* isEmpty(String str) 是否为空，空格字符为false
* isNotEmpty(String str) 是否为非空，空格字符为true
* isBlank(String str) 是否为空，空格字符为true
* isNotBlank(String str) 是否为非空，空格字符为false
* trim(String str)去除字符串两端的控制符，空字符串、null 返回 null
* trimToEmpty(String str) 去除字符串两端的控制符，空字符串、null 返回""
* stripToNull(String str) 去除字符串两端的空白符，空字符串、null 返回null
* stripToEmpty(String str) 去除字符串两端的空白符，空字符串、null 返回""
* strip(String str, String stripChars) 去掉str两端的在stripChars中的字符

StringUtils.strip("000000134\_76539000","0")="134\_76539"

* stripStart (String str,String stripChars) 去除str 前端在stripChars中的字符
* stripEnd (String str,String stripChars) 去除str 后端在stripChars中的字符
* equals(String str1,String str2) 比较两个字符串是否相等，如果两个均为空则认为相等
* indexOf(String str,char searchChar) 返回searchChar 在字符串中第一次出现的位置，如果没找到则返回 -1，如果str 为null 或者 "",也返回-1
* indexOf(String str,char searchChar,int startPos) 返回字符searchChar从startPos开始在字符串str中第一次出现的位置。
* contains(String str,char searchChar) str中是否包含字符searchChar，str为null 或者 searchChar为null,返回false 。

StringUtils.contains("", "")  = true

StringUtils.contains("dfg", "")  = true

* containsIgnoreCase(String str,String searchStr) str中是否包含字符searchChar，不区分大小写
* int indexOfAny(String str, char[] searchChars) 找出字符数组searchChars中的字符第一次出现在字符串str中的位置。 如果字符数组中的字符都不在字符串中，则返回-1 ，如果字符串为null或""，则返回-1
* subString(String str,int start) 从start 开始，包含start 那个字符，得到字符串str 的子串，如果start为负数，则从后面开始数起。如果str 为null 或者 "" 则返回其本身
* subStringBefore(String str,String separator) 得到字符串separator第一次出现前的子串。不包含那个字符，如果str 为null 或者 "" 则返回其本身。
* subStringAfter(String str,String separator) 得到字符串separator第一次出现后的子串，不包含那个字符，如果 str 为null，或者"",则返回其本身
* subString(String str,int start,int end) 同上
* left(String str,int len) 得到字符串str从左边数len长度的子串，如果str 为null 或者 "",则返回其本身，如果len小于0，则返回""
* right(String str,int len)得到字符串str从右边数len长度的子串
* mid(String str,int pos,int len) 得到字符串str从pos开始len长度的子串，pos小于0，则设为0。
* split(String str) 把字符串拆分成一个字符串数组，用空白符 作为分隔符，字符串为null 返回null，字符串为"",返回空数组{}
* split(String str,char c) 按照 char c 拆分字符串
* join(Object[] arrey)把数组中的元素连接成一个字符串返回
* join(Object[] arrey,char c) 把数组中的元素拼接成一个字符串返回，把分隔符 c 也带上
* deleteWhitespace(String str) 删除字符串中的所有空白符，包括转义字符
* removeStart(String str,String remove) 如果字符串str是以remove开始，则去掉这个开始，然后返回，否则返回原来的串
* removeEnd(String str,String remove) 如果字符串str是以字符串remove结尾，则去掉这个结尾，然后返回，否则返回原来的串。
* remove(String str,char remove) 去掉字符串str中所有包含remove的部分，然后返回
* replace(String str,String reql,String with) 在字符串text中用with代替repl，替换所有
* replaceChars(String str,char old,char new) 在字符串中 new 字符代替 old 字符
* replaceChars(String str, String searchChars, String replaceChars) 这个有点特别，先看下面三个例子

StringUtils.replaceChars("asssdf","s","yyy"))    =    "ayyydf"

StringUtils.replaceChars("asdf","sd","y"))        = "ayf"

StringUtils.replaceChars("assssddddf","sd","y"))= "ayyyyf"

解释:为什么会出现上面这样的结果呢？原来这个置换规则是这样的，他是拿searchChars的index，去replaceChars找相应的index然后替换掉，怎么说呢？比如说第一个例子 s 的index 是0，找到yyy相对应的index为0的字符是y。第二个例子 's' 的index是0，'d'的index是1, 字符's' 可以找到对应index为0的 'y'，d就找不到index为'1'的的字符了，所以就直接过滤掉了，听明白了吗？

* overlay(String str,String new,int start,int end) 用字符串new 覆盖字符串str从start 到 end 之间的串
* chop(String str) 去掉字符串的最后一个字符,比如/r/n
* repeat(String str,int repart) 重复字符串repeat次
* rightPad(String str,int size,String padStr) size长度的字符串，如果不够用padStr补齐
* leftPad(String str,int size,String padStr)同上
* center(String str,int size)产生一个字符串，长度等于size，str位于新串的中心
* swapCase(String str) 字符串中的大写转小写，小写转换为大写