## 目录写作注意点：

1. 目录是一个与读者交流的最好窗口，而不仅仅是将一种技术或话题拆分成几个相关子主题，所以在拟标题的时候，既要注意直接展现主题又要具有亲和力，写出自己的个性。
2. 目录是销售过程一个重要组织部分，告诉读者他们将要学到什么。
3. 在写目录之前，你要对读者开始学习这本书有可能要做什么有个非常明确的概念——清晰定位你的读者是在初级还是中级，拟或高级。
4. 对你的读者学习完整本图书后他们会得到什么，或者能做什么，也要明确——定义图书终点。
5. 上面两点是帮助你确认写这本书的起点和终点，所以你的目录就是读者从起点走到终点的路线图。
6. 每一章名要让所有读者，尤其是还没有翻开图书正文的读者一眼能清楚是在讲什么。
7. 每一个章名连接起来要向读者表明一个技巧或能力提升的过程。
8. 要从读者能力提升变化的角度来拟章名，而不是单纯地描述软件。
9. 章名越具体越好，如“高级”这样的形容词过于抽象，要避免使用，这样的词于读者理解作者意图毫无用处。
10. 章名要写你正在教读者做什么，而不是写功能名，比如写SQL，“从数据库抽取数据”就要比“SELECT语句”好，因为前者很清楚地描述了SELECT语句的功能，而初学者基本不太理解SELECT语句的原本意思。
11. 在申报选题时的目录与最终写作的目录可以有所变动，但是变动最好请编辑确认。

## 目录总体模板

请用以下模板构建您的目录。

**前言**

**估计页数：**

2-3 段简述下您的前言

**第1 章: 章名**

**估计页数：**

2-3 段简述下读者在第1章将要学到什么，怎样学。

* **1.1 节名**
* **1.2 节名**
* **1.3节名**
* **等等**

**第2章：章名**

**估计页数：**

2-3 段简述下读者在第2章将要学到什么，怎样学。

* **2.1 节名**
* **2.2 节名**
* **2.3节名**
* **等等**

**第3章: 章名**

**估计页数：**

2-3 段简述下读者在第3章将要学到什么，怎样学。  
List of topics that will be covered in the chapter:

* **3.1 节名**
* **3.2 节名**
* **3.3节名**
* **等等**

**第4章: 章名**

**估计页数：**

2-3 段简述下读者在第4章将要学到什么，怎样学。

List of topics that will be covered in the chapter:

**• 4.1 节名**

**• 4.2 节名**

**• 4.3节名**

**• 等等**

**第5章: 章名**

**估计页数：**

2-3 段简述下读者在第5章将要学到什么，怎样学。

List of topics that will be covered in the chapter:

**• 5.1 节名**

**• 5.2 节名**

**• 5.3节名**

**• 等等**

**第6章: 章名**

**估计页数：**

2-3 段简述下读者在第6章将要学到什么，怎样学。

List of topics that will be covered in the chapter:

