

个人资料



_John_Tian_



访问： 420428次
积分： 7096
等级： BLOG > 6
排名： 第2799名

原创： 282篇
转载： 84篇
译文： 4篇
评论： 120条

文章搜索

Remark

JohnTian's Blog
JohnTian
Github
JohnTian

博客专栏



机器学习
文章： 6篇
阅读： 5048



VB实践学习
文章： 14篇
阅读： 14238



python视频学习
文章： 21篇
阅读： 23227



opencv例题学习
文章： 56篇
阅读： 79755

C/C++中printf和C++中cout的输出格式

标签： printf cout

2013-06-30 20:26 1352人阅读 评论(0) 收藏 举报

分类： C++/C (58)

一、Printf 输出格式

C中格式字符串的一般形式为： %[标志][输出最小宽度][.精度][长度]类型，其中方括号[]中的项为可选项。各项的意义介绍如下：

1.类型类型字符用以表示输出数据的类型，其格式符和意义下表所示：

表示输出类型的格式字符	格式字符意义
a (C99)	浮点数、十六进制数字和p-计数法
A (C99)	浮点数、十六进制数字和p-计数法
c	输出单个字符
d	以十进制形式输出带符号整数(正数不输出符号)
e	以指数形式输出单、双精度实数
E	以指数形式输出单、双精度实数
f	以小数形式输出单、双精度实数
g	以%f%e中较短的输出宽度输出单、双精度实数,%e格式在指数小于-4或者大于等于精度时使用
G	以%f%e中较短的输出宽度输出单、双精度实数,%e格式在指数小于-4或者大于等于精度时使用
i	有符号十进制整数(与%d相同)
o	以八进制形式输出无符号整数(不输出前缀O)
p	指针
s	输出字符串
x	以十六进制形式输出无符号整数(不输出前缀OX)
X	以十六进制形式输出无符号整数(不输出前缀OX)
u	以十进制形式输出无符号整数

```
#include "stdio.h"
#include "conio.h"
main()
{
```

文章分类

- HEVC (29)
- JMVC (4)
- HTM (10)
- GNU/Linux (23)
- C++/C (59)
- JAVA (12)
- OpenCV (76)
- Python (56)
- MATLAB (14)
- VB (14)
- 科技相关 (16)
- Favorite (7)
- DataBase (7)
- JavaScript (3)
- MachineLearning (12)
- Algorithm (33)
- golang (2)

文章存档

- 2017年05月 (2)
- 2017年03月 (4)
- 2017年02月 (4)
- 2017年01月 (5)
- 2016年12月 (5)

展开

最新评论

- VSR3.5配置问题, error LINK2**
_John_Tian_ : @hxwab:1. lib库文件的路径未找到; 2. lib库文件的生成环境不同, x86和x64
- 《OpenCV2 计算机视觉编程手册》
_John_Tian_ : @xyxd190:请帖出报错信息。视频无法正常播放,有可能是解码的问题。
- 《OpenCV2 计算机视觉编程手册》
xyxd190: 博主, 你好, 我测试了下, 视频无法正常播放, 代码运行的了, 不知道哪里出现问题
- 《OpenCV2 计算机视觉编程手册》
xyxd190: 博主, 你好, 我测试了下, 视频无法正常播放, 代码运行的了, 不知道哪里出现问题
- OpenCV && Tesseract-OCR in /
_John_Tian_ : @guanzenbin00: 源码都在博客文章中的哦。旋转识别的话, 我没有尝试过哦。如果是整个工程的代...
- OpenCV && Tesseract-OCR in /
guanzenbin00: 源码可以分享一下吗? 不知道支持 旋转后的识别吗?
- OpenCV && Tesseract-OCR in /
_John_Tian_ : @DJY1992:谢谢!
- OpenCV && Tesseract-OCR in /
奥特曼超人: 恭喜您, 文章已被推荐收录。
- 测试序列下载地址
cheng_alan: @tianzhaixing:这个是HEVC的 和SCC的不一样, 能找到SCC的吗
- CU分割信息提取(更新续)
_John_Tian_ :
@zhangchun3211:是的

```
printf("The program test print style!\n");
printf("%d\n" , 223);
printf("%d\n" , -232);
printf("\n");

printf("%o\n" , 223);
printf("%o\n" , -232);
printf("\n");

printf("%x\n" , 223);
printf("%x\n" , -232);
printf("\n");

printf("%u\n" , 223);
printf("%u\n" , -232);
printf("\n");

printf("%f\n" , 223.11);
printf("%f\n" , 232.11111111);
printf("%f\n" , -223.11);
printf("%f\n" , -232.11111111);
printf("\n");

printf("%e\n" , 223.11);
printf("%e\n" , 232.11111111);
printf("%e\n" , -223.11);
printf("%e\n" , -232.11111111);
printf("\n");

printf("%g\n" , 223.11);
printf("%g\n" , 232.111111111111);
printf("%g\n" , -223.11);
printf("%g\n" , -232.111111111111);
printf("\n");

printf("%c\n" , 'a');
printf("%c\n" , 97);
printf("\n");

printf("%s\n" , "this is a test!");
printf("%s\n" , "2342o34uo23u");
printf("\n");
getch();
}
```

