

---

# 异步图书作者写作指南

---

## ↓ 0. 导言

---

### 0.1 为什么要写书

技术图书的作者，往往在所在的领域或行业深耕数年、十数年甚至数十年，有深厚的学科知识，也有了一些心得体会和独到见解，希望以图书的形式分享出来，影响和帮助到更多的人。正如这些作者所说，写书不是为了成就作者，而是为了成就每一位读者！

这些作者有着强烈的求知欲和很高的学习觉悟，他们渴望学习，以阅读引领自我的求知，靠写作萃取专精的知识（Yearning For Learning, Leading by Reading, Distilling by Writing）。

这些作者有熟练的写作技巧和知识传授技巧。他们认为，书写是一种锻炼思维的方式，因为书写是对思维的备忘，书写是对思维的缓存；书写也是与自己的对话、与别人交流的高级手段。

这些作者对成功有自己理解和看法，所以，他们认为写作是最好的个人投资。

基于这些理念，我们可以看到写作和出版技术图书，能给你带来的一些实际的效果。

- 传播知识，同时也完善自身知识体系；
- 扩大朋友圈，提高交友质量；
- 打造个人品牌，技术图书作者是很好的身份标签；
- 丰富职业经历，带来更多职业机会和商业合作，甚至能够为个人创业或企业品牌背书；
- 有一份稿酬作为回报；
- .....

### 0.2 了解出版流程

在立下雄心壮志写一本技术图书之前，我们不妨先来了解一下图书出版流程。

一般来说，一本技术图书的出版要经过以下几个阶段：



绿色表示需要出版社编辑和作者协作完成的工作。

蓝色表示需要出版社编辑独立完成的工作。

红色表示需要作者独立完成的工作。

本指南主要涵盖了第 1-2 个阶段，也就是从选题计划到写作交稿阶段。这两个阶段是决定一本图书品质的关键阶段。

第 3 到第 5 个阶段会在出版社内部进行，所以尽管（作者）会参与一些工作（如第一次校对样的核对、出版后的图书营销推广），但作者这些阶段所承担的工作，相对要少一些。

## 1. 确定选题

出版社对于准备出版的图书的设想和构思，称为选题。

选题是图书出版流程的起点。

### 1.1 选题计划

要写一本书并出版，你首先需要和编辑讨论并制定一个选题计划。

一般来说，选题计划至少要包括如下的内容：

- 写作的主题
- 作者的介绍

- 读者对象
- 写作的目录（大纲）
- 所写作图书的特色和卖点
- 同类书（竞品）分析
- 样章
- 其他有助于判断和确立选题的信息和内容

## 1.2 研究你的读者

作者写作图书是为了读者。图书是否成功，读者的评判是最重要的标准，正所谓“金奖银奖，不如读者的夸奖”。因此，无论出版社还是作者，都应该树立“读者至上”的意识。也不用把读者当上帝，先做到把读者当兄弟就很好了。

那么，在开始写作之前，在选题阶段，最重要的事情就是确定读者对象，研究你的读者。

### 1.2.1 目标读者是谁

正如前面所述，目标读者是选题计划中最重要的问题。

要弄清楚目标读者是谁，我们可以先拆分出几个小问题去思考：

- 什么人会读这本书？
- 本书主要读者是做什么工作的（或是什么职位，几年经验）？
- 他们为什么读这本书，会怎样阅读和使用这本书？
- 阅读这本书需要具备什么基础知识？

研究目标读者的一种具体方法是读者画像。

要做好读者画像，至少要搞清楚下面几个问题

问题	说明
目标读者是谁	1.目标读者的一般性说明
	2.假设他们具备什么知识
	3.目标读者的关键特点/独特挑战是什么？
对目标读者来说，什么最重要	1.目标读者要处理的关键问题是什么？
	2.他们需要知道哪些功能？

### 1.2.2 读者的知识水平

技术图书最大的读者群体是从业者，俗称程序员。

程序员每天都很忙碌地工作，他们并不总是有时间来学新东西。他们的学习呈现碎片化，需要从工作中插空学习。这是一种非正式的学习，工作中需要什么，他们学什么。

在策划图书之前，我们先要了解读者的“知识水平”。

知识水平	起点：读本书之前	终点：读完本书后
小白	没有相关知识或没接触过	工作知识：对该技术基本熟知
工作知识	了解基本功能性知识/熟练	高级知识：专门水平，能够用该技术来完成自己的工作
高级知识	专业且熟练/对技术有了丰富的实用知识	专家知识：权威用户，提炼和巩固，关注专门方向和小众领域

