

Perl 编程语言学习笔记

Roger Young

编译时间: March 30, 2017

Contents

1 语言简介	5
1.1 语言进化简史	5
1.2 基本概念	5
1.3 基本语法	5
1.4 标准或推荐的程序框架	5
1.5 基本编辑过程	6
1.6 基本运行方法	6
2 关键字	7
3 输入输出	9
3.1 输出	9
3.2 获取用户输入	9
4 基本数据结构	11
4.1 数字	11
4.1.1 数字的一般操作	12
4.2 字符串	13
4.2.1 转移字符	13
4.2.2 字符串中的变量插入: variable interpolated	14
4.2.3 字符串程序示例	14
4.3 基本数据类型之间的互换	15
4.4 内置类型	15
5 内置变量	17
6 控制语句	19
6.1 条件	19
6.2 循环	19

7 函数	21
7.1 内置函数	21
8 模块化机制和实践	23
9 面向对象机制	25
9.1 定义类	25
9.2 封装	25
9.3 继承	25
9.4 多态	25
9.5 接口	25
10 标准库介绍	27
11 常用应用场景	29
11.1 网络	29
11.2 字符操作	29
11.3 数据库	29
11.4 Json 操作	29
11.5 xml 操作	29
11.6 文件、目录操作	29
11.7 进程管理	29
11.8 线程管理	29
11.9 正则表达式	29

Chapter 1

语言简介

1.1 语言进化简史

Perl的全称可以认为是“*Practical Extraction and Report Language*”，也可以认为是“*Pathological Eclectic Rubbish Listeria*”。一般来说，以大写的“P”开头的“Perl”是指 Perl 编程语言，以小写字母“p”开头的“perl”是指用来解释和运行 Perl 程序的程序（或曰解释器，interpreter）。

1.2 基本概念

1.3 基本语法

本部分的基本语法是 Perl语法体系中最最基础的部分，包括：

- Perl中所有的语句需要以“;”结束；

1.4 标准或推荐的程序框架

对于一般的 Perl程序（当然也包括其他编程语言写得脚本、程序），建议采用 utf-8 编码，对于个人来说，常用的脚本框架如下：

```
#!/usr/bin/perl
```

```
use strict;  
use utf8;  
use warnings;
```

```
# 其他代码
```

1.5 基本编辑过程

1.6 基本运行方法

Chapter 2

关键字

Chapter 3

输入输出

3.1 输出

3.2 获取用户输入

Chapter 4

基本数据结构

Perl语言中没有类似于 C、C++中的 *int*、*long* 等表示对象或者变量类型的关键字。在 Perl 中变量的最基本类型根据其所参与的运算来决定。最基本类型是指数、字符、字符串等。当个最基本类型类型组成的变量或对象，在 Perl 中称为标量 (Scalar)。由多个标量，可以组成数组、字典等其他相对复杂类型。在 Perl 可以通过变量标识符前的 sigil，来确定变量的类型：

Sigil	变量类型	举例
\$	标量、scalar	<code>my \$days = 30;</code>
@	数组	<code>my @months = ("January", "February", "March")</code>
%	字典、Hash	<code>my \%months = ('Jan' => 'January', 'Feb' => 'February', 'Mar' => 'March');</code>

4.1 数字

Perl 依赖于低层的 C 库来表示数字。在 Perl 中，所有的数字在内部都以双精度浮点数 (double-precision floating-point value) 保存的。这意味着，Perl 对数字的精度和位数可能有些要求。同时，也意味着，Perl 中没有整数 (integer) 或者浮点数 (floating-number) 的概念。

Perl 数字举例：

```
#!/usr/bin/perl
```

```
use strict;
```

```
use utf8;
```

```
use warnings;
```

```
use diagnostics;
```

```
use v5.10;
```

```
my $chhogori_altitude = 8611;
```

```

my $mariana_trench_altitude = -11034;

my $chhogori_altitude_oct = 0206;
my $chhogori_altitude_hex = 0x21A3;
my $chhogori_altitude_bin = 0b10000110100011;
my $chhogori_altitude_bin_sep = 0b10_000_110_100_011;

my $planck_length = 1.616229e35;
my $planck_time = 5.39116E-44;
my $height = 1.73;
my $light_speed = 299_792_458;
my $sackur_tetrode_constant = -1.1517084;
my $atomic_mass = 1.660539040e-27;

use v5.22;
my $hex_exponent = 0x1f.0p3;

