## 首先关于String s = new String("xyz"); 创建几个对象的问题

你知道在java中除了8中基本类型外，**其他的都是类对象以及其引用**。所以 "xyz "在java中它是一个String对象.对于string类对象来说他的对象值是不能修改的，也就是具有不变性。   
看：   
String s= "Hello ";   
s= "Java ";   
String s1= "Hello ";   
String s2=new String( "Hello ");   
  
在jvm的工作过程中，会创建一片的内存空间专门存入string对象。我们把这片内存空间叫做string池，先存放在Java堆中；   
  
String s= "Hello ";当jvm看到 "Hello "，在string池创建string对象存储它，并将他的引用返回给s。   
s= "Java "，当jvm看到 "Java "，在string池创建新的string对象存储它，再把新建的string对象的引用返回给s。而原先的 "Hello "仍然在string池内。没有消失，他是不能被修改的。   
  
所以我们仅仅是改变了s的引用，而没有改变他所引用的对象，因为string对象的值是不能被修改的。   
  
String s1= "Hello ";jvm首先在string池内里面看找不找到字符串 "Hello ",找到,返回他的引用给s1，否则,创建新的string对象，放到string池里。这里由于s= "Hello "了,对象已经被引用，所以依据规则s和s1都是引用同一个对象。所以 s==s1将返回true。(==,对于非基本类型，是比较两引用是否引用内存中的同一个对象)   
  
String s2=new String( "Hello ");jvm首先在string池内里面看找不找到字符串 "Hello ",找到,不做任何事情，否则,创建新的string对象，放到string池里面。由于遇到了new，还会在内存上（不是string池里面）创建string对象存储 "Hello "，并将内存上的（不是string池内的）string对象返回给s2。所以s==s2将返回false，不是引用同一个对象。   
  
好现在我们看题目：   
String s = new String( "xyz ");   
首先在string池内找，找到？不创建string对象，否则创建， 这样就一个string对象   
遇到new运算符号了，在内存上创建string对象，并将其返回给s，又一个对象   
  
所以总共是2个对象

一个例子:   
public class Test   
{   
public static void main(String [] args)   
{   
String s1=new String("test");//创建2个对象，一个Class和一个堆里面   
String s2="test";//创建1个对象，s2指向pool里面的"test"对象   
String s3="test";//创建0个对象,指向s2指想pool里面的那个对象   
String s4=s2;//创建0个对象,指向s2,s3指向pool里面的那个对象   
String s5=new String("test");//创建1个对象在堆里面，注意，与s1没关系   
  
System.out.println(s2=="test");//true s2=="test"很明显true   
System.out.println(s2==s3);//true,因为指向的都是pool里面的那个"test"   
System.out.println(s2==s4);//true,同上,那么s3和s4...:)   
System.out.println(s1==s5);//false,很明显,false   
System.out.println(s1==s2);//false,指向的对象不一样,下面再说   
System.out.println(s1=="test");//false,难道s1!="tset"？下面再说   
  
System.out.println("---------------");   
  
s1=s2;   
System.out.println(s1=="test");//true,下面说   
}   
}

说明：

1,System.out.println(s1==s2);很明显，s2指向的对象"test"是在pool里面，而s1指向的是堆里面的"test"对象(s1指向的内存区)，所以返回false.   
2,System.out.println(s1=="test");s1指向的是堆里面的"test"对象(s1指向的内存区)，而"test"是程序刚刚建立的(其实是共用pool里面的那个已经创建了的"test"对象,也就是我们s2="test"时候，在pool里面创建的)，所以s1指向的堆里的"test"对象   
和"test"(pool里面)并不是一样个对象，所以返回false.   
3,当我们s1=s2;的时候，很明显，把s2的指给s1,s1指向pool里面的"test",这个时候，s2也指向了pool里面的"test"对象了，当System.out.println(s1=="test");时候，java虚拟机创建"test"对象，注意，其实没创建，和前面讲的一 样，公用s1="test"创建的"test"对象(pool里面的),所以，s1=="test"(pool里面的)，同样，s1=s2=s3=s4！   
  
而为什么在网上都说String s=new String("test");创建了2个对象？那可能因为它就写这么一句代码，误让人默认的认为执行代码之前并不实例任何一个String对象过（也许很多人不会这么想，），跟着别人或者不经思考的就说2个，如果说存放在栈内存中专门存放String对象引用的s变量是一个对象！实在不可原谅！