### 1.2.3 读者的学习特点

了解读者的学习特点，可以帮助你理清思路。

如果你能够更细致地了解读者的学习特点和学习过程，那么你可以围绕读者的学习特点来进行选题策划和写作，做到有的放矢，达到超出预期的效果。

如果你的读者对象是程序员，你要知道——程序员很忙，所以他们对信息的要求是：

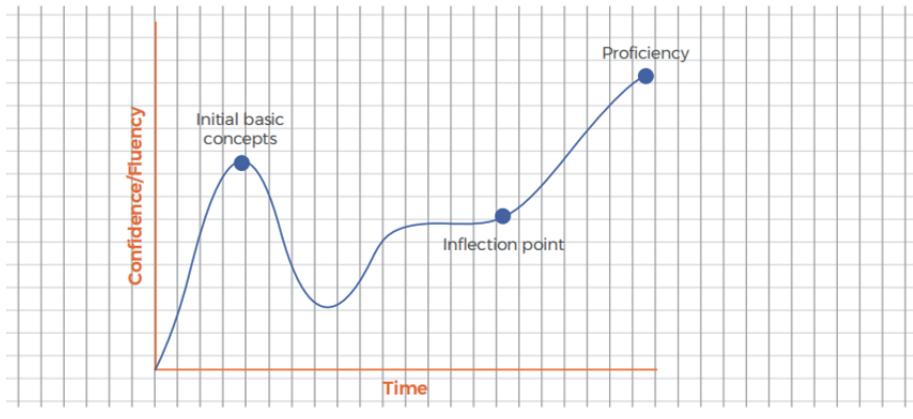
- 信息要以有逻辑的、线性的方式展示。
- 要易于获取，要能快速有效地找到他们需要的内容。
- 碎片化学习体验。内容要独立于语境，弱关联。读者不需要阅读上下文就能找到所需的答案。
- 能够投入实际使用。

### 1.1.4 了解读者的学习过程

在分析读者对象及其学习过程的时候，我们要考虑两个方面的需求：

- (1) 图书的目录结构和读者的预期、知识水平是否匹配？
- (2) 是否符合读者的学习体验。

人们学习知识和技能的过程是存在一定的规律性的。一般来讲，人们学习一个知识或掌握项技能，会经过如下的 4 个阶段。



(1) 入门期。刚开始学习一项新技术（技能），由于最先学习的都是最初的基本概念，有一个阶段是随着时间的增加，感觉学习有趣而充实。

(2) 低谷期。稍后，开始意识到一项新技术（技能）实际上很难。这说明那些直观的和基础的知识都已经学完了，开始感到灰心丧气了，对学习过程感到沮丧。

(3) 胶着期。稍后，会走出低谷期，这时候，大脑的神经开始适应所学的东西。大脑将这些知识带入到深层记忆，而不只是表层记忆。最终，学习过程逐渐达到转折点。

(4) 熟练期。经过转折点，学习开始变得容易，开始对所学技能变得越来越熟练。

### 1.3 研究竞品

这里说的竞品是一个比较广义的概念，不仅包括同形态的产品，也包括其他形态的、替代性或互补性的知识产品。从这个角度来看，研究竞品就是要搞清楚如下的一些问题：

- 研究同类的畅销图书：仔细查看目录和它们所涵盖的主题，阅读读者评论来了解读者，翻阅内容看看作者是如何处理各个主题的。目的：了解读者喜欢这本书的哪些方面，他们对此是如何评价的。
- 除了本书，读者还能够找到什么样的资料和信息（包括其他的书）？
- 如果别人想要学这门技术，你会推荐什么资料（视频、课程、融合出版机会）？
- 我的这本书和这些竞品比，有什么差异或优势？
- 我的图书的独特性/独特价值主张是什么？

