

初学 L^AT_EX

王旭

Wireless Tech Innovation (WTI) Institute,
Beijing University of Posts and Telecommunications (BUPT)

Slides are manipulated by L^AT_EX 2_ε & Beamer.



"Distributed under Creative Commons Lincens: Some Right Reserved, BY-NC-ND"

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料



大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料



大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料



大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料



大纲

- 1 L^AT_EX 导引
 - L^AT_EX 是什么
 - L^AT_EX 的特点
 - 安装 L^AT_EX
 - L^AT_EX 是如何工作的
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个排版软件,就像 MS Word
- 一个可编程、可扩展的工具,富于精确控制能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

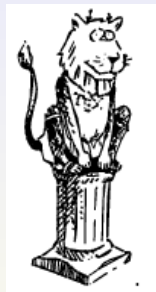


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个排版软件,就像 MS Word
- 一个可编程、可扩展的工具,富于精确控制能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

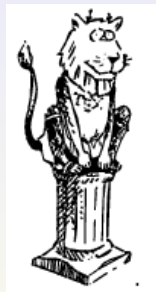


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个可编程、可扩展的工具,富于精确控制能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

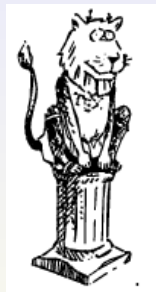


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个可编程、可扩展的工具,富于精确控制能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

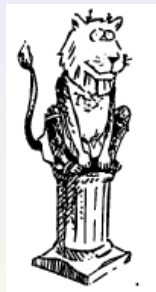


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于精确控制能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

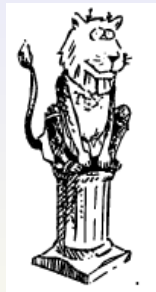


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制能力**
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

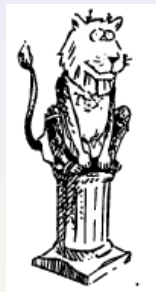


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是科学论文的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

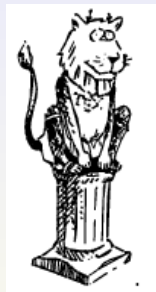


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是**科学论文**的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

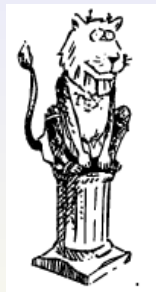


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是**科学论文**的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

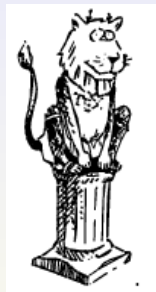


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是**科学论文**的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所见即所得,而是所想即所得的工具

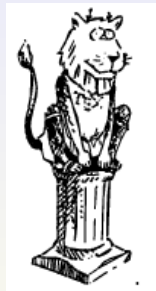


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是**科学论文**的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所**见**即所得,而是所想即所得的工具

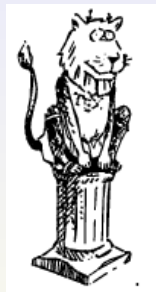


FIGURE:
T_EX

什么是 T_EX

- 右边这位狮子先生
- 一个字处理软件,或说是一个**排版软件**,就像 MS Word
- 一个**可编程**、可扩展的工具,富于**精确控制**能力
- 一个擅长逻辑结构严密的文章、特别是**科学论文**的排版工具
- 一个真正的跨平台工具,不论你在什么系统上都能使用
- 不是所**见**即所得,而是所**想**即所得的工具

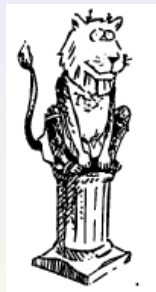


FIGURE:
T_EX

第一关:如何发音

T_EX

/tekʰ/, 关键点: X 是希腊字母 X (chi), 绝对不能读成 /ks/, 如果觉得不好发音, 可以读成 /k/。

L^AT_EX

/'lei'tekʰ/ 或 /'la:'tekʰ/。

第一关:如何发音

T_EX

/tekʰ/, 关键点: X 是希腊字母 X (chi), **绝对不能**读成 /ks/, 如果觉得不好发音, 可以读成 /k/。

L^AT_EX

/'lei'tekʰ/ 或 /'la:'tekʰ/。

第一关:如何发音

T_EX

/tekʰ/, 关键点: X 是希腊字母 X (chi), **绝对不能**读成 /ks/, 如果觉得不好发音, 可以读成 /k/。

L^AT_EX

/'lei'tekʰ/ 或 /'lɑ:'tekʰ/。

T_EX 之父



FIGURE: T_EX 的缔造者: Knuth

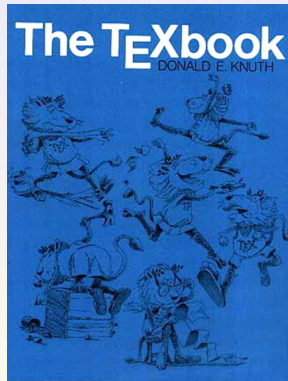


FIGURE: T_EXBook

这些是什么？

T_EX 的马甲

Plain T_EX, L^AT_EX, ConT_EXt

穿马甲的 T_EX

TeXLive, CTeX

T_EX 的徒弟

OMEGA...

以后我们的话题就要围绕 L^AT_EX 进行了

这些是什么？

T_EX 的马甲

Plain T_EX, L^AT_EX, ConT_EXt

穿马甲的 T_EX

TeXLive, CTeX

T_EX 的徒弟

OMEGA...

以后我们的话题就要围绕 L^AT_EX 进行了

这些是什么？

T_EX 的马甲

Plain T_EX, L^AT_EX, ConT_EXt

穿马甲的 T_EX

TeXLive, CTeX

T_EX 的徒弟

OMEGA...

以后我们的话题就要围绕 L^AT_EX 进行了

这些是什么？

T_EX 的马甲

Plain T_EX, L^AT_EX, ConT_EXt

穿马甲的 T_EX

TeXLive, CTeX

T_EX 的徒弟

OMEGA...

以后我们的话题就要围绕 L^AT_EX 进行了

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- 漂亮而方便的数学公式编辑
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- 漂亮而方便的数学公式编辑
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- **源文件的编辑更加高效**
- 漂亮而方便的数学公式编辑
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- **漂亮而方便的数学公式编辑**
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- 漂亮而方便的数学公式编辑
- **便捷的交叉索引和参考文献编排**
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- 漂亮而方便的数学公式编辑
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

为什么使用 (不用) L^AT_EX

- 使用简便、样式和内容分开,“所想即所得”。
- 印刷级的优质排版,特别是英文
- 源文件的编辑更加高效
- 漂亮而方便的数学公式编辑
- 便捷的交叉索引和参考文献编排
- 众多的用户和大量成熟好用的模板
- 完美的 UNIX 哲学

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt...
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt...
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 —要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt...
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 —要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt...
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt. . .
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt. . .
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt. . .
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt. . .
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

一个完整的 T_EX 系统需要具备:

- T_EX 执行程序
- 多种样式,对于我们来说,需要的是 L^AT_EX 2_ε, 实际上还有 plain, ConT_EXt. . .
- 大量宏包,比如,要包含图形就要用 graphicx 宏包。
- 需要的字体 — 要排出漂亮的文档,当然要有漂亮的字体。
- 辅助工具:
 - MetaFont&MetaPost 是很多精美绘图的缔造者
 - 维护工具最能看出套装的独到之处
 - 当然也有些套装中甚至提供源文件编辑器。

T_EX 套装

常见的 T_EX 套装

- CT_EX, 中文 Windows 用户最常用的套装。
CT_EX 网站提供的一套中文 T_EX 套装, 对 GBK 中文支持的更好, 可以利用 Windows 系统中的标准中文字体, 随软件包附带了 WinEdt 编辑器构成一个集成开发环境。
- T_EXLive, 最完整的多平台套装。
CTAN 提供的全平台的 T_EX 套装, 实际包含了几个服务于不同平台的核心执行程序以及大量的宏包, 每年发布更新。
- teT_EX, Linux 用户最常用的套装。
Thomas Esser 维护的 UNIX 类操作系统使用的 T_EX 套装, 几乎每个 Linux 发布版都会包含它, 内容非常丰富, 您看到的这个幻灯片就是 teT_EX 的杰作。
- fpT_EX, MikT_EX...

