

# 第三讲 格式化输入 / 输出

## 3.1 printf函数

■功能：显示**格式串**的内容

```
printf (string, expr1, expr2, ...);
```

■格式串包括：普通字符、转换说明

■普通字符：原样输出

■转换说明：以字符“%”开头，表示输出过程中**待填充值的占位符**

■%d：按整数格式进行填充

■%f：按浮点数格式进行填充

# 3. 格式化输入 / 输出

## 3.1 printf函数

```
/* 03-01-printf.c */
#include <stdio.h>
int main(void)
{
    int i, j;
    float x, y;
    i = 10;
    j = 20;
    x = 43.2892f;
    y = 5527.0f;
    printf("i = %d, j = %d, x = %f, y = %f\n", i, j, x, y);

    return 0;
}
```

```
i = 10, j = 20, x = 43.289200, y = 5527.000000
```

## 3.1 printf函数

■ 编译器不会检查转换说明的**数量**和**类型**是否与输出项相匹配

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

```
printf("%d\n", i, j);
```

```
printf("%f %d\n", i, x);
```

# 3. 格式化输入 / 输出

## 3.1 printf函数

### ■ 转换说明格式化信息

■ %m.pX or %-m.pX

■ m, p都是可选的

■ %10.2f: m=10, p=2, X 为f

■ %.2f: m未定义, p=2, X 为f

■ m: 最小字段宽度

■ %4d 将数值123显示为·123(·代表空格)

■ %-4d 将数值123显示为123·(·代表空格)

■ p: 精度

■ X=f/e时, p代表小数点后数字的个数

■ X=d时, 整数位数, 不够前面补零

```
/*03-02tprintf.c*/
/* Prints int and float values in various formats */
#include <stdio.h>
int main(void)
{
    int i;
    float x;

    i = 40;
    x= 839.21f;

    printf("|%d|%5d|%-5d|%5.3d|\n", i, i, i, i);
    printf("|%10.3f|%10.3e|%-10g|\n", x, x, x);

    return 0;
}
```

40	40	40	040
839.210	8.392e+002	839.21	

## 3.1 printf函数

### ■ 转义序列

■ 换行符: \n

■ 水平制表符: \t

■ 字符 " : \"

■ 字符 \ : \\

■ 字符 ' : \'

```
printf("Item\tUnit\tPurchase\n\tPrice\tDate\n");
```

Item	Unit	Purchase
Price	Date	

```
printf("\\"Hello!\\\"");
```

"Hello!"

## 3.1 scanf函数

- 根据特定的格式读取输入
- 格式串包括：普通字符、转换说明

```
#include <stdio.h>
int main(void)
{
    int i, j;
    float x, y;

    scanf("%d%d%f%f", &i, &j, &x, &y);
    return 0;
}
```

1 -20 .3 -4. 0e3

- 将1, -20, 0.3, -4000.0 分别赋值给i,j,x,y四个变量

### ■注意

- 转换说明的数量需要与输入变量的数量相匹配
- “&” 符号

## 3.1 scanf函数

### ■ scanf函数的工作方法

- 用户输入的是字符串
- 从左边开始处理字符串中信息
- 对于格式串中每一个转换说明，scanf函数从输入数据中定位适当类型的项，并在必要时跳过空格，然后scanf读入数据项，并在不可能属于此项的字符时停止。
- 如果读入数据项成功，那么scanf函数继续处理格式串剩余部分
- 如果一项不能成功读入，那么scanf函数将不再查看格式串的剩余部分
- 寻找数的起始位置时，会忽略空白字符（包括空格、制表符、换行符、换页符）

## 3.1 scanf函数

```
scanf ("%d%d%f%f", &i, &j, &x, &y);
```

```
1  
-20      .3  
-4.0e3
```

```
••1•-20•••.3•••-4.0e3•  
ssrsrrrrssssrrssssrrrrrr (s = skipped; r = read)
```

```
1-20.3-4.0e3
```

## 3.1 scanf函数

■ 格式串包括：普通字符（原样输入）、转换说明

■ 匹配机制

■ 如果输入和格式串不匹配，函数就会返回，不匹配的输入字符将被“放回”留待以后读取

■ 转换说明：转换说明（%[，%c，和%n除外）会跳过输入项开始处的空白字符；转换说明不会跳过尾部的空白字符。

■ 空白字符：一个或多个连续的空白字符与输入流中的零个或多个空白字符相匹配

■ 非空白字符：除了%之外的非空白字符和输入流中的相同字符相匹配

# 3. 格式化输入 / 输出

## 3.1 scanf函数

	输入	变量
<code>n = scanf ("%d%d", &amp;i, &amp;j);</code>	<code>12•,•34↙</code>	<code>n: 1</code> <code>i: 12</code> <code>j: 不变</code>
<code>n = scanf ("%d,%d", &amp;i, &amp;j);</code>	<code>12•,•34↙</code>	<code>n: 1</code> <code>i: 12</code> <code>j: 不变</code>
<code>n = scanf ("%d ,%d", &amp;i, &amp;j);</code>	<code>12•,•34↙</code>	<code>n: 2</code> <code>i: 12</code> <code>j: 34</code>
<code>n = scanf ("%d, %d", &amp;i, &amp;j);</code>	<code>12•,•34↙</code>	<code>n: 1</code> <code>i: 12</code> <code>j: 不变</code>

# 3. 格式化输入 / 输出

## 3.1 scanf函数

	输入	变量
<code>n = scanf("%*d%d", &amp;i);</code>	<code>12•34↙</code>	n: 1 i: 34
<code>n = scanf("%*s%s", str);</code>	<code>My•Fair•Lady↙</code>	n: 1 str: "Fair"
<code>n = scanf("%1d%2d%3d", &amp;i, &amp;j, &amp;k);</code>	<code>12345↙</code>	n: 3 i: 1 j: 23 k: 45
<code>n = scanf("%2d%2s%2d", &amp;i, str, &amp;j);</code>	<code>123456↙</code>	n: 3 i: 12 str: "34" j: 56

## ■课程内容

0. 引言

1、C语言概述

2、C语言基本概念

3、格式化输入输出

### 4、表达式

5、选择语句

6、循环语句

7、基本类型

8、数组

9、函数

10、程序结构

11、指针

12、指针和数组

13、字符串操作

14、预处理器

15、编写大型程序

16、结构、联合和枚举

17、指针的高级应用

18、声明

19、程序设计

20、底层程序设计