## ↙ 2. 确定目录结构

---

### 2.1 目录的重要性

目录的作用——读者买书的时候，通过目录来判断这本书是否适合自己。目录对于图书销售至关重要。

把目录当做一张“藏宝图”，它引导读者按照特定顺序学习。

确定目录的过程大概可以参考下面的几个步骤：

1. 构思。研究一下其他的作者是如何安排主题的？读者按照什么顺序来学习？
2. 搭建一个目录草案。
3. 和其他的专家或编辑，讨论目录草案。
4. 确定目录。

### 2.2 搭建目录

#### 2.3.1 一般原则

人们认知事物的时候，常常自觉地应用一些结构性原则，如分类、逻辑排序、对应等等。

同样，组织搭建目录的时候，要遵从一些符合认知规律的原则：

1. 从易到难。
2. 从常用的到较少用到的知识。最有用的知识放在图书的最前面。
3. 从先到后。比如，按照你做项目的顺序来，部署应该是最后一步。

搭建目录的时候还需要考虑读者的用户体验。

读者如何与图书交互？要注意整体概念和局部知识的关系，符合组件化学习的特点；图书内容的安排，要让读者容易找到相应内容，容易理解知识点之间的关联性。

目录是否让读者更容易阅读？读者更关注自己的工作内容，而不是图书本身。他们买书是为了解决问题。

#### 2.3.2 好目录的特点

要精心安排目录结构，仔细确定各章标题和子标题。好的目录标题，一般有如下的一些特点：

- 易于阅读：目录标题长度保持在 10 个字内，甚至更少
- 易于理解：尽量避免使用术语，能够打比方就更好了
- 抓住要点：标题不需要太多创意，抓住要点，要直接而具体
- 语法正确：错别字、不通顺，读者直接就不买了
- 实用性：使用动词，如构建、创建、探索等
- 要有逻辑顺序：从易到难、从先到后等
- 标题尽量用数字，更吸引人。

### 2.3.3 举个例子

假设要编写一本 Python 编程的图书

导言

第一部分 基本概念和技巧

第二部分 其他概念和技巧

第三部分 面向对象编程

第四部分 数据库和 GUI 编程

附录

导言

第一部分 基本概念和技巧

    第 1 章 编写第一个程序

    第 2 章 用控制语句编写代码

    第 3 章 定义并使用函数和模块

    第 4 章 测试和调试程序

    第 5 章 使用列表和元组

    第 6 章 使用文件 I/O

    第 7 章 异常处理

第二部分 其他概念和技巧

第三部分 面向对象编程

第四部分 数据库和 GUI 编程

附录

#### 2.3.4 检查和回顾

最后，通过如下表格来检查目录，确保目录符合各项标准。

问题	说明
全面性	有漏掉的内容吗？
	目录结构和读者的熟练程度一致吗？
指引性	目录能帮助读者轻易找到内容吗？
	目录体现了逻辑性吗？
	容易学习吗——读者能够很容易地学并学完？
关键词	重要：章标题包含了用于搜索优化的关键词吗？
标题	切中要点：具体而不要有太多创意/要有趣
	语法正确吗（避免错别字和错误标点）？
	容易理解吗（标题要和读者的水平对应）

## 3. 写作的准备

### 3.1 选择写作工具

工欲善其事必先利其器。磨刀不误砍柴工。

在开始高效写作之前，不妨先结合自己的需求和写作习惯，选一款高效的写作工具。

工具	优点	缺点	推荐指数
Markdown	格式简单、版本控制方便、导出其他格式快捷、代码和公式支持性强	使用门槛高，无法批注	不推荐
桌面 Office	门槛低、格式丰富、可批注	对代码不友好，无法多人写作	保留性推荐
在线文档服务（Google 石墨等）	支持协作、方便快捷、适合碎片化写作	版式功能不够强大、对代码不友好	保留性推荐（可用于选题计划、目录搭建、写草稿等）

工具	优点	缺点	推荐指数
WPS	桌面和在线可以很好切换、可批注、支持协作、轻量级、基本功能免费	代码不友好、高级功能付费、有时候版本控制出错	强烈推荐
LaTeX	格式简单、对代码和公式支持性强、排版功能强大	使用门槛高、导出不方便	保留性推荐（用于人工智能、数学类等专业性较强图书写作）

## 3.2 写作的步骤

写作图书的过程虽然很漫长，但基本上也可以分为如下几个步骤。

当然，也可以按照如下的步骤，依次完成一本技术图书的每一个部分，实现模块化的写作。

- 确定基本的写作风格
- 写草稿
- 初审草稿
- 修改草稿

## 3.3 技术图书的风格

理想的技术图书应该：技术图书要切实解决读者的问题，不要太过理论化，毕竟不是学术图书。

技术图书要立足于把知识付诸实际应用。

总而言之，技术图书要坚持自己特有的风格。

- 介绍关于一个主题的必要细节。
- 对概念和理论，提供直观和信息充分的说明。
- 向读者展示如何使用这些概念和理论，以及如何避免常见的陷阱。
- 尽可能快地进入到实用的核心内容，并且提供实际的示例来教授理论。
- 模块化。技术图书的每一章应该是一个独立的碎片化学习单位。把每章都做一个单独的课程来规划，应该和前面的章相关，但更应该独立，便于碎片化学习。

