java常用函数（持续更新）

把**char**类型的数组转化为**String**类型

```
char a[]={'s','d','f'};
String str1 = new String(a);//把char类型的数组转化为String类型;
需要注意的是，如果数组的长度超过了数组中数据的长度转化为String后末尾会有空格
也就是说转化为String类型后的长度和数组的长度一样;
```

把**String**类型的变量**str**转化为**int**类型

```
String str="12345678";
int n=Integer.parseInt(str);把String类型的变量str转化为int类型;
需要注意的是，str中只能含有数字，并且不能有空格;
```

把**char**类型的数组转换成**String**类型的字符串;

```
 此方法可以把int类型的数组转化成String的字符串;
 char a[]=new char[10];
 int y=0;
 for(int i=70;i<80;i++)
 {
  a[y++]=(char)i;
 }
 System.out.println(Arrays.toString(a));
 输出样例：`
 [F, G, H, I, J, K, L, M, N, O]`
 此方法不可以把int类型的数组转化成String的字符串;
 char a[]=new char[10];
 int y=0;
 for(int i=70;i<80;i++)
 {
  a[y++]=(char)i;
 }
 System.out.println(String.valueOf(a));
 输出样例：`
 FGHIJKLMNO
```

以某字符为标志分割**String**字符串

```
String s="dgjcvhjjh@ertyuj@wghjkl"
String k[]=s.split("@");以@为标志分割String字符串;
把各部分分别放进看k[0],k[1],k[2].....中;
String s="dgjcvhjjh^erty^uj^wghjkl"
String k[]=s.split("\\^");
注意：如果分割符是特殊符号，比如'^','+','-','/'等，需要加转义符\\;
String s="dgjc^vhjjh^erty^uj^wghjkl"
String k[]=s.split("\\^",2);
规定分割前几个代码;
```

把所有的某字符转换成另一个字符

```
String s="ghjkdfg@ghjh@fdfghj";
s.replaceAll("@,"$");//把所有的@转换成$;
注意：特殊符号要加"\\"进行转义;
```

进制转换

```
Integer.toBinaryString(n);把十进制转换成2进制;
Integer.toHexString(n);把十进制转换成16进制;
Integer.toOctalString(n);把十进制转换成8进制;
Integer.toString(n, m);把十进制转换成m进制;
```

返回值是布尔类型的函数

```
Character.isDigit(ch);判断是否为数字
Character.isJavaIdentifierPart(ch);判断是否为java合法标识符的非首位;
Character.isJavaIdentifierStart(ch);判断是否为java合法标识符的首位;
Character.isLowerCase(ch);判断是否为小写字母;
Character.isUpperCase(ch);判断是否为大写字母;
```

字母大小写转换

```
Character.toLowerCase(ch);大写转小写
Character.toUpperCase(ch);小写转大写
```

找子串

```
/*Java中字符串中子串的查找共有四种方法，如下：*/
1、int indexOf(String str)：
返回第一次出现的指定子字符串在此字符串中的索引。
2、int indexOf(String str, int startIndex):
从指定的索引处开始，返回第一次出现的指定子字符串在此字符串中的索引。
3、int lastIndexOf(String str)：
返回在此字符串中最右边出现的指定子字符串的索引。
4、int lastIndexOf(String str, int startIndex)：
从指定的索引处开始向后搜索，返回在此字符串中最后一次出现的指定子字符串的索引。
```

检测字符串是否以指定的前缀开始

```
public boolean startsWith(String prefix, int toffset)
或
public boolean startsWith(String prefix)
prefix -- 前缀。
toffset -- 字符串中开始查找的位置。
```

```
如果字符串以指定的前缀开始，则返回 true；否则返回 false。

public class Test {

public static void main(String args[]) {

   String Str = new String("www.runoob.com");

   System.out.print("返回值 :" );

   System.out.println(Str.startsWith("www") );

   System.out.print("返回值 :" );

   System.out.println(Str.startsWith("runoob") );

   System.out.print("返回值 :" );

   System.out.println(Str.startsWith("runoob", 4) );
}}

以上程序执行结果为:
返回值 :true
返回值 :false
返回值 :true
```

截取**String**字符串中的一段字符串

ubstring——从某处到最后

```
String s = "Hello";
String  ss = s.substring(int beginIndex)
返回一个新的字符串，它是此字符串的一个子字符串。
该子字符串始于指定索引处的字符，一直到此字符串末尾。

例如:
"unhappy".substring(2) returns "happy"
"Harbison".substring(3) returns "bison"
"emptiness".substring(9) returns "" (an empty string)

参数:
beginIndex - 开始处的索引（包括开始处的索引位置的字符）
```

substring——从某处到某处

```
public String substring(int beginIndex, int endIndex)
返回一个新字符串，它是此字符串的一个子字符串。
该子字符串从指定的beginIndex 处开始,到指定的 endIndex-1处结束。

示例:
"hamburger".substring(3,8) returns "burge"
"smiles".substring(0,5) returns "smile"

参数:
beginIndex - 开始处的索引（包括开始处的索引位置的字符）。
endindex 结尾处索引（不包括结尾处的索引位置的字符）。
```