可以再配置一下“快速构建”命令。



**Figure 1.3.** Revise quick build.

As shown in Figure 1.3, click the “Quick Build” icon on the left side. Then, chose “LaTeX + divpdfm + View PDF” as the quick build command. Now, we can use the shortcut key “F1” to quick build.

如 Figure 1.3所示，在左侧图标中选择“快速构建”，然后将右侧的快速构建命令选为“LaTeX + divpdfm + 查看 PDF”。就可以使用快捷键 F1 一键编译。

### 1.3.2 Configuration for Overleaf/配置 Overleaf

First, as shown in Figure 1.4, click “Menu”.

首先，如 Figure 1.4所示，点击“菜单”。

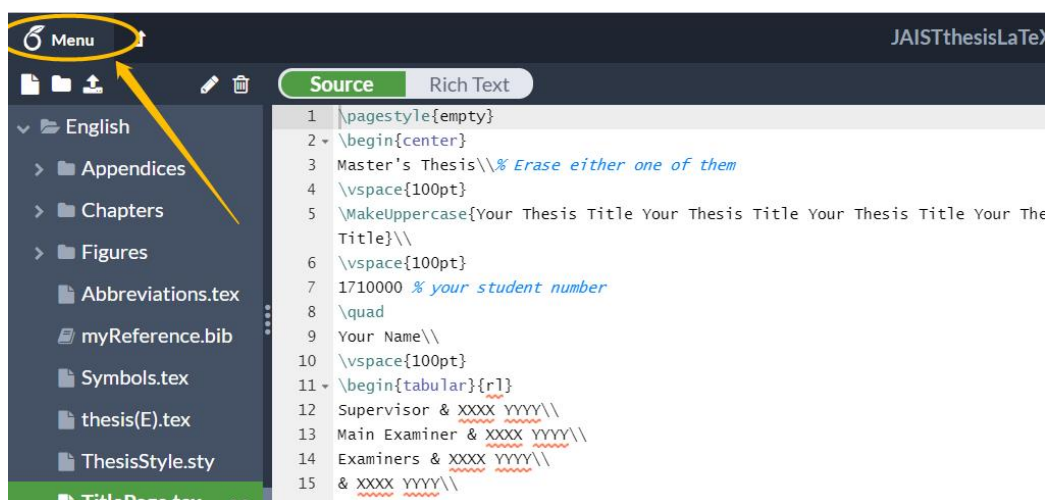


Figure 1.4. Overleaf menu.

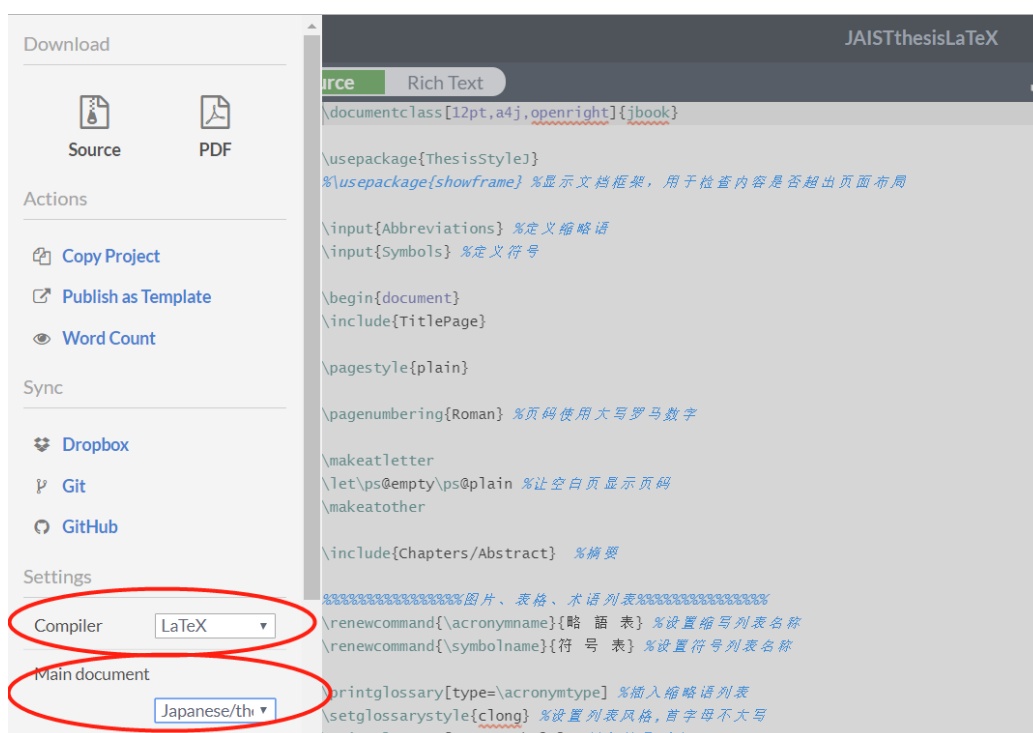


Figure 1.5. Overleaf configuration for Japanese.

Then, as shown in Figure 1.5, choose “LaTeX” as the compiler, and



set the Japanese document entry as the main document, i.e., “thesis(J).tex”. Now, you can successfully compile the Japanese documents. Because in the root path of the project, there is already a configuration file “latexmkrc”, containing the following codes for setting platex as the L<sup>A</sup>T<sub>E</sub>X compiler.

```
1 $latex = 'platex';  
2 $bibtex = 'pbibtex';  
3 $dvi2pdf = 'dvipdfmx %O -o %D %S';  
4 $makeindex = 'mendex %O -o %D %S';  
5 $pdf_mode = 3;
```

然后，如 Figure 1.5所示，把编译器设置为“LaTeX”，把主目录选择为日文模板的主文件，即“thesis(J).tex”。现在你已经可以成功编译日文文档了。因为在项目的根目录下，已经有一个“latexmkrc”的配置文件把 platex 设置为 L<sup>A</sup>T<sub>E</sub>X 的编译器，里面写好了以下代码：

```
1 $latex = 'platex';  
2 $bibtex = 'pbibtex';  
3 $dvi2pdf = 'dvipdfmx %O -o %D %S';  
4 $makeindex = 'mendex %O -o %D %S';  
5 $pdf_mode = 3;
```



Figure 1.6. Overleaf configuration for English.

