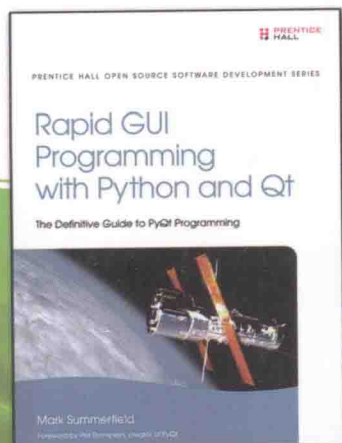


Python Qt GUI 快速编程 —— PyQt编程指南

Rapid GUI Programming with Python and Qt
The Definitive Guide to PyQt Programming



[英] Mark Summerfield 著

闫锋欣 黄琳雅 王军锋 等译

Python Qt GUI 快速编程

——PyQt 编程指南

Rapid GUI Programming with Python and Qt
The Definitive Guide to PyQt Programming

[英] Mark Summerfield 著

闫锋欣 黄琳雅 王军锋 等译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要讲述如何利用 Python 和 Qt 开发 GUI 应用程序的原理、方法和关键技术。本书共分四个部分：第一部分主要讲述 Python 基础知识，第二部分通过三个例子给出 PyQt GUI 应用程序的初步印象，第三部分深入讲述窗口部件布局、事件处理、窗口部件子类化、Qt 图形架构和 Qt 的模型/视图等内容，第四部分介绍国际化、网络化和多线程化等内容。

本书结构合理，内容翔实，适合于对 Python、Qt 和 PyQt 编程感兴趣的科教人员和广大的计算机编程爱好者阅读，也可作为相关机构的培训教材。

Authorized translation from the English language edition, entitled Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming, 9780132354189 by Mark Summerfield, published by Pearson Education, Inc., Copyright © 2008 Pearson Education Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2016.

本书中文简体字版专有出版权由 Pearson Education(培生教育出版集团)授予电子工业出版社。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书贴有 Pearson Education(培生教育出版集团)激光防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2015-1609

图书在版编目(CIP)数据

Python Qt GUI 快速编程: PyQt 编程指南/(英)马克·萨默菲尔德(Mark Summerfield)著;闫锋欣等译.

北京:电子工业出版社,2016.9

书名原文:Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming

ISBN 978-7-121-29806-6

I. ①P… II. ①马… ②闫… III. ①软件工具-程序设计-指南 IV. ①TP311.561-62

中国版本图书馆 CIP 数据核字(2016)第 203337 号

策划编辑:冯小贝

责任编辑:李秦华

印 刷:三河市华成印务有限公司

装 订:三河市华成印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:28.75 字数:811 千字

版 次:2016 年 9 月第 1 版

印 次:2016 年 9 月第 1 次印刷

定 价:79.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: fengxiaobei@phei.com.cn。

译者序

Python 是一种面向对象、解释型程序设计语言，由 Guido van Rossum 于 1989 年发明并于 1991 年发布第一个公开发行版。Python 简洁而清晰的语法，丰富而强大的库，使其能够将其其他语言编写代码模块(尤其是 C/C++)轻松联结在一起，从而在不影响程序性能的前提下，大大降低整个应用程序的开发成本和资源占用。Python 自 2004 年以来，已成为全球最受欢迎的程序设计语言之一，用户数呈线性激增，被 TIOBE 计算机编程语言排行榜评为 2010 年的年度语言。Python 语言在卡耐基·梅隆大学、麻省理工学院、清华大学等国内外高校和科研机构的用户众多，也进而推动了 Python 的快速发展。

Qt 是一个性能卓越的跨平台 C++ 图形用户界面应用程序开发框架。自 1991 年由奇趣科技(Trolltech)推出至今的 25 年间，深受业界赞誉，并先后于 2008 年、2012 年被诺基亚(Nokia)和大奇(Digia)收购，实现了由单一的桌面计算机和嵌入式应用领域到全 IT 行业的转变。2014 年 4 月，Qt 开发团队正式发布跨平台的集成开发环境 Qt Creator 3.1.0，实现了对 iOS 的完全支持，新增了 WinRT、Beautifier 等插件，集成了基于 Clang 的 C/C++ 代码模块，调整了对 Android 的支持，实现了 iOS、Android、WP 的全面支持。目前，Qt 的用户已经涵盖了全球众多知名厂商，如 Google、Adobe、IBM、华为、中国移动等，而诸如 Google Earth、AutoDesk Maya、Opera、KDE、Skype 这样的应用程序大家也都是耳熟能详的。

