

L^AT_EX 入门

zkli

zkli@ir.hit.edu.cn

哈尔滨工业大学 SCIR 研究中心



目录

- 1 L^AT_EX 是什么
- 2 helloworld.tex
- 3 章节、段落
- 4 数学公式
- 5 算法、代码、表格、图片和引用
- 6 中文支持
- 7 Beamer
- 8 参考文献
- 9 几个有意思的工具

- 由 Donald E. Knuth (高德纳, 唐纳德·克努特) 1977 年开始设计
- 由三个希腊字母组成, 发音为 “Tech” (泰克)
- 最初用于出版工业的数字印刷设备
- 从第 3 版之后的版本号越来越接近圆周率 π , 目前的版本是 3.1415926
- 非常稳定, 高德纳悬赏奖励任何能够在 T_EX 中发现程序漏洞 (bug) 的人

- 由 Donald E. Knuth (高德纳, 唐纳德 · 克努特) 1977 年开始设计
- 由三个希腊字母组成, 发音为 “Tech” (泰克)
- 最初用于出版工业的数字印刷设备
- 从第 3 版之后的版本号越来越接近圆周率 π , 目前的版本是 3.1415926
- 非常稳定, 高德纳悬赏奖励任何能够在 T_EX 中发现程序漏洞 (bug) 的人

- 由 Donald E. Knuth (高德纳, 唐纳德·克努特) 1977 年开始设计
- 由三个希腊字母组成, 发音为 “Tech” (泰克)
- 最初用于出版工业的数字印刷设备
- 从第 3 版之后的版本号越来越接近圆周率 π , 目前的版本是 3.1415926
- 非常稳定, 高德纳悬赏奖励任何能够在 T_EX 中发现程序漏洞 (bug) 的人

- 由 Donald E. Knuth (高德纳, 唐纳德 · 克努特) 1977 年开始设计
- 由三个希腊字母组成, 发音为 “Tech” (泰克)
- 最初用于出版工业的数字印刷设备
- 从第 3 版之后的版本号越来越接近圆周率 π , 目前的版本是 3.1415926
- 非常稳定, 高德纳悬赏奖励任何能够在 TeX 中发现程序漏洞 (bug) 的人

- 由 Donald E. Knuth (高德纳, 唐纳德·克努特) 1977 年开始设计
- 由三个希腊字母组成, 发音为“Tech”(泰克)
- 最初用于出版工业的数字印刷设备
- 从第 3 版之后的版本号越来越接近圆周率 π , 目前的版本是 3.1415926
- 非常稳定, 高德纳悬赏奖励任何能够在 TeX 中发现程序漏洞 (bug) 的人

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

高德纳的传奇



Figure: 高公萌照

- 《The Art of Computer Programming》作者
- 美国国家科学院院士
- 美国工程院院士
- 美国艺术与科学院院士
- 斯坦福大学计算机系教授（30 岁）
- 最年轻的图灵奖获得者（36 岁）

<http://www-cs-faculty.stanford.edu/~knuth/>

- 发音为“Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- 发音为“Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- 发音为 “Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- 发音为“Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- 发音为“Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- 发音为“Lay-Tech” (雷态克)
- 由最早由计算机学家 Lamport 在 20 世纪 80 年代初开发
- L^AT_EX 是在 Plain T_EX 的基础上开发出的一种更为简单的语言
- 提供了预先定义好的专业页面设置
- 短时间内生成具有书籍质量的印刷品
- 还可以用来生成矢量图形

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构
- **省时省力**。可以免去很多费力不讨好的页面样式设计工作

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构
- **省时省力**。可以免去很多费力不讨好的页面样式设计工作
- **资源丰富**。有大量的模版可以借鉴，很容易套用

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构
- **省时省力**。可以免去很多费力不讨好的页面样式设计工作
- **资源丰富**。有大量的模版可以借鉴，很容易套用
- **统一标准**。L^AT_EX 是科研界标准，很多期刊提供模板，甚至提供在线编译功能

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构
- **省时省力**。可以免去很多费力不讨好的页面样式设计工作
- **资源丰富**。有大量的模版可以借鉴，很容易套用
- **统一标准**。L^AT_EX 是科研界标准，很多期刊提供模板，甚至提供在线编译功能
- **当前国情**。用 L^AT_EX 写算法导论报告分数会高一些

- **模板漂亮**。让你的文档足够漂亮以应对各种场合 -
- **编写方便**。可以容易地编辑公式、生成脚注、索引、目录、参考文献等复杂的文档结构
- **省时省力**。可以免去很多费力不讨好的页面样式设计工作
- **资源丰富**。有大量的模版可以借鉴，很容易套用
- **统一标准**。L^AT_EX 是科研界标准，很多期刊提供模板，甚至提供在线编译功能
- **当前国情**。用 L^AT_EX 写算法导论报告分数会高一些
- **业界良心**。体会码农的乐趣

- 不是所见即所得，上手不如 word 简单，但是一劳永逸。
- 组织结构混乱的文章不太容易写，但我们追求的就是清晰的结构。
- 自己重新设计整个排版很花时间，但我们没有设计排版的需求。
- 很难做出很花哨的效果，但我们不会去做花哨的效果。

