

# 1 NAME

perl - Practical Extraction and Report Language

## 2 语法

```
perl [ -sTuU ] [ -hv ] [ -V[:configvar] ] [ -cw ] [ -d[:debugger] ] [ -D[number/list] ] [ -pna ] [ -Fpattern ] [ -l[octal] ] [ -O[octal] ] [ -ldir ] [ -m[-]module ] [ -M[-]'module...' ] [ -P ] [ -S ] [ -x[dir] ] [ -i[extension] ] [ -e 'command' ] [ -- ] [ programfile ] [ argument ]...
```

如果你是一个 Perl 的初学者，那么你最好先阅读一下 perlintro, 那是一个为初学者专门准备的简单介绍，为你讲解一些 Perl 的基础知识，并且帮助你遨游于剩余的大量 Perl 文档。

为了便于阅读，Perl 手册被分成了一下几个部分：

### 概述

perl	Perl 概述(本文档)
perlintro	为初学者准备的 Perl 简介
perltoc	Perl 文档目录
activeperl	ActivePerl 概述

### 教程

perlreftut	Perl 引用简介
perldsc	Perl 数据结构介绍
perllool	Perl 高级数据结构: 数组的数组(二维数组)
perlrequick	Perl 正则表达式快速入门
perlretut	Perl 正则表达式教程
perlboot	Perl OO 入门教程
perltoot	Perl OO 教程(一)
perltooc	Perl OO 教程(二)
perlbot	Perl OO 高级技巧与实例
perlstyle	Perl 程序风格指南
perlcheat	Perl 夹带
perltrap	Perl 陷阱
perldebtut	Perl 调试教程
perlfaq	Perl 常见问题(FAQ)
perlfaq1	普通问题
perlfaq2	开始使用和学习 Perl 时碰到的问题
perlfaq3	编程工具
perlfaq4	数据操作
perlfaq5	文件和格式
perlfaq6	正则表达式
perlfaq7	Perl 语言问题
perlfaq8	系统交互
perlfaq9	网络编程

## 参考手册

perlsyn	Perl 语法
perldata	Perl 数据结构
perlop	Perl 操作符和优先级
perlsub	Perl 子程序
perlfunc	Perl 内建函数
perlopentut	Perl open() 教程
perlpacktut	Perl pack() 和 unpack() 教程
perlpod	Perl POD 文档
perlpodspec	Perl POD 文档格式说明
perlrun	Perl 运行和选项
perldiag	Perl 诊断信息
perllexwarn	Perl 警告
perldebug	Perl 调试
perlvar	Perl 预定义变量
perlre	Perl 正则表达式高级教程
perlrefref	Perl 正则表达式快速参考
perlref	Perl 引用高级教程
perlform	Perl 格式
perlobj	Perl 对象
perltie	Perl 绑定(将对象隐藏在普通变量背后)
perldbmfilter	Perl DBM 过滤器
perlipc	Perl 进程间通信
perlfork	Perl fork() 说明
perlnumber	Perl 数字符号
perlthrtut	Perl 线程教程
perlothrtut	以前的 Perl 线程教程
perlport	Perl 移植手册
perllocale	Perl 区域支持
perluniintro	Perl Unicode 介绍
perlunicode	Perl Unicode 支持
perlebcdic	考虑将 Perl 运行在 EBCDIC 编码的平台上
perlsec	Perl 安全
perlmod	Perl 模块编程: 入门教程
perlmodlib	Perl 模块编程: 如何书写并使用一个新模块
perlmodstyle	Perl 模块编程: 编写模块的规范
perlmodinstall	Perl 模块编程: 如何安装一个 CPAN 上发布的模块
perlnewmod	Perl 模块编程: 准备发布自己的模块
perlutil	Perl 打包、发布工具
perlcompile	Perl 编译器套件介绍
perlfilter	Perl source filters

