

11장. 레이아웃

[정리]

1. 파이 레이아웃

D3.layout.pie()
D3.svg.arc()
 .innerRadius
 .outerRadius
*** grouping: transform ~ translate

2. 링 레이아웃

3. 스택 레이아웃

Dataset 형태
D3.layout.stack()
Y0, Y 구분하기

4. 포스레이아웃

Dataset 형태
D3.layout.force()
Tick 설정하기

0. 레이아웃 기본 구성

Dataset 형태

d3.layout.**레이아웃**

d3.svg.**도형**

도형에 대한 위치 지정

1. 파이 레이아웃 – d3.layout.pie() / d3.svg.arc()

```
19
20 //Width and height
21 var w = 300;
22 var h = 300;
23
24 var dataset = [5, 10, 20, 45, 8, 25 ];
25
26 var outerRadius = w / 2;
27 var innerRadius = 0;
28 var arc = d3.svg.arc()
29   .innerRadius(innerRadius)
30   .outerRadius(outerRadius);
31
32 var pie = d3.layout.pie();
33
34 //Easy colors accessible via a 10-step ordinal scale
35 var color = d3.scale.category10();
36
37 //Create SVG element
38 var svg = d3.select("body")
39   .append("svg")
40   .attr("width", w)
41   .attr("height", h);
42
43 //Set up groups
44 var arcs = svg.selectAll("g.arc")
45   .data(pie(dataset))
46   .enter()
47   .append("g")
48   .attr("class", "arc")
49   .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");
50
51 //Draw arc paths
52 arcs.append("path")
53   .attr("fill", function (d, i) {
54     return color(i);
55   })
56   .attr("d", arc);
57
58 //Labels
59 arcs.append("text")
60   .attr("transform", function (d) {
61     return "translate(" + arc.centroid(d) + ")";
62   })
63   .attr("text-anchor", "middle")
64   .text(function (d) {
65     return d.value;
66   });
67
```

Dataset의 형태는 리스트.

파이 레이아웃의 경우, `***arc(호)` 로 구성되며,
innerRadius와 outerRadius를 인자로 전달 받음.

1. 파이 레이아웃 – grouping: transform ~ translate

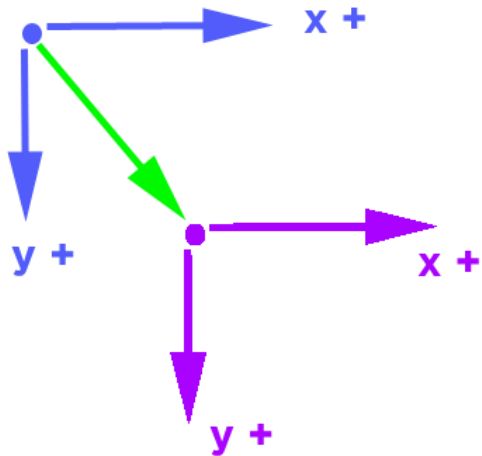
Ref: <https://www.dashingd3js.com/svg-group-element-and-d3js>

SVG group element → A container

1) Grouping: To group a set of SVG elements that share the same attribute

2) Transforming: To define a new coordinate system for a set of SVG elements by applying a transformation to each coordinate specified in this set of SVG elements

3) Translating



ex) moving the (0, 0) point of the blue coordinate system to the (0, 0) point of the purple system.

1. 파이 레이아웃 – grouping: transform ~ translate

```
19
20 //Width and height
21 var w = 300;
22 var h = 300;
23
24 var dataset = [5, 10, 20, 45, 8, 25 ];
25
26 var outerRadius = w / 2;
27 var innerRadius = 0;
28 var arc = d3.svg.arc()
29   .innerRadius(innerRadius)
30   .outerRadius(outerRadius);
31
32 var pie = d3.layout.pie();
33
34 //Easy colors accessible via a 10-step ordinal scale
35 var color = d3.scale.category10();
36
37 //Create SVG element
38 var svg = d3.select("body")
39   .append("svg")
40   .attr("width", w)
41   .attr("height", h);
42
43 //Set up groups
44 var arcs = svg.selectAll("g.arc")
45   .data(pie(dataset))
46   .enter()
47   .append("g")
48   .attr("class", "arc")
49   .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");
50
51 //Draw arc paths
52 arcs.append("path")
53   .attr("fill", function (d, i) {
54     return color(i);
55   })
56   .attr("d", arc);
57
58 //Labels
59 arcs.append("text")
60   .attr("transform", function (d) {
61     return "translate(" + arc.centroid(d) + ")";
62   })
63   .attr("text-anchor", "middle")
64   .text(function (d) {
65     return d.value;
66   });
67
```

모든 arc들을 그룹핑한 후,
각각의 호를 svg 컨테이너의 ***중심점으로 모음.

