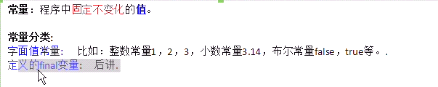
Day 02

**01 字面量和常量**

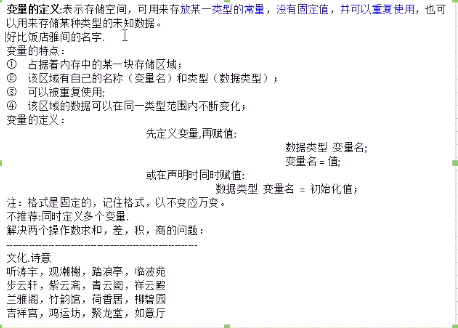


常量分为字面常量和自定义常量。

字面量就是直接给出的一个值（可以是整数小数true false 字符串等）

**02 引出变量**

**003 变量的定义和基本使用**

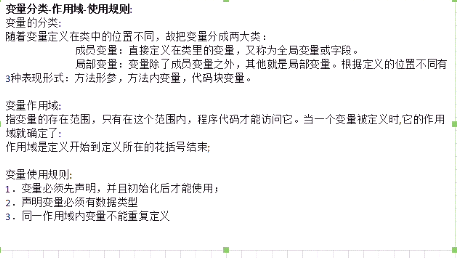


寄存器 内存堆栈 内存堆 静态区 常量区 持久化存储

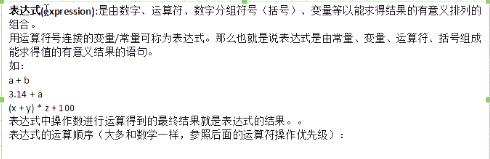
局部变量在使用前需要初始化，类变量自动初始化。

Java内存分为三大区：堆栈 堆内存和常量方法区

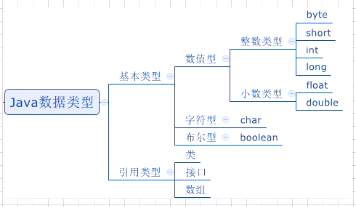
**04 变量的分类、作用域和使用规则**



**05 java表达式**



**06 数据类型和分类**



基本数据类型/原生数据类型

数值型：

整数型：byte short int long

小数型：float double

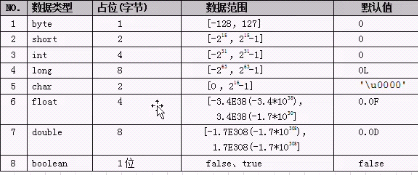
不同整数类型在内存中占用的空间大小不同。

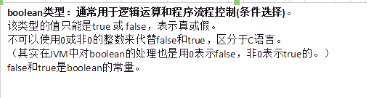
字符型:char

布尔型:Boolean,true和false

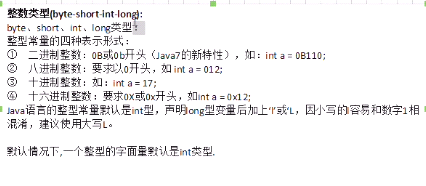
引用数据类型/对象数据类型

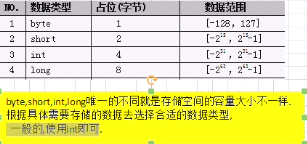
007 布尔类型



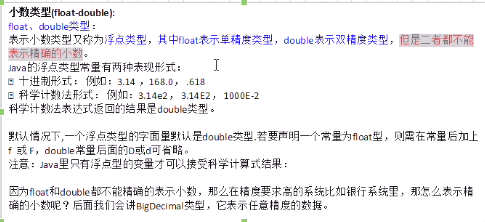


**08 整数类型**

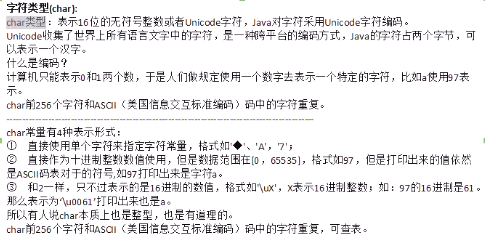




09 小数类型



**10 字符类型**



Char是无符号的。可以赋值为整数，范围为0~65535（0~2^16-1）

从某方面来讲，char就是整数类型，0~2^16-1，最高位不是符号位

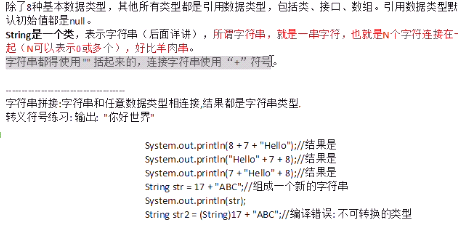
Boolean在jvm中也是由0和1来表示的

char c1=65;