2.标志

标志字符为-、+、#、空格和0五种，其意义下表所示：

关闭

标志格式字符	标志 意义
-	结果左对齐, 右边填充空格
+	输出符号(正号或负号)
空格	输出值为正时冠以空格, 为负时冠以负号
#	对c, s, d, u类无影响; 对o类, 在输出时加前缀0; 对x类, 在输出时加前缀0x或者0X; 对g, G 类防止尾随0被删除;
	对于所有的浮点形式, #保证了即使不跟任何数字, 也打印一个小数点字符
0	对于所有的数字格式, 用前导0填充字段宽度, 若出现-标志或者指定了精度(对于整数), 忽略

3.输出最小宽度

用十进制整数来表示输出的最少位数。若实际位数多于定义的宽度, 则按实际位数输出, 若实际位数少于定义的宽度则补以空格或0。

```
#include "stdio.h"
#include "conio.h"
main()
{

    printf("%*-10d*\n", 223);
    printf("%*%+10d*\n", -232);
    printf("%*-*\n", 223);
    printf("%*%#d*\n", -232);
    printf("\n");
    getch();

    printf("%*-10o*\n", 223);
    printf("%*%+10o*\n", -232);
    printf("%*%o*\n", 223);
    printf("%*%#o*\n", -232);
    printf("\n");
    getch();

    printf("$%-10x$\n", 223);
    printf("$0x$\n", -232);
    printf("$% x$\n", 223);
    printf("$%#x$\n", -232);
    printf("\n");

    printf("%*-10u*\n", 223);
    printf("%*%+10u*\n", -232);
    printf("%% u*\n", 223);
    printf("%%#u*\n", -232);
    printf("\n");
    getch();
}
```

```

printf("%-10f\n" , 223.11);
printf("%+10f\n" , 232.11111111);
printf("% f\n" , -223.11);
printf("%#f\n" , -232.11111111);
printf("\n");
getch();

printf("%-10e\n" , 223.11);
printf("%+10e\n" , 232.11111111);
printf("% e\n" , -223.11);
printf("%#e\n" , -232.11111111);
printf("\n");
getch();

printf("%-10g\n" , 223.11);
printf("%+10g\n" , 232.111111111111);
printf("% g\n" , -223.11);
printf("%#g\n" , -232.111111111111);
printf("\n");
getch();

printf("%-10c\n" , 'a');
printf("%+10c\n" , 97);
printf("% c\n" , 'a');
printf("%#c\n" , 97);
printf("\n");
getch();

printf("%-20s\n" , "this is a test!");
printf("%+20s\n" , "2342o34uo23u");
printf("% 20s\n" , "this is a test!");
printf("%#s\n" , "2342o34uo23u");
printf("\n");
getch();
}

