# 变量：

变量将对象状态存储到field中。Field在对象中介绍了。命名一个field的规则是什么，还有什么类型。如果field没有初始化，是否分配一个默认值。我们在后面探讨这些问题。在Java中，field和变量都被使用，有时候会混淆。

Java定义下面类型的变量:

实例变量（非静态域）：每个对象自己的变量

类变量（静态域）：全局变量（java中没有global关键字）

局部变量：只在method中

参数：

在后文中，

一般field是指实例变量和类变量，不包含局部变量和参数。

变量是指上面四个。

成员：类定义中的filed，嵌套类型，方法，都是成员。

命名规则：

变量名字：

子字符串可以是字母，数字，美元符和下划线。

如果名字只有一个单词，那么小写则个单词。如果有多个，那么后面的单词首字母大写。如果存储常数值，那么每个单词的字母大写，单词之间用下划线隔开。

## **Primitive Data Types**

### 数据大小以及类型

Pass

### 类型默认值

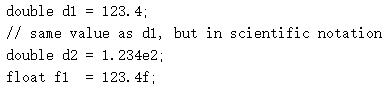
Pass

Literals（数值）

在初始化内置类型时，没有使用new关键字，内置类型。Literal是固定值。

### 整数literal

### 浮点literal



### 字符和String literal

Char和string的literal可能包含Unicode字符，如果你的editor和文件系统允许，你可以在你的代码中使用这些字符。如果不使用，你可以使用Unicode escape，

Char要使用单引号

String要使用双引号

Java支持一些特殊的char和string literal:

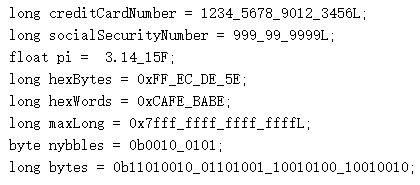


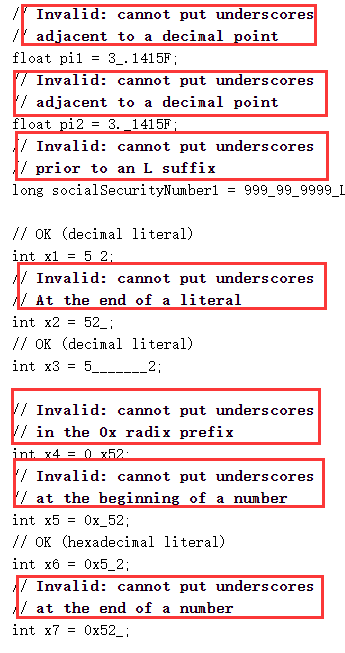
也有一个特殊的literal可以用作任何引用类型。Null可以分配到任意类型。

String.class指向类名字。

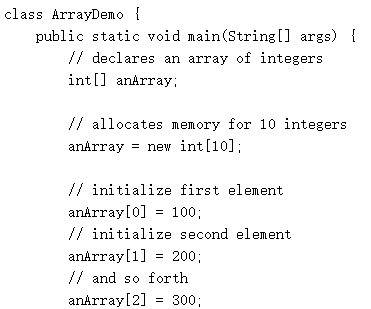
### 在数值Literal中使用下划线

下划线可以在一个数值literal中任意部分出现，这些特征能够让分出在数值literal中的组，从而**增加可读性**



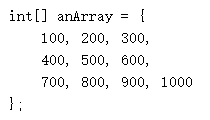


## 数组：

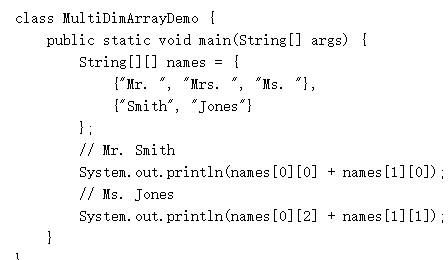


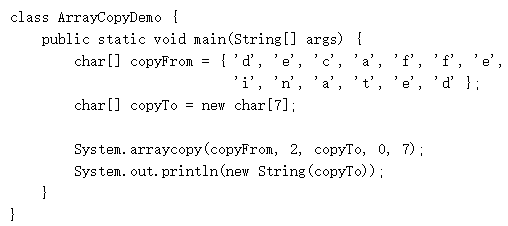
声明数组变量。任意类型有两个部分组成。数组类型和名字。如果是一个数组类型，那么在类型后面加上[]，也可以像C一样放在后面变量名字后面。





多维数组用{}分开。



数组复制  


### 数组操作：

Java提供了数组操作最基本的方法。