

# D3.JS

## 互動式資料視覺化

**Lecturer: LinJer 林哲**

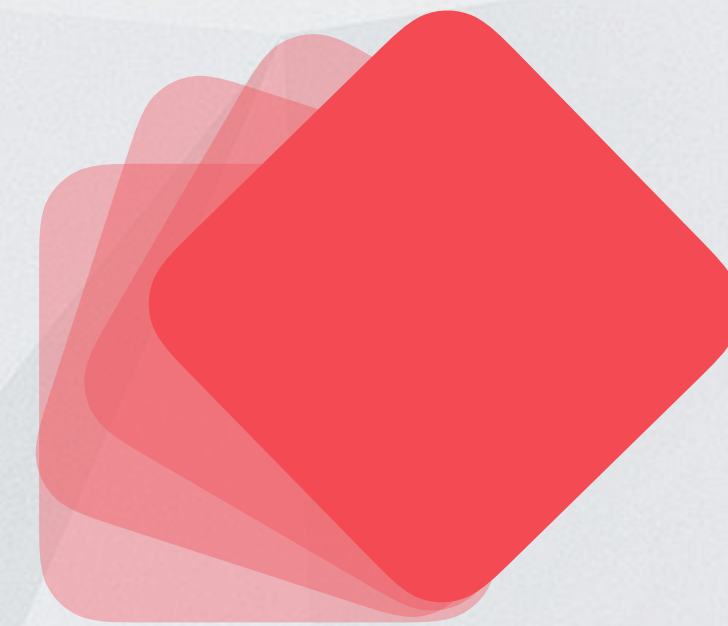
evin92@gmail.com





# 用D3玩動畫

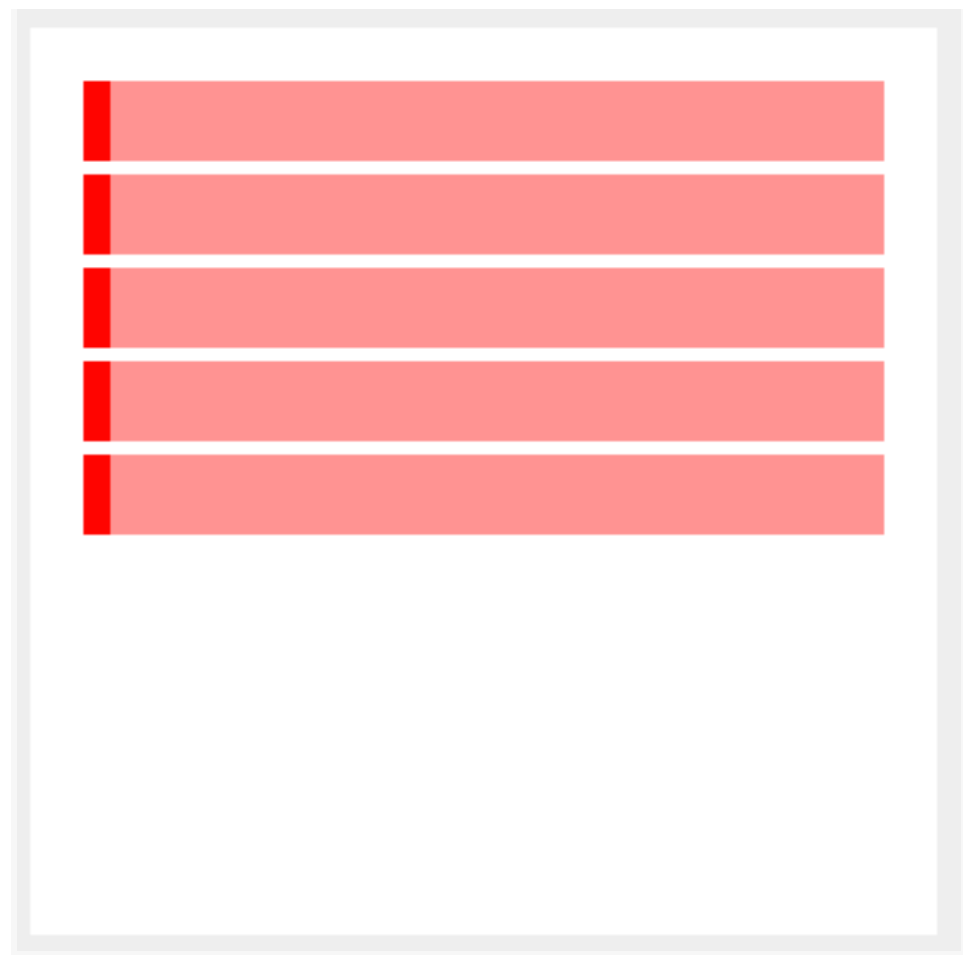
## 讓視覺元素動起來





# 基本D3動畫

## 開始與結束兩個狀態



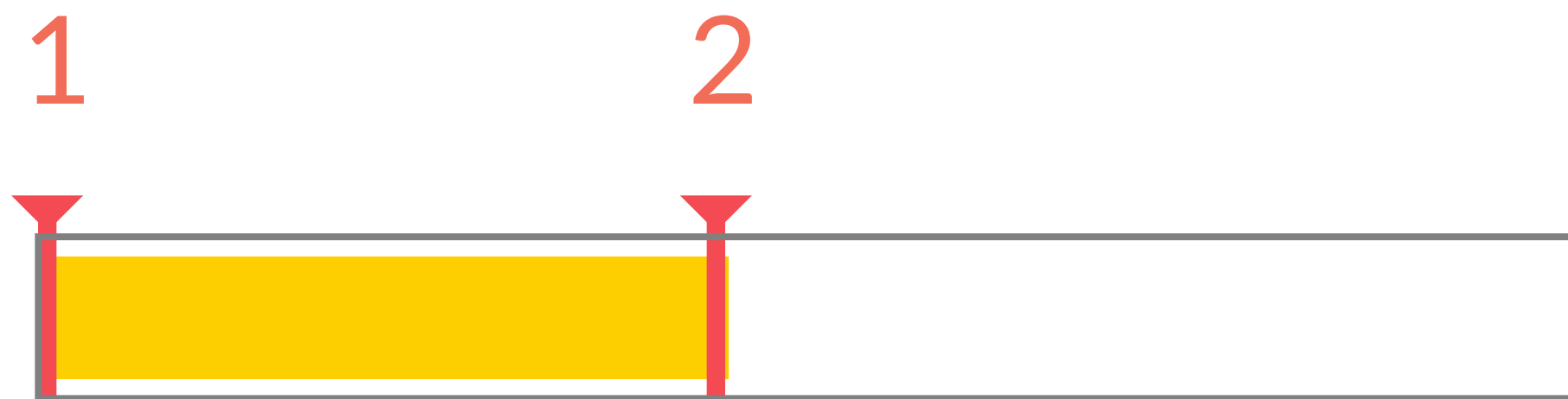
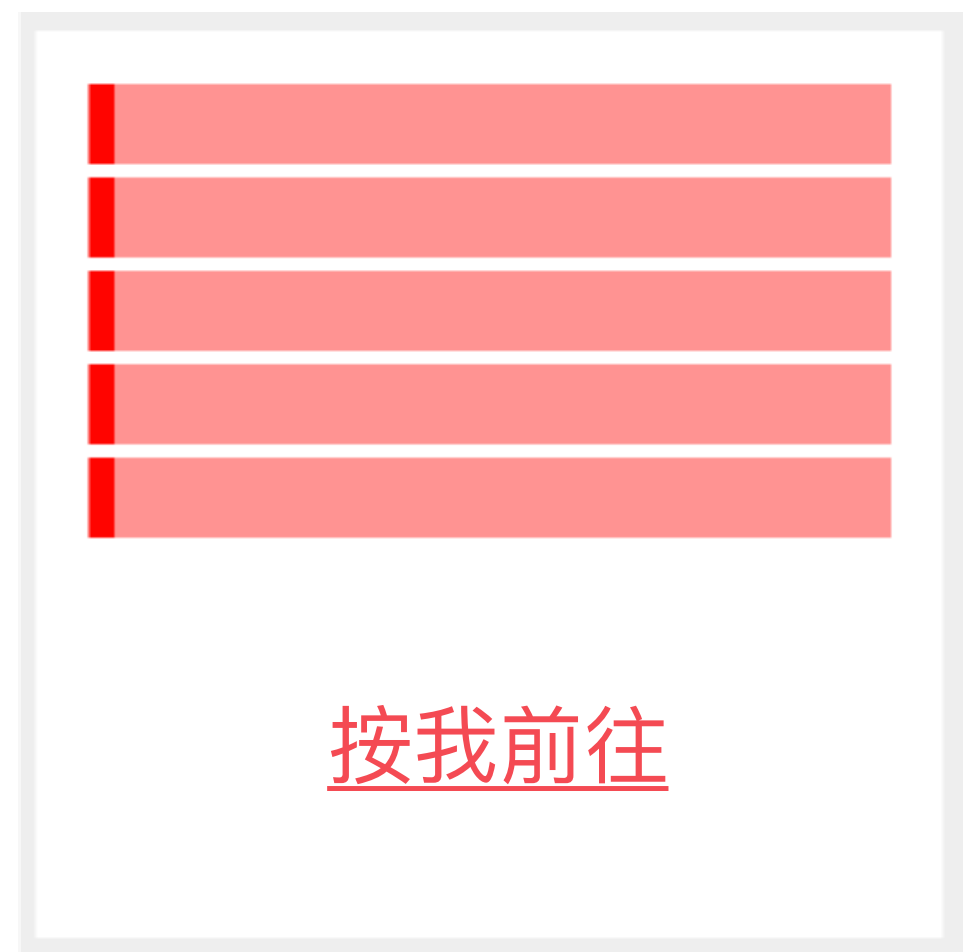
[注意]屬性需有具體值才會有效果

```
d3.selection.transition()
```

```
d3.selectAll("rect#bar")  
  .attr({width: 10})  
  .transition()  
  .attr({width: 300});
```

