

Perl 5 并不完美，至少其默认行为是如此。核心部分存在一些改变这种状况的选择。更多的则来源于 CPAN。有经验的 Perl 开发者对理想状态下 Perl 5 应有的行为有着各自的理解，并且他们常常高效地使用他们各自对 Perl 5 的配置。

新手也许并不知道 Perl 如何帮助他们更好地编写程序。一手把核心模块将使你变得更加富有成效。

缺少的默认特性

Perl 5 的设计过程从 1993 年持续至 1994 年，尝试为语言的新方向做好准备，却无法预言其未来。Perl 5 增加了不少优秀的新特性，但它仍和 Perl 1 到 Perl 4 这过去的七年保持兼容。十六年之后，编写整洁、可维护、强大和简明的 Perl 5 代码最佳途径已经和 Perl 5.000 时代相差甚大。默认行为有时会对你有所妨碍，幸运的是，还有更佳的行为可用。

strict 编译命令

strict 编译命令 (*pragmas*) 允许你禁用 (或重新启用) 诸多功能强大但可能导致意外误用的语言结构。

strict 执行三种功能：禁用符号引用、要求变量声明并且禁用对未声明裸字 (*barewords*) 的使用。虽然偶尔使用符号引用对于进行符号表修改和导出 (除对辅助模块的使用之外，比如 *Moose*) 来说是必须的，将变量用作变量名称使隐晦的超距作用 (*action at a distance*) 错误有机可乘——或者更糟糕，使未经严格验证的用户输入恶意修改对内数据成为可能。

要求变量声明有助于防止变量名的打字错误并鼓励将词法变量限定于正确作用域。如果所有词法变量都在合适的作用域内由 *my* 或 *our* 声明，则很容易就能看出其预期的作用域。

strict 在词法范围内有效，基于其用法的编译期作用域。你可以用 *no strict* 禁用某些 **strict** 的特性 (当然，在可能的最小范围内)。详情请参见 *perldoc strict*。

warnings 编译命令

warnings 编译命令 (*handling_warnings*) 控制着 Perl 5 中诸多警告的汇报，例如试图字符串化 *undef* 值或对值使用错误的操作符类型。它同时对使用不推荐的特性做出警告。

最有用的警告解释了 Perl 难以理解你的意思，只能猜测正确的解释方法。即便 Perl 通常能猜对，由你来消除歧义总是能保证程序的正常运行。

warnings 变量命令对其用法的编译期作用域有词法效果。你可以用 *no warnings* 禁用部分或全部警告 (当然，在可能的最小范围内)。详情请参见 *perldoc perllexwarn* 和 *perldoc warnings*。

结合 *use warnings* 和 *use diagnostics*，Perl 5 将对程序中出现的每一处警告显示扩展诊断信息。这些扩展诊断信息来自于 *perldoc perldiag*。此行为在 Perl 学习阶段非常有用，但是在部署产品代码时则非如此，因为它可能产生非常详细的错误输出。

IO::Handle

Perl 5.6.0 增加了词法文件句柄。在此之前，文件句柄都是包全局变量。这样做偶尔会显得散乱和令人迷惑。现在你可以这样写：

```
open my $fh, '>', $file or die "Can't write to '$file': $!\n";
```

.....*\$fh* 中的词法文件句柄简单易用。词法文件句柄的实现创建了对象；*\$fh* 是 *IO::Handle* 的实例。不幸的是，即使 *\$fh* 是一个对象，你无法在其上调用任何方法，因为 *IO::Handle* 类尚未加载。

举例来说，冲洗相关文件句柄的缓冲区偶尔也会令人痛苦。它也可以这样简单：

```
$fh->flush();
```

.....但仅在你程序的某处包含 *use IO::Handle* 时才可以这样做。解决方法就是在你程序里加上这一行，使得词法文件句柄——原本就是对象——在行为上也和对象一致。

autodie 编译命令

Perl 5 的默认错误检查有些吝啬。例如，如果你不仔细对每一个 *open()* 调用的返回值进行检查，你可能会尝试从一个已经关闭的文件句柄读取——或者更糟，向其写入而导致数据丢失。*autodie* 编译命令改变此默认行为。如果你编写：

```
use autodie;
```

```
open my $fh, '>', $file;
```

.....一个不成功的 *open()* 调用将通过 *Perl 5* 的常规异常机制抛出异常。考虑到应对失败系统调用的最佳手段是抛出异常，该编译命令可以去掉大量刻板的代码并让你因为知道没有忘记检查返回值而感到轻松。

该编译命令于 *Perl 5.10.1* 进入 *Perl 5* 核心部分。详情请参见 *perldoc autodie*。