C 基本语法 C 变量

C 数据类型

在 C 语言中，数据类型指的是用于声明不同类型的变量或函数的一个广泛的系统。变量的类型决定了变量存储占用的空间，以及如何解释存储的位模式。

C 中的类型可分为以下几种：

序号	类型与描述
1	**基本类型：** 它们是算术类型，包括两种类型：整数类型和浮点类型。
2	**枚举类型：** 它们也是算术类型，被用来定义在程序中只能赋予其一定的离散整数值的变量。
3	**void 类型：** 类型说明符 *void* 表明没有可用的值。
4	**派生类型：** 它们包括：指针类型、数组类型、结构类型、共用体类型和函数类型。

数组类型和结构类型统称为聚合类型。函数的类型指的是函数返回值的类型。在本章节接下来的部分我们将介绍基本类型，其他几种类型会在后边几个章节中进行讲解。

整数类型

下表列出了关于标准整数类型的存储大小和值范围的细节：

类型	存储大小	值范围
char	1 字节	-128 到 127 或 0 到 255
unsigned char	1 字节	0 到 255
signed char	1 字节	-128 到 127
int	2 或 4 字节	-32,768 到 32,767 或 -2,147,483,648 到 2,147,483,647
unsigned int	2 或 4 字节	0 到 65,535 或 0 到 4,294,967,295
short	2 字节	-32,768 到 32,767
unsigned short	2 字节	0 到 65,535
long	4 字节	-2,147,483,648 到 2,147,483,647

unsigned long	4 字节	0 到 4,294,967,295

注意，各种类型的存储大小与系统位数有关，但目前通用的以64位系统为主。

以下列出了32位系统与64位系统的存储大小的差别（windows 相同）：

Windows vc12		Linux gcc-5.3.1		Compiler
win32	x64	i686	x86_64	Target
1		1	1	char
1		1	1	unsigned char
2		2	2	short
2		2	2	unsigned short
4		4	4	int
4		4	4	unsigned int
4		4	8	long
4		4	8	unsigned long
4		4	4	float
8		8	8	double
4		4	8	long int
8		8	8	long long
8		12	16	long double

为了得到某个类型或某个变量在特定平台上的准确大小，您可以使用 **sizeof** 运算符。表达式 *sizeof(type)* 得到对象或类型的存储字节大小。下面的实例演示了获取 int 类型的大小：

实例

```
#include <stdio.h>
#include <limits.h>
int main()
{
printf("int 存储大小 : %lu \n", sizeof(int));
return 0;
}
```

%lu 为 32 位无符号整数，详细说明查看 C 库函数 - printf()。

当您在 Linux 上编译并执行上面的程序时，它会产生下列结果：

```
int 存储大小 : 4
```

浮点类型

下表列出了关于标准浮点类型的存储大小、值范围和精度的细节：

类型	存储大小	值范围	精度
float	4 字节	1.2E-38 到 3.4E+38	6 位小数
double	8 字节	2.3E-308 到 1.7E+308	15 位小数
long double	16 字节	3.4E-4932 到 1.1E+4932	19 位小数

头文件 float.h 定义了宏，在程序中可以使用这些值和其他有关实数二进制表示的细节。下面的实例将输出浮点类型占用的存储空间以及它的范围值：

实例

```
#include <stdio.h>
#include <float.h>
int main()
{
printf("float 存储最大字节数 : %d \n", sizeof(float));
printf("float 最小值: %E\n", FLT_MIN );
printf("float 最大值: %E\n", FLT_MAX );
printf("精度值: %d\n", FLT_DIG );
return 0;
}
```

%E 为以指数形式输出单、双精度实数，详细说明查看 C 库函数 - printf()。

当您在 Linux 上编译并执行上面的程序时，它会产生下列结果：

```
float 存储最大字节数 : 4
float 最小值: 1.175494E-38
float 最大值: 3.402823E+38
精度值: 6
```

void 类型

void 类型指定没有可用的值。它通常用于以下三种情况下：

序号	类型与描述
1	**函数返回为空** C 中有各种函数都不返回值，或者您可以说它们返回空。不返回值的函数的返回类型为空。例如 **void exit (int status);**
2	**函数参数为空** C 中有各种函数不接受任何参数。不带参数的函数可以接受一个 void。例如 **int rand(void);**
3	**指针指向 void** 类型为 void * 的指针代表对象的地址，而不是类型。例如，内存分配函数 **void *malloc(size_t size);** 返回指向 void 的指针，可以转换为任何数据类型。

如果现在您还是无法完全理解 void 类型，不用太担心，在后续的章节中我们将会详细讲解这些概念。

C 基本语法 C 变量

2 篇笔记 写笔记

常用基本数据类型占用空间（64位机器为例）

- char ： 1个字节
- int ：4个字节
- float：4个字节
- double：8个字节

基本类型书写

整数

- a，默认为10进制 ，10 ，20。
- b，以0开头为8进制，045，021。
- c.，以0b开头为2进制，0b11101101。
- d，以0x开头为16进制，0x21458adf。

小数

单精度常量：2.3f 。

双精度常量：2.3，默认为双精度。

字符型常量

用英文单引号括起来，只保存一个字符'a'、'b' 、'*' ，还有转义字符 '\n' 、'\t'。

字符串常量

用英文的双引号引起来 可以保存多个字符："abc"。

Justforyou 2年前 (2017-09-01)

1、数据类型转换：C 语言中如果一个表达式中含有不同类型的常量和变量，在计算时，会将它们自动转换为同一种类型；在 C 语言中也可以对数据类型进行强制转换；

2、自动转换规则：

- a) 浮点数赋给整型，该浮点数小数被舍去；
- b) 整数赋给浮点型，数值不变，但是被存储到相应的浮点型变量中；

3、强制类型转换形式: **(类型说明符)(表达式)**

实例程序：

```
#include<stdio.h>

int main()
{
    float f,x=3.6,y=5.2;
    int i=4,a,b;
```

```
    a=x+y;
    b=(int)(x+y);
    f=10/i;
    printf("a=%d,b=%d,f=%f,x=%f\n",a,b,f,x);
}
```

例中先计算 x+y 值为 8.8，然后赋值给 a，因为a为整型，所以自取整数部分8，a=8;

接下来 b 把 x+y 强制转换为整型;

最后 10/i 是两个整数相除，结果仍为整数 2，把 2 赋给浮点数 f;

x 为浮点型直接输出。

zsz311 1年前 (2018-03-08)