

# 判断

2020年7月10日 14:23

if ( 条件成立 ) {

...

}

没有分号

有分号, 先执行空语句 (if空语句),

再执行下面大括号中的语句块 (普通语句)

```
if ( age > 60 )
{
    salary = salary * 1.2;
    printf("%f\n", salary);
}
```

- 计算两个值之间的关系, 所以叫做关系运算

运算符	意义
==	相等
!=	不相等
>	大于
>=	大于或等于
<	小于
<=	小于或等于

- 当两个值的关系符合关系运算符的预期时, 关系运算的结果为整数1, 否则为整数0

- printf("%d\n", 5==3);
- printf("%d\n", 5>3);
- printf("%d\n", 5<=3);

- 判断是否相等的==和!=的优先级比其他的低, 而连续的关系运算是从左到右进行的

- (5 > 3) == (6 > 4)
- (6 > 5) > 4    1 > 4 → 0
- (a == b) == 6
- a == (b > 0)

- 所有的关系运算符的优先级比算术运算的低, 但是比赋值运算的高

- 7 >= 3 + 4
- int r = (a > 0);

## else

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
} else {
    printf("你的钱不够\n");
}
```

- else = 否则的话

## if语句

- 一个基本的if语句由一个关键字if语句这一行结束的时候并没有表上在括号里的一个表示条件的逻辑表达式, 而后面跟着一对大括号"{}"之间的语句写在if的下一行, 并且缩进语句。如果表示条件的逻辑表达式, 在这一行结束的时候有一个果不是零, 那么就执行后面跟着表示语句结束的";"。这表明这条大括号中的语句, 否则就跳过这条赋值语句是if语句的一部分, if语句不执行, 而继续下面的其他语句。if语句拥有和控制这条赋值语句, 决定它是否要被执行。

if ( total > amount )

total += amount + 10;

```
int a,b,c;
scanf("%d %d %d", &a, &b, &c);
```

```
int max = 0;
```

```
if ( a>b ) {
    if ( a>c ) {
        max = a;
    } else {
        max = c;
    }
} else {
    if ( b>c ) {
        max = b;
    } else {
        max = c;
    }
}
printf("The max is %d\n", max);
```

```
int a,b,c;
scanf("%d %d %d", &a, &b, &c);
```

```
int max = 0;
```

```
if ( a>b )
    if ( a>c )
        max = a;
    else
        max = c;
else
    if ( b>c )
        max = b;
    else
        max = c;
printf("The max is %d\n", max);
```

为什么强调if和else后面要用{}

为了让程序在条件成立时或者不成立时执行一个语句块, 需要在if()之后和else之后使用 {}。

如果在条件成立或不成立时需要执行的语句只有一条, 就没有必要使用 {} 了。

## 嵌套的判断

- 当if的条件满足或者不满足的时候要执行的语句也可以是一条if或if-else语句, 这就是嵌套的if语句

```
if ( code == READY )
    if ( count < 20 )
        printf("一切正常\n");
    else
        printf("继续等待\n");
```

## 缩进

- 缩进格式不能暗示else的匹配

## else的匹配

- else总是和最近的那个if匹配

```
if ( code == READY ) {
    if ( count < 20 )
        printf("一切正常\n");
    } else
        printf("继续等待\n");
```

```
if ( code == READY )
    if ( count < 20 )
        printf("一切正常\n");
    else
        printf("继续等待\n");
```

```
if ( gameover == 0 )
    if ( player2move == 2 )
        printf("Your turn\n");
```

```
if ( gameover == 0 )
    if ( player2move == 2 )
        printf("Your turn\n");
    else
        printf("GAME OVER\n");
```

• 在if或else后面总是用{}

• 即使只有一条语句的时候

## 级联的if-else if