T_EX 套装

常见的 T_EX 套装

- CT_EX, 中文 Windows 用户最常用的套装。
CT_EX 网站提供的一套中文 T_EX 套装, 对 GBK 中文支持的更好, 可以利用 Windows 系统中的标准中文字体, 随软件包附带了 WinEdt 编辑器构成一个集成开发环境。
- T_EXLive, 最完整的多平台套装。
CTAN 提供的全平台的 T_EX 套装, 实际包含了几个服务于不同平台的核心执行程序以及大量的宏包, 每年发布更新。
- teT_EX, Linux 用户最常用的套装。
Thomas Esser 维护的 UNIX 类操作系统使用的 T_EX 套装, 几乎每个 Linux 发布版都会包含它, 内容非常丰富, 您看到的这个幻灯片就是 teT_EX 的杰作。
- fpT_EX, MikT_EX...

T_EX 套装

常见的 T_EX 套装

- CT_EX, 中文 Windows 用户最常用的套装。
CT_EX 网站提供的一套中文 T_EX 套装, 对 GBK 中文支持的更好, 可以利用 Windows 系统中的标准中文字体, 随软件包附带了 WinEdt 编辑器构成一个集成开发环境。
- T_EXLive, 最完整的多平台套装。
CTAN 提供的全平台的 T_EX 套装, 实际包含了几个服务于不同平台的核心执行程序以及大量的宏包, 每年发布更新。
- teT_EX, Linux 用户最常用的套装。
Thomas Esser 维护的 UNIX 类操作系统使用的 T_EX 套装, 几乎每个 Linux 发布版都会包含它, 内容非常丰富, 您看到的这个幻灯片就是 teT_EX 的杰作。
- fpT_EX, MikT_EX...

T_EX 套装

常见的 T_EX 套装

- CT_EX, 中文 Windows 用户最常用的套装。
CT_EX 网站提供的一套中文 T_EX 套装, 对 GBK 中文支持的更好, 可以利用 Windows 系统中的标准中文字体, 随软件包附带了 WinEdt 编辑器构成一个集成开发环境。
- T_EXLive, 最完整的多平台套装。
CTAN 提供的全平台的 T_EX 套装, 实际包含了几个服务于不同平台的核心执行程序以及大量的宏包, 每年发布更新。
- teT_EX, Linux 用户最常用的套装。
Thomas Esser 维护的 UNIX 类操作系统使用的 T_EX 套装, 几乎每个 Linux 发布版都会包含它, 内容非常丰富, 您看到的这个幻灯片就是 teT_EX 的杰作。
- fpT_EX, MikT_EX...

L^AT_EX 工作机制

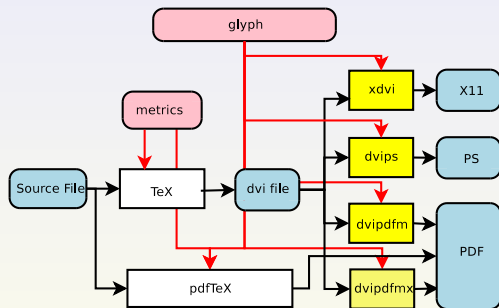


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

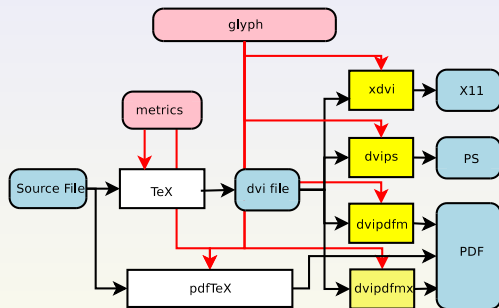


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

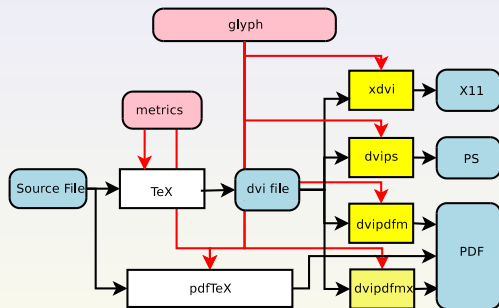


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

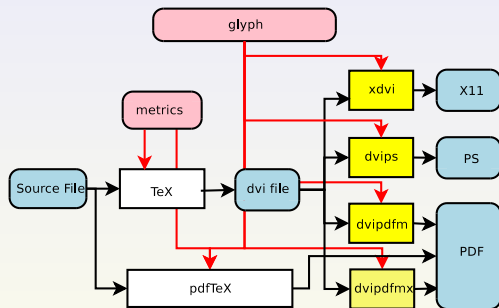


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

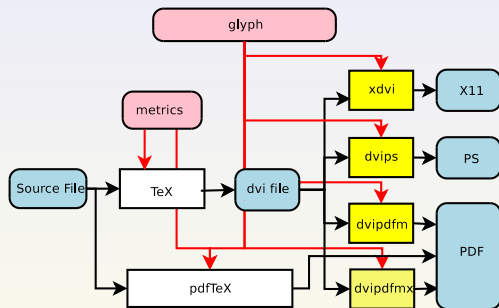


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

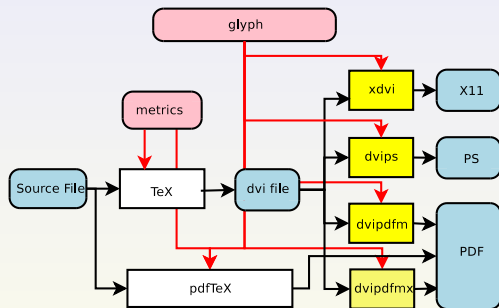


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 工作机制

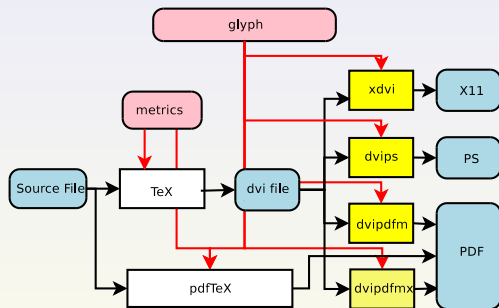


FIGURE: T_EX 的工作流程

- 1 载入源文件
- 2 读入需要的字体维度信息 (盒子)
- 3 根据宏指令进行排版, 得到排版信息
- 4 输出为 DVI。
- 5 载入字形信息 (具体的字)
- 6 最终输出。

L^AT_EX 的源文件

- 选择你最喜欢的编辑器,写入一段源文件,这是我们的第一个 L^AT_EX 文件:

```
\documentclass{article}
\title{First \LaTeX Document}
\author{Wang Xu}
\begin{document}

\maketitle

Hello, \LaTeX!

\end{document}
```

制作输出文件

- 可以这样:

```
latex first.tex⇒first.dvi  
dvips first.dvi⇒first.ps  
pspdf first.ps ⇒first.pdf
```

- 也可以这样:

```
latex first.tex ⇒first.dvi  
dvipdfm first.dvi⇒first.pdf
```

- 还可以直接这样:

```
pdflatex first.tex⇒first.pdf
```

制作输出文件

- 可以这样:

```
latex first.tex⇒first.dvi  
dvips first.dvi⇒first.ps  
pspdf first.ps ⇒first.pdf
```

- 也可以这样:

```
latex first.tex ⇒first.dvi  
dvipdfm first.dvi⇒first.pdf
```

- 还可以直接这样:

```
pdflatex first.tex⇒first.pdf
```

制作输出文件

- 可以这样:

```
latex first.tex⇒first.dvi  
dvips first.dvi⇒first.ps  
pspdf first.ps ⇒first.pdf
```

