x.co/d3course/

D3.js 課程資源

- 課程投影片
- 範例程式檔
- 線上範例

Environment Setup

1. unzip Example.zip
2. Create index.html inside Example/
3. Create index.html and type:

```
<body></body>
<script src="d3.min.js"></script>
<script>
d3.select("body").text("Hello World!");
</script>
```

Continuous Transition

http://bl.ocks.org/mbostock/1256572


SVG

```
<body>
<svg width="100%" height="100%">
</svg>
</body>
<script src="d3.min.js"></script>
<script></script>
```

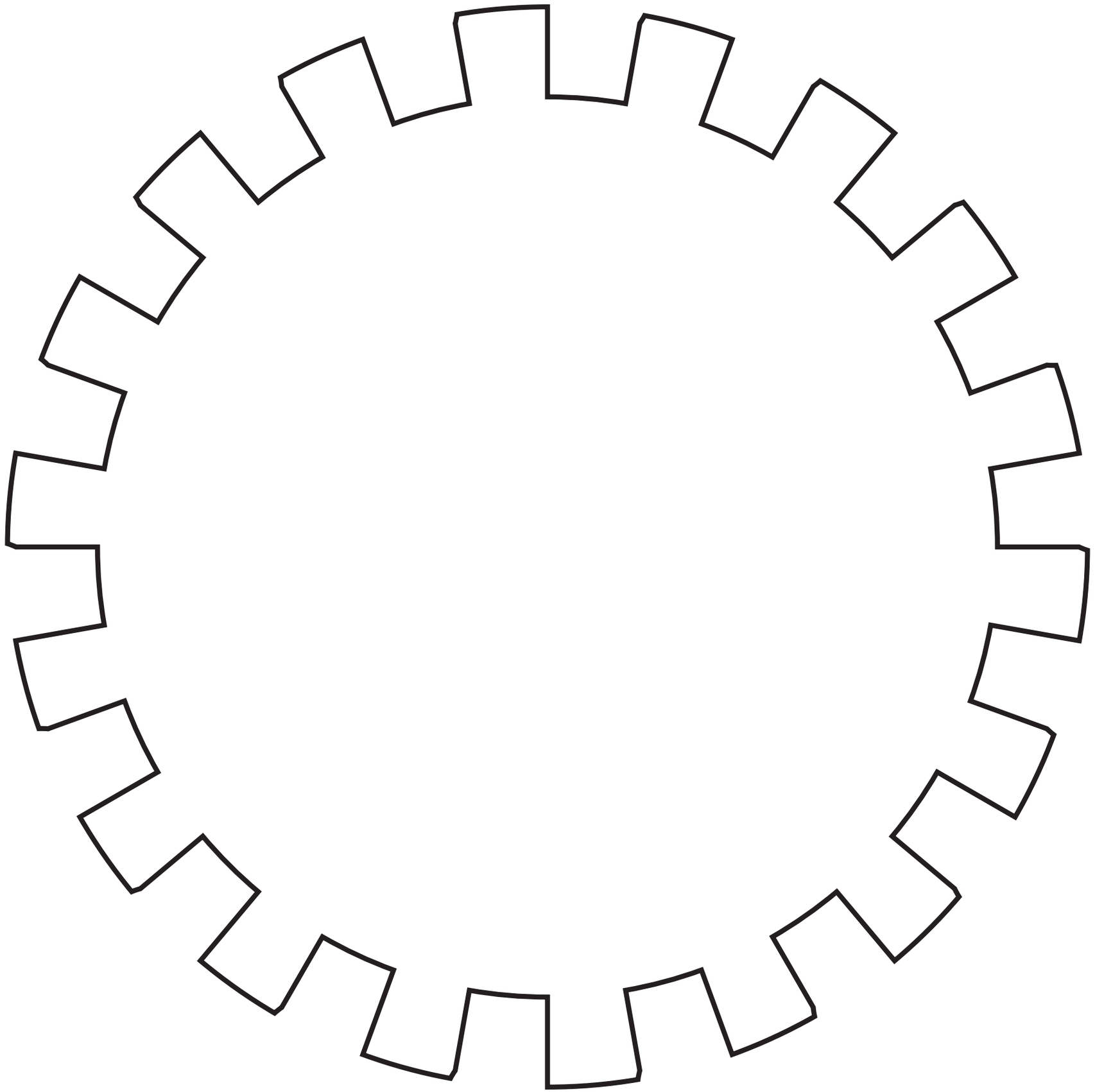
練習 #01　利用 SVG 畫一個紅色的圓

```
<body>
<svg width="100%" height="100%">
<circle
   cx="200"
   cy="200"
   r="100"
   fill="red"/>
</svg>
</body>
<script src="d3.min.js"></script>
<script></script>
```

加分題: 加上邊框

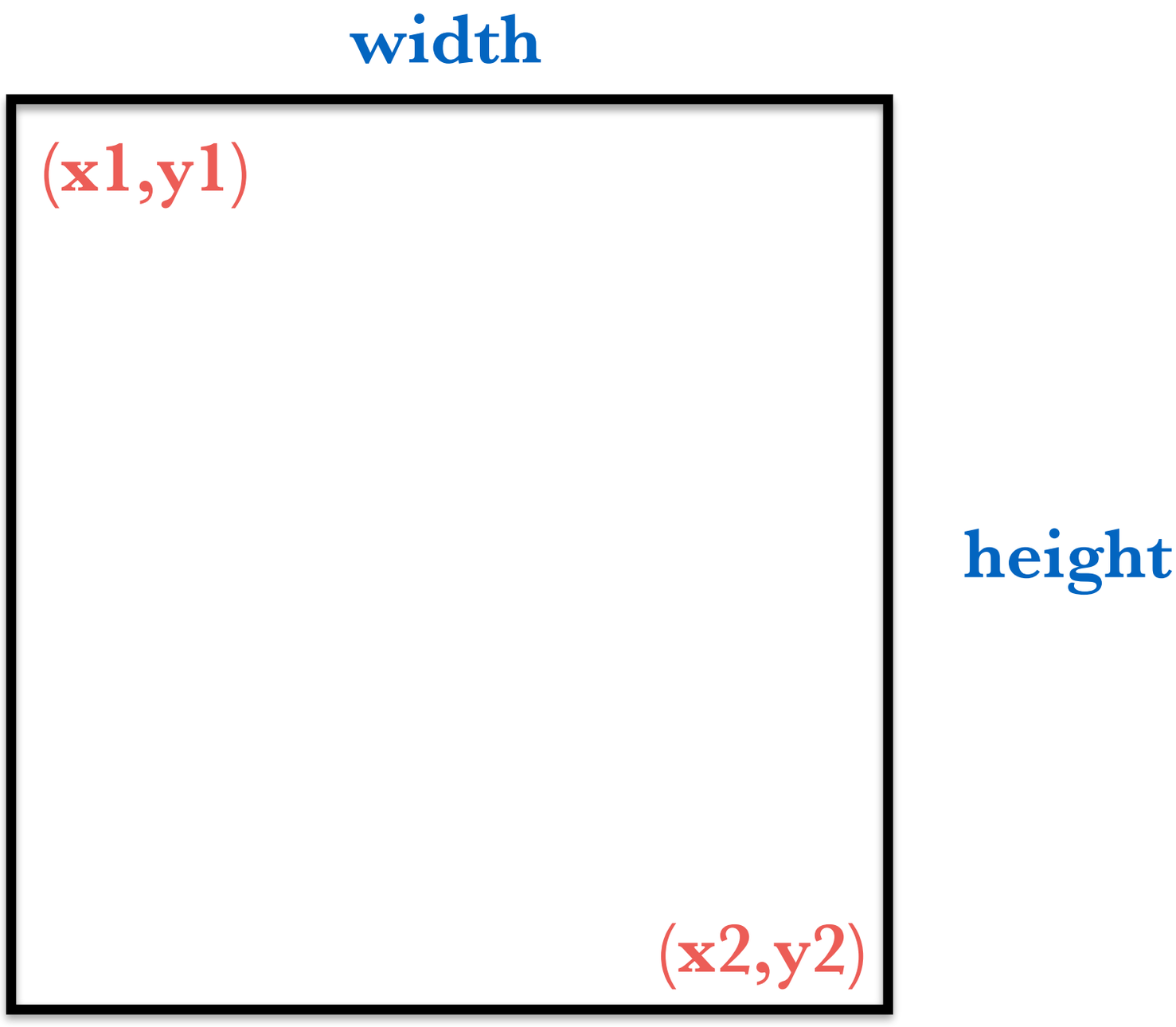
```
stroke: "blue",
"stroke-width": 5
```

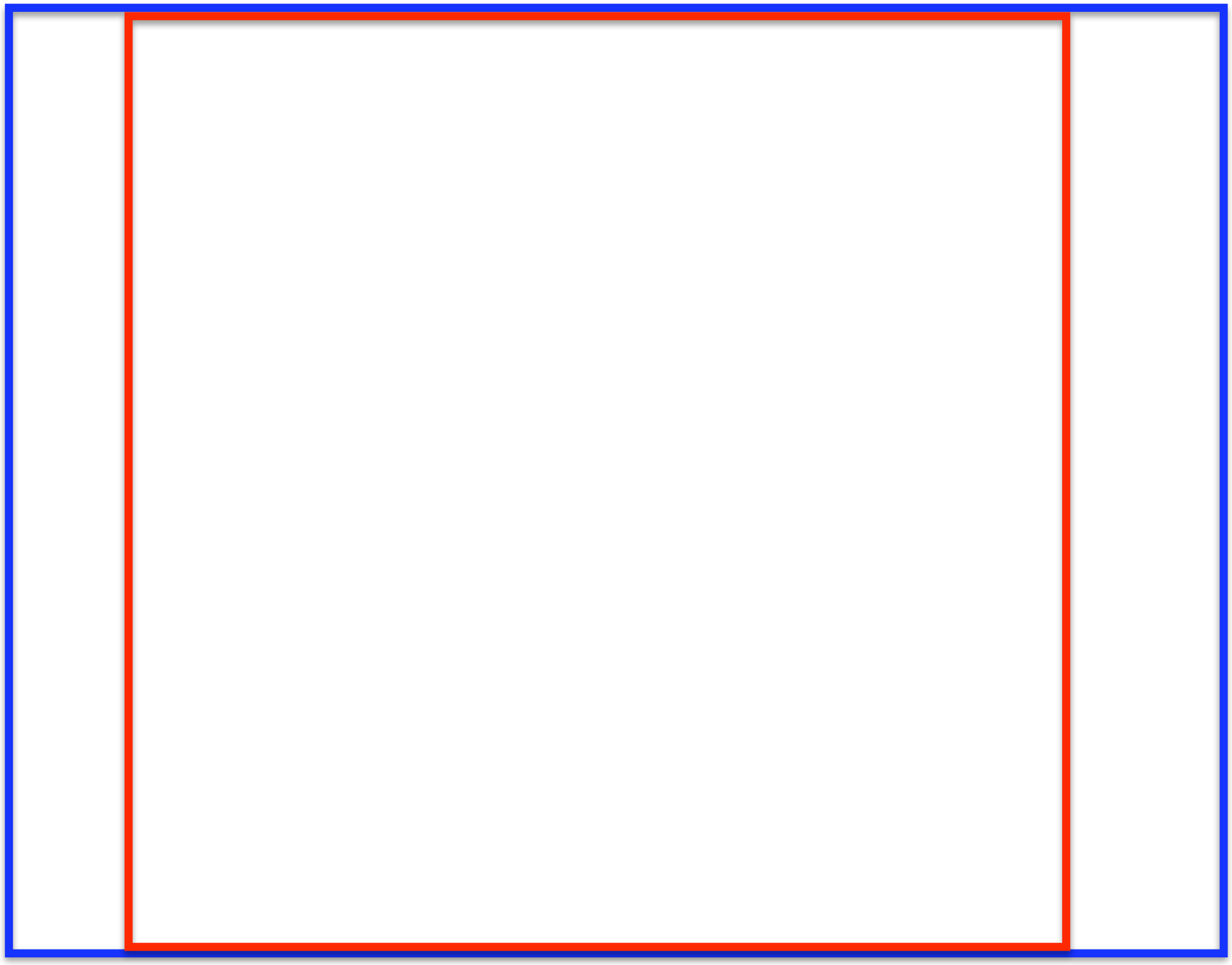
<circle/>	cx, cy, r
<rect/>	x, y, width, height
<line/>	x1, y1, x2, y2
<path/>	d
<text></text>	x, y, dx, dy



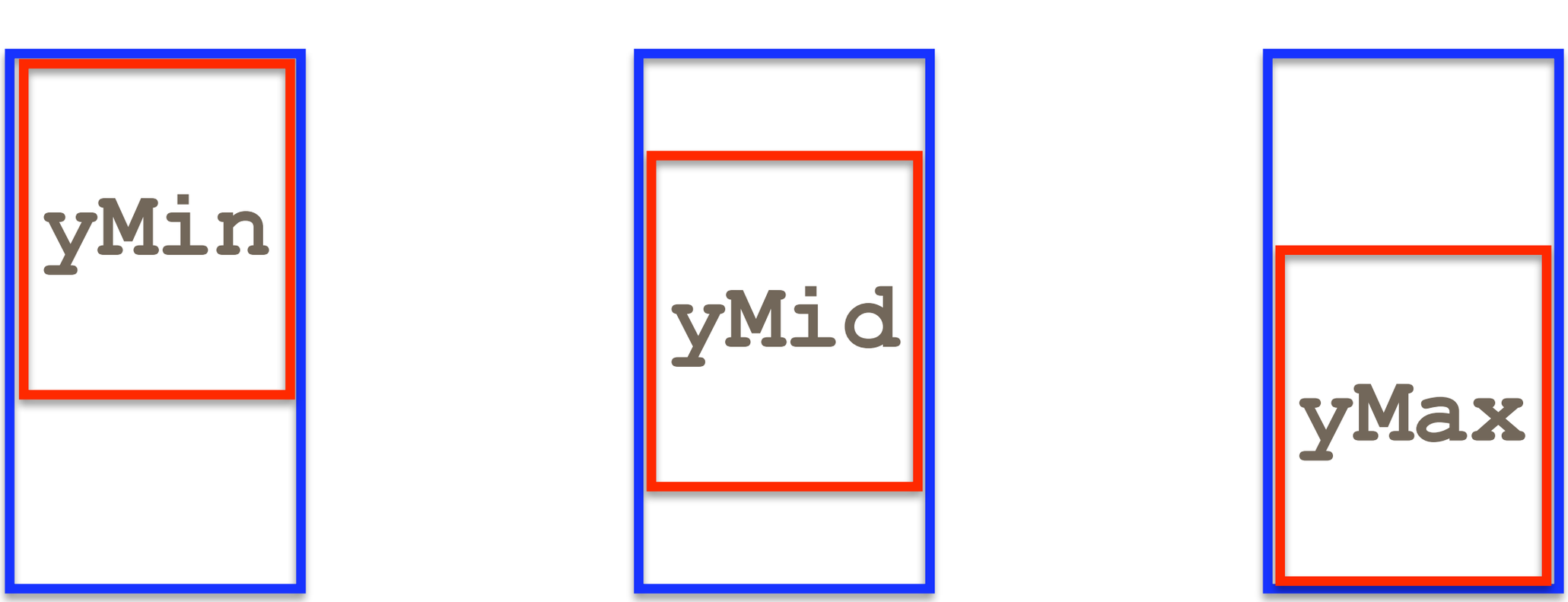
<?xml version="1.0" encoding="utf-8"?> <!-- Generator: Adobe Illustrator 19.0.0, SVG Export Plug-In . SVG Version: 6.00 Build 0) --> <svg version="1.1" baseProfile="tiny" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 1024 576" xml:space="preserve"> <path id="XMLID_39_" fill="#FFFFFF" stroke="#231F20" stroke-miterlimit="10" d="M574.4,270.8c1.1-6.1,1.6-12.1,1.6-18.1 c-0.6-0.2-1.1-0.4-1.7-0.7H558c0-5.3-0.5-10.6-1.4-15.6l17.7-3.1c-1.1-6.1-2.6-11.9-4.6-17.6c-0.6,0-1.2,0-1.8,0l-15.3,5.6 c-1.8-5-4-9.7-6.6-14.2l15.6-9c-3.1-5.3-6.6-10.3-10.4-14.9c-0.6,0.2-1.1,0.4-1.7,0.6l-12.5,10.5c-3.4-4-7.1-7.7-11.1-11.1 l11.6-13.8c-4.7-4-9.7-7.5-14.8-10.5c-0.4,0.4-0.9,0.8-1.4,1.1L513,174c-4.5-2.6-9.3-4.8-14.2-6.6l6.1-16.9 c-5.8-2.1-11.7-3.7-17.5-4.8c-0.3,0.5-0.6,1.1-0.9,1.6l-2.8,16.1c-5.1-0.9-10.3-1.4-15.6-1.4v-18c-6.2,0-12.2,0.5-18.1,1.5 c-0.1,0.6-0.2,1.2-0.4,1.8l2.8,16.1c-5.2,0.9-10.3,2.3-15.2,4.1l-6.1-16.9c-5.8,2.1-11.3,4.7-16.5,7.6c0.1,0.6,0.2,1.2,0.3,1.8 L423,174c-4.5,2.6-8.8,5.7-12.8,9l-11.6-13.8c-4.7,4-9,8.3-12.9,12.8c0.3,0.5,0.6,1,0.9,1.6l12.5,10.5c-3.4,4-6.4,8.3-9,12.8 l-15.6-9c-3.1,5.3-5.7,10.8-7.7,16.4c0.5,0.4,0.9,0.8,1.4,1.2l15.3,5.6c-1.8,4.9-3.1,9.9-4.1,15.2l-17.7-3.1 c-1.1,6.1-1.6,12.1-1.6,18.1c0.6,0.2,1.1,0.4,1.7,0.7H378v0c0,5.3,0.5,10.6,1.4,15.6l-17.7,3.1c1.1,6.1,2.6,11.9,4.6,17.6 c0.6,0,1.2,0,1.8,0l15.3-5.6c1.8,5,4,9.7,6.6,14.2l-15.6,9c3.1,5.3,6.6,10.3,10.4,14.9c0.6-0.2,1.1-0.4,1.7-0.6l12.5-10.5 c3.4,4,7.1,7.7,11.1,11.1l-11.6,13.8c4.7,4,9.7,7.5,14.8,10.5c0.4-0.4,0.9-0.8,1.4-1.1L423,330c4.5,2.6,9.3,4.8,14.2,6.6l-6.1,16.9 c5.8,2.1,11.7,3.7,17.5,4.8c0.3-0.5,0.6-1.1,0.9-1.6l2.8-16.1c5.1,0.9,10.3,1.4,15.6,1.4v18c6.2,0,12.2-0.5,18.1-1.5 c0.1,0.6,0.2,1.2,0.4,1.8l-2.8-16.1c5.2-0.9,10.3-2.3,15.2-4.1l6.1,16.9c5.8-2.1,11.3-4.7,16.5-7.6c-0.1-0.6-0.2-1.2-0.3-1.8 L513,330c4.5-2.6,8.8-5.7,12.8-9l11.6,13.8c4.7-4,9-8.3,12.9-12.8c-0.3-0.5-0.6-1-0.9-1.6l-12.5-10.5c3.4-4,6.4-8.3,9-12.8l15.6,9 c3.1-5.3,5.7-10.8,7.7-16.4c-0.5-0.4-0.9-0.8-1.4-1.2l-15.3-5.6c1.8-4.9,3.1-9.9,4.1-15.2L574.4,270.8z"/></svg>

```
<svg
    width="width" height="height"
    viewBox="x1 y1 x2 y2">
```





