API(Application Programming Interface),应用程序编程接口。Java API是一本程序员的字典，是JDK中提供给我们使用的类的说明文档。（说白了，API就是好多的类和好多的接口）

1.Scanner类

引用类型的一般使用步骤：

1、导包

import 包路径.类名称;

如果需要使用的目标类，和当前类位于同一个包下，则可以省略导包语句不写。

**只有java.lang**包下的内容不需要导包，其他的包都需要import语句。

2、创建

类名称 对象名 = new 类名称（）；

3、使用

对象名.成员方法名（）；

Scanner sc = new Scanner(System.in);//System.in表示从键盘进行输入。

获取键盘输入的一个int数字： int num = sc.nextInt();

获取键盘输入的一个字符串： String str = sc.next();

2.匿名对象

1、创建对象的标准格式：

类名称 对象名 = new 类名称（）；

匿名对象就是只有右边的对象，没有左边的名字和赋值运算符。

2、格式：**new 类名称（）；**

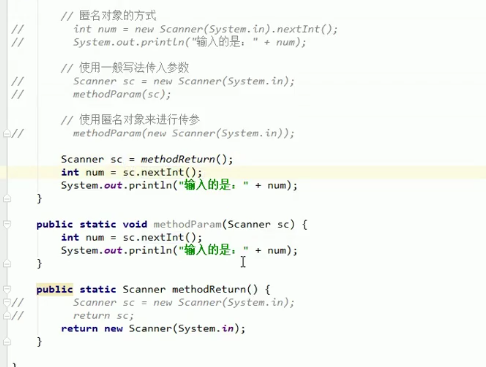
**（**每次new关键字一出现，就是新的对象被创建了。**）**

注意事项：匿名对象只能使用唯一的一次，下次使用不得不再创建一个新对象。

使用建议：如果确定有一个对象只需要使用唯一的一次，就可以使用匿名对象。

**【任何数据类型都能作为参数和返回值】**

3、匿名对象作为参数和返回值



3.Random类

1、Random类用来生成随机数字。

2、Random r = new Random();

获取一个随机的int数字：

1、int num = r.nextInt（）//num的范围是int所有范围，有正负两种

2、int num = r.nextInt（x）//num的访问是**[0,x）( 左闭右开区间)**

**注意：此处的x必须大于0**

4.ArrayList集合

1、对象数组

普通数组（一维）的定义：int[] array = new int[**x**];(其中x必须赋值)（没有括号，分号结尾）

定义一个**对象数组**，用来存储3个Person对象:Person[] person = new Person[3];

而**对象的创建**是这样的 Person one = new Person();(括号是因为调用的构造函数)

**2、数组有一个缺点，一旦创建，程序运行期间长度不可以发生改变。**

3、**数组**的长度不可以发生改变，但是ArrayList**集合**的长度是可以随意变化的。

4、对于ArrayList来说，有一个<E>代表**泛型**。

泛型：也就是装在集合当中的所有元素，全都是统一的类型。

注意：泛型**只能是引用类型**，不能是基本类型。

/\*举例子：ArrayList<String> list = new ArrayList<>();\*/

/\*代表创建了一个ArrayList集合，集合的名称是list，里面装的都是String类型的数据\*/

/\*备注：从JDK1.7开始，右侧<>内部可以不写内容，但是<>本身还是要写的\*/

注意事项：

对于**ArrayList集合**来说，直接打印得到的**不是地址值**，**而是内容**。如果内容是空，得到的是空的中括号。**[]** （重点）

ArrayList集合不为空举例：[赵丽颖，迪丽热巴，古力娜扎]

5、ArrayList当中的常用方法（**必背**）

**1.public boolean add(E e):向集合中添加元素，参数的类型和泛型一致.返回值代表添加是否成功。**

**备注：对于Arra集合来说，add添加动作一定是成功的，所以返回值可用可不用。但是对于其他集合来说，add添加动作不一定成功。**

**2.public E get(int index):从集合当中获取元素，参数是索引编号，返回值就是对应位置的元素。**

**3.public E remove(int index):从集合当中删除元素，参数是索引编号，返回值就是被删掉的元素。**

**4.public int size（）：获取集合的尺寸长度，返回值是集合中包含的元素个数。**

**5.数组和集合遍历的区别及注意事项**



6.集合里面的泛型是引用类型，不能是基本类型。为什么？

其实集合里面保存的都是地址值，但是基本类型保存的都是数据，没有地址值，所以集合里面存储基本数据类型是不行的。但是还是有办法存储基本数据类型的数据的，用基本类型的包装类即可。

