
浮点数怎么做比较



华清远见 | 创客学院 小美老师

浮点数怎么做比较

- 浮点数存储的时候，存在精度损失，大多浮点数不是一个精确表示的数。
- 在浮点数比较的时候，可不可以使用`==`进行比较？

测试代码-1

```
#include <stdio.h>
int main() {
    double a = 1.0;
    double b = 0.9;
    if ((a - b) == 0.1) {
        printf("equal\n");
    }
    else {
        printf("not equal\n");
    }
    return 0;
}
```

思考： $a - b = 1.0 - 0.9 = 0.1$ 即程序运行成功后会输出equal，那么输出结果为是否为这样呢？

```
$ gcc float_demo6.c -Wall
$ ./a.out
not equal
```

测试代码-2

为了更好地进行代码分析，将 $a - b$ 的值、 0.1 的值都打印

```
#include <stdio.h>
int main() {
    double a = 1.0;
    double b = 0.9;
    printf("%.30lf\n", a-b);
    printf("%.30lf\n", 0.1);
    return 0;
}
```

```
$ gcc float_demo7.c -Wall
$ ./a.out
0.0999999999999999977795539507497
0.1000000000000000005551115123126
```

$a - b \neq 0.1$

浮点数在存储的过程中存在精度损失，会导致结果存在细微的差别。

浮点数跟零值进行比较的方案

- 引入绝对值函数

```
#include <math.h>  
double fabs(double x);  
Link with -lm.
```

绝对值函数-示例

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    double n = -3.14159;
```

```
    n = fabs(n);
```

```
    printf("%lf\n",n);
```

```
    return 0;
```

```
}
```

```
$ gcc abs1.c -lm
```

```
$ ./a.out
```

```
3.141590
```

浮点数跟零值进行比较

```
#include <stdio.h>
#include <math.h>
int main() {
    double a = 1.0;
    double b = 0.9;
    double t = fabs((a - b) - 0.1);

    if (t < 0.0000000000000001 ){
        printf("equal\n");
    }
    else {
        printf("not equal\n");
    }
    return 0;
}
```

```
$ gcc float_demo8.c -Wall
6$ ./a.out
equal
```

自定义精度

待比较的两个浮点数做减法，差值的绝对值小于精度，认为相等，反之不等

扫一扫，获取更多信息



THANK YOU