

今年早些时候，Perl 就 23 岁了。这门语言已经由介于 Shell 脚本和 C 程序之间(Perl 1)的简易系统管理工具，转变成为一种强大的通用编程语言。他承接着丰富的代码资产(Perl 5)，并在对通用编程的不断反思中，续写又一个 25 年的辉煌(Perl 6)。

尽管如此，Perl 语言提供的强大特性还是未能得到现存大部分 Perl 5 程序的充分利用。你可以写出与 Perl 4 (或 Perl 3 或 Perl 2 或 Perl 1) 相仿的 Perl 5 程序，但用到整个 Perl 5 社区已经发明、发扬、发现的神奇特性的程序，会更短、更快、更强大，相比其它版本，也更易维护。

Modern Perl 仅是对老练而高效的 Perl 5 程序员的工作方式的一种粗略描述。他们使用惯用语。他们充分利用 CPAN。他们的程序带有明显的“Perl 的味道”，他们展现出对事物的良好品味、独到的匠心以及对 Perl 通透的理解。

你也可以学着成为这样的 Perl 程序员。

除非另作说明，每一处代码片段总是假设如下基本程序框架：

```
#!/usr/bin/perl

use Modern::Perl;

# 这里是示例代码
...
```

Modern::Perl 模块可以从 CPAN 获取。你可以自行安装它，或用如下代码代替：

```
use strict;
use warnings;
use 5.010;
```

其余用到诸如 ok(), like(), 及 is() (testing) 等测试函数的代码片段，该框架则为：

```
#!/usr/bin/perl

use Modern::Perl;
B<use Test::More;>

# 这里是示例代码
...
done_testing();
```

书中给出的例子假设你正使用 Perl 5.10 或更新版本，理想的话至少是 Perl 5.10.1。许多例子经适当修改就可以在 Perl 5 的早期版本上运行，但是早于 Perl 5.10.0 的版本会给你带来更多的困难。本书同样描述（但不要求使用）Perl 5.12 中新引入的功能。

你通常可以自行安装一份最新的 Perl。Windows 用户可以从 <http://www.strawberryperl.com/> 下载 Strawberry Perl。操作系统自带 Perl 5（以及 C 编译器还有其他一些开发工具）的用户，可以以安装 CPAN 模块 App::perlbrew 安装指南参见 [http://search.cpan.org/perl/doc?App::perlbrew](http://search.cpan.org/perl/doc/App::perlbrew) 作为开始。

perlbrew 允许你安装并管理多个版本的 Perl 5。默认地。它将它们安装在你的家目录下。你不仅可以在不影响系统自带 Perl 的情况下拥有多个版本的 Perl 5，而且不用劳烦系统管理员给你特定权限，就可以把你喜欢的任何模块安装到这些目录中。

Perl 5 和 Perl 6

你应该学习 Perl 5 还是 Perl

6？它们有着同样的哲学和语法以及库和社区，它们有各自合适的位置。如果：

- * 你有现成的 Perl 5 代码要维护
- * 你需要利用 CPAN 模块

- * 你的部署策略要求严格的稳定性

那么学 Perl 5 吧。如果：

- * 你对频繁升级习以为常
- * 你能够承受试验新语法和新功能
- * 你需要的功能仅由 Perl 6 提供
- * 你可以为它的开发做出贡献（无论是补丁、缺陷报告、文档、赞助或其他资源）

那么学 Perl 6 吧。

一般说来，Perl 5

的开发会保持有关核心语言的部分。有好有坏，变化产生得相当缓慢。Perl 6 更具实验性质，它更注重找到最合理的设计而非保证旧的代码可以工作。所幸的是，你可以学习并使用两者（并且它们影响着互相进步）。

本书讨论 Perl 5。如要学习 Perl 6，请浏览 <http://perl6.org/>，试用 Rakudo (<http://www.rakudo.org/>)，并参考同样由 Onyx Neon Press 出版的 Using Perl 6 一书。