### 3.3.1 语言

不同的作者写作，有不同的语言风格。

正如前面所述，技术图书的读者大多为程序员，他们有自己的学习习惯，因而也有自己所喜欢或习惯的语言风格。

写作技术图书的时候，所使用的语言应该尽量符合以下的要求：

- 使用日常语言，不要使用太官方、正式、文绉绉的语言。
- 避免用长句。
- 避免太啰嗦。
- 避免太形式化和教条。
- 尽量用简单的句子。
- 用把字句，少用被字句。

### 3.3.2 代码

实用的代码示例很重要，它能够讲清楚概念，并且帮助把所学的技术付诸使用。

- 代码最好不要以图片的格式给出，以免后期排版还需要重新录入为文字格式。
- 确保图书文本中代码的长度不要跨页。
- 如果要分享大段代码，考虑将其分成几段，每一段用来说明一个关键点。
- 正确保留缩进。每次缩进一般以 4 个空格为一个缩进单位，最好是使用空格来控制对齐，不建议使用制表符，因为制表符不同编辑器缩进不同，虽然能设置，但是后期排版会麻烦一些。
- 彩色印刷的图书，代码显示颜色和编程环境中保持一致。
- 示例代码要分章存放，放到和对应的图书章号一致的文件夹下面。完整的代码包，将会放到社区网站图书页面的“配套资源”下，供读者下载。

### 3.3.3 图片

一图胜千言。

要用好图片，也有一些规范和规律需要遵循。

**图片要符合规范**

格式：jpg/png/bmp/tif

**分辨率：**分辨率最好为 600 像素/英寸，最低不低于 300 像素/英寸

**图片文件大小：**2MB 以内

### 一般性的图片要求

- 图要有标题，图题尽量简单直接。
- 图应该按章编号。
- 先文后图。正文中先提到图号，然后在出现图。
- 图在正文中应该居中，图号、图题和图居中对齐，图号、图题和图的内容不能背页。
- 所有图片按照图中出现的章的顺序，编号保存文件。社区网站会提供图书所有图片的下载。
- 不要用深色背景，使用白色背景（仅适用黑白印刷）

### 框线线条图

- 避免把 PPT 直接截图作为正文图片（而是要提炼和重绘）。
- PPT/xmind/visio/PSD( 保留图层)等其他可复制文字的原图文件，方便绘图过程中复制文字并修改。
- 原始文件未输出成图片格式前无尺寸、分辨率要求。

### 代码输出可视化图片

- 代码输出 PDF 或 SVG 格式。
- PDF 或 SVG 原图文件保留文字可复制编辑功能，不要转曲，以便于排版绘图过程中按印刷要求处理图片。
- 可编辑的原始 PDF 或 SVG 无尺寸、分辨率要求。

### 界面截屏图

- 截取图片尽量尺寸和分辨率大一些，清晰度会更好。
- 截屏软件若截取出的图分辨率是固定的，图片尽量放大截取界面图。
- 软件界面颜色设置亮度尽量选择浅色，比较暗的截取出来不如浅色图印刷出来效果好。
- 界面截屏图上如有框线，可直接加在图中存成图片用。如果有指示线和说明文字，需要后期统一字体字号的，尽量不要做成不可编辑的图片如

PNG/JPG 等，如果只能提供图片形式的，也请尽量避开压界面原图上的字或其他需要体现出来的内容，后期处理时有的地方无法修改，只能画框覆盖住原内容再统一字体字号。

- 尺寸小的尽量分辨率要高一些（如宽 10cm，分辨率可 300 像素/英寸）；尺寸大的分辨率可适当小一点（如宽 50cm，分辨率可 72 像素/英寸）。

### 照片或由其他软件输出后的图片

- JPG/PNG/TIF 照片图、框线线条图等如直接使用不重绘处理，分辨率最好为 600 像素/英寸，最低不低于 300 像素/英寸。
- 尺寸小的话尽量分辨率要高（如宽 10cm，分辨率 300 像素/英寸）；尺寸大分辨率可适当变小（如宽 50cm，分辨率 72 像素/英寸）。

