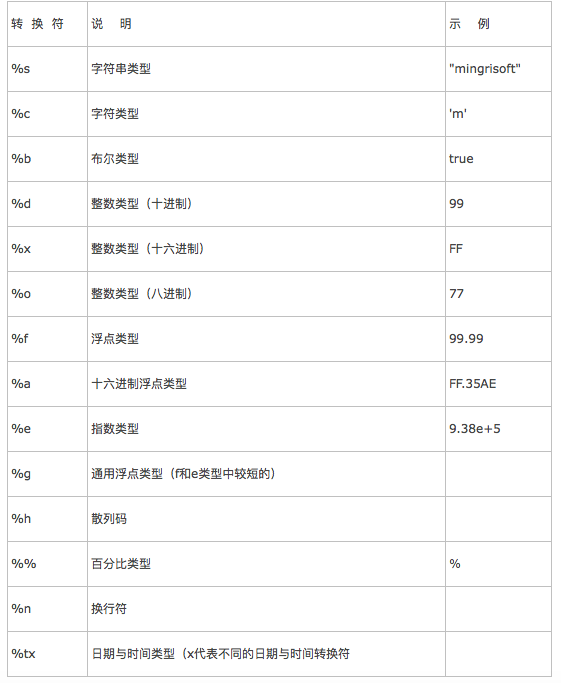
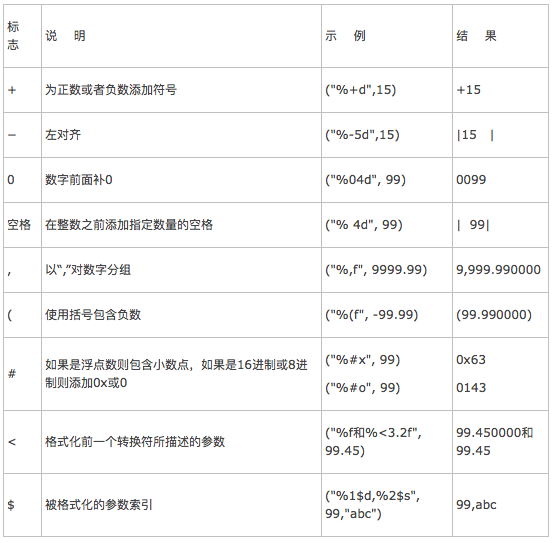
# 基本编程结构

## 格式化输出





out.**printf**("%,+.2f\n", 10000.0/3.0);

String message = String.**format**("The averag number is: %,+.2f, total number: %d ", 10000.0/3, 10000);

out.**println**(message);

%d、%x、%o分别是10、16和8进制，没有2进制。

## 数组的复制

int[] array = new int[]{1,2,3};

int[] arrayCopy = array;

arrayCopy[2] = 30;

for(int i : array)

out.**print**(i);

int[] arrayDeepCopy = Arrays.**copyOf**(array, array.length);

arrayDeepCopy[2] = 300;

for(int i : array)

out.**print**(i);

## 实例变量和局部变量

实例变量是在class中定义的变量，属于实例对象。**实例变量**可以不初始化，系统会自动给它复赋值，比如数字为0，boolean为假，对象引用为null。而局部变量必须要初始化之后才可以使用。

# 对象比较

## Comparable

每个类比较元素的规则应该是不同的，排序算法只应该调用类提供的比较方法，只要所有的类就调用什么方法达成一致，排序算法就能开始工作。

如果一个类想启用对象排序，那么它应该实现Comparable接口。

Comparable接口有一个类型参数

public interface **Comparable**<T> {

public int **compareTo**(T o);

}

在调用x.compareTo(y)的时候，返回值为负整数，说明x在y的前面；返回值为0，说明x和y是相等的；返回值为正整数，说明x在y的后面。

public class **Employee** implements **Comparable**<Employee> {

int id;

public int **compareTo**(Employee o) {

return this.id - o.id;

}

public static void **main**(String[] args){

Employee e1 = new **Employee**(); e1.id = 2;

Employee e2 = new **Employee**(); e2.id = 1;

Employee[] es = new Employee[]{e1, e2};

Arrays.**sort**(es);

//out.**println**(es[0].id);

}

}

Arrays.sort方法可以对任何Comparable对象的数组进行排序。

## Comparator

对于String对象，我们不想按照字典顺序排序，但是我们也没有办法修改String的compareTo方法。Arrays.sort方法提供了一个办法，它接受一个实现了Comparator接口类的实例。