- 不是所见即所得，上手不如 word 简单，但是一劳永逸。
- 组织结构混乱的文章不太容易写，但我们追求的就是清晰的结构。
- 自己重新设计整个排版很花时间，但我们没有设计排版的需求。
- 很难做出很花哨的效果，但我们不会去做花哨的效果。

- 不是所见即所得，上手不如 word 简单，但是一劳永逸。
- 组织结构混乱的文章不太容易写，但我们追求的就是清晰的结构。
- 自己重新设计整个排版很花时间，但我们没有设计排版的需求。
- 很难做出很花哨的效果，但我们不会去做花哨的效果。

- 不是所见即所得，上手不如 word 简单，但是一劳永逸。
- 组织结构混乱的文章不太容易写，但我们追求的就是清晰的结构。
- 自己重新设计整个排版很花时间，但我们没有设计排版的需求。
- 很难做出很花哨的效果，但我们不会去做花哨的效果。

——能抓到老鼠的猫就是好猫

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

- L^AT_EX 各种字母体型优美，仪态万方
- 文档大小较之 Word 小很多，并且是文本格式

——能抓到老鼠的猫就是好猫

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

- L^AT_EX 各种字母体型优美，仪态万方
- 文档大小较之 Word 小很多，并且是文本格式

——能抓到老鼠的猫就是好猫

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

$$\sum_n f(n) = \int_{t>1} f(t)dt$$

- L^AT_EX 各种字母体型优美，仪态万方
- 文档大小较之 Word 小很多，并且是文本格式

接下来要介绍的内容

- `helloworld.tex`
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

接下来要介绍的内容

- helloworld.tex
- 基本语法介绍
- 章节、段落
- 数学公式
- 插入代码、图片、表格和引用
- 中文支持
- Beamer(Slides NOT PPT)
- 参考文献的加入
- 几个省事的软件

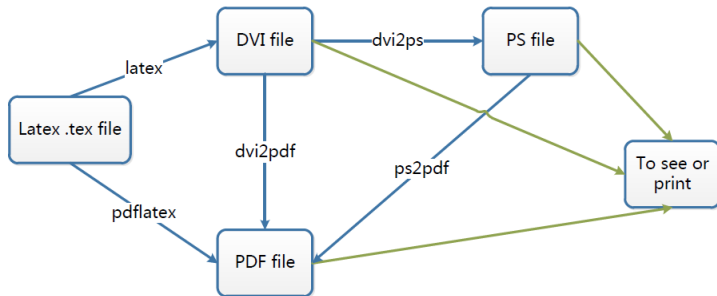
- TEX 套装

- Windows: MiKTeX, CTeX
- Linux: teTeX
- 跨平台: TeX Live, MacTeX, ConTeXt

- 编译器

- Windows: TeXnicCenter, MeWa, WinShell, BakoMa TeX, Inlage, WinEdt, ...
- Linux: Gedit LaTeX Plugin, Gummi, Winefish, Kile, ...
- 跨平台: LyX, Texmaker, AUCTION, TeXlipse, TeXworks, ...
- 详细比较:
http://en.wikipedia.org/wiki/Comparison_of_TeX_editors
- vim 和 emacs 可以通过相应插件来支持 L^AT_EX

编译过程



其中 DVI、PostScript 和 PDF 为三种输出格式。

动手写一个 hello L^AT_EX

HelloLatex.tex

```
% !Mode:: "TeX:UTF-8"  
\documentclass{article}  
\author{fool}  
\title{My First \LaTeX{} article}  
  
\begin{document}  
  
\maketitle  
  
    Wow! This is my FIRST \LaTeX{} Article!  
  
    Hello World!  
  
\end{document}
```

基本语法介绍

- **空格**: 连续的空格被认为只有一个, 用~ 表示空格
- 有些特殊的符号是不能直接使用的:
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行: \\
- 分段: 文字之后的一个空行是段落结束的标志
- 注释: % 之后都文字都是注释, 是无效的语句
- LaTeX 的命令: 以\ 开始:

```
\section{第一段}
```

```
\emph{强调}
```

基本语法介绍

- 空格：连续的空格被认为只有一个，用~ 表示空格
- 有些特殊的符号是不能直接使用的：
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行：\\
- 分段：文字之后的一个空行是段落结束的标志
- 注释：% 之后都文字都是注释，是无效的语句
- LaTeX 的命令：以\ 开始：

```
\section{第一段}
```

```
\emph{强调}
```

基本语法介绍

- 空格：连续的空格被认为只有一个，用~ 表示空格
- 有些特殊的符号是不能直接使用的：
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行：\\
- 分段：文字之后的一个空行是段落结束的标志
- 注释：% 之后都文字都是注释，是无效的语句
- LaTeX 的命令：以\ 开始：

```
\section{第一段}
```

```
\emph{强调}
```


基本语法介绍

- 空格：连续的空格被认为只有一个，用~ 表示空格
- 有些特殊的符号是不能直接使用的：
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行：\\
- 分段：文字之后的一个空行是段落结束的标志
- 注释：% 之后都文字都是注释，是无效的语句
- LaTeX 的命令：以\ 开始：

```
\section{第一段}
```

```
\emph{强调}
```

