



# C 语言和程序设计思想 I

## 数据和 C



# 课程大纲

## 周 主题

---

**5** 概览、C 语言概述、数据和 C

---

**6** 数据和 C

---

**9** 字符串的格式化输出与输出

---

**10** 运算符、表达式和语句

---

**11** 控制语句之循环

---

**12** 控制语句之分支和跳转

---

**13** 字符的输入/输出和输入确认

---

**14** \*函数

---



# 内容

- 数据和 C



# 简单示例

```
#include <stdio.h>
int main (void)
{
    int num;
    num = 1;

    printf ("I am a simple ");
    printf ("computer.\n");
    printf ("My favorite number is %d because it is first.\n", num);
    return 0;
}
```

*/\* 一个简单的 C 程序 \*/*

*/\* 定义一个名为 num 的变量 \*/*  
*/\* 为 num 赋一个值 \*/*

*/\* 使用 printf () 函数 \*/*



# 简单的 C 程序结构

```
#include <stdio.h>
int main (void)
{
    statements
    return 0;
}
```

*/\* 一个简单的 C 程序 \*/*

*/\* 函数、代码块等 \*/*



# 函数

- printf 函数
- 赋值运算符“=”

```
int a=5;
```

## Standard C 语言标准函数库速查 (Cheat Sheet)

### 说明

本页面包含C语言标准版的15个头文件解释以及函数，137个函数和演示，可以点击函数名字进去查看。  
纯粹无责任乱译，参照本页产生的任何错误概不负责。本页也不一定会保证能持续更新。(查看 [更新记录](#))

C语言函数搜索框:

最后更新: 2013年4月24日

如果您喜欢它, 请考虑 [捐赠](#) 一点资金, 感激不尽!

### C语言标准头

- <assert.h> 断言
- <ctype.h> 字符类测试
- <errno.h> (部分) 库函数抛出的错误代码
- <float.h> 浮点数运算
- <limits.h> 检测整型数据类型值范围
- <locale.h> 本土化
- <math.h> 数学函数
- <setjmp.h> “非局部跳转”
- <signal.h> 信号
- <stdarg.h> 可变参数列表
- <stddef.h> 一些常数, 类型和变量
- <stdio.h> 输入和输出
- <stdlib.h> 实用功能
- <string.h> 字符串函数
- <time.h> 时间和日期函数

### <assert.h> 断言

头文件<assert.h>唯一的目的是提供宏assert的定义。如果断言非真 (expression==0), 则程序会在

### <math.h> 数学函数

<math.h> 是C语言中的数学函数库

三角函数

- double sin(double x); 正弦
- double cos(double x); 余弦
- double tan(double x); 正切

\*cot三角函数, 可以使用tan(PI/2-x)来实现。

反三角函数

- double asin(double x); 结果介于[-PI/2, PI/2]
- double acos(double x); 结果介于[0, PI]
- double atan(double x); 反正切(主值), 结果介于[-PI/2, PI/2]
- double atan2(double y, double x); 反正切(整圆值), 结果介于[-PI, PI]

双曲三角函数

- double sinh(double x); 计算双曲正弦

### <stdio.h> 输入和输出

<stdio.h>头文件定义了用于输入和输出的函数、类型和宏。最重要的类型是用于声明文件指针的FILE。另外两个常用的类型是 size\_t和fpos\_t, size\_t是由运算符sizeof产生的无符号整型; fpos\_t类型定义能够唯一说明文件中的每个位置的对象。由头部 定义的最有用的宏是EOF, 其值代表文件的结尾。

变量:

```
typedef size_t
typedef FILE
typedef fpos_t
```

常量:

- NULL 空值
- \_IOFBF 表示完全缓冲
- \_IOLBF 表示线缓冲
- \_IONBF 表示无缓存
- BUFSIZ setbuf函数所使用的缓冲区的大小
- EOF EOF是负整数表示END OF FILE
- FOPEN\_MAX (20)同时打开的文件的最大数量
- FILENAME\_MAX 文件名的最大长度
- L\_tmpnam 整数, 最大长度的临时文件名
- SEEK\_CUR 取得目前文件位置

