

# 8장 축

## 8장 축

축(axis) : 정의한 파라미터를 가진 함수. 함수를 호출하면 어떤 값을 반환하는 대신 선과 라벨, 구분자를 가진 시각적 요소 생성됨.

d3.svg.axis() : 축 함수 생성.

xAxis.orient("bottom") : 라벨 기본 위치 지정. x축은 bottom, top y축은 left, right

svg.append("g") : SVG 끝에 append()로 g 문서를 추가. 여기서 g 문서요소는

그룹(group)을 뜻함. 눈에 보이지도 않고 시각적 위치를 차지안함.

call() : 메서드 체인 앞단의 선택물을 호출해서 함수로 건네줌.

## 8장 축

```
//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]);

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

```
//Define X axis
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom");
```

축 설정

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

```
//Create circles
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]);
    });
```

```
//Create labels
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");
```

```
//Create X axis
svg.append("g")
    .call(xAxis);
```

축 생성

## 8장 축

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3: Axis moved to the bottom</title>
    <script type="text/javascript" src="../../d3/d3.v3.js"></script>
    <style type="text/css">
      .axis path,
      .axis line {
        fill: none;
        stroke: black;
        shape-rendering: crispEdges;
      }

      .axis text {
        font-family: sans-serif;
        font-size: 11px;
      }
    </style>
  </head>
  <body>
    <img alt="A line graph showing a linear increase from 0 to 100 over 1000 iterations. The x-axis is labeled 'Iteration' and the y-axis is labeled 'Value'. The line is black and starts at (0,0) and ends at (1000,100)." data-bbox="100 100 600 500"/>
  </body>
</html>
```

## CSS 선택자 설정(문서요소 스타일 설정)

```
//Create labels
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");
```

```
//Create X axis
svg.append("g")
    .attr("class", "axis")
    .attr("transform", "translate(0," + (h - padding) + ")")
    .call(xAxis);
```

## css class 할당

## 8장 축

```
//Define X axis  
var xAxis = d3.svg.axis()  
    .scale(xScale)  
    .orient("bottom")  
    .ticks(5);
```

d3는 ticks()의 전달 인자 값을 의견 정도로 받아드리고 요청한 값보다 조금 크거나 같더라도 간격을 사람이 읽기 쉽고 깔끔하게 눈금을 조정해 준다.

## 8장 축

### y축 설정

```
//Define Y axis
var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left")
    .ticks(5);
```

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

```
//Create circles
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]);
    });
```

```
//Create labels
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");
```

```
//Create X axis
svg.append("g")
    .attr("class", "axis")
    .attr("transform", "translate(0," + (h - padding) + ")")
    .call(xAxis);
```

```
//Create Y axis
svg.append("g")
    .attr("class", "axis")
    .attr("transform", "translate(" + padding + ",0)")
    .call(yAxis);
```

### y축 생성

## 8장 축

```
//Dynamic, random dataset
var dataset = []; //Initialize empty array
var numDataPoints = 50; //Number of dummy data points to create
var xRange = Math.random() * 1000; //Max range of new x values
var yRange = Math.random() * 1000; //Max range of new y values
for (var i = 0; i < numDataPoints; i++) { //Loop numDataPoints times
    var newNumber1 = Math.floor(Math.random() * xRange); //New random integer
    var newNumber2 = Math.floor(Math.random() * yRange); //New random integer
    dataset.push([newNumber1, newNumber2]); //Add new number to array
}
```

랜덤 데이터 생성.  
데이터 개수 설정  
x축 y축 데이터 범위 설정  
랜덤 데이터 생성 후 정수화  
데이터셋에 저장

## 8장 축

```
/*
//Create labels
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");
*/
```

라벨 제거 : 보기 좋게 하기 위함

```
//Create X axis
svg.append("g")
  .attr("class", "axis")
  .attr("transform", "translate(0," + (h - padding) + ")")
  .call(xAxis);

//Create Y axis
svg.append("g")
  .attr("class", "axis")
  .attr("transform", "translate(" + padding + ",0)")
  .call(yAxis);
```



## 8장 축

```
//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]);

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

`var formatAsPercentage = d3.format(".1%");` 라벨 포매팅 : 원하는 형태로 포매팅 설정

```
//Define X axis
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom")
    .ticks(5)
    .tickFormat(formatAsPercentage);

//Define Y axis
var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left")
    .ticks(5)
    .tickFormat(formatAsPercentage);
```

x축 y축에 포매팅 설정 적용