基本语法介绍

- 空格：连续的空格被认为只有一个，用~ 表示空格
- 有些特殊的符号是不能直接使用的：
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行：\\
- 分段：文字之后的一个空行是段落结束的标志
- 注释：% 之后都文字都是注释，是无效的语句
- LaTeX 的命令：以\ 开始：

```
\section{第一段}
```

```
\emph{强调}
```

基本语法介绍

- 空格：连续的空格被认为只有一个，用~ 表示空格
- 有些特殊的符号是不能直接使用的：
\$ & % # _ { } 应该写成 \ \$ \& \% \# _ \{ \}
- 断行：\\
- 分段：文字之后的一个空行是段落结束的标志
- 注释：% 之后都文字都是注释，是无效的语句
- LaTeX 的命令：以\ 开始：

`\section{第一段}`

`\emph{强调}`

章节、段落

- in article:

章节、段落

- in article:
 - `\section{section name}`

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`
- in book:

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`
- in book:
 - `\chapter{chapter name}`

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`
- in book:
 - `\chapter{chapter name}`
 - `\part{part name}`

章节、段落

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`
- in book:
 - `\chapter{chapter name}`
 - `\part{part name}`
- in beamer:

- in article:
 - `\section{section name}`
 - `\subsection{subsection name}`
 - `\paragraph{paragraph name}`
 - `\subparagraph{subparagraph name}`
- in book:
 - `\chapter{chapter name}`
 - `\part{part name}`
- in beamer:
 - `\section{section name}`

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)

- $\langle \dots \rangle$
- $\backslash\begin{math} \dots \backslash\end{math}$
- $\$ \dots \$$

- 行间公式 (display mode)

- $\backslash\begin{equation} \dots \backslash\end{equation}$
- $\langle \dots \rangle$
- $\backslash\begin{displaymath} \dots \backslash\end{displaymath}$
- $\$\$ \dots \$\$$

- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - (\cdots)
 - $\begin{math} \cdots \end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\begin{equation} \cdots \end{equation}$
 - $[\cdots]$
 - $\begin{displaymath} \cdots \end{displaymath}$
 - $\$ \$ \cdots \$ \$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslash\begin{math} \cdots \backslash\end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslash\begin{equation} \cdots \backslash\end{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslash\begin{displaymath} \cdots \backslash\end{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslash\begin{math} \cdots \backslash\end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslash\begin{equation} \cdots \backslash\end{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslash\begin{displaymath} \cdots \backslash\end{displaymath}$
 - $\$ \$ \cdots \$ \$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslash\begin{math} \cdots \backslash\end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslash\begin{equation} \cdots \backslash\end{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslash\begin{displaymath} \cdots \backslash\end{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslashbegin{math} \cdots \backslashend{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslashbegin{equation} \cdots \backslashend{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslashbegin{displaymath} \cdots \backslashend{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslash\begin{math} \cdots \backslash\end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslash\begin{equation} \cdots \backslash\end{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslash\begin{displaymath} \cdots \backslash\end{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslashbegin{math} \cdots \backslashend{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslashbegin{equation} \cdots \backslashend{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslashbegin{displaymath} \cdots \backslashend{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslashbegin{math} \cdots \backslashend{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslashbegin{equation} \cdots \backslashend{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslashbegin{displaymath} \cdots \backslashend{displaymath}$
 - $$$ \cdots $$$
- 多行公式

传说中的方便插入公式?

article 中的两种公式:

- 行内公式 (inline mode)
 - $\backslash(\cdots \backslash)$
 - $\backslash\begin{math} \cdots \backslash\end{math}$
 - $\$ \cdots \$$
- 行间公式 (display mode)
 - $\backslash\begin{equation} \cdots \backslash\end{equation}$
 - $\backslash[\cdots \backslash]$
 - $\backslash\begin{displaymath} \cdots \backslash\end{displaymath}$
 - $\$\$ \cdots \$\$$
- 多行公式

用例子说明两种公式的区别

Input

I know that you know $1+1=2$, but I know $2-1=1$, which you don't know. Now look at it $2-1=1$ I DO know more than you

Output

I know that you know $1 + 1 = 2$, but I know $2 - 1 = 1$, which you don't know. Now look at it

$$2 - 1 = 1$$

I DO know more than you.

分式、上下标和开方

Input

```
$$\frac{2011}{2012}, x_1, x_2, \dots, x_n, a^2+b^2=c^2, \\ x_1^2+x_2^2+\dots+x_n^2=r^{100}, \sqrt{x+1}, \\ \sqrt[3]{x^2+1}$$
```

Output

$$\frac{2011}{2012}, x_1, x_2, \dots, x_n, a^2 + b^2 = c^2, x_1^2 + x_2^2 + \dots + x_n^2 = r^{100}, \sqrt{x+1}, \sqrt[3]{x^2+1}$$

三角函数

Input

```
$$\sin x, \cos x, \tan x, \arctan x, \sinh x, \cosh x,  
\max x, \min x, \ln x, \log x, \log_2 x.$$
```

Output

$\sin x, \cos x, \tan x, \arctan x, \sinh x, \cosh x, \max x, \min x, \ln x, \log x, \log_2 x.$

