LuaTeX-CN 文档

— 古籍版式复刻 · 遵循 clreq 的现代中文排版 —

从 GitHub Wiki 自动生成

生成时间 2026-11-18 23:04 UTC · 仓库 9cc66cd · wiki 078a7d6

目录

LuaTeX-CN Wiki

安装指南

快速入门

古籍版式要素

示例

模板使用与自定义

字体设置

功能详解

现代竖排

- 标点系统
- 句读

    夹注与侧批
    批注与眉批
    改字与装饰
    抬头
    文本框
    印章
调试模式
命令索引
更新日志
开发文档
发布流程

LuaTEX-CN Wiki

欢迎来到 LuaTEX-CN 的用户手册。本 Wiki 涵盖了从安装到高级排版的完整文档，帮助你用 LuaTEX-CN 排版高质量的中文古籍、现代竖排书籍与符合 clreq 的现代横排文档。

> **当前版本**: v0.4.5 | **CTAN**: v0.4.5

从这里开始

1. **安装指南** — 从零安装 TEX Live 和 luatex-cn
2. **快速入门** — 5 分钟跑通你的第一个排版示例
3. **示例** — 浏览《史记》、《红楼梦》等完整复刻

排版模式

竖排：四个文档类

文档类	用途	详情
`ltc-guji`	传统古籍排版（版心、鱼尾、丝栏）	快速入门
`ltc-guji-digital`	古籍数字化录入：一行源码 = 一个竖排列，手工指定每列内容以精确复刻底本版面	源码示例
`ltc-cn-vbook`	现代中国大陆竖排书籍（标点挤压、脚注）	ltc-book
`ltc-tw-vbook`	现代台湾竖排书籍（居中标点）	ltc-book

> 旧名 `ltc-book` 已不再提供，请用 `ltc-cn-vbook` ／ `ltc-tw-vbook`。

横排：一个宏包

宏包	用途	详情
`\usepackage{luatex-cn}`（横排）	按 W3C clreq 排版的现代中文横排：中西间距、四级禁则、行内挤压／拉伸、行间标号与拼音、段落与孤行控制。加载到 `article` 等任意横排文档类之上，非文档类	现代横排

核心文档

排版基础

- 版心与鱼尾 — 理解古籍排版的核心"版心"概念，预设模板与自定义
- 字体设置 — 自动探测、手动设置、字体递补链
- 布局与功能 — 章节标题、目录条目及布局控制
- 现代横排 — `\usepackage{luatex-cn}`：clreq 横排管道、选项与命令
- clreq 符合度矩阵 — 逐条款的横竖排符合状态、证据与有意偏离

标点与句读

- 标点系统 — 现代标点挤压、禁则处理，中国大陆/台湾风格
- 句读 — 传统句读标记（句/读/白文三种模式）

注释系统

- 夹注与侧批 — 双行小字注、行间侧批
- 批注与眉批 — 页面浮动批注、顶部眉批

装饰与辅助

- 改字与装饰 — 勘误改字、专名号、书名号、着重号
- 抬头 — 敬语抬头格式（单抬、双抬、三抬、平抬）
- 文本框 — 独立文本框、绝对定位
- 印章 — 电子印章叠加

数字化与转换

- 数字化排版 — 语义模式到布局模式的转换工作流与工具

调试与参考

- 调试模式 — 网格可视化与排版调试
- **命令索引** — 全部命令、环境及参数的速查表

开发者文档

- 架构与开发 — 三阶段渲染引擎、源码结构
- 版本发布规范 — 发布流程与 CI/CD
- 更新日志 — 版本历史

联系方式

- **联系人**: Sheldon Li（sheldonli.dev@gmail.com）、Frank Lin（ctan@linshuang.info）
- **Issue**: GitHub Issues

安装指南

本页面手把手教你完成安装，包括 TEX 环境本身和 `luatex-cn` 宏包。

前提条件：安装 TEX Live

`luatex-cn` 运行在 **LuaTEX** 引擎上，需要安装 **TEX Live**（推荐 2024 或更新版本）。如果你已经安装了 TEX Live，可以跳过此节。

Windows

1. 访问 TEX Live 安装页面，下载 `install-tl-windows.exe`
2. 运行安装程序，选择"完整安装"（Full scheme），或至少确保勾选了 `luatex`、`fontspec`、`luaotfload` 等包
3. 安装完成后，打开命令提示符（CMD），输入 `lualatex --version`，看到版本号即表示安装成功

macOS

1. 下载 MacTeX（约 4GB，包含完整 TEX Live）
2. 双击 `.pkg` 文件安装
3. 打开终端，输入 `lualatex --version` 验证

Linux

```
# Ubuntu / Debian
sudo apt install texlive-full

# Fedora
sudo dnf install texlive-scheme-full

# Arch Linux
sudo pacman -S texlive
```

安装后运行 `lualatex --version` 验证。

> **提示**：如果不想安装完整版，最小依赖包括：`texlive-base`、`texlive-luatex`、`texlive-latex-extra`、`texlive-fonts-recommended`、`texlive-lang-chinese`。但推荐使用完整安装以避免缺包问题。

安装 luatex-cn

方法 1：从 GitHub Release 安装（推荐）

1. **下载**：前往 GitHub Releases 下载最新版本的 `luatex-cn-tex-v*.zip`
2. **定位 texmf 目录**：
   - **Windows**：`C:\Users\<你的用户名>\texmf`（如果不存在则手动创建）
   - **macOS/Linux**：`~/texmf`（如果不存在则运行 `mkdir -p ~/texmf`）
3. **放置文件**：将压缩包中 `tex/` 下的所有内容放入：

   ```
   texmf/tex/latex/luatex-cn/
   ```

4. **刷新 TEX 数据库**：

   ```
   texhash
   ```

5. **（可选）安装 PDF 手册**：将压缩包中 `doc/` 下的内容（`luatex-cn-wiki-zh.pdf` / `luatex-cn-wiki-en.pdf` 等）放入：

   ```
   texmf/doc/latex/luatex-cn/
   ```

   之后即可用 `texdoc luatex-cn` 直接打开 PDF 手册（v0.4.0 起 Release 压缩包内含 `doc/`）。

方法 2：通过 CTAN 包管理器安装

```
# TeX Live (Windows/Linux)
tlmgr install luatex-cn

# MiKTeX
# 打开 MiKTeX Console → Packages → 搜索 luatex-cn → 安装
```

> **注意**：CTAN 版本可能落后于 GitHub Release，建议使用方法 1。经 CTAN/包管理器安装时文档随包进入 texmf 文档树，`texdoc luatex-cn` 可直接打开 PDF 手册。

方法 3：直接放到项目目录

如果只是想快速试用，可以将 `tex/` 下的内容直接放到你的 `.tex` 文件所在的同一目录下，无需安装到 texmf。

验证安装

创建测试文件 `test.tex`：

```
\documentclass[四库全书]{ltc-guji}
\begin{document}
\begin{正文}
天地玄黄，宇宙洪荒。
\end{正文}
\end{document}
```

编译（注意：必须使用 **`lualatex`**，不是 `pdflatex` 或 `xelatex`）：

```
lualatex test.tex
```

如果编译成功并生成 `test.pdf`，恭喜你安装完成！打开 PDF 应该能看到竖排的文字。

> 如果遇到 `File 'ltc-guji.cls' not found` 错误，说明 luatex-cn 没有正确安装，请检查文件路径和 `texhash` 是否已执行。

字体说明

`luatex-cn` 会**自动检测**你系统上已安装的中文字体。大多数情况下无需额外配置。

如果你希望获得更好的古籍效果，推荐安装：

- **思源宋体** (Noto Serif CJK SC / Source Han Serif) — 开源高质量宋体

- **楷体** (KaiTi) — 模拟手抄本效果
- **仿宋** (FangSong) — 模拟刻本效果

安装字体后，可在 `.tex` 文件中指定：

```
\设置字体{Noto Serif SC}
```

更多字体配置请参阅 字体设置。

下一步

安装完成后，前往 **快速入门** 学习基本命令和排版流程。

快速入门

本页面帮助你在 5 分钟内用 `luatex-cn` 排出第一个古籍竖排页面。

> 还没安装？先看 安装指南。

第一步：最小示例

创建文件 `test.tex`，写入以下内容：

```
\documentclass[四库全书]{ltc-guji}

\title{钦定四库全书}
\chapter{史记\\卷一}

\begin{document}
\begin{正文}
太史公曰：學者多稱五帝，尚矣。
然尚書獨載堯以來，而百家言黃帝，
其文不雅馴，薦紳先生難言之。
\end{正文}
\end{document}
```

用 `lualatex` 编译（注意：**不是** `pdflatex` 或 `xelatex`）：

```
lualatex test.tex
```

打开生成的 `test.pdf`，你应该能看到一个竖排的古籍页面，包含版心、鱼尾和书名。

第二步：加入更多功能

在 `\begin{正文}` … `\end{正文}` 之间，你可以使用以下命令。每个命令都有中文名称，直接写在正文中即可：

夹注（双行小字注释）

```
學者多稱五帝\夹注{三皇五帝之五帝}，尚矣。
```

夹注会自动分为左右两列、自动平衡长度、自动跨列跨页。详见 夹注与侧批。

侧批（行间批注）

```
太史公曰\侧批{此为开篇}：學者多稱五帝。
```

侧批出现在两列之间。详见 夹注与侧批。

段落缩进

```
\begin{段落}[indent=2]
    天地玄黄\\
    宇宙洪荒
\end{段落}
```

`indent=2` 表示缩进 2 格，`\\` 表示强制换列。

空格

```
前面\空格[2]后面
```

插入 2 个全角空格。竖排中空格不会自动产生，需要手动声明。

强制换列

```
这一列结束\\
新的一列开始
```

`\\` 在正文中表示强制换到下一列。

第三步：一个完整示例

把上面的命令组合起来：

```
\documentclass[四库全书]{ltc-guji}

\title{钦定四库全书}
\chapter{史记\\卷一}

\begin{document}
\begin{正文}

太史公曰\侧批{此为开篇}:
學者多稱五帝\夹注{三皇五帝之五帝}，尚矣。
然尚書獨載堯以來，而百家言黃帝，
其文不雅馴，薦紳先生難言之。\\

\begin{段落}[indent=2]
孔子所傳宰予問五帝德及帝繫姓，
儒者或不傳。
\end{段落}

\end{正文}
\end{document}
```

编译后你会看到一个包含版心、夹注、侧批和缩进段落的竖排页面。

预设模板

通过文档类选项选择不同的预设风格：

guji（古籍排版）

选项	说明
`四库全书`	清代四库全书风格，单鱼尾、乌丝栏
`四库全书彩色`	彩色版四库全书
`红楼梦甲戌本`	手抄本风格，无鱼尾，支持侧批与眉批

```
\documentclass[四库全书]{ltc-guji}
\documentclass[红楼梦甲戌本]{ltc-guji}
```

cn-vbook / tw-vbook（现代竖排）

文档类	说明
`ltc-cn-vbook`	简洁现代竖排，中国大陆标点风格

文档类	说明
`ltc-tw-vbook`	简洁现代竖排，台湾标点风格

```
\documentclass{ltc-cn-vbook}
\documentclass{ltc-tw-vbook}
```

详见 模板与自定义 和 现代竖排。

更多命令速览

命令	用途	详细文档
`\夹注{...}`	双行小字注释	夹注与侧批
`\侧批{...}`	行间侧批	夹注与侧批
`\批注[x=, y=]{...}`	页面浮动批注	批注与眉批
`\眉批{...}`	页面顶部批注	批注与眉批
`\印章[page=1]{img.png}`	添加电子印章	印章
`\句读模式`	切换到传统句读	句读
`\专名号{人名}`	人名/地名标记线	改字与装饰
`\书名号{书名}`	书名标记波浪线	改字与装饰
`\改{正}`	古籍勘误改字	改字与装饰
`\抬头[1]`	敬语提行	抬头
`\脚注{...}`	脚注	现代竖排
`\LtcDebugOn`	开启调试模式	调试模式

完整命令列表请查看 **命令索引**。

元数据设置

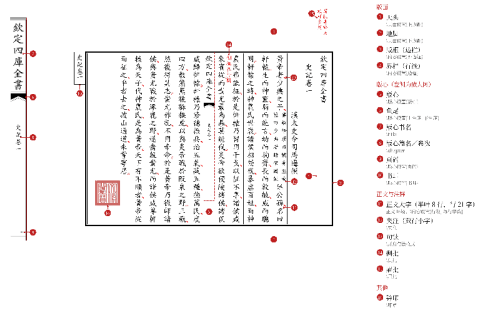
```
\title{书名}          % 显示在版心
\chapter{卷一}        % 章节标题，重置页码
\publisher{出版社}    % 显示在版心底部
```

下一步

- **示例** — 查看《史记》、《红楼梦》的完整排版源码
- **模板与自定义** — 创建自己的排版模板
- **命令索引** — 查找任何命令的详细参数
- **字体设置** — 配置中文字体和字体递补

古籍版式要素

不熟悉古籍版式的读者，可以先在这一页对照：图上每个编号是一种版式要素，下表给出它的标准名称、luatex-cn 里对应的命令或参数，以及详细说明所在的页面。



> 图为一整版（左右两个半叶连同中间的版心），用 `\documentclass[四库全书]{ltc-guji}` 排出，加 `\关闭分页` 不拆成两页。左侧是版心的放大图。 底稿与生成脚本见仓库 `资源/wiki-figures/anatomy/` 。

名称对照表

编号	古籍要素	含义	luatex-cn 对应	详见
1	天头	版框上方的空白	`\页面设置{上边距}`	命令索引
2	地脚	版框下方的空白	`\页面设置{下边距}`	命令索引
3	版框（边栏）	围住整版的外框；四周双边即内外两道线	`\内容设置{外边框，外边框粗细，外边框间距}`	模板与自定义
4	界栏（行线）	行与行之间的竖线	`\内容设置{边框，边框粗细，边框色}`	模板与自定义
5	版心	一版正中、折叠处的窄行	`\版心设置{版心}`；宽度 `\内容设置{版心比例}`	功能详解
6	鱼尾	版心中的分界符号	`\版心设置{上鱼尾，下鱼尾}`	

编号	古籍要素	含义	luatex-cn 对应	详见
				功能详解
7	版心书名	刻在版心上部的书名	`\title{...}`	功能详解
8	版心篇名／卷次	刻在版心中部的篇名或卷次	`\chapter{...}`	功能详解
9	页码	刻在版心下部的叶次	`\版心设置{页码={...}}`	命令索引
10	书耳	边栏外上角记篇名的小框	`\版心设置{书耳}`（v0.4.4 之后的新版本）	—
11	正文大字、行款	正文；「半叶 8 行，行 21 字」这类规格叫行款	`正文` 环境；`\内容设置{行数,每行字数}`	模板与自定义
12	夹注（双行小字）	夹在正文中的双行小字注	`\夹注{...}`、`\单行夹注{...}`	夹注与侧批
13	句读	字旁的断句圈点	`\开启句读模式`、`\句读设置`	句读
14	侧批	写在行间的批语	`\侧批{...}`	夹注与侧批
15	眉批	写在天头的批语	`\眉批{...}`；任意位置用 `\批注`	批注与眉批
16	钤印	藏书印等印章	`\印章[...]{图片}`	印章

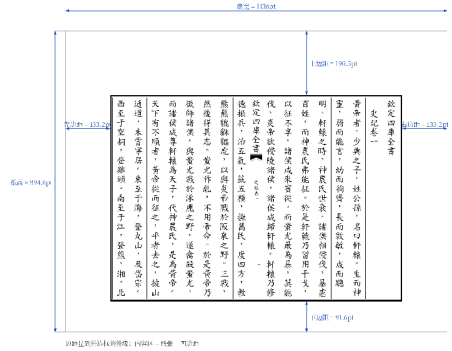
图里没有画出、但同样常用的要素：

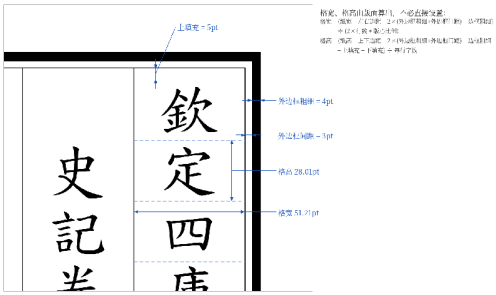
古籍要素	luatex-cn 对应	详见
抬头（敬称另起一行并高出）	`\抬头`、`\平抬`、`\单抬`、`\双抬`、`\挪抬`	抬头
牌记（刊刻者、刊刻时地的题记）	`\牌记{...}`	文本框
筒子页（一版对折成前后两个半叶）	默认开启；`\关闭分页` 关闭	命令索引

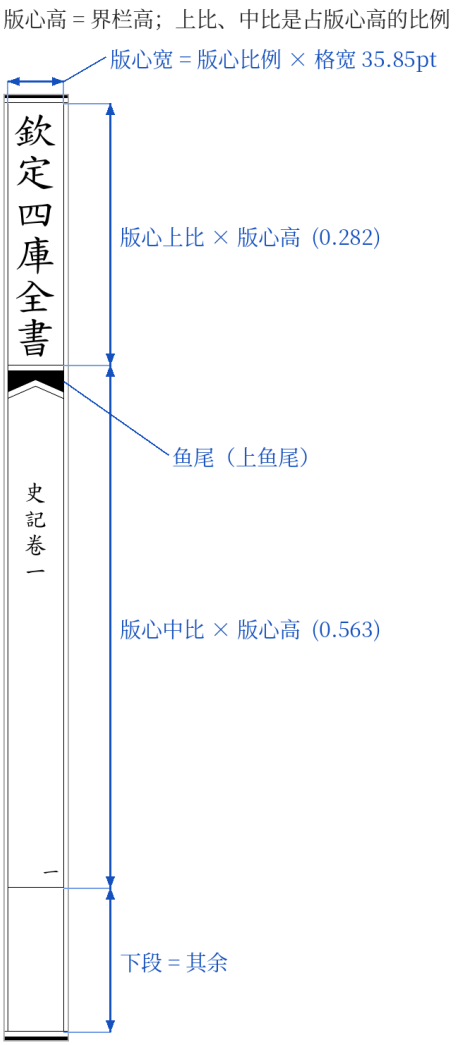
> 象鼻（版心上下端到鱼尾之间）的黑口、花鱼尾等变化，目前还没有专门参数。

参数示意图

常用几何参数在实际页面上的位置。图中每个尺寸都由脚本从排版结果里量出，并校验等于参数设定值，所以不会和代码脱节。







延伸阅读

· 维基百科：线装书

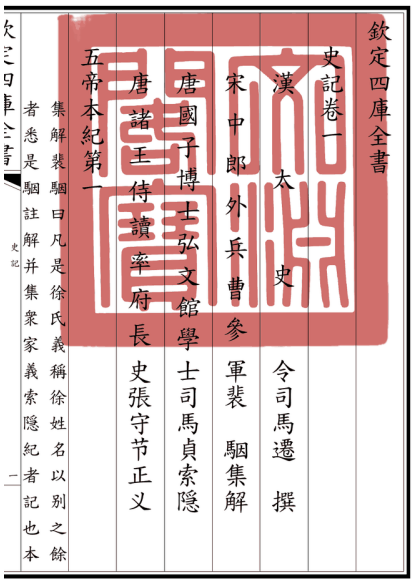
示例

以下示例展示了 `luatex-cn` 从传统古籍、现代竖排到现代横排的排版能力。每个示例都可以直接下载源码编译。

> 如果你还没跑通第一个示例，建议先看 快速入门。

1. 史记·五帝本纪 — 四库全书排版

文档类: `ltc-guji` | **模板**: `四库全书` / `四库全书彩色`



演示功能:

- 版心与单鱼尾（`\banxinSetup`）
- 复杂夹注（`\夹注{...}`）— 自动平衡、跨列跨页
- 印章绝对定位（`\印章[page=1]{...}`）

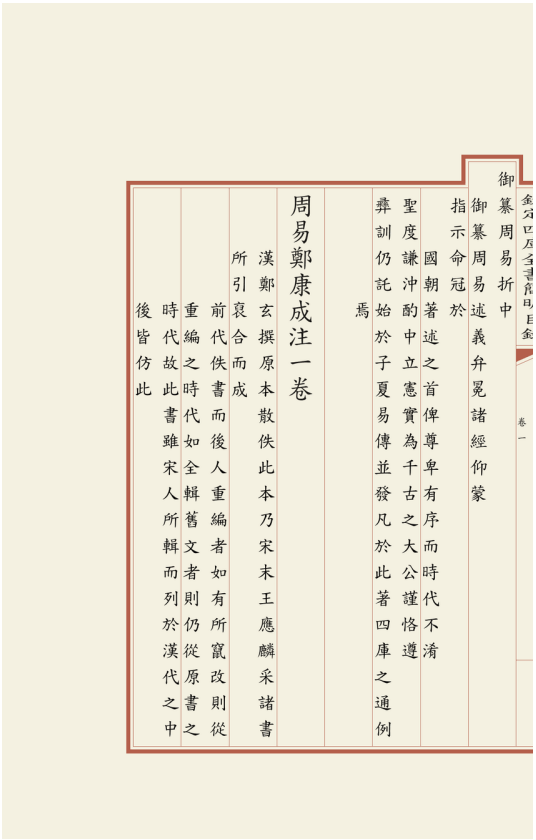
· 乌丝栏边框与排版缩进

查看源码 | 查看 PDF

---

2. 钦定四库全书简明目录 — 文渊阁本目录页

文档类: `ltc-guji` | **模板**: `SiKuMuLu`（示例自带 `.cfg`）



演示功能:

· 标准古籍目录布局（`\条目[层级]{...}` 命令）
· "八行二十一字"的行款限制
· 白口、四周双边、单鱼尾版式
· 注文双行小字（`\注{...}`）与文渊阁宝印叠加（`\印章[page=1, opacity=0.7]{...}`）

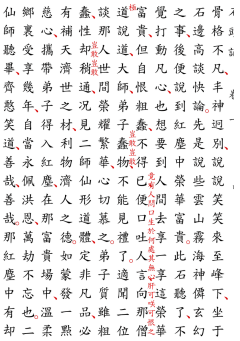
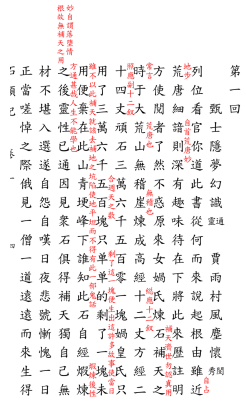
查看源码 | 查看 PDF

> 全书复刻版（多册）见 全书复刻/欽定四庫全書簡明目錄。

---

3. 红楼梦甲戌本 — 手抄批改本

文档类: `ltc-guji` | **模板**: 红楼梦甲戌本

第二页（标点）	第一页（眉批）
	

演示功能:

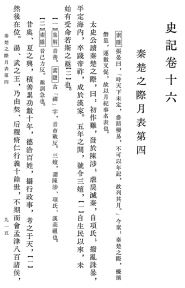
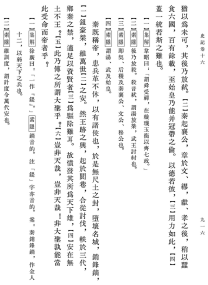
- 侧批（`\侧批{...}`）— 行间红色批注
- 批注（`\批注{...}`）— 页面浮动批注框
- 夹注（`\夹注{...}`）— 双行小字
- 句读模式（`\开启句读模式`）— 传统句/读标记
- 无鱼尾版心，底部页号

> 眉批（`\眉批{...}`）不在本示例中，见下方 全书复刻的《红楼梦》甲戌本。

查看源码 | 查看 PDF

4. 史记·秦楚之际月表 — 现代繁体竖排

文档类: `ltc-cn-vbook`（默认配置）

第一页	第二页
	

演示功能:

- 现代标点挤压（`\punctSetup{punct-squeeze=true}`）
- 禁则处理（`\punctSetup{kinsoku=true}`）
- 脚注系统（`\脚注{...}`）— 鹿角编号
- 段落缩进（`\begin{段落}[indent=2]`）
- 中国大陆标点风格

查看源码 | 查看 PDF

---

5. 论辩的魂灵（鲁迅）— 现代横排（clreq）

文档类: `article` + `\usepackage{luatex-cn}`（v0.4.0 起）

同一篇文章的两种排法对照：繁体部分用 `style=taiwan`（标点居中），简体部分用 `style=mainland` + `quote-style=curly`（原文的直角引号自动转为弯引号，嵌套先双后单）。

演示功能:

- 中西间距与四级换行禁则（`kinsoku-level=basic`）
- 连续标点缩减（2 → 1.5 字宽）、行首行尾标点挤压
- 段末孤字避免（前行借字）
- 段末行对齐 `\horiSetup{last-line=right}`（文末落款）
- 号数字号 `\字号{二号}`

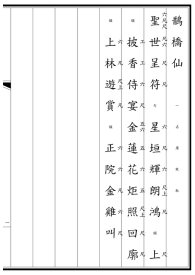
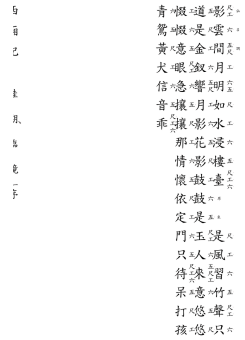
查看源码 | 查看 PDF | 📖 现代横排

---

6. 工尺谱 — 昆曲曲谱

文档类: `ltc-guji` | **模板**: 默认（印本）与 `红楼梦甲戌本`（手写本）

两份复刻，对应两种底本形态：

印本《鹊桥仙·圣世呈符》	手写本《西厢记·佳期·临镜序》
	

演示功能:

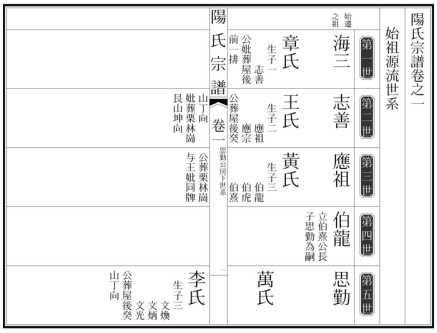
- 工尺主字与音字（`\工尺{影}{\音[板]{尺}\音[眼]{工}}`）— 板眼标记以红色小圆点区分
- 宫调注（`\宫调{仙吕調}`）、调名注（`\调名注{一名廣寒秋}`）
- 句读字（`\句读字`）与句读模式配合曲词断句
- 收紧行款（`\contentSetup{n-char-per-col=12, n-column=8}`）以匹配印本密度

查看源码 | 印本 PDF | 手写本 PDF

---

7. 阳氏宗谱 — 欧式（苏式）谱表

文档类: `ltc-guji` | **模板**: `欧式族谱`



复刻《陽氏宗譜》卷之一「始祖源流世系」首开印面：以世次横栏分格，每格右为讳名、左为配氏与葬向小注。

演示功能：

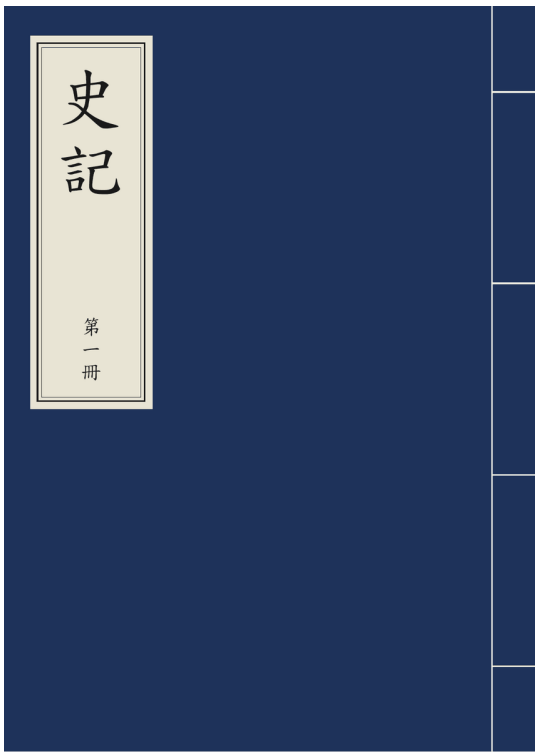
- **表格系统**（唯一的表格示例）：`\begin{表格}[分栏=6，栏高度=43.2pt，行填充=页]`、`\栏格式`、`\单元格`、`\换栏`
- 八角墨围反白（`\八角墨围反白[高度=3，填充]{第一世}`）作世次标签
- 样式覆盖（`\样式[格高=28pt，字号=28pt]{...}`）
- 免安装字体族（`\设置字体族{Jigmo}`）
- 调试开关（`\开启调试`，源码中注释保留）

查看源码 | 查看 PDF

---

8. 线装书封面 — 现代蓝色封面

文档类：`ltc-guji` | **模板**：无（内置命令）



中华书局、广陵书社等影印古籍常见的现代蓝色线装书封面（参照 CY/T 247—2021 图 1、图 2）：藏青封面、左上角米色签条、文武线双框。四页分别演示短书名 + 册次、卷次、长书名 + 册次卷次（签条自动增高）、小字随书名。

演示功能：

- `\线装书封面[册次=…，卷次=…]{书名}`（v0.4.3+）
- `小字位置=随后`、`订线` 预览开关
- `\pageSetup{paper-width=370mm, paper-height=260mm}`（16 开线装本对开尺寸）

查看源码 | 查看 PDF

全书复刻

`示例/` 下是单页或数页的功能演示；`全书复刻/` 放的是成册的完整复刻项目，页数多、编译慢。

项目	文档类	进度	独有演示
欽定四庫全書簡明目錄	`ltc-guji` + `ltc-guji-digital`	第一册（共 10 册 20 卷）	**数字化模式**（`\双列{\右小列{…}\左小列{…}}`、`\缩进`、`\换页`）、完整抬头体系（`\单抬` / `\平抬` / `\挪抬` / `\相对抬头`）、`\文本框`、`\校对页`

项目	文档类	进度	独有演示
紅樓夢·甲戌本	`ltc-guji`	凡例 + 第一回	**眉批**（`\眉批{...}`，35 处）、**改字**（`\改{...}`）

