

第二章：简单变量

生命科学学院

简单变量

- 基本上，简单变量就是一个数据单元，这个单元可以是数字或字符串。
- 一、整型
- 1、整型
- PERL 最常用的简单变量， 由于其与其它语言基本相同， 不再赘述。
- 例：


```
$x = 12345;
if (1217 + 116 == 1333)
{
    # statement block goes here
}
```



简单变量

- ▶ PERL 实际上把整数存在你的计算机中的浮点寄存器中，所以实际上被当作浮点数看待。在多数计算机中，浮点寄存器可以存贮约16 位数字，长于此的被丢弃。整数实为浮点数的特例。

8 进制和16 进制数

- ▶ 8 进制以0 打头， 16 进制以0x 打头。
- ▶ 例： `$var1 = 047;`（等于十进制的39）
`$var2 = 0x1f;`（等于十进制的31）

浮点数

- ▶ 如 11.4 、 -0.3 、 .3 、 3. 、 54.1e+02 、 5.41e03 浮点寄存器通常不能精确地存贮浮点数，从而产生误差，在运算和比较中要特别注意。指数的范围通常为 -30^9 到 $+30^8$ 。

字符串

- 惯用C 的程序员要注意，在PERL 中，字符串的末尾并不含有隐含的NULL 字符， NULL 字符可以出现在串的任何位置。
- 双引号内的字符串中支持简单变量替换， 例如：
- `$number = 11 ;`
- `$text = "This text contains the number $number."`;
- 则\$text 的内容为： "This text contains the number 11 . "



双引号内的字符串中支持转义字符

- ▶ `\a` Bell (beep)
- ▶ `\b` Backspace
- ▶ `\cn` The Ctrl+n character
- ▶ `\e` Escape
- ▶ `\E` Ends the effect of `\L`, `\U` or `\Q`
- ▶ `\f` Form feed
- ▶ `\l` Forces the next letter into lowercase
- ▶ `\L` All following letters are lowercase



双引号内的字符串中支持转义字符

- ▶ `\n` Newline
 - ▶ `\r` Carriage return
 - ▶ `\Q` Do not look for special
 - ▶ pattern character s
 - ▶ `\t` Tab
 - ▶ `\u` Force next l e t t e r i n t o
 - ▶ uppercase
 - ▶ `\U` A l l fol l owing letters are
 - ▶ uppercase
 - ▶ `\v` Vertical tab
-



双引号内的字符串中支持转义字符

- ▶ 要在字符串中包含双引号或反斜线， 则在其前加一个反斜线，

- ▶ 反斜线还可以取消变量替换， 如：

```
$res = "A quote \" and A backslash \\";
```

```
$result = 14 ;
```

```
print ( "The value o f \"$result is $result.\n")
```

的结果为：

The value of \$result is 14 .

双引号内的字符串中支持转义字符

- ▶ 可用 `\nnn` (8 进制) 或 `\xnn` (16 进制) 来表示 ASCII 字符, 如:

```
$result = "\377" ; # this is the character
255, or EOF
```

```
$result = "\xff" ; #this is also 255
```



单引号字符串

- ▶ 单引号字符串与双引号字符串有两个区别，一是没有变量替换功能，二是反斜线不支持转义字符，而只在包含单引号和反斜线时起作用。单引号另一个特性是可以跨多行， 如：

```
$text = 'This is two  
lines of text';
```

与下句等效：

```
$text = "This is two\nlines of text\n";
```



字符串和数值的互相转换

▶ 例1 :

```
$string = "43" ;
```

```
$number = 28 ;
```

```
$result = $string + $number ; # $result = 71
```

▶ 若字符串中含有非数字的字符， 则从左起至第一个非数字的字符，

如：

```
$result = "hello" * 5 ; # $result = 0
```

```
$result = "12a34"+1; # $result = 13
```

变量初始值

- 在PERL 中，所有的简单变量都有缺省初始值： "", 即空字符。但是建议给所有变量赋初值，否则当程序变得大而复杂后，很容易出现不可预料且很难调试的错误。