# 基本D3動畫

## 開始與結束兩個狀態



`d3.selection.transition()`

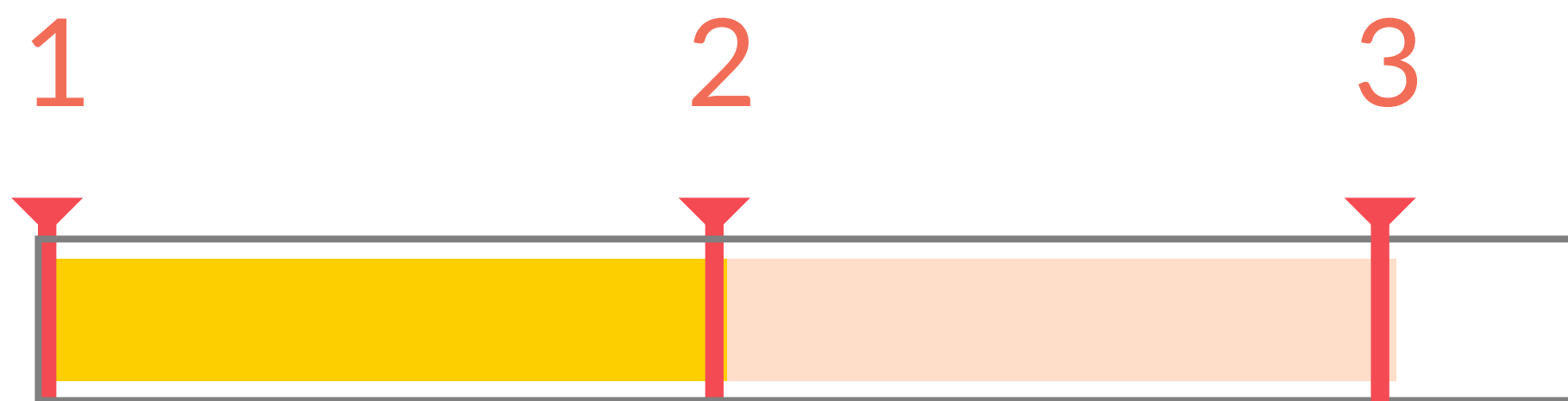
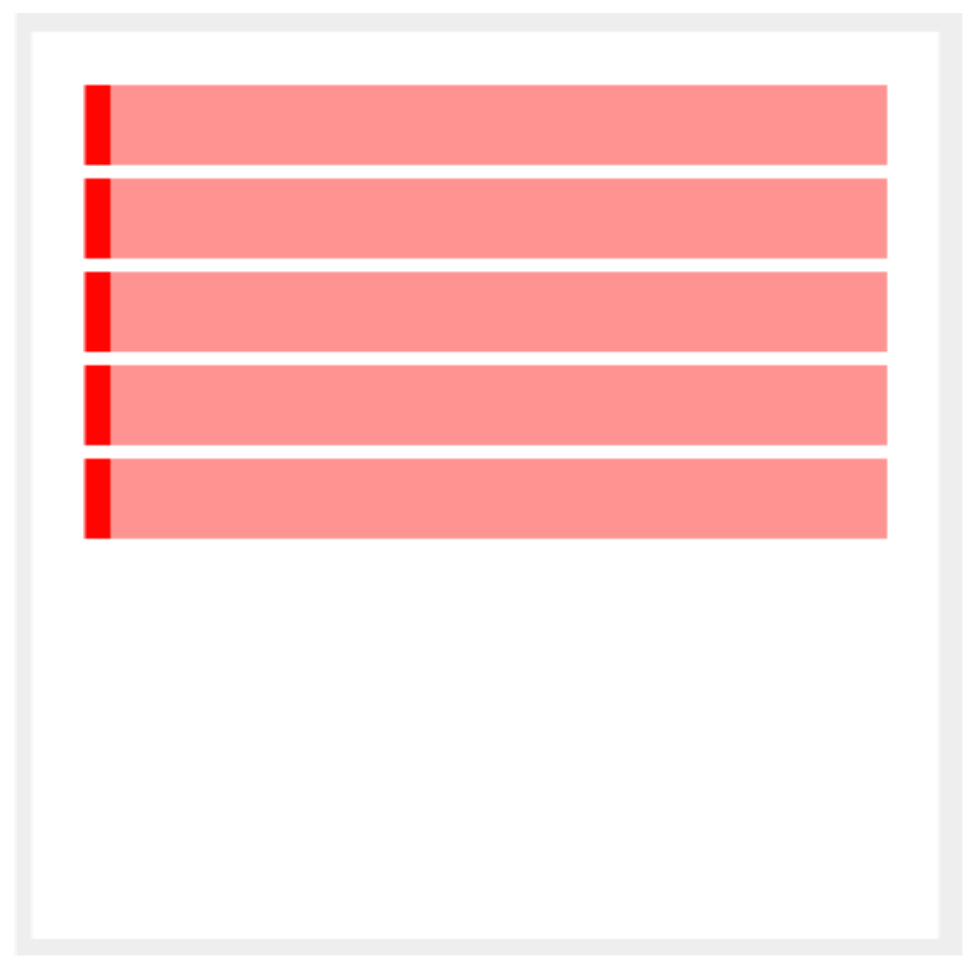
也可以分開設定

1  $\longrightarrow$  `d3.selectAll("rect#bar")  
                  .attr({width: 10})`

`d3.selectAll("rect#bar")  
          .transition()  
2  $\longrightarrow$  .attr({width: 300});`

# 基本D3動畫

## 開始與結束兩個狀態



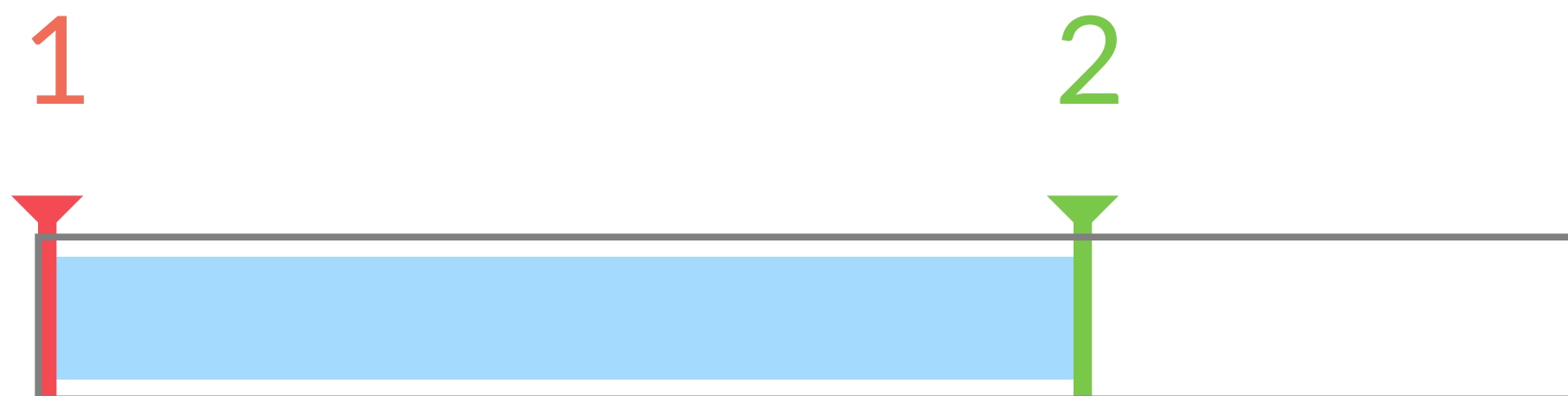
`d3.selection.transition()`

可以接續串接成連續動畫

```
d3.selectAll("rect#bar")  
  .transition().attr({width: 300})  
  .transition().attr({width: 100});
```

