# 1.StringBuffer基本概念

String和StringBuffer相比,String对象是不能被修改的,

但是StringBuffer对象可以改变

数据的追加

Public StringBuffer append(数据类型 变量)

|  |
| --- |
| StringBuffer buf = new StringBuffer();  buf.append("haahah").append(123).append("!!"); |

注意:StringBuffer不能直接和String一样接受一个字符串,即StringBuffer buf = “hello” 错误的

# 2.String和StringBuffer对比

|  |  |
| --- | --- |
| **String类** | **StringBuffer类** |
| public final class String  extends Object  implements Serializable, Comparable<String>, CharSequence | public final class StringBuffer  extends Object  implements Serializable,  CharSequence |

:

# 3. String和StringBuffer相互转化

## 3.1String转化为StringBuffer

有两种方式:

1. 利用StringBuffer类的构造方法:

|  |
| --- |
| public StringBuffer(String str) |
| public StringBuffer(CharSequence seq) |

1. 利用append方法

|  |
| --- |
| public StringBuffer append(String str) |

范例:

|  |
| --- |
| StringBuffer buf = new StringBuffer("helllo,world");  StringBuffer buf1 = new StringBuffer().append("hello,world"); |

## 3.2StringBuffer转化为String

有两种方式:

1. 利用toString()可以将StringBuffer变为String
2. 利用String的构造函数

|  |
| --- |
| public String(StringBuffer buffer) |

范例:

|  |
| --- |
| StringBuffer buf = new StringBuffer("hello,world");  String str1 = buf.toString();  String str2 = new String(buf); |

# 4.String和StringBuffer内容是否相等

|  |
| --- |
| public boolean contentEquals(StringBuffer sb) |

|  |
| --- |
| StringBuffer buf = new StringBuffer("hello,world");  String str1 = "hello,world";  System.out.print(str1.contentEquals(buf)); |

# 5.StringBuffer字符串反转

**public StringBuffer reverse();**

范例:

|  |
| --- |
| StringBuffer buf = new StringBuffer("abc");  StringBuffer x = buf.reverse();  System.out.print(buf == x); //true 同一个地址  System.out.print(x); //cba |

# 6.再指定位置前插入字符串

Public StringBuffer insert(int offset,数据类型 变量)

范例:

|  |
| --- |
| StringBuffer buf = new StringBuffer("abc");  buf.insert(1,"ww"); //在下标为1的前面插入字符串  System.out.print(buf); // awwbc |

# 7.删除部分数据

public StringBuffer delete(int start, int end)

范例

|  |
| --- |
| StringBuffer buf = new StringBuffer("abcde");  buf.delete(1,3); //删除从下标为1开始到下标为3(不包括下标为3)的元素  System.out.print(buf); //ade |