### 3.3.5 表格

表格能够高效地展示汇总或提炼后的信息。

表格需要遵从以下规范：

- 表要有标题，表题尽量简单直接。
- 表格内容不要用图片格式。
- 表要有编号，按章编号，编号和标题放表上方靠左。
- 表格的格式尽可能做到全书统一。
- 先文后表。正文中先提到表号，再给出表。
- 表号、表题和表的内容不能跨页，即必须在同一页中（但表的内容太多的时候，可以跨页）。

### 3.3.6 公式

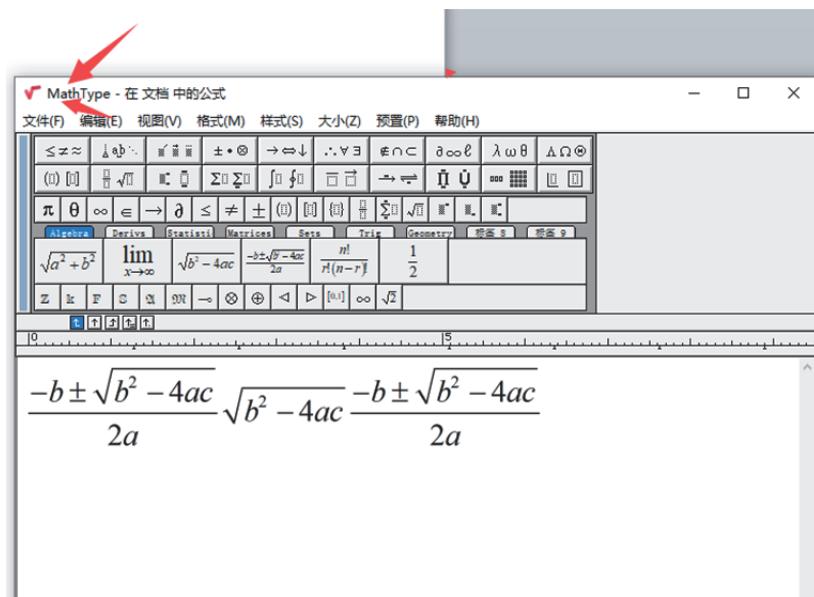
随着人工智能、深度学习技术的普及，现在很多的技术书会涉及公式。作者有必要更多地了解公式的规范和录入要求。

为了便于编辑检查核对公式以及排版人员录入修改公式，建议作者在写作中，对于公式的录入遵从如下的建议：

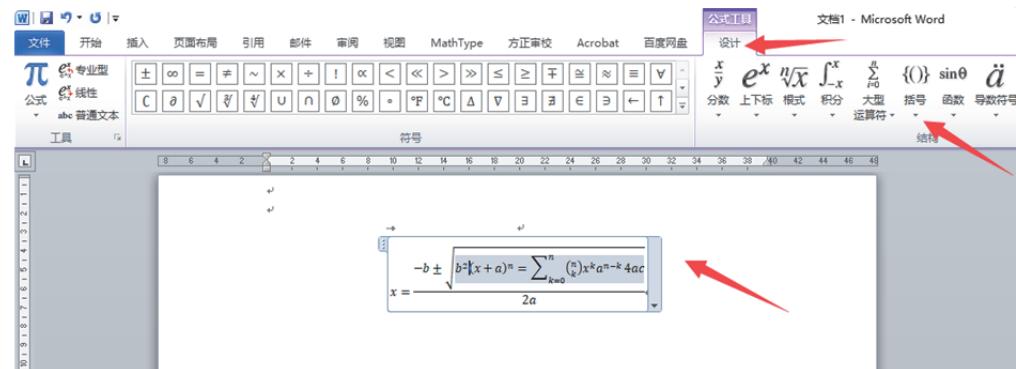
(1) 公式尽量同时提供 Pdf 版本和源文件（可以是 Word/Markdown/LaTeX）。

PDF 是为了方便编辑核对正确的公式格式，源文件是为了方便排版录入操作。

(2) 公式最好是用数学公式编辑器制作；



(3) 不要使用 Word 自带的公式编辑器，因为它在特殊字体控制和显示方面不稳定。如果您用 Word 自带的公式编辑器输入公式，为了保持公式的一致性和稳定性，排版人员仍然需要用 MathType 进行转换或重新录入。



