



# 两种实例化方式的区别

　　　　　　1）直接赋值（String str = "hello"）：只开辟一块堆内存空间，并且会自动入池，不会产生垃圾。

　　　　　　2）构造方法（String str=  new String("hello");）:会开辟两块堆内存空间，其中一块堆内存会变成垃圾被系统回收，而且不能够自动入池，需要通过public  String intern();方法进行手工入池。

Static

可以修饰的内容：1变量2方法3类4静态代码块

的特点

# String：方法系列

### String类的获取功能

int length():获取字符串的长度，其实也就是字符个数

char charAt(int index):获取指定索引处的字符

int indexOf(String str):获取str在字符串对象中第一次出现的索引

String substring(int start):从start开始截取字符串

String substring(int start,int end):从start开始，到end结束截取字符串。包括start，不包括end

### String的判断功能

　　boolean equals(Object obj)：比较字符串的内容是否相同

　　boolean equalsIgnoreCase(String str)： 比较字符串的内容是否相同,忽略大小写

　　boolean startsWith(String str)： 判断字符串对象是否以指定的str开头

　　boolean endsWith(String str)： 判断字符串对象是否以指定的str结尾

### String的转换功能

　　char[] toCharArray()：把字符串转换为字符数组

　　String toLowerCase()：把字符串转换为小写字符串

　　String toUpperCase()：把字符串转换为大写字符串

### 其他常用方法

　　去除字符串两端空格：String trim()

　　按照指定符号分割字符串：String[] split(String str)

### 在String的情况下==与equals的区别

基本数据类型：比较的是基本数据类型的值是否相同  
 引用数据类型：比较的是引用数据类型的地址值是否相同  
 所以在这里的话：String类对象==比较，比较的是地址，而不是内容

### 代码1 eq与==的区别

String str1 = **"Lance"**;  
String str2 = **new** String(**"lance"**);  
String str3 = str2; *//引用传递，str3直接指向st2的堆内存地址*String str4 = **"Lance"**;  
System.***out***.println(str1+str4);  
*/\*\*  
 \* ==:  
 \* 基本数据类型：比较的是基本数据类型的值是否相同  
 \* 引用数据类型：比较的是引用数据类型的地址值是否相同  
 \* 所以在这里的话：String类对象==比较，比较的是地址，而不是内容  
 \*/*System.***out***.println(str1.equalsIgnoreCase(str2));  
System.***out***.println(str4.equals(str3));  
System.***out***.println(str1 == str2);*//false*System.***out***.println(str1 == str3);*//false*System.***out***.println(str3 == str2);*//true*System.***out***.println(str1 == str4);*//true*System.***out***.println(str4.toUpperCase());

### 代码2 手机号

*//存放的是电话号码包含的数字***int**[] nums = {4, 2, 1, 0, 8, 3};  
*//电话号码出现的下标***int**[] index = {2, 4, 5, 2, 3, 0, 3, 2, 1, 0, 0};  
*// 求电话号码15753049998***for** (**int** i = 0; i < index.**length**; i++) {  
 System.***out***.print(nums[index[i]]);  
}  
System.***out***.println();  
**for** (**int** a : index) {  
 System.***out***.print(nums[a]);  
}

### 代码3 二维数组

**int**[][] arr = {{1}, {2, 2}, {3, 3, 3}, {4, 4, 4, 4}, {5, 5, 5, 5, 5}, {6, 6, 6, 6, 6, 6}};  
**for** (**int** i = 0; i < arr.**length**; i++) { *//遍历二维数组，遍历出来的每一个元素是一个一维数组* **for** (**int** j = 0; j < arr[i].**length**; j++) { *//遍历对应位置上的一维数组* System.***out***.print(arr[i][j]);  
 }  
 System.***out***.println();  
}  
**for** (**int**[] i:arr){  
 **for** (**int** j:i){  
 System.***out***.print(j);  
 }  
 System.***out***.println();  
}

### 代码4 算出出现次数

**public static void** main(String[] args) {  
 String str = **"吃葡萄不吐葡萄皮，不吃葡萄倒吐葡萄皮"**;  
 *//请用四种方法统计 葡萄 出现的次数  
 // 第一种* String str1 = str.replace(**"葡萄"**, **" "**);  
 **char**[] data = str1.toCharArray();  
 **int** count = 0;  
 **for** (**char** a : data) {  
 **if** (a == **' '**) {  
 count++;  
 }  
 }  
 System.***out***.println(count);  
 *// 第二种* **int** countt = 0;  
 **int** x = 0;  
 **while** (x != -1) {  
 x = str.indexOf(**"葡萄"**);  
 **if** (x != -1) {  
 countt++;  
 str = str.replaceFirst(**"葡萄"**, **""**);  
 }  
 }  
 System.***out***.println(countt);  
 *// 第五种* **int** counte = 0;  
 **int** xx = 0;  
 **while** (xx != -1) {  
 xx = str.indexOf(**"葡萄"**);  
 **if** (xx != -1) {  
 **if** ((xx + 2) < str.length()) {  
 str = str.substring(xx + 2);  
 }  
 counte++;  
 }  
 }  
 System.***out***.println(count);  
  
  
}