PyQt 是 Qt 与 Python 的成功融合，或者也可以认为 PyQt 是 Qt 库的 Python 版本。PyQt 初次发布于 1998 年，创始人是 Phil Thompson。PyQt 的版本包括支持 Qt 1 到 Qt 3 的 PyQt3，支持 Qt 4 的 PyQt4 和现今支持 Qt 5 的 PyQt5。不过，由于 Qt 开发团队已经明确宣布自 2015 年年底后不再支持 Qt 4，所以对于 PyQt 的新手来说，最好能够直接从 PyQt5 开始学习。

本书是迄今为止最受公众认可的 PyQt 编程学习用书之一。作者 Mark Summerfield 在 Qt 公司初创时期就任职于此，近年来也一直作为 Qt 和 Python 开发的知名代码贡献者，在两者中均拥有良好的经验。Mark 在撰写本书的过程中，就不断跟踪 Python 和 Qt 的实时发展动态，因而无论是书中的例子和风格，还是课后习题的设置，始终秉承传授编程思想和原理方法为主、解决和分析技术难点为辅的写作风格，因而他所撰写的多部作品都获得了有软件业界“奥斯卡”之称的“震撼奖”(Jolt Award)。本书与获奖作品《C++ GUI Qt 4 编程》一书的写作风格类似，案例设置通俗易懂，因而是学习 PyQt 不可多得的一本好书。

感谢电子工业出版社的冯小贝编辑。为了能够把握书中的关键技术和最新发展，译者不得不多方求证、字斟句酌，也形成了近似电影《疯狂动物城》中“闪电”先生的工作模式和工作效率，译稿一拖就是一年多的时间。不过，这一年多的时间中，我核对了书中的每一处链接，验证了它们的有效性；依据作者的勘误信息，订正并更新了本版图书中的相应错误内容，以确保文字的正确性和可读性。因此，无论使用的 Qt 4 还是 Qt 5，无论是 Python 2.x 还是最新的 Python 3.5.x，绝大部分的内容都可以运行无误。

还要感谢参与本书翻译和审校工作的各位战友们，感谢你们的鼓励和支持。大家的工作分工是：西安交通大学的黄琳雅翻译了第 10 章、第 12 章和第 13 章，北京工商大学的张君施翻

译了第 8 章、第 9 章、第 15 章和第 17 章，西南科技大学的王军锋翻译了第 1 章，西北农林科技大学的张雷锋翻译了附录 B 和附录 C，浙江大学的薛一翻译了第 11 章，广东技术师范学院的刘溪翻译了第 14 章、第 16 章、第 18 章和 19 章，西北农林科技大学的闫锋欣翻译了本书第 2 章至第 7 章以及附录 A、前言和致谢等剩余部分。我们还邀请了北京交通大学的王海波、武汉深之度科技有限公司的丁江锋、山东济南初创公司的王翔凯和徐景亮作为外部审稿人，他们的细致和耐心，为我们的工作增色不少。

感谢农业部现代农业装备重点实验室开放课题(项目编号:201603002)和中央高校基本科研业务费项目(项目编号:Z109021423)为译者提供了宽松且安心舒适的工作环境。

书中所用到的示例程序的源代码可从原书站点 www.qtrac.eu(英文)下载,也可直接从站点 www.qtcn.org/pyqtbook(中文)下载。有关本书的讨论和勘误信息,也会及时在 www.qtcn.org 网站公布,并在此向网站负责人 XChinux 表示感谢。

由于书中概念和术语数目繁多,加之译者水平所限,译稿中难免存在曲解或误解作者原意的地方,恳请读者谅解。

闫锋欣

2016 年 4 月 24 日

序

作为 PyQt 的创始人，非常高兴能够看到本书终于编写完成了。尽管是本书的技术审稿人之一，还是很高兴地要承认，自己的确从书中学到了不少东西。

PyQt 文档中涵盖了 PyQt 中各个层次的全部 API 类。本书则用来讲述如何使用这些类，以及如何将这些类组合起来创建对话框、主窗口和各类应用程序（它们不仅看起来美观漂亮而且功用良好，没有任何不良缺陷），使得大家乐于使用这一编程语言。

我最喜欢这本书的地方在于，即使是用来说明简单知识点的样例都显得极不平凡，并能够以其正确的方式给出极富潜力的应用方法。这些不寻常的方法，将可馈赠那些打算把 PyQt 用于开发更大维度、更高品质应用程序的读者们。