# 基本D3動畫

## 可設定動畫時間長度



```
d3.selection  
.transition().duration(ms)
```

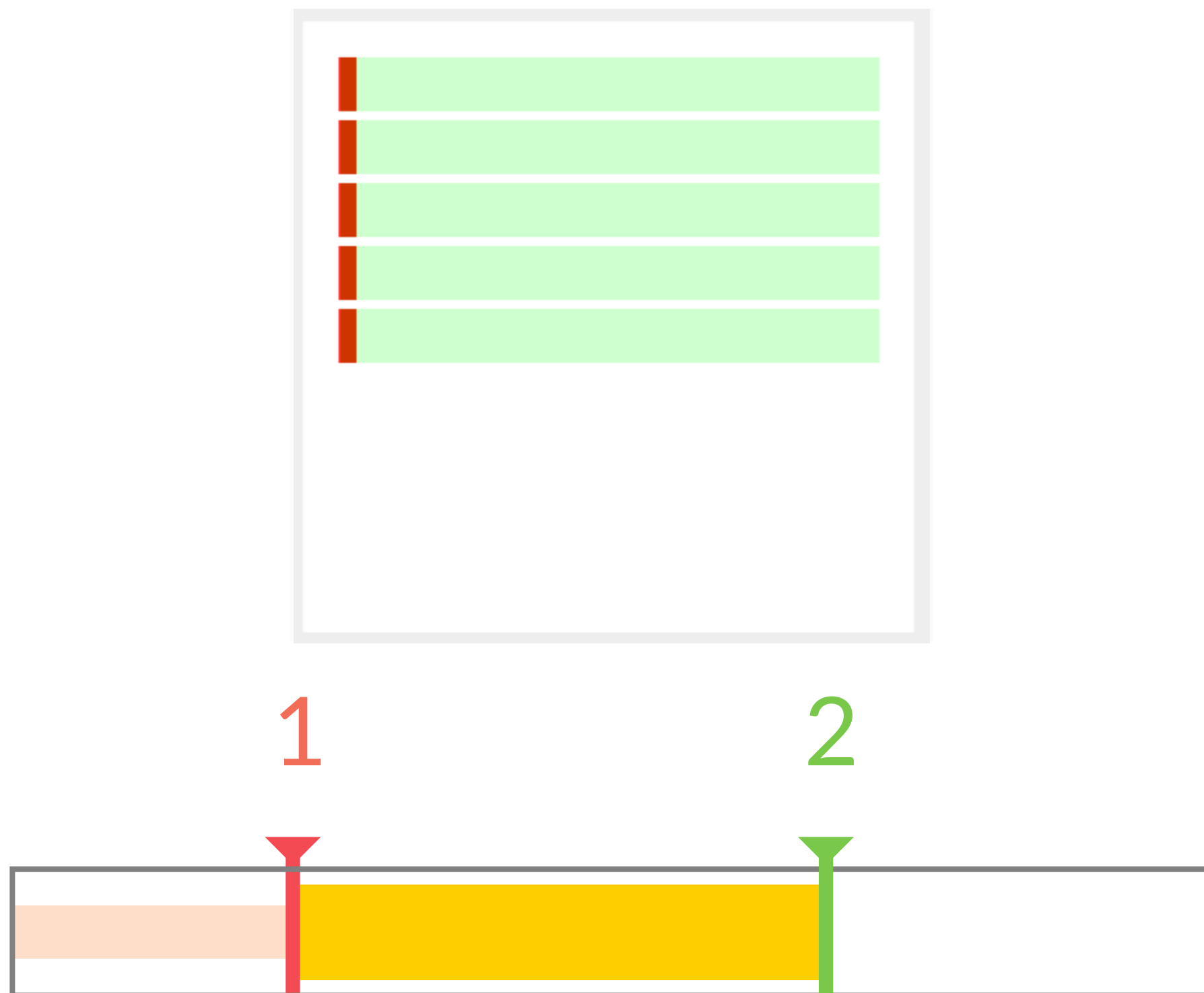
```
d3.selectAll("rect#bar")  
  .transition().duration(1000)  
  .attr({fill: "rgb(0,255,0)"});
```

# 基本D3動畫

可設定延遲播放時間

```
d3.selection  
.transition().delay(ms)
```

```
d3.selectAll("rect#bar")  
  .transition().delay(1000)  
  .attr({fill: "rgb(0,255,0)"});
```

