

变量

翁恺

做计算

```
printf("23+43=%d\n", 23+43);
```

算找零

```
printf("23+43=%d\n", 23+43);
```

```
printf("100-23=%d\n", 100-23);
```

- 如何能在程序运行时输入那个数字23，然后计算输出结果？

算找零

```
printf("23+43=%d\n", 23+43);
```

```
printf("100-23=%d\n", 100-23);
```

- 如何能在程序运行时输入那个数字23，然后计算输出结果？

需要：

1. 有地方放输入的数字；
2. 有办法输入数字；
3. 输入的数字能参与计算。

change.c

```
int price = 0;  
  
printf("请输入金额 (元) : ");  
scanf("%d", &price);  
  
int change = 100 - price;  
  
printf("找您%d元。 \n", change);
```

今后课件上的程序都只有main的{}里面的部分

如何输入

- 输入也在终端窗口中
- 输入是以行为单位进行的，行的结束标志就是你按下了回车键。在你按下回车之前，你的程序不会读到任何东西

变量

- `int price = 0;`
- 这一行，定义了一个变量。变量的名字是 `price`，类型是 `int`，初始值是 `0`。
- 变量是一个保存数据的地方，当我们需要在程序里保存数据时，比如上面的例子中要记录用户输入的价格，就需要一个变量来保存它。用一个变量保存了数据，它才能参加到后面的计算中，比如计算找零。

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

变量定义

- 变量定义的一般形式就是：
 - `<类型名称> <变量名称>;`
- `int price;`
- `int amount;`
- `int price, amount;`

变量的名字

- 变量需要一个名字，变量的名字是一种“标识符”，意思是它是用来识别这个和那个的不同的名字。
- 标识符有标识符的构造规则。基本的原则是：标识符只能由字母、数字和下划线组成，数字不可以出现在第一个位置上，C语言的关键字（有的地方叫它们保留字）不可以用做标识符。

C语言的保留字

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,
inline,restrict

不需要背诵!

赋值和初始化

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

- `int price = 0;`
- 这一行，定义了一个变量。变量的名字是 `price`，类型是 `int`，初始值是 `0`。
- `price=0` 是一个式子，这里的“=”是一个赋值运算符，表示将“=”右边的值赋给左边的变量。

赋值

- 和数学不同， $a=b$ 在数学中表示关系，即a和b的值一样；而在程序设计中， $a=b$ 表示要求计算机做一个动作：将b的值赋给a。关系是静态的，而动作是动态的。在数学中， $a=b$ 和 $b=a$ 是等价的，而在程序设计中，两者的意思完全相反

初始化

- 当赋值发生在定义变量的时候，就像程序1中的第7行那样，就是变量的初始化。虽然C语言并没有强制要求所有的变量都在定义的地方做初始化，但是所有的变量在第一次被使用（出现在赋值运算符的右边）之前被应该赋值一次
- 如果没有初始化？

变量初始化

- `<类型名称> <变量名称> = <初始值>;`
 - `int price = 0;`
 - `int amount = 100;`
- 组合变量定义的时候，也可以在这个定义中单独给单个变量赋初值，如：
 - `int price = 0, amount = 100;`

读整数

- `scanf("%d", &price);`
- 要求scanf这个函数读入下一个整数，读到的结果赋值给变量price
- 小心price前面的&

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

如果输入的不是整数会怎样?

表达式

- “=”是赋值运算符，有运算符的式子就叫做表达式。
- `price=0;`
- `change=100-price;`

变量类型

- `int price = 0;`
- 这一行，定义了一个变量。变量的名字是 `price`，类型是 `int`，初始值是 `0`。
- C是一种有类型的语言，所有的变量在使用之前必须定义或声明，所有的变量必须具有确定的数据类型。数据类型表示在变量中可以存放什么样的数据，变量中只能存放指定类型的数据，程序运行过程中也不能改变变量的类型。

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

第二个变量

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

- `int change = 100 - price;`
- 定义了第二个变量`change`
- 并且做了计算

C99!

ANSI C

- 只能在代码开头的地方定义变量

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

C99

```
int price = 0;
int change = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

change = 100 - price;

printf("找您%d元。 \n", change);
```

ANSI C

常量

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

- `int change = 100 - price;`
- 固定不变的数，是常数。直接写在程序里，我们称作直接量（literal）。
- 更好的方式，是定义一个常量：
 - `const int AMOUNT = 100;`

C99!

