**[String Formatting](https://github.com/mbostock/d3/wiki/Formatting)**

[Wiki](https://github.com/mbostock/d3/wiki/Home) ▸ [API Reference](https://github.com/mbostock/d3/wiki/API-Reference) ▸ [Core](https://github.com/mbostock/d3/wiki/Core) ▸ **String Formatting**

首先声明，咱是纯业余的，不过翻译错不了多少，就看你敢不敢用谷歌，哈哈哈！ 不矫情了，直接说重点，原文简单得很，是爷们儿就别怕！ 给你们看个数字，看好了哈，"0.30000000000000004"，想象一下，把这个数字放到坐标轴上，你自己都想吐吧，更何况给你的客户看？另外，你可能还需要对你的数字做一些千分位等的标记，让它更具可读性，如 "$1,240.10" 等。再然后，你可能只是想展示一些有意义的或特定的数……总之，想对你展示的数字很好的格式化，兄弟，用D3的这套标准 “**number format**”吧，TMD真的很简单！

举个简单的例子，创建一个用0填齐数字到4位的例子，意思是让你的数字总是满足最小四位，不够的，前面添零：



现在，你就可以调用 zero 来很方便的格式化你的数字了！



当然，除了数字，还有[日期](https://github.com/mbostock/d3/wiki/Time-Formatting) [dates](https://github.com/mbostock/d3/wiki/Time-Formatting)呀，[逗号分隔的一些字串comma-separated values](https://github.com/mbostock/d3/wiki/CSV)呀等等乱七八糟的东东，必须都能支持！

## 重点: Numbers

## [#](https://github.com/mbostock/d3/wiki/Namespaces" \l "wiki-prefix) d3.format (*specifier*)

将数字转化成指定格式的字符串。转化的格式非常丰富，且非常智能。指定格式，就是那个叫 specifier规范的东东（等同于适用默认的美国英语语言环境的[locale.numberFormat](https://github.com/mbostock/d3/wiki/Localization#wiki-locale_numberFormat) ） 格式化函数唯一的入参是数字，返回代表格式化数字的字符串。这个格式化规范模拟的是Python 3.1内置的格式化规范语言 [format specification mini-language](http://docs.python.org/release/3.1.3/library/string.html#formatspec). 长得样子是这样的:

[​[**fill**]**align**][**sign**][**symbol**][**0**][**width**][**,**][**.precision**][**type**]

这种命名法叫“[巴科斯范式](http://zh.wikipedia.org/wiki/%E5%B7%B4%E7%A7%91%E6%96%AF%E8%8C%83%E5%BC%8F)”，大部分编程语言的语法都用它的，必须懂哈！ 在这里，你只要知道，方括号( **[ ]** )内包含的为可选项就好了！

**fill** : 可以是任意字符，除了 “{“ 和 “}”，**fill** 必须是 **align** 的一个选项，往往都是跟着它的！

**align**: 三个字符 “**<**”, “**>**”[默认], “**^**”! 分别表示在可用的区域 **左**对齐，**右**对齐，**居中**！

**sign**: 也就三个，“+”，“-”[默认]，“ “，加号可以用于正数或负数，减号仅仅用于负数，前面的空格应该用在正数前面，而减号必须用在负数！(翻译的真垃圾)

**symbol**: 美元符号 “$”, 必须被用在每一种情况下，不管作为前缀还是后缀； “#”作为二进制，八进制，十六进制的输出，用 "0b", "0o", or "0x"来分别指定！

**0**: 表示是否激活补零功能！

**width**: 指定最小宽度，如果没有指定，将由具体内容决定!

**,** : 分号表示是否激活千分符分隔数字！

**.precision**: 精度, 你懂的! 表示使用“f”和“％”格式化的数值小数点后应该显示几位，或者“G”，“r”和“p”格式化的数值小数点前后应该显示几位!

**type**: 类型，地球人都懂！ (能不翻译么？)

其中精度的可选项有：

* exponent ("e") – 指数，参照 [Number.toExponential](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toExponential).
* general ("g") -  将定点或者指数方式调整至精确数字。参照[Number.toPrecision](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toPrecision).
* fixed ("f") – 指定小数点后数字位数，不足补0，参照 [Number.toFixed](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toFixed).
* integer ("d") – 数字， 参照[Number.toString](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toString)，但是忽略任何不是整数的值。
* rounded ("r") – 取整，类似fixed
* percentage ("%") - 类似fixed，但乘以100和后缀为“％”。
* rounded percentage ("p") - 类似rounded，但乘以100和后缀为“％”。
* binary ("b") –输出二进制数字
* octal ("o") -输出八进制数字
* hexadecimal ("x") -输出16进制数字，超过9的数字使用小写字母。
* hexadecimal ("X") -输出16进制数字，超过9的数字使用大写字母。
* character ("c") -打印前转换整数到相应的Unicode字符。
* SI-prefix ("s") –类似rounded，但是使用一个后缀单元例如 "9.5M" 代表百万或者 "1.00µ" 代表百万分之一。

类型 "n" 是 “,g”的一种缩写形式!

## [#](https://github.com/mbostock/d3/wiki/Namespaces" \l "wiki-qualify) d3.formatPrefix (*value*[, *precision*])

以指定的值和精度获得一个[SI prefix](http://en.wikipedia.org/wiki/Metric_prefix)对象。这个函数可用来自动判断数据的量级， 如K(千)，M(百万)等等。如果指定一个可选的精度，

该值计算前缀之前，使用相应d3.round函数四舍五入。 返回的前缀对象有两个属性：

•symbol（符号） - 前缀符号，如“M”百万。

•scale（规模） - scale函数，将数字转换为相应的前缀scale。

示例:

var prefix = d3.formatPrefix(1.21e9);

console.log(prefix.symbol); // "G";

console.log(prefix.scale(1.21e9)); // 1.21

这个方法用在d3.format对字符串的格式化中

## [#](https://github.com/mbostock/d3/wiki/Namespaces#wiki-qualify) d3.round (*x*[, *n*])

返回小数点后的值x四舍五入到n位。如果n被省略，它默认为零。其结果是一个数。值被调整为10^n最接近的倍数。如果两个倍数同样接近，则该值按照内置的round函数向上舍入。例如：

d3.round(1.23); // 1

d3.round(1.23, 1); // 1.2

d3.round(1.25, 1); // 1.3

d3.round(12.5, 0); // 13

d3.round(12, -1); // 10

请注意，当转换为字符串，得到的数字可能会由于IEEE浮点精度不精确；为了将固定小数位数字转换为字符串可以使用d3.format代替。

## Strings

## [#](https://github.com/mbostock/d3/wiki/Namespaces#wiki-qualify) d3.requote (*string*)

* 将字符串转义成可在正则表达式中使用的格式。



## Dates

参照 [d3.time](https://github.com/mbostock/d3/wiki/Time-Formatting) 模块。

Colin翻译，咕噜校正 2014年4月6日0:50:35