**• 6.1 节名**

**• 6.2 节名**

**• 6.3节名**

**• 等等**

**依此类推**

**Total 估计页数 =**

## 故事+案例带出技术类目录示例

**第1章 计算机的世界你不懂 1**

1.1 我是一个线程 1

初生牛犊 1

渐入佳境 4

虎口脱险 5

江湖再见 8

1.2 TCP/IP之大明内阁 9

1.3 TCP/IP之大明邮差 12

1.4 CPU阿甘 17

启动 17

运行 18

新装备：缓存 19

自我提升：流水线 20

1.5 我是一个进程 21

批处理系统 22

多道程序 22

地址重定位 24

分时系统 26

分块装入内存 27

虚拟内存：分页 28

分段 分页 30

程序的装载 31

线程 35

1.6 我是一块硬盘 36

内部结构 37

文件 38

文件的存放 39

管理空闲块 44

文件系统 45

1.7 我是一个键盘 46

二等公民 46

总线和端口 47

轮询（程序式I/O） 48

中断 49

DMA 50

1.8 数据库的奇妙之旅 51

无纸化办公 51

数据的冗余和不一致 52

李氏查询 52

并发访问 54

原子性问题 56

安全 58

1.9 搞清楚Socket 59

1.10 从1加到100：一道简单的数学题挑战一下你的大脑 64

CPU和内存 64

从1加到100 65

热身 65

正式出发 65

1.11 一个翻译家族的发家史 68

机器语言 68

汇编语言 69

高级语言 70

1.12 编程世界的那把锁 74

共享变量惹的祸 74

争抢吧，线程 75

改进 76

信号量 77

1.13 绕不开的加法器 81

1.14 递归那点事儿 85

**第2章 Java帝国 91**

2.1 Java：一个帝国的诞生 91

C语言帝国的统治 91

反抗 92

一鸣惊人 93

开拓疆土 93

帝国的诞生 94

2.2 我是一个Java Class 96

陌生警察 96

刺探信息 97

初识虚拟机 98

快乐假期 102

真相大白 103

2.3 持久化：Java帝国反击战 104

断电的威胁 104

数据库联合酋长国 105

表面风光的EJB 106

轻量级O/R Mapping框架 106

帝国的反击 107

2.4 JDBC的诞生 108

谈判 108

统一接口 110

面向接口编程 111

简单工厂 112

数据驱动 114

工厂方法 115

2.5 Java帝国之宫廷内斗 119

JDBC大臣 119

密谋 120

两阶段提交 121

JTA 123

塞翁失马，焉知非福 124

基本可用 125

走漏风声 127

宫廷激辩 128

2.6 JSP：一个装配工的没落 131

黑暗岁月 131

服务器端动态页面 132

标签库 134

模板引擎 134

草根搅局 135

2.7 Java 帝国之消息队列 137

张家村的历史 137

拆分 138

新问题 139

消息队列 139

互不兼容的MQ 140

消息队列接口设计 141

配置和代码的分离 143

再次抽象 144

2.8 Java帝国之动态代理 147

深夜奏对 147

明察暗访 149

Java 动态代理 150

2.9 Java注解是怎么成功上位的 153

XML大臣 153

安翰林献计 154

早朝争斗 157

2.10 Java帝国之泛型 159

新王登基 159

C 使者 159

泛型实现 162

泛型方法 163

泛型和继承 164

2.11 一个著名的日志系统是怎么设计出来的 166

前言 166

张家村 167

小张的设计 167

正交性 170

Log4j 171

尾声 171

2.12 序列化：一个老家伙的咸鱼翻身 172

寒冬的蛰伏 172

XML和JSON的挑战 173

新协议的崛起 175

2.13 加锁还是不加锁，这是一个问题 176

互斥锁 176

要不要加锁 177

CAS的扩展 179

2.14 Spring 的本质 182

问题来源 183

设计模式：模板方法 184

设计模式：装饰者 186

AOP 187

实现AOP 188

对象的创建 189

IoC与DI 191

**第3章 浪潮之巅的Web 192**

3.1 Web的起源 192

3.2 两个程序的爱情故事 195

好感 195

分离 196

网络 197

Web 198

3.3 一个故事讲完HTTPS 199

总有一种被偷窥的感觉 199

RSA：非对称加密 200

非对称加密 对称加密 202

中间人劫持 202

你到底是谁 203

HTTPS 206

3.4 机房夜话 207

第一夜 207

第二夜 209

第三夜 211

3.5 从密码到token，一个有关授权的故事 215

我把密码献给你 215

token 216

授权码 token 218

后记 220

3.6 后端风云 220

数据库老头儿 220

危机 221

党委扩大会议 221

分家 223

Redis 224

余数算法 227

一致性Hash算法 229

Hash槽 （Hash Slot） 233

故障转移 235

高可用的Nginx 237

高可用的Tomcat 238

数据库的读写分离 240

3.7 我是一个函数 243

3.8 从SOA到微服务 246

3.9 什么是框架 249

3.10 HTTP Server：一个差生的逆袭 252

HTTP Server 1.0 252

HTTP Server 2.0：多进程 253

HTTP Server 3.0：select模型 254

HTTP Server 4.0：epoll模型 255

**第4章 代码管理那些事儿 258**

4.1 版本管理简史 258

