

인터랙티브 데이터 시각화

WEEK

02

데잇걸즈

한국정보화진흥원

SVG 활용

D3.js로 SVG 다루기

송한나 Hannah Sookyong Song

2017.09.11.MON

1

D3.js로 SVG 도형 생성

SVG 요소 혹은 rect 요소 생성
한 개의 rect 요소 생성
여러 개의 rect 요소 생성

1. svg 요소 혹은 rect 요소 생성

D3.js를 이용하면 페이지 안에 있는 HTML 요소나 SVG 요소를 마음대로 조작할 수 있다. 조작 방법은 HTML DOM이나 SVG 모두 마찬가지이다.

먼저 페이지 안에 있는 div 요소 안에 svg 요소 혹은 SVG 도형의 하나인 rect 요소를 생성해본다. 생성되었음을 알 수 있도록 svg 요소에는 스타일시트를 설정해둔다.

select() 메서드로 요소 지정

우선 svg 요소를 생성할 대상이 되는 div 요소를 지정해야 한다. D3.js에서는 select() 메서드를 사용하여 요소를 지정한다. 다음과 같이 하면 ID이름 'myGraph'의 div 요소를 지정할 수 있다.

```
D3.select("#myGraph")
```

Select() 메서드의 파라미터에 CSS와 마찬가지로의 셀렉터를 지정한다. Select() 메서드는 셀렉터와 일치하는 첫 번째 요소만을 그 대상으로 한다. 같은 CSS 클래스를 가진 여러 개의 요소가 있다고 해도 가장 먼저 발견된 요소 하나만을 조작할 수 있다.

* D3.js에서 지정할 수 있는 것은 자바스크립트의 네이티브 메서드인 querySelector() 또는 querySelectorAll()로 지정할 수 있는 것에 해당한다.

1. svg 요소 혹은 rect 요소 생성

append() 메서드로 요소 추가

select() 메서드로 div 요소를 특정했으면 append() 메서드를 사용하여 svg 요소를 생성한다. append() 메서드는 파라미터에 지정한 요소를 추가한다. 이때 지정된 요소의 안쪽에 추가된다. 이후 사각형을 생성할 때 요소의 안쪽에 생성되는 것에 유의해야한다.

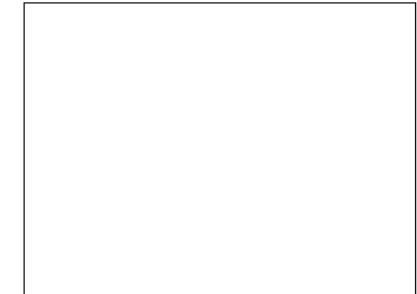
D3.select().append() 를 사용하여 svg 요소를 다음과 같이 생성한다.

1. svg 요소 혹은 rect 요소 생성

D3.js를 이용한 SVG 요소 생성

Sample01/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>SVG</title>
    <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <style>
      svg { width: 320px; height: 240px; border: 1px solid black; }
      rect { fill : orange; }
    </style>
  </head>
  <body>
    <h1>D3.js를 이용한 SVG 요소 생성</h1>
    <div id="myGraph"></div>
    <script src="js/sample.js"></script>
  </body>
</html>
```



svg 요소와 rect 요소를 생성한다.
rect 요소가 생성되었으나 넓이 등의 속성을 지정하지
않아 화면에 표시되지는 않는다.

svg 요소를 생성할 대상이 되는 Id="myGraph"인 div 요소를 생성한다.
D3.js 코드가 있는 sample.js 파일을 호출한다.

Sample01/js/sample.js

```
d3.select("#myGraph") // ID 이름 myGraph를 지정
  .append("svg") // svg 요소 추가
```

Select() 메서드로 id="myGraph"인 div를 특정한다.
append() 메서드로 지정한 svg 요소를 추가한다.
이후 이 svg 안쪽으로 사각형이 추가되게 된다.

2. 한 개의 rect 요소 생성

사각형 rect 요소를 추가한다. d3.select()로 div 요소를 지정하고 append()로 svg 요소를 추가하는 append("svg") 다음 줄에 append("rect")를 추가한다. 이것이 svg요소에 rect 요소를 추가하는 부분이다. append()를 사용하면 요소 안에 지정된 요소가 생성된다. 즉 이 경우에는 svg 요소 안에 rect 요소가 생성된다.

또한 생성할 그래프에 따라 다음과 같이 변수에 조작할 대상이 되는 요소를 넣어 둘 때도 있다. 이렇게 하면 나중에 특정 요소를 자유로이 조작할 수 있게 된다.

