# Static

在《Java编程思想》P86页有这样一段话：

“static方法就是没有this的方法。在static方法内部不能调用非静态方法，反过来是可以的。而且可以在没有创建任何对象的前提下，仅仅通过类本身来调用static方法。这实际上正是static方法的主要用途。”

## Static可以用在什么地方

### 修饰成员变量

static变量也称作静态变量，静态变量和非静态变量的区别是：静态变量被所有的对象所共享，在内存中只有一个副本，它当且仅当在类初次加载时会被初始化。而非静态变量是对象所拥有的，在创建对象的时候被初始化，存在多个副本，各个对象拥有的副本互不影响。static成员变量的初始化顺序按照定义的顺序进行初始化。

另外记住，即使没有显示地声明为static，类的构造器实际上也是静态方法。

static是不允许用来修饰局部变量

### 修饰成员函数

static方法一般称作静态方法，由于静态方法不依赖于任何对象就可以进行访问，因此对于静态方法来说，是没有this的，因为它不依附于任何对象，既然都没有对象，就谈不上this了。并且由于这个特性，在静态方法中不能访问类的非静态成员变量和非静态成员方法，因为非静态成员方法/变量都是必须依赖具体的对象才能够被调用。

### 类的静态代码块

static关键字还有一个比较关键的作用就是 用来形成静态代码块以优化程序性能。static块可以置于类中的任何地方，类中可以有多个static块。在类初次被加载的时候，会按照static块的顺序来执行每个static块，并且只会执行一次。因此，很多时候会将一些只需要进行一次的初始化操作都放在static代码块中进行。

## 特点

1.被对象共享，在内存中只存在一份，存在于方法区中。

2.除被对象调用外还可以直接被类名调用。——（ class.method() ）

3.随类的加载而加载，随类的消失而消失（类中其生命周期最长）

4.优先于对象存在

## Static利弊

利：对类的共享数据进行单独空间存储，在类的实例较多时能节省空间，不需要每个对象都 生成一份

弊：访问局限(只能访问静态)，生命周期过长。

## 关于java的主函数main是static的

public static void main(String[] args)

JVM识别的入口main格式固定唯一（包括参数类型），但可以重载

JVM调用main时传的args为 new String[0]

被JVM调用作为程序入口

public： 访问权限最大

static ：随类的加载必定存在

void ：主函数没有具体返回值

main ：不是关键字，可以被JVM唯一识别

String[] args：字符串数组

## 用法示例

1. 如果类中所有的方法都为静态方法，可以将类的构造函数初始为私有private，以使类不能创建对象。（能隐藏的都隐藏起来）

## 静态代码块

class p

{

static{} //静态构造代码块

{} //构造代码块

}

p a = null; //类并未加载进内存

new p();//先类加载进内存，再在内存中创建无实体对象引用的对象内存

p a = new p(); //先类加载进内存，再在堆内存中创建有实体对象引用的内存

用到类中内容才会加载类到内存中

## 详解对象的初始化过程 p a = new p();

第一步.JVM找到并加载p.class加载进内存

第二步.若有static代码块，则先执行进行类的初始化

第三步.在堆内开辟空间，分配内存地址

第四步.在堆内存中建立对象的特有属性，并进行默认初始化

第五步.对属性进行显示初始化

第六步.对对象进行构造代码块初始化

第七步.对对象进行对应的构造函数初始化、

第八步.将内存地址赋给栈内存中的a变量