同一部书的语义版（`tex/`）与数字化版（`ditigal_tex/`、`*-digital.tex`）并存，可直接对照两种录入方式的差异。

功能对照表

功能	在哪个示例里看
版心与鱼尾	史记五帝本纪、四库简明目录（各模板不同）
夹注 `\夾注`	史记五帝本纪、四库简明目录、红楼梦、欧式族谱
侧批 `\側批`	红楼梦甲戌本
批注 `\批注`	红楼梦甲戌本
眉批 `\眉批`	仅全书复刻《红楼梦》甲戌本
印章 `\印章`	史记五帝本纪、四库简明目录
句读	红楼梦甲戌本、工尺谱
目录条目 `\条目`	四库简明目录
抬头（单抬/平抬/挪抬/相对抬头）	四库简明目录、全书复刻
文本框 `\填充文本框` / `\文本框`	史记五帝本纪、全书复刻
表格	欧式族谱
工尺谱	工尺谱（两份）
墨围 / 八角墨围反白	史记卷十六·现代、欧式族谱
样式 `\样式`	欧式族谱、全书复刻
改字 `\改`	仅全书复刻《红楼梦》甲戌本
现代标点与禁则	史记卷十六·现代
脚注 `\脚注`	史记卷十六·现代
数字化双列	仅全书复刻《四库简明目录》

功能	在哪个示例里看
横排 clreq	论辩的魂灵
线装书封面 `\线装书封面`	线装书封面

> **示例未覆盖的功能**：`ltc-tw-vbook` 文档类、线标（`\线标` / `\专名号` / `\书名号`）、装饰与圈点（`\装饰` / `\圈点` / `\发圈`）、缺字、旁注（`\旁注`）、牌记与悬浮文本框、插图页与换半页、排版坐标导出，以及横排的行间标号／行间注／均排／段落排式（示例只用到 `\字号` 与 `\horiSetup`）。这些能力目前只在 `test/regression_test/` 的测试文档里有用例。

如何运行示例

1. 下载或克隆项目仓库
2. 进入示例目录，如 `示例/史记五帝本纪/`
3. 运行 `lualatex 史记.tex`
4. 打开生成的 PDF 查看效果

```
cd 示例/史记五帝本纪
lualatex 史记.tex
```

所有古籍示例均基于历史文献扫描件进行数字重构，旨在展示 `luatex-cn` 对各种复杂竖排场景的适配能力。

模板使用与自定义

`luatex-cn` 的「模板」就是一份配置文件（`.cfg`），里面写 `\pageSetup`、`\contentSetup`、`\banxinSetup` 等设置命令。在文档类选项里写模板名即可加载。

内置模板

模板名称	文档类	说明
四库全书（`default`）	`ltc-guji`	经典官修书籍风格（黑白版），`ltc-guji` 的默认模板
四库全书彩色（`SiKuQuanShu-colored`）	`ltc-guji`	经典官修书籍风格（彩色版）
红楼梦甲戌本（`HongLouMengJiaXuBen`）	`ltc-guji`	手抄本风格，支持侧批与眉批
欧式族谱（`oushizupu`）	`ltc-guji`	族谱风格
（默认配置）	`ltc-cn-vbook`	现代竖排，中国大陆标点
（默认配置）	`ltc-tw-vbook`	现代竖排，台湾标点

现代竖排两类没有命名模板，各自加载默认配置，详见 现代竖排。

使用模板

在文档类选项中指定模板名称：

```
\documentclass[四库全书彩色]{ltc-guji}
\documentclass[红楼梦甲戌本]{ltc-guji}

% 也可以在导言区切换（中文名、英文名都行，v0.4.5+）
\gujiSetup{ template = 红楼梦甲戌本 }
```

自定义模板

方法 1：创建配置文件（推荐）

模板名 `<名>` 的查找顺序：先找文档目录下的 `<名>.cfg`，再找 TEX 路径中的 `configs/` 目录。

文档类	文件名
`ltc-guji`	`<名>.cfg` 或 `configs/luatex-cn-guji-<名>.cfg`
`ltc-cn-vbook` ／ `ltc-tw-vbook`	`<名>.cfg` 或 `configs/luatex-cn-vbook-<名>.cfg`

然后 `\documentclass[<名>]{ltc-guji}` 即可加载。

guji 模板

```
% MyTemplate.cfg（放在文档目录下）

\gujiSetup{ template = default }   % 先套用四库全书，再改下面几项

\pageSetup{
    paper-width = 1077.2pt,
    paper-height = 1077.2pt,
    margin-top = 226.8pt,
    margin-bottom = 113.4pt,
    margin-left = 25.5pt,
    margin-right = 25.5pt,
}

\contentSetup{
    n-column = 12,
    font-size = 30pt,
    line-spacing = 45pt,
    n-char-per-col = 18,
    border = true,
}

\banxinSetup{
    banxin-upper-ratio = 0.18,
    banxin-middle-ratio = 0.38,
    upper-yuwei = true,
    lower-yuwei = true,
}

\endinput
```

cn-vbook / tw-vbook 模板

模板在默认配置之后加载，只需写要改的项：

```
% mytemplate.cfg

\pageSetup{
    paper-width = 148mm,
    paper-height = 210mm,
    margin-top = 20mm,
    margin-bottom = 15mm,
    margin-inner = 18mm,
    margin-outer = 15mm,
    page-number-style = digits,
}

\contentSetup{
    font-size = 11pt,
    line-spacing = 13pt,
    grid-width = 13pt,
    grid-height = 13pt,
}

\punctSetup{punct-style=taiwan}

\endinput
```

方法 2：在文档中直接配置

在导言区直接写设置命令，书名、卷名用 `\title`、`\chapter`：

```
\documentclass{ltc-guji}

\contentSetup{
    n-column = 10,
    font-size = 24pt,
    border = true,
}

\title{我的书名}

\begin{document}
\begin{正文}
\chapter{第一章}
% ...
\end{正文}
\end{document}
```

> v0.4.5 起，`\gujiSetup` 和 `\defineGujiTemplate` 里除 `template` 外还可以直接写版面参数（`n-column`、`font-size`、`margin-top`、`border` 等），它们会自动交给

> `\pageSetup` / `\contentSetup` / `\banxinSetup` 中认识这个键的那个；三者都不认识的键只发警告、不报错。旧版 wiki 里的 `\gujiSetup{book-name=…}` 仍然无效（会得到这条警告）。
>
> ```
> \defineGujiTemplate{窄版}{ template = default, n-column = 5, font-size = 20pt, margin-top = 100pt }
> \gujiSetup{ template = 窄版 }
> ```

在模板上改参数要注意 v0.4.5+

古籍模板（四库全书、红楼梦甲戌本等）都设了 `行数` 和 `每行字数`，格宽、格高由版面算出，所以在模板上直接写 `格高` / `格宽` 不起作用：想要更大的字格，应减小 `每行字数` 或 `行数`；想直接给格高，先设 `每行字数 = 0`。同样，有版心时不要用 `版面行数`，字号也不要大于格高。这些情况编译时都会给出提醒（模板自己设的值不会触发）。完整列表见 命令索引：参数之间的覆盖关系。

常用配置项

下表默认值是不加载模板时的代码默认值；各模板会覆盖其中一部分。完整参数见 命令索引。

配置命令	参数	说明	默认值
`\gujiSetup`	`template`	模板名称（中文名或英文名）	—
`\contentSetup`	`n-column`	每半页列数（0 = 自由列宽）	8
	`n-char-per-col`	每列字数（0 = 自动）	0
	`font-size`	字号	12pt
	`grid-width` / `grid-height`	格宽／格高（古籍模板里由 `行数`、`每行字数` 算出，直接写不生效）	1.5em／1.2em
	`line-spacing`	行距（natural 模式）	`\baselineskip`
	`border`	显示丝栏	false
	`border-color`	丝栏颜色	black
	`font-color`	字体颜色	(空)
	`layout-mode`	排版模式：`grid` / `natural`	`grid`
	`horizontal-align`		`right`

配置命令	参数	说明	默认值
		行块水平位置：`right`／`center`／`left`（v0.4.5+）	
`\pageSetup`	`paper-width`／`paper-height`	纸张宽高	210mm／297mm
	`margin-*`	页边距	0pt
	`twoside`	对页（奇偶页镜像边距）	false
	`page-number-style`	页码样式	(空)
`\banxinSetup`	`banxin-upper-ratio`	版心上部比例	0.28
	`upper-yuwei`／`lower-yuwei`	上／下鱼尾	true
	`book-ear`	书耳（v0.4.5+）	false
`\punctSetup`	`punct-style`	标点风格	`taiwan`
	`punct-squeeze`	标点挤压	`true`
	`kinsoku`	禁则处理	`true`

下一步：查看 调试模式 或 功能详解

字体设置 (Fonts)

`luatex-cn` 提供了智能的跨平台字体探测机制，以及强大的多字体递补（Fallback）功能，确保生僻字和缺字情况能够被优雅处理。

1. 自动字体探测

在默认情况下，使用 `ltc-guji`（或 `ltc-cn-vbook` / `ltc-tw-vbook`）文档类会自动根据操作系统选择最合适的已安装字体：

- **Windows**: 优先使用中易宋体 (SimSun)、中易黑体 (SimHei) 等。
- **macOS**: 优先使用宋体-简 (Songti SC)、苹方-简 (PingFang SC) 等。
- **Linux**: 优先使用 Fandol 系列或 Noto CJK 系列。

如果需要手动触发自动探测并应用：

```
\ApplyAutoFont
```

中文字体名称支持（v0.2.2+）

v0.2.2 起，字体探测支持使用简体或繁体中文字体名称：

```
\设置字体{宋体}          % 简体中文名称
\设置字体{標楷體}        % 繁体中文名称
\设置字体{思源宋体}      % 开源字体中文名
```

系统会自动将中文字体名称映射到对应的系统字体。

2. 标准字体设置

如果你只需要设置单一字体，可以使用标准命令或别名：

```
\设置字体{KaiTi}
% 等同于 \setmainfont{KaiTi}
```

3. 设置字体族 (Font Family & Fallback)

v0.1.5 新增了**字体族**（Font Family）的概念。通过设置字体族，系统会建立一个“字体递补链”。当主字体缺少某个字符（如生僻字）时，会自动到链中的下一个字体里查找。

`\设置字体族`

该命令接受一个字体列表（逗号分隔）。列表中的第一个字体将被设为主字体，其余字体作为后备。

```
% 将 "Source Han Serif SC" 设为主字体
% 如果缺字，则尝试 "SimSun"，再缺字尝试 "SimHei"
\设置字体族{Source Han Serif SC, SimSun, SimHei}
```

英名别名：

```
\setFontFamily{Source Han Serif SC, SimSun, SimHei}
```

注册表别名与字体文件（免安装）

`\设置字体族` 还接受**注册表别名**和**字体文件名/路径**，三种规格可混用，字体无需装进系统：

```
\设置字体族{Jigmo}                      % 别名：自动展开为 Jigmo + Jigmo2 回退链
\设置字体族{fonts/MyFont.ttf, TW-Kai}   % 文件路径与字体名混搭
\设置字体族{TW-Kai, Jigmo2.ttf}         % 裸文件名也可以
```

每个字体按三级顺序解析：

1. **系统已装** → 按名字使用系统字体；
2. **本地文件** → 依次查 `./fonts/`、文档目录、kpse（覆盖 TEXMFHOME 与 OSFONTDIR）；
3. **都没有** → 报错中断编译，并给出下载脚本的完整命令（见 §4），避免缺字渲染成豆腐块。

系统与本地文件可逐字体混搭——例如系统装了 Jigmo 而本地只有 `Jigmo2.ttf`，别名 `{Jigmo}` 仍能正确组链。目前注册表内置别名：`Jigmo`（Jigmo + Jigmo2）。

> 对于竖排文档，手动设置字体时，通常无需额外添加竖排特性（`RawFeature={+vert}`），`luatex-cn` 底层通常会自动处理。但在某些极端自定义场景下，你可能需要手动指定 `Renderer=Harfbuzz` 或相关特性。

4. 免安装字体：下载脚本

不想（或没有权限）把字体装进系统字体库？LuaTEX 本身就支持**按文件加载字体**——只要 `.ttf` / `.otf` 文件在文档目录、`./fonts/` 子目录或 TEXMFHOME 里，写文件名即可使用。

仓库自带下载脚本（需要 Python 3），按 `scripts/font-manifest.json` 清单获取推荐的开源字体（TW-Kai 楷体系列、Jigmo 字面、京華老宋体、朱雀仿宋等，均校验 SHA-256）：

```
# 方式一：安装到 TeX 个人目录（推荐，一次下载全局可用）
# 位置为 TEXMFHOME/fonts/truetype/luatex-cn/，免管理员权限、不进系统字体库
python3 scripts/download_fonts.py --all --user

# 方式二：下载到文档项目目录（文档与字体可整体打包分发）
python3 scripts/download_fonts.py --all --dest ./fonts
```

之后在文档中直接按**文件名**引用：

```
% 字体在 TEXMFHOME 或文档同目录时，裸文件名即可
\setmainfont{TW-Kai-98_1.ttf}

% 字体在子目录时，用 Path 指定
\setmainfont{TW-Kai-98_1.ttf}[Path=./fonts/]
```

> [!NOTE]
>
> - `--user` 安装后如需按**字体名**（而非文件名）引用，再执行一次 `luaotfload-tool --update` 更新字体索引。
> - 朱雀仿宋等来自 npm woff2 切片的字体，脚本会自动下载并用 fonttools 合并为单个 TTF（竖排特性保留），需要先 `pip install fonttools brotli`。

☞ 下一步：查看 改字与装饰 或 功能详解。

功能详解

`luatex-cn` 提供了丰富的中文排版功能：从核心的竖排引擎、古籍版式元素，到按 clreq 实现的现代横排管道。本页面概览了宏包的主要功能模块。逐 clreq 条款的符合状态与有意偏离见 clreq 符合度矩阵（119 条 PDF 度量断言逐条核验）。

1. 核心排版引擎

基于 LuaTEX 的网格布局系统，实现高精度的竖排文本流。

- **RTT 流向**：原生支持从上到下、从右到左的流向。
- **网格定位**：字符精确对齐到定义的网格中。
- **分页与换列**：自动处理复杂的页面分割和列流转。

2. 竖排：四个文档类

文档类	用途	详情
`ltc-guji`	传统古籍排版，支持版心、鱼尾、丝栏等传统元素	快速入门
`ltc-guji-digital`	古籍数字化录入：一行源码 = 一个竖排列，`\双列{\右小列{…}\左小列{…}}` 手工分栏，精确复刻底本版面	源码示例
`ltc-cn-vbook` (v0.2.8+)	现代中国大陆竖排书籍，简洁设计，内置脚注和标点系统	ltc-book
`ltc-tw-vbook` (v0.2.8+)	现代台湾竖排书籍，居中标点	ltc-book

> `ltc-guji-digital` 的命令见 数字化排版。旧名 `ltc-book` 已不再提供，请用 `ltc-cn-vbook` ／ `ltc-tw-vbook` 。

3. 横排：`\usepackage{luatex-cn}`

按 W3C clreq 实现的现代中文横排管道，主包名 `luatex-cn` 即其统一入口（内部实现名 `luatex-cn-hori` 仍可直接加载）。它是宏包而非文档类，加载到 `article` 等任意横排文档类之上即可；规则内核与竖排共用 `tex/shared/` 下的同一份实现。

```
\usepackage{fontspec}
\setmainfont{Source Han Serif SC}[Script=CJK, Language={Chinese Simplified}]
\usepackage[style=taiwan]{luatex-cn}
```

- **中西间距**：1/4 em，可挤至 1/8、拉至 1/2，含 clreq 规定的三类例外。
- **标点宽度调整**：中国大陆式／台式／不调整三套预设，连续标点缩减，行末标点半字宽，行尾点号悬挂（可选）。
- **换行控制**：四级禁则（none/basic/gb/strict）与符号分离禁则；行内按 clreq 优先顺序做二次分配（挤压 7 级／拉伸 2 级 + 兜底均分）。
- **行间标号与行间注**：`\专名`、`\书名号甲`、`\着重`；`\拼音`／`\对照` 词对齐标音。
- **段落与页面**：段首缩排两字、段末孤字避免、页首页尾孤行控制、凸排与段落缩排环境、段末行四种对齐、号数字号 `\字号`。
- **引号体例转换**：默认不改来稿，可按风格自动或显式转为弯引号／直角引号。

规则实现由 `test/clreq_test.py` 解析 PDF 内容流逐条度量断言，已并入 CI。详见 **现代横排**。

> ⚠ 需要 LuaTeX，且与 `luatexja`（`ctex` 在 LuaLaTeX 下会载入）互斥。

4. 标点处理

- **标点系统** (v0.2.7+)：现代标点挤压、禁则处理、竖排引号替换，支持中国大陆/台湾风格。竖排标点宽度调整为上下文相关（clreq），字面分布按字形墨迹度量锚定，不依赖字体自身的字面设计。v0.4.0 起另有：行尾点号悬挂（`punct-hanging`）、中西混排间距（1/4em，`western-space`）、西文横置（`\横置`）、短串横排入一格（`\中横排`，縱中横）、连接号/间隔号/分隔号扩类（中国大陆式间隔号半字宽）。
- **句读**：传统句读标记，三种模式（normal/judou/none）随时切换。

5. 古籍版式元素

完整支持传统装帧所需的视觉元素：

- **版心与鱼尾**：支持单鱼尾、双鱼尾、乌丝栏等。
- **筒子页**：支持电子阅览与传统装订的纸张分割。
- **边框系统**：支持四周双边、内框等。
- **书耳** (v0.4.5+)：`\banxinSetup{书耳=true}` 在边栏外上角刻一个记篇名的小框，文字默认取当前 `\chapter`，可设位置（左／右／两侧）、宽高、字号。参数见 命令索引「版心」一节 (#87)。

6. 注释与批改系统

提供多种维度的文本注释方式：

- **夹注**：正文中的双行小字，支持自动平衡算法。
- **侧批**：行间的垂直注释。
- **批注**：页面任意位置的浮动注释框。
- **眉批** (v0.2.3+)：页面顶部的批注。
- **脚注** (v0.2.7+)：段末注和页下注，支持鹿角/带圈编号。
- **目录条目**：`\条目` (v0.2.5+)，支持层级缩进。

7. 样式控制 (v0.2.8+)

`\Style`（或 `\样式`）命令允许在正文中临时覆盖字号、字体、字体颜色、格高、间距等样式属性，而不影响当前布局模式。

```
% 行内用法
\样式[字号=16pt，字体颜色=red]{重要文本}

% 环境用法（适合多行内容）
\begin{样式设置}[字号=14pt，间距=0.5em]
  多行样式化文本……
\end{样式设置}
```

支持参数：

参数	中文别名	说明
`font-size`	`字号`	覆盖字号

参数	中文别名	说明
`font`	`字体`	覆盖字体
`font-color`	`字体颜色`	覆盖字色
`grid-height`	`格高`	覆盖格高
`spacing-top`	`上间距`	上方附加间距
`spacing-bottom`	`下间距`	下方附加间距
`spacing`	`间距`	同时设置上/下间距

样式通过内部注册表管理，支持跨页保持和嵌套使用。详见 命令索引。

8. 封面 (v0.2.8+)

`Cover`（或 `\begin{封面}`）环境用于排版封面页。它会自动关闭边框、外边框和版心，并支持设置页面背景色。

```
\begin{封面}[底色=200,200,200]
  \行[对齐=center，字号=24pt]{书名}
  \行[对齐=center，字号=14pt]{作者}
\end{封面}
```

参数	中文别名	说明
`background-color`	`底色`	封面背景色（颜色名或 RGB）

封面环境内部套用了 `SinglePage`（单页）+ `BodyText`（正文），无需手动处理分页。背景色仅作用于封面页，不会泄漏到后续页面。

线装书封面 (v0.4.3+)

`\线装书封面[键值]{书名}`（`\ThreadBoundCover` ／ `\線裝書封面`）输出一页现代蓝色线装书封面（中华书局、广陵书社等影印古籍常见样式，参照 CY/T 247—2021《线装书籍要求》图 1、图 2）：藏青色封面，左上角米色签条，签条内为文武线双框（外粗内细），书名大字自上而下。签条有最小高度（默认页高 50%），书名长了随之增高，超出页面时书名与小字按比例缩小。封面上只有签条，不印作者、出版者。

```
\线装书封面[册次=第一冊]{史記}
\线装书封面[卷次=卷一]{李白詩集}
\线装书封面[册次=第九冊，卷次=卷二十一至二十三]{欽定四庫全書簡明目錄}
```

```
\线装书封面[册次=第一册，小字位置=随后]{四書章句集注}
\线装书封面[册次=第一册，订线]{史記}    % 预览：画出订线
```

键（英 / 简）	默认	说明
`volume` / `册次`	空	如「第一冊」
`juan` / `卷次`	空	如「卷一」；与册次同时给出时并排两列（册次在右）
`small-position` / `小字位置`	`bottom`（底部）	`follow`（随后）：紧随书名
`binding-thread` / `订线`	关	画出订线，仅供预览；印制封面时不要开
`thread-holes` / `订线眼数`	4	六眼订法写 6
`thread-distance` / `订线距`	页宽 × 0.08	订线距页面右边
`thread-color` / `订线颜色`	`236, 236, 230`	
`background-color` / `底色`	`30, 50, 90`	封面色（RGB 0–255 或颜色名）
`label-color` / `签条底色`	`232, 228, 212`	签条纸色
`ink-color` / `墨色`	`25, 25, 25`	书名、小字、框线颜色
`label-width` / `签条宽`	页宽 × 0.23	
`label-left` / `签条左距`	页宽 × 0.05	签条左缘距页面左边
`label-top` / `签条上距`	页高 × 0.04	签条上缘距页面上边
`label-min-height` / `签条最小高`	页高 × 0.5	
`title-font-size` / `书名字号`	签条宽 × 0.56	
`small-font-size` / `小字字号`	书名字号 × 0.3	

长度类键不给（或给 `0pt`）时按页面比例自动取值；页面尺寸沿用文档设置（对开模式下取半页宽）。完整示例见 `示例/线装书封面/`。

9. 线标——专名号与书名号 (v0.2.8+)

`\LineMark`（或 `\线标`）在字符旁绘制装饰线条（直线或波浪线），用于标示**专名号**和**书名号**。与传统 Unicode 字符不同，线标使用 PDF 图形命令绘制，精度更高、样式更灵活。

```
% 专名号（直线）
\专名号{孔子}

% 书名号（波浪线）
\书名号{红楼梦}

% 自定义样式
\线标[类型=wavy，颜色=blue，振幅=large]{强调内容}
```

快捷命令：

命令	中文别名	效果
`\ProperNameMark`	`\专名号` `\下划线`	直线标记
`\BookTitleMark`	`\书名号` `\波浪线`	波浪线标记

支持参数：

参数	中文别名	说明	默认值
`type`	`类型`	`straight`（直线）或 `wavy`（波浪）	`straight`
`color`	`颜色`	线条颜色	`black`
`offset`	`偏移`	线条与文字的距离	`0.6em`
`amplitude`	`振幅`	波浪幅度：`small` `medium` `large`	`medium`
`line-width`	`线宽`	线条粗细	`0.8pt`
`style`	`风格`	`standard`（紧凑）或 `cursive`（手写感）	`standard`

线标在夹注、侧批等小字环境中会自动缩放适配。

10. 布局控制与排版命令

- **强制分列**：`\行`（或 `\Column`）把内容单独排成一列。
- **末列控制**：`\末行`（或 `\LastColumn`）将内容排在当前页的最后一列。
- **灵活列宽**：`\行[width=2cm]{...}` (v0.2.7+) 支持不等宽列布局。

- **分页控制**：支持标准的 `\newpage` 和 `\clearpage`。
- **换半页** (v0.3.6+)：`\换半页` / `\NewHalfPage` 在筒子页模式下跳到下一半页。
- **插图页** (v0.3.6+)：`\插图页` / `\IllustrationPage` 隐藏丝栏保留边框，用于插入全页插图 (fix #29)。
- **自然排版** (v0.2.7+)：`layout-mode=natural` 非等距字符排列。
- **自由列宽模式** (v0.2.8+)：`n-column=0` 时启用可变列宽布局，列宽随内容自动调整（`ltc-cn-vbook` / `ltc-tw-vbook` 默认启用）。
- **对页补页** (v0.4.4+)：`\pageSetup{对页补页=true}`（`facing-pages`）让每章从奇数页（右页）起：章（`\chapter`）落在偶数页顶端时，前面自动补一页。默认关闭；只认 `正文` 块内的 `\chapter`；筒子页模式下不自动补。`\补空白页`（`\InsertBlankPage`）在任意处强制补一页，连写两次补两页。补出的页默认带版框、版心／页眉页码，`补页样式=空白`（`blank-page-style=blank`）则整页空白，`\pageSetup` 与 `\补空白页[补页样式=空白]` 都可写 (#176)。
- **居中页** (v0.4.5+)：`\begin{居中页}` … `\end{居中页}`（`CenteredPage`）在新的一页上排一段正文，整块行在内容区内水平居中，适合扉页、卷标题页；各行的上下位置用 `\行[align=center]` 控制。一般正文也可用 `\contentSetup{horizontal-align=center}`（`横对齐=居中`，可选 `right`／`center`／`left`，默认 `right`）(#175)。

```
\begin{居中页}
  \行[align=center]{王又朴集}
  \行[align=center]{上}
\end{居中页}
```

11. 装饰与辅助功能

- **印章**：在页面任意位置添加馆藏印或私印，支持透明度。
- **字体设置**：支持自动探测、字体族递补机制。v0.2.2 起支持简繁体中文字体名称。
- **文本框**：支持网格布局、自动扁平化压缩及自由填充。透明度控制 (v0.3.6+)。
- **牌记** (v0.3.6+)：`\牌记` / `\colophon` 不透明双层边框悬浮文本框。
- **缺字**：`\缺字`（`\MissingChar`）插入白方框「□」作缺字占位，`\缺字[3]` 连缺三字。
- **改字与装饰**：古籍勘误改字（\改）与字符级装饰（\装饰）。
- **抬头** (v0.2.6+)：敬语抬头格式（单抬、双抬、三抬、平抬）。

12. 架构与调试功能（v0.2.0+）

- **样式注册表**：多属性样式存储（颜色、字号、网格高度），支持跨页样式保持。
- **元数据系统**：集中化书籍元数据管理（`\title`、`\publisher`、`\ChapterMarker`）。
- **调试模式**：`\LtcDebugOn` 开启后显示页面网格坐标，`\LtcShowGrid` 显示坐标网格。

- **排版坐标导出**：`\开启排版导出`（`\enableLayoutExport`）将排版结果导出为 JSON 文件，包含每个字符的精确坐标、页面/列/行信息、夹注和侧批数据。
- **数字化排版** (v0.3.0+)：专为古籍数字化设计的“布局模式”，提供 `\缩进`、`\双列` 等显示控制命令，配合 `converter.py` 实现语义稿到数字化稿的转换。
- **插件架构**：标准化插件 API，模块统一实现 initialize/layout/render 三阶段接口。

13. 模板系统

预设了多种经典风格，开箱即用。详见 模板与自定义。

- `四库全书`：清代内府排版风格。
- `四库全书彩色`：彩色版四库全书风格。
- `红楼梦甲戌本`：手抄本批改风格。
- `欧式族谱`：族谱风格。
- `ltc-cn-vbook` ／ `ltc-tw-vbook` 默认配置：现代竖排，分别为中国大陆／台湾标点。

下一步：查看 模板与自定义 | 调试模式 | 示例 | 命令索引

现代竖排文档类 (v0.2.7+)

luatex-cn 提供两个面向现代竖排中文书籍的文档类，与 `ltc-guji`（古籍排版）不同，它们采用简洁的现代设计：无版心、无鱼尾、无丝栏，适合当代出版物的竖排需求。

文档类	标点风格	说明
`ltc-cn-vbook` (v0.2.8+)	中国大陆	简体中文竖排书籍
`ltc-tw-vbook` (v0.2.8+)	台湾	繁体中文竖排书籍

基本用法

```
\documentclass{ltc-cn-vbook}
\setmainfont{Noto Serif SC}

\begin{document}
\begin{正文}
天地玄黄，宇宙洪荒。日月盈昃，辰宿列张。
寒来暑往，秋收冬藏。闰余成岁，律吕调阳。
\end{正文}
\end{document}
```

模板系统

两个文档类各自加载一份默认配置，共用 `configs/luatex-cn-vbook-common.cfg`：16 开（170×240mm），14pt 字号，自由列宽（`n-column=0`），对页（`twoside=true`，书脊侧 24mm、翻口侧 16mm），运行页眉与页码，脚注鹿角编号。

文档类	默认配置文件	标点风格
`ltc-cn-vbook`	`luatex-cn-vbook-default.cfg`	中国大陆（`punct-style=mainland`）
`ltc-tw-vbook`	`luatex-cn-tw-vbook-default.cfg`	台湾（`punct-style=taiwan`）

文档类选项中写模板名会再加载 `<名>.cfg`（文档目录）或 `configs/luatex-cn-vbook-<名>.cfg`，见 模板与自定义。目前没有内置的命名模板（旧版 wiki 提到的 `中华书局` 模板并不存在）。

与 guji 的区别

特性	cn-vbook / tw-vbook	guji
设计风格	现代简洁	传统古籍
版心/鱼尾	无	支持
边框/丝栏	无	支持
布局模式	自由列宽（`n-column=0`）	固定网格（`n-column=8`）
标点默认	现代标点 (normal)	可选句读/正常
脚注	内置支持	需手动加载
页码	自动竖排	版心中缝

标点风格配置

`ltc-cn-vbook` / `ltc-tw-vbook` 默认启用现代标点系统。详见 标点系统。

台湾风格（`ltc-tw-vbook` 默认）

```
\documentclass{ltc-tw-vbook}
% 标点居中，这是 tw-vbook 的默认设置
```

中国大陆风格（`ltc-cn-vbook` 默认）

```
\documentclass{ltc-cn-vbook}
% 或在任一类中显式指定：
\punctSetup{punct-style=mainland}
```

句读模式

虽然现代竖排文档类默认使用现代标点，但也可以切换到句读模式：

```
\documentclass{ltc-cn-vbook}
\开启句读模式  % 切换到句读模式

\begin{document}
\begin{正文}
天地玄黄，宇宙洪荒。
% 逗号变为右下角小点，句号变为右下角小圈
```

```
\end{正文}
\end{document}
```

双面镜像页边距 (v0.3.2+)

竖排书在右侧装订：**奇数页是对开的左页**，书脊在它右边；偶数页是右页，书脊在它左边。vbook 默认启用双面模式（`twoside=true`），奇偶页自动镜像内外边距：