```

在 **Perl** 中，数字可以直接通过字面值输入。所谓“字面值”是指，在源程序中直接存在的、不需要在程序运行过程中经过运算或者 IO 操作就能得到的值。

通过数字字面值可以直接以十进制、八进制、十六进制、二进制表示整数。（计算机科学中的）整数的字面值可以是以下几种形式：

- 以阿拉伯数字开头的、连续的一个或多个阿拉伯数字；
- 以“+”或者“-”开头的，连续的一个或多个阿拉伯数字；
- 以“+0b”或者“-0b”开头的，连续的一个或多个“0”、“1”；
- 以“+0”或者“-0”开头的，连续的一个或多个小于 8 的阿拉伯数字；
- 以“+0x”或者“-0x”开头的，连续的一个或多个的字符，这些字符包括所有的阿拉伯数字、a、b、c、d、e、f、A、B、C、D、E、F；
- 以字符“_”分隔的上述类型；

浮点数的表示，跟我们常见的写法一致。浮点数可以以“.”开始、可以以表示整数的“+”或者“-”开始。浮点数也可以包含字母“E”或者“e”来表示科学计数法。对于 5.22 版以后的 **Perl**，也可以使用形如“\$hex_exponent”的形式表示浮点数。

4.1.1 数字的一般操作

Perl 支持数字常用的加、减、乘、除等算法，其对应的运算符如下：

操作	符号	举例	结果
加	+	3 + 2	5
减	-	3 - 2	1
乘	*	3 * 2	6
除	/	3 / 2	1.5
模 ¹	%	3 % 2	1
幂	**	3 ** 2	9

4.2 字符串

作为一门以字符处理为特长的语言，**Perl**的字符串功能相当强大。字符串字面值可以有两种表示形式，分别是：

- 以英文单引号，“'”括起来的字符串：在这种情况下，除了“'”和“\”外其他的字符都是代表其自身。（之所以是这两个，可能是因为，“'”已经用来分隔字符串了，如果想要在字符串中包含“'”，必须采用特殊的办法，而“\”被选中了。相应的，字符串中可以通过“\’”和“\\”来包含这两个特殊字符。）
- 以英文双引号，“””括起来的字符串。这类字符串与其他编程语言中的字符串比较类似。字符串中含有特殊的命令，可以对字符串进行必要的操作。也是在后续描述中篇幅比较大的一种。

4.2.1 转移字符

在 **Perl**中，以英文双引号，“””括起来的字符串，英文称为“Double-Quoted String Literal”。Double-Quoted String Literal 可以包含一些特殊意义的字符组合：

¹对于有负数参与的求模运算，其结果可能在不同的机器上不同。这意味着“say -10 % 3;”在可能产生不同的结果！！

转义字符组合	意义
<code>\n</code>	换行
<code>\r</code>	回车
<code>\t</code>	Tab
<code>\f</code>	Formfeed
<code>\b</code>	退格
<code>\a</code>	alarm, 响一声
<code>\e</code>	ASCII 码中的 Escape 字符
<code>\Odd</code>	以八进制表示的 ASCII 字符
<code>\xhh</code>	以十六进制表示的 ASCII 字符
<code>\N{charactername}</code>	charactername 表示的 Unicode 字符
<code>\cC</code>	控制字符
<code>\\</code>	<code>\</code>
<code>\"</code>	<code>"</code>
<code>\l</code>	下一个字符小写
<code>\L</code>	其后的字符小写, 直到遇到 <code>\E</code>
<code>\u</code>	下一个字符大写
<code>\U</code>	其后的字符小写, 直到遇到 <code>\E</code>
<code>\Q</code>	Quote nonword characters by adding a backslash until <code>\E</code>
<code>\E</code>	终止 <code>\L</code> 、 <code>\U</code> 、 <code>\Q</code>

4.2.2 字符串中的变量插入: variable interpolated

可以直接在以英文双引号, “” 括起来的字符串中插入变量。

4.2.3 字符串程序示例

```
#!/usr/bin/perl
```

```
use strict;
use utf8;
use warnings;
use diagnostics;
```

```
binmode STDOUT, ':utf8';
```

```
my $light_speed = 299_792_458;
```

```
print 'The speed of light in vacuum is important in physics.';
```

```
print "\n";  
print "可能乱码呀，稍后解决： \N{SNOWMAN、 }\x{2668}";  
print "The speed of light in vacuum is important in physics.\a\n";  
print "The speed of light in vacuum is $light_speed!"
```

4.3 基本数据类型之间的互换

4.4 内置类型

Chapter 5

内置变量

Chapter 6

控制语句

6.1 条件

6.2 循环

Chapter 7

函数

7.1 内置函数

Chapter 8

模块化机制和实践

Chapter 9

面向对象机制

9.1 定义类

9.2 封装

9.3 继承

9.4 多态

9.5 接口

Chapter 10

标准库介绍

Chapter 11

常用应用场景

11.1 网络

11.2 字符操作

11.3 数据库

11.4 Json 操作

11.5 xml 操作

11.6 文件、目录操作

11.7 进程管理

11.8 线程管理

11.9 正则表达式