<http://ganquan.info/standard-c/>





# 示例

- `printf("The weather today is fine.\n");`
- `printf("The price is %d dollars.\n", 20);`
- `apples = 3;`
- `pi = 3.1415926535;`



# 关键词和保留标识符

ISO/ANSI C 关键字

1	2	3	4
auto	<b>enum</b>	<i>restrict</i>	unsigned
break	extern	return	<b>void</b>
case	float	short	<b>volatile</b>
char	for	<b>signed</b>	while
<b>const</b>	goto	sizeof	<i>_Bool</i>
continue	if	static	<i>_Complex</i>
default	<i>inline</i>	struct	<i>_Imaginary</i>
do	int	switch	
double	long	typedef	
else	register	union	

PS: 粗体显示的关键字是有 ISO/ANSI C90 标准新增；斜体为 C99 标准新增。

重点： int, char,  
return, void, unsigned,  
long, short, float,  
double





# 数据类型及其关键字

原来的 K&R 关键字	C90 关键字	C99 关键字
int	signed	_Bool
long	void	_Complex
short		_Imaginary
unsigned		
char		
float		
double		

重点： 第一列  
K&R C 关键字



# 示例程序

```
/* rhodium.c -- 用金属铑衡量您的体重 */
#include <stdio.h>
int main (void)
{
    float weight;                /* 用户的体重 */
    float value;                 /* 相等重量的铑的值 */
    printf ("Are you worth your weight in rhodium?\n");
    printf ("Let's check it out.\n");
    printf ("Please enter your weight in pounds: ");
    /* 用户获取输入 */
    scanf("%f", &weight);
    /* 假设铑为每盎司 770 美元 */
    /* 14.5833 把常衡制的英镑转换成金衡制的盎司 */
    value = 770 * weight *14.5833;
    printf ("Your weight in rhodium is worth $%.2f.\n", value);
    printf ("You are easily worth that! If rhoium prices drop, \n");
    printf ("eat more to maintain your vaule.\n");
    return 0;
}
```

程序的错误（**error**）会导致无法编译，而警告（**warning**）则不会终止编译



# C 基本数据类型

- 整型
- 浮点型
- 字符型
- Bool 型
- 可移植类型
- 复数和虚数类型
- 指针



# 整型

- 以二进制存储
- 有符号、无符号，signed 可省略, unsigned 不可省略；
- short int
- int
- long int
- long long int
- unsigned short int
- unsigned int
- unsigned long int
- unsigned long long int



# 整型

范围：

short < int < long int  
< long long int

源代码：

<https://github.com/Spacebody/My-Codes/blob/master/Codes%20in%20C:C%2B%2B/printMax%26Min.c>

```
#include <stdio.h>;  
#include <limits.h>;
```

```
int main(void)  
{  
    printf("The minimum value of short int = %d\n", SHRT_MIN);  
    printf("The maximum value of short int = %d\n", SHRT_MAX);  
  
    printf("The minimum value of int = %d\n", INT_MIN);  
    printf("The maximum value of int = %d\n", INT_MAX);  
  
    printf("The minimum value of long int = %ld\n", LONG_MIN);  
    printf("The maximum value of long int = %ld\n", LONG_MAX);  
  
    printf("The minimum value of long long int = %lld\n", LLONG_MIN);  
    printf("The maximum value of long long int = %lld\n", LLONG_MAX);  
  
    printf("The maximum value of unsigned short int = %d\n", USHRT_MAX);  
  
    printf("The minimum value of unsigned int = %u\n", UINT_MAX);  
  
    printf("The maximum value of unsigned long int = %lu\n", ULONG_MAX);  
  
    printf("The maximum value of unsigned long long int = %llu\n", ULLONG_MAX);  
  
    printf("The minimum value of char = %d\n", CHAR_MIN);  
    printf("The maximum value of char = %d\n", CHAR_MAX);  
  
    printf("The maximum value of unsigend char = %d\n", UCHAR_MAX);  
  
    return 0;  
}
```



