

## 3 一个简单的C语言程序

在数学中，函数是一个映射，但是，C语言中的函数是一个用“{}”括起来的单元，这个单元里是一组为了实现某一功能的语句的集合。

### 3.0 Hello world!

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* 这是一个简单的C语言程序 */
6      // 它的作用就是在屏幕的输出窗口打印：Hello world!
7      printf("Hello world!\n");
8      return 0;
9  }
```

上面的程序的功能十分简单，编译运行之后，它将在屏幕终端输出“Hello world!”。

要注意的一点是：**程序中的所有字符都是英文输入法状态下的字符！**

### 3.1 简单释义

按行解释

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

这是一个预包含语句，它以“#”符号开头，紧接着一个 include（包含的意思）关键字，然后用尖括号括起来的要包含的文件名 stdio.h。在最新标准中，<> 表示导入的头文件为系统自带库文件，如果导入用户自己编写的头文件，则使用""，双引号会优先在用户文件中查找头文件，如果没有目标文件，则会在系统文件中继续查找，所以它也可以导入系统自带库文件，但随着语义化编程的推进，不推荐这么使用，最好还是按用途区别使用。

stdio.h是标准输入输出库头文件，后面程序中调用的 printf() 函数就是在这个头文件中定义的，stdio 即为 standard input & output 之意。预包含语句不需要分号结束符。

```
1  int main()
```

这行是 main() 函数（主函数）声明语句，一个C语言程序都必须有并且只有一个 main() 函数，前面的关键字 int 规定函数返回值的类型，这里是一个整型。其中 int 与 main() 之间最少有一个空格。一个程序从 main() 函数开始执行。另，函数声明时在结尾也需要语句结束符“;”，但如果后面紧跟着函数定义，则不需要分号做结束符。

```
1  { ... }
```

大括号中间包含的即为实现某一功能的语句的集合，这里包含的是 main() 函数的定义部分。这样的大括号的结束符在其内部包含语句的结尾，所以在括号的结尾不需要结束符（定义结构体等特殊情况则另作说明）。

```
1  /* 这是一个简单的C语言程序 */  
2  // 它的作用就是在屏幕的输出窗口打印: Hello world!
```

这两行是注释语句，注释语句的内容将在程序编译时被忽略，不对程序执行造成影响，仅用于注释说明。其中“//”属于行注释，其后一行的字符为注释，“/\* \*/”属于块注释，其间的所用字符为注释，两种方式按需择优使用。注释不得嵌套，也不能在字符串或字符值中使用。

```
1  printf("Hello world!\n");
```

该语句用于格式化输出到屏幕。printf() 函数调用自之前导入的 stdio.h 头文件，要输出的内容即为双引号引起来的内容。在双引号中的最后，有一个“\n”，其中“\”是转义符，其后的内容不发挥其直接的作用。连上“n”即表示换行，而不是一个字符“n”。语句最后以分号作为结束符。

```
1  return 0;
```

函数返回语句，返回函数的返回值。终止函数，返回函数返回值 0，返回值的数据类型在函数声明时被规定，可以返回一个确定的值，也可以返回一个变量或运算式。主函数的return语句即为终止程序。语句最后以分号作为结束符。

## 3.2 简单总结

C语言程序中，所有的括号都是相互匹配的，左括号表示开始，相应的右括号表示结束；其次双引号也是匹配的；所有的语句以分号表示结束（但有例外，比如 include 语句）。

一个C语言程序中可以有多个函数，但只能有一个主函数。

函数与函数类型之间至少有一个以作分隔的空格，之后变量等的类型也是如此。

**注意：**在C语言中，缩进并不是必须的，更多的时候，规范的缩进只是起到使代码更美观的作用，它不影响语句间的语法关系。