字符串（String）

@M了个J
李明杰

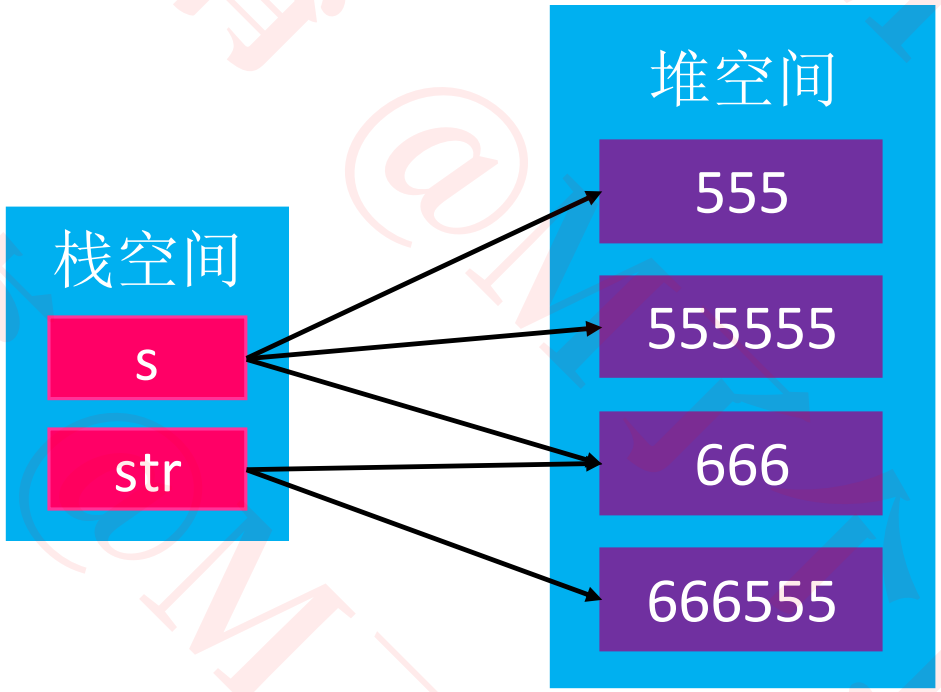
https://github.com/CoderMJLee
http://cnblogs.com/mjios

码拉松

字符串（String）

- Java 中用 java.lang.String 类代表字符串
  - 底层使用 `char[]` 存储字符数据，从 Java 9 开始，底层使用 `byte[]` 存储字符数据
  - 所有字符串字面量（比如`"abc"`）都是 String 类的实例
  - String 对象一旦创建完毕，它的字符内容是不可以修改的

```
String s = "555";
s += "555";
s = "666";
test(s);
System.out.println(s);

void test(String str) {
    str += "555";
}
```

