选题

好的选题和内容切入点是一本小册成功的基础，但同时也是最难的地方。要兼顾内容的难度和受众大小，并从中找出有差异性的角度是我们思考小册选题的基础。

举例来说，Git 基本语法的内容难度不高、但受众很大，因而就需要作者能够想出好的内容差异性；高并发架构设计的内容难度高、但受众有限，差异性依赖于作者是否有特别的实践经历；程序的设计模式是一个有难度的内容，而且大多数开发者如果想要提升自己的能力都需要学习，这个品类就很有可能出现大家需要的小册。

当然，作者并不是一个人在战斗！掘金的团队非常愿意根据掘金产品上的数据和我们的社区运营经验，来和每位作者一起分析当下的开发者最需要的内容。


```
1  掘金小册的选题原则：以一个明确的开发者需要的问题为出发点
```

👎 欠佳例子

TensorFlow 入门

- 问题：小册的选题问题不明确。入门？入门到怎样的水平？要能解决怎样的问题？是要把 TensorFlow 全都讲一遍？是要服务于什么具体的工作需求？要讲多细？这个题目没有办法把这些需求明确的表达清楚。
- 👍 选题优化：
  - TensorFlow 基本语法和模型运行指南
  - MNIST 机器学习方法及 TensorFlow 的运行实现
  - 基于 TensorFlow 实现一个图片自动分类器

Vue.js 2.0 升级踩坑记

- 问题：小册的选题过于细节。文段内容不足以支撑小册的产品模式，更适用于文章或是问答形式，且每一个业务的具体需求不同在升级中遇到的问题不尽相同，因而小册凝练出的内容不够有价值。
- 👍 选题优化：

- Vue.js 的前端组件化实践

搞下 ARKit

- 问题：小册的选题过于随意。既无法表达具体问题，标题整性不足，让读者觉得内容本身质量欠思考、甚至不佳。
- 👍 选题优化：
  - ARKit 快速实现：做一个桌子上的小动物
  - iOS ARKit 组件功能详解

确定小册大纲

提纲挈领，抓住要领才能让小册价值得以发挥。因而，在我们选择好了一个合适的选题之后，就需要现对整个小册的内容进行一个拆解，从而确定小册的大纲。

与市面上的书籍章节不同，掘金小册只支持第一级目录，我们称之为**小节**。这背后的目的是希望在掘金小册的内容大纲足够干练精准，减少读者获取信息的成本。章节上篇幅亦不要过于冗长，小节呈现一个有体系、有逻辑性的顺序，从而完成整个小册的小节结构。

另外一点，我们看到纸质书的一个共通的问题：

1. 一本书中包含大量线上可以找到且更新更及时的内容
2. 一本书冗长的内容中有很多章节读者不知道是否和自己的需求直接相关

注意，我这里绝对没有说一本完整的书籍不好，完整性和全面性是参考内容和深入理解内容很重要的部分。但是，我们面对的技术市场是一个高速迭代的市场，内容日新月异的同时，读者需要快速理解一个新技术的优势和具体实践；而作者也要应对技术的快速迭代，捕捉到足以让大多数人快速学习的内容，进而再持续深入。

几个掘金小册的推荐大纲结构

一、开发实现顺序

许多技术类内容都是要顺着具体的开发引导来的，因而按照实现的顺序一层一层、一步一步地讲解，是可以帮助读者从头到尾把内容连贯起来的。

读者获得了内容就可能出现版权问题？最后，讲明白掘金小册的需求，就希望更多人来写，因而留下申请成为小册作者的方法。

- 第一部分：介绍我们要做什么
- 第二部分：按照实现顺序按部就班地详解流程
- 第三部分：完成的结果，照应主题，留下读者可以继续深入

二、概念到细节

很多技术内容的讲解，尤其是一些新的技术，最先要讲明白背后的概念，让读者对这个内容有了一个概括性的了解，然后再深入背后的细节完成实体的内容。

以Git 原理详解及实用指南为例：Git 虽然为大家熟悉，但它本身是一个分布式版本管理工具（DVCS）及其基本的概念需要让读者先理解。然后来讲一些核心概念，如 repository、branch、stage 等等；然后讲基本操作以及最后的一些工作中会遇到的实用方法。一步步从概念走到细节。

- 第一部分：技术的基本概念
- 第二部分：技术的核心概念
- 第三部分：技术的基本功能和实用方法
- 第四部分：技术的高级实用技巧
- 第五部分：总结 + 深入学习的资源列表

三、总分平铺

还有一种小册的大纲结构是以经验总结、最佳实践、问题解决方案为主的平行内容。这样往往一开始先讲明白小册的核心目的和价值，然后就会围绕主题一个一个讲解决方法、或者一个个的补充内容。

以响应式编程 —— RxJava 高阶指南为例：小册的目标是去帮助读者理解 ReactiveX 概念和 RxJava 的高阶问题。在第一节解释清楚 Rx 的价值之后就会围绕着 Rx 和 RxJava 逐一讲解概念和高阶问题，但是各个问题之间的关系相对是平行的。

- 第一部分：小册要去解决的问题的重要性和核心内容
- 第二部分：一个一个地讲解内容
- 第三部分：总结

开头和结尾的重要性

我们建议一个小册的开头：

- 清楚讲解小册的核心价值和目的
- 配以合理的介绍性的文字、图片、代码，甚至是 Demo
- 对整本小册的结构大纲有一个概述性的介绍，为后面的核

读者完成一本小册的阅读，到达最后一个小节，可以获得其期待的知识信息。但是，读者往往在这一刹那希望可以获得更多、可以与作者和其他读者交流、可以有更多的资源和技术可以更深入的学习下去。

因而我们建议一个小册的结尾：

- 回顾整本小册，对内容、核心产出、重点做一次总结
- 援引一些资源、开源库、书籍、文章链接，让读者可以继续学习
- 给读者以与作者继续保持联系的方法

组织小册内容


```
1  掘金小册的内容组织方法论（不断更新）
```

1. 题目应简洁直接、直达主题
2. 小节的标题同样要简洁明了，一眼就明白小节的价值
3. 小节之间应连贯有逻辑，遵从一个大纲排布规则
4. 每一小节的平均阅读时长在 10 分钟左右
5. 每一个小节都应该在文段里有一个标题
6. 每一个小节最后要有一个小结来总结本小节的核心内容
7. 可以根据小册的特点，适当地设置一些练习题或问答题
8. 最后一个小节应该有一个总结和延伸阅读的入口
9. 图片要保证无版权风险，且大小合宜、描述清晰
10. 准确的代码区块，切忌大段粘贴代码
11. 内容排版中善用 `h2` 、 `h3` 、 `blockquote` 、 `code` 、 `ul` 、 `ol` 等元素
12. 在内容上多使用链接去援引外部内容

掘金责任编辑

申请成为小册作者

留言

掘金已支持深色模式

在"我的设置-通用设置"中即可配置

点击前往

看完啦， 立刻购买 分享一下感受吧~