

# Java-常用库-String相关

## Java常见包及其含义

包名	含义
Java.lang	语言包
Java.util	实用工具
Java.awt	抽象窗口工具
Java.applet	Applet应用程序
Java.text	文本工具
Java.io	输入输出流及文件
Java.net	网络功能
Java.rmi	远程方法调用
Java.sql	数据库API

## String与StringBuffer

### String - 字符串类

- 用来表示字符序列，符合Unicode标准，初始化方式如下：

```
// 实质上是构建对象
String s1 = "Hello!";
String s2 = new String("Hello!");
```

- 特点：对象不可变，所有看起来修改对象的方法，实际上是创建了一个新的String对象。
- String对象作为参数传递的特点：传引用，即形参和传入函数的实参指向同一个对象。但是由于String不可变，对形参的改变并不会影响实参的内容。
- String之间的比较：区分引用比较与值比较

```
public class Test {
    public static void main(String args[]) {
        String s = new String("Hello");
        String t = new String("Hello");
        // 引用比较：比较变量所引用的是不是同一个对象
        if (s==t) {
            System.out.println("相等");
        } else {
            System.out.println("不相等");
        }
    }
}
```

```

        // 值比较：比较两个对象的值是否相同
        if (s.equals(t)) {
            System.out.println("相等");
        } else {
            System.out.println("不相等");
        }
    }
}

```

以上结果输出：

```

不相等
相等

```

## String, StringBuffer和StringBuilder

- String内部基于常量字符数组，内容不可变；StringBuffer和StringBuilder内部基于字符数组，内容可变。
  - StringBuffer可调用 `append(String s)` 方法，在当前字符串后面追加内容。
- 处理字符串的性能：StringBuilder > StringBuffer > String
- 线程安全：StringBuilder不具备线程安全。

## String与正则表达式

- String对象调用 `public boolean matches(String regex)` 方法可以判断当前String对象的字符序列是否和参数regex指定的正则表达式匹配。方法返回布尔值。
- 常用的正则表达式
  - ①匹配整数（十进制）的正表达式regex: `String regex = "-?[1-9]\\d*";`
  - ②匹配浮点数的正表达式regex: `String regex = "-?[0-9][0-9]*[.][0-9]+";`
  - ③匹配email的正表达式regex: `String regex = "\\w+@\\w+\\. [a-z]+(\\. [a-z]+)?";`
  - ④匹配18位身份证号码（最后一位是数字或字母）的正表达式regex:
 

```
String regex = "[1-9][0-9]{16}[a-zA-Z0-9]{1}";
```
  - ⑤匹配日期的正则表达式
 

不考虑二月的特殊情况，匹配日期（年限制为4位）的正则表达式regex:

```
String year = "[1-9][0-9]{3}";
String month = "((0?[1-9])|(1[012]))";
String day = "((0?[1-9])|([12][0-9])|(3[01]?))";
String regex = year+"[-./]+" + month+"[-./]+" + day;
```
- 使用正则表达式分割字符串
  - `public String[] split(String regex)` 使用参数指定的正则表达式regex做为分隔标

记分解出其中的单词，并将分解出的单词存放在字符串数组中。

- 分割标记不会被存入返回的字符串数组中。

## String与Scanner类

- Scanner对象可以解析字符序列中的单词，例如，对于String对象NBA `String NBA= "I Love This Game"`. 为了解析出NBA的字符序列中的单词，可以如下构造一个Scanner对象：

```
Scanner scanner = new Scanner(NBA);
```

- 然后Scanner对象可以调用方法 `useDelimiter(正则表达式)` 来分析字符串。