

Day 04---循环结构



一、为什么需要循环

思考：100 遍 “好好学习，天天向上”

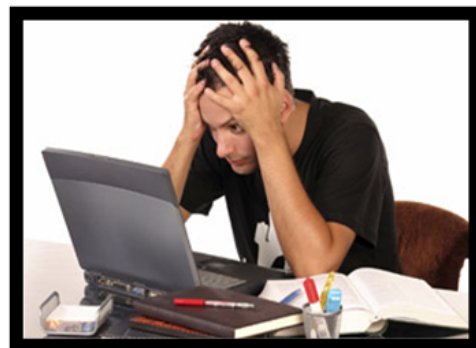
生活中的循环



打印 50 份试卷



10000 米赛跑



做 100 道编程题

二、三大循环结构

for循环、while循环、do...while循环。这三大循环在所有的编程语言中几乎都存在。这三种循环本质上是一样的，只是在语法上稍有不同。

另外，JavaScript为了提高遍历对象的属性和数组元素的性能，增加了for-in循环，待面向对象阶段再细讲。

2.1 for循环

语法：

```
for(表达式1; 表达式2; 表达式3){  
    //循环体  
}
```

说明：

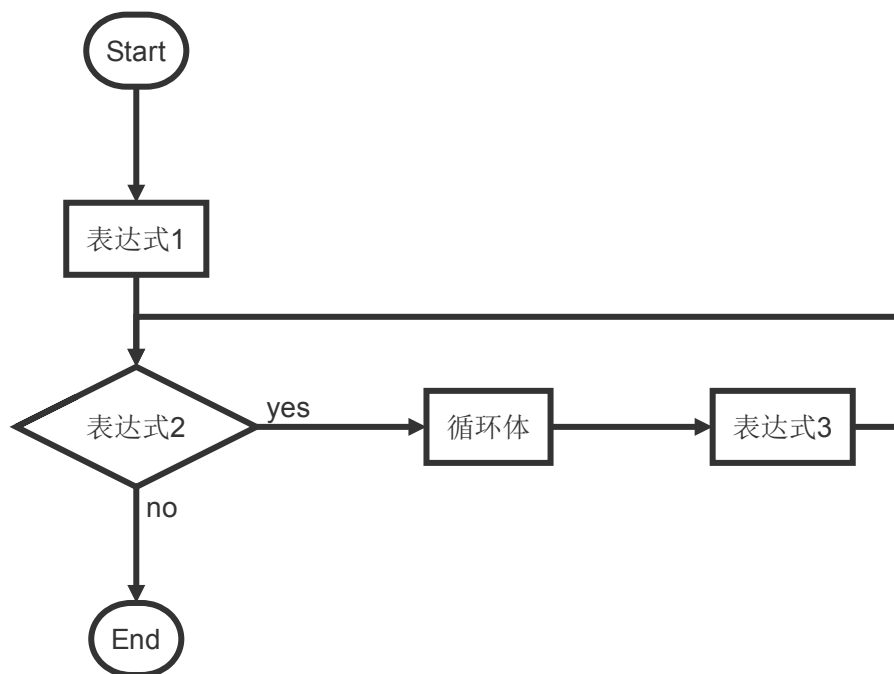
1. for循环开始执行，首先执行表达式1，表达式1一般是对循环变量做初始化的操作。表达式1在整个for循环执行期间只会执行一次。

2. 表达式1执行完毕后开始执行表达式2，如果表达式2最终的结果是true(或者可以通过Boolean()转化函数转换为true)，则开始执行循环体。如果表达式2最终的结果是false，则循环语句结束。
3. 循环执行完毕之后，开始执行表达式3，表达式3一般是对循环做自增或自减的操作。
4. 表达式3执行完毕后，继续执行表达式2。继续上面第2步的操作。

注意：

- 3个表达式的都可以省略。
- 对表达式1和表达式3省略，对for循环没有任何影响，只是少执行了代码而已。
- 如果表达式2省略，表示此处为true，那么这个循环就是死循环。
- 如果第一次检查表达式2的时候就是false，则循环体内的代码可能一次也不执行。