preserveAspectRatio



```
<SVG
width="800px" height="600px"
viewBox="0 0 400 300"
preserveAspectRatio="xMidYMid">
```

```
d3.select("svg")
    .append("circle")
    .attr({
      cx: 200,
      cy: 200,
      r: 100,
      fill: "red"
});
```

```
d3.select("svg")
    .append("circle")
    .attr({
        cx: 200,
        cy: 200,
        r: 100,
        fill: "red"
});
```

functional style

（注意傳回值）

config with object

練習 #02 利用 D3.js 畫一個紅色的長方形

```
<body>
<svg width="100%" height="100%">
</svg>
</body>
<script src="d3.min.js"></script>
<script>
d3.select("svg")
  .append("rect")
  .attr({
    x: 10,
    y: 10,
    width: 100,
    height: 20,
    fill: "red"
  });
</script>
```

加分題: 畫一個 Quby

(使用 circle + rect 即可辦到)


```
var node = d3.select("body");

node.text("Hello World!")

node.html("<b>Hello World!</b>")

node.attr({"data-toggle": "#me"})

node.style({"width": "100%"})
```

```
node.text("Hello World!")
```

```
<body>Hello World!</body>
```

```
node.html("<b>Hello World!</b>")
```

```
<body><b>Hello World!</b></body>
```

```
node.attr({"data-toggle": "#me"})
```

```
<body data-toggle="#me"></body>
```

```
node.style({"width": "100%"})
```

```
<body style="width:100%"></body>
```

```
node.style(
    {
        "width": "100%",
        "height": "100%",
        "background: "red"
    }
)
```

```
var node = d3.select("body");

newnode = node.append("div");
```

```
<body><div></div></body>
```

return "div"

```
newnode.remove();
```

```
<body></body>
```

練習 #02 利用 D3.js 畫一個紅色的長方形

```
<body>
<svg width="100%" height="100%">
</svg>
</body>
<script src="d3.min.js"></script>
<script>
d3.select("svg")
  .append("rect")
  .attr({
    x: 10,
    y: 10,
    width: 100,
    height: 20,
    fill: "red"
  });
</script>
```

我要打十個

暴力法

```
= d3.select(“svg”).append(“rect”).attr({…})
```

x 10

```
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
= d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
  d3.select(“svg”).append(“rect”).attr({…})
```

使用迴圈

```
for(var i =1; i<=10;i++) {
   d3.select("svg")
      .append("rect")
      .attr({
        width: data[i]
      });
}
```

使用迴圈

```
for(var i =1; i<=10;i++) {
  d3.select("svg")
    .append("rect")
    .select("rect")
    .style({
      width: data[i]
    });
}
```

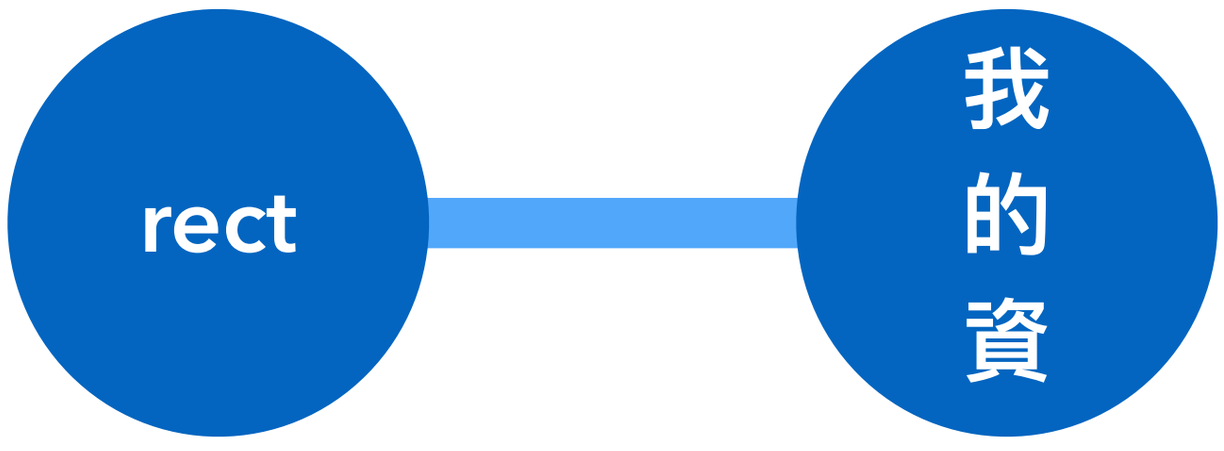
reorder?

```
d3.select("rect:nth-of-type(5)")
```

data?

Data Binding

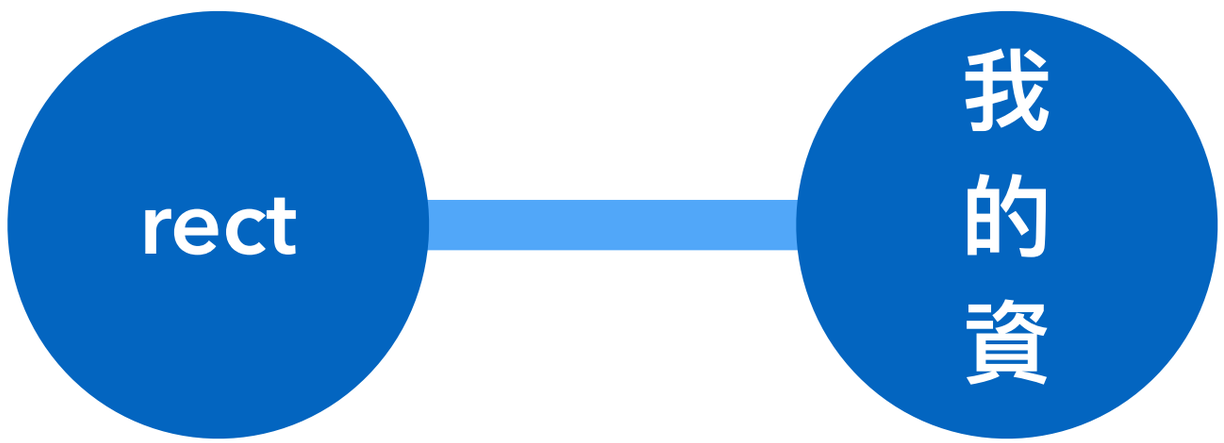
```
d3.select("rect")
    .datum( 我的資料 )
```



Data Binding

d3.select("rect").datum(我的資料)

```
d3.select("rect").datum();
```



使用迴圈

reorder?

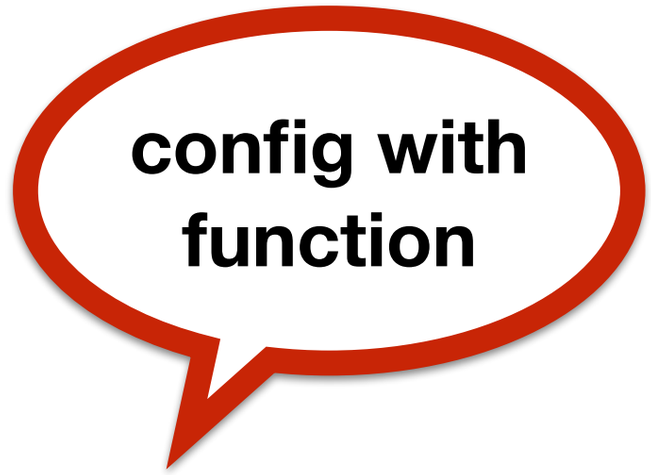
```
for(var i =1; i<=10;i++) {
    d3.select("svg")
      .append("rect")
      .select("rect")
      .style({
        width: data[i] + “px”
      });
}
```

```
d3.select("rect:nth-of-type(5)")
```

data?

.datum()

```
d3.select("rect")
   .datum( 我的資料 )
   .attr({
      "width":function(d,i) {
         return this;
      }
});
```




```
   d    = 我的資料
this    = <rect></rect>
   i    = 0
width = 回傳值
```

```
for(var i =1; i<=10;i++) {
   d3.select("svg")
      .append("rect")
      .datum(data[i])
      .attr({
         width: function(d,i) {
            return d * 10;
         }
      });
}
```

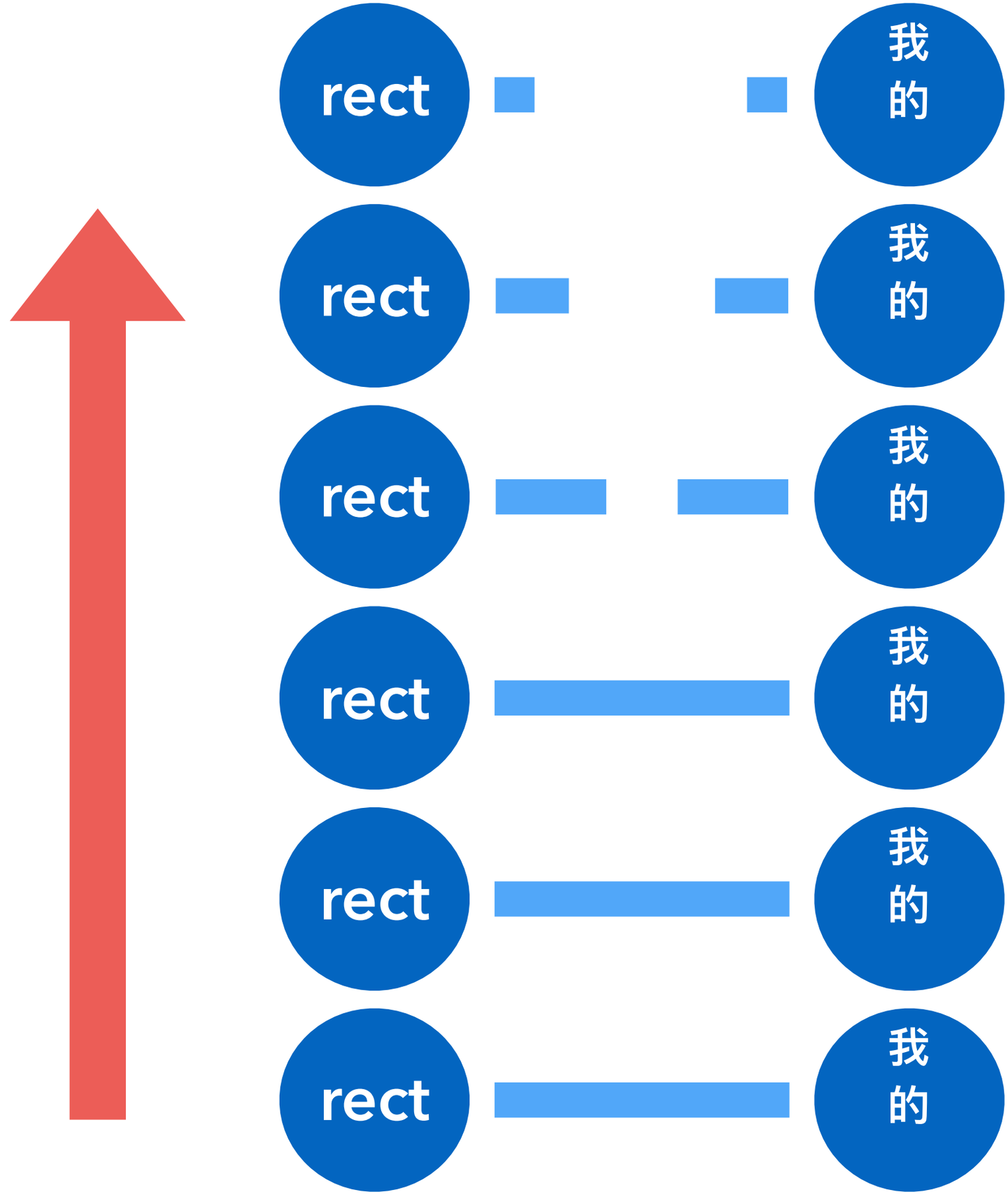
```
for(var i =1; i<=10;i++) {
    d3.select("svg")
        .append("rect")
        .datum(data[i])
        .attr({
            width: function(d,i) {
                return d * 10;
            }
        });
    }
```

