

String

Java 通过 String 类来创建和操作字符串数据。

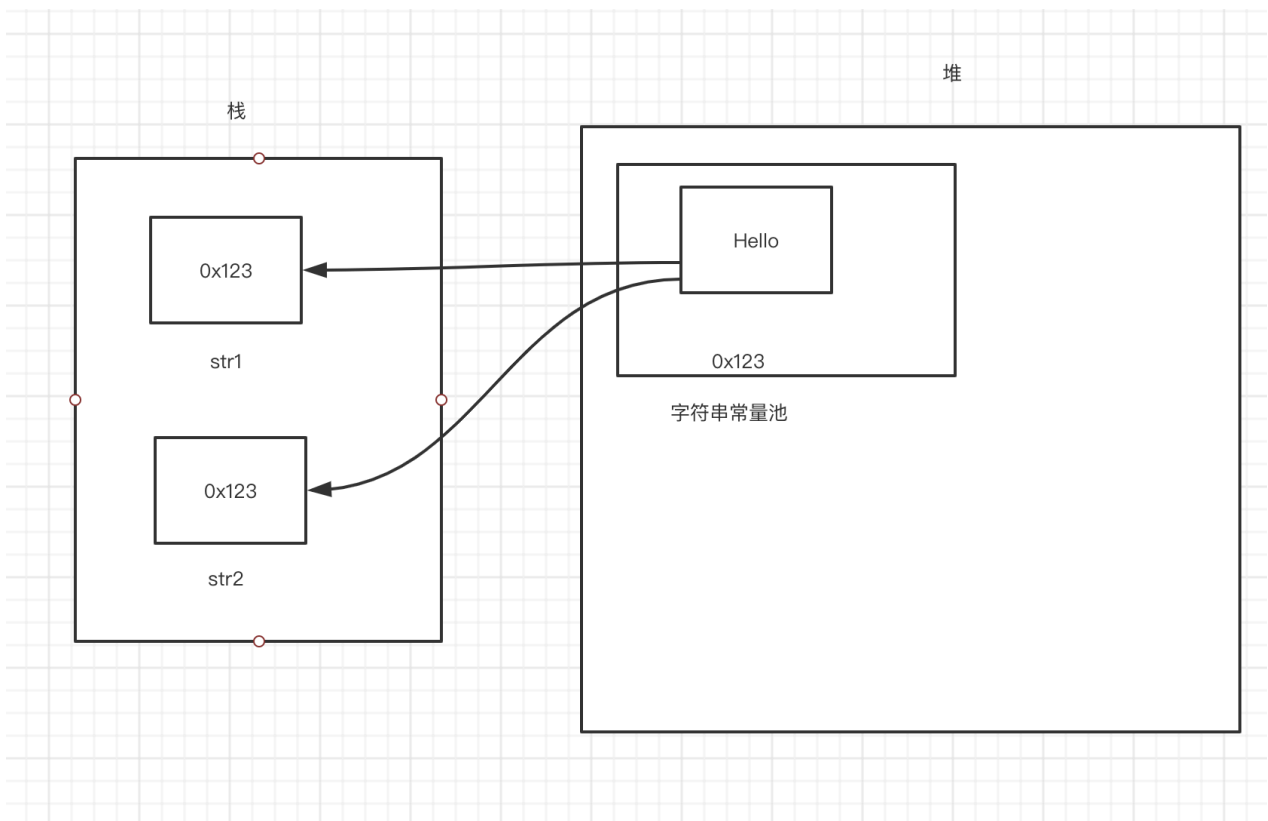
- String 实例化

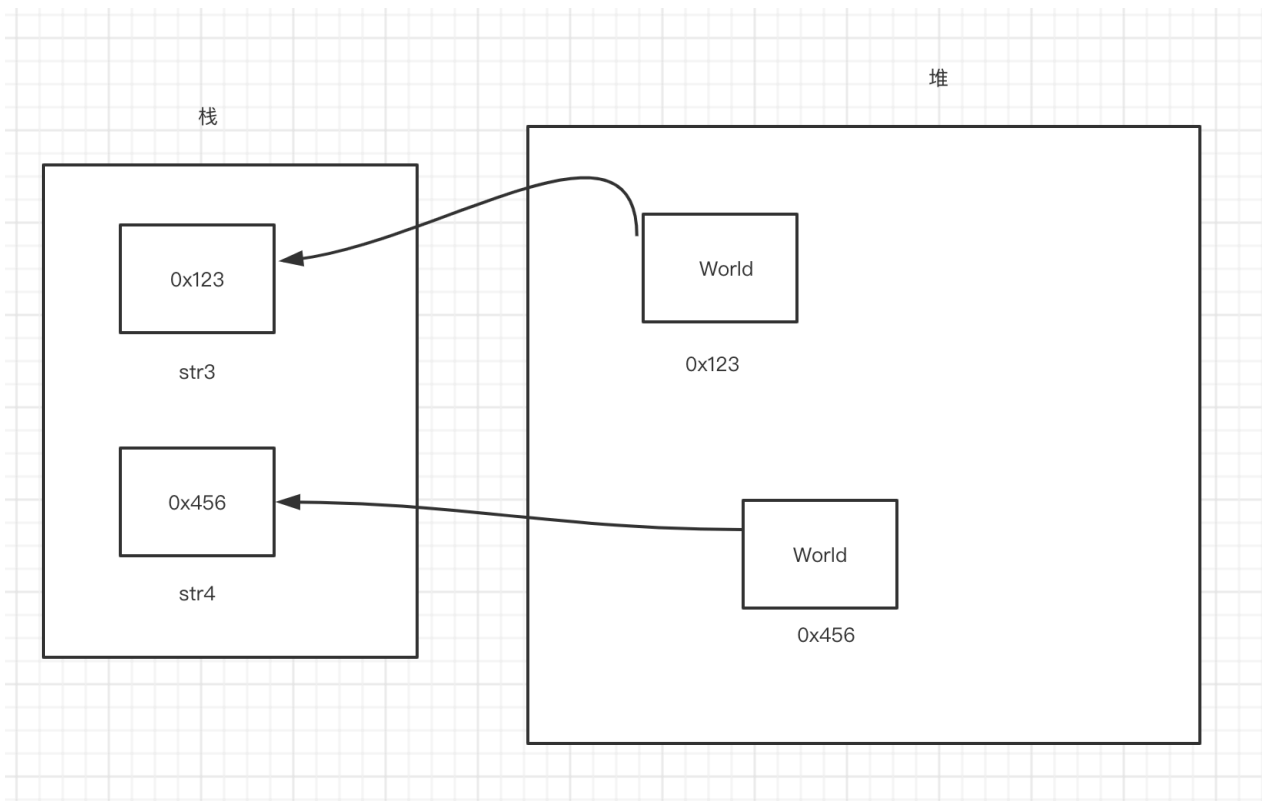
1、直接赋值

```
String str = "Hello World";
```

2、通过构造函数创建对象

```
String str = new String("Hello World");
```





```
isLatin1() ? StringLatin1.equals(value,
aString.value)
:StringUTF16.equals(value, aString.value);
```

三目运算符 三元表达式

String 常用方法

方法	描述
<code>public String()</code>	创建一个空的字符串对象
<code>public String(String value)</code>	创建一个值为 <code>value</code> 的字符串对象
<code>public String(char value[])</code>	将一个char数组转换为字

	字符串对象
public String(char value[],int offset, int count)	将一个指定范围的char数组转为字符串对象
public String(byte value[])	将一个byte数组转换为字符串对象
public String(byte value[],int offset, int count)	将一个指定范围的byte数组转为字符串对象
public int length()	获取字符串的长度
public boolean isEmpty()	判断字符串是否为空
public char charAt(int index)	返回指定下标的字符
public byte[] getBytes()	返回字符串对应的byte数组
public boolean equals(Object anObject)	判断两个字符串值是否相等
public boolean equalsIgnoreCase(Object anObject)	判断两个字符串值是否相等（忽略大小写）
public int compareTo(String value)	对字符串进行排序
public int compareToIgnoreCase(String value)	忽略大小写进行排序
public boolean startsWith(String value)	判断字符串是否以 value 开头

public boolean endsWith(String value)	判断字符串是否以 value 结尾
public int hashCode()	返回字符串的 hash 值
public int indexOf(String str)	返回 str 在字符串中的下标
public int indexOf(String str,int formIndex)	从指定位置查找字符串的下标
public String subString(int beginIndex)	从指定位置开始截取字符串
public String subString(int beginIndex,int endIndex)	截取指定区间的字符串
public String concat(String str)	追加字符串
public String replaceAll(String o,String n)	将字符串中所有的 o 替换成 n
public String[] split(String regex)	用指定的字符串对目标进行分割，返回数组
public String toLowerCase()	转小写
public String toUpperCase()	转大写
public char[] toCharArray()	将字符串转为字符数组

null 和空是两个概念。

null 是指对象不存在，引用地址为空。

空是指对象存在，没有内容，长度为零。

