

Perl & MySQL Web编程

杨 焱 徐泽平 等 编著
高 波 审校

人民邮电出版社
www.pptph.com.cn

Perl & MySQL Web 编程

杨 焱 徐泽平 等编著

高 波 审 校

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

Perl & MySQL Web 编程/杨焱, 徐泽平编著.—北京:

人民邮电出版社, 2001.7

ISBN 7-115-09368-7

I.P... II.①杨...②徐... III. ①PERL 语言—程序设计②关系数据库—数据库管理系统, MySQL IV.TP312

中国版本图书馆 CIP 数据核字 (2001) 第 034891 号

内 容 提 要

Perl 是 Internet 上相当流行的编程语言, 有着强大的字符处理能力和丰富的函数库。MySQL 作为中型的免费网络数据库, 由于其具有强大的功能, 良好的安全性、伸缩性及鲁棒性, 已成为广受欢迎的 Web 数据库。

本书分为三篇, 第一篇讲解 Perl 语言, 第二篇讲解 MySQL 网络数据库的管理与应用, 第三篇结合大量典型实例讲解如何基于 MySQL 用 Perl 编写实用的 Web 程序。书后附录给出了 Perl 和 MySQL 函数的详细参考, 供读者查阅。

本书条理清楚、内容丰富, 实例剖析深入浅出, 对于使用 Perl 基于 MySQL 进行 Web 编程的读者具有很好的参考价值。

Perl&MySQL Web 编程

◆ 编 著 杨 焱 徐泽平 等
审 校 高 波
责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线:010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16
印张:31.5
字数:766 千字 2001 年 7 月第 1 版
印数:1—5 000 册 2001 年 7 月北京第 1 次印刷

ISBN 7-115-09368-7/TP·2259

定价:49.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

前 言

在金融、制造、遗传学、军事等人类知道的几乎所有应用领域，都可以看到 Perl 的踪迹。在 Internet 和 WWW 应用中，Perl 就用得更多了。使用 Perl，用户仅用少量的代码就能够创建出非常出色的程序。使用少量的 Perl，结合其他应用程序、语言和技术，也可以很快就能创建非常有用的程序，所以学会 Perl 的使用技巧无疑是一种非常值得的智力投资。相信本书对 Perl 的较为详尽的介绍会使读者尽快掌握 Perl 语言。

MySQL 是完全网络化的跨平台关系型数据库系统，又是具有客户机/服务器体系结构的分布式数据库管理系统。MySQL 具有功能强大、使用简便、管理方便、运行速度快、安全可靠性等优点，用户可利用许多语言编写访问 MySQL 数据库的程序。MySQL 可以利用 SQL（结构化查询语言），也可以利用支持 ODBC（开放式数据库连接）的应用程序。因为 MySQL 功能强大又具有如此多的优势，所以在本书专门用一篇来系统地介绍了 MySQL 的基础知识和概念，这些内容对具有一定关系数据库知识，并希望在网络上建立分布式数据库或建立基于因特网的应用系统的读者会有很大帮助。

本书除了介绍 Perl、MySQL 之外，还对利用 Perl 使用 MySQL 数据库进行 Web 编程做了尽可能多的讲述。

本书共分 3 篇、14 章。第一篇是专门介绍 Perl 语言的，本篇分为 4 章，第一章介绍了 Perl 的历史、优点、用途以及如何编写一个 Perl 程序。本章的目的是希望读者能够从一开始就对 Perl 有一个总体的认识。第二章介绍了如何建立 Perl 的工作环境，该章描述了 Perl 的获得和在一些常用的环境下如何安装 Perl，这对于一个初学者来说是非常必要的。那些对 Perl 比较熟悉的人可以跳过该章的学习。第三章对 Perl 语言的语法做了较为详尽的介绍，由于篇幅有限，只对一些最常用的知识点做了讲解，对于那些想了解关于 Perl 的更多知识的读者请参阅专门针对 Perl 编程的书籍。第四章实际是面向对象编程的一部分，该章详细讲述了 Perl 中的包和模块的很多知识，读者能够从中学到 Perl 编程的一些特别的技巧。

第二篇整篇都是对 MySQL 数据库系统的讲解，用 3 章的篇幅对 MySQL 数据库进行了初步的描述。第五章是 MySQL 数据库系统的入门知识。第六章分别就安装及基本使用、一些基本语法和 MySQL 函数做了较为全面的介绍。通过本章的学习，读者可以应付一般的 MySQL 编程。第七章讲述了如何管理和维护 MySQL 数据库系统。