“人肉” 版本管理 258

锁定文件：避免互相覆盖 259

允许冲突：退一步海阔天空 260

分支：多版本并行 261

分布式管理：给程序员放权 262

程序员也爱社交 264

4.2 Build的演进之路 265

手工Build的烦恼 265

自动化Build 267

Java 与 XML 268

消除重复 270

4.3 烂代码传奇 275

4.4 敏捷下的单元测试 277

敏捷运动 277

困惑 278

讨论 279

一年以后 280

4.5 再见！Bug 281

**第5章 我的编程语言简史 284**

5.1 JavaScript ：一个草根的逆袭 284

出世 284

成长 286

第一桶金 288

发明JSON 289

人生巅峰 291

5.2 Node.js：我只需要一个店小二 292

5.3 C老头儿和Java小子的硬盘夜话 295

5.4 编程语言的“爱恨情仇” 300

让人怀疑的C 语言 300

被忘却的 VB & Visual FoxPro 301

蹂躏我的C 302

赖以谋生的Java 303

优雅的Ruby 304

5.5 命令式编程 VS 声明式编程 306

一则小故事 306

命令式编程 306

声明式编程 307

**第6章 老司机的精进 309**

6.1 凡事必先骑上虎背，和性格内向的程序员聊几句 309

6.2 码农需要知道的“潜规则” 311

上帝的规矩：局部性原理 311

坐飞机的怎么和坐驴车的打交道：缓存 311

抛弃细节：抽象 312

我只想和邻居打交道：分层 312

我怕等不及：异步调用 313

大事化小，小事化了：分而治之 314

6.3 15年编程生涯，一名架构师的总结 314

好奇心 315

养成计算机的思维方式 315

扎实基础，融会贯通 317

要透彻地理解一门技术的本质 317

能写漂亮的代码 318

抽象的能力 319

技术领导力 319

6.4 对自己狠一点，开始写作吧 320

6.5 学习编程的四兄弟 322

摇摆不定的老大 322

“小仓鼠”老二 322

不看书的老三 323

半途而废的老四 323

## Cookbook技术点细分目录示例

**第1章　数据结构和算法 1**  
1．1　将序列分解为单独的变量 1  
1．2　从任意长度的可迭代对象中分解元素 3  
1．3　保存最后N个元素 5  
1．4　找到最大或最小的N个元素 7  
1．5　实现优先级队列 9  
1．6　在字典中将键映射到多个值上 11  
1．7　让字典保持有序 13  
1．8　与字典有关的计算问题 14  
1．9　在两个字典中寻找相同点 15  
1．10　从序列中移除重复项且保持元素间顺序不变 17  
1．11　对切片命名 18  
1．12　找出序列中出现次数最多的元素 20  
1．13　通过公共键对字典列表排序 22  
1．14　对不原生支持比较操作的对象排序 23  
1．15　根据字段将记录分组 25  
1．16　筛选序列中的元素 26  
1．17　从字典中提取子集 29  
1．18　将名称映射到序列的元素中 30  
1．19　同时对数据做转换和换算 33  
1．20　将多个映射合并为单个映射 34  
**第2章　字符串和文本 37**2．1　针对任意多的分隔符拆分字符串 37  
2．2　在字符串的开头或结尾处做文本匹配 38  
2．3　利用Shell通配符做字符串匹配 40  
2．4　文本模式的匹配和查找 42  
2．5　查找和替换文本 45  
2．6　以不区分大小写的方式对文本做查找和替换 47  
2．7　定义实现最短匹配的正则表达式 48  
2．8　编写多行模式的正则表达式 49  
2．9　将Unicode文本统一表示为规范形式 50  
2．10　用正则表达式处理Unicode字符 52  
2．11　从字符串中去掉不需要的字符 53  
2．12　文本过滤和清理 54  
2．13　对齐文本字符串 57  
2．14　字符串连接及合并 59  
2．15　给字符串中的变量名做插值处理 62  
2．16　以固定的列数重新格式化文本 64  
2．17　在文本中处理HTML和XML实体 66  
2．18　文本分词 67  
2．19　编写一个简单的递归下降解析器 70  
2．20　在字节串上执行文本操作 80  
**第3章　数字、日期和时间 83**3．1　对数值进行取整 83  
3．2　执行精确的小数计算 85  
3．3　对数值做格式化输出 87  
3．4　同二进制、八进制和十六进制数打交道 89  
3．5　从字节串中打包和解包大整数 90  
3．6　复数运算 92  
3．7　处理穷大和NaN 94  
3．8　分数的计算 96  
3．9　处理大型数组的计算 97  
3．10　矩阵和线性代数的计算 101  
3．11　随机选择 103  
3．12　时间换算 105  
3．13　计算上周5的日期 107  
3．14　找出当月的日期范围 108  
3．15　将字符串转换为日期 110  
3．16　处理涉及到时区的日期问题 112  
**第4章　迭代器和生成器 114**4．1　手动访问迭代器中的元素 114  
4．2　委托迭代 115  
4．3　用生成器创建新的迭代模式 116  
4．4　实现迭代协议 118  
4．5　反向迭代 121  
4．6　定义带有额外状态的生成器函数 122  
4．7　对迭代器做切片操作 123  
4．8　跳过可迭代对象中的前一部分元素 124  
4．9　迭代所有可能的组合或排列 127  
4．10　以索引-值对的形式迭代序列 129  
4．11　同时迭代多个序列 131  
4．12　在不同的容器中进行迭代 133  
4．13　创建处理数据的管道 134  
4．14　扁平化处理嵌套型的序列 137  
4．15　合并多个有序序列，再对整个有序序列进行迭代 139  
