

第三章

最简单的C程序设计





赋值语句

编程时需要解决的问题

(1) 赋值语句

赋值表达式;

本例可用语句:

a=3;

b=7;

s=a*b;

l=(a+b)*2;

输出示例:

`printf("%d\n",s);`

`printf("%d\n",l);`

`printf("%d,%d\n",s,l);`

`printf("面积=%d,周长=%d\n",s,l);`

输出函数:

`printf(格式控制,输出项列表)`

(3) 输出





赋值语句

■ 程序代码

```
#include<stdio.h>
void main()
{int a,b,s,l;
 a=3;
 b=7;
 s=a*b;
 l=(a+b)*2;
 printf("s=%d,l=%d\n",s,l);
}
```

运行结果：
s=21,l=20



数据的输入、输出—输入、输出概念



1. 输入:

为程序提供数据。

数据流向程序。

为程序提供原始数据。

2. 输出:

数据由程序流向外界。

将程序的计算结果通知
编程者或使用者。

```
#include "stdio.h"
#include "math.h"
#define PI 3.14
void main( )
{
    float r, s; s=12.5
    scanf("%f", &s).↑
    r=sqrt(s/PI);
    printf("半径为: %f
", r);
}
```



数据的输入、输出—输入、输出实现



1. 方法:

使用系统函数。

scanf, printf, putchar, getchar等。

2. #include命令

功能： 使用将函数库包含到用户程序中。

位置： 在主函数前面。

格式： **#include “stdio.h”**

#include <stdio.h>



数据的输入、输出—格式输入、输出



■ 格式输出

printf(格式控制, 输出项列表)

编程者
指定的样式

例： **printf(“面积=%d,周长=%d\n”, s,l);**

格式控制：用双引号括起来的一串字符。



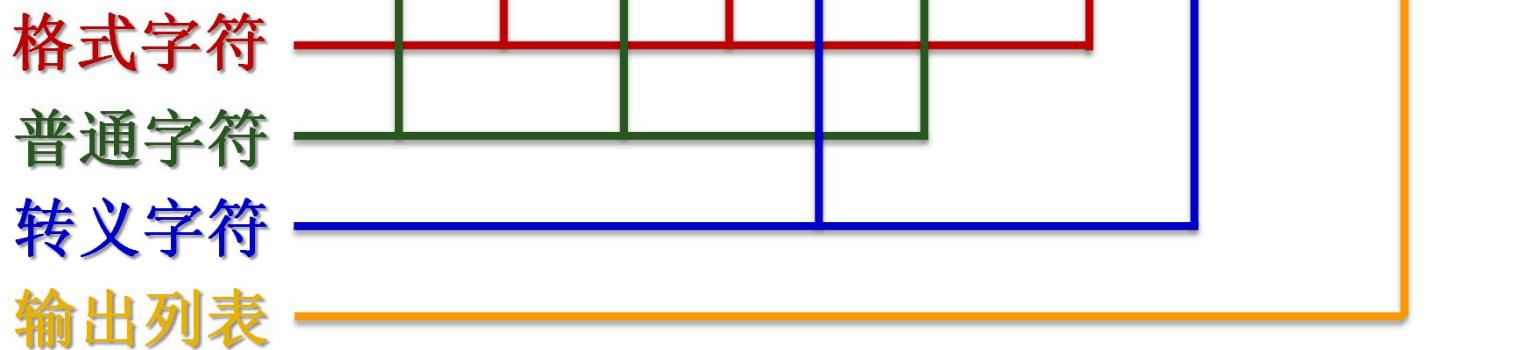
数据的输入、输出—格式输入、输出



■ 格式输出

输出项列表：用逗号间隔的常量、变量、表达式

```
printf("x=%d,y=%d\nx+y=%d\n",3,4,3+4)
```



```
printf("x=%d,y=%d\n,x+y=%d\n", 3 , 4 , 3+4 )
```

x=3 ,y= 4

x+y= 7



数据的输入、输出—格式输入、输出



■ 格式输出

格式符：

整型数据格式： **%d** (%**o**,%**x**,%**ld**)

