\*\*一.创建字符串\*\*

·\*\*String str="java";\*\*

如下图：修改str2的不会影响str1的值，只会重新创建一个新的对象让str2指向新的对象

![在这里插入图片描述](https://img-blog.csdnimg.cn/20191128192548103.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlYnN0dWR5MTI=,size\_16,color\_FFFFFF,t\_70)

·\*\*String str=new String("java")\*\*;//和方式一比较，虽然都是“java”这个字符串，但地址不同，此方法是重新创建了一个新的对象，拥有新的地址。

![在这里插入图片描述](https://img-blog.csdnimg.cn/20191128192803572.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlYnN0dWR5MTI=,size\_16,color\_FFFFFF,t\_70)

·\*\*char[] array=new char[]{'j','a','v','a'};

String str=new String(array);\*\*

\*\*二.字符串比较相等\*\*

字符串比较相等是比较内容是否相等，用的不是“==”，等号在String字符串比较的是字符串地址是否相同，用的是equals（）方法。

```java

String str1="java";

String str2="java";

String str3=new String("java");

System.out.println(str1==str2);//结果是true，

//因为他们指的是同一字符串,所以地址相同

System.out.println(str1==str3);//结果是false

//他们字符串内容虽然相同，但不属于同一对象，所以地址不相同；

System.out.println(str1.equals(str3));//true

```

\*\*使用equals（）\*方法需要注意！\*

```java

String str1=new String("hello");

//用str1与"hello"这个字符比较内容是否相同

//方式一

System.out.println(str1.equals("hello"));

//方式二

System.out.println("hello".equals(str1));

```

\*\*第二种方式更加确切，原因是，如果str1是null，那么方式一会出现空字符串异常（java.lang.NullPointerException 异常），因为空的字符串没有equals（）方法\*\*

\*\*三.字符串常量池\*\*

![在这里插入图片描述](https://img-blog.csdnimg.cn/2019112820023596.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlYnN0dWR5MTI=,size\_16,color\_FFFFFF,t\_70)

1.str1,str2,str3指向同一个对象的原因是，String类使用了\*\*共享模式设计\*\*，str1通过\*\*直接赋值\*\*的模式进行了实例化对象操作，该实例化对象（字符串内容）将会被保存在\*\*字符串常量池\*\*中，如果后续操作还是采用\*\*直接赋值\*\*实例化对象，如果字符串内容在常量池中有\*\*相同内容\*\*，那么\*\*直接引用\*\*，如果没有，那么\*\*开辟新的空间\*\*将新的字符串对象保存在常量池中，供下一次使用

2.\*\*intern（）\*\*，已经知道直接赋值，如果字符串内容相同，不会开辟新的空间来存储相同内容字符串，如果采用String构造方法创建一个字符串，不管内容是否相同都会开辟新的空间。此时可以使用\*\*intern（）方法\*\*，作用就是让采用String构造方法创建一个内容在常量池当中存在的字符串加到字符串常量池当中，请看下图来理解。

![在这里插入图片描述](https://img-blog.csdnimg.cn/20191128202951248.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlYnN0dWR5MTI=,size\_16,color\_FFFFFF,t\_70)

\*\*面试题：\*\*请解释String类中两种对象实例化的区别？

\*\*1. 直接赋值：只会开辟一块堆内存空间，并且该字符串对象可以自动保存在对象池中以供下次使用。

2. 构造方法：会开辟两块堆内存空间，其中一块成为垃圾空间，不会自动保存在对象池中，可以使用\*\*

\*\*四.字符串的不可变性\*\*

1.

```java

String str="hello";

str=str+"world";

str=str+"!";

System.out.println(str);

//结果是：helloworld!

```

看似字符串内容改变，但只是表象，实际并没有改变！

![在这里插入图片描述](https://img-blog.csdnimg.cn/20191128210939483.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlYnN0dWR5MTI=,size\_16,color\_FFFFFF,t\_70)

2.可以使用以下方法改变内容

```java

String str="hello";

str="H"+str.subString(1);//截取从下标为一开始的所有字符；

//str内容为Hello

```

String 类中有很多方法：可以使用jdk-Api文档来查找