

# Nestoc 文档插件

xieby1

一个模块+嵌套的 Typst 文档插件。

# 目录

1. Nestoc 简介 .....	3
2. Nestoc 能力展示 .....	3
2.1. 父模块的题目 .....	3
2.1.1. 父模块的一级标题 .....	3
2.1.2. 子模块的题目 .....	3
2.1.2.1. 子模块的一级标题 .....	4
2.1.2.2. 孙模块的题目 .....	4
2.1.2.2.1. 孙模块的一级标题 .....	4
3. 原理和使用 .....	4
3.1. nestoc(nestoc_fn, heading_offset:0, template:default_template) => body .....	4
3.2. nestoc_fn(heading_offset:0) => nestoc_obj .....	5
3.3. template(title:"", author:"", abstract:[], body) => _body .....	5
3.4. Nestoc 文档的代码框架 .....	5
3.5. 例子 .....	6
4. 总测试 .....	6
4.1. 空格测试 .....	6
4.2. 智能引用 .....	6
图索引 .....	7
表格索引 .....	7
代码块索引 .....	7

# 1. Nestoc 简介

Nestoc 的名称来源于 Nest（嵌套）与 Doc（文档）的组合，亦可理解为 Nest + ToC（Table of Contents 目录）。Nestoc 旨在构建一个支持文档嵌套（或称“模块化”）的 Typst 插件。

设想一个包含多个章节的大型文档项目，需要由多位协作者分模块完成各章节内容，最终将这些模块嵌套整合为统一的完整文档。

- **模块** 每个章节均为独立、可编辑、可编译、可阅读的自包含文档。
- **嵌套** 所有章节组合时，标题、编号等元素将自动调整，形成协调一致的最终文档。

## 2. Nestoc 能力展示

用 Nestoc 的文档 `doc/` 作为例子，展示 Nestoc 的模块和嵌套的能力。

Nestoc 的 `doc/` 目录包含了模块化的文档。其模块层次如下

- 总文档: `./main.typ => ./parent/main.pdf`
- 父模块: `./parent/main.typ => ./parent/main.pdf`
- 子模块: `./parent/child/main.typ => ./parent/child/main.pdf`
- 孙模块: `./parent/child/grandchild/main.typ => ./parent/child/grandchild/main.pdf`

每个模块均可独立地编辑、编译生成 pdf。当然每个模块也可以被嵌套到其他模块中，比如 `grandchild` 被嵌套到了 `child` 里；`child` 被嵌套到了 `parent` 里；`parent` 被嵌套到了这个总文档的下面。你可以仔细观察这些 pdf 文档，然后理解它们间的嵌套关系。

### 2.1. 父模块的题目

xieby1

父模块被嵌套到了总文档中，以二级标题为基础。即各级标题偏移+2。其各级标题映射关系如下：

父模块	总文档
题目	二级标题
一级标题	三级标题
..	..

表 1 父模块题目/标题映射关系

#### 2.1.1. 父模块的一级标题

父模块的一级标题下方的内容

#### 2.1.2. 子模块的题目

xieby1

子模块被嵌入到父模块的题目之下。因此相对父模块，子模块的标题等级都会+1，这个现象能在父模块文档中观察到。当然，子模块间接地随着父模块被嵌套到了总文档中。相对总文档，子模块的标题登记都会+3 于是有映射关系如下：

子模块	父模块	总文档
题目	一级标题	三级标题
一级标题	二级标题	四级标题
..	..	..

表 2 子模块题目/标题映射关系

### 2.1.2.1. 子模块的一级标题

子模块的一级标题下方的内容

### 2.1.2.2. 孙模块的题目

xieby3

孙模块被嵌套到子模块中，当然间接地随着子模块被嵌套到了父模块和总文档中。其各级标题的映射关系如下：

孙模块	子模块	父模块	总文档
题目	一级标题	二级标题	四级标题
一级标题	二级标题	三级标题	五级标题
..	..	..	..

表 3 孙模块题目/标题映射关系

### 2.1.2.2.1. 孙模块的一级标题

孙模块的一级标题下方的内容

## 3. 原理和使用

Nestoc 的所有功能围绕着函数 `nestoc(nestoc_fn, heading_offset:0) => body` 函数展开

### 3.1. `nestoc(nestoc_fn, heading_offset:0, template:default_template) => body`

- 参数 `nestoc_fn` 为函数，其类型为 `nestoc_fn(heading_offset:0) => nestoc_obj` :
- 参数 `heading_offset` 为 int，表示给 `nestoc_fn` 的题目和标题施加 `heading_offset` 的偏移。

例如：

- `heading_offset:0` 表示不偏移
- `heading_offset:1` 表示 `nestoc_fn` 题目=>一级标题，`nestoc_fn` 的一级标题=>二级标题，...
- `heading_offset:2` 表示 `nestoc_fn` 题目=>二级标题，`nestoc_fn` 的一级标题=>三级标题，...
- 参数 `template:default_template` 模板，其类型为：  
`template(title:"", author:"", abstract:[], body) => _body`。目前默认模板为修改过的 `ilm` 模板。
- 返回值 为文档内容

