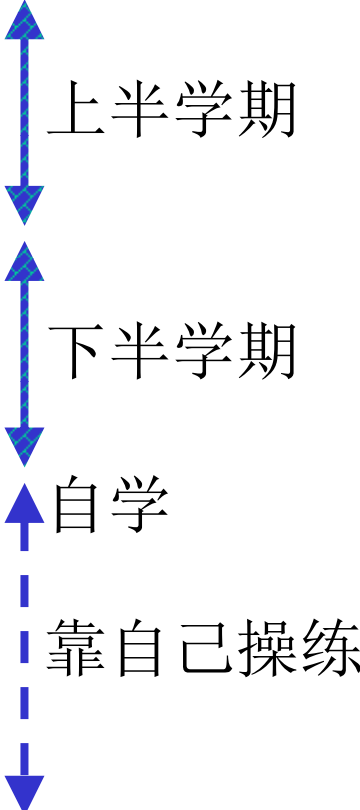


Perl 入门和提高 Lesson 1

周晓方

Perl等级

- 菜鸟级
 - 学会RE, Scalar, Array, Hash, 内部函数, perl变量, 子程序, 会写完整的Perl程序。
 - 进阶级
 - Perl引用和复杂的数据结构, PerlObj
 - 会用各种模块, 自己上cpan找模块用
 - 会用几种方法编写同样功能的程序
 - 高手级
 - perlXS, 把Perl嵌入其他语言中
 - 编写模块, 在CPAN上和大家共享
 - 骨灰级 ☠️?
- 
- 上半学期
- 下半学期
- 自学
- 靠自己操练

Perl的特色/参考书

- Perl and Larry Wall <http://www.wall.org/~larry/>
- Replace less powerful UNIX tools : sed, grep, awk, ...
- Easy to learn/to code, Free, Platform independent
- Many powerful & ready to use modules on CPAN
 - Perl/TK, Perl/Internet, Perl/DB, Perl/Bio-info
 - Perl/CGI, Perl/XS, Perl/Win32, Perl/OLE
- Perl Books:
 - “Learning Perl” 8th ed. yr.2021 “Programming Perl” 4th ed. yr.2012
 - “Quick Reference To Perl” “Perl 5 In Examples”
 - Perl编程24学时教程 "Perl CD book shelf ver1/2/3"

Perl安装和版本

- 多种平台 www.perl.com, 其中Windows平台
 - (推荐)草莓Perl: <http://strawberryperl.com/>
 - (不再推荐)~~ActivePerl: www.activestate.com 安装perl~~
- Android: Termux, pkg-install clang和autoconf
- 参看版本信息 perl -v perl -V (详细配置详细)
- 本课程的实例采用Win-32环境下的草莓Perl
- Perl版本和C编译器的关系
 - 编写或安装新的perl模块时可能用到C编译器
 - 安装某些Perl模块时, Perl版本和编译器版本要匹配
 - 草莓Perl内置mingw32和gcc, Make工具是gmake
 - ~~active perl 请找到对应的VC.net编译器和nmake~~
 - 建议采用文本编辑器和命令行调试Perl脚本
 - 有免费IDE, 如 <http://padre.perlide.org/>



Recommended version:

strawberry-perl-5.32.1.1-64bit.msi
strawberry-perl-5.32.1.1-32bit.msi

More downloads (all releases):



ZIP, Portable, special editions

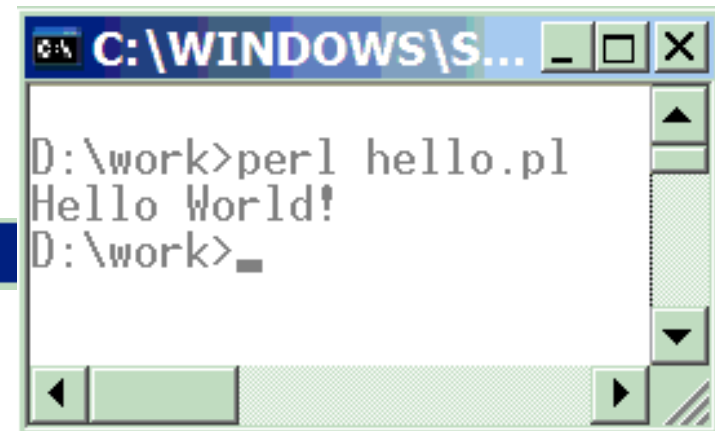
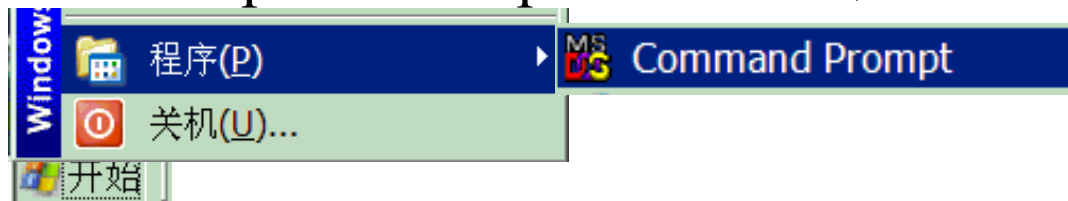
You can find here release notes and other details.