- 也可以这样:

```
latex first.tex ⇒first.dvi  
dvipdfm first.dvi⇒first.pdf
```

- 还可以直接这样:

```
pdflatex first.tex⇒first.pdf
```

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始,不过,就算是没选好的话,将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要,对需要强调的内容加以强调;
- 高手的话,可能做更多格式的设置;
- 生成输出文档,适当调整格式,直到完成目标为止。

用 L^AT_EX 写文章的方式

- 选好要用的模板开始, 不过, 就算是没选好的话, 将来换也可以;
- 按照章节组织好文章;
- 插入公式、表格、图形...
- 在需要被引用的地方加上标记 (label), 在引用的地方引用标记;
- 添加参考文献和参考文献的引用 (随用随加也是常见的);
- 根据行文的需要, 对需要强调的内容加以强调;
- 高手的话, 可能做更多格式的设置;
- 生成输出文档, 适当调整格式, 直到完成目标为止。

大纲

1 L^AT_EX 导引

2 L^AT_EX 的基本使用

- 源文件的基本要素: 文本、命令和注释
- 特殊字符的输入与逐字显示
- 列表与枚举

3 书写中文

4 插入公式、图形、表格以及程序代码

5 参考文献

6 更多技巧与常用宏包

7 参考文献与网上资料

源文件中的要素

L^AT_EX 的源文件,尤其是开头,更接近于程序而不是文章或书,文章中有一些基本要素

- 注释: 以 % 开头的是注释。这一行后面的内容都不会影响最终输出结果。
- 以 \ 开头的词是命令,更确切地说是需要 L^AT_EX 特殊处理的内容。
 - 命令如

<code>\LaTeX</code>	<code>\textit{italic}</code>	<code>\rule[.5em]{5em}{1pt}</code>
L ^A T _E X	<i>italic</i>	<hr/>

- 还有中间可以加入大量内容的环境。如:

```
1 \begin{quote}
2 We use ``quote'' environment for words from others.
3 \end{quote}
```

- 除此之外的内容就是要被排版的文字了。

源文件中的要素

L^AT_EX 的源文件,尤其是开头,更接近于程序而不是文章或书,文章中有一些基本要素

- 注释: 以 % 开头的是注释。这一行后面的内容都不会影响最终输出结果。
- 以 \ 开头的词是命令,更确切地说是需要 L^AT_EX 特殊处理的内容。
 - 命令如

<code>\LaTeX</code>	<code>\textit{italic}</code>	<code>\rule[.5em]{5em}{1pt}</code>
L ^A T _E X	<i>italic</i>	<hr/>

- 还有中间可以加入大量内容的环境。如:

```
1 \begin{quote}
2 We use ``quote'' environment for words from others.
3 \end{quote}
```

- 除此之外的内容就是要被排版的文字了。

源文件中的要素

L^AT_EX 的源文件,尤其是开头,更接近于程序而不是文章或书,文章中有一些基本要素

- 注释: 以 % 开头的是注释。这一行后面的内容都不会影响最终输出结果。
- 以 \ 开头的词是命令,更确切地说是需要 L^AT_EX 特殊处理的内容。
 - 命令如

<code>\LaTeX</code>	<code>\textit{italic}</code>	<code>\rule[.5em]{5em}{1pt}</code>
L ^A T _E X	<i>italic</i>	<hr/>

- 还有中间可以加入大量内容的环境。如:

```
1 \begin{quote}
2 We use ``quote'' environment for words from others.
3 \end{quote}
```

- 除此之外的内容就是要被排版的文字了。

源文件中的要素

L^AT_EX 的源文件,尤其是开头,更接近于程序而不是文章或书,文章中有一些基本要素

- 注释: 以 % 开头的是注释。这一行后面的内容都不会影响最终输出结果。
- 以 \ 开头的词是命令,更确切地说是需要 L^AT_EX 特殊处理的内容。
 - 命令如

<code>\LaTeX</code>	<code>\textit{italic}</code>	<code>\rule[.5em]{5em}{1pt}</code>
L ^A T _E X	<i>italic</i>	<hr/>

- 还有中间可以加入大量内容的环境。如:

```
1 \begin{quote}
2 We use ``quote'' environment for words from others.
3 \end{quote}
```

- 除此之外的内容就是要被排版的文字了。

源文件中的要素

L^AT_EX 的源文件,尤其是开头,更接近于程序而不是文章或书,文章中有一些基本要素

- 注释: 以 % 开头的是注释。这一行后面的内容都不会影响最终输出结果。
- 以 \ 开头的词是命令,更确切地说是需要 L^AT_EX 特殊处理的内容。
 - 命令如

<code>\LaTeX</code>	<code>\textit{italic}</code>	<code>\rule[.5em]{5em}{1pt}</code>
L ^A T _E X	<i>italic</i>	<hr/>

- 还有中间可以加入大量内容的环境。如:

```
1 \begin{quote}
2 We use ``quote'' environment for words from others.
3 \end{quote}
```

- 除此之外的内容就是要被排版的文字了。

文字与空白

多于一个的空格以及换行都会被当作一个空格,而连续的两个换行 (也就是一个空行) 则会被当作是分段。

示例

```
This is an          example for illustrate the organ%  
ization of  
a \LaTeXe file.
```

```
You can learn many commands and environments of it in  
this example.
```

相应的结果

This is an example for illustrate the organization of a L^AT_EX 2_ε file.
You can learn many commands and environments of it in this example.

文字与空白

多于一个的空格以及换行都会被当作一个空格,而连续的两个换行 (也就是一个空行) 则会被当作是分段。

示例

```
This is an           example for illustrate the organ%  
ization of  
a \LaTeXe file.
```

```
You can learn many commands and environments of it in  
this example.
```

相应的结果

This is an example for illustrate the organization of a L^AT_EX 2_ε file.
You can learn many commands and environments of it in this example.

文字与空白

多于一个的空格以及换行都会被当作一个空格,而连续的两个换行 (也就是一个空行) 则会被当作是分段。

示例

```
This is an           example for illustrate the organ%  
ization of  
a \LaTeXe file.
```

```
You can learn many commands and environments of it in  
this example.
```

相应的结果

This is an example for illustrate the organization of a L^AT_EX 2_ε file.
You can learn many commands and environments of it in this example.

源文件的结构

引言

```
\documentclass[a4paper,12pt]{article}
\usepackage{fancyhdr,listings,graphicx}
\usepackage{CJK,CJKnumb}
\usepackage[a4paper,%
             includehead,%
             includefoot,%
             top=2cm,%
             bottom=2cm,%
```

前导部分的功能就是引入宏包、定义命令...

源文件的结构

引言

```
\documentclass[a4paper,12pt]{article}
\usepackage{fancyhdr,listings,graphicx}
\usepackage{CJK,CJKnumb}
\usepackage[a4paper,%
             includehead,%
             includefoot,%
             top=2cm,%
             bottom=2cm,%
```

前导部分的功能就是引入宏包、定义命令...