到底要多少個？

多
出來的

幫我自動

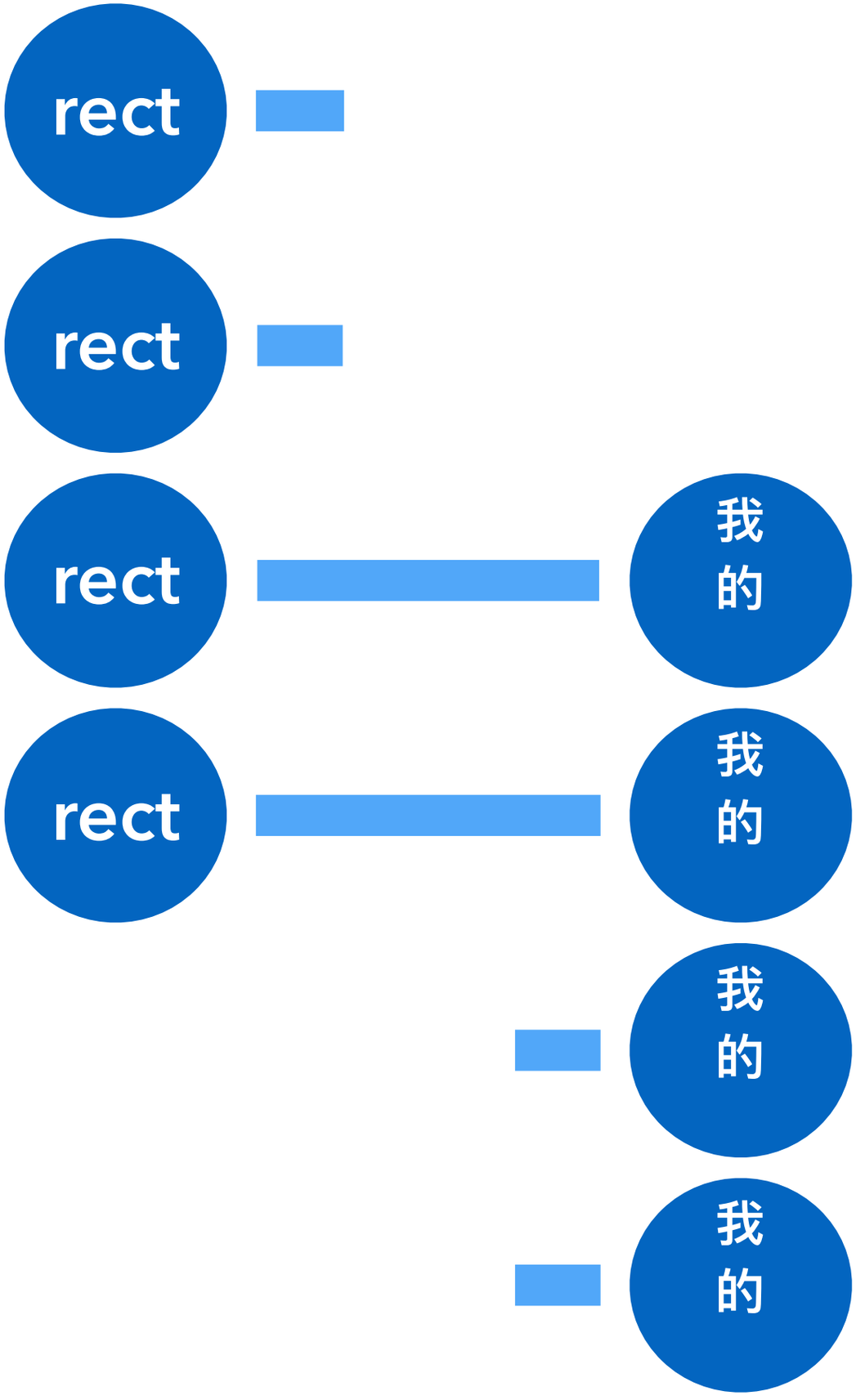
綁定資料



資料已經
不見的

之前已經
綁好的

還沒建立
元素的



我們需要

元素陣列

資料陣列

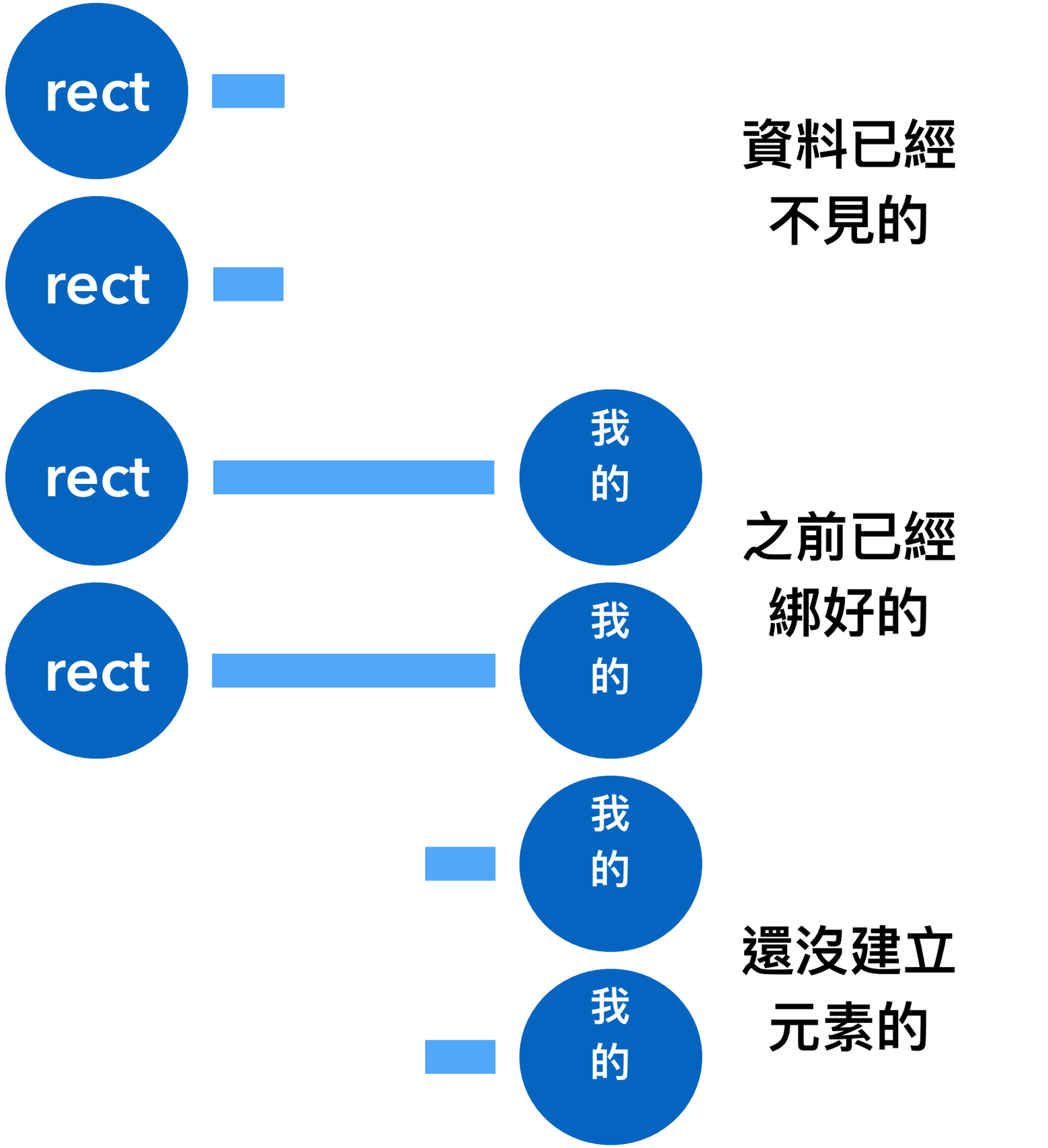
綁定動作

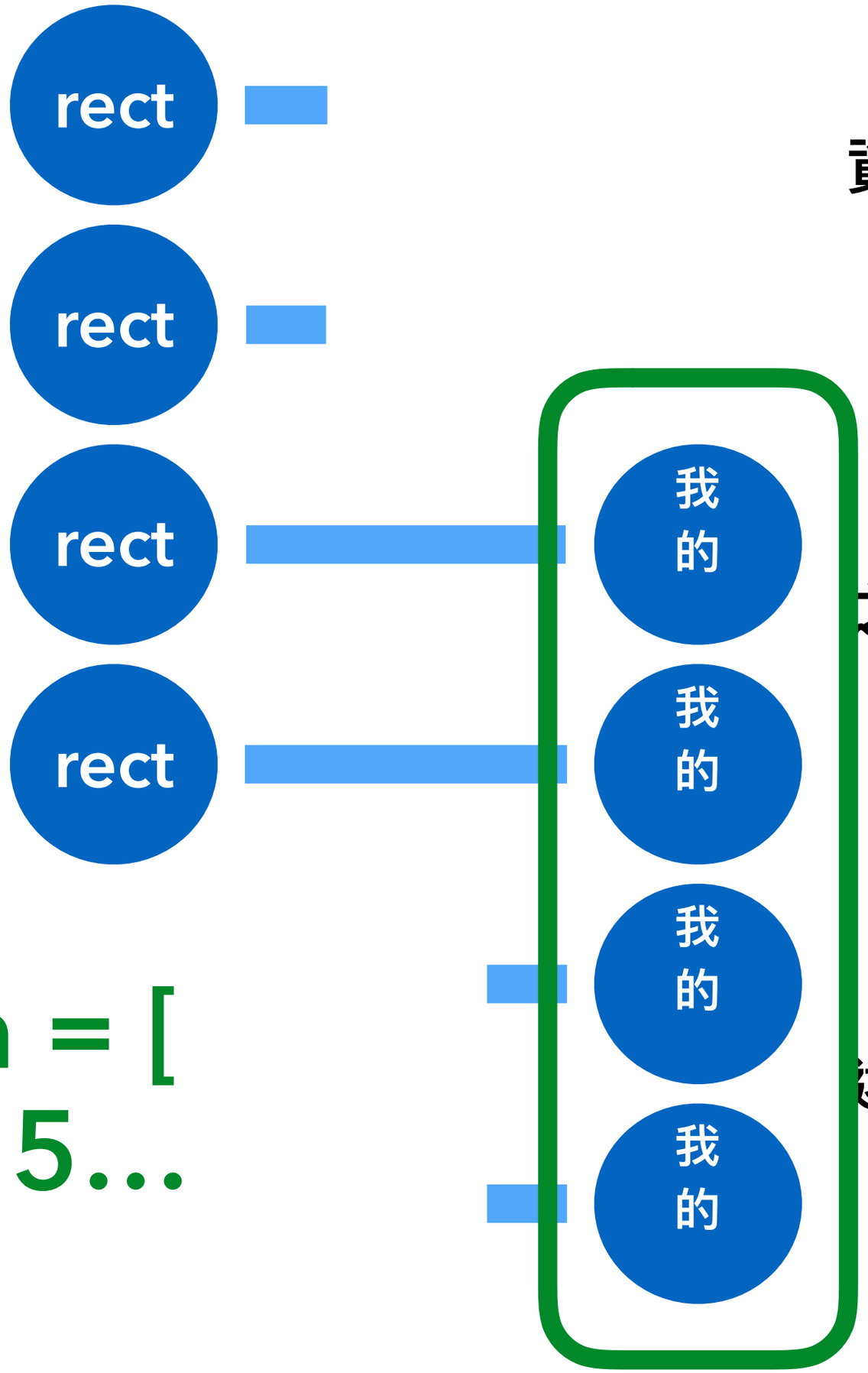
我們想要

A 集合

B 集合

C 集合





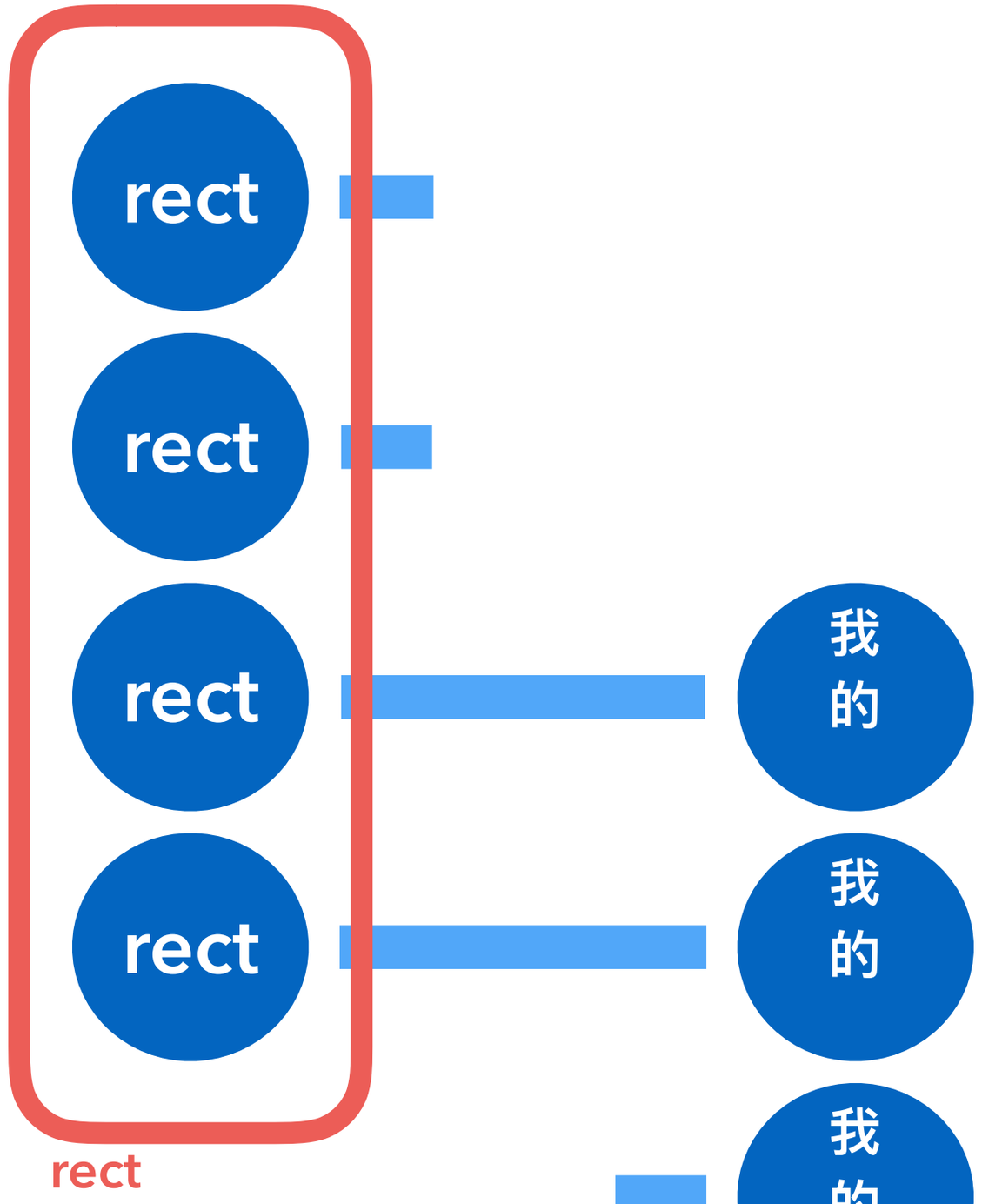
資料已經
不見的

之前已經
綁好的

還沒建立
元素的

```
var mydata = [
   3, 1, 4, 1, 5...
];
```

```
var mydata = [
   3, 1, 4, 1, 5...
];
```



資料已經不見的

之前已經綁好的

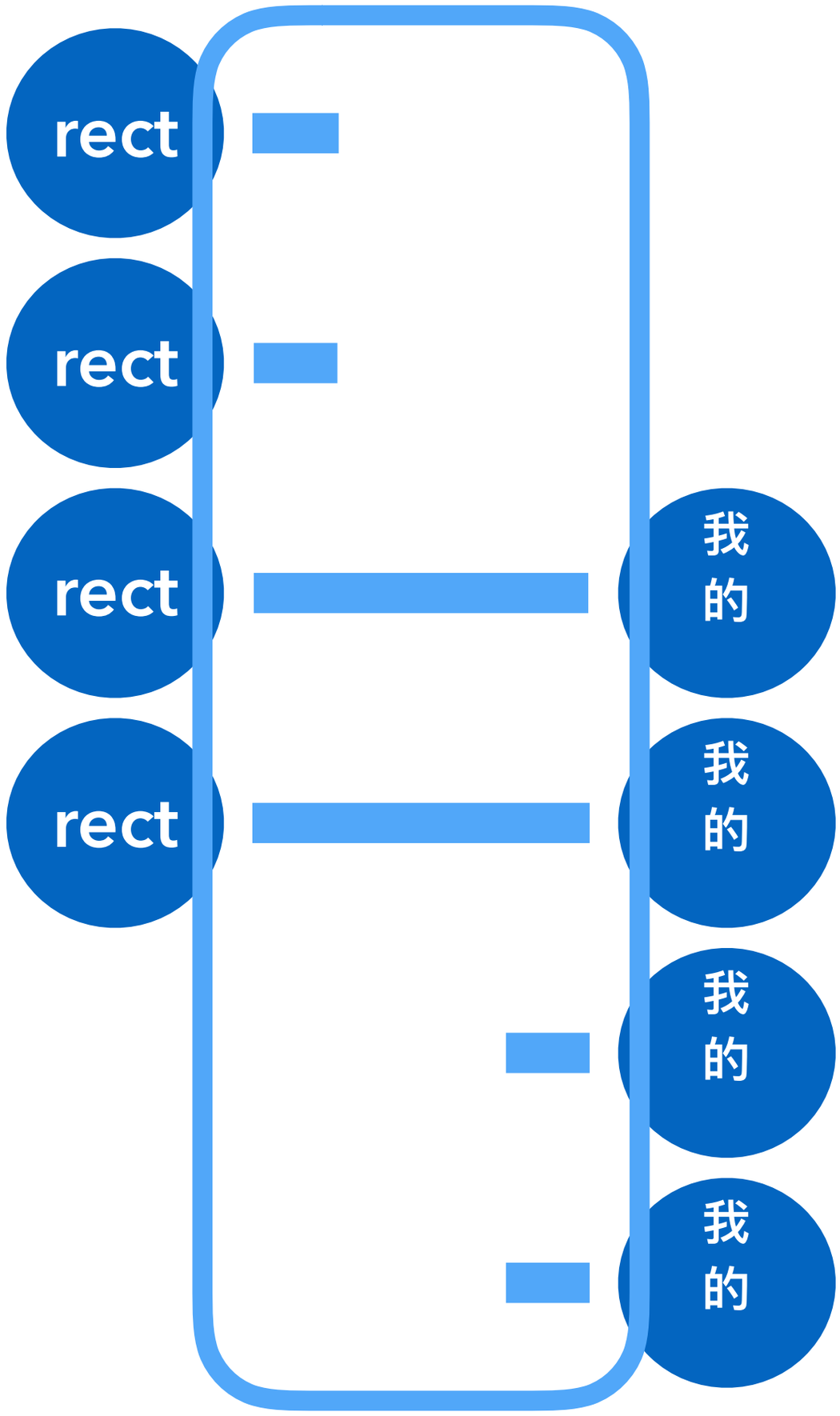
還沒建立元素的

d3.selectAll("rect")

```
var mydata = [
   3, 1, 4, 1, 5...
];
```

d3.selectAll("rect")

.data(mydata)

