

Perl turns 23 years old later this year. The language has gone from a simple tool for system administration somewhere between shell scripting and C programming (Perl 1) to a powerful, general purpose language steeped in a rich heritage (Perl 5) and a consistent, coherent, rethinking of programming in general intended to last for another 25 years (Perl 6).

Even so, most Perl 5 programs in the world take far too little advantage of the language. You *can* write Perl 5 programs as if they were Perl 4 programs (or Perl 3 or 2 or 1), but programs written to take advantage of everything amazing the worldwide Perl 5 community has invented, polished, and discovered are shorter, faster, more powerful, and easier to maintain than their alternatives.

*Modern Perl* is a loose description of how experienced and effective Perl 5 programmers work. They use language idioms. They take advantage of the CPAN. They're recognizably Perl-ish, and they show good taste and craftsmanship and a full understanding of Perl.

You can learn this too.

## Running Modern Perl

You don't have to install a new version of Perl to take advantage of most of this book, but the examples given assume that you're using Perl 5.10 or newer. Unless otherwise mentioned, code snippets always assume the basic skeleton of a program:

```
#!/usr/bin/perl

use Modern::Perl;

# example code here
...
```

Other code snippets use testing functions such as `ok()`, `like()`, and `is()` (*testing*). That skeleton program is:

```
#!/usr/bin/perl

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

# example code here
...
```

If you don't already have a modern version of Perl installed, you can install it yourself, using:

```
* perlbrew
* Strawberry Perl
* local::lib
```