- **内边距**（`margin-inner`）：书脊侧边距（奇数页=右边距，偶数页=左边距），默认 `24mm`
- **外边距**（`margin-outer`）：翻口侧边距（奇数页=左边距，偶数页=右边距），默认 `16mm`

```
\pageSetup{
    twoside = true,
    margin-inner = 24mm,   % 书脊侧
    margin-outer = 16mm,   % 翻口侧
}
```

> 读哪一组边距由 `twoside` 决定（v0.4.3+）：`twoside=true` 只读 `margin-inner` / `margin-outer`，`twoside=false` 只读 `margin-left` / `margin-right`；设置了被忽略的一组会收到警告。`twoside=false` 时页眉页码固定在左侧，不随奇偶页交替。

> **v0.4.4 方向修正**：此前 `margin-inner` 按横排习惯放在奇数页左侧，实际落在翻口一侧。现在按书脊算，vbook 默认值同时互换为 inner 24mm／outer 16mm，默认版面不变。自己写了 `margin-inner ≠ margin-outer` 的文档左右会对调，把两个值互换即可。古籍模板的 inner／outer 相等，不受影响。

对页固定内容区 (v0.4.4+)

想让奇偶页的正文落在同一位置（类似 InDesign／Word 按上下左右边距固定内容区），用 `mirror-margins = false`：

```
\pageSetup{
    twoside = true,
    mirror-margins = false,     % 镜像边距 = false
    margin-left = 22mm, margin-right = 22mm,
    binding-offset = 8mm,       % 装订边距（要装订时加）
```

```
    header-anchor = content,    % 页眉锚点 = 内容区
}
```

参数	中文键	默认值	说明
`mirror-margins`	`镜像边距` / `鏡像邊距`	`true`	设为 `false` 时，`twoside = true` 也读 `margin-left` / `margin-right`，内容区在奇偶页同一位置；书眉、页码仍奇偶交替
`binding-offset`	`装订边距` / `裝訂邊距`	`0pt`	加在书脊一侧：奇数页加在右边，偶数页加在左边；`twoside = false` 时加在右边。三种边距模式都生效

三种边距模式：

`twoside`	`mirror-margins`	读取的边距	适用
`true`	`true`（默认）	`margin-inner` / `margin-outer`，奇偶镜像	装订成书的对开版式
`true`	`false`	`margin-left` / `margin-right`，每页相同	内容区固定、书眉页码仍交替
`false`	—	`margin-left` / `margin-right`，每页相同	不装订打印、屏幕阅读

运行页眉 (v0.3.2+)

vbook 支持运行页眉，自动在奇偶页交替显示章节名和书名。

配置

```
\pageSetup{
    header-enabled = true,    % 启用页眉
    header-font-size = 9pt,   % 页眉字号
    header-y = 52mm,          % 页眉纵向位置
}
```

显示规则

页面	页眉内容	位置
奇数页（左页）	章节名（`\chapter{...}`）	左侧（翻口侧）
偶数页（右页）	书名（`\title{...}`）	右侧（翻口侧）

页眉默认在外侧（翻口侧）上角，页码在外侧下角。`header-position = inner`（`页眉位置 = 内侧`）可把页眉移到书脊侧（v0.4.4+）。

页眉锚点 (v0.4.4+)

默认 `header-anchor = paper`（`页眉锚点 = 纸边`）：页眉、页码按 `header-y` ／ `page-number-x` ／ `page-number-y` 从纸边量。改了边距后要手动调这几个值。

`header-anchor = content`（`页眉锚点 = 内容区`）让页眉、页码跟着内容区走：放在内容区外侧 `header-sep`（`页眉间距`，默认 `6mm`）处，页眉顶端对齐内容区上边，页码底端对齐内容区下边。此时 `header-y` ／ `page-number-y` 若写了，就作为相对这两条边的偏移。

跳过页眉

在标题页等不需要页眉的页面使用 `\skipHeader`：

```
\skipHeader
\卷名{史記卷六}
\章节名{秦始皇本紀第六}
```

段落与首行缩进 (v0.3.2+)

vbook 支持段落首行缩进，通过 `\contentSetup` 或 `StyleEnv` 环境配置：

```
\contentSetup{
    first-indent = 2,   % 首行缩进2格
}
```

布局引擎自动检测段落边界（通过 `\par` 或空行），对每段首行应用缩进。

也可在 `\Style` 中临时设置：

```
\Style[first-indent=0]{不缩进的段落内容}
```

卷名与章节名命令 (v0.3.2+)

vbook 提供专用的标题命令：

命令	功能	示例
`\卷名{...}`	卷名标题（24pt，上间距48pt）	`\卷名{史記卷六}`
`\章节名{...}`	章节名标题（20pt，首行缩进2字）	`\章节名{秦始皇本紀第六}`

`\章节名` 会自动调用 `\chapter`，同步更新页眉和版心中的章节标题。

脚注系统

现代竖排文档类内置脚注支持，提供段末注和页下注两种模式。

基本用法

```
\documentclass{ltc-cn-vbook}
\脚注设置{number-style=lujiao, indent=2}

\begin{document}
\begin{正文}
天地玄黄\脚注{此为千字文首句}，宇宙洪荒\脚注{天地初开之意}。
\输出脚注  % 在段落末尾输出所有脚注
\end{正文}
\end{document}
```

配置参数

通过 `\footnoteSetup` (或 `\脚注设置`) 配置：

参数	说明	默认值	可选值
`mode`	脚注模式	`endnote`	`endnote` (段末注), `page` (页下注)
`number-style`	编号样式	`lujiao`	`lujiao` (【一】), `circled` (①)
`separator`	各条之间的分隔	`blank`	`blank` （1em 空白），`none`
`font-size`	脚注字号	`0.8em`	任意尺寸
`font-color`	脚注颜色	(继承)	任意颜色
`indent`	悬挂缩进（格数）	`2`	整数

参数	说明	默认值	可选值
`marker-height`	编号占用格数	`2`	整数
`spacing`	间距	`0.5em`	任意尺寸

命令一览

命令	中文别名	说明
`\Footnote{内容}`	`\脚注{内容}`	添加脚注
`\FlushFootnote`	`\输出脚注`	输出段末注
`\FootnoteParagraph{内容}`	`\脚注段落{内容}`	无编号脚注段落（如"索隱曰…"）
`\footnoteSetup{...}`	`\脚注设置{...}`	配置脚注参数
`\ResetFootnoteCounter`	`\重置脚注计数`	重置脚注编号

页码

现代竖排文档类自动显示竖排页码。启用 `twoside` 时，页码默认在外侧（翻口侧）下角，奇偶页交替。

页码配置

通过 `\pageSetup` 配置：

参数	说明	默认值
`page-number-style`	页码样式	`digits`
`page-number-font-size`	页码字号	`9pt`
`page-number-position`	页码位置：`outer` 翻口侧、`inner` 书脊侧、`center` 页脚横向居中（中文键 `页码位置 = 外侧 / 内侧 / 居中`，v0.4.4+）	`outer`
`header-position`	页眉位置：`outer` 翻口侧、`inner` 书脊侧（`页眉位置 = 外侧 / 内侧`，v0.4.4+）	`outer`
`header-anchor`	页眉页码的定位基准：`paper` 纸边、`content` 内容区（`页眉锚点 = 纸边 / 内容区`，v0.4.4+）	`paper`

参数	说明	默认值
`header-sep`	`header-anchor = content` 时，页眉页码距内容区外边的距离（`页眉间距`，v0.4.4+）	`6mm`
`page-number-x`	页码距页边的水平距离（`页码横移`）	`3mm`
`page-number-y`	页码距页底的垂直距离（`页码纵移`）；`header-anchor = content` 时为相对内容区下边的偏移	`52mm`
`header-y`	页眉距页顶的垂直距离（`页眉纵移`）；`header-anchor = content` 时为相对内容区上边的偏移	`52mm`

`page-number-position` 只移动页码，`header-position` 只移动页眉。

> v0.4.4 起「内侧／外侧」按真正的装订方向（书脊）算，默认改为 `outer`，默认版面与之前相同。之前写了 `页码位置 = 内侧` 的文档，现在要写 `外侧` 才是原来的位置。

页码样式：

- `digits`：逐位中文数字（如 一二三）
- `chinese`：传统中文数字（如 一百二十三）
- `arabic`：阿拉伯数字
- `none`：不显示页码

自定义模板

在文档目录下创建 `<模板名>.cfg`（或放到 TEX 路径中的 `configs/luatex-cn-vbook-<模板名>.cfg`）。模板在默认配置之后加载，只写要改的项：

```
% mytemplate.cfg
\pageSetup{
    paper-width = 148mm,
    paper-height = 210mm,
    margin-top = 20mm,
    margin-bottom = 15mm,
    margin-inner = 18mm,
    margin-outer = 15mm,
    page-number-style = digits,
}

\contentSetup{
    font-size = 11pt,
    line-spacing = 13pt,
```

```
    grid-width = 13pt,
    grid-height = 13pt,
}

\punctSetup{punct-style=taiwan}
```

然后在文档中使用：

```
\documentclass[mytemplate]{ltc-cn-vbook}
```

自由列宽模式

`ltc-cn-vbook` / `ltc-tw-vbook` 默认使用 `n-column=0`，即**自由列宽模式**。与 `ltc-guji` 的固定 8 列网格不同，自由模式下每列的宽度由内容决定，布局引擎自动追踪各列宽度并在内容超出页面宽度时分页。

这种模式适合现代书籍中列宽不固定的场景，如混合不同字号、使用 `\行[宽度=3cm]{...}` 指定不等宽列等。

在自定义模板中启用：

```
\contentSetup{
    n-column = 0,           % 启用自由列宽
    grid-width = 13pt,
    grid-height = 13pt,
    border = false,
}
```

下一步：查看 标点系统 | 功能详解 | 模板与自定义

标点系统 (v0.2.7+)

`luatex-cn` 提供了完整的现代标点处理系统，支持标点挤压、禁则处理、竖排引号替换，以及中国大陆/台湾两种标点风格。

标点系统 vs 句读系统

`luatex-cn` 有两套标点处理机制，**互斥**运行：

系统	适用场景	punct-mode	说明
标点系统 (本页)	现代书籍排版	`normal` (默认)	处理挤压、禁则、引号替换
句读系统	古籍排版	`judou` 或 `none`	将标点转为圆圈/点或移除

当 `punct-mode=normal` 时，标点系统启用；当 `punct-mode=judou` 或 `none` 时，句读系统启用、标点系统自动禁用。

标点风格

台湾风格 (默认)

所有标点居中显示在网格单元内，这是默认设置。

```
\punctSetup{punct-style=taiwan}
% 或
\台湾标点风格
```

中国大陆风格

句号（。）、逗号（，、）等偏向网格单元的右上角，更符合中国大陆出版物的排版习惯。

```
\punctSetup{punct-style=mainland}
% 或
\大陆标点风格
```

配置参数

通过 `\punctSetup` 配置标点行为：

```
\punctSetup{
    punct-style = taiwan,    % 标点风格：taiwan 或 mainland
    punct-squeeze = true,    % 标点挤压：将标点压缩为半格
    kinsoku = true,          % 禁则处理：防止标点出现在不当位置
    punct-hanging = false,   % 行尾点号悬挂（默认关）
}
```

参数	说明	默认值	可选值
`punct-style`	标点风格	`taiwan`	`taiwan`，`mainland`
`punct-squeeze`	标点挤压	`true`	`true`，`false`
`kinsoku`	禁则处理	`true`	`true`，`false`
`punct-hanging`	行尾点号悬挂（详见下文）	`false`	`true`，`false`

clreq 预设键（与横排同名同值）

除上表的原有参数外，`\punctSetup`（中文别名 `\标点设置`）还提供一组与横排管道 **键名与取值完全一致**的 clreq 预设，方便同一份文稿在竖排与横排之间保持体例：

参数	中文别名	说明	默认值	可选值
`style`	`标点风格` / `风格预设`	整体风格预设；`none` = clreq「不调整」，既不挤压也不偏靠	随文档类	`mainland`，`taiwan`，`none`
`squeeze-mode`	`宽度调整`	标点宽度调整的判定方式：`legacy` 无条件挤压（`ltc-guji` 默认，版面零变化）；`context` 上下文相关，仅相邻标点／列首列末触发（`ltc-cn-vbook` / `ltc-tw-vbook` 默认）	`legacy`	`legacy`，`context`
`adjacent-punct`	`连续标点`	连续标点缩减量（字幅）	`1.5`	`1.5`，`1`，`natural`
`line-start-bracket`	—	列首的开始夹注符号缩减始侧半字	`trim`	`trim`，`natural`

参数	中文别名	说明	默认值	可选值
`line-end-punct`	—	列末点号缩减末侧半字	`compress`	`compress`, `natural`
`western-space`	`中西间距`	汉字与西文字母/数字间自动加 1/4em 间距（详见下文「中西混排」）	`true`	`true`, `false`

> **上下文相关的宽度调整**（clreq）：夹在汉字之间的单个标点占满一字幅，只有相邻标点连排时才缩减（夹注符号参与时 2 → 1.5 字幅，`adjacent-punct=1` 可收到 1 字幅）。挤压方向按 clreq「空白在哪一侧就往哪边让」。

> **字面分布**按字形墨迹度量锚定，不依赖字体自身的字面设计：中国大陆式直排偏靠右上，台湾式横竖一律居中。因此用台湾字体排中国大陆式、或用中国大陆字体排台湾式，落点都与规范一致。

行尾点号悬挂 (punct-hanging)

clreq 的行尾点号悬挂：落在**列末**的点号（句号、逗号等）整个字幅移出列内，挂在版口之外，列内正文不再因它被压缩。clreq 注明悬挂适用于直排（横排港台式因点号居中不宜悬挂）。

```
\punctSetup{punct-hanging=true}   % 中文别名：标点悬挂
```

默认关闭——悬挂会改变既有文档的版面，是否启用属于本书体例的选择。仅在 `squeeze-mode=context` 挡位（`ltc-cn-vbook` / `ltc-tw-vbook` 默认）生效。

中西混排 (clreq)

中西间距

按 clreq，汉字与西文字母/数字之间自动加 **1/4 汉字宽**的间距，参与行内调整（可挤压至 1/8、拉伸至 1/2）。三类例外不加间距：中文点号前后、开始夹注符号之后、结束夹注符号之前。

- 规则与横排宏包同源（`tex/shared/luatex-cn-cjk-western.lua`），两种模式体例一致。
- 数字串仍是不可拆的整体（符号分离禁则）：串内保持基准字距，两端照常有中西间距。
- 仅在 `squeeze-mode=context` 挡位生效；`western-space=false`（中文别名 `中西间距`）可关闭。

西文横置 `\横置`

clreq 直排中西混排的三种配置中，「顺时针旋转 90°」由 `\横置`（别名 `\横置` / `\Sideways`）实现——适合较长的西文词句；单个字母或缩略语通常直立逐字入格即可（引擎默认行为）：

```
引用\横置{LuaTeX}之名继续排版。
年份\横置{2026}与长词\横置{Typography}混排。
```

横置串的字母连排成词（内部无字距、不被均排撑散），共用一条竖直基线，横向居中于列，与前后汉字照常有 1/4em 中西间距。

短串横排 `\中横排`

三种配置中的「横排入一个字格」（即日文縦中横）由 `\中横排`（别名 `\中横排` / `\CombineUpright`）实现——两三位数字或极短西文横排挤进**一个字位**，适合日期、卷次等：

```
今年\中横排{12}月\中横排{8}日刊行。
第\中横排{365}回。
```

整组共占一个字幅，天然不可拆；组内字形基线同高、横向排开；总宽超过 1em 时只做横向压缩（字高与笔画竖向粗细不变）；与前后汉字照常有 1/4em 中西间距。较长的词句请用 `\横置`。

标点扩类

连接号（～ - –）、间隔号（· • ·）、分隔号（/ ／）已并入分类体系：三类均受行首禁则约束；**中国大陆式间隔号 `•` 固定占半个字宽**（clreq 附录 A GB 式，如人名「爱新觉罗·玄烨」），台湾式占一字宽。

标点挤压 (punct-squeeze)

启用后，句号、逗号等标点会被压缩为半个网格单元的高度，使排版更加紧凑美观。

```
% 启用标点挤压（默认）
\punctSetup{punct-squeeze=true}

% 禁用标点挤压
\punctSetup{punct-squeeze=false}
```

禁则处理 (kinsoku)

遵循中文排版规范 (CLREQ)，防止标点出现在不当位置：

- **行首禁则**：闭括号 `」』）〉》】〕` 、句号 `。．` 、逗号 `，、` 、中间标点 `：；！？` 不允许出现在列首
- **行末禁则**：开括号 `「『（〈《【〔` 不允许出现在列末

```
% 启用禁则处理（默认）
\punctSetup{kinsoku=true}

% 禁用禁则处理
\punctSetup{kinsoku=false}
```

竖排引号替换

标点系统会自动将水平形式的括号和引号替换为 Unicode 竖排形式：

水平形式	竖排形式
「	﹁
」	﹂
『	﹃

水平形式	竖排形式
』	﹄
（	︵
）	︶
《	︽
》	︾
—	︱
…	⋮

在 cn-vbook / tw-vbook 中使用

`ltc-cn-vbook` / `ltc-tw-vbook` 文档类默认启用标点系统，两者的默认风格不同：

```
% 中国大陆风格（默认 punct-style=mainland）
\documentclass{ltc-cn-vbook}

% 台湾风格（默认 punct-style=taiwan）
\documentclass{ltc-tw-vbook}
```

也可以在文档中随时切换：

```
\documentclass{ltc-cn-vbook}
\punctSetup{punct-style=mainland}  % 覆盖默认的台湾风格

\begin{document}
\begin{正文}
天地玄黄，宇宙洪荒。日月盈昃，辰宿列张。
\end{正文}
\end{document}
```

在 guji 中使用

`ltc-guji` 文档类也支持标点系统。当 `punct-mode=normal` 时，标点系统自动生效：

```
\documentclass[四库全书]{ltc-guji}
\正常标点模式  % 使用现代标点（而非句读）
\punctSetup{punct-style=mainland}

\begin{document}
```

```
\begin{正文}
天地玄黄，宇宙洪荒。
\end{正文}
\end{document}
```

与句读模式的切换

可以在文档中动态切换标点系统和句读系统：

```
% 使用现代标点
\正常标点模式
天地玄黄，宇宙洪荒。  % 现代标点显示

% 切换到句读模式
\句读模式
天地玄黄，宇宙洪荒。  % 逗号变为小点，句号变为小圈

% 切换到白文模式（无标点）
\无标点模式
天地玄黄宇宙洪荒      % 所有标点被移除
```

技术细节

标点处理在 Lua 层通过 `luatex-cn-core-punct.lua` 模块实现，分三个阶段：

1. **Flatten 阶段**：分类标点字符（开括号、闭括号、句号、逗号、中间标点、不可断），替换竖排形式
2. **Layout 阶段**：执行标点挤压和禁则处理
3. **Render 阶段**：应用中国大陆/台湾风格的位置偏移

下一步：查看 句读 | 现代竖排 | 功能详解

句读 (Judou)

句读（jù dòu）是传统汉文书籍中对文本进行断句的标记系统。`luatex-cn` 能够自动整理输入的现代标点，并将其转化为符合古籍排版规范的"句读"样式。

标点模式概览

`luatex-cn` 提供三种标点模式（`punct-mode`），通过 `\judouSetup` 或便捷命令切换：

模式	说明	标点系统
`normal` (默认)	显示现代全角标点，支持挤压和禁则	标点系统
`judou`	将标点转为传统句读标记（圆圈/点）	句读系统
`none`	移除所有标点（白文）	无

注意：标点系统和句读系统**互斥**。当 `punct-mode=normal` 时，标点系统处理挤压/禁则；当 `punct-mode=judou` 或 `none` 时，句读系统接管。

什么是句读模式?

在传统的古籍排版中，通常不使用现代的占位全角逗号和句号，而是使用位于正文字符（通常是右下角）的微小圆圈（句）或点（读/逗）。

- **句 (Ju)**：对应现代的句号（。）、问号（？）、叹号（！）。表现为一个小圆圈 ○。
- **读 (Dou)**：对应现代的逗号（，）、分号（；）、顿号（、）。表现为一个小实心点 ● 或 、。

支持的标点符号（v0.2.1+）

v0.2.1 起扩展了标点符号支持范围，包括：

- 基本标点：句号、逗号、顿号、分号、冒号
- 成对标点：引号（「」『』""''）、书名号（《》 〈〉）、括号
- 特殊标点：破折号、省略号、间隔号等

使用方法

1. 切换标点模式

```
% 开启句读模式
\开启句读模式
\句读模式

% 恢复现代标点
\关闭句读模式
\正常标点模式

% 白文模式（无标点）
\无标点模式
\开启无标点模式
```

2. 在不同文档类中使用

guji（古籍排版）

```
\documentclass[四库全书]{ltc-guji}
\开启句读模式  % 使用传统句读

\begin{document}
\begin{正文}
人之初，性本善。性相近，习相远。
% "，"变为右下角小点，"。"变为右下角小圈
\end{正文}
\end{document}
```

cn-vbook / tw-vbook（现代竖排）

```
\documentclass{ltc-cn-vbook}
% 现代竖排文档类默认使用现代标点（punct-mode=normal）

\begin{document}
\begin{正文}
% 现代标点，支持挤压和禁则
天地玄黄，宇宙洪荒。日月盈昃，辰宿列张。

% 也可以切换到句读模式
\句读模式
寒来暑往，秋收冬藏。
\end{正文}
\end{document}
```

3. 文档中动态切换

```
\句读模式
人之初，性本善。  % 句读标记

\正常标点模式
现代文：他说："一二三。"  % 现代标点

\无标点模式
子曰学而时习之不亦说乎  % 白文

\句读模式
子曰：「学而时习之，不亦说乎？」  % 恢复句读
```

配置参数

通过 `\judouSetup` 微调句读外观：

参数	说明	默认值
`punct-mode`	标点模式	`normal`
`judou-color`	句读标记的颜色（别名：`color`、`font-color`、`句读颜色`、`句讀顏色`）	`red` (朱批)
`judou-size`	标记的大小	`1em` (相对字号)
`judou-pos`	标记的位置	`right-bottom`

```
\judouSetup{
    punct-mode = judou,
    judou-color = black,
    judou-size = 0.8em,
}
```

命令参考

命令	中文别名	说明
`\NormalPunctuationMode`	`\正常标点模式`、`\关闭句读模式`、`\正常標點模式`	切换到现代标点
`\JudouPunctuationMode`	`\句读模式`、`\开启句读模式`、`\句讀模式`	切换到句读模式
`\NonePunctuationMode`	`\无标点模式`、`\开启无标点模式`、`\無標點模式`	切换到白文模式

命令	中文别名	说明
`\judouSetup{...}`	`\句读设置`、`\句讀設置`	配置句读参数

技术细节

标点处理是在 Lua 层通过 `luatex-cn-guji-judou.lua` 模块实现的：

1. **识别**：扫描节点流，通过 Unicode 编码识别标点符号。
2. **转换**：在 `judou` 模式下，将标点字符节点移除，并紧随前一个可见汉字节点插入一个装饰标记节点。
3. **渲染**：在渲染阶段，在其前一个汉字的右下定位点绘制对应的圆圈或点。

☞ 下一步：查看 标点系统 | 功能详解

夹注与侧批

`luatex-cn` 提供了完善的注释系统，包括正文内的夹注和位于行间的侧批。

夹注 (Interlinear Notes)

夹注是古籍中常见的双行小字注释，位于正文汉字之间。

使用方法

```
这是正文\\夹注{这是夹注内容，如果内容很长，它会自动分成两列并平衡长度。}继续正文。
```

核心特性

- **自动平衡算法**：系统会自动计算当前列的剩余空间，并将夹注内容平分为左右两列。
- **跨列/跨页流动**：当夹注在一列内排不下时，会自动流向下一列或下一页。
- **对齐控制**：默认外侧对齐，支持 `left`，`right`，`center`，`inward`，`outward` 模式。
- **平衡控制（v0.2.6+）**：通过 `auto-balance=false` 选项可以禁用自动平衡算法。此模式常用于手动分行或特定古籍样式的复现。

夹注配置

```
\jiazhuSetup{
  font-size = 12pt,
  font = {FandolSong-Regular},
  font-color = red,
  auto-balance = true, % 是否自动平衡左右列长度
  align = outward,     % 对齐方式
  n-char-per-col = 20  % 夹注每小列字数（v0.4.4+，见下）
}
```

夹注每小列字数与正文不同 v0.4.4+

古籍常见正文每列 15 字、夹注每小列 20 字。`n-char-per-col`（`每行字数`）按「正文列高 ÷ N」取夹注格高，字号另用 `font-size` 配：

```
\contentSetup{ n-char-per-col = 15 }                 % 正文每列 15 字
\jiazhuSetup{ n-char-per-col = 20, font-size = 22pt } % 全书夹注每小列 20 字
正文……\夹注[n-char-per-col=15]{临时改回 15 字}……      % 单个夹注覆盖
```

- 夹注从列首起时每小列恰好 N 字；夹注前面还有正文时，该列的小列只剩余下的格数，不足 N 字是正常的。
- 同时给 `grid-height` 时以 `grid-height` 为准；夹注内 `\样式[grid-height=…]` 仍可覆盖局部格高。
- 旧写法 `\夹注[grid-height=22.5pt]` 依然可用，但要自己算格高。

侧批 (Side Notes / Sidenote)

侧批是位于两列正文之间或页面边缘的小字批注。

使用方法

支持三种语法：

```
% 基础用法
正文内容\\侧批{这是侧向批注的内容}继续正文。

% 位置参数（自动转换为 em）
正文内容\\侧批[5pt]{调整高度的批注}继续正文。
正文内容\\侧批[0.5]{相当于 0.5em}继续正文。

% 命名参数
正文内容\\侧批[yshift=10pt, color=red]{颜色和位置都指定}继续正文。
```

参数说明

- `yshift`：垂直偏移量，支持位置参数。纯数字自动转换为 em 单位（如 `0.5` → `0.5em`）。
- `xshift` v0.3.5+：水平偏移量（默认 0pt）。
- `color`：批注颜色（默认红色）。
- `font-size`：批注字体大小（默认 10pt）。
- `grid-height`：自定义网格高度（默认等于字号）。
- `border-padding-top`：边框上边距（默认 0pt）。
- `border-padding-bottom`：边框下边距（默认 0pt）。

- `background-color` v0.3.8+: 侧批底色，遮挡其下方的界行。中文别名 `底色` / `背景色`。特殊值 `page` / `inherit` / `继承` / `页面` 会继承当前页面背景色（未配置时退化为白色）。默认空值，行为同旧版本（不绘制底色）。

全局配置

```
\sidenoteSetup{
  color = red,
  font-size = 19pt,
  grid-height = 20pt,
  xshift = 0pt,   % 水平偏移 (v0.3.5+)
  yshift = 0em,
  border-padding-top = 0.5em,
  border-padding-bottom = 0.5em
}
```

侧批开关 v0.3.6+

通过 `sidenote=false` 参数可全局关闭侧批输出（内容仍然存在但不渲染），适用于只想查看正文排版效果的场景：

```
\sidenoteSetup{sidenote=false}  % 全局关闭侧批
```

组件级标点模式 v0.3.5+

侧批可以独立于全局设置使用不同的标点模式：

```
% 侧批使用句读模式（全局可能是 normal 模式）
\侧批[标点模式=judou]{批注内容}
```

全文的标点模式用 `\句读模式` ／ `\正常标点模式` ／ `\无标点模式` 切换，见 句读。

中文别名

- 侧批设置：`\侧批设置` (简体) / `\側批設置` (繁体)
- 纵移参数：`纵移` (简体) / `縱移` (繁体)
- 横移参数：`横移` (简体) / `橫移` (繁体)

颜色跨页保持（v0.2.1+）

v0.2.1 引入了样式注册表机制，侧批的颜色设置现在可以正确跨页保持。即使侧批内容跨越多页，颜色也会始终一致。

夹注捷径 (Jiazhu Shortcut) v0.3.4+

`\夾注捷径` 命令可以注册一对括号符号，使其在源代码中自动替换为 `\夾注{...}`，简化大量夹注的输入。

使用方法

```
% 默认：【...】 自动替换为 \夾注{...}
\夾注捷径

% 自定义括号对
\夾注捷径{（）}

% 带样式参数：所有捷径产生的夹注都使用指定样式
\夾注捷径[font-color=red, font-size=9pt]{（）}
```

效果

注册后，源代码中的：

```
正文内容【这是注释内容】继续正文
```

等效于：

```
正文内容\夾注{这是注释内容}继续正文
```

命令别名

英文	简体	繁体
`\jiazhuShortcut`	`\夹注捷径`	`\夾注捷徑`

注意事项

- 捷径基于 `process_input_buffer` 预处理，在 TEX 解析之前进行文本替换
- 可以同时注册多对不同的括号
- 可选参数支持所有夹注参数（字号、字体、颜色、对齐等）

技术实现 (仅供开发者)

- **夹注**：利用 LuaTEX 属性标记节点。在布局阶段，系统检测到特定属性后挂起正常排版，调用 `core_textflow.lua` 进行分列计算，然后重新映射到网格。v0.2.1 起夹注已迁移至样式栈架构。
- **侧批**：通过 `core_sidenote.lua` 管理。它将内容捕获为独立的盒子，并根据当前基准字符的位置计算偏移地址。v0.2.1 起支持 Phase 2 样式注册表进行颜色保持。
- **样式注册表**：`luatex-cn-style-registry.lua` 提供多属性样式存储，支持 color、font_size、grid_height 等属性的去重和跨页保持。

☞ 下一步：查看 批注 或 句读。

批注 (PiZhu)

`\批注` 命令允许在页面任意位置放置一个浮动的批注框，常用于模拟古籍中的题跋或后人的批改。

使用方法

```
\批注[x=5cm, y=2cm, height=6, color={1 0 0}]{这是批注内容}
```

参数说明

- `x`：批注框左上角的水平位置（相对于版心）。
- `y`：批注框左上角的垂直位置（相对于版心）。
- `height`：批注框占据的网格行数（高度）。
- `color`：颜色。支持 RGB 格式（如 `{1 0 0}` 代表红色）。
- `font-size`：批注内的文字大小。
- `grid-width`：批注专用网格宽度。
- `grid-height`：批注专用网格高度。

开关控制 v0.3.6+

批注和眉批均支持 `pizhu=false` / `meipi=false` 开关，可在 Setup 或单次调用中关闭输出：

```
\pizhuSetup{pizhu=false}  % 全局关闭批注
\meipiSetup{meipi=false}  % 全局关闭眉批
```