流程图：



2.2 while循环

语法：

```
while(condition){  
    // 循环体  
}
```

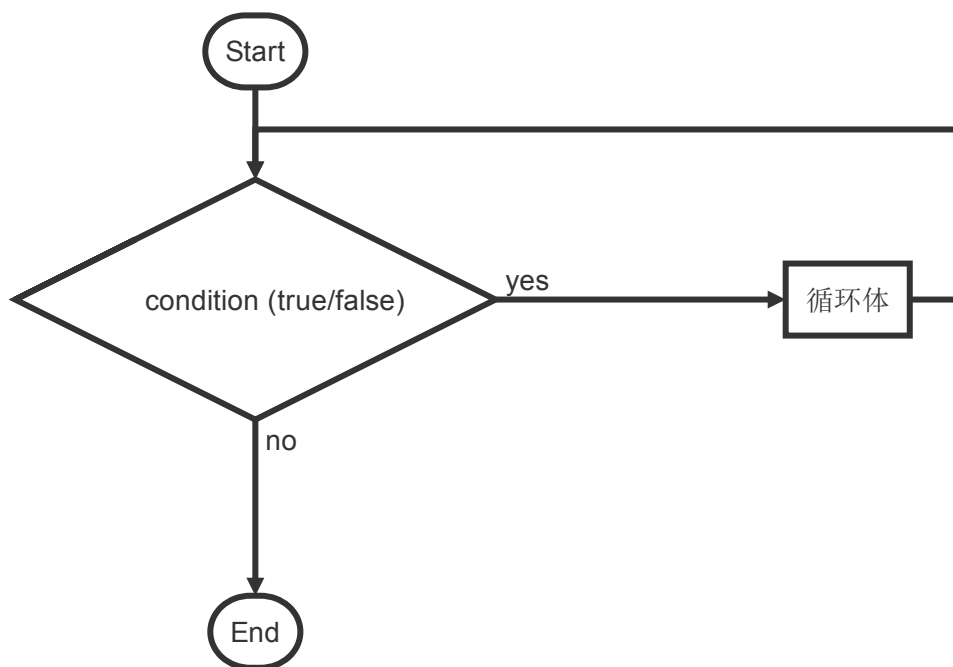
说明：

1. 先判断condition是true还是false，如果是true，则执行循环体，循环体执行完毕，再次判断condition
2. 如果condition为false，则结束循环。

注意：

- condition不能省略。如果省略为语法错误
- while循环也有可能一次也不执行。

流程图：



2.3 do...while循环

语法：

```
do{  
    //循环体  
}while(condition);
```

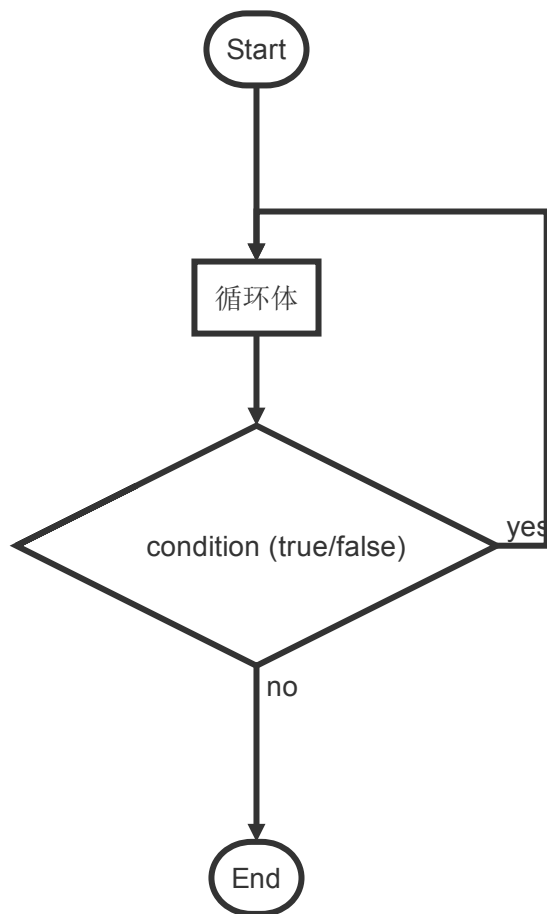
说明：

1. 先执行循环体。
2. 循环体执行结束后，去判断condition，如果condition是true，则再次执行循环体，否则循环结束。

注意：

- condition条件不能省略，省略语法错误。
- 由于先执行在判断，所以，对do...while 循环来说，循环体至少执行一次。

流程图：



2.4 三大循环比较

从本质上来讲，三大循环都可以完成相同的工作。其实一个for循环可以适应所有的需要循环的场景。但是，有些地方用某个可能更方便一些。

从实际情况来看，大部分人更新换用for，比如我。

循环结构	特点和用途
for	一般用于精确控制循环次数的场景。
while	不能提前知道循环次数
do...while	不能提前知道循环次数，且循环体的代码至少执行一次的场景

三、两个特殊的流程控制语句

在循环内部，有的时候需要提前结束循环，或者结束本轮循环进入下一轮循环，就需要用到两个特殊的控制语句：break和continue

3.1 break语句

把break语句放入到循环中，代表提前结束循环。

```
for (var i = 0; i < 5; i++) {  
    if(i == 3){  
        break; // 如果i == 3 则直接结束循环(跳出循环)，不会再执行以后的代码。 也不会去执行i++和判断i<5  
    }  
    document.write(i + " ");  
}  
//结果输出: 0 1 2
```

3.2 continue语句

把continue语句放在循环中，表示提前结束本轮循环，不在执行循环体内剩下的代码，而是继续执行for中的表达式3或while中的条件。

```
for (var i = 0; i < 5; i++) {  
    if(i == 3){  
        //如果i == 3，执行continue语句，则提前结束本轮循环，不再执行循环体剩下的代码。直接去执行i++，然后判断，开启  
        //下一轮循环。  
        continue;  
    }  
    document.write(i + " ");  
}
```