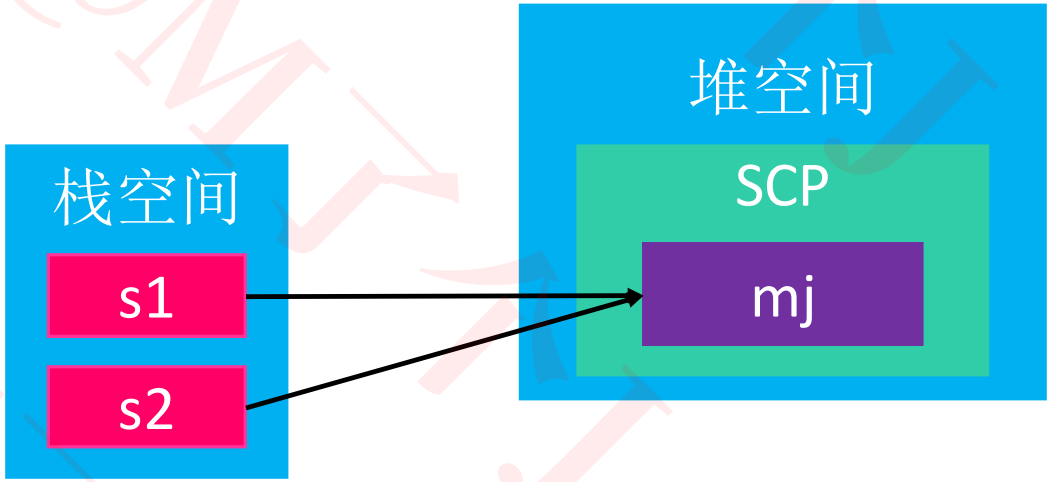


字符串常量池（String Constant Pool）

- Java 中有个字符串常量池（String Constant Pool，简称 SCP）
  - 从 Java 7 开始属于堆空间的一部分（以前放在方法区）

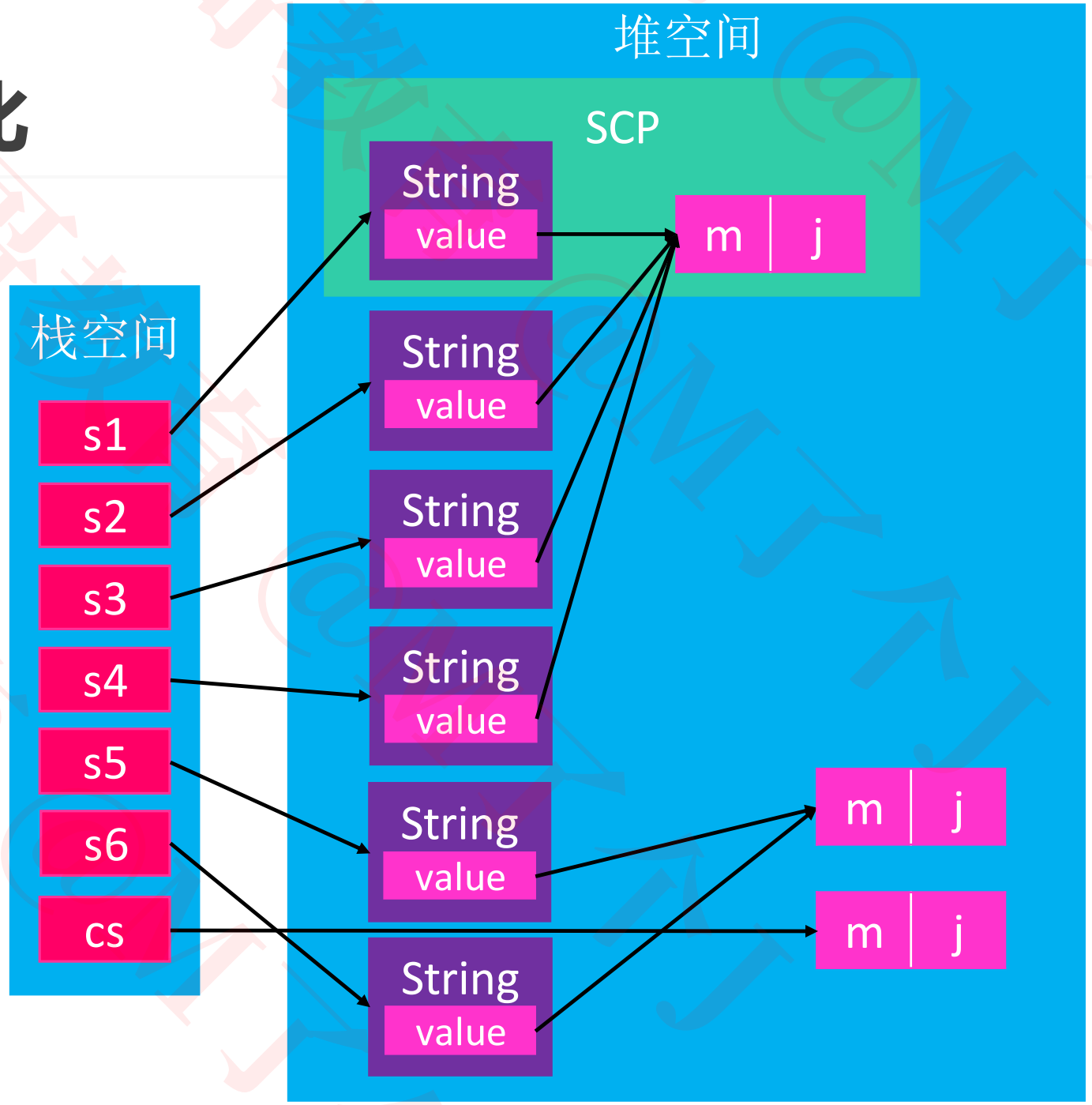
- 当遇到字符串字面量时，会去查看 SCP
  - 如果 SCP 中存在与字面量内容一样的字符串对象 A 时，就返回 A
  - 否则，创建一个新的字符串对象 D，并加入到 SCP 中，返回 D

```
String s1 = "mj";
String s2 = "mj";
System.out.println(s1 == s2); // true
```