1. 파이 레이아웃 – path (경로정보) 정의

```
19
20 //Width and height
21 var w = 300;
22 var h = 300;
23
24 var dataset = [5, 10, 20, 45, 8, 25 ];
25
26 var outerRadius = w / 2;
27 var innerRadius = 0;
28 var arc = d3.svg.arc()
29     .innerRadius(innerRadius)
30     .outerRadius(outerRadius);
31
32 var pie = d3.layout.pie();
33
34 //Easy colors accessible via a 10-step ordinal scale
35 var color = d3.scale.category10();
36
37 //Create SVG element
38 var svg = d3.select("body")
39     .append("svg")
40     .attr("width", w)
41     .attr("height", h);
42
43 //Set up groups
44 var arcs = svg.selectAll("g.arc")
45     .data(pie(dataset))
46     .enter()
47     .append("g")
48     .attr("class", "arc")
49     .attr("transform", "translate(" + ou
50
51 //Draw arc paths
52 arcs.append("path")
53     .attr("fill", function (d, i) {
54         return color(i);
55     })
56     .attr("d", arc);
57
58 //Labels
59 arcs.append("text")
60     .attr("transform", function (d) {
61         return "translate(" + arc.centroid(d) + ")";
62     })
63     .attr("text-anchor", "middle")
64     .text(function (d) {
65         return d.value;
66     });
67
```

Path 문서요소의 경로 정보 정의는,
D라는 속성에 하고, 그 안에서 arc 생성 함수를 호출.

1. 파이 레이아웃 – arc.centroid()

```
19
20 //Width and height
21 var w = 300;
22 var h = 300;
23
24 var dataset = [5, 10, 20, 45, 8, 25 ];
25
26 var outerRadius = w / 2;
27 var innerRadius = 0;
28 var arc = d3.svg.arc()
29     .innerRadius(innerRadius)
30     .outerRadius(outerRadius);
31
32 var pie = d3.layout.pie();
33
34 //Easy colors accessible via a 10-step ordinal scale
35 var color = d3.scale.category10();
36
37 //Create SVG element
38 var svg = d3.select("body")
39     .append("svg")
40     .attr("width", w)
41     .attr("height", h);
42
43 //Set up groups
44 var arcs = svg.selectAll("g.arc")
45     .data(pie(dataset))
46     .enter()
47     .append("g")
48     .attr("class", "arc")
49     .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");
50
51 //Draw arc paths
52 arcs.append("path")
53     .attr("fill", function (d, i) {
54         return color(i);
55     })
56     .attr("d", arc);
57
58 //Labels
59 arcs.append("text")
60     .attr("transform", function (d) {
61         return "translate(" + arc.centroid(d) + ")";
62     })
63     .attr("text-anchor", "middle")
64     .text(function (d) {
65         return d.value;
66     });
67
```

*** arc.centroid() → 각 호의 중심점을 계산
*** d.value() → arcs.data 가 pie(dataset)이기 때문.

2. 링 레이아웃

```
17 <body>
18 <script type="text/javascript">
19
20     //Width and height
21     var w = 300;
22     var h = 300;
23
24     var dataset = [5, 10, 20, 45, 8, 25 ];
25
26     var outerRadius = w / 2;
27     var innerRadius = w / 3;
28     var arc = d3.svg.arc()
29         .innerRadius(innerRadius)
30         .outerRadius(outerRadius);
31
32     var pie = d3.layout.pie();
33
34     //Easy colors accessible via a 10-step ordinal scale
35     var color = d3.scale.category10();
36
37     //Create SVG element
38     var svg = d3.select("body")
39         .append("svg")
40         .attr("width", w)
41         .attr("height", h);
42
43     //Set up groups
44     var arcs = svg.selectAll("g.arc")
45         .data(pie(dataset))
46         .enter()
47         .append("g")
48         .attr("class", "arc")
49         .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");
50
51     //Draw arc paths
52     arcs.append("path")
53         .attr("fill", function (d, i) {
54             return color(i);
```