# 示例

- long int estine;
- long johns;
- short int erns;
- unsigned int s\_count;
- unsigned players;
- unsigned long headcount;
- unsigned short yesvotes;
- long long ago;



# 整型

以字节单位输出大小：

源代码：

<https://github.com/Spacebody/My-Codes/blob/master/Codes%20in%20C:C%2B%2B/Typesize.c>

Type int has a size of 4 bytes.  
Type char has a size of 1 bytes.  
Type long has a size of 8 bytes.  
Type double has a size of 8 bytes.

```
/* typesize.c -- 输出类型的大小 */
#include <stdio.h>
int main(void)
{
    /* C99 为类型大小提供了一个 %zd 说明符*/
    printf ("Type int has a size of %zd bytes.\n" ,sizeof(int));
    printf ("Type char has a size of %zd bytes.\n" ,sizeof(char));
    printf ("Type long has a size of %zd bytes.\n" ,sizeof(long));
    printf ("Type double has a size of %zd bytes.\n" ,sizeof(double));
    return 0;
}
```





# 浮点型

- 整数和小数分别存储
- 采用 2 的幂
- 对于一些算数运算，浮点数会损失更多精度
- 浮点数往往表示实际值的近似结果
- 浮点数的运算比整数慢
- **float** ，至少能精确表示 6 位有效数字
- **double** ，通常能够表示至少 10 位有效数字以及更大的指数
- **long double** ，能够表示更大的指数

PS: 相关的头文件 `float.h`



# 字符型

- 存储字母和标点符号
  - 从技术实现上是整数类型
  - 常用的编码 ASCII 码（美国）、Unicode 字符集
  - **char**
  - 单引号
  - 可直接使用字符对应的数值对变量声明
  - 转义字符
- 
- e.g. `char letter = 'a';`  
      `char name[] = "Jerry";`



# 转义字符

字符	意义	字符	意义	字符	详细作用
\a	警报 (ANSI C)	\\	反斜杠 (\)	\a	产生一个能听或者看到的警报，取决于系统，蜂鸣最常见
\b	退格	\	单引号 (')	\b	使活动位置在当前行上退回一个空格
\f	走纸	\"	双引号 (")	\f	将活动位置移到下一页纸
\n	换行	\?	问号 (?)	\n	将活动位置移到下一行的开始处
\r	回车	\0oo	八进制值 (o 为八进制数)	\r	将活动位置移到当前行的开始处
\t	水平制表符	\xhh	十六进制值 (h 为十六进制值)	\t	将活动位置移到下一个水平制表点
\v	垂直制表符			\v	将活动位置移到下一个垂直制表点

本次重点：\n, \t



# Bool 型

- Bool
- 整数类型，只有两个值 0 和 1
- true = 1 , false = 0
- 相关的头文件: stdbool.h -- **bool**



# 进制初步

- 二进制 e.g. 1010
- 八进制 e.g. 0123
- 十六进制  
A (10) 、 B (11) 、 C (12) 、 D (13) 、 E (14) 、 F (15)  
e.g. 0x123 or 0X123
- 十进制



# 进制使用示例

```
/* 以十进制、八进制、十六进制输出 100*/
```

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

```
int main(void)
```

```
{
```

```
    int x = 100;
```

```
    printf("dec = %d; octal = %o; hex = %x\n",x, x, x);
```

```
    printf("dec = %d; octal = %#o; hex = %#x\n",x, x, x);
```

```
    return 0;
```

```
}
```



