

算术运算符和表达式

一元运算符，二元运算符，唯一一个三元运算符

其的结合性，以及结合的优先性

整数的除法和浮点数的除法的区别，求余运算必须求两个整数进行。

余数与被除数的结果相同

简单赋值运算符和多重赋值

简单赋值运算 =

符合赋值运算 += *= /= %= -=

变量值x 运算符 op = 表达式

变量x = 变量x 运算符op 表达式;

a+= b

a = a+b;

自增和自减，增1，减1

放在变量的前面是前缀 ++a

放在变量的后面是后缀 a--

看法，从左向右执行完这一句的结果和其实现的顺序

比如：

m = ++n; m = 3

m = n++; m = 4

另外，不能对表达式进行自加或者自减的操作

比如：m = -n++其的实际意思为：m = - (n++)

宏常量和宏替换

啥是幻数？

在程序中直接使用的常数就是幻数。

可以使用宏定义常量来解决这个问题

name宏名

#define name sizemax 宏替换过程

宏替换不做任何的语法检查，只有在对已展开后的源程序进行编译的时候才会发现语法错误。

宏定义的宏名和字符串之间可以有多个空白符，

但是无需增加等号，并且无需增加分号，

因为宏定义不是C语句，只是一种编译命令，编译器并不对其进行语法检查

可以使用来完常数这个问题

宏定义没有数据类型，编译器不对其进行检查。容易产生错误。

const常量，只读，不允许进行修改，称为const常量

方式如下const double pi = 3.1415926

自动类型转换和强制类型转换

表达式中的自动转换，类型提升，就是自动转换

类型提升的自动转换

long	double	double	float	unsigned long	long
unsigned			int	char, short	

一般的long>unsigned>double>float>int>short, char

类型下降的自动转换

经常会发生自动转换的类型的格式

比如

低类型 = 高类型 保留前位，与存储有关

int = float 小数部分丢失，整数部分四舍五入

强制类型转换

（类型）表达式；

强转不能改变原有的数据的类型的格式和类型，只是改变了其的表达，若要赋值

可以赋值出去

如果不赋值的话，下一个语句，其任然保持原有的类型不进行变化

比如

```
int a;
```

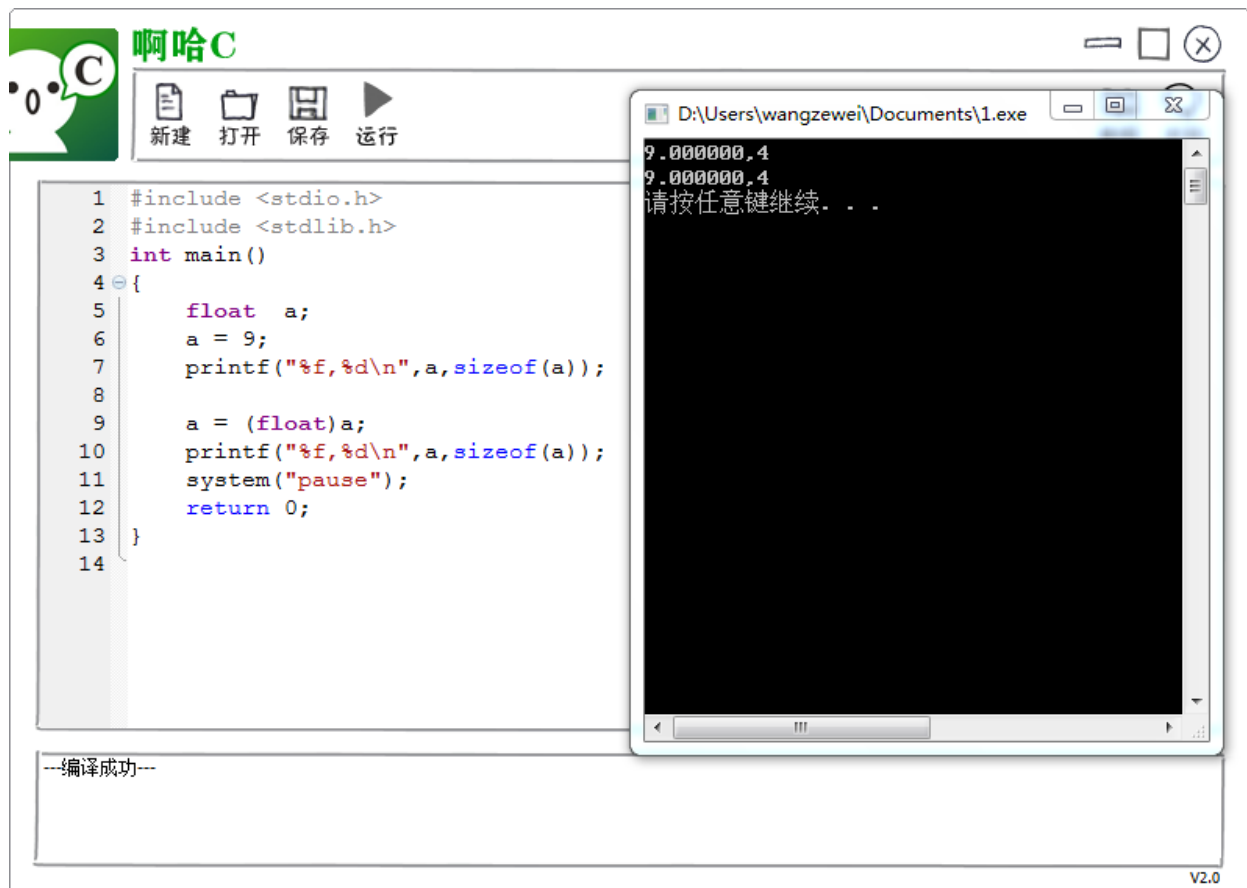
```
a = 9;
```

```
a = (float) a;
```



```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    float a;
    a = 9;
    printf("%f,%d\n",a,sizeof(a));

    a = (int)a;
    printf("%f,%d\n",a,sizeof(a));
    system("pause");
    return 0;
}
```



