

## 第七章：控制结构

生命科学学院

# 一、条件判断

---

```
if ( <expression> ) {  
    <statement_block_1>  
}  
  
elsif ( <expression> ) {  
    <statement_block_2>  
}  
  
...  
  
else {  
    <statement_block_3>  
}
```

---



## 二、循环：

---

### 1 、while 循环

```
while ( <expression> ) {  
    <statement_block>  
}
```

### 2 、until 循环

```
until ( <expression> ) {  
    <statement_block>  
}
```



### 3、类C 的for 循环 ， 如

```
for ($count=1; $count <= 5; $count++) {
# statements inside the loop go here
}
```

下面是在for 循环中使用逗号操作符的例子：

```
for($line = <STDIN>, $count = 1; $count <= 3; $line =<STDIN>, $count++)
{
    print ($line) ;
}
```

它等价于下列语句：

```
$line = <STDIN>;
$count = 1;
while ($count <= 3) {
    print ($line);
    $line = <STDIN>;
    $count++;
}
```



4、针对列表(数组)每个元素的循环： **foreach**， 语法为：

---

```

foreach localvar ( list expr ) {
    statement_block;
}

```

例：

```

foreach $word (@words) {
    if ( $ word eq "the" ) {
        print ( "found the word ' the' \n" ) ;
    }
}

```



注：

- (1) 此处的循环变量 `localvar` 是个局部变量，如果在此之前它已有值，则循环后仍恢复该值。
- (2) 在循环中改变局部变量，相应的数组变量也会改变，如：

```
@list = ( 1 , 2 , 3 , 4 , 5 ) ;
foreach $temp (@list) {
    if( $temp == 2 ) {
        $temp = 20 ;
    }
}
```

此时 `@list` 已变成了 ( 1 , 20 , 3 , 4 , 5 ) 。



## 5 、 do 循环

---

```
do {  
    statement_block  
} while_or_until (condexpr);
```

do 循环至少执行一次循环。



## 6、循环控制

---

退出循环为last， 与C 中的break 作用相同； 执行下一个循环为next， 与C 中的continue 作用相同； PERL 特有的一个命令是redo， 其含义是重复此次循环， 即循环变量不变， 回到循环起始点， 但要注意， redo 命令在do 循环中不起作用。

7、传统的goto label;语句。

---





### 三、单行条件

---

语法为 `statement keyword condexpr`。其中 `keyword` 可为 `if`、`unless`、`while` 或 `until`，如：

```
print ("This is zero. \n ") if ($var == 0);
print ("This is zero.\n") unless ($var != 0);
print ( "Not zero yet.\n ") while ($var-- > 0);
print ("Not zero yet. \n ") until ($var-- == 0);
```

虽然条件判断写在后面， 但却是先执行的。

---

