C语言第一讲

简介

- C语言是一种底层语言
- C语言是一种小型语言
- C语言是一种包容性语言

1.2 C语言的特点

语言简洁、紧凑，使用方便、灵活。

只有32个关键字、9种控制语句,程序形式自由。

运算符丰富。34种运算符

数据类型丰富，具有现代语言的各种数据结构。

具有结构化的控制语句，是完全模块化和结构化的语言。

语法限制不太严格，程序设计自由度大。

1.2 C语言的特点

允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可直接对硬件进行操作。

兼有高级语言和低级语言的特点，适合写操作系统。

目标代码质量高，程序执行效率高。只比汇编程序生成的目标代码效率低10%-20%。

程序可移植性好（与汇编语言比）。很少修改就能用于不同型号的计算机和操作系统。

C程序的基本结构

/*.....*/ 表示注释，注释只是给人看的，对编译和运行不起作用

```
/* hello.c */
#include <stdio.h>
int  main(){
    printf("hello, word. \n");
    return 0;
}
```

以#开头的是指令。

#include ，导入标准输入输出库stdio.h，printf 函数来自于该库。

每个C程序必须有一个主函数main，int是函数的返回类型。

{ }是函数开始和结束的标志,不可省略。

每个C语句以分号结束。

return 0：表示程序结束时向操作系统返回值0 。

```
#include〈stdio.h〉
int main()
{int a,i;
a=6;
for(i=1;i< = 3;i++)
printf("%d", a);
return 0;
}
```



```c
#include<stdio.h>
int main(){
int a,i;
a=6;
for(i=1;i<=3;i++)
 printf("%d",a);
 return 0;
}
```

hello world

```
#include <stdio.h>

int main()
{
    printf("Hello World!");
    return 0;
}
```

> #include <stdio.h>
> 头文件，包含C语言标准输入/输出库的相关信息

C语言规定每条语句都需要以分号结尾（当然，也有例外）

注释

文档说明

> 符号/标记注释的开始，符号/标记注释的结束

```
/* This is a comment */
```

注释几乎可以出现在程序的任何位置上。
注释也可以占用多行

```
#include <stdio.h>
/* Author:Bio Sheep
   website:https://algernon98.github.io/
*/

int main(void) //不带参数
{
    printf("Hello World!");
    return 0;
}
```

一般的注释放在代码部分之前，或者 同一行

```
int main(void)
{
    printf("欲买桂花同载酒，终不似，少年游");  /*文本部分*/
    return 0;
}
```

注意：大小写 “ ” "
句尾分号；

另一种注释：

```
int main(void)
{
    printf("欲买桂花同载酒，终不似，少年游");  //文本部分
    return 0;
}
```

