

6장. 데이터를 이용한 드로잉

[복습]

데이터 불러오기
(loading)

데이터 엮기
(binding)

시각적 요소 지정
(transforming)

사용자 반응 지정
(transitioning)

[복습]

데이터 불러오기 (loading)

- 데이터 불러오기
- `d3.csv('data.csv', function(data) {
 dataset = data;
/ d3.json()`

데이터 엮기 (binding)

- 데이터를 시각적 요소와 연동
- `d3.select('body').selectAll('p') // 요소 선택
.data(dataset) // 데이터 불러오기
.enter() // 데이터와 요소 엮기
.append('p') // 문서 요소 정의
.text('New text')`

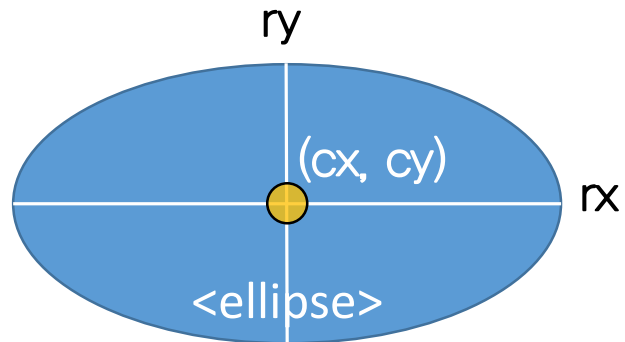
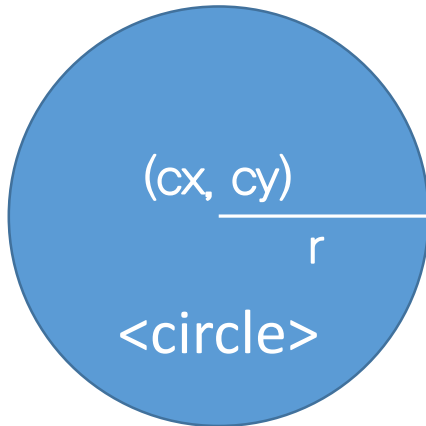
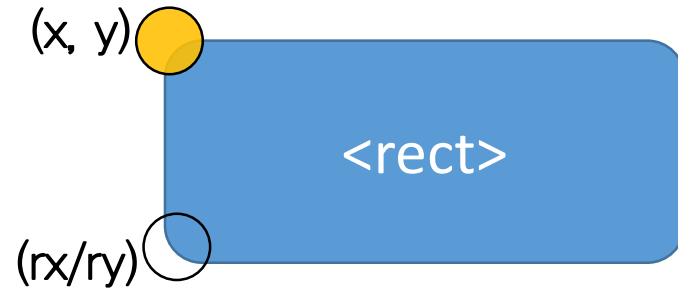
시각적 요소 지정 (transforming)

- 시각적 요소 지정(그래프 유형, 요소의 색상, 축 등)
- `d3.svg.axis()` (추후 6장부터 나옴)

사용자 반응 지정 (transitioning)

- 클릭, 드래그 사용자 반응에 따른 효과 지정

[복습]



[복습]

D3. 데이터 형식 (“ 데이터 경로 ” , 함수)

데이터를 불러온 후, 할 수 있는 작업들은
이 함수에서 진행
Ex) 데이터를 불러오고 dataset이라는
변수에 넣는 작업

[복습]

함수

```
function name() {  
  실행할 작업;  
};  
// name은 함수 이름
```

익명함수

```
function (d) {  
  return d;  
};  
// d는 전달인자
```

attr 메소드

```
selection.attr ( “name” , value )  
// name은 설정하고자 하는 속성명
```

6.1 div로 드로잉하기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Drawing divs with data</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">
```

```
div.bar {
  display: inline-block;
  width: 20px;
  height: 75px;
  background-color: teal;
}
```

css display 내장 스타일 추가

```
</style>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var dataset = [ 5, 10, 15, 20, 25 ];
```

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

```
  .data(dataset)
```

```
  .enter()
```

```
  .append("div")
```

```
  .attr("class", "bar");
```

bar 클래스 적용

직접 style 속성 지정을 한다면

.classed("bar", true) 로 하면 됨

```
</script>
```

```
</body>
```

```
</html>
```

6.1 div로 드로잉하기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Drawing divs with data</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = [ 5, 10, 15, 20, 25 ];

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          return d + "px";
        });

    </script>
  </body>
</html>
```

