

这里是封面

指导成员小组

目录

CHAPTER	TITLE	PAGE
1	不怕命令行	1
1.1	任务	1
1.2	Windows 命令行	1
2	不怕命令行	2
2.1	任务	2
2.2	Windows 命令行	2
3	不怕命令行	3
3.1	任务	3
3.2	Windows 命令行	3
4	不怕命令行	4
4.1	任务	4
4.2	Windows 命令行	4
5	不怕命令行	5
5.1	任务	5
5.2	Windows 命令行	5
6	不怕命令行	6
6.1	任务	6
6.2	Windows 命令行	6
	附录	7
	附录一	8
	附录二	9

摘要

ConTeXt 是一个强大的排版系统，适用于对版面和结构有高度控制需求的用户，特别适合多语言文档、技术手册和学术出版。其基于 LuaTeX 引擎，支持模块化设计、样式重用与动态内容生成。与 LaTeX 相比，ConTeXt 提供更一致的语法和更灵活的样式定义方式，用户可以通过关键字（如 `\startsection`、`\definefontfamily`、`\setuphead`）自定义页面布局、字体设定、章节格式、图文混排等内容。它内建对中文等 CJK 文本支持（通过 `zhfonts` 模块或自定义字体设定），并支持 PDF 输出、图形绘制（MetaFun）及自动化任务（如引用、目录、索引等）。ConTeXt 的使用依赖于命令行操作（如 `context` 编译、`mtxrun` 管理工具），适合希望在一个统一框架下进行高质量排版设计的用户。

Abstract

ConTeXt is a powerful typesetting system designed for users who require fine-grained control over layout and structure, making it particularly suitable for multilingual documents, technical manuals, and academic publishing. It is built on the LuaTeX engine and supports modular design, style reuse, and dynamic content generation. Compared to LaTeX, ConTeXt offers a more consistent syntax and more flexible style definitions, allowing users to customize page layout, font settings, section formatting, and image-text integration using commands such as `\startsection`, `XXXX XXXXXX XXXXX XXXXXXXX XXXXXX \definefontfamily`, and `\setuphead`. It includes built-in support for CJK scripts like Chinese (via the `zhfonts` module or custom font settings), and supports PDF output, graphical illustration (via MetaFun), and automation tasks such as cross-referencing, table of contents, and indexing. ConTeXt is operated primarily through the command line (e.g., using `context` for compilation and `mtxrun` for management), and is well-suited for users seeking high-quality typesetting within a unified framework.

第 1 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

1.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

1.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

第 2 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

2.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

2.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

第 3 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

3.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

3.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

第 4 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

4.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

4.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

第 5 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

5.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

5.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

第 6 章 不怕命令行

无论是安装还是使用 ConTeXt，皆需要对命令行环境有所了解。原本未有介绍这方面知识的计划，但是考虑到我正在写一份世上最好的 ConTeXt 入门文档，便有了些许动力。本章先分别介绍 Windows、Linux 和 macOS 系统的命令行环境的基本用法，以刚好满足安装和运行 ConTeXt 的需求为要。倘若对命令行环境已颇为熟悉，可直接阅读?? 和?? 节。

6.1 任务

使用命令行环境，在文件系统中，创建一个名为 foo 的目录，在该目录内创建一份 Shell 脚本，令其可在命令行窗口中输出「不怕命令行」，执行该脚本，查看其输出。

6.2 Windows 命令行

Windows 用户似乎畏惧甚至厌憎命令行环境，甚至很多人认为命令行环境是早已被淘汰的上个世纪的产物，因此要教会他们如何使用命令行环境，通常会有些麻烦，我当勉力为之。

在 Windows 系统中打开一个命令行窗口，有很多种方法，其中最快的应当是使用如图?? 所示快捷键「Win + R」，打开「运行」对话框，在其中输入「cmd」后点击「确定」按钮或单击「Enter」键，即可打开与图?? 类似的命令行窗口。

附录

这里是附录的内容。

附录一

这里是附录一的内容。

附录二

这里是附录二的内容。

参考文献

攻读学位期间发表的学术论文

致谢

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。论文中除特别标注的内容外，不包含任何其他个人或机构已经发表或撰写过的研究成果。对本研究做出重要贡献的个人和集体，均已在论文中作了明确的声明并表示了谢意。本声明的法律结果由本人承担。

本人完全了解复旦大学有关收藏和利用博士、硕士学位论文的规定，即：学校有权收藏、使用并向国家有关部门或机构送交论文的印刷本和电子版本；允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。涉密学位论文在解密后遵守此规定。

