

Java基础(Java SE)

字符串



String

- ▷ String类适用于描述字符串事物。
- ▷ 那么它就提供了多个方法对字符串进行操作。 常见的操作有哪些？

String

▷ 1.获取

▷ 1.1字符串中包含的字符数，也就是字符串的长度。 `int length():获取长度。`

▷ 1.2根据位置获取位置上的某个字符。 `char charAt(int index)`

▷ 1.3根据字符获取该字符在字符串中的位置。

`int indexOf(int ch):`返回的是ch在字符串中第一次出现的位置。

`int indexOf(int ch,int fromIndex):`从fromIndex指定位置开始，获取ch在字符串中出现的位置。

`int indexOf(String str):`返回的是str在字符串中第一次出现的位置。

`int indexOf(String str,int fromIndex):`从fromIndex指定位置开始，获取str在字符串中出现的位置。

String

▷ 2.判断

▷ 2.1字符串中是否包含某一个子串。

`boolean contains(str);`

特殊之处：`indexOf(str)`:可以索引str第一次出现位置，
如果返回-1，表示该str不在字符串中存在。

所以，也可以用于对指定判断是否包含。

`if(str.indexOf("aa") != -1)`

而且该方法既可以判断，也可以获取出现的位置。

▷ 2.2字符串中是否有内容。

`boolean isEmpty();`原理就是判断长度是否为0。

▷ 2.3字符串是否是以指定内容开头。

`boolean startsWith(str);`

▷ 2.4字符串是否是以指定内容结尾。

`boolean endsWith (str)`

▷ 2.5判断字符串的内容是否相同，复写了Object类中的equals方法。

`boolean equals(str);`

▷ 2.6判断内容是否相同，并忽略大小写。

`boolean equalsIgnoreCase();`

String

▷ 3.转换。

▷ 3.1将字符数组转换成字符串

▷ 构造函数:

```
String (char[])
```

```
String (char[], offset,count) :将一部分字符数组转换成字符串
```

▷ 静态方法:

```
static String copyValueOf(char[]);
```

```
static String copyValueOf(char[] data,int offset,int count);
```

```
static String valueOf(char[]);
```

▷ 3.2将字符串转成字符数组。 `char[] toCharArray();`

▷ 3.3将字节数组转成字符串

```
String(byte[])
```

```
String(byte[],offset,count):将字节数组中的一部分转成字符串
```

String

▷ 3.4将字符串转成字节数组

```
byte[] getBytes();
```

▷ 3.5将基本数据类型转成字符串。

```
static String valueOf(int)  
static String valueOf(double)
```

String

▷ 4. 替换

```
String replace(oldchar,newchar);
```

▷ 5. 切割

```
String substring(begin);  
String substring(begin,end);
```

▷ 6. 子串，获取字符串中的一部分。

```
String[] split(regex);
```

String

▷ 7.转换：去除空格，比较。

▷ 7.1将字符串转成大写或者小写。

```
String toUpperCase()  
String toLowerCase()
```

▷ 7.2将字符串两端的多个空格去除。

```
String trim()
```

▷ 7.3对两个字符串进行自然顺序的比较。

```
int compareTo(string)
```


StringBuffer

▷ StringBuffer是字符串缓冲区。是一个容器。

▷ 特点：

▷ 1.而且长度是可变化的

▷ 2.可以字节操作多个数据类型。

▷ 3.最终会通过toString方法变成字符串。

StringBuffer

▷ 1.存储

`StringBuffer append()` 将指定参数添加到已有数据的结尾处。

`StringBuffer insert(index,数据)` 可以将数据插入到指定index位置

▷ 2.删除

`StringBuffer delete(start,end)` 删除缓冲区中的数据，包含start，不包含end

`StringBuffer deleteCharAt(index)` 删除指定位置的字符

StringBuffer

▷ 3.获取

```
char charAt(int index)
int indexOf(String str)
int lastIndexOf(String str)
int length()
String substring(int start,int end)
```

▷ 4.修改

```
StringBuffer replace(start,end,string);
void setCharAt(int index,char ch);
```

StringBuilder

- ▷ JDK1.5之后出现了StringBuilder
- ▷ StringBuffer是线程同步。
- ▷ StringBuilder是线程不同步。和StringBuffer使用方法相同。
- ▷ 开发建议使用StringBuilder提高效率。

THE END