特性

- **绝对定位**：不占据正文空间，直接叠加在页面指定坐标。
- **自动分列**：如果内容超过了设定的 `height`（行数），它会自动开启新的一列。
- **防裁剪逻辑**：批注位置计算会自动处理版心（Banxin）位置，防止内容被边框截断。

- **多列支持**：通过设置 `n-cols` 可以精确控制批注框内部的列数。
- **独立缩进**（v0.2.2+）：批注内容不再继承正文的段落缩进设置 (fix #37)。

技术实现

批注底层基于 `core_textbox.lua`。它将内容渲染到一个独立的垂直网格系统中，然后通过 `render_page.lua` 将该系统产生的整个盒子（Box）在页面渲染阶段通过 `shipout` 逻辑叠加输出。

眉批 (MeiPi) v0.2.3+

`\眉批` 命令用于在页面顶部（正文上方）放置批注，是古籍批注中常见的形式。与 `\批注` 不同，眉批会自动计算位置，从右向左排列，避免互相重叠。

使用方法

```
% 自动定位（推荐）
\眉批[height=5]{这是眉批内容}

% 手动指定 Y 坐标
\眉批[height=5, y=50pt]{这是眉批内容}

% 完全手动定位
\眉批[height=5, x=100pt, y=50pt]{这是眉批内容}
```

参数说明

- `height`：眉批占据的网格行数（高度），**必填**。
- `x`：水平位置。不指定时自动计算，从右向左依次排列。
- `y`：垂直位置。不指定时自动计算，底边对齐正文顶部。
- `color`：颜色，默认红色 `{1 0 0}`。
- `font-size`：字体大小，默认 18pt。
- `grid-width`：网格宽度，默认 20pt。
- `grid-height`：网格高度，默认 19pt。

特性

- **自动排列**（v0.2.3+）：多个眉批自动从右向左排列，间距可通过 `\meipiSetup{spacing=...}` 设置。
- **智能定位**：Y 坐标自动计算，使眉批底边紧贴正文顶部。
- **跨页支持**（v0.2.3+）：眉批会正确显示在其锚点所在的页面，而非前一页。
- **独立缩进**：眉批内容不继承正文的段落缩进设置。

配置

```
\meipiSetup{
  spacing = 10pt,  % 眉批之间的间距
  gap = 0pt        % 眉批底边与正文顶部的间距
}
```

技术实现

眉批底层基于 `guji/luatex-cn-guji-meipi.lua`。它通过在节点列表中插入锚点（anchor）来记录眉批的位置，然后在布局阶段根据锚点**后面**第一个有布局信息的节点来确定眉批应该出现在哪一页。

👉 下一步：查看 印章 或 句读。

改字与装饰 (Correction & Decoration)

`luatex-cn` 提供了一套用于古籍校勘和文本装饰的工具体系，包括改字标记、着重号、下划线（专名号/书名号）等功能。

改字 (Correction)

在古籍整理与校勘中，常需对底本文字进行勘误。`\改` 命令提供了一种标准的视觉呈现方式：保留原文并标记删除（、），同时在字符右下方附注改正后的文字。

使用方法

```
\改{替换字}
% 或使用英文别名
\fix{replacement}
```

示例

```
这是一段有错\改{正}误的文本。
```

在此例中，“错”字会被标注删除号（、），而“正”字会以原字号的 60% 显示在“错”字的右下角。

技术细节

- **删除标记**：自动在目标字符中心叠加一个删除点（、）。
- **改正文字**：放置于目标字符右下角，字号为正文的 50% - 60%。
- **无损排版**：该操作不影响原网格布局，保持版面整洁。

文本装饰 (Text Decoration)

针对古籍和现代学术规范，提供了一系列预设的文本标记命令。

1. 划线标记 (Lines)

命令	别名	说明	效果
`\下划线`	`\Underline`	左侧直线（垂直排版）	用于强调或现代习惯
`\专名号`	-	左侧直线	用于人名、地名、朝代等
`\波浪线`	`\WavyUnderline`	左侧波浪线	用于强调
`\书名号`	-	左侧波浪线	用于书籍、篇名等

参数列表 (`[options]`):

- `color` : 颜色，默认为 `red` 。
- `offset` : 划线距离文字的偏移量，默认为 `0.6em` 。

示例:

```
\专名号[color=black]{孔子}
\书名号{论语}
```

技术改进 (v0.2.6+): 下划线功能已从基于字形 (`|`) 的实现迁移到 **PDF 矢量绘图** 实现。这彻底解决了:

- 划线在行末字符不连贯的问题。
- 划线与字体粗细/位置强绑定的问题。
- 支持在侧批、夹注及文本框等复杂环境中使用。

抬头 (Taitou / Elevation) v0.2.6+

`luatex-cn` 提供了完整的抬头（提行/抬格）支持。详见：**抬头**。

---

2. 圈点与着重 (Emphasis)

命令	别名	说明
`\着重号`	`\EmphasisMark` / `\著重號`	逐字在右侧添加着重标记
`\着重号设置`	`\emphasisMarkSetup` / `\圈点设置`	全局设置

参数	中文别名	默认值	说明
`char`	`字符`	`○`	标记字符
`scale`	`缩放`	`0.3`	标记缩放比例
`color`	`颜色`	`black`	标记颜色
`skip-punct`	`跳过标点`	`false`	标点上不加着重号（clreq 5.3.1）

按文档类分档的默认体例（v0.4.0 起）：`ltc-guji` 保持古籍圈点体例——空心圈 `○`、逐字都加（含标点）；`ltc-cn-vbook` / `ltc-tw-vbook` 在配置里预设 `char=●, skip-punct=true`，即 clreq 5.3.1 规定的实心圆点、标点上不加。

> 台湾式说明：台湾教育部《重订标点符号手册》（2008 年修订版）未收录着重号，但实务出版仍在使用；`ltc-tw-vbook` 取 clreq 的通用形态（●），可用 `\着重号设置{char=…}` 改成本书体例。

示例:

```
\着重号{非常重要}
\着重号[skip-punct=false]{标点，也加。}   % 单次覆盖
\着重号设置{char=●, 颜色=red}             % 全局改体例
```

3. 发圈/圈发 — 平上去入声调圈点 v0.3.8+

依宋岳珂《九经三传沿革例》"音有平上去入之分，则随圈发"。在字符四角放置声调圈点。

命令	别名	说明
`\发圈`	`\faSheng` / `\圈发` / `\發圈` / `\圈發`	单字声调圈点
`\发圈批`	`\fashengBatch` / `\發圈批`	平行批量（声调串 + 字符串）
`\发圈设置`	`\fashengSetup` / `\發圈設置`	全局设置
`\平声` / `\上声` / `\去声` / `\入声`	—	声调简写命令

四角约定：平 = 左下，上 = 左上，去 = 右上，入 = 右下。

使用方法:

```
% 简写形式：方括号内直接写声调
\发圈[平]{字}
\发圈[上]{字}
\发圈[去]{字}
\发圈[入]{字}

% 显式 key=value
\发圈[声=去，颜色=red]{字}

% 简写命令
\平声{字}\上声{字}\去声{字}\入声{字}

% 平行批量：声调串 + 主字串（一一对应）
\发圈批{平上去入}{春夏秋冬}
```

参数说明:

参数	中文别名	默认	说明
`tone`	`声` / `聲`	`ping`	声调，可选 `ping` / `shang` / `qu` / `ru`，或简写 `平` / `上` / `去` / `入`
`color`	`颜色` / `顏色`	`black`	圈点颜色
`char`	`字符`	`○`	圈点字符
`scale`	`缩放` / `縮放`	`0.3`	圈点缩放
`offset`	`偏移`	`0.45em`	距字符中心的偏移量

通用装饰命令 (Generic Decoration)

`\装饰`（`\decorate`）是一个底层命令，允许用户在**每个字符**上叠加任意内容的“装饰层”。

使用方法

```
\装饰[参数]{内容}
% 或
\decorate[options]{content}
```

参数说明

参数	说明	默认值
`char`	装饰字符	`。`
`color`	颜色	`red`
`xoffset`	水平偏移量 (xshift)	`0pt`
`yoffset`	垂直偏移量 (yshift)	`0pt`
`scale`	缩放比例	`1.0`
`font-size`	强制指定字号	自动

示例:

```
% 在每个字上叠加红色的圈
文字\装饰[scale=0.8, color=red, char=○]{文字}
```

👉 下一步：查看 填充文本框 或 功能详解。

抬头 (Taitou / Elevation) v0.2.6+

`luatex-cn` 提供了完整的抬头（提行/抬格）支持，用于复现古籍中对特定称谓的尊敬性格式。

使用方法

命令	效果	说明
`\抬头[N]`	提行并上移 N 格	核心命令，N 可为正/负
`\平抬`	提行并顶格	相当于 `\抬头[0]`
`\单抬`	提行并高出 1 格	相当于 `\抬头[1]`
`\双抬`	提行并高出 2 格	相当于 `\抬头[2]`
`\三抬`	提行并高出 3 格	相当于 `\抬头[3]`
`\相对抬头[N]`	相对当前缩进上移 N 格	用于夹注等已有缩进的环境

自动边框包裹

抬头功能与版心系统深度集成。当文字“冲出”正常版面进入天头时，`luatex-cn` 会自动识别并绘制阶梯状的边界线，确保护栏（乌丝栏/朱丝栏）能正确包裹抬高的文字。

示例

```
至圣\单抬 先师孔子
```

技术实现

抬头功能在 `luatex-cn-core-paragraph.sty` 中定义，其逻辑在 `luatex-cn-core-textflow.lua` 和 `luatex-cn-layout-grid.lua` 中实现。当调用抬头命令时，布局引擎会调整当前列的起始

行号（`cur_row`），并记录抬头高度。渲染层 `luatex-cn-core-render-border.lua` 则根据这些信息绘制阶梯状的边框。

☞ 下一步：查看 版心与鱼尾 或 功能详解。

文本框 (TextBox)

`luatex-cn` 提供了强大的竖排文本框功能，支持网格对齐、自动分栏以及特定样式的填充。

基础文本框

`\文本框`（`\TextBox`）用于创建一个独立的竖排文本区域。

使用方法

```
\文本框[参数列表]{文本内容}
```

常用参数

参数	说明	默认值
`height` / `高度`	文本框的高度（以网格数为单位）。设为 0 则自动延长。	`0`
`n-cols` / `列数`	强制分栏的列数。	`0` (不分栏)
`border` / `边框`	是否显示边框框线。	`false`
`background-color` / `底色`	背景颜色。	无
`fill` / `填充`	均匀分布内容（等价于 `box-align=fill`）。	—
`floating`	是否为浮动文本框（需配合 x, y 坐标）。	`false`
`transparent` / `透明`	背景是否透明（v0.3.6+）。false 时使用页面背景色填充。	`true`

自动延长与压缩

- **自然延长**：如果不指定 `height`（或设为 0），文本框会根据内容长度自然向下延伸。
- **字体压缩（扁字）**：如果指定了固定 `height` 但内容过多，系统会自动垂直压缩字体（扁字）以确保内容完全放入框内（v0.1.5 新增）。

- **独立缩进**（v0.2.2+）：文本框内容不再继承正文的段落缩进设置，确保布局独立性(fix #37)。

填充文本框

`\填充文本框`（`\FillTextBox`）是 `\TextBox` 的一个变体，默认启用填充对齐模式，常用于需要充满特定区域的场景。

```
\填充文本框[17]{端明殿学士兼翰林侍读学士……赐紫金鱼袋}   % 只写整数 = 格数
\填充文本框[height=17]{……}                              % 等价的键值写法
\填充文本框[height=120pt]{……}                             % 绝对高度
\填充文本框[height=4]{天地玄黄宇宙}洪荒                  % 6 字压进 4 格
```

等同于 `\TextBox[height=…, box-align=fill]{内容}`。

- **扁字**：内容多于格数时整体纵向压缩，字宽不变；压缩比 = 格数 ÷ 字数，字数越多越扁（83 字压进 17 格约 1/5，接近横线，此时应减小 `font-size` 或增加格数）。内容不足时均匀拉开。
- 方括号里只写整数时视为 `height=N`；要同时给别的键（`font-size`、`border` 等）请用键值写法。
- 与 `\文本框` 的区别：`\文本框[height=N]` 不压缩，只占 N 格并顶端对齐。
- 在 `正文`、`列表`、`段落` 内都可直接使用，无需额外宏包。

边框便捷命令 (v0.3.2+, 扩展于 v0.3.4)

一系列快捷命令，用于快速对正文中的标注文字添加边框和/或反白效果。命令会根据内容字数自动计算尺寸。

命令一览

命令	英文	繁体	效果
`\墨围{文字}`	`\bordered`	`\墨圍`	矩形黑色边框
`\反白{文字}`	`\inverted`	—	黑底白字（无边框）
`\墨围反白{文字}`	`\borderedInverted`	`\墨圍反白`	矩形边框 + 黑底白字
`\八角墨围{文字}`	`\octagonBordered`	`\八角墨圍`	八角形边框
`\八角墨围反白{文字}`	`\octagonBorderedInverted`	`\八角墨圍反白`	八角形边框 + 黑底白字

命令	英文	繁体	效果
`\带圈{文字}`	`\circled`	`\帶圈`	圆形边框

使用示例

```
\墨围{集解}裴駰曰……
\墨围{索隱}司馬貞曰……
\反白{正義}張守節曰……
\八角墨围{校勘}此字疑誤……
\带圈{一}
```

反白跨列换行 (v0.4.4+)

`\反白` ／ `\墨围反白` ／ `\八角墨围反白` 是行内排版：剩余格数不够时逐字换到下一列（夹注里换到下一小列），不再整块推到下一列、空出前面的格子。底色与边框按每列连续的一段分别绘制。

参数	中文键	说明
`break`	`断口` ／ `斷口`	被换列截断处的角型：`square` ／ `直角`（默认，只有整段首尾画八角）、`octagon` ／ `八角`（每段都画完整八角）
`inline`	`可断行` ／ `可斷行`	设为 `false` 时回退为旧的整块文本框实现
`border-shape`	`边框形状` ／ `邊框形狀`	`none` ／ `rect` ／ `octagon` 仍走行内实现，可跨列断开（v0.4.5+, #210）；`circle` 回退整块

行内实现还认 `font-color`（`color` ／ `字体颜色` ／ `颜色`）、`background-color`（`底色`）、`border-color`（`边框色`）、`border-width`（`边框宽`）。

传入文本框专有的 key（`高度`、`填充`、`padding-top` 等）时会回退为整块文本框（不能跨列断开），如 `\八角墨围反白[高度=3, 填充]{...}`。从 v0.4.5 起回退时会在日志里给出警告，说明是哪个 key 触发的；确实想要整块时写 `inline=false` 即可消除警告。

随字号缩放 (v0.4.4+)

在夹注里或临时改小字号时，`\带圈` 的圈、`\墨围` ／ `\八角墨围` ／ `\反白` 的框随所在字号等比缩放，并在所占行内垂直居中，不再按正文行高画而压到邻字。

边框间距控制

通过 TextBox 的 `border-margin-x` / `border-margin-y` 参数可分别控制横向和纵向的边框与内容间距：

```
\TextBox[border, border-margin-x=2pt, border-margin-y=1pt]{内容}
```

参数	说明	默认值
`border-margin`	统一边框间距	`0pt`
`border-margin-x`	横向边框间距	(跟随 border-margin)
`border-margin-y`	纵向边框间距	(跟随 border-margin)
`outer-margin`	文本框外部间距	`0pt`

浮动批注 (Floating Box)

通过设置 `floating=true`，可以将文本框放置在页面的绝对坐标位置，常用于制作**批注**。`\悬浮文本框[…]{…}`（`\FloatingTextBox`）是 `\文本框[floating=true, …]` 的简写。

```
\悬浮文本框[
  x=2cm, y=3cm,
  height=5,
  border=true
]{这是浮动批注}
```

也可以直接使用简化的 `\批注` 命令（`\PiZhu`）。全部参数见 命令索引，`\文本框设置`（`\textboxSetup`）可设默认值。

☞ 下一步：查看 功能详解。

印章 (YinZhang)

`\印章` 命令用于在页面上添加馆藏印、私印等图像，支持绝对定位和透明度。

使用方法

```
\印章[page=1, xshift=2cm, yshift=5cm, opacity=0.8, width=3cm]{seal.png}
```

参数说明

- `page`：印章出现的页码（默认为 1）。
- `xshift` / `yshift`：距离版心（或纸张边缘，取决于模式）的偏移。
- `opacity`：不透明度（0.0 - 1.0）。古籍印章通常建议设为 0.7-0.9 左右以模拟纸张渗透感。
- `width`：印章图像展示的宽度。
- `color`：（可选）如果使用矢量印章，可指定颜色。

特性

- **层级管理**：印章默认在 `shipout/background` 阶段渲染，位于文字层下方；也可根据需要调整为前景。
- **多点布局**：可以在同一页插入多个印章命令，系统会自动根据坐标分别放置。
- **绝对定位**：基于 TikZ 的坐标系统，能够精确锁定页面位置。

技术实现

印章功能利用了 LaTeX 的 `shipout` 钩子（Hooks）。每调用一次 `\印章`，都会在相应的钩子中注册一个带有 `\ifnum\value{page}=N` 判断的 `tikzpicture` 块，确保图像在正确的渲染层和正确的页码输出。

☞ 下一步：查看 句读 或 功能详解。

调试模式

`luatex-cn` 提供了完整的调试功能，帮助排查排版问题和定位元素位置。

---

1. 基础调试开关

1.1 全局开关

```
\LtcDebugOn    % 开启调试（或 \开启调试）
\LtcDebugOff   % 关闭调试（或 \关闭调试）
```

也可以在文档类选项中开启：

```
\documentclass[四库全书彩色, debug=true]{ltc-guji}
```

1.2 模块级调试

针对特定模块开启调试，减少日志量：

```
\LtcDebugModuleOn{vertical}   % 或 \开启调试模块{vertical}
\LtcDebugModuleOff{vertical}  % 或 \关闭调试模块{vertical}
```

可用模块：

- `vertical` - 竖排引擎
- `banxin` - 版心系统
- `sidenote` - 侧批
- `textflow` - 夹注分列
- `page` - 页面设置
- `page-split` - 筒子页分割
- `render` - 渲染阶段

---

2. 显示坐标网格

2.1 基本用法

在页面上叠加坐标网格，用于精确定位元素位置。

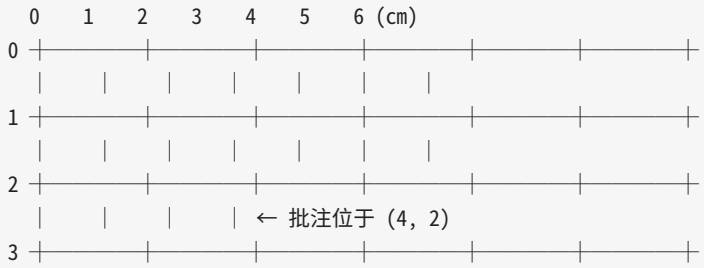
```
% 显示网格（默认单位 cm）
\LtcShowGrid              % 或 \显示网格

% 隐藏网格
\LtcHideGrid              % 或 \隐藏网格
```

2.2 指定单位

```
\LtcShowGrid[measure=cm]  % 厘米（默认）
\LtcShowGrid[measure=mm]  % 毫米
\LtcShowGrid[measure=pt]  % 点（1/72 英寸）
```

2.3 效果示例



2.4 别名

```
\显示坐标[measure=pt]  % 等同于 \LtcShowGrid[measure=pt]
\隐藏坐标              % 等同于 \LtcHideGrid
```

---

3. 显示页面边框

显示页面边界和内容区域：

```
\LtcShowFrame   % 或 \显示边框
```

效果：

- 红色线框：页面物理边界
- 蓝色线框：内容区域（margin 内）

---

4. 调试颜色设置

为不同模块的调试输出设置不同颜色（便于区分）：

```
\LtcDebugColor{vertical}{blue}
\LtcDebugColor{banxin}{red}
\LtcDebugColor{sidenote}{green}
```

---

5. 完整使用示例

5.1 定位批注位置

```
\documentclass[四库全书彩色]{ltc-guji}

% 开启调试和坐标网格
\LtcDebugOn
\显示坐标[measure=cm]

\begin{document}
\begin{正文}
测试文本\批注[x=4cm, y=2cm]{批注内容}
\end{正文}
\end{document}
```

5.2 调试侧批跨页

```
\documentclass[四库全书彩色, debug=true]{ltc-guji}

\开启调试模块{sidenote}
\显示网格

\begin{document}
\begin{正文}
正文内容\侧批{这是一条很长的侧批，可能会跨页显示，用于测试跨页颜色保持}
% ... 更多内容 ...
\end{正文}
\end{document}
```

5.3 检查版心渲染

```
\documentclass[四库全书彩色]{ltc-guji}

\开启调试模块{banxin}
\显示边框

\begin{document}
\begin{正文}
内容
\end{正文}
\end{document}
```

---

6. 调试输出说明

开启调试后，编译日志中会输出详细信息。

6.1 日志格式

```
[模块名] 消息内容
```

6.2 常见日志示例

页面设置:

```
[page] Configured: paper=400mm x 280mm, margins=133pt/84pt/133pt/84pt
[page-split] Enabled - target=200mm x 280mm
```

侧批处理:

```
[sidenote] Processing registry entry 1
[sidenote] Column wrap: base_indent=2, next_r=2
```

渲染阶段:

```
[render] Processing page 1, nodes=256
[render] Glyph: char=测, pos=(5, 3)
```

6.3 过滤日志

```
# 只看调试输出
lualatex file.tex 2>&1 | grep "\["

# 只看特定模块
lualatex file.tex 2>&1 | grep "\[sidenote\]"

# 只看错误
lualatex file.tex 2>&1 | grep -E "\[ERROR\]|\[WARNING\]"
```

---

7. Lua 端调试

7.1 输出调试信息

```
-- 使用项目的调试模块
local debug = require('debug.luatex-cn-debug')
local dbg = debug.get_debugger('my-module')

dbg.log("Processing node: " .. tostring(node))
dbg.log(string.format("Position: page=%d, col=%d, row=%d", p, c, r))
```

7.2 直接输出

```
-- 始终输出（不受调试开关影响）
texio.write_nl("term and log", "[DEBUG] message")

-- 格式化输出
texio.write_nl("term and log", string.format(
    "[DEBUG] x=%.1fpt, y=%.1fpt",
    x / 65536, y / 65536
))
```

7.3 打印节点信息

```
-- 打印节点类型和属性
local function debug_node(n)
    local D = node.direct
    local id = D.getid(n)
    local attr = D.get_attribute(n, ATTR_JIAZHU)
    texio.write_nl("term and log", string.format(
        "[DEBUG] Node: id=%d, attr=%s",
        id, tostring(attr)
```

```
    ))
end
```

---

8. TEX 端调试

8.1 输出变量值

```
% 显示 token list 内容
\tl_show:N \l_my_tl

% 输出到终端
\iow_term:x { value:~|\l_my_tl| }

% 输出到日志
\iow_log:x { Debug:~height=\l_my_height_tl }
```

8.2 条件调试

```
\bool_if:NT \g__luatexcn_debug_bool
  {
    \iow_term:n { [DEBUG]~Entering~environment }
  }
```

---

9. 坐标系统说明

9.1 页面坐标

```
(0, 0) ──────────────────────────────────────────→ x (paperwidth)
  │
  │    ┌──────────────────────────────┐
  │    │  内容区域                    │
  │    │  (margin 内)                 │
  │    └──────────────────────────────┘
  │
  ↓
  y (paperheight)
```

9.2 竖排网格坐标

```
        列 1   列 2   列 3   (从右向左)
       ┌──────────┬──────────┬──────────┐
行 1   │  字 │  字 │  字 │  ↑
       ├──────────┼──────────┼──────────┤   │
行 2   │  字 │  字 │  字 │   │ 从上向下
       ├──────────┼──────────┼──────────┤   │
行 3   │  字 │  字 │  字 │  ↓
       └──────────┴──────────┴──────────┘
```

9.3 筒子页坐标

筒子页模式下，物理页面分为左右两半：

```
物理页面（40cm 宽）：
┌──────────────────────────────────────┬──────────────────────────────────────┐
│    左半（20cm）      │     右半（20cm）     │
│   → 输出为 Page 2   │   → 输出为 Page 1   │
│                     │                     │
│  x: 0 ──────→ 20cm   │  x: 0 ──────→ 20cm   │
└──────────────────────────────────────┴──────────────────────────────────────┘
```

注意：批注、印章等的 x 坐标是相对于**逻辑页面**（半页），不是物理页面。

---

10. 常见调试场景

10.1 元素位置不对

1. 开启坐标网格：`\显示坐标[measure=cm]`
2. 检查元素的 x, y 参数
3. 注意筒子页模式下坐标是相对于半页

10.2 颜色丢失

1. 开启模块调试：`\开启调试模块{render}`
2. 检查日志中是否有颜色信息
3. 确认是否跨页导致颜色堆栈断开

10.3 内容消失

1. 开启全局调试：`\LtcDebugOn`
2. 检查是否有节点被错误移除
3. 检查属性是否被 `\selectfont` 清除

10.4 版心位置错误

1. 显示边框：`\显示边框`
2. 开启版心调试：`\开启调试模块{banxin}`
3. 检查 margin 和 paper 尺寸设置

👉 返回：开发文档 | 功能详解 | 首页

命令索引 (Command Reference)

> 本页列出 luatex-cn 全部用户命令、环境及其参数。每个命令/环境均提供英文、简体中文、繁体中文三套名称（同形则省略）。

通用类型说明

参数表中的"类型"列使用以下名称：

类型名	说明	示例
度量	带单位的长度值。支持的单位：`pt`（磅）、`mm`（毫米）、`cm`（厘米）、`em`（当前字号的宽度）、`ex`（当前字号的 x 高度）、`sp`（TEX 最小单位，65536sp = 1pt）。	`12pt` `1.5em` `210mm` `0.4cm`
字号	字体大小，本质也是度量值。常用 `pt` 或 `em`（相对于基准字号）。	`12pt` `0.8em` `16pt`
颜色	颜色值。支持三种格式：颜色名（如 `red` `black` `blue`）；0-1 小数 RGB（如 `1 0 0` 表示红色）；0-255 整数 RGB（如 `255,128,0`）。	`red` `1 0 0` `255,128,0`
字符串	一般文本值，含义由具体参数决定。	`grid` `taiwan` `right-bottom`
整数	整数值，通常表示格数或列数。	`0` `8` `2`
是/否	布尔值，`true` 或 `false`。也可省略值，直接写参数名等价于 `true`。	`true` `false`（或直接 `border`）
选项	只能从给定选项中选一个。可选值会在说明中列出。	`top` `center` `bottom`
	组合参数，自动设置多个关联参数。	

类型名	说明	示例
快捷		`spacing=5pt` → 同时设置 `spacing-top` 和 `spacing-bottom`

中文取值

v0.4.5+ 类型为**选项**或**是/否**的参数都可以写中文值，与英文值等价，例如 `对齐=顶`、`单列=左`、`页码位置=内侧`、`显示=是`。常用对照：

中文值	英文值
是、真、开、开启、启用	`true`
否、假、关、关闭、禁用	`false`
无	`none`（该参数无此值时为 `false`）
上、顶 ／ 下、底	`top` ／ `bottom`
左、居左 ／ 右、居右 ／ 中、居中	`left` ／ `right` ／ `center`
两侧	`both`
内侧、书脊侧 ／ 外侧、切口侧	`inner` ／ `outer`
直角 ／ 八角	`square` ／ `octagon`
大陆 ／ 台湾	`mainland` ／ `taiwan`
自动	`auto`

繁体写法（如 `開`、`頂部`、`內側`）同样可用。写了参数不接受的值会报错「选项 'x' 不接受值 'y'」。

Core 核心宏包

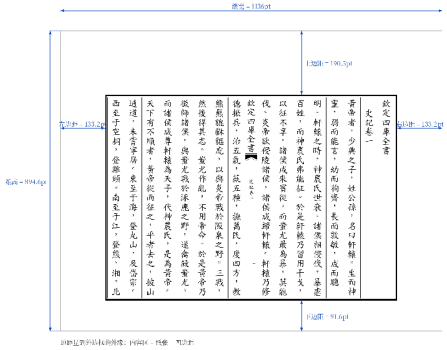
核心排版引擎，提供页面、内容区域、网格布局、列、段落、文本框、样式等基础功能。

页面 (Page)

文件: `core/luatex-cn-core-page.sty`

`\pageSetup` `\页面设置` `\頁面設置`

配置页面几何尺寸、边距、筒子页等参数。调用后同步到 Lua 层。



> 边距量到外边框的外缘。图中数值由脚本从实际排版结果里量出，并校验与参数设定值一致（见仓库 `scripts/build/build_figures.py`）。

参数	简体	繁体	类型	默认值	说明
paper-width	纸宽	紙寬	度量	210mm	纸张宽度
paper-height	纸高	紙高	度量	297mm	纸张高度
margin-top	上边距	上邊距	度量	0pt	上边距

参数	简体	繁体	类型	默认值	说明
margin-bottom	下边距	下邊距	度量	0pt	下边距
margin-left	左边距	左邊距	度量	0pt	左边距（仅 `twoside=false` 时生效，v0.4.3+）
margin-right	右边距	右邊距	度量	0pt	右边距（仅 `twoside=false` 时生效，v0.4.3+）
background-color	底色	底色	颜色	(空)	页面背景色
footskip	页脚距	頁腳距	度量	0pt	页脚偏移
topskip	页头距	頁頭距	度量	0pt	页头偏移
page-split-enabled	开启分页	開啟分頁	是/否	true	是否启用筒子页裁剪
page-split-right-first	右页优先	右頁優先	是/否	true	筒子页先印右页
twoside	—	—	是/否	false	双面模式：为 true 时只读 `margin-inner` ／ `margin-outer`（奇偶页镜像），为 false 时只读 `margin-left` ／ `margin-right`；被忽略的一组若被设置会发出警告（v0.4.3+）
margin-inner	—	—	度量	(空)	书脊侧边距（仅 `twoside=true` 且 `mirror-margins=true` 时生效）。竖排书奇数页是左页，书脊在右（v0.4.4 起按书脊算）

