var关键字声明可修改变量。

val关键字声明只读变量。

const关键字：编译时常量，只能在函数外定义。因为编译时常量都是在编译的时候赋值，而函数都是在运行的时候才调用，函数内的变量也是在函数运行的时候赋值，编译时常量要在这些变量赋值前就已经存在。

注：编译时常量只能是基本数据类型：String ，Int，Double，Float，Long，Short，Byte，Char，Boolean。

类型推断：对于已声明并赋值的变量，允许省略类型定义

例如：

# String模板

模板支持在字符串的引号内放入变量值

还支持字符串里计算表达式的值并插入结果，添加在${}中的任何表达式，该表达式的值都会作为字符串的一部分。

例如下面代码：



输出结果



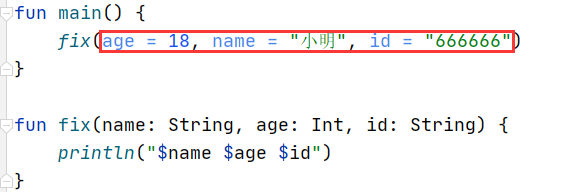
# 函数

函数头：private public等，若函数有函数头修饰，必须注明返回类型

没有返回值的函数叫Unit函数，也就是它们返回Unit类型

默认参数值：kotlin中可以给函数传入默认参数值

命名函数参数：调用函数时，如果使用命名值参，就可以不用值参的顺序

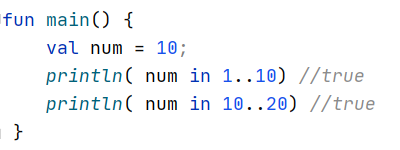


# Rang表达式

关键字in

格式X in A..B

in关键字用来检查某个值X是否在指定的范围之内，



# 反引号中的函数名

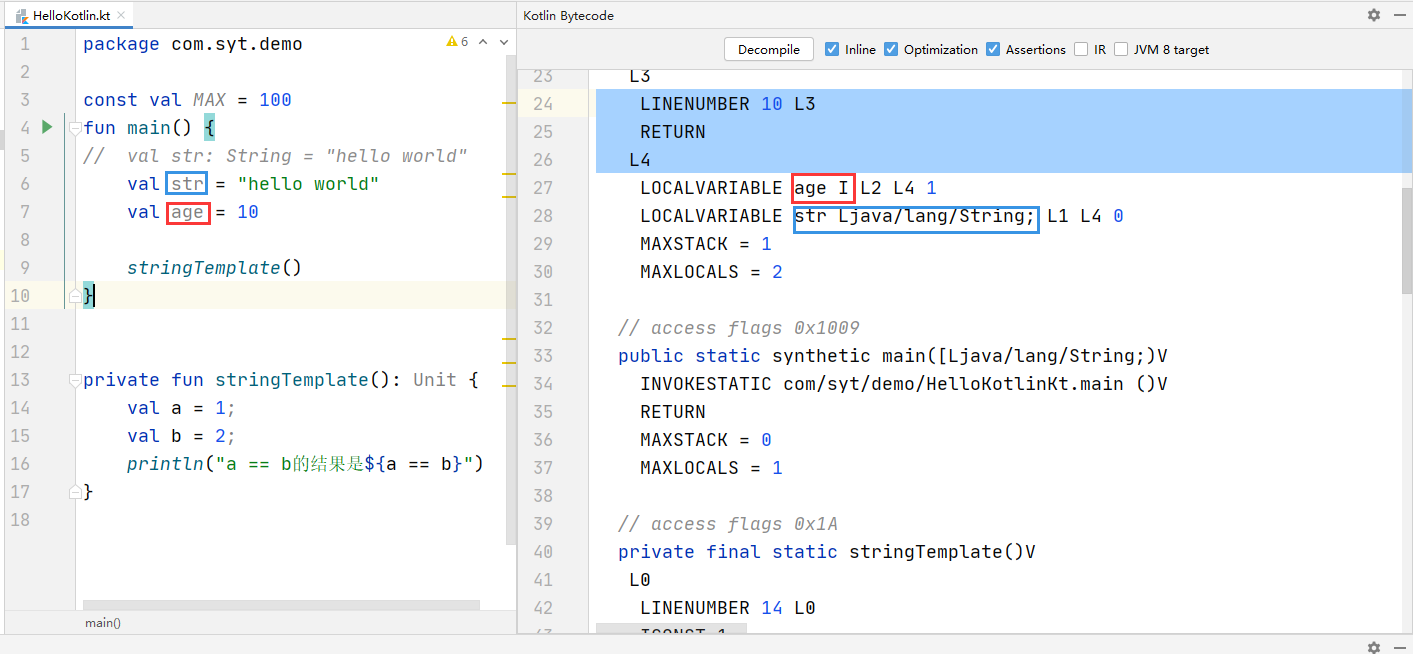
1.Kotlin可以使用空格和特殊字符对函数命名，不过函数名要用一对反应号 `` 括起来。可以用于测试函数的命名。

2.为了支持Kotlin和Java的互操作，而Kotlin和Java各自却有着不同的保留关键字，不能作为函数名，使用反引号包住函数名就能避免冲突



# 关于Kotlin中只有引用数据类型问题

由于刚刚进入Kotlin的学习当中，我起先对这个问题抱有怀疑态度，原因如下图



我定义了一个age变量（很明显它会是Int类型），如果说Kotlin只有应用数据类型的话，那么Kotlin中的Int类型也将是应用数据类型，但是编译后的字节码文件中age的类型是I ，而Java代码通过编译后代码中的int类型变量在字节码文件中的标识也是I ，所以我对该问题产生了疑问。

但是再看看Kotlin中的String类型编译过后是Ljava/lang/String，我的观点就变成了这样：

假设Kotlin中只有引用数据类型，只不过在编译的阶段，Kotlin的编译器会将一些特定的引用类型比如Int，Char等自动转换成Java中的基本数据类型。

# Kotlin协程

# 协程

谷歌在协程的官方介绍中说可以将协程看成是轻量级的线程。我对于这句话的理解是这样的

1. 协程必须依赖于线程运行
2. 协程可以在线程之间切换，并且不会造成当前线程的阻塞

【今日进度】  
1.落地完善 Kotin协程 + Retrofit + MVVM + Jetpcack的网络请求Demo  
【明日计划】  
1.对Kotlin的协程进行更加深入地学习并做好学习总结  
【风险点】  
无