这种注释会在行末自动终止

标识符

在编写程序时，需要对变量、函数、宏和其他实体进行命名。这些名字称为标识符（identifier）。

在C语言中，标识符可以含有字母、数字和下划线， 但是必须以字母或者下划线开头。

下面是合法标识符的一些示例：

times10 get_next_char _done

接下来这些则是不合法的标识符：

10times get-next-char

关键字

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

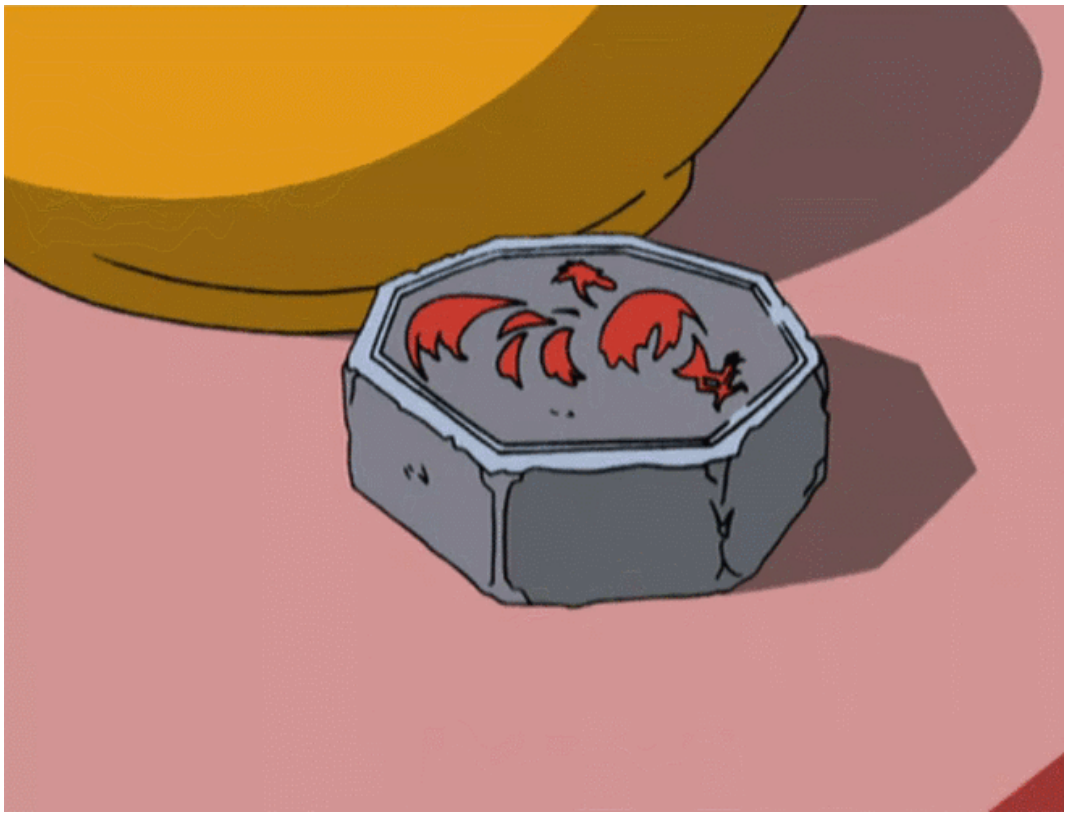
变量

每一种变量都必须有一种类型，用来说明变量所存储的数据的种类。
int型变量可以存储整数，但取值范围有一定限制；float型变量可以存储带小数位的数。//integer floating point

类型	举例
int	0、114、-255、2^31-1、-2^31(2147483648)
float	114.514、-0.11

声明

> 老爹的侄子是一名考古学家，这一天他带回了一个奇妙的小玩意儿，想让见多识广的老爹掌掌眼。但即使是老爹也需要时间研究，于是老爹想定制一个精美的小盒子装下这个看似不同寻常的宝物，你可以设计一个程序帮助老爹确定这个 盒子的体积吗?



首先，我们需要知道盒子的长宽高。
长、宽、高就是程序中的**变量**
在C语言中，使用变量前必须对其进行**声明**

声明变量，首先指定变量类型，然后说明变量名字，
比如我们需要引入两个变量，物体的高(整数)和重量（浮点数)

```
int height;
float weight;
```

赋值

从右往左赋值

```
//对于一个高为8，长为12，宽为10的物体，依此对三个变量赋值：
height =8;
length =12;
width = 10;
```

但是，在对一个变量进行赋值之前，需要确保在此之前已经声明过变量：

```
int height;
float length;
float width;
height =8;
length =12.14159;
width = 10;
```

让我们看看输出：

```
printf("%d",height);
printf("%f",length);
printf("%f",width);
```

```
812.14159010.000000
```

显然，这不是我们想要的结果

我们将代码进行一定调整：

```
int height;
float length;
float width;
height =8;
length =12.14159;
width = 10;
printf("height=%d\n",height);
printf("length=%f\n",length);
printf("width=%f",width);
```

输出：

```
height=8
length=12.141590
width=10.000000
```

可以看到，在不指定保留小数位数的情况下，浮点数默认六位小数

声明的时候，我们也可以一并赋值，这样可以对变量进行**初始化**

```
int height=8,length=12,width=10; //分别对高、长和宽初始化
int height,length,width=10;//声明三个变量，但仅对宽初始化
```

格式化输入输出

> 老爹最近很烦！上一个古董还没有研究完，这个不省心的侄子又给他带了好几个小玩意儿，要知道，带有魔力的盒子是很不好定制的，不同的古董也有不同的参数，你能否改进程序，可以根据老爹的要求来求出适合的盒子体积呢?