## Perl 内部实现和 C 语言接口

perlembed	嵌入 Perl 到你的 C/C++ 程序中去
perldebbugs	Perl debugging guts and tips
perlxsut	Perl XS 教程

perlxs	Perl XS 编程接口(API)
perlclib	C 标准库的 Perl 替代
perlguts	使用 Perl 内部函数扩展 Perl
perlcall	从 C 中调用 Perl 时的约定
perlapi	Perl API 一览
perlintern	Perl 内部函数
perliol	C API for Perl's implementation of IO in Layers
perlapiio	Perl 内部 IO 抽象接口
perlhack	Perl 黑客指南

## 杂项

perlbook	Perl 书籍讯息
perltodo	Perl 未来展望
perldoc	Look up Perl documentation in Pod format
perlhist	Perl 修改记录
perldelta	相对于前一个版本的修改
perl586delta	5.8.6 版所做的修改
perl585delta	5.8.5 版所做的修改
perl584delta	5.8.5 版所做的修改
perl583delta	5.8.3 版所做的修改
perl582delta	5.8.2 版所做的修改
perl581delta	5.8.1 版所做的修改
perl58delta	5.8.0 版所做的修改
perl573delta	5.7.3 版所做的修改
perl572delta	5.7.2 版所做的修改
perl571delta	5.7.1 版所做的修改
perl570delta	5.7.0 版所做的修改
perl561delta	5.6.1 版所做的修改
perl56delta	5.6 版所做的修改
perl5005delta	5.005 版所做的修改
perl5004delta	5.004 版所做的修改
activeperl-release	ActivePerl 发布日志
activeperl-changes	ActivePerl 修订历史
perlartistic	Perl Artistic License
perlgpl	GNU GPL 许可协议内容

## 语言相关

perlcn	Perl 简体中文简介(采用 EUC-CN 编码)
perljp	Perl 日文简介(采用 EUC-JP 编码)
perlko	Perl 朝鲜文简介(采用 EUC-KR 编码)
perltw	Perl 繁体中文简介(采用 Big5 编码)

## 平台相关

perlaix	AIX 平台说明
perlamiga	AmigaOS 平台说明
perlapollo	Apollo DomainOS 平台说明
perlbeos	BeOS 平台说明

perlbs2000	POSIX-BC BS2000 平台说明
perlce	WinCE 平台说明
perlcygwin	Cygwin 平台说明
perldgux	DG/UX 平台说明
perldos	DOS 平台说明
perlepoc	EPOC 平台说明
perlfreesbd	FreeBSD 平台说明
perlhpux	HP-UX 平台说明
perlhurd	Hurd 平台说明
perlirix	Irix 平台说明
perlmachten	Power MachTen 平台说明
perlmacos	Mac OS (Classic) 平台说明
perlmacosx	Mac OS X 平台说明
perlmint	MiNT 平台说明
perlmpaix	MPE/iX 平台说明
perlnetware	NetWare 平台说明
perlopenbsd	OpenBSD 平台说明
perlos2	OS/2 平台说明
perlos390	OS/390 平台说明
perlos400	OS/400 平台说明
perlplan9	Plan 9 平台说明
perlqnx	QNX 平台说明
perlsolaris	Solaris 平台说明
perltru64	Tru64 平台说明
perluts	UTS 平台说明
perlvmesa	VM/ESA 平台说明
perlvms	VMS 平台说明
perlvos	Stratus VOS 平台说明
perlwin32	Windows 平台说明

如果你使用的是类 UNIX 系统，那么上述手册页默认情况下安装在 /usr/local/man/ 目录。

除此之外，还有大量的 Perl 模块的文档。这些附加文档默认被放置在 /usr/local/lib/perl5/man 目录或者是 Perl 库目录下的 man 子目录 这里边有一部分附加文档是 Perl 的标准发布包中自带的，其余则是自行安装的 第三方模块的文档。

译者注：如果你使用的是 windows 平台下的 ActiveState Perl，并且安装在了 C:\Perl，那么上述手册页的 html 版本都在 C:\Perl\html 目录，pod 版本则在 C:\Perl\lib\pod 目录。如果你是用 .msi 安装包形式安装的话，安装包就会在 [开始] -> [程序] -> [ActiveState ActivePerl 5.x] -> [Documentation] 这位置生成一个快捷方式，你点击它就可以浏览所有的 ActivePerl 文档。