求和、极限和积分

Input

```
$$\lim_{n \to \infty} a_n = 1, \sum_{n=1}^{\infty} n = 5050, \\ \int_a^b f(x) \mathrm{d}x = I$$
```

Output

$$\lim_{n \rightarrow \infty} a_n = 1, \sum_{n=1}^{\infty} n = 5050, \int_a^b f(x) dx = I$$

关系符号、希腊字母和部分数学标记

Input

```
$$a \times b, c \div d, a < b, b = c, c \neq d, d > e,
e \geq f, f \leq g $$
$$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\phi\varphi\omega, |A|, \|A\|, \vec{a}, \overrightarrow{AB}, \tilde{x},
\widetilde{xyz}, \mathrm{sin}$$
```

Output

$a \times b, c \div d, a < b, b = c, c \neq d, d > e, e \geq f, f \leq g$

$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\phi\varphi\omega, |A|, \|A\|, \vec{a}, \overrightarrow{AB}, \tilde{x}, \widetilde{xyz}, \sin$

矩阵

Input

```
\begin{equation}
\left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
\end{equation}
```

Output

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (1)$$

矩阵 v2.0

Input

```
\begin{equation}
\left\{
\begin{array}{c|c|c}
a_{11} & a_{12} & \\
\hline
a_{21} & & a_{23} \\
& a_{32} & a_{33}
\end{array}
\right.
\end{equation}
```

Output

$$\left\{ \begin{array}{c|c|c} a_{11} & a_{12} & \\ \hline a_{21} & & a_{23} \\ & a_{32} & a_{33} \end{array} \right. \quad (2)$$

分段函数

Input

```
\begin{equation}
\chi_A(x)=
\left\{
\begin{array}{ll}
1, & x \in A \\
0, & x \notin A
\end{array}
\right.
\end{equation}
```

Output

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (3)$$

Input

```
\begin{algorithm}[H]
\caption{An Algorithm}
\begin{algorithmic}[1]
\FOR{each  $i$  in  $[1,9]$ }
\STATE initialize  $T_{\{i\}}$ ;
\STATE  $T_{\{i\}}$ ;
\ENDFOR
\end{algorithmic}
\end{algorithm}
```

Output

Algorithm 1 An Algorithm

- 1: **for** each i in $[1,9]$ **do**
 - 2: initialize T_i ;
 - 3: T_i ;
 - 4: **end for**
-

Input

```
\begin{algorithm}[H]
\caption{Description}
\begin{algorithmic}[1]
\REQUIRE ~~\\
The samples,  $P_n$ ;\\
\ENSURE ~~\\
Classifiers,  $E_n$ ;\\
\STATE Samples  $T_n$ ;
\STATE Classifiers  $E$ 
\end{algorithmic}
\end{algorithm}
```

Output

Algorithm 2 Description

Require:

The samples, P_n ;

Ensure:

Classifiers, E_n ;

1: Samples T_n ;

2: Classifiers E

插入代码 v1.0

Input

```
\usepackage{listings}
\begin{lstlisting}
[language=C]
int main(void)
{
printf("Hello world!\n");
return 0;
}
\end{lstlisting}
```

```
int main(void)
{
printf("Hello world!\n");
return 0;
}
```

插入代码 v6.0

Input

```
\lstset
{numbers=left,blarblar}
\begin{lstlisting}
[language=C]
int main(void)
{
printf("Hello world!\n");
return 0;
}
\end{lstlisting}
```

```
1 int main(void)
2 {
3 /* print a string*/
4 printf("Hello □world!\n");
5 return 0;
6 }
```

插入表格

Input

```
\begin{tabular}{l|l}  
Name & score \\  
\hline  
You & 100 \\  
Me & 59  
\end{tabular}
```

Output

Name	score
You	100
Me	59

图片格式支持什么？

ps : PostScript. 由 Adobe 公司推出，是一种页面描述语言。独立于设备，能综合处理文字和图像，擅长于描述矢量图形。

eps : Encapsulated PostScript. 封装的 PostScript，是 PostScript 的一个子集，每个 eps 文件只有一个页面。eps 格式的图片与 \LaTeX 最兼容。

pdf : Portable Document Format

非矢量图 :jpg, png, bmp, …: 各种其他图片格式，也被 \LaTeX 支持

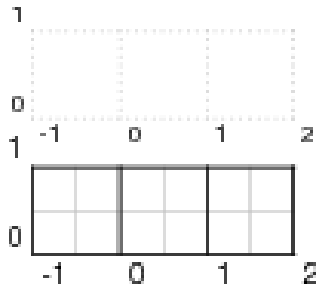
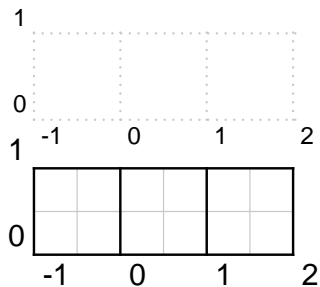
插入 jpg 图片

Input

```
\begin{figure}[h]  
  \centering  
  \includegraphics[width=0.3\textwidth,angle=20]{jpg_figure}  
\end{figure}
```