常量

```
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = 100 - price;

printf("找您%d元。 \n", change);
```

```
const int AMOUNT = 100;
int price = 0;

printf("请输入金额 (元) : ");
scanf("%d", &price);

int change = AMOUNT - price;

printf("找您%d元。 \n", change);
```

- `int change = 100 - price;`
- 100这个固定不变的数，叫做常数。直接写在程序里，我们称作直接量（literal，字面量）。
- 更好的方式，是定义一个常量变量：
 - `const int AMOUNT = 100;`

C99!

const

- `const`是一个修饰符，加在`int`的前面，用来给这个变量加上一个`const`（不变的）的属性。这个`const`的属性表示这个变量的值一旦初始化，就不能再修改了。
- `int change = AMOUNT - price;`
- 如果你试图对常量做修改，把它放在赋值运算符的左边，就会被编译器发现，指出为一个错误。

C99!

try

- 让用户输入变量AMOUNT的值，而不是使用固定的初始值
- 这个变量在哪里定义合适呢？

tips

- 程序要求读入多个数字时，可以在一行输入，中间用空格分开，也可以在多行输入
- 在scanf的格式字符串中有几个%d，它就等待用户输入一个整数，当然，字符串后面也需要对应有那么多个整数

plus.c

```
int a;  
int b;  
  
printf("请输入两个整数: ");  
scanf("%d %d", &a, &b);  
printf("%d + %d = %d\n", a, b, a + b);
```

身高5尺7寸？

- 美国人固执地使用英制计量单位，他们习惯用几尺几寸（英尺英寸）来报自己的身高。如果遇到一个美国人告诉你他是5英尺7，他的身高应该是一米几呢？
- $(5 + 7 \div 12) \times 0.3048 = 1.7018$ 米

计算身高的程序

```
printf("请分别输入身高的英尺和英寸，"  
      "如输入\"5 7\"表示5英尺7英寸：");  
  
int foot;  
int inch;  
  
scanf("%d %d", &foot, &inch);  
  
printf("身高是%f米。\\n",  
      ((foot + inch / 12) * 0.3048));
```

运行一下

- 为什么总是1.524米？

因为...

- 因为两个整数的运算的结果只能是整数
 - $10/3*3=>?$
- 10和10.0在C中是完全不同的数
- 10.0是浮点数

浮点数

- 带小数点的数值。浮点这个词的本意就是指小数点是浮动的，是计算机内部表达非整数（包含分数和无理数）的一种方式。另一种方式叫做定点数，不过在C语言中你不会遇到定点数。人们借用浮点数这个词来表达所有的带小数点的数。

改进

- $(\text{foot} + \text{inch} / 12) * 0.3048;$



- $(\text{foot} + \text{inch} / 12.0) * 0.3048;$

- 当浮点数和整数放到一起运算时，C会将整数转换成浮点数，然后进行浮点数的运算。

double

- `inch`是定义为`int`类型的变量，如果把`int`换成`double`，我们就把它改为`double`类型的浮点数变量了。
- `double`的意思是“双”，它本来是“双精度浮点数”的第一个单词，人们用来表示浮点数类型。除了`double`，还有`float`（意思就是浮点！）表示单精度浮点数。

另一个版本

```
printf("请分别输入身高的英尺和英寸，"  
      "如输入\"5 7\"表示5英尺7英寸：");  
  
double foot;  
double inch;  
  
scanf("%lf %lf", &foot, &inch);  
  
printf("身高是%f米。\\n",  
      ((foot + inch / 12) * 0.3048));
```

数据类型

- 整数
 - int
 - printf("%d",...)
 - scanf("%d",...)
- 带小数点的数
 - double
 - printf("%f",...)
 - scanf("%lf",...)

还有更多的，以后再仔细研究！

整数

- 整数类型不能表达有小数部分的数，整数和整数的运算结果还是整数。计算机里会有纯粹的整数这种奇怪的东西，是因为整数的运算比较快，而且占地方也小。其实人们日常生活中大量做的还是纯粹整数的计算，所以整数的用处还是很大的。