基本类型 包装类（**引用类型**，包装类都位于java.lang包下）

byte Byte

short Short

int Integer 【特殊】

long Long

float Float

double Double

char Character 【特殊】

boolean Boolean

备注：从JDK1.5+开始，支持自动装箱，自动拆箱。

自动装箱：基本类型 ---》包装类型

自动拆箱：包装类型 ---》基本类型

5.String类型（字符串）

1、Java程序中的所有字符串字面值（如“abc”）都作为此类的实例实现。

其实就是说，程序当中所有的双引号字符串，都是String类的对象。（就算没有new，也是）

2、字符串的特点

1、字符串的内容永不可变。【**字符串是常量，他们的值在创建之后不能更改。**】【重点】

2、正是因为字符串不可改变，所以字符串是可以共享使用的。

3、字符串效果上相当于是char[]字符数组，但是**底层原理是byte[]字节数组**。

3、创建字符串的常见3+1种方式

三种构造方法：

1、public String():创建一个空白字符串，不含有任何内容。

2、public String （char[] array）:根据字符数组的内容，来创建对应的字符串

3、public String（byte[] array）：根据字节数组的内容，来创建对应的字符串。

一种直接创建：

String str = “hello”;//右边直接用双引号

注意：直接写上双引号，就是字符串对象.（就算没new，也是对象，其实是JVM帮我们new了）



4、字符串的常量池

/\*程序当中直接写上的双引号字符串，就在字符串常量池中。也就是说只有直接写双引号的字符串，才在字符串常量池中\*/

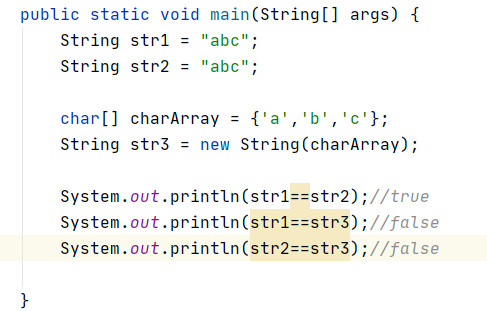
/\*

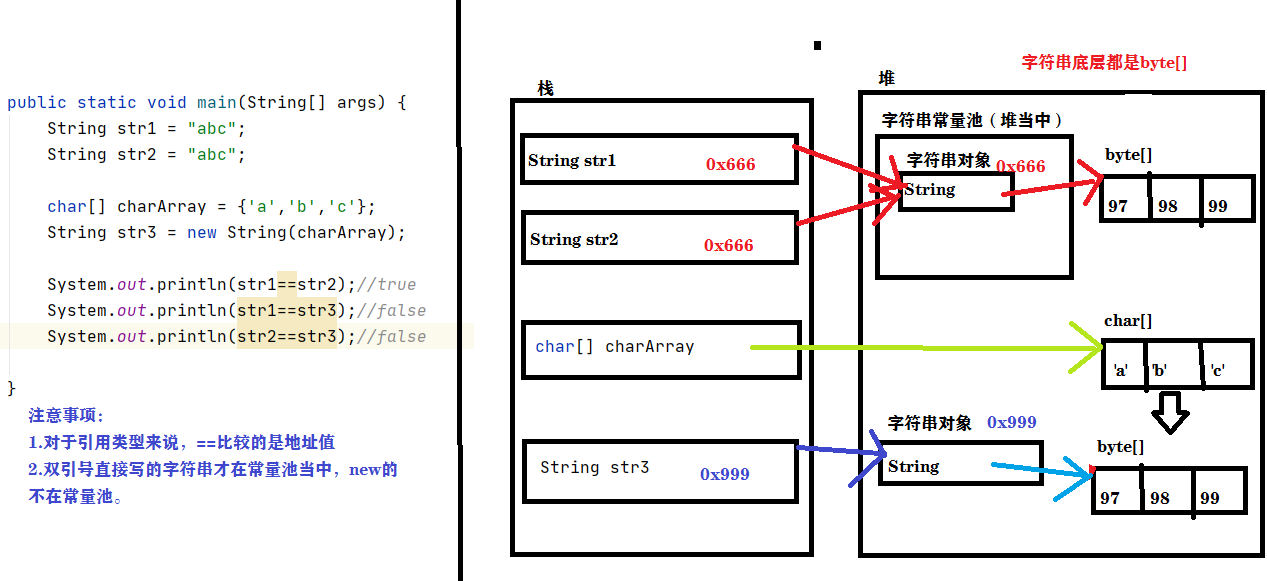
对于**基本类型**来说， ==是进行**数值**的比较。

对于【引用类型】来说， ==是进行【地址值】的比较。

\*/







4、字符串中常用的方法（必背）

1、字符的比较相关方法

==是进行对象的地址值比较（引用类型），如果确实**需要字符串的内容比较**，可以使用两个方法：

**①public boolean equals(Object obj):**参数可以是任何对象，**只有参数是一个字符串并且内容相同的才会給true**，否则返回false。

注意事项：

1、任何对象都能用Object进行接收。

2、equals方法具有对称性，也就是a.equals(b)和b.equals(a)效果一样

3、如果比较双方一个常量一个变量，推荐把常量字符串写在前面。

推荐：“abc”.euqals(str1)【有经验的程序员都这么写】 不推荐：str1.equals(“abc”)