4．16　用迭代器取代while循环 140  
**第5章　文件和I/O 142**5．1　读写文本数据 142  
5．2　将输出重定向到文件中 145  
5．3　以不同的分隔符或行结尾符完成打印 145  
5．4　读写二进制数据 146  
5．5　对已不存在的文件执行写入操作 149  
5．6　在字符串上执行I/O操作 150  
5．7　读写压缩的数据文件 151  
5．8　对固定大小的记录进行迭代 152  
5．9　将二进制数据读取到可变缓冲区中 153  
5．10　对二进制文件做内存映射 155  
5．11　处理路径名 157  
5．12　检测文件是否存在 158  
5．13　获取目录内容的列表 159  
5．14　绕过文件名编码 161  
5．15　打印法解码的文件名 162  
5．16　为已经打开的文件添加或修改编码方式 164  
5．17　将字节数据写入文本文件 166  
5．18　将已有的文件描述符包装为文件对象 167  
5．19　创建临时文件和目录 169  
5．20　同串口进行通信 171  
5．21　序列化Python对象 172  
**第6章　数据编码与处理 177**  
6．1　读写CSV数据 177  
6．2　读写JSON数据 181  
6．3　解析简单的XML文档 186  
6．4　以增量方式解析大型XML文件 188  
6．5　将字典转换为XML 192  
6．6　解析、修改和重写XML 194  
6．7　用命名空间来解析XML文档 196  
6．8　同关系型数据库进行交互 198  
6．9　编码和解码十六进制数字 201  
6．10　Base64编码和解码 202  
6．11　读写二进制结构的数组 203  
6．12　读取嵌套型和大小可变的二进制结构 207  
6．13　数据汇总和统计 218  
**第7章　函数 221**  
7．1　编写可接受任意数量参数的函数 221  
7．2　编写只接受关键字参数的函数 223  
7．3　将元数据信息附加到函数参数上 224  
7．4　从函数中返回多个值 225  
7．5　定义带有默认参数的函数 226  
7．6　定义匿名或内联函数 229  
7．7　在匿名函数中绑定变量的值 230  
7．8　让带有N个参数的可调用对象以较少的参数形式调用 232  
7．9　用函数替代只有单个方法的类 235  
7．10　在回调函数中携带额外的状态 236  
7．11　内联回调函数 240  
7．12　访问定义在闭包内的变量 242  
**第8章　类与对象 246**8．1　修改实例的字符串表示 246  
8．2　自定义字符串的输出格式 248  
8．3　让对象支持上下文管理协议 249  
8．4　当创建大量实例时如何节省内存 251  
8．5　将名称封装到类中 252  
8．6　创建可管理的属性 254  
8．7　调用父类中的方法 259  
8．8　在子类中扩展属性 263  
8．9　创建一种新形式的类属性或实例属性 267  
8．10　让属性具有惰性求值的能力 271  
8．11　简化数据结构的初始化过程 274  
8．12　定义一个接口或抽象基类 278  
8．13　实现一种数据模型或类型系统 281  
8．14　实现自定义的容器 287  
8．15　委托属性的访问 291  
8．16　在类中定义多个构造函数 296  
8．17　不通过调用init来创建实例 298  
8．18　用Mixin技术来扩展类定义 299  
8．19　实现带有状态的对象或状态机 305  
8．20　调用对象上的方法，方法名以字符串形式给出 311  
8．21　实现访问者模式 312  
8．22　实现非递归的访问者模式 317  
8．23　在环状数据结构中管理内存 324  
8．24　让类支持比较操作 327  
8．25　创建缓存实例 330  
**第9章　元编程 335**9．1　给函数添加一个包装 335  
9．2　编写装饰器时如何保存函数的元数据 337  
9．3　对装饰器进行解包装 339  
9．4　定义一个可接受参数的装饰器 341  
9．5　定义一个属性可由用户修改的装饰器 342  
9．6　定义一个能接收可选参数的装饰器 346  
9．7　利用装饰器对函数参数强制执行类型检查 348  
9．8　在类中定义装饰器 352  
9．9　把装饰器定义成类 354  
9．10　把装饰器作用到类和静态方法上 357  
9．11　编写装饰器为被包装的函数添加参数 359  
9．12　利用装饰器给类定义打补丁 362  
9．13　利用元类来控制实例的创建 364  
9．14　获取类属性的定义顺序 367  
9．15　定义一个能接受可选参数的元类 370  
9．16　在\*args和\*\*kwargs上强制规定一种参数签名 372  
9．17　在类中强制规定编码约定 375  
9．18　通过编程的方式来定义类 378  
9．19　在定义的时候初始化类成员 382  
9．20　通过函数注解来实现方法重载 384  
9．21　避免出现重复的属性方法 391  
9．22　以简单的方式定义上下文管理器 393  
9．23　执行带有局部副作用的代码 395  
9．24　解析并分析Python源代码 398  
9．25　将Python源码分解为字节码 402  
**第10章　模块和包 406**  
10．1　把模块按层次结构组织成包 406  
10．2　对所有符号的导入进行精确控制 407  
10．3　用相对名称来导入包中的子模块 408  
10．4　将模块分解成多个文件 410  
10．5　让各个目录下的代码在统一的命名空间下导入 413  
10．6　重新加载模块 415  
10．7　让目录或zip文件成为可运行的脚本 416  
10．8　读取包中的数据文件 417  
10．9　添加目录到sys．path中 418  
10．10　使用字符串中给定的名称来导入模块