源文件的结构

正文

```
\begin{document}
```

这里就是正文部分了...

```
\end{document}
```

另外

一篇 L^AT_EX 文章可以写在不同的源文件里, 通过 `\input{filename}` 或 `\include{filename}` 命令载入, 对于保持源文件的清晰和美观有很大好处。

源文件的结构

正文

```
\begin{document}
```

这里就是正文部分了...

```
\end{document}
```

另外

一篇 L^AT_EX 文章可以写在不同的源文件里, 通过 `\input{filename}` 或 `\include{filename}` 命令载入, 对于保持源文件的清晰和美观有很大好处。

文章的标题

文章的标题部分包括文章的标题、作者和日期:

首先是文章的题目:

```
\title{Hello, \LaTeX!\\  
My First \LaTeX Work}
```

这是整篇文章的题目,如果需要对题目分行,可以使用双反斜线。

文章的标题

文章的标题部分包括文章的标题、作者和日期:

首先是文章的题目:

```
\title{Hello, \LaTeX!\\  
My First \LaTeX Work}
```

这是整篇文章的题目,如果需要通过让题目分行,可以使用双反斜线。

文章的标题

文章的标题部分包括文章的标题、作者和日期:

作者列表:

```
\author{Wang Xu\\  
Beijing Univ. of Posts and Telecomm.\\  
gnawux@gmail.com  
&  
Mr. Sleepy\\  
Who knows where he come from\\  
nobody@anywhere.net%  
}
```

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 **ARTICLE** 文档类

- 有一个特殊的 **abstract** 环境, 作为摘要;
- 文章的不同大纲级别包括
section, subsection, subsubsection
- 对于超长的章节名称, 可以有个短名字出现在页眉和目录中:

```
1 \section[short name]%  
2 {section has a long name}
```

- 带有 * 的章节没有编号。

```
1 \section*{asterisk means \ldots}
```

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 **ARTICLE** 文档类

- 有一个特殊的 **abstract** 环境, 作为摘要;
- 文章的不同大纲级别包括
section, subsection, subsubsection
- 对于超长的章节名称, 可以有个短名字出现在页眉和目录中:

```
1 \section[short name]%  
2 {section has a long name}
```

- 带有 * 的章节没有编号。

```
1 \section*{asterisk means \ldots}
```

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 **ARTICLE** 文档类

- 有一个特殊的 **abstract** 环境, 作为摘要;
- 文章的不同大纲级别包括
section, subsection, subsubsection
- 对于超长的章节名称, 可以有个短名字出现在页眉和目录中:

```
1 \section[short name]%  
2 {section has a long name}
```

- 带有 * 的章节没有编号。

```
1 \section*{asterisk means \ldots}
```

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 **ARTICLE** 文档类

- 有一个特殊的 **abstract** 环境, 作为摘要;
- 文章的不同大纲级别包括
section, subsection, subsubsection
- 对于超长的章节名称, 可以有个短名字出现在页眉和目录中:

```
1 \section[short name]%  
2 {section has a long name}
```

- 带有 * 的章节没有编号。

```
1 \section*{asterisk means \ldots}
```


章节与段落

L^AT_EX 按照章节组织文章的内容

对于 **ARTICLE** 文档类

- 有一个特殊的 **abstract** 环境, 作为摘要;
- 文章的不同大纲级别包括
section, subsection, subsubsection
- 对于超长的章节名称, 可以有个短名字出现在页眉和目录中:

```
1 \section[short name]%  
2 {section has a long name}
```

- 带有 * 的章节没有编号。

```
1 \section*{asterisk means \ldots}
```

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

章节与段落

L^AT_EX 按照章节组织文章的内容

对于 BOOK 文档类

- `\maketitle` 命令会产生一个扉页,而不是 `article` 那样的标题栏。
- 多出了两个更高的大纲级别:
part, chapter
- 缺省情况下,每一章都从右边的页开始,可以通过 `openany` 参数改变。
- 目录 (`\tableofcontents`) 等内容会成为一章,但没有编号
- 书的内容可以分为三个部分:
frontmatter, mainmatter & appendix

特殊文本元素

有些特殊的文本元素是由程序生成的,用于辅助文章的理解

- 标题栏或扉页, 章节, 已经介绍过了。
- 脚注

```
1 This sentence have  
2 footnote\footnote{blah blah\ldots}.
```

- 标记

```
1 \section{Introduction}\label{sec:intro}
```

- 引用标记

```
1 We have mentioned in \S~\ref{sec:intro} on  
2 pp.~\pageref{sec:intro}\ldots
```


特殊文本元素

有些特殊的文本元素是由程序生成的,用于辅助文章的理解

- 标题栏或扉页, 章节, 已经介绍过了。

- 脚注

```
1 This sentence have  
2 footnote\footnote{blah blah\ldots}.
```

- 标记

```
1 \section{Introduction}\label{sec:intro}
```

- 引用标记

```
1 We have mentioned in \S~\ref{sec:intro} on  
2 pp.~\pageref{sec:intro}\ldots
```

特殊文本元素

有些特殊的文本元素是由程序生成的,用于辅助文章的理解

- 标题栏或扉页, 章节, 已经介绍过了。

- 脚注

```
1 This sentence have  
2 footnote\footnote{blah blah\ldots}.
```

- 标记

```
1 \section{Introduction}\label{sec:intro}
```

- 引用标记

```
1 We have mentioned in \S~\ref{sec:intro} on  
2 pp.~\pageref{sec:intro}\ldots
```

特殊文本元素

有些特殊的文本元素是由程序生成的,用于辅助文章的理解

- 标题栏或扉页, 章节, 已经介绍过了。
- 脚注

```
1 This sentence have  
2 footnote\footnote{blah blah\ldots}.
```

- 标记

```
1 \section{Introduction}\label{sec:intro}
```

- 引用标记

```
1 We have mentioned in \S~\ref{sec:intro} on  
2 pp.~\pageref{sec:intro}\ldots
```

特殊文本元素

有些特殊的文本元素是由程序生成的,用于辅助文章的理解

- 标题栏或扉页, 章节, 已经介绍过了。
- 脚注

```
1 This sentence have  
2 footnote\footnote{blah blah\ldots}.
```

- 标记

```
1 \section{Introduction}\label{sec:intro}
```

- 引用标记

```
1 We have mentioned in \S~\ref{sec:intro} on  
2 pp.~\pageref{sec:intro}\ldots
```

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

● 引号

```
1 we use ``quote sign'' like this. So pretty.
```

● 横线

```
1 one means hyphe-nation\\  
2 two is a -- short bar\\  
3 three --- oh yeah\\
```

● 波浪符

```
1 Tilde(\~{ }) can generate~nbsp
```

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

● 引号

```
1 we use ``quote sign'' like this. So pretty.
```

● 横线

```
1 one means hyphe-nation\\
2 two is a -- short bar\\
3 three --- oh yeah\\
```

● 波浪符

```
1 Tilde(\~{ }) can generate~nbsp
```

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

● 引号

```
1 we use ``quote sign'' like this. So pretty.
```

● 横线

```
1 one means hyphe-nation\\  
2 two is a -- short bar\\  
3 three --- oh yeah\\
```

● 波浪符

```
1 Tilde(\~{ }) can generate~nbsp
```

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

● 引号

```
1 we use ``quote sign'' like this. So pretty.
```

● 横线

```
1 one means hyphe-nation\\  
2 two is a -- short bar\\  
3 three --- oh yeah\\
```

● 波浪符

```
1 Tilde(\~{ }) can generate~nbsp
```

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

显示效果

- 引号

we use "quote sign" like this. So pretty.

- 横线

one means hyphe-nation

two is a – short bar

three — oh yeah

- 波浪符

Tilde(˘) can generate nbsp

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

显示效果

- 引号

we use “quote sign” like this. So pretty.

- 横线

one means hyphenation

two is a – short bar

three — oh yeah

- 波浪符

Tilde(˜) can generate nbsp

¹参考 \short.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

显示效果

- 引号

we use “quote sign” like this. So pretty.

- 横线

one means hyphenation

two is a – short bar

three — oh yeah

- 波浪符

Tilde(~) can generate nbsp

¹参考 lshort.

特殊字符

我们在上面已经看到很多特殊字符了,这里要介绍几个最常用的¹:

显示效果

- 引号

we use “quote sign” like this. So pretty.

- 横线

one means hyphenation

two is a – short bar

three — oh yeah

- 波浪符

Tilde(~) can generate nbsp

¹参考 lshort.

逐字显示

有的时候,我们需要刻意避免 L^AT_EX 将我们输入的字符当作是特殊指令。

- 单行的逐字显示

```
1 \verb|%$#@~\  oh yeah/ --- all are verb
```

- 整段的显示

```
1 \begin{verbatim}  
2 here,  
3 #@%$  
4 verbatims\ldots  
5 \end{verbatim}
```

- 更复杂的逐字显示: listings, fancyverb...