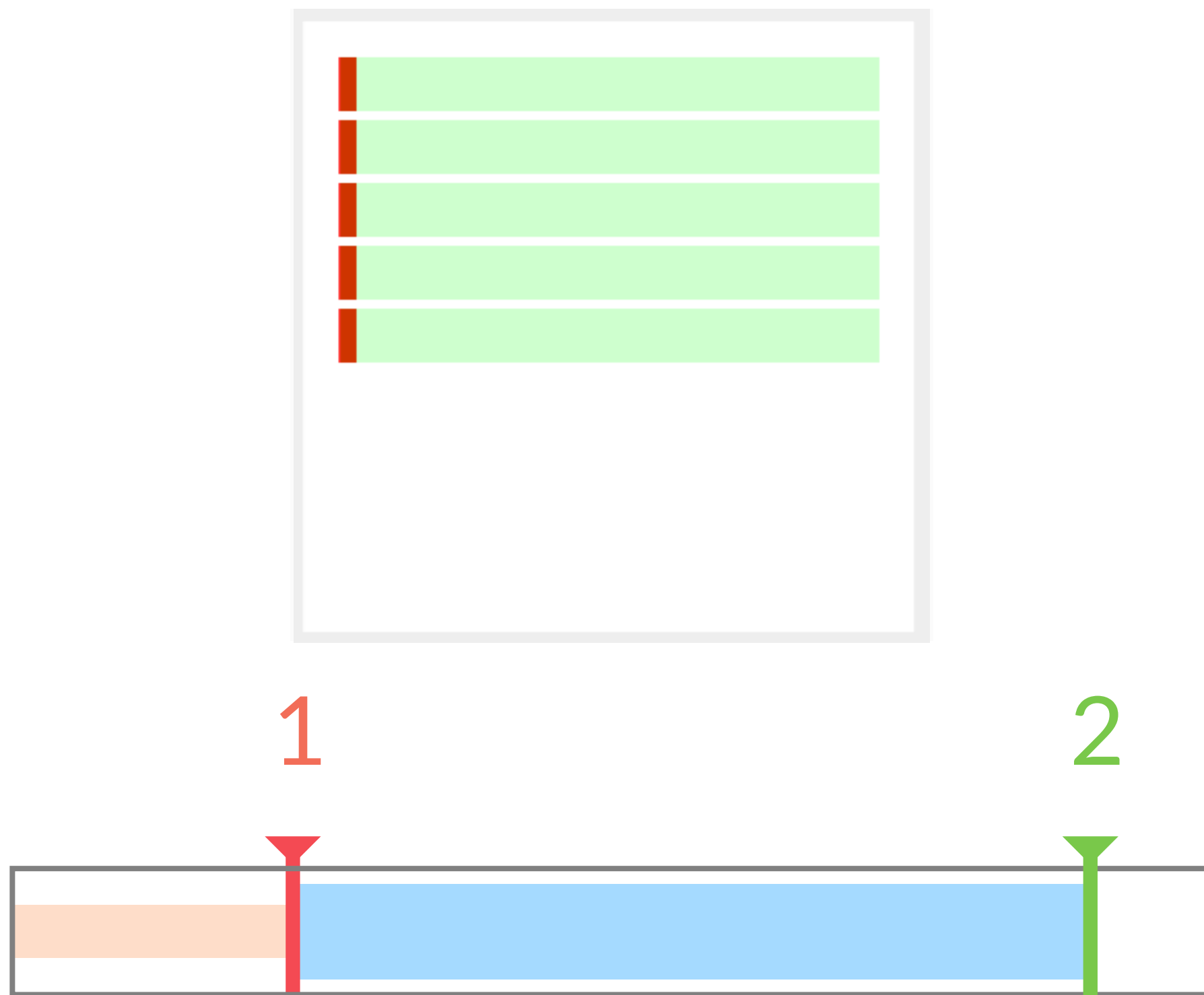


# 基本D3動畫

## 同時設定延遲與長度

```
d3.selection  
.transition().delay(ms)
```

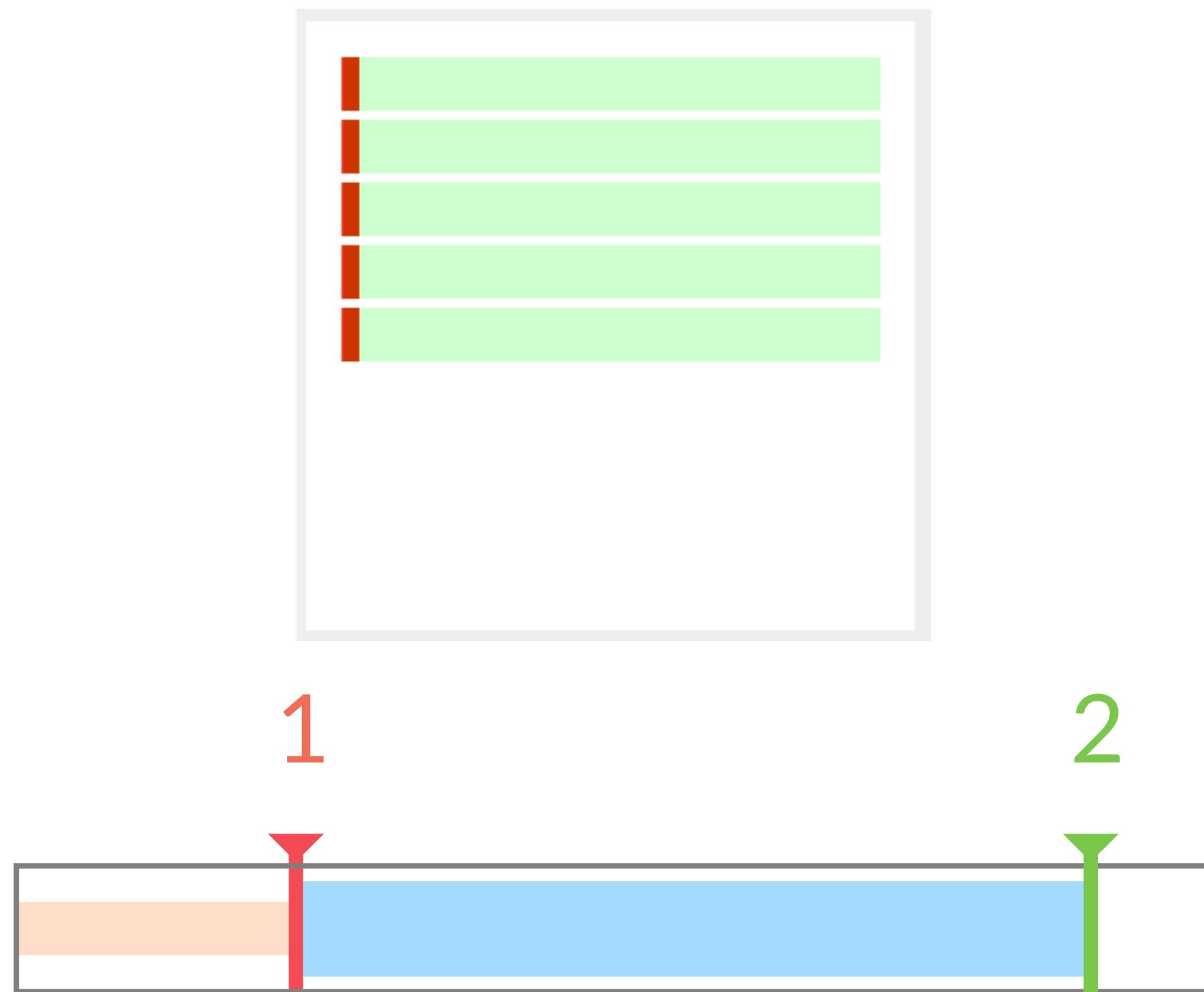
```
d3.selectAll("rect#bar")  
  .transition()  
  .duration(1000).delay(1000)  
  .attr({fill: "rgb(0,255,0)"});
```





# 基本D3動畫

## 同時設定延遲與長度-時間差



```
d3.selection  
  .transition().delay(ms)
```

括號可以放函式(取得綁定資料)

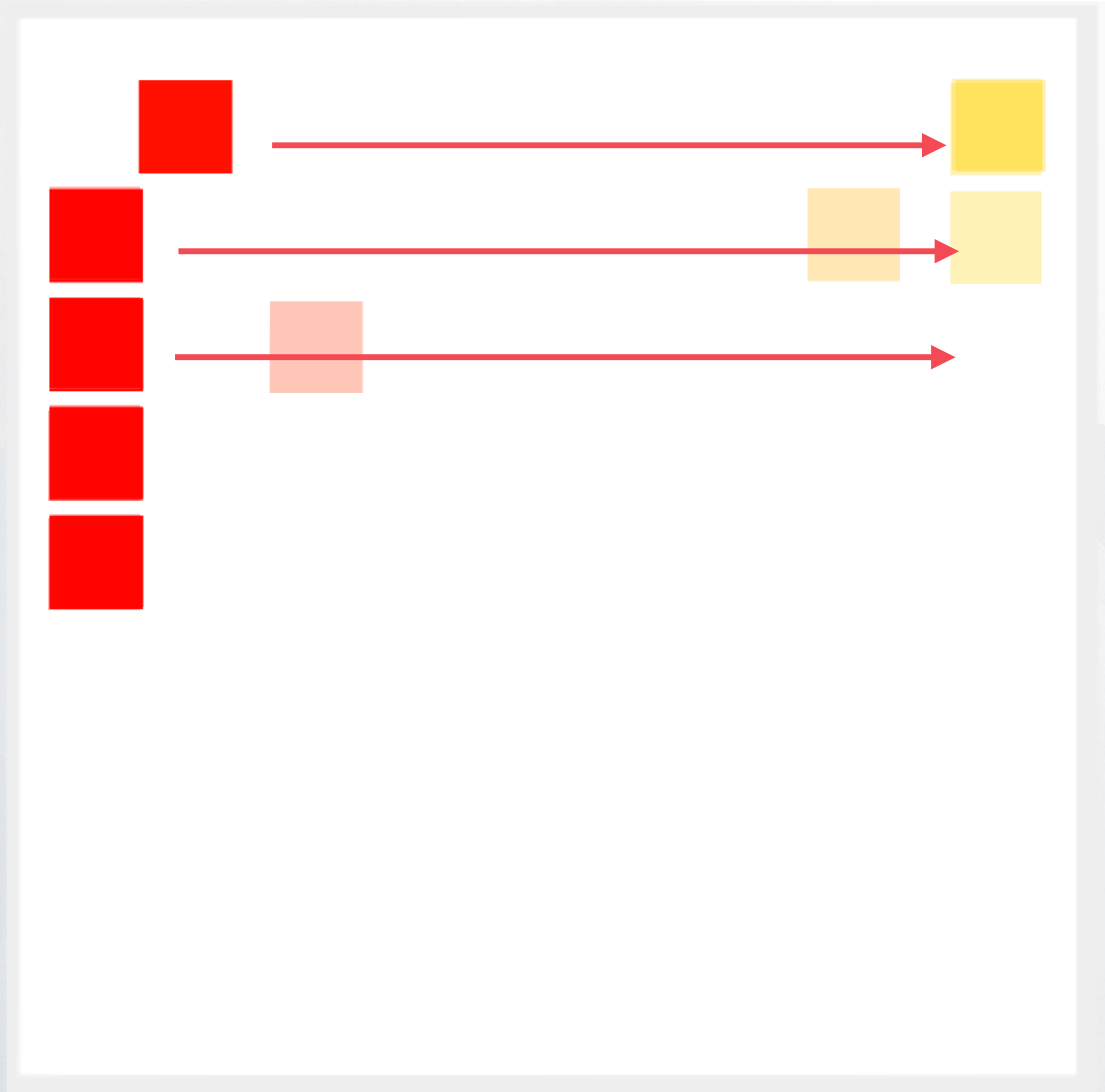
```
d3.selectAll("rect#bar")  
  .transition()  
  .duration(1000)  
  .delay(function(d, i){  
    return i*500;  
  })  
  .attr({fill: "rgb(0,255,0)"});
```