## 产品生命周期发展目录示例

引子 1

**第1章　互联网系统架构的挑战 2**

1.1 云应用架构技术发展 4

1.2 云平台下架构的不同点 5

1.2.1 开发模式的区别 6

1.2.2 交付模式的区别 7

1.2.3 架构设计的区别 8

1.3 云原生应用架构 10

1.4 架构演化发展历程 21

1.4.1 初创期架构 22

1.4.2 快速成长期架构 24

1.4.3 分布式服务架构 26

1.5 云计算服务介绍 29

1.6 云计算解决方案 31

1.7 案例概述 34

1.7.1 背景介绍 34

1.7.2 环境要求 36

1.7.3 项目构建 36

1.7.4 项目运行 36

1.7.5 相关技术介绍 37

小结 40

**第2章 从0到1工程实践 41**

2.1 工程化 41

2.1.1 工程模板 41

2.1.2 模块化 45

2.1.3 工程化构建 50

2.1.4 代码规范及检查 53

2.1.5 代码版本管理 54

2.1.6 环境划分 61

2.2 基于容器工程化 62

2.2.1 Docker及作用 63

2.2.2 Docker镜像及操作 66

2.2.3 Docker容器及操作 73

2.2.4 基于容器工程化 77

2.3 实战示例 78

小结 84

**第3章 初创期应用架构实践 85**

3.1 技术选型 85

3.1.1 业务框架选型 85

3.1.2 结构化数据存储 92

3.1.3 缓存选型 102

3.1.4 静态资源存储 106

3.2 架构实践 109

3.2.1 快速迭代 109

3.2.2 高可用与负载均衡 111

3.2.3 交付与部署 117

3.2.4 Web应用安全 119

3.3 应用监控 127

3.3.1 应用监控指标 127

3.3.2 应用进程监控 128

3.3.3 操作系统监控 129

小结 136

**第4章 快速成长期应用架构实践 137**

4.1 关键业务需求 137

4.1.1 计数与排序 137

4.1.2 秒杀 146

4.1.3 全文检索 149

4.1.4 日志收集 154

4.2 架构实践 156

4.2.1 前端系统扩展 157

4.2.2 无状态服务设计 157

4.2.3 在线水平扩展 160

4.2.4 后端系统扩展 163

4.2.5 系统通信 173

4.2.6 消息中间件 176

4.3 系统优化 181

4.3.1 静态资源分离 182

4.3.2 数据库调优 185

4.3.3 系统高可用 193

4.4 应用诊断 200

4.4.1 应用健康检查 200

4.4.2 性能问题诊断 204

4.4.3 基于日志的故障诊断 210

4.5 数据库诊断 214

4.6 DevOps 223

4.6.1 持续集成 224

4.6.2 持续交付 227

4.6.3 灰度发布 229

4.6.4 大应用编排 231

4.7 安全设计 246

4.7.1 入侵检测 247

4.7.2 防劫持攻击 249

小结 255

**第5章 稳定期服务化应用架构实践 256**

5.1 业务拆分 256

5.2 统一配置中心 259

5.3 分布式定时任务 261

5.3.1 分布式定时任务设计 262

5.3.2 业界流行的开源框架 264

5.4 分布式锁系统 274

5.5 微服务化架构 277

5.5.1 服务发现 279

5.5.2 服务治理 302

5.5.3 微服务框架 307

5.5.4 服务编排 313

5.5.5 微服务测试 321

5.6 分布式数据一致性 333

5.6.1 CAP和BASE理论 333

5.6.2 一致性模型 336

5.6.3 典型的解决方案 337

5.7 同城多活 344

5.7.1 应用同城多活 345

5.7.2 跨AZ负载均衡 347

5.8 故障诊断 348

小结 353

参考文献 354

技术术语 356