Recommended downloads

Version	Date	MSI edition	Portable	PDL edition	ZIP edition
5.32.1.1	2021-01-24	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit
5.30.3.1	2020-07-13	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit
5.28.2.1	2019-05-02	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit
5.26.3.1	2018-12-02	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit
5.24.4.1	2018-04-15	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit
5.22.3.1	2017-01-15	32bit/64bit	32bit/64bit	32bit/64bit	32bit/64bit

系统路径，运行脚本

- 确认perl在系统路径中
 - Linux: `echo $PATH` `which perl`
 - Win10/8/7等
 - 英文版: Control Panel → System → Advanced → Environment → System → Path
 - 中文版: 控制面板 → 系统 → 高级 → 环境变量 → 系统变量 → Path 是否包含 `C:\Strawberry\perl\bin\perl.exe`
 - Win9x: 查看 `c:\autoexec.bat` 在 `command.com` 下运行
- 运行perl脚本 开始菜单 → 运行 → `cmd.exe`
 - 打开命令窗 Console windows
 - `perl hello.pl`



My 1st perl program

```
#!/usr/local/bin/perl -w
use strict;
print "Hello world!\n";
1;      #program terminates successfully
```

- File extension "pl", "hello.pl" (also .p .cgi)
perl hello.pl ↵
- **#!** "she-bang", followed by a path to perl
- **;** statement separator
- ***use strict*** and ***-w***
- String, **"** and **'**, **\n**, **\t**, ****, **\? \? \033**, **\x1b**
- **1** and **0**, return value of your script
- One line perl script:
perl -e "print qq(Hello world!\n); "↵



Scalar ——— *\$foo*

- 脚本语言，几乎不区分整数、实数、和字符串
- *\$Variable_Name*, name is case-sensitive
- Value can be string, number, reference (object)
- Numbers (What about arbitrary size numbers?)
 - Decimal 1234 Octal 01234
 - Hexadecimal 0x1234 Fraction/float 12.34
 - Scientific notation 0.00001234, 1.234e-5

```
#!/usr/bin/perl -w
use strict;
my ($Hello, $TimeAround);      # my说明本地变量
$Hello = "Hello, World";
$TimeAround = 2;
print "$Hello for the ", $TimeAround , "nd time!\n";
1;
```

Hello, World for the 2nd time!

Interpolation, string

- "有插值"小心\$@, 用\转义 '不插值'

```
print "$Hello for the ${TimeAround}nd time!\n";
=> Hello, World for the 2nd time!
print '$Hello for the ${TimeAround}nd time!\n';
=>$Hello for the ${TimeAround}nd time!\n
print "My email is courses\@xfzhou.homeftp.org\n";
=>My email is courese@xfzhou.homeftp.org
```


- 字符串的另类表示法

- "有插值的串" qq/.../ qq(...) qq!...!
- '不插值的串' q/.../ q(...) q!...!
- `执行系统命令` qx/.../ *back quote*

```
print `dir *`; print "oops!\n";
=> ...
=> oops!
```

函数undef, 0, ""

弱类型
语言

- Undefined variable has the value *undef*
 - Treat as 0 when used as a number
 - Treat as zero length string when as a string
- The keyword **undef** has the value *undef*
- Function **defined**(EXPR) check if the value is defined or undefined.
- Function **undef**(LEXPR) clears the LEXPR and makes it undefined.
- Some functions return **undef** for special meanings
- 和C语言不同，写条件判断时要特别小心
 - if (\$A) 表示\$A!=0 and \$A ne "" and defined \$A
 - if (defined \$A)判断\$A是否有定义  (defined(\$num=shift @_))
 - 例如buggy的写法 while (\$num=shift @_){.....}

Homework

- Download and review viewgraphs
- Download & Install strawberry perl on your Windows
- Rewrite and run the 1st perl script hello.pl
- 编写perl程序，打印小九九乘法表(表格要对齐)

– 提示: Perl中的两重for循环可以是这样的:

```
my ($i, $j);
foreach $i (1..9) {
    foreach $j ($i..9) {
        ...print $i*$j...
    }
}
```

1	2	3	4	5	6	7	8	9
	4	6	8	10	12	14	16	18
		9	12	15	18	21	24	27
			16	20	24	28	32	36
				25	30	35	40	45
					36	42	48	54
						49	56	63
							64	72
								81

- Submit your 九九乘法表 answers by email to ~~homework@xfzhou.homeftp.org~~
- 主题写明学号姓名,程序放在附件中,学号-01.pl

About Homeworks

- Veni Vidi Vici
---*Julius Caesar*

- I came, I saw, I ~~charged~~
conquered



Nastasic / Getty Images

- Reserve you right to ..., for even to ... wrongly
is better than not to ... at all.