# 實作一群 由左至右的滑動方塊

## [要求]

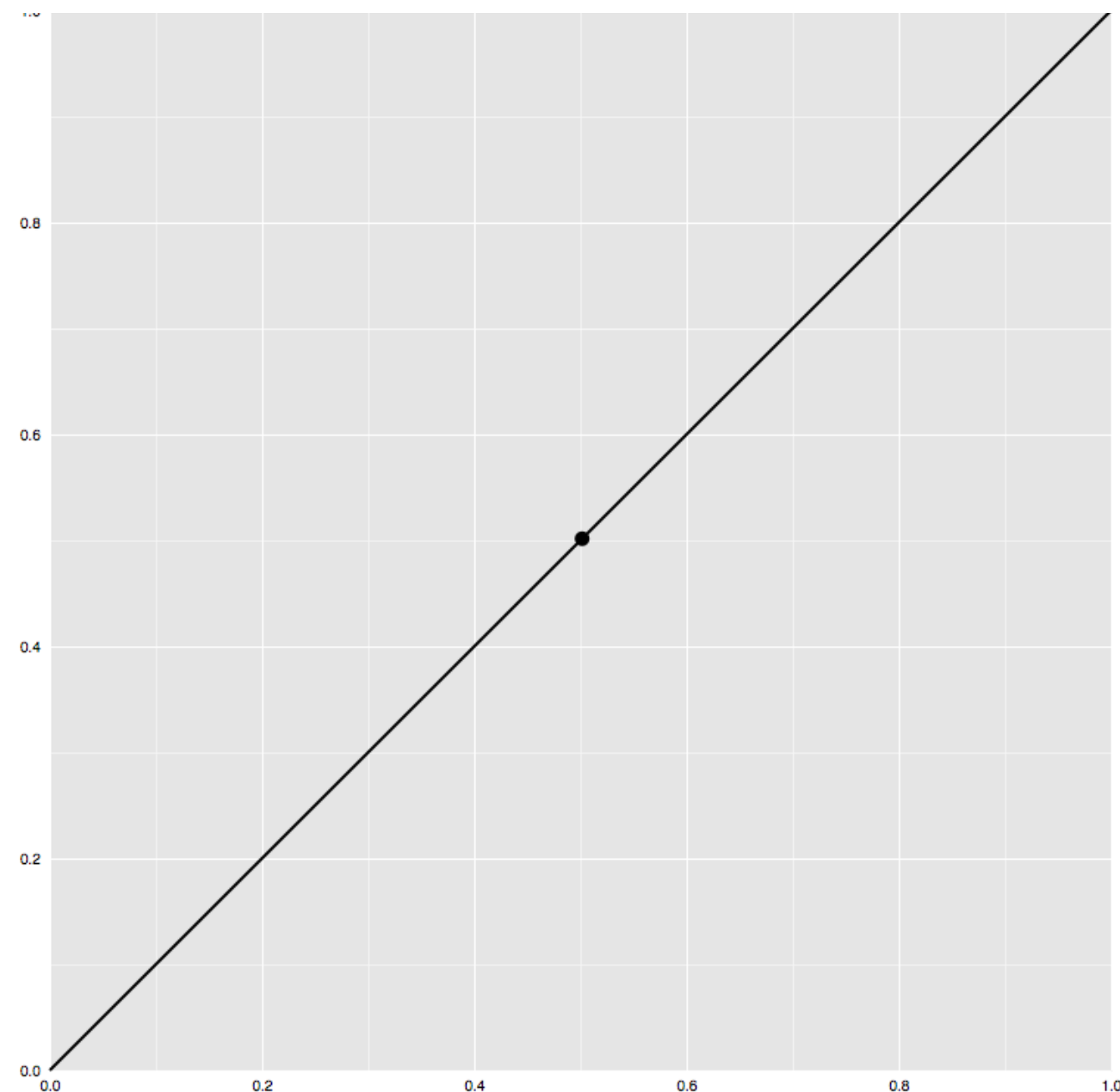
1. 每個移動時間1s
2. 一個到位下一個才能動
3. 顏色從 red 變成 gold

從我開始



# 基本D3動畫

## 可設定元件速度函式



什麼是速度函式?

```
d3.selection  
.transition().ease("linear")
```

```
d3.selectAll("rect#bar")  
.transition()  
.duration(1000)  
.ease("bounce")  
.attr({x: "300"});
```

[查看上一頁解答](#)

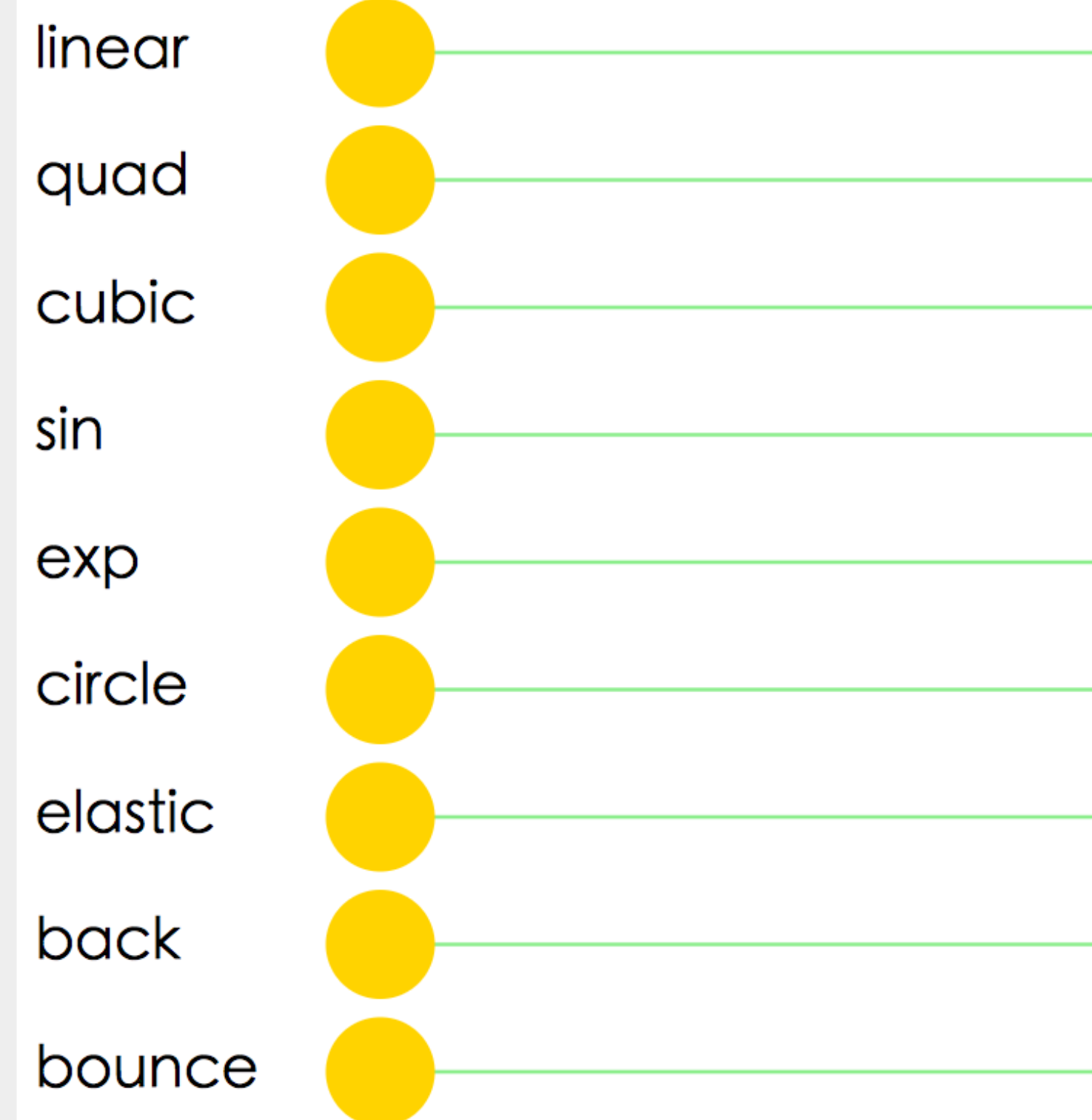




# 實作一個速度函式滑桿 `.transition().ease("linear")`

按下: 播放動畫、在右邊按會返回

[按我開始練習](#)