If you need compile English document, you can according to Figure 1.6, set the compiler back to “pdfLaTeX”, and choose the English document entry as the main document, i.e., “thesis(E).tex”.

如果你需要编译英文文档，可以按 Figure 1.6所示，将“pdfLaTeX”设为编译器，同时将主目录选择为英文模板的主文件，即“thesis(E).tex”

## 1.4 Command, Argument, Macro and Environment/命令、参数、宏、环境

### 1.4.1 Command/命令

As mentioned above, writing in  $\text{\LaTeX}$  is equivalent to programming. You can only write plain text in  $\text{\LaTeX}$  source files; thus,  $\text{\LaTeX}$  provides commands to allow you control the text style.

A command starts with “\”, followed by the command name. For instance, the command “\LaTeX” outputs the “ $\text{\LaTeX}$ ” logo. Since “\” is for special use in  $\text{\LaTeX}$ , if you want to type it out, you should use “\textbackslash”. Note that “\textbackslash” is also a  $\text{\LaTeX}$  command, which let  $\text{\LaTeX}$  type out “\”. There are also some other special characters required commands to type out. You can just Google them if necessary.

$\text{\LaTeX}$  的文档由纯文本代码构成，因此除了文章的内容之外，还需要额外的命令来告诉编译器哪些内容需要什么样式。命令由反斜杠 “\” 开头，后面接上命令的名称。如 “\LaTeX” 这个命令用于输出 “ $\text{\LaTeX}$ ” 这个 logo。

由于 “\” 在  $\text{\LaTeX}$  中有特殊功能，所以如果你想在文章中打出 “\”，你需要使用转义字符 “\textbackslash”，注意到 “\textbackslash” 本质上也是一条  $\text{\LaTeX}$  命令。此外还有一些其它的转义字符，在需要使用时请自行上网搜索。

### 1.4.2 Argument/参数

Some commands require arguments, so that you can do more things. For example, the command “\documentclass” sets the class of a document; consequently, it requires the name of class as an argument.

Arguments includes mandatory arguments and optional arguments. Mandatory arguments are written within “{...}”, while optional arguments are within “[...]”. More than one arguments are divided by English comma “,”. For example, the following command

```
\documentclass[twocolumn, 12pt]{article}
```

sets “article” as the document class, 12pt as the fontsize, and text aligned in two columns.

有些命令需要指定参数，比如设定文档格式的命令：\documentclass，你需要提供文档样式模板的名称作为命令参数。

参数包括必选参数和可选参数，大括号{...} 里面的的是必选参数，中括号[...] 里面的的是可选参数。多个参数之间使用英文半角逗号隔开。例如以下命令：

```
\documentclass[twocolumn, 12pt]{article}
```

指定了文档的模板为文章 “article”，双栏排版 “twocolumn”，字号为 12pt。

### 1.4.3 Macro/宏

Macro is the combination of a series of commands, and it is also written in the form of command. Most of the commands are macros, except the most basic commands.

A macro package defines a series of macros for related functions. You can easily use the macros defined in a macro package by including the package file. For example, “\usepackage{showframe}” will show the frame of your document, so that you can check whether the contents are overfull or not.

However, some macro packages will conflict with each other, because a macro package is essentially a patch of previous command. Some macro packages do not work, due to the changes of a command patched by the previous included packages. Therefore, the order of including macro package is very important in some cases.

“宏 (Macro)” 在形式上也是一条命令，它是多条命令的集合。除了最基本最原始的命令以外，都可以认为是宏。一个“宏包”中定义了具有相关功能的许多条宏，导入宏包就可以使用其中的宏，可以方便地在不同文档中使用。例如 “\usepackage{showframe}” 会显示文档框架，可以用于检查文档内容是否超出页面布局。

但是有些宏包之间会相互冲突，因为宏包的本质是给之前的命令打补丁，如果先导入的一个宏包改变了某条命令，之后导入的宏包就有可能无法修改这条更新过的命令。所以导入宏包的顺序很关键，也尽量不要导入非必须的宏包。

### 1.4.4 Environment/环境

A pair of “\begin{\*}” and “\end{\*}” can switch the text to a specified environment, where “\*” represents the environment name.

```
1 \begin{*}  
2 ...  
3 \end{*}
```

---

In an environment, the text is interpreted as codes in a special way. Essentially, environment is also a kind of macro, i.e., some commands will be executed by calling `\begin{*}` to start an environment, while some other commands will be executed by calling `\end{*}` to end the environment .

一对匹配的“`\begin{环境名}`”和“`\end{环境名}`”就能切换到相应的代码环境。

```
1 \begin{*}  
2   ...  
3 \end{*}
```

每种环境中都有不同的代码运行方式，例如公式环境中会采用不同的字体、并给每行公式增加编号。环境的原理其实也是一种宏，在`\begin`的时候会运行一段开始环境的命令，在`\end`的时候会运行另一段结束环境的命令。

### 1.4.5 Including Files/引用文件

If you divide your document into different files, you can include the style file “`style_file_name.sty`” by “`\usepackage{style_file_name}`”.

You can include the “`file.tex`” by “`\input{file}`” or “`\include{file}`”. Compare to “`\input{*}`”, “`\include{*}`” will execute a “`\clearpage`” command before import the file, and switch to another temporary “`.aux`” file for generating “`.pdf`” file.

If there are some files in sub-folders, you have to include them with relative path. Note that even in the files in sub-folders, if you want to include some figures, codes or other files, the relative path is still relative to the main document file. Because all files are recursively included into the main document file.