常用的标准数学函数库

#include<math.h>

第一点：数学中成立的格式和式子放在计算机中不一定成立

函数的名称

sqrt平方根

fabs绝对值

log求解lnx

log10求解lgx

exp计算e的x方

pow (x, y) x的y次方

sin求解sinx的值

cos求解cosx的值

拓展知识：#include< >和#include的区别

< >引用的是编译器的类库路径里面的头文件

” ”引用的是你程序目录的相对路径中的头文件

””的话编译器环境首先会在你当前所在的目录来找寻头文件，当所在目录没有的话，它就会到标准头文件库来找，找不到就报错；<>的话它一开始就到标准头文件库来找，找不到就报错！

假如你编译器定义的自带头文件引用在C:\Keil\c51\INC\下面

则#include<[stdio.h](#)>引用的就是C:\Keil\c51\INC\[stdio.h](#)这个头文件

不管你的项目在什么目录里，C:\Keil\c51\INC\[stdio.h](#)这个路径就定下来了

一般是引用自带的一些头文件：stdio.h、[conio.h](#)、[string.h](#)、[stdlib.h](#)等等之类的。。

假如你的项目目录是在D:\Projects\tmp\

则#include "my.h" 引用的就是D:\Projects\tmp\my.h这个头文件，其一般是用
来引用自己写的一些头文件

如果使用” ”，它是会先在你项目的[当前目录](#)查找是否有对应头文件

如果没有，它还是会在[对应编译器定义的自带头文件目录下](#)查找对应的头文件

意思就是，使用#include "stdio.h"如果你项目目录里面，没有stdio.h这个头文件，它还是会定位到C:\Keil\c51\INC\stdio.h这个头文件的

相对路径和绝对路径

绝对路径是指目录下的绝对位置，直接到达目标位置，**通常是从盘符开始的路径。完整的描述文件位置的路径就是绝对路径，**

绝对路径需要输入完整的文件路径，比如D:\images\郊游照片\2016年郊游照片
\photo1.jpg

相对路径就是指由这个文件所在的路径引起的跟其它文件（或文件夹）的路径关系。使用相对路径可以为我们带来非常多的便利。方便进行到达。

相对路径的结构一般是：在同一个目录：photo1.jpg 或./photo1.jpg； 在下一个目录：xxx/photo1.jpg； 在下下个目录：xxx/xxx/photo1.jpg； 在上一个目录：../photo1.jpg； 在上上个目录：../../photo1.jpg也就是以现在的路径为基础下进行到达其他路径的方式，称为相对路径。

补充下：在html中，[绝对路径](#)还可以是一个完整的网址。

上面的xxx指的是下级目录所对应的文件名，并不可以直接使用哦，还有就是[相对路径](#)引用的文件必须在同一个磁盘中，如果觉得解决你的疑问的话请采纳啦啦啦啦啦

csdn上对于jsp路径的解释

https://blog.csdn.net/qg_34970891/article/details/77102781