在类 UNIX 环境下，通过正确设置相应的系统文件，或者修改 MANPATH 环境变量，你可以使用 man(1) 程序来打开手册页。

用以下命令可以知道你的 Perl 手册页被安装到了什么地方：

```
perl -V:man.dir
```

如果返回结果中包含有一个常用的部分，比如是 /usr/local/man/man1 和 /usr/local/man/man3，那么你只需要添加 (/usr/local/man) 到你的 man(1) 的配置文件或者 MANPATH 环境变量中。

如果返回结果不包括这些常用目录，那你只好将每一行都添加到 man(1) 的配置文件或者 MANPATH 环境变量了。

此外，你还可以用 Perl 自带的 perldoc 脚本来浏览 Perl 手册页或者模块文档。(译者注：在 windows 平台下，因为没有 man 命令，所以要想在命令行下浏览 Perl 文档，那么这是唯一的方式。)

如果你的程序不能正确工作而你又不知道该去看哪个手册页，那么请试一下 `-w` 命令行选项，它通常会给你一个相对准确的信息以指出到底是哪儿出了麻烦。

### 3 说明

Perl 可以轻易的实现扫描任意文本文件、从中提取有用的信息、然后按照要求的格式打印出报表。Perl 也是一种优秀的系统管理工具。

Perl 语言被设计成是“实用的”(易于使用、高效、功能丰富)，而不是“华丽的”(简单、优美、精致)。『译者注：译文无法体现原文的意思，原文为 "The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal)."』

据作者来看，Perl 结合了 C、sed、awk、sh 等一些工具的许多优秀特性。这样熟悉以上工具的人可以只花费较少的代价就可以使用 Perl。(语言历史学家们可能还记得 csh, Pascal, 甚至还有 BASIC-PLUS.)

以下是 Perl 的一些特点：1, Perl 的表达式语法和 C 的表达式语法非常相似。2, 和大多数 UNIX 工具不同，Perl 本身不会限制你的数据大小，只要你拥有足够的内存。3, Perl 可以一口气读取整个文件的内容并且保存到一个普通的字符串变量中。4, 递归嵌套层数没有限制。5, 使用“哈希表(hash)”(有些资料也称作“关联数组(associative arrays)”)来表示表格。在必要的时候哈希表会自动增长额外的空间以免妨碍到性能。6, 使用“模式匹配”来快速扫描大量数据。7, Perl 也可以和二进制数据打交道，通过某些技术，可以使一个 DBM 文件象一个哈希表一样操作。8, 出于 setuid 方式运行的 Perl 脚本要比同样的 C 程序安全，Perl 拥有一个贯穿数据流的污染检测系统，这可以防止大多数令人生厌的安全漏洞。

如果你碰到了一个任务需要用 sed、awk 或者 sh 来完成，结果却发现无法做到或者是想运行起来更快一些，并且你又不想写 C 程序来处理，那么 Perl 就是你最好的选择。Perl 允许你将现有的 sed 脚本或者 awk 脚本转换为 Perl 脚本。

早在 1993 年(参见 perlhist)的时候，Perl 5 几乎被完全重写了一次，经历了这次版本更新以后，Perl 增加了一下几个重要特性：

- 模块化和可重用的编程思想，以及无数的模块。

在 perlmod、perlmodlib 和 perlmodinstall 中有介绍。

- 嵌入和扩展

在 perlembed, perlxtut, perlxs, perlcalls, perlguys, 和 xsubpp 中有介绍。

- 变量绑定(包含许多关联的 DBM 工具)

在 perltie 和 AnyDBM\_File 中介绍。

- 子程序可以被重载、自动加载、指定原型

在 perlsub 中介绍。

- 任意嵌套的数据结构和匿名子程序

在 perlreftut, perlref, perldsc, 和 perllol 中介绍。

- 面向对象编程

在 perlobj, perlboot, perltoot, perltooc, 和 perlbot 中介绍。

- 支持“轻量级进程”(线程)

在 perlthrtut 和 threads 中介绍。

- 支持 Unicode, 国际化, 区域化  
在 perluniintro, perllocale 和 Locale::Maketext 中介绍。
- 词法作用域  
在 perlsub 中介绍。
- 增强了正则表达式功能  
在 perlre 中介绍, 增加的示例参见 perlop。
- 增强了调试器和交互式环境, 可以集成外部编辑器。  
在 perldebtut, perldebug 中 perldebguts 介绍。
- POSIX 1003.1 兼容库  
参见 POSIX。

好了, 有了以上这些宣传, 我想肯定已经足够了。

## 4 可用性

在大多数操作系统上 Perl 都是可用的。实际上所有类 UNIX 平台都是可用的。详情参见 "Supported Platforms" Supported Platforms in perlport, 那里有一个清单。

## 5 运行环境

参见 perlrun.