如果你把文档分割成多个文件，你可以通过 “`\usepackage{file}`” 来引用样式文件 “`file.sty`” 。

你还可以通过 “`\input{file}`” 或 “`\include{file}`” 来引用 “`file.tex`”

文件。它们的区别在于，“\include{\*}”会在导入文件之前执行一条“\clearpage”命令，并且会生成一个新的“.aux”文件。“aux”文件是生成“.pdf”文件过程中的一个临时文件类型。

如果有一些文件放在子文件夹中，你需要通过相对路径来引用它们。注意：即使是在子文件夹的文件中引用其它的图片、代码之类的文件，也是使用相对于主文档的相对路径。因为所有的文件都是先递归地引用到主文档中，然后再进行编译的。

# Chapter 2

## Introduction of the Template/模板介绍

### 2.1 Files and Folders/文件介绍

There are the following files in the template:

- Main document: “thesis(E).tex” for English, and “thesis(J).tex” for Japanese. It defines the structure of documents and the order of chapters.
- Style file: “ThesisStyle.sty” for English, and “ThesisStyleJ.sty” for Japanese.
- Title page files: “TitlePage.tex”.
- Chapter files: “Chapters/\*.tex”.
- Appendix files: “Appendices/\*.tex”.
- Figure files: “Figures/\*.”.
- Bibliography files: “myReference.bib”.
- Abbreviation files: “Abbreviations.tex”.
- Symbol files: “Symbol.tex”.

模板中有以下文件：

- 主文档：英语为 “thesis(E).tex”，日语为 “thesis(J).tex”。它定义了文档框架和章节顺序；
- 文档样式：英语为 “ThesisStyle.sty”，日语为 “ThesisStyleJ.sty”；
- 封面：“TitlePage.tex”；

- 章: “Chapters/\*.tex”;
- 附录: “Appendices/\*.tex”;
- 图片: “Figures/\*.”;
- 参考文献: “myReference.bib”;
- 缩略语: “Abbreviations.tex”;
- 符号: “Symbol.tex”。

## 2.2 Insert Chapters and Sections/插入章节

The following commands are used to insert a new chapter, section, subsection, and subsubsection, respectively.

```
1 \chapter{Chapter Title} \label{cha:exalple}
2 \section{Section Title} \label{sec:exalple}
3 \subsection{Subsection Title} \label{sec:exalple2}
4 \subsubsection{Subsubsection Title} \label{sec:exalple3}
```

where the command “\label{label\_name}” set a label for cross-reference.

The following commands are used for cross-reference of chapter, appendix, and section/subsection/subsubsection, respectively.

```
1 \charef{chapter_label}
2 \appref{appendix_label}
3 \secref{section_label}
```

For example, Chapter [2](#) ..., Appendix [A](#) ..., [2.4](#) ....

以下命令分别用于插入章、节、小节、四级小节:

```
1 \chapter{Chapter Title} \label{cha:exalple}
2 \section{Section Title} \label{sec:exalple}
3 \subsection{Subsection Title} \label{sec:exalple2}
4 \subsubsection{Subsubsection Title} \label{sec:exalple3}
```

其中 “\label{label\_name}” 命令用于, 给章节添加标签, 以便交叉引用。

以下命令分别用于交叉引用章、附录、节/小节/四级小节:

```
1 \charef{chapter_label}
2 \appref{appendix_label}
```



例如: Chapter 2 ..., Appendix A ..., 2.4 ....

## 2.3 Appendices/附录

Appendix is also a kind of Chapter. In main document, i.e., “thesis(E).tex” or “thesis(J).tex”, there is one line of code as “\appendix”, which makes the following chapters to be appendices.

In the table of contents, there is an additional line of “Appendices” before “Appendix A”. If you do not want this additional line, you can add the command “\FistAppdfalse” below “\appendix”.

附录也是章的一种。在主文档 “thesis(E).tex” 或 “thesis(J).tex” 中，有一行代码 “\appendix”，它会把之后的章都变为附录。

在目录页中，在 “Appendix A” 之前，会有额外一行 “Appendices”。如果你不想额外增加这一行 “Appendices”，可以在 “\appendix” 下方增加一条 “\FistAppdfalse” 命令。

## 2.4 Figures/图片

### 2.4.1 Inserting Figures/插入图片

You can insert eps, pdf, jpg and png figures in  $\LaTeX$ . Eps and pdf support vector images, and you should embed fonts when generating eps and pdf figures. You can use Adobe Illustrator to handle eps files. If you are using Microsoft Visio, you can export the figure to a pdf file.

In the Japanese template, the platex compiler cannot read the image size of jpg and png files, resulting wrong alignment. Therefore, it is suggested to first convert jpg and png figures into pdf figures, and then insert the pdf figures.

$\LaTeX$  支持的图片格式有: eps, pdf, jpg, png。其中 eps 和 pdf 支持矢量图，生成 eps 和 pdf 图片时要注意嵌入字体。可以通过 Adobe Illustrator 来

处理 eps 图片，Microsoft Visio 则可以导出成 PDF 格式。如果需要将 Visio 导出 eps 图片，则需要先把 Visio 图片导出成“可缩放的向量图形 (.svg)”，然后再用 Adobe Illustrator 来把.svg 另存为 eps 图片。

在日文模板中，对于 jpg 和 png 图片，会因为日文编译器无法读取图片尺寸，导致排版出错。所以在日文模板在，建议将图片转存为 pdf 格式。



Figure 2.1. JPG figure title.

The following commands are used to insert a figure:

```
1 \begin{figure}[ht]
2   \centering
3   \includegraphics[scale=0.4]{figure_name.jpg}
4   \caption{Figure title.}
5   \label{fig:label}
6 \end{figure}
```

“\centering” makes the figure align in the center.

“\includegraphics[scale=0.4]{figure\_name.jpg}” inserts a figure and set the display size of the figure, where the argument in “[ ]” can be “scale”, “width” or “height”.

“caption{Figure title.}” sets the figure title.

