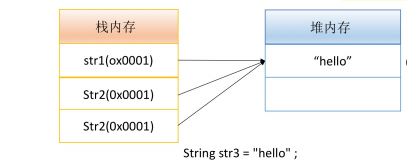
**String类(引用类型）：**

**实例化方式**

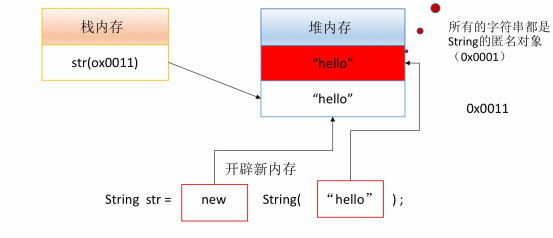
直接赋值（数据保存在常量池中）

String str1="hello";
String str2="hello";
String str3="hello";



构造方法（开辟两块堆内存，一块称为垃圾空间）

String str=new String("hello");

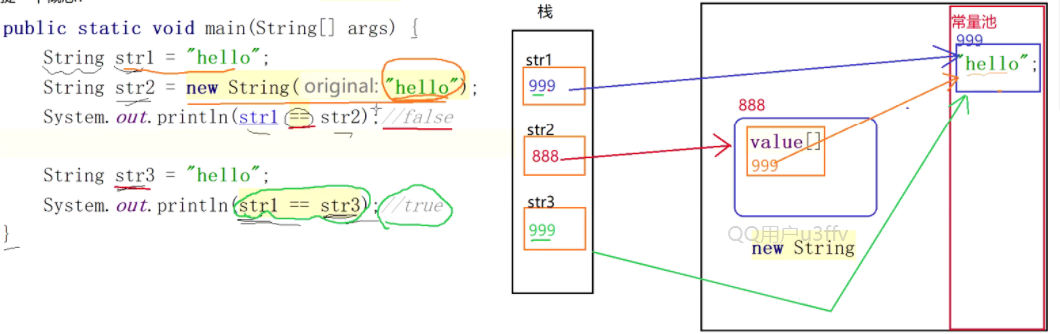


**字符串相等比较（String.equals()方法）**

**==** 与 **equals** 的区别：

1. ”==”：比较的是两个字符串对象。

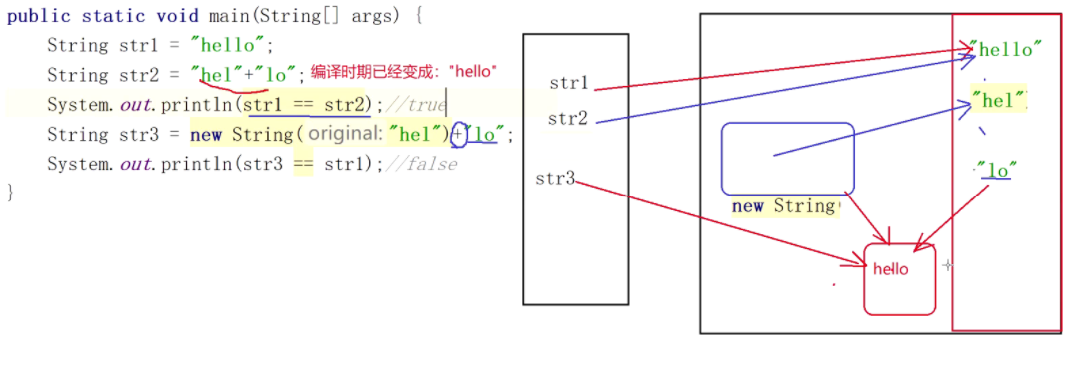
2. “equals（）”：比较字符串的内容



**不可变更性：字符串一旦定义不可改变**

修改：1.字符串拼接截取 2.反射

不可变的好处：1.方便实现字符串对象池 2.线程安全 3. 便于缓存hashcode



**字符与字符数组**

public char[] toCharArray();字符串转字符数组
public String(char value[]);字符数组转字符串
public char[] charAt(int index);获取指定索引位置的字符，下标从0开始

**字节与字符串**

public byte[] getBytes();将字符串以字节数组返回
public String(byte bytes[]);字节数组转字符串

**比较**

equals(Object obj) 区分大小写的比较

equalaIgnoreCase(String str) 不区分大小写的比较

compareTo(String str) 比较两个字符串的大小关系

**查找**

使⽤indexOf()需要注意的是，如果内容重复，它只能返回查找的第⼀个位置



**替换**

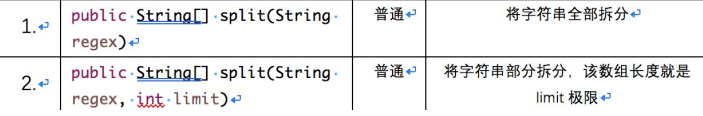


**拆分**

1. 字符"|","\*","+"都得加上转义字符，前面加上"\".

2. 而如果是""，那么就得写成"\\".

3. 如果一个字符串中有多个分隔符，可以用"|"作为连字符.



**截取**



**其他方法**



**StringBuffer:内容可以修改**

String——>StringBuffffer：StringBuffffer的构造方法或append()方法

StringBuffffer——>String：toString()⽅法

字符串反转:reverse()

删除指定范围数据:delete(int start,int end)

插入数据:insert(int offset,数据）

**请解释String、StringBuffffer、StringBuilder的区别:**

String的内容不可修改，StringBuffffer与StringBuilder的内容可以修改.

StringBuffffer采用同步处理，属于线程安全操作；

StringBuilder采⽤异步处理，属于线程不安全操作。

**Object类**

**Object**可以接收任意的对象，因为**Object**是所有类的父类，但是**Obejct**并不局限于

此，它可以接收所有数据类型，包括：类、数组、接⼝。

**包装类**

对象型(Object的直接子类):Boolean、Character(char)；

数值型(Number的直接子类): Byte、Double、Short、Long、Integer(int)、Float；

**1.装箱与拆箱：**

装箱：将基本数据类型包装为包装类

拆箱：将基本数据从包装类中拆出

Integer num = new Integer(55) ; // 装箱
int data = num.intValue() ;

所有的相同类型的包装类对象之间值的比较，全部使⽤ equals ⽅法⽐较。

**2.字符串与基本数据类型的转换：**

字符串转基本数据：

String变为int 类型（Integer类）： parseInt(String s)

String变为double类型（Double类）：parseDouble(String s)

String变为Boolean类型（Boolean类）： parseBoolean(String s)

注意：将字符串转为数字的时候，字符串的组成日过有非数字，就会出现错误（NumberFormatException）

基本数据转字符串：

“+”连接符

String的valueOf()方法，不产生垃圾