# Java StringBuffer & StringBuilder



## 一、StringBuffer和StringBuilder的区别是什么？

Java中的StringBuffer和StringBuilder类是用于处理字符串的可变对象；

它们与String类类似，但是不同之处在于，StringBuffer和StringBuilder可以修改字符串内容，而String对象是不可变的；

StringBuffer和StringBuilder类的用法基本相同，但StringBuilder类的效率更高，因为它不是线程安全的；

## 二、StringBuffer的示例

// 创建一个空的StringBuffer对象  
StringBuffer sb = new StringBuffer();  
  
// 向StringBuffer对象中添加字符串  
sb.append("Hello");  
sb.append(" ");  
sb.append("World");  
  
// 将StringBuffer对象转换为String对象  
String str = sb.toString();  
System.out.println(str); // 输出: Hello World  
  
// 在指定位置插入字符串  
sb.insert(5, " my");  
System.out.println(sb.toString()); // 输出: Hello my World  
  
// 删除指定位置的字符  
sb.deleteCharAt(5);  
System.out.println(sb.toString()); // 输出: Hellomy World  
  
// 反转字符串  
sb.reverse();  
System.out.println(sb.toString()); // 输出: dlroW ymolleH

## 三、StringBuilder的示例

// 创建一个空的StringBuilder对象  
StringBuilder sb = new StringBuilder();  
  
// 向StringBuilder对象中添加字符串  
sb.append("Hello");  
sb.append(" ");  
sb.append("World");  
  
// 将StringBuilder对象转换为String对象  
String str = sb.toString();  
System.out.println(str); // 输出: Hello World  
  
// 在指定位置插入字符串  
sb.insert(5, " my");  
System.out.println(sb.toString()); // 输出: Hello my World  
  
// 删除指定位置的字符  
sb.deleteCharAt(5);  
System.out.println(sb.toString()); // 输出: Hellomy World  
  
// 反转字符串  
sb.reverse();  
System.out.println(sb.toString()); // 输出: dlroW ymolleH

## 四、为什么StringBuffer和StringBuilder比String更适合在循环中使用？

由于String对象是不可变的，每次修改String对象都会创建一个新的对象，这会造成大量的对象创建和销毁，导致性能问题。相比之下，StringBuffer和StringBuilder都是可变的，可以直接修改其内部字符数组的内容，避免了创建新的对象，因此更适合在循环中使用。

## 五、如何将String对象转换为StringBuilder或StringBuffer对象？

可以使用StringBuilder或StringBuffer类的构造函数将String对象转换为相应的可变字符串对象。例如，可以使用以下代码将String对象"hello"转换为StringBuilder对象：

String str = "hello";  
StringBuilder sb = new StringBuilder(str);

## 六、如何将StringBuilder或StringBuffer对象转换为String对象？

可以使用StringBuilder或StringBuffer对象的toString()方法将其转换为String对象。例如，可以使用以下代码将StringBuilder对象sb转换为String对象：

StringBuilder sb = new StringBuilder("hello");  
String str = sb.toString();