为什么呢？如果String str1 = null;不推荐的那个写法会报空指针异常(NullPointException),所以不推荐。

**②public boolean equalsIgnoreCase(String str):**忽略英文字母大小写，进行内容比较。



2、String当中与获取相关的常用方法：

**①public int length()**:获取字符串当中含有的字符串个数，拿到字符串长度。**【length()方法】**

**②public String concat(String str)**:将当前字符串与参数字符串拼接成为返回值新的字符串。

**③public char charAt(int index)**:获取指定索引位置的单个字符。（索引从0开始）

**④public int indexOf(String str):**查找参数字符串在本字符串当中首次出现的索引位置，如果没有返回-1。（索引从0开始，没找着就是-1）

**小备注：（获取长度或者叫元素个数）**

**数组**（如int[] array = new int[3]）获取长度,**用属性length**。

**集合**（ArrayList<E> list = new ArrayList<>();）获取长度，**用方法size（）**。

**字符串**（String str = “hello”）获取长度，**用方法length（）**。

3.String中的截取方法

**①public String subString（int index）：**截取从参数位置一直到字符串末尾，返回新字符串。

**②public String subString（int begin，int end）**：截取从begin开始，一直到end结束，中间的字符串。

备注：[begin，end）左边右开

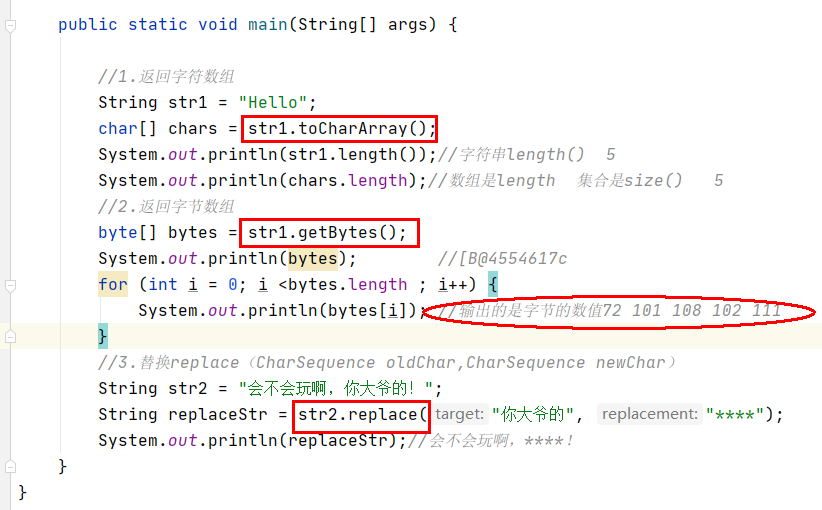
4.String（字符串）中的转换方法

**①public char[] toCharArray（）：**将当前字符串拆分为**字符数组**作为返回值。

**②public byte[] getBytes()**：获得当前字符串底层的**字节数组**。（学字节流的时候有用）

**③public Sting replace（CharSequence oldString,CharSequence newString）**：将所有出现的老字符串成为新的字符串，返回替换之后的结果新字符串。

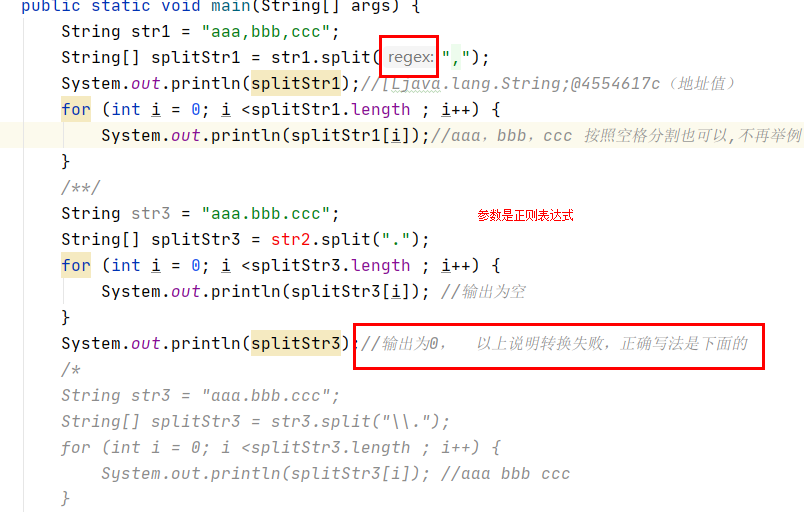