innerRadius를 조정하여 링 차트를 만듦

3. 스택 레이아웃 – dataset 형태

```
//Original data
var dataset = [
  [
    { x: 0, y: 5 },
    { x: 1, y: 4 },
    { x: 2, y: 2 },
    { x: 3, y: 7 },
    { x: 4, y: 23 }
  ],
  [
    { x: 0, y: 5 },
    { x: 1, y: 12 },
    { x: 2, y: 19 },
    { x: 3, y: 23 },
    { x: 4, y: 17 }
  ],
  [
    { x: 0, y: 22 },
    { x: 1, y: 28 },
    { x: 2, y: 32 },
    { x: 3, y: 35 },
    { x: 4, y: 43 }
  ]
];
```

```
> dataset
< [▼ Array[5], ▼ Array[5], ▼ Array[5]]
  ▼ 0: Object
    x: 0
    y: 5
    y0: 0
    ▶ __proto__: Object
  ▼ 1: Object
    x: 1
    y: 4
    y0: 0
    ▶ __proto__: Object
  ▼ 2: Object
    x: 2
    y: 2
    y0: 0
    ▶ __proto__: Object
  ▼ 3: Object
    x: 3
    y: 7
    y0: 0
    ▶ __proto__: Object
  ▼ 4: Object
    x: 4
    y: 23
    y0: 0
    ▶ __proto__: Object
    length: 5
  ▼ 0: Object
    x: 0
    y: 5
    y0: 5
    ▶ __proto__: Object
  ▼ 1: Object
    x: 1
    y: 12
    y0: 4
    ▶ __proto__: Object
  ▼ 2: Object
    x: 2
    y: 19
    y0: 2
    ▶ __proto__: Object
  ▼ 3: Object
    x: 3
    y: 23
    y0: 7
    ▶ __proto__: Object
  ▼ 4: Object
    x: 4
    y: 17
    y0: 23
    ▶ __proto__: Object
    length: 5
  ▼ 0: Object
    x: 0
    y: 22
    y0: 10
    ▶ __proto__: Object
  ▼ 1: Object
    x: 1
    y: 28
    y0: 16
    ▶ __proto__: Object
  ▼ 2: Object
    x: 2
    y: 32
    y0: 21
    ▶ __proto__: Object
  ▼ 3: Object
    x: 3
    y: 35
    y0: 30
    ▶ __proto__: Object
  ▼ 4: Object
    x: 4
    y: 43
    y0: 40
    ▶ __proto__: Object
    length: 5
```

3. 스택 레이아웃 – rect 좌표 설정

```
31     { x: 1, y: 12 },
32     { x: 2, y: 19 },
33     { x: 3, y: 23 },
34     { x: 4, y: 17 }
35   ],
36   [
37     { x: 0, y: 22 },
38     { x: 1, y: 28 },
39     { x: 2, y: 32 },
40     { x: 3, y: 35 },
41     { x: 4, y: 43 }
42   ]
43 ];
44
45 //Set up stack method
46 var stack = d3.layout.stack();
47
48 //Data stacked
49 stack(dataset);
50
51 //Set up scales
52 var xScale = d3.scale.ordinal()
53   .domain(d3.range(dataset[0].length))
54   .rangeRoundBands([0, w], 0.05);
55
56 var yScale = d3.scale.linear()
57   .domain([0,
58     d3.max(dataset, function (d) {
59       return d3.max(d, function (d) {
60         return d.y0 + d.y;
61       });
62     })
63   ])
64   .range([0, h]);
65
66 //Easy colors accessible via a 10-step ordinal scale
67 var colors = d3.scale.category10();
68
69 //Create SVG element
70 var svg = d3.select("body")
71   .append("svg")
```