직접 CSS 프로퍼티와
값을 지정할 때 사용

6.1 div로 드로잉하기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Drawing divs, spaced out</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = [ 5, 10, 15, 20, 25 ];

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });

    </script>
  </body>
</html>
```

div bar 사이 간격 추가

d를 5배로 늘리는 경우 변수 사용

6.2 data() 메서드의 힘

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Drawing divs, spaced out</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = [ 5, 10, 15, 20, 25 ];

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });

    </script>
  </body>
</html>
```

data()에 들어가는 값의 개수만큼
데이터가 생성

Python에서 len()의 개수만큼 생성되는
점을 생각해보면 이해하기 쉬울 듯!

6.2 data() 메서드의 힘

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Bar chart with random values</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = []; //Initialize empty array
      for (var i = 0; i < 25; i++) { //Loop 25 times
        var newNumber = Math.random() * 30; //New random number (0-30)
        dataset.push(newNumber); //Add new number to array
      }

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });
    </script>
  </body>
</html>
```

6.2 data() 메서드의 힘

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Bar chart with random values</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = []; //Initialize empty array
      for (var i = 0; i < 25; i++) { //Loop 25 times
        var newNumber = Math.floor(Math.random() * 30); //New random integer (0-29)
        dataset.push(newNumber); //Add new number to array
      }

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });
    </script>
  </body>
</html>
```

6.2 data() 메서드의 힘

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Drawing divs, spaced out</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = [ 5, 10, 15, 20, 25 ];

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });

    </script>
  </body>
</html>
```

d를 5배로 늘리는 경우 변수 사용

6.3 SVG로 드로잉하기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Creating an empty SVG</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">
      /* No style rules here yet */
    </style>
  </head>
  <body>
    <script type="text/javascript">

      //Create SVG element
      var svg = d3.select("body")
        .append("svg")
        .attr("width", 500)
        .attr("height", 50);

    </script>
  </body>
</html>
```

문서 요소 생성

문서 요소 속성 지정

6.3 SVG로 드로잉하기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Creating an empty SVG</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">
      /* No style rules here yet */
    </style>
  </head>
  <body>
    <script type="text/javascript">
```

```
//Width and height
var w = 500;
var h = 50;
```

```
//Create SVG element
var svg = d3.select("body")
  .append("svg")
  .attr("width", w)
  .attr("height", h);
```

변수에 값을 넣고, 변수를 부르게 코딩

6.3 SVG로 드로잉하기

```
//Create SVG element  
var svg = d3.select("body")  
    .append("svg")  
    .attr("width", 500)  
    .attr("height", 50);
```

```
var circles = svg.selectAll("circle")  
    .data(dataset)  
    .enter()  
    .append("circle");
```

circle 생성

```
circles.attr("cx", function(d, i) {  
    return (i * 50) + 25;  
})  
    .attr("cy", h/2)  
    .attr("r", function(d) {  
        return d;  
    });
```

circle 속성 지정

6.3 SVG로 드로잉하기

```
//Create SVG element
var svg = d3.select("body")
    .append("svg")
    .attr("width", 500)
    .attr("height", 50);

var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
    .attr("cy", h/2)
    .attr("r", function(d) {
        return d;
    });

    .attr("fill", "yellow")
    .attr("stroke", "orange")
    .attr("stroke-width", function(d) {
        return d/2;
    });
```

circle 속성 지정

6.4 막대 차트 만들기 – div

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: A bar chart with divs</title>
    <script type="text/javascript" src="d3/d3.js"></script>
    <style type="text/css">

      div.bar {
        display: inline-block;
        width: 20px;
        height: 75px; /* Gets overridden by D3-assigned height below */
        margin-right: 2px;
        background-color: teal;
      }

    </style>
  </head>
  <body>
    <script type="text/javascript">

      var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                      11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];

      d3.select("body").selectAll("div")
        .data(dataset)
        .enter()
        .append("div")
        .attr("class", "bar")
        .style("height", function(d) {
          var barHeight = d * 5;
          return barHeight + "px";
        });

    </script>
  </body>
</html>
```

6.4 막대 차트 만들기 – SVG

```
//Width and height
var w = 500;
var h = 100;

var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
```

```
//Create SVG element
```