为了达到老爹的要求，这套程序需要允许用户自行录入尺寸。

获取用户输入的函数是scanf函数，而输出使用printf函数
二者都需要使用格式串来指定输入或输出数据的形式。

格式化字符串（简称：格式串）包含普通字符和转换说明（conversion specification），转换说明以%开头。

%d 表示把int型数值从二进制转换成十进制数字组成的字符串。

%f 表示对float型的转换。

读入一个int型值：

```
scanf("%d".&i); //将输入值存入int型变量i中
```

同样的，float型值的读入：

```
scanf("%f".&j); //将输入值存入float型变量j中
```

改进的程序：

```
#include <stdio.h>
/* Author:Bio Sheep
   website:https://algernon98.github.io/
*/

int main(void)
{
int height,length,width,volume;
printf("输入盒子的高:") ;
scanf("%d",&height);
printf("输入盒子的长:") ;
scanf("%d",&length);
printf("输入盒子的宽:") ;
scanf("%d",&width);
volume=height*length*width;
printf("盒子的体积是: %d",volume);
    return 0;
}
```

输出：

```
输入盒子的高:1
输入盒子的长:2
输入盒子的宽:3
盒子的体积是: 6
```

如果我们希望保留三位小数呢?

```
printf("盒子的体积是: %.3f",volume);
```

转换说明

我们可以用%.1f来显示小数点后带一位数字的float型值。
%m.pX形式
m是最小 字段宽度，p是精度，X是转换说明符
对于X：

- d表示十进制形式的整数

- e表示指数（科学计数法）形式的浮点数
- f是浮点数（没有指数）

```
%d
%5d
%-5d
%5.3d
%5.3f
%5.3e
```

转义序列

格式串中常用的“\n”，被称为转义序列（Escape Sequence）

常用的：

警报（响铃）符：\a

回退符：\b

换行符：\n

水平制表符：\t

> 换行符 ： \n

```
printf("欲买桂花同载酒，终不似，少年游");
printf("欲买桂花同载酒，\n终不似，\n少年游。");
```

输出：

> 欲买桂花同载酒，终不似，少年游欲买桂花同载酒，
> 终不似，
> 少年游。

修改：

```
printf("欲买桂花同载酒，终不似，少年游\n");
printf("欲买桂花同载酒，\n终不似，\n少年游。");
```

```
欲买桂花同载酒，终不似，少年游
欲买桂花同载酒，
终不似，
少年游。
```

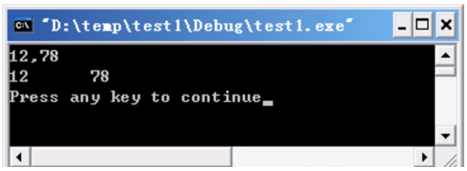
转义序列 " ，表示字符"

```
printf("老爹说：\"还有一件事!\"");
```

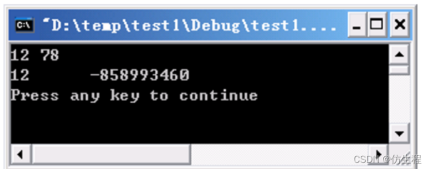
```
#include <stdio.h>
  void main()
{
    int num1,num2;
    scanf("%d,%d",&num1,&num2);
   printf("%d\t%d\n",num1,num2);
}
```

scanf的格式串用逗号，分割两个数据，所以输入必须用逗号分隔，否则就会读入错误数据

程序输出结果:



注意输入时避免如下错误：



表达式

算数运算符

一元运算符

+	-

二元运算符

加法类	乘法类
+加法运算符	*乘法运算符
-减法运算符	/除法运算符
	%求余运算符

> 10%3的值是1,10/3的值是3

复合赋值

```
i=i+2;
```

可以简写为:

```
i+=2;
```

其他同上

> v-=e表示v加上e，然后将结果存储到v中

自增、自减

```
i=i+1;
j=j-1;
```

可以利用自增运算符（++）和自减运算符（--）缩短为：

```
i++;
j--;
```

> 思考：++i和i++有什么区别？
>
> **只需一个操作数**
>
> **幅度为1**
>
> **前缀使用 ++i; --i;**
>
> **i立即自增或自减；如果有，再做其它。**
>
> **后缀使用 i++; i--;**
>
> **先用i的原始值；随后（在本语句执行结束前）i再自增或自减。**
>
> **整型变量或浮点型变量都可以做自增或自减运算**
>
>

编程题，两位数逆序打印

```
#include <stdio.h>
int main(){
int i=1,n;
n=i++;
printf("%d\n",n);
printf("%d",i);
return 0;
}
```

2-7: 计算账单 (Bills)

编写一个程序，要求用户输出一个美元数量，然后显示出如何用最少20美元、10美元、5美元和1美元来付款。

```
Enter a dollar amount: 93
$20 bills: 4
$10 bills: 1
$5 bills: 0
$1 bills: 3
```