参数	简体	繁体	类型	默认值	说明
margin-outer	—	—	度量	(空)	翻口侧边距（仅 `twoside=true` 且 `mirror-margins=true` 时生效）
mirror-margins	镜像边距	鏡像邊距	是/否	true	为 false 时 `twoside=true` 也读 `margin-left` ／ `margin-right`，内容区在奇偶页同一位置，书眉页码仍奇偶交替（v0.4.4+）
binding-offset	装订边距	裝訂邊距	度量	0pt	加在书脊一侧的装订边距：奇数页右、偶数页左；`twoside=false` 时加在右侧（v0.4.4+）
facing-pages	对页补页	對頁補頁	是/否	false	章（`\chapter`）落在偶数页顶端时，在它前面自动补一页，使每章从奇数页起；只认 `正文` 块内的 `\chapter`，筒子页模式下不自动补（v0.4.4+）
blank-page-style	补页样式	補頁樣式	字符串	header	补出来的页的样式：`header`（默认）带版框、版心／页眉页码；`blank` ／ `空白` 整页完全空白（v0.4.4+）
header-enabled	启用页眉	啟用頁眉	是/否	false	启用运行页眉
header-font-size	页眉字号	頁眉字號	字号	9pt	页眉字号
page-number-style	页码样式	頁碼樣式	字符串	空（未设置）	页码显示风格：`digits`（逐位中文）／ `chinese`（位值中文）／ `arabic` ／ `none`。未设置时竖排页码不输出，版心页码沿用位值中文数字
page-number-font-size	页码字号	頁碼字號	字号	9pt	页码字号

`\enableSplitPage` `\开启分页` `\開啟分頁`

启用筒子页（电子简策）裁剪模式，可接受 key-value 参数配置。

`\disableSplitPage` `\关闭分页` `\關閉分頁`

关闭筒子页裁剪。

`\SetPageNumber` `\设置页码` `\設置頁碼`

将页码计数器设为指定值。语法：`\SetPageNumber{5}`

`\ResetPageNumber` `\重置页码` `\重置頁碼`

将页码重置为 1。

`\换页` `\換頁`

强制换页，等价于 `\newpage`。

`\InsertBlankPage` `\补空白页` `\補空白頁`

在此处强制补一页（v0.4.4+）。`正文` 块内、块间都可以写，连写两次补两页，紧跟在 `\换页` 之后不会多补。可选参数 `补页样式`（`blank-page-style`）只对这一页生效：`\补空白页[补页样式=空白]`。

环境 `Page` `页`

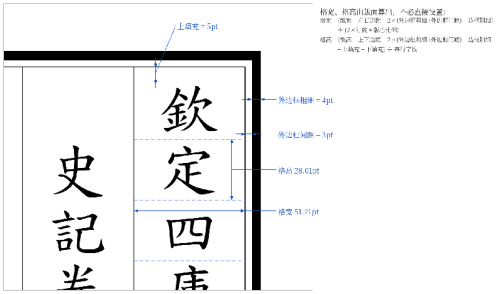
在环境内临时修改页面参数（纸张尺寸、边距等），环境结束后自动恢复。支持全部页面参数。

内容区域 (Content)

文件：`core/luatex-cn-core-content.sty`

`\contentSetup` `\内容设置` `\內容設置`

配置正文区域的所有排版参数：网格尺寸、字体、行列数、边框、布局模式等。



> 设了 `每行字数`（n-char-per-col）后，格高由版面高度算出；格宽由版面宽度、`行数` 与 `版心比例` 算出。公式见图右上。这时再写 `格高` ／ `格宽` 不起作用，见下文 参数之间的覆盖关系。

参数	简体	繁体	类型	默认值	说明
n-column	行数	行數	整数	8	每半页行数（竖排时"行"即列）
n-char-per-col	每行字数	每行字數	整数	0	每行字数（0=自动）
grid-width	格宽	格寬	度量	1.5em	字格宽度。`行数` > 0 时（古籍类）由版面算出，此值不生效
grid-height	格高	格高	度量	1.2em	字格高度。`每行字数` > 0 时由版面算出，此值不生效
font-size	字号	字號	字号	12pt	正文字号
font	字体	字體	字符串	(空)	正文字体名
font-features	字体特性	字體特性	字符串	(空)	字体 OpenType 特性
font-color	字体	字體	颜色	(空)	正文字体颜色

参数	简体	繁体	类型	默认值	说明
	颜色	顏色			
vertical-align	纵对齐	縱對齊	字符串	center	字符在格中的纵向对齐
horizontal-align	横对齐	橫對齊	选项	right	整块行在内容区内的水平位置：`right`（居右，默认，从右边起排）`center`（居中）`left`（居左）；也可写中文值 `居右` ／ `居中` ／ `居左`。别名 `h-align`（v0.4.5+，#175）
height	高度	高度	度量	(自动)	内容区高度
layout-mode	布局模式	佈局模式	选项	grid	布局引擎：`grid` 或 `natural`
spacing-col	—	—	度量	(空)	列间距（中文 `行间距` 现指 `column-spacing`，见下）
line-spacing	行距	行距	度量	\baselineskip	行距（natural 模式）
page-columns	版面行数	版面行數	字符串	(空)	直接指定每页列数，格宽按内容区等分；覆盖 `行数`，且不给版心留位，有版心时请用 `行数`
banxin-ratio	版心比例	版心比例	字符串	0.7	版心在行中的比例（0-1）
border	边框	邊框	是/否	false	是否显示栏线
border-thickness	边框粗细	邊框粗細	度量	0.4pt	栏线粗细

参数	简体	繁体	类型	默认值	说明
border-color	边框色	邊框色	颜色	black	栏线颜色
padding-top	上填充	—	度量	0pt	界栏框顶边与第一个字格之间的留白，算在界栏框之内（别名 `column-padding-top` ／ `栏上填充` ）
padding-bottom	下填充	—	度量	0pt	最后一个字格与界栏框底边之间的留白（别名 `column-padding-bottom` ／ `栏下填充` ）
outer-border	外边框	外邊框	是/否	false	是否显示外边框
outer-border-thickness	外边框粗细	外邊框粗細	度量	1pt	外边框线宽
outer-border-sep	外边框间距	外邊框間距	度量	2pt	外边框与栏线间距
inter-cell-gap	字间隙	字間隙	度量	0pt	字格间附加间距
cell-height	字格高	字格高	度量	(空)	单字格高度覆盖
cell-width	字格宽	字格寬	度量	(空)	单字格宽度覆盖
cell-gap	字格	字格	度量	(空)	字格间距覆盖

参数	简体	繁体	类型	默认值	说明
	间距	間距			
auto-column-width	自动行宽	自動行寬	是/否	false	自动计算列宽
column-spacing-top	行上间距	行上間距	度量	0pt	列顶部留白
column-spacing-bottom	行下间距	行下間距	度量	0pt	列底部留白
column-spacing	行间距	行間距	快捷	—	同时设置列上/下间距
paragraph-spacing	段间距	段間距	度量	0pt	段落间距
n-bands	分栏	分欄	整数	1	分栏数（水平分区）
band-gap	栏间距	欄間距	度量	0pt	栏间距
band-mode	栏模式	欄模式	选项	auto	栏流动模式：`auto`（跨页）`per-page`（逐页）
band-heights	栏高度	欄高度	列表	(空)	各栏高度（逗号分隔）
band-columns	栏行数	欄行數	字符串	(空)	每栏列数

参数	简体	繁体	类型	默认值	说明
column-border	行边框	行邊框	字符串	(空)	列（丝栏）边框
band-border	栏边框	欄邊框	字符串	(空)	栏间分隔线
digital-mode	数字模式	數字模式	是/否	false	启用数字化模式（obeylines，每行=一列）
ignore-spaces	忽略空格	—	是/否	true	忽略源码中的半角空格
indent	缩进	縮進	整数	(空)	非首行缩进（格数）
first-indent	首行缩进	首行縮進	整数	(空)	首行缩进（格数）

> **注意**: 未识别的参数会自动转发到页面（Page）参数。

`\Content` `\内容` `\內容`

将文本交给网格排版引擎处理。语法：`\Content[keys]{文本内容}`。支持全部 content 参数。

`\Space` `\空格`

插入 N 个全角空格（U+3000），用于网格中的手动间距。语法：`\Space[3]`

`\MissingChar` `\缺字`

插入白方框「□」（U+25A1）作缺字／漫漶字占位。语法：`\缺字` 或 `\缺字[3]`（连缺 3 字）。

`\Sideways` `\横置` `\横置`

西文串顺时针旋转 90° 排入竖排列（clreq 直排中西混排配置）。字母连排成词、共用竖直基线、横向居中于列。语法：`\横置{LuaTeX}`。适合较长西文词句；短串直立逐字入格即可（默认行为）。

`\CombineUpright` `\中横排` `\中橫排`

两三位数字/极短西文横排挤进一个字位（clreq 直排中西混排配置之「横排入一个字格」，即縱中横）。整组共占一个字幅、不可拆列；超过 1em 只做横向压缩（字高不变）。语法：`\中横排{12}`。适合日期「`\中横排{12}`月」、卷次等；较长词句用 `\横置`。

环境 `BodyText` `正文`

`\Content` 的环境形式，支持全部 content 参数。

```
\begin{正文}[行数=10, 格高=1.5em]
  此处为正文内容……
\end{正文}
```

环境 `CenteredPage` `居中页` `居中頁` v0.4.5+

扉页、卷标题页等：前后各换一页，在新页上排一段正文，整块行在内容区内水平居中（即 `正文[horizontal-align=center]`）。各行的上下位置用 `\行[align=center]` 等控制；可选参数传给正文，如 `[horizontal-align=left]`。不要嵌套在另一个 `正文` 里。(#175)

```
\begin{居中页}
  \行[align=center]{王又朴集}
  \行[align=center]{上}
\end{居中页}
```

`\NewHalfPage` `\换半页` `\換半頁` v0.3.6+

强制换半页。在筒子页模式下，从当前半页跳到下一半页。

`\IllustrationPage` `\插图页` `\插圖頁` v0.3.6+

将当前页面标记为插图页，自动隐藏丝栏（保留内外边框），默认先换半页。适用于需要在古籍中插入全页插图的场景。

`\EndIllustrationPage` `\恢复丝栏` `\恢復絲欄` v0.3.6+

结束 `\插图页` 的效果，恢复丝栏。通常在 `\clearpage` 之后调用。

参数之间的覆盖关系 v0.4.5+

有几组参数会互相覆盖：同时设了，其中一个就静默不起作用，页面看起来就「变形」了。下表列出这些组合、实际生效的是哪一个，以及怎样才能用另一个。

同时设置	实际效果	想用另一个时
`每行字数`（n-char-per-col）> 0 与 `格高`（grid-height）	格高 =（版框内高 − 上下填充）÷ 每行字数，`格高` 不生效	设 `每行字数 = 0`，`格高` 才起作用
`行数`（n-column）> 0 与 `格宽`（grid-width），`ltc-guji` ／ `ltc-guji-digital`	格宽 = 内容区宽 ÷（2 × 行数 + 版心比例），`格宽` 不生效	改 `行数`、左右边距或 `版心比例`
`版面行数`（page-columns）与版心	每页列数 = 版面行数，格宽按内容区等分，**不给版心留位**，正文压在版心上	有版心时用 `行数`；`版面行数` 只用于没有版心的版面
`版面行数` 与 `行数`（无版心）	以 `版面行数` 为准，`行数` 不生效	去掉 `版面行数`
`字号`（font-size）大于格高	字比字格高，上下相邻的字互相压叠	减小 `字号`，或减小 `每行字数` 让字格变高
`版心上比` + `版心中比` > 1	版心下段没有空间，下分隔线与版心页码被挤出版心	让两者之和 ⩽ 1（下段比例 = 1 − 上比 − 中比）
`twoside = true` 与 `左边距` ／ `右边距`	只读 `margin-inner` ／ `margin-outer`，左右边距不生效	设 `镜像边距 = false`，或改写 `margin-inner` ／ `margin-outer`（见 `\pageSetup`）

这些组合都会在编译时提醒，例如：

```
luatex-cn Warning: 格高（grid-height）不生效：每行字数（n-char-per-col）= 21 时格高由版面高度算出（现为 28.00pt）；要用格高请
```

提醒只针对**你自己设的值**：文档类和模板（`.cfg`）设的值不算。比如四库全书模板设了 `每行字数 = 21`，在它上面写 `\contentSetup{格高 = 30pt}` 就会得到上面这条提醒；只用模板、不改这几个参数的文档不会有提醒。版心比例之和超过 1 不论谁设的都会提醒；字号大于格高只在你自己设了 `字号` 或 `每行字数` 时提醒。每条提醒每份文档只出一次。

样式 (Style)

文件：`core/luatex-cn-core-style.sty`

`\Style` `\样式` `\樣式`

行内样式覆盖，可改变字号、字体、字体颜色、格高、间距。不影响布局模式。

语法：`\Style[font-size=16pt, font-color=red]{内容}`

参数	简体	繁体	类型	默认值	说明
font-size	字号	字號	字号	(空)	覆盖字号
font	字体	字體	字符串	(空)	覆盖字体
font-color	字体颜色	字體顏色	颜色	(空)	覆盖字体颜色
grid-height	格高	格高	度量	(空)	覆盖格高
grid-width	格宽	格寬	度量	(空)	覆盖格宽
spacing-top	上间距	上間距	度量	(空)	上方附加间距
spacing-bottom	下间距	下間距	度量	(空)	下方附加间距
spacing	间距	間距	快捷	—	同时设置上/下间距
xshift	横移	橫移	度量	(空)	X方向平移
yshift	纵移	縱移	度量	(空)	Y方向平移
padding-top	上填充	—	度量	(空)	上方填充
padding-bottom	下填充	—	度量	(空)	下方填充
padding	填充	—	快捷	—	同时设置上/下填充

参数	简体	繁体	类型	默认值	说明
indent	缩进	縮進	整数	(空)	非首行缩进（格数）
first-indent	首行缩进	首行縮進	整数	(空)	首行缩进（格数）
bottom-indent	末行缩进	末行縮進	整数	(空)	末行缩进（格数）

环境 `StyleEnv` `样式设置` `樣式設置`

`\Style` 的环境形式，允许内容中使用命令展开。

行/列 (Column)

文件：`core/luatex-cn-core-column.sty`

`\Column` `\行`

单列竖排文本，支持对齐方式控制。

语法：`\Column[align=center, font-size=16pt]{内容}`

参数	简体	繁体	类型	默认值	说明
align	对齐	對齊	选项	top	对齐方式：`top` `bottom` `center` `stretch`
last	末行	末行	是/否	false	放置在当前半页末列
font-size	字号	字號	字号	(空)	覆盖字号
font	字体	字體	字符串	(空)	覆盖字体
font-color	字体颜色	字體顏色	颜色	(空)	覆盖字体颜色（别名 `color` ／ `颜色` ／ `顏色`）
width	宽度	寬度	度量	(空)	列宽
column-width	行宽	行寬	度量	(空)	列宽（别名）
auto-width	自动宽度	自動寬度	是/否	true	自动计算列宽

参数	简体	繁体	类型	默认值	说明
width-scale	宽度缩放	寬度縮放	字符串	1.2	自动宽度缩放系数
spacing-top	上间距	上間距	度量	(空)	列顶部间距
spacing-bottom	下间距	下間距	度量	(空)	列底部间距
spacing	间距	間距	快捷	—	同时设置上/下间距
padding-top	上填充	—	度量	(空)	列内上方填充
padding-bottom	下填充	—	度量	(空)	列内下方填充
padding	填充	—	快捷	—	同时设置上/下填充
grid-height	格高	格高	度量	(空)	覆盖格高

`\LastColumn` `\末行`

放置内容到当前半页末列。等价于 `\Column[last]{内容}`。

文本流 (TextFlow)

文件: `core/luatex-cn-core-textflow.sty`

`\TextFlow` `\文本流`

双行小注（夹注）排版。将内容分为左右两小列排版。

语法: `\TextFlow[font-size=8pt]{注释内容}`

参数	简体	繁体	类型	默认值	说明
only-column	单列	單列	选项	none	仅使用单列: `none` (双列) `right` `left`
auto-balance	自动均衡	自動均衡	是/否	true	自动均衡左右列字数（也可写 `自动平衡` ／ `自動平衡` ）

参数	简体	繁体	类型	默认值	说明
font-size	字号	字號	字号	(空)	覆盖字号
font	字体	字體	字符串	(空)	覆盖字体
font-color	字体颜色	字體顏色	颜色	(空)	覆盖字体颜色（别名 `color` ／ `颜色` ／ `顏色` ）
align	对齐	對齊	字符串	(空)	对齐方式
grid-height	格高	格高	度量	(空)	小字格高
n-char-per-col	每行字数	每行字數	整数	(空)	每小列字数，格高取「正文列高 ÷ N」（v0.4.4+，#82）

`\DanHangJiaZhu` `\单行夹注` `\單行夾注`

单列夹注。语法：`\DanHangJiaZhu[right]{内容}`，可选参数为 `right`（默认）或 `left`，即只用右小列或左小列。

环境 `TextFlowEnv` `文本流环境` `文本流環境`

`\TextFlow` 的环境形式。

文本框 (TextBox)

文件: `core/luatex-cn-core-textbox.sty`

`\TextBox` `\文本框`

多列网格文本框，支持边框、悬浮定位、背景色等。

语法：`\TextBox[height=5, border, background-color=yellow]{内容}`

参数	简体	繁体	类型	默认值	说明
height	高度	高度	整数	0	文本框高度（格数）

参数	简体	繁体	类型	默认值	说明
n-cols	列数	列數	整数	0	内部列数
outer-cols	外列数	外列數	整数	0	外部占用列数
fill	填充	填充	快捷	—	均分排布内容（= `box-align=fill`）
box-align	框对齐	框對齊	字符串	top	文本框在列中的对齐
border	边框	邊框	是/否	false	显示边框
border-shape	边框形状	邊框形狀	选项	none	边框形状：`none` `rect` `octagon` `circle`
border-color	边框色	邊框色	颜色	(空)	边框颜色
border-width	边框宽	邊框寬	度量	0.4pt	边框线宽
border-margin	边框间距	邊框間距	度量	0pt	边框与内容间距（统一）
border-margin-x	边框横距	邊框橫距	度量	(空)	边框横向间距
border-margin-y	边框纵距	邊框縱距	度量	(空)	边框纵向间距
border-offset	—	—	度量	0pt	边框画在底色之外的距离（`\墨围反白` 用 0.1em 留出白缝）
outer-margin	外间距	外間距	度量	0pt	文本框外部间距
outer-border	外边框	外邊框	是/否	false	显示外边框
outer-border-thickness	外边框粗细	外邊框粗細	度量	1pt	外边框线宽
outer-border-sep			度量	2pt	外边框与边框间距

参数	简体	繁体	类型	默认值	说明
	外边框间距	外邊框間距			
banxin	版心	版心	是/否	—	是否显示版心线
column-align	列对齐	列對齊	字符串	(空)	内部列对齐
vertical-align	纵对齐	縱對齊	字符串	(空)	纵向对齐（别名 box-align）
floating	悬浮	懸浮	是/否	false	悬浮定位模式
floating-paper-width	悬浮纸宽	懸浮紙寬	度量	0pt	悬浮模式下的纸宽
x	横坐标	橫坐標	度量	0pt	悬浮 X 坐标
y	纵坐标	縱坐標	度量	0pt	悬浮 Y 坐标
background-color	底色	底色	颜色	(空)	背景色
font-color	字体颜色	字體顏色	颜色	(空)	字体颜色
font-size	字号	字號	字号	(空)	字号
grid-width	格宽	格寬	度量	(空)	内部网格宽度
grid-height	格高	格高	度量	(空)	内部网格高度
padding-top	—	—	度量	(空)	框内上方填充（别名 `column-padding-top`、`fill-padding-top`）
padding-bottom	—	—	度量	(空)	框内下方填充（别名 `column-padding-bottom`、`fill-padding-bottom`）

`fill` / `填充` 是快捷参数，等价于 `box-align=fill`（均分排布）。`font-color` 也可写 `color` / `颜色`。

TextBox 参数补充 v0.3.6+:

参数	简体	繁体	类型	默认值	说明
transparent	透明	透明	是/否	true	背景是否透明（false 时使用页面背景色填充）

`\textboxSetup` `\文本框设置` `\文本框設置`

设置之后所有 `\TextBox` 的默认参数，参数同上。

`\FloatingTextBox` `\悬浮文本框`

悬浮文本框，等价于 `\TextBox[floating=true, …]{内容}`，用 `x` / `y` 指定在页面上的位置。

`\FillTextBox` `\填充文本框`

向后兼容的文本框命令，接受整数高度（旧式）或 key-value 参数。

`\反白` `\inverted`

反白效果快捷命令：黑底白字。语法：`\反白[参数]{内容}`

`\反白`、`\墨围反白`、`\八角墨围反白` 默认是行内实现（v0.4.4+）：跨列时逐字换列，底色与边框按每列连续的一段分别绘制。行内实现认的参数如下；写了其他文本框参数（如 `高度`、`填充`）会退回整块文本框（不能跨列断开），并在日志中给出警告（v0.4.5+）。

参数	简体	繁体	类型	默认值	说明
font-color	字体颜色	字體顏色	颜色	white	字色（别名 `color` / `颜色` / `顏色`）
background-color	底色	底色	颜色	0.2 0.2 0.2	底色
border-color				(空)	边框颜色

参数	简体	繁体	类型	默认值	说明
	边框色	邊框色	颜色		
border-width	边框宽	邊框寬	度量	0.4pt	边框线宽
break	断口	斷口	选项	square	跨列断口：`square`（直角）或 `octagon`（八角，每段都画完整八角）
border-shape	边框形状	邊框形狀	选项	—	`none` / `rect` / `octagon` 仍走行内实现（v0.4.5+，#210）；`circle` 退回整块
inline	可断行	可斷行	是/否	true	`false` 时有意使用整块文本框，不发警告

`\墨围` `\bordered` `\墨圍`

墨围（细边矩形框），用于强调标注（如"索隐"、"集解"等）。语法：`\墨围{内容}`

`\牌记` `\colophon` v0.3.6+

牌记命令：不透明双层边框悬浮文本框，用于排版古籍牌记页。

语法：`\牌记[key=value]{牌记内容}`

`\墨围反白` `\borderedInverted` `\墨圍反白`

矩形黑色边框 + 黑底白字。语法：`\墨围反白{内容}`

`\八角墨围` `\octagonBordered` `\octagon` `\八角墨圍`

八角形黑色边框。语法：`\八角墨围{内容}`

`\八角墨围反白` `\octagonBorderedInverted` `\八角墨圍反白`

八角形黑色边框 + 黑底白字。语法：`\八角墨围反白{内容}`

`\带圈` `\circled` `\帶圈`

圆形边框。语法：`\带圈{内容}`

> `\八角墨围` 另有英文别名 `\octagon`。墨围、八角墨围、带圈的装饰框随所在字号等比缩放（v0.4.4+）。

表格 (Table) v0.3.4+

文件: `core/luatex-cn-core-table.sty`

环境 `Table` `表格`

内联表格环境，嵌入在 `\begin{正文}` 中使用。

参数	简体	繁体	类型	默认值	说明
n-bands	分栏	分欄	整数	2	分栏数
band-gap	栏间距	欄間距	度量	0pt	栏间距
band-heights	栏高度	欄高度	列表	(空)	各栏高度（逗号分隔）
n-columns	行数	行數	字符串	(空)	每栏列数
header-rows	表头栏数	表頭欄數	整数	0	表头栏数
column-border	行边框	行邊框	字符串	(空)	列边框样式
band-border	栏边框	欄邊框	字符串	(空)	栏间分隔线
column-fill	行填充	行填滿	字符串	(空)	列填充方式：`page`(页/頁)

`\Cell` `\单元格` `\單元格`

表格单元格。语法：`\Cell[col-width=2]{内容}`

参数	简体	繁体	类型	默认值	说明
col-width	行宽	行寬	数字	1	列跨度（支持小数）
vertical-align	纵对齐	縱對齊	字符串	(空)	纵向对齐
column-border	行边框	行邊框	字符串	(空)	本格列边框

`\BandFormat` `\栏格式` `\欄格式`

设置当前栏格式，在栏第一个 `\Cell` 之前使用。

参数	简体	繁体	类型	默认值	说明
column-border	行边框	行邊框	字符串	(空)	当前栏列边框
padding-top	上填充	—	度量	(空)	栏顶部填充
padding-bottom	下填充	—	度量	(空)	栏底部填充
padding	填充	—	度量	(空)	同时设置上下填充

`\换栏` `\換欄`

正文分栏模式中强制切换到下一个栏。

段落 (Paragraph)

文件: `core/luatex-cn-core-paragraph.sty`

环境 `Paragraph` `段落`

块级缩进环境，支持首行/悬挂/末行缩进。

```
\begin{段落}[缩进=2，首行缩进=0]
  第一列缩进0格……\\
  后续列缩进2格……
\end{段落}
```

参数	简体	繁体	类型	默认值	说明
indent	缩进	縮進	整数	0	基础缩进（格数）
first-indent	首行缩进	首行縮進	整数	(空)	首行缩进（空=跟随 indent，明确值覆盖）
bottom-indent	末行缩进	末行縮進	整数	0	行尾缩进（行末要空几格）

`\Indent` `\缩进` `\縮進`

强制设置当前行缩进值，在 `\\` 或行尾自动恢复。负值等同抬头（伸入天头）。

语法：`\缩进[3]`（默认 0）

`\抬头` `\TaiTou`

另起一列，文字提升 N 格。N=0 为平抬，N>0 伸入天头区域。

语法：`\抬头[2]`（提升2格）

快捷命令	英文	等价于
`\平抬`	`\PingTai`	`\抬头[0]`
`\单抬` `\單抬`	`\DanTai`	`\抬头[1]`
`\双抬` `\雙抬`	`\ShuangTai`	`\抬头[2]`
`\三抬`	`\SanTai`	`\抬头[3]`

`\挪抬` `\NuoTai`

不换行，在当前位置插入 N 个全角空格。

语法：`\挪抬[2]`（插入2格空白）

快捷命令	英文	等价于
`\空抬`	`\KongTai`	`\挪抬[1]`

`\相对抬头` `\XiangDuiTaiTou` `\相對抬頭`

相对当前缩进位置上移 N 格。如当前 indent=2，`\相对抬头[3]` 则 indent=-1（伸入天头1格）。

`\换行` `\HuanHang` `\換行`

强制换列。

侧批 (Sidenote)

文件: `core/luatex-cn-core-sidenote.sty`

`\Sidenote` `\侧批` `\側批`

侧批注释，渲染在页面边缘。也可用拼音别名 `\CePi`。

语法:

- 基础: `\Sidenote{批注内容}`
- 位置参数（自动转换为 em）: `\Sidenote[5pt]{高度调整}` 等同于 `\Sidenote[yshift=5pt]{高度调整}`
- 数值自动转换为 em: `\Sidenote[0.5]{内容}` 等同于 `\Sidenote[0.5em]{内容}`
- 命名参数: `\Sidenote[yshift=5pt, color=blue]{批注内容}`

参数	简体	繁体	类型	默认值	说明
sidenote	侧批	側批	是/否	true	是否启用侧批输出 v0.3.6+
yshift	纵移	縱移	度量	0pt	纵向偏移（支持位置参数）
xshift	横移	橫移	度量	0pt	水平偏移 v0.3.5+
grid-height	格高	格高	度量	(空)	侧批格高（默认=字号）
font-size	字号	字號	字号	10pt	侧批字号
color	颜色	顏色	颜色	red	侧批颜色（别名 `font-color` ／ `字体颜色` ／ `字體顏色`）
background-color	底色	背景色	颜色	(空)	侧批底色（简体也可写 `背景色`）
column-padding-top	栏上填充	—	度量	0pt	上方填充
column-padding-bottom	栏下填充	—	度量	0pt	下方填充

另外，`punct-mode`（`标点模式`）等标点设置也可直接写在侧批参数里，只对这条侧批生效，例如 `\侧批[标点模式=judou]{…}`。

`\sidenoteSetup` `\侧批设置` `\側批設置`

配置侧批默认参数。

脚注 (Footnote)

文件: `core/luatex-cn-footnote.sty`

`\Footnote` `\脚注` `\腳注`

插入脚注。Mode 1（段末注）：存储到序列，在 `\FlushFootnote` 时输出。Mode 2（页下注）：注册到 Lua 层，页面渲染时自动输出。

语法：`\Footnote{校勘记内容}`

`\FlushFootnote` `\输出脚注` `\輸出腳注`

输出所有已存储的脚注并重置计数器。仅 Mode 1（段末注）有效。

`\footnoteSetup` `\脚注设置` `\腳注設置`

配置脚注参数。

参数	简体	繁体	类型	默认值	说明
mode	模式	模式	选项	endnote	脚注模式：`endnote`(段末注) `page`(页下注)
number-style	编号样式	編號樣式	选项	lujiao	编号风格：`lujiao`(【一】) `circled`(①)
separator	分隔符	分隔符	选项	blank	各条之间的分隔：`blank`（1em 空白） `none`
font	字体	字體	字符串	(空)	脚注字体
font-size	字号	字號	字号	0.8em	脚注字号（相对于正文字号）

参数	简体	繁体	类型	默认值	说明
font-color	字体颜色	字體顏色	颜色	(空)	脚注字体颜色
indent	缩进	縮進	整数	2	悬挂缩进首行空格数
spacing	间距	間距	度量	0.5em	脚注间距
marker-height	序号高度	序號高度	整数	2	标记符号占用行数（格数）
counter-start	起始编号	起始編號	整数	(空)	脚注计数器起始值

`\FootnoteParagraph` `\脚注段落` `\腳注段落`

无编号脚注段落，用于独立注释块（如"索隱曰"、"正義曰"等），不带序号标记。

语法：`\FootnoteParagraph{索隱紀者記也……}`

`\ResetFootnoteCounter` `\重置脚注计数` `\重置腳注計數`

重置脚注计数器（Mode 2 页下注按页重置时使用）。

元数据 (Metadata)

文件：`core/luatex-cn-core-metadata.sty`

`\metadataSetup` `\元数据设置` `\元數據設置`

配置书籍元数据。

参数	简体	繁体	类型	默认值	说明
book-name	书名	書名	字符串	(空)	书名
chapter-title	章节标题	章節標題	字符串	(空)	章节标题
publisher	出版商	出版商	字符串	(空)	出版商名称