In “\begin{figure}[ht]”, “[ht]” specifies the position of the figure. Options are:

- h, Place here relative to the location in  $\text{\LaTeX}$  code as much as possible.
- t, Place at the top of the page.
- b, Place at the bottom of the page.
- p, Place at a new page separately.
- !, Do not determine "good" float positions.
- H, Force the figure to be placed here.

You can use more than one options to let compiler try them in order. For example, "`[htb]`" will try to place the figure here first. If there is not enough space, the figure will then be placed at the top. If it fails to place at the top, it will be place at the bottom.

以下命令用于插入图片：

```

1 \begin{figure}[ht]
2   \centering
3   \includegraphics[scale=0.4]{figure_name.jpg}
4   \caption{Figure title.}
5   \label{fig:label}
6 \end{figure}
```

"`\centering`" 让图片居中。

"`\includegraphics[scale=0.4]{figure_name.jpg}`" 插入一张图片，并指定图片尺寸。"`[ ]`" 中的参数可以为缩放比例 "scale"，宽度 "width"，高度 "height"。

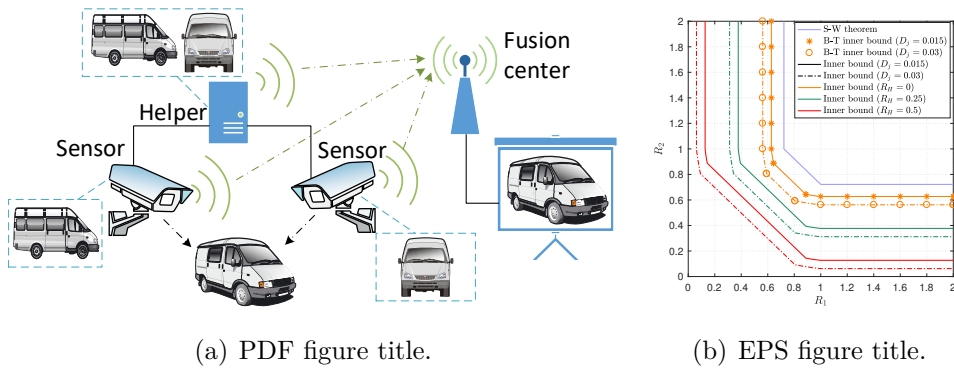
"`caption{Figure title.}`" 设置图片的标题。

"`\begin{figure}[ht]`" 中的 "`[ht]`" 指定图片的排版方式。可能的选项为：

- h, 尽量放在相对于  $\text{\LaTeX}$  代码的当前位置，与上下文呼应
- t, 放在页首
- b, 放在页末
- p, 在单独一页显示图片
- !, 不需要服从排版审美规则，来决定图片的浮动位置
- H, 强制放在当前位置

同时指定多个选项，则按顺序先后尝试符合页面要求的布局。如：“[htb]”让图片优先排在当前位置，如果排不下，则放到下一页的页首。如果下一页，页首还排不下，则放到下一页页尾。如果需要让图片强制排到当前位置，保证与上下文的顺序，则可以用 “[H]” 选项。

## 2.4.2 Inserting Sub-Figures/插入子图



**Figure 2.2.** Subfigure Examples.

The following commands are used to insert sub-figures.

以下命令用于插入子图：

```

1 \begin{figure}[ht]
2   \centering
3   \subfigure[Sub figure title 1.]{
4     \includegraphics[height=4cm]{subfig1.pdf}
5     \label{fig:subfig1}
6   }
7   \subfigure[Sub figure title 2.]{
8     \includegraphics[height=4cm]{subfig2.eps}
9     \label{fig:subfig2}
10  }
11  \caption{Figure title.}
12  \label{fig:examples}
13 \end{figure}

```

where “`\subfigure[Sub figure title 1.]{...}`” inserts the sub-figure, and its title is written within “[ ]”.

“`\subfigure[Sub figure title 1.]{...}`” 用于插入子图，“[ ]” 中是子图的标题。

### 2.4.3 Referring to Figures/引用图片

The command for cross-reference of figures is “`\fref{fig_label}`”. For example, Figure 2.1, Figure 2.2, Figure 2.2(b).

引用图片的命令为 “`\fref{fig_label}`”，例如：Figure 2.1, Figure 2.2, Figure 2.2(b).

### 2.4.4 List of Figures/插入图片列表

In main document, i.e., “thesis(E).tex” or “thesis(J).tex”, there is one line of code as “`\listoffigures`”, which prints the list of figures. You can move it to print the list of figures at another place, or delete it to remove the the list of figures.

在主文档 “thesis(E).tex” 或 “thesis(J).tex” 中，有一行代码 “`\listoffigures`”，它会自动生成和插入一页图片列表。你可以通过移动这行代码来移动图片列表，也可以删除它来移除图片列表。

## 2.5 Tables/表格

### 2.5.1 Inserting Tables/插入表格

The following commands insert a table.

```
1 \begin{table}[ht]
2   \centering
3   \begin{tabular}{...}{r|rr}
4     & a & b \\
5     \hline
6     1 & 0.25 & 0.33 \\
```

	first		
	s		s s s
	second		
test	s s		
	s	first	s s s
	second		
test	s s		
test	test		
test			

**Table 2.1.** Caption of the table

```

7 \end{tabular}
8 \caption{Caption of the table}\label{table:example}
9 \end{table}

```

In the command “`\begin{tabular}[...]{...}`”, the argument within “`[...]`” sets the vertical position of the text. The options are

- t, the line at the top is aligned with the text baseline.
- b, the line at the bottom is aligned with the text baseline.
- c or “none”, the table is centred to the text baseline.

And “`{...}`” defines the alignment and the borders of each column. The options are:

- l, left-justified column.
- c, centred column.
- r, right-justified column.
- p{‘width’}, paragraph column with text vertically aligned at the top.
- m{‘width’}, paragraph column with text vertically aligned in the middle (requires “array” package).
- b{‘width’}, paragraph column with text vertically aligned at the bottom (requires “array” package).
- |, vertical line.
- ||, double vertical line.

