> *Wiki* ▸ *[[API--中文手册]]* ▸ *[[核心函数]]* ▸ ***格式化***

- 如发现翻译不当或有其他问题可以通过以下方式联系译者:
- 邮箱：zhang_tianxu@sina.com
- QQ群：D3数据可视化205076374，大数据可视化436442115

格式化数字是不经常用到的，只有在例如丑陋的"0.30000000000000004"出现在你的数轴标签上时，或者你想要使用固定精度将几千的数字组织为更加可读的形式，例如"$1,240.10"，再或者你可能只想展示一个特定的数字的显著位。D3使用标准的数字格式化使得一切变得简单，例如，创建一个用0补齐4位数字的函数，可以这样：

```
var zero = d3.format("04d");
```

现在，你就可以调用 zero 来很方便的格式化你的数字了：

```
zero(2); // "0002"
zero(123); // "0123"
```

当然，除了数字，D3还支持格式化和解析日期，逗号分隔字串。

Numbers

格式化数字示例

d3.**format**(*specifier*)

返回给定的字符串（specifier）的格式化函数（等同于适用默认的美国英语语言环境的locale.numberFormat)。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]]。规范（specifier）的通常格式如下:

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

*fill*可以是任意字符，除了 “{” 和 “}”，fill 由紧跟它的*align*选项标识。

*align*有三种选项：

- ("<") 在可用的区域左对齐。
- (">") 在可用的区域右对齐（默认）。
- ("^") 在可用的区域居中。

*sign*可能是：

- plus ("+") - 可以用于正数或负数。
- minus ("-") - 仅仅用于负数（默认）。
- space (" ") - 前面的空格应该用在正数前面，而减号必须用在负数！

*symbol*可能是：

- currency ("$") - 本地货币符号的前端或后缀
- base ("#") - 对于二进制、八进制或十六进制的输出,前缀分别是 "0b", "0o", or "0x".

*0*选项允许补零.

width 定义最小字段宽度. 如果没有指定,那么宽度将取决于内容。

*,*选项允许使用逗号作为千位分隔符.

The *precision* indicates how many digits should be displayed after the decimal point for a value formatted with types "f" and "%", or before and after the decimal point for a value formatted with types "g", "r" and "p".

The available *type* values are:

- exponent ("e") - use [[Number.toExponential|https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toExponential]].
- general ("g") - use [[Number.toPrecision|https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toPrecision]].
- fixed ("f") - use [[Number.toFixed|https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toFixed]].
- integer ("d") - use [[Number.toString|https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Number/toString]], but ignore any non-integer values.
- rounded ("r") - round to *precision* significant digits, padding with zeroes where necessary in similar fashion to fixed ("f"). If no *precision* is specified, falls back to general notation.
- percentage ("%") - like fixed, but multiply by 100 and suffix with "%".
- rounded percentage ("p") - like rounded, but multiply by 100 and suffix with "%".
- binary ("b") - outputs the number in base 2.
- octal ("o") - outputs the number in base 8.
- hexadecimal ("x") - outputs the number in base 16, using lower-case letters for the digits above 9.
- hexadecimal ("X") - outputs the number in base 16, using upper-case letters for the digits above 9.
- character ("c") - converts the integer to the corresponding unicode character before printing.
- SI-prefix ("s") - like rounded, but with a unit suffixed such as "9.5M" for mega, or "1.00µ" for micro.

The type "n" is also supported as shorthand for ",g".

d3.**formatPrefix**(*value*[, *precision*])

Returns the SI prefix for the specified *value*. If an optional *precision* is specified, the *value* is rounded accordingly using d3.round before computing the prefix. The returned prefix object has two properties:

- symbol - the prefix symbol, such as "M" for millions.
- scale - the scale function, for converting numbers to the appropriate prefixed scale.

For example:

```
var prefix = d3.formatPrefix(1.21e9);
console.log(prefix.symbol); // "G"
console.log(prefix.scale(1.21e9)); // 1.21
```

This method is used by d3.format for the `s` format.

d3.**round**(*x*[, *n*])

Returns the value *x* rounded to *n* digits after the decimal point. If *n* is omitted, it defaults to zero. The result is a number. Values are rounded to the closest multiple of 10 to the power minus *n*; if two multiples are equally close, the value is rounded up in accordance with the built-in [[round|https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Math/round]] function. For example:

```
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
```

Note that the resulting number when converted to a string may be imprecise due to IEEE floating point precision; to format a number to a string with a fixed number of decimal points, use d3.format instead.

Strings

d3.**requote**(*string*)

Returns a quoted (escaped) version of the specified *string* such that the string may be embedded in a regular expression as a string literal.

```
d3.requote("[]"); // "\[\]"
```

Dates

See the [[d3.time|Time-Formatting]] module.

name	time
gafish	20160423