

在编程时，小挫折可能与大难题一样令人痛苦。没人希望在费劲心思之后，只是做到弹出消息窗口或是快速写入数据库。因此，程序员都会喜欢那些能够快速处理这些问题，同时长远来看也很健壮的解决方案。

下面这6个Python库既可以快速解决眼前的棘手问题，同时也能够作为大型项目的基础。

是什么：**Pyglet**是一个纯Python语言编写的跨平台框架，用于开发多媒体和窗口特效应用。

为什么需要它：从头开发图形界面应用所需要的功能模块是十分繁琐的，**Pyglet**提供了大量现成的模块，省去了很多的时间：窗口函数，**OpenGL**特效，音频与视频回放，键鼠处理以及图片处理。（不过，**Pyglet**并没有提供类似按钮、工具栏或菜单等UI部件。）

所有上述模块都是在**Windows**、**OS X**或**Linux**平台下原生实现的，并不依赖外部二进制包；它是纯Python语言编写的。**Pyglet**通过**BSD**协议发布，可以用于任何商业和开源项目。

Peewee

是什么：**Peewee**是一个小型但是十分强大的库，支持通过**ORM**的方式访问数据库，原生支持**SQLite**、**MySQL**和**PostgreSQL**等数据库。

为什么需要它：任何一个需要经常使用外部数据的应用基本都会用到数据库，但是通过临时连接从数据库中读写数据会带来很多麻烦。

Peewee提供了一条访问数据库资源的安全、稳定的通道。对于Python程序员和数据库工程师来说，该库所提供的Python类使用起来将会得心应手。有了**Peewee**的支持，我们可以快速便捷地访问数据库，后续还可以扩展加入更多的选项，不需要重新设计。**Peewee**同时原生支持数据库事务（**transaction**），并有许多可选的额外模块，提供了从数据库连接池（**connection pooling**）到类似多对多（**many-to-many**）的高级**field**类型等功能。

Bottle

是什么：**Bottle**是一个小型的轻量网络开发框架，同时速度也很快。

为什么需要它：如果你只是想快速创建一个**Restful API**接口，或者只想用网络开发框架的做一个简单的应用，**Bottle**可以轻松地满足你的要求。它具备了你将需要的所有功能：路由、模板、访问请求与响应数据（**request and response data**）、支持多种网络服务器以及**WebSockets**等高级功能。

创建一个应用所需的工作极少，而且**Bottle**在设计时就考虑了可扩展性，如果需要更多高级功能，随时就可以接入。

Invoke

是什么：简单来说，**Invoke**让你通过一个Python库便捷地执行系统管理任务。

为什么需要它：谁不想要一个“可以运行shell命令、定义并归类执行任务的简洁、高级接口”呢？利用Python替代一般的shell脚本，并执行相应的任务，是完全合理的。**Invoke**提供了执行常见命令行任务并进行管理的解决方案。对于**Invoke**来说，每个管理任务就像是Python函数一样，可以在此基础上优雅地设计更为复杂的任务。

需要注意的是，**Invoke**当前仍是预览版；如果你想使用稳定的工具（即使是不再积极开发），可以考虑**Invoke**的前身——**Fabric**。

Splinter

是什么：**Splinter**是一个自动化测试网络应用的Python库。

为什么需要它：大家都知道，没有什么比自动化网络应用测试更无聊的事了。有了**Splinter**，就可以将打开浏览器、输入URL、填写表单、点击按钮等全部操作自动化。

特定的浏览器需要使用相应的驱动器（drivers），不过还好已经自带了**Chrome**和**Firefox**驱动器。另外，**Splinter**还可以通过**Selenium Remote**来远程控制其他机器上的浏览器。你甚至可以在目标浏览器中手动执行JavaScript代码。

如果你想知道某个浏览器在浏览指定网站时的具体情况，那么**Splinter**将是一个很有用的工具。如果了解不依赖浏览器与网站进行交互，可以查看**Twill**。（译者：**Twill**是一种脚本语言，支持用户通过命令行浏览网络。）

Arrow

是什么：**Arrow**这个库可以更好地处理Python中的日期和时间（data/time）。

为什么需要它：处理时区、日期转换、应对不同的日期格式以及其他日期相关的东西，足够让你头疼一天半的。如果使用Python自带标准库中的模块，那么估计你得头疼两天了。

改用**Arrow**库的话有四大好处，不管长期还是短期都是很有用的。第一，它可以完美替代Python中的datetime模块，这意味着你仍可以使用类似 `.now()` 库 `.utcnow()` 这些常见的函数调用形式。第二，它提供了满足转换时区等常见需求的方法。第三，它提供了“人性化”的日期/时间信息——也就是，它可以很轻松地告诉你某件事是在“一小时以前”发生的，或是“将在两小时后”发生。第四，它很容易地将日期/时间信息转换为当地时间。