逐字显示

有的时候,我们需要刻意避免 L^AT_EX 将我们输入的字符当作是特殊指令。

- 单行的逐字显示

```
1 \verb|%%$#@~\  oh yeah/ --- all are verb
```

- 整段的显示

```
1 \begin{verbatim}  
2 here,  
3 #@%$  
4 verbatims\ldots  
5 \end{verbatim}
```

- 更复杂的逐字显示: listings, fancyverb...

逐字显示

有的时候,我们需要刻意避免 L^AT_EX 将我们输入的字符当作是特殊指令。

- 单行的逐字显示

```
1 \verb|%$#@~\ oh yeah/ --- all are verb
```

- 整段的显示

```
1 \begin{verbatim}  
2 here,  
3 #@$  
4 verbatims\ldots  
5 \end{verbatim}
```

- 更复杂的逐字显示: listings, fancyverb...

调整显示效果

为了表示不同的内容,我们时常会指定不同的字体族

- 强调命令与下划线

```
1 \emph{emphasis} and \underline{underline}
```

- 粗体

```
1 \textbf{bold face}
```

- 斜体

```
1 \textit{italic}
```

- 等线字体

```
1 \texttt{typewriter}
```


调整显示效果

为了表示不同的内容,我们时常会指定不同的字体族

- 强调命令与下划线

```
1 \emph{emphasis} and \underline{underline}
```

- 粗体

```
1 \textbf{bold face}
```

- 斜体

```
1 \textit{italic}
```

- 等线字体

```
1 \texttt{typewriter}
```

调整显示效果

为了表示不同的内容,我们时常会指定不同的字体族

- 强调命令与下划线

```
1 \emph{emphasis} and \underline{underline}
```

- 粗体

```
1 \textbf{bold face}
```

- 斜体

```
1 \textit{italic}
```

- 等线字体

```
1 \texttt{typewriter}
```

调整显示效果

为了表示不同的内容,我们时常会指定不同的字体族

- 强调命令与下划线

```
1 \emph{emphasis} and \underline{underline}
```

- 粗体

```
1 \textbf{bold face}
```

- 斜体

```
1 \textit{italic}
```

- 等线字体

```
1 \texttt{typewriter}
```

调整显示效果

为了表示不同的内容,我们时常会指定不同的字体族

- 强调命令与下划线

```
1 \emph{emphasis} and \underline{underline}
```

- 粗体

```
1 \textbf{bold face}
```

- 斜体

```
1 \textit{italic}
```

- 等线字体

```
1 \texttt{typewriter}
```

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{itemize}
    \item ooollllwww
    \item tttoiirrr
    \item nnnmmmeee
\end{itemize}
```

效果

- ooollllwww
- tttoiirrr
- nnnmmmeee

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{itemize}
  \item ooollllwww
  \item ttтииirrr
  \item nnnmmmeee
\end{itemize}
```

效果

- ooollllwww
- ttтииirrr
- nnnmmmeee

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{enumerate}  
    \item first one  
    \item another one  
    \item yet another  
\end{enumerate}
```

效果

- 1 first one
- 2 another one
- 3 yet another

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{enumerate}  
    \item first one  
    \item another one  
    \item yet another  
\end{enumerate}
```

效果

- ① first one
- ② another one
- ③ yet another

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{description}
  \item[itemize] bullet
  \item[enumerate] number
  \item[description] the term
\end{description}
```

效果

ITEMIZE bullet
ENUMERATE number
DESCRIPTION the term

列表与枚举

将要说明的内容分点列出,是最常用的说明手法之一。

示例

```
\begin{description}
  \item[itemize] bullet
  \item[enumerate] number
  \item[description] the term
\end{description}
```

效果

ITEMIZE bullet

ENUMERATE number

DESCRIPTION the term

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文**
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

$\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持

- $\text{T}_{\text{E}}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_{\text{E}}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_{\text{E}}\text{X}$ 也可以处理中文,如 Con $\text{T}_{\text{E}}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 应用广泛。
- 早期将 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 中使用中文。
- 还有 C $\text{T}_{\text{E}}\text{X}$ 为 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持作出了巨大贡献。

$\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持

- $\text{T}_{\text{E}}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_{\text{E}}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_{\text{E}}\text{X}$ 也可以处理中文,如 Con $\text{T}_{\text{E}}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 应用广泛。
- 早期将 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 中使用中文。
- 还有 C $\text{T}_{\text{E}}\text{X}$ 为 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持作出了巨大贡献。

$\text{T}_\text{E}\text{X}/\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持

- $\text{T}_\text{E}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_\text{E}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_\text{E}\text{X}$ 也可以处理中文,如 Con $\text{T}_\text{E}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 应用广泛。
- 早期将 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 中使用中文。
- 还有 C $\text{T}_\text{E}\text{X}$ 为 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持作出了巨大贡献。

$\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持

- $\text{T}_{\text{E}}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_{\text{E}}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_{\text{E}}\text{X}$ 也可以处理中文,如 Con $\text{T}_{\text{E}}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 应用广泛。
- 早期将 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 中使用中文。
- 还有 C $\text{T}_{\text{E}}\text{X}$ 为 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的中文支持作出了巨大贡献。

T_EX/L^AT_EX 的中文支持

- T_EX 本身并不支持中文编码,无法处理中文。
- T_EX 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理,T_EX 也可以处理中文,如 ConT_EXt 可以处理中文。
- 但这些都不及 L^AT_EX 应用广泛。
- 早期将 L^AT_EX 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 L^AT_EX 处理方法进行了改动,和大量的 L^AT_EX 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 L^AT_EX 中使用中文。
- 还有 CTeX 为 L^AT_EX 的中文支持作出了巨大贡献。

$\text{T}_\text{E}\text{X}$ / $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持

- $\text{T}_\text{E}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_\text{E}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_\text{E}\text{X}$ 也可以处理中文,如 Con $\text{T}_\text{E}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 应用广泛。
- 早期将 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 中使用中文。
- 还有 C $\text{T}_\text{E}\text{X}$ 为 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持作出了巨大贡献。

$\text{T}_\text{E}\text{X}$ / $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持

- $\text{T}_\text{E}\text{X}$ 本身并不支持中文编码,无法处理中文。
- $\text{T}_\text{E}\text{X}$ 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理, $\text{T}_\text{E}\text{X}$ 也可以处理中文,如 Con $\text{T}_\text{E}\text{X}$ t 可以处理中文。
- 但这些都不及 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 应用广泛。
- 早期将 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 处理方法进行了改动,和大量的 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 中使用中文。
- 还有 C $\text{T}_\text{E}\text{X}$ 为 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的中文支持作出了巨大贡献。

T_EX/L_AT_EX 的中文支持

- T_EX 本身并不支持中文编码,无法处理中文。
- T_EX 的一个全新实现,OMEGA 更适于处理中文这样的多字节编码。
- 经过一些包装和处理,T_EX 也可以处理中文,如 ConT_EXt 可以处理中文。
- 但这些都不及 L_AT_EX 应用广泛。
- 早期将 L_AT_EX 引入中国的张林波老师开发了 CCT 用以支持中文。
- 但当时的 CCT 对 L_AT_EX 处理方法进行了改动,和大量的 L_AT_EX 宏包并不兼容。
- 还好德国人 Werner Lemberg 开发了 CJK 宏包,支持在 L_AT_EX 中使用中文。
- 还有 CTeX 为 L_AT_EX 的中文支持作出了巨大贡献。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理源文件,一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的 `fontmap` 文件,建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 `GB.cpx`,标准的 `article` 和 `book` 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 1. 处理源文件, 一个一个读入亚洲字符集输入字符;
 2. 将中日韩大字符集的多种字体拆分成多个字体, 建立逻辑字体到多个具体字体的映射;
 3. 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理源文件, 一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的多种字体按需求建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件,一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体,建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx,标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件,一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体,建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx,标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件, 一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体, 建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件, 一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体, 建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件, 一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体, 建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

CJK 宏包

- CJK—Chinese, Japanese, Korean
- CJK 宏包是德国人 Werner Lemberg 的杰出贡献。
- CJK 的基本处理方法
 - ① 处理输入源文件, 一个一个读入亚洲字符集输入字符;
 - ② 将中日韩大字符集的逻辑字体拆分成多个字体, 建立逻辑字体到多个具体字体的映射;
 - ③ 在输出文件中插入对应的字体。
- 大部分 \LaTeX 宏包不加修改或稍加修改就可以和 CJK 和平共处。
- 配合 CTeX 修改后的 GB.cpx, 标准的 article 和 book 文档类都可以支持中文。

书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。
SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 \CJKfamily{kai}
```