### 3.4 杜绝抄袭

未经版权所有者的许可，使用文本、代码或者图片，将会使作者面临法律风险。这是不可接受的做法，并且代价昂贵；所以请不要这么做。

编辑会有专门的工具查重，并且对抄袭行为零容忍，一旦发现这种行为，会取消合同和选题。

### 3.5 规范引用

引用和抄袭是不同的，但合理引用往往难以证实。

因此，请特别留意如下几种情况，并且有任何疑问，直接联系和咨询编辑。

- 引用需要注明所引用的资源的来源和版权。
- 大段的引用也是有风险的，必须要得到版权所有者（作者或出版者）的正式许可。如果你拿不准，请咨询编辑。
- 引用自己发表的博客、论文等，需要考虑读过文章的人的感受，他们花钱购买本书并发现内容相近，是否会有上当的感觉。引用的内容应进行适当更新和改写。
- 引用自己以前出版的图书的内容，需要提前征得出版方的同意。
- 图片也是有版权的，直接使用通过百度搜索到的图片，可能会面临法律风险。
- 最好使用自己操作截屏图片，或者请出版社专门绘制图片。

## 4. 写作初稿

---

### 4.1 图书常见的结构

一般来说，技术图书包含如下一些组成部分

- 序言。邀请相关领域的知名 KOL 写的序是推荐序，作者自己写的序是自序。有助于图书的推广营销。对图书内容品质的加持和背书。序言一般概括阐述图书的思路和定位、意义和价值等。
- 前言。对于将要介绍的主要概念给出一个蓝图，以便读者能够熟悉它。讲清楚图书的内容、纳入这些内容的原因，以及介绍它们的顺序。描述学习的成果（读者学完了能够干什么）。最后，以提要的方式给出主要的内容。
- 正文。正文要介绍主要话题。分为若干个部分（篇），每个部分（篇）包含若干章。划分的方式主要是考虑技术内容、主题和读者的学习过程。
- 结尾篇章。对于图书所写内容总结以及对于未来学习路径的展望。告诉读者已经学到了什么。指出未来学习的方向和路径。

其中版权页、内容提要、赞誉、序言、致谢、前言等统称为文前。

文末的附录和文前一起，称为附文。

除此之外的部分，称为正文。

## 4.2 正文的篇章结构

学习目的决定了图书内容，图书内容决定了篇章结构和写作形式。

不同的技术图书，可能存在不同的学习目的，导致内容安排上有所差异，篇章结构和写作形式也需要有所差异。不一而足，我们举例说明。

以编程语言类的图书为例。这类图书帮助读者掌握一门编程语言的基础知识、语法、功能和应用案例，学会使用该语言编程解决实际问题。这类图书一般可以分为三个或四个部分，第一部分介绍语言的背景、应用场景和基础知识，第二部分介绍语言的核心语法和特性，第三部分介绍该语言的高级语法和功能，第四部分通过1个或数个典型应用案例介绍编程语言的实战技巧。

以“技术内幕/源码剖析”类的图书为例。该类图书从底层架构设计/源码剖析的角度，帮助读者了解某种技术的架构、设计思想和重要功能模块的实现原理。这类图书一般会分为

四个部分：第一部分是阅读源码前的准备（包括知识准备、软件安装、代码获取等），第二部分是解析核心功能模块的代码，第三部分可能是进阶内容，深入拓展，剖析一般人不太会用到或接触的高级功能模块；第四部分介绍与其他相关技术结合的应用场景。

为了便于说明，也为了方便技术写作者理解和掌握，我们选择最为常见的一种技术图书类型——某种技术的实战类图书为例，来详细介绍篇章结构的特点和各部分的写作方式。

一般来说，技术实战类图书的正文可以分为引入篇章、核心篇章和结尾篇章。

### 4.2.1 引入篇章

第1章：这一章是全书内容的一般性概览。

本章要告诉读者如下几个问题：

- 这本书的技术内容是什么？它和读者已经掌握的其他知识和技术有什么关系？
- 这项技术如何用到整个技术生态的更大的背景中？
- 从更高的层面来看，这项技术是如何工作的？如何绘制一张图（或思维导图）来说明这项技术？它和其他的技术或方法有什么区别？

- 这项技术的实用的好处是什么？请给出一些简短的例子来说明这些好处。

第 2 章：第 2 章通常是一个简单的教程示例，展示这项技术的实际用法。我们不可能通过这一个例子来让读者理解所有内容，但是，这个例子应该足够简单，能够展示出该技术的重要特性。理想情况下，读者能够自己跑通这个例子。

