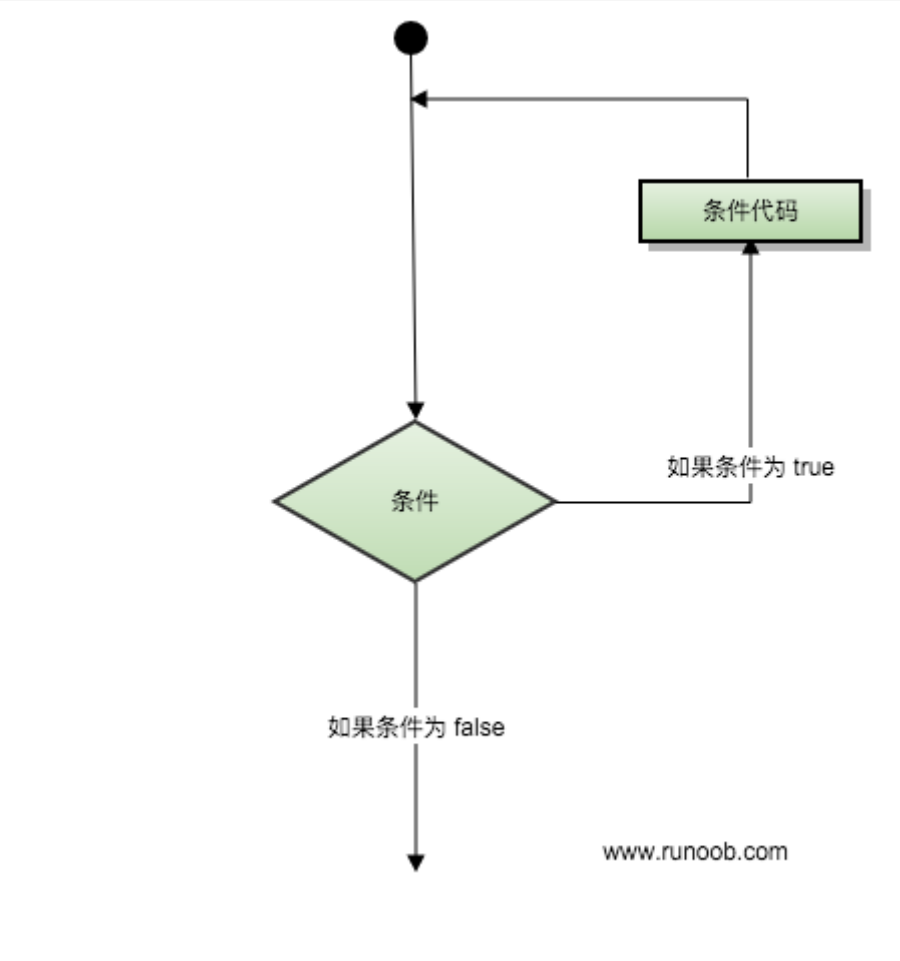


Perl 循环

有的时候，我们可能需要多次执行同一块代码。一般情况下，语句是按顺序执行的：函数中的第一个语句先执行，接着是第二个语句，依此类推。

编程语言提供了更为复杂执行路径的多种控制结构。

循环语句允许我们多次执行一个语句或语句组，下面是大多数编程语言中循环语句的 流程图：



注意，数字 0，字符串 '0'、""，空 list ()，和 undef 为 **false**，其他值均为 **true**。true 前面使用 **!** 或 **not** 则返回 false。

Perl 语言提供了以下几种循环类型:

循环类型	描述
while 循环	当给定条件为 true 时，重复执行语句或语句组。循环主体执行之前会先测试条件。
until 循环	重复执行语句或语句组，直到给定的条件为 true。循环主体执行之前会先测试条件。
for 循环	多次执行一个语句序列，简化管理循环变量的代码。

foreach 循环	foreach 循环用于迭代一个列表或集合变量的值。
do...while 循环	除了它是在循环主体结尾测试条件外，其他与 while 语句类似。
嵌套循环	您可以在 while、for 或 do..while 循环内使用一个或多个循环。

循环控制语句

循环控制语句改变了代码的执行顺序，通过它你可以实现代码的跳转。

Perl 提供了下列的循环控制语句：

控制语句	描述
next 语句	停止执行从next语句的下一语句开始到循环体结束标识符之间的语句，转去执行continue语句块，然后再返回到循环体的起始处开始执行下一次循环。
last 语句	退出循环语句块，从而结束循环
continue 语句	continue 语句块通常在条件语句再次判断前执行。
redo 语句	redo 语句直接转到循环体的第一行开始重复执行本次循环，redo语句之后的语句不再执行，continue语句块也不再执行；
goto 语句	Perl 有三种 goto 形式：got LABEL，goto EXPR，和 goto &NAME。

无限循环

如果条件永远不为 false，则循环将变成无限循环。

for 循环在传统意义上可用于实现无限循环。

由于构成循环的三个表达式中任何一个都不是必需的，您可以将某些条件表达式留空来构成一个无限循环。

实例

```
#!/usr/bin/perl
for( ; ; )
{
    printf "循环会无限执行。 \n";
}
```

你可以按下 **Ctrl + C** 键来终止循环。

当条件表达式不存在时，它被假设为 true 。您也可以设置一个初始值和增量表达式，但是一般情况下，Perl 程序员偏向于使用 for(;;) 结构来表示一个无限循环。

 点我分享笔记