# 用D3玩動畫 Transition

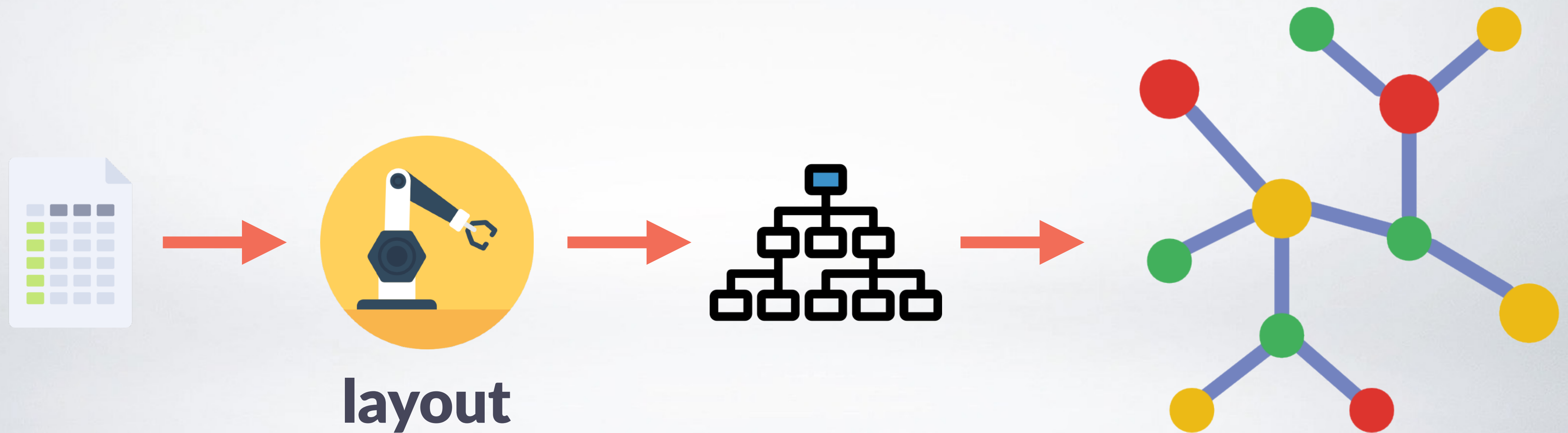
[查看解答](#)

1. 開始與結束兩個狀態 -> `transition()`
2. 屬性之值平滑的轉變 -> `attr({ ... })`
3. 可設定動畫時間長度 -> `duration({ ... })`
4. 可設定延遲播放時間 -> `delay({ ... })`
5. 可設定元件速度函式 -> `ease("...")`