```
var s=d3.select("#myGraph")           // ID 이름 myGraph를 지정
```

* D3.js에서는 메서드 체인에 따라 연속된 처리를 기술할 수 있다. 메서드는 실행된 후 무언가의 반환 값이 있다. d3.select() 메서드의 반환값은 지정한 요소이다. append() 메서드의 반환값은 바로 생성한 요소가 된다. 위의 코드를 다음과 같이 바꾸어도 마찬가지로 rect 요소를 생성할 수 있다.

```
var s=d3.select("#myGraph")           // ID 이름 myGraph를 지정
    .append("svg")                     // svg 요소 추가
s.append("rect")                       // rect 요소 추가
```

2. 한 개의 rect 요소 생성

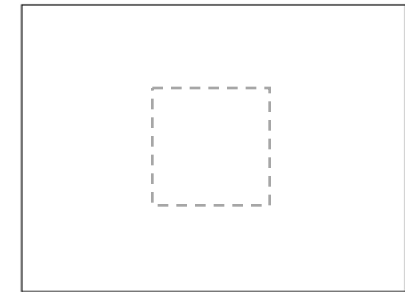
D3.js를 이용한 rect 요소 생성

Sample02/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>SVG</title>
    <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <style>
      svg { width: 320px; height: 240px; border: 1px solid black; }
      rect { fill : orange; }
    </style>
  </head>
  <body>
    <h1>D3.js를 이용한 rect 요소 생성</h1>
    <div id="myGraph"></div>
    <script src="js/sample.js"></script>
  </body>
</html>
```

Sample02/js/sample.js

```
d3.select("#myGraph") // ID 이름 myGraph를 지정
  .append("svg") // svg 요소 추가
  .append("rect") // rect 요소 추가
```



svg 요소와 rect 요소를 생성한다.
rect 요소가 생성되었으나 넓이 등의 속성을 지정하지 않아 화면에 표시되지는 않는다.

svg 요소를 생성할 대상이 되는 Id="myGraph"인 div 요소를 생성한다.
D3.js 코드가 있는 sample.js 파일을 호출한다.

Select() 메서드로 id="myGraph"인 div를 특정한다.
append() 메서드로 지정한 svg 요소를 추가한다.
svg 안쪽으로 사각형이 추가되었다(표시되지는 않는다).

3. 여러 개의 rect 요소 생성

또다른 svg 요소 안에 rect 요소를 생성해본다. 여러 개의 rect 요소를 생성하려면 다음과 같이 코드를 작성한다. append()를 사용하면 마음대로 요소를 추가할 수 있다. 그러나 생성한다고 해서 사각형으로 표시되지는 않으며, 요소의 표시에 필요한 속성을 지정해야한다.

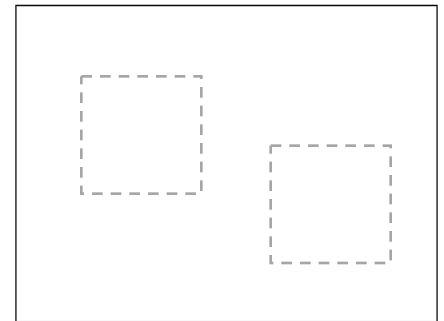
Sample03/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>SVG</title>
    <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <style>
      svg { width: 320px; height: 240px; border: 1px solid black; }
      rect { fill : orange; }
    </style>
  </head>
  <body>
    <h1>여러 개의 rect 요소 생성</h1>
    <div id="myGraph"></div>
    <script src="js/sample.js"></script>
  </body>
</html>
```

Sample03/js/sample.js