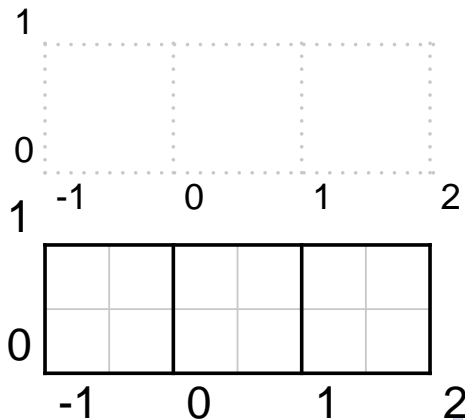
矢量图 vs 非矢量图



插入 eps 图片

Input

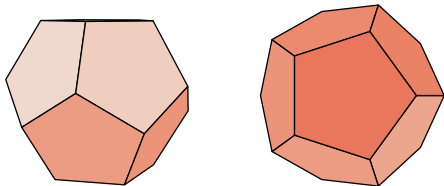
```
\includegraphics[width=0.5\textwidth]{eps_figure.eps}
```



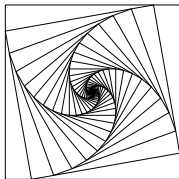
插入 pdf 图片

Input

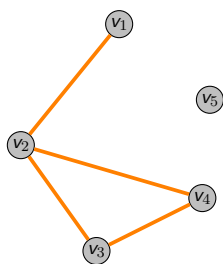
```
\includegraphics [width=0.5\textwidth] {pdf_figure.pdf}
```



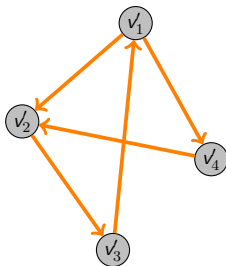
插入 MetaPost



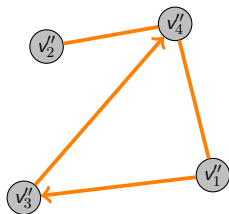
插入 TikZ



(a) G (无向图)



(b) G' (有向图)



(c) G'' (混合图)

Input

```
\begin{equation}
\lim_{x \to 0} \frac{\sin x}{x} = 1
\label{myequation}
\end{equation}
(\ref{myequation}) 式是一个很重要的
的极限
```

Output

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad (4)$$

(4) 式是一个很重要的极限

- CJK

- CJK
 - 一个德国人开发的中、日、韩文字处理包

- CJK

- 一个德国人开发的中、日、韩文字处理包
- 是 \LaTeX 系统的一个宏包，比较通用

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯
 - 缺点是需要进行预处理，引入其它 LaTeX 资源时会有一些问题

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯
 - 缺点是需要进行预处理，引入其它 \LaTeX 资源时会有一些问题
- 天元

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯
 - 缺点是需要进行预处理，引入其它 LaTeX 资源时会有一些问题
- 天元
 - 华东师大的肖刚、陈志杰等开发的中文 $\text{T}_{\text{E}}\text{X}$ 系统

中文支持

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯
 - 缺点是需要进行预处理，引入其它 LaTeX 资源时会有一些问题
- 天元
 - 华东师大的肖刚、陈志杰等开发的中文 $\text{T}_{\text{E}}\text{X}$ 系统
- $\text{X}_{\text{E}}\text{\LaTeX}$

- CJK
 - 一个德国人开发的中、日、韩文字处理包
 - 是 \LaTeX 系统的一个宏包，比较通用
- CCT
 - 中科院张林波教授开发的中文系统
 - 中文字体比较多，排版方式考虑中文使用习惯
 - 缺点是需要进行预处理，引入其它 LaTeX 资源时会有一些问题
- 天元
 - 华东师大的肖刚、陈志杰等开发的中文 $\text{T}_{\text{E}}\text{X}$ 系统
- $\text{X}_{\text{E}}\text{\LaTeX}$
 - 从底层支持中文

CJK 格式

```
\documentclass{article}
\usepackage{CJK}
\begin{document}
\begin{CJK*}{GBK}{kai}
这是中文楷体字。
\end{CJK*}
\end{document}
```


CCT 中文支持

老版本 CCT 格式

```
\documentclass{cctart}
\begin{document}
\kaishu 这是中文楷体字。
\end{document}
```

新版本 CCT 格式

```
\documentclass [CJK]{cctart}
\begin{document}
\kaishu 这是中文楷体字。
\end{document}
```

X_YTeX 中文支持

X_YTeX 格式 1 (编码保存为 UTF-8)

```
\documentclass{ctexart}
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

X_YTeX 格式 2

```
\documentclass{article}
```

```
\usepackage{ctex}
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

Beamer 是一个用于制作演示文稿的 LaTeX 文档类，由 Till Tantau 编写。相对于其它同类工具，Beamer 有如下这些优点：

Beamer 是一个用于制作演示文稿的 LaTeX 文档类，由 Till Tantau 编写。相对于其它同类工具，Beamer 有如下这些优点：

- **标准统一**。标准的 \LaTeX 指令在 Beamer 文稿中可直接使用；

Beamer 是一个用于制作演示文稿的 LaTeX 文档类，由 Till Tantau 编写。相对于其它同类工具，Beamer 有如下这些优点：

- **标准统一**。标准的 \LaTeX 指令在 Beamer 文稿中可直接使用；
- **主题丰富**。提供了许多主题 (theme)，可以很容易改善简档的外观；

