

# String, StringBuffer and StringBuilder

---

## 一、String

### 1、特性

不可变：String 被声明为 final，因此它不可被继承。

内部使用 char 数组存储数据，该数组被声明为 final，这意味着 value 数组初始化之后就不能再引用其它数组。并且 String 内部没有改变 value 数组的方法，因此可以保证 String 不可变。

```
public final class String implements java.io.Serializable, Comparable<String>,
CharSequence {
    /** The value is used for character storage. */
    private final char value[];
}
```

### 2、不可变的好处

#### 1) 可以缓存 hash 值

因为 String 的 hash 值经常被使用，例如 String 用做 HashMap 的 key。不可变的特性可以使得 hash 值也不可变，因此只需要进行一次计算。

#### 2) String Pool 的需要

如果一个 String 对象已经被创建过了，那么就会从 String Pool 中取得引用。只有 String 是不可变的，才可能使用 String Pool。

#### 3) 安全性

String 经常作为参数，String 不可变性可以保证参数不可变。例如在作为网络连接参数的情况下如果 String 是可变的，那么在网络连接过程中，String 被改变，改变 String 对象的那一方以为现在连接的是其它主机，而实际情况却不一定是。

#### 4) 线程安全

String 不可变性天生具备线程安全，可以在多个线程中安全地使用。

### 3、String.intern()

使用 `String.intern()` 可以保证相同内容的字符串变量引用同一的内存对象

## 四、String, StringBuffer , StringBuilder的区别

### 1. 可变性

- String 不可变
- StringBuffer 和 StringBuilder 可变

### 2. 线程安全

- String 不可变，因此是线程安全的
- StringBuilder 不是线程安全的，但比StringBuffer效率更高
- StringBuffer 是线程安全的，内部使用 `synchronized` 进行同步