集训课-03：选择结构

课程目标

01 关系运算符

02 单分支结构

03 双分支结构

04 分支嵌套

05 总结回顾

Part 1

关系运算符

关系运算符

符号	关系	举例	返回结果类型
>	大于	10 > 8 9 >10	bool型数据
>=	大于或者等于	100 >= 90 100 >= 100	
<	小于	128 < 120 50 < 50	
<=	小于或者等于	10 <= 3 88 <= 88	
==	相等	11 == 11	
!=	不等于	256 != 128	

关系运算符

想一想1

将下面的关系运算符和表达含义进行连线

==	大于等于
!=	相等
>=	小于
<	不等于

关系运算符

- 打印关系表达式必须加上小括号

例如

```
cout << (888 > 666);

cout << (114514 < 54188);
```

关系运算符

- 关系表达式返回的结果类型为bool型数据

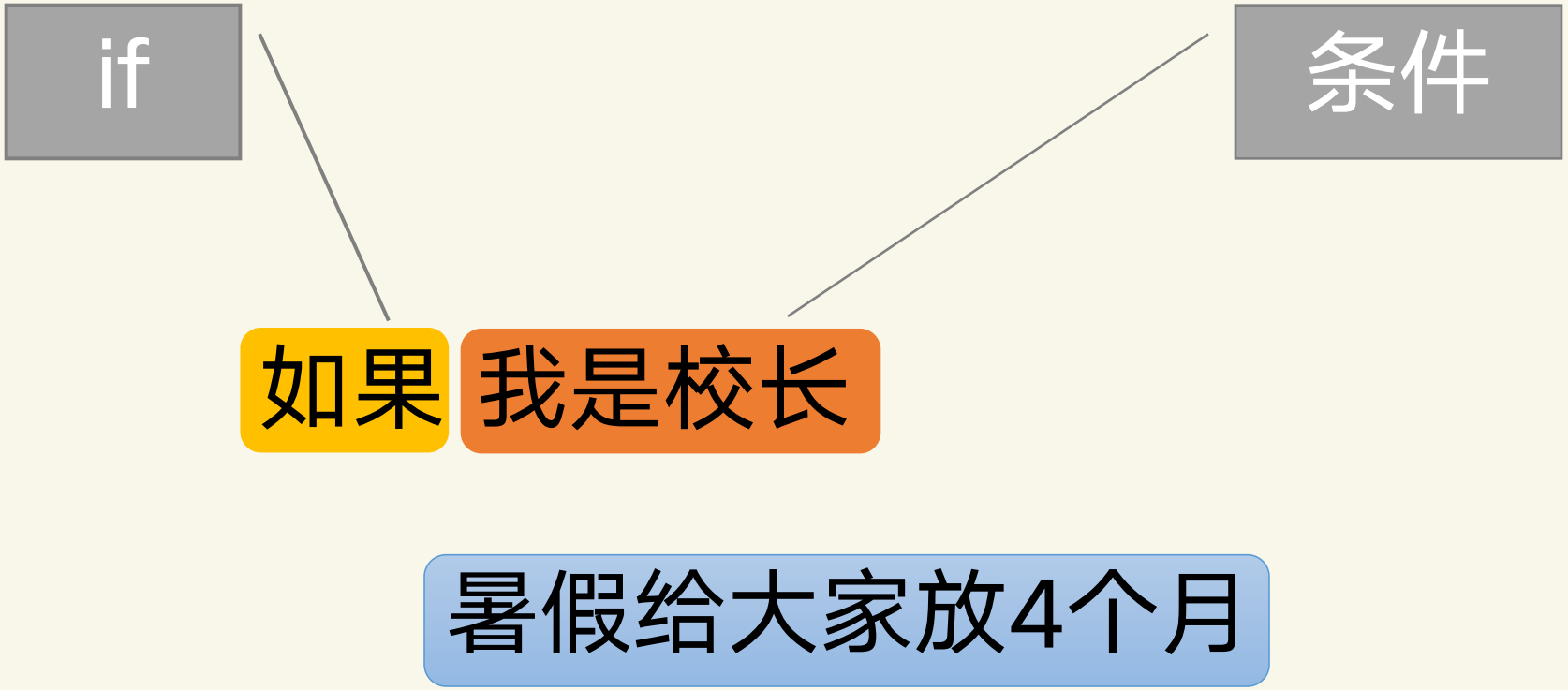
1表示true,即条件成立

0表示false,即条件不成立

Part 2

单分支结构

单分支结构



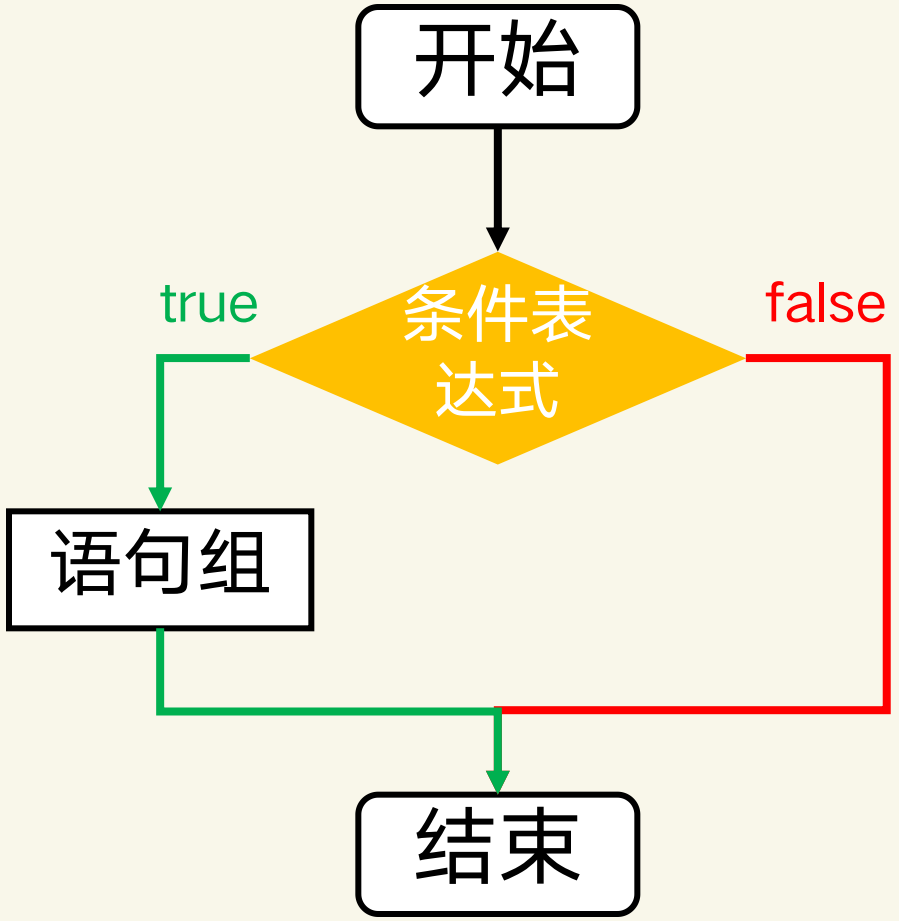
单分支结构

- 单分支框架：

```
if （条件表达式）
{
    语句组;
}
```

单分支结构

- 判断逻辑



单分支结构

想一想2

下列框架表述单分支结构正确的是？

A
```
if [上课]
{
    班主任督查;
}
```

B
```
if (吃了狗粮);
{
    汪汪汪;
}
```

C
```
if (星期天)
{
    我要睡一天;
}
```

单分支结构

偶数概念：能被2整除的数（2的倍数）

程序中：

如果对2求余数，结果为0的

单分支结构

判断偶数

```
int num;
cin >> num;
if (num % 2 == 0)
{
        cout << "偶数";
}
```

单分支结构

奇数概念：不能被2整除的数

程序中：
如果对2求余结果不等于0的数，
那么为奇数

单分支结构

判断奇数

```
int num;
cin >> num;
if (num % 2 != 0)
{
        cout << "奇数";
}
```

单分支结构

- 多个单分支组合：

```
if （条件表达式1）
{
        语句组1;
}
if （条件表达式2）
{
        语句组2;
}
if （条件表达式n）
{
        语句组n;
}
```

单分支结构

判断奇偶数

```
int num;
cin >> num;
if (num % 2 != 0)
{
        cout << "奇数";
}
if (num % 2 == 0)
{
        cout << "偶数";
}
```

单分支结构

想一想3

```
int a, b;
cin >> a >> b;
if (a > b)
{
        cout << a / b;
}
if (a < b)
{
        cout << a * b;
}
```

