

## 1.If 语句格式 1 及执行流程

### 1.1 if 语句第一种格式：

```
if(关系表达式) {  
    语句体  
}
```

\*如果 if 语句中只有一句语句体，可以省略大括号{}

```
if(关系表达式)  
    语句体
```

执行流程:

首先判断关系表达式看其结果是 true 还是 false

如果是 true 就执行语句体

如果是 false 就不执行语句体

### 1.2 if 语句第二种格式：

```
if(关系表达式) {  
    语句体 1;  
}  
else {  
    语句体 2;  
}
```

执行流程

首先判断关系表达式看其结果是 true 还是 false

如果是 true 就执行语句体 1

如果是 false 就执行语句体 2

### 1.3 if 语句第三种格式：

```
if(关系表达式 1) {
```

```

        语句体 1;

    }else if (关系表达式 2) {

        语句体 2;

    }

    ...

    else {

        语句体 n+1;

    }

```

#### 执行流程

首先判断关系表达式 1 看其结果是 true 还是 false

如果是 true 就执行语句体 1

如果是 false 就继续判断关系表达式 2 看其结果是 true 还是 false

如果是 true 就执行语句体 2

如果是 false 就继续判断关系表达式...看其结果是 true 还是 false

...

如果没有任何关系表达式为 true , 就执行语句体 n+1

\*if 嵌套

```

    if(关系表达式 1) {

        If(关系表达式)

    }else {

        If(关系表达式)

        语句体 n+1;

    }

```

if 语句的三种格式：

- \* 第一种格式适合做一种情况的判断
- \* 第二种格式适合做二种情况的判断
- \* 第三种格式适合做多种情况的判断

## 2 switch 语句格式和执行流程

switch 语句格式：

```
switch(表达式) {  
    case 值 1:  
        语句体 1;  
        break;  
    case 值 2:  
        语句体 2;  
        break;  
    ...  
    default:  
        语句体 n+1;  
        break;  
}
```

**\*\*break 可以不写，但程序会贯穿执行到下一个 break 时再停止运行！**

格式解释：

表达式:byte,short,int,char

JDK5 以后可以是枚举

JDK7 以后可以是字符串

case:就是要和表达式进行比较的值

break:表示中断,结束的意思。

default:表示所有的情况都不匹配的时候,就执行语句体 n+1。和 if 语句的 else 相似。

执行流程：

A:计算出表达式的值

B:拿计算出来的值和 case 后面的值依次比较,一旦有对应的值,就执行该处的语句,在执行过程中,遇到 break,就结束。

C:如果所有的 case 都不匹配,就会执行 default 控制的语句,然后结束。

### 3.for 循环的格式及基本使用

#### 3.1 for 循环语句格式：

```
for(初始化语句;判断条件语句;控制条件语句) {  
    循环体语句;  
}
```

#### 3.2 执行流程

A:执行初始化语句

B:执行判断条件语句,看其结果是 true 还是 false

如果是 false,循环结束。

如果是 true,继续执行。

C:执行循环体语句

D:执行控制条件语句

E:回到 B 继续

### 4.while 循环语句格式

#### 4.1 基本格式

```
while(判断条件语句) {  
    循环体语句;  
}
```

扩展格式

```
初始化语句;  
  
while(判断条件语句) {  
    循环体语句;  
}
```

## 5.do...while 循环语句格式

基本格式

```
do {  
    循环体语句;  
}while((判断条件语句);
```

扩展格式

```
初始化语句;  
  
do {  
    循环体语句;  
    控制条件语句;  
} while((判断条件语句);
```

\*模拟连接 wifi , 登录

执行流程：

\*       A:执行初始化语句;

- \* B:执行循环体语句;
- \* C:执行控制条件语句;
- \* D:执行判断条件语句，看是 true 还是 false
- \* 如果是 true，回到 B 继续
- \* 如果是 false，就结束

## 6.三种循环的区别

### 6.1 区别概述

虽然可以完成同样的功能，但是还是有小区别：

do...while 循环至少会**执行一次**循环体。

for 循环和 while 循环只有在**条件成立**的时候才会去执行循环体

for 循环语句和 while 循环语句的小区别：

使用区别：控制条件语句所控制的那个**变量**，在 for 循环结束后，就不能再被访问到了，而

while 循环结束还可以继续使用，如果你想继续使用，就用 while，否则推荐使用 for。原

因是 for 循环结束，该变量就从内存中消失，能够提高内存的使用效率。

**\*知道循环次数时建议使用 for 循环，不知道循环次数时使用 while 循环**