> *Wiki* ‣ *[[API--中文手册]]* ‣ *[[行为]]* ‣ ***拖动***

- 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

该行为会自动创建事件监听器来处理元素的拖动，可支持鼠标事件和触摸事件。

d3.behavior.**drag**()

构造一个新的拖拽行为. 构造后, 可以用 `selection.call` 将拖放行为应用于所选择的元素:

```
var drag = d3.behavior.drag();
selection.call(drag);
```

所有注册的监听器使用 "drag" 命名空间, 故如下可以移除拖放行为:

```
selection.on(".drag", null);
```

drag.**on**(*type*[, *listener*])

注册指定的监听器 *listener* 来接收拖动行为中指定类型 *type* 的事件；如果监听器 *listener* 未指定，则为指定的类型*type* 的事件返回当前已注册的监听器(事件类型可能包含命名空间，查看详细参见： dispatch.on)，可支持的事件类型包括:

- dragstart - 拖动开始时.
- drag - 拖动移动时.
- dragend - 拖动结束时（放下时）.

拖动事件(除了dragstart 和 dragend) 使用 *x* 和 *y* 属性来表示本地坐标系中拖动行为当前位置；默认情况下，这个位置即是鼠标 mouse(或触摸 touch) 的位置，然而，这个位置可以通过指定一个原点(origin)修改；拖动事件同时也提供 "dx" 和 "dy" 属性来表示鼠标的“瞬时”偏移量（相对于上一时刻的偏移：正数或负数），这两个属性有时比指定一个明确地原点更方便。

在拖动的动作中，一些浏览器的默认动作会自动被阻止（如：选择文本），另外，点击事件的默认行为随即变成一个非空拖动动作也会被禁止，以便允许链接的拖动。当在可拖动元素中注册了点击事件，你可以这样来检查click事件是否被禁止：

```
selection.on("click", function() {
  if (d3.event.defaultPrevented) return; // click suppressed
  console.log("clicked!");
});
```

当拖拽事件结合其他事件一起使用时(例如使拖拽行为优先于缩放)，你可以考虑阻止源事件，来避免多个动作的发生：

```
drag.on("dragstart", function() {
  d3.event.sourceEvent.stopPropagation(); // silence other listeners
});
```

drag.**origin**([*origin*])

如果指定了原点 *origin* ， 设置原点访问器为指定的函数；如果未指定原点*origin* ， 则返回当前原点访问器，原点默认为 `null` ；

原点访问器函数*origin*被用来确定拖拽开始时的起始位置，这就允许拖拽行为保存鼠标位置相对于开始元素位置的偏移量；如果原点访问器为空，元素位置被设置为鼠标位置时可能会有明显的跳动；如果指定了原点访问器，该函数会在鼠标被按下时调用，该函数会像其他函数调用方式一样被调用，即：会传递当前元素的数据 `d` 和元素索引 `i` ，还有 `this` 上下文代表当前点击的DOM元素；要访问当前发生的事件，可以使用d3.event；指定的原点访问器必须返回一个包含被拖动元素开始坐标 `x` 和 `y` 的对象；

原点访问器常常指定为恒等函数： `function(d) { return d; }` ，当前元素已绑定了一个包含坐标位置 `x` 和 `y` 的对象时是最合适的，详细参见： http://bl.ocks.org/1557377。

- 魏飞T20141125
- guluP20141208 2014-12-8 21:53:51