我从事 PyQt 的故事可以追溯到 20 世纪 90 年代。当时我用过一段时间的 Tcl/Tk，但觉得 Tk 应用程序看起来并不美观，尤其是当在第一版 KDE 中看到它运行时所做的那些事时，就打算要使用 Python，于是我认为，是把语言的变化和 GUI 库的变化结合起来的时候了。

起初，使用了一些基于 SWIG 写成的封装类，但随后就说服了自己，应该自己做一套更适合的封装工具^①。工作就从创建 SIP 开始，并在 1998 年 11 月发布了支持 Qt 1.41 的 PyQt 0.1 版。开发工作自此开始不断正规起来，不仅与新发布的 Qt 保持一致，而且还在不断拓展 PyQt 的应用范围，例如，各类额外的支持工具和改良后的文档。到 2000 年时，PyQt 2.0 就已经可以在 Linux 和 Windows 上同时支持 Qt 2.2 了。对于 Qt 3 的支持始于 2001 年，而 2002 年就支持 Mac OS X 了。PyQt4 系列始于 2006 年 6 月的 PyQt 4.0，它可以支持 Qt 4。

我的基本目标是要让 Python 和 Qt 能够协同工作，这种工作方式要让 Python 编程人员觉得非常自然，同时允许他们能够以 C++ 来做那些像在 Python 中一样可以想做的任何事。达到这一点的关键在于 SIP 的开发工作。这就给予了一个特殊的代码生成器，让我能够完全控制并确保 Python 和 Qt 是始终如一的。

开发和维护 PyQt 的必要步骤目前都已建立完毕。大多数工作现在都可自动完成，这就意味着，与 Trolltech 公司的 Qt 新版本保持同步已不再像从前那样显得是个问题，并可相信，PyQt 在未来几年中会依旧向前发展^②。

如今，非常欣慰于能够目睹 PyQt 社区在过去的数年中不断成长。如果此书能够把您引入 PyQt 的天地，那么欢迎您！

—Phil Thompson

温伯恩，多赛特，英国

① SWIG 是一种简化脚本语言与 C/C++ 接口的开发工具。简而言之，SWIG 是一个通过包装和编译 C 语言程序来达到与脚本语言通信目的的工具——译者注。

② Trolltech 是挪威的一家公司，Qt 最先源自该公司，先后被诺基亚和大奇 (Digia) 公司收购。国内一般将其称为奇趣科技公司——译者注。

前 言

本书主要讲述如何利用 Python 程序语言和 Qt 应用程序开发框架来开发 GUI 应用程序。仅需要的一点必备知识是，要能够使用一些面向对象编程语言来编程，诸如 C++、C#、Java，当然，也包括 Python 自己。在有关 Rich 文本的章节中，可能还会假定你了解了 HTML 和正则表达式的一些知识；而在数据库和多线程的那些章节中，也还会假定你已了解了相关话题的基本知识。至于 GUI 编程方面的知识就不需要了，因为书中会包含其相关的所有关键概念。

本书将对那些以专业编程为其工作的那些人大有帮助，无论是专业的软件开发人员，还是其他行业的编程人员，如科学家、工程师等，都需要通过编程来支撑自己的工作。本书也同样适用于那些大学生和研究生，他们在课题或者研究工作中上经常需要大量的计算要素。为帮助同学们理解书中讲授的内容，还特地提供了一些练习题（并给出了这些习题的解答思路）。

Python 有可能是世界上广泛应用的最易学习、最漂亮的脚本语言了，而 Qt 则有可能是开发 GUI 应用程序最好的库。Python 和 Qt 的结合，称为 PyQt，使得在所有支持它们的平台，如 Windows、Linux、Mac OS X 和类 UNIX 系统的各个现有版本上，开发应用程序并且不做任何改变地运行程序成为可能。无须编译得益于 Python 这一解释性脚本语言，而针对不同操作系统都无须更改源代码的好处则源于 Qt 的抽象方式，能够避开那些与平台相关的细节。我们要做的仅仅就是将写好的一个或者多个源代码文件复制到安装了 Python 和 PyQt 的目标机器上运行即可。

如果对 Python 一无所知，那么欢迎您！您将会探索一门易读易写的语言，其语法简洁而不神秘。Python 支持很多编程范式，但由于我们侧重于 GUI 编程，因而将会在除本书最前面的几章之外的各处均使用面向对象的编程方法。

