# 数据类型：

## 基础类型（主类型）：

### java基础类型，又叫做主数据类型，主要有八种：

### byte类型：

1. byte类型是1个字节（8位二进制）、有符号的，以二进制补码表示的数。
2. 取值范围： -128（-2^7）~127（2^7）
3. 默认值是0
4. byte类型用在大型数组中节约空间，主要代替整数，因为byte类型变量占用的栈空间只有int类型的四分之一。
5. 正确范例： byte b = -56; byte b = 0; byte b = 80;
6. 错误范例： byte b = -129; byte b = 128;（都是错误的，超出了取值范围）

### short类型：

1. short类型是2个字节（16位二进制）、有符号的，以二进制补码表示的整数。
2. 取值范围： -32768（-2^15）~32767（2^15）
3. 默认值是0
4. short类型也可以像byte类型那样节省空间，一个short类型变量所占的栈空间是int型变量的二分之一。
5. 正确范例： short s = -1020; short s = 0; short s = 100;
6. 错误范例： short s = -32679; short s = 32768; （都超出了范围）

### int类型：

1. int类型是4个字节（32位二进制），有符号的，以二进制补码表示的整数。
2. 取值范围： -2,147,483,648（-2^31）~2,147,483,647（2^31 - 1）
3. 默认值是0
4. 一般的整型变量默认为int类型

### long：

1. long类型是8个字节（64位二进制）、有符号的，以二进制补码的形式表示的整数。
2. 取值范围： -2^63~2^63-1
3. 默认值是0L
4. 这种类型主要使用在需求比较大整数的系统上。

### float：

1. float数据类型是单精度、32位、符合IEEE 754标准的浮点数
2. float在存储大型浮点数族的时候可节省内存空间
3. 取值范围：-3.403E38~3.403E38（比double值还要大）
4. 默认值是0.0f
5. 浮点数不能用来表示精确的值，如货币

### double：

1. double数据类型是双精度、64位、符合IEEE 754标准的浮点数。
2. 浮点数的默认值是double类型，即如果浮点数后面什么不加”f”声明，则默认是double类型
3. double类型同样不能表示精确的值，如：货币
4. 取值范围：-1.798E308~1.798E308
5. 默认值是0.0d

### 8、boolean：

1）boolean数据类型表示一位二进制数，只有true和false两个值

2）默认值是false

### 9、char：

1）char类型占两个字节，是一个单一的16位Unicode字符。

2）char类型是无符号位的，即都是正数

3）取值范围： \u0000（0）~\uffff（65535）

4）char数据类型可以存储任何字符

## 引用类型：

### 数组：

### 接口：

### 3、类：

## 类型转换：

### 自动转换（隐式转换）：

类型转换分为自动转换(隐式转换)和强制转换(显式转换)

自动转换:分两种情况：

1. 常量赋值给变量：(常量默认是int或者double类型,这里以int为例)

1)常量值赋值给表值范围比int大的变量：

-进行了自动类型转换,将常量的类型转换成了表值范围大的类型(向上转换总是自动,并且安全的)

2)常量值赋值给表值范围比int小的变量：

①常量值没有超出变量类型的取值范围：

-进行了自动类型转换(向下转型),将常量的类型转换成了变量的类型

②常量值超出了变量类型的取值范围：

-编译报错,若想不报错,就要使用强制转换,但是会丢失精度

1. 变量赋值给变量：

1)变量赋值给比它自己的类型的表值范围大的变量：

-进行了自动类型转换,将变量的类型向上提升了(向上转型总是自动,并且安全的)

2)变量赋值给比它自己的类型的表值范围小的变量：(这种情况下都是要强制转换的,而且都会丢失精度,区别在于变量的值是否发生了变化)

①变量的值没有超出要赋给的变量的表值范围：

-需要进行强制类型转换,否则会报错,变量的值不变

②变量的值超出了要赋给的变量的表值范围：

-需要进行强制类型转换,否则会报错,变量的值变了

## 四、类型转换的原理：

以 byte = 200为例：

1、10的二进制表示：（10是int类型，int类型是32位）

|  |
| --- |
| 0000-0000 0000-0000 0000-0000 1100-1000 |

2、byte是8位二进制，所以需要将前面24个二进制全部去掉，二进制表示就成了这样

|  |
| --- |
| 1100-1000 |

3、那 1100-1000 这个二进制表示的数十进制又是多少呢？

|  |
| --- |
| 1100-1000 |
| 取反 0011-0111 |
| 加1 0011-1000 |
| 十进制 32+16+8=56 |
| 所以 1100-1000 就是 -56 |

1. 引用类型：