书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。
SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 \CJKfamily{kai}
```

书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。

SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 {\CJKfamily{kai}}
```

书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。

SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 {\CJKfamily{kai}}
```


书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。

SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 {\CJKfamily{kai}}
```

书写中文

- ① 包含 CJK 宏包,以及辅助的 CJKnumb 包。
- ② 使用 CJK 环境

```
1 \begin{CJK*}{GB}{song}
```

在其中就可以输入中文了。环境的参数:

GB 字符编码,对于中文,还可能是 GBK, UTF8。

SONG 字体族,有很多种字体可选

- ③ 使用的字体还是可以改变的

```
1 {\CJKfamily{kai}}
```

细节调整

● 章节标题中的汉字

```
1 \CJKcaption{GB}
```

这会载入 GB.cpx 文件,它原本是为了一组叫 Komascript 的文档类设计的,CTeX 的 `aloft` 修改了它,于是,标准的 `article`, `book` 都可以使用它了,标题会使用“第一章”这样的字样,以及“参考文献”,“目录”...

● CJKtilde。

CJK 的设计并不能很好地处理中英文之间的衔接,源文件中,中文和英文之间如果空一格会导致中文中的英文单词的前面紧贴着中文,后面空一格,这样并不美观,我们使用 CJKtilde 命令后,‘~’的行为被改变,成为一个很小的空间,在中英文之间放上它,可以让中英文混合的文章更漂亮。

● CJKindent。标准的英文文章第一段的段首不缩进,使用这条命令后,会让第一段的段首有空格,而且将所有缩进都设置成中文习惯的缩进长度。



细节调整

● 章节标题中的汉字

1

```
\CJKcaption{GB}
```

这会载入 GB.cpx 文件,它原本是为了一组叫 Komascript 的文档类设计的,CTeX 的 **aloft** 修改了它,于是,标准的 article, book 都可以使用它了,标题会使用“第一章”这样的字样,以及“参考文献”,“目录”...

● CJKtilde。

CJK 的设计并不能很好地处理中英文之间的衔接,源文件中,中文和英文之间如果空一格会导致中文中的英文单词的前面紧贴着中文,后面空一格,这样并不美观,我们使用 CJKtilde 命令后,‘~’的行为被改变,成为一个很小的空间,在中英文之间放上它,可以让中英文混合的文章更漂亮。

● CJKindent。标准的英文文章第一段的段首不缩进,使用这条命令后,会让第一段的段首有空格,而且将所有缩进都设置成中文习惯的缩进长度。



细节调整

- 章节标题中的汉字

1

```
\CJKcaption{GB}
```

这会载入 GB.cpx 文件,它原本是为了一组叫 Komascript 的文档类设计的,CTeX 的 **aloft** 修改了它,于是,标准的 article, book 都可以使用它了,标题会使用“第一章”这样的字样,以及“参考文献”,“目录”...

- CJKtilde。

CJK 的设计并不能很好地处理中英文之间的衔接,源文件中,中文和英文之间如果空一格会导致中文中的英文单词的前面紧贴着中文,后面空一格,这样并不美观,我们使用 CJKtilde 命令后,‘~’ 的行为被改变,成为一个很小的空间,在中英文之间放上它,可以让中英文混合的文章更漂亮。

- CJKindent。标准的英文文章第一段的段首不缩进,使用这条命令后,会让第一段的段首有空格,而且将所有缩进都设置成中文习惯的缩进长度。



细节调整

● 章节标题中的汉字

1

```
\CJKcaption{GB}
```

这会载入 GB.cpx 文件,它原本是为了一组叫 Komascript 的文档类设计的,CTeX 的 **aloft** 修改了它,于是,标准的 article, book 都可以使用它了,标题会使用“第一章”这样的字样,以及“参考文献”,“目录”...

● CJKtilde。

CJK 的设计并不能很好地处理中英文之间的衔接,源文件中,中文和英文之间如果空一格会导致中文中的英文单词的前面紧贴着中文,后面空一格,这样并不美观,我们使用 CJKtilde 命令后,‘~’的行为被改变,成为一个很小的空间,在中英文之间放上它,可以让中英文混合的文章更漂亮。

● CJKindent。标准的英文文章第一段的段首不缩进,使用这条命令后,会让第一段的段首有空格,而且将所有缩进都设置成中文习惯的缩进长度。



大纲

- ① L^AT_EX 导引
- ② L^AT_EX 的基本使用
- ③ 书写中文
- ④ 插入公式、图形、表格以及程序代码
 - 插入公式
 - 插入表格
 - 插入图形
 - 插入程序代码
- ⑤ 参考文献
- ⑥ 更多技巧与常用宏包
- ⑦ 参考文献与网上资料

T_EX 的数学模式

数学模式

- 数学模式几乎是 T_EX 诞生的原因,而很多人使用 L^AT_EX 也是因为漂亮的数学排版。
- 数学模式是 T_EX 的内建支持,包含了各种数学用的符号和排版的规则,所有你看到的数学书中的数学公式排版,都可以由 T_EX 实现。
- 数学模式中的所有的公式都可以用纯文本录入,这意味着你在输入公式的时候无须将手移开键盘。

T_EX 的数学模式

数学模式

- 数学模式几乎是 T_EX 诞生的原因,而很多人使用 L^AT_EX 也是因为漂亮的数学排版。
- 数学模式是 T_EX 的内建支持,包含了各种数学用的符号和排版的规则,所有你看到的数学书中的数学公式排版,都可以由 T_EX 实现。
- 数学模式中的所有的公式都可以用纯文本录入,这意味着你在输入公式的时候无须将手移开键盘。

T_EX 的数学模式

数学模式

- 数学模式几乎是 T_EX 诞生的原因,而很多人使用 L^AT_EX 也是因为漂亮的数学排版。
- 数学模式是 T_EX 的内建支持,包含了各种数学用的符号和排版的规则,所有你看到的数学书中的数学公式排版,都可以由 T_EX 实现。
- 数学模式中的所有的公式都可以用纯文本录入,这意味着你在输入公式的时候无须将手移开键盘。

公式环境

- 我们可以在文章的字里行间插入公式,如

```
1 $\min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml}$
2 \sum_{l=1}^L R_{ml}$
```

- 结果就像 $\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml}$
- 也可以断开行,突出显示

```
1 \begin{equation}
2 \min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml},
3 \sum_{l=1}^L R_{ml},
4 \end{equation}
```

- 显示结果如

$$\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml},$$

公式环境

- 我们可以在文章的字里行间插入公式,如

```
1 $\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml}$
2 \sum_{l=1}^L R_{ml}$
```

- 结果就像 $\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml}$
- 也可以断开行,突出显示

```
1 \begin{equation}
2 \min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml},
3 \sum_{l=1}^L R_{ml},
4 \end{equation}
```

- 显示结果如

$$\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml},$$

公式环境

- 我们可以在文章的字里行间插入公式,如

```
1 $\min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml}$
2 \sum_{l=1}^L R_{ml}$
```

- 结果就像 $\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml}$
- 也可以断开行,突出显示

```
1 \begin{equation}
2 \min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml},
3 \sum_{l=1}^L R_{ml},
4 \end{equation}
```

- 显示结果如

$$\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml},$$

公式环境

- 我们可以在文章的字里行间插入公式,如

```
1 $\min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml}$
2 \sum_{l=1}^L R_{ml}$
```

- 结果就像 $\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml}$
- 也可以断开行,突出显示

```
1 \begin{equation}
2 \min_R D_{\text{rec}}(R)+\lambda\sum_{m=1}^M\sum_{l=1}^L R_{ml},
3 \sum_{l=1}^L R_{ml},
4 \end{equation}
```