```
if ( exp1 )
    st1;
else if ( exp2 )
    st2;
else
    st3;
```

```
int f;
if ( x < 0 ) {
    f = -1;
} else if ( x == 0 ) {
    f = 0;
} else {
    f = 2 * x;
}
printf("%d", f);
```

单一出口

写程序最好用单一出口

## if语句常见的错误

• 永远在if和else后面加上大括号，即使当时后面只有一条语句

- 忘了大括号
- if后面的分号
- 错误使用==和=
- 使人困惑的else

### 错误使用==和=

- if只要求()里的值是零或非零

```
if ( a = b )
{
    printf("A=B");
}
```

### 代码风格

- 在if和else之后必须加上大括号形成语句块;

- 大括号内的语句缩进一个tab的位置;

注意这里a=b是把b的值赋给a，当b不等于0时，条件永远是对的，执行if语句，当b等于0，条件不成立，不执行if语句

## switch-case

```
switch ( 控制表达式 ) {
case 常量:
    语句
    .....
case 常量:
    语句
    .....
default:
    语句
    .....
}
```

- 控制表达式只能是整型的结果
- 常量可以是常数，也可以是常数计算的表达式
- 根据表达式的结果，寻找匹配的case，并执行case后面的语句，一直到break为止
- 如果所有的case都不匹配，那么就执行default后面的语句；如果没有default，那么就什么都不做

```
switch ( type )
{
case 1:
    printf("你好\n");
    break;
case 2:
    printf("晚上好\n");
    break;
case 3:
    printf("再见\n");
    break;
case 4:
    printf("啊，什么啊？\n");
    break;
default:
    printf("啊，什么啊？\n");
    break;
}
```

## break

- switch语句可以看作是一种基于计算的跳转，计算控制表达式的值后，程序会跳转到相匹配的case（分支标号）处。分支标号只是说明switch内部位置的路标，在执行完分支中的最后一条语句后，如果后面没有break，就会顺序执行到下面的case里去，直到遇到一个break，或者switch结束为止。

type = 1, type=2执行结果一样！！

type = 3: 晚上好 + 再见

type = 4: 再见

```
//printf("输入成绩 (0-100) ");
int grade;
scanf("%d", &grade);
```

## 分支-06. 成绩转换(15)

主页  
题目集  
基本信息  
题目列表  
提交列表  
排名  
帮助

本题要求编写程序将一个百分制成绩转换为五分制成绩。转换规则：

- 大于等于90分为A；
- 小于90且大于等于80为B；
- 小于80且大于等于70为C；
- 小于70且大于等于60为D；
- 小于60为E。

输入格式：

输入在一行中给出1个整数的百分制成绩。

输出格式：

输出在一行中给出1个字符。

输入样例：

85

输出样例：

B

```
grade /=10;
switch ( grade ) {
    case 10:
    case 9:
        printf("A\n");
        break;
    case 8:
        printf("B\n");
        break;
    case 7:
        printf("C\n");
        break;
    case 6:
        printf("D\n");
        break;
    default:
        printf("F\n");
        break;
}
```

这段代码不符合“单一出口”的原则，因为我们还没学过字符或字符串数据的处理

编程时，先想需要几个变量！需要输入输出什么，几个变量  
printf scanf 需要 %d 或者其他，什么什么浮点数什么的  
需要的是整数还是浮点数要想好

Scanf 里面的&不要忘记！！还有不要乱加中文，是要你输入的东西！！

判断的分支括号注意啦

f(x) = -1; x<0  
0; x=0  
2x; x>0

- 这个分段函数能否用switch-case?

```
1. #include<stdio.h>
2. int main()
3. {
4.     int x,y,f;
5.     scanf("%d",&x);
6.     y=x/abs(x);
7.
8.     switch(y){
9.         case -1: f=-1; break;
10.        case 0: f=0; break;
11.        default: f=2*x; break;
12.    }
13.    printf("%d\n",f);
14.    return 0;
15. }
```

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int x=0;
    scanf("%d",&x);
    switch (x<0){
        case 1:x=-1;break;
        case 0:x=2*x;break;
    }
    printf("%d\n", x);
    return 0;
}
```