arduino中String的用法全解

string是c语言中自带的变量类型，等同于char[]。
String是需要使用扩展库实现的变量类型，可以相对方便的操作。

String的+操作，也可以使用+=：

```
stringThree =  stringOne + 123;// adding a constant integer to a string
stringThree = stringOne + 123456789;// adding a constant long integer to a string
stringThree =  stringOne + 'A';// adding a constant character to a string:
stringThree =  stringOne +  "abc";// adding a constant string to a string:
stringThree = stringOne + stringTwo; // adding two Strings together:
stringThree = stringOne + millis();
stringThree = stringOne + analogRead(A0);
```

charAt() and setCharAt() 的用法：
The String functions charAt() and setCharAt() are used to get or set the value of a character at a given position in a String.

At their simplest, these functions help you search and replace a given character. For example, the following replaces the colon in a given String with an equals sign:

```
String reportString = "SensorReading: 456";
  int colonPosition = reportString.indexOf(':');
  reportString.setCharAt(colonPosition, '=');
```

toUpperCase() and toLowerCase()

```
stringOne.toUpperCase();
```

比较操作符：
The String comparison operators ==, !=,>, < ,>=, <= , and the equals() and equalsIgnoreCase() methods allow you to make alphabetic comparisons between Strings. They’ re useful for sorting and alphabetizing, among other things.

The operator == and the method equals() perform identically. In other words,

if (stringOne.equals(stringTwo)) {

is identical to

if (stringOne ==stringTwo) {

The “>” (greater than) and “<” (less than) operators evaluate strings in alphabetical order, on the first character where the two differ. So, for example “a” < “b” and “1” < “2”, but “999” > “1000” because 9 comes after 1.

Caution: String comparison operators can be confusing when you’ re comparing numeric strings, because the numbers are treated as strings and not as numbers. If you need to compare numbers, compare them as ints, floats, or longs, and not as Strings.

String()的用法，堪称神器。

```
String stringOne = "Hello String";                   // using a constant String
String stringOne = String('a');                      // converting a constant char into a String
String stringTwo = String("This is a string");       // converting a constant string into a String object
String stringOne = String(stringTwo + " with more"); // concatenating two strings
String stringOne = String(13);                       // using a constant integer
String stringOne = String(analogRead(0), DEC);       // using an int and a base
String stringOne = String(45, HEX);                  // using an int and a base (hexadecimal)
String stringOne = String(255, BIN);                 // using an int and a base (binary)
String stringOne = String(millis(), DEC);            // using a long and a base
String stringOne = String(5.698, 3);                 // using a float and the decimal places
```

indexOf() and lastIndexOf(), length() and trim() ,toInt(), substring(),startsWith() and endsWith(),replace()
以下为转载:
此库中包含
1 charAT()
2 compareTo()
3 concat()
4 endsWith()
5 equals()
6 equalsIgnoreCase()
7 getBytes()
8 indexOf()
9 lastIndexOf()
10 length()
11 replace()
12 setCharAt()
13 startsWith()
14 substring()
15 toCharArray()
16 toInt()
17 toLowerCase()
18 toUpperCase()
19 trim()
charAT(n)

描述
获取字符串的第n个字符

参数
n： 是一个索引，作为下标
str1.compareTo(str2)

描述
compareTo函数是比较两个字符串，相同返回两个字符串当前比较字符串的差值。前-后

参数
str1： 第一个字符串
str2： 第二个字符串

返回
compareTo()的返回值是整型,它是先比较对应字符的大小(ASCII码顺序),如果第一个字符和参数的第一个字符不等,结束比较,返回他们之间的差值,如果第一个字符和参数的第一个字符相等,则以第二个字符和参数的第二个字符做比较,以此类推,直至比较的字符或被比较的字符有一方全比较完,这时就比较字符的长度
str1.concat(str2)

描述
字符串拼接，其实在C++语言中字符串拼接可以直接让前字符串+后字符串

参数
str1： 第一个字符串
str2： 第二个字符串

返回
无返回值，拼接好的字符串在str1中
str1.endsWith(str2)

描述
字符串尾部判断对比，判断str1尾部是否是字符串str2，当然你也可以用它来判断’ \n’

参数
str1 ： 第一个字符串
str2 ： 第二个字符串

返回
布尔类型
str1.equals(str2)

描述
判断字符串是否相等

参数
str1 ： 第一个字符串
str2 ： 第二个字符串

返回值
布尔类型
str1.equalsIgnoreCase(str2)

描述
判断字符串是否相等，忽略大小写
str1.getBytes(buf,len)

描述
字符串的复制，和和函数 toCharArray()功能非常相识。

参数
string1：原本的字符串
buf：要搬移的目的变量
len：字符串长度
indexOf()

描述
在字符串中选择特定的字符，并返回位置的功能函数(正向)。如果你想在一个很长的字符串中查找这个特别的关键字，可以使用这个函数。
string1.indexOf(val);
string1.indexOf(val,from);

参数
string1：原本的字符串
val：想要找的关键字，可以是char或 string 字符串。
from：选择性参数，你可以特别指定从那个位置开始寻找这个关键字。

返回值
成功返回位置，失败返回-1
lastIndexOf()

描述
和indexOf一样，只不过是反向查找
str1.length()

描述
测量字符串长度

参数
str1：被测字符串变量

返回值
字符串长度
replace()

描述
字符串替换，string1.replace(string2,string3)

参数
string1：原本的字符串。
string2：在字符串中欲被替换的字符串。
string3：要替换之后的新字符串。

返回值
无
setCharAt()

描述
字符替换，string1.setCharAt(i,charl)

参数
string1：原本的字符串。
i：字符串中欲被换掉的字符的位置。
charl：要替换的字符，注意只有一个字符，而不是字符串。

返回值
无
startsWith()

描述
判断字符串是否已某个特殊的字符串开始的，string1.startsWith(string2)

参数
string1：原本的字符串。
string2：判断是不是已这个字符串开始。

返回值
布尔代数，true 和 false
substring()

描述
用来截取字符串中间的某一位置。另外是决定那里结束，
string1.substring(from); //返回 from 之后的
string1.substring(from,to); //返回 from 和 to 之间的

参数
string1：原本的字符串
from：要截取的字符串从第几个字开始。
to：要截取后面的第几个字

返回值
字符串
toCharArray()

描述
把string拷贝char中，toCharArray函数是字符串处理中常用的一个函数，你可以把他当成string转char[]的转换函数，或者复制文字的函数，他的功能和getBytes()非常相似，toCharArray()函数转换后是呈现文字，而getBytes()函数转换后是呈现数字。
string1.toCharArray(buf,len);

参数
string：原本的字符串。
buf：指定的char[]的位置，注意char[]的空间，一定要等于或大于复制的大小，不然存储器和程序都会产生不可预期的问题。
len：要复制的字符串长度。

返回值
要复制的字符串长度
toInt()

描述
字符串转成int
string1.toInt();

参数
string1：字符串，如"123"

返回值
整数，如 123
toLowerCase()

描述
把英文全部转换成小写

使用方式
str1.toLowerCase();

返回值
无
toUpperCase()

描述
把英文全部转换成大写

使用方式
str1.toUpperCase();

返回值
无
trim()

描述
自动清除字符串中最前面和最后面的空白数据。
string1.trim();

参数
string1：原本的字符串

返回值
无