第 3 章（备选）：如果需要的话，这章可以概述你认为读者应该已经具备的所有预备知识。这章内容也可以放到附录中。

#### 4.2.2 核心篇章

接下来的各章需要介绍这项技术的核心部分。

基于读者已有的知识，接下来首先要学习什么？

还需要介绍一些其他的概念以便读者能够用这一技术做一些有用的事情吗？

我们能够提供哪些示例？

#### 4.2.3 结尾篇章

通常，我们要保证读者需要的所有信息都在核心篇章中有所涉及。

在结尾篇章中，我们通常会总结通过本书所学到的内容，并且为读者展示进一步的学习路径，此外，这里可以介绍某一部分特定读者可能会感兴趣或关心的主题。

### 4.3 前言的写法

前言的目的是帮助读者对本书的内容有一个概览。它是对一本书的一般性介绍。其主要内容包括：

- 图书的写作背景——包括相关领域或行业的发展情况，写作本书的目的。
- 主要内容介绍。本书涵盖哪些内容，可以依次介绍每篇或每章的内容。
- 目标读者：本书适合哪些读者阅读，能够解决他们的哪些问题；学完本书，他们能够干些什么。
- 如何阅读和使用图书：读者阅读本书需要具备的预备知识，学习本书所需的软件和下载素材。
- 体例说明。说明一些重要提示的含义，或者说明一些体例、字体、代码的特定用途。
- 服务和反馈方式：图书提供哪些辅助材料下载、是否有支持答疑，读者如何反馈意见。

- 致谢。图书写作和出版过程中需要感谢的人，包括家人、导师或同事、评测者、作序者、出版团队等。

## 4.4 一般章的写法

要写好一章内容，首先要写好每章的简介和知识要点。

一章的主要结构包括如下的部分：

- 本章简介。各章简介用 50-100 字，三四句话，说清楚该章内容；包括本章要写什么内容、适合谁阅读、为什么要安排这些内容，如何学习等。
- 知识要点。知识要点尽量和该章的二级标题对应。可以以列表的形式列出知识要点。关键是要讲清楚本章的主要话题，以及读者通过学习这些主题后能够做些什么。
- 技术需求：介绍学习本章所需要的具体技术知识。
- 本章小结（200-300 字）：告诉读者，通过本章学到了什么，可以继续学习什么。
- 建议每章篇幅控制在 30 页左右。

## 4.5 结尾篇章的写法

结尾篇章可以有如下一些内容。

- 仪式：告诉读者已经学到了哪些知识，并表示祝贺。
- 验证：在前言中，已经告诉读者学完本书将能解决什么问题，现在，展示一下他们能够如何解决这些问题。
- 继续：读者继续提升技术水平，还可以有哪些学习方向和路径。有哪些异步图书可以推荐？
- 反馈：鼓励读者给出反馈意见。

# 5. 检查初稿

---

## 5.1 一般性检查

完成初稿后，我们需要进行一些检查。

- 确保章节标题能够清晰地告诉读者将要做什么或学什么。最好的方法是使用“创建”“实现”这样的动词。
- 在黑白印刷的图中，避免提到图片中的颜色（全彩印刷的图书除外）。
- 总是记住向读者说明任务的目的，并且强化你要传递的观点。
- 检查简介部分——是否清晰表明了我们将要学习什么，读者将获得什么知识和技能，将要涉及哪些主题，以及为什么这对于读者来说很重要。
- 检查小结部分——是否清晰阐述了介绍了什么、学到了什么，这些知识和技能如何变得有用，以及下一章将要学习什么，如何继续进步。
- 图表、代码、链接等是否符合要求，正文中是否先提及这些内容。

## 5.2 分项列表检查

本节中的各个小节，针对需要检查的各个部分，给出一个分项检查列表。技术图书写作者可以根据具体情况，灵活使用这些检查列表。

### 5.2.1 目录大纲

- ✓ 是否包含了策划本书所准备的目录大纲中的所有主题和知识技能。
- ✓ 篇幅和目录大纲中设定的篇幅是否相符。章节不要太长或太短。
- ✓ 确保各章满足目录大纲中设定的学习目标。

### 5.2.2 技术内容

- ✓ 所有技术内容和代码，使用了最新版本。
- ✓ 所有代码都用一句话或一段话给出解释说明。
- ✓ 更新下载服务中的代码文件。
- ✓ 正文中不要给出 15 行以上的代码，对于太长的代码，只提供部分片段。

