前端視覺化實戰 - D3.js

Kuro Hsu @ TWMVC

tw MVC

About me

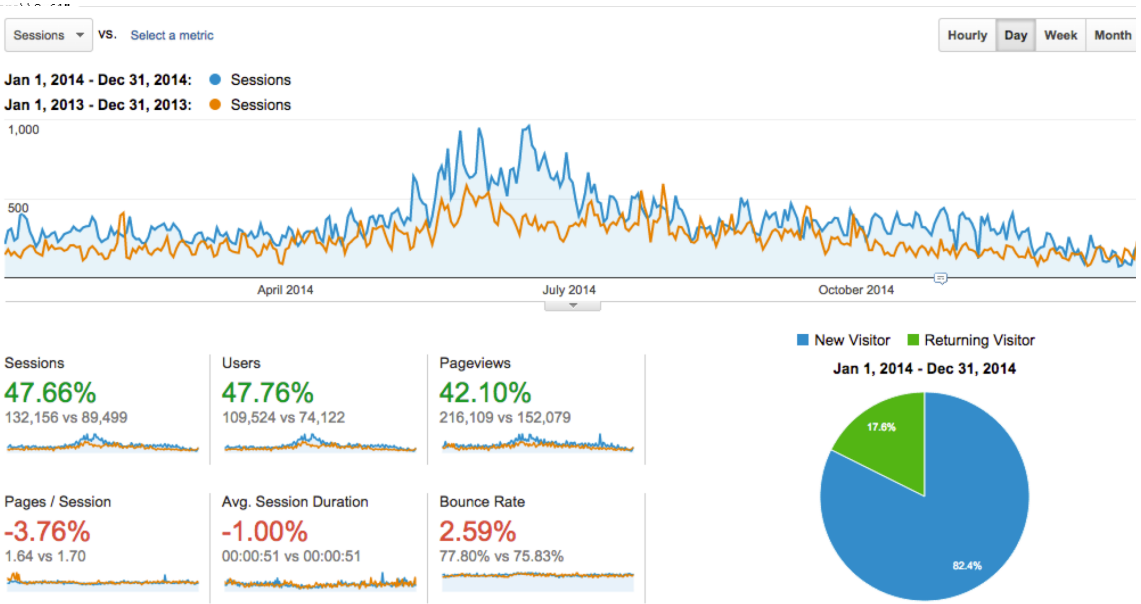


Kuro Hsu

- 前端工程師 @ 永慶房產集團
- kurotanshi [at] gmail.com
- http://kuro.tw

同樣是記錄，哪個易懂？

```
18525 24 - - [10/May/2010:15:24:54 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18526 12 - - [10/May/2010:15:24:55 -0500] "GET /wp-content/plugins/cforms/js/include/picsof.exe HTTP/1.1" 404 8816 "-" "Opera\\9.64"
18527 99 - - [10/May/2010:15:24:56 -0500] "GET /wp-content/plugins/cforms/js/include/picsof.exe HTTP/1.1" 404 8788 "-" "Opera\\9.64"
18528 72 - - [10/May/2010:15:24:56 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 8796 "-" "Opera\\9.64"
18529 24 - - [10/May/2010:15:24:56 -0500] "GET /wp-content/plugins/cforms/js/include/picsof.exe HTTP/1.1" 404 9012 "-" "Opera\\9.64"
18530 99 - - [10/May/2010:15:24:56 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18531 72 - - [10/May/2010:15:24:56 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18532 24 - - [10/May/2010:15:24:57 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18533 21 - - [10/May/2010:15:24:57 -0500] "POST /wp-content/plugins/cforms/js/include/installed.php HTTP/1.1" 200 5249 "http:/
18534 70 - - [10/May/2010:15:24:58 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 9013 "-" "Opera\\9.64"
18535 71 - - [10/May/2010:15:24:58 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 9013 "-" "Opera\\9.64"
18536 12 - - [10/May/2010:15:24:58 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 9013 "-" "Opera\\9.64"
18537 70 - - [10/May/2010:15:24:58 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18538 12 - - [10/May/2010:15:24:58 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18539 71 - - [10/May/2010:15:24:58 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18540 12 - - [10/May/2010:15:24:59 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 9013 "-" "Opera\\9.64"
18541 12 - - [10/May/2010:15:24:59 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18542 98 - - [10/May/2010:15:24:59 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 8824 "-" "Opera\\9.64"
18543 98 - - [10/May/2010:15:25:00 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18544 75 - - [10/May/2010:15:25:01 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 8817 "-" "Opera\\9.64"
18545 69 - - [10/May/2010:15:25:01 -0500] "GET /wp-content/plugins/cforms/js/include/usapart.exe HTTP/1.1" 404 8817 "-" "Opera\\9.64"
18546 75 - - [10/May/2010:15:25:01 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Opera\\9.64"
18547 69 - - [10/May/2010:15:25:01 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "C
18548 72 - - [10/May/2010:15:25:02 -0500] "GET /wp-content/plugins/cforms/js/incl
18549 72 - - [10/May/2010:15:25:02 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "
18550 98 - - [10/May/2010:15:25:03 -0500] "GET /wp-content/plugins/cforms/js/includ
18551 68 - - [10/May/2010:15:25:03 -0500] "GET /wp-content/plugins/cforms/js/inclu
18552 98 - - [10/May/2010:15:25:03 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Op
18553 67 - - [10/May/2010:15:25:03 -0500] "GET /wp-content/plugins/cforms/js/include
18554 68 - - [10/May/2010:15:25:03 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "C
18555 67 - - [10/May/2010:15:25:04 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Ope
18556 68 - - [10/May/2010:15:25:04 -0500] "GET /wp-content/plugins/cforms/js/includ
18557 68 - - [10/May/2010:15:25:05 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Op
18558 64 - - [10/May/2010:15:25:07 -0500] "GET /wp-content/plugins/cforms/js/include/
18559 24 - - [10/May/2010:15:25:07 -0500] "GET /wp-content/plugins/cforms/js/inclu
18560 64 - - [10/May/2010:15:25:07 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Oper
18561 24 - - [10/May/2010:15:25:07 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "C
18562 96 - - [10/May/2010:15:25:08 -0500] "GET /wp-content/plugins/cforms/js/include
18563 96 - - [10/May/2010:15:25:08 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Ope
18564 80 - - [10/May/2010:15:25:09 -0500] "GET /wp-content/plugins/cforms/js/incl
18565 79 - - [10/May/2010:15:25:09 -0500] "GET /wp-content/plugins/cforms/js/includ
18566 21 - - [10/May/2010:15:25:09 -0500] "POST /wp-content/plugins/cforms/js/incl
18567 79 - - [10/May/2010:15:25:10 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Op
18568 71 - - [10/May/2010:15:25:11 -0500] "GET /wp-content/plugins/cforms/js/include/
18569 71 - - [10/May/2010:15:25:11 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Oper
18570 86 - - [10/May/2010:15:25:12 -0500] "GET /wp-content/plugins/cforms/js/inclu
18571 24 - - [10/May/2010:15:25:12 -0500] "GET /wp-content/plugins/cforms/js/includ
18572 24 - - [10/May/2010:15:25:12 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Op
18573 99 - - [10/May/2010:15:25:14 -0500] "GET /wp-content/plugins/cforms/js/include
18574 24 - - [10/May/2010:15:25:14 -0500] "GET /wp-content/plugins/cforms/js/inclu
18575 99 - - [10/May/2010:15:25:14 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "Ope
18576 24 - - [10/May/2010:15:25:14 -0500] "GET /xmlrpc.php HTTP/1.1" 200 54 "-" "C
```



Data-Driven Documents



http://d3js.org/


MAJOR BROWSER
COMPATIBILTY

D3 是什麼

- D3 = Data-Driven Documents.
- 使用**網頁標準技術** (HTML / CSS / JavaScript / SVG)
- 強大的 Data-Driven 將資料對應到 DOM 上，包括元素的增刪及屬性變化
- 提供 SVG 相關繪圖工具庫，號稱 SVG 界的 jQuery
- 彈性超大但實作較其他視覺化工具繁瑣



選擇器與 DOM 操作

選擇器

JavaScript

```
document.querySelectorAll(".block")
```

jQuery

```
$(".block")
```

D3.js

```
d3.selectAll(".block")
```

選擇器

```
d3.selectAll("body .block")
```

```
d3.select("body").selectAll(".block")
```

d3 selection

```
<div class="block">block 1</div>
<div class="block">block 2</div>
<div class="block">block 3</div>

<script>
  var blocks = d3.selectAll('.block');
  console.log( blocks.empty() );
  console.log( blocks.node() );
  console.log( blocks.size() );
</script>
```

```
false
<div class="block">
3
```

方法鏈 (method chaining)

```
var box = d3.selectAll('.box');
box.style('color', '#f00');
box.text('Hello World!');
```

```
var box = d3.selectAll('.box')
            .style('color', '#f00')
            .text('Hello World!');
```

attr

```
// attr()
d3.selectAll("circle")
    .attr("cx", 50)
    .attr("cy", 50)
    .attr("r", 25)
    .style("fill", "red");
```

attr

```
// 以「物件」的方式指定屬性
d3.selectAll("circle")
   .attr({ "cx": 50, "cy": 50, "r": 25 })
   .style("fill", "red");
```

```
// 取得 circle cx 屬性的值
d3.select("circle").attr("cx");
```

property

```
d3.selectAll('.name')
  .property('value', 'Kuro');
```

```
d3.selectAll('.name')
  .property('disabled');
```

用法與 attr() 相同，但 property() 使用在**無字面指定的屬性**。
如: disabled, checked 以及 value 等。

class

```
d3.selectAll('.block')
  .classed('item', true);

d3.selectAll('.block')
  .classed('item', false);
```

```
d3.selectAll('.block')
  .classed('item big', true);
```

styles

```
d3.selectAll('.block')
  .style('color', '#f00');
```

```
d3.selectAll('.block')
  .style({
    'color': '#f00',
    'font-size': '15px'
  });
```

text & html

jQuery

```
var foo = $('.foo');
foo.text('Hello World!');
foo.html('<div class="bar">Hello</div>');
```

D3

```
var foo = d3.selectAll('.foo');
foo.text('Hello World!');
foo.html('<div class="bar">Hello</div>');
```

d3 append

```
// d3
d3.select('.box1').append('div')
        .classed('.text', true)
        .text('TWMVC');

// jquery
$('.box2').append('div')
        .addClass('.text')
        .text('TWMVC');
```

Hello
TWMVC

TWMVC

注意： D3.append 與 jQuery.append 回傳的元素不同。

d3 insert, remove

```
d3.select('.box2')
  .insert('div', '.t')
    .text('TWMVC');
```

```
d3.select('.box3 .t')
  .remove();
```

d3.js vs. jQuery

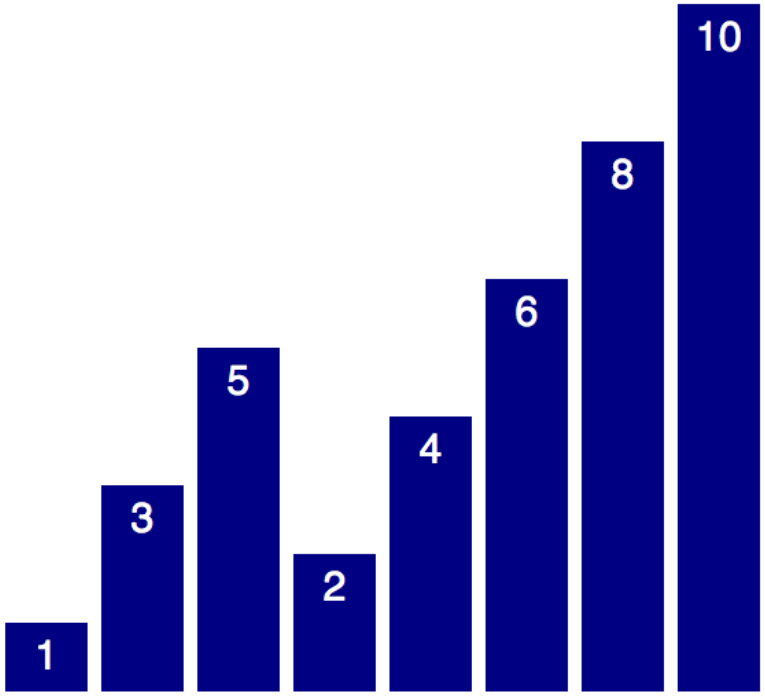
	jQuery	D3.js
Selector	`$("#id001");`	`d3.select("#id001");` `d3.selectAll(".class001");`
Attribute	`SELECTION.attr("key", "val")`	`SELECTION.attr("key", "val")`
Class	`SELECTION.addClass("…")` `SELECTION.removeClass("…")`	`SELECTION.classed("…", true)` `SELECTION.classed("…", false)`
DOM append	`SELECTION.append("<div></div>")`	`SELECTION.append("div")`



從資料到圖像元素

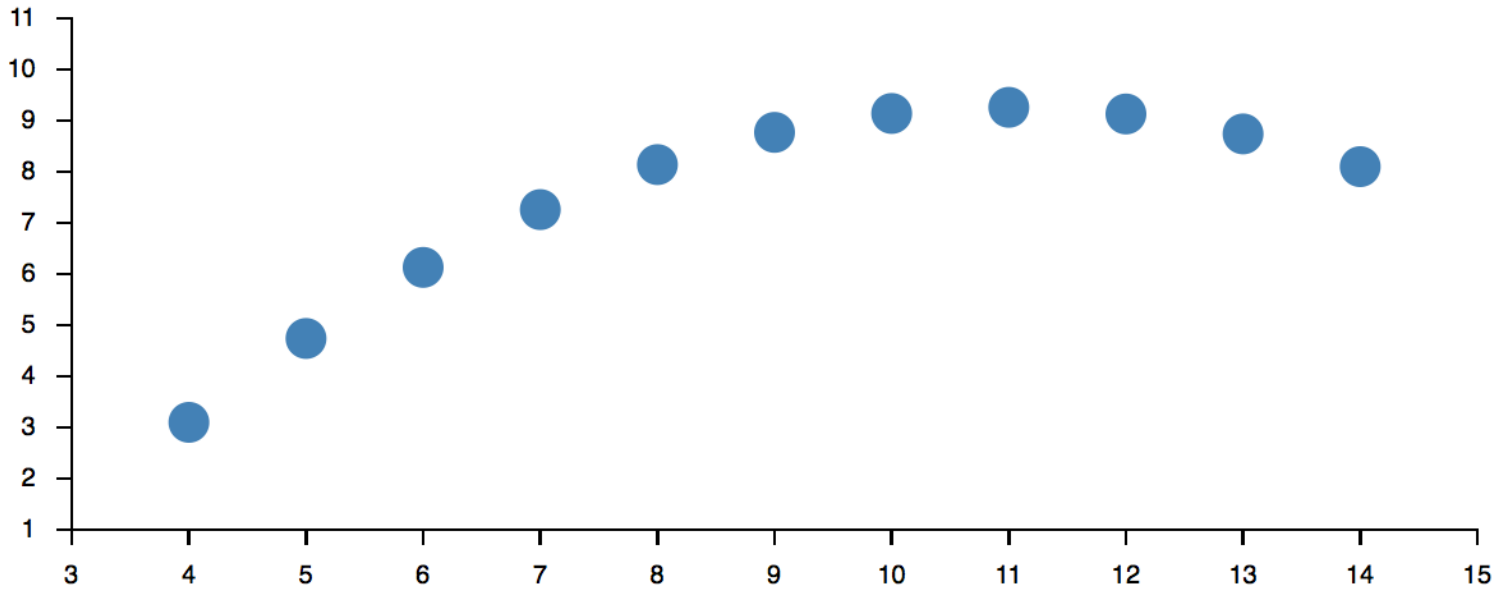
Data - 所有的資料都是陣列

```
var data = [1, 3, 5, 2, 4, 6, 8, 10];
```



Data - 所有的資料都是陣列

```
var data = [
  {x: 10.0, y: 9.14}, {x:  8.0, y: 8.14},
  {x: 13.0, y: 8.74}, {x:  9.0, y: 8.77}, ……
];
```

