# 为什么要进行数据转换?

参与运算的两个操作数的数据类型必须相同!

# 类型转换的类别

1.[隐式类型转换](#_隐式类型转换)

1. [算数转换](#_算数转换)
2. [赋值转换](#_赋值转换)
3. [输出转换](#_输出转换(主要是在C语言))

2.[强制类型转换](#_强制类型转换)

# 隐式类型转换

## 算数转换

(+ - \* / %)

char, int, long, long long, float, double

当两种不同类型的数据进行算数运算时,转换为此序列的靠后类型

15+3.14=>15.0+3.14

## 赋值转换

int x;

x=3.14\*10.0;

cout<<x; //x=31;

以赋值运算符左侧数据类型为准

## 输出转换(主要是在C语言)

printf(“%c\n”,255+50); //305->49(‘1’)

这里的305需要两个字节,但是char类型仅1个字节,故舍去高位(305:1 0011 0001=> 0011 0001)

printf(“%d\n”,255+50); //305

注意:

int 按照%f格式输出时会得到错误信息

float（或 double) 类型数据，按照%d 格式输出时，将得到错误的输出

# 强制类型转换(此处尚未完成,注意)

## 简单强制类型转换

int x=257+100;

cout<<x<<endl; //357

x=(char)257+100;

//257: 1 0000 0001=>0000 0001 =1

(char 类型仅仅一个字节,则舍去高位)

x=1+100;

cout<<x<<end; //101

## 高级强制类型转换(此处尚未完成,注意)

### static\_cast

### dynamic\_cast

### const\_cast

### reinteroert\_cast