### 5.2.3 结构

- ✓ 每章开头给出主题简短介绍并且列出学习目标。
- ✓ 使用正确的标题/子标题格式来提高可读性。
- ✓ 在标题/子标题中尽可能使用动词。
- ✓ 在主要部分之间添加了过渡语句或段落，以体现内容之间的逻辑性。
- ✓ 章内的内容之间建立关联，以创建学习路径。

- ✓ 确保读者能够实现简介中提到的目标。

#### 5.2.4 实用性

- ✓ 使用现实的示例，保证内容可操作和实用性。
- ✓ 有限的背景知识和理论介绍。
- ✓ 使用图片来支持讲解，或者简化说明。
- ✓ 在每章结尾，阐述所教授技能的价值和实际应用。
- ✓ 当读者快要实现本章目标的时候，在正文中加以提醒。
- ✓ 使用“现在，我们已经学会了/理解了...”来强调所学的技能。
- ✓ 本章小结：回顾本章所学内容或强化其价值和应用。
- ✓ 在本章小结的结尾，介绍下一章的主题。
- ✓ 避免重复信息，可以交叉引用而不要重复。
- ✓ 避免任何导向性或政治不正确的叙述。
- ✓ 避免以任何形式抄袭任何内容（文本、图片、数据集、代码都要考虑）。
- ✓ 使用有版权或知识产权的内容，要寻求得到相应的许可。

#### 5.2.5 可读性

- ✓ 编写的时候就要考虑到读者，例如，写给初学者的图书，要较详细地介绍主题并辅以示例。
- ✓ 保证内容精简且直接，专注于主题，并且保持语句完整。
- ✓ 使用多种可视化的元素，如列表、信息提示框、图、表等。
- ✓ 在说明复杂的任务或涉及多步的代码执行的时候，添加编号的步骤。
- ✓ 在代码块和图像之前或之后，增加一些正文，以框定其位置。

## ↓ 6. 持续写作

---

### 6.1 写作计划

凡事预则立，不预则败。

要成功地写完一本书，需要制定一个切实可行的写作计划。

制定计划有以下几个基本要求：

(1) 自律：大多数人都不喜欢自律！

可是，无论你是在学习一门编程语言还是在写一本书，抽出专门的时间非常重要——即使你不喜欢这样做。每天留出一两个小时固定的、不被打扰的时间。

(2) 设定目标：每天写大约 1000 字。

(3) 制定一个大纲：这将让你保持专注。我们将在下一节中介绍这个大纲。

## 6.2 排除干扰

写作一本书，是一个长期的过程。在这个过程中你需要排除外界的干扰，持续把这件事情做下去，直到有一个阶段性的成果。

请注意，有时会感到沮丧或挫折，这是很正常的事情。

有些技巧可以帮你缓解：

- 散步
- 读一本书
- 消除掉分散你注意力的事情
- 随机地涂鸦
- 改变一下工作环境
- 听音乐
- 和朋友电话聊聊

## 说明和致谢

---

技术图书写作是一项比较复杂的技能。对于任何复杂的技能，人们总是可以通过总结方法和刻意练习来掌握并逐渐达到纯熟。

这个写作指南旨在总结和提炼方法，帮助那些想要成为技术图书作者的朋友，了解写作的方法、相关的规范和要求，以及完成技术写作的技巧，从而降低技术写作的门槛。

在编写这个写作指南的过程中，我参考了几家著名的欧美出版社的规范，包括 Packt Publishing 的作者写作指南、Manning Publications 的写作指导和选题表、NoStarch 的作者指南和选题表等。异步社区 5 周年大会上的演讲底稿，以及此前张涛编辑针对技术作者的宣讲材料等，为这个写作指南的编写提供了基本的思路和蓝本。国内一些出版同行的作者写作指导计划和讲座，也给我们提供了一些有益的启发。

汉魂公司的同事郭丽娟等对于代码、图片、表格和公式的规范，给出了较为详细的说明和指导。Lisa、余洁等同事结合自己的原创图书策划工作经验，站在作者的视角给出了很多非常切实有用的建议。在此对以上协助此项工作的公司、同行和同事，表示深深的感谢！

这个写作指南还远没达到完善的程度，也就是说还在继续迭代中。如果您在阅读和使用中有任何的意见和建议，欢迎反馈给我（[chenjikang@ptpress.com.cn](mailto:chenjikang@ptpress.com.cn)），以帮助我把它不断改进和完善，让更多的技术作者从中受益。