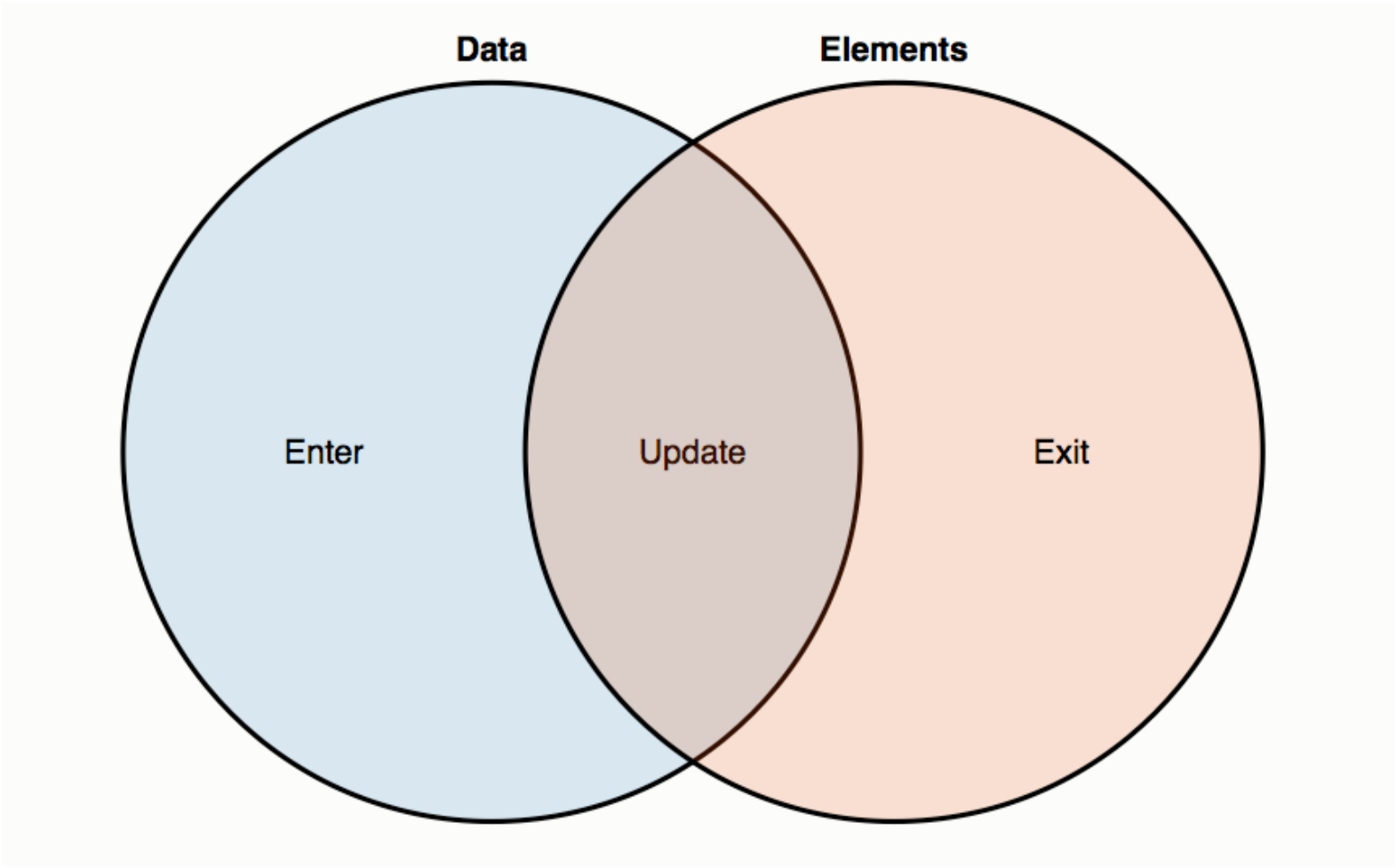


key function

```
var data = [
  { 'name': 'kuro', 'age': 30 },
  { 'name': 'John', 'age': 20 },
  { 'name': 'Mary', 'age': 18 }
];

p.data(data)
  .text(function(d, i){
    return d.name + ': ' + d.age + '歲';
  });
```

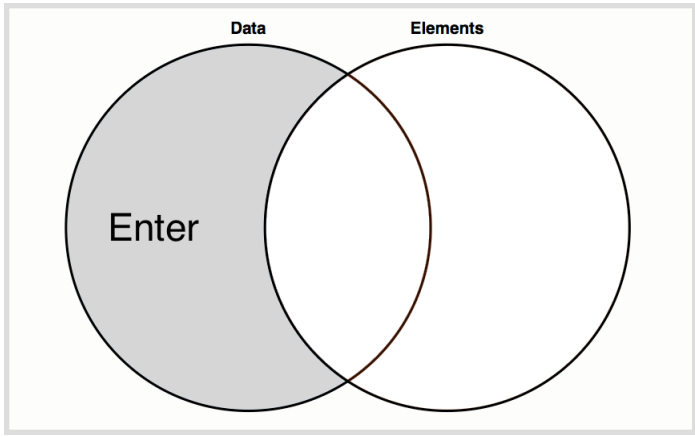
從資料到圖像元素



Enter, Update, Exit Pattern

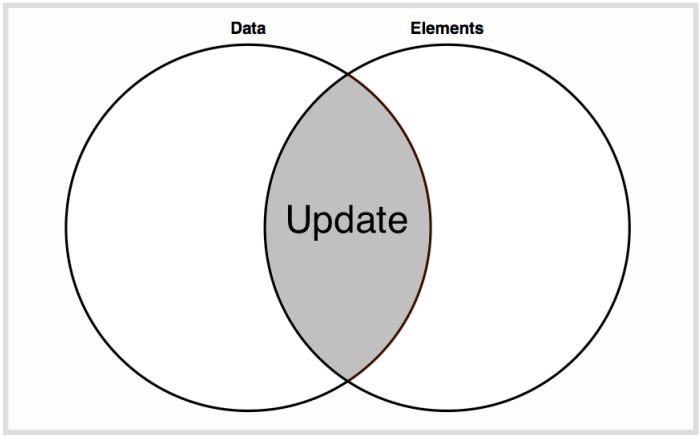
```
var data = [10, 20, 30];

d3.select('body')
    .selectAll('div')
      .data( data )
      .enter()
      .append('div')
        .text(function(d, i){ return d; });
```





Enter, Update, Exit Pattern




```
data = [1, 3, 5, 7, 9];

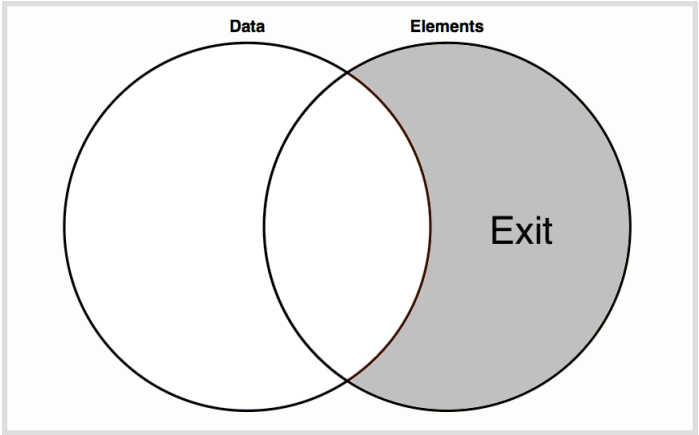
d3.select('body')
    .selectAll('div')
      .data( data )
      .text(function(d, i){ return d });
```

1	3	5	7	9

Enter, Update, Exit Pattern

```
data = [10, 20, 30];

d3.select('body')
    .selectAll('div')
      .data( data )
      .exit()
      .remove();
```

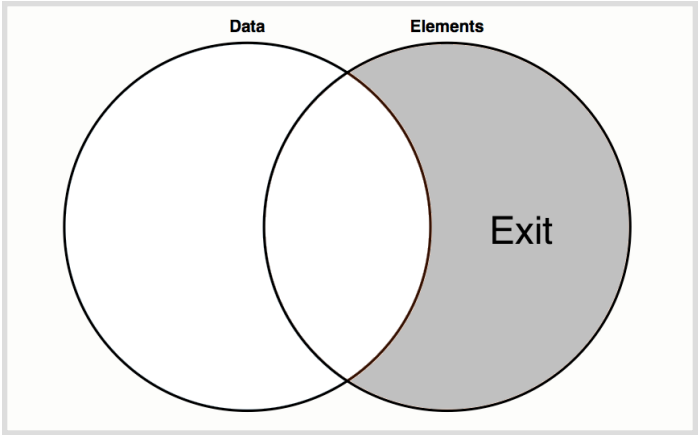


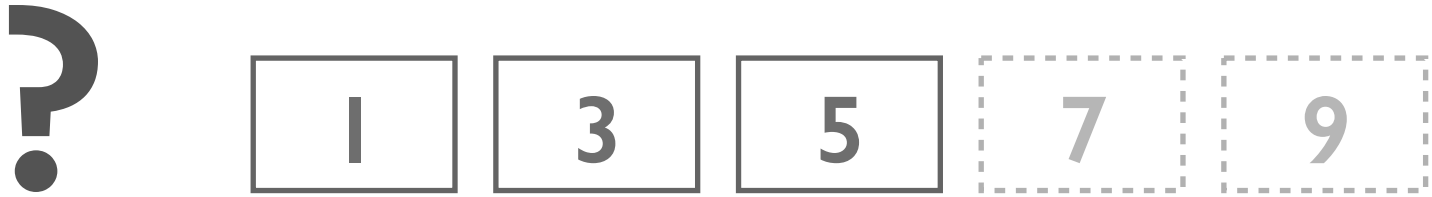


Enter, Update, Exit Pattern

```
data = [10, 20, 30];

d3.select('body')
    .selectAll('div')
      .data( data )
      .exit()
      .remove();
```

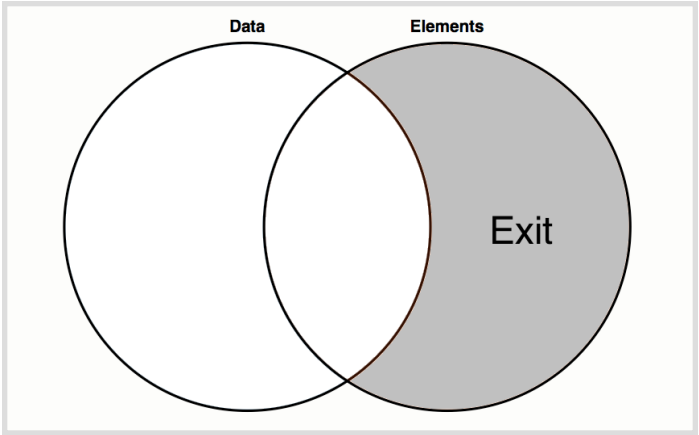




Enter, Update, Exit Pattern

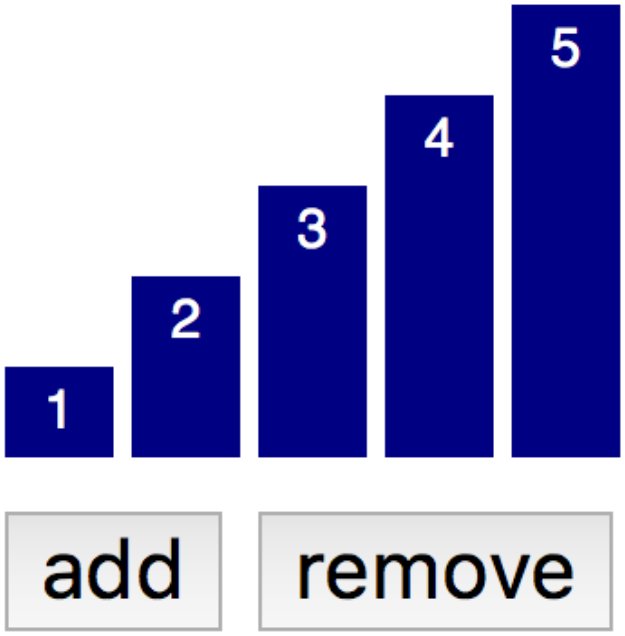
```
data = [10, 20, 30];

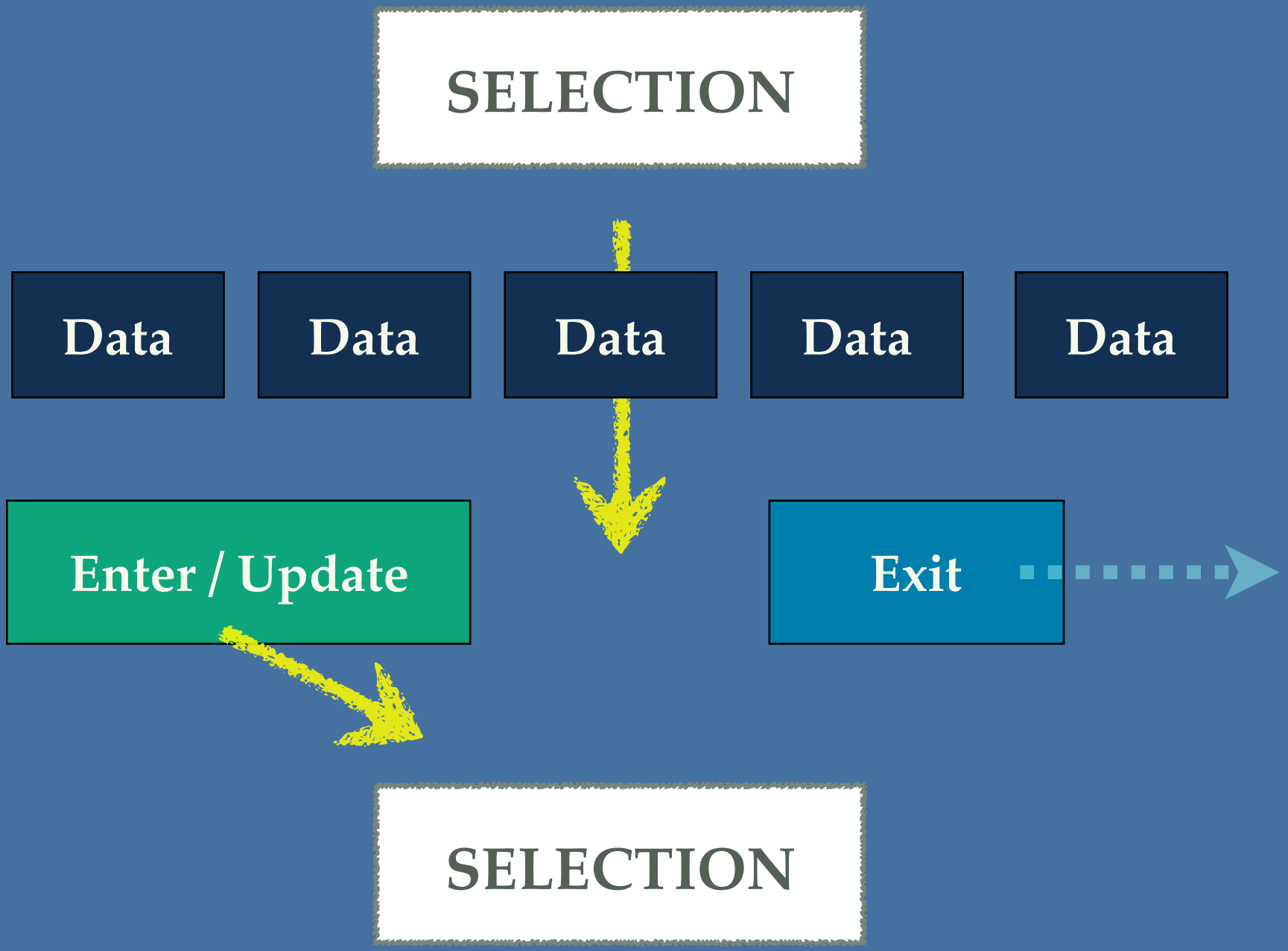
d3.select('body')
    .selectAll('div')
      .data( data )
      .text(function(d, i){
          return d;
      })
      .exit()
      .remove();
```





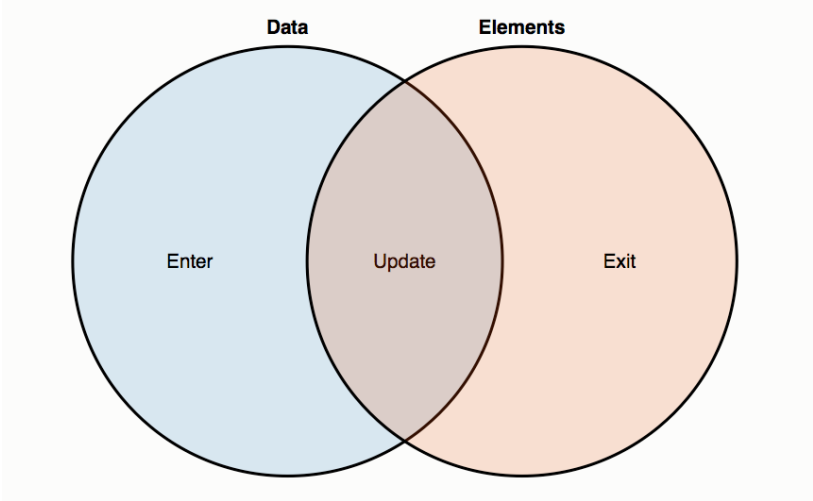
從資料到圖像元素




SELECTION
Data
Data
Data
Data
Data
Enter / Update
Exit
SELECTION

Enter, Update, Exit Pattern

- Enter: Data > Elements.
- Update: Data = Elements.
- Remove: Data < Elements.