字符串的初始化

```
String s1 = "mj";
String s2 = new String("mj");
String s3 = new String(s1);
String s4 = new String(s2);
char[] cs = { 'm', 'j' };
String s5 = new String(cs);
String s6 = new String(s5);
```



小码哥教育 SEEMYGO

intern 方法

- A.intern 方法的作用
  - 如果 SCP 中存在与 A 内容一样的字符串对象 C 时，就返回 C
  - 否则，将 A 加入到 SCP 中，返回 A

```
int a = 1, b = 2, c = 3;
String s1 = String.format("%d%d%d", a, b, c);
String s2 = String.format("%d%d%d", a, b, c);
String s3 = s1.intern();
String s4 = s2.intern();
String s5 = "123";

System.out.println(s1 == s2); // false
System.out.println(s1 == s3); // true
System.out.println(s1 == s4); // true
System.out.println(s1 == s5); // true
```

小码哥教育 SEEMYGO

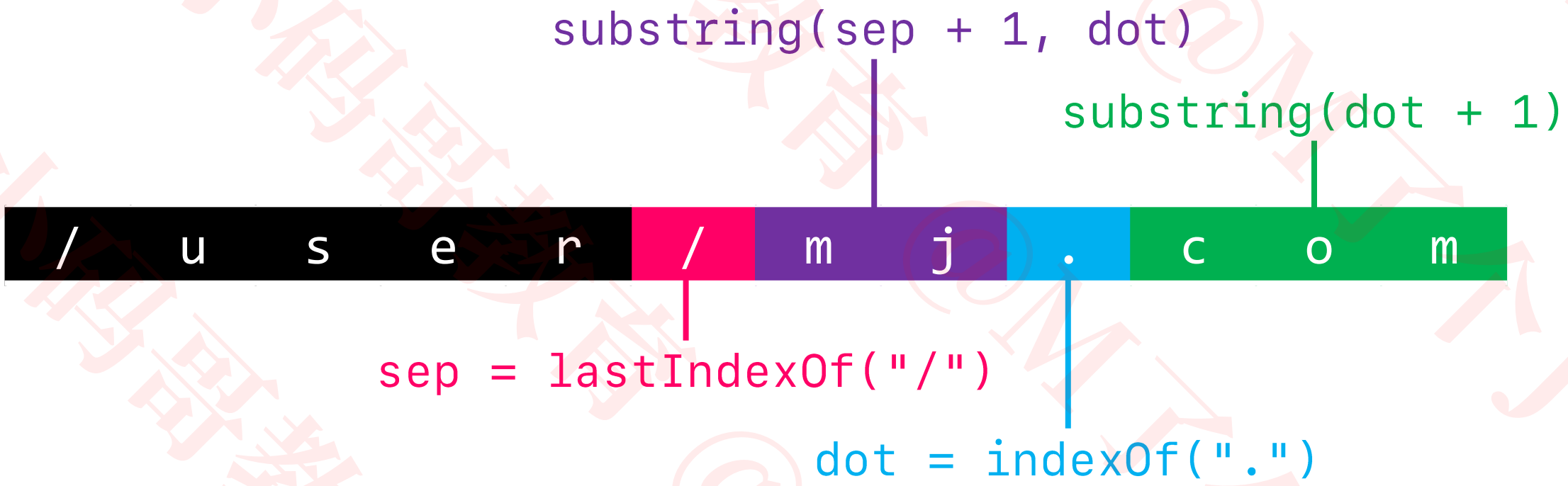
字符串的常用方法

```
// 去除左右的空格：“123  456”
" 123  456  ".trim()

// 转为大写字母：ABC
"abc".toUpperCase()
// 转为小写字母：abc
"ABC".toLowerCase()

// 是否包含某个字符串：true
"123456".contains("34")
// 是否以某个字符串开头：true
"123456".startsWith("123")
// 是否以某个字符串结尾：true
"123456".endsWith("456")

// 将字符串分割为数组：[1, 2, 3, 4]
"1_2_3_4".split("_")
```

```
// 比较字符串的大小：< 0
"abc".compareTo("adc")
// 忽略大小写比较字符串的大小：< 0
"abc".compareToIgnoreCase("ADC")

String s1 = "abc";
String s2 = new String("abc");
// 查看字符串的内容是否相等：true
s1.equals(s2)
// 忽略大小写查看字符串的内容是否相等：true
"abc".equalsIgnoreCase("aBc")
```