### 3.2. `nestoc_fn(heading_offset:0) => nestoc_obj`

- 参数 `heading_offset` 同 `nestoc` 的 `heading_offset`
- 返回值 `nestoc_obj` 为字典类型，有如下成员变量：
  - `title` 字符串类型，文档的题目
  - `author` 字符串类型，作者
  - `abstract` 文档类型，摘要
  - `body` 文档内容

### 3.3. `template(title:"", author:"", abstract:[], body) => _body`

- 参数 `title` 标题字符串
- 参数 `author` 作者字符串
- 参数 `abstract` 摘要内容
- 参数 `body` 文档内容
- 返回值 应用了模板之后的文档内容

## 3.4. Nestoc 文档的代码框架

每个 Nestoc 文档：

- 必须提供函数 `nestoc_fn` 供其他 Nestoc 文档调用
- 可选调用 `nestoc(nestoc_fn)` 来渲染当前文档

因此一个 Nestoc 文档的代码框架如下：

```
1 //导入nestoc
2 #import "@local/nestoc:0.1.0"
3 //定义nestoc_fn函数
4 #let nestoc_fn(heading_offset: 0) = (
5 title: "题目", author: "作者", abstract: [摘要], body: [
6 =一级标题
7
8 正文...
9 ])
10 //调用nestoc函数，将nestoc_fn的文档内容插入于此。
11 #nestoc.nestoc(nestoc_fn)
```

代码 1 Nestoc 文档的代码框架

若用图表示上面的 Nestoc 文档代码框架，则如下：

`nestoc_fn` — `nestoc(..)+编译` → 文档.pdf

图 1 Nestoc 文档代码框架的图示

- 第 1-7 行定义函数 `nestoc_fn(heading_offset: 0) => nestoc_obj`。返回的 `nestoc_obj` 的字典包含了 `title`, `author`, `abstract` 以及文档主体 `body`。
- 第 8-9 行将上面定义的 `nestoc_fn` 传给 `nestoc` 函数。`nestoc` 内部会进行标题级别的协调，然后调用 `nestoc_fn` 函数，最后返回处理好的文档主体。

你可能会好奇若不调用 `nestoc(..)` 函数会怎样？则该 typst 文件仅是一个无法编译出 pdf 的模块。这个模块仅能用于嵌入其他模块。

### 3.5. 例子

本文档提供了一个具体的 Nestoc 例子。本文档一共包含了 4 个模块。

- 每个模块均定义了自己的 `nestoc_fn` 函数。
- 每个模块均调用了 `nestoc(自己的nestoc_fn)`。这让该模块可以编译成独立的 pdf 文档。
- 文档的嵌套通过调用 `nestoc(别的模块的nestoc_fn, heading_offset: xx)` 来实现。

所有 `nestoc` 和 `nestoc_fn` 的关系如下图所示：

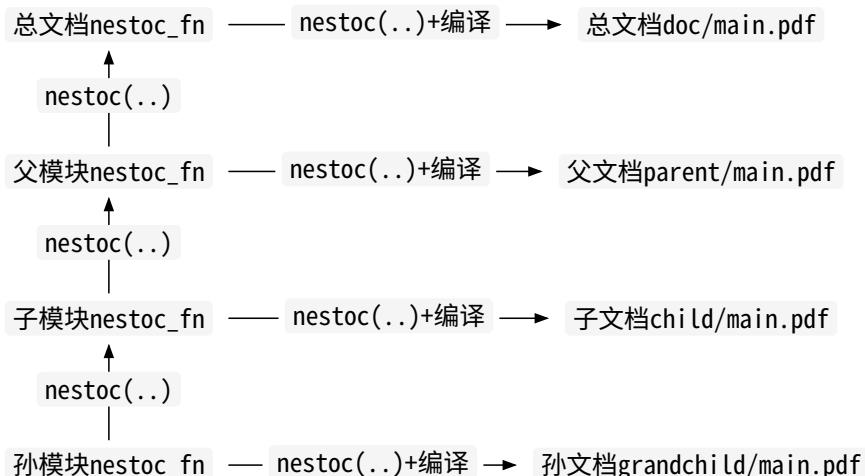


图 2 测试一下图

## 4. 总测试

### 4.1. 空格测试

多行不会出现空格

中间也不要空格

交替空格也不要

强制 空 格 可以使用~

### 4.2. 智能引用

"不要搞智能引用"

“我自己知道切换中文引号”

“I know what I am doing”

"So don't be over smart"

## 图索引

图 1 Nestoc 文档代码框架的图示 .....	5
图 2 测试一下图 .....	6

## 表格索引

表 1 父模块题目/标题映射关系 .....	3
表 2 子模块题目/标题映射关系 .....	4
表 3 孙模块题目/标题映射关系 .....	4

## 代码块索引

代码 1 Nestoc 文档的代码框架 .....	5
---------------------------	---