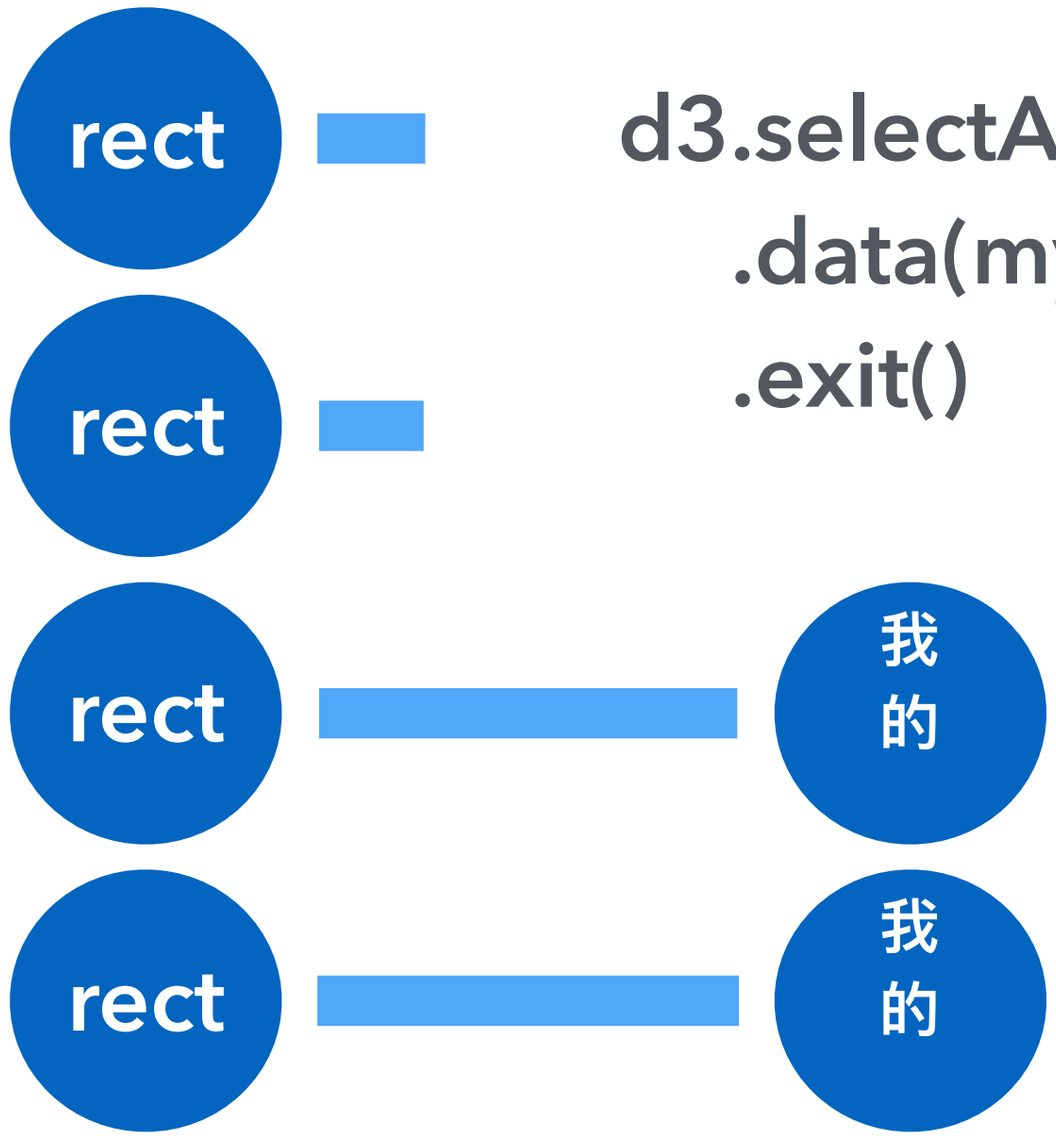


資料已經不見的

之前已經綁好的

還沒建立元素的

d3.selectAll("rect")
.data(mydata)

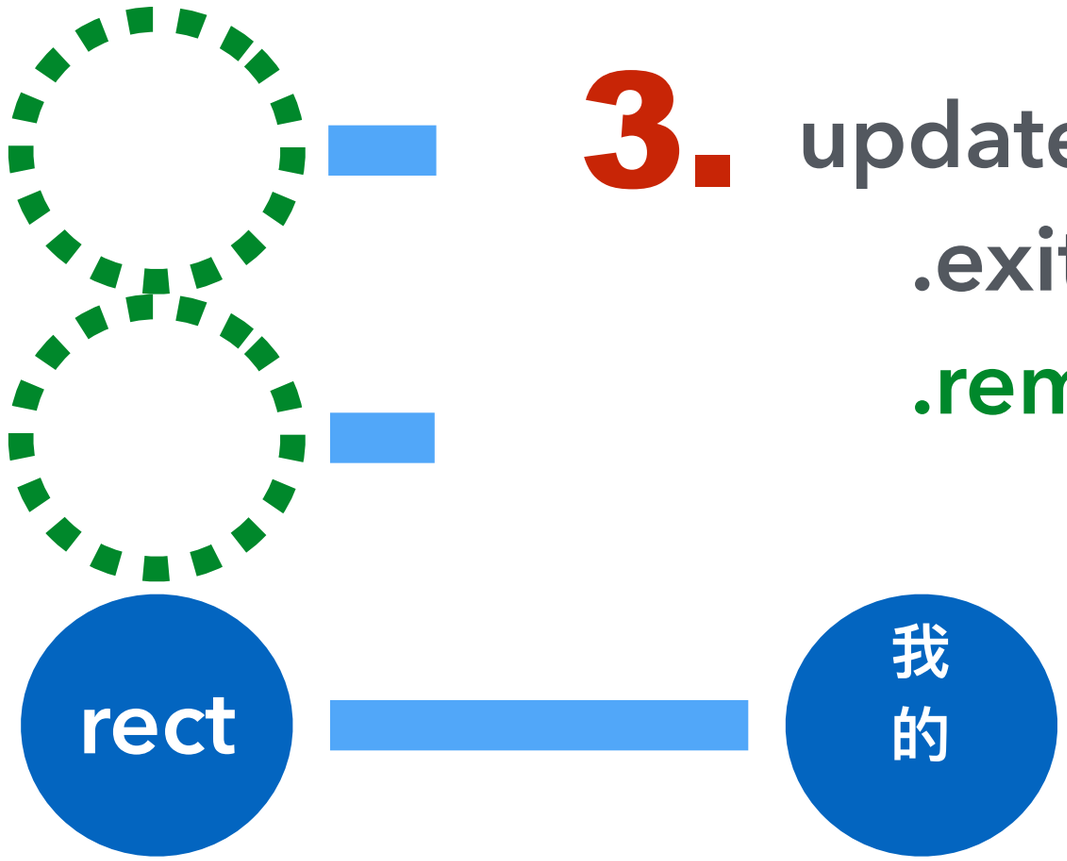


d3.selectAll("rect")
.data(mydata)
.exit()

d3.selectAll("rect")
.data(mydata)
.enter()

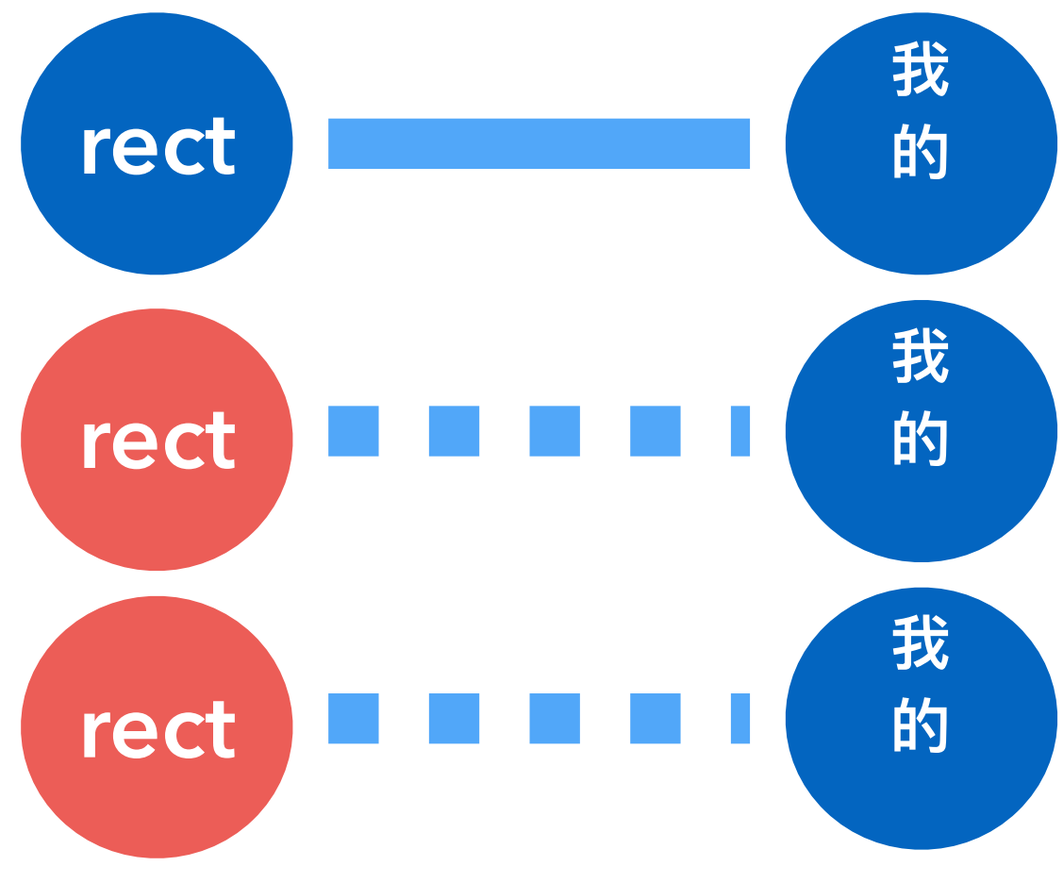
1.

```
var update = d3
    .selectAll("rect")
    .data(mydata);
```



2.

```
update
    .enter()
    .append("rect")
```



3.

```
update
    .exit()
    .remove()
```

```
var update = d3
    .select("svg")
    .selectAll("rect")
    .data([3,1,4,1,5,9]);

update.enter()
    .append("rect");

update.exit()
    .remove();
```

```
var update = d3
   .select("svg")
   .selectAll("rect")
   .data([3,1,4,1,5,9]);
update.enter().append("rect");
update.exit().remove();
```

```
d3.selectAll("rect")
   .attr({
      width: function(d,i) {
         ....
      },
      ....
   });
```

```
var update = d3
   .select("svg")
   .selectAll("rect")
   .data([3,1,4,1,5,9]);
update.enter().append("rect");
update.exit().remove();
```

```
d3.selectAll("text")
   .text(function(d,i) {
     return "hello world!";
   });
```

```
var update = d3.select("svg")
  .selectAll("rect").data([3,1,4,1,5,9]);
update.enter().append("rect");
update.exit().remove();
d3.select("svg").selectAll("rect").attr({
  x: 10,
  y: function(d,i) {
    return i * 12;
  },
  width: function(d,i) {
    return d * 10;
  },
  height: 10,
  fill: "red"
});
```

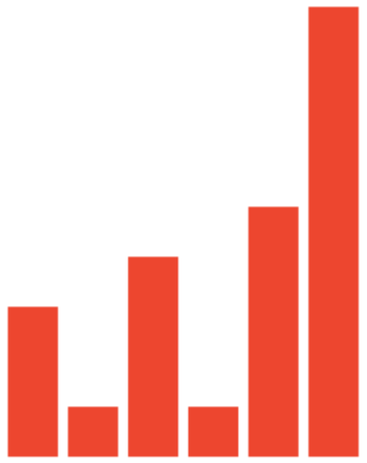
07.html
08.html
08-04.html

練習 #03　畫一個垂直的長條圖

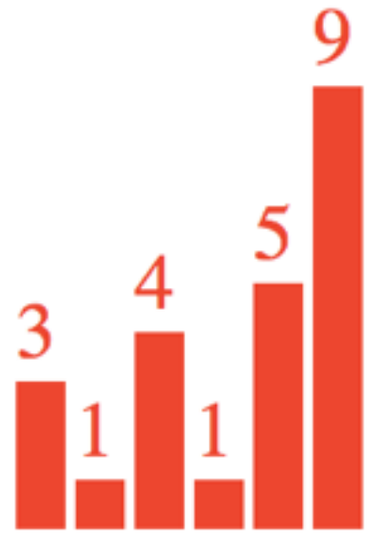
範例資料：[3,1,4,1,5,9,2,6,5]

範例如下:

加分題1



加分題2

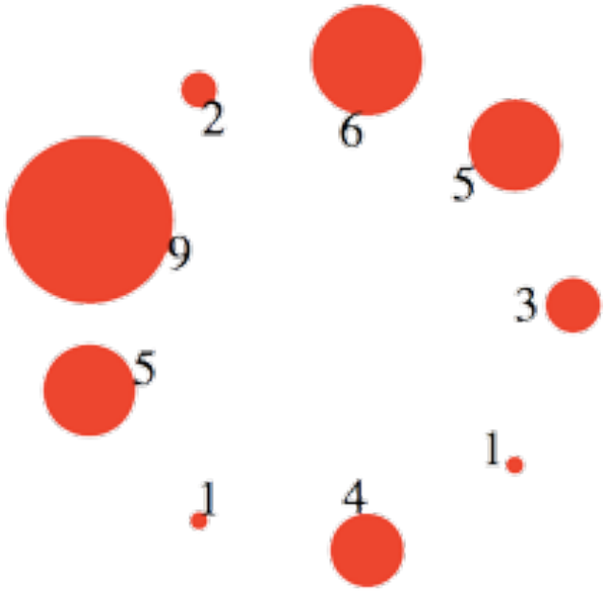


(使用 <text> 與 .text)

加分題3

(使用 circle)

加分題4



來點動畫吧！

```
d3.selectAll("rect")
  .attr({width: 0})
  .attr({width: 100})
```

來點動畫吧！

```
d3.selectAll("rect")
  .attr({width: 0})
  .transition()
  .attr({width: 100})
```

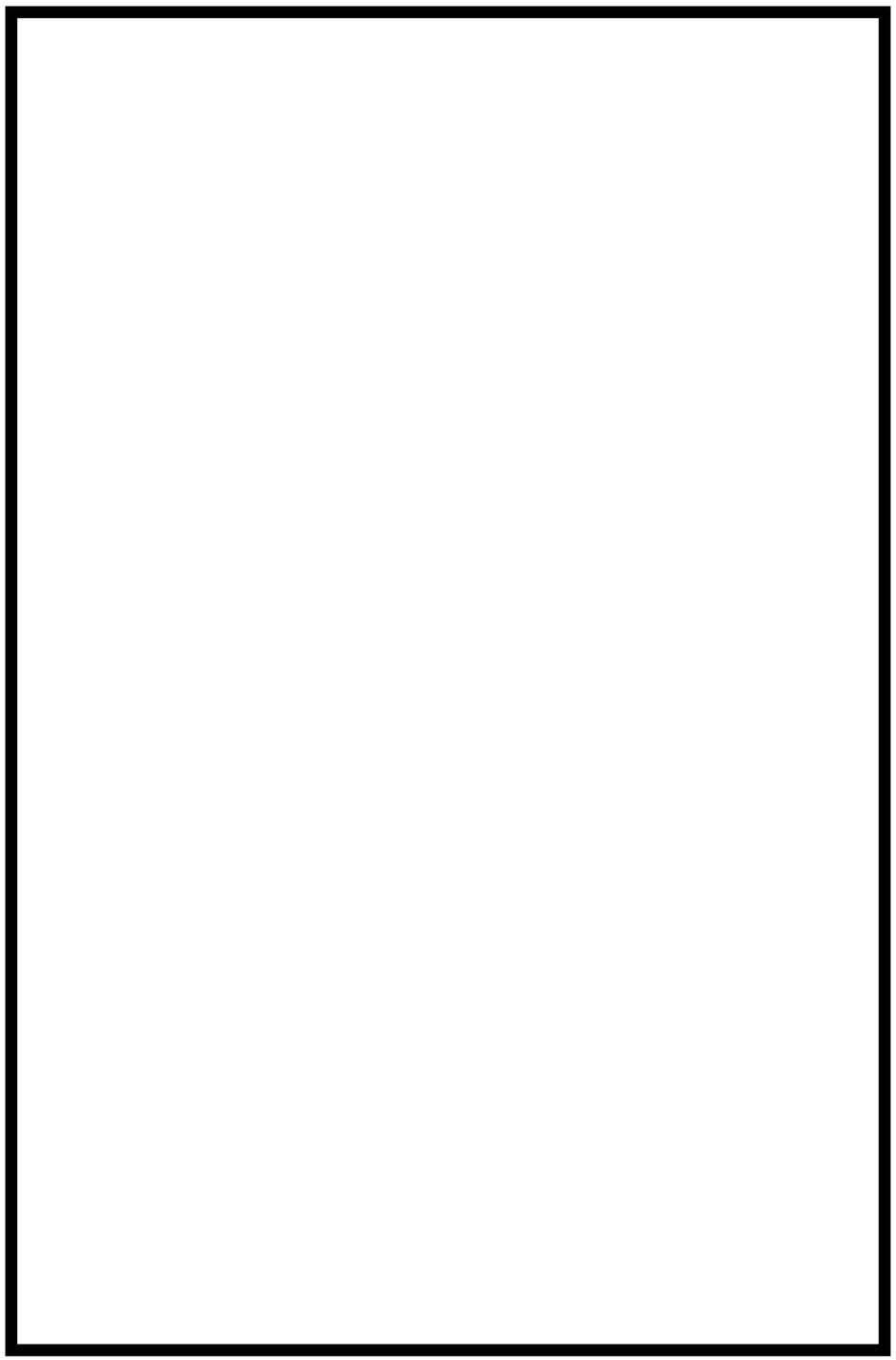
接下來的
設定
都要做動畫

```
d3.selectAll("rect")
  .attr({width: 0})
  .transition()
  .attr({width: 100})
```

分開寫也行

```
d3.selectAll("rect")
  .attr({width: 0})
d3.selectAll("rect")
  .transition()
  .attr({width: 100})
```

練習 #04 畫一個長出來的長條圖



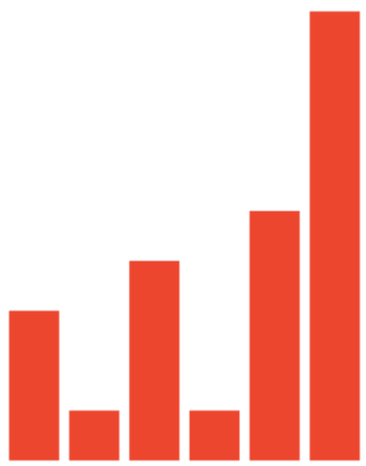
加分題1

```
transition()
  .duration(ms)
  .delay(ms)
```