## 6 作者

Larry Wall <larry@wall.org>, 以及其他许多普通人

如果你认为你的 Perl 成功案例能够帮助那些需要在他们的项目中使用 Perl 的人, 或者你想简单地表达一下你对于 Larry 以及其他 Perl 开发人员的谢意, 那么请写信给 perl-thanks@perl.org。

## 7 文件

"@INC"                    定位 Perl 模块搜索路径

## 8 参见

a2p    awk 脚本到 perl 脚本的翻译器  
s2p    sed 脚本到 perl 脚本的翻译器

<a href="http://www.perl.org/">http://www.perl.org/</a>	Perl 主页
<a href="http://www.perl.com/">http://www.perl.com/</a>	Perl 文章(O'Reilly 维护)
<a href="http://www.cpan.org/">http://www.cpan.org/</a>	CPAN: 完整的 Perl 档案网络
<a href="http://www.pm.org/">http://www.pm.org/</a>	Perl Mongers

## 9 诊断

使用 `use warnings` 语句(或者 `-w` 命令行选项)可以得到很多有用的诊断信息。

查看 `perldiag` 可以得到所有的 Perl 诊断信息的解释。使用 `use diagnostics` 语句可以自动地将错误信息转换成更加详细的形式。

编译时产生的错误信息可以告诉你错误发生的行号, 以及具体出错的位置。(译者注: 因为 Perl 总是在错误发生之后才会发现错误, 因此实际的错误位置 往往要比提示的靠前一些)

如果 Perl 脚本是通过命令行参数 `-e` 传递过去的, 那么每一个 `-e` 参数 将被当作一行来计算行号。

`Setuid` 脚本可以添加约束, 从而产生类似于 “Insecure dependency” (不安全的依赖) 之类的消息。参见 `perlsec`。

我们是不是应该提倡在每个程序中都使用 `-w` 开关呢?

## 10 BUGS

`-w` 命令行选项不是强制的。

Perl 的有些操作是和你的机器相关的, 比如 `type casting` (类型转换)、`atof()` 操作、还有使用 `sprintf()` 进行浮点数的输出, 等等。

如果你的标准输入输出库需要在每次读或者写操作之后移动文件指针, 那么 Perl 中会 同样如此, 但是 `sysread()` 和 `syswrite()` 函数中不会这样。

尽管没有一种内建数据类型的尺寸会受到除了物理内存大小之外的其它任何限制, 但是 仍然有一些东西在制约着你: 变量名的长度不能超过 251 个字母; 另外, 诊断时显示 出来的行号在 Perl 内部是使用短整型存储的, 因此最大不能超过 65535 (超过这个数 之后将发生环绕)。

你可以将你发现的 bug 汇报到 `perlbug@perl.org`。不过最好附上一份完整的配置信息, `perl` 源代码树中的 `myconfig` 可以帮你做这件事, 或者用 `perl -V` 也行。如果你成功地编译了 Perl, 那么你可以在 `utils/` 目录下找到 `perlbug` 脚本, 它可以帮你邮寄一份 bug 报表。

Perl 事实上是一个 “夭折的垃圾列表器”, 但是千万不要告诉任何人我这么说过。^\_^

## 11 注意事项

Perl 的格言是 “条条大路通罗马” (There's more than one way to do it.), 具体有多少作为 练习留给读者去猜测好了。

程序员的三种主要美德是: 懒惰、急躁和傲慢。至于为什么这么说请参考骆驼书。

## 12 TRANSLATORS

王兴华 "flw" <flw@cpan.org>