**函数是一块代码，接收零个或多个参数做一件事，并返回零个或一个值。**

**Java的函数必须定义在类的内部，成为类的成员。**

**定义一个函数：**

**<返回类型> <函数名称>(<参数表>) {**

**<函数体>**

**}**

**返回类型是这个函数运行结束时要返回给调用者的数据的类型，函数可以返回。基本数据类型、对象或者void。返回void表示这个函数不返回任何值。**

**函数名称是一个Java名字，一样要遵循和变量一样的命名规则。**

**参数表是0个或1个或多个参数定义，用逗号’,’分隔。**

**在这个阶段，我们要在所有的函数的返回类型前面加上关键字“static”。**

**static表示这个函数属于这个类，而不属于这个类的任何对象，因此我们才可以不制造这个类的对象，而直接从main()函数中调用它。**

**变量的生存期：什么时候变量开始出现，到什么时候变量消亡。**

**变量的作用域：在代码的什么范围内可以访问这个变量（变量可以起作用）。**

**本地变量：函数每次运行，就产生一个独立的变量空间，在这个空间中的变量，是函数这次运行所独有的。（定义在函数内部的变量就是本地变量，参数也是本地变量）**

**本地变量的规则：**

**本地变量定义在块内**

**可定义在函数内**

**可定义在语句内**

**可随意用一对大括号定义变量**

**程序运行进入块之前，块中的变量不存在；离开块时，变量消失**

**块外面定义的变量在里面仍有效**

**不能在一个块内定义同名的变量，不能定义块外面定义过的变量**

**本地变量不会默认初始化**

**参数进入函数时被初始化了**

### 例一：

**函数定义\_素数求和**

**代码**

**package** Function;

**import** java.util.Scanner;

**public** **class** 函数定义\_素数求和

{

**private** **static** Scanner *in*;

//判断素数

**public** **static** **boolean** isPrime(**int** i)

{

**boolean** pd = **true**;

**for** (**int** k=2; k<i; k++)

{

**if**( i % k == 0 )

{

pd =**false**;

**break**;

}

}

**return** pd;

}

//主函数

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

{

//函数定义

//对象的操作 对象可执行函数

*in* = **new** Scanner(System.***in***);

System.***out***.print("请输入n：");

**int** n = *in*.nextInt();

System.***out***.print("请输入m：");

**int** m = *in*.nextInt();

**int** cnt = 0;

**int** sum = 0;

**if**(n==1) n=2;

**for** (**int** i=n; i<m; i++)

{

//计算

**if**(*isPrime*(i))

{

cnt ++;

sum += i;

}

}

System.***out***.println( n + "-->" + m + "之间有" + cnt + "个素数，总和为：" + sum);

}

}

### 例二：

**函数定义\_求和**

**代码**

**package** Function;

**import** java.util.Scanner;

**public** **class** 函数定义\_求和

{

**private** **static** Scanner *in*;

//求和

**public** **static** **void** Sum(**int** n,**int** m)

{

**int** sum = 0;

**for**(**int** i=n; i<=m; i++)

sum += i;

System.***out***.println( n + "-->" + m + "之间总和为：" + sum);

}

//主函数

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

{

//函数定义

//对象的操作 对象可执行函数

*in* = **new** Scanner(System.***in***);

System.***out***.print("请输入n：");

**int** n = *in*.nextInt();

System.***out***.print("请输入m：");

**int** m = *in*.nextInt();

*Sum*(n,m);

}

}