- 显示结果如

$$\min_R D_{\text{rec}}(R) + \lambda \sum_{m=1}^M \sum_{l=1}^L R_{ml},$$

多行公式

使用 `EQNARRY` 支持多行公式

```
\begin{eqnarray}
D_{dec} &=& D_0 + \theta / (R - R_0) + \nonumber \\
&& \kappa (P_r + (1 - P_r) e^{-(C-R)T/L}) . \\
\label{equ:distort}
\end{eqnarray}
```

显示效果如

$$D_{dec} = D_0 + \theta / (R - R_0) + \kappa (P_r + (1 - P_r) e^{-(C-R)T/L}) . \quad (2)$$

多行公式

使用 `EQNARRY` 支持多行公式

```
\begin{eqnarray}
D_{dec}=&D_0+\backslashtheta/(R-R_0)+ \backnonumber\\
&\&\backkappa(P_r+(1-P_r)e^{\{-(C-R)T/L\}}.
\label{equ:distort}
\end{eqnarray}
```

显示效果如

$$D_{dec} = D_0 + \theta / (R - R_0) + \kappa (P_r + (1 - P_r) e^{-(C-R)T/L}). \quad (2)$$

多行公式

使用 **EQNARRY** 支持多行公式

```
\begin{eqnarray}
D_{dec} &=& D_0 + \theta / (R - R_0) + \nonumber \\
&& \kappa (P_r + (1 - P_r) e^{-(C-R)T/L}) . \\
\label{equ:distort}
\end{eqnarray}
```

显示效果如

$$D_{dec} = D_0 + \theta / (R - R_0) + \kappa (P_r + (1 - P_r) e^{-(C-R)T/L}). \quad (2)$$

绘制表格

使用 TABULAR 环境绘制表格

```
\begin{tabular}{|l|lc|p{5em}|}
\hline
number & type & example & comments \\
\hline
\hline
\hline
1          & bf      & \textbf{bf}      & for the
defination etc.\\
2          & italic & \textit{italic} & emph.\\
\hline
```

绘制表格

显示效果如

number	type	example	comments
1	bf	bf	for the defi- nation etc.
2	italic	<i>italic</i>	emph.

每一列的对齐方式对应环境的声明

```
\begin{tabular}{|l|l|lc|p{5em}|}
```

浮动表格环境

- 我们常常希望表格的位置不是固定的,而是在页面的某些位置浮动,于是,使用 `table` 环境构成浮动图形。
- 同时,我们还用 “`caption`” 为表格加标题,用 “`label`” 来打标记。
- 如示例

```
1 \begin{table}[htb]
2 \centering
3 \caption{Sample Table}
4 \label{tab:sample}
5 \begin{tabular}{|l|l|l|c|p{5em}|}
```

浮动表格环境

- 我们常常希望表格的位置不是固定的,而是在页面的某些位置浮动,于是,使用 `table` 环境构成浮动图形。
- 同时,我们还用 “`caption`” 为表格加标题,用 “`label`” 来打标记。
- 如示例

```
1 \begin{table}[htb]
2 \centering
3 \caption{Sample Table}
4 \label{tab:sample}
5 \begin{tabular}{|l|l|l|c|p{5em}|}
```

浮动表格环境

- 我们常常希望表格的位置不是固定的,而是在页面的某些位置浮动,于是,使用 `table` 环境构成浮动图形。
- 同时,我们还用 “`caption`” 为表格加标题,用 “`label`” 来打标记。
- 如示例

```
1 \begin{table}[htb]
2 \centering
3 \caption{Sample Table}
4 \label{tab:sample}
5 \begin{tabular}{|l|l|c|p{5em}|}
```

支持的图片格式

- 我们在 \LaTeX 文档中, 通常使用 `graphicx` 宏包来插入图形。
- 以下格式的图片是被支持的:

	图形格式	说明
\LaTeX	eps	嵌入式 PS 文件
<code>pdf\LaTeX</code>	pdf	对就是那个
	png	一种位图
	jpg	另一种位图

- eps 图形格式
 - 是一种矢量图
 - 可以直接绘制
 - 可以从 PS 文件中剪裁或通过 PS 虚拟打印机得到
 - 可以通过其他工具, 如 (Imagimagick) 转换格式得到。

支持的图片格式

- 我们在 \LaTeX 文档中, 通常使用 `graphicx` 宏包来插入图形。
- 以下格式的图片是被支持的:

	图形格式	说明
\LaTeX	eps	嵌入式 PS 文件
pdf \LaTeX	pdf	对就是那个
	png	一种位图
	jpg	另一种位图

- eps 图形格式
 - 是一种矢量图
 - 可以直接绘制
 - 可以从 PS 文件中剪裁或通过 PS 虚拟打印机得到
 - 可以通过其他工具, 如 (Imagimagick) 转换格式得到。

插入图片



插入图片的命令行

以上图片使用如下命令行插入:

```
\includegraphics[height=2cm]{figures/wti-logo}
```



插入图片



常用命令行参数

参数	用途
width/height	图片宽度或高度,如本例
scale	图片缩放的系数,如 <code>scale=0.5</code>
angle	图片旋转的角度,如 <code>angle=90</code>
clip	eps 图片切边

浮动图形

- 与表格类似, 图片也可以以浮动图形的方式出现在文档之中。
- 格式如下

```
1 \begin{figure}[htb]
2     \begin{center}
3     % \includegraphics{}
4     \end{center}
5     \caption{Sample figure}
6     \label{fig:sample}
```

listings 宏包

- **verb** 命令和 **verbatim** 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (`\lstinputlisting` 和 `\lstinputlisting`) 与一个环境 (`\lstlisting`), 其中 `\lstinputlisting` 这个命令最为贴心, 可以读入其它文件。
 - 支持多种程序语言的语法, 对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

listings 宏包

- **verb** 命令和 **verbatim** 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (`\lstinline` 和 `\lstinputlisting`) 与一个环境 (`\lstlisting`), 其中 `\lstinputlisting` 这个命令最为贴心, 可以读入其它文件。
 - 支持多种程序语言的语法, 对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

listings 宏包

- `verb` 命令和 `verbatim` 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (`\lstinline` 和 `\lstinputlisting`) 与一个环境 (`\lstlisting`),其中 `\lstinputlisting` 这个命令最为贴心,可以读入其它文件。
 - 支持多种程序语言的语法,对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

listings 宏包

- **verb** 命令和 **verbatim** 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (**lstinline**listing 和 **lstinput**listing) 与一个环境 (**lstlisting**),其中 **lstinput**listing 这个命令最为贴心,可以读入其它文件。
 - 支持多种程序语言的语法,对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

listings 宏包

- **verb** 命令和 **verbatim** 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (**lstinline**listing 和 **lstinput**listing) 与一个环境 (**lstlisting**),其中 **lstinput**listing 这个命令最为贴心,可以读入其它文件。
 - 支持多种程序语言的语法,对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

listings 宏包

- `verb` 命令和 `verbatim` 环境可以用来原封不动的保持输入的内容,理论上讲,我们可以使用它们来展示程序代码。
- listings 宏包的特色
 - 两个命令 (`\lstinline` 和 `\lstinputlisting`) 与一个环境 (`\lstlisting`), 其中 `\lstinputlisting` 这个命令最为贴心, 可以读入其它文件。
 - 支持多种程序语言的语法, 对关键字、注释等进行语法加亮
 - 支持边框、背景色、字体风格、行号等特征
- 本幻灯片中大部分代码是通过 listings 宏包加入的。

设置与调用 listings 宏包

本幻灯片中的 LISTINGS 设置

```
\usepackage{pgf,pgfarrows,pgfnodes,pgfautomata,pgfheaps,pgf}
%\usepackage{CJK,CJKnumb}
%\usepackage{indentfirst}
\usepackage{tipa}
\usepackage{listings}

\setbeamercovered{dynamic}
```

