

## awk从放弃到入门（8）：awk动作总结之二

在本博客中，AWK是一个系列文章，本人会尽量以通俗易懂的方式递进的总结awk命令的相关知识点。

awk系列博文直达链接：[AWK命令总结之从放弃到入门（通俗易懂，快进来看）](#)



这篇文章中的知识点是建立在前文的基础上的，如果你还没有掌握前文中的知识，请先参考之前的文章。

**注：**在阅读这篇文章之前，最好已经了解了一些开发的基本语法，比如，for、while、do...while 等，否则在阅读时 有可能遇到障碍。

前文中，我们介绍了awk中的条件判断语句。

比如 "if"

比如 "if...else..."

比如 "if...else if...else..."

我们说过，这些语句都可以归类为"控制语句"，但是这些并不是"控制语句"的全部，今天我们继续聊聊awk的"控制语句"。

今天的话题主要与"循环"有关，说到"循环"，你一定会想起 for、while 等字眼，没错，awk中也有for循环、while循环等控制语句。

其实，只要我们有一点点开发基础，理解这些都不是什么难事，因为它们的用法与其他编程语言中的用法并没与什么区别。

我们先来看一下这些循环控制语句的语法，然后再进行举例。

```
1  #for循环语法格式1
2  for(初始化; 布尔表达式; 更新) {
3  //代码语句
4  }
5
6  #for循环语法格式2
7  for(变量 in 数组) {
8  //代码语句
9  }
10
11 #while循环语法
12 while( 布尔表达式 ) {
13 //代码语句
14 }
15
16 #do...while循环语法
17 do {
18 //代码语句
19 }while(条件)
```

好了，了解了语法，现在来看看示例，就从for循环开始吧。

因为我们还没有介绍过数组，所以此处只演示上述语法中的格式1的用法。

```
[www.zsythink.net]# awk 'BEGIN{ for( i=1;i<=6;i++ ){ print i } }'
1
2
3
4
5
6
[www.zsythink.net]#
```

zsythink.net 未双印博客

上例中，我们使用了BEGIN模式，BEGIN模式对应的动作中，包含了for循环语句，看到这里，是不是感觉与其他语言中的for循环完全没有区别嘛？只不过，上例中都写在一行中而已。

再来看看while循环的具体使用，为了方便演示，仍然使用BEGIN模式，示例如下。

```
[www.zsythink.net]# awk -v i=1 'BEGIN{ while(i<=5){print i;i++} }'
1
2
3
4
5
[www.zsythink.net]# awk 'BEGIN{i=1;while(i<=5){print i;i++} }'
1
2
3
4
5
[www.zsythink.net]#
```

zsythink.net 未双印博客

当while对应的条件满足时，则执行对应的语句，语句执行完成后，对条件进行修改。

同理，do...while的示例如下，它与while循环的不同之处在于，while循环只有当满足条件时才会执行对应语句，而do...while循环则是无论是否满足条件，都会先执行的代码，然后再判断是否满足while中对应的条件，满足条件，则执行do对应的代码，如果不满足条件，则不再执行do对应的代码。

```
[www.zsythink.net]# awk 'BEGIN{ i=1; do{print "test";i++}while(i<1) }'
test
[www.zsythink.net]#
[www.zsythink.net]# awk 'BEGIN{ do{print "test";i++}while(i<=5) }'
test
test
test
test
test
test
test
[www.zsythink.net]#
```

zsythink.net 未双印博客

正如下图所示，无论是否满足while中的条件，都会先执行一遍do对应的代码。

那么，说到循环，就不能不说与循环有关的跳出语句。

没错，与其他编程语言中一样，在awk中，同样可以使用break与continue跳出循环。

continue的作用：跳出“当前”循环

break的作用：跳出“整个”循环

示例如下，先看看continue的示例

由于在for循环中添加了条件判断，所以当i的值为3时，跳过了“当前本次”循环，没有执行当前本次循环需要执行的动作，所以上例中，数字“3”并没有被打印出来

当然，如果你想“结束”的更加“彻底”，可以使用break结束循环，示例如下。

如上图所示，break结束的更加彻底，当使用break时，整个循环都将被结束，循环中的动作将不会再被执行。

continue与break同样可以用于while循环与do...while循环，此处就不再赘述了。

当然，如果你经常编写过shell脚本，你可能会问，awk中有类似exit的语句吗？必须有啊，在shell中，exit命令表示退出当前脚本，在awk中，它的含义也是类似的执行awk命令，相当于退出了当前的awk命令，示例如下。

如上图所示，上图中第一条命令中，执行了多个动作（多条语句），上图中的第二条命令中，也执行了多个动作，但是当在awk中执行了exit语句以后，之后的所有被执行，相当于退出了整个awk命令。

其实，这样描述exit的作用并不准确，因为，当在awk中使用了END模式时，exit的作用并不是退出整个awk命令，而是直接执行END模式中的动作，示例如下。

如上图所示，当awk中使用了END模式时，如果执行了exit语句，那么exit语句之后的所有动作都将不会再被执行，END模式中的动作除外。

换句话说就是，当执行了exit语句后，如果使用了END模式，将直接执行END模式中的动作，其他动作将不会被执行，如果没有使用END模式，当执行了exit语句后整个awk命令。

在awk中，除了能够使用"exit命令"结束"整个awk"，还能够使用"next命令"结束"当前行"，什么意思呢？我们慢慢聊。

在前文中，我们提到过，awk是逐行对文本进行处理的，也就是说，awk会处理完当前行，再继续处理下一行，那么，当awk需要处理某一行文本的时候，我们能不："不用处理这一行了，直接从下一行开始处理就行了"。

没错，使用next命令即可让awk直接从下一行开始处理，换句话说就是，next命令可以促使awk不对当前行执行对应的动作，而是直接处理下一行，示例如下。

其实，next与continue有些类似，只是，continue是针对"循环"而言的，continue的作用是结束"本次循环"，而next是针对"逐行处理"而言的，next的作用是结束"处理"，从而直接处理"下一行"，其实，awk的"逐行处理"也可以理解成为一种"循环"，因为awk一直在"循环"处理着"每一行"，不是吗？

awk的常用的流程控制语句与循环语句都已经总结完毕了，希望这篇文章能够帮助到你。

