

# 逻辑运算

# 逻辑运算

- 逻辑运算是对逻辑量进行的运算，结果只有0或1
- 逻辑量是关系运算或逻辑运算的结果

运算符	描述	示例	结果
!	逻辑非	!a	如果a是true结果就是false， 如果a是false结果就是true
&&	逻辑与	a && b	如果a和b都是true，结果就是true； 否则就是false
	逻辑或	a    b	如果a和b有一个是true，结果为true； 两个都是false，结果为false

# TRY

`x > 4 && x < 6`

- 如果要表达数学中的区间，如： $x \in (4,6)$ 或 $x \in [4,6]$ ，应该如何写C的表达式？

像  $4 < x < 6$  这样的式子，不是C能正确计算的式子，因为  $4 < x$  的结果是一个逻辑值（0或1）

- 如何判断一个字符c是否是大写字母?
  - `c >= 'A' && c <= 'Z'`

# 理解一下

- `age > 20 && age < 30`
- `index < 0 || index > 99`
- `! age < 20`

# 优先级

- `! > && > ||`
  - `!done && (count > MAX)`

# 优先级

优先级	运算符	结合性
1	()	从左到右
2	! + - ++ --	从右到左（单目的+和-）
3	* / %	从左到右
4	+ -	从左到右
5	< <= > >=	从左到右
6	== !=	从左到右
7	&&	从左到右
8		从左到右
9	= += -= *= /= %=	从右到左

```

for ( i = a; i<=a+3; i++ ) {
    for ( j=a; j<=a+3; j++ ) {
        for ( k=a; k<=a+3; k++ ) {
            if ( i!=j ) {
                if ( i!=k ) {
                    if ( j!=k ) {
                        cnt++;
                        printf("%d%d%d",i,j,k);
                        if ( cnt == 6 ) {
                            printf("\n");
                            cnt = 0;
                        } else {
                            printf(" ");
                        }
                    }
                }
            }
        }
    }
}

```

```

for ( i = a; i<=a+3; i++ ) {
    for ( j=a; j<=a+3; j++ ) {
        for ( k=a; k<=a+3; k++ ) {
            if ( i!=j && i!=k && j!=k ) {
                cnt++;
                printf("%d%d%d",i,j,k);
                if ( cnt == 6 ) {
                    printf("\n");
                    cnt = 0;
                } else {
                    printf(" ");
                }
            }
        }
    }
}

```



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

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

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

?

# 短路

- 逻辑运算是自左向右进行的，如果左边的结果已经能够决定结果了，就不会做右边的计算

- `a==6 && b==1`

- `a==6 && b+=1`

- 对于`&&`，左边是`false`时就不做右边了
- 对于`||`，左边是`true`时就不做右边了

不要把赋值，包括复合赋值组合进表达式！

# 条件运算

# 条件运算符

- `count = (count > 20) ? count - 10 : count + 10;`
- 条件、条件满足时的值和条件不满足时的值

```
if ( count > 20 )  
    count = count - 10;  
else  
    count = count + 10;
```

# 优先级

- 条件运算符的优先级高于赋值运算符，但是低于其他运算符

`m < n ? x : a + 5`

`a++ >= 1 && b-- > 2 ? a : b`

`x = 3 * a > 5 ? 5 : 20`

# 嵌套条件表达式

- `count = (count > 20) ? (count < 50) ? count - 10 : count - 5 : (count < 10) ? count + 10 : count + 5;`

- 条件运算符

我们不希望你使用嵌套的条件表达式

- `w < x ? x + w : x < y ? x : y`

逗号运算

# 逗号运算符

- 逗号用来连接两个表达式，并以其右边的表达式的值作为它的结果。逗号的优先级是所有的运算符中最低的，所以它两边的表达式会先计算；逗号的组合关系是自左向右，所以左边的表达式会先计算，而右边的表达式的值就留下来作为逗号运算的结果。



# 在for中使用,

- `for ( i=0, j=10; i<j; i++, j- - ) .....`