

final修饰变量的初始化方式：

```
package com.wxiang;  
// 非静态final赋值有三种：定义初始化、非静态代码块、构造方法。
```

```
public class FinalTest {  
    // 1、声明时赋值  
    private final String name = "hello";  
  
}
```

```
class FinalTest1{  
    private final int id;  
    // 在静态代码块中赋值  
    {  
        id = 10;  
    }  
}
```

```
class FinalTest2{  
    private final boolean flag;  
    // 3、在构造函数中赋值  
    public FinalTest2(boolean flag){  
        this.flag = flag;  
    }  
}
```

```
// 静态final赋值有两种：定义初始化、静态代码块。
```

```
class StaticFinalTest{  
    private static final String name = "张三";  
}
```

```
class StaticFinalTest1{  
    private static final String sex;  
    static {  
        sex = "female";  
    }  
}
```

被final修饰的变量其实就相当于定义了一个常量，无法被修改的变量，如果final修饰的是一个基本数据类型的变量，那么这个变量的值就定了，不能变了，而如果修饰的是一个引用变量，那么该变量存的是一个内存地址，该地址就不能变了，但是该内存地址所指向的那个对象还是可以变的

单独用final修饰的变量也有可能在字节码找到对应的ConstantValue属性，但是会被JVM忽略掉。

`final`修饰的实例属性，在实例创建的时候才会赋值。

`static`修饰的类属性，在类加载的准备阶段赋初值，初始化阶段赋值。

`static+final`修饰的`String`类型或者基本类型常量，JVM规范建议在初始化阶段赋值，但是HotSpot VM直接在准备阶段就赋值了。

`static+final`修饰的其他引用类型常量，赋值步骤和第二点的流程是一样的。