本文中使用的 LISTINGS

```
\lstinputlisting[firstline=295,lastline=295]{tex.tex}
```

设置与调用 listings 宏包

常用参数

参数名称	说明
caption	一般的 caption 的内容,文章的题目
label	一般的 label 的含义,引用此列表的标记
frame	边框设置
language	设置的语言
firstline	引用文件的起始行号
lastline	引用文件的终止行号

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献**
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

参考文献列表

参考文献列表很象 ITEMIZE 环境

```
\begin{thebibliography}{10}
\bibitem{bib:ubi91weiser}
  M. Weiser,
  The computers for the twenty-first century,
  Scientific American,
  vol. 265, pp. 94-100, Sept. 1991.
\bibitem{bib:wwrf}
  The Wireless World Research Forum,
  http://www.wireless-world-research.org .
\end{thebibliography}
```

参考文献的引用

使用 `CITE` 命令引用参考文献

- 可以使用最简单的引用方式

```
1 in \cite{bib:ubi91weiser}, \ldots
```

- 也可以一次引用多个参考文献

```
1 in \cite{bib:ubi91weiser,bib:wwrf}, \ldots
```

- 还可以加上更多的说明

```
1 in \cite[pp. 99-105]{bib:wwrf}, \ldots
```

大纲

- ① L^AT_EX 导引
- ② L^AT_EX 的基本使用
- ③ 书写中文
- ④ 插入公式、图形、表格以及程序代码
- ⑤ 参考文献
- ⑥ 更多技巧与常用宏包**
 - 页面维度设置
 - 页脚页眉设置
 - 自定义命令与环境
- ⑦ 参考文献与网上资料

geometry 宏包

GEOMETRY 的典型使用

```
\usepackage[a4paper,%  
    includehead,%  
    includefoot,%  
    top=2cm,%  
    bottom=2cm,%  
    left=2cm,%  
    right=2cm,%  
    head=15pt,%  
    foot=15pt%  
{geometry}
```

页面风格

L^AT_EX 的内建风格

- **empty** 是最简单的,也就是什么都没有。
- **plain** 是很简单的,缺省的风格,article 的首页和 book 的每章开始的地方通常也会是这个风格,只有下面有页码。
- **heading** 页眉放置章节的标题。

FANCYHDR 宏包

- 页脚上方和页眉下方划线
- 左中右三段式的页脚和页眉的设置

页面风格

L^AT_EX 的内建风格

- **empty** 是最简单的,也就是什么都没有。
- **plain** 是很简单的,缺省的风格,article 的首页和 book 的每章开始的地方通常也会是这个风格,只有下面有页码。
- **heading** 页眉放置章节的标题。

FANCYHDR 宏包

- 页脚上方和页眉下方划线
- 左中右三段式的页脚和页眉的设置

页面风格

L^AT_EX 的内建风格

- **empty** 是最简单的,也就是什么都没有。
- **plain** 是很简单的,缺省的风格,article 的首页和 book 的每章开始的地方通常也会是这个风格,只有下面有页码。
- **heading** 页眉放置章节的标题。

FANCYHDR 宏包

- 页脚上方和页眉下方划线
- 左中右三段式的页脚和页眉的设置

页面风格

L^AT_EX 的内建风格

- **empty** 是最简单的,也就是什么都没有。
- **plain** 是很简单的,缺省的风格,article 的首页和 book 的每章开始的地方通常也会是这个风格,只有下面有页码。
- **heading** 页眉放置章节的标题。

FANCYHDR 宏包

- 页脚上方和页眉下方划线
- 左中右三段式的页脚和页眉的设置

页面风格

L^AT_EX 的内建风格

- **empty** 是最简单的,也就是什么都没有。
- **plain** 是很简单的,缺省的风格,article 的首页和 book 的每章开始的地方通常也会是这个风格,只有下面有页码。
- **heading** 页眉放置章节的标题。

FANCYHDR 宏包

- 页脚上方和页眉下方划线
- 左中右三段式的页脚和页眉的设置

fancyhdr 宏包

ARTICLE 中常用的 FANCYHDR 用法

```
\fancyhf{}  
\pagestyle{fancy}  
\lhead{\rightmark}  
\chead{{\color{blue} $\spadesuit$ gnawux $\spadesuit$}}  
\rhead{\bfseries\thepage}  
\lfoot{Manipulated by \LaTeXe.}  
\rfoot{K.I.S.S.---Keep It Simple Stupid}  
\renewcommand{\headrulewidth}{0.5pt}  
\renewcommand{\footrulewidth}{0.5pt}  
\fancypagestyle{plain}{%  
    \fancyhf{} % get rid of headers on plain pages  
    \cfoot{\bfseries\thepage}  
    \renewcommand{\headrulewidth}{0pt} % and the line  
    \renewcommand{\footrulewidth}{0.5pt}  
}
```

自定义命令与环境

我们还可以修改一些命令或环境的行为

- newcommand 和 newenvironment 命令可以定义新命令和环境
- renewcommand 和 renewenvironment 命令可以修改已有的命令和环境
- 这里也有一个例子

```
1 \renewcommand{\sectionmark}[1]  
2 {\markright{\thesection\ #1}}
```


自定义命令与环境

我们还可以修改一些命令或环境的行为

- newcommand 和 newenvironment 命令可以定义新命令和环境
- renewcommand 和 renewenvironment 命令可以修改已有的命令和环境
- 这里也有一个例子

```
1 \renewcommand{\sectionmark}[1]  
2 {\markright{\thesection\ #1}}
```

大纲

- 1 L^AT_EX 导引
- 2 L^AT_EX 的基本使用
- 3 书写中文
- 4 插入公式、图形、表格以及程序代码
- 5 参考文献
- 6 更多技巧与常用宏包
- 7 参考文献与网上资料

文章推荐

入门必读

- ① lshort-cn
- ② Latex2e 使用手册(Latex Manual, latex 科技文献排版指南电子版)
- ③ latex 插图指南(epslatex)中文版

常用快速参考手册

- \LaTeX Command.
- \LaTeX Cookbook.

文章推荐

入门必读

- ① lshort-cn
- ② Latex2e 使用手册 (Latex Manual, latex 科技文献排版指南电子版)
- ③ latex 插图指南 (epslatex) 中文版

常用快速参考手册

- \LaTeX Command.
- \LaTeX Cookbook.

文章推荐

入门必读

- ① [lshort-cn](#)
- ② [Latex2e 使用手册 \(Latex Manual, latex 科技文献排版指南电子版 \)](#)
- ③ [latex 插图指南 \(epslatex \) 中文版](#)

常用快速参考手册

- [L^AT_EX Command.](#)
- [L^AT_EX Cookbook.](#)

文章推荐

入门必读

- ① lshort-cn
- ② Latex2e 使用手册 (Latex Manual, latex 科技文献排版指南电子版)
- ③ latex 插图指南 (epslatex) 中文版

常用快速参考手册

- \LaTeX Command.
- \LaTeX Cookbook.

文章推荐

入门必读

- ① lshort-cn
- ② Latex2e 使用手册 (Latex Manual, latex 科技文献排版指南电子版)
- ③ latex 插图指南 (epslatex) 中文版

常用快速参考手册

- \LaTeX Command.
- \LaTeX Cookbook.

文章推荐

入门必读

- ① lshort-cn
- ② Latex2e 使用手册 (Latex Manual, latex 科技文献排版指南电子版)
- ③ latex 插图指南 (epslatex) 中文版

常用快速参考手册

- \LaTeX Command.
- \LaTeX Cookbook.

其它推荐文章:

- CteX-FAQ
- lshort-en
- latex 插图指南(using imported graphics in latex2e)英文版
- texbook
- latex companion ch8(经典书籍 latex companion 的第八章,详细讲解数学输入)
- latex-help-book
- short math guide
- math mode

网站推荐

- CT_EX: 中文 T_EX 中坚站, 文档、下载、论坛
- 新水木 T_EX 版
- CTAN: Comprehensive T_EX Archive Network, T_EX 大全

谢 谢 !

2006 年 2 月