第三篇是本书的重点，本篇的主要目的是使读者通过阅读本篇能够胜任几乎所有的 Perl 及 MySQL 的 Web 编程工作，这是读者需要认真学习重点篇章。本篇共分为 7 章，分别就建立 Web 应用编程环境、CGI 编程的知识和 Perl 编程实例进行了详尽的讲解。第八章讲述了如何建立 Web 应用编程环境以及如何让 Perl 使用 MySQL 数据库。第九章对 CGI 程序设计做了详尽的描述，深谙 CGI 编程的读者可以跳过该章。第十章介绍了如何使用 Perl 编写 CGI 程序，其中有一些知识在前面也讲解过，之所以在这里再介绍，目的是希望读者从另一个角度去认识 Perl。Perl 和 MySQL 数据库的联系方法在第十一章中做了细致的描述，这些在前面的章节中也有所涉及。后面的 3 章都是 Perl 及 MySQL 编程实例讲解。第十二章介绍了一个大的数据库检索系统，读者通过该章的学习后编程经验会有很大的提高。第十三章讲了六

个编程实例，这些实例有最常用的邮件列表、计数器，也有现在最热门的电子商务，这些实例会给读者不少启示。第十四章是一些较高级的编程实例，那些做大型的通用软件或通信编程的读者可以研究这一章。

Perl 和 MySQL 是在不断更新的，写此书时尽量反映最新版本的内容。由于水平有限，一些地方可能写得不妥，希望读者能尽可能发现它们，共同对本书进行完善和改进。

编 者

2001 年 6 月

目 录

第一篇 Perl 语言

第一章 Perl 语言概述	1
1.1 Perl 的历史	1
1.2 Perl 的优点	1
1.3 Perl 的用途	2
1.4 基本概念	2
1.5 Perl 预览	3
1.5.1 “Hey!My reader” 程序	3
1.5.2 提出问题并记录结果的程序	3
1.5.3 条件语句示例程序	4
1.5.4 猜密码程序	5
1.5.5 一个人不止一个密码时的猜密码程序	5
1.5.6 不同的人对应不同的密码时的猜密码程序	7
1.5.7 几种密码输入格式的处理程序	8
1.5.8 对每个人都像 Randal 一样处理的程序	9
1.5.9 更模块化的程序	11
1.5.10 把密码表挪到单个文件里边的程序	13
1.5.11 确保尽可能安全的程序	15
1.5.12 当密码出错时给出警告的程序	16
1.5.13 在当前目录下有多个密码文件的处理程序	17
1.5.14 列出密码清单的程序	18
1.5.15 使旧的密码表更明显的程序	19
1.5.16 维护一个上次用户猜测的数据库的程序	20
1.5.17 最后的程序	21
第二章 建立 Perl 工作环境	25
2.1 获取 Perl 源代码	25
2.1.1 Unix 版本	26
2.1.2 MS-DOS 版本	26
2.1.3 Perl 的 NetWare 版本	26
2.1.4 Perl 的 Windows 95 和 Windows NT 版本	26
2.1.5 Perl 的 OS/2 版本	26

2.1.6 Perl 的 Linux 版本	27
2.2 Perl 的安装	27
2.2.1 Unix 版本的安装	27
2.2.2 Win32 版本的安装	28
2.2.3 OS/2 版本的安装	28
2.2.4 Linux 版本的安装	28
第三章 Perl 语言速成	29
3.1 数据类型	29
3.1.1 Scalars (标量)	29
3.1.2 Arrays (数组)	31
3.1.3 Hashes (哈希表)	34
3.1.4 handles (句柄)	37
3.2 运算符及表达式	40
3.3 Perl 的控制语句	45
3.3.1 语句块	45
3.3.2 if...elseif...else/unless 语句	45
3.3.3 for 语句	47
3.3.4 foreach 语句	47
3.3.5 while/until 语句	48
3.3.6 跳出控制结构的语句	50
3.4 Perl 的函数	51
3.4.1 用户自定义函数	51
3.4.2 Perl 的内部函数	54
3.5 模式匹配	89
3.5.1 规则表达式及模式简介	89
3.5.2 模式匹配操作符	89
3.5.3 规则表达式语法	90
3.5.4 转义序列	91
3.5.5 字符类	92
3.5.6 锚点	92
3.5.7 限量符	93
3.5.8 模式匹配变量	93
3.5.9 扩展规则表达式	94
3.5.10 split 和 join 函数	94
3.6 面向对象设计	95
3.6.1 对象及类的简介	95
3.6.2 Perl 的对象	96
3.6.3 对象只是一个被引用的事物	96
3.6.4 类只是一个包	98
3.6.5 方法只是一个子程序	98

