

02 | 第一个程序：教你输出彩色的文字

2020-01-07 胡光

人人都能学会的编程入门课

[进入课程 >](#)



讲述：胡光

时长 17:36 大小 14.11M



你好，我是胡光。从今天开始，我将带你正式进入 C 语言的编程世界。我们总会听到这样一种说法：兴趣是最好的老师。引起你对编程的兴趣，就是今天这讲的目的。如果你之前对于 C 语言的认知还停留在黑白纯色阶段的话，今天就让我们一起来进入一个彩色的 C 语言世界，来，跟我一起给它点儿颜色！

今日任务

先来看看今天这 10 分钟，我们要完成的任务。今天你将会学到如何设计一个读入字符串，并且按照字符串信息中规定的颜色，输出相应颜色文本内容的方法程序。

例如当我们：

输入：red:color content, 输出红色的：color content

输入：blue:color content, 输出蓝色的：color content

输入：yellow:color content, 输出黄色的：color content

下面这张图呢，就是当你完成这个任务时，你程序应该有的一个输出效果：

```
red:hello world
hello world
blue:hello world
hello world
yellow:hello world
hello world
```


那么想完成这个任务，我们都需要做哪些准备呢？又有哪些概念需要我们理解和掌握的？请你多点耐心，听我一步步讲。

必知必会，查缺补漏

1. 输出函数：让计算机“说出话”


我还记得我当年学习 C 语言的时候，最兴奋的地方就是计算机可以根据我的“指令”打印出一句我想说的话来。这个功能虽然简单，但它也预示着我们可以初步“控制”计算机了。下面我附了一段代码，代码后面有相应的中文说明，你可以先看看。

```
1 #include <stdio.h> // 文件包含，之所以能使用 printf，跟这行代码有关
2
3 int main() {        // 主函数部分，整个程序的入口位置
4     printf("hello my friend!"); // 打印函数
5     return 0;       // 暂不介绍，不是重点
6 }
```

 复制代码

不知道你有没有理解这段代码，我多啰嗦两句。上面代码中，// 后面的内容属于注释，它是用来说明代码功能的，不属于程序部分，而且就算写在程序里面，也不会影响程序逻辑的正确性。

如果你在编译器中运行上面这段程序，就会看到如下一段输出：

 复制代码


```
1 hello my friend!
```

恭喜你，现在你已经成功与我们的计算机朋友打招呼了，这是一个好的开始，不是吗？

printf 函数就是所谓的“输出函数”，现在你可以尝试在 printf 函数的双引号中间换一些其他内容，来试试效果了。但要注意，printf 后面的括号和双引号（且是英文输入法环境下）是必须要有的，其中双引号里面的内容才是最后程序输出的内容。至于为什么是这样，你也不用想，一开始，先死记住就可以了，或者换句话说，这就是规则。有些规则，本来就没有为什么。


我学习计算机的时候，我的老师就让我把上面的代码敲了 N 遍，最后都成了肌肉记忆。现在想想，也是这么回事。

上面那段代码，如果你玩够了的话，可以将 printf 函数换成下面的内容，看看输出结果：

 复制代码

```
1 printf("Hi, my friend:\n\tthis is the first day I know you.");
```

你所看到的输出内容，应该与下面这段内容相似：

 复制代码

```
1 Hi, my friend:
2     this is the first day I know you.
```

我们看到有了换行效果，且第二行开头有了缩进。

你要是使用过 Word 的话，应该知道 Tab 键吧，对，\t 的效果就相当于在相应位置敲击了 Tab 键，\n 的效果就相当于敲击了换行。在 C 语言中，\t、\n 都属于转义字符，还是和上面一样，它是 C 语言定义的规则，你也先不用问为什么，记住它，然后多用几次就可以了。下面这个表里，我还给出了一些更多的转义字符，你可以拿来玩一玩。

转义字符	名称	解释
\n	换行符	用于输出内容的换行
\t	Tab 字符	相当于敲击了Tab键
\a	响铃	在不扰民的前提下，打开外放，聆听来自电脑的怒吼
\b	退格	这个效果比较有趣，你试试就知道了
\033	设置颜色	用这个，给C语言点儿颜色

2. 类型与变量：组织语言让计算机理解你的意思

现在我们来假设一个场景，在一片硝烟弥漫的战场上，你身处其中，需要将战况传回指挥部，以便指挥官做出下一步的战斗指示，你可能会将如下信息回传：

报告指挥官，敌军兵力大致如下：
重步兵：100 人
坦克：4 辆
火箭炮：6 门
报告完毕，请总部做出下一步战斗指示！

这个场景中，100、4、6 都是对抽象的战场环境的客观且具体的数字描述。

现实世界就像这个战场一样，是抽象的，而计算机的世界是具体的、可计算的。从现在开始，你应该注意学习如何将现实世界的“**信息**”，映射到计算机世界中“**数据**”的技巧。