Python 是一种非常富有表现力的语言，这就意味着，要完成具有相当功能的应用程序，用 Python 编写的代码要比使用其他诸如 C++ 或者 Java 等语言的代码少得多。这就使得通过文本来展示一些小而完整的样例成为可能，也使得 PyQt 成为能够快速并且简易开发 GUI 应用程序的一件理想工具，无论是用做原型设计还是用做最终的产品。

由于本书的重点是有关 GUI 编程的，第一部分会像其他 PyQt 文献一样也给出一个 Python 快速入门教程。在本书的第二部分、第三部分和第四部分，都与 PyQt 相关并会假定读者已经可以用 Python 编程，无论是前期的经验，还是从第一部分中阅读而来的。

在编程时，当有数种可能采用的可行方法时就会经常遇到决策点 (decision point)。参阅书籍和网上文献来识别出可用的那些类、方法和函数，还可以参考某些情况下所给出的示例，但那样的文献很少会给出一个上下文背景 (broader context)。本书则会给出必要的上下文背景，强调那些用于 GUI 编程的决策点，深入解析器优劣，以便让读者自行决断特定情况下的正确策略。例如，在创建对话框的时候，应该是使用模态 (modal) 对话框还是非模态 (modeless) 对话框 (请参阅第 5 章中有关这一主题的解释说明和推荐策略中的内容)。

PyQt 可用于编写各类 GUI 应用程序，从会计类应用程序到被科学家和工程师所使用的各种可视化工具。例如，在图 1 中，给出了一个示例 Eric4，这是一个使用 PyQt 编写的强大集成开发环境。编写一个仅有 10 行代码长短的 PyQt 应用程序，或者也有可能是编写一个拥有

1000 ~ 10 000 行 PyQt 代码的中型工程，都已司空见惯。某些商业公司利用从一个人到数十人不等的编程团队，已经构建出超过 100 000 行代码的 PyQt 应用程序。许多公司内部使用的工具就是用 PyQt 编写的，但由于这些工具通常是直接用于获利的，相关公司一般不会将自己使用 PyQt 的事情公之于众。PyQt 也会广泛用于开源世界，包括游戏、应用设施、可视化工具和各类集成开发环境 (IDE) 等都会用到它。

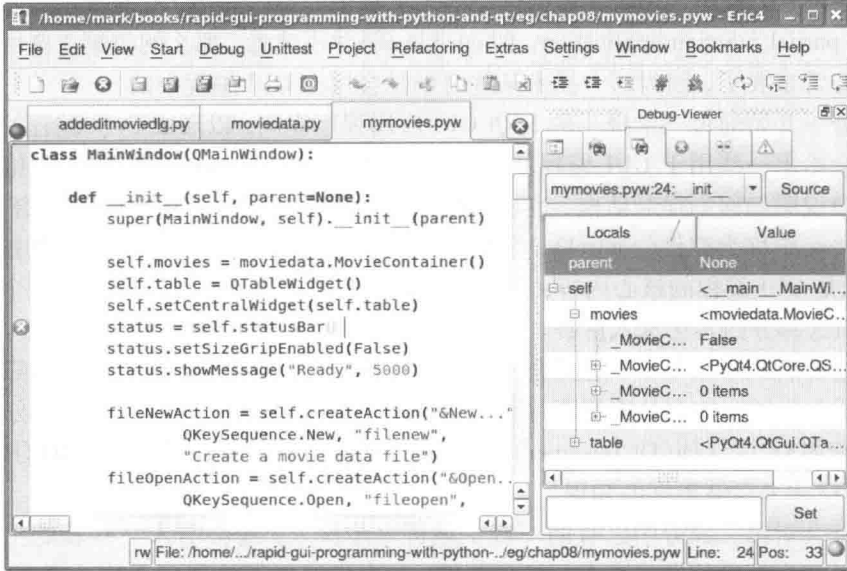


图 1 Eric4 集成开发环境，是一个 PyQt4 应用程序

本书重点放在 PyQt4 和用于 Qt 4 C++ 应用程序开发框架^①的 Python 绑定上。会以 10 个 Python 模块的形式给出 PyQt4，在这些模块中会大约含有 400 个类和 6000 个左右的方法和函数。所有样例程序均已使用 Python 2.5、Qt 4.2 和 PyQt4.2 在 Windows、Linux 和 Mac OS X 上进行了测试。此外，在 Windows 和 Linux 上，还使用 Qt 4.3 和 PyQt4.3 进行了测试。某些情况下，还有可能可以回溯到以前更早期的版本，但还是建议尽量使用最新版本的 Python、Qt 和 PyQt。

