double类型是双精度浮点数，占用8字节（Byte）即64位（bit），其精度是由32个bit的二进制尾数来确定的，因此准确精度是二进制精度而不是十进制精度，通常可以保证十进制小数点后15位有效精度和第16位的部分精度

System.out.printf("%f",d);//"f"表示格式化输出浮点数。

System.out.printf("%9.2f",d);//"9.2"中的9表示输出的长度，2表示小数点后的位数。

System.out.printf("%+9.2f",d);//"+"表示输出的数带正负号。

System.out.printf("%-9.4f",d);//"-"表示输出的数左对齐（默认为右对齐）。

System.out.printf("%+-9.3f",d);//"+-"表示输出的数带正负号且左对齐。

System.out.printf("%d",i);//"d"表示输出十进制整数。

System.out.printf("%o",i);//"o"表示输出八进制整数。

System.out.printf("%x",i);//"d"表示输出十六进制整数。

System.out.printf("%#x",i);//"d"表示输出带有十六进制标志的整数。

System.out.printf("%s",s);//"d"表示输出字符串。

System.out.printf("输出一个浮点数：%f，一个整数：%d，一个字符串：%s",d,i,s);//可以输出多个变量，注意顺序。

System.out.printf("字符串：%2$s，%1$d的十六进制数：%1$#x",i,s);//"X$"表示第几个变量。

**举例1：控制输出小数位（注意这里是printf）**

|  |
| --- |
| **double** d = 3.234566;  System.***out***.printf("%.2f",d); //输出结果：3.23 |

举例2：控制输出小数位：

|  |
| --- |
| DecimalFormat df=**new** DecimalFormat(".##"); // 定义一个DecimalFormat类，能够将小数位保留位数输入  **double** d = 12.3456; //定义double类型变量d  System.***out***.println(df.format(d));//按照“.##”的格式，将d输出 结果为：12.35 |