- `*{num}{form}`, the format form is repeated num times; for example `*{3}{|1|}` is equal to `|1|1|1|`.

In one row, you can use the following commands:

- `&`, column separator.
- `\\`, start new row (additional space may be specified after `\\` using square brackets, such as `\\[6pt]`).
- `\hline`, horizontal line between rows.
- `\newline`, start a new line within a cell (in a paragraph column).
- `\cline{i-j}`, partial horizontal line beginning in column i and ending in column j.

以下命令用于插入表格：

```

1 \begin{table}[ht]
2   \centering
3   \begin{tabular}{...}{r|rr}
4     & a & b\\
5     \hline
6     1& 0.25 & 0.33\\
7   \end{tabular}
8   \caption{Caption of the table}\label{table:example}
9 \end{table}

```

其中，命令“`\begin{tabular}{...}{...}`”中的“`[...]`”用于设置 tabular 整体的垂直对齐位置，所以只有在表格中嵌套表格的时候该选项才有效。选项如下：

- t, 顶端对齐；
- b, 底端对齐；
- c 或空参数, 中央对齐。

“`{...}`”中的参数定义了列的对齐方式和垂直表格线。选项如下：

- l, 左对齐的一列；
- c, 居中对齐的一列；
- r, 右对齐的一列；
- p{width}, 段落样式的一列，指定宽度，竖直方向顶端对齐；

- `m{'width'}`, 段落样式的一列, 指定宽度, 垂直方向居中对齐 (需要引用 “array” 宏包);
- `b{'width'}`, 段落样式的一列, 指定宽度, 垂直方向底部对齐 (需要引用 “array” 宏包);
- `|`, 垂直表格线;
- `||`, 垂直表格双线;
- `*{num}{form}`, 重复 num 次 “form” 规定的样式, 例如 `*{3}{|1|}` 就等于 `|1|1|1|`。

在一行里, 你可以使用下列命令:

- `&`, 分隔每列单元格;
- `\\`, 开始新的一行, 在 `\\` 后面可以加上宽度选项来额外增加空白高度, 例如 `\\[6pt]`;
- `\hline`, 水平表格线;
- `\newline`, 在段落样式的单元格中换行;
- `\cline{i-j}`, 只在 i 列到 j 列之间显示的水平表格线。

## 2.5.2 Referring to Tables/引用表格

For cross-reference of a table, you can use “`\tref{table:example}`”. For example, Table 2.1.

你可以使用 “`\tref{table:example}`” 来交叉引用表格, 例如 Table 2.1。

## 2.5.3 List of Tables/插入表格列表

In main document, i.e., “thesis(E).tex” or “thesis(J).tex”, there is one line of code as “`\listoftables`”, which prints the list of tables. You can move it to print the list of tables at another place, or delete it to remove the the list of tables.

在主文档 “thesis(E).tex” 或 “thesis(J).tex” 中, 有一行代码 “`\listoftables`”, 它会自动生成和插入一页表格列表。你可以通过移动这行代码来移动表格列表, 也可以删除它来移除表格列表。



## 2.6 Equations/公式

## 2.7 Inline Mode/行内公式

You can write inline equations within  $\dots$ , such as  $a+b=c$  outputs  $a + b = c$ .

你可以在 $\dots$ 之间撰写嵌入行内的公式, 比如 $a+b=c$ 会显示  $a + b = c$ 。

## 2.8 Display Mode/独行公式

You can use `\begin{equation} \dots \end{equation}` to write an equation in a separate line. For example, the following codes

你可以使用`\begin{equation} \dots \end{equation}`环境让公式另起一行, 例如, 以下代码

```
1 \begin{equation}
2   x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad \text{\label{eq:example}}
3 \end{equation}
```

outputs

会输出

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (2.1)$$

## 2.9 Aligning Equations/多行公式

You can use `\begin{align} \dots \end{align}` to align more than one lines of equations. “&” and “\\” works at the same way as tabular environment. For example, the following codes

你可以使用`\begin{align} \dots \end{align}`环境来输出并对齐多行公式。“&”用于标志公式对齐的位置, “\\”用于换行, 它们的功能和用法跟 tabular 中的一样。例如, 以下代码

```

1 \begin{align}
2   x^2 &= -(2x + 1) \\
3   x^2 &= -2x - 1 \nonumber \\
4   x^2 + 2x + 1 &= 0 \label{eq:example2} \\
5   (x+1)^2 &= 0 \label{eq:example3}.
6 \end{align}

```

outputs

会输出

$$x^2 = -(2x + 1) \tag{2.2}$$

$$x^2 = -2x - 1$$

$$x^2 + 2x + 1 = 0 \tag{2.3}$$

$$(x + 1)^2 = 0. \tag{2.4}$$

If you do not want one line to be numbered, you can use “`\nonumber`” at the end of this line.

如果你想要某一行不被编号，你可以在行尾加上 “`\nonumber`” 命令。

### 2.9.1 Referring to Equations/引用公式

For cross-reference of equations, you can use “`\eqref{eq_label}`” to refer to a equation, such as (2.4). You can also use “`\eqsref{eq_label1}{eq_label2}`” to refer to equations from “eq\_label1” to “eq\_label2”, such as (2.1 – 2.3).

“`\eqref{eq_label}`” 用来引用一行公式，例如(2.4)。

“`\eqsref{eq_label1}{eq_label2}`” 用于引用 “eq\_label1” 到 “eq\_label2” 之间的公式，例如 (2.1 – 2.3)。

## 2.10 Abbreviations and Symbols/缩略语和符号

### 2.10.1 Abbreviations Management/管理缩略语

An abbreviation should be defined when it appears for the first time. In some cases of paper revision, you may move one paragraph with the definition of an abbreviation to another place, which may make the definition appears in a wrong place. Therefore, you need to manually check and revision the position of abbreviation definitions. However,  $\LaTeX$  can automatically manage the abbreviations, so that the definition must appear at the place where the abbreviation is used for the first time.