Python、PyQt 和 Qt 可免费用于非商业用途，但 Python 所采用的授权则不同于 PyQt 和 Qt 的授权。基于较为宽泛的授权条款就可以获得 Python，并可将其用于开发商用或者非商用应用程序。而 PyQt 和 Qt 则使用双重授权模式：这就从本质上使得它们只能用于开发非商用应用程序——相应需要使用一种可接受的开源授权，如 GPL (GNU General Public License, GNU 通用公共授权)；或者用于开发商业程序，此时，就必须分别购买商业版的 PyQt 授权和商业版的 Qt 授权。

本书的结构

本书共分四个部分。第一部分是快速转换课程，主要面向那些熟悉面向对象语言的非 Python 编程人员，尽管会有一些 PyQt 的内容。这是因为核心的 Python 语言非常简单并且相当

^① 对于较老的 Qt 3 库也有 Python 绑定，但在新的工程项目中没有理由再去使用该库，特别是自 Qt 4 以来，又提供了更多的功能和更为简单的使用方法。

小巧，这些章节会讲述一些 Python 的基础知识，以便能够为编写真正的 Python 应用程序做好准备。

如果认为自己可以通过阅读一些 Python 程序就能够了解 Python 语法的话，不妨直接跳过第一部分并转入第二部分前面有关 GUI 编程的那些章节。第二部分的开头章节会向前引用一些位于第一部分相关页码的内容，以便为读者选择该方法提供支撑依据。然而，即使是熟悉 Python 的那些读者，也还是建议能够阅读一下第 1 章中 QString 的相关内容。如果您对偏函数应用程序 (partial function application, 即 currying^①) 并不熟悉，那么阅读第 2 章中有关小节的相关内容就非常重要了，因为这一技术某些时候是会用在 GUI 编程中的。

第二部分一开始通过三个极小的 PyQt GUI 应用程序样例，以形成 PyQt 编程的一个初始印象。其中也会给出一些用于 GUI 编程基本概念的说明，包括 PyQt 的高级信号和槽的通信机制。第 5 章会说明如何创建对话框，也会说明如何创建和布局对话框中的各个窗口部件 [widget, 在 Windows 中称为控件 (control)，是用来构成用户界面的一些图形元素，如按钮、列表框等]。对话框是 GUI 编程的核心内容：多数 GUI 应用程序都有单一的主窗口、数个或者数十个对话框，因而这部分内容会深入讲解。

对话框之后就到了第 6 章，这里会涉及一些主窗口，包括菜单、工具栏、停靠窗口和键盘的快捷键，还会包括一些有关应用程序设置的加载和保存方面的内容。第二部分的剩余章节会讲述如何使用 Qt 设计师 (Qt Designer) 和 Qt 的可视化设计工具，还会讲述如何用二进制、文本和 XML 的格式来存储数据的知识。

第三部分会对第二部分中已有的一些主题进一步深入，还会引入一些新的主题。第 9 章会讲述如何以相当烦琐的方式来布局窗口部件，并会讲述如何处理多文档。第 10 章会讲述一些低级事件处理器 (event handler)，还会讲述如何使用剪切板 (clipboard) 和拖拽、文本、HTML 及二进制数据。第 11 章给出了如何对已有的窗口部件进行修改和子类化，还会讲述如何用 scratch^② 创建一些全新的窗口部件，并可以完全控制它们的外观和行为。这一章还会给出一些图形基础的示例。第 12 章会讲述如何使用 Qt4.2 的图形视图架构，它尤其适合用来处理大量的独立图形对象。Qt 的 HTML Rich 文本引擎处理能力会在第 13 章中讲述。这一章还会包括一些打印到纸张和 PDF 文件的内容。

第三部分中与模型/视图 (model/view) 相关的章节有两章：第 14 章会介绍该主题，并会展示一些如何使用 Qt 内置视图的方法，用以创建自定义数据模型和自定义代理 (delegate)；第 15 章会讲述如何使用模型/视图架构来执行数据库编程。

第四部分继续模型/视图这一主题，会在第 16 章中给出三个不同的模型/视图的用法。第 17 章的 17.1 节会介绍一些用于提供在线帮助的技术，17.2 节会讲述如何来国际化一个应用程序，包括如何来使用 Qt 的翻译工具创建翻译文件等。Python 标准库提供了一些用于网络化和多线程化的自有类，但在第四部分的最后两章，还是会给出一些网络化和多线程化时使用了 PyQt 的类。