`\title`

设置书名（同步到版心和元数据）。

`\publisher` `\出版商`

设置出版商。

`\ChapterMarker` `\章节标记` `\章節標記`

插入章节标记，后续页面的版心章节标题自动更新。

`\chapter` `\卷`

设置章节标题并重置页码。v0.3.5+ 支持可选参数区分版心与正文标题。

语法：

- `\chapter{标题}` — 设置章节标题并输出正文
- `\chapter*{标题}` — 仅设置元数据，不输出
- `\chapter[版心标题]{正文标题}` — 版心显示"版心标题"，正文输出"正文标题"
- `\chapter*[版心标题]{正文标题}` — 设置版心但不输出正文

`\section` v0.3.5+

小节标题（篇级）。语法：`\section{标题}` / `\section*{标题}`（带星号仅设置元数据）。

`\subsection` v0.3.5+

子节标题（节级）。语法：`\subsection{标题}` / `\subsection*{标题}`（带星号仅设置元数据）。

`\sectionSetup` `\标题设置` `\標題設置` v0.3.5+

配置多级标题参数。

参数	简体	繁体	类型	默认值	说明
chapter-indent	章节缩进	章節縮進	整数	0	

参数	简体	繁体	类型	默认值	说明
					章节标题缩进（格数）
chapter-font-size	章节字号	章節字號	字号	(空)	章节标题字号
chapter-font-color	章节字体颜色	章節字體顏色	颜色	(空)	章节标题颜色
chapter-font	章节字体	章節字體	字符串	(空)	章节标题字体
chapter-spacing-top	—	—	度量	(空)	章节前附加间距
section-indent	篇缩进	—	整数	1	小节标题缩进
section-font-size	篇字号	—	字号	(空)	小节字号
section-font-color	篇字体颜色	—	颜色	(空)	小节颜色
section-font	篇字体	—	字符串	(空)	小节字体
section-spacing-top	—	—	度量	(空)	小节前附加间距
subsection-indent	节缩进	節縮進	整数	2	子节标题缩进
subsection-font-size	节字号	節字號	字号	(空)	子节字号
subsection-font-color	节字体颜色	節字體顏色	颜色	(空)	子节颜色
subsection-font	节字体	節字體	字符串	(空)	子节字体
subsection-spacing-top	—	—	度量	(空)	子节前附加间距

组件级语义设置 (Settings)

文件: `core/luatex-cn-core-settings.sty`

没有单独的全局设置命令（旧版 wiki 写的 `\GlobalSettings` ／ `\全局设置` 并不存在）。下面这几个键可以直接写在侧批、批注、眉批、文本框、`\样式` 等组件的可选参数里，只对该组件生效：

参数	简体	繁体	类型	默认值	说明
punct-mode	标点模式	標點模式	选项	(空)	标点模式：`normal` `judou` `none`
punct-style	标点标准	標點標準	选项	(空)	标点风格：`mainland` `taiwan`
debug	调试	調試	是/否	(空)	启用调试输出

全文的标点模式用 `\正常标点模式` ／ `\句读模式` ／ `\无标点模式` 切换（见「句读」一节），标点风格用 `\标点设置{标点风格=…}`。

文档样式 (Document)

文件: `core/luatex-cn-core-document.sty`

`\documentSetup` `\文档设置` `\文檔設置`

设置文档级默认样式（样式栈的最底层），各组件未指定时沿用。只有英文键：

参数	类型	默认值	说明
font-color	颜色	(空)	全文字色
font-size	字号	(空)	全文字号
font	字符串	(空)	全文字体
background-color	颜色	(空)	页面底色

标点 (Punct)

文件: `core/luatex-cn-core-punct.sty`

`\punctSetup` `\标点设置` `\標點設置`

配置现代标点处理规则。

参数	简体	繁体	类型	默认值	说明
punct-style				taiwan	标点位置风格：`taiwan` `mainland`

参数	简体	繁体	类型	默认值	说明
	标点风格	標點風格	选项		
punct-squeeze	标点压缩	標點壓縮	是/否	true	连续标点压缩
punct-hanging	标点悬挂	標點懸掛	是/否	false	标点悬挂
kinsoku	禁则	禁則	是/否	true	禁则处理
western-space	中西间距	中西間距	是/否	true	竖排中西文之间加 1/4em 间距
style	风格预设	風格預設	选项	—	clreq 预设：`mainland` `taiwan` `none`（与横排同名同值，见 标点系统）
squeeze-mode	宽度调整	寬度調整	选项	legacy	标点挤压算法：`legacy` `context`（上下文相关）
adjacent-punct	连续标点	連續標點	选项	1.5	相邻标点合计宽度：`1.5` `1` `natural`
line-start-bracket	—	—	选项	trim	列首开始夹注符号（开括号、开引号）：`trim`（缩减始侧半字）`natural`
line-end-punct	—	—	选项	compress	列末点号：`compress`（挤半宽）`natural`

`\大陆标点风格` `\DaLuBiaoDianFengGe` `\大陸標點風格`

快捷命令：切换到中国大陆标点风格。

`\台湾标点风格` `\TaiWanBiaoDianFengGe` `\臺灣標點風格`

快捷命令：切换到台湾标点风格。

Guji 古籍宏包

古籍专用功能：句读、夹注、批注、眉批、印章、书名页、封面、目录条目。

古籍文档类 (Guji Class)

文件: `ltc-guji.cls`

`\gujiSetup` `\古籍设置` `\古籍設置`

古籍文档类的配置命令：

参数	简体	繁体	类型	默认值	说明
template	—	—	字符串	—	加载预设模板

除 `template` 外，也可以直接写版面参数（v0.4.5+）：`\gujiSetup{ n-column = 10, margin-top = 80pt }` 会把每个键交给 `\pageSetup` ／ `\contentSetup` ／ `\banxinSetup` 中认识它的那个；都不认识的键发警告、不报错。

内置模板:

英文	简体	繁体
default	四库全书	四庫全書
SiKuQuanShu-colored	四库全书彩色	四庫全書彩色
HongLouMengJiaXuBen	红楼梦甲戌本	紅樓夢甲戌本
oushizupu	欧式族谱	歐式族譜

中英文模板名在文档类选项和 `\gujiSetup{template=…}` 里都可用（v0.4.5+），如 `\documentclass[四库全书彩色]{ltc-guji}` 或 `\gujiSetup{template=红楼梦甲戌本}`。`ltc-guji-digital` 同样适用，模板不存在时只发警告。

`\defineGujiTemplate` `\定义古籍模板` `\定義古籍模板`

给 `\gujiSetup` 定义一个模板名，模板体按 `\gujiSetup` 的键解析，所以可以写 `template`（在某个模板基础上改）和各种版面参数（v0.4.5+）：

```
\defineGujiTemplate{窄版}{ template = default, n-column = 5, font-size = 20pt, margin-top = 100pt, border = true }
\gujiSetup{ template = 窄版 }
```

版式改动较多、要在多个文档间共用时，仍建议用配置文件（见 模板与自定义）。

句读 (Judou)

文件: `guji/luatex-cn-guji-judou.sty`

`\judouSetup` `\句读设置` `\句讀設置`

配置句读显示参数。

参数	简体	繁体	类型	默认值	说明
judou-pos	句读位置	句讀位置	字符串	right-bottom	句读符号位置
judou-size	句读大小	句讀大小	字号	1em	句读符号大小
judou-color	句读颜色	句讀顏色	颜色	red	句读颜色（别名 `color` ／ `font-color` ／ `颜色` ／ `字体颜色` ）
punct-mode	标点模式	標點模式	选项	normal	标点模式

`\NormalPunctuationMode` `\正常标点模式` `\正常標點模式`

切换到正常标点模式。

`\JudouPunctuationMode` `\句读模式` `\句讀模式`

切换到句读标点模式。

`\NonePunctuationMode` `\无标点模式` `\無標點模式`

移除所有标点。

夹注 (JiaZhu)

文件: `guji/luatex-cn-guji-jiazhu.sty`

`\JiaZhu` `\夹注` `\夾注`

双行小字注释。语法: `\JiaZhu[font-size=8pt]{注释内容}`

参数	简体	繁体	类型	默认值	说明
font-size	字号	字號	字号	(空)	夹注字号
font	字体	字體	字符串	(空)	夹注字体
font-color	字体颜色	字體顏色	颜色	(空)	夹注字体颜色
align	对齐	對齊	字符串	outward	对齐方式
grid-height	格高	格高	度量	(空)	夹注格高（不给则按字号）
n-char-per-col	每行字数	每行字數	整数	(空)	夹注每小列字数，格高 = 正文列高 ÷ N；与 grid-height 同时给时以 grid-height 为准（v0.4.4+）

`\jiazhuSetup` `\夹注设置` `\夾注設置`

配置夹注默认参数，可写上表全部参数（含 `n-char-per-col`）。

环境 `JiaZhuEnv` `夾注环境` `夾注環境`

夹注的环境形式。

`\jiazhuShortcut` `\夹注捷径` `\夾注捷徑`

注册一对括号符号作为夹注快捷输入。注册后，源代码中匹配的括号对会自动替换为 `\夹注{...}`。

语法:

- `\jiazhuShortcut` — 默认注册 `【...】`
- `\jiazhuShortcut{（}{）}` — 自定义括号对
- `\jiazhuShortcut[font-color=red]{（}{）}` — 带样式参数

可选参数支持所有夹注参数（font-size、font、font-color、align 等）。

批注 (PiZhu)

文件: `guji/luatex-cn-guji-pizhu.sty`

`\PiZhu` `\批注`

浮动批注框。语法: `\PiZhu[x=5cm, y=3cm]{批注内容}`

参数	简体	繁体	类型	默认值	说明
pizhu	批注	批註	是/否	true	是否启用批注输出 v0.3.6+
x	横坐标	橫坐標	度量	(空)	X 坐标
y	纵坐标	縱坐標	度量	(空)	Y 坐标
height	高度	高度	整数	(空)	批注高度（格数）
font-size	字号	字號	字号	18pt	批注字号
color	颜色	顏色	颜色	1 0 0	批注颜色（别名 `font-color` / `字体颜色`）

参数	简体	繁体	类型	默认值	说明
grid-width	格宽	格寬	度量	20pt	批注格宽
grid-height	格高	格高	度量	19pt	批注格高

`\pizhuSetup` `\批注设置` `\批注設置`

配置批注默认参数。

眉批 (MeiPi)

文件: `guji/luatex-cn-guji-meipi.sty`

`\MeiPi` `\眉批`

自动定位的眉批注释，出现在页面天头区域。

语法: `\MeiPi[font-size=14pt]{眉批内容}`

参数	简体	繁体	类型	默认值	说明
meipi	眉批	眉批	是/否	true	是否启用眉批输出 v0.3.6+
x	横坐标	横坐標	度量	(空)	X 坐标覆盖
y	纵坐标	縱坐標	度量	(空)	Y 坐标覆盖
height	高度	高度	整数	7	眉批高度（格数）
font-size	字号	字號	字号	18pt	眉批字号
color	颜色	顏色	颜色	1 0 0	眉批颜色（别名 `font-color` ／ `字体颜色` ）
grid-width	格宽	格寬	度量	20pt	眉批格宽
	格高	格高	度量	19pt	眉批格高

参数	简体	繁体	类型	默认值	说明
grid-height					
spacing	间距	間距	度量	10pt	眉批间距
gap	间隔	間隔	度量	0pt	眉批额外间隔

`\meipiSetup` `\眉批设置` `\眉批設置`

配置眉批默认参数。

印章 (YinZhang)

文件: `guji/luatex-cn-guji-yinzhang.sty`

`\YinZhang` `\印章`

在指定页面覆盖馆藏印/私印图像。

语法: `\YinZhang[opacity=0.5, page=2]{seal.png}`

参数	简体	繁体	类型	默认值	说明
opacity	不透明度	不透明度	字符串	1	印章不透明度（0-1）
xshift	横移	橫移	度量	0cm	水平偏移
yshift	纵移	縱移	度量	0cm	垂直偏移
width	宽度	寬度	度量	10cm	印章宽度
page	页	頁	整数	1	目标页码
color	颜色	顏色	颜色	(空)	印章着色

`\yinzhangSetup`（`\印章设置` / `\印章設置`）设置以上参数的默认值。

书名页 (TitlePage)

文件: `guji/luatex-cn-guji-titlepage.sty`

环境 `TitlePage` `书名页` `書名頁`

古籍书名页，自动处理列布局。

参数	简体	繁体	类型	默认值	说明
border	边框	邊框	是/否	false	显示边框
outer-border	外边框	外邊框	是/否	false	显示外边框

单页 (SinglePage)

文件: `guji/luatex-cn-guji-danye.sty`

环境 `SinglePage` `单页` `單頁`

单页模式（筒子页的半页），自动计算纸宽/2 和边距/2。无专属参数，继承页面（Page）参数。

环境 `EmptySinglePage` `空白页` `空白頁`

无边框（无丝栏、无外框）的单页，内容可空。语法：`\begin{空白页}[参数] … \end{空白页}`。

封面 (Cover)

文件: `guji/luatex-cn-guji-cover.sty`

环境 `Cover` `封面`

封面页环境。自动关闭边框和版心。

参数	简体	繁体	类型	默认值	说明
background-color	底色	底色	颜色	(空)	封面背景色

`\ThreadBoundCover` `\线装书封面` `\線裝書封面` (v0.4.3+)

一页现代蓝色线装书封面（内部即 `\begin{Cover}`）。语法：`\线装书封面[键值]{书名}`

参数	简体	繁体	类型	默认值	说明
volume	册次	冊次	文本	(空)	签条下部小字，如「第一册」
juan	卷次	卷次	文本	(空)	签条下部小字，如「卷一」；与册次并排两列
small-position	小字位置	小字位置	bottom / follow	bottom	`底部` / `随后`（`隨後`）：小字排在签条底部或紧随书名
binding-thread	订线	訂線	是/否	false	画出订线（仅供预览）
thread-holes	订线眼数	訂線眼數	整数	4	订眼数（四眼／六眼）
thread-distance	订线距	訂線距	度量	页宽 × 0.08	订线距页面右边
thread-color	订线颜色	訂線顏色	颜色	236, 236, 230	订线颜色
background-color	底色	底色	颜色	30, 50, 90	封面色
label-color	签条底色	簽條底色	颜色	232, 228, 212	签条纸色
ink-color	墨色	墨色	颜色	25, 25, 25	书名、小字、框线颜色
label-width	签条宽	簽條寬	度量	页宽 × 0.23	签条宽
label-left	签条左距	簽條左距	度量	页宽 × 0.05	签条左缘距页面左边
label-top			度量		签条上缘距页面上边

参数	简体	繁体	类型	默认值	说明
	签条上距	簽條上距		页高 × 0.04	
label-min-height	签条最小高	簽條最小高	度量	页高 × 0.5	签条最小高度，书名长则增高
title-font-size	书名字号	書名字號	度量	签条宽 × 0.56	书名字号
small-font-size	小字字号	小字字號	度量	书名字号 × 0.3	册次／卷次字号

长度类键不给（或给 `0pt`）时按页面比例自动取值。

目录条目 (Mulu)

文件: `guji/luatex-cn-guji-mulu.sty`

`\条目` `\TiaoMu` `\條目`

输出一个目录/索引条目，支持缩进层级 (0-4)。

语法: `\条目[2]{卷三 论语}`

Banxin 版心宏包

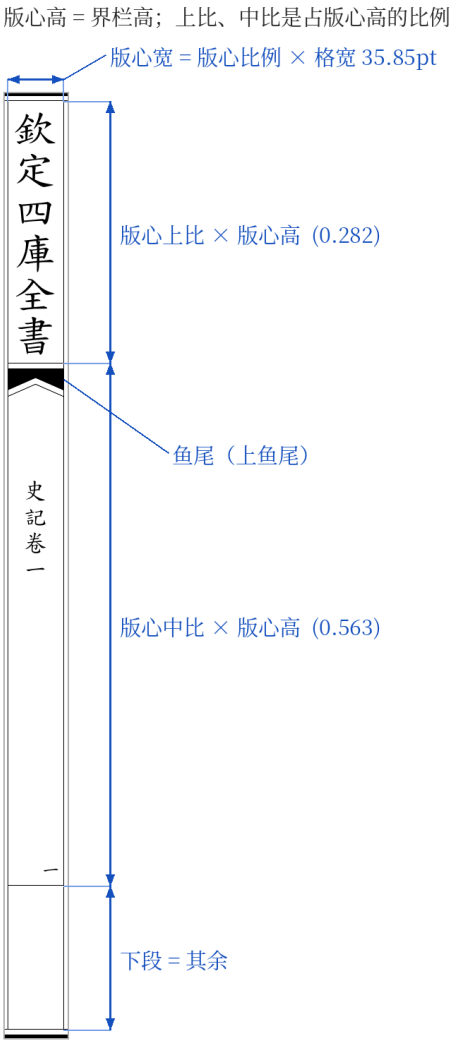
版心装饰功能：鱼尾、分隔线、书名/章节/页码/出版商标注。

版心 (Banxin)

文件: `banxin/luatex-cn-banxin.sty`

`\banxinSetup` `\版心设置` `\版心設置`

配置版心参数。版心包含上鱼尾、书名区、分隔线、章节标题区、页码区、下鱼尾、出版商区。



主参数:

参数	简体	繁体	类型	默认值	说明
banxin	版心	版心	是/否	false（`ltc-guji` 默认开启）	是否启用版心
banxin-upper-ratio	版心上比	版心上比	字符串	0.28	上部（书名区）占比（0-1）
banxin-middle-ratio	版心中比	版心中比	字符串	0.56	中部（卷名区）占比（0-1）。上比 + 中比须 ⩽ 1，余下的是下段（页码区）
banxin-divider	版心分隔线	版心分隔線	是/否	true	是否显示分隔线
upper-yuwei	上鱼尾	上魚尾	是/否	true	是否显示上鱼尾
lower-yuwei	下鱼尾	下魚尾	是/否	true	是否显示下鱼尾
middle-section-text	中部文字	—	字符串	(空)	中部文字（通常由 `\chapter` 自动设置）
upper-section	上部	—	文字或键值	—	上部（书名区）：写文字即书名，写键值则设下表参数
middle-section	中部	—	键值	—	中部（卷名区）参数
page-number	页码	頁碼	键值	—	页码区参数
lower-section	下部	—	文字或键值	—	下部（刻工／出版者区）：写文字即内容，写键值则设下表参数
book-ear	书耳	書耳	是/否、文字或键值	false	书耳，见下（v0.4.5+，#87）

书名默认取 `\title{...}`，卷名取 `\chapter{...}`。

上部参数（`upper-section={...}` ／ `上部={...}`）:

参数	简体	类型	默认值	说明
font-size	字号	字号	(空)	书名字号
grid-height	格高	度量	(空)	书名格高
align	对齐	字符串	center	书名对齐方式
top-padding	—	度量	2pt	上方留白
bottom-padding	—	度量	10pt	下方留白
background-color	底色	颜色	(空)	底色
font-color	字体颜色	颜色	(空)	字色

中部参数（`middle-section={...}` ／ `中部={...}`）：

参数	简体	繁体	类型	默认值	说明
font-size	字号	—	字号	(空)	卷名字号
grid-height	格高	—	度量	(空)	卷名格高
top-margin	上留白	—	度量	20pt	中部上方留白
cols	列数	列數	整数	1	卷名列数
align	对齐	對齊	字符串	center	对齐方式
background-color	底色	—	颜色	(空)	底色
font-color	字体颜色	—	颜色	(空)	字色

页码区参数（`page-number={...}` ／ `页码={...}`）：

参数	简体	类型	默认值	说明
show	显示	布尔	true	是否显示页码
font-size	字号	字号	14pt	页码字号
grid-height	格高	度量	(空)	页码格高
align				

参数	简体	类型	默认值	说明
	对齐	字符串	right-bottom	页码位置：`right-bottom` 版心下部靠右；`bottom-center` 版心下部横向居中；`center` 版心中部居中（紧接卷名之下，常见于明清刻本）
bottom-padding	—	度量	10pt	页码区底部留白（`right-bottom` / `bottom-center` 时生效）

```
\banxinSetup{ page-number = { align = center } }
\版心设置{ 页码 = { 对齐 = center } }
```

下部参数（`lower-section={...}` ／ `下部={...}`）：

参数	简体	类型	默认值	说明
font-size	字号	字号	10pt	字号
grid-height	格高	度量	(空)	格高
align	对齐	字符串	right	对齐
bottom-margin	下留白	度量	5pt	下方留白
top-padding	—	度量	2pt	上方留白
bottom-padding	—	度量	10pt	下方留白
background-color	底色	颜色	(空)	底色
font-color	字体颜色	颜色	(空)	字色

书耳（`book-ear` ／ `书耳` ／ `書耳`）v0.4.5+：

书耳是刻在边栏外上角的小框，记篇名。`书耳=true` 开启，文字默认取当前 `\chapter` 的篇名（换篇后跟着变）；也可直接写文字 `书耳=五帝本紀`，或写键值：

参数	简体	繁体	类型	默认值	说明
show	显示	顯示	是/否	false	是否显示（写了键值即开启）
text	文字	文字	字符串	(空)	书耳文字，空则取篇名
position			选项	left	

参数	简体	繁体	类型	默认值	说明
	位置	位置			`left`（左）`right`（右）`both`（两侧／兩側）。默认在左侧，筒子页拆分后只落在左半叶上
width	宽度	寬度	度量	0.8 × 列宽	框宽
height	高度	高度	度量	按字数	框高
font-size	字号	字號	字号	0.6 × 框宽	字号

```
\banxinSetup{ 书耳 = true }
\banxinSetup{ book-ear = { position = both, font-size = 9pt } }
```

Decorate 装饰宏包

字符装饰与线标功能。

装饰 (Decorate)

文件: `decorate/luatex-cn-decorate.sty`

`\decorate` `\装饰` `\裝飾`

对内容中的每个字符附加装饰符号。

语法: `\decorate[char=●, color=blue]{内容}`

参数	简体	繁体	类型	默认值	说明
char	字符	字符	字符串	。	装饰字符
color	颜色	顏色	颜色	red	装饰颜色
xshift	横偏 / 横移	橫偏 / 橫移	度量	0pt	水平偏移
yshift	纵偏 / 纵移	縱偏 / 縱移	度量	0pt	垂直偏移
font-size	字号	字號	字号	(空)	装饰字号
scale	缩放	縮放	字符串	1.0	装饰缩放比例

`\EmphasisMark` `\着重号` `\著重號`

着重号（圈点）快捷命令。逐字在右侧加着重标记。全局设置: `\着重号设置` / `\emphasisMarkSetup` / `\圈点设置`。

参数	简体	繁体	类型	默认值	说明
char	字符	—	字符	○	标记字符（vbook 两类预设 ●）
scale	缩放	縮放	数值	0.3	标记缩放比例

参数	简体	繁体	类型	默认值	说明
color	颜色	顏色	颜色	black	着重号颜色
skip-punct	跳过标点	跳過標點	是/否	false	标点上不加着重号（clreq 5.3.1，vbook 两类预设 true）

`\decorateSetup` `\装饰设置` `\裝飾設置`

设置 `\装饰` 的默认参数，参数同上。

`\faSheng` `\发圈` `\發圈` `\圈发` `\圈發`

发圈（四声圈点）：在字的四角画小圈标声调。语法：`\发圈[tone=shang]{字}`，或用快捷命令 `\平声{字}` `\上声{字}` `\去声{字}` `\入声{字}`（繁体 `\平聲` 等）。批量：`\发圈批{平上去入}{春夏秋冬}`（`\fashengBatch`，声调串与字串逐个配对）。默认参数用 `\发圈设置`（`\fashengSetup` ／ `\發圈設置`）。

参数	简体	繁体	类型	默认值	说明
tone	声	聲	选项	ping	`ping` `shang` `qu` `ru`（也可写 `平` `上` `去` `入`，或直接写 `上声` 当开关）
color	颜色	顏色	颜色	black	圈的颜色
char	字符	字符	字符	○	圈的字符
scale	缩放	縮放	数值	0.3	缩放比例
offset	偏移	偏移	度量	0.45em	离字中心的偏移

详见 改字与装饰。

`\SideText` `\旁注` `\旁註`

在字旁加小字（如注音、校字）。语法：`\旁注{正文}{右側小字}` 或 `\旁注{正文}{右側小字}{左側小字}`。默认参数用 `\旁注设置`（`\sideTextSetup` ／ `\旁註設置`）：`scale`（缩放，0.5）、`color`（颜色，black）、`offset`（偏移，0pt）。

`\fix` `\改`

改字命令。在前一字上叠加「、」删除标记，并在右下角添加替换字。

语法：`甲\改{乙}` — 表示"甲"应改为"乙"。

线标 (LineMark)

文件：`decorate/luatex-cn-linemark.sty`

`\LineMark` `\线标` `\線標`

通用线标命令，支持直线/波浪线样式。

语法：`\LineMark[type=wavy, color=red]{内容}`

参数	简体	繁体	类型	默认值	说明
type	类型	類型	选项	straight	线型：`straight` `wavy`
color	颜色	顏色	颜色	black	线色
offset	偏移	偏移	度量	0.6em	线距文字偏移
amplitude	振幅	振幅	字符串	medium	波浪振幅
line-width	线宽	線寬	度量	0.8pt	线宽
style	风格	風格	字符串	standard	渲染风格

`\ProperNameMark` `\专名号` `\專名號`

专名号（直线标记）。也可用 `\Underline` `\下划线` `\下劃線`。

`\BookTitleMark` `\书名号` `\書名號`

书名号（波浪线标记）。也可用 `\WavyUnderline` `\波浪线` `\波浪線`。

Fonts 字体宏包

文件: `fonts/luatex-cn-font-autodetect.sty`

根据操作系统自动检测并应用中文字体。

获取自动检测到的字体名称。

获取自动检测到的字体特性字符串。

`\setfontfamily` `\设置字体族` `\設置字體族`

一步设置主字体及递补链。语法: `\setfontfamily{Font1,Font2}`

`\setmainfont` `\设置字体` `\設置字體`

标准 fontspec 命令透传。

其他命令所在页面

- 横排（`\usepackage{luatex-cn}`）的 `\horiSetup`、行间标号、行间注、`\字号` 等：见 现代横排
- 数字化文档类 `ltc-guji-digital` 的 `\双列`、`\右小列`、`\左小列`、`数字化内容` 环境 等：见 数字化
- 工尺谱：见 工尺谱
- 排版坐标导出 `\enableLayoutExport`（`\开启排版导出`）：见 排版坐标导出
- 抬头相关命令：见 抬头

Debug 调试宏包

文件: `debug/luatex-cn-debug.sty`

`\LtcDebugOn` `\开启调试` `\开启调试模式` `\開啟調試`

全局启用调试输出。

`\LtcDebugOff` `\关闭调试` `\關閉調試`

全局关闭调试输出。

`\LtcDebugModuleOn` `\开启调试模块`

启用指定模块的调试输出。语法：`\LtcDebugModuleOn{render}`

`\LtcDebugModuleOff` `\关闭调试模块`

关闭指定模块的调试输出。

`\LtcDebugColor` `\调试颜色` `\調試顏色`

设置模块的调试输出颜色。语法：`\LtcDebugColor{render}{blue}`

`\LtcShowFrame` `\显示边框` `\顯示邊框`

显示页面文本区域边框。

`\LtcShowGrid` `\显示网格` `\显示坐标` `\顯示網格`

显示页面坐标网格。原点在右上角。

参数	简体	繁体	类型	默认值	说明
measure	单位	單位	选项	(空)	坐标单位：`cm` `pt` `mm`

`\LtcHideGrid` **`\隐藏网格`** **`\隐藏坐标`** **`\隱藏網格`**

隐藏页面坐标网格。

文档类 (Document Classes)

cn-vbook

文件：`ltc-cn-vbook.cls`

简体中文竖排书籍文档类。自动加载 `luatex-cn-core` 和默认模板 `luatex-cn-vbook-default.cfg`。

类选项中传入模板名可加载对应配置文件。

`\skipHeader` `\跳过页眉`

跳过当前页的页眉和页码显示（如标题页、封面页）。语法：`\skipHeader`

vbook 页面配置

通过 `\pageSetup` 配置 vbook 专有参数（配合 `header-enabled=true`）：

参数	简体	繁体	类型	默认值	说明
page-number-x	页码横移	頁碼橫移	度量	3mm	页码距纸边的横向距离
page-number-y	页码纵移	頁碼縱移	度量	52mm	页码纵向位置（`header-anchor=content` 时为相对偏移）
header-y	页眉纵移	頁眉縱移	度量	52mm	页眉纵向位置（`header-anchor=content` 时为相对偏移）
page-number-position	页码位置	頁碼位置	选项	outer	`outer`（外侧，翻口侧）`inner`（内侧，书脊侧）`center`（居中，页脚居中）（v0.4.4+）
header-position	页眉位置	頁眉位置	选项	outer	`outer`（外侧）`inner`（内侧）（v0.4.4+）
header-anchor	页眉锚点	頁眉錨點	选项	paper	`paper`（纸边）按纸边定位；`content`（内容区）书眉页码跟随内容区（v0.4.4+）

参数	简体	繁体	类型	默认值	说明
header-sep	页眉间距	頁眉間距	度量	6mm	`header-anchor=content` 时书眉页码离内容区外缘的距离（v0.4.4+）

> 竖排书右侧装订：奇数页是左页，书脊在其右侧。默认书眉页码在翻口侧（外侧）。详见 现代竖排。

`\卷名`

vbook 卷名标题（定义在 `configs/luatex-cn-vbook-common.cfg`）。语法：`\卷名{史記卷六}`。自动设置 24pt 字号和 48pt 上间距。