# 数据类型的使用

- 变量和类型匹配
- 声明变量
- 初始化变量
- 类型之间的转化
- 使用系统化的变量命名规则

要定义一个变量存储班上单身狗人数

int 型

假设有 11 个

变量名: singleDogs

```
int singleDog = 10;
```







# 格式说明符

格式说明符	意义
%a(%A)	浮点数、十六进制数字和p-(P-)记数法(C99)
%c	字符
%d	有符号十进制整数
%f	浮点数(包括float和doulbe)
%e(%E)	浮点数，指数输出（e-(E-)记数法）
%g(%G)	浮点数，不显无意义的零"0"
%i	有符号十进制整数(与%d相同)
%u	无符号十进制整数
%o	八进制整数
%x(%X)	十六进制整数
%p	指针
%s	字符串
%%	输出符号"%"

重点：%d, %c, %f,



# scanf() 函数

```
int money;
```

```
scanf("%d", &money);
```

```
char a;
```

```
scanf("%c", &a);
```

```
char name[5];
```

```
scanf("%s", name);
```



# 回顾

- C 的基本程序结构、赋值符“=”、main() 和 printf() 函数、整型变量的声明和赋值、注释
- sizeof 运算符、scanf() 函数、C 的基本数据类型、整型和浮点型的区别、使用 scanf() 和 printf() 函数读写数据类型的值



# 课堂练习

```
include stdio.h
int main{void}
(
int s
s:=56;
print(There are s weeks in a year);
return 0;
```



# 课堂练习

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

```
int main(void)
```

```
{
```

```
    int s; /*tab/
```

```
    s = 52;
```

```
    printf("There are %d weeks in a year\n", s);
```

```
    return 0;
```

```
}
```



# 课堂练习

- `printf("Baa Baa Black Sheep.");`
- `printf("Have you any wool?\n");`
- `printf("Begone!\n0 creature of lard!");`
- `printf("What?\nNo/nBonzo?\n");`
- `int num;`  
`num = 2;`  
`printf("%d + %d = %d ", num, num, num + num);`



# 课堂练习

使用 printf 输出 words 和 lines 的值

words = 3020    lines = 350

```
int words = 3020, lines = 350;  
printf("There were %d words and %d lines", words, lines);
```





# 课堂练习

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

```
int main(void)
```

```
{
```

```
    int a, b;
```

```
    a = 5;
```

```
    b = 2;
```

```
    b = a; // a = ?
```

```
    a = b; // b = ?
```

```
    printf("%d %d", b, a);
```

```
    return 0;
```

```
}
```

如何交换两个变量的值呢？

```
tmp = a;
```

```
a = b;
```

```
b = tmp;
```



# 课堂练习

- '\b'

char

- 1066

int

- 99.44

double

- 0XAA

unsigned int

- 2.0e30

double



# 课堂练习

- |           |     |
|-----------|-----|
| • 012     | %#o |
| • 2.9e05L | %Lf |
| • 's'     | %c  |
| • 100000  | %ld |
| • '\n'    | %c  |
| • 20.0f   | %f  |
| • 0x44    | %#x |



# 课堂练习

\*观察系统处理整数上溢、浮点数上溢和下溢的方式

```
#include <stdio.h>
#include <limits.h>
#include <float.h>
int main(void)
{
    printf("%d %d\n",INT_MAX, INT_MAX+1);
    printf("%d %d\n",INT_MIN, INT_MIN-1);
    printf("%e %e\n",DBL_MAX, DBL_MAX+DBL_EPSILON);
    printf("%e %e\n",DBL_MIN, DBL_MIN-DBL_EPSILON);
    printf("%e %e\n",FLT_MAX, FLT_MAX+FLT_EPSILON);
    printf("%e %e\n",FLT_MIN, FLT_MIN-FLT_EPSILON);
    return 0;
}
```



# 作业

- 第二章: 2.11全部、2.12—1, 5
- 第三章: 3.10—1, 4, 5, 6, 8, 11、3.11—2, 4, 7