下面就来看现实生活中几种常见的信息，以及相应信息在计算机中的表示：


现实生活	计算机中	类型解释
这里有107家时髦的金拱门	int	整数类型
艾伦巨人身高15.3米	float/double	浮点数类型
他的名字叫Ticki	char[]	字符串类型，用来存储文本信息

在上表中，金拱门有多少家，是一个整数，因为不可能出现 0.5 这样的半家，所以在计算机中表示为 int 的整数类型；巨人的身高则有零有整，所以在计算机中表示为 float 或 double 的浮点数类型；而一个人的名字就不能用数字类型来表示了，而是采用字符串类型来存储。

可以看到，我们说到的这几种基础数据类型，用来代表不同种类的信息。

在现实生活中，你可能会把各种信息记录在纸条上，或者本子的某个地方。在程序中，我们把这些信息，记录在一些叫做“变量”的东西里面。注意，类型和变量是两个完全不同的概念。

下面我们来看一个简单的变量示例：

 复制代码

```
1 #include <stdio.h>
2
3 int main() {
4     int a = 167, b = 543;    // 定义变量 a,b
5     printf("%d + %d = %d\n", a, b, a + b);
6     return 0;
7 }
```

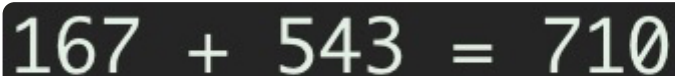
在上面的程序中，a、b 就是变量，167、543 就是数据。那么 167、543 这样的数据是什么类型呢？我们看到它们是整数，所以用的是 int。可以看到，我们定义了两个整数型变量 a、b，并把数据 167、543 分别放到了变量 a 和变量 b 里，进而实现了程序目的。

所以，这里我划个重点，**变量是用来存储数据的**。你理解了吧？

上面例子中的 printf 函数虽然复杂一点儿，但其实简单来说就只有如下两部分：

1. “%d + %d = %d\n” 叫做**格式控制字符串**，其中 %d 是整型的“格式占位符”。
2. a, b, a + b 叫做**参数列表**，每一项依次对应一个“格式占位符”要输出的内容。

“格式占位符”与参数列表中的项一一对应，按照顺序，第一个 %d 与 a 对应，第二个 %d 与 b 对应，第三个 %d 与 a+b 对应。在输出内容时，会被替换成为对应项的内容。例如，上述程序的输出结果如下图所示：



167 + 543 = 710

你会看到，相应的 %d 被按顺序替换成了变量 a、变量 b 以及表达式 a + b 的内容。

我们利用 printf 函数输出了二者加法表达式的值。至此，这个程序之于我们而言，已经具备了一个简单计算器的基本功能了。

下表是一些常用的“类型”与其“格式占位符”之间的对应关系，同样，还是不用问为什么，先试着去用，把它当做规则记住就可以了。

格式占位符	作用	对应类型
%d	十进制形式的整型数据	int
%f	小数形式的浮点型数据	float/double
%s	字符串数据	char[]

3. 输入函数：让计算机“捡起”你的话

前面，我们已经看过了 printf 函数的输出功能，它可以把程序中数据信息输出到屏幕上，这个屏幕，就是现在我们与程序交互的最基本的场所，以后你还会接触其他交互形式，但不急，慢慢来。

你可以把这个屏幕想象成一个桌子，你和计算机坐在桌子的两侧，当程序执行到 printf 的时候，计算机会把数据放到这个桌子上。那么这个时候，如果你往桌子上放了一些数据，计算机如何把这些数据信息“捡起来”呢？

看到这里，请在你的编译器中，输入如下程序：


 复制代码

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b; // 定义变量 a,b
5     scanf("%d%d", &a, &b); // 输入函数
6     printf("%d + %d = %d", a, b, a + b); // 输出函数
7     return 0;
8 }
```

代码的第 5 行中，有一个 `scanf` 函数，它会帮计算机把你放在屏幕上的数据“捡起来”，就像上文中说的“变量是用来存放数据的”，计算机会把捡起来的数据存储在 `a`、`b` 两个变量中。

注意：`a`、`b` 两个变量前面有一个特殊的符号 `&`（取地址符），在这里暂不做过多解释。

当你运行上面这段程序后，可能会得到如下效果：

 复制代码