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);
```

SVG 생성

```
svg.selectAll("rect")
    .data(dataset)
    .enter()
    .append("rect")
    .attr("x", 0)
    .attr("y", 0)
    .attr("width", 20)
    .attr("height", 100);
```

rect 생성

6.4 막대 차트 만들기 – SVG

```
//Width and height
var w = 500;
var h = 100;

var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
```

```
//Create SVG element
```

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);
```

SVG 생성

```
svg.selectAll("rect")
    .data(dataset)
    .enter()
    .append("rect")
    .attr("x", function(d, i) {
        return i * 21; //Bar width of 20 plus 1 for padding
    })
    .attr("y", 0)
    .attr("width", 20)
    .attr("height", 100);
```

겹치는 문제를 방지하기 위한 코딩

6.4 막대 차트 만들기 – SVG

```
//Width and height
var w = 500;
var h = 100;

var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
```

```
//Create SVG element
```

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);
```

SVG 생성

```
svg.selectAll("rect")
    .data(dataset)
    .enter()
    .append("rect")
    .attr("x", function(d, i) {
        return i * (w / dataset.length);
    })
    .attr("y", 0)
    .attr("width", 20)
    .attr("height", 100);
```

확장성을 고려한 코드

6.4 막대 차트 만들기 – SVG

```
<script type="text/javascript">
```

```
//Width and height
```

```
var w = 500;
```

```
var h = 100;
```

```
var barPadding = 1;
```

여백 변수 추가

```
var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,  
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
```

```
//Create SVG element
```

```
var svg = d3.select("body")  
    .append("svg")  
    .attr("width", w)  
    .attr("height", h);
```

```
svg.selectAll("rect")  
    .data(dataset)  
    .enter()  
    .append("rect")  
    .attr("x", function(d, i) {  
        return i * (w / dataset.length);  
    })
```

```
    .attr("y", function(d) {  
        return h - (d * 4);  
    })
```

Svg는 위부터 좌표설정이 되서 이렇게 코딩

```
    .attr("width", w / dataset.length - barPadding)  
    .attr("height", function(d) {  
        return d * 4;  
    })  
    );
```

```
</script>
```

6.4 막대 차트 만들기 – SVG

```
<script type="text/javascript">
```

```
//Width and height
```

```
var w = 500;
```

```
var h = 100;
```

```
var barPadding = 1;
```

```
var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,  
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
```

```
//Create SVG element
```

```
var svg = d3.select("body")
```

```
    .append("svg")
```

```
    .attr("width", w)
```

```
    .attr("height", h);
```

```
svg.selectAll("rect")
```

```
    .data(dataset)
```

```
    .enter()
```

```
    .append("rect")
```

```
    .attr("x", function(d, i) {
```

```
        return i * (w / dataset.length);
```

```
    })
```

```
    .attr("y", function(d) {
```

```
        return h - (d * 4);
```

```
    })
```

```
    .attr("width", w / dataset.length - barPadding)
```

```
    .attr("height", function(d) {
```

```
        return d * 4;
```

```
    })
```

```
    .attr("fill", "teal");
```

색상 추가

```
</script>
```

6.4 막대 차트 만들기 – SVG

```
<script type="text/javascript">

  //Width and height
  var w = 500;
  var h = 100;
  var barPadding = 1;

  var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                  11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];

  //Create SVG element
  var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);

