# Java变量命名规范

## Package(包) 的命名

Package 的名字应该都是由一个小写单词组成, 例如 net.ebseries.modules。一般都是反着写，比如com.baidu.controller

## Class(类) 或 Interface(接口)的命名

Class 的名字首字母大写, 通常由多个单词合成一个类名，要求每个单词的首字母也要大写, 例如：DataFile 或 InfoParser。叫做“驼峰命名法”。

## 变量的命名

1. **首字母是英文字母、$ 和下划线，由字母、数字和下划线组成。**
2. 变量的命名遵循见名知义的原则。 ［很重要，比如名字就用 name ，而不是用 a、b、c 这样的命名，不然又要找找找］
3. 用驼峰命名法命名多个单词组成的变量名。 比如： sumScore ］
4. 变量名［方法名］首字母建议不用大写字母。 ［首字母大写一般是用来标识类名的］ －－ 看起来更规范而已
5. 变量名不要使用 Java 关键字。

## Static Final 变量（相当于常量）的命名

Static Final 变量的名字应该都大写，并且指出完整含义，例如：final MAXUPLOADFILESIZE=1024。

## 方法的命名

1. 方法名的第一个单词应该是动词，大小写可混用，**但首字母应小写**。
2. 在每个方法名内，大写字母将词分隔并**限制使用下划线**。
3. 参数的名字必须和变量的命名规范一致。

## 数组的命名

1. 数组应该总是用下面的方式来命名：byte[] buffer; 而不是：byte buffer[](习惯性问题而已)。
2. 一般应采用小写字母，但类名、接口名以及任何非初始单词的第一个字母要大写。

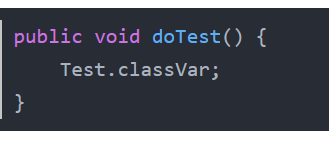
# Java关键字



# 变量

## 类变量

1. 类变量在类中**用 static 关键字声明**，**在任意在方法、构造函数或代码块外部。**
2. 类变量在程序**启动时被创建**，在程序停止时被销毁。**类变量被存储在静态存储器中**。
3. 类变量通过调用的类名来访问。



1. 类变量与对象毫无关系，无论有没有创建对象，创建了多少个对象，一个类变量永远只有一个副本。

## 常量（不可修改）

1. 常量是标记为 final 关键字的类变量，类变量的特性常量均有。
2. 声明常量时必须赋值，且赋值后不能改变其值。
3. 常量名称一般都需要大写。
4. **常量被存储在常量存储器中。（常量存储器和静态存储器不一样）**

## 实例变量

1. 实例变量在类中声明，在任意在方法、构造函数或代码块外部。
2. 实例变量在一个对象 "new" 时创建，在对象被销毁时销毁。
3. **实例变量是对象的一部分，同对象一样存储在堆中。**
4. 实例变量通过对象来访问。

## 局部变量

1. 局部变量在方法、构造函数或代码块中声明。
2. 局部变量在进入方法、构造函数或块时创建，退出时被销毁。
3. 局部变量在堆栈级别内部实现。

# 常用运算符

## 算术运算符



## 关系运算符



## 逻辑运算符



1. & 称为 逻辑与，需要判断完两个条件。&& 称为 短路与，如果第一个为false直接得结果。逻辑或，短路或同理。

## 位运算符

位运算符可以应用到**整数类型，长型，整型，短整型，字符和字节**。它作用于位，并执行逐位操作。



## 赋值运算符



1. 异或操作：相同为0，不同为1。

1^1=0 0^0=0 1^0=1 0^1=1

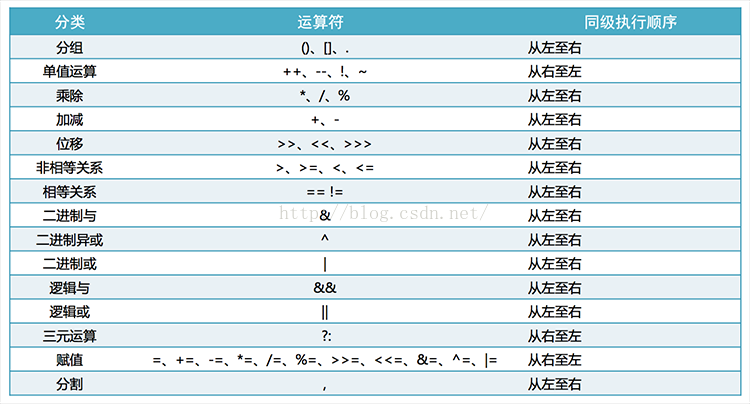
## 其它运算符\_条件运算符

条件运算符也被称为三元运算符，可以作为赋值运算符种很特殊的一种，此运算符是确定哪些值应分配给变量。语法：

variable x = (expression) ? value if true : value if false

"?" 号左侧为条件表达式 ture 或 false，如果 true 则将 ":" 左侧值赋值给 "=" 左侧的变量；如果 false 则将 ":" 右侧值赋值给 "=" 左侧的变量。

## 运算符优先级别



# Math类常用函数

## 常量

自然基数：public static final double E

圆周率：public static final double PI

## 四舍五入

Math.ceil() 执行向上舍入，将数值向上舍入为最接近的整数；

Math.floor() 执行向下舍入，将数值向下舍入为最接近的整数；

Math.round() 执行标准舍入，将数值四舍五入为最接近的整数

## 其他

Math.abs 求绝对值

Math.max 求两数中最大

Math.min 求两数中最小

Math.sqrt 求开方

Math.pow 求某数的任意次方, 抛出 ArithmeticException 处理溢出异常

Math.random 返回 0，1 之间的一个随机数