

PHP 中跳出多重循环使用 break, continue, goto, return, exit 的用法和区别

Php 中的循环结构大致有 for, while, do...while, foreach 四种。

跳出循环方式 break, continue, return, exit, die, goto.

1. break

break 是被用在上面所提出的各种循环和 switch 语句中。他的作用是跳出当前的语法结构，执行下面的语句。Break 可以带一个参数 n, 表示跳出的循环的层数，如果要跳出多重的循环的话，可以用 n 来表示跳出的层数，如果不带参数默认是跳出本重循环。

```
For ($i=1;$i<=10;$i++) {  
    For ($j=1;$j<=10;$j++) {  
        $m=$i * $i+$j*$j;  
        Echo $m  
        If ($m<90 || $m>190) {  
            Break 2;  
        }  
    }  
}
```

这里使用了 `break 2` 跳出了两重循环，如果不使用参数，跳出的是本次循环，第一层循环会继续执行下去。

2. `continue`

`continue` 是用来用在循环结构中，控制程序放弃本次循环 `continue` 语句之后的代码并转而进行下一次循环。`Continue` 本身并不跳出循环结构，只是放弃这次循环。如果在肺循环结构中使用（例如 `if`，`switch` 中）会报错。

```
For ($i=1;$i<=100;$i++) {  
    If ($i%3==0 || $i%7==0) {  
        Continue;  
    } else {  
        Echo $i;  
    }  
}
```

3. `return`

`return` 语句是用来结束一段代码，并返回一个参数的。可以从一个函数里调用，也可以从一个 `include ()` 或者 `require ()`

语句包含的文件里来调用，也可以是在主程序里调用，如果是在函数中调用程序会马上结束运行并返回参数，如果是 `include ()` 或者是 `require ()` 语句包含的文件中被调用，程序执行将会马上返回到调用该文件的程序，而返回值将作为 `include ()` 或 `require ()` 的返回值。而如果是在主程序中调用，那么主程序将会马上停止执行。结束当前执行脚本。

```
For ($I=1000;$i>=1;$i-){  
    If(sqrt($i)>=29){  
        Echo $i;  
    }else{  
        Return;  
    }  
}  
  
Echo ‘本行将不会被输出’;
```

这里的例子和使用 `exit` 的效果是一样的。

在循环结束条件，自然跳出，当循环满足循环临界条件是就是自己退出。

4. exit

`exit` 是用来结束程序执行的。可以用在任何地方，本身没有跳出循环的含义，`exit` 可以带一个参数，如果参数是字符串，php 将会直接把字符串输出，如果参数是 `integer` 整形（范围是 0 - 254），那个参数将会被作为结束状态使用。

```
For ($i=1000;$i>=1;$i-) {  
    If(sqrt($i)>=29) {  
        Echo $i,  
    }else{  
        Exit;  
    }  
}  
  
Echo '本行将不会输出' ;
```

上面的这个例子中直接在循环里结束了代码的运行，这样会导致后面的代码都不会被执行，如果是在一个 php web 页面里面，甚至连 `exit` 后面的 html 代码都不会被输出。

5. die

die 退出循环和 exit 退出是一样的。都是语法结构。

6. goto

goto 实际上只是一个运算符，和其他语言一样，PHP 中也不鼓励滥用 goto，滥用 goto 会导致 程序的可读性严重下降。goto 的作用是将程序的执行从当前位置跳转到其他任意位置，goto 本身并没有要结束的循环的作用，但其跳转位置的作用使得其可以作为跳出循环使用。但 PHP5.3 及以上版本停止了对 goto 的支持，所以应该尽量避免使用 goto。

```
For ($i=100;$i>=1;$i-) {  
    If(sqrt($i)<=29) {  
        Goto a;  
    }  
    Echo $i;  
}  
  
A:  
Echo "this is the end" ;
```