Ajax with D3

```
d3.json([URL], function(error, data){
  if (error) return console.warn(error);
  // do something
});

d3.csv([URL]).get(function(error, data){
  // do something
});
```

Data / Array APIs

- d3.min([…]): 回傳陣列最小值
- d3.max([…]): 回傳陣列最大值
- d3.extent([…]): 同時回傳陣列最小及最大值
- d3.sum([…]): 回傳陣列數值總和
- d3.medium([…]): 回傳陣列中間值
- d3.mean([…]): 回傳陣列平均值

Data / Array APIs

- d3.shuffle([…]): 隨機排列陣列元素
- d3.merge([…]): 合併兩個陣列
- d3.pairs([…]): 回傳相鄰的陣列組
- d3.range([…]): 回傳指定的等差數列
- d3.permute([…]): 依照指定的索引回傳對應順序的陣列



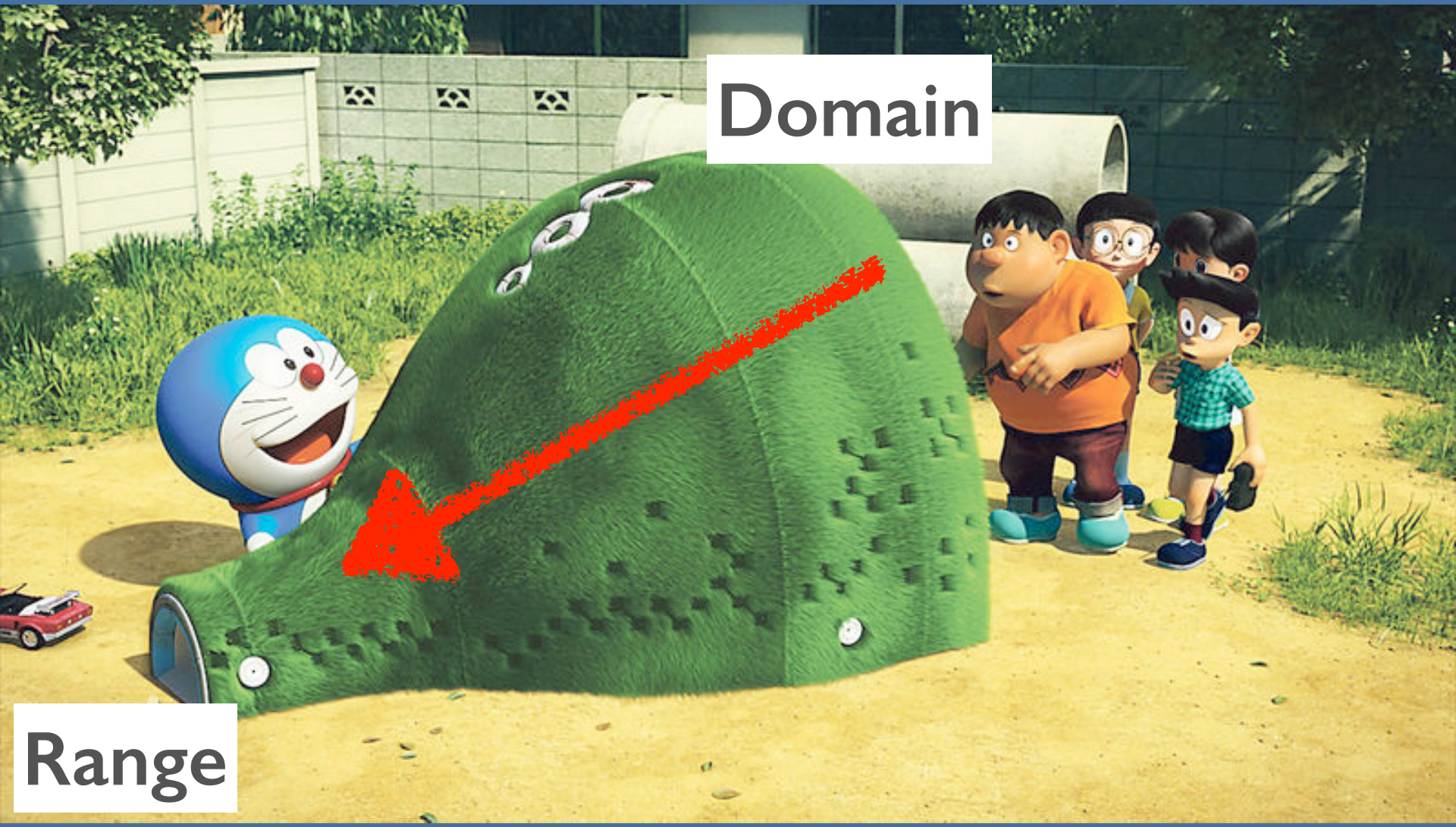
比例尺與座標軸

```
d3.scale.linear().domain().range()
```

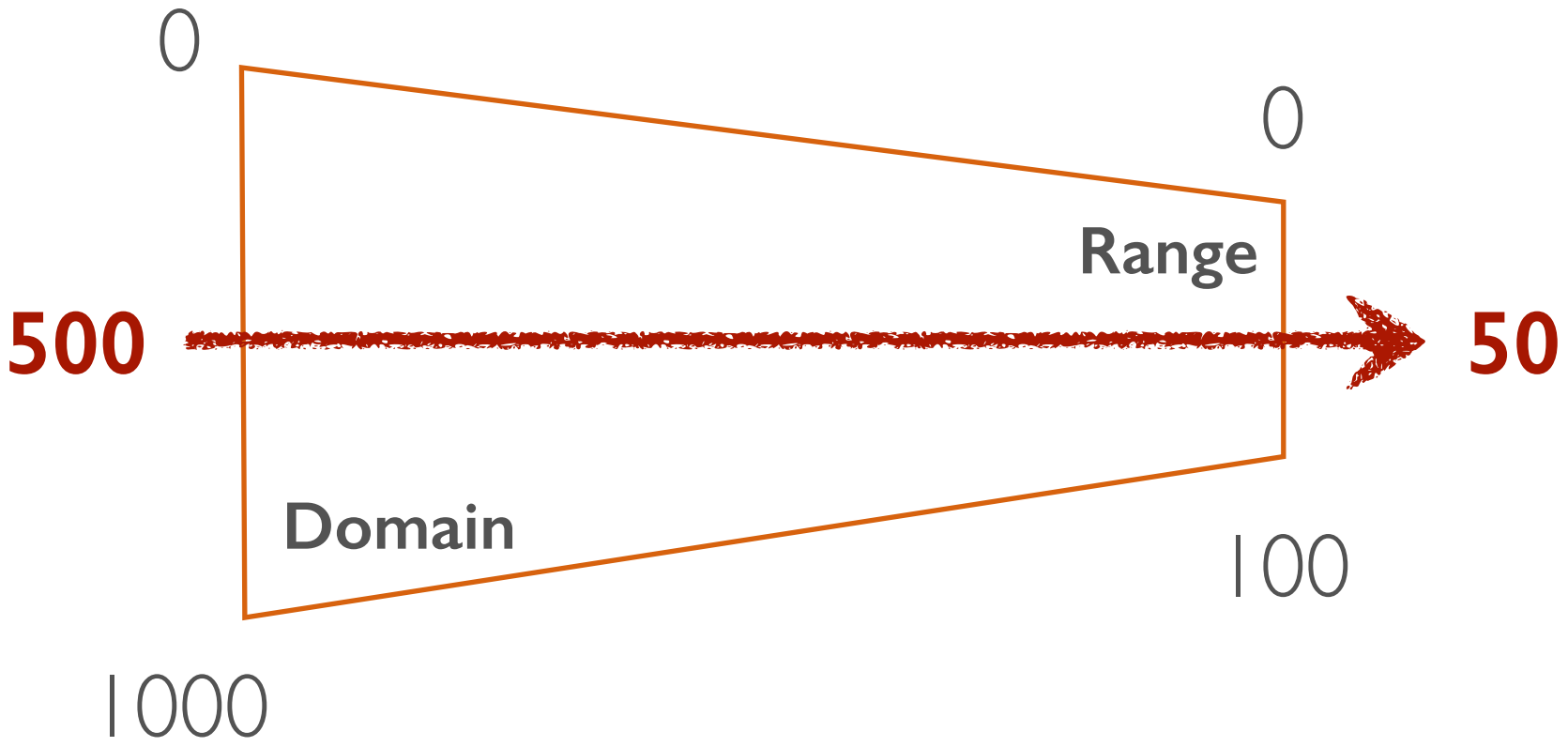
補間方法

輸入範圍

輸出範圍


Domain
Range

比例尺 Domain -> Range



scales

```
var d3Scale = d3.scale.linear()    // linear
        .domain([0, 1000])         // 傳入範圍
        .range([0, 100]);          // 輸出範圍

 console.log( d3Scale(500) );      // 50
 console.log( d3Scale(123) );      // 12.3
```

scales

```
// scale.pow()，回傳 n^2
var d3Scale = d3.scale.pow().exponent(2);

console.log( d3Scale(1) );       // 1
console.log( d3Scale(10) );      // 100
console.log( d3Scale(5) );       // 25
```

scales

```
// scale.pow()，回傳 (n^2) * 5
var d3Scale = d3.scale.pow()
                .exponent(2)
                .domain([0, 1])    // 輸入
                .range([0, 5]);    // 輸出

console.log( d3Scale(1) );         // 5
console.log( d3Scale(10) );        // 500
console.log( d3Scale(5) );         // 125
```

Quantitative Scales

- d3.scale.**linear**(): 線性比例尺
- d3.scale.**pow**(): 指數比例尺
- d3.scale.**log**(): 對數比例尺
- d3.scale.**quantize**(): 量子比例尺
- d3.scale.**ordinal**():序數比例尺

Scales

```
var colorScale = d3.scale.linear()
                   .domain([0, 20])
                   .range(["#f00", "#0f0"]);


for (var i = 0; i <= 20; i++) {
    body.append('div')
        .style('background-color', colorScale(i));
}
```



Scales

```
var widthScale = d3.scale.linear()
                   .domain([0, 12])
                   .range(["0px", "720px"]);

for (var i = 0; i <= 12; i++) {
    body.append('div').text( widthScale(i) );
}
```

0px 60px 120px 180px 240px 300px 360px 420px 480px 540px 600px 660px 720px

Scales

```
var domain_data = ["a", "b", "c", …… , "j"];
var domain_range = ["甲", "乙", "丙", …… "癸"];

var ordinalScale = d3.scale.ordinal()
      .domain(domain_data)
      .range(domain_range);
```

a: 甲	b: 乙	c: 丙	d: 丁	e: 戊	f: 己	g: 庚	h: 辛	i: 壬	j: 癸

interpolate 插值

```
var red   = d3.rgb(255,0,0);
var green = d3.rgb(0,255,0);
var compute = d3.interpolate(red, green);

console.log( compute(0) );          // #ff0000
console.log( compute(0.3) );        // #b34d00
console.log( compute(0.5) );        // #808000
console.log( compute(0.7) );        // #4db300
console.log( compute(1) );          // #00ff00
```

interpolate 插值

```
var a = "50px";
var b = "1000px";
var compute = d3.interpolate(a, b);

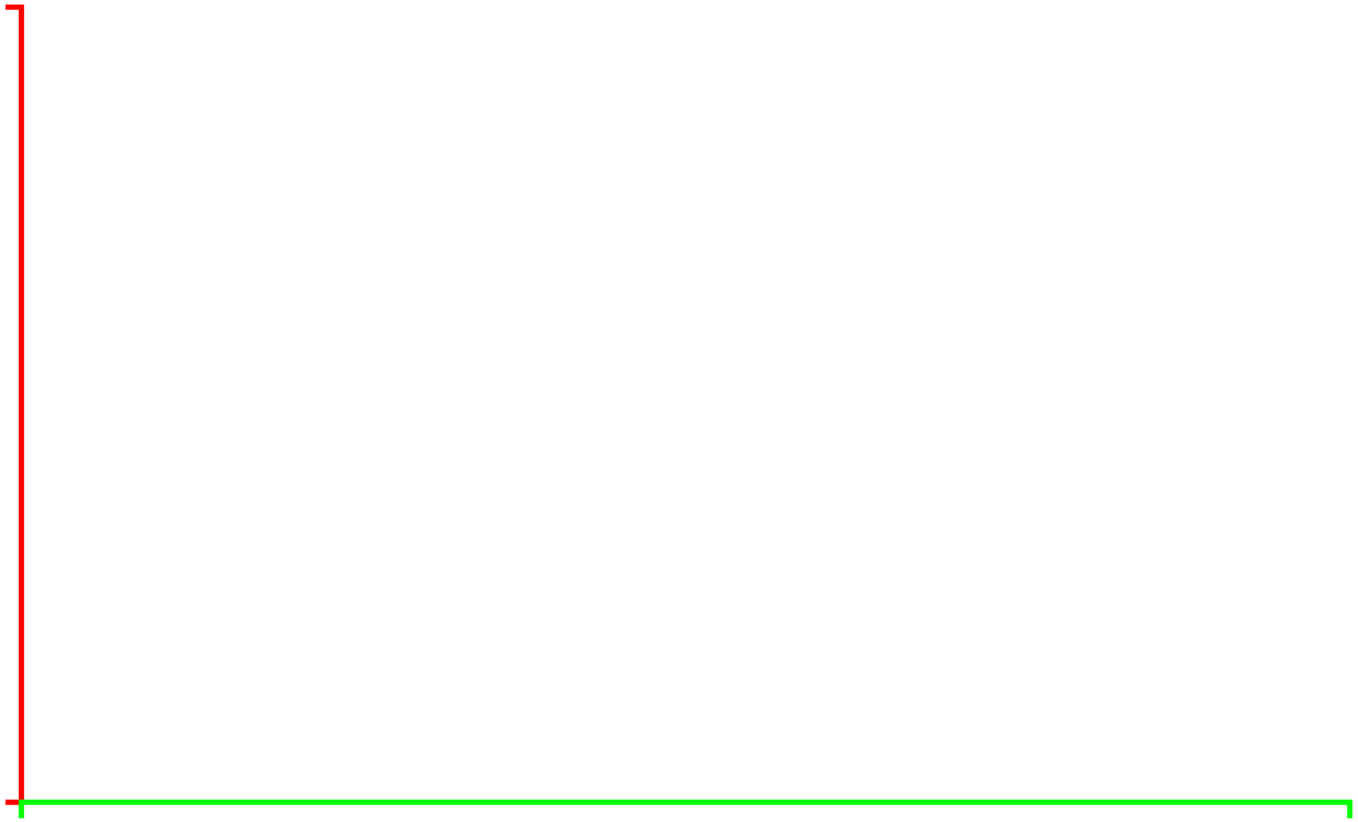
console.log( compute(0) );      // 50px
console.log( compute(0.3) );    // 335px
console.log( compute(0.5) );    // 525px
console.log( compute(0.7) );    // 715px
console.log( compute(1) );      // 1000px
```

Categorical Colors

- d3.scale.category10()
- d3.scale.category20()
- d3.scale.category20b()
- d3.scale.category20c()



Axis - 軸線



Axis - 軸線

```
var data  = [0, 150, 200, 300, 500, 1000];
// x 軸 scale
var xScale = d3.scale.linear()
  .domain(d3.extent(data))  // [0, 1000]
  .range([0, 500]);         // x-range 用來表示寬度

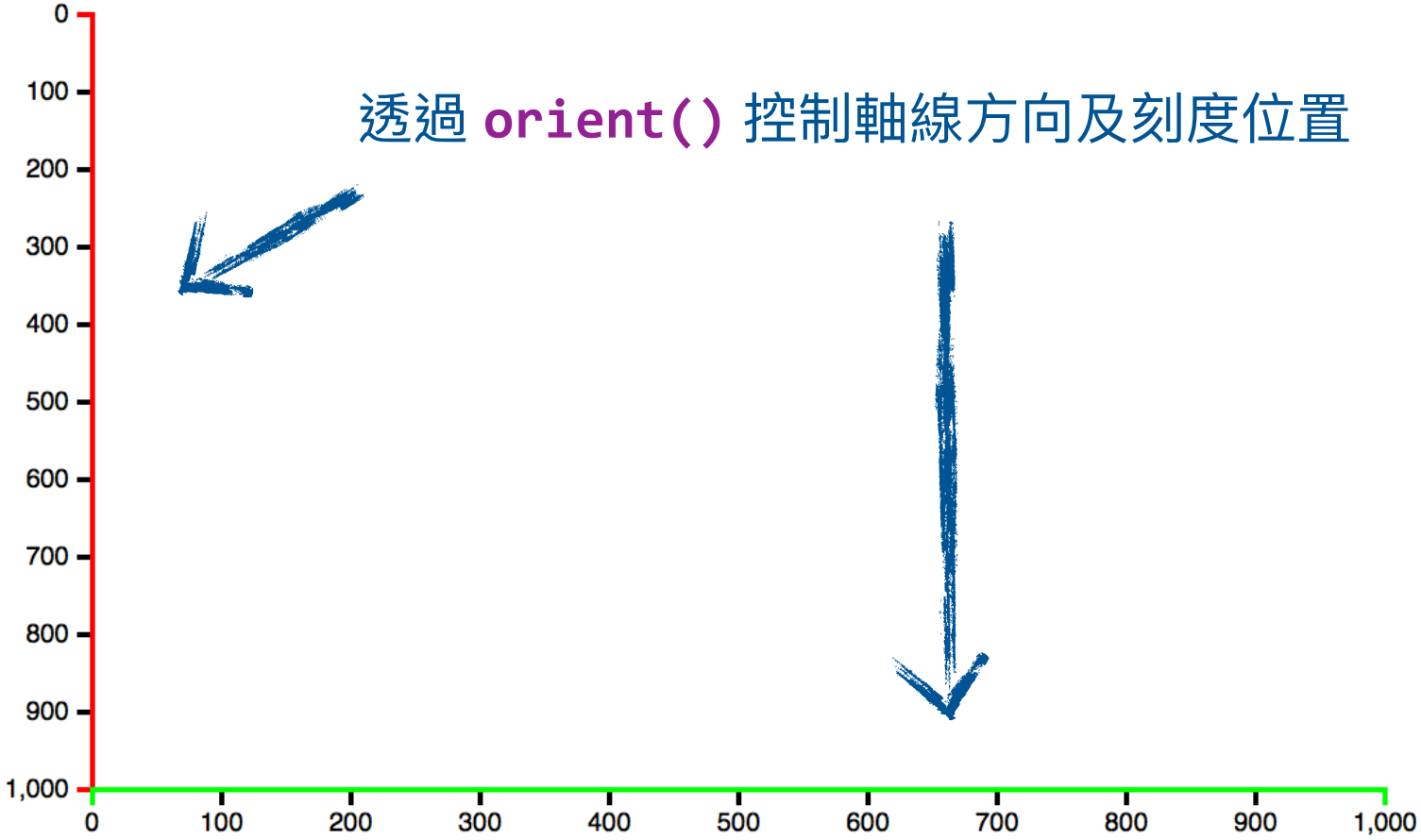
// y 軸 scale
var yScale = d3.scale.linear()
  .domain(d3.extent(data))  // [0, 1000]
  .range([0, 300]);         // y-range 用來表示高度
```

Axis - 軸線

```
// x 軸
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom");          // 刻度位置

// y 軸
var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left");            // 刻度位置
```