```
1 192 567
2 192 + 567 = 759
```

第一行是你放到屏幕上的信息，第二行是计算机放到屏幕上的信息。

有了输入函数以后，面对每次不同的计算需求，就不需要修改程序代码了，我们可以直接在屏幕上输入两个需要计算的数据，计算机就会给我们一个满意的答案。

最后我们来看看输入输出函数的返回值：

`printf` 函数的返回值，代表向屏幕上打印了多少个字符。

`scanf` 函数的返回值，代表成功给多少个变量赋了值（后续再展开讲）。

一起动手，搞事情

下面我给出两道思考题，希望你能尽量自己动手查资料解决。

以后，基本每堂课我都会留一些你抬抬脚就能解决的问题，不太简单，可绝不会难上天。我尽量控制，也欢迎你在专栏下面留下意见和建议，更欢迎你将思考过程留在专栏下面。

思考题（1）：位数输出

计算一个输入整数的十进制表示的位数？

条件 1：允许有多余输出的情况下，怎么实现？

条件 2：只允许输出数字位数的时候，又该怎么实现？

请单纯考虑使用 `printf` 一族函数方法实现。

思考题（2）：读入一行字符串

请大家自行实现一个读入一行字符串，并且输出相关内容的程序，思考如下：

条件 1：如果字符串中没有空格，怎么实现？

条件 2：如果字符串中有空格，又该怎么实现？

请单纯考虑用 scanf 一族函数方法实现。

“hello world”显示什么颜色，你做主

前面我们讲了如何使用输出函数让计算机显示什么，又讲了如何利用类型与变量等组织一句计算机可以理解的话语，以及如何让计算机接收到你想传达信息的渠道。接下来，就回到我们今天的任务：按照字符串信息中规定的颜色，输出相应颜色文本内容的方法程序。

在我讲输出函数的时候，提到转义字符，其中有一个转义字符就是用来操作颜色的，它就是：\033。下面就让我们具体看一下，它是如何工作的。

设置颜色，以 \033 开始，也以 \033 结束，这种首尾呼应的结构对记忆比较友好。具体格式如下：

```
1 格式：\033[属性代码{；属性代码...}m 输出内容\033[0m
```

 复制代码

我们来介绍几个属性代码，并加以使用：

0 代表关闭所有属性

1 代表高亮 / 粗体

4 代表下划线

33 代表黄色

如果你在你的 Linux 环境中输入如下代码：