```

4.精度

精度格式符以“.”开头，后跟十进制整数。本项的意义是：如果输出数字，则表示小数的位数；如果输出的是字符，则表示输出字符的个数；若实际位数大于所定义的精度数，则截去超过的部分。

```

#include "stdio.h"
#include "conio.h"
main()
{
    printf("%.3d\n" , 5555);
}

```

```

    getch();
    printf("%.3f\n", 0.88888);
    getch();
    printf("%.3f\n", 0.9999);
    getch();
    printf("%.4s\n", "this is a test!");
    getch();
}

```

5. 长度

长度格式符为h,l两种，h表示按短整型量输出，l表示按长整型量输出。

h和整数转换说明符一起使用，表示一个short int 或者unsigned short int类型的数值，示例：

```
%hu, %hx, %6.4hd
```

hh和整数转换说明符一起使用，表示一个short int 或者unsigned short类型的数值，示例：

```
%hhu, %hhx, %6.4hhd
```

j和整数转换说明符一起使用，表示一个intmax_t或者uintmax_t类型的数值，示例：

```
%jd, %8jx
```

l和整数转换说明符一起使用，表示一个long int 或者unsigned long int类型的数值，示例：

```
%ld, %8lu
```

ll和整数转换说明符一起使用，表示一个long int 或者unsigned long int类型的数值(C99)，示例：

```
%lld, %8llu
```

L和浮点转换说明符一起使用，表示一个long double的值，示例：%Lf, .4Le

t和整数转换说明符一起使用，表示一个ptrdiff_t值(两个指针之间的差相对应的类型)(C99)，示例：

```
%td, ti
```

z和整数转换说明符一起使用，表示一个size_t值(sizeof返回的类型)(C99)，示例：

```
%zd, zx
```

```

main(){
int a=15;
float b=138.3576278;
double c=35648256.3645687;
char d='p';
printf("a=%d,],%o,%x\n",a,a,a,a);
printf("b=%f,%lf,%5.4lf,%e\n",b,b,b,b);
printf("c=%lf,%f,%8.4lf\n",c,c,c);
printf("d=%c,◆\n",d,d);
}
a<--15
b<--138.3576278
c<--35648256.3645687
d<--'p'
main()

```

```

{
    int a=29;
    float b=1243.2341;
    double c=24212345.24232;
    char d='h';
    printf("a=%d,],%o,%x\n",a,a,a,a);

    printf("b=%f,%lf,%5.4lf,%e\n",b,b,b,b);

    printf("c=%lf,%f,%8.4lf\n",c,c,c);

    printf("d=%c,◆\n",d,d);
    getch();
}

```

使用printf函数时还要注意一个问题，那就是输出表列中的求值顺序。不同的编译系统不一定相同，可以从左到右，也可从右到左。**Turbo C**是按从右到左进行的

```

main(){
    int i=8;
    printf("%d\n%d\n%d\n%d\n%d\n",++i,--i,i--,i++,-i--);
}

```

6.特殊用法

对于m.n的格式还可以用如下方法表示（例）

```

char ch[20];
printf("%*.*s\n",m,n,ch);

```

前边的*定义的是总的宽度，后边的定义的是输出的个数。分别对应外面的参数m和n。我想这种方法的好处是可以在语句之外对参数m和n赋值，从而控制输出格式。

今天(06.6.9)又看到一种输出格式 %n 可以将所输出字符串的长度值赋给一个变量，见下例：

```

int slen;
printf("hello world%n", &slen);