3.6.6	方法引用	99
3.6.7	管理实例数据	101
3.6.8	管理类数据	102
3.6.9	把类用作结构	104
3.6.10	克隆对象	105
3.6.11	写一个可继承的类	106
3.6.12	毁坏器	107
3.6.13	自动调用方法	108
3.6.14	垃圾收集	109
3.7	Perl 如何访问数据库	111
第四章	包、库和模块	113
4.1	引言	113
4.1.1	模块	114
4.1.2	进口/出口规则	115
4.1.3	别的种类的库文件	115
4.1.4	不要重复编写代码	116
4.2	模块接口的定义	116
4.3	捕获在 require 或 use 中的错误	118
4.4	延迟到运行时才用	119
4.5	使模块独占变量	120
4.6	确定调用者的包	122
4.7	自动模块清理	124
4.8	使用自己的模块目录	125
4.9	发布模块如何准备	126
4.10	使用 SelfLoader 加速模块调入	127
4.11	使用 AutoLoader 加速模块调入	128
4.12	覆盖内部函数	129
4.13	像内部函数那样报告错误和警告信息	130
4.14	间接引用包	131
4.15	使用 h2ph 转换 C 语言的#include 文件	132
4.16	使用 h2xs 建立一个 C 语言代码模块	134
4.17	使用 pod 为模块嵌入文档	137
4.18	建立和安装一个 CPAN 模块	138
4.19	模块模板示例	140
4.20	取得已安装的模块的版本号和描述信息的程序实例	141

第二篇 MySQL 数据库管理系统

第五章	MySQL 数据库概述	146
5.1	MySQL 的历史	146

5.2	MySQL 的主要特点	146
5.3	MySQL 的稳定性如何	148
5.4	2000 年问题	149
第六章	MySQL 的安装及基本使用	151
6.1	安装 MySQL	151
6.1.1	怎样获得 MySQL	151
6.1.2	MySQL 支持的操作系统	153
6.1.3	确定使用哪一个版本	153
6.1.4	安装布局	154
6.1.5	安装 MySQL 二进制发布	155
6.1.6	安装 MySQL 源代码发布	159
6.1.7	如何解决使用 Perl 的 DBI/DBD 接口出现的问题	162
6.2	MySQL 数据库语言	163
6.2.1	基本数据类型	163
6.2.2	数据库、表、索引、列和别名的名称规范	165
6.2.3	列类型	166
6.3	MySQL 数据库函数	178
6.3.1	分组函数	178
6.3.2	一般的算术操作	178
6.3.3	位函数	179
6.3.4	逻辑运算函数	179
6.3.5	比较操作符	180
6.3.6	字符串比较函数	183
6.3.7	强制操作符	184
6.3.8	流控制函数	184
6.3.9	数学函数	185
6.3.10	字符串函数	189
6.3.11	日期和时间函数	194
6.3.12	其他函数	200
6.3.13	用于 GROUP BY 子句的函数	202
6.4	常用 SQL 语句的语法	203
第七章	数据库系统的管理与维护	220
7.1	MySQL 管理介绍	220
7.1.1	管理职责概述	220
7.1.2	常规管理	221
7.1.3	安全性	222
7.1.4	数据库修复和维护	222
7.2	MySQL 存取权限系统	222
7.2.1	权限系统做什么	222
7.2.2	MySQL 用户名和口令	223

7.2.3	与 MySQL 服务器连接	223
7.2.4	使用用户的口令安全	224
7.2.5	MySQL 提供的权限	224
7.2.6	权限系统工作原理	226
7.2.7	存取控制, 阶段 1: 连接证实	228
7.2.8	存取控制, 阶段 2: 请求证实	231
7.2.9	权限更改何时生效	233
7.2.10	建立初始的 MySQL 权限	233
7.2.11	向 MySQL 增加新用户权限	234
7.2.12	怎样设置口令	237
7.2.13	Access denied 错误的原因	237
7.2.14	怎样使 MySQL 安全以对抗解密高手	240
7.3	维护 MySQL 安装	241
7.3.1	使用 myisamchk 进行表的维护和崩溃恢复	241
7.3.2	建立一个数据库表维护规范	244
7.3.3	获得关于一个表的信息	244
7.3.4	使用 myisamchk 进行崩溃恢复	250
7.3.5	日志文件维护	252