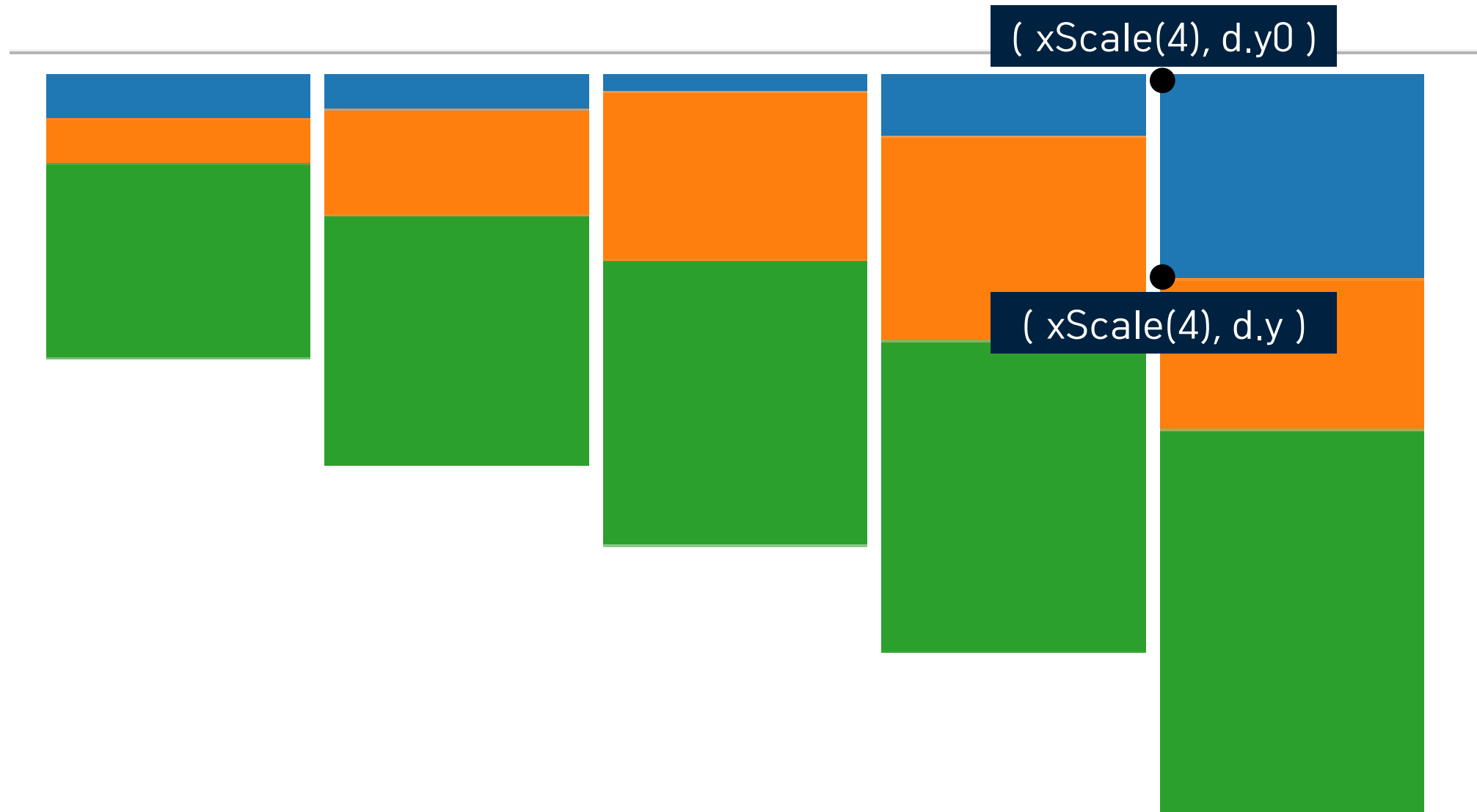
***yScale.domain 설정 부분이 중요!
➔ 나눠서 뜯어보기!