Beamer 是一个用于制作演示文稿的 LaTeX 文档类，由 Till Tantau 编写。相对于其它同类工具，Beamer 有如下这些优点：

- **标准统一**。标准的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 指令在 Beamer 文稿中可直接使用；
- **主题丰富**。提供了许多主题 (theme)，可以很容易改善简档的外观；
- **注重内容**。致力于更好的表现演讲内容，而不是仅仅为了让页面好看；

Beamer 是一个用于制作演示文稿的 LaTeX 文档类，由 Till Tantau 编写。相对于其它同类工具，Beamer 有如下这些优点：

- **标准统一**。标准的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 指令在 Beamer 文稿中可直接使用；
- **主题丰富**。提供了许多主题 (theme)，可以很容易改善简档的外观；
- **注重内容**。致力于更好的表现演讲内容，而不是仅仅为了让页面好看；
- **方便定制**。页面布局、色彩、字体都可以实现全局调控；

```
\documentclass{beamer}
\begin{document}
\begin{frame}
Hello Beamer!
\end{frame}%
\end{document}
```

从这个例子可以看出，Beamer 中每张幻灯片的内容都是放置在一个 frame 环境里面的。


```
\documentclass{beamer}
\usepackage[UTF8]{ctex}
\begin{document}
\begin{frame}
你好 Beamer!
\end{frame}
\end{document}
```

对于中文文档，建议用 UTF8 编码，然后用 xelatex 程序编译。另外，可以在载入 ctex 宏包时加上 noindent 选项以取消段落的缩进。

Beamer 基本语法

```
\begin{frame}{幻灯片标题}{我是一个副标题}  
Hello Beamer!  
\end{frame }
```

或者

```
\begin{frame}  
\frametitle{幻灯片标题}  
\framesubtitle{我是一个副标题}  
Hello Beamer!  
\end{frame }
```

Beamer 文档结构

在 Beamer 文档中，最常用的分节命令是 `\section`:

```
\section{Section Name}
```

类似于标题页面，我们可以在幻灯片中用 `\tableofcontents` 命令生成目录页。

```
\begin{frame}  
\tableofcontents[hideallsubsections]  
\end{frame }
```

其中 `hideallsubsections` 选项表示不显示小节标题。

简单的列表环境

```
\begin{itemize}
\item<1-> 我是第一项
\item<2-> 我是第二项
\item<3-> 我是第三项
\end{itemize}
```

- 我是第一项

简单的列表环境

```
\begin{itemize}
\item<1-> 我是第一项
\item<2-> 我是第二项
\item<3-> 我是第三项
\end{itemize}
```

- 我是第一项
- 我是第二项

简单的列表环境

```
\begin{itemize}  
\item<1-> 我是第一项  
\item<2-> 我是第二项  
\item<3-> 我是第三项  
\end{itemize}
```

- 我是第一项
- 我是第二项
- 我是第三项

简单的区块环境

```
\begin{block}{区块环境}
```

区块环境为了突出显示某些内容。

```
\end{block}
```

区块环境

区块环境为了突出显示某些内容。

警示区块环境

警示区块环境为了警示突出某些内容。

例子区块环境

例子区块环境为了突出例子内容。

beamer themes

Beamer 的整体主题包含了结构、颜色、字体各方面的设置。

```
\usebeamertheme{主题名}
```

无导航栏 无导航栏 default、boxes、Bergen、Pittsburgh 和 Rochester。

带顶栏 Antibes、Darmstadt、Frankfurt、JuanLesPins、Montpellier 和 Singapore。

带底栏 Boadilla 和 Madrid。

带顶栏底栏 AnnArbor、Berlin、CambridgeUS、Copenhagen、Dresden、Ilmenau、Luebeck、Malmoe、Szeged 和 Warsaw。

带侧栏 Berkeley、Goettingen、Hannover、Marburg 和 PaloAlto。

http://deic.uab.es/~iblanes/beamer_gallery/

<http://www.hartwork.org/beamer-theme-matrix/>

Beamer 的各部分的内容都可以自己定制和修改，和主题的划分类似，可以从如下这三个方面来定制自己的主题：

定制模板 用 `\setbeamertheme` 命令

定制颜色 用 `\setbeamercolor` 命令

定制字体 用 `\setbeamerfont` 命令

主要介绍两种方法：

主要介绍两种方法:

- ① 手工输入法

主要介绍两种方法：

- ① 手工输入法
- ② BibTeX

参考文献手工输入法

```
As is stated in \cite{bibitem1} \dots
```

```
\begin{thebibliography}{9}
```

```
\bibitem{bibitem1} 大傻瓜. 如何做一个合格的大傻瓜 [M]. 傻瓜帝国: 傻瓜出版社.2013.
```

```
\bibitem{bibitem2} 小傻瓜. 如何成为一个大傻瓜 [M]. 傻瓜帝国: 傻瓜出版社.2013.
```

```
\end{thebibliography}
```

As is stated in [1] ...