  svg.selectAll("rect")
    .data(dataset)
    .enter()
    .append("rect")
    .attr("x", function(d, i) {
      return i * (w / dataset.length);
    })
    .attr("y", function(d) {
      return h - (d * 4);
    })
    .attr("width", w / dataset.length - barPadding)
    .attr("height", function(d) {
      return d * 4;
    })
    .attr("fill", function(d) {
      return "rgb(0, 0, " + (d * 10) + ")";
    });
</script>
```

RGB 형태로 색 지정 가능

6.4 막대 차트 만들기 – SVG

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function(d) {
    return d;
  })
  .attr("x", function(d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function(d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

라벨 부여하기 위해 위치 지정 및 속성 부여

</script>

6.4 막대 차트 만들기 – SVG

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function(d) {
    return d;
  })
  .attr("text-anchor", "middle")
  .attr("x", function(d, i) {
    return i * (w / dataset.length) + (w / dataset.length - barPadding) / 2;
  })
  .attr("y", function(d) {
    return h - (d * 4) + 14;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

중앙 정렬

1 픽셀 위로 올림 (15→14)

6.4 막대 차트 만들기 – SVG

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function(d) {
    return d;
  })
  .attr("text-anchor", "middle")
  .attr("x", function(d, i) {
    return i * (w / dataset.length) + (w / dataset.length - barPadding) / 2;
  })
  .attr("y", function(d) {
    return h - (d * 4) + 14;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

중앙 정렬

1 픽셀 위로 올림 (15→14)

7장 척도

척도 : 요소의 위치를 크기에 비례해 자동으로 계산해줌

d3.scale() : 매핑한 함수. Linear, ordinary, category 등

d3.domain() : 입력되는 데이터 값의 범위
d3.domain([100,500])

d3.range() : 출력되는 데이터 값의 범위
d3.range([10, 350])

d3.min() / d3.max() 를 사용해서 값을 불러옴
단, 인덱스 개념을 이용해야 함

7장 척도

척도 설정

```
var xScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[0]; })])
    .range([0, w]);

var yScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[1]; })])
    .range([0, h]);
```

```
//Create SVG element
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);

svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle")
    .attr("cx", function(d) {
        return xScale(d[0]);
    })
    .attr("cy", function(d) {
        return yScale(d[1]);
    })
    .attr("r", function(d) {
        return Math.sqrt(h - d[1]);
    });
```

```
svg.selectAll("text")
    .data(dataset)
    .enter()
    .append("text")
    .text(function(d) {
        return d[0] + "," + d[1];
    });
```

```
.attr("x", function(d) {
    return xScale(d[0]);
})
.attr("y", function(d) {
    return yScale(d[1]);
})
```

```
.attr("font-family", "sans-serif")
.attr("font-size", "11px")
.attr("fill", "red");
```

척도 적용

7장 척도

```
var xScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[0]; })])
    .range([0, w]);

var yScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[1]; })])
    .range([h, 0]);

//Create SVG element
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);

svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle")
    .attr("cx", function(d) {
        return xScale(d[0]);
    })
    .attr("cy", function(d) {
        return yScale(d[1]);
    })
    .attr("r", function(d) {
        return Math.sqrt(h - d[1]);
    });

svg.selectAll("text")
    .data(dataset)
    .enter()
    .append("text")
    .text(function(d) {
        return d[0] + "," + d[1];
    })
    .attr("x", function(d) {
        return xScale(d[0]);
    })
    .attr("y", function(d) {
        return yScale(d[1]);
    })
    .attr("font-family", "sans-serif")
    .attr("font-size", "11px")
    .attr("fill", "red");
```

Y값이 큰 점들이 위로
Y의 치역을 변경

7장 척도

```
var w = 500;
var h = 100;
var padding = 20;

var dataset = [
    [5, 20], [480, 90], [250, 50], [100, 33], [330, 95],
    [410, 12], [475, 44], [25, 67], [85, 21], [220, 88]
];

//Create scale functions
var xScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[0]; })])
    .range([padding, w - padding * 2]);

var yScale = d3.scale.linear()
    .domain([0, d3.max(dataset, function(d) { return d[1]; })])
    .range([h - padding, padding]);
```

Padding(여백) 설정

7장 척도

```
var rScale = d3.scale.linear()  
    .domain([0, d3.max(dataset, function(d) { return d[1]; })])  
    .range([2, 5]);
```

```
svg.selectAll("circle")  
    .data(dataset)  
    .enter()  
    .append("circle")  
    .attr("cx", function(d) {  
        return xScale(d[0]);  
    })  
    .attr("cy", function(d) {  
        return yScale(d[1]);  
    })  
    .attr("r", function(d) {  
        return rScale(d[1]);  
    });
```

rScale 설정

7장 척도

```
var rScale = d3.scale.linear()  
    .domain([0, d3.max(dataset, function(d) { return d[1]; })])  
    .range([2, 5]);
```

```
svg.selectAll("circle")  
    .data(dataset)  
    .enter()  
    .append("circle")  
    .attr("cx", function(d) {  
        return xScale(d[0]);  
    })  
    .attr("cy", function(d) {  
        return yScale(d[1]);  
    })  
    .attr("r", function(d) {  
        return rScale(d[1]);  
    });
```

rScale 설정