There is an “Abbreviations.tex” in the same folder of main document. You can define abbreviations by

```
1 \newacronym{ID}{Short_Name}{Long_Name}
```

- “`\gls{ID}`” refers to an abbreviation, such as [independent and identically distributed \(i.i.d.\)](#), and for the second time referring to [i.i.d.](#).
- “`\Gls{ID}`” makes the first letter to be uppercase, such as [Joint photographic experts group \(JPEG\)](#).
- “`\glspl{ID}`” prints the abbreviation at plural form, such as [chief executive officers \(CEOs\)](#), and the singular is [CEO](#).
- “`\Glspl{ID}`” makes the abbreviation plural, and the first letter to be uppercase, such as [Wireless sensor networks \(WSNs\)](#).

在撰写文献中，常常需要使用缩略语，书写规范是将缩略语定义在第一次出现的地方。在修改文章的某些情况下，需要将定义了缩略语的某一整段移动到后面的位置，就有可能导致缩略语定义的位置也移动了。因此你可能经常要检查和修改缩略语定义的位置。 $\LaTeX$  提供了缩略语的管理功能，会自动在第一次出现的位置展开缩略语。使用方法如下：

在主文件目录下，“Abbreviations.tex”定义了许多缩略语，你可以使用以下命令来定义一条缩略语：

```
1 \newacronym{ID}{Short_Name}{Long_Name}
```

- “\gls{ID}” 用于引用缩略语，例如moving picture experts group (MPEG)，再次引用就变为MPEG.
- “\Gls{ID}” 会将首字母大写，如Bit error rate (BER)。
- “\glspl{ID}” 会引用复数形式，如probability mass functions (*pmfs*), 单数为*pmf*
- “\Glspl{ID}” 会将首字母大写并引用复数形式，如Signal-to-noise ratios (SNRs).

## 2.10.2 Symbols/符号

There is a “Symbols.tex” in the same folder of main document. You can define symbols by

```

1 \newglossaryentry{ID}{
2   sort=xxx,
3   type=symbols,
4   name={...},
5   description={...}
6 }
7 \glsadd{ID}
```

- “\newglossaryentry{ID}”, create a new entry with “ID”.
- “sort=xxx”, sort the symbols according to the string “xxx”.
- “type=symbols”, the type of glossary is “symbol”.
- “name={...}”, the display name of the symbol.
- “description={...}”, description of the symbol.
- “\glsadd{ID}”, add the symbol with “ID” into the “List of Symbols”.

在主文件目录下，“Symbols.tex”定义了许多符号，你可以使用以下命令来定义一个符号:

```

1 \newglossaryentry{ID}{
2   sort=xxx,
3   type=symbols,
4   name={...},
5   description={...}
```

```

6 }
7 \glsadd{ID}

```

- “\newglossaryentry{ID}”, 定义了一个符号, 给它一个 “ID”。
- “sort=xxx”, 以字符串 “xxx” 为顺序依据, 排列该符号。
- “type=symbols”, 术语的类型设定为符号 “symbol”。
- “name={...}”, 符号的定义 (显示名称)。
- “description={...}”, 符号的描述。
- “\glsadd{ID}”, 把该 “ID” 的符号加进符号列表 “List of Symbols”。

## 2.10.3 List of Glossaries/术语列表

### 2.10.3.1 Inserting the Lists/插入列表

In main document, i.e., “thesis(E).tex” or “thesis(J).tex”, there are two lines of code as

- “\printglossary[type=\acronymtype]”, which prints the list of abbreviations.
- “\printglossary[type=symbols]”, which prints the list of symbols.

You can move it to print the list at another place, or delete it to remove the the list.

在主文档 “thesis(E).tex” 或 “thesis(J).tex” 中, 有两行代码:

- “\printglossary[type=\acronymtype]”, 生成和插入缩略语列表;
- “\printglossary[type=symbols]”, 生成和插入符号列表。

你可以通过移动这些代码来移动术语列表, 也可以删除它们来移除术语列表。

### 2.10.3.2 Sorting the Abbreviations/缩略语排序

If you want to sort the list of abbreviations in different order, you can specify the sort string when you define the abbreviation, as:

```

1 \newacronym[sort=something]{ID}{Short_Name}{Long_Name}

```

如果你需要指定缩略语的排列顺序，可以在定义缩略语时指定 `sort` 的值，然后就会依据 `sort` 字符串来排序。例如：

```
1 \newacronym[sort=something]{ID}{Short_Name}{Long_Name}
```

### 2.10.3.3 Configuration of Texmaker/配置 Texmaker

No additional configuration is required for Overleaf. However, you need to configure Texmaker for printing the list of glossaries. First, as shown in Figure 2.3, click “User → User Commands → Edit User Commands”.

Overleaf 中不需要额外配置就可以插入术语列表。如果使用 Texmaker，则需要配置。首先，如 Figure 2.3所示，点击“自定义 → 自定义命令 → 编辑自定义命令”。

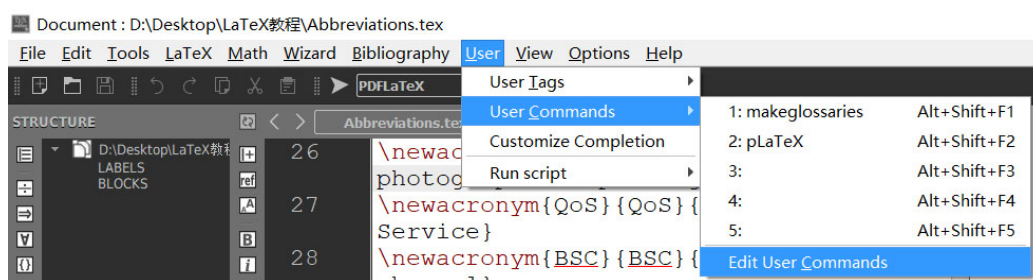


Figure 2.3. Edit User Commands.

