

# 一、介绍

## 1、简介

变量相当于内存中一个数据存储空间的表示，可以比喻成一个房间的门，可以根据门牌号找到房间，也可以通过变量名访问到变量值。

## 2、基础用法

```
// 1、声明变量
int a;
// 2、赋值
a = 100;
// 3、使用变量
System.out.println(a);
```

## 3、注意事项

- 1、变量表示内存中的一个存储区域（不同的变量，类型不同，占用空间也不同，如：`int` 4字节，`double` 8字节）
- 2、该区域有自己的名称[变量名]和类型[数据类型]
- 3、变量必须先声明，后使用
- 4、该区域的数据/值可以在同一类型范围内不断变化
- 5、变量 = 变量名 + 值 + 数据类型

# 二、+号

- 1、当左右两边都是数值型时，则做加法运算
- 2、当左右两边有一方为字符串，则做拼接运算

# 三、数据类型

## 1、介绍

每一种数据都定义了明确的数据类型，在内存中分配了不同大小的内存空间（字节）。

## 2、基本数据类型

```
# 数值型
// 整数型
    byte[1]
    short[2]
    int[4]
    long[8]
// 浮点型
    float[4]
    double[8]
```

```
# 字符型
    char[2]
```

```
# 布尔型
    boolean[1]
```

### 3、引用数据类型

```
# 类
    class
```

```
# 接口
    interface
```

```
# 数组
    []
```

### 4、整型

```
# 分类
byte(字节)      1字节      -128 ~ 127
short(短整型)   2字节      -32768 ~ 32767
int(整型)        4字节      -2147483648 ~ 2147483647
long(长整型)     8字节      -2(63次方) ~ 2(63次方)-1
```

```
# 细节
1、Java各整数据类型有固定的范围和字段长度，不受具体OS的影响，以保证Java程序的可移植性
2、Java的整型常量默认为int型，声明long型常量须后加"L"或"L"
3、Java程序中变量常声明为int型，除非不足以表示大数，才使用long
4、bit：计算中的最小存储单位。byte：计算机中基本存储单位，1byte=8bit
```

### 5、浮点型

```
# 分类
float(单精度)    4字节      -3.403E38 ~ 3.403E38
double(双精度)   8字节      -1.798E308 ~ 1.798E308
```

### # 细节

- 1、与整数类型类似，Java浮点类型也有固定的范围和字段长度，不受具体OS的影响
- 2、Java的浮点型常量默认为double型，声明float型常量，须在后面加'f'或'F'
- 3、浮点型常量表达方式：  
十进制：5.12      512.0f    .512  
科学计数法：5.12e2      5.12e-2
- 4、通常情况下，应该使用double型，因为它比float型更加精确
- 5、浮点数使用陷阱：2.7和8.1/3比较

## 6、字符型

### # 介绍

字符类型可以表示单个字符，字符类型是char，char是两个字节（可以存放汉字），多个字符使用String

### # 细节

- 1、字符常量是用单引号''括起来的单个字符
- 2、Java中允许使用'\''将其后的字符转变为特殊字符型常量
- 3、Java中，char的本质是一个证书，在输出时，是unicode码对应的字符
- 4、可以直接给char赋一个整数，输出时，会按照对应的unicode字符输出
- 5、char类型是可以进行运算的，相当于一个整数，因为它都对应有Unicode码

### # 本质

- 1、字符型存储到计算机中，需要将字符对应的码值找出来
  - 存储：'a' => 码值97 => 二进制(1100001) => 存储
  - 读取：二进制(1100001) => 码值97 => 'a' => 显示
- 2、字符和码值的对应关系是通过字符编码表决定的
  - ASCII
  - Unicode
  - utf-8
  - gbk
  - gb2312
  - big5

## 四、编码

### 1、ASCII码

#### # 简介

上个世纪60年代，美国制定了一套字符编码（使用一个字节），对英语字符与二进制位之间的关系，做了统一规定。这被称为ASCII码。ASCII码一共规定了128个字符的编码，只占用了一个字节的后面7位，最前面的1位统一规定为0。一个字节可以表示为256个字符，ASCII码只用了128个字符

#### # 缺点

不能表示所有字符

## 2、Unicode编码

### # 简介

Unicode编码将世界上所有的符号都纳入其中，每一个符号都给予一个独一无二的编码，使用Unicode没有乱码的问题。Unicode码兼容ASCII码

### # 缺点

一个英文字母和一个汉字都占用2个字符，这对于存储空间来说是浪费

## 3、UTF-8编码

### # 简介

UTF-8是在互联网上使用最广的一种Unicode的实现方式。UTF-8是一种变长的编码方式，它可以使用1-6个字节表示一个符号，根据不同的符号而变化字节长度。字母占1个字节，汉字占3个字节

## 五、数据类型转换

### 1、自动类型转换

#### # 简介

自动类型转换，当Java程序在进行赋值或者运算时，精度小的类型自动转换为精度大的数据类型

#### # 精度排序

```
char -> int -> long -> float -> double  
byte -> short -> int -> long -> float -> double
```

### 2、自动类型转换细节

- 1、有多种类型的数据混合运算时，系统首先自动将所有数据转换成容量最大的那种数据类型，然后再进行计算
- 2、当我们把精度大的数据类型赋值给精度小的数据类型时，就会报错，反之就会进行自动类型转换
- 3、(byte, short)和char之间不会相互自动转换
- 4、byte, short, char 他们三者可以计算，在计算时首先转换为int类型
- 5、boolean 不参与转换
- 6、自动提升原则：表达式结果的类型自动提升为操作数中最大的类型

### 3、强制类型转换

#### # 简介

自动类型转换的逆过程，将容量大的数据类型转换为容量小的数据类型。使用时要加上强制转换符()，但可能造成精度降低或溢出，格外要小心

## 4、强制类型转换细节

- 1、当进行数据的大小 从大 -> 小，就需要使用到强制转换
- 2、强制符号只针对于最近的操作数有效，往往会使用小括号提升优先级
- 3、`char`类型可以保存`int`的常量值，但不能保存`int`的变量值，需要强转
- 4、`byte`和`short`类型在进行运算时，当作`int`类型处理

## 5、基本数据类型与String

# 基本数据类型转String类型  
将基本类型的值 + ""即可

# String类型转基本数据类型  
通过基本数据类型的包装类调用`parseXX`方法即可

## 6、基本数据类型与String细节

在将String类型转成基本数据类型时，要确保String类型能够转成有效的数据，比如：我们可以把"123"转成一个整数，但不能把"Hello"转成一个整数