**备注**：CharSequence意思是可以接受字符串类型（可以把它当做是String）



5.String（字符串）中的分割方法

**①public String[] split（String regex）：**按照参数的规则，将字符串切分成为若干部分。

**注意事项：**这个参数其实是个正则表达式。今天要注意，如果按照英文据点“.”进行切分，必须写” \\. ” (两个反斜杠)。



5、数组工具类Arrays

Java.util.Arrays是一个与数组相关的工具类，里面提供了大量静态方法，用来实现数组常见的操作。

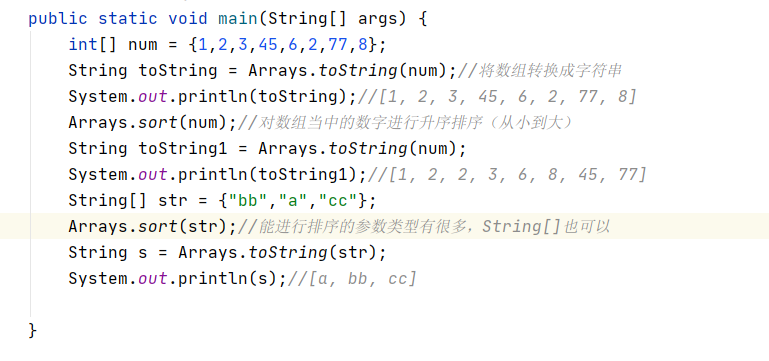
**public static String toString（数组）;**将**参数数组**变成**字符串**（按照默认格式：[元素1，元素2，元素3,…]）

**public static void sort（数组）;**按照默认升序（从小到大）对数组的元素进行排序。

备注：1、如果是数值，sort默认按照升序从小到大

2、如果是字符串，sort默认按照字母升序。

3、如果是自定义的类型，那么这个自定义的类要有Comparable或者Comparator接口的支持。



5、数学工具类Math

Java.util.Math类是数学相关的工具类，里面提供了大量的静态方法，完成与数学相关的运算操作。

**public static doubule abs(double num)**；获取绝对值。（获取绝对值的返回值可以是int float double long）

**public static double ceil(double num)**;向上取整。（返回值类型只有double）

**public static double floor（double num）**;向下取整。 （floor地板的意思）（和ceil一样）

**public static long round(double num):**四舍五入。(四舍五入的结果不带小数点)（返回值类型有double和float，但是不带小数点）