

第一章 Perl 总览

概念型

Perl 中的变量类型

<i>Type</i>	<i>Character</i>	<i>Example</i>	<i>Is a name for:</i>
Scalar	\$	\$cents	An individual value (number or string)
Array	@	@large	A list of values, keyed by number
Hash	%	%interest	A group of values, keyed by string
Subroutine	&	&how	A callable chunk of Perl code
Typeglob	*	*struck	Everything named struck

构造复杂数据类型：引用

文件句柄 Filehandles

a name you give a file, device, socket or pipe.

运算符

控制结构

if unless

while until

for foreach

正则表达式

重要型

使用引用来构造复杂数据结构

上下文

运算符（单目、双目、三目）or（前置运算符、中置运算符）or（数字运算符、字符串运算符、文件操作符）

正则表达式

用途：1、判断字符串是否符合条件；2、定位字符串位置及替换；3、指出匹配和不匹配的地方并保存起来。

量词：+, {min, max}

定位符：^, \$, \b\b

反引用，记住匹配部分和不匹配部分。

列表处理，上下文，排序，匹配。

琐碎型

An reference example:

`$wife = ["Leah", "Rachel", "Bilhah", "Zilpah"];`

That statement creates an unnamed array and puts a reference to it into the Scalar `$wife`.

调用 `open` 函数来创建并关联一个文件句柄到某个文件，`open` 函数有至少两个参数，一个为文件句柄，另一个为文件名，`STDIN`，`STDOUT`，`STDERR` 为预定义的文件句柄。在文件句柄，需要指明操作方式。默认读，有读(<)，写(>)，追加(>>)，从管道读(| `output-pipe-command`)，向管道写(`input-pipe-command` |); 读行操作符(<>)可以操作文件句柄

双目运算符：加+、减-、乘*、除/、乘方**

字符串操作符：连字符(.)，将两个字符串连接成一个字符串

重复运算符（小写字母 `x`），将 `x` 前的内容重复 `x` 后的数字指明的次数

赋值操作符：

一元算术运算符：++，--

逻辑运算符：共有两种逻辑操作运算符集，一组从 C 语言中借鉴而来，一组从 BASIC 语言中借鉴而来。分别是 (`&&`, `||`, `!`) 和 (`and`, `or`, `not`, `xor`)

数字和字符串比较运算符：

用于数字：==, !=, <, >, <=, >=, <=>

用于字符串：eq, ne, lt, gt, le, ge, cmp

注：<=> 和 `cmp`，等于是结果为 0，如果左运算符大，则为 1，如果右边大，则为 -1

文件测试操作符：

Example	Name	Result
-e \$a	Exists	True if file named in \$a Exists
-r \$a	Readable	True if file named in \$a is readable
-w \$a	Writable	True if file named in \$a is writable
-d \$a	Directory	True if file named in \$a is a directory
-f \$a	File	True if file named in \$a is a regular file
-T \$a	Text File	True if file named in \$a is a text file
...

regular file 是相对于 irregular file 来说的，比如目录和设备文件

text file 是相对于 binary file 来说的，不可混淆。

Perl 中的真值：

除了 " " 和 "0" 外所有的字符串

除了 0 的所有数字

所有的引用

所有 undefined 值为假

心得型

You'd use array when you want to look something up by number. You'd use a hash when you want to look something up by name.

`${Timesecond}nd`，可以根据 `Timesecond` 的不同改变字符串。

`Map` 函数对列表中的每一个元素进行指定的处理并返回

`splice` 函数将列表指定范围内的元素进行替换