**C语言基本语法**

C语言基本语法 实例代码教程- 我们已经看到的C程序的基本结构，所以这将是很容易理解其他的C语言编程的基本构建块。

你已经看到的C程序的基本结构，所以这将是很容易理解其他的C语言编程的基本构建块。

C语言的记号

C程序由各种令牌，令牌可以是关键字，标识符，常量，字符串文字或符号。例如，下面的C语句包含五个令牌：

printf("Hello, World! \n");

The individual tokens are:

printf

(

"Hello, World! \n"

)

;

分号 ;

In C program, the semicolon is a statement terminator. That is, each individual statement must be ended with a semicolon. It indicates the end of one logical entity.

For example, following are two different statements:

printf("Hello, World! \n");

return 0;

注释

Comments are like helping text in your C program and they are ignored by the compiler. They start with /\* and terminates with the characters \*/ as shown below:

/\* my first program in C \*/

You can not have comments with in comments and they do not occur within a string or character literals.

标识符

C标识符是用来标识变量，函数的名称，或任何其它用户定义的项目。开始的标识符以字母A到Z或a〜z或下划线\_由零个或多个字母，下划线和数字（0〜9）。

C does not allow punctuation characters such as @, $, and % within identifiers. C is a **case sensitive**programming language. Thus *Manpower* and *manpower* are two different identifiers in C. Here are some examples of acceptable identifiers:

mohd zara abc move\_name a\_123

myname50 \_temp j a23b9 retVal

关键字

The following list shows the reserved words in C. These reserved words may not be used as constant or variable or any other identifier names.

|  |  |  |  |
| --- | --- | --- | --- |
| auto | else | long | switch |
| break | enum | register | typedef |
| case | extern | return | union |
| char | float | short | unsigned |
| const | for | signed | void |
| continue | goto | sizeof | volatile |
| default | if | static | while |
| do | int | struct | \_Packed |
| double |  |  |  |

C语言中的空白行

A line containing only whitespace, possibly with a comment, is known as a blank line, and a C compiler totally ignores it.

Whitespace is the term used in C to describe blanks, tabs, newline characters and comments. Whitespace separates one part of a statement from another and enables the compiler to identify where one element in a statement, such as int, ends and the next element begins. Therefore, in the following statement:

int age;

必须有至少一个空白字符（通常是int和年龄的编译器能够区分它们之间有一个空格）。另一方面，在下面的语句

fruit = apples + oranges; // get the total fruit

任何空白字符是必要的之间水果和=，=苹果，虽然你是自由的，包括一些，如果你想可读性目的。

词法分析序列表：

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符表 | i |  | |
| 常数表 | C |  | |
| 关键字表 | K | Int  Float  Char  Void  If  Else  Switch  Case  For  Do  While  Continue  Break  Default  Sizeof  return | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 |
| 界符表 | P | >=  <=  ==  =  >  <  +  -  \*  /  {  }  ,  ;  (  ) | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 |
| 字符表 | Ch |  | |
| 字符串表 | st |  | |

# [简易c语言LL(1)文法](http://www.cnblogs.com/www924121851/p/6135760.html)

〈程序〉：：〈声明〉|〈函数〉  
〈声明〉：：（A|〈声明〉'）〈函数声明〉  
A：：〈头文件〉|〈宏定义〉  
〈声明〉'：：〈声明〉|null  
〈头文件〉：：〈头文件〉〈头文件〉'  
〈头文件〉'：：〈头文件〉|null  
〈宏定义〉：：#define B〈宏定义〉|null  
B：：〈字符串 〉C  
C：：〈字符串〉|〈数字〉  
〈字符串〉：：〈变量名〉  
〈函数声明〉：：〈返回值类型 〉〈函数名〉(〈形参〉) ;〈函数声明〉|null  
〈形参〉：：〈数据类型 〉〈变量名〉|〈数据类型 〉〈变量名〉〈,〉〈形参〉|void  
〈变量名〉：：〈字母〉D  
D：：〈字母〉|〈数字〉|〈\_〉|D|null  
〈字母〉：：F〈字母〉'  
F：：a|b|...|y|z|（大写字母。。。）  
〈字母〉'：：〈字母〉|null  
〈数字〉：：E〈数字〉'  
E：：0|1|...|8|9  
〈数字〉'：：〈数字〉|null  
〈返回值类型〉：：int|float|char|void  
〈函数名〉：：〈变量名〉  
〈函数〉：：〈返回值类型〉〈函数名〉{〈函数主体〉}|函数名〉{〈函数主体〉}〈函数〉  
〈函数主体〉：：〈变量的申请〉〈处理语句〉  
〈变量的申请〉：：〈数据类型 〉〈变量名〉;〈变量的申请〉'  
〈变量的申请〉'：：〈变量的申请〉|null  
〈数据类型 〉：：int|float|char  
〈处理语句〉：：F〈处理语句〉'  
F：：〈赋值语句〉|〈判断语句〉|〈循环语句〉|〈函数调用〉  
〈处理语句〉'：：〈处理语句〉|null  
〈赋值语句〉：：〈变量名〉=G  
G：：〈字母〉|〈数字〉|〈变量名〉|〈运算〉  
〈运算〉：：〈变量名〉〈运算符〉〈数字〉|〈数字〉〈运算符〉〈数字〉  
〈运算符〉：：+|-|\*|/  
〈判断语句〉：：if(〈条件〉){〈处理语句〉|null}H  
H：：〈Else if〉else{〈处理语句〉|null}|null  
〈Else if〉：：〈Else if〉'〈else if〉(〈条件〉){〈处理语句〉|null}|null  
〈Else if〉'：：〈Else if〉|null  
〈条件〉：：〈变量名〉〈判断运算符〉〈变量名〉|〈变量名〉〈判断运算符〉〈数字〉|〈数字〉〈判断运算符〉〈数字〉|〈数字〉  
〈判断运算符〉：：=|>|<|>=|<=|!=  
〈循环语句〉：：while(〈条件〉){〈处理语句〉}|do{〈处理语句〉}while(〈条件〉)|for(〈赋值语句〉;〈条件〉;〈运算〉){〈处理语句〉}  
〈函数调用〉：：〈变量名〉=〈函数名〉(〈形参〉)|〈函数名〉(〈形参〉)