目录

第1章 如何学习Visual Studio Code 1

1.1 学会搜索 1

1.2 学会提问 2

1.3 学会学习 3

1.3.1 自己的思考 3

1.3.2 知其然知其所以然 3

1.3.3 举一反三 4

第2章 Visual Studio Code简介 5

2.1 Visual Studio Code概览 5

2.1.1 跨平台 5

2.1.2 IntelliSense 5

2.1.3 代码调试 6

2.1.4 内置的Git支持 6

2.2 Visual Studio Code简史 6

2.3 Visual Studio Code的优势 7

2.3.1 学习曲线 7

2.3.2 用户体验 8

2.3.3 性能 8

2.3.4 插件 8

2.3.5 生态 9

2.4 Visual Studio Code开发团队 9

2.5 Visual Studio Code是如何做开源的 10

2.5.1 代码开源 10

2.5.2 Issues和Pull requests 11

2.5.3 开源的开发流程 11

2.5.4 开源的生态 11

第3章 核心组件 12

3.1 Electron 12

3.2 Monaco Editor 13

3.3 TypeScript 13

3.4 Language Server Protocol 15

3.5 Debug Adapter Protocol 16

3.6 Xterm.js 18

第4章 安装与配置 20

4.1 概览 20

4.1.1 硬件要求 20

4.1.2 平台支持 20

4.1.3 跨平台 21

4.1.4 更新频率 21

4.1.5 附加组件 21

4.2 Linux 21

4.2.1 安装 21

4.2.2 更新 23

4.2.3 把Visual Studio Code设置为默认编辑器 23

4.2.4 使用Windows系统进行Linux开发 24

4.3 macOS 24

4.3.1 安装 24

4.3.2 从终端命令行启动 24

4.3.3 触控栏 25

4.3.4 首选项菜单 25

4.4 Windows 25

4.4.1 安装 25

4.4.2 从终端命令行启动 25

4.4.3 用户安装与系统安装 25

第5章 快速入门 27

5.1 Visual Studio Code Insiders 27

5.2 设置 27

5.2.1 两种不同范围的设置 28

5.2.2 两种设置方法 28

5.2.3 设置编辑器 28

5.2.4 JSON设置文件 30

5.2.5 语言的特定设置 31

5.2.6 设置与安全 33

5.2.7 常用的设置项 33

5.3 用户界面 34

5.3.1 基本布局 34

5.3.2 命令面板 35

5.3.3 并排编辑 37

5.3.4 缩略图 38

5.3.5 面包屑导航 39

5.3.6 文件资源管理器 40

5.3.7 禅模式 42

5.3.8 Tab（标签页） 42

5.3.9 窗口管理 43

5.4 编辑功能 44

5.4.1 多光标 44

5.4.2 列选择 44

5.4.3 自动保存 44

5.4.4 热退出 45

5.4.5 搜索与替换 45

5.4.6 跨文件搜索 46

5.4.7 IntelliSense 47

5.4.8 代码格式化 47

5.4.9 代码折叠 48

5.4.10 缩进 49

5.4.11 文件编码 49

5.5 主题 50

5.5.1 设置颜色主题 51

5.5.2 设置文件图标主题 52

5.5.3 插件市场中的主题 53

5.6 快捷键 53

5.6.1 快捷键编辑器 53

5.6.2 快捷键大全 54

5.6.3 键盘映射插件 55

5.6.4 解决快捷键冲突 56

5.6.5 查看默认的快捷键 56

5.6.6 查看更改的快捷键 57

5.6.7 高级配置 57

5.6.8 快捷键规则 58

5.6.9 有效的按键组合 58

5.6.10 常用的快捷键 59

5.7 集成终端 61

5.7.1 打开集成终端 61

5.7.2 管理多个终端 61

5.7.3 配置终端 62

5.7.4 终端的显示样式 63

5.7.5 终端的快捷键 63

5.7.6 运行选中的文本 64

5.7.7 重命名终端 65

5.7.8 设置终端的打开路径 65

5.8 中文显示 65

第6章 进阶应用 67

6.1 命令行 67

6.1.1 命令行帮助 67

6.1.2 通过命令行启动Visual Studio Code 68

6.1.3 命令行核心选项 68

6.1.4 打开文件和文件夹 68

6.1.5 通过命令行管理插件 69

6.1.6 命令行高级选项 69

6.2 IntelliSense 69

6.2.1 不同编程语言的IntelliSense 69

6.2.2 IntelliSense功能 70

6.2.3 自动补全的类型 71

6.2.4 自定义IntelliSense 71

6.3 代码导航 73

6.3.1 文件快速导航 73

6.3.2 面包屑导航 73

