

C语言教程

C语言教程

简单介绍

0.C语言简述

1.第一个程序-Hello,World!

1-0我们来逐行解读一下

```
1. #include <stdio.h>
2. int main()
3. printf("Hello,world!\n");
4. return 0;
```

2变量与数据类型

2-0数据类型

2-1变量

1.变量类型转换

隐式转换 自动类型转换

显示转换 强制类型转换

简单介绍

本教程主要是C语言的快速入门教程，具体不会涉及的很深。

本片教程已经默认你已经配置好了C语言环境。

0.C语言简述

[自己去百度百科上看看，这是直达链接-百度百科\(baidu.com\)\]](#)

1.第一个程序-Hello,World!

在很多编程语言的教程中，首先叫你的便是打印 `Hello,world!` (你好，世界！)这句话。

为什么要打印 `Hello,world!` 这句话而不是其他话呢？*以下内容来源于网络*

*1、hello world的起源要追溯到1972年，贝尔实验室著名研究员Brian Kernighan在撰写“B语言教程与指导(Tutorial Introduction to the Language B)”时初次使用（程序），这是目前已知最早的在计算机著作中将hello和world一起使用的记录。之后，在1978年，他在他和Dennis Ritchie合作撰写的C语言圣经“The C Programming Language”中，延用了“hello,world”句式，作为开篇第一个程序。在这个程序里，输出的“hello,world”全部是小写，没有感叹号，逗号后有一空格。虽然之后几乎没能流传下来这个最初的格式，但从此用hello world向世界打招呼成为惯例。几乎每一个程序设计语言的教材中的第一个范例都是hello world程序，因此在学习一门新语言的时候用hello world作为起步已经成为计算机程序界的一个传统。

*2、编写一个简单的程序，可以让我们知道编译器能正常编译；

*3、Hello World程序就是打印这一行文字，一般来说这个程序是一个非常简单而又经典的程序，一般教程喜欢把它作为第一个例子，来让学习者初步认识程序的编写。hello world的迷人之处在于：它是简单的，即便没有任何基础，你也可以轻松理解，但它又不是苍白的，它带有一个程序语言鲜明的特色，体现了该语言最基本的思想和特征。同样也是hello world，让人体会到第一份成功的喜悦。

*4、相当于中国人到一处地方的留言记录语“xx 到此一游”一样，可以作为你学习编程语言的回忆。

以下便是C语言的 Hello,world! 程序

```
1    #include <stdio.h>
2
3    int main()
4    {
5        printf("Hello,world!\n");
6        return 0;
7    }
8
```

1-0我们来逐行解读一下

1.#include <stdio.h>

这个句的意思是导入stdio.h这个文件。这个文件被称为头文件(扩展为.h)，这个头文件包含了C语言常用的一些语句。如 while for printf scanf 等等，当然由于 printf 语句的常用性，所以C语言允许不导入库就可以使用 printf。

2.int main()

这个语句定义了一个名为 main 的函数，返回值为int(整形)

在C语言中 main 函数就是程序的入口，程序先执行 main 内的代码。

当然你可能注意到了 main 函数下面的一对大括号 {} 此括号用于定义函数或是其他语句的范围，除了函数要使用 {} 之外，还有 for if else elif 等也要使用 {} 来划定执行范围。

3.printf("Hello,world!\n");

这个语句的意思是打印 Hello,world! 这句话到终端

在这给语句中你可能注意到 \n，\n 不会被打印出来，因为它是ASCII的控制字符，意为换行。

控制字符还有 \t：制表符 \r：回车 \e 转义** 等等。

现在语句后面有一个分号，这个分号用于标记一个语句的结束。当然，并不是所有语句都要加这个分号，例如上面的 #include <stdio.h> 就没加，因为这预处理语句是不用加分号的，还有规定了执行范围的语句也不用加如上文中声明的 main 它就没加分号。

4.return 0;

这个语句的意思是函数调用完毕返回 0，由于上文中我们把 main 函数的类型设为了 int(整形)也就是整数类型，所有 return 只能返回数字，当然这里不一定要填 0 其他数字也可以，不过填 0 已经成了一个约定俗成的规范了，象征着程序正常退出。同时填负数，意味着程序是因某种错误而退出，当然这也是一个约定俗成的规范。

而return用于调用函数后返回的东西，一旦执行到 return 语句，函数便会立马结束，并返回 return 后跟的内容。

*注意 return 0; 也可以不用写，程序也是可以照常退出的，因为下面已经没有代码了。

2变量与数据类型

本章回简要概述数据的几种样式，与变量的创建与使用。

2-0数据类型

C语言的数据类型如下表

整数类型表

关键字	中文名称	占内存字节	值范围
char	字节型	1字节	-128 ~ 127
unsigned char	无符号字节型	1字节	0 ~ 255
int	整形	4字节	-32,768 ~ 32,767
unsigned int	无符号整型	4字节	0 ~ 65,535
short	短整形	2字节	-32,768 ~ 32,767
unsigned short	无符号短整形	2字节	0 ~ 65,535
long	长整形	4字节	-2,147,483,648 ~ 2,147,483,647
unsigned long	无符号长整形	4字节	0 ~ 4,294,967,295

浮点数类型表

关键字	中文名称	占内存字节	值范围	精度
float	单精度浮点数	4字节	1.2E-38 ~ 3.4E+38	6位有效
double	双精度浮点数	8字节	2.3E-308 ~ 1.7E+308	15位有效
long double	长双精度浮点数	16字节	3.4E-4932 ~ 1.1E+4932	19位有效

可以看到C语言的数据类型还挺多，这么多的数据类型能可以根据实际情况来自由选择，这样可以极大程度上减少资源的浪费。

2-1变量

与Python不同，C语言不会检测你的变量类型，所以在分配变量时C语言会要求你规定变量的数据类型，具体格式大体如下

变量的定义格式通常为

数据类型 变量名称;

或 数据类型 变量名称 = 变量内容;

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a = 2;
6      printf("The a is %d\n", a);
7  }
```

其中%d是变量格式化字符串，对应int整形系列

其他常用格式化字符串有

%c: char(字符型)

%lf: float/double(单/双精度浮点型)

1.变量类型转换

变量的转换有两种方式，分别是显示转换与隐式转换，或者还一种说法为强制类型转换与自动类型转换。

隐式转换 自动类型转换

```
1 float a = 0.1;
2 double b = 0.2;
3 double c = a + b;
```

在上述代码片段中变量 `a` 的变量类型与 `b`, `c` 的数据类型并不相同，此时C语言会自动进行数据类型转换，顺序为从低到高。优先度为下表

整形: `short -> int -> long`

浮点数(小数): `float -> double`

所以说此时 `a` 已经被隐式地转换为 `double` 了

显示转换 强制类型转换