```
1 #include <stdio.h>
2
3 int main() {
```

 复制代码

```
4     printf("\033[1;4;33mhello color\033[0m");  
5     return 0;  
6 }
```

运行以后，你将会在终端看到一行高亮且带有下划线的“hello color”字符串。如下图所示：



至此，我们就准备好了完成课程任务的所有基本元素了，下面，就请你自行尝试一下本任务吧，即使做不出来，也不用担心，我会上传参考代码。

参考代码中，会涉及一些我们后续才会学到的编码技巧，你暂时看不懂也没关系，只需要欣赏它就好了。毕竟，想要进入一个行业的前提，是要懂得这个行业的审美标准。

课程小结

今天是我们第一次真正接触 C 语言，所涉及的专业词汇可能有点多，你可能看完后对一些概念也是分不太清楚，但不要担心，当你接触的多了，这些术语渐渐都会清晰明白。下面呢，我来给你总结以下今天的重点内容：

1. printf 函数是用来输出内容的方法，包含了格式控制字符串和参数列表两部分。
2. 类型和变量是两个完全不同的概念，变量是用来存储数据的。
3. 使用格式占位符的时候，需要对应到相关类型，整型对应到 %d，字符型对应到 %c，浮点型对应到 %f，双精度浮点型对应到 %lf。

总之，今天这节课你已经知道如何和计算机打招呼，以及如何让计算机“听”你说的话了。

我是胡光，这是我带你第一次接触 C 语言，你还有什么疑惑或其他想知道的，我们留言区见。

人人都能学会的编程入门课

>>> 每天 10 分钟，轻松学编程

胡光

原百度高级算法研发工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | 学习编程，我到底该选择哪门语言？

下一篇 03 | 判断与循环：给你的程序加上处理逻辑

精选留言 (13)

写留言



奔跑的八戒

2020-01-08

通过百度完成二道思考题，对格式化的输入输出，又有了新的认识。

思考2：

条件 1：如果字符串中没有空格，怎么实现？： %s

条件 2：如果字符串中有空格，又该怎么实现？： %[^\n]，原来scanf 还支持正则表达式。...

展开

作者回复: d(^_^o)，非常棒！

只输出数字位数的话 fprintf想一想。

1

1



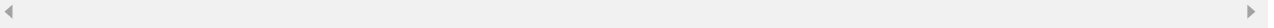
一步

2020-01-07

为什么用printf打印字符串后面会多一个%号呢?

展开 ▾

作者回复: 我看看你的代码?



💬 1

👍 1

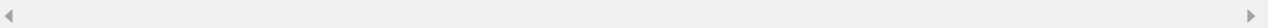


徐洲更

2020-01-07

\033这类打印颜色的方式是shell终端的解读格式，而非windows系统下。

作者回复: 对的，windows的话可以调用API实现颜色功能，这个要是有一定基础的话，可以自行查阅资料试一下。



💬

👍 1



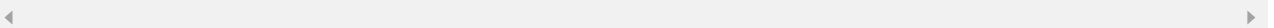
仁

2020-01-11

老师...什么是多余输出啊...第一题两个条件不太看得懂

展开 ▾

作者回复: 哈哈，你看文章的留言区中已经有人给出了自己的答案，并且已经很接近标准答案了。你先看一看。



💬

👍



潮汐

2020-01-10

现实世界就像这个战场一样，是抽象的，而计算机的世界是具体的、可计算的。从现在开始，你应该注意学习如何将现实世界的“信息”，映射到计算机世界中“数据”的技巧。点赞!

展开 ▾

💬

👍

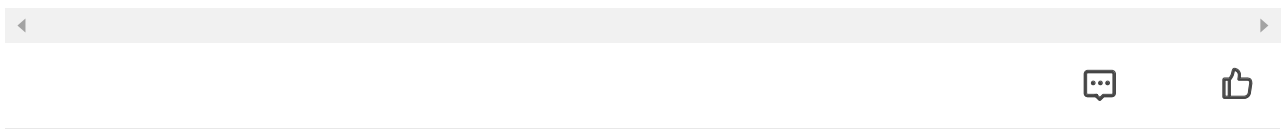


MagisDing

2020-01-09

老师可以在下节课上课之前发布本节课思考题的参考代码吗?

作者回复: 嗯嗯, 咱们后面会定期的发送思考题的参考答案。不仅会有参考答案, 还会有你们的精彩留言。

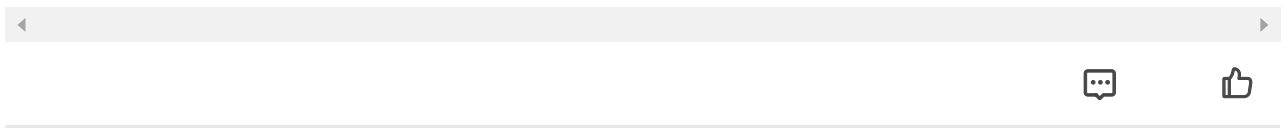


一溢孤行

2020-01-09

我还是不清楚Windows系统下该怎么去改变输出代码的颜色

作者回复: 调用windows环境下的API方法, 可以上网搜一下。



一步

2020-01-07

为什么用printf打印字符串后面会多一个%号呢?

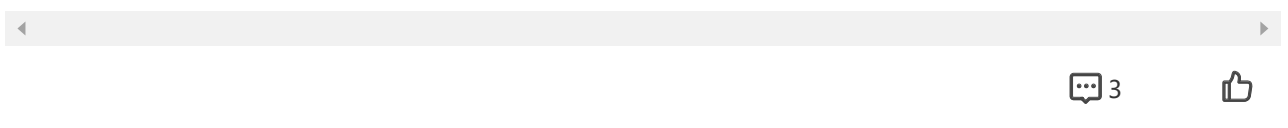
作者回复: 我看看你的代码?

代码如下:

```
printf("\033[1;4;33mhello color\033[0m"); // 就是文中的代码...
```

展开 ∨

作者回复: 不可能啊, 文字有颜色么和下划线么?



一步

2020-01-07

不能回复老师的评论, 我就重新写了

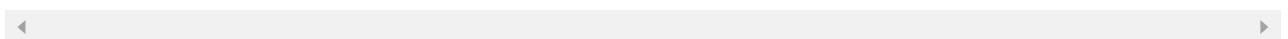
Mac打印字符串使用\a 没有声音

作者回复: 你用的什么环境?

环境为: 编译环境 gcc: Apple LLVM version 10.0.1 (clang-1001.0.46.4)...

展开 ∨

作者回复: 没啥问题



**fengtalk**

2020-01-07

大家用的都是什么编辑器呢？我在MacOS下用的CodeRunner。

**一步**

2020-01-07

Mac打印字符串使用[\a](#) 没有声音

展开

作者回复: 你用的什么环境？

**行问**

2020-01-07

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

```
int main() {  
    printf("\033[1;4;33mhello color\033[0m");  
    return 0;...
```

展开

作者回复: 装个虚拟机Ubuntu、CentOS或者DeepIn系统，或者使用windows 10里面的bash Ubuntu都行。^_^

**行问**

2020-01-07

字符型对应到 %c 还是 %s ?

没看到有上传代码的地址

展开

作者回复: %c, 字符串对应的是%s



