

# 常量与变量

DATA TYPES AND EXPRESSIONS  
2 数据类型与表达式



## 编程：计算圆的周长和面积

- 输入：半径
- 输出：圆的周长和面积
- 计算公式： $AREA = PI * R * R$

$$CIRCUM = 2 * PI * R;$$



## 计算圆的周长和面积

Input r: 5.3 ✓  
circumference = 33.30  
area = 88.24

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

```
int main()
```

```
{
```

```
float r;
```

```
float circum;
```

```
float area;
```

```
printf("input r:\n");
```

```
scanf_s("%f", &r);
```

```
circum = 2 * 3.1415926 * r;
```

```
area = 3.1415926 * r * r;
```

```
printf("circumference = %.2f\n", circum);
```

```
printf("area = %.2f\n", area);
```

```
system("pause");
```

```
return 0;
```

```
}
```

常数

## 使用常数存在的问题？

- 假如直接使用常数，会有什么影响？
  1. 程序的可读性变差
  2. 容易发生书写错误
  3. 当常数需要改变时，要修改所有引用它的代码，工作量大，还可能有遗漏
- 解决方案：
  1. 避免使用常数
  2. 把常数定义为常量（宏常量、CONST常量.....）



## 计算圆的周长和面积

//编译时，宏替换为  $2 * 3.1415926 * r$

//编译时，宏替换为  $3.1415926 * r * r$

```
#include <stdio.h>
#define PI 3.1415926 //定义宏常量PI
int main()
{
    float r;
    float circum;
    float area;
    printf("input r:\n");
    scanf_s("%f", &r);
    circum = 2 * PI * r;
    area = PI * r * r;
    printf("circumference = %f\n", circum);
    printf("area = %f\n", area);
    system("pause");
    return 0;
}
```



## 计算圆的周长和面积

```
#include <stdio.h>
#define PI 3.14159;
#define R 5.3;
main()
{
    printf("area = %f\n", PI * R * R);
    printf("circumference = %f\n", 2 * PI * R);
}
```

相当于执行

```
#include <stdio.h>
main()
{
    printf("area = %f\n", 3.14159*5.3*5.3);
    printf("circumference = %f\n", 2*3.14159*5.3);
}
```



## const常量

```
#include <stdio.h>
int main()
{
    const double pi = 3.1415926;
    float r;
    float circum;
    float area;
    printf("input r:\n");
    scanf_s("%f", &r);
    circum = 2 * pi * r;
    area = pi * r * r;
    printf("circumference = %f\n", circum);
    printf("area = %f\n", area);
    system("pause");
    return 0;
}
```

## 结 论

const常量与宏常量相比的优点是什么？

- const常量有数据类型
- 某些集成化调试工具可以对const常量进行调试



## 2.2 常量

1. 数值常量：整型常量、浮点数常量
2. 字符型常量：字符常量、字符串常量、转义字符、符号常量

## 2.2.1 整型常量

- 三种不同的数制

十进制	八进制	十六进制
65	0101	0x41
125	0175	0x7d

1. 长整型：-56l ; 3567L ; 04L
2. 无符号整型：586u ; 077765U ; 0xfffdU
3. 无符号长整型：0375ul ; 0x45feUL

## 2.2.2 浮点数常量

浮点数是实数在计算机中的表现形式，只能以十进制数表示。

- 小数形式：3.141; -25.3; 23.; 0.45
- 指数形式:
- (整数部分).(小数部分)e(或E)(指数部分)

如:  $3.14 \times 10^{-3}$  可表示为：

3.14e-3; 0.314e-2; 31.4E-4



## 2.2.3 字符常量

- 字符常量是指用单引号括起来的一个字符。
- 字符按其对应的ASCII码值(附录C)存储，占一个字节。

'a' 97 ; 'A' 65 ; '9' 57 ; '0' 48

- 字符常量可以像整数一样在程序中参与运算。

'a' -32          等价于          97-32=65

'9' -9          等价于          57-9=48

'A' +32          等价于          65+32=97

## 2.2.4 字符串常量

用双引号括起的一串字符。

在内存中存储时，末尾自动加“串结束标识符号”，  
NULL 或 \0 (ASCII码值为0)。

例如：“China” 占6个字节的存储形式如下：

C h i n a NULL

67	104	105	112	97	\0
----	-----	-----	-----	----	----

## 2.2.5 转义字符

- 表示字符的一种特殊形式。表示不可打印字符或具有特定用途的字符。例如：

\a 响铃 007      \" 双引号 034

\n 换行 010      \ddd 三位八进制数

\\ 反斜杠 092      \xhh 二位十六进制数

\' 单引号 039

- 用\ddd 或\xhh形式可以表示任意字符。如：

A 65 \101 \x41

5 53 \065 \x35

Ω 234 \352 \xea

## 2.2.6 符号常量

将常量定义为一个标识符,称为符号常量。通常用大写表示。

如：

```
#define PI 3.1415926
```

```
#define TRUE 1
```

```
#define STAR '*'
```

便于程序调试和修改

## 2.3 变量

在程序执行过程中其值可以改变的量。

### 2.3.1 变量定义

变量必须先定义后使用,定义时包括说明:

数据类型

变量名称

例如：

```
int num;
```

```
char ch1, ch2;
```

```
double length;
```

- 名称一般用小写字母,符合标识规定，不能与关键字同名。

length


num


ch1

--

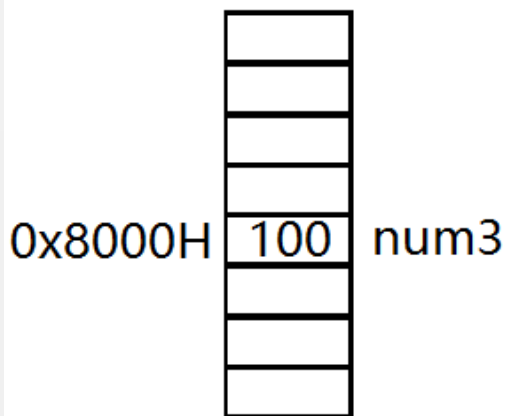
ch2

--





## 2.3.2 变量的初始化



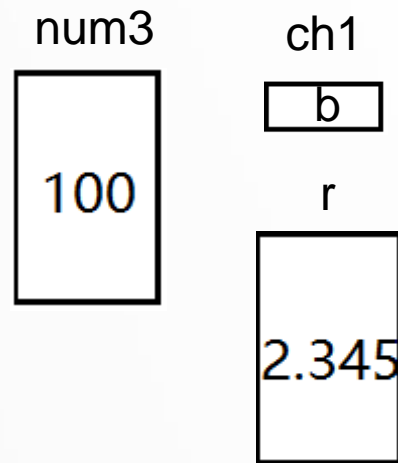
- 定义变量时对变量赋初值。

例如：`int num2, num3=100;`

`char ch1= 'b' , ch2;`

`float r=2.345;`

使程序简洁,提高程序的可读性。



## 2.3.3 变量地址

- 存放变量的内存空间中的首单元地址称为**变量地址**。  
存放的内容称为**变量的值**。