小码哥教育 SEEMYGO

字符串截取



小码哥教育 SEEMYGO

StringBuilder

- 在进行大量字符串的改动操作时（比如拼接、替换）
  - 使用 String 会非常消耗内存、降低程序性能
  - 使用 StringBuilder 可以节省内存、提高程序性能

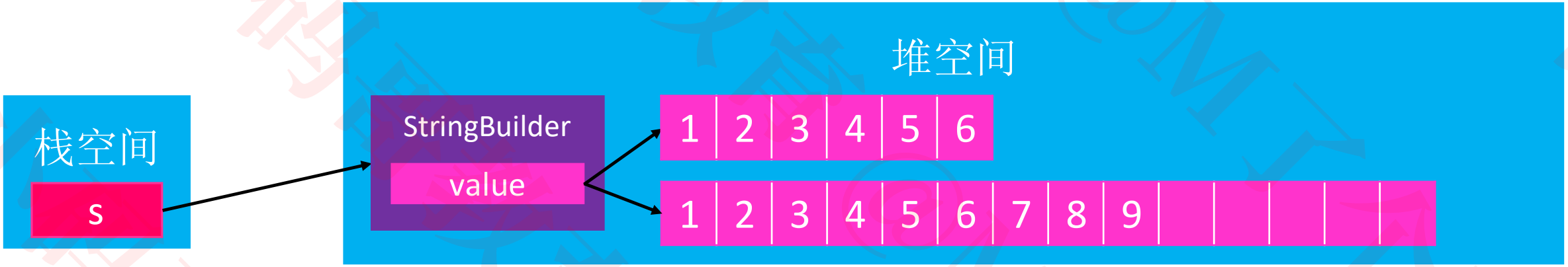
```
String s1 = "";
s1 += "123";
s1 += "456";

StringBuilder s2 = new StringBuilder();
s2.append("123").append("456");
```

- StringBuilder 的常用方法有：append、insert、delete、replace、reverse等

- 注意
  - StringBuilder 并不是 String 的子类 或者 父类
  - StringBuilder、String 都实现了 CharSequence 接口

StringBuilder 的 append 原理

```
StringBuilder s = new StringBuilder();
s.append("123").append("456").append("789");
```



- StringBuilder 的默认容量是 16，扩容后的新容量是原来容量的 2 倍 + 2
  - 16 扩容为 34
  - 34 扩容为 70
  - 70 扩容为 142
  - ……