 大傻瓜. 如何做一个合格的大傻瓜 [M]. 傻瓜帝国: 傻瓜出版社.2013.

 小傻瓜. 如何成为一个大傻瓜 [M]. 傻瓜帝国: 傻瓜出版社.2013.

传说中的高级玩法 BIB_TE_X

BIB_TE_X 是一个使用数据库的的方式来管理参考文献程序, 用于协调 LaTeX 的参考文献处理.

```
@article{Gettys90,  
author = {Jim Gettys and Phil Karlton and Scott McGregor},  
title = {The {X} Window System, Version 11},  
journal = {Software Practice and Experience},  
volume = {20},  
year = {1990},  
abstract = {A technical overview of the X11 functionality.}  
}
```

BIB_TE_X 文件的后缀名为.bib

```
As is stated in \cite{fool} \dots
\bibliographystyle{plain}
\nocite{*}\bibliography{reference}
```

As is stated in [1] ...



Little fool.

How to become a fool.

Research on the fool, 2013.



Super fool.

How to become a right fool.

Research on the right fool, 2013.

L^AT_EX 中 B_IB_TE_X 的编译流程

X_YL^AT_EX \implies B_IB_TE_X \implies X_YL^AT_EX \implies X_YL^AT_EX

每步的解释:

L^AT_EX 中 B_IB_TE_X 的编译流程

X_YL^AT_EX \implies B_IB_TE_X \implies X_YL^AT_EX \implies X_YL^AT_EX

每步的解释:

- 1 用 X_YL^AT_EX 编译你的.tex 文件, 这是生成一个.aux 的文件, 这告诉 B_IB_TE_X 将使用那些引用

L^AT_EX 中 B_IB_TE_X 的编译流程

X_YL^AT_EX \implies B_IB_TE_X \implies X_YL^AT_EX \implies X_YL^AT_EX

每步的解释:

- 1 用 X_YL^AT_EX 编译你的.tex 文件, 这是生成一个.aux 的文件, 这告诉 B_IB_TE_X 将使用那些引用
- 2 用 B_IB_TE_X 编译.bib 文件, 后台将.bst 文件和.bib 文件编译成.bbl 文件

L^AT_EX 中 B_IB_TE_X 的编译流程

$X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} \implies \text{BIB}\text{T}_{\text{E}}\text{X} \implies X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} \implies X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

每步的解释:

- 1 用 $X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译你的.tex 文件, 这是生成一个.aux 的文件, 这告诉 B_IB_TE_X 将使用那些引用
- 2 用 B_IB_TE_X 编译.bib 文件, 后台将.bst 文件和.bib 文件编译成.bbl 文件
- 3 再次用 $X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译你的.tex 文件, 这个时候在文档中已经包含了参考文献, 但此时引用的编号可能不正确

L^AT_EX 中 BIB_TE_X 的编译流程

$X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} \implies \text{BIB}\text{T}_{\text{E}}\text{X} \implies X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} \implies X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

每步的解释:

- ① 用 $X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译你的.tex 文件, 这是生成一个.aux 的文件, 这告诉 BIB_TE_X 将使用那些引用
- ② 用 BIB_TE_X 编译.bib 文件, 后台将.bst 文件和.bib 文件编译成.bbl 文件
- ③ 再次用 $X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译你的.tex 文件, 这个时候在文档中已经包含了参考文献, 但此时引用的编号可能不正确
- ④ 最后用 $X_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译你的.tex 文件, 如果一切顺利的话, 这是所有东西都已正常了

BIBTEX 管理辅助软件

如果要管理大量参考文献，就需要接下来讲的 BIBTEX 管理辅助软件了，这里我们主要介绍 JabRef:

如果要管理大量参考文献，就需要接下来讲的 BIBTEX 管理辅助软件了，这里我们主要介绍 JabRef:

- ① 自动导入。支持 CiteSeer、JSTOR、SPIRES、IEEEXplore、ArXiv.org、ACM Portal、Medline 以及 ScienceDirect 八大电子资源数据库的文献查找和索引自动导入功能。

如果要管理大量参考文献，就需要接下来讲的 BIBTEX 管理辅助软件了，这里我们主要介绍 JabRef:

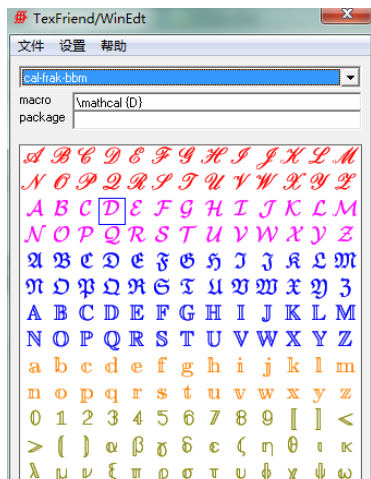
- ① 自动导入。支持 CiteSeer、JSTOR、SPIRES、IEEEXplore、ArXiv.org、ACM Portal、Medline 以及 ScienceDirect 八大电子资源数据库的文献查找和索引自动导入功能。
- ② 转换方便。支持不同文献索引格式文件的导入和导出。可以广泛读取其他文献管理工具，如 EndNote、Reference Manager、Refworks 等保存的文献索引格式。

