

## 2.1 String 概述

String 类在 **java.lang** 包下，所以使用的时候不需要导包

**String** 类代表**字符串**，Java 程序中的所有字符串文字（例如 “abc”）都被实现为此类的实例  
也就是说，**Java 程序中所有的双引号字符串，都是 String 类的对象**

### 字符串的特点

- 字符串不可变，它们的值在创建后不能被更改
- 虽然 String 的值是不可变的，但是它们可以被共享
- 字符串效果上相当于字符数组 (**char[]**)，但是底层原理是字节数组 (**byte[]**)

## 2.2 String 构造方法

方法名	说明
public String()	创建一个空白字符串对象，不含有任何内容
public String(char[] chs)	根据字符数组的内容，来创建字符串对象
public String(byte[] bys)	根据字节数组的内容，来创建字符串对象
String s = "abc" ;	直接赋值的方式创建字符串对象，内容就是abc

## 2.4 字符串的比较

使用 **==** 做比较

- 基本类型：比较的是**数据值**是否相同
- 引用类型：比较的是**地址值**是否相同

字符串是对象，它比较内容是否相同，是通过一个方法来实现的，这个方法叫：**equals()**

- public boolean **equals(Object anObject)**：将此字符串与指定对象进行比较。由于我们比较的是字符串对象，所以参数直接传递一个字符串

### 案例：遍历字符串

需求：键盘录入一个字符串，使用程序实现在控制台遍历该字符串

思路：

- ① 键盘录入一个字符串，用 Scanner 实现
- ② 遍历字符串，首先要能够获取到字符串中的每一个字符
  - public char charAt(int index)：返回指定索引处的char值，字符串的索引也是从0开始的
- ③ 遍历字符串，其次要能够获取到字符串的长度
  - public int **length()**：返回此字符串的长度
  - 数组的长度：数组名.length
  - 字符串的长度：字符串对象.length()
- ④ 遍历字符串的通用格式

```
for(int i=0; i<s.length(); i++) {  
    s.charAt(i); // 就是指索引处的字符值  
}
```

### 3.1 StringBuilder概述

```
public static void main(String[] args) {
    String s = "hello";
    s += "world";
    System.out.println(s);
}
```

输出: "helloworld"



如果对字符串进行拼接操作，每次拼接，都会构建一个新的String对象，既耗时，又浪费内存空间，而这种操作还不可避免。那么有没有一种比较好的方式可以解决这个问题呢？答案是肯定的，我们可以通过Java提供的**StringBuilder**类来解决这个问题。

### 3.3 StringBuilder 的添加和反转方法

方法名	说明
public StringBuilder append(任意类型)	添加数据，并返回对象本身
public StringBuilder reverse()	返回相反的字符序列

### 3.4 StringBuilder 和 String 相互转换

#### 1. StringBuilder 转换为 String

public String **toString()**: 通过 toString() 就可以实现把StringBuilder转换为String

#### 2. String 转换为 StringBuilder

public **StringBuilder(String s)**: 通过构造方法就可以实现把String转换为StringBuilder

### 3.5 通过帮助文档查看 StringBuilder 中的方法

方法名	说明
public StringBuilder append (任意类型)	添加数据，并返回对象本身
public StringBuilder reverse()	返回相反的字符序列
public String toString()	通过toString()就可以实现把StringBuilder转换为String

```
import java.util.Scanner;

public class Text3{

    public static void main(String[] args) {

        String name = "dy";

        String password = "111";

        for(int i=0;i<3;i++){

            Scanner sc = new Scanner(System.in);

            System.out.println("请输入登录名");

            String inputname = sc.nextLine();

            System.out.println("请输入登录密码");

            String inputword = sc.nextLine();

            if(inputname.equals(name) && inputword.equals(password)){

                System.out.println("登陆成功");

                break;

            }else if(2-i == 0){

                System.out.println("登陆失败，请与管理员联系");

            }else{

                System.out.println("登陆失败，你还有" + (2-i) + "次机会");

            }

        }

    }

}
```

注意使用 equals 来比较输入的 String 内容和定义的 String 的内容是否相同。