3. 스택 레이아웃 – rect 좌표 설정

```
65
66 //Easy colors accessible via a 10-step ordinal scale
67 var colors = d3.scale.category10();
68
69 //Create SVG element
70 var svg = d3.select("body")
71   .append("svg")
72   .attr("width", w)
73   .attr("height", h);
74
75 //Add a group for each row of data
76 var groups = svg.selectAll("g")
77   .data(dataset)
78   .enter()
79   .append("g")
80   .style("fill", function (d, i) {
81     return colors(i);
82   });
83
84 //Add a rect for each data value
85 var rects = groups.selectAll("rect")
86   .data(function (d) {return d; })
87   .enter()
88   .append("rect")
89   .attr("x", function (d, i) {
90     return xScale(i);
91   })
92   .attr("y", function (d) {
93     return yScale(d.y0);
94   })
95   .attr("height", function (d) {
96     return yScale(d.y);
97   })
98   .attr("width", xScale.rangeBand());
99
100 </script>
101 </body>
102 </html>
```

*** y 좌표 → d.y0 / height → d.y

3. 스택 레이아웃 – rect 좌표 설정



4. 포스 레이아웃 – dataset 형태

```
//Original data
var dataset = {
  nodes: [
    { name: "Junhwan" },
    { name: "Minsu" },
    { name: "Yong-hwan" },
    { name: "Seong_yun" },
    { name: "Donovan" },
    { name: "Iris" },
    { name: "Julia" },
    { name: "Jerry" },
    { name: "Adam" },
    { name: "Levine" }
  ],
  edges: [
    { source: 0, target: 1 },
    { source: 0, target: 2 },
    { source: 0, target: 3 },
    { source: 1, target: 4 },
    { source: 2, target: 5 },
    { source: 2, target: 6 },
    { source: 3, target: 5 },
    { source: 5, target: 4 },
    { source: 6, target: 3 },
    { source: 5, target: 7 },
    { source: 8, target: 8 },
    { source: 7, target: 9 },
    { source: 8, target: 3 }
  ]
};
```

```
> dataset
< ▼ Object {nodes: Array[10], edges: Array[13]} ⓘ
  ► edges: Array[13]
  ▼ nodes: Array[10]
    ▼ 0: Object
      index: 0
      name: "Junhwan"
      px: 218.5948979404358
      py: 125.02660123249143
      weight: 3
      x: 218.55988596782947
      y: 125.00077522364005
      ► __proto__: Object
    ► 1: Object
```

```
> dataset
< ▼ Object {nodes: Array[10], edges: Array[13]} ⓘ
  ▼ edges: Array[13]
    ▼ 0: Object
      ▼ source: Object
        index: 0
        name: "Junhwan"
        px: 218.5948979404358
        py: 125.02660123249143
        weight: 3
        x: 218.55988596782947
        y: 125.00077522364005
        ► __proto__: Object
      ▼ target: Object
        index: 1
        name: "Minsu"
        px: 232.77704515864892
        py: 76.14710085917552
        weight: 2
        x: 232.76866565184008
        y: 76.10382831666301
        ► __proto__: Object
      ► __proto__: Object
```

4. 포스 레이아웃 – d3.layout.force()

```
49  
50  
51 //Initialize a default force layout, using the nodes and edges in dataset  
52 var force = d3.layout.force()  
53   .nodes(dataset.nodes)  
54   .links(dataset.edges)  
55   .size([w, h])  
56   .linkDistance([50])  
57   .charge([-100])  
58   .start();
```

```
59  
60 var colors = d3.scale.category10();
```

```
61  
62 //Create SVG element  
63 var svg = d3.select("body")  
64   .append("svg")  
65   .attr("width", w)  
66   .attr("height", h);
```

```
67  
68 //Create edges as lines  
69 var edges = svg.selectAll("line")  
70   .data(dataset.edges)  
71   .enter()  
72   .append("line")  
73   .style("stroke", "#ccc")  
74   .style("stroke-width", 1);
```

```
75  
76 //Create nodes as circles  
77 var nodes = svg.selectAll("circle")  
78   .data(dataset.nodes)  
79   .enter()  
80   .append("circle")  
81   .attr("r", 10)  
82   .style("fill", function(d, i) {  
83     return colors(i);  
84   })  
85   .call(force.drag);
```

```
86  
87 //Every time the simulation "ticks", this will be called  
88 force.on("tick", function() {  
89  
90   edges.attr("x1", function(d) { return d.source.x; })  
91     .attr("y1", function(d) { return d.source.y; })  
92     .attr("x2", function(d) { return d.target.x; })  
93     .attr("y2", function(d) { return d.target.y; });  
94  
95   nodes.attr("cx", function(d) { return d.x; })  
96     .attr("cy", function(d) { return d.y; });  
97  
98 });  
99
```

다른 레이아웃과 다르게,
마지막에 start() 메서드를 호출.

