[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) ▸ **Selections**

# **Selections**

## **0 Selections概览**

* [selection.size](https://github.com/mbostock/d3/wiki/Selections#wiki-size) - 返回选择集中的元素个数。
* [selection.select](https://github.com/mbostock/d3/wiki/Selections#wiki-select) - 选择所选的元素中的第一个子元素组成新的选择集。
* [selection.selectAll](https://github.com/mbostock/d3/wiki/Selections#wiki-selectAll) - 选择所选的元素中的多个子元素组成新的选择集。
* [d3.selection](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_selection) - 选择集对象原型（可通过 d3.selection.prototype 为选择集增强功能）。
* [d3.event](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_event) - 获取当前交互的用户事件。
* [d3.mouse](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_mouse) - 获取鼠标的相对某元素的坐标。
* [d3.touches](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_touches) - 获取相对某元素的触控点坐标。

[#](https://github.com/mbostock/d3/wiki/Selections#wiki-size) selection.**size**()

返回当前选择元素的总数量。

[#](https://github.com/mbostock/d3/wiki/Selections" \l "wiki-select) selection.**select**(*selector*)

对当前选中的每个元素，选中第一个匹配特定的选择器字符串的子代元素。如果当前元素没有元素匹配特定的选择器，当前索引处的元素在返回的选择中将是空值null。运算符（除了数据）自动跳过null元素，从而保持现有选择的索引。如果当前元素具有相关联的数据，该数据由返回的子选择继承，并且自动绑定到新选定的元素。如果有多个元素匹配选择器，只有在文档遍历顺序中第一个匹配的元素会被选中。选择器也可以被指定为一个函数，返回一个元素，或者null（如果没有匹配的元素）。在这种情况下，指定的选择器以和其他操作函数同样的方式被调用，被传递的当前数据D和索引i，与此上下文作为当前的DOM元素。(being passed the current datum d and index i, with the this context as the current DOM element)

[#](https://github.com/mbostock/d3/wiki/Selections" \l "wiki-selectAll) selection.**selectAll**(*selector*)

对当前选择的每个元素，选取匹配指定选择器字符串的后代元素。返回的选择是通过在当前选择的祖先节点进行分组。如果没有元素和当前元素的指定选择器相匹配，返回的选择中当前索引处的组将是空。部分选取不从当前选择继承数据，但如果数据值指定为一个函数，这个函数会被调用，带有的祖先节点的数据D和组索引i来确定部分选定的数据绑定。   
  
通过selectAll的分组也将影响后续进入的占位节点。因此，为追加进入的节点指定父节点，使用select后紧跟selectAll：

**d3.select("body").selectAll("div")**

您可以通过检查各组数据的parentNode属性查看每个组的父节点，例如selection[0].parentNode。   
  
选择器也可以被指定为一个函数，返回元素的数组（或节点列表NodeList），或者空数组（如果没有匹配的元素）。在这种情况下，指定的选择器以和其他操作函数同样的方式被调用，正在传递的电流基准D和索引i，与此上下文作为当前DOM元​​素..

[#](https://github.com/mbostock/d3/wiki/Selections#wiki-selectAll) selection.**selectAll**(*selector*)

返回根的选择，相当于d3.select（document.documentElement）。此函数也可以用来检查一个对象是否是一个选择：o instanceof d3.selection。您还可以为选择原型添加新的方法。例如，要添加一个方便的方法来设置复选框的“checked”属性，你可能会写：

d3.selection.prototype.checked = function(value) {

return arguments.length < 1

? this.property("checked")

: this.property("checked", value);

};

[#](https://github.com/mbostock/d3/wiki/Selections" \l "wiki-d3_event) d3.**event**

存储当前的事件，如果有的话。这全局函数是事件侦听器回调过程中和运算符注册的。当监听器在finally块被通知后当前事件被重置。这使侦听器函数与其他运算符函数具有相同的形式，当被传递当前数据D和索引i时 。

该d3.event对象是DOM事件，并实现了标准事件字段，像时间戳和键代码，以及preventDefault()方法和的stopPropagation()方法。当然你可以使用原生事件的pageX和pageY ，它往往是更方便转变事件位置为接收该事件容器的局部坐标系。例如，如果你在网页的正常流程嵌入SVG，你可能想事件的位置是相对于SVG图像的左上角的。如果您的SVG包含转换，你也可能想知道事件的相对于那些变换的位置。标准鼠标指针使用d3.mouse运算符，d3.touches用于在iOS多点触控事件。

[#](https://github.com/mbostock/d3/wiki/Selections" \l "wiki-d3_mouse) d3.**mouse**(*container*)

返回当前d3.event相对于指定的容器的x和y坐标，。该容器可以是一个HTML或SVG容器元素，如svg:g或svg:svg 。该坐标返回为一个包含两个元素的数组[ X，Y ] 。

[#](https://github.com/mbostock/d3/wiki/Selections" \l "wiki-d3_touches) d3.**touches**(*container*[, *touches*])

返回与当前d3.event相关联的每个触摸x和y坐标，基于触摸的属性，相对于指定的容器中。该容器可以是一个HTML或SVG容器元素，如svg:g 或 svg:svg。坐标返回两个元素的数组的数组[[X1，Y1]，[X2，Y2]，...]。如果指定了触摸，返回指定触摸的位置，如果没有指定触摸时，则默认为当前事件的触摸属性上。

# 简单记忆：

# 补充知识：

# 单词一览

|  |  |
| --- | --- |
| preserving | 保持 |
| inherited | 继承 |
| traversal | 遍历 |
| invoked | 调用 |
| manner | 方式 |
| descendant | 后代 |
| ancestor | 祖先 |
| placeholder | 后续 |
| equivalent | 相当于 |
| inspecting | 检查 |
| coordinate | 坐标 |
| embed | 嵌入 |
| multitouch | 多点触控 |

Gulu 2014年3月27日22:22:39