System.out.println(c1); //输出A

**11 最常用的引用类型**

String



Java字符串源码实现就是字符数组

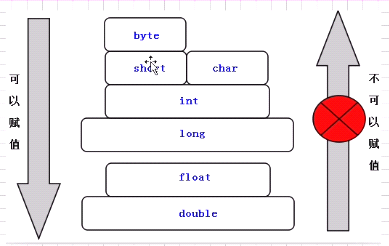
**12 数据过大和溢出**

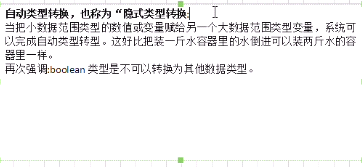


因为数组访问时每次都要检查是否出现索引越界所以相对c，java数组访问较慢，但增加了安全性

数字运算在计算机底层都是二进制运算，即转换为二进制运算

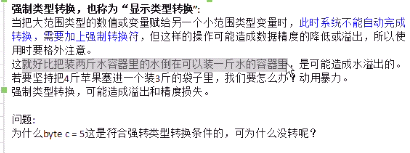
**13 基本类型的自动转换**





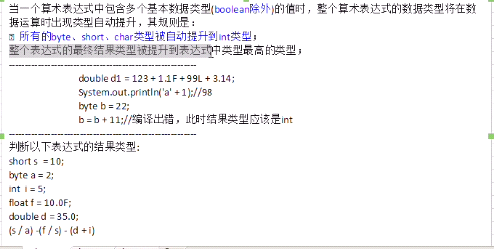


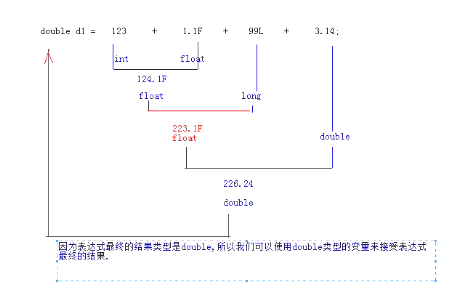
**14 基本数据类型的强制转换**



因为所有的字面量数字都是int类型，5在byte的范围之内，因此可以强制转换成功

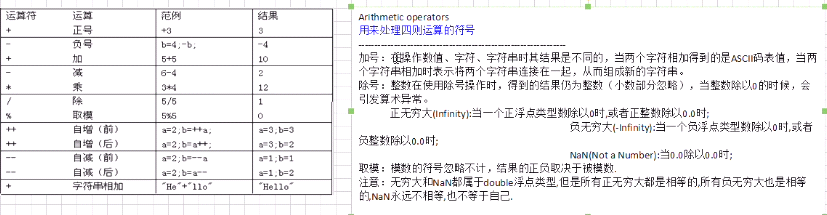
**15 表达式类型的自动提升**



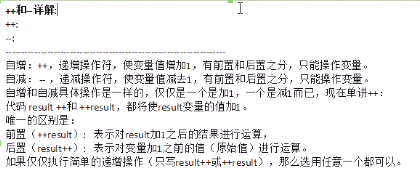


自动提升永远是小转大（占用空间）

**16 算数运算符**

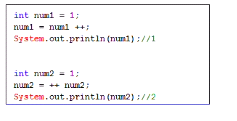


**17 ++和--的详解**

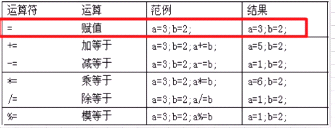


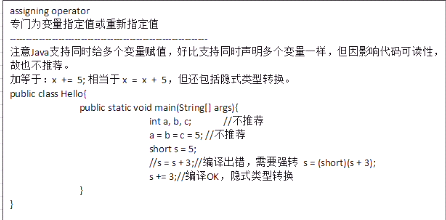
++ -- 都只能操作变量，不能操作常量（字面常量和定义的常量）

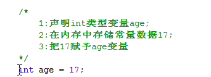
018 自增的面试题



019 赋值运算符

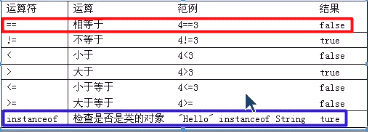




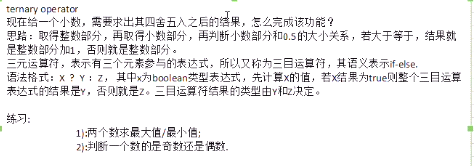


+= -= \*= /= 包含隐式类型转换

020 比较运算符



021 三元运算符

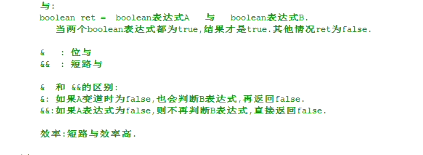


022 分页业务逻辑算法

边界条件判断

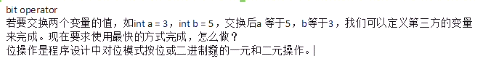
023 逻辑运算符

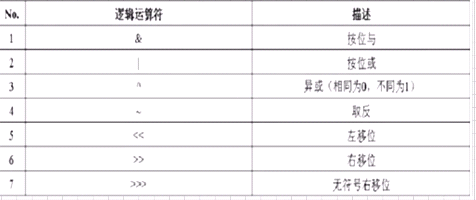


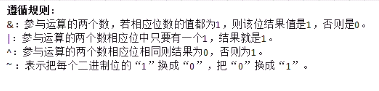


短路运算符效率更高

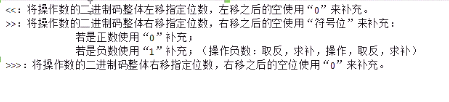
024 为运算符





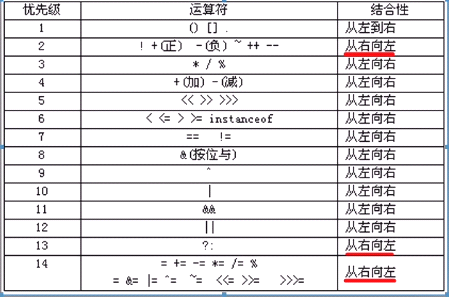


026 移位操作



027 交换两个变量的值

028 运算符操作的优先级



028 小结