当左侧代码输入2和4时，则输出的结果是

A、2

B、8

C、6

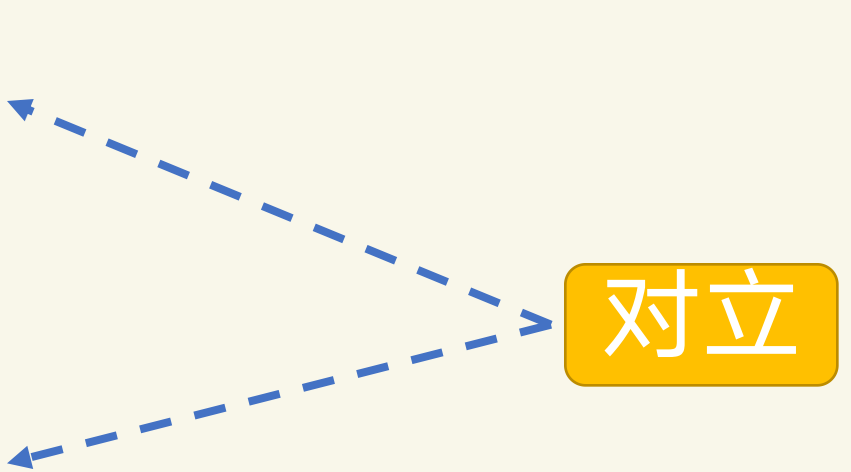
Part 3

双分支结构

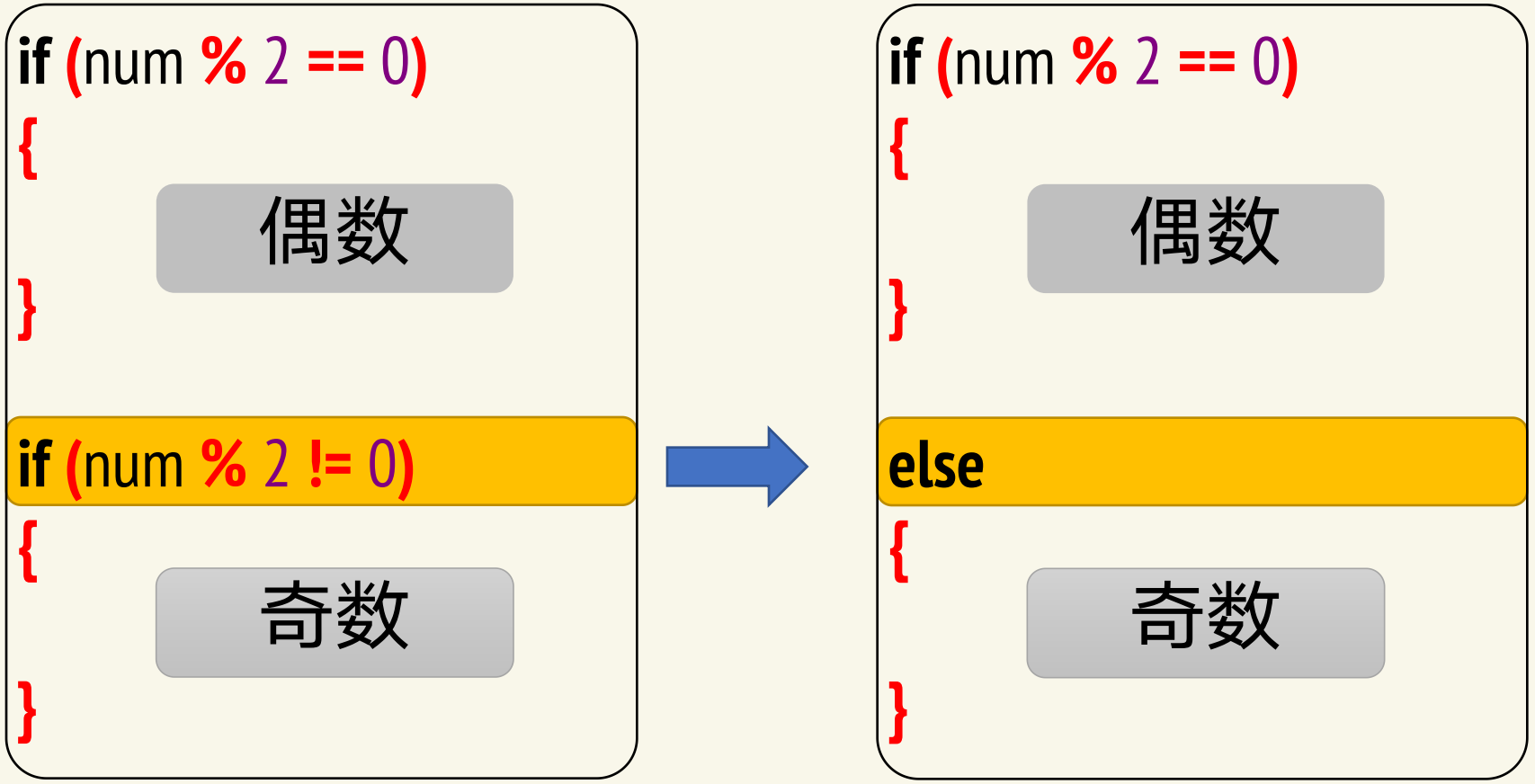
双分支结构

```
if (num % 2 == 0)
{
        偶数
}

if (num % 2 != 0)
{
        奇数
}
```



