

第十节 输出格式控制printf

赵冯平



输出格式控制printf函数——命令格式

● `printf("<格样化字符串>", <表达式表>);`

- <格样化字符串> 包括两种内容:

正常字符, 这些字符将按原样输出;

格式控制符 (占位符), 以"%"开头, 后跟一个或几个规定字符, 用来确定输出表达式的格式。

- <表达式表>是用“, ”号分开的多个表达式 (算式、变量、常量), 格式控制符(占位符)的个数要与表达式的个数一致。



输出格式控制printf函数——示例

例中双引号内，红色是占位符（格式规定符），黑色是正常字符，\n是换行符

- `printf("weight=%.3lf\n",w);`
- `printf("%d:%02d:%02d:%02d:%03d\n",d,h,m,s,ms);` // %02d 输出不足2位时，前面补0
- `printf("%d %d\n",c,n);` //按整数输出，%d %d中间的两个空格原样输出，2个占位符对应c,n
- `printf("%c %c\n",c,n);` // %c格式符控制按字符输出



```

#include <iostream>
using namespace std;
int main()
{
    double w=1234.56789;
    printf("%.2lf\n",w); //在格式符%.2lf位置按2位小数输出变量w的值, \n是换行符, 使后面输出在新行上
    printf("weight=%.3lf\n",w); //在格式符%.3lf位置按3位小数输出 w, 原样输出: weight=
    double x=1234.56789;
    printf("%g\n",x); //最多输出6位有效数字。 \n的作用是换行
    double y=1234567.89;
    printf("%g\n",y); //6位有效数字不能表达, 按科学计数法输出
    double z=12.34;
    printf("%g\n",z); //最多输出6位有效数字, 实际输出4位
    int d=2,h=4,m=28,s=8,ms=74; //天、小时、分、秒、毫秒
    printf("%d %d:%d:%d:%d\n",d,h,m,s,ms); // %d按整数输出的格式符, 5个格式符对应5个变量
    printf("%d:%2d:%2d:%2d:%3d\n",d,h,m,s,ms); // %3d 指定输出占3位数的位宽, 原样输出冒号 :
    printf("%d:%02d:%02d:%02d:%03d\n",d,h,m,s,ms); // %02d 输出不足2位时, 前面补0
    char c; int n;
    c=65; n='b';
    printf("%d %d\n",c,n); //按整数输出, %d %d中间的两个空格原样输出, 2个占位符, 对应c,n
    printf("%c %c\n",c,n); // %c格式符控制按字符输出
    printf("$%.2lf %.2lf%%\n",w,z); //两个格式符%.2lf %.2lf控制w,z输出格式。 %有特殊用途, %%输出%
    return 0;
}

```

输出

1234.57
weight=1234.568

1234.57

1.23457e+006

12.34

2 4:28:8:74
2: 4:28: 8: 74
2:04:28:08:074

65 98

A b

\$1234.57 12.34%

再见！