可以改變動畫長度跟延遲,
試著用用看吧！

(單位是千分之一秒)

加分題2: 從下面往上長



延遲與動畫長度

```
transition()
  .duration(ms)
  .delay(ms)
```

Good Pattern

Binding 與 Styling

```
d3.selectAll(“rect”)
   .data(data)
   .enter()
   .append(“rect”)
   .attr({
      class: “…”
   });
```

只做初始設定

```
d3.selectAll(“rect”)
   .attr({
      …
   }).style({
      …
   })
```

要分開

Good Pattern

Binding

```
function bind(data) {
  d3.selectAll(“rect”)
    .data(data)
    .enter()
    .append(“rect”)
    .attr({
      class: “…”
    });
}
```

Styling

```
function render() {
    d3.selectAll(“rect”)
      .attr({
          …
      }).style({
        …
      })
}
```

包裝起來，可重複使用

Good Pattern

Binding

```
function bind(data) {
  d3.selectAll(“rect”)
    .data(data)
    .enter()
    .append(“rect”)
    .attr({
      class: “…”
    });
}
```

Styling

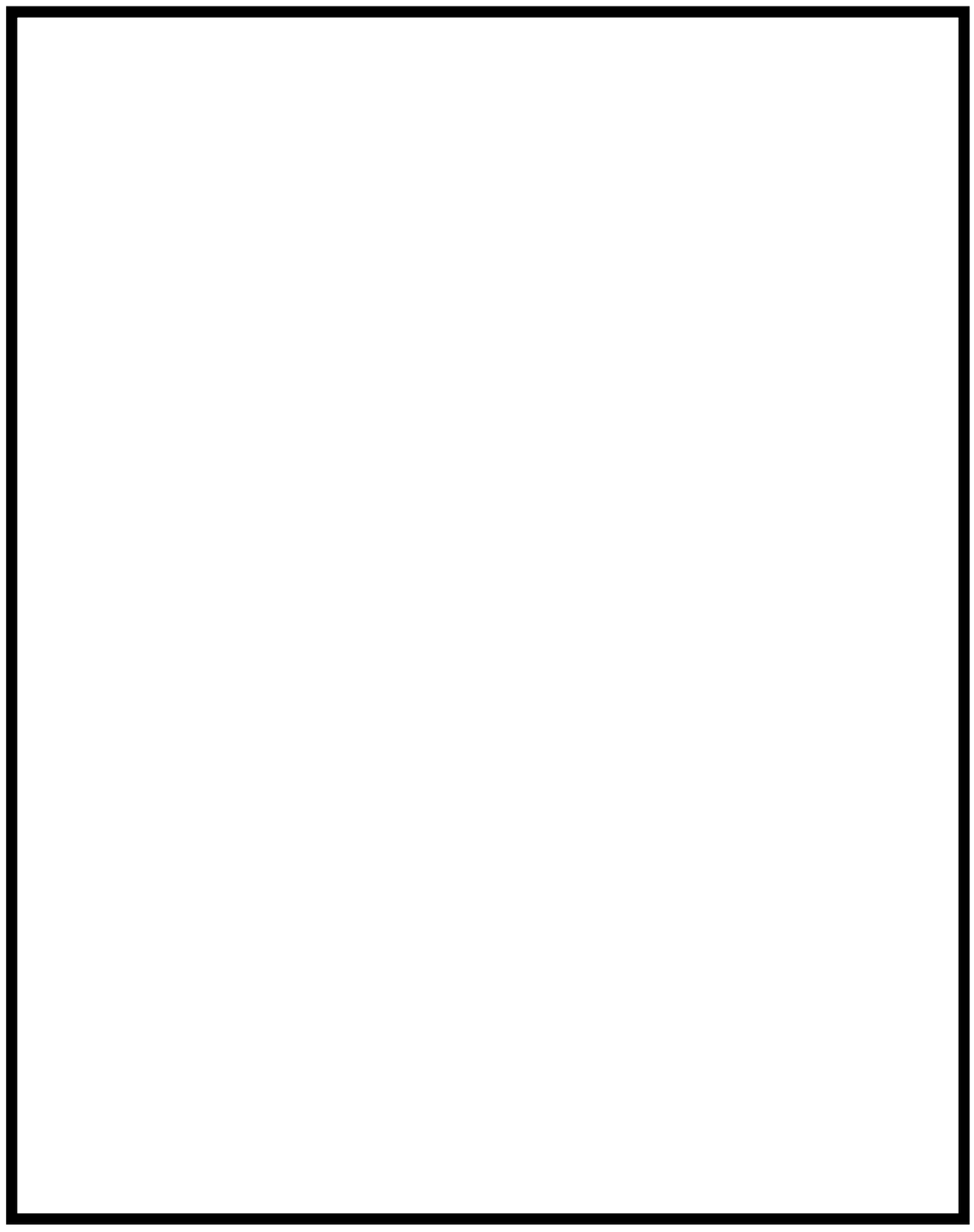
```
function render(delay) {
    d3.selectAll(“rect”)
      .transition()
      .delay(delay)
      .attr({
          …
      }).style({
        …
      })
}
```

為了動畫，帶個延遲參數

Rendering

```
bind([3,1,4,1,5,9]);
render(0);
bind([9,5,1,4,1,3]);
render(1000);
```

練習 #05 畫一個長條圖，從 [3,1,4,1,5,9] 變形到 [9,5,1,4,1,3]



加分題1
一開始似乎閃了一下黑色
你能不能讓他一開始就是紅色呢？

加分題2
讓圖表在兩組資料間不斷來回變形

```
function bind(data) {
  var update = d3.select("svg").
    selectAll("rect").data(data);
  update.enter().append("rect");
  update.exit().remove();
}

function render(delay) {
  d3.select("svg").selectAll("rect")
    .transition().delay(delay).attr({
      x: function(d,i) { return i * 12; },
      y: 10, width: 10,
      height: function(d,i) { return d * 10; },
      fill: "red"
    });
}

bind([3,1,4,1,5,9]);
render(0);
bind([9,5,1,4,1,3]);
render(1000);
```

#05 解答

```
function bind(data) {
  var update = d3.select("svg").
    selectAll("rect").data(data);
  update.enter().append("rect")
    .attr({/* 初始化設定 */});
  update.exit().remove();
}
function render(delay) {
  d3.select("svg").selectAll("rect").attr({
    /* 不想做動畫的設定 */
  });
  d3.select("svg").selectAll("rect")
    .transition().delay(delay).attr({
      x: function(d,i) { return i * 12; },
      y: 10, width: 10,
      height: function(d,i) { return d * 10; },
      fill: "red"
    });
}
bind([3,1,4,1,5,9]);
render(0);
bind([9,5,1,4,1,3]);
render(1000);
```

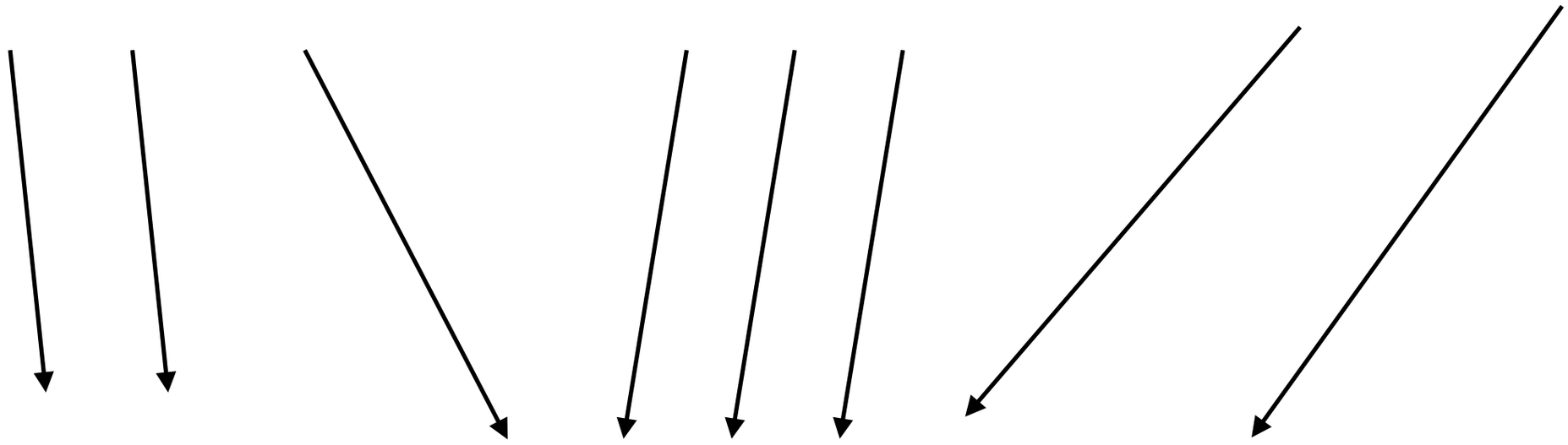
#05 加分題

Quiz

字串參數可以做動畫嗎？

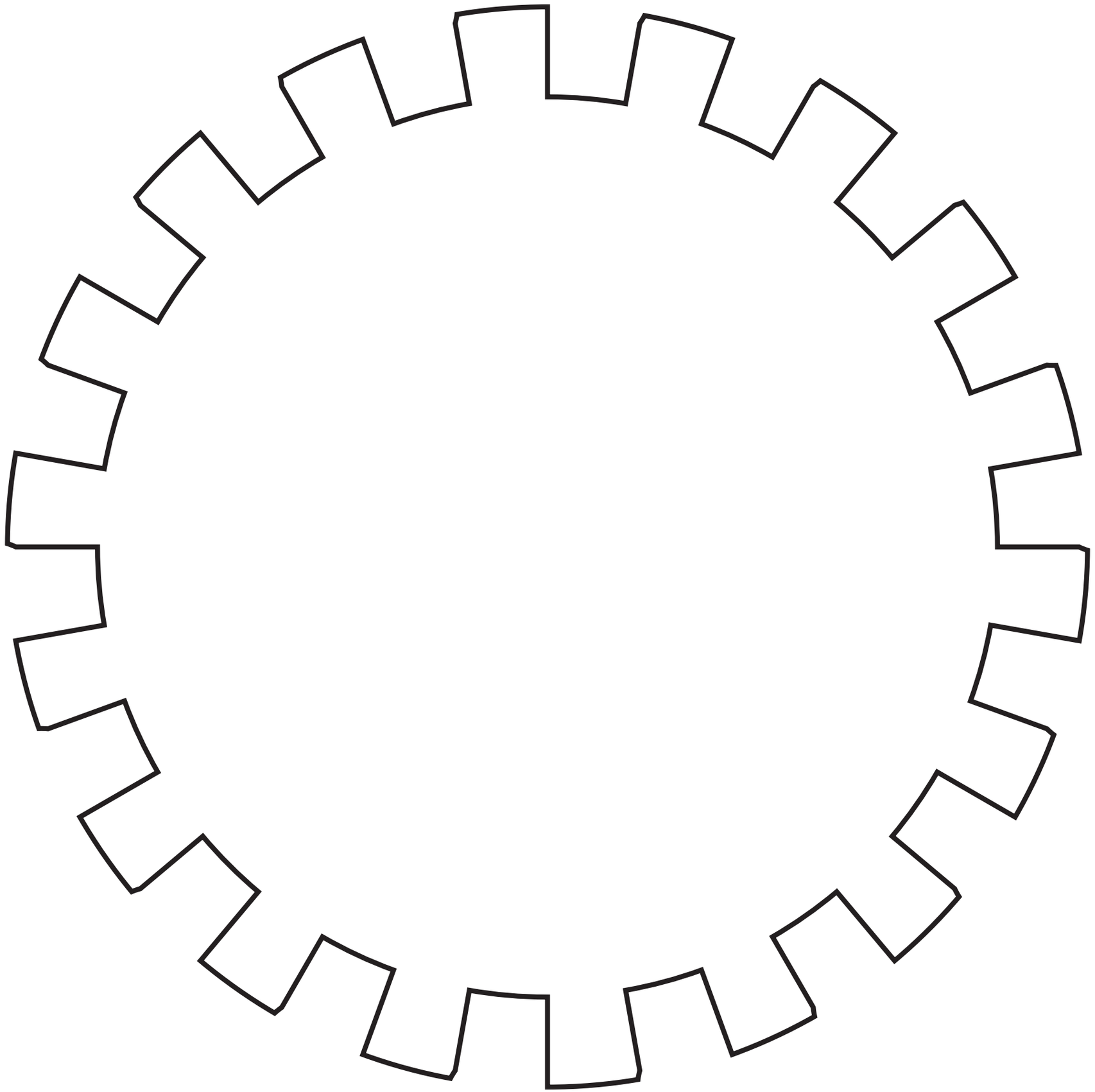
```
d3.selectAll("rect")
  .attr("fill", "red")
  .transition().duration(100)
  .attr("fill", "blue");
```

1鄉2里共3夫子，不識4書5經6義，竟敢教789子，10分大膽



10室9貧，湊得8兩7錢6分5毫4厘，尚且3心2意，1等下流

1.94790193749999997室2.7372570625貧，湊得3.526612187499999 6兩4.3159673125錢5.1053224375分5.894677562499999毫706.3218865625厘，尚且9.2627429374999 99心2意，1等下流



<?xml version="1.0" encoding="utf-8"?> <!-- Generator: Adobe Illustrator 19.0.0, SVG Export Plug-In . SVG Version: 6.00 Build 0) --> <svg version="1.1" baseProfile="tiny" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 1024 576" xml:space="preserve"> <path id="XMLID_39_" fill="#FFFFFF" stroke="#231F20" stroke-miterlimit="10" d="M574.4,270.8c1.1-6.1,1.6-12.1,1.6-18.1 c-0.6-0.2-1.1-0.4-1.7-0.7H558c0-5.3-0.5-10.6-1.4-15.6l17.7-3.1c-1.1-6.1-2.6-11.9-4.6-17.6c-0.6,0-1.2,0-1.8,0l-15.3,5.6 c-1.8-5-4-9.7-6.6-14.2l15.6-9c-3.1-5.3-6.6-10.3-10.4-14.9c-0.6,0.2-1.1,0.4-1.7,0.6l-12.5,10.5c-3.4-4-7.1-7.7-11.1-11.1 l11.6-13.8c-4.7-4-9.7-7.5-14.8-10.5c-0.4,0.4-0.9,0.8-1.4,1.1L513,174c-4.5-2.6-9.3-4.8-14.2-6.6l6.1-16.9 c-5.8-2.1-11.7-3.7-17.5-4.8c-0.3,0.5-0.6,1.1-0.9,1.6l-2.8,16.1c-5.1-0.9-10.3-1.4-15.6-1.4v-18c-6.2,0-12.2,0.5-18.1,1.5 c-0.1,0.6-0.2,1.2-0.4,1.8l2.8,16.1c-5.2,0.9-10.3,2.3-15.2,4.1l-6.1-16.9c-5.8,2.1-11.3,4.7-16.5,7.6c0.1,0.6,0.2,1.2,0.3,1.8 L423,174c-4.5,2.6-8.8,5.7-12.8,9l-11.6-13.8c-4.7,4-9,8.3-12.9,12.8c0.3,0.5,0.6,1,0.9,1.6l12.5,10.5c-3.4,4-6.4,8.3-9,12.8 l-15.6-9c-3.1,5.3-5.7,10.8-7.7,16.4c0.5,0.4,0.9,0.8,1.4,1.2l15.3,5.6c-1.8,4.9-3.1,9.9-4.1,15.2l-17.7-3.1 c-1.1,6.1-1.6,12.1-1.6,18.1c0.6,0.2,1.1,0.4,1.7,0.7H378v0c0,5.3,0.5,10.6,1.4,15.6l-17.7,3.1c1.1,6.1,2.6,11.9,4.6,17.6 c0.6,0,1.2,0,1.8,0l15.3-5.6c1.8,5,4,9.7,6.6,14.2l-15.6,9c3.1,5.3,6.6,10.3,10.4,14.9c0.6-0.2,1.1-0.4,1.7-0.6l12.5-10.5 c3.4,4,7.1,7.7,11.1,11.1l-11.6,13.8c4.7,4,9.7,7.5,14.8,10.5c0.4-0.4,0.9-0.8,1.4-1.1L423,330c4.5,2.6,9.3,4.8,14.2,6.6l-6.1,16.9 c5.8,2.1,11.7,3.7,17.5,4.8c0.3-0.5,0.6-1.1,0.9-1.6l2.8-16.1c5.1,0.9,10.3,1.4,15.6,1.4v18c6.2,0,12.2-0.5,18.1-1.5 c0.1,0.6,0.2,1.2,0.4,1.8l-2.8-16.1c5.2-0.9,10.3-2.3,15.2-4.1l6.1,16.9c5.8-2.1,11.3-4.7,16.5-7.6c-0.1-0.6-0.2-1.2-0.3-1.8 L513,330c4.5-2.6,8.8-5.7,12.8-9l11.6,13.8c4.7-4,9-8.3,12.9-12.8c-0.3-0.5-0.6-1-0.9-1.6l-12.5-10.5c3.4-4,6.4-8.3,9-12.8l15.6,9 c3.1-5.3,5.7-10.8,7.7-16.4c-0.5-0.4-0.9-0.8-1.4-1.2l-15.3-5.6c1.8-4.9,3.1-9.9,4.1-15.2L574.4,270.8z"/></svg>