附录 A 会说明哪些地方可以获得 Python、PyQt 和 Qt，并说明如何在 Windows、Mac OS X

① 在计算机科学领域，偏函数应用程序是指通过固定原函数的一部分参数生成新函数的过程，新函数的参数数目会少于原函数；currying 是把接受多个参数的函数变换成接受一个单一参数 (最初函数的第一个参数) 的函数，返回接受余下的参数并返回结果的新函数。currying 和偏函数应用程序有些关系，实为两个并不完全相同的概念——译者注。

② Scratch 是一种新的编程语言，据说是面向中小学生的，可认为是一种基于 Squeak 并用其实现的可视化程序设计语言——译者注。

和 Linux 上安装它们。如果安装了 PyQt 并试着通过阅读一些示例代码做些练习,就会发现 PyQt 要更易于学习。附录 B 给出了一些截图和一些选取的 PyQt 窗口部件的简单介绍,这对于那些 GUI 编程新手们会很有帮助。附录 C 展示了一些 PyQt 的关键类的层次示意图,这在要知道 PyQt 提供了哪些类以及它们和谁相关的时候很有用。

如果之前从来就没有用过 Python,那么就应当依次阅读第 1 章到第 6 章。如果已经了解 Python,那么至少要阅读有关字符串方面的规定,然后就可以跳过第 2 章了(除了 2.1 节之外,因为这些内容应当熟练掌握)。要相信你对 lambda 和偏函数应用程序(lambda and partial function application)并不反感,这些内容都在第 2 章讨论。第 3 章很有可能也是可以略过的。然后就可以依次阅读第 4 章至第 6 章了。

一旦涉猎了前六章,那么就有了 Python 的基本知识和 PyQt 的基础。

如果打算知道如何使用可视化工具而不是纯粹的手工编码来创建对话框,那么第 7 章就很有用了,这样就可以节省出大量的时间。对于文件的处理,则至少要阅读过第 8 章的前三节。如果打算对文件进行读写,还需要阅读第 8 章的 8.4 节;而如果打算使用 XML 文件,那么就需要阅读 8.5 节。

对于第三部分,至少需要阅读第 10 章的 10.1 节,即事件处理,以及第 11 章的全部内容。第 12 章和第 13 章的 13.1 节会认为你已经阅读过 PyQt 事件处理的内容了,并同时还会认为也阅读了第 11 章。在这一部分中,第 9 章和第 14 章可以单独阅读,但第 15 章会认为你已经阅读了第 14 章。

对于第四部分,在阅读第 16 章之前,假设已阅读了第 14 章和第 15 章。而其他章节可单独阅读。

如果发现了文字或者示例中的错误,或者有其他建议,请发邮件到 mark@qtrac.eu, 并在标题栏中备注“PyQt book”字样。在本书的主页会列出一些勘误信息,其中也可以下载各个样例和练习题的答案,主页的网址是: <http://www.qtrac.eu/pyqtbook.html>。

如果打算参与 PyQt 社区,就非常值得加入邮件列表了。可以在 <http://www.riverbankcomputing.com/mailman/listinfo/pyqt> 找到该邮件列表存档的链接,以便可以看到邮件列表的形式,还可以找到加入邮件列表的表单。Python 还有一些邮件列表和其他的活跃社区。要查看它们,可以浏览 <http://www.python.org/community>。

致谢

要感谢很多人,那么就先从参与本书的这些人开始吧。

Jasmin Blanchette 是 Trolltech 公司的一名高级软件开发人员,也是一名 Qt 专家,还是一名好编辑和特立独行的作家。我曾与他一起合著过两本 C++/Qt 的书籍。Jasmin 提出了大量的建议和批评,从而极大地提高了本书的质量。

David Boddie 是 Trolltech 公司的一名文档经理,也是一名为 PyQt 做出过很多贡献的 PyQt 活跃开源开发人员。他的投入很大程度上为我保证了所需的各类东西,并会给出一些合理的次序。

Richard Chamberlain 是 Jadu 有限公司的合伙人和首席技术官(CTO),该公司主营内容管理。他的反馈和远见确保了本书能够在尽可能宽泛的领域内获得认可。他还帮助精炼和改良了那些用于示例和练习题中的代码。