```

执行后变量被赋值为11。

二、c++ cout 输出格式

在c++程序里面经常见到下面的头文件

```
#include <iomanip>
```

io代表输入输出，manip是manipulator（操纵器）的缩写

iomanip的作用：

主要是对cin,cout之类的一些操纵运算符，比如setfill,setw,setbase,setprecision等等。它是I/O流控制头文件,就像C里面的格式化输出一样.以下是一些常见的控制函数的：

dec 置基数为10 相当于"%d"

hex 置基数为16 相当于"%X"

oct 置基数为8 相当于"%o"

setfill('c') 设填充字符为c

setprecision(n) 设显示小数精度为n位

setw(n) 设域宽为n个字符

这个控制符的意思是保证输出宽度为n。如：

```
cout << setw( 3 ) << 1 << setw( 3 ) << 10 << setw( 3 ) << 100 << endl;
```

输出结果为

1 10100 （默认是右对齐）当输出长度大于3时(<<1000)，setw(3)不起作用。

▲setw(n)用法：通俗地讲就是预设宽度

如 cout<<setw(5)<<255<<endl;

结果是：

(空格)(空格)255

▲setfill(char c) 用法：就是在预设宽度中如果已存在没用完的宽度大小，则用设置的字符c填充

如 cout<<setfill('@')<<setw(5)<<255<<endl;

结果是：

@@255

▲setbase(int n)：将数字转换为 n 进制。

如 cout<<setbase(8)<<setw(5)<<255<<endl;

cout<<setbase(10)<<setw(5)<<255<<endl;

cout<<setbase(16)<<255<<endl;

结果是：

(空格) (空格)377

(空格)(空格) 255

(空格)(空格) f f

▲ setprecision用法

使用setprecision(n)可控制输出流显示浮点数的数字个数。C++默认的流输出数值有效位是6。

如果setprecision(n)与setiosflags(ios::fixed)合用，可以控制小数点右边的数字个数。setiosflags(ios::fixed)是用定点方式表示实数。

如果与setiosnags(ios::scientific)合用，可以控制指数表示法的小数位数。setiosflags(ios::scientific)是用指数方式表示实数。

setiosflags(ios::fixed) 固定的浮点显示

setiosflags(ios::scientific) 指数表示

setiosflags(ios::left) 左对齐

setiosflags(ios::right) 右对齐

setiosflags(ios::skipws) 忽略前导空白

setiosflags(ios::uppercase) 16进制数大写输出

setiosflags(ios::lowercase) 16进制小写输出

setiosflags(ios::showpoint) 强制显示小数点

setiosflags(ios::showpos) 强制显示符号

举例：

```
#include <iostream.h>
```

```
#include <iomanip.h>
```

```
using namespace std;
```

```
int main()
{
    cout<<12345.0<<endl;//输出"12345"
    cout<<setiosflags(ios::fixed)<<setprecision(3)<<1.2345<<endl;输出"1.235"
    cout<<setiosflags(ios::scientific)<<12345.0<<endl;//输出"1.234500e+004 "
    cout<<setprecision(3)<<12345.0<<endl;//输出"1.235e+004"（1.235e+004应改为1.23e+004）
    return 0;
}
```

<http://blog.csdn.net/tjunxin/article/details/8163635>

顶

0

踩

0

上一篇 计算机视觉期末报告---项目二


下一篇 摄像机标定：像素焦距与毫米焦距转换

相关文章推荐


- C++常见问题解答
- C++常见问题解答
- log4cpp的配置
- GCC新手入门
- gcc

- 从 Solaris 向 Linux 移植应用程序的技术指导版本 10
- GCC 中文手册上
- 深入了解scanfgetchar和gets等函数转
- CC++中cout和printf的运行机制


参考知识库



iOS知识库
4568 关注 | 1563 收录



.NET知识库
3883 关注 | 839 收录



Swift知识库
3934 关注 | 1239 收录

猜你在找

- Visual Studio 2015开发C++程序的基本使用

深入浅出C++程序设计（基础篇）

C++程序设计

Swift与Objective-C\C\C++混合编程

C++语言基础
- 零基础学会在Linux上编译调试C++项目

C++标准模板库从入门到精通

C++面试笔试精要

C++ DirectX 游戏开发初级教程

Android JNI专题高级实战 安卓项目整合C/C++网络通

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) [webmaster@csdn.net](#) 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 