双分支结构



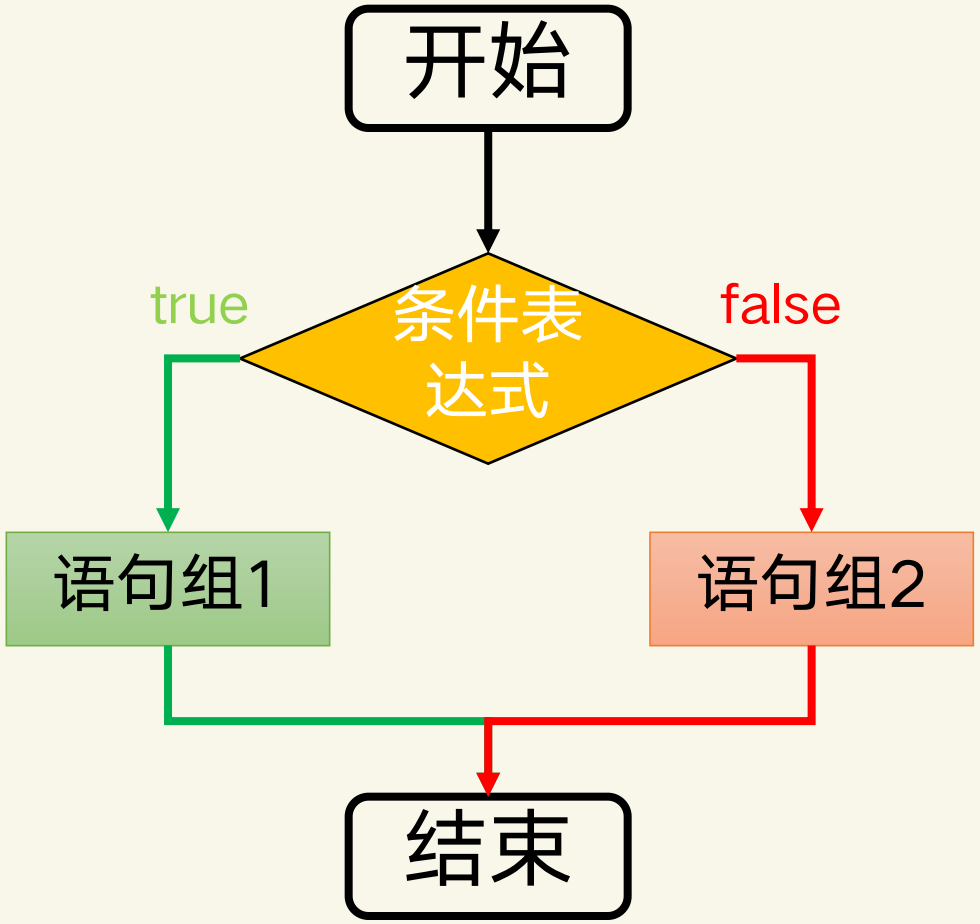
双分支结构

双分支框架：

```
if (条件表达式)
{
        语句组1;
}
else
{
        语句组2;
}
```

双分支结构

- 判断逻辑



双分支结构

想一想4

```
int x;
cin >> x;
if (x % 5 == 0)
{
        cout << x / 10;
}
else
{
        cout << x % 10;
}
```

左侧程序输入19的时候，该程序的结果是

A、19

B、9

C、1

Part 4

分支嵌套

分支嵌套

```
if (条件表达式1)
{
        if (条件表达式2)
        {
                语句组;
        }
}
```

由外到内，条件表达式依次成立后，执行语句组

分支嵌套

下面语句组什么时候会执行？

```
if (n >= 90)
{
        if (n < 100)
        {
                语句组;
        }
}
```

分支嵌套

想一想5

```
int n;
cin >> n;
if (n >= 80)
{
   if (n <= 90)
   {
      cout << "良好";
   }
}
```

左侧程序输入85的时候，该程序的结果是

A、良好
B、无输出内容

Part 5

总结回顾

总结回顾

符号	关系	举例	返回结果类型
>	大于	10 > 8 9 >10	bool型数据
>=	大于或者等于	100 >= 90 100 >= 100	
<	小于	128 < 120 50 < 50	
<=	小于或者等于	10 <= 3 88 <= 88	
==	相等	11 == 11	
!=	不等于	256 != 128	

总结回顾

1、单分支结构

```
if (条件表达式)
{
    语句组;
}
```

2、双分支结构

```
if (条件表达式)
{
    语句组1;
}
else
{
    语句组2;
}
```

3、多个单分支

```
if （条件表达式1）
{
        语句组1;
}
if （条件表达式2）
{
        语句组2;
}
if （条件表达式n）
{
        语句组n;
}
```

4、分支嵌套结构

```
if (条件表达式)
{
    if (条件表达式)
    {
        语句组;
    }
}
```

Class is over