# 问：字符串比较中“A==B”和A.equals(B)两种比较的区别？

“==”是java提供的关系运算符，主要功能是进行数值的相等判断，如果用在了String对象上，表示的是内存地址的比较。

equals：是由String类提供的一个方法，主要供字符串内容的比较。

## 数组的操作：

### 数组的拷贝：

System.arraycopy(源数组名称，源数组拷贝开始索引，目标数组名称，目标数组拷贝开始索引，长度)；//注意数组开始的索引是0

### 数组的排序：

格式： java.util.Arrays.sort(数组); 即可完成从小到大的排序。

# 对象数组（important）：

就是引用数据类型中包含了多个引用数据类型。

## 格式：

动态初始化：类名称 对象数组名称[]=new 类名称[长度]；

分步完成：类名称 对象数组名称[]=null; //开辟之后都是null值。

类名称 对象数组名称[]=new 类名称[长度]；

静态初始化：类名称 对象数组名称[]=new 类名称[]{实例化对象，实例化对象…}

**对象数组中的每个对象都需要分别实例化，对象数组一般定义为一维。数组有一个天生短板，不可改变数组长度。**

**对于笔试题，排序问题，不要写 java.util.Arrays.sort(数组名);**

# String类的常用方法：

1. **、new的入池，但是有手工入池的方法eg: String str=new String(“Hello”).intern();**

**方法：A.intern()**

## 转换：

|  |  |  |  |
| --- | --- | --- | --- |
| **NO:** | **方法名称** | **类型** | **功能即说明** |
| **1** | **public String(char[] value)** | **构造** | **将字符数组变为String类对象** |
| **2** | **public String(char[] value,int offset,int count)** | **构造** | **将部分字符类数组变为String** |
| **3** | **public char[] charAt(int index)** | **普通** | **返回指定索引对应的字符信息（index表示第几个，记住数组是从0开始的）** |
| **4** | **public char toCharArray()** | **普通** | **将字符串以字符数组的形式返回，输出的时候按照字符串数组用for循环输出。格式:A.toCharArray();** |
|  |  |  |  |

**字符大小写的转换，小的转大的-32即可。**

判断是否全为字符的时候，可以转换为

# 在定义boolean方法的时候，建议定义为isXxx这样是一个好习惯！

## 转换：

|  |  |  |  |
| --- | --- | --- | --- |
| NO: | 方法 | 类型 | 功能及说明 |
| 1 | public String(byte[] bytes) | 构造 | 将全部字节数组变为字符串 |
| 2 | public String(byte[] bytes,int offset,int length) | 构造 | 将部分字节变为字符串 |
| 3 | public byte[] getBytes() | 普通 | 将字符串变为字节数组 |
| 4 | public byte[] getBytes(String charsetName)throws UnspportedEncodingException | 普通 | 进行编码转换 |

## 区别：

|  |  |  |  |
| --- | --- | --- | --- |
| NO： | 方法 | 类型 | 功能及说明 |
| 1 | public Boolean equals(String anObject) | 普通 | 进行相等判断，区分大小写 |
| 2 | public boolean equalsIgnoreCase(String anotherString) | 普通 | 进行相等的判断不区分大小写 |
| 3 | public int compare To(String anotherString) | 普通 | 判断两个字符串的大小（按编码比较），有三种结果：   1. =0，表示要比较的两个字符串内容相等 2. >0，表示大于的部分 3. <0，表示小于的部分 |

**格式：A·方法（B）;**

## 查找：

|  |  |  |  |
| --- | --- | --- | --- |
| NO： | 方法 | 类型 | 功能及说明 |
| 1 | public boolean countains(String s) | 普通 | 判断指定内容是否存在 |
| 2 | public indexOf(String str)  或者用A.contains(B)直接返回boolean型。（contains成为了查找的代名词。） | 普通 | 由前向后查找指定字符串的位置，如果查到了则返回（第一个字母）的索引，如果找不到则返回-1。 |
| 3 | public indexOf(String str,int formIndex) | 普通 | 由指定位置从前向后查找指定字符串的位置，找不到返回-1. |
| 4 | public int lastIndexOf(String str) | 普通 | 由后向前查找指定字符串的位置，zhaobudao 返回-1. |
| 5 | public int lastIndexOf(String str,int formIndex) | 普通 | 从指定的位置由后向前查找字符串的位置，找不到返回-1. |
| 6 | public boolean startsWith(String prefix) | 普通 | 判断是否以指定的字符串开头 |
| 7 | public boolean startsWith(String prefix,int toffset) | 普通 | 从指定的位置开始判断是否以指定的字符串开头 |
| 8 | public Boolean endsWith(String suffix) | 普通 | 判断是否以指定的字符串结尾。 |

## 替换：

|  |  |  |  |
| --- | --- | --- | --- |
| NO： | 方法 | 类型 | 功能及说明 |
| 1 | **Public String replaceAll(String regex, String replacement)** | 普通 | 用新的字符串替换全部 |
| 2 | public String replaceFirst(String regex, String replacement) | 普通 | 替换首字母 |

## 字符串截取：

|  |  |  |  |
| --- | --- | --- | --- |
| No: | 方法 | 类型 | 功能及说明 |
| 1 | public String substring(int beginIndex) | 普通 | 从指定索引截取到结尾 |
| 2 | public String substring(int beginIndex,int endIndex) | 普通 | 从指定索引截取到指定索引 |

数据库中的函数可考虑到非专业人士的使用，但是考虑到函数的严谨性，是不可以用负数来作为截取的开始点的。

## 拆分：

|  |  |  |  |
| --- | --- | --- | --- |
| No: | 方法 | 类型 | 功能及说明 |
| 1 | public String[] split(String regex) | 普通 | 按指定的字符串拆分若（“”）逐个拆分，记得拆出来的是数组。 |
| 2 | public String[] split(String regex,  int limit) | 普通 | 按指定的字符串进行部分拆分，最后由数组的长度也就是limt就决定（如果能拆的结果很多，数组的长度才会由limt决定），即前面拆，后面不拆，（2起步） |

有时候按敏感字符（即正则标记）拆的时候，是没结果的，所以要加上转义字符[\\加上原来的标记，即可拆分](file:///\\加上原来的标记，即可拆分)。

|  |  |  |  |
| --- | --- | --- | --- |
| No: | 方法 | 类型 | 功能及说明 |
| 1 | public String concat(String str) | 普通 | 相当于“+”，连接字符串，和＋不同，一定要注意区分 |
| 2 | public String toLowerCase（） | 普通 | 转小写，除字母外的字符都不会转换 |
| 3 | public String toUpperCase（Locale locale） | 普通 | 转大写，除字母外的字符都不会转换 |
| 4 | public String trim（） | 普通 | 将字符串头尾的空格删除，中间的空格是不会删除的，防止接受数据时无用的空格。 |
| 5 | public int length（） | 普通 | 取得字符串长度 |
| 6 | public String intern（） | 普通 | 数据入池 |
| 7 | public boolean isEmpty（） | 普通 | 判断是否是空字符串（不是null，是””,length是0），如果A.isEmpty(B)不方便，可以用A.equals(B); |

## 其他方法：

有些情况下，用户输入的数据长度是有限制的，可以利用此方式判断。数组中也有一个length属性。但是调用的形式不同

**·数组对象.length; //在这里是没有括号的**

**·String 对象.length() //在这里是有括号的，原因是：String是一个类**