

# Arduino Data types

## String()

### 描述

构造String类的实例。有多个版本可以从不同的数据类型构造字符串(即将它们格式化为字符序列)，包括：

- 用双引号括起来的常量字符串(即char数组)
- 单引号内的一个常量字符
- String对象的另一个实例
- 常数整数或长整数
- 使用指定基数的常数整数或长整数
- 整型或长整型变量
- 使用指定基数的整型或长整型变量
- 使用指定小数点的浮点数或双精度数

从数字构造String将得到一个包含该数字的ASCII表示形式的字符串。默认是十进制，所以

```
1 String thisString = String(13);
```

输出字符串“13”。不过，你也可以使用其他进制。例如，

```
1 String thisString = String(13, HEX);
```

输出字符串“d”，它是十进制值13的十六进制表示。如果你喜欢二进制，

```
1 String thisString = String(13, BIN);
```

输出了字符串“1101”，它是13的二进制表示。

### 语法

```
1 String(val);
2 String(val, base);
3 String(val, decimalPlaces);
```

### 参数

val : 要格式化为String类型的变量。允许的数据类型:string, char, byte, int, long, unsigned int, unsigned long, float, doublebase。

base : (可选)格式化整数值的进制表达。

decimalPlaces : **仅当val是float或double时**。小数点后几位。

## 返回值

String类的一个实例

## 例程

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

## Reference List

1. [String\(\) - Arduino Reference](#)