--- *Hypatia's father, Theon of Alexandria*

- O friendship too unfriendly!
--- "*Confessions*" Augustine

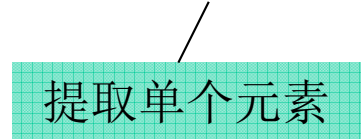
独自完成
作业

如何学好Perl

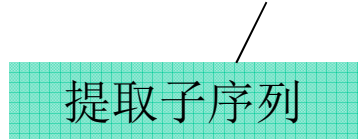
- 入门
 - 跟上课程的进度学习, 独立完成作业, 不缺课
 - 课后学习Perl入门书籍, 如《Learning Perl》等
 - 阅读在线帮助文档 `perldoc`
 - 多编写一些格式转换方面的脚本
 - 编程时要写文档、有良好的编程风格、有版本控制
- 提高
 - 听课, 学习例题, 掌握课上讲解的Perl模块
 - 看《Programming Perl》等书籍
 - 查阅CPAN了解更多的模块, 自学这方面的书
 - 尝试综合运用网络、用户界面、数据库等各种CPAN模块, 编写一些实用程序

Array/list — (*foo, bar, ...*), *@baz*

- 脚本语言，数组空间无须申明、释放，大小动态可变
- Perl array is always one-dimension. Perl数组总是一维的
- *@ArrName*, *\$ArrName[index]*, *@ArrName[i1, i2, i3]*
- (1, 2, 3, "oops", 5, 6)
(5, 1 .. 6, 7)



提取单个元素



提取子序列
- `qw(aaa bbb ccc)` non-white space delimited, non-interpolation
- Empty list ()

```
@junk = (1, "I", undef, 66..69, "foo");
```

```
($one, $two, $three, @more) = (1..9);
```

```
($a, $b, @many, $last) = (1..9);
```

```
($one, $two) = ($two, $one);
```

\$last得到undef

```
@evenmore = ($one, $two, @more, $three);
```

```
@who = (qw(fred barney betty wilma))[2,3];
```

Element access

- `$Name[Index]`, `$Name[-Rindex]`, `@Name[i1, i2, ..]`
- `$#Name`. Returns -1 when apply to empty list
 - *There're better ways to tell the size of an array*

```
@fred = (7,8,9);  
$b = $fred[0]; # give 7 to $b (first element of @fred)  
$fred[0] = 5; # now @fred = (5,8,9)  
@fred[1,2] = (9,10);  
@who = (qw(fred barney betty wilma))[2,3];
```

```
@fred = (7,8,9); # as in previous example  
@barney = (2,1,0); @backfred = @fred[@barney];
```

```
@fred = (1,2,3);  
$fred[3] = "hi"; # @fred is now (1,2,3,"hi")  
$fred[6] = "ho"; # @fred is now (1,2,3,"hi",undef,undef,"ho")
```

```
@fred = ("fred", "wilma", "pebbles", "dino");  
print $fred[-1];          # prints "dino"  
print $#fred;             # prints 3  
print $fred[$#fred];      # prints "dino"
```

push, pop, shift, unshift, splice

- push/pop, append/extract from right side
- unshift/shift, insert/extract from left side
- splice, insert/replace/extract/remove freely
- These built-in funcs have side effects.

```
@a = (4, 5, 6);
push(@a, 7..10);
unshift (@a, 1..3);
$a = shift @a; #返回1
$b = pop @a;   #返回10
```

4	5	6							
4	5	6	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>			
<u>1</u>	<u>2</u>	<u>3</u>	4	5	6	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
<u>2</u>	<u>3</u>	4	5	6	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	
<u>2</u>	<u>3</u>	4	5	6	<u>7</u>	<u>8</u>	<u>9</u>		

```
@c = splice(@a, 3, 4, @b);
# replace @a[3,4,5,6] with @b
# return original @a[3,4,5,6]
```


Perl reference to scalar and list

- Reference: \

```
$Rscalar = \ $scalar;    ${$Rscalar} is scalar;  
$Rarray  = \ @array;     @{$Rarray} is the arr  
$Rref    = \ $ref;       ${$Rref} is the ref
```

- DeReference: \$ @

- Reference to anonymous list: []

```
$r_a = ['A', 55, 1..6, 'oops'];
```

```
print "$r_a\n";  
print "@{$r_a}\n";  
print ${$r_a}[1] + 5;
```

显示结果

```
ARRAY(0x183efdc)  
A 55 1 2 3 4 5 6 oops  
60
```

Complex Structure in Perl

- list of list

```
$arr = ['A', 55, [1..6], 'oops'];  
print ${$arr}[0], "\n";  
print "@{${$arr}[2]}\n";  
print ${$arr}[2][0];
```

显示结果

A

1 2 3 4 5 6

1

[2]号元素也是个数组

[2][0]号元素的值是1

- This is multi-dimension array in perl5
- Perl-obj is also based on perl reference.

本课小结

- Perl标量(scalar)
 - 不区别类型
- Perl数组(array, list)
 - 数组深度和类型无需事先说明，可动态变化
- Perl不使用指针，使用引用(Reference)
 - 没有类似C/Cpp的struct/class，完全由引用构成复杂的数据结构，也就无需预定义struct结构
- Perl的双引号或qq//有插值作用，用\转义
- 真假判断要小心，undef, "", 0都是假
 - 空的list ()是不是假？