SVG Path

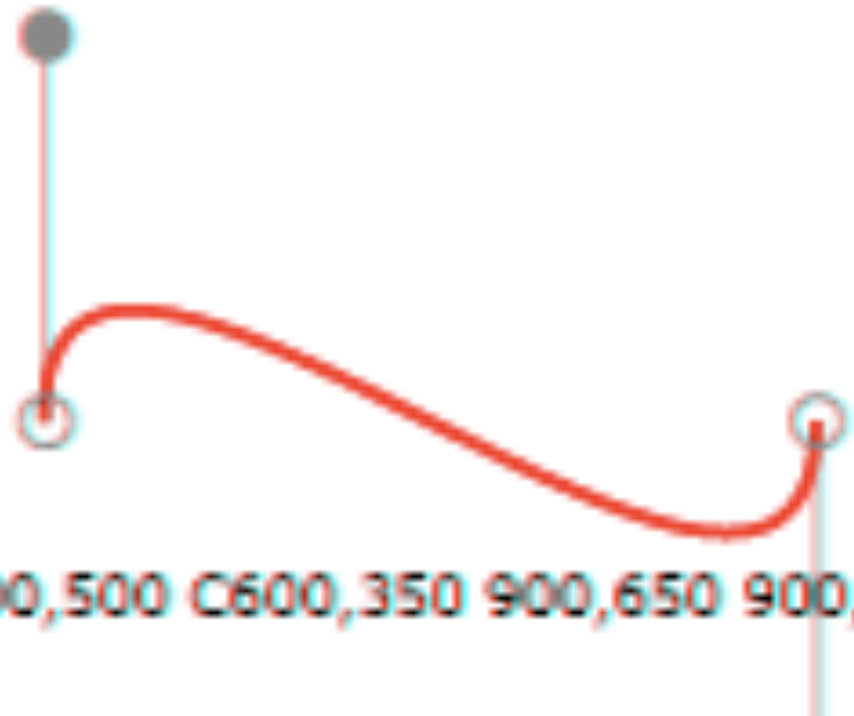
```
<path
fill="red" stroke="black"
d="M10 10L20 20"/>
```

SVG Path

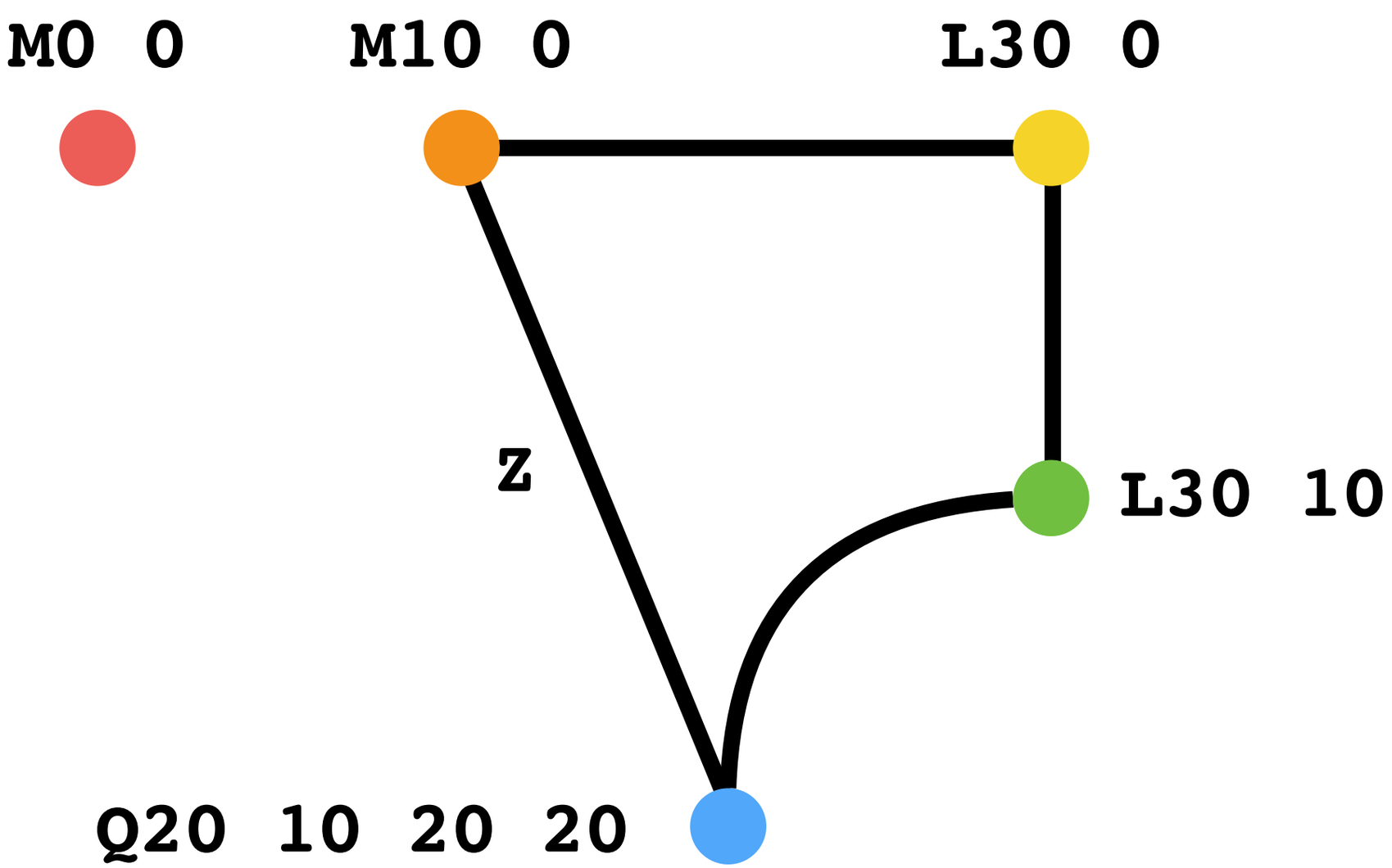
M100,200 C100,100 400,100 400,200

M600,200 C675,100 975,100 900,200

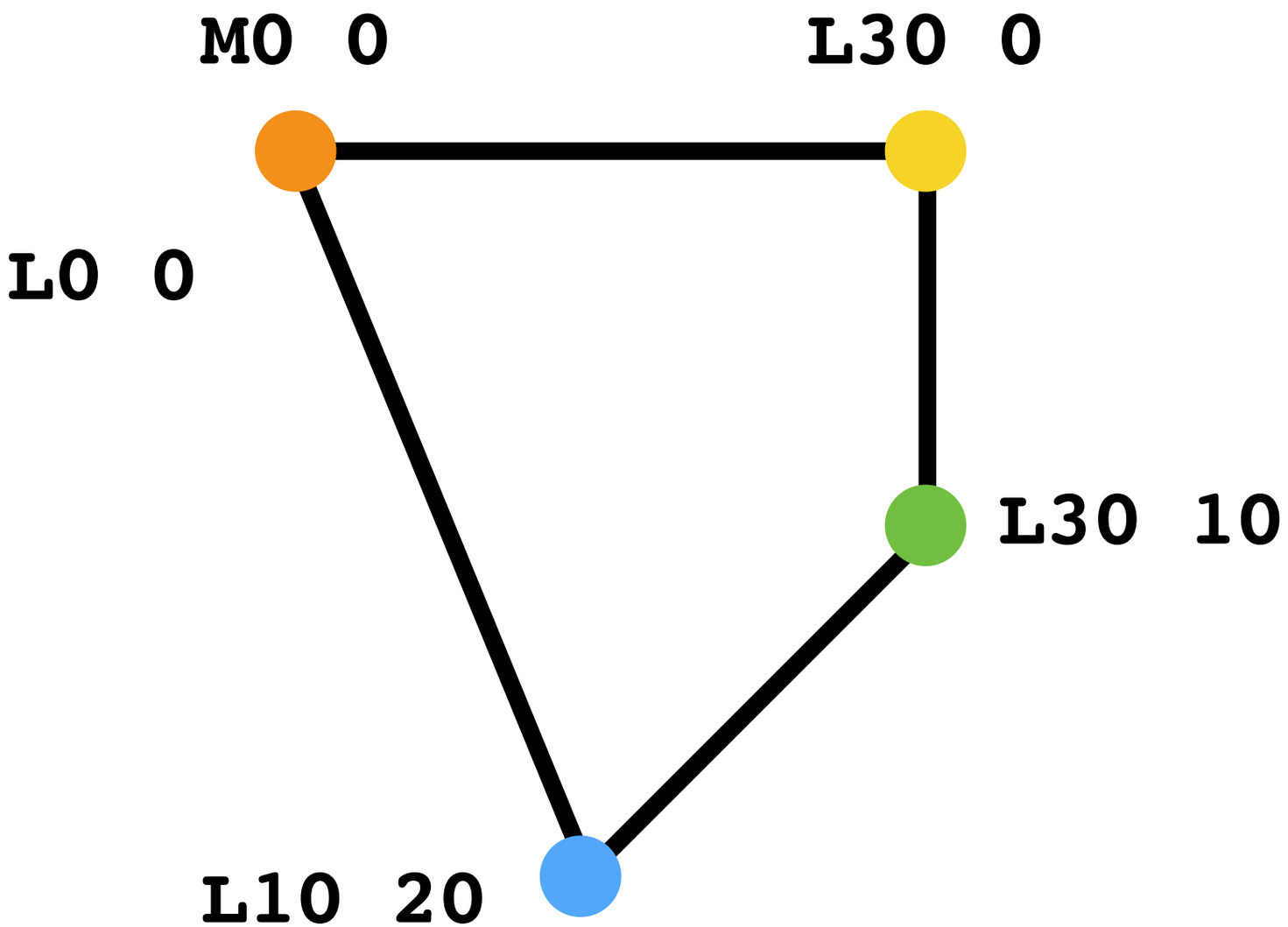
M100,500 C25,400 475,400 400,500



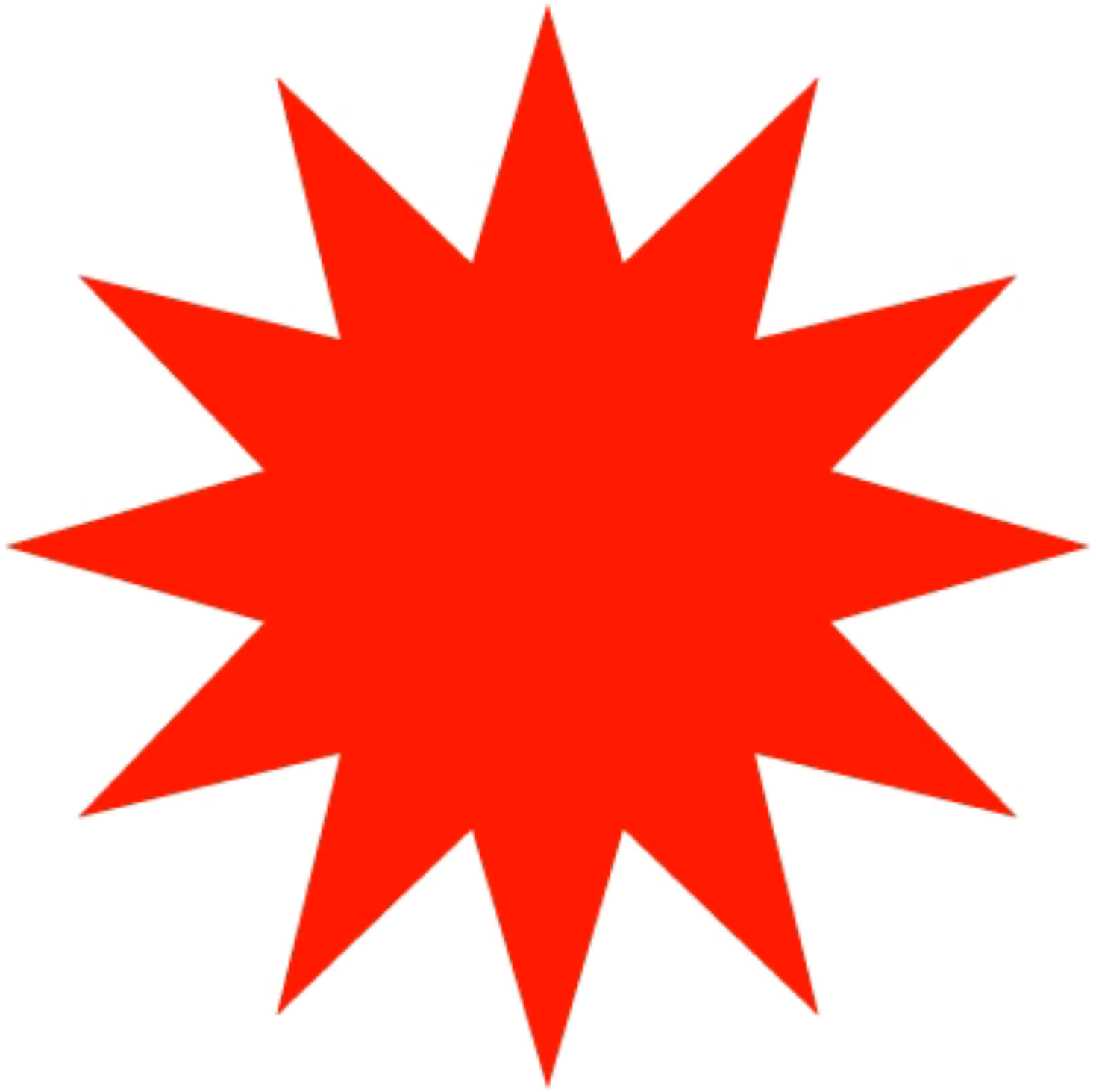
M600,500 C600,350 900,650 900,500



M0 0 M10 0 L30 0 L 30 10 Q20 10 20 20 Z

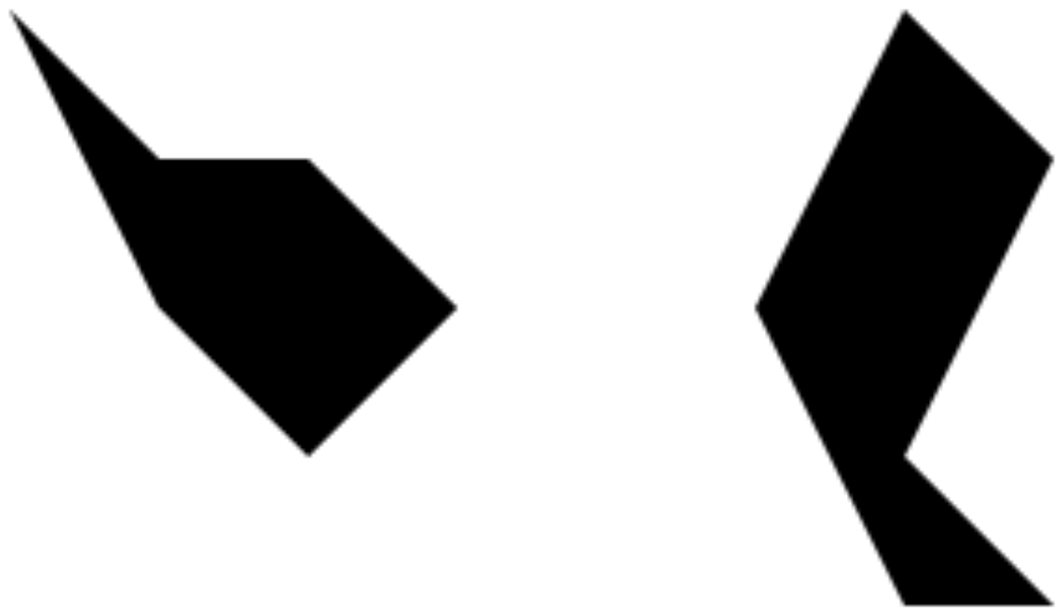



```
<path d="M0 0 L30 0 L 30 10 L10 20 L0 0"/>
```