第三篇 Perl 及 MySQL Web 编程

第八章	建立 Web 应用环境	254
8.1	引言	254
8.2	建立 Linux 下 Web 开发环境	254
8.2.1	网络操作系统采用 Linux	255
8.2.2	Web 服务器采用 Apache	255
8.2.3	数据库采用 MySQL	255
8.2.4	脚本语言采用 Perl	255
8.2.5	客户端的软件	255
8.2.6	开发人员软件	256
8.2.7	Apache 和 Perl 模块的安装	256
8.3	建立 Windows 98 下 Web 开发环境	257
8.3.1	准备好需要的文件	257
8.3.2	软件安装	257
8.3.3	软件测试	259
8.4	Perl 与 MySQL 的数据库开发环境	260
8.4.1	Perl 5.0	260
8.4.2	Perl 数据库接口	263
8.5	Perl 与 MySQL 连接时应注意的问题	266
8.5.1	使用 PerlDBI/DBD 接口的问题	266

8.5.2 系统特定的问题	267
第九章 CGI 程序设计	276
9.1 引言	276
9.1.1 CGI 的提出	276
9.1.2 CGI 的工作原理	276
9.1.3 CGI 与其他 WWW 技术的关系	277
9.2 CGI 编程简介	277
9.2.1 CGI 的调用方式	278
9.2.2 环境变量	278
9.2.3 一个 Perl 编写的 CGI 例子	279
9.3 CGI 程序的结构	280
9.3.1 Get 和 Post	280
9.3.2 初始化	281
9.3.3 处理	281
9.3.4 输出结果	281
9.3.5 终止	282
9.3.6 设计 CGI 程序的几点建议	282
9.4 CGI 程序环境	285
9.4.1 标准环境变量	285
9.4.2 平台独立性	287
9.4.3 服务器独立性	287
9.4.4 CGI 库	288
9.5 CGI 的局限	291
第十章 使用 Perl 编写 CGI 程序	293
10.1 引言	293
10.1.1 模块简介	293
10.1.2 Perl 中的类	293
10.1.3 方法	296
10.2 Perl 中的包和模块	298
10.2.1 require 函数	298
10.2.2 包	300
10.2.3 模块	302
10.3 CGI.pm	304
10.3.1 CPAN	304
10.3.2 CGI.pm	305
10.4 CGI 方法	311
10.4.1 保存表单状态	311
10.4.2 创建 HTTP 头部	313
10.4.3 生成 HTML 文档	314
10.4.4 创建表单	320

第十一章 使用 Perl 来操作 MySQL 数据库	330
11.1 引言	330
11.2 DBI-通用数据库接口模块	332
11.2.1 概论	332
11.2.2 DBI 类	335
11.2.3 DBI 数据库对象	340
11.2.4 DBI 语句对象	347
11.3 MySQL 数据库驱动模块	351
11.3.1 概论	351
11.3.2 MySQL 数据库句柄的方法	353
11.3.3 MySQL 语句句柄对象	354
11.3.4 用 Perl 操作 MySQL 数据库的一个例子	355
第十二章 数据库检索系统 Genquery	357
12.1 引言	357
12.1.1 系统功能	357
12.1.2 系统调用方法	358
12.2 Genquery 的公共函数	358
12.2.1 打开数据库连接	358
12.2.2 打开游标	359
12.2.3 提取数据	359
12.2.4 执行语句	359
12.2.5 关闭游标	359
12.2.6 关闭数据库连接	359
12.2.7 生成 Insert 语句	360
12.2.8 生成 Update 语句	361
12.2.9 生成 Where 语句	361
12.2.10 生成 HTML 表格	362
12.2.11 生成打印输出	364
12.2.12 生成表单	364
12.2.13 其他函数	367
12.3 表单输出	368
12.3.1 查询表单	368
12.3.2 显示表单	369
12.3.3 添加表单	369
12.3.4 更新表单	371
12.3.5 删除表单	372
第十三章 Perl 编程实例	374
13.1 引言	374
13.2 表格处理器	374
13.3 动态创建图像	377