Axis - 軸線



透過 `orient()` 控制軸線方向及刻度位置

Axis - 軸線

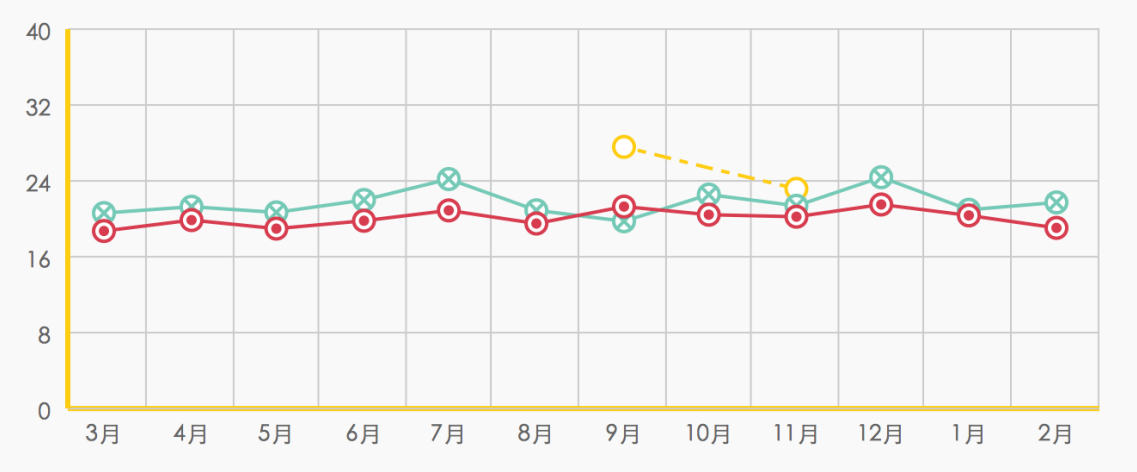
```
var data  = [0, 150, 200, 300, 500, 1000];
// x 軸 scale
var xScale = d3.scale.linear()
  .domain(d3.extent(data))  // [0, 1000]
  .range([0, 500]);         // x-range 用來表示寬度

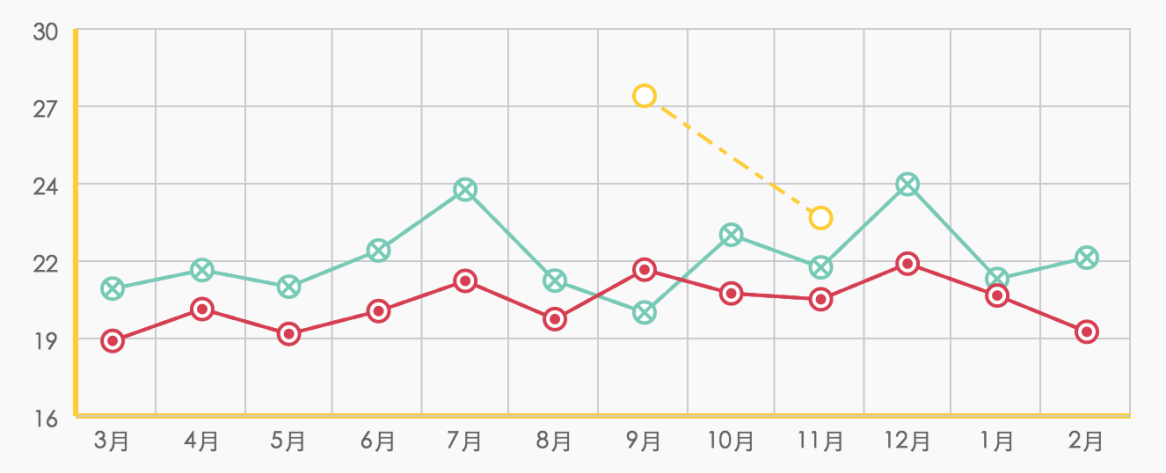
// y 軸 scale
var yScale = d3.scale.linear()
  .domain(d3.extent(data))  // [0, 1000]
  .range([300, 0]);         // y-range 用來表示高度
```

Ticks - 刻度

```
// x 軸
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom")
        .ticks(15)                  // 指定刻度數量, 單位
        .tickFormat(function(d){ return d + "px"; });
// y 軸
var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left");
```

Ticks - 刻度





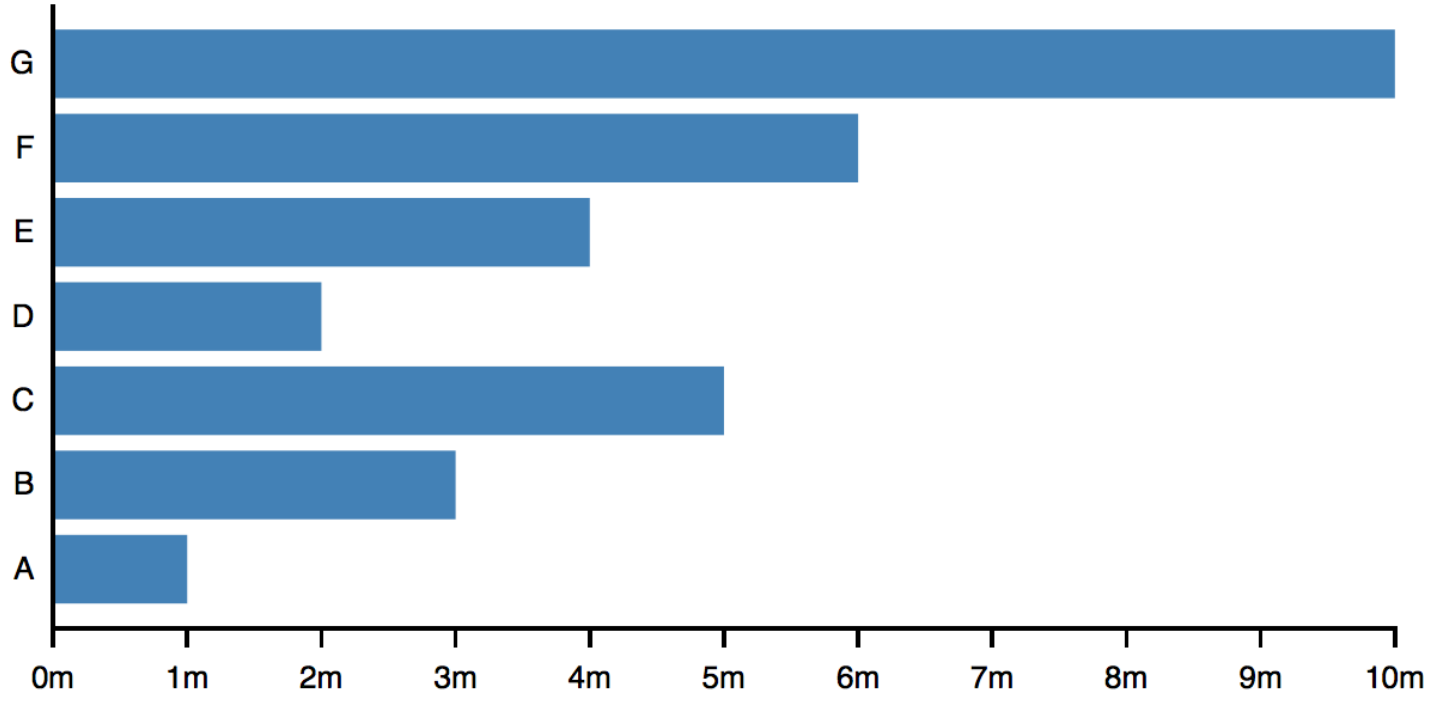
善用

d3.min()

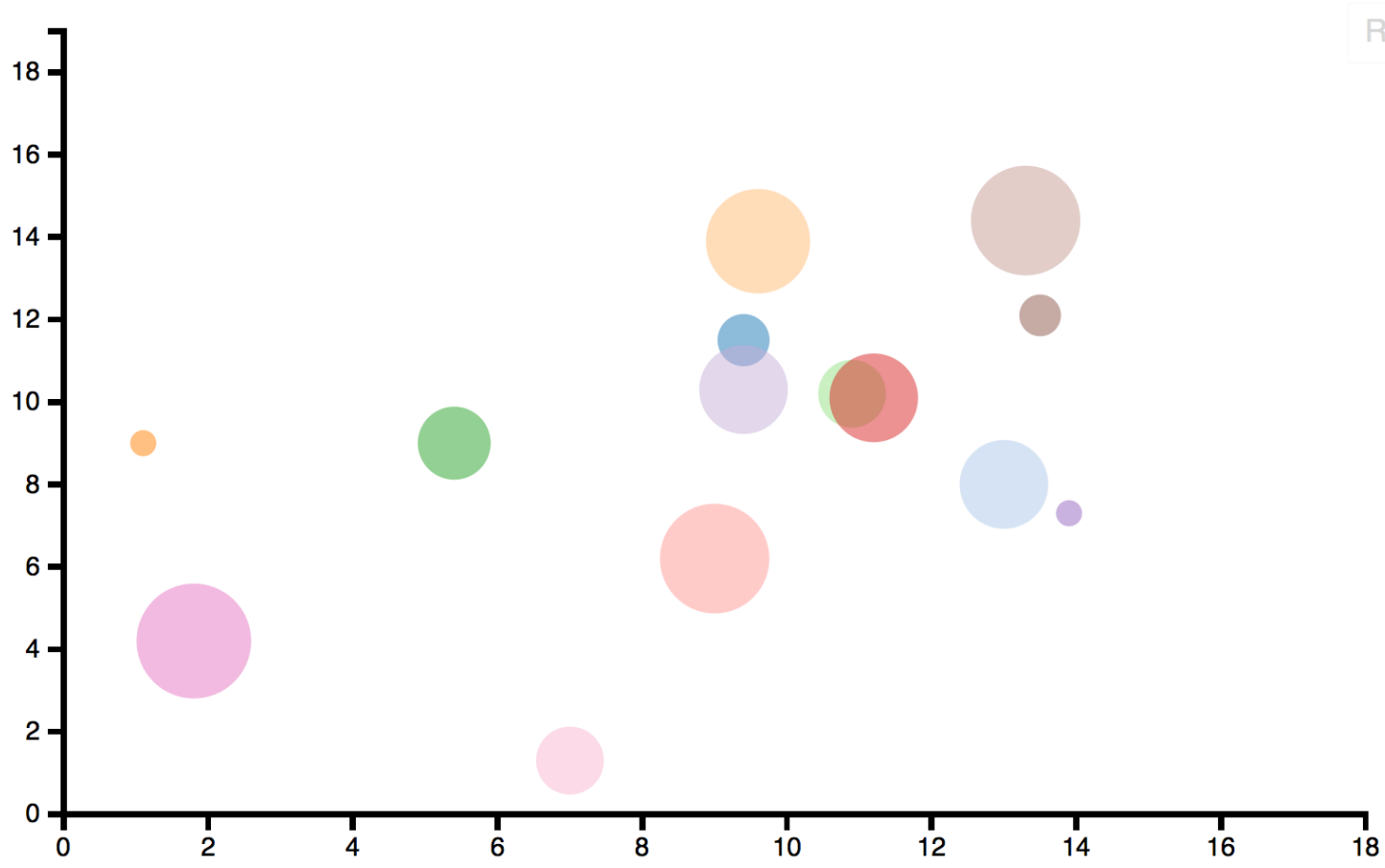
d3.max()

長條圖 Demo

```
var data = [1, 3, 5, 2, 4, 6, 10];
```



氣泡圖 Demo



http://jsbin.com/yepapa



動畫與事件

事件綁定

```
SELECTION.on('click', function(){
  alert('Hello D3!');
});
```

```
SELECTION.on('click', null);
```

事件綁定

```
SELECTION.on('click.event1', function(){
  alert('Hello D3 event 1');
});
```

```
SELECTION.on('click.event2', function(){
  alert('Hello D3 event 2');
});
```

事件綁定

- **mouse events**
  - click / dbclick / mouseover / mouseout

    mousemove / mousedown / mouseup
- **keyboard events**
  - keydown / keypress / keyup
- **touch events**
  - touchstart / touchmove / touchend

過場動畫

- Transitions have a four-phase life cycle:
  - The transition is scheduled.
  - The transition starts.
  - The transition runs.
  - The transition ends.

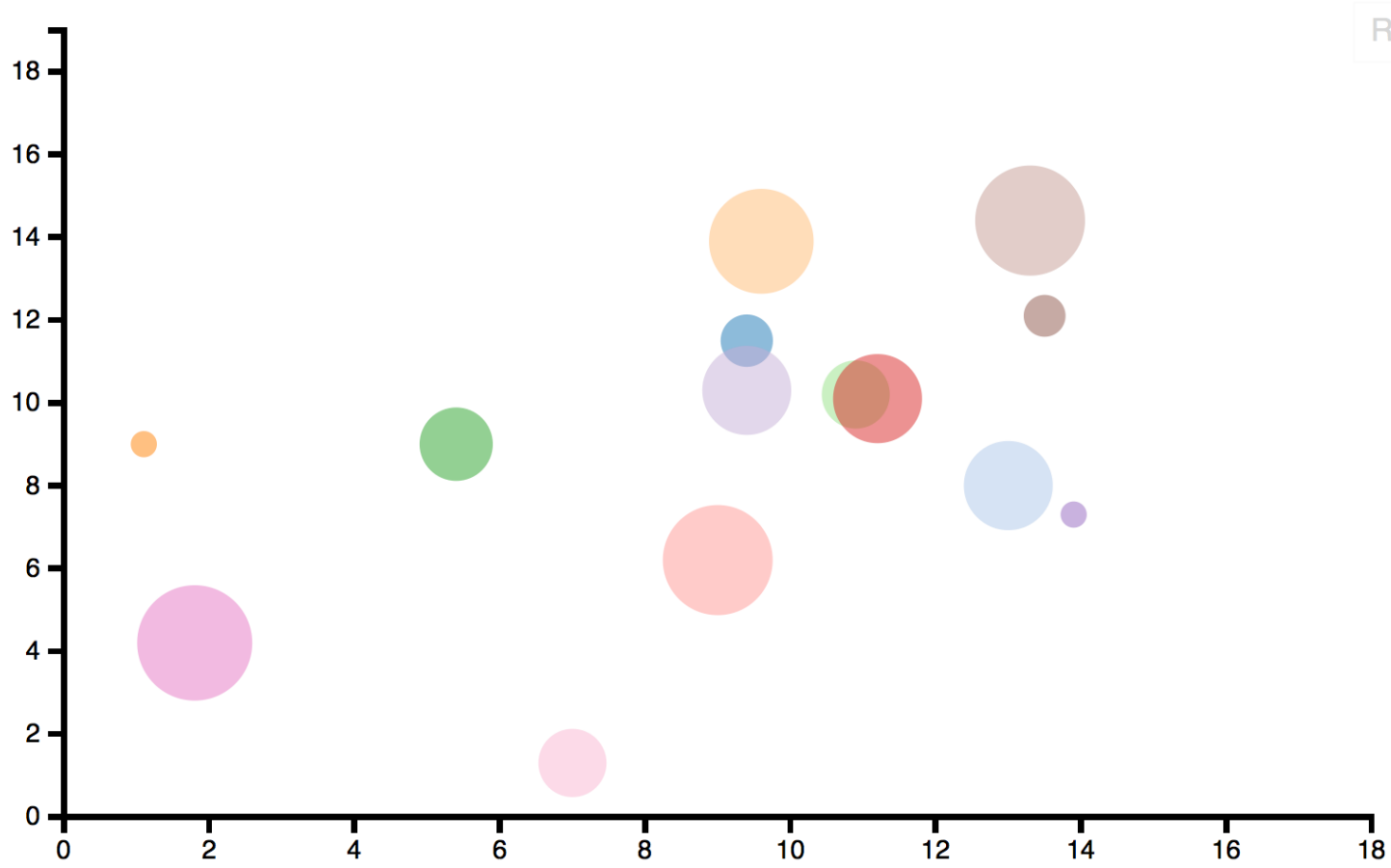
過場動畫

```
SELECTION.transition()
    .duration(1000)          // 動畫執行時間
    .delay(1000)             // 延遲
    .ease( ... )             // Easing 函數
    .attr({
       'transform': function(d){ ..... }
    });
```

過場動畫

```
SELECTION.transition()
    .duration(1000)
    .each('start', function(){ … })  // 動畫開始前執行
    .attr({ … })
    .each('end', function(){ … });   // 動畫結束後執行
```

氣泡圖加上動畫效果



http://jsbin.com/ketexa

zoom In / Out

```
var zoom = d3.behavior.zoom()    // 縮放
  .scaleExtent([1, 10])          // 指定縮放比例 [min, max]
  .on("zoom", zoomed);

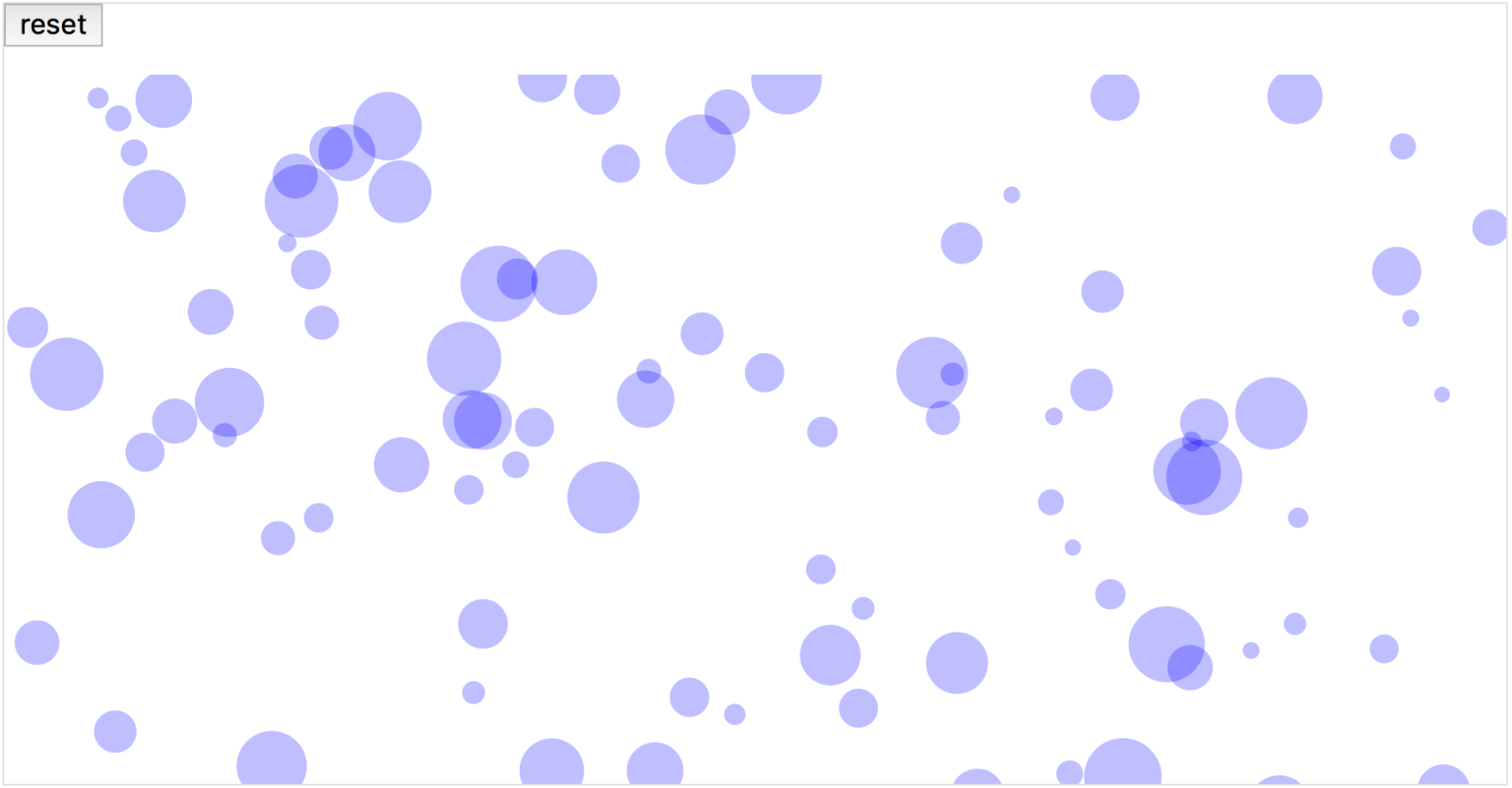
function zoomed() {
  container.attr("transform",
    "translate(" + d3.event.translate + ")" +
    "scale(" + d3.event.scale + ")");
}
```