6.3.3 代码导航右键菜单 76

6.3.4 跳转到定义 76

6.3.5 跳转到类型定义 77

6.3.6 查找所有引用 77

6.3.7 通过内联编辑器查看定义和引用 78

6.3.8 引用信息 78

6.3.9 跳转到实现 79

6.3.10 跳转到文件中的符号 79

6.3.11 跳转到工作区中的符号 80

6.3.12 括号匹配 80

6.3.13 错误与警告 80

6.4 玩转Git 81

6.4.1 版本控制插件 81

6.4.2 安装Git 82

6.4.3 克隆Git仓库 82

6.4.4 源代码管理视图 82

6.4.5 Git commit 82

6.4.6 Git diff 83

6.4.7 Git分支 84

6.4.8 Git状态栏 85

6.4.9 Gutter提示 86

6.4.10 合并冲突 86

6.4.11 把Visual Studio Code作为Git编辑器 87

6.5 打造自己的主题 87

6.5.1 自定义工作台的颜色主题 88

6.5.2 自定义编辑器的颜色主题 89

6.5.3 颜色主题配置大全 90

6.6 快速创建可复用的代码片段 90

6.6.1 代码片段插件 90

6.6.2 使用代码片段 91

6.6.3 创建自定义的代码片段 92

6.6.4 代码片段的生效范围 93

6.6.5 代码片段的语法 94

6.6.6 为代码片段添加快捷键 96

6.7 Task，把重复的工作自动化 97

6.7.1 配置第一个Hello World的Task 97

6.7.2 配置一个更复杂的Task 99

6.7.3 Task自动检测 100

6.7.4 自定义自动检测的Task 102

6.7.5 问题匹配器 103

6.7.6 命令参数 103

6.7.7 输出行为 104

6.7.8 运行行为 105

6.7.9 变量替换 105

6.7.10 命令面板 109

6.7.11 快捷键绑定 109

6.7.12 操作系统的相关属性 110

6.7.13 后台运行的Task 110

6.8 Multi-root Workspaces 112

6.8.1 管理文件夹 112

6.8.2 工作区文件 114

6.8.3 用户界面 116

6.8.4 设置 117

6.8.5 源代码管理 119

6.8.6 插件推荐 119

6.9 调试与运行 120

6.9.1 调试器插件 120

6.9.2 调试与运行视图 121

6.9.3 调试与运行菜单 121

6.9.4 调试模式与运行模式 122

6.9.5 launch.json调试配置 122

6.9.6 launch.json属性 123

6.9.7 变量替换 123

6.9.8 与操作系统相关的属性 124

6.9.9 全局的launch.json配置 124

6.9.10 多目标调试 124

第7章 插件 126

7.1 插件市场 126

7.1.1 插件市场主页 126

7.1.2 插件搜索 128

7.1.3 插件页面 131

7.2 插件管理 133

7.2.1 搜索与浏览 133

7.2.2 通过插件管理视图管理插件 138

7.2.3 通过命令行管理插件 141

7.2.4 离线安装插件 142

7.2.5 插件推荐 142

7.2.6 插件的安装目录 143

7.3 那些不错的插件 143

7.3.1 REST Client：也许是比Postman更好的选择 143

7.3.2 Code Runner：代码一键运行，支持40多种语言 147

7.3.3 为你的代码再添上一抹亮色 150

7.3.4 更强的Git集成 154

7.3.5 Web开发利器 157

7.3.6 轻松管理数据库 167

7.3.7 提升开发效能 169

7.3.8 好用的工具类插件 171

7.3.9 容器开发 172

7.3.10 移动开发 172

7.3.11 LeetCode插件：程序员的Offer收割利器 173

7.3.12 有点儿好看的主题插件 173

7.3.13 不止代码！放松一下，那些劳逸结合的插件 174

第8章 语言深入 176

8.1 概览 176

8.1.1 编程语言插件 176

8.1.2 对编程语言的支持 177

8.1.3 为文件设置编程语言的类型 177

8.1.4 语言ID 178

8.1.5 把文件扩展名添加到编程语言中 179

8.2 Python 180

8.2.1 快速开始 180

8.2.2 代码编辑 185

8.2.3 静态代码检查 190

8.2.4 调试 192

8.2.5 Jupyter Notebooks 198

8.2.6 Python交互式窗口 204

8.2.7 测试 207

8.2.8 Python插件推荐 212

8.3 JavaScript 213

8.3.1 JavaScript、ECMAScript与Node.js 213

8.3.2 快速开始 214

8.3.3 代码编辑 216

8.3.4 调试 222

8.2.5 静态代码检查 230

8.2.6 测试 231