如果要管理大量参考文献，就需要接下来讲的 BIBTEX 管理辅助软件了，这里我们主要介绍 JabRef:

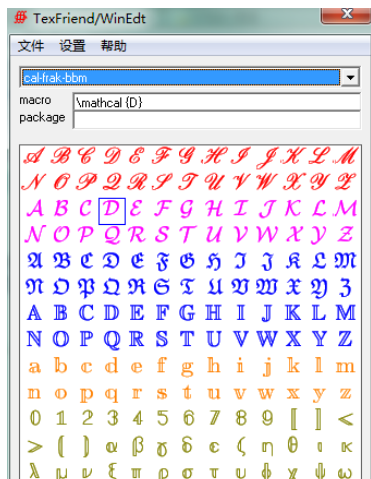
- ① 自动导入。支持 CiteSeer、JSTOR、SPIRES、IEEEXplore、ArXiv.org、ACM Portal、Medline 以及 ScienceDirect 八大电子资源数据库的文献查找和索引自动导入功能。
- ② 转换方便。支持不同文献索引格式文件的导入和导出。可以广泛读取其他文献管理工具，如 EndNote、Reference Manager、Refworks 等保存的文献索引格式。
- ③ 自动分类。支持任意分类和自动分类，可自动根据题目、作者、关键词或摘要自动分类。

如果要管理大量参考文献，就需要接下来讲的 BIBTEX 管理辅助软件了，这里我们主要介绍 JabRef:

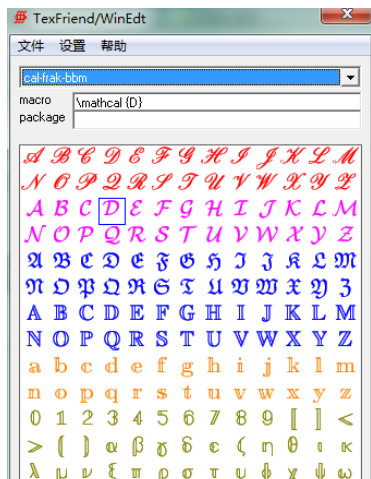
- ① 自动导入。支持 CiteSeer、JSTOR、SPIRES、IEEEXplore、ArXiv.org、ACM Portal、Medline 以及 ScienceDirect 八大电子资源数据库的文献查找和索引自动导入功能。
- ② 转换方便。支持不同文献索引格式文件的导入和导出。可以广泛读取其他文献管理工具，如 EndNote、Reference Manager、Refworks 等保存的文献索引格式。
- ③ 自动分类。支持任意分类和自动分类，可自动根据题目、作者、关键词或摘要自动分类。
- ④ 兼容性强。支持在各种 LaTeX 编辑器中和很多文本编辑器中插入文献记录，可以推送文献索引至写作用档。



- 1 南开大学孙文昌开发，WinEdt 自带。



- 1 南开大学孙文昌开发，WinEdt 自带。
- 2 TeXFrined 提供几千个 LaTeX 字符的自动输入。





<http://wiki.lyx.org/uploads/LyX/Screencasts/LyXIntroPalette.htm>



- ① LyX 是一个“所见即所指”（what you see is what you mean）的文件编辑软件。

<http://wiki.lyx.org/uploads/LyX/Screencasts/LyXIntroPalette.htm>



- ① LyX 是一个“所见即所指”（what you see is what you mean）的文件编辑软件。
- ② LyX 利用 \LaTeX 来排版。

<http://wiki.lyx.org/uploads/LyX/Screencasts/LyXIntroPalette.htm>



- ① LyX 是一个“所见即所指”（what you see is what you mean）的文件编辑软件。
- ② LyX 利用 \LaTeX 来排版。
- ③ 通过 CJK-LyX 提供中文支持。

<http://wiki.lyx.org/uploads/LyX/Screencasts/LyXIntroPalette.htm>



- ① LyX 是一个“所见即所指”（what you see is what you mean）的文件编辑软件。
- ② LyX 利用 \LaTeX 来排版。
- ③ 通过 CJK-LyX 提供中文支持。
- ④ 可以生成 \LaTeX 文件和 PostScript 文件。

<http://wiki.lyx.org/uploads/LyX/Screencasts/LyXIntroPalette.htm>



- 1 在 Word 中编辑文章



Word2TeX

- ① 在 Word 中编辑文章
- ② 安装 Word2TeX 软件



- ① 在 Word 中编辑文章
- ② 安装 Word2TeX 软件
- ③ 将 Word 文档转化为 TeX 文件



- ① 在 Word 中编辑文章
- ② 安装 Word2TeX 软件
- ③ 将 Word 文档转化为 TeX 文件
- ④ 对生成的 TeX 文件修改



- ① 在 Word 中编辑文章
- ② 安装 Word2TeX 软件
- ③ 将 Word 文档转化为 TeX 文件
- ④ 对生成的 TeX 文件修改
- ⑤ 再用 LaTeX 编译为 PDF 文档