Drag

```
var drag = d3.behavior.drag()
    .origin(function(d) { return d; })    // 設定拖曳起點座標
    .on("dragstart", dragstarted)
    .on("drag", dragged)
    .on("dragend", dragended);

 function dragstarted(d) { … }
 function dragged(d) { … }
 function dragended(d) { … }
```

元氣彈



http://bl.ocks.org/kurotanshi/33d614d5f4ab46b18bd0



SVG 圖形基礎

Why SVG

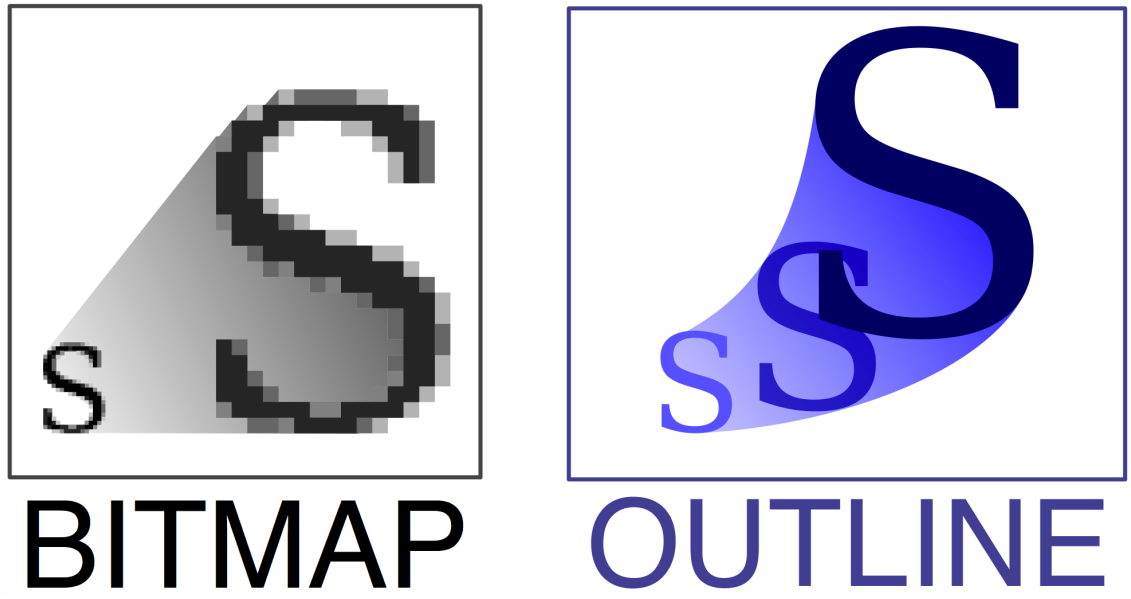
點陣式圖片放大會產生失真效應





Why SVG

向量式圖片由一連串座標計算組成，放再大也不怕



SVG

(毫無反應，就只是個 300*300 的畫布)

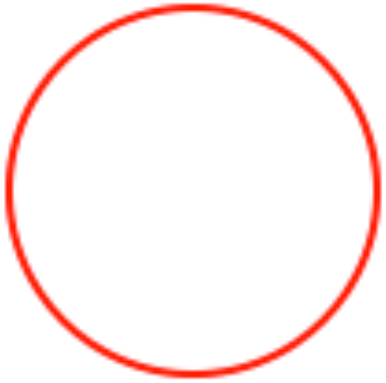
```
<svg width="300" height="100"></svg>
```

rect



```
<rect x="20" y="20"
      width="150" height="100" rx="10"></rect>
```

circle



```
<circle cx="55" cy="55" r="50"></circle>
```

line



```
<line x1="10" y1="30" x2="230" y2="60"></line>
```

text

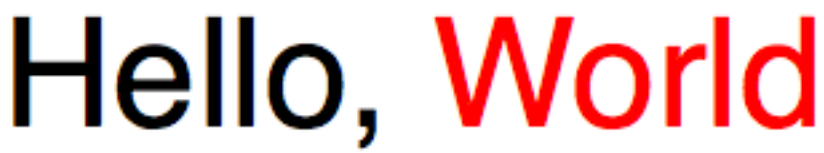



```
<text
     x="0" y="0" dx="0" dy="0"
     font-size="50" text-anchor="start">
  Hello
</text>
```

text-anchor="start/middle/end"

text - tspan

```
<text x="0" y="20" style="font-size: 17px;">
  <tspan>Hello, </tspan>
  <tspan fill="red">World </tspan>
</text>
```

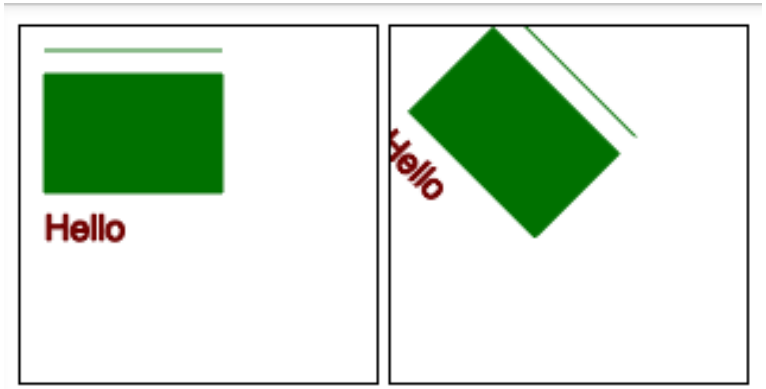


text - tspan

```
<text x="0" y="20" style="font-size: 17px;">
    <tspan x="10" dy="10">Hello, </tspan>
    <tspan x="10" dy="20">World </tspan>
</text>
```



g (group)

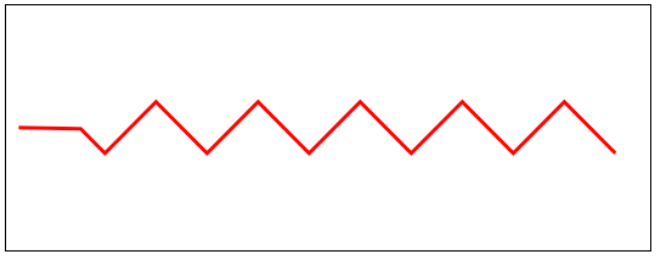



```
<g transform="rotate(45 50 50)">
  <line> … </line>
  <rect> … </rect>
  <text> … </text>
</g>
```

path

- 任意線段的組合，所有圖形都畫得出來
- d 屬性：一群「指令」+「座標」的組合

```HTML
<svg>
    <path d="M10,100 L61,101 L62,102 L63,103 L64,104 L65,105 L66,106 L67,107 L68,108 L69,109 L70,110 L71,111 L72,112 L73,113 L74,114
L75,115 L76,116 L77,117 L78,118 L79,119 L80,120 L81,121 L82,120 L83,119 L84,118 L85,117 L86,116 L87,115 L88,114 L89,113 L90,112
L91,111 L92,110 L93,109 L94,108 L95,107 L96,106 L97,105 L98,104 L99,103 L100,102 L101,101 L102,100 L103,99 L104,98 L105,97 L106,96
L107,95 L108,94 L109,93 L110,92 L111,91 L112,90 L113,89 L114,88 L115,87 L116,86 L117,85 L118,84 L119,83 L120,82 L121,81 L122
L123,79 L124,80 L125,81 L126,82 L127,83 L128,84 L129,85 L130,86 L131,87 L132,88 L133,89 L134,90 L135,91 L136,92 L137,93 L138
L139,95 L140,96 L141,97 L142,98 L143,99 L144,100 L145,101 L146,102 L147,103 L148,104 L149,105 L150,106 L151,107 L152,108 L15
L154,110 L155,111 L156,112 L157,113 L158,114 L159,115 L160,116 L161,117 L162,118 L163,119 L164,120 L165,121 L166,120 L167,11
L169,117 L170,116 L171,115 L172,114 L173,113 L174,112 L175,111 L176,110 L177,109 L178,108 L179,107 L180,106 L181,105 L182,10
L184,102 L185,101 L186,100 L187,99 L188,98 L189,97 L190,96 L191,95 L192,94 L193,93 L194,92 L195,91 L196,90 L197,89 L198,88 L
L200,86 L201,85 L202,84 L203,83 L204,82 L205,81 L206,80 L207,79 L208,80 L209,81 L210,82 L211,83 L212,84 L213,85 L214,86 L215
L216,88 L217,89 L218,90 L219,91 L220,92 L221,93 L222,94 L223,95 L224,96 L225,97 L226,98 L227,99 L228,100 L229,101 L230,102 L
L232,104 L233,105 L234,106 L235,107 L236,108 L237,109 L238,110 L239,111 L240,112 L241,113 L242,114 L243,115 L244,116 L245,11
L247,119 L248,120 L249,121 L250,120 L251,119 L252,118 L253,117 L254,116 L255,115 L256,114 L257,113 L258,112 L259,111 L260,11
L262,108 L263,107 L264,106 L265,105 L266,104 L267,103 L268,102 L269,101 L270,100 L271,99 L272,98 L273,97 L274,96 L275,95 L27
L277,93 L278,92 L279,91 L280,90 L281,89 L282,88 L283,87 L284,86 L285,85 L286,84 L287,83 L288,82 L289,81 L290,80 L291,79 L292
L293,81 L294,82 L295,83 L296,84 L297,85 L298,86 L299,87 L300,88 L301,89 L302,90 L303,91 L304,92 L305,93 L306,94 L307,95 L308
L309,97 L310,98 L311,99 L312,100 L313,101 L314,102 L315,103 L316,104 L317,105 L318,106 L319,107 L320,108 L321,109 L322,110 L323,111
L324,112 L325,113 L326,114 L327,115 L328,116 L329,117 L330,118 L331,119 L332,120 L333,121 L334,120 L335,119 L336,118 L337,117 L338,116
L339,115 L340,114 L341,113 L342,112 L343,111 L344,110 L345,109 L346,108 L347,107 L348,106 L349,105 L350,104 L351,103 L352,102 L353,101
L354,100 L355,99 L356,98 L357,97 L358,96 L359,95 L360,94 L361,93 L362,92 L363,91 L364,90 L365,89 L366,88 L367,87 L368,86 L369,85
L370,84 L371,83 L372,82 L373,81 L374,80 L375,79 L376,80 L377,81 L378,82 L379,83 L380,84 L381,85 L382,86 L383,87 L384,88 L385,89
L386,90 L387,91 L388,92 L389,93 L390,94 L391,95 L392,96 L393,97 L394,98 L395,99 L396,100 L397,101 L398,102 L399,103 L400,104 L401,105
L402,106 L403,107 L404,108 L405,109 L406,110 L407,111 L408,112 L409,113 L410,114 L411,115 L412,116 L413,117 L414,118 L415,119 L416,120
L417,121 L418,120 L419,119 L420,118 L421,117 L422,116 L423,115 L424,114 L425,113 L426,112 L427,111 L428,110 L429,109 L430,108 L431,107
L432,106 L433,105 L434,104 L435,103 L436,102 L437,101 L438,100 L439,99 L440,98 L441,97 L442,96 L443,95 L444,94 L445,93 L446,92 L447,91
L448,90 L449,89 L450,88 L451,87 L452,86 L453,85 L454,84 L455,83 L456,82 L457,81 L458,80 L459,79 L460,80 L461,81 L462,82 L463,83
L464,84 L465,85 L466,86 L467,87 L468,88 L469,89 L470,90 L471,91 L472,92 L473,93 L474,94 L475,95 L476,96 L477,97 L478,98 L479,99
L480,100 L481,101 L482,102 L483,103 L484,104 L485,105 L486,106 L487,107 L488,108 L489,109 L490,110 L491,111 L492,112 L493,113 L494,114
L495,115 L496,116 L497,117 L498,118 L499,119 L500,120 L501,121" style="stroke:#f00; fill:none; stroke-width:3;" />
</svg>
```



path

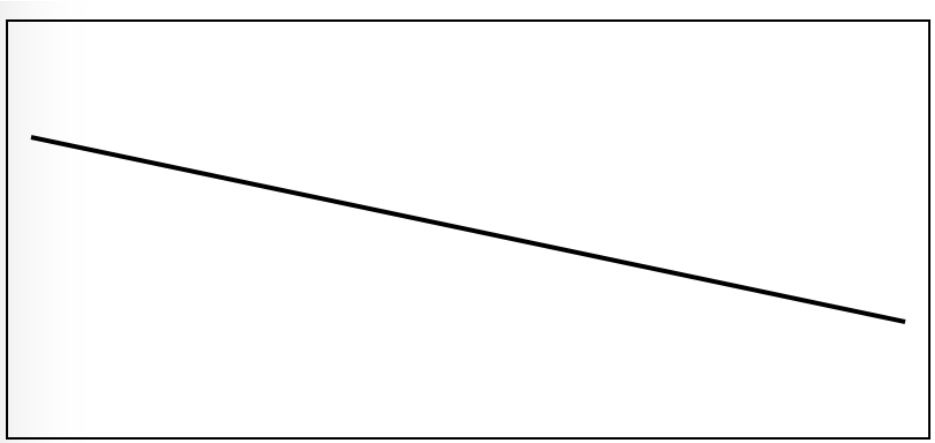
- 移動: **M** (MoveTo)
- 直線: **L** (LineTo) / **H** (Horizontal) / **V** (Vertical)
- 曲線: **C** (CurveTo) / **S** (Shorthand/Smooth CurveTo)

  Q (Quadratic Bezier CurveTo)

  T (smooth quadratic Bezier CurveTo)
- 弧線: **A** (elliptical Arc)
- 閉合: **Z** (close-path)

path

```
<line x1="10" y1="50" x2="390" y2="130">
```

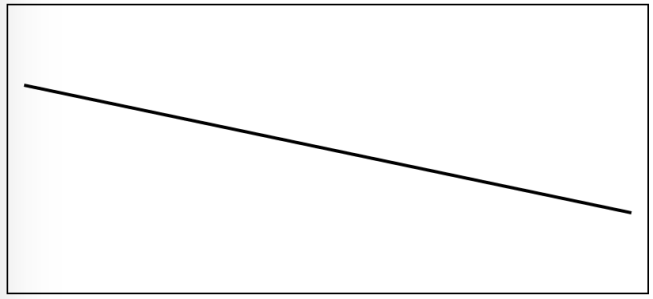


```
<path d="M10,50L390,130">
```

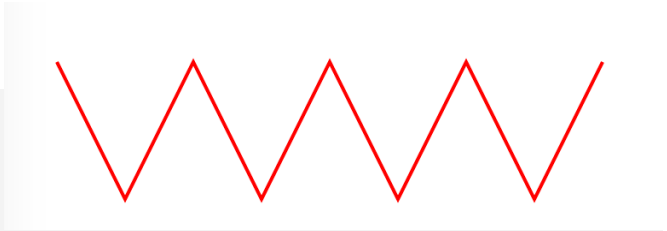
Path Generators - d3.svg.line()

```
var svg = d3.select('.svg2');
var linePath = d3.svg.line();
var data = [[10,50], [390, 130]];

svg.append('path')
   .attr('d', linePath(data));
```



Path Generators - d3.svg.line()



```
var svg = d3.select('.svg');
var linePath = d3.svg.line()
      .x(function(d, i){ return d; })
      .y(function(d, i){ return (i%2 === 0) ? 40 : 120;})

var data = [40, 80, 120, 160, 200, 240, 280, 320, 360];

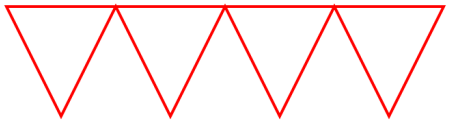
svg.append('path')
    .attr({ 'd': linePath(data) });
```

Path Generators - interpolate

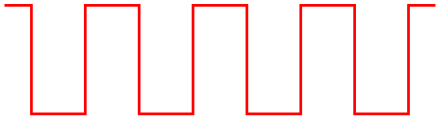
```
d3.svg.line()
  .x(…).y(…)
  .interpolate("linear");
```



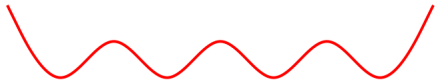
```
  .interpolate("linear-closed");
```