实型数据格式： **%f,%e,%lf**

字符型数据格式： **%c**

字符串格式： **%s**

实例：

```
#include "stdio.h"
void main( )
{ float r=25.34;
  int a=30
  char m= 'R' ;
```

```
printf("%f, %d, %c",
```



数据的输入、输出—格式输入、输出



%d的使用

```
#include <stdio.h>
void main()
{int a=123,b=567;
printf("%d%d\n",a,b);
printf("%6d%6d\n",a,b);
printf("%-6d%-6d\n",a,b);
printf("%d,%d\n",a,b);
}
```



数据的输入、输出—格式输入、输出



%c的应用

```
#include <stdio.h>
void main()
{char c='a';
int m=97;
printf("c=%c,c=%d\n",c,m);
printf("m=%c,m=%d",c,m);
}
```

- 字符数据可用“%d”格式输出；
- 整型数据可用“%c”格式输出。



数据的输入、输出—格式输入、输出



%f的应用

```
#include <stdio.h>
void main()
{float a=1234.123456
printf("%f\n",a),
printf("%.3f\n",a),
printf("%12.4f\n",a);
printf("%-12.4f\n",a);}
```

按数据

.3

保留小数位数

12

-

指定的宽度大于数据的实际宽度时，在数据后面补空格(左对齐)。



数据的输入、输出—格式输入、输出



%s的应用

```
#include <stdio.h>
void main()
{ printf("%s\n","china");
  printf("%7s\n","china");
  printf("%-7s\n","china");
}
```

输出结果

china

china

china



数据的输入、输出—格式输入、输出



■ 格式输入

scanf(格式控制，地址表列)

例： int a,b;

~~scanf("%d%d",&a,&b);~~

~~scanf("%d,%d",&a,&b)~~

使用与printf函数相同的

地址
变

用逗号做数据
间的间隔符

两种方
法的区别？

用空格做数据
间的间隔符



数据的输入、输出—格式输入、输出



■ scanf() 函数执行过程

读取数据：

从输入数据中按指定格式

最简洁实用
的输入格式

赋值：

将读取的数据赋值给地址列表中相应的变量。

`scanf("%d%d%d",&a,&b,&c)`

数据输入格式为： 2 3 4



数据的输入、输出—格式输入、输出



例：

```
#include "stdio.h"
void main()
{ float a,b,c;
scanf("%f%f",&a,&b);
c=a+b;
printf("%f,%f,%f\n",a,b,c);
}
```



数据的输入、输出—格式输入、输出



例：

```
#include "stdio.h"  
void main()  
{char a,b,c;  
scanf("%c%c",&a,&b);  
a=a+32;  
b=b-32;  
printf("%c,%c,\n",a,b);  
}
```

用“%c”格式读
数据时，数据间
不用间隔符。



数据的输入、输出—字符输入、输出



■ 输出字符函数

用法: **putchar(字符数据)**

功能: 一次输出一个字符

输出单个字符

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

```
void main()
```

```
{char a,b,c;
```

```
a=getchar();
```

```
b=getchar();
```

```
c=getchar();
```

```
putchar(a);putchar(b);
```

```
putchar(c);putchar('\n');}
```

■ 输入字符函数

用法: 字符变量=**getchar()**

功能: 一次读入一个字符





顺序结构程序设计

■ 顺序结构：

最简单的程序结构。

特点：完全按语句的排列顺序执行程序。

构成：赋值语句、输入、输出。

如：**#include "stdio.h"**

void main()

{int a,b,c;

scanf("%d%d",&a,&b);

c=a+b;

printf("%d,%d,%d\n",a,b,c);

}





顺序结构程序设计

例3：从键盘输入一个三位数，正确分离出它的个位、十位和百位，然后求三个数之和。

■ 问题的关键：

如何分离一个数的每一位？

个位： $m \% 10$

十位： $m / 10 \% 10$

百位： $m / 100$

如何确定m的值？

■ 数据结构：5个变量，整型；



本章小结