`\章节名`

vbook 章节名标题。语法：`\章节名{秦始皇本紀}`。自动设置 20pt 字号和首行 2 字缩进。

tw-vbook

文件：`ltc-tw-vbook.cls`

繁体中文竖排书籍文档类。加载 `luatex-cn-tw-vbook-default.cfg`。其他与 `ltc-cn-vbook` 相同。

ltc-guji

文件：`ltc-guji.cls`

古籍文档类。加载全部古籍功能模块（句读、夹注、批注、眉批、印章、书名页、封面等）。

更新日志

本项目的所有重大更改都将记录在此。

[0.4.5] - 2026-11-18

- ✨ **书耳**：`\banxinSetup{书耳=true}` 在版框左上角外侧画小框，竖排写本页篇名（随 `\chapter` 变），也可 `书耳=<文字>`；`位置 = 左 | 右 | 两侧`，宽度、高度、字号可调 (fix #87)
- ✨ **整页居中**：正文新增 `横对齐`（`horizontal-align`）`= 居右 | 居中 | 居左`；新环境 `居中页`（`CenteredPage`）排扉页、卷标题页。古籍网格版式下不生效并警告 (fix #175)
- ✨ **中文取值**：所有选项与是/否参数都可写中文值，如 `对齐=顶`、`页码位置=内侧`、`显示=是`，见命令索引 (fix #85)
- ✨ **参数覆盖提醒**：每行字数与格高、行数与格宽、版面行数与版心、字号大于格高、版心上比 + 中比 > 1 这几组互相覆盖的参数，同时设置时发警告，见命令索引 (fix #195)
- 📖 **版式图解**：新增古籍版式要素标注图与参数标注图 (fix #195)
- ♻ **模板设置**：`\gujiSetup` ／ `\defineGujiTemplate` 可直接写 `n-column`、字号、边距等版面参数；`template=` 接受模板中文名，未知模板改为警告
- 🐛 **夹注句读**：夹注设了 `n-char-per-col` 或 `grid-height` 时句读不再偏离所附的字 (fix #216)
- 🐛 **夹注缩进**：缩进段落里带 `n-char-per-col` 的夹注换列后不再冲出版框 (fix #212)
- 🐛 **反白 border-shape**：`\反白[border-shape=none]` 等写法不再退回整块文本框；因其他 key 退回时会警告 (fix #210)
- 🐛 **筒子页**：两个 `正文` 块之间不再多出一张两倍宽的整张筒子页
- 🐛 **版心字号**：`\版心设置{font-size=12pt}` 不再中止编译；`\LineMark[color=gray]` 颜色生效
- ♻ **代码清理**：删死代码、提取重复、luacheck 进 CI，排版结果不变

[0.4.4] - 2026-11-16

- ✨ **夹注每列字数**：`\夹注[n-char-per-col=20]`（`每行字数`）— 夹注每小列字数可与正文不同，夹注格高取「正文列高 ÷ N」；`\夹注设置{n-char-per-col=20}` 作全书默认 (fix #82)
- ✨ **对页补页**：`\pageSetup{对页补页=true}`（`facing-pages`）— 章落在偶数页顶端时自动补一页，使每章从奇数页起；`\补空白页`（`\InsertBlankPage`）在任意处强制补一页；`补页样式=空白`（`blank-page-style=blank`）让补出来的页完全空白。默认关闭 (fix #176)
- ✨ **对页固定内容区**：`镜像边距 = false`（`mirror-margins`）让 `twoside = true` 时奇偶页都用左右边距，内容区固定；`装订边距`（`binding-offset`）在书脊一侧加装订边距；现代竖排类新增 `页眉锚点 = 内容区`（`header-anchor = content`）、`页眉间距`（`header-sep`）、`页眉位置`（`header-position`），书眉页码跟着内容区走 (fix #198)
- ✨ **现代竖排页码位置**：`页码位置 = 外侧 | 内侧 | 居中`（`page-number-position`），默认外侧 (fix #193)
- ♻️ **竖排书内外侧按书脊算**：竖排书右侧装订，奇数页是左页。`margin-inner` 与 `页码位置=内侧` 此前落在翻口一侧，现改按书脊算；vbook 默认值同步互换为 inner 24mm／outer 16mm，默认版面不变。自己写了 `margin-inner ≠ margin-outer` 或 `页码位置` 的文档左右会对调，把两个值互换即可 (fix #198)
- 🐛 **反白跨列**：`\反白` ／ `\墨围反白` ／ `\八角墨围反白` 跨列时逐字换行，不再整块推到下一列；新增 `断口=直角|八角` (fix #201)
- 🐛 **vbook 提前换页**：`\include` 多个文件、或 `\clearpage` ／ `\换页` 之后，下一页不再只排几列就换页 (fix #199)
- 🐛 **装饰框随字号缩放**：夹注里或小字号下，`\带圈`、`\墨围`、`\八角墨围`、`\反白` 的圈和框随所在字号等比缩放，不再压到邻字 (fix #196)
- 🐛 **回退字体按需加载**：`\设置字体族` 的回退链只在某字号真需要时才载入，长书不再因字号多而 OOM，排版结果不变 (fix #197)
- 🐛 **对页边距**：从偶数页开始的 `正文` 块不再整块用错一侧边距；自由列宽模式最后一列不再越过左边距 (fix #198)
- 🐛 **vbook 页眉页码键**：`page-number-x`、`page-number-y`、`header-y` 经 `\pageSetup` 真正生效，并有中文键 `页码横移` ／ `页码纵移` ／ `页眉纵移`
- 🐛 **古籍模板 twoside**：古籍模板补上 `margin-inner` ／ `margin-outer` 默认值，`twoside = true` 不再变成零边距
- ✅ **测试**：浮动元素独占页的分页补上端到端单测 (fix #98)；新增 `test/margin_test.py` 边距度量断言
- 📖 **文档**：`\填充文本框` 扁字用法说明与示例 (fix #79)

[0.4.3] - 2026-10-28

- ✨ **线装书封面**：`\线装书封面[键值]{书名}`（`\ThreadBoundCover` / `\線裝書封面`）— 现代蓝色线装书封面（中华书局、广陵书社等影印古籍常见样式，参照 CY/T 247—2021 图 1、图 2）：藏青封面，左上角米色签条，签条内文武线双框；签条取最小高度，书名长了随字数增高，超出页面时自动缩字；`册次`、`卷次` 小字排在签条下部或 `小字位置=随后`；`订线` 开关可在预览时画出订线。示例见 `示例/线装书封面/` (#170)
- 🐛 **反白字色**：`\反白` / `\墨围反白` / `\八角墨围反白` 在四库全书彩色模板的正文里、以及任何模板的夹注里不再只画黑底不变白字 (#187)
- 🐛 **侧批换列缩进**：缩进段落里的长侧批换到下一列后与正文齐平，不再比正文高出 indent 格 (#188)
- 🐛 **嵌套段落缩进**：嵌套 `段落` 时内层缩进不再回退成外层缩进 (#168)
- 🐛 **纯浮动页**：只含悬浮文本框／印章的页不再被整页丢弃并截掉其后的正文页 (#98)
- 🐛 **悬浮文本框叠放次序**：同页多个悬浮文本框按注册先后绘制，不再随机
- 🐛 **句读与 vert 换形**：`vert` 特性把 `，` `。` 换成无 Unicode 码位的竖排字形（如 Windows 宋体）时，句读照常判型，不再退回正文下方 (PR #186)
- ♻️ **边距读取规则（不向后兼容）**：`twoside=true` 只读 `margin-inner` / `margin-outer`（奇偶页镜像），`twoside=false` 只读 `margin-left` / `margin-right`；被忽略的一组若被设置会发出警告；`twoside=false` 时 vbook 页眉页码固定在左侧。旧的 `twoside=false` + `margin-inner/outer` 写法不再生效，请改用 `margin-left/right` (#174)

[0.4.2] - 2026-09-30

- 🐛 **抬头后的夹注**：`\抬头` / `\平抬` / `\单抬` 之后的夹注不再跳到列首、落进天头；抬头的强制缩进不再泄漏到后面的夹注，夹注内部的 `\单抬` 照旧凸进天头 (#165)
- 🐛 **连续两个「正文」**：同一页上连续两个 `正文` 环境不再叠印（版心同时印出「一」和「二」），也不再报 `\prevdepth` 错误；`分栏` + `\换栏` 的双栏对照版式照旧共用一页 (#165)
- 🐛 **版心页码样式**：`\pageSetup{页码样式=digits}`（`digits` / `chinese` / `arabic` / `none`）对版心页码生效，与竖排页码走同一个格式化函数；`页码样式` 初始值改为「未设置」，`\pageSetup{纸宽=...}` 不再顺带关掉版心页码 (#161)
- 🐛 **`\关闭分页` 之后再 `\pageSetup`**：整本书纸宽不再被砍半、页数不再加倍，`\begin{document}` 之后调用也不再多出空白页；`\pageSetup` 不再无条件设 `\topskip`，正文里设的 `\topskip` 不被覆盖 (#166)

- 🐛 **横排中西间距**：空格窄于 1/4 em 的西文字体（如 Source Sans 3）纯西文行不再溢出版心 (#171)；`1/4 em`、`a/b` 里的 `/` `-` `·` `–` 视为西文半角标点，不再左右各加 1/4em (#160)
- 🐛 **句读颜色**：四库全书彩色模板下句读不再被画成正文色；`\句读设置{句读颜色=...}` 真正生效 (#157)
- 🐛 **颜色 key 别名与警告**：夹注、textflow、行、文本框、句读、批注、眉批、侧批的 `color`／`颜色` 与 `font-color`／`字体颜色` 两种写法互通，`\夹注设置{color=red}` 不再无效或把全文染色；写错的 key 会发出警告而不是被静默丢弃 (#163)
- ♻ **大部头提速**：版心竖排文字的缩放字体按 (字体, 字号) 缓存，《欽定儀象考成》（875 页）170s → 141s，输出 PDF 逐字节相同 (fix #167, #172)
- ✅ **测试**：新增颜色 key 自校验 `test/color_test.py`、clreq 中西间距断言补充

[0.4.1] - 2026-08-07

- 🐛 **批注/TextBox 西文断词修复**：TextBox 收集内容不再让 TEX 预断行（旧逻辑会把批注中的英文拆成 S-/tate 并产生列错位），断列全部交给网格引擎；grid 模式（ltc-guji 等固定格高）下 `\横置` 字幅改按 advance 计，整词可连排 (fix #50)
- 📦 **CTAN 打包修正**：按 CTAN 审核反馈调整——上传包改名 `luatex-cn-v<版本>.zip`、解包顶层唯一目录 `luatex-cn/`、`README.md` 置于包顶层并增加英文摘要；本地打包、CI 发布资产与 CTAN 上传为同一文件
- 📖 **Wiki PDF 瘦身**：示例预览图降采样至 300 dpi 并以 64 色 Indexed 嵌入，emoji 改单色矢量字体子集嵌入，zh 4.8→2.7 MB、en 4.3→1.8 MB

[0.4.0] - 2026-08-06

clreq 大版本：统一入口 `\usepackage{luatex-cn}`、完整横排 clreq 管道、竖排 clreq 达标（P1–P3）、横竖排共享规则内核、符合度声明矩阵与 119 条度量断言、免安装字体体系。

新增：

- ✨ **统一入口**：`\usepackage{luatex-cn}` 成为本包全部行为的统一入口——默认加载横排 clreq 管道（选项透传，如 `[style=taiwan]`）；`[vertical]` 或 banxin 类键加载竖排 bundle（0.3.x 语义不变）；ltc-* 文档类之下加载则自动走竖排分支 (#148)
- ✨ **横排 clreq 管道**（luatex-cn-hori）：中西间距 1/4em（可挤 1/8、拉 1/2，含三类例外）、标点宽度调整三预设（中国大陆式/台式/不调整）、引号体例转换、四级换行禁则与符号分离禁则、按 clreq 优先顺序的行内挤压/拉伸二次分配、行末标点半字宽、行尾点号悬挂、均排 (#127–#131)

- ✨ **横排行间标注**：`\专名` / `\书名号甲` / `\着重`（相邻标注两侧各缩 1/16em、标点上不加着重号）、`\拼音` / `\对照` 词对齐标音 (#129)
- ✨ **横排段落与页面级**：段末孤字避免、页首页尾孤行控制、凸排与缩排环境、段末行四种对齐、号数字号 `\字号`（初号…八号，横竖排共享） (#131)
- ✨ **竖排中西混排三配置齐备**（clreq P3）：汉字与西文边界 1/4em 间距接入求解器；`\横置` 西文串顺时针旋转 90° 排入竖列；`\中横排` 短串（两三位数字等）横排挤入一个字格（縦中横） (#146, #147, #149)
- ✨ **竖排行尾点号悬挂**（clreq 6.1.3）：`\标点设置{punct-hanging=true}`（中文别名 `标点悬挂` ）把列末点号挂到版口外，默认关
- ✨ **竖排着重号 clreq 达标**（5.3.1）：vbook 两类默认实心圆点 ● 且标点上不加；`\着重号设置` 新增 `char` / `scale` / `skip-punct` 键；ltc-guji 保持古籍圈点体例
- ✨ **clreq 共享规则内核** `tex/shared/`：标点全表、行内空间分配求解器、四级禁则、字面锚点，横竖排单一数据源、纯 Lua 可独立单测 (#125)
- ✨ **clreq 符合度声明矩阵** docs/CLREQ-CONFORMANCE.md：逐 clreq 条款列横竖排状态与证据，含有意偏离与不实现项的理由
- ✨ **clreq 断言测试**：解析 PDF 内容流做度量断言（横排逐字形坐标、直排按列基线距离），119 条并入 CI (#128)
- ✨ **免安装字体**：`\设置字体族` 接受注册表别名与文件路径；`download_fonts.py --user/--dest` 免管理员安装；字体清单 SHA-256 钉定 (#109, #122, #123)
- ✨ **几何自校验测试**：断言每列字形基线等距，不依赖基线图像即可发现整字偏移 (#113)
- ✨ **示例《论辩的魂灵》**（鲁迅）：横排 clreq 管道实战文档

变更：

- ♻️ **竖排接入共享行内调整求解器**（clreq P2）：挤压 7 级/拉伸 2 级 + 兜底均分，禁则「挤进/推出」按词典序比价；数字串/西文词/两字宽标点锁为刚性单元；ltc-guji 三套基线零变化 (#143)
- ♻️ **竖排标点宽度上下文相关**：单个标点占满字幅、连排才缩减；vbook 默认 `squeeze-mode=context`，ltc-guji 保持 legacy 无条件挤压
- ♻️ **标点字面分布度量驱动**：读字形 boundingbox 把墨迹锚到规范位置（中国大陆/台湾 × 横/竖全矩阵），字体自身字面设计不再影响落点
- ♻️ **`\标点设置` 与横排统一**：新增 `style` / `squeeze-mode` / `adjacent-punct` / `line-start-bracket` / `line-end-punct` 键

修复：

- 🐛 **直排夹注号字面归上/下半格**（clreq 6.3.2.1）：墨心相对位移锚定，四款字体落点收敛 (fix #105)

- 🐛 **破折号/省略号两字幅整体**：不拆列、内部字距归零，连排墨迹连成一线；ccmp 合字拆回逐个 em dash (fix #119)
- 🐛 **脚注标号几何修正**：标号不参与标点宽度调整、括号按真实墨迹缩放居中、中文数字支持千位
- 🐛 **格内垂直居中改用 em 框基线对齐**：修复「一」「丶」等墨迹不跨基线的字被整字偏移 (#113)
- 🐛 其他：装饰符号错位 (#118)、数字化空左小列 (#120)、换列守卫 (#108)、`os.name` 跨平台探测 (#100)、`run_all.lua` 失败漏报 (#121)

维护：

- ☑️ 测试与示例全面改用可确认授权的开源字体；回归测试零容差逐像素比较；CI 对每个 PR 跑 unit + regression (#101, #106–#117)

[0.3.8] - 2026-05-09

新增工尺谱排版命令族、声调圈点装饰、侧批底色支持，多项侧批与版面修复。

- ✨ **工尺谱排版**：新增 `\音[板/眼/高/低/昆低]{尺}` 主字命令、`\拖音` 延音/续拍单元、句读与宫调小字注释，支持完整传统乐谱版面 (#90)
- ✨ **发圈/圈发**：新增 `\发圈` / `\圈发` 命令，按平上去入在四角添加声调圈点装饰 (fix #92)
- ✨ **侧批底色**：`\侧批` 新增 `background-color` 参数（中文别名 `底色` / `背景色`），遮挡底层界行；特殊值 `page` / `inherit` / `继承` / `页面` 继承页面底色 (#93)
- 🐛 **侧批换行与抬头对齐**：修复侧批向前流转换行、抬头列对齐正文顶部、列底越界自动换列 (fix #91)
- 🐛 **侧批反向流转兜底**：page 0 / banxin 列处理修复；提取 `wrap_to_next_column` 助手消除重复换列逻辑 (review #97)
- 🐛 **悬浮文本框空页**：anchor 标记 `page_has_content`，防止整页只有 floating textbox 时被误判为空页 (fix #98)
- 🐛 **行填充=页 表格内边框**：起始页左侧内边框丢失修复 — 按页区分 `actual_band_cols`
- 🐛 **插图页整页抑制**：整页无真实文本时丝栏抑制扩展到两个半页

[0.3.7] - 2026-04-25

完善表格功能、旁注 API 重设计，多项夹注与 textbox 修复。

- ✨ **表格增强**：表格跨页、栏线自定义；新增 `行填充=半页` 模式
- ✨ **旁注 API 重设计**：`\旁注{主字}{右内容}{左内容?}`，支持 `\文本框` 内容、间距按主字大小自适应 (fix #94)
- ✨ **古籍模式空格选项**：新增 `ignore-spaces` 选项，默认忽略源码空格
- ✨ **版心页码开关**：增加显示开关
- 🐛 **夹注内 textbox**：八角框尺寸/对齐/竖排/sub-column x 偏移多项问题 (fix #96)
- 🐛 **parallel 模式表格跨页**：保留 band 结构与水平栏线、cell 紧密排列、cell 级列边框不继承到 overflow 页
- 🐛 **textbox 行高与空格**：内行高使用外网格高度，修复 body `\octagon{}` 溢出；textbox 后多余空格占用错误行数
- 🐛 **句读列换行定位**：标记在列换行时错误定位到旧列
- 🐛 **样式字体应用**：`\样式[字体=X]` 不带字号时也应用字体到 TEX 层
- 🐛 **fill 模式 padding**：`padding_top` 双重应用
- ♻️ **chapter 换行**：改用 `\夹注`，适配文本框渲染模式

[0.3.6] - 2026-03-20

新增插图页、牌记、注解开关等功能，版心全面 TextBox 化。

- ✨ **插图页**：新增 `\插图页` / `\IllustrationPage` 命令，支持半页粒度丝栏抑制，默认先换半页 (fix #29)
- ✨ **牌记**：新增 `\牌记` / `\colophon` 命令，不透明双层边框悬浮文本框 (fix #58)
- ✨ **注解开关**：侧批/眉批/批注支持 `sidenote=false` / `meipi=false` / `pizhu=false` 开关设置，可在 Setup 或单次调用中关闭 (fix #89)
- ✨ **换半页**：新增 `\换半页` / `\NewHalfPage` 命令
- ✨ **版心 TextBox 化**：版心上部/下部/中部全面通过 TextBox 管道排版，支持紧凑排列、夹注、fill-padding
- ✨ **TextBox 透明度**：新增 `transparent` / `透明` 布尔参数（默认 true），控制文本框背景透明度
- 🐛 **侧批标点转换**：修复 normal 模式下侧批竖版标点未正确转换的问题 (fix #89)
- 🐛 **表格修复**：修复自动宽度单元格后续列重叠、固定宽度空单元格不生效、纵对齐未扣除 padding 等问题

- 🐛 **插图页丝栏**：修复 `\插图页` 导致所有页面丝栏消失的样式栈污染问题
- 🐛 **表格跨页**：修复 parallel 模式跨页表格后续栏排版位置错误
- 🐛 **下划线位置**：修复 `\Underline` 在 Free Mode 下位置偏移
- ♻️ **parallel 默认**：表格分栏默认使用 parallel 模式（栏满换页而非换栏）
- ♻️ **padding 统一**：统一 `padding-top` / `padding-bottom` 参数，取代多套命名
- ♻️ **参数重命名**：版心参数从内容导向重命名为位置导向，`横位` / `纵位` → `横坐标` / `纵坐标`

[0.3.5] - 2026-03-14

新增组件级设置系统和多级标题支持。

- ✨ **Setting Stack**：新增 setting stack 模块，侧批/批注/眉批/TextBox 支持组件级独立标点模式配置 (fix #84)
- ✨ **全局设置命令**：新增 `\全局设置` / `\GlobalSettings` 命令，统一管理标点模式、标点标准等全局参数；侧批新增 xshift 横移参数 (fix #86)
- ✨ **多级标题**：新增 `\chapter` / `\section` / `\subsection` 多级标题命令，`\chapter` 支持可选参数 `\chapter[版心标题]{正文标题}` 区分版心与正文显示；新增 `\sectionSetup` 配置命令 (fix #64)
- ✨ **样式参数统一**：所有组件统一接受样式参数（.unknown 双层转发 settings → style）
- 🐛 **印章修复**：修复多个 `\印章` 命令互相覆盖只显示最后一个的问题 (fix #80)
- 🐛 **中文别名统一**：统一中文别名命名，列→行、SideNode→Sidenote (fix #83)
- 🐛 **换行空格**：修复正文环境中源码换行产生多余空格的问题 (fix #86)
- 🐛 **排版修复**：修复 auto_balance 分割时子列溢出、textflow 在表格 band 中列高约束、版心渲染字体匹配、banxin_ratio=1 横线穿过版心列、表格列边框=false 时边界竖线缺失等问题

[0.3.4] - 2026-03-12

新增表格、族谱排版（实验中）。

- ✨ **表格环境**：实现表格（Table）环境，支持单元格分组、多栏布局、栏高度/列数参数、细粒度边框控制
- ✨ **分栏功能**：实现分栏（Band）功能，支持竖排页面水平分栏及边框渲染
- ✨ **欧式族谱**：初步实现欧式族谱（欧阳修首创）排版，参见示例

- ✨ **夹注捷径**：新增 `\夹注捷径{开始符}{结束符}`，支持自定义括号自动替换为 `\夹注{}` (fix #76)
- 🐛 **抬头换行**：修复抬头后换行失效的问题（字数 ⩽ 上抬格数时 `\\` 不换列）(fix #78)
- 🐛 **夹注 key 转发**：修复 jiazhu key 系统转发未知 key 到 textflow 的问题 (fix #81)
- 🐛 **脚注标记**：修复脚注标记【一】【二】等字符居中对齐不一致的问题 (fix #77)
- 🐛 **表格修复**：修复表格单元格纵对齐、栏间线位置、列边框参数继承、多表格参数覆盖等多个问题
- 🐛 **排版修复**：修复换页/夹注/抬头多一行的 bug，修复 textflow pending 状态泄漏
- ♻️ **坐标系统重构**：pos.x/pos.y 使用绝对坐标，渲染阶段统一 RTL→LTR 转换

[0.3.3] - 2026-03-04

修复多个排版问题。

- 🐛 **专名号**：修复专名号在 natural mode 下每个字断开的问题 (fix #74)
- 🐛 **装饰符号偏移**：修复 Noto CJK 等字体装饰符号水平偏移问题，兼容不同字体 rawdata.descriptions key 格式 (fix #73)
- 🐛 **改字命令**：修复 `\改` 命令在夹注环境中多空两格及顿号位置偏上的问题 (fix #72)
- 🐛 **标点识别**：修复特定字体标点无法识别的问题 (fix #71)

[0.3.2] - 2026-03-02

完整实现现代竖排（仿中华书局排版）。

- ✨ **段落首行缩进**：实现段落检测和 vbook 首行缩进，indent 样式合并到 StyleEnv
- ✨ **运行页眉**：vbook 奇偶页交替显示章节名/书名，页码内侧对齐
- ✨ **双面镜像页边距**：添加 twoside 镜像页边距支持，vbook 默认启用
- ✨ **脚注增强**：实现脚注悬挂缩进、`\FootnoteParagraph` 无编号脚注段落命令、脚注序号对齐
- ✨ **墨围命令**：添加 `\墨围` 命令及分离的边框边距控制
- ✨ **列间距**：实现 Free Mode 列间距，支持自然模式接近满列的两端对齐
- ✨ **叠加对比工具**：添加 `overlay_compare.py` 红蓝叠加对比工具

- 🐛 **禁断规则**：修复 natural mode 禁断规则（kinsoku）在所有布局路径生效
- 🐛 **脚注修复**：修复脚注段落跨列后缩进丢失的问题
- 🐛 **vbook 修复**：修复 Lua 数字格式错误和台湾繁体命令别名
- 🐛 **批注颜色**：修复批注默认颜色中 expl3 空格吞并导致 RGB 解析错误
- 🐛 **字间距**：统一自然模式字间距策略，字间距改为 0.1em 比例间距
- 🐛 **列拉伸**：脚注序号和墨围不参与列拉伸

[0.3.1] - 2026-02-26

重点优化 guji-digital 文档类，用于影印本排版。

- ✨ **文档级样式系统**：新增 `\ConfigDocument` 命令，支持文档级样式预设和属性传递
- ✨ **数字化工具链**：提供 `semantic_to_digital.py` 和 `digital_to_semantic.py` 双向转换工具
- ✨ **Layout 导出功能**：通过 `ENABLE_EXPORT=1` 环境变量导出 layout JSON，支持自动化验证
- ✨ **PDF 页面映射**：简化 spread 页面语义，正确处理单页/双页映射关系
- 🐛 **缩进系统重构**：拆分 taitou（抬头）和 suojin（缩进）两类编码，修复属性传递 bug
- 🐛 **Textflow 布局**：修复 textflow 中 taitou indent 被误清除的问题
- 🐛 **装饰符号调整**：圈点/着重号字符从 `。` 改为 `○`，视觉效果更清晰
- 🐛 **换页处理**：修复 `\双列` 在 `\换页` 后的排版问题
- 🐛 **平抬功能**：修复 `\平抬` 在夹注中的失效问题
- 📦 **模板更新**：新增五凉全誌六德集模板，重构四库全书模板文件夹结构

[0.3.0] - 2026-02-18

项目已基本稳定，可进行整书排版。

- ✨ **使用体验优化**：整理参数命名规范，更新 Wiki 和 README 文档
- ✨ **模板系统**：新增模板文件夹，提供开箱即用的排版模板
- 🐛 **Paragraph 修复**：修复 `bottom-indent` 默认值设置 (fix #xx)
- 🐛 **版心修复**：修复版心章节对齐处理

注意：v1.0 之前，命令和参数可能会变化，请及时更新最新版本。

[0.2.9] - 2026-02-18

完整排版《欽定四庫全書簡明目錄冊一》，同时新增"数字化古籍"模式。

- ✨ **辅助命令**：新增 `\臣`、`\xshift` 等排版辅助命令
- ✨ **页面处理**：新增封面、校勘页、空白页等处理支持
- ✨ **数字化古籍模式**：新增 `ltc-guji-digital` 文档类，支持古籍 OCR 数字化录入与校对（实验性）
- ✨ **侧批增强**：`\SideNode` 支持位置参数和数值自动转换（`\SideNode[5pt]` = `\SideNode[0.5]` 自动转换为 em）
- ♻️ **参数改名**：
  - 重命名：`font-color` 的中文别名 `字色` → `字体颜色`（`\Style`、`\Paragraph`、`\SideNode` 等）
  - 重命名：`yoffset` → `yshift`（`\SideNode`，中文别名 `纵偏` → `纵移`）
  - 重命名：`distribute` → `fill`（`\TextBox`，中文别名 `均分` → `填充`）
  - 重命名：`font-name` → `font`（`\Content`，中文别名 `字体名` → `字体`）
  - 改进：Paragraph `first-indent` / `bottom-indent` 默认值改为空，自动跟随 `indent` 参数
- 🐛 **抬头修复**：修复大量抬头缩进、换行的问题
- 🐛 **句读修复**：修复句读模式在夹注中的问题
- 🐛 **版心修复**：修复单页版心处理
- 🐛 **背景颜色修复**：修复背景颜色继承与覆盖的处理逻辑

[0.2.8] - 2026-02-14

- ✨ **自由排版模式**：新增自动列宽和灵活列间距支持
- ✨ **样式命令**：新增 `\样式` 命令和 `样式环境`，支持行内样式覆盖（字号、字色、格高等）
- ✨ **封面功能**：新增古籍封面 `Cover` 环境及卷名/章节名命令
- ✨ **命令重命名**：文档类统一为 `ltc-guji`、`ltc-cn-vbook`、`ltc-tw-vbook`，命令参数全面支持英/简/繁三套名称
- 🐛 **Bug 修复**：更完整地支持现代繁体竖排，修复脚注和标题的排版问题
- ♻️ **代码重构**：大规模重构 layout-grid 模块，统一 grid/natural 引擎，重写单元测试

[0.2.7] - 2026-02-09

- ✨ **现代标点系统**：新增完整的现代标点支持（标点挤压、禁则处理、竖排引号替换），支持大陆/台湾两种标点风格
- ✨ **ltc-book 文档类**：新增 `ltc-book` 现代竖排书籍文档类，支持模板系统（默认/中华书局）
- ✨ **脚注系统**：新增 `\脚注` 命令，支持段末注和页下注两种模式，鹿角/带圈数字编号
- ✨ **自然排版模式**：新增 `layout-mode=natural`，支持非等距排列、标点半格、底端对齐
- ✨ **灵活列宽**：`\行` (Column) 支持 `width` 参数，实现不等宽列布局
- ✨ **书名页列宽**：书名页 (TitlePage) 支持 `column-widths` 参数
- 🐛 **Bug 修复**：修复列满自动翻页后 `\newpage` 导致内容丢失的问题 (fix #63)
- 🐛 **Bug 修复**：修复 `\Space` 命令在列首空格被跳过的问题 (fix #62)
- 🐛 **Bug 修复**：修复 `\挪抬` 命令尾部空格产生额外空白
- ♻️ **代码重构**：拆分 render-page.lua 为两个模块，拆分回归测试为三个套件

[0.2.6] - 2026-02-07

- ✨ **抬头支持**：新增 `\抬头[N]` 系统（单抬、双抬、三抬、平抬、相对抬头），支持自动外边框包裹
- ✨ **专名号/书名号**：改用 PDF 矢量绘图实现，解决连接性及字体适配问题 (fix #53, #56, #57)
- ✨ **夹注平衡控制**：新增 `auto-balance` 选项，支持不平衡排版
- 🐛 **字形旋转修复**：增加对省略号和破折号在缺少竖排字形时的自动旋转逻辑
- 🐛 **Bug 修复**：修复侧批跨页颜色泄漏问题 (fix #44)
- 🐛 **Bug 修复**：修复夹注换列逻辑问题

[0.2.5] - 2026-02-05

- ✨ **目录条目命令**：新增 `\条目` 命令，支持层级缩进，方便目录排版
- ✨ **列控制命令**：新增 `\列` 、`\末列` 命令，更好控制排版 (fix #30)
- 🐛 **Bug 修复**：修复专名号/书名号在行末最后一个字上未绘制的问题 (fix #54)
- 🐛 **Bug 修复**：修复部分字符（如"漢"）居中偏移问题，正文改用宽度居中 (fix #55)
- 🐛 **Bug 修复**：修复侧批跨页时颜色泄漏导致版心染色的问题 (fix #48)