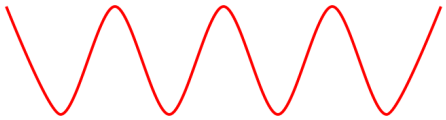
```
  .interpolate("step");
```



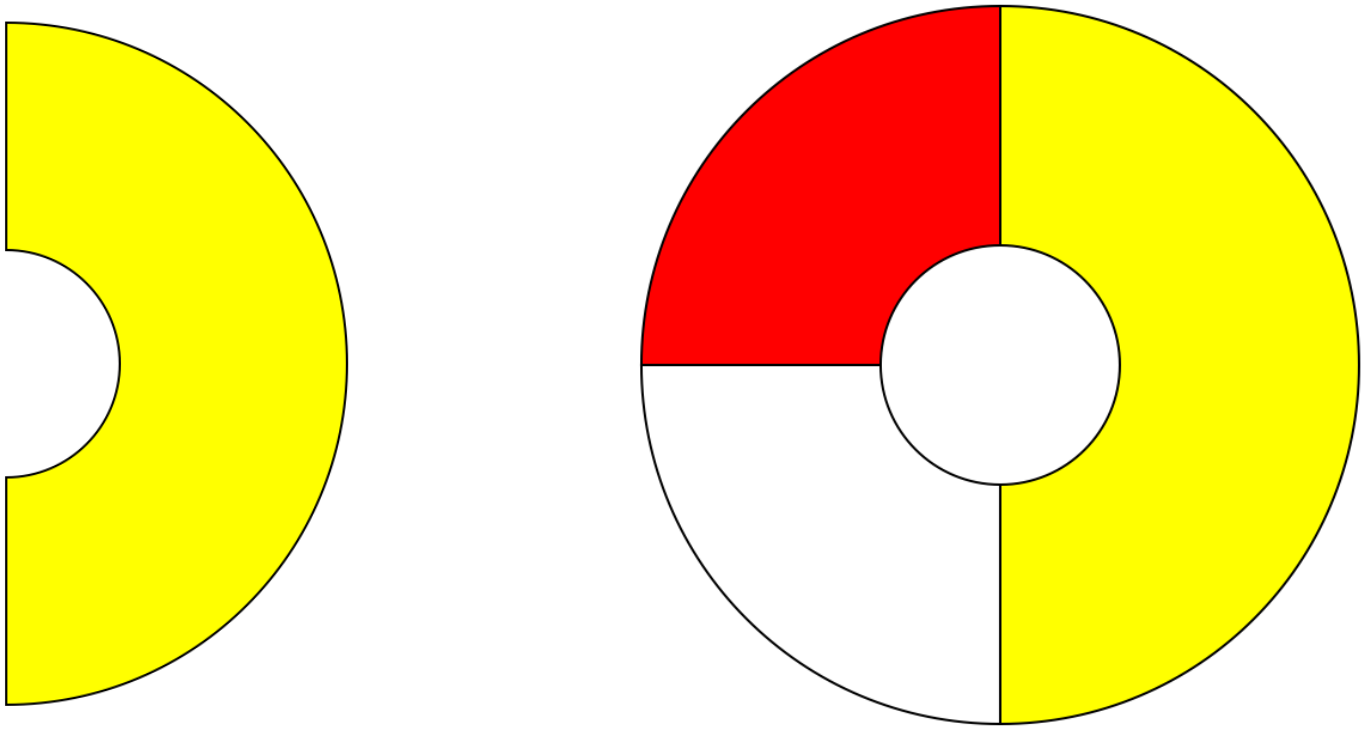
```
  .interpolate("basis");
```



```
  .interpolate("cardinal");
```



Path Generators - d3.svg.arc()



Path Generators - d3.svg.arc()

```
var myArc1 = {
  "startAngle": 0,
  "endAngle": Math.PI
};

var myArc2 = {
  "startAngle": Math.PI,
  "endAngle": Math.PI * 1.5
};
```

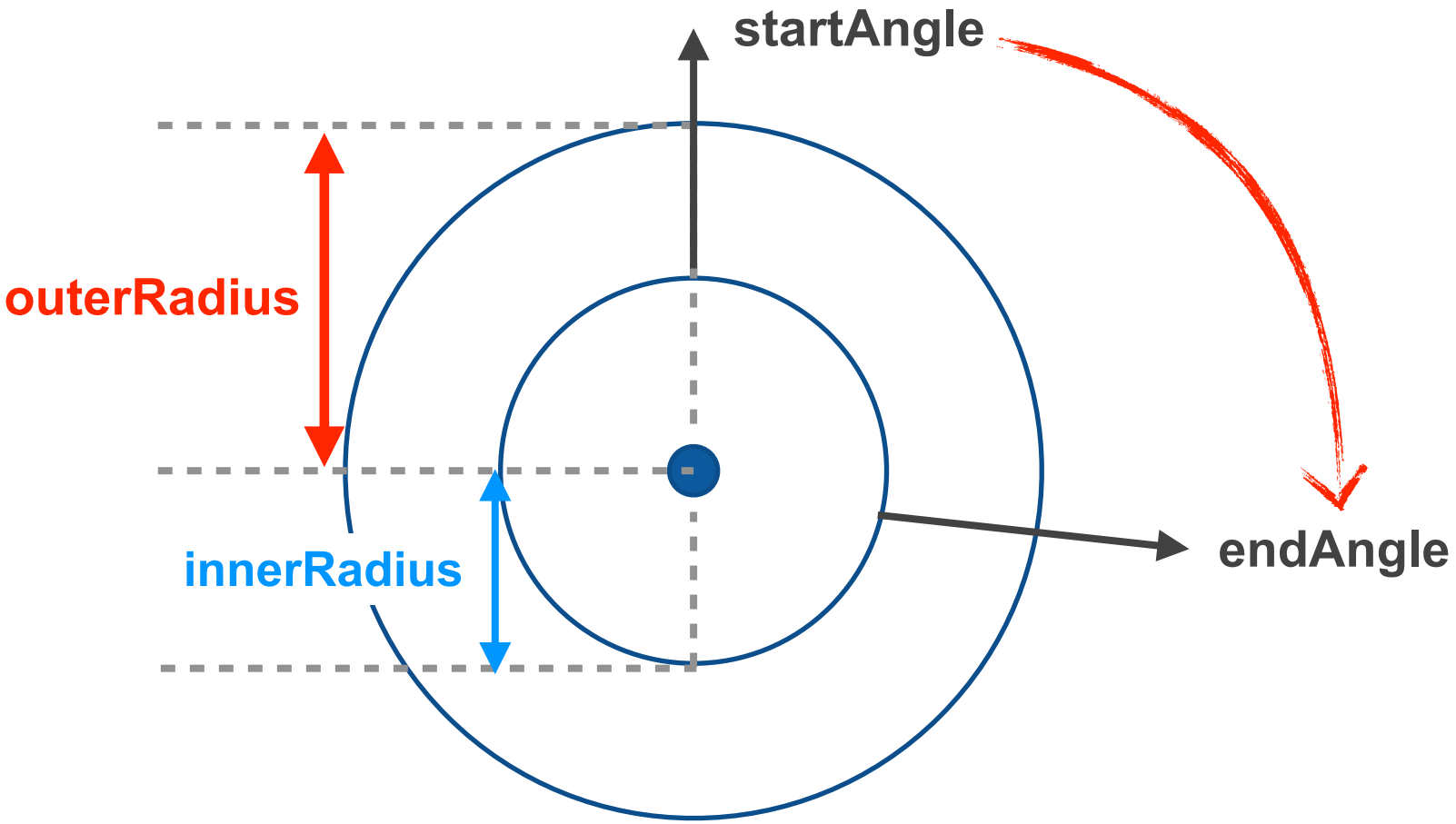
```
var myArc3 = {
  "startAngle": Math.PI * 1.5,
  "endAngle": Math.PI * 2
};
```

Path Generators - d3.svg.arc()

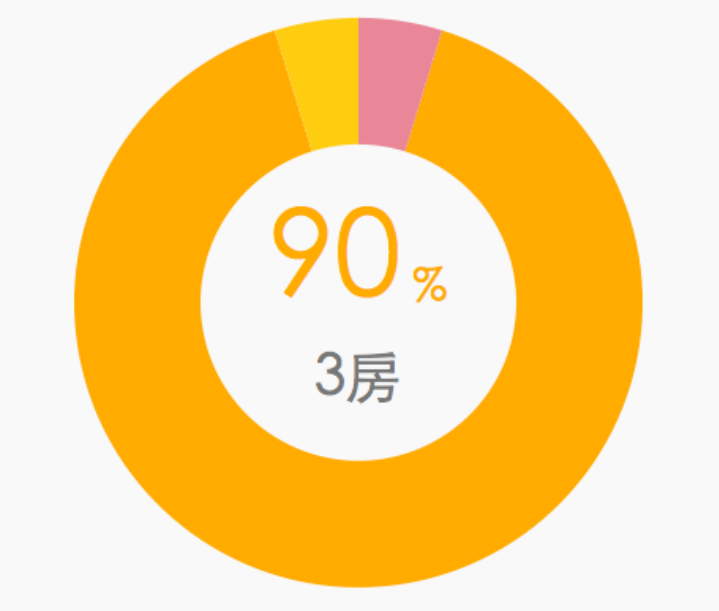
```
var arc = d3.svg.arc()
            .innerRadius(50)
            .outerRadius(150);

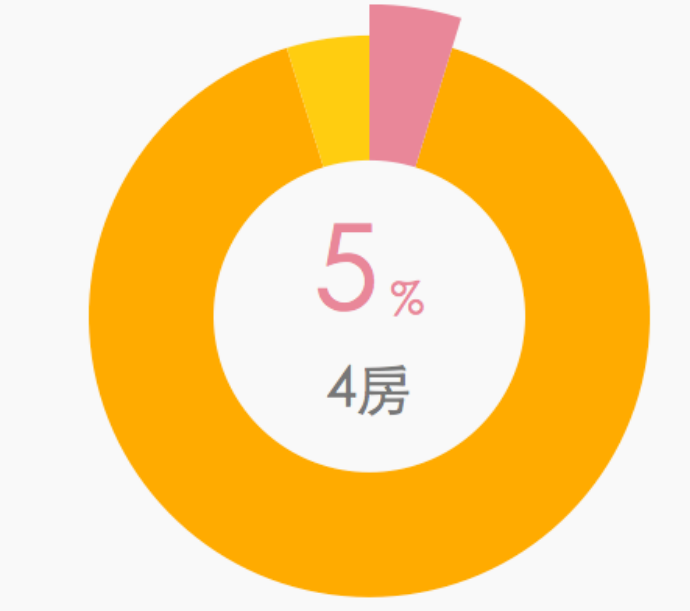
svg.append('path')
   .attr({ 'd': arc(myArc1) });
```

Path Generators - d3.svg.arc()

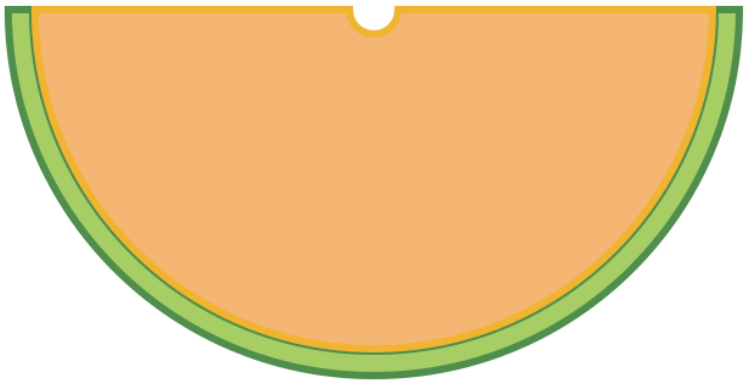


Path Generators - d3.svg.arc()

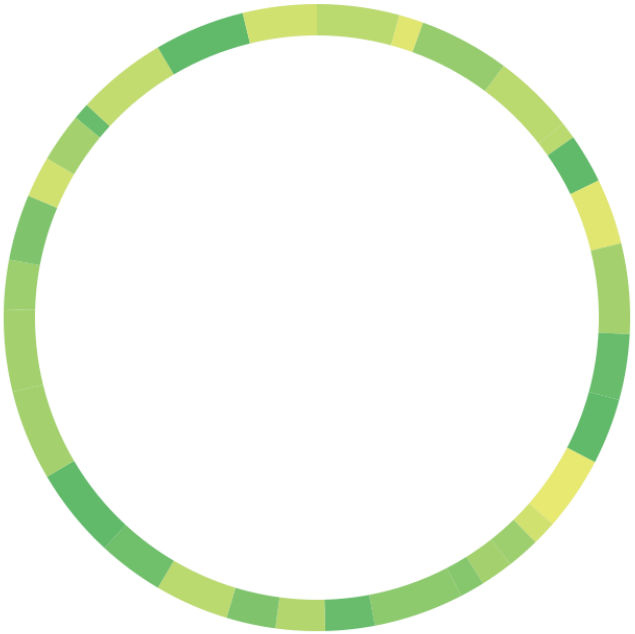




Path Generators - d3.svg.arc()



http://jsbin.com/xomaxi



http://jsbin.com/fuqiho



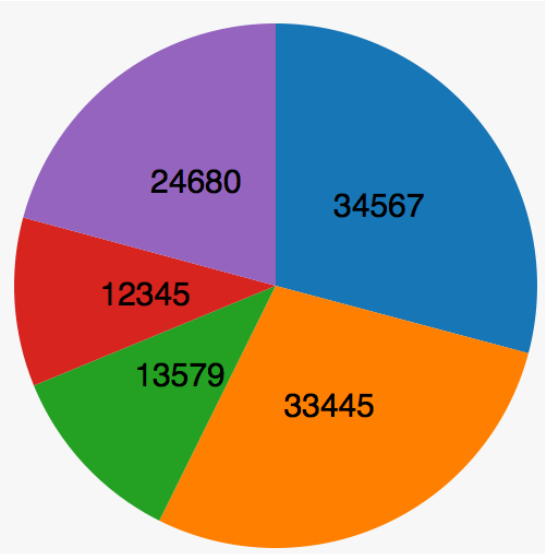
D3.js Layouts

https://github.com/mbostock/d3/wiki/Layouts

D3.js Layouts

- d3.js 提供了 11 種版型 (layouts)
- 只需提供該版型指定的資料格式就可以直接套用
- layouts 本身**不提供繪圖**功能，而是將原始的資料轉換成**適合被 d3 繪圖**的資料格式
- ~~懶得計算 top, left, d 等等屬性的時候特別好用~~

Pie Layout

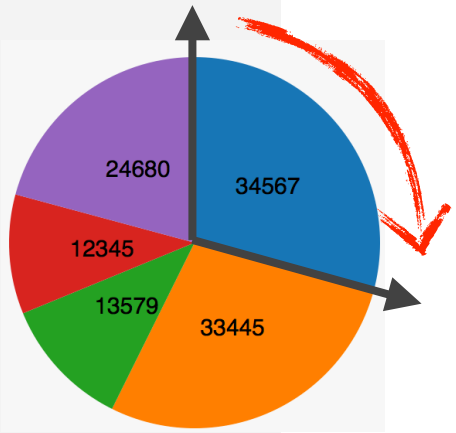



```
// 建立資料
var data = [34567, 33445, 13579, 12345, 24680];

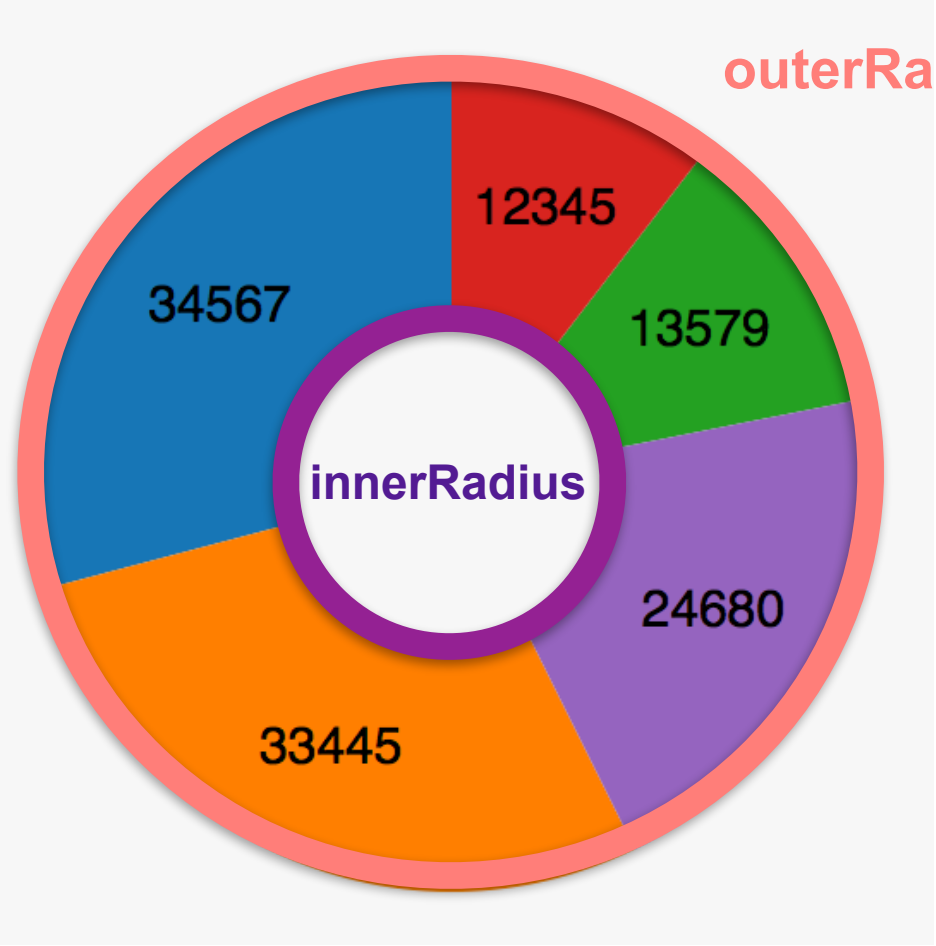

// 透過 layout.pie() 轉成適合給 d3.arc 用的結構
var g = svg.selectAll("g path")
    .data(d3.layout.pie(data))
    .enter()
    .append("g");
```