13.3.1	创建图像	377
13.3.2	颜色控制	378
13.3.3	画图命令	379
13.3.4	动态生成图像实例	380
13.4	页面计数器	382
13.4.1	记录(log)文件	383
13.4.2	创建自己的计数器	384
13.5	用户管理	387
13.5.1	用户注册	387
13.5.2	用户验证	390
13.5.3	用户删除	391
13.6	邮件列表	392
13.6.1	加入邮件列表	392
13.6.2	删除邮件列表	393
13.6.3	发送邮件	394
13.6.4	几个常用函数	395
13.7	构筑电子商务数据库系统	397
13.7.1	电子商务应用的三种类型	397
13.7.2	电子商务应用	397
13.7.3	电子商务系统实例	398
第十四章	Perl 高级编程实例	409
14.1	Socket 编程	409
14.1.1	SMTP 协议编程实例	410
14.1.2	POP3 协议编程实例	413
14.2	ODBC 编程	416
14.2.1	ODBC 编程简介	416
14.2.2	Perl 通过 ODBC 开发数据库应用实例	418
14.2.3	查询及返回结果 (se.pl)	418
14.3	网页文件上传	419
14.4	进程间通信	422
14.4.1	信号	422
14.4.2	命名管道	424
14.4.3	Socket	426
附录 A	Perl 函数库及其说明	429
附录 B	MySQL 函数库及其说明	464

第一篇 Perl 语言

第一章 Perl 语言概述

1.1 Perl 的历史

Perl 是“实用提取报告语言 (Practical Extraction and Report Language)”的缩写，有时也被称为“精悍的电子垃圾列表器 (Pathologically Eclectic Rubbish Lister)”。这两种名称实际上都有一定的道理，且都是由 Larry Wall 题的名。Larry Wall 是 Perl 的编制者和总设计师，也是 Perl 的维护者。当他需要为问题报告系统生成类似于 USENET NEWS 的层次文件报告时，觉得特别耗费精力，所以就想法做一个通用的工具，这个工具不仅仅能解决现在遇到的问题，而且在其他类似的情况下也能用得着。这样最初版本的 Perl 就诞生了。

在最初版本的 Perl 实现之后，Larry 及时把它发布给 USENET 读者社区，在那里遍及世界的使用者给了他一些信息反馈，要求他做进一步的改进，其中好多建议都是 Larry 没有想到的，结果 Perl 与 Unix 操作系统大约以同样的速度发展，Perl 的结构、可移植性都变得越来越好，过去只是小不点的它现在有 1000 多页的文档（分布在许多不同的传媒上），有 600 页的参考手册，有一个成员达 200000 的 USENET NEWS 组，这只是初步的统计。

Perl 的 USENET NEWS（如 comp.lang.perl.misc）早在 20 世纪 80 年代晚期就已出现。最早关于 Perl 的书《Perl 编程》(Programming Perl) 在 1991 年出版。由于人们对网络的应用越来越感兴趣，并且该语言用于 CGI 编程比较容易，它的用户数和使用量迅猛增长。Larry 现在不再是 Perl 唯一的维护人员，但仍然是首席设计师。Perl 仍然在不断发展，Perl 的最初版本重点在系统管理和文本处理方面，后来的几个版本还能够处理规则表达式、信号 (signal) 和网络套接字 (Socket)。

1.2 Perl 的优点

Perl 借鉴了好多 Unix 工具如 sed 和 awk 的优点，而且做得比它们更好。本书中所说的 Perl 可以指语言，也可以指语言的解释器。Perl 确实能够做很多 sed、awk 和 Unix 命令脚本可以做的事情。Perl 对低级任务处理得特别好，Perl 5.0 更是如此。Perl 和 C 语言有大量的共同点，但 Perl 比 C 语言更简洁。Perl 能够自动无缝地处理内部数据类型、内存分配等问题。在 Perl 5.0 中增加或去掉一些模块非常容易。比如可以通过使用模块把喜欢的软件包加入到 Perl 中，也可以把网络套接字处理从内核及模块中移走。Perl 现在如此受欢迎，当然有它的独到之处。首先它是免费的，用户不花钱或花很少的钱就能使用 Perl 和一些 Perl 的工具。事