8.2.7 JavaScript插件推荐 232

8.4 TypeScript 233

8.4.1 快速开始 233

8.4.2 一键运行TypeScript 235

8.4.3 编译 237

8.4.4 调试 238

8.4.5 代码编辑 241

8.5 Java 242

8.5.1 快速开始 242

8.5.2 代码编辑 245

8.5.3 调试 248

8.5.4 测试 252

8.5.5 Java项目管理 254

8.5.6 Java插件推荐 259

8.6 C# 261

8.6.1 快速开始 261

8.6.2 代码编辑 263

8.6.3 调试 266

8.6.4 测试 268

8.6.5 C#插件推荐 270

8.7 C/C++ 271

8.7.1 快速开始 271

8.7.2 调试 273

8.7.3 设置 278

8.7.4 代码编辑 282

8.7.5 C/C++插件推荐 284

8.8 Go 284

8.8.1 快速开始 284

8.8.2 调试 286

8.8.3 代码编辑 288

8.8.4 测试 291

8.9 更多语言支持 291

8.9.1 PHP 292

8.9.2 Rust 292

8.9.3 Dart 293

8.9.4 Ruby 293

8.9.5 Lua 294

8.9.6 R 294

8.9.7 Matlab 296

8.9.8 D 296

8.9.9 F# 296

第9章 前端开发 298

9.1 HTML 298

9.1.1 IntelliSense 298

9.1.2 自动闭合标签 299

9.1.3 颜色选择器 299

9.1.4 验证嵌入的JavaScript和CSS 300

9.1.5 代码折叠 300

9.1.6 代码格式化 301

9.1.7 自定义HTML数据格式 302

9.1.8 HTML插件推荐 303

9.2 CSS、SCSS和Less 305

9.2.1 IntelliSense 305

9.2.2 颜色预览 305

9.2.3 颜色选择器 305

9.2.4 代码折叠 306

9.2.5 静态代码检查 307

9.2.6 跳转到CSS符号 307

9.2.7 悬停预览 307

9.2.8 自定义CSS数据格式 307

9.2.9 CSS插件推荐 309

9.3 Emmet 310

9.3.1 Emmet的支持范围 310

9.3.2 在HTML中使用Emmet 310

9.3.3 在CSS中使用Emmet 311

9.3.4 使用Tab键展开Emmet缩写 311

9.3.5 在建议列表中禁用Emmet缩写 312

9.3.6 Emmet缩写在建议列表中的顺序 312

9.3.7 在其他文件中启用Emmet缩写 312

9.3.8 Emmet设置项 312

9.4 React 313

9.4.1 快速开始 313

9.4.2 调试React 314

9.4.3 IntelliSense 316

9.4.4 代码导航 317

9.4.5 静态代码检查 317

9.4.6 React插件推荐 317

9.5 Angular 318

9.5.1 快速开始 318

9.5.2 调试Angular 319

9.5.3 IntelliSense 321

9.5.4 代码导航 321

9.5.5 Angular插件推荐 321

9.6 Vue 322

9.6.1 快速开始 322

9.6.2 Vetur插件 323

9.6.3 调试Vue 323

9.6.4 IntelliSense 326

9.6.5 代码导航 326

9.6.6 静态代码检查 326

9.6.7. Vue插件推荐 327

9.7 前端插件推荐 327

9.7.1 Beautify 327

9.7.2 Prettier - Code formatter 327

9.7.3 JavaScript (ES6) code snippets 328

第10章 云计算开发 329

10.1 微软Azure 329

10.1.1 Azure插件 329

10.1.2 轻松上云 331

10.1.3 Serverless开发 333

10.1.4 Web应用开发 338

10.1.5 数据库开发 340

10.1.6 更多Azure插件推荐 342

10.2 AWS 343

10.3 Google Cloud Platform 345

10.4 阿里云 348

10.4.1 Alibaba Cloud Toolkit 348

10.4.2 Aliyun Serverless 349

10.5 腾讯云 351

第11章 物联网开发 353

11.1 设备端开发 353

11.1.1 PlatformIO开发生态 353

11.1.2 了解PlatformIO IDE 355

11.1.3 使用PlatformIO IDE 355

11.2 设备上云 359

11.2.1 了解Azure IoT Hub 360

11.2.2 了解Azure IoT Hub插件 360

11.2.3 使用Azure IoT Hub插件进行物联网开发 360