加入動畫

```
g.append("path")
  .attr("fill", function(d, i) { return color(i); })
  .transition()
  .ease('bounce')
  .duration(800)
  .attrTween("d", function (d) {
    var start = { startAngle: 0, endAngle: 0 };
    var interpolate = d3.interpolate(start, d);
    return function (t) { return arc(interpolate(t)); };
  });
```

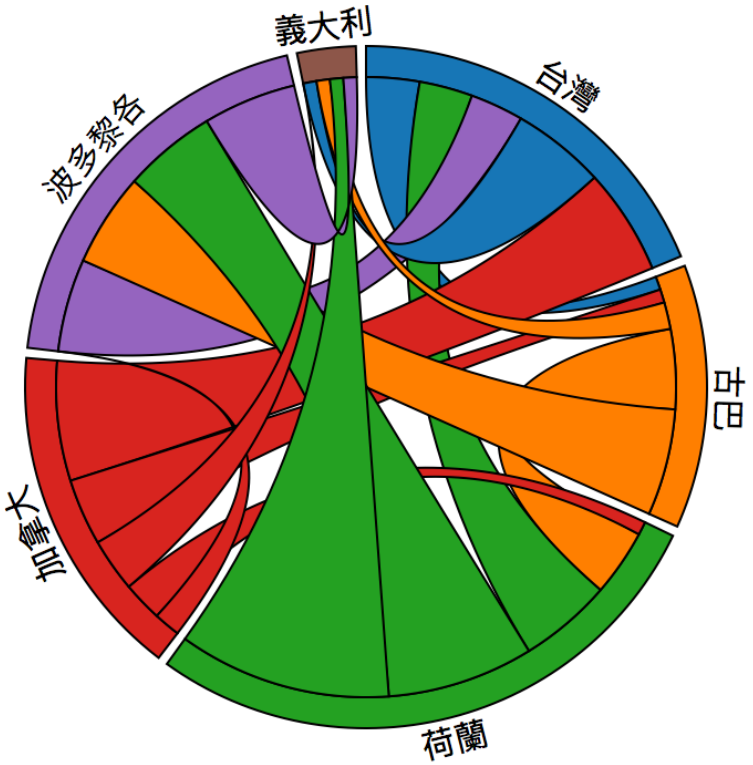


一秒變甜甜圈



Chord Layout

2015 第一屆世界12強棒球賽 A 組預賽對戰結果



Chord Layout

2015 第一屆世界12強棒球賽 A 組預賽對戰結果



日期	時間	場次	客隊	比數	主隊	備註	場地
11/08 日							
1回戰 11/09 一	18:30	2	荷蘭	7：4	中華		台中洲際
11/10 星期二	12:30	4	義大利	1：7	波多黎各		台中洲際
	18:30	6	加拿大	5：1	古巴		台中洲際
2回戰 11/11 星期三	12:30	8	古巴	6：5	荷蘭		台中洲際
	18:30	11	波多黎各	0：2	加拿大		雲林斗六
	18:30	12	中華	7：1	義大利		台中洲際
3回戰 11/12 星期四	12:30	14	波多黎各	7：8	古巴	十局	台中洲際
	18:30	17	荷蘭	16：1	義大利	七局	雲林斗六
	18:30	18	加拿大	9：8	中華		台中洲際
11/13							十三
4回戰 11/14 星期六	12:30	20	波多黎各	7：11	荷蘭		台中洲際
	18:30	23	義大利	0：4	加拿大		雲林斗六
	18:30	24	古巴	1：4	中華		台中洲際
5回戰 11/15 星期日	12:30	26	中華	4：7	波多黎各	12局	台中洲際
	18:30	29	義大利	1：2	古巴		雲林斗六
	18:30	30	加拿大	3：1	荷蘭		台中洲際

得分＼失分	台灣	古巴	荷蘭	加拿大	波多黎各	義大利
台灣	-	4	4	8	4	7
古巴	1	-	6	1	8	2
荷蘭	7	5	-	1	11	16
加拿大	9	5	3	-	2	4
波多黎各	7	7	7	0	-	7
義大利	1	1	1	0	1	-

Chord Layout

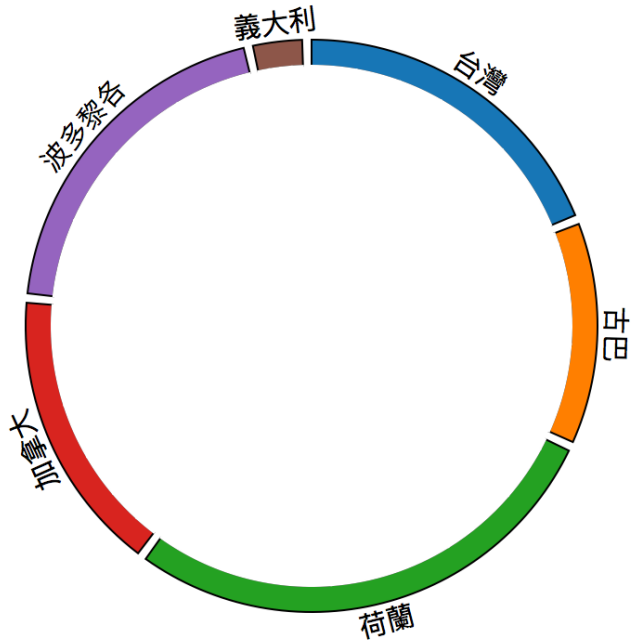
```
var area = ["台灣", "古巴", "荷蘭", "加拿大", "波多黎各", "義大利"];

var scores = [
  [ 0, 4, 4, 8, 4, 7 ],
  [ 1, 0, 6, 1, 8, 2 ],
  [ 7, 5, 0, 1, 11, 16 ],
  [ 9, 5, 3, 0, 2, 4 ],
  [ 7, 7, 7, 0, 0, 7 ],
  [ 1, 1, 1, 0, 1, 0 ]
];
```

Chord Layout

```
// 外圈 d3.svg.arc()
var arcOuter = d3.svg.arc()
        .innerRadius(innerRadius)
        .outerRadius(outerRadius);

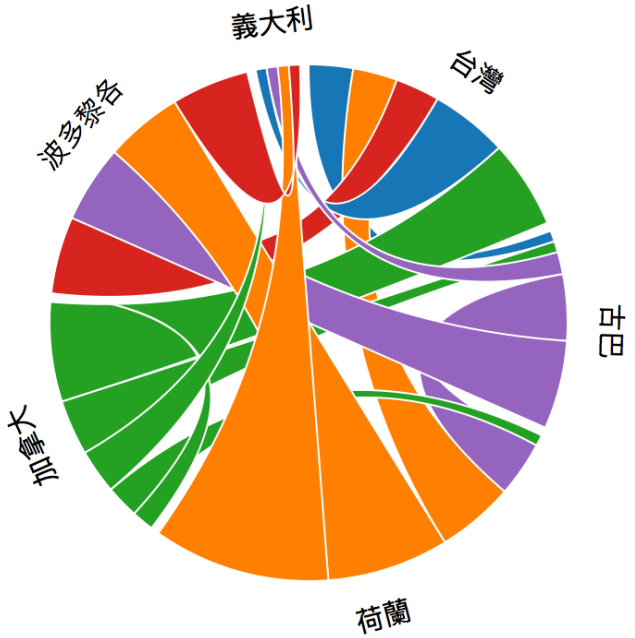
gOuter.selectAll(".outerPath")
  .data(chord.groups())
  .enter()
  .append("path")
    .attr({
      "d": arcOuter,
      "class": "outerPath"
    })
    .style("fill", function(d) { return color20(d.index); });
```



Chord Layout

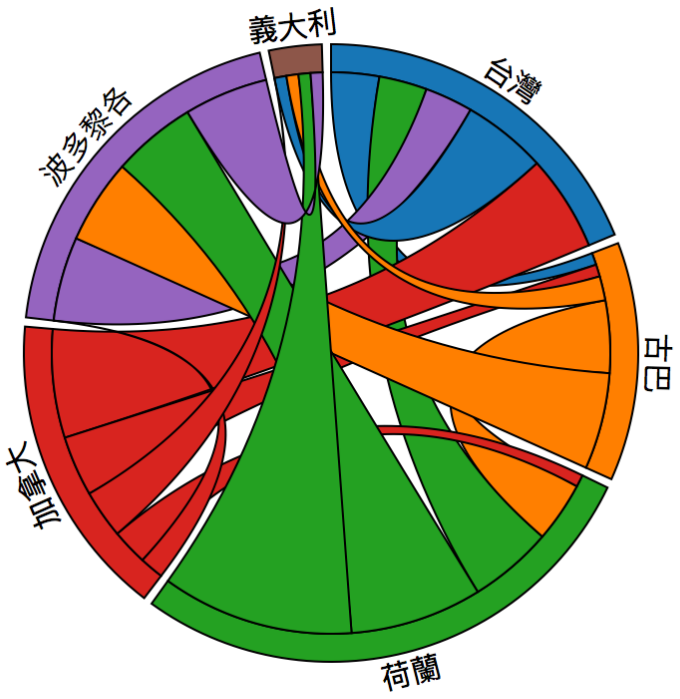
```
// 內圈 d3.svg.chord()
var arcInner =  d3.svg.chord()
                  .radius(innerRadius);

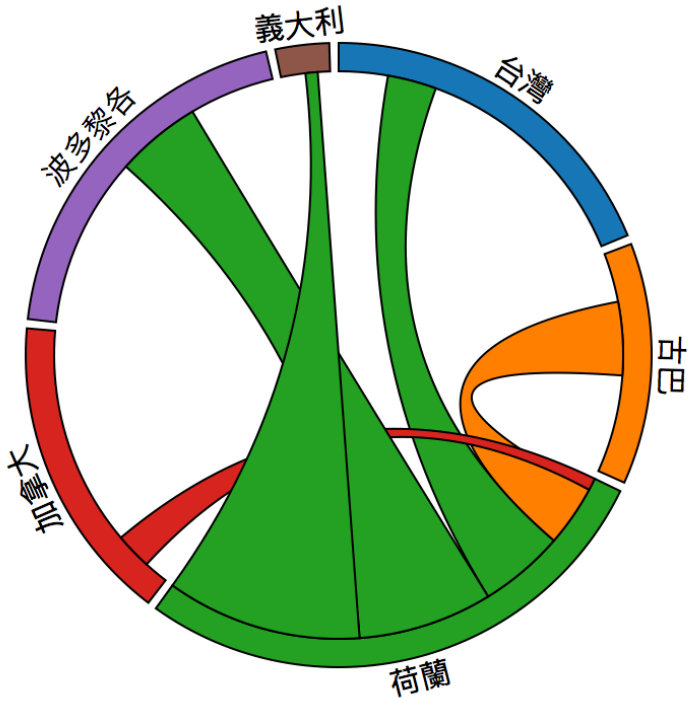
gInner.selectAll(".innerPath")
  .data(chord.chords())
  .enter()
  .append("path")
  .attr({
    "class": "innerPath",
    "d": arcInner
  })
  .style("fill", function(d) {
    return color20(d.source.index);
  });
```



Chord Layout

2015 第一屆世界12強棒球賽 A 組預賽對戰結果





Hierarchy Layout

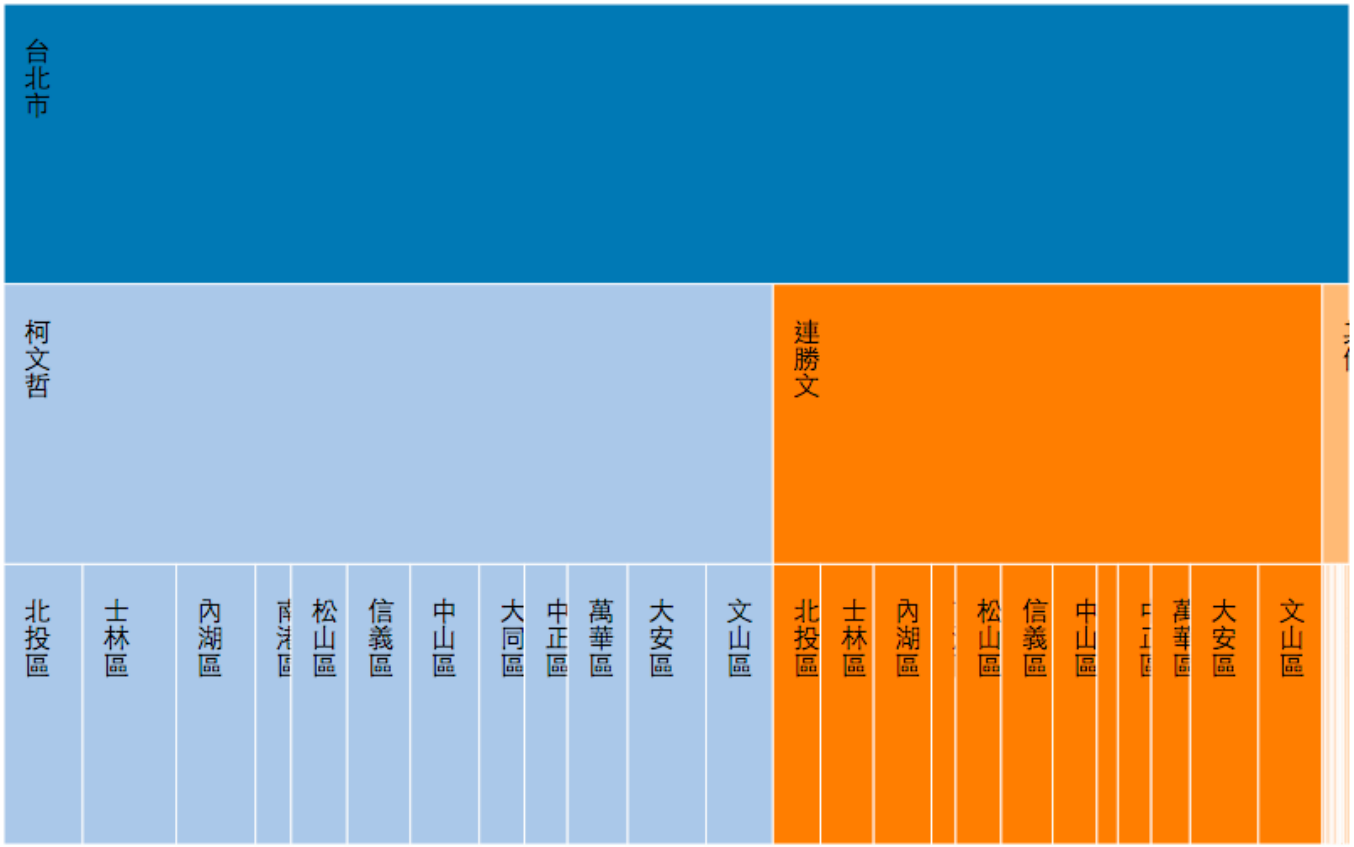
右圖範例資料: 2014 台北市長選舉各行政區投票統計


```
{
 "name": "flare",
 "children": [
  {
   "name": "analytics",
   "children": [
    {
     "name": "cluster",
     "children": [
      {"name": "AgglomerativeCluster", "size": 3938},
      {"name": "CommunityStructure", "size": 3812},
      {"name": "MergeEdge", "size": 743}
     ]
    },
    {
     "name": "graph",
     "children": [
      {"name": "BetweennessCentrality", "size": 3534},
      {"name": "LinkDistance", "size": 5731}
     ]
    }
   ]
  }
 ]
}
```



```
{
   name: "台北市",
 ▼ children: [
    ▼ {
         name: "柯文哲",
       ▼ children: [
          ▼ {
               name: "北投區",
               value: 87646
            },
          ▼ {
               name: "士林區",
               value: 103891
            },
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…},
          ▶ {…}
         ]
      },
    ▼ {
         name: "連勝文",
       ▼ children: [
          ▼ {
               name: "北投區",
               value: 52728
```

Partition Layout

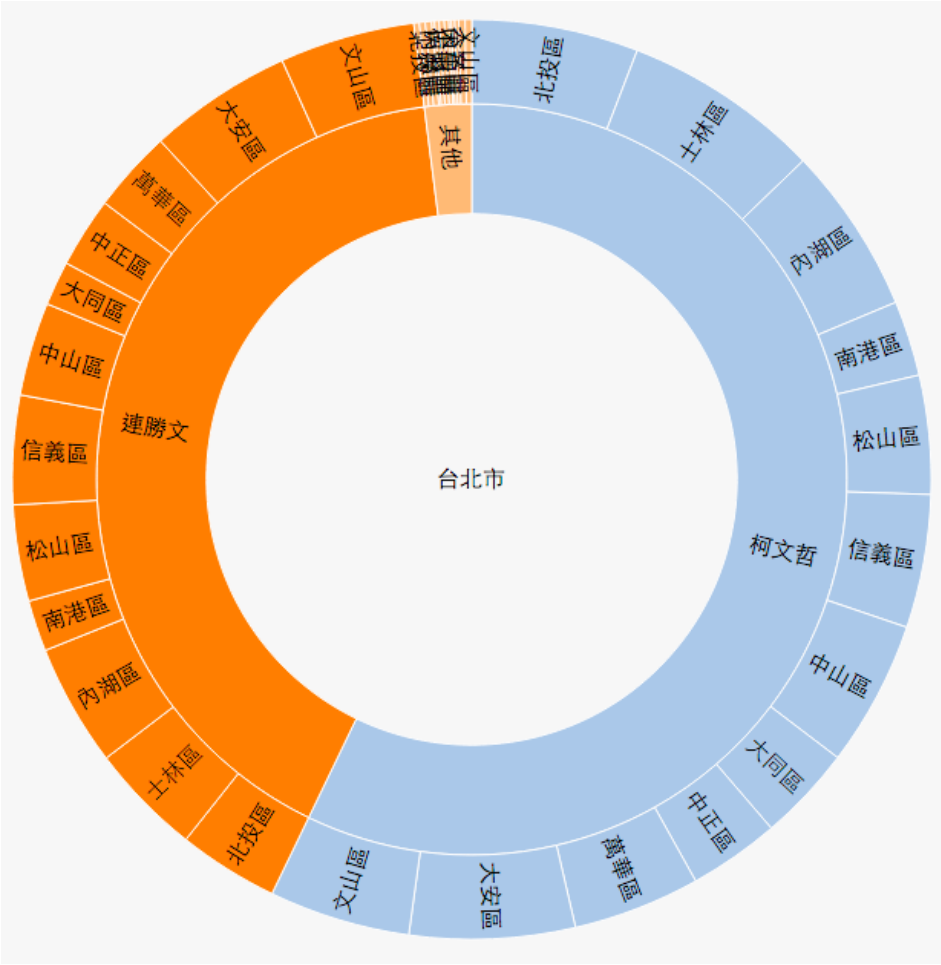


Partition Layout

```
// 矩形階層圖: d3.layout.partition()
// size 指定圖案大小, 利用 .nodes 帶入資料
var nodes = d3.layout.partition()
                .size([600, 400])       // [width, height]
                .nodes(data);

// 將生成好的 nodes 透過 data() 帶入
var gRects = svg.selectAll("g").data(nodes).enter().append("g");
// 加入 rect
gRects.append("rect").attr({ ... }).style({ ... });
// 加入文字
gRects.append("text").attr({ ... }).style({ ... });
```

