关系运算符与逻辑运算符

March 21, 2017

1 关系运算符

关系运算符用于比较两个值。

1. 运算符 == 检查两个给定的操作数是否相等。若相等，返回true；否则返回false。如 5 == 5 返回true。
2. 运算符 != 检查两个给定的操作数是否相等。若不相等，返回true；否则返回false。如 5 != 5 返回false。
3. 运算符 > 检查第一个操作数是否大于第二个操作数。若成立，返回true；否则返回false。如 6 > 5 返回true。
4. 运算符 < 检查第一个操作数是否小于第二个操作数。若成立，返回true；否则返回false。如 6 < 5 返回false。
5. 运算符 >= 检查第一个操作数是否大于或等于第二个操作数。若成立，返回true；否则返回false。如 5 >= 5 返回true。
6. 运算符 <= 检查第一个操作数是否小于或等于第二个操作数。若成立，返回true；否则返回false。如 5 <= 5 返回true。

```
// C program to demonstrate working of relational operators
#include <stdio.h>

int main()
{
    int a=10, b=4;

    // relational operators
    // greater than example
    if (a > b)
        printf("a␣is␣greater␣than␣b\n");
    else printf("a␣is␣less␣than␣or␣equal␣to␣b\n");

    // greater than equal to
    if (a >= b)
        printf("a␣is␣greater␣than␣or␣equal␣to␣b\n");
    else printf("a␣is␣lesser␣than␣b\n");

    // less than example
    if (a < b)
        printf("a␣is␣less␣than␣b\n");
    else printf("a␣is␣greater␣than␣or␣equal␣to␣b\n");

    // lesser than equal to
    if (a <= b)
```

```
        printf("a is lesser than or equal to b\n");
    else printf("a is greater than b\n");

    // equal to
    if (a == b)
        printf("a is equal to b\n");
    else printf("a and b are not equal\n");

    // not equal to
    if (a != b)
        printf("a is not equal to b\n");
    else printf("a is equal b\n");

    return 0;
}
```

```
Output:
a is greater than b
a is greater than or equal to b
a is greater than or equal to b
a is greater than b
a and b are not equal
a is not equal to b
```

2 逻辑运算符

逻辑运算符用于连接两个及以上条件，或对原条件取补。

1. 逻辑与： 当两个条件同时满足时，运算符 && 返回true；否则返回 false。如，当 a 和 b 均为 true (即非零)时，a && b 返回true。
2. 逻辑或：当至少有一个条件满足时，运算符 || 返回true；否则返回 false。如，当 a 和 b 至少有一个为 true (即非零)时，a || b 返回true。当然，当 a 和 b 均为 true 时， a || b返回true。
3. 逻辑非：当条件不满足时，运算符 ! 返回 true ；否则返回 false 。如，若 a 为 false 时，a 返回 true。

```
// C program to demonstrate working of logical operators
#include <stdio.h>

int main()
{
    int a=10, b=4, c = 10, d = 20;

    // logical operators

    // logical AND example
    if (a>b && c==d)
        printf("a is greater than b AND c is equal to d\n");
    else printf("AND condition not satisfied\n");

    // logical AND example
    if (a>b || c==d)
        printf("a is greater than b OR c is equal to d\n");
    else printf("Neither a is greater than b nor c is equal "
                "to d\n");
```

```
    // logical NOT example
    if (!a)
        printf("a is zero\n");
    else printf("a is not zero");

    return 0;
}
```

```
AND condition not satisfied
a is greater than b OR c is equal to d
a is not zero
```

3 逻辑运算符中的短路现象

对于逻辑与，若第一个操作数为 false ，则第二个操作数将不会被计算。如以下程序将不会打印 Hello World。

```
#include <stdio.h>
#include <stdbool.h>
int main()
{
    int a=10, b=4;
    bool res = ((a == b) && printf("Hello World"));
    return 0;
}
```

但下面的程序将打印 Hello World。

```
#include <stdio.h>
#include <stdbool.h>
int main()
{
    int a=10, b=4;
    bool res = ((a != b) && printf("GeeksQuiz"));
    return 0;
}
```

对于逻辑或，若第一个操作数为 true ，则第二个操作数不会被计算。如以下程序不会打印 Hello World。

```
#include <stdio.h>
#include <stdbool.h>
int main()
{
    int a=10, b=4;
    bool res = ((a != b) || printf("GeeksQuiz"));
    return 0;
}
```

但下面的程序将打印 Hello World。

```
#include <stdio.h>
#include <stdbool.h>
int main()
{
    int a=10, b=4;
    bool res = ((a == b) || printf("GeeksQuiz"));
    return 0;
}
```