# 函数

1. 概念

一段逻辑代码可以完成特定的功能，可以反复使用

1. 语法

位置：类的内部，其他函数的外部

函数的声明和实现：

修饰符 返回值类型 函数名(参数表){

//完成特定的功能的逻辑代码

}

Eg:public static void 函数名(){

//逻辑代码

}

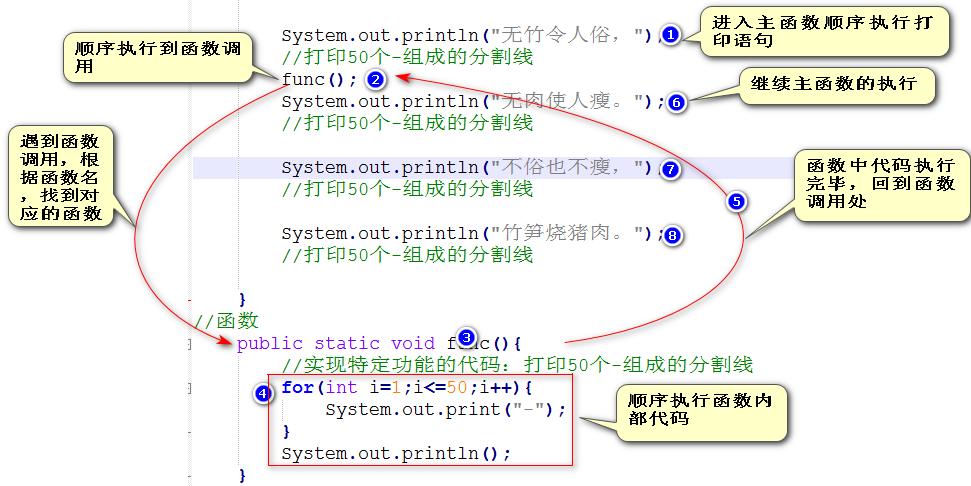
函数的调用：

在需要函数的地方：

函数名(参数表);

Eg:函数名();

函数的执行流程：



1. 参数

多数情况下，函数和调用者之间都存在数据交互，只有调用者提供必要的数据，函数才能提供相应的功能，调用者提供的数据叫做参数。

声明处：修饰符 返回值类型 函数名（形参1,形参2...）

形参列表：声明局部变量 int a,double b告知调用者需要什么类型的参数

形参可以是所有的数据类型：8中基本类型+引用类型

形参可以有无数个

调用处：方法名(实参1,实参2...);

实参列表：给局部变量赋值 12,12.1依次给形参列表赋值

实参赋值必须类型与形参一致

实参个数和形参个数一致

注：参数中间用 , 分割

1. 返回值

函数声明处：修饰符 返回值类型 函数名（参数表）

返回值类型：所有的数据类型+void

Void:当函数没有返回值，需要void占位，说明函数没有返回值

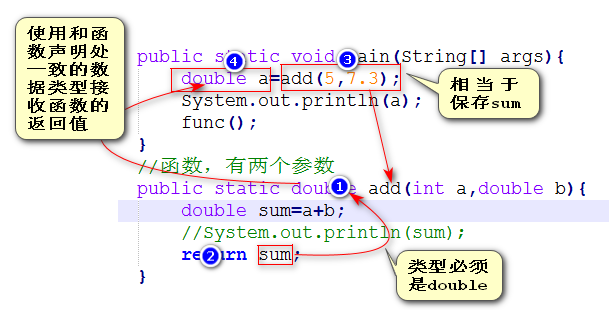
有函数的返回值：

修饰符 返回值类型（int） 函数名（参数表）{

Return 返回值；//返回值类型必须和函数声明处的返回值类型相同

}

当函数有返回值，函数调用相当于表达式



函数调用：

数据类型 变量名=函数名（参数）；

数据类型必须保持和函数声明处的返回值类型相同

作用：

约束函数内部返回值的结果

告知函数调用者接收返回值的类型

Return：

用在有返回值的函数中：

返回函数的结果，并且结束函数

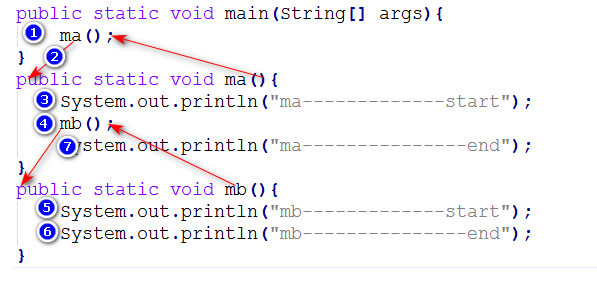
用在没有返回值的函数中：

结束函数

如果函数有返回值类型，并且函数中出现分支语句，必须保证每条分支语句都有返回值

1. 函数嵌套和递归

函数嵌套：在函数中调用一个函数





递归：在当前函数中调用自己本身，必须存在收敛条件