# D3-Layout 版型

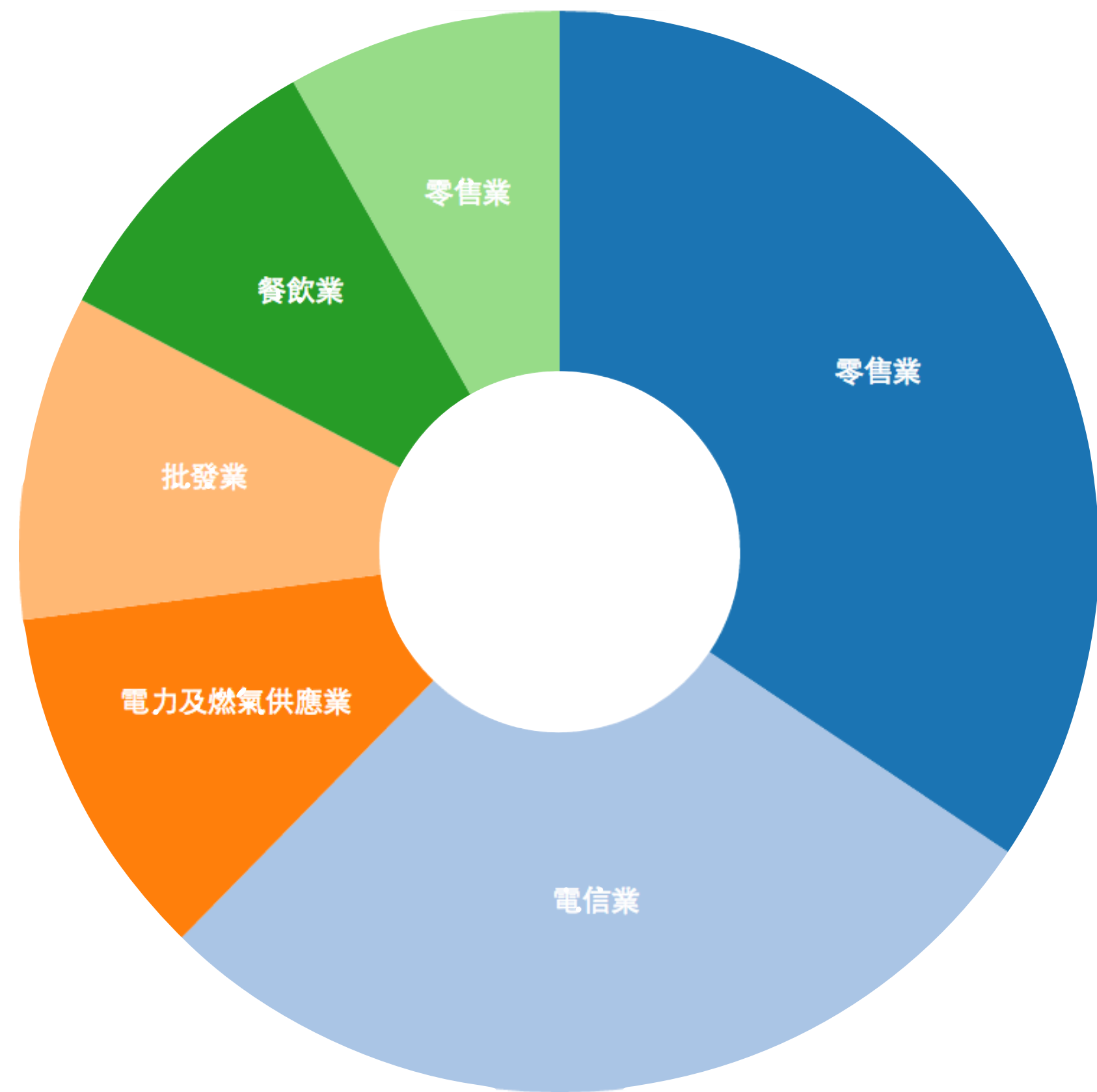
## 整理來源資料的好幫手





# Layout: Pie

## 圓餅版型



### d3.layout.pie()

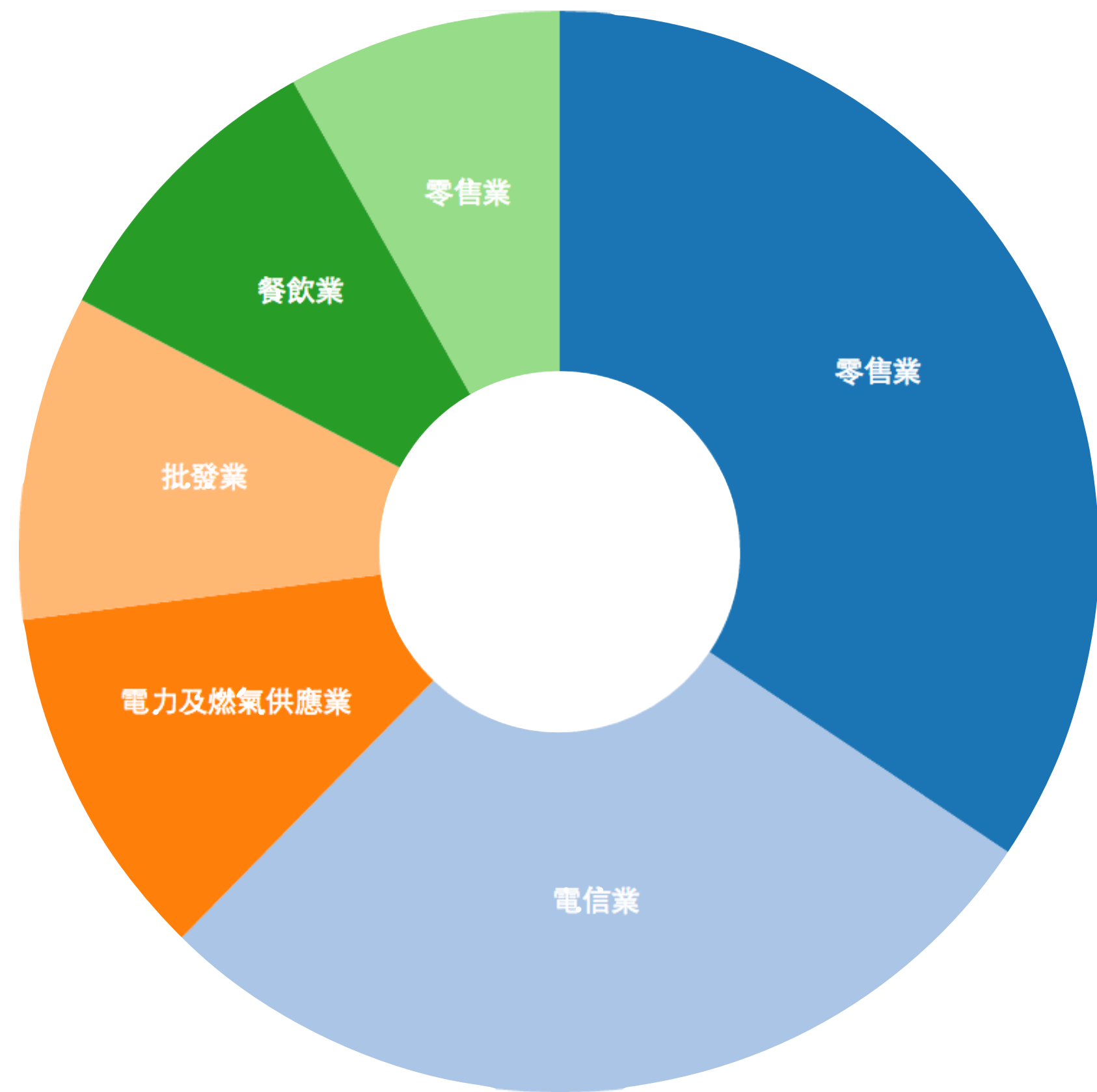
```
var pie = d3.layout.pie()  
    .value(function(d) {  
        return d.amount;  
    });
```

```
var selection = d3.select("svg")  
    .selectAll("g.arc")  
    .data(pie(dataSet));
```

[按我下載練習](#)

# Layout: Pie

## 圓餅版型



## 如何用svg畫圓餅圖

```
var arc = d3.svg.arc()  
    .outerRadius(300)  
    .innerRadius(100);
```

```
d3.selectAll("g.arc")  
    .select("path")  
    .attr("d", arc)
```



升序排序

`d3.ascending(a, b)`



1  
9

# D3排序

降序排序

`d3.descending(a, b)`



9  
1

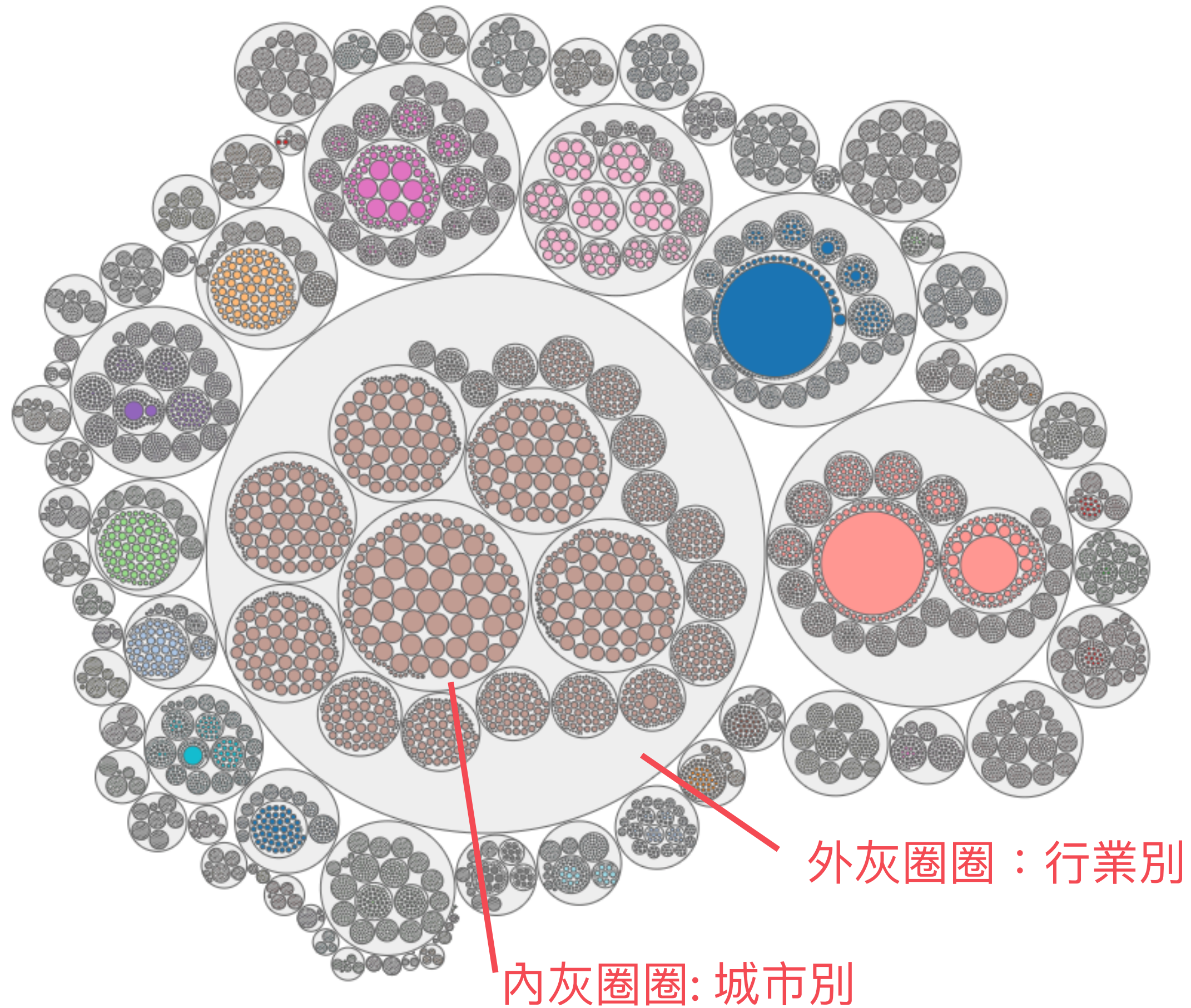
```
var ARRAY = [{num: 4},{num: 6},{num: 8}];  
  
var newArr = ARRAY.sort(function(a, b){  
    return d3.descending(a.num, b.num);  
});
```

[查看範例](#)