Trenton Schulz 是一名 Trolltech 公司的开发人员,也是我之前几本书中不可多得的审阅人员。对于本书,他带来了大量的 Python 和 Qt 知识,对本书手稿给出了相当多的建议。此外,

他还指出了我所遗漏的一些错误。

Phil Thompson 是 PyQt 的创始人和维护人。他从一开始就很支持本书，甚至还将我们针对本书讨论而来的一些结果加入到 PyQt 的新特性和改进之中。他对改进本书提出了无数的建议，并纠正了书中的很多错误和误解。

尤其感谢 Samuel Rolland，让我能够轻松自在地使用他的 Mac 笔记本，安装 PyQt，测试示例和截图。

还要感谢 Python 的创始人 Guido van Rossum，同样要感谢为 Python 发展做出了巨大贡献的 Python 社区，特别是其各个类库，是那么有用，那么好用。

同样要感谢 Trolltech 公司，开发和维护着 Qt，特别感谢 Trolltech 公司中那些过去和现在的开发人员，他们中的很多人我都曾经与之快乐共事，并始终坚信 Qt 会是目前已有的最好跨平台 GUI 开发框架。

特别感谢 Lout 排版语言的创始人 Jeff Kingston。我在自己的全部书籍和大部分著述工程中都使用了 Lout。在这些年中，Jeff 依据用户的建议对 Lout 做出了很多改进并增加了许多新特性，其中就包括了许多我曾自问的东西。也要感谢 James Cloos，他是 DejaVu Sans Mono 字体（派生自 Jim Lyles 的 Vera 字体）及其压缩版的创始人，从该字体派生了本书所使用的 mono-spaced 字体。

从出版社来说，主编 Karen Gettman 是从最最开始就非常支持本书的。特别感谢我的编辑 Debra Williams Cauley 的支持，使得整个出版流程尽可能顺畅自如。还要感谢 Lara Wysong 那么好地管理了出版工作，感谢校对人员 Audrey Doyle 做了那么多的细致工作。

最后，我想感谢我的妻子 Andrea。她的爱、忠诚和支持总是给予我力量和希望。

关于作者

Mark Summerfield 毕业于威尔士斯旺西大学(the University of Wales Swansea)，他曾获得计算机科学专业的一等荣誉学位。在进入企业之前，曾进行过为期一年的研究生学习阶段。在加入 Trolltech 公司前的多年间，担任过多个公司的软件工程师。他有大约三年的时间担任 Trolltech 公司的文档管理产品经理，在这段时期内，创立并编写了 Trolltech 公司的技术杂志，*Qt Quarterly*，并与他人共同编写了 *C++ GUI Programming with Qt 3* 一书，之后又编写了 *C++ GUI Programming with Qt 4* 一书。Mark 创立了 Qttrac 有限公司(www.qttrac.eu)，在那里他作为独立撰稿人、编辑和技术顾问(特别是有关 C++、Qt、Python 和 PyQt)。

目 录

第一部分 Python 编程

第 1 章 数据类型和数据结构	2
1.1 执行 Python 代码	3
1.2 变量和对象	4
1.3 数字和字符串	7
1.3.1 整数和长整型	8
1.3.2 浮点数和小数	9
1.3.3 字节字符串、Unicode 字符串和 QString	11
1.4 集合	18
1.4.1 元组	18
1.4.2 列表	20
1.4.3 字典	23
1.4.4 集	24
1.5 内置函数	25
小结	27
练习题	28
第 2 章 控制结构	30
2.1 条件分支	31
2.2 循环	33
列表解析和生成器	37
2.3 函数	37
2.3.1 生成器函数	40
2.3.2 关键字参数的使用	41
2.3.3 lambda 函数	43
2.3.4 动态函数的创建	43
2.3.5 偏函数应用程序	44
2.4 异常处理	46
小结	51
练习题	51
第 3 章 类和模块	54
3.1 实例的创建	55
3.2 方法和特殊方法	57
3.2.1 静态数据、静态方法和装饰器	61

3.2.2 例: Length 类	62
3.2.3 集合类	67
3.2.4 例: OrderedDict 类	67
3.3 继承和多态	72
3.4 模块和多文件应用程序	76
小结	78
练习题	79