Partition Layout

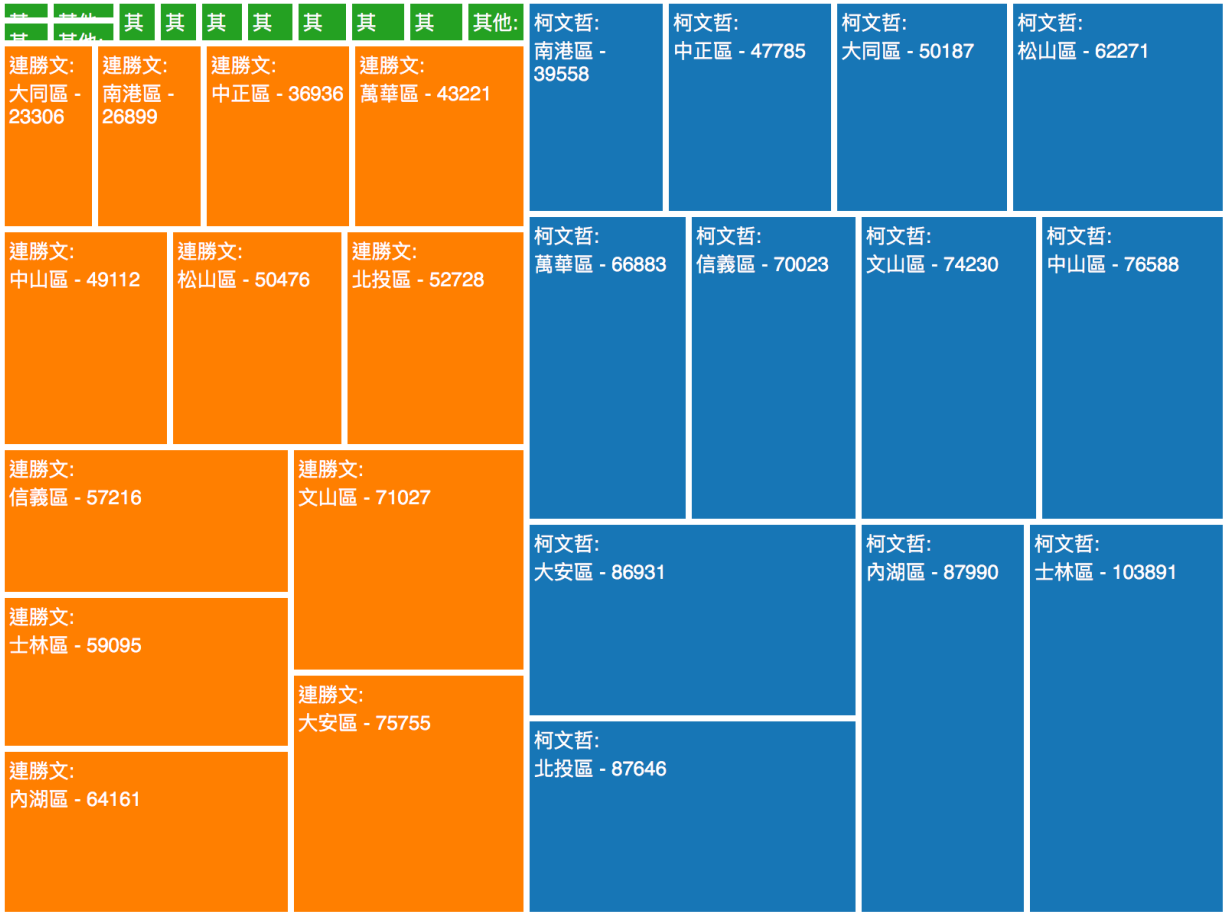


Partition Layout

```
// 圓形階層圖: d3.layout.partition()
// size 指定圖案大小, 利用 .nodes 帶入資料
var nodes = d3.layout.partition()
              .size([2 * Math.PI, radius * radius])
              .nodes(data);

// d3.svg.arc 製作弧形
var arc = d3.svg.arc()
    .startAngle(function (d) { return d.x; })
    .endAngle(function (d) { return d.x + d.dx; })
    .innerRadius(function (d) { return Math.sqrt(d.y); })
    .outerRadius(function (d) { return Math.sqrt(d.y + d.dy); });
```

Treemap Layout



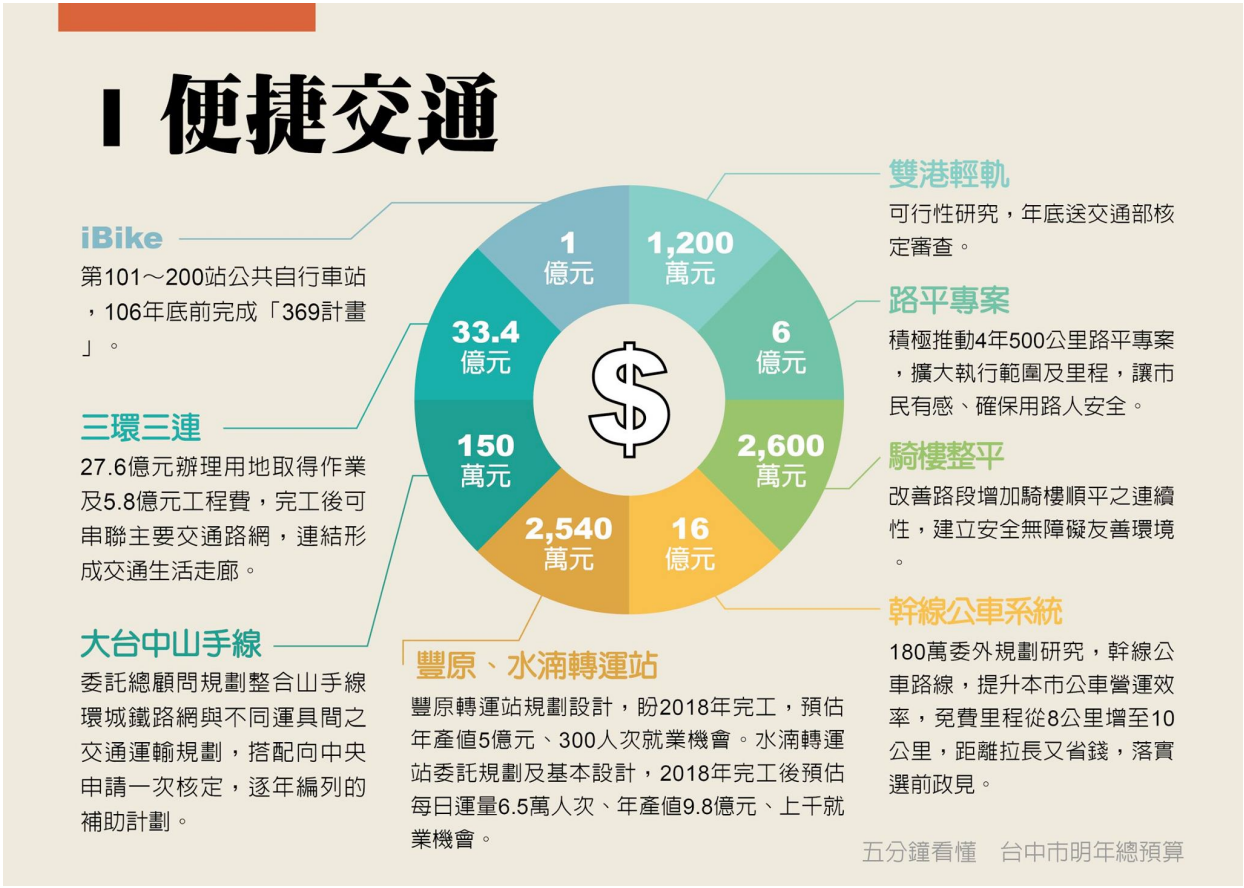
Treemap Layout

```
// d3.layout.treemap(),
// size 指定圖案大小, nodes 將資料帶入
var nodes = d3.layout.treemap()
              .size([width, height])
              .nodes(data);
```

Treemap Layout

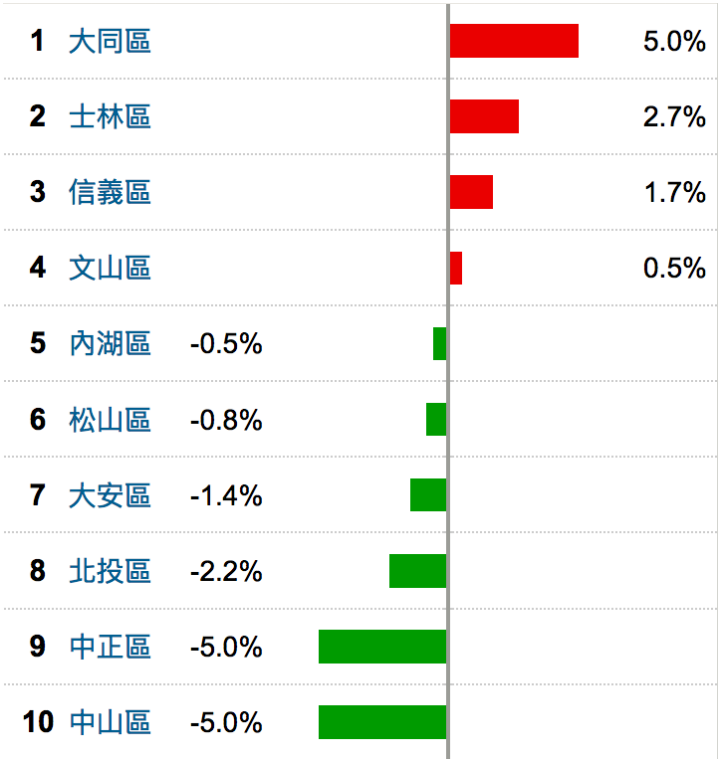
```
// 此範例不用 svg, 改以 div 實作
var rects = groups.append("div")          // 填入 div
      .attr({ "class": "nodeRect", })     // 設定 class
      .style({                            // 指定寬高, 位置與顏色
        "top":    function(d) { return d.y + 'px'; },
        "left":   function(d) { return d.x + 'px'; },
        "width":  function(d) { return d.dx + 'px'; },
        "height": function(d) { return d.dy + 'px'; },
        "background-color":
          function(d){ return color(d.parent.name); }
      })
      .html({ … }});                      // 加入文字
```

視覺化不是有圖就好

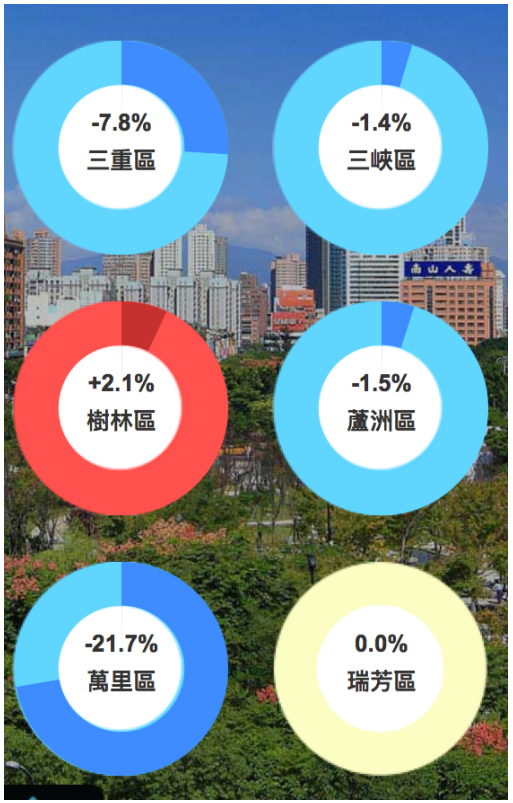


圖片來源：
https://www.facebook.com/forpeople/photos/a.989035831154301.1073741835.153819538009272/989036197820931/

視覺化不是漂亮就好



(永慶房仲網行情頻道)

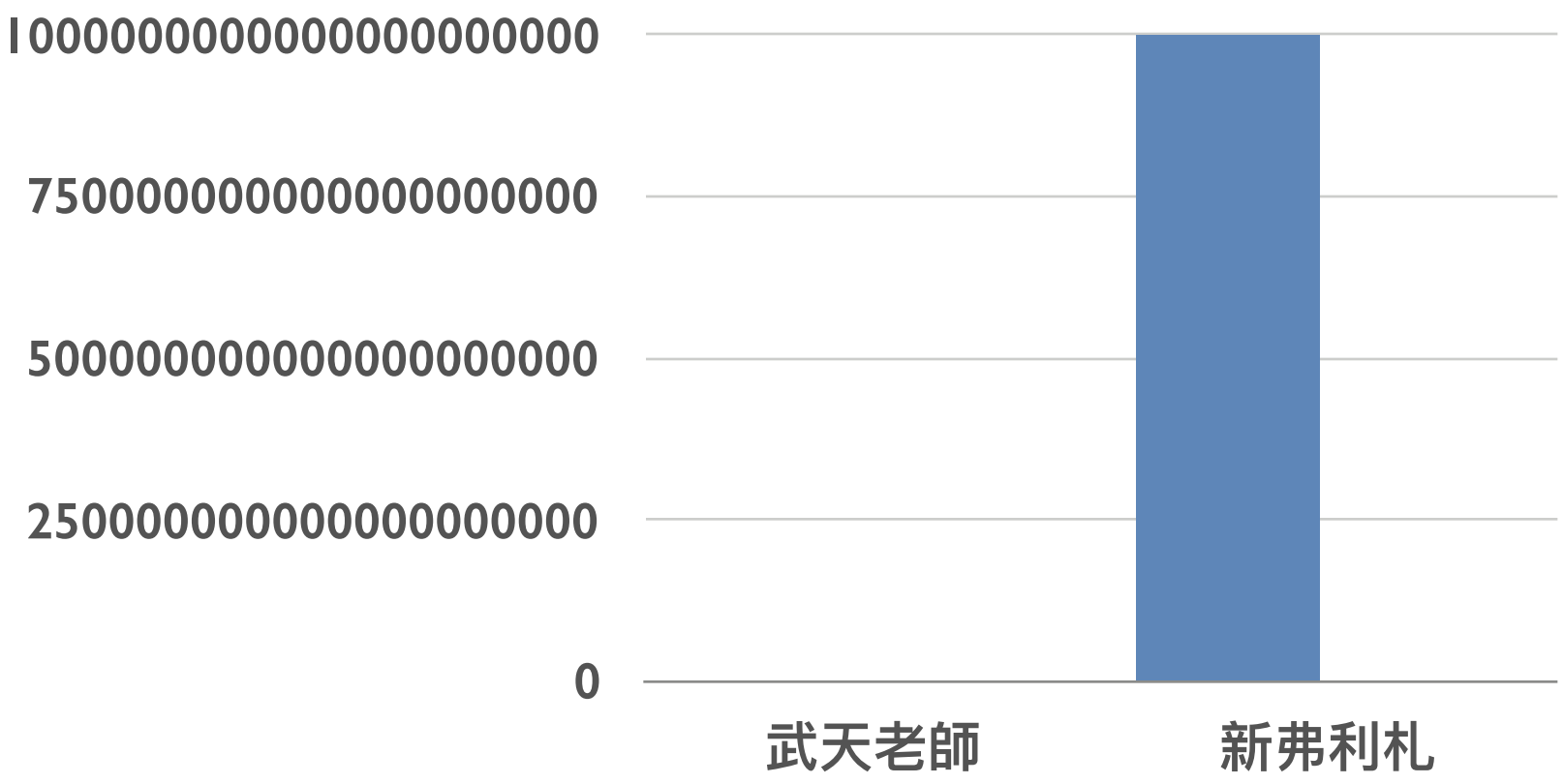


慎選主題


變成容易了解的表示文字了…
呃…139…
嘩嘩…
這就是武天師父的戰鬥力了！


最恐フリーザ 戦闘力は
100,000,000,000,000,000,000!!!!!

慎選主題


100000000000000000000
75000000000000000000
50000000000000000000
25000000000000000000
0
武天老師
新弗利札

結語

- D3 Selectors
- Data-Binding (Enter-Update-Exit)
- Scales / Axis
- D3 Events
- SVG Basic
- D3 Layouts

感謝 KKTIX 贊助活動報名平台



http://kktix.com

感謝網際智慧贊助自然輸入法



感謝 Jetbrains 贊助贈品



ReSharper

https://www.jetbrains.com/resharper/

感謝 OzCode 贊助贈品



http://www.oz-code.com/

業界師資、實戰教學



http://skilltree.my

謝謝各位

http://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.