**Arduino语言**  
  
Arduino使用C/C++编写程序，虽然C++兼容C语言，但这是两种语言，C语言是一种面向过程的编程语言，C++是一种面向对象的编程语言。早期的Arduino核心库使用C语言编写，后来引进了面向对象的思想，目前最新的Arduino核心库采用C与C++混合编写而成。  
通常我们说的Arduino语言，是指Arduino核心库文件提供的各种应用程序编程接口（Application Programming Interface，简称API）的集合。这些API是对更底层的单片机支持库进行二次封装所形成的。例如，使用AVR单片机的Arduino的核心库是对AVR-Libc（基于GCC的AVR支持库）的二次封装。  
传统开发方式中，你需要厘清每个寄存器的意义及之间的关系，然后通过配置多个寄存器来达到目的。  
  
而在Arduino中，使用了清楚明了的API替代繁杂的寄存器配置过程，如以下代码：

[C++] *纯文本查看* *复制代码*

[?](http://www.arduino.cn/)

|  |  |
| --- | --- |
| 代码 | |
| 001  002 | pinMode(13,OUTPUT);  digitalWrite(13,HIGH); |

pinMode(13,OUTPUT)即是设置引脚的模式，这里设定了13脚为输出模式；而digitalWrite(13,HIGH) 是让13脚输出高电平数字信号。  
  
这些封装好的API，使得程序中的语句更容易被理解，我们不用理会单片机中繁杂的寄存器配置，就能直观的控制Arduino，增强程序的可读性的同时，也提高了开发效率。  
  
在上一章我们已经看到第一个Arduino程序Blink，如果你使用过C/C++语言，你会发现Arduino的程序结构与传统的C/C++结构的不同——Arduino程序中没有main函数。  
  
其实并不是Arduino没有main函数，而是main函数的定义隐藏在了Arduino的核心库文件中。Arduino开发一般不直接操作main函数，而是使用Setup和loop这个两个函数。  
  
通过**Arduino IDE菜单>文件>示例>01.Basics>BareMinimum**可以看到Arduino程序的基本结构：

[C++] *纯文本查看* *复制代码*

[?](http://www.arduino.cn/)

|  |  |
| --- | --- |
| 代码 | |
| 001  002  003  004  005  006  007  008  009 | void setup()  {    // 在这里加入你的setup代码，它只会运行一次：  }    void loop()  {    // 在这里加入你的loop代码,它会不断重复运行：  } |

Arduino程序基本结构由setup() 和loop() 两个函数组成：

[C++] *纯文本查看* *复制代码*

[?](http://www.arduino.cn/)

|  |  |
| --- | --- |
| 代码 | |
| 001 | setup() |

Arduino控制器通电或复位后，即会开始执行setup() 函数中的程序，该部分只会执行一次。  
通常我们会在setup() 函数中完成Arduino的初始化设置，如配置I/O口状态，初始化串口等操作。

[C++] *纯文本查看* *复制代码*

[?](http://www.arduino.cn/)

|  |  |
| --- | --- |
| 代码 | |
| 001 | loop() |

在setup() 函数中的程序执行完后，Arduino会接着执行loop() 函数中的程序。而loop()函数是一个死循环，其中的程序会不断的重复运行。  
通常我们会在loop() 函数中完成程序的主要功能，如驱动各种模块，采集数据等。