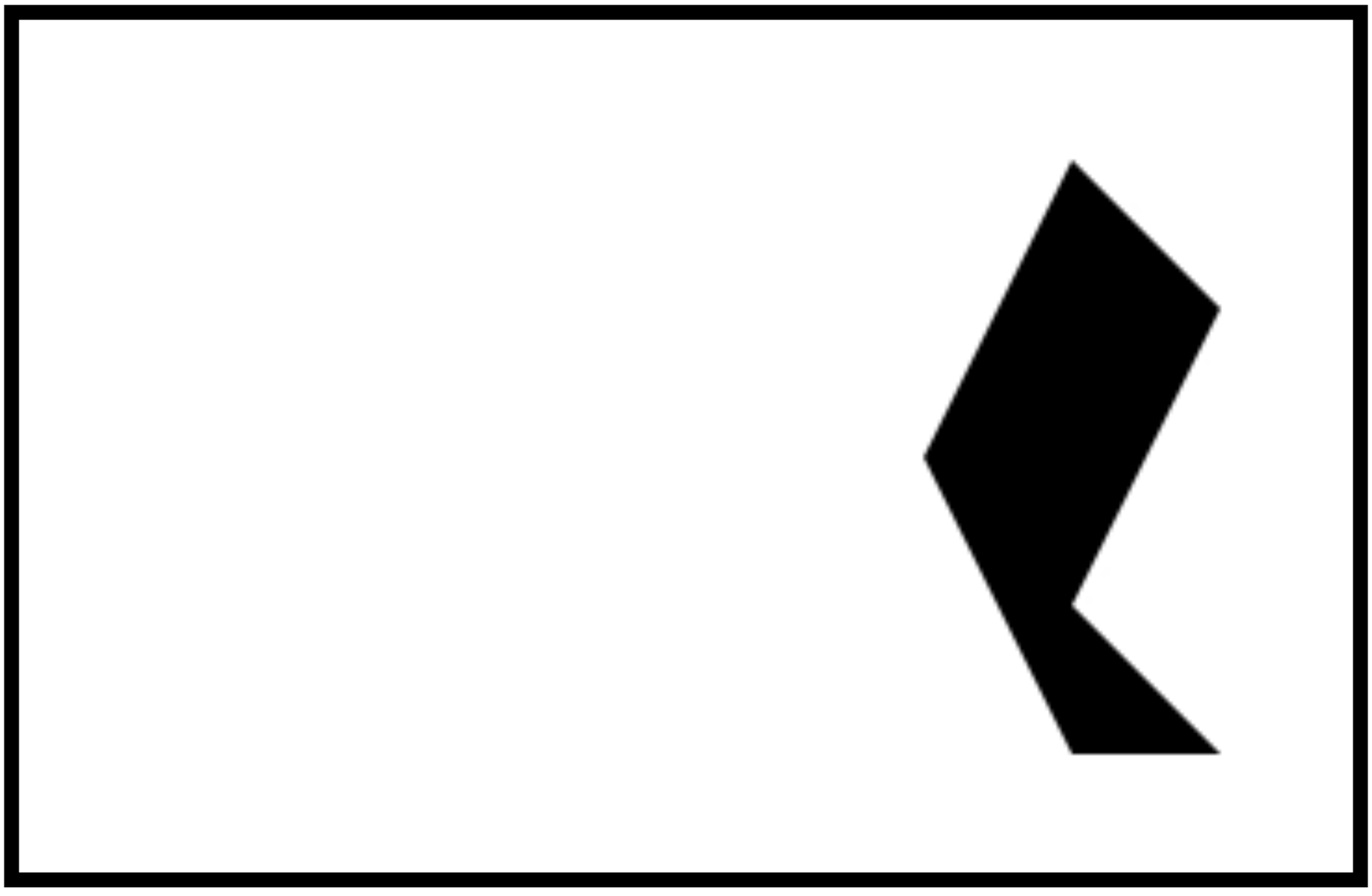
練習 #07　利用 SVG Path 隨便畫兩個不一樣的六邊形

```
<svg width="100%" height="100%">
<path d="M10 10 L20 20 L30 20 L40 30 L30 40 L20 30 L10 10"/>
<path d="M70 10 L80 20 L70 40 L80 50 L70 50 L60 30 L70 10"/>
</svg>
```



加分題: 嘗試使用 "C" 指令製作曲線線段

練習 #08 將練習 #07 中的第一個六邊形利用 D3.js 變形到第二個六邊形



提示: 建立一個 path, 先設定為第一個六邊形，再用 transition 設定為第二個六邊形

Continuous Transition

http://bl.ocks.org/mbostock/1256572

台灣歷年地震分布

統計自 1995 年起至 2015 年止
資料來源：中央氣象局

深色圈代表地震深度較深

大圈圈代表地震

921

TAIPEI
DAXI
HSINCHU
TAICHUNG
NANTOU
DOULIU
TAINAN
KAOHSIUNG
MA-KUNG
PUTIAN
Taiwan Strait

全國醫院列表

goo.gl/Ws6dkN

地址轉經緯度

- 使用批次網路服務
- 使用 Google Spreadsheet
- 利用 CartoDB 的內建服務

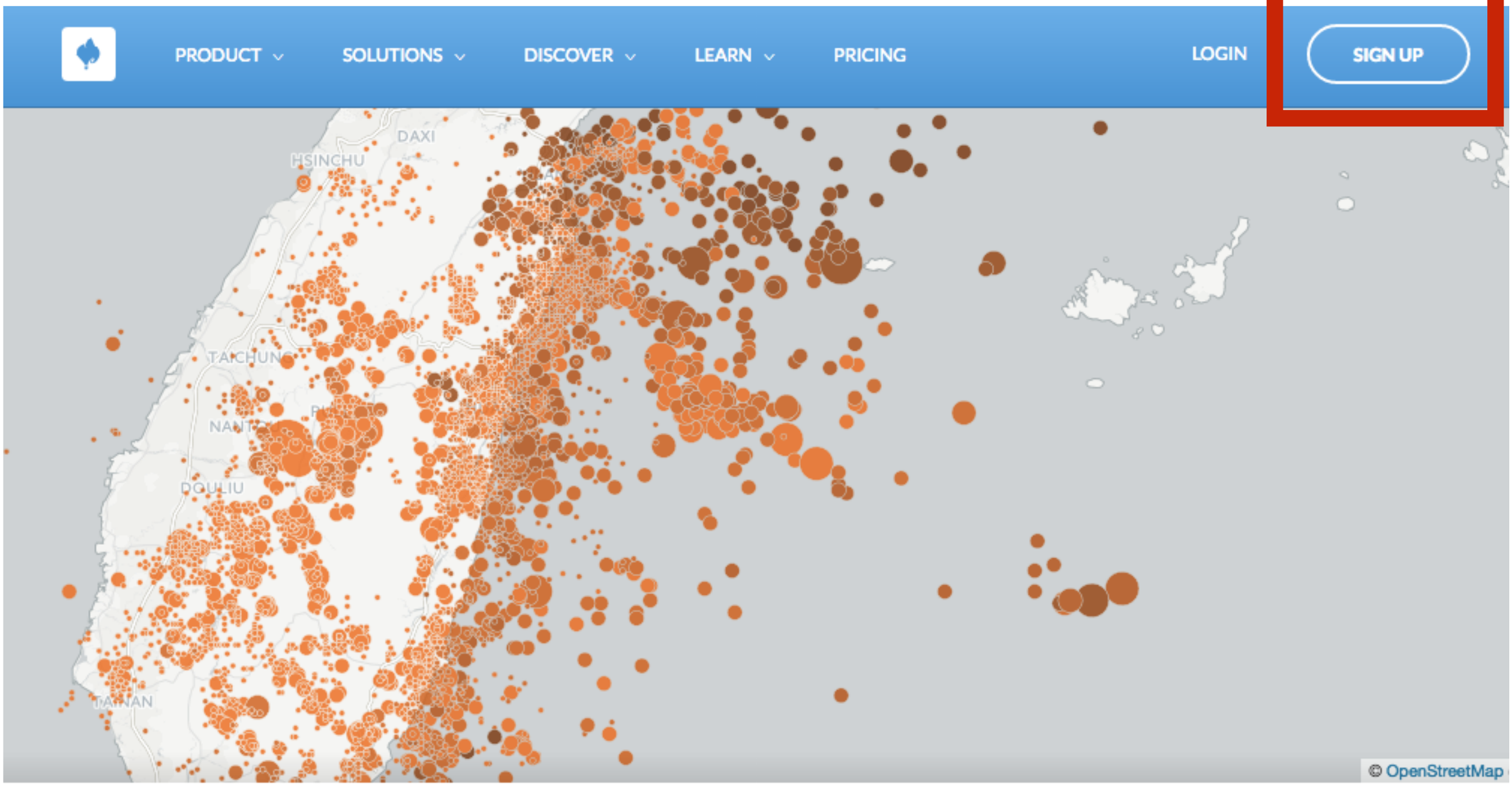
使用CartoDB 內建服務

- 每天免費 100 次

- 連至 **http://cartodb.com**

- 註冊後登入

- 匯入範例資料



DATA VISUALIZATION

INFOGRAPHICSTW 5 MAPS 9 DATASETS

tkrby / Maps

GALLERY

Your maps

Your datasets

Your locked maps

Your locked datasets

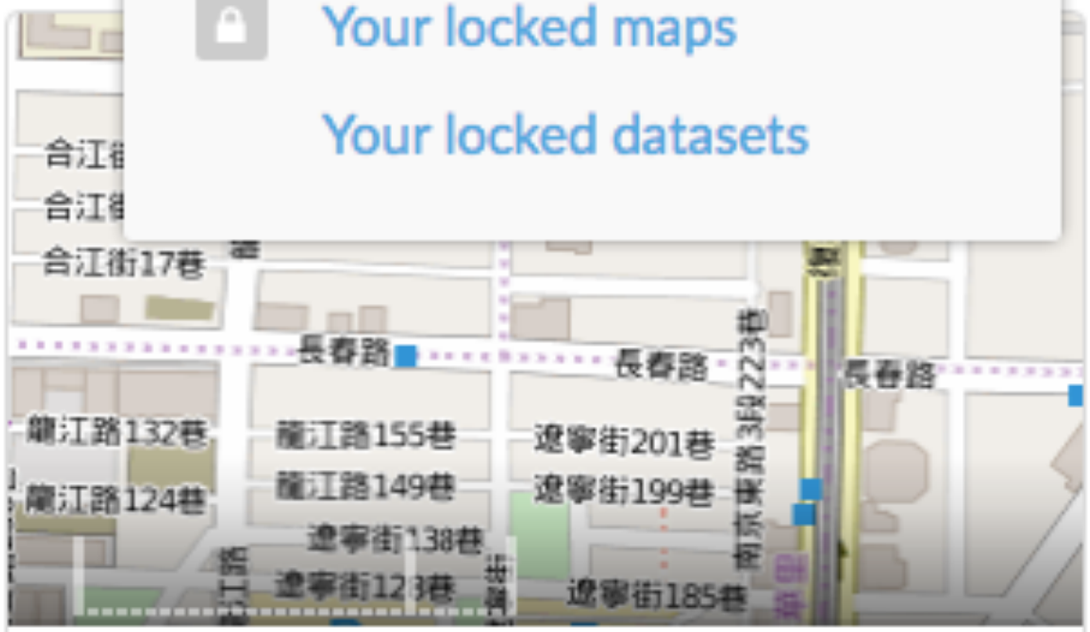


table_3 2

Add description...

Add tags...

public 5 months ago

table_3 1

Add description...

Add tags...

public 5 months ago

GALLERY DOCUMENTATION

Data library

NEW DATASET

Search | Connect dataset | Data library

Data file | Google Drive | Dropbox | box Box

Upload a file or a URL

Paste a URL or select a file like CSV, XLS, ZIP, KML, GPX, see all format

Drag & drop your file BROWSE or http://www.cartodb.com/libra

DATA VIEW | MAP VIEW

Edit | VISUA

m GEO ry	address string	cartodb_georef_status boolean	nam strin
24, 25.1209	基隆市安樂區麥金路222號	true	長庚醫
33, 25.1305	基隆市信義區信二路268號	true	行政院

- Export...
- Georeference
- Duplicate dataset...
- Merge with dataset...
- Change privacy...
- Lock dataset
- Delete this dataset...

Lon/Lat Columns | City Names | Admin. Regions | Postal Codes | IP Addresses | **Street Addresses**

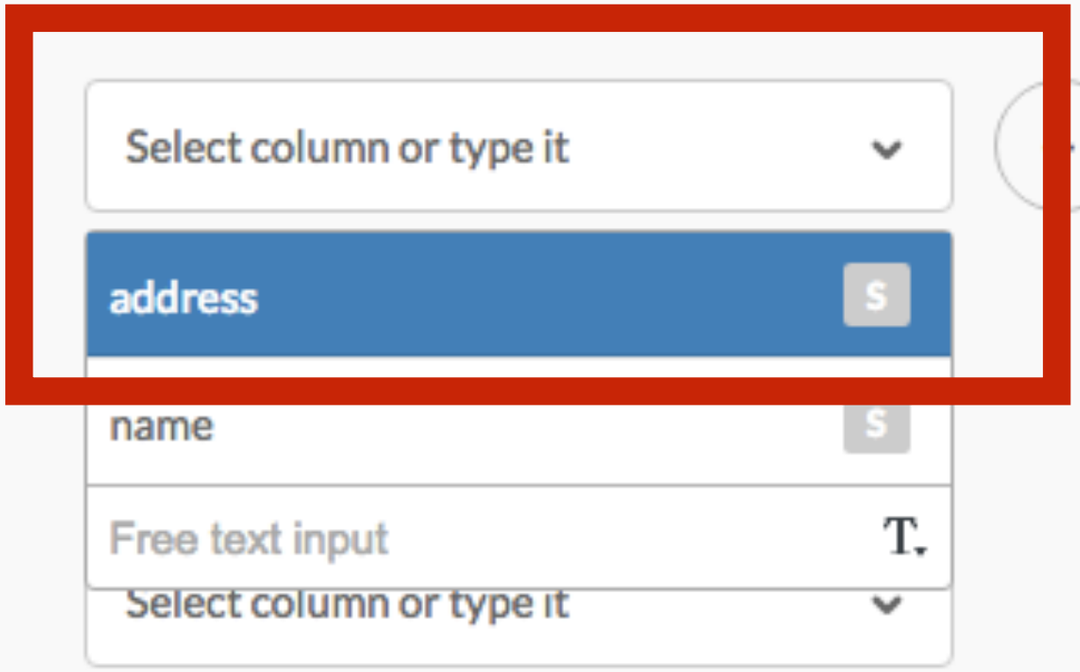
Select the column(s) that has your street address

Use this option if you need high resolution geocoding of your street adresses data.

1 Which Column Are Your Street Addresses Stored In?

2 State/Province Where Address Is Located, If Known

3 Country Where Street Address Is Located, If Known



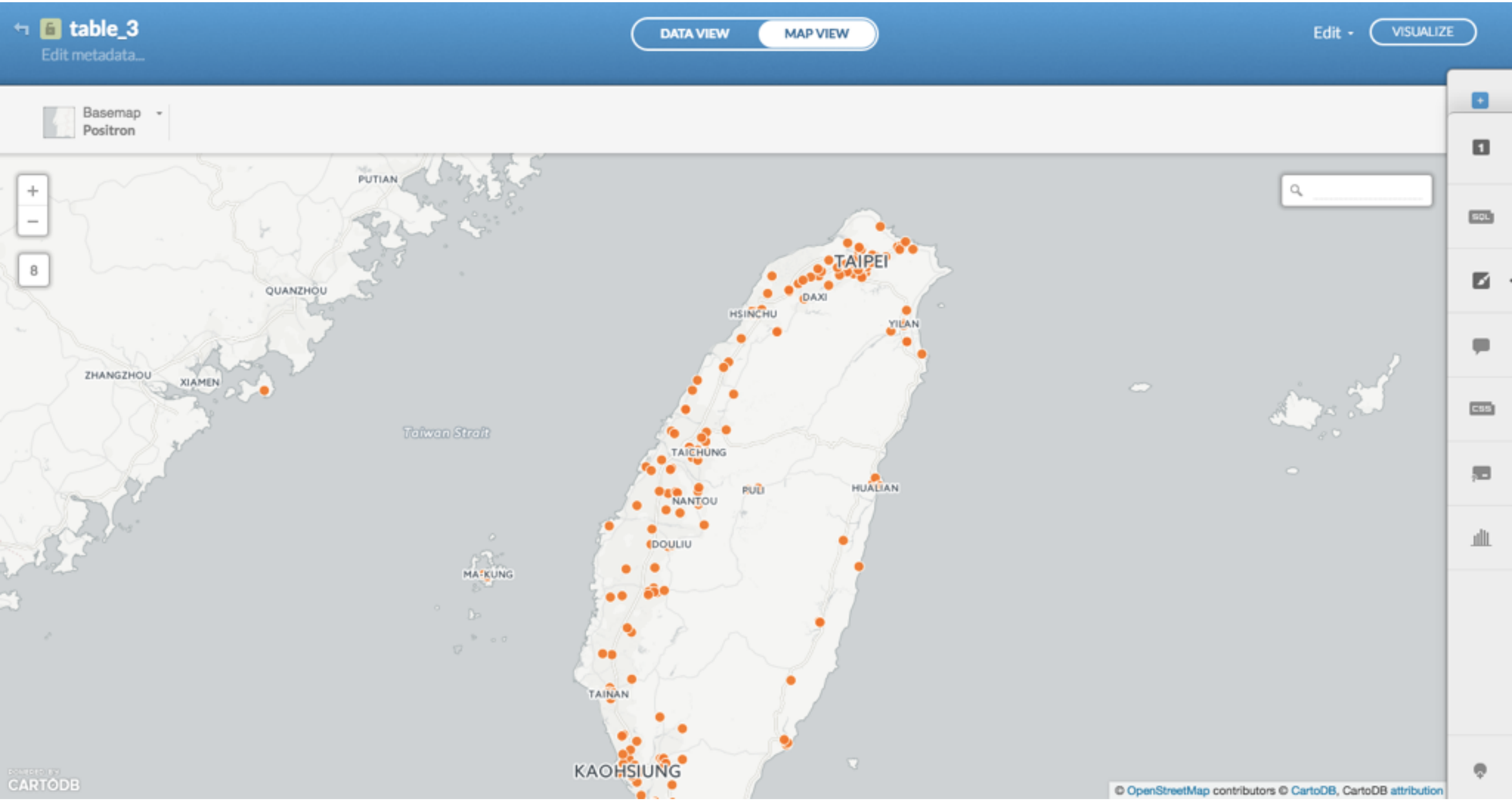


table_3
Edit metadata...
DATA VIEW
MAP VIEW
Edit
VISUALIZE
Basemap
Positron
PUTIAN
QUANZHOU
ZHANGZHOU
XIAMEN
Taiwan Strait
MA-KUNG
TAIPEI
DAXI
HSINCHU
YILAN
TAICHUNG
NANTOU
PULI
HUALIAN
DOULIU
TAINAN
KAOHSIUNG
8
POWERED BY
CARTODB
© OpenStreetMap contributors © CartoDB, CartoDB attribution