实上, Perl 除了可免费利用这一优点外, 还有更大的优点, 那就是它几乎可以在任何带有 C 语言编译器的 Unix 或类 Unix 的系统中运行得非常好。它带有一个被称作 “Configure” 的神奇配置脚本, 这个脚本能够窥视和探测系统目录, 寻找它需要的东西, 调整头文件, 定义相应的符号, 用户可辅助验证它找到的东西。其次, 可以用少量的代码实现强大的功能。Perl 代码公开, 因此可以像使用 awk 一样充分使用它。第三, 在数分钟内就能编制一个小程序并立即执行, 没有必要去建立 makefile 文件、编译程序或者执行任何别的进程。但是如果用像 C 语言那样的编译语言时, 就必须得执行一些进程。第四, Perl 的许多库函数、模块、例程代码和脚本在网上随处可见, 完全可以免费获得。最后, Perl 可以专门用于网络应用开发。因为 Perl 可以用于多种平台, 所以对于做好的 Perl 程序, 只要针对操作系统对 Perl 脚本作一些小修改即可移植到另一个平台上运行, 而基本组件、概念和实现功能保持不变 (当然平台必须支持用于实现程序的 Perl 语言版本)。

1.3 Perl 的用途

Perl 是专门为帮助程序员解决一些常见任务而设计的, 这些任务可能特别繁重, 对环境很敏感 (即可移植性不好), 也可能太古怪, 存活时间短, 错综复杂以致没法用 C 语言或一些 Unix 附带语言编写。

一旦熟悉了 Perl, 就会在可移植性问题上花费很少的时间, 而把大部分时间用在读 USENET NEWS 上。Perl 确实是一种用来节省时间的很好的工具, 它强大的构建能力使得用户可以毫不费力地创建一些很好的通用工具。Perl 易于移植, 又便于使用, 使得用户可以在下一步工作中使用现有的工具。

像任何一门语言一样, Perl 是 “只写不读” 的, 也就是说写完后就不会再去读它。但应该尽量不要这样做, 在编写程序时, 尽量按照一定规范, 这样既便于读, 也便于维护。

1.4 基本概念

一个命令脚本实际上是由若干命令构成的文本文件。通过设置执行位, 该文件就变成可执行文件, 然后该文件名可在命令行提示符状态下输入。类似地, 一个 Perl 程序就是一簇 Perl 语句和定义所构成的文件。可以设置执行位, 然后在命令行状态下敲入文件名。但文件必须指示它是一个 Perl 程序, 而不是命令脚本, 因而在大多数情况下应该把 `#!/usr/bin/perl` 作为文件的第一行。但如果 Perl 放在非标准位置, 或者系统不理解该行时, 除了刚才说的作法外还得做另外的工作, 即检查 Perl 安装程序。Perl 像 C 语言一样, 格式随意, 程序的各元素 (有意义的语言单位) 之间的空格、制表符、换行符和回车符数是随意的, 但两元素合在一起可能会被误解成另一种程序元素, 在这种情况下空格、制表符、换行符或回车符必须至少有一个。某些结构在特定位置需要特定的空白符如空格、制表符、换行符和回车符, 在讲到这些特殊结构时我们将会指明。可以认为, 程序元素之间的空白符可以是任意的。

一个 Perl 程序可以写在一行上, 但我们建议最好写得像 C 语言一样具有层次结构。一个 Perl 程序是由 Perl 语句构成的, 可以被看作一个大的例程, 而不像 C 语言有 main 例程的概念。Perl 的注释像命令行脚本的注释, 任何从 “#” 开始到该行结束的内容都被看作注释。Perl

没有像 C 语言那样的多行注释功能。

不同于命令行脚本，Perl 解释器在执行任何 Perl 程序时首先得把该程序完全分析并编译成内部表示格式。这意味着一旦程序开始执行，就无法看到任何语法错误，程序的空白符和注释都被剔除。编译确保了一旦程序启动后 Perl 能够迅速执行。编译确实费时间，但程序的运行时间通过编译将会锐减。

Perl 既是编译器又是解释器。说它是编译器是因为在第一条语句执行以前，必须对整个程序阅读分析；说它是解释器是因为不需要生成中间代码占用磁盘空间。在某种程度上说，无论在哪一方面，Perl 都是最好的。因为无论是在两种引用之间的目标代码被缓存起来，还是被翻译成本地机器码，都是很好的。