[0.2.4] - 2026-02-03

- ✨ **下划线支持**：新增 `\Underline` / `\下划线` 命令，使用 `|` 字符实现 (fix #5)
- ✨ **书名号/专名号**：新增 `\书名号`（波浪线）、`\专名号`（直线）命令 (fix #16)
- 🐛 **Bug 修复**：修复侧批换列位置对齐问题 (fix #47)
- 🐛 **Bug 修复**：修复侧批换页走到第二列的问题 (fix #49)
- 🐛 **Bug 修复**：修复夹注对齐问题 (fix #51)
- 🐛 **Bug 修复**：修复 `\newgeometry` 不支持 paperwidth/paperheight 的问题 (fix #52)
- 🐛 **Bug 修复**：修复侧批跨页时颜色泄漏导致版心染色的问题

[0.2.3] - 2026-02-01

- ✨ **眉批功能**：新增 `\眉批` 命令，支持自动排版和手动指定坐标 (fix #46)
- ✨ **特殊文本框**：TextBox frame 功能，支持边框形状设置 (fix #25)
- ✨ **出版商显示**：新增出版商显示功能，在版心下方显示，字体可自定义 (fix #26)
- 🐛 **Bug 修复**：修复批注和文本框错误继承段落缩进的问题 (fix #37)
- 🐛 **Bug 修复**：修复眉批跨页时显示在错误页面的问题
- 🐛 **Bug 修复**：修复字体检测支持简繁体中文名称 (fix #35)
- ♻️ **代码重构**：重构边框样式系统，引入样式栈管理 border、outer_border 等属性

[0.2.2] - 2026-01-31

- 🐛 **Bug 修复**：修复前言章节在首页显示问题 (fix #45)
- 🐛 **Bug 修复**：修复夹注在列尾跳列问题 (fix #42)
- ✨ **句读增强**：支持更多中文标点符号 (fix #40)
- 🐛 **Bug 修复**：修复 TextBox 和批注继承段落缩进问题 (fix #37)
- ✨ **调试功能**：新增调试标尺，显示页面网格坐标 (fix #31)
- ✨ **字体探测**：支持简体/繁体中文字体名称识别 (fix #35)
- ♻️ **代码重构**：版心文本渲染逻辑整合
- ♻️ **代码重构**：标题/出版社命令移至 metadata.sty
- ✅ **模块清理**：清理版心、古籍、印章模块依赖

[0.2.1] - 2026-01-31

- ✨ **颜色保持**：侧批颜色跨页保持功能 (fix #38, #44)
- ✨ **样式栈架构**：实现 Phase 3 样式栈多属性支持
- ✨ **样式注册表**：Phase 2 多属性样式存储（颜色、字号、网格高度）
- 🐛 **Bug 修复**：修复侧批渲染顺序问题，现在正确显示在丝栏上方 (fix #38)
- 🐛 **Bug 修复**：句读模式现在正确应用于侧批内容 (fix #39)
- ♻️ **夹注迁移**：夹注功能迁移至样式栈架构
- ♻️ **代码重构**：统一颜色注册机制

[0.2.0] - 2026-01-31

- ✨ **插件架构**：重构为标准化插件 API，实现一致的组件行为
- ✨ **全局状态管理**：统一配置通过全局表管理所有模块
- ✨ **元数据系统**：集中化书籍元数据管理（标题、出版社、章节标记）
- ♻️ **模块命名空间**：清晰分离 core/、guji/、banxin/、util/、decorate/
- ♻️ **大规模重构**：插件系统重构，标准化模块接口

[0.1.5] - 2026-01-28

- ✨ **多字体支持**：新增多字体递补链功能及一套中文化字体设置命令 (fix #23)
- ✨ **功能完善**：实现 `\装饰` 和 `\改` 功能，支持垂直排版校勘标记 (fix #12)
- ✨ **文本框增强**：文本框支持垂直压缩（扁字）以适配网格位置 (fix #24)
- 🐛 **Bug 修复**：修复了维度解析 0pt 报错、句读位置偏移、长标题挤压等布局 Bug (fix #19, #20)
- ✅ **回归测试**：引入基于 PDF 渲染对比的回归测试自动化程序
- ♻️ **代码重构**：重构渲染核心逻辑并提升整体代码稳定性

[0.1.4] - 2026-01-27

- ✨ 添加了句读功能
- 🐛 修复了段落缩进、侧批换列的一些问题
- ♻️ 对代码进行了大量重构，更加易于理解和维护
- ✅ 添加更多 testfile

[0.1.3] - 2026-01-25

- 🐛 修复分页不准确、页码位置的问题
- ✨ 批注坐标更加准确，且会跳过版心位置防止被截断
- 📚 完整排版了《红楼梦甲戌本》第一回
- ♻️ 更加符合 Expl3 语法
- ✅ 添加了 l3build testfile 进行回归测试
- ✅ 建立了完善的 Lua 单元测试框架，支持 `l3build test` 命令
- 🐛 彻底修复了单元测试中的 Mock 逻辑与 API 不一致问题

[0.1.2] - 2026-01-22

- ♻️ src 重命名为 tex
- 🐛 彻底修复了 Windows 下 CRLF、BOM 字符问题
- 📦 采用 l3build，详见 build.lua
- 🐛 修复 package require 路径，确保 release 的版本解压即用

[0.1.1] - 2026-01-19

功能

- ✨ 增加了批注命令，底层仍是可悬浮文本框

修复

- 🐛 修复了 CRLF、BOM 在 Windows 下字符问题

文档

- 📖 更新了文档，中英双语

[0.1.0] - 2026-01-18

主要功能

- **原生几何垂直排版引擎**：基于 Lua 深度定制，专为古籍排版设计的网格布局系统

- **古籍核心元素支持**：完整支持版心、鱼尾、乌丝栏/朱丝栏等传统装帧元素
- **完善的注释系统**：提供灵活的夹注和侧批功能
- **开箱即用的模板系统**：预设《四库全书》、《红楼梦甲戌本》等多种传统书籍视觉风格
- **跨平台字体自动检测**：智能识别 Windows、macOS 和 Linux 系统字体
- **电子简策/筒子页支持**：支持传统"筒子页"逻辑及电子阅览适配
- **绝对定位印章系统**：支持在页面任意位置覆盖馆藏印/私印

项目支撑

- 自动化的 CTAN 打包与发布流程
- 详尽的中英文用户手册及多个经典古籍复现示例

开发文档

本文档详细介绍 `luatex-cn` 排版引擎的核心架构、源文件结构、测试体系以及开发流程。

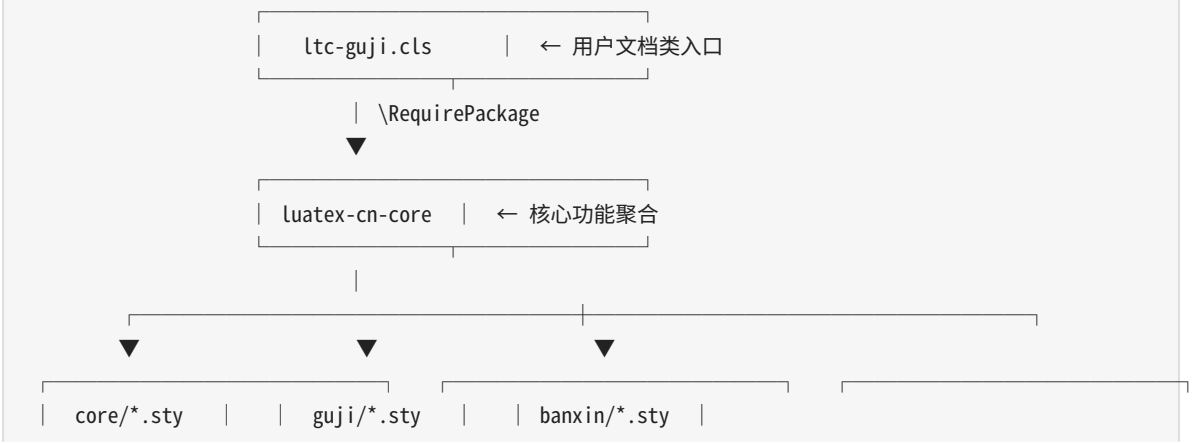
1. 源文件结构

1.1 目录总览

```
tex/
├── ltc-guji.cls                 # 古籍文档类（推荐入口）
├── ltc-ltc-guji.cls             # 古籍文档类（兼容别名）
├── ltc-cn-vbook.cls             # 中国大陆竖排书籍
├── ltc-tw-vbook.cls             # 台湾竖排书籍
├── ltc-guji-digital.cls         # 古籍数字化文档类
├── luatex-cn.sty             # 顶层宏包（兼容性入口）
├── luatex-cn-core.sty        # 核心功能聚合包
│
├── core/                     # 核心渲染引擎（最关键）
│   ├── luatex-cn-core-main.lua          # 主流程协调器
│   ├── luatex-cn-core-flatten-nodes.lua # 节点展平（Stage 1）
│   ├── luatex-cn-layout-grid.lua        # 网格布局计算（Stage 2）
│   ├── luatex-cn-core-render-page.lua   # 页面渲染（Stage 3）
│   ├── luatex-cn-core-content.sty/.lua  # 正文环境
│   ├── luatex-cn-core-page.sty/.lua     # 页面设置
│   ├── luatex-cn-core-page-split.lua    # 筒子页分割
│   ├── luatex-cn-core-column.sty/.lua   # 列控制
│   ├── luatex-cn-core-style.sty         # 样式命令（v0.2.8+）
│   ├── luatex-cn-core-sidenote.sty/.lua # 侧批系统
│   ├── luatex-cn-core-textbox.sty/.lua  # 文本框/悬浮框
│   ├── luatex-cn-core-textflow.sty/.lua # 夹注分列
│   ├── luatex-cn-core-paragraph.sty     # 段落缩进/抬头
│   ├── luatex-cn-core-metadata.sty/.lua # 元数据（书名、章节）
│   ├── luatex-cn-core-punct.sty/.lua    # 标点挤压与禁则
│   ├── luatex-cn-footnote.sty/.lua      # 脚注系统
│   ├── luatex-cn-constants.lua          # 常量定义
│   ├── luatex-cn-render-position.lua    # 位置计算
│   └── digital/                  # 数字化专用逻辑
│       └── luatex-cn-digital.sty        # 数字化命令实现
│
├── guji/                     # 古籍特有功能
│   ├── luatex-cn-guji.sty               # 古籍功能聚合
│   ├── luatex-cn-guji-judou.sty/.lua    # 句读
│   ├── luatex-cn-guji-jiazhu.sty        # 夹注（双行小注）
│   ├── luatex-cn-guji-pizhu.sty         # 批注（悬浮框）
│   ├── luatex-cn-guji-meipi.sty/.lua    # 眉批
│   ├── luatex-cn-guji-yinzhang.sty      # 印章
```

```
│   ├── luatex-cn-guji-danye.sty/.lua    # 单页模式
│   ├── luatex-cn-guji-cover.sty         # 封面（v0.2.8+）
│   ├── luatex-cn-guji-titlepage.sty     # 书名页
│   └── luatex-cn-guji-mulu.sty          # 目录条目
│
├── banxin/                   # 版心系统
│   ├── luatex-cn-banxin.sty             # 版心配置
│   ├── luatex-cn-banxin-main.lua        # 版心主逻辑
│   ├── luatex-cn-banxin-layout.lua      # 版心布局
│   ├── luatex-cn-banxin-render-banxin.lua  # 版心渲染
│   └── luatex-cn-banxin-render-yuwei.lua   # 鱼尾渲染
│
├── decorate/                 # 文本装饰
│   ├── luatex-cn-decorate.sty           # 装饰接口
│   ├── luatex-cn-decorate.lua           # 装饰实现
│   ├── luatex-cn-linemark.sty           # 线标（专名号/书名号）
│   └── luatex-cn-linemark.lua           # 线标实现
│
├── fonts/                    # 字体系统
│   ├── luatex-cn-font-autodetect.sty    # 字体自动检测
│   └── luatex-cn-font-autodetect.lua
│
├── debug/                    # 调试工具
│   ├── luatex-cn-debug.sty              # 调试接口
│   └── luatex-cn-debug.lua              # 调试实现
│
├── util/                     # 工具模块
│   ├── luatex-cn-utils.lua              # 通用工具
│   ├── luatex-cn-text-utils.lua         # 文本工具
│   ├── luatex-cn-drawing.lua            # 绘图工具
│   └── luatex-cn-style-registry.lua     # 样式注册表
│
└── configs/                  # 预设配置
    ├── luatex-cn-guji-default.cfg
    ├── luatex-cn-guji-SiKuQuanShu.cfg
    ├── luatex-cn-guji-SiKuQuanShu-colored.cfg
    ├── luatex-cn-guji-HongLouMengJiaXuBen.cfg
    ├── luatex-cn-vbook-default.cfg
    └── luatex-cn-vbook-ZhongHuaShuJu.cfg
```

1.2 组件关系图





1.3 核心模块依赖

```
luatex-cn-core-main.lua (主协调器)
    |
    ├─── luatex-cn-core-flatten-nodes.lua  (Stage 1: 展平)
    |
    ├─── luatex-cn-layout-grid.lua         (Stage 2: 布局)
    |       |
    |       ├─── luatex-cn-constants.lua
    |       └─── luatex-cn-style-registry.lua
    |
    └─── luatex-cn-core-render-page.lua    (Stage 3: 渲染)
            |
            ├─── luatex-cn-render-position.lua
            ├─── luatex-cn-utils.lua
            └─── luatex-cn-drawing.lua
```

---

2. 核心架构：三阶段渲染

`luatex-cn` 采用三阶段流水线架构：



Stage 1: Flatten (展平)

将 TEX 的嵌套节点链表转换为线性 Lua 表。

```
-- luatex-cn-core-flatten-nodes.lua
function flatten.process(head)
    local list = {}
    for node in node.traverse(head) do
        if node.id == GLYPH then
            table.insert(list, node)
```

```
        elseif node.id == HLIST then
            flatten_hlist(node, list)
        end
    end
    return list
end
```

Stage 2: Layout (布局)

计算每个字符的网格坐标，填充 `layout_map`。

```
-- luatex-cn-layout-grid.lua
layout_map[node] = {
    page = 1,        -- 页码
    col = 5,         -- 列号
    row = 3,         -- 行号（字符位置）
    x_sp = 123456,   -- x 坐标（sp 单位）
    y_sp = 789012,   -- y 坐标（sp 单位）
    color = "1 0 0", -- 颜色（可选）
    style_id = 42    -- 样式 ID（可选）
}
```

Stage 3: Render (渲染)

根据 `layout_map` 将节点重新组装并输出 PDF 指令。

```
-- luatex-cn-core-render-page.lua
function render.process(head, layout_map, params)
    for node, pos in pairs(layout_map) do
        local x, y = position.calculate(pos, params)
        insert_at_position(head, node, x, y)
    end
    return head
end
```

---

3. 组件详解

3.1 文档类 (`.cls`)

文件	用途	加载的包
`ltc-guji.cls`	古籍排版	`luatex-cn-core` + `guji/*.sty`
`ltc-cn-vbook.cls`	中国大陆竖排书籍	`luatex-cn-core` + 标点系统

文件	用途	加载的包
`ltc-tw-vbook.cls`	台湾竖排书籍	`luatex-cn-core` + 标点系统

ltc-guji.cls 加载顺序：

```
\LoadClass{article}
\RequirePackage{luatex-cn-core}       % 核心功能
\RequirePackage{guji/luatex-cn-guji}  % 古籍功能聚合
```

3.2 核心宏包（`.sty`）

文件	功能	关键命令
`luatex-cn-core-content.sty`	正文环境	`\begin{正文}`
`luatex-cn-core-page.sty`	页面设置	`\pageSetup{}`
`luatex-cn-core-column.sty`	列控制	`\Column{}` / `\行{}`
`luatex-cn-core-style.sty`	样式覆盖	`\Style{}` / `\样式{}`
`luatex-cn-core-sidenote.sty`	侧批	`\Sidenote{}`
`luatex-cn-core-textbox.sty`	文本框	`\TextBox{}`
`luatex-cn-core-paragraph.sty`	段落/抬头	`\begin{段落}` / `\抬头`
`luatex-cn-core-punct.sty`	标点系统	`\punctSetup{}`
`luatex-cn-footnote.sty`	脚注	`\Footnote{}` / `\脚注{}`

3.3 古籍功能 (guji/)

文件	功能	关键命令
`luatex-cn-guji-judou.sty`	句读	`\JudouPunctuationMode` / `\句读模式`
`luatex-cn-guji-jiazhu.sty`	夹注	`\JiaZhu{}` / `\夹注{}`
`luatex-cn-guji-pizhu.sty`	批注	`\PiZhu{}` / `\批注{}`
`luatex-cn-guji-meipi.sty`	眉批	`\MeiPi{}` / `\眉批{}`
`luatex-cn-guji-yinzhang.sty`	印章	`\YinZhang{}` / `\印章{}`
`luatex-cn-guji-cover.sty`	封面	`\begin{封面}`

文件	功能	关键命令
`luatex-cn-guji-titlepage.sty`	书名页	`\begin{书名页}`
`luatex-cn-guji-mulu.sty`	目录条目	`\条目`

---

4. 排版上下文系统

4.1 问题背景

在跨页渲染时（如侧批、夹注跨页），需要保持以下属性：

类别	属性示例	典型场景
字体	family, size, features	正文 → 夹注 → 正文
颜色	text_color, bg_color	正文 → 侧批 → 跨页
布局	indent, align, spacing	段落 → 文本框 → 段落
网格	grid_height	大字 → 小字 → 大字

4.2 样式注册表

`luatex-cn-style-registry.lua` 提供多属性样式存储：

```
local style_id = style_registry.register({
    color = {1, 0, 0},
    font_size = 19,
    grid_height = 20
})

D.set_attribute(node, ATTR_STYLE_REG_ID, style_id)
```

4.3 上下文继承

```
[全局上下文: 黑色, 20pt, indent=0]
  ↓ 进入段落
[段落上下文: 继承颜色和字体, indent=2]
    ↓ 进入夹注
  [夹注上下文: 继承 indent=2, 覆盖 color=red, size=14pt]
    ↓ 退出夹注
[段落上下文: 恢复到 黑色, 20pt, indent=2]  ← 自动恢复
```

5. 插件系统

v0.2.0 引入标准化插件 API：

```
plugin = {
    initialize = function(params, engine_ctx)
        return processed_params
    end,
    layout = function(list, layout_map, engine_ctx, context)
    end,
    render = function(head, layout_map, params, context, ...)
        return modified_head
    end
}
```

已实现的插件

插件	阶段	功能
`judou`	layout + render	句读符号
`sidenote`	layout + render	侧批
`textflow`	layout	夹注分列
`decorate`	render	文本装饰
`banxin`	render	版心绘制

6. 测试体系

luatex-cn 有两层测试：**单元测试**（Lua 逻辑）和**回归测试**（PDF 视觉对比）。

6.1 测试执行顺序

> **必须先跑单元测试，再跑回归测试**。 不允许跳过单元测试直接运行回归测试。

```
# 第一步：运行全部单元测试
texlua test/run_all.lua
```

```
# 第二步：运行回归测试
python3 test/regression_test.py check
```

6.2 单元测试

概述

- **运行环境**：`texlua`（LuaTEX 内置的 Lua 解释器）
- **测试框架**：`test/test_utils.lua`（自定义轻量框架）
- **测试文件**：24 个，约 3850 行代码，418 个测试用例
- **入口**：`test/run_all.lua`

目录结构

```
test/
├─── run_all.lua           # 运行全部单元测试
├─── test_utils.lua        # 测试框架（mock + assert）
└─── unit_test/
    ├─── util/             # 工具函数测试
    │    ├─── utils-test.lua
    │    ├─── text-utils-test.lua
    │    ├─── constants-test.lua
    │    └─── drawing-test.lua
    ├─── core/             # 核心引擎测试
    │    ├─── flatten-nodes-test.lua
    │    ├─── layout-grid-test.lua
    │    ├─── layout-grid-helpers-test.lua
    │    ├─── render-page-test.lua
    │    ├─── render-position-test.lua
    │    ├─── style-registry-test.lua
    │    ├─── core-column-test.lua
    │    ├─── core-metadata-test.lua
    │    ├─── core-page-split-test.lua
    │    └─── core-punct-test.lua
    ├─── guji/             # 古籍功能测试
    │    ├─── judou-test.lua
    │    ├─── meipi-test.lua
    │    └─── danye-test.lua
    ├─── decorate/         # 装饰元素测试
    │    ├─── decorate-test.lua
    │    └─── linemark-test.lua
    ├─── banxin/           # 版心测试
    │    ├─── luatex-cn-banxin-main-test.lua
    │    ├─── luatex-cn-banxin-render-banxin-test.lua
    │    └─── luatex-cn-banxin-render-yuwei-test.lua
    ├─── fonts/            # 字体检测测试
    │    └─── luatex-cn-font-autodetect-test.lua
    └─── debug/            # 调试模块测试
         └─── debug-test.lua
```

测试框架 API

`test/test_utils.lua` 提供以下核心 API：

```
local test_utils = require("test.test_utils")

-- 运行一个测试用例
test_utils.run_test("测试名称", function()
    -- 断言函数
    test_utils.assert_eq(actual, expected)        -- 相等
    test_utils.assert_neq(actual, expected)       -- 不等
    test_utils.assert_true(condition)             -- 为真
    test_utils.assert_false(condition)            -- 为假
    test_utils.assert_nil(value)                  -- 为 nil
    test_utils.assert_not_nil(value)              -- 不为 nil
    test_utils.assert_type(value, "string")       -- 类型检查
    test_utils.assert_contains(str, substring)    -- 字符串包含
    test_utils.assert_error(function() ... end)   -- 期望抛出错误
    test_utils.assert_near(actual, expected, tol) -- 数值近似
end)
```

Mock 基础设施

单元测试运行在 `texlua` 环境，没有完整的 TEX 引擎。`test_utils.lua` 提供了以下模块的 Mock：

Mock 模块	说明
`node` / `node.direct`	节点创建、遍历、属性操作
`tex`	`tex.box`、`tex.count`、属性分配
`font`	`font.getfont()`、字符信息
`texio`	日志输出
`luatexbase`	`new_attribute()` 等
`token`	Token 扫描器
`utf8`	UTF-8 编码工具

编写新测试

1. 在对应目录下创建 `xxx-test.lua` 文件
2. 引入测试框架和被测模块：

```
local test_utils = require("test.test_utils")
local my_module = require("path.to.module")

test_utils.run_test("功能描述", function()
    -- 准备
    local input = ...
    -- 执行
    local result = my_module.some_function(input)
    -- 断言
    test_utils.assert_eq(result, expected)
end)
```

1. 在 `test/run_all.lua` 中添加对应的 `require` 行
2. 运行 `texlua test/run_all.lua` 验证

常见陷阱

陷阱	说明
`new_attribute` ID 碰撞	Mock 中必须用递增计数器，返回不同 ID
0-indexed 页码	`group_nodes_by_page` 使用 0-indexed 页码
`node.direct` vs `node`	Mock 中两套 API 使用不同的内部 key
`_internal` 导出	模块通过 `_internal` 表导出内部函数用于白盒测试

6.3 回归测试

概述

通过编译 `.tex` 文件生成 PDF，将 PDF 渲染为图像后与基线图像进行像素级比较。

- **脚本**：`test/regression_test.py`
- **依赖**：Python 3.8+、ImageMagick（`compare` 命令）、Poppler（`pdftoppm`）

测试套件

```
test/regression_test/
├── basic/              # 基础功能测试
│   ├── tex/             # .tex 源文件
│   ├── baseline/        # 基准图像 (.png)
│   └── current/         # 当前输出 (运行时生成)
├── past_issue/         # 历史 Bug 回归测试
│   ├── tex/
│   ├── baseline/
│   └── current/
```

```
└──── example/            # 示例文件测试
    ├──── tex/
    ├──── baseline/
    └──── current/
```

常用命令

```
# 检查所有套件
python3 test/regression_test.py check

# 检查特定套件
python3 test/regression_test.py check --suite basic

# 检查单个文件
python3 test/regression_test.py check test/regression_test/basic/tex/shiji.tex

# 保存新基线（确认输出正确后）
python3 test/regression_test.py save test/regression_test/basic/tex/shiji.tex
```

添加回归测试

1. 创建 `.tex` 文件到对应套件的 `tex/` 目录
2. 运行 `python3 test/regression_test.py save <路径>` 生成基线
3. 验证基线图像是否正确
4. 提交 `.tex` 文件和基线图像

为 Bug 修复添加回归测试

修复 GitHub Issue 后，在 `past_issue/tex/` 下添加最小复现文件：

```
% issue_description_issue42.tex - 回归测试: 问题简述 (issue #42)
\documentclass{ltc-guji}
\关闭分页

\begin{document}
\begin{正文}
% 最小复现代码
\end{正文}
\end{document}
```

---

7. 参与开发

7.1 环境准备

1. **Fork 仓库**：点击 GitHub 右上角的 Fork
2. **Clone**：`git clone git@github.com:YOUR_NAME/luatex-cn.git`
3. **安装依赖**：
   - LuaTeX（TeX Live 2024+）
   - Python 3.8+（用于回归测试）
   - ImageMagick（`apt install imagemagick` 或 `brew install imagemagick`）
   - Poppler（`apt install poppler-utils` 或 `brew install poppler`）
4. **链接源码**：

```
ln -s $(pwd)/tex ~/texmf/tex/latex/luatex-cn
texhash
```

7.2 开发工作流

```
# 1. 创建功能分支
git checkout -b feat/my-feature dev

# 2. 修改代码

# 3. 运行单元测试
texlua test/run_all.lua

# 4. 如果修改了 Lua 逻辑，更新对应的单元测试

# 5. 运行回归测试
python3 test/regression_test.py check

# 6. 如果输出有预期变化，更新基线
python3 test/regression_test.py save <路径>

# 7. 提交
git add -p
git commit -m "feat: 简短描述"
```

7.3 代码规范

TeX 代码：必须使用 expl3

```
\cs_new_protected:Npn \luatexcn_func:n #1
  {
    \tl_set:Nn \l_tmpa_tl { #1 }
```

```
    \lua_now:e { module.process("\tl_use:N \l_tmpa_tl") }
  }
```

Lua 代码：复杂逻辑放独立文件

```
-- luatex-cn-module.lua
local M = {}

function M.process(text)
    -- 实现
end

package.loaded['module.luatex-cn-module'] = M
return M
```

命名规范：

- 所有用户命令提供三套名称：英文、简体中文、繁体中文
- 英文命令使用 PascalCase（如 `\Column`、`\TextBox`）
- Key-value 参数使用 kebab-case（如 `font-size`、`grid-height`）

7.4 提交规范

```
类型: 简短描述

详细说明（可选）

Fixes #issue_number
```

类型：`fix`, `feat`, `refactor`, `docs`, `test`, `chore`

---

8. 关键技术细节

8.1 竖排流向 (RTT)

```
\pardir RTT \textdir RTT
```

- 文本从上到下流动
- 行从右到左堆叠
- `\hsize` 定义列高度

8.2 属性管理

```
D.set_attribute(node, ATTR_JIAZHU, 1)
local val = D.get_attribute(node, ATTR_JIAZHU)
```

> ⚠ `\selectfont` 会清除所有属性，务必在设置字体后重新设置属性。

8.3 节点所有权

```
-- 交给 TeX（转移所有权）
tex.box[box_num] = node_list
-- 需要重用时必须复制
tex.box[box_num] = node.copy_list(original)
```

8.4 PDF 绘制顺序

PDF 遵循画家模型（后画覆盖先画）：

1. 背景色（最先）
2. 边框
3. 版心
4. 文字内容
5. 调试框架（最后）

---

9. 开发经验总结

详细经验请参阅项目中的 `ai_must_read/LEARNING.md`。

常见陷阱

问题	原因	解决方案
颜色不生效	使用颜色名称	使用归一化 RGB：`0 0 0 rg`
跨页颜色丢失	颜色堆栈断开	通过 layout_map 传递颜色
`\directlua` 失效	`--` 注释被线性化	使用 `----` 或独立 lua 文件
expl3 空格丢失	`ExplSyntaxOn` 忽略空格	使用 `~` 代替空格

问题	原因	解决方案
类型错误	Lua 弱类型	比较前检查 `type(var) == "number"`
嵌套 `\NewEnviron` 死循环	内层重定义 `\BODY`	先保存到独立 tl 变量

☞ 返回：功能详解 | 调试模式 | 首页

发布流程

`luatex-cn` 的发布遵循自动化流程，确保符合 CTAN (Comprehensive TEX Archive Network) 的要求。

开发工作流

测试

```
# 1. 运行单元测试（必须先通过）
texlua test/run_all.lua

# 2. 运行回归测试
python3 test/regression_test.py check

# 测试单个回归文件
python3 test/regression_test.py check test/regression_test/basic/tex/shiji.tex

# 更新基线（确认改动正确后）
python3 test/regression_test.py save test/regression_test/basic/tex/shiji.tex
```

本地安装

源码目录 `tex/` 可通过 symlink 链接到本地 texmf 目录进行开发测试：

```
# macOS/Linux
ln -s $(pwd)/tex ~/texmf/tex/latex/luatex-cn
texhash
```

自动化发布 (CI/CD)

通过 GitHub Actions 实现自动化打包：

1. **源码同步**：主分支（`main`）用于稳定代码，`dev` 分支用于日常研发。
2. **打包过程**：
    - 提取 `tex/`、`文档/` 和 `示例/` 文件夹。
    - 根据 `scripts/file_name_translation.json` 将文件夹和文件名转换为 CTAN 推荐的英文格式。

    - 自动生成符合规范的目录结构（doc, example, tex）。
3. **发布分支**：打包后的内容会被推送到 `ctan` 分支并发布到 GitHub Releases。
4. **Gitee 镜像**：通过 GitHub Actions 自动同步主仓库和 Wiki 到 Gitee。

版本管理

- 版本号记录在根目录的 `VERSION` 文件中。
- 所有重大变更记录在 更新日志。
- 遵循语义化版本（SemVer）规范。

发布检查清单

1. 确保 `dev` 分支上所有单元测试和回归测试通过
2. 更新 `VERSION` 文件
3. 更新 更新日志
4. 将 `dev` 合并到 `main`
5. 创建 Git tag：`git tag v0.x.x`
6. Push tag 触发 CI 自动打包发布

☞ 下一步：查看 开发文档。