DATA VIEW | MAP VIEW

Edit | VISUALIZE

create new map

TAIPEI
DAXI
HSINCHU
YILAN
TAICHUNG
PULI
HUALIAN
NANTOU
DOULIU
MA-KUNG
TAINAN
KAOHSIUNG
Taiwan Strait

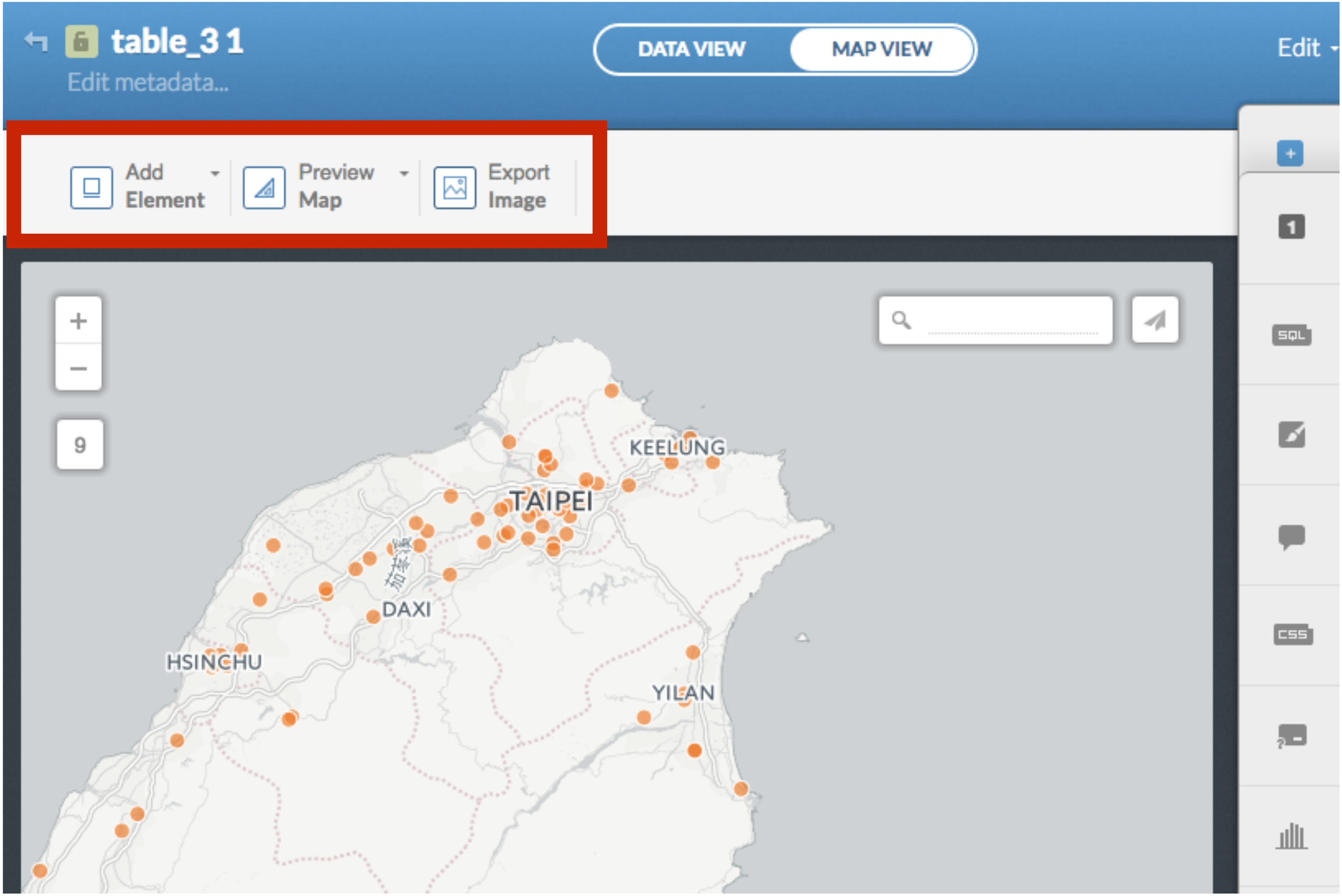
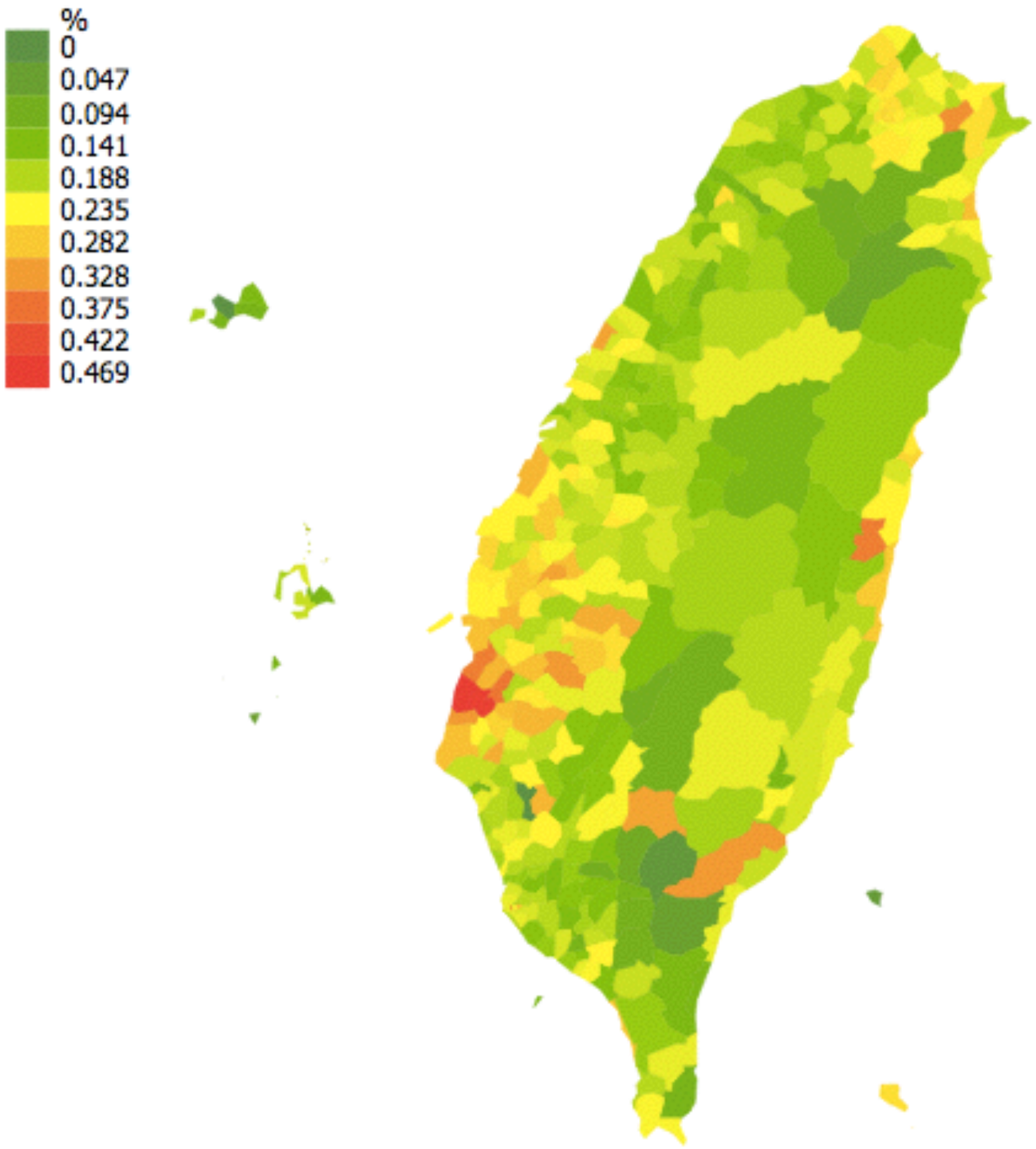


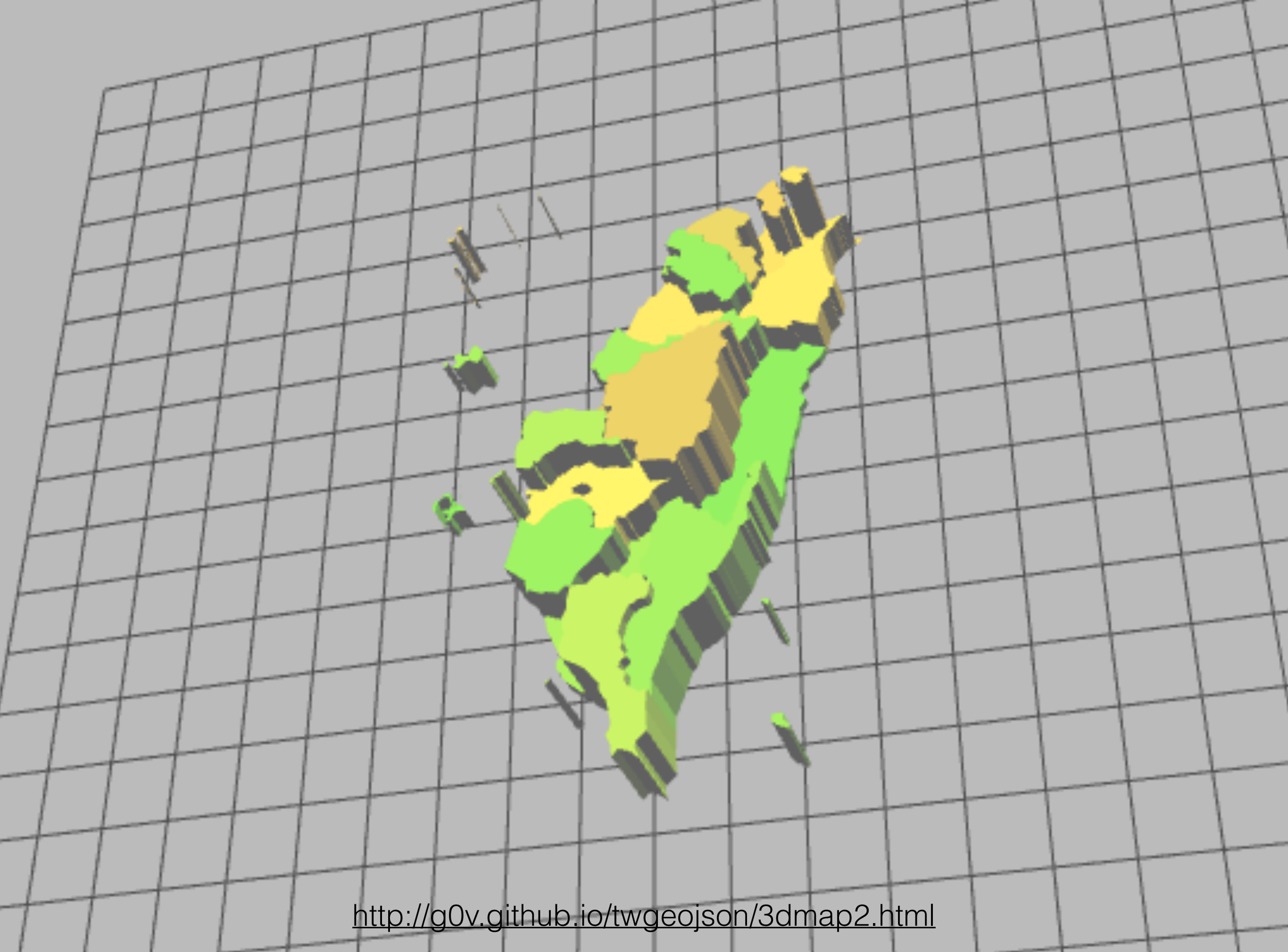
table_3 1
Edit metadata...
DATA VIEW
MAP VIEW
Edit
Add Element
Preview Map
Export Image
9
KEELUNG
TAIPEI
DAXI
HSINCHU
YILAN
SQL
CSS

鄉鎮級台灣癌症地圖



面量圖 Choropleth Map

- 需要 geojson 格式地理區塊檔
- g0v 是你的好朋友
  http://github.com/g0v/twgeojson



http://g0v.github.io/twgeojson/3dmap2.html

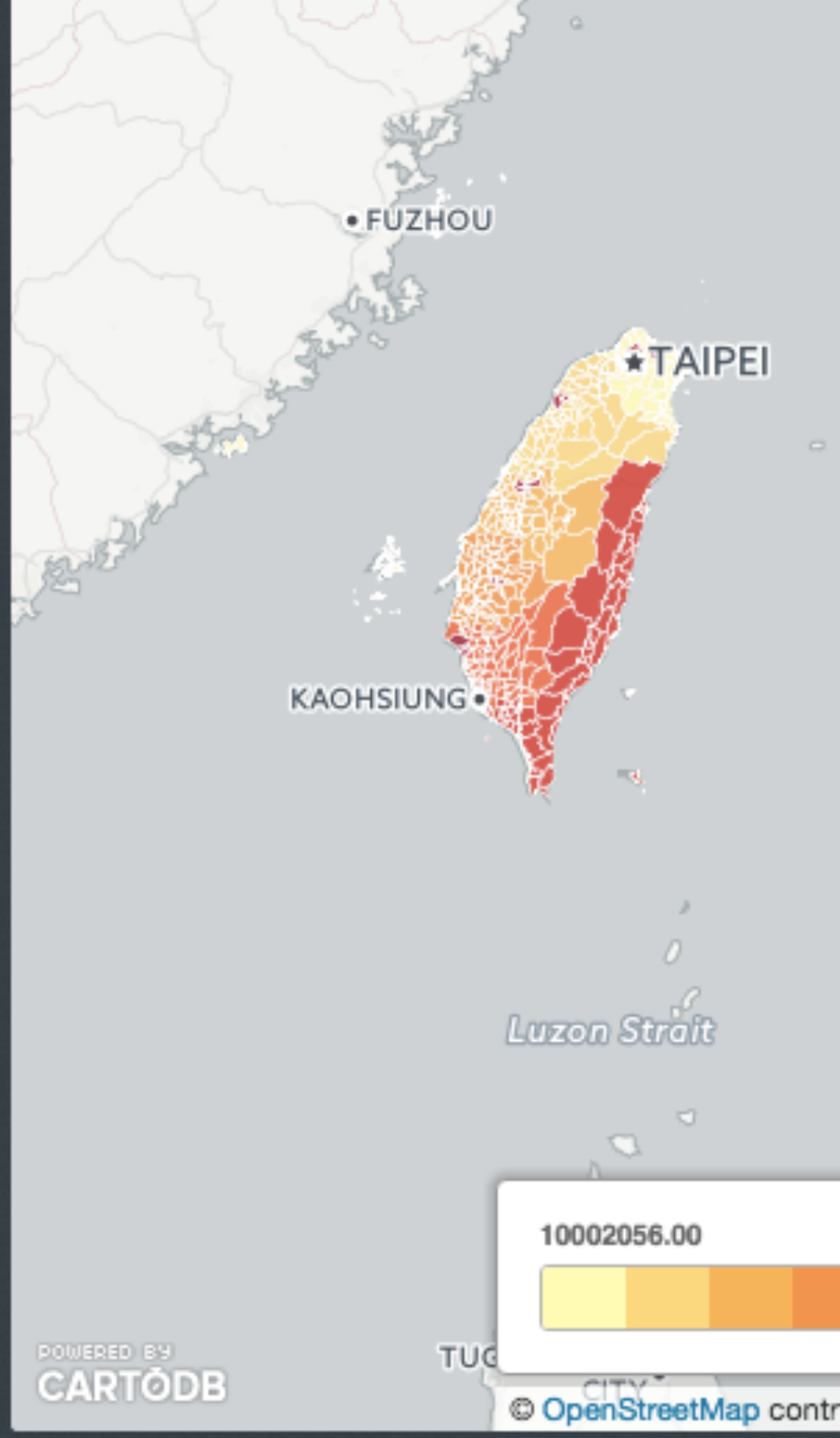
利用現成資料

https://tkrby.cartodb.com/tables/county

或

goo.gl/l15qlr

市/旗山區	GeoJSON	1001219	旗山區	10012011	null
市/東區	GeoJSON	1002001	東區	10020001	null
市/六甲區	GeoJSON	1001109	六甲區	10011010	null
縣/西湖鄉	GeoJSON	1000514	西湖鄉	10005009	null
市/東區	GeoJSON	1001801	東區	10018002	null
市/官田區	GeoJSON	1001110	官田區	10011012	null
縣/彰化市	GeoJSON	1000701	彰化市	10007004	null
縣/獅子鄉	GeoJSON	1001332	獅子鄉	10013029	null
縣/湖口鄉	GeoJSON	1000405	湖口鄉	10004002	null
縣/內埔鄉	GeoJSON	1001313	內埔鄉	10013010	null



DATASETS

API CALL

DOWNLOAD

CREATE MAP

製作資料對應 (1)

- 切換至 Data View
- 點擊右方「SQL」鈕
- SELECT * FROM cartodb_query_1 改為
  SELECT **name** FROM cartodb_query_1
- 右下角 Apply query
- 點擊右上角「Edit → Export Layer」，選 CSV

製作資料對應 (2)

- 匯入 Google Spreadsheet
- 利用 left 函式製作醫院的分區資訊
- 利用 unique + counta 計算各分區醫院數
- 利用 filter 對應地理區塊地名與醫院分區表
- 匯出！

製作資料對應 (3)

- 回到 CartoDB，進入 dataset dashboard
- 點選稍早複製的鄉鎮資料集
- 切換至 Data view ，選擇右下角的 merge dataset
- Column Join, 利用「name」欄位合併

面量圖 Choropleth Map

- 進入 Map View，在 Wizard 選「Choropleth」
- Column 選稍早建立的值欄位
- add layer to Create Map
- Publish!