11.3 设备模拟 364

11.4 边缘计算 366

11.4.1 了解Azure IoT Edge 366

11.4.2 Azure IoT Edge插件 367

11.5 物联网插件推荐 368

11.5.1 Espressif IDF 368

11.5.2 Arduino 368

11.5.3 Workbench 369

11.5.4 Cortex-Debug 369

11.5.5 Azure IoT Tools 369

第12章 远程开发 371

12.1 远程开发概览 371

12.2 远程开发插件 372

12.3 SSH 372

12.3.1 快速开始 373

12.3.2 系统要求 382

12.3.3 管理SSH远程机器 383

12.3.4 管理插件 384

12.3.5 端口转发 386

12.3.6 打开远程SSH主机的终端 387

12.3.7 远程SSH主机的设置 387

12.3.8 清理远程SSH主机上的Visual Studio Code服务器 388

12.4 容器 388

12.4.1 快速开始 388

12.4.2 系统要求 392

12.4.3 devcontainer.json文件 392

12.4.4 样例项目 394

12.4.5 直接打开Git项目 394

12.4.6 管理容器 394

12.4.7 管理插件 396

12.4.8 端口转发 398

12.4.9 打开开发容器的终端 398

12.4.10 开发容器的设置 398

12.5 WSL 399

12.5.1 快速开始 400

12.5.2 管理WSL 402

12.5.3 管理插件 402

12.5.4 打开WSL的终端 404

12.5.5 远程WSL的设置 404

第13章 Visual Studio family 405

13.1 Visual Studio、Visual Studio Code、Visual Studio Codespaces，你都分清楚了吗 405

13.2 Visual Studio Codespaces 406

13.2.1 概览 406

13.2.2 4种开发模式 408

13.2.3 使用Visual Studio Codespaces 409

13.2.4 自托管的环境 426

13.2.5 自建Web版Visual Studio Code 427

13.3 Visual Studio Live Share 429

13.3.1 概览 429

13.3.2 使用Visual Studio Live Share 429

13.3.3 Visual Studio Live Share插件推荐 436

13.4 Visual Studio IntelliCode 437

13.4.1 概览 437

13.4.2 使用Visual Studio IntelliCode 438

第14章 成为Visual Studio Code的贡献者 440

14.1 GitHub Issues 440

14.1.1 报告bug 440

14.1.2 功能请求 441

14.1.3 分享你的反馈与想法 441

14.1.4 翻译中文Issue 442

14.2 提问 442

14.3 讨论 443

14.4 GitHub Pull requests 443

14.5 插件 443

14.6 翻译 443

第15章 插件开发 445

15.1 如何打造一款优秀的Visual Studio Code插件 445

15.1.1 设计 445

15.1.2 实现 448

15.1.3 推广 449

15.1.4 维护 450

15.2 你的第一个Visual Studio Code插件 451

15.2.1 搭建开发环境 451

15.2.2 创建插件项目 451

15.2.3 运行插件 452

15.2.4 开发插件 452

15.2.5 调试插件 453

15.2.6 插件项目的文件结构 453

15.2.7 package.json插件清单文件 454

15.2.8 extension.ts插件入口文件 455

15.3 Visual Studio Code插件的扩展能力 456

15.3.1 Visual Studio Code插件的设计理念 456

15.3.2 通用功能的扩展能力 457

15.3.3 工作区用户界面的扩展能力 459

15.3.4 主题的扩展能力 460

15.3.5 编程语言的扩展能力 460

15.3.6 调试功能的扩展能力 461

15.4 插件开发面面观 461

15.4.1 插件样例 461

15.4.2 Command命令 464

15.4.3 树状视图 468

15.4.4 网页视图 472

15.4.5 集成终端 482

15.4.6 存储 483

15.4.7 主题 485

15.4.8 编程语言 490

15.4.9 更多常用的API 495

15.5 插件开发的生命周期 496

15.5.1 插件测试 496

15.5.2 插件发布 498

15.5.3 持续集成 500