# Layout: Pack

## 泡泡版型



`d3.layout.pack()`

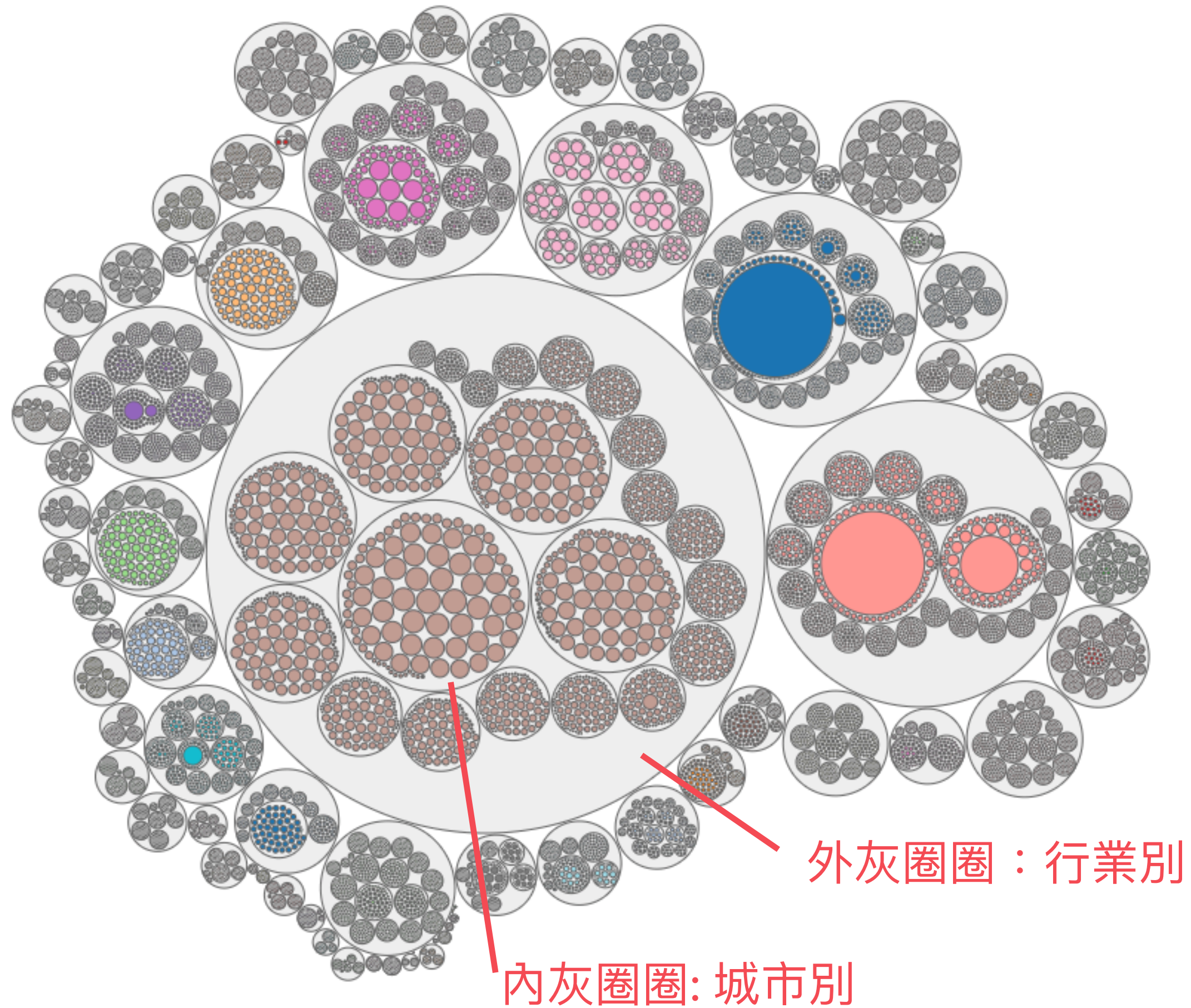
```
d3.layout.pack()  
  .size([width, height])  
  .children( ... )  
  .nodes(root)
```

[按我下載練習](#)



# Layout: Pack

## 泡泡版型



d3.layout.pack()

資料需要先是這樣的格式

```
var root = {  
  value: (數值大小),  
  children: [  
    {value: ... , children},  
    {...},  
    .....  
  ]  
}
```



巢狀:

**d3.nest()**

**.key()** → 分類依據

**.entries()** → 來源資料

key-value pair  
鍵-值 對

```
var arr = [  
  {value: 1, char: "A", num: 1},  
  {value: 2, char: "B", num: 1},  
  {value: 3, char: "B", num: 2},  
  {value: 4, char: "C", num: 1},  
  {value: 5, char: "C", num: 2},  
  {value: 6, char: "C", num: 3},  
];
```

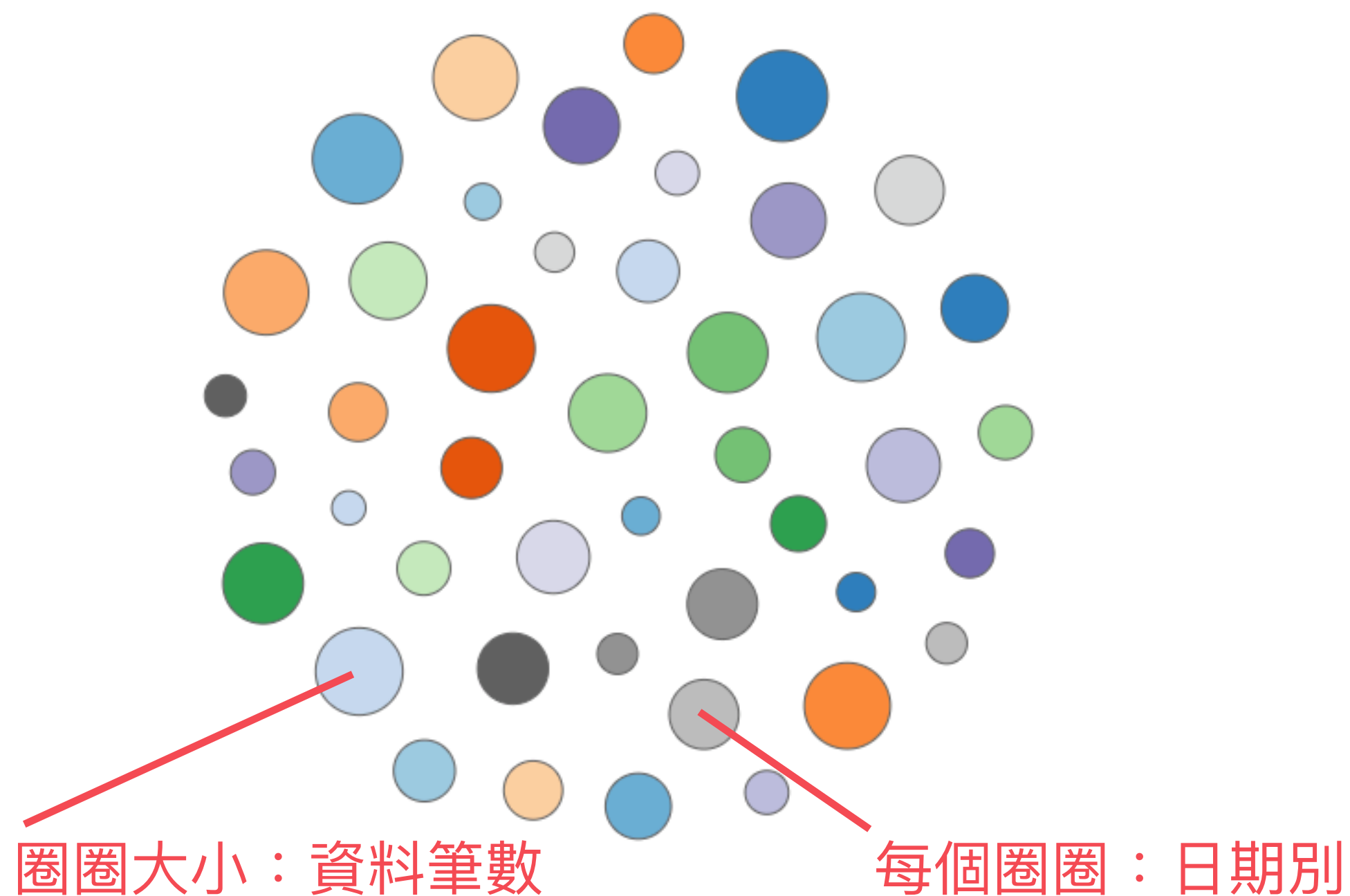
```
var nested_a = d3.nest()  
  .key(function(d){  
    return d.char;  
  }).entries(arr);
```

[查看範例](#)



# Layout: Force

## 重力場版型



## d3.layout.force()

```
var force = d3.layout.force()  
  .size([w, h])  
  .charge(-30)  斥力(負值)  
  .friction(0.9)  摩擦力  
  .gravity(0.1)  重力  
  .nodes(nested_dataSet)  
  .on("tick", render)  
  .start();
```

[按我下載練習](#)



# 期末實戰專題 成果發表

2016/11/13 (日)

發表內容:

1. 作品簡述
2. 作法分享
3. 遭遇困難
4. Demo