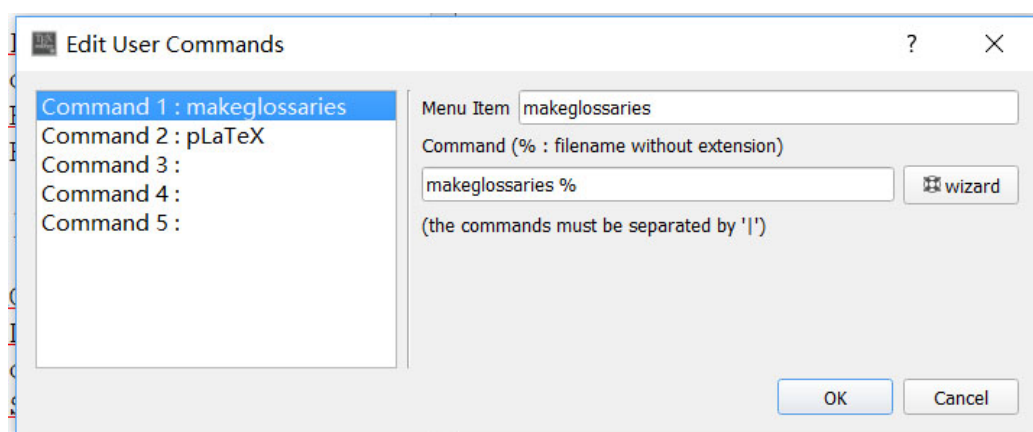


Figure 2.4. Make glossaries.

Then, as shown in Figure 2.4, set the Menu Item as “makeglossaries”, and the Command as “`makeglossaries %`”. Now you can use shortcut key “Alt + Shift + F1” to update and print the lists of glossaries. When you want to update the list of glossaries, note that you should do in the order: “compiling to pdf” → “makeglossaries” → “compiling to pdf again”.

然后，如 Figure 2.4所示，在菜单项中填上名称 “makeglossaries”，在命令中填上 “`makeglossaries %`”。至此，你可以使用快捷键 “Alt + Shift + F1” 来更新和打印术语列表。如果你需要更新术语列表，注意要按如下顺序编译文件：生成一次 pdf 文件 → “makeglossaries” → 再生成一次 pdf 文件。

## 2.11 Bibliography and Citations/引用参考文献

In main document, i.e., “thesis(E).tex” or “thesis(J).tex”, there are two lines of code as

- “`\bibliographystyle{...}`”, which sets the style of bibliography.
- “`\bibliography{myReference}`”, which includes the bib file “myReference.bib”, and prints the list of bibliography.

There is a “myReference.bib” in the same folder of main document. The bibliography entry is defined by BibTeX code as:

```
1 @BibType{ID,  
2   title = {...},  
3   author = {...},  
4   year = {2019},  
5   ...  
6 }
```

You can use Google Scholar to easily get the BibTeX of a reference.

You can use “`\cite{ID1, ID2, ...}`” to cite one or more references, such as [3], [4–7], [3, 6].

在主文档 “thesis(E).tex” 或 “thesis(J).tex” 中，有两行代码：

- “`\bibliographystyle{...}`” 设置了参考文献的样式；

- “\bibliography{myReference}”, 导入参考文献数据库 “myReference.bib”, 并打印参考文献列表。

“myReference.bib” 与主文件在同一目录下。一条文献数据通过以下 BibTeX 代码来定义:

```
1 @BibType{ID,  
2   title = {...},  
3   author = {...},  
4   year = {2019},  
5   ...  
6 }
```

你可以通过谷歌学术等网站来轻松获取一条文献的 BibTeX 代码。

你可以通过 “\cite{ID1, ID2, ...}” 来引用一条或多条参考文献, 例如: [3], [4–7], [3, 6].

### 2.11.1 Compiling Order/编译顺序

In Overleaf, it will automatically compile the files in default order. However, in other software, you may need to manually compile the files according to the following order:

1. Compile to pdf file for the first time. It records the bib ID cited in the document with the order.
2. Compile bib file by BibTeX. It generates the bib list according to the recorded bib ID.
3. Compile to pdf file for the second time. It prints the bib list into the document and give order numbers to them.
4. Compile to pdf file for the third time. It updates the order number at cited place in the document.

在 Overleaf 中会自动按顺序编译, 在其它软件中, 你需要按如下顺序编译:

1. 第一次编译成 pdf 文件。这次会将文件中引用的文献 ID 按顺序记录到临时文件中。



2. 用 BibTeX 编译 bib 文件。这次会根据临时文件中记录的文献 ID 生成文献列表。
3. 第二次编译成 pdf 文件。这次会将文献列表加入到文档中，并为文献列表中的文献生成编号。
4. 第三次编译成 pdf 文件。这次会将文献列表中的编号更新到文章中引用的位置。

## 2.12 Algorithms/算法

The following commands show an example of algorithm.

以下代码是一个插入算法的例子：

```
1 \begin{algorithm}
2   \caption{An Example}
3   \label{alg:example}
4   \begin{algorithmic}
5     \REQUIRE {$x$, $n$}
6     \ENSURE {$y$}
7     \STATE {set $y=1$;}
8
9     \IF {$n==0$}
10      \STATE {set $y=1$;}
11    \ELSIF {$n>0$}
12      \FOR {$i=1$ \TO $n$}
13        \STATE {set $y = y \times x$;}
14      \ENDFOR
15    \ELSE
16      \FOR {$i=n$ \TO $-1$}
17        \STATE {set $y = y \div x$;}
18      \ENDFOR
19    \ENDIF
20  \end{algorithmic}
21 \end{algorithm}
```

outputs:

会输出：

---

**Algorithm 2.1.** An Example

---

**Input:**  $x, n$   
**Output:**  $y$   
set  $y = 1$ ;  
**if**  $n == 0$  **then**  
    set  $y = 1$ ;  
**else if**  $n > 0$  **then**  
    **for**  $i = 1$  **to**  $n$  **do**  
        set  $y = y \times x$ ;  
    **end for**  
**else**  
    **for**  $i = n$  **to**  $-1$  **do**  
        set  $y = y \div x$ ;  
    **end for**  
**end if**

---

You can use “`\algref{alg_label}`” to refer to an algorithm, such as Algorithm 2.1.