1.5 Perl 预览

在进入 Perl 的全面学习前，我们先对 Perl 做一个大致的了解。本节通过一些小的应用程序来体现 Perl 的一些不同特征，使你可以快速了解 Perl 语言。

1.5.1 “Hey!My reader” 程序

我们来看一个小程序，这是一个很简单的程序：

```
#!/usr/bin/perl -w
print ("Hey!My reader");
```

第一行是 Perl 注释行，指明本程序是一个 Perl 程序，注释是从“#”号开始到该行结束的部分（在许多解释编程语言中都这样使用）。不像程序中其他的注释，第一行的注释是专用的，Perl 会在该行中查找可选项参数。如在本例中，使用了一个非常重要的转换器——“-w”转换器。这时，它要求 Perl 对潜在的危险建构产生附加报警信息。应该总使用“-w”模式编写程序。

第二行是整个程序的执行部分。这里使用了 print 函数，它是一个内部函数，只带有一个字符串参数。在字符串中，字符组合“\n”代表换行符，print 语句以分号“;”结束。像 C 语言一样，Perl 中所有的语句都以分号结束。当执行该程序时，内核利用 Perl 解释器分析整个程序，然后执行编译的结果。第一个也是唯一的一个操作是执行 print 函数，把函数的参数传给标准的输出设备。当程序执行完毕，Perl 就退出进程，并返回给父进程 shell 一个退出码。注意：在 Perl 程序中 print 和其他的一些函数有时在调用时使用括号，有时则不使用。这是因为在 Perl 中，内部函数可以用括号也可以不用括号，用户完全可以凭自己的判断来确定是否使用括号以提高程序的清晰程度。

1.5.2 提出问题并记录结果的程序

我们来编一个更复杂一点的程序，让程序询问用户的姓名。首先这需要一个位置来保存姓名，另外还得提问姓名并做出响应。保存姓名的那个位置就是标量变量。在本程序中，使用标量变量 \$name 来保存姓名。（有关标量变量的内容，如变量保存什么，如何保存等，会在以后的章节中讨论。）现在假定要在标量变量中保存一个单精度实数或字符串。程序要提问

姓名，那就需要一种提示方式并使用一种接收输入的方法。前面的程序已经表明可以利用 `print` 函数提示。要在终端接收一行输入可以使用结构 `<STDIN>`，该结构可以捕获一行输入，然后把输入赋值给变量 `$name`。下面是实现代码：

```
print "What's your name? ";
$name = <STDIN>;
```

`$name` 的值以换行符结束，可以使用 `chomp` 函数除掉换行符，`chomp` 函数把标量变量作为唯一的参数，从该变量的字符串值中除掉尾部的换行符（如果存在）并把新的值保存在该标量变量中，具体用法如下：

```
chomp ($name);
```

现在需要做的是程序收到字符串后的响应，这里仅需要输出“Hi.”，后面紧跟着输出变量 `$name` 的值就可以了。显然用 `print` 语句就能做到。注意：可以把嵌入变量的字符串作为 `print` 的参数。执行时它会自动换成相应的变量值输出：

```
print "Hi,$name!\n";
```

如果想输出一个“\$”符而不是把它作为一个标量变量的引用，那只需要在“\$”符前加一个反斜杠“\”即可。

把上面这几条语句合在一块，得到如下程序：

```
#!/usr/bin/perl -w
print "What's your name? ";
$name = <STDIN>;
chomp ($name);
print "Hi, $name!\n";
```

该程序就完全实现了要求的功能。执行后屏幕输出如下：

```
What's your name?
```

这时光标在问号后闪烁，要求输入姓名。敲入 Yang Yi 并回车，将看到

```
Hi,Yang Yi!
```

最后在屏幕上看到的内容如下：

```
What's your name? Yang Yi
```

```
Hi,Yang Yi!
```

1.5.3 条件语句示例程序

以上面的程序为基础，现在，想要求程序能给 Yang Yi 一个特别的问候，而对其他人给出一般性问候。为此，要把输入的名字与字符串“Yang Yi”比较，如果相同，作特殊处理。给上面的程序加入 `if-then-else` 分支结构和比较，就可得到程序：

```
#!/usr/bin/perl -w
print "What's your name? ";
$name = <STDIN>;
chomp ($name);
if ($name eq "Yang Yi") {
    print "Hi, Yang Yi! How are you?\n";#特别的问候
```