选择语句

- 选择语句 if switch
- 重复语句 while do for 循环
- 跳转语句 break continue goto

逻辑表达式

关系运算符

在C语言中，诸如i<j这种比较运算会产生整数：
0（假）或1（真）
10<11的值为1（true）
11<10的值为0（false）

符号	含义
<	小于
>	大于
<=	小于等于
>=	大于等于

判等运算符

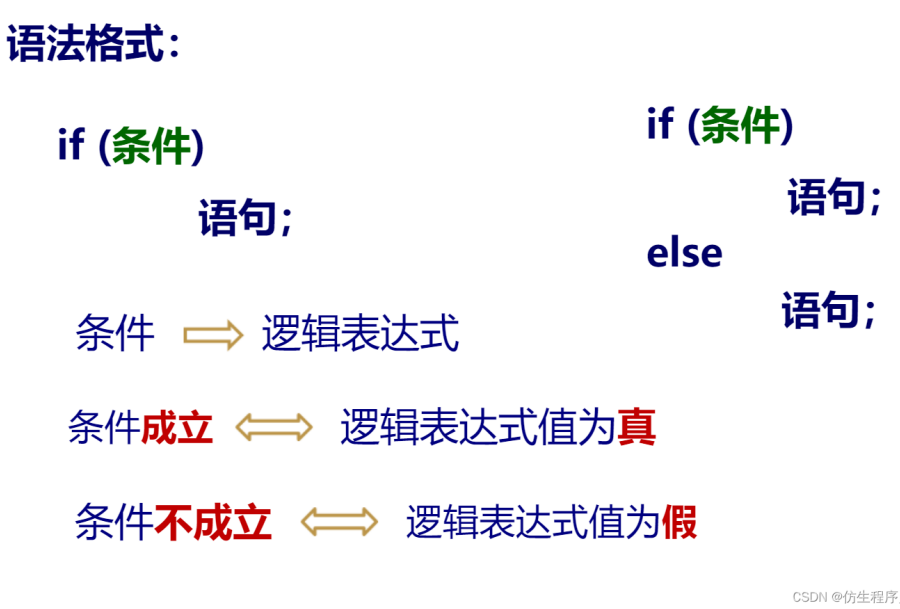
符号	含义
==	等于
! =	不等于

逻辑运算符

符号	含义
!	逻辑非
&&	逻辑与

符号	含义
||	逻辑或

if 语句




```
if(表达式)语句
```

判定0<i<n是否成立：

```
if (0<i && i<n)
```

复合语句：

```
if (i>0) {
i--;
j++;
}
```

交换两个数

例4：输入两个数，存入变量a、b，比较两个数大小，将较大的一个数存储在a中，较小数存在b中

步骤：

定义两个变量存储该数据；

接受用户输入，存储在该变量中；

判断a和b中值的大小，将较大值存入a ，较小值存入b ；

输出判断结果

```
if (a < b)
{
    t = a;
    a = b;
    b = t;
}
```



```
if(a<b){
     t=a;
     a=b;
     b=t;

}
```

else语句

```
//获取两个数中的最大值
if  (i>j)
  max=i; //把i赋给max
else
  max=j;
```

if嵌套

语法格式：

```
if (条件)
        语句;
else
        语句;
```

利用缩进匹配else和if

```
if (条件)
        语句;
else    if (条件)
                语句;
        else
                语句;
```

完整的if 语句


```
#include <stdio.h>
/* Author:Bio Sheep
   website:https://algernon98.github.io/
*/

int main(void)
{
```

```
int i=5,j=4,max;
if(i>0){
   if(i>j){
   max=i;
   }
   else{
   max=j;
   }
}
else{
max=j;
}
printf("%d",max);
    return 0;
}
```

级联式if语句

```
if  (i>j)
   printf("i is greater than j\n");
else if (i<j)
   printf("i is less than j\n");


else
   printf("i is equal to  j\n")
```

switch语句

格式：

switch(e)

{ case 常量表达式c_1： 语句1

case 常量表达式c_2： 语句2

...

case 常量表达式c_n ： 语句n

default: 语句n+1 }

```
switch(grade){

case 4:
case 3:
case 2:
case 1: printf("passing");
        break;
case 0: printf("falling");
       break;

}
```

多个case语句可以共用一组执行语句:

```
switch (grade) {
  case 4: case 3: case 2: case 1:
        printf("Passing");
        break;
  case 0:  printf("Failing");
        break;
  default: printf("Illegal grade");
        break;
}
```

Grade	Level
4	Passing
3	
2	
1	
0	Failing
others	Illegal grade

break语句

break语句会使得程序“跳”出switch语句，继续执行switch后面的语句