“`\algref{alg_label}`” 用于引用一个算法，例如 Algorithm 2.1。

## 2.13 Codes/代码

### 2.13.1 Inserting Code/插入代码

The following commands insert a part of codes.

```
\begin{lstlisting}[...]  
... (Source codes)  
\end{lstlisting}
```

Arguments in “[...]” can be:

- “`language=C++`”, set the programming language of the code.
- “`title={No Number Title}`”, set no number title.

- “caption={Numbered Title}”, set numbered title.
- “label={code:example} ”, set the label of the code.

For example,

No Number Title

```
1 int main() {
2   for (int i=0; i<3; i++){
3     cout<<i<<endl;
4   }
5   return 0;
6 }
```

以下命令用于插入一段代码

```
\begin{lstlisting}[...]
... (Source codes)
\end{lstlisting}
```

“[...]” 中的参数可以为：

- “language=C++”, 设置编程语言；
- “title={No Number Title}”, 设置无编号的标题；
- “caption={Numbered Title}”, 设置带编号的标题；
- “label={code:example} ”, 设置引用标签；

一个例子：

No Number Title

```
1 int main() {
2   for (int i=0; i<3; i++){
3     cout<<i<<endl;
4   }
5   return 0;
6 }
```

## 2.13.2 Importing Codes from Files/从文件中导入代码

The following command imports code from the file “filename.extension”.

以下命令用于从文件中导入代码。

```
1 \lstinputlisting [...] { filename.extension }
```

The arguments in “[...]” are the same usage as “\begin{lstlisting} [...]”. For example,

“[...]” 中的参数与 “\begin{lstlisting}[...]” 的用法相同。一个例子：

Code 2.1: Input Code from File

```
1 \chapter{Conclusion}\label{cha:conclusion}  
2  
3 This thesis ...
```

### 2.13.3 Referring to Codes/交叉引用代码

You can use “\coderef{code\_label}” to refer to the code, such as Code 2.1.

“\coderef{code\_label}” 用于交叉引用代码，例如：Code 2.1。

## 2.14 Definition, Lemma and Proposition/定义、引理和命题

The following commands are used to insert a new definition, lemma, and proposition, respectively.

以下代码分别用于插入定义、引理和命题：

```
1 \Definition \label{def:example} Example Definition  
2 \Lemma \label{lem:example} Example Lemma  
3 \Proposition \label{pro:example} Example Proposition
```

which outputs:

以上代码会得到如下内容：

**Definition 2.1:** Example Definition

**Lemma 2.1:** Example Lemma

**Proposition 2.1:** Example Proposition

You can simply use the same command “`\ref{label}`” to refer to a definition, lemma, or proposition. For example, [Definition 2.1](#), [Lemma 2.1](#), [Proposition 2.1](#).

你可以使用同一条命令 “`\ref{label}`” 来引用定义、引理或命题，例如：[Definition 2.1](#), [Lemma 2.1](#), [Proposition 2.1](#)。



# Appendix A

## Example

Number test in Appendices.

$$x(i) = x^i, \text{ for } i = \{1, 2, \dots, n\} \tag{A.1}$$

(A.1)

Table A.1

Parameter	Value
Block length	10000 bits
Number of Blocks	1000

**Table A.1.** Table in Appendix





# Appendix B

## Code Example

```
1 function result = H2(d)
2     if d==0 || d==1
3         result=0;
4     else
5         result=-d*log2(d)-(1-d)*log2(1-d);
6     end
7 end
8
9 function mid = H2_inv(r)
10     accuracy=1e-9;
11     if r<=0
12         mid=0;
13
14         return;
15     end
16     min=0;
17     max=0.5;
18     mid=0.25;
19     while max-min>accuracy
20         tmp=H2(mid);
21         if abs(tmp-r)<accuracy
22             break;
23         end
24         if tmp>r
25             max=mid;
26         else
27             min=mid;
28         end
```

```

29         mid=(max+min) / 2;
30     end
31     mid;
32 end
33
34 function h=joint_entropy(P)
35     h=0;
36     n=numel(P);
37     P_=ones(1,n)-P;
38     if n==0
39         h=0;
40     elseif n==1
41         h=1;
42     else
43         for i=1:2^n
44             A=bitget(i,1:n);
45             j=A==1;
46             k=A==0;
47             Pj=P(j);
48             Pk=P(k);
49             Pj_=P_(j);
50             Pk_=P_(k);
51             q=0.5*(prod(Pj)*prod(Pk_)+prod(Pj_)*prod(Pk));
52             h=h-q*log2(q);
53         end
54     end
55 end

```

# References

- [1] linwest, “JAISTthesisLaTeX,” 2019. [Online]. Available: <https://github.com/linwest/JAISTthesisLaTeX>
- [2] JAIST, “修士（情報科学）学位論文 / 課題研究報告書作成要領,” 2019. [Online]. Available: <https://www.jaist.ac.jp/education/degree-private/is-degree.html>
- [3] T. Berger, “Multiterminal source coding,” in *The Information Theory Approach to Communications*, G. Longo, Ed. New York: Springer-Verlag, 1978, pp. 171–231.
- [4] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
- [5] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec.*, vol. 4, no. 142-163, p. 1, Mar. 1959.
- [6] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge, UK: Cambridge University Press, 2011.
- [7] ISO/IEC International Standard 11172-3, “Coding of moving pictures and associated audio for digital storage media at up to about 15 Mbit/s - Part3: Audio,” 1993.



# Publications

- [1] Author 1, Author 2 and Author 3, “This is the paper title”, Journal name, vol. xx, no. xx, pp. xx-xx, Feb, 2019.
- [2] Author 1 and Author 2, “This is another paper title”, Conference name, City name, Country name, Jan, 2019.