➔ 계속해서 dataset을 불러서 시각적으로 보여줌을 의미.
➔ 이후, tick 설정을 왜 하는지에 대한 해답.

4. 포스 레이아웃 – nodes / edges

```
49  
50  
51 //Initialize a default force layout, using the nodes and edges in dataset  
52 var force = d3.layout.force()  
53   .nodes(dataset.nodes)  
54   .links(dataset.edges)  
55   .size([w, h])  
56   .linkDistance([50])  
57   .charge([-100])  
58   .start();  
59  
60 var colors = d3.scale.category10();  
61  
62 //Create SVG element  
63 var svg = d3.select("body")  
64   .append("svg")  
65   .attr("width", w)  
66   .attr("height", h);  
67  
68 //Create edges as lines  
69 var edges = svg.selectAll("line")  
70   .data(dataset.edges)  
71   .enter()  
72   .append("line")  
73   .style("stroke", "#ccc")  
74   .style("stroke-width", 1);  
75  
76 //Create nodes as circles  
77 var nodes = svg.selectAll("circle")  
78   .data(dataset.nodes)  
79   .enter()  
80   .append("circle")  
81   .attr("r", 10)  
82   .style("fill", function (d, i) {  
83     return colors(i);  
84   })  
85   .call(force.drag);  
86  
87 //Every time the simulation "ticks", this will  
88 force.on("tick", function () {  
89  
90   edges.attr("x1", function (d) { return d.  
91     .attr("y1", function (d) { return  
92     .attr("x2", function (d) { return  
93     .attr("y2", function (d) { return d.target.y; });  
94  
95   nodes.attr("cx", function (d) { return d.x; })  
96     .attr("cy", function (d) { return d.y; });  
97  
98  
99
```

.edges → svg의 *** line ***

.nodes → svg의 *** circle ***

.call(force.drag) → 상호작용을 위해
드래그 앤 드롭 호출

4. 포스 레이아웃 – tick 설정하기

```
49  
50  
51 //Initialize a default force layout, using the nodes and edges in dataset  
52 var force = d3.layout.force()  
53   .nodes(dataset.nodes)  
54   .links(dataset.edges)  
55   .size([w, h])  
56   .linkDistance([50])  
57   .charge([-100])  
58   .start();  
59  
60 var colors = d3.scale.category10();  
61  
62 //Create SVG element  
63 var svg = d3.select("body")  
64   .append("svg")  
65   .attr("width", w)  
66   .attr("height", h);  
67  
68 //Create edges as lines  
69 var edges = svg.selectAll("line")  
70   .data(dataset.edges)  
71   .enter()  
72   .append("line")  
73   .style("stroke", "#ccc")  
74   .style("stroke-width", 1);  
75  
76 //Create nodes as circles  
77 var nodes = svg.selectAll("circle")  
78   .data(dataset.nodes)  
79   .enter()  
80   .append("circle")  
81   .attr("r", 10)  
82   .style("fill", function(d, i) {  
83     return colors(i);  
84   })  
85   .call(force.drag);  
86  
87 //Every time the simulation "ticks", this will be called  
88 force.on("tick", function() {  
89  
90   edges.attr("x1", function(d) { return d.source.x; })  
91     .attr("y1", function(d) { return d.source.y; })  
92     .attr("x2", function(d) { return d.target.x; })  
93     .attr("y2", function(d) { return d.target.y; });  
94  
95   nodes.attr("cx", function(d) { return d.x; })  
96     .attr("cy", function(d) { return d.y; });  
97  
98  
99 });
```

*** tick 코드는,
'매 순간(tick)마다 line 과 circle의 x, y 값을 갱신하고
반영해라'
➔ x, y 값은 자동으로 입력됨.

*** tick을 설정하는 이유는,
포스 레이아웃 특성상, dataset을 계속해서 불러와서
실행되기 때문