第二部分 GUI 编程基础

第4章 GUI 编程简介	82
4.1 25 行的弹出式闹钟	83
4.2 30 行的表达式求值程序	86
4.3 70 行的货币转换程序	90
4.4 信号和槽	94
小结	101
练习题	102
第5章 对话框	103
5.1 简易对话框	104
5.2 标准对话框	109
OK/Cancel 型模态对话框	110
5.3 智能对话框	115
5.3.1 非模态应用/关闭型对话框	115
5.3.2 非模态的实时对话框	119
小结	121
练习题	122
第6章 主窗口	123
6.1 主窗口的创建	124
6.1.1 动作和按键顺序	127
6.1.2 资源文件	129
6.1.3 创建和使用动作	130
6.1.4 恢复和保存主窗口的状态	135
6.2 用户动作的处理	142
6.2.1 文件动作的处理	142
6.2.2 编辑动作的处理	147
6.2.3 帮助动作的处理	149
小结	150
练习题	151
第7章 使用 Qt 设计师	152
7.1 用户界面的设计	154

7.2	对话框的实现	161
7.3	对话框的测试	165
	小结	166
	练习题	167
第8章	数据处理和自定义文件格式	169
8.1	主窗口的职责	170
8.2	数据容器的职责	175
8.3	二进制文件的保存和加载	179
8.3.1	用 QDataStream 读写	179
8.3.2	使用 pickle 模块读写	183
8.4	文本文件的保存和加载	185
8.4.1	使用 QTextStream 读写	186
8.4.2	使用 codecs 模块读写	190
8.5	XML 文件的保存和加载	191
8.5.1	XML 的写	191
8.5.2	用 PyQt 的 DOM 类来读取和解析 XML	193
8.5.3	用 PyQt 的 SAX 类读取和解析 XML	195
	小结	198
	练习题	199

第三部分 中级 GUI 编程

第9章	布局和多文档	202
9.1	布局策略	203
9.2	Tab 标签页窗口部件和堆叠窗口部件	204
9.3	窗口切分条	211
9.4	单文档界面(SDI)	213
9.5	多文档界面(MDI)	219
	小结	227
	练习题	228
第10章	事件、剪贴板和拖放	229
10.1	事件处理机制	229
10.2	重新实现事件处理程序	230
10.3	使用剪贴板	235
10.4	拖放	236
	小结	240
	练习题	241
第11章	自定义窗口部件	242
11.1	使用窗口部件样式表	242
11.2	创建复合窗口部件	245

11.3	子类化内置窗口部件	246
11.4	子类化 QWidget	247
11.4.1	例: 分数滑块	249
11.4.2	例: 流体混合窗口部件	255
	小结	260
	练习题	261
第 12 章	基于项的图形	262
12.1	图形项的自定义和交互	263
12.2	动画和复杂形状	277
	小结	285
	练习题	286
第 13 章	Rich 文本和打印	287
13.1	Rich 文本的编辑	288
13.1.1	使用 QSyntaxHighlighter	288
13.1.2	Rich 文本的行编辑	293
13.2	文档打印	300
13.2.1	图片的打印	302
13.2.2	使用 HTML 和 QTextDocument 打印文档	302
13.2.3	使用 QTextCursor 和 QTextDocument 打印文档	304
13.2.4	使用 QPainter 打印文档	307
	小结	310
	练习题	311
第 14 章	模型/视图编程	312
14.1	使用简便项窗口部件	313
14.2	创建自定义模型	320
14.2.1	实现视图逻辑	320
14.2.2	实现自定义模型	323
14.3	创建自定义委托	329
	小结	334
	练习题	335
第 15 章	数据库	336
15.1	连接数据库	336
15.2	执行 SQL 查询	337
15.3	使用数据库窗体视图	341
15.4	使用数据库表视图	345
	小结	356
	练习题	356

第四部分 高级 GUI 编程

第 16 章 高级模型/视图编程	360
16.1 自定义视图	360
16.2 泛型委托	366
16.3 树中表达表格数据	373
小结	383
练习题	383
第 17 章 在线帮助和国际化	385
17.1 在线帮助	385
17.2 国际化	387
小结	393
练习题	393
第 18 章 网络应用	394
18.1 创建 TCP 客户端	396
18.2 创建 TCP 服务器	400
小结	404
练习题	404
第 19 章 多线程	406
19.1 创建线程服务器	407
19.2 创建和管理次线程	412
19.3 实现次线程	418
小结	422
练习题	423
这并非结束	424
附录 A 安装	425
附录 B PyQt 的部分窗口部件	437
附录 C 部分 PyQt 类的层次	441

第一部分

Python 编程

- 第 1 章 数据类型和数据结构
- 第 2 章 控制结构
- 第 3 章 类和模块