```
var svg1 = d3.select("#myGraph") // ID 이름 myGraph를 지정
    .append("svg") // svg 요소 추가
svg1.append("rect") // rect 요소를 svg 요소에 추가
svg1.append("rect") // rect 요소를 svg 요소에 추가
```

여러 개의 rect 요소 생성



변수 svg1을 선택하여 svg를 생성하고,
여기에 rect 2개를 연달아 추가한다.

3. 여러 개의 rect 요소 생성

여러 개의 rect 요소를 생성할 때 흔히 하는 실수에 주의한다. 다음 코드에서도 여러 개의 rect 요소를 생성할 것 같으나, 실제로는 맨 처음의 rect 요소 안에 다음 rect 요소가 생성되어 버린다.

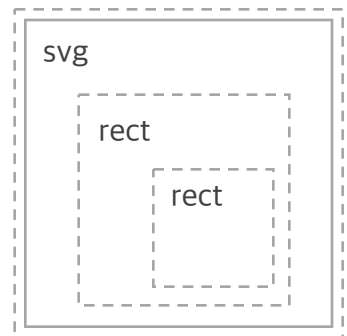
Sample03-2/js/sample.js

```
d3.select("#myGraph") // ID 이름 myGraph를 지정
.append("svg") // svg 요소 추가
.append("rect") // rect 요소를 svg 요소에 추가
.append("rect") // rect 요소를 svg 요소에 추가
```

rect 요소 안에 rect 요소가 추가된다.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body> == $0
    <h1>여러 개의 rect 요소 생성</h1>
    <div id="myGraph">
      <svg>
        <rect>
          <rect></rect>
        </rect>
      </svg>
    </div>
    <script src="js/sample.js"></script>
  </body>
</html>
```

<div Id="myGraph">



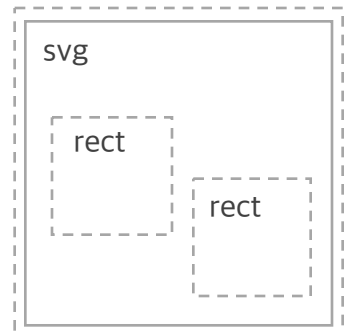
[비교]

Sample03/js/sample.js

```
var svg1 = d3.select("#myGraph") // ID 이름 myGraph를 지정
.append("svg") // svg 요소 추가
svg1.append("rect") // rect 요소를 svg 요소에 추가
svg1.append("rect") // rect 요소를 svg 요소에 추가
```

변수명 svg1

<div Id="myGraph">



2

도형의 위치나 크기 지정

도형의 위치나 크기 지정하기

D3.js로 rect 요소의 위치나 크기 지정하기

특정 도형 요소 다루기

스타일이나 속성을 한번에 지정하기

attr() 메서드의 다른 사용법 :

애니메이션 효과

1. 도형의 위치나 크기 지정하기

SVG에서는 요소를 생성하는 것만으로는 아무것도 표시되지 않을 때가 있다. 그러므로 도형의 위치나 크기, 칠하기 색 등을 지정해야 한다.

D3.js에서 도형의 위치나 크기를 지정하려면 `attr()` 메서드를 사용한다. 이 `attr()` 메서드만으로 모든 요소의 색과 크기를 지정할 수 있다. D3.js에서는 많은 메서드를 기억할 필요가 없는 대신, 조작할 svg 요소의 속성 이름 등은 알아야한다. `rect` 요소(사각형)에는 다음과 같은 속성이 있다.

요소	설명
x	X 좌표, 생략하면 0
y	Y 좌표, 생략하면 0
width	넓이
height	높이

`attr()` 메서드는 첫 번째 파라미터에 속성 이름을, 두 번째 파라미터에는 설정값을 지정한다. 속성 이름은 큰따옴표(“”)로 감싼 문자열로 지정한다.

```
attr("x", 50)
```

```
// x속성에 값 50을 설정(단위 없이 지정하면 px이 된다)
```

2. D3.js로 rect 요소의 위치나 크기 지정하기

Rect 요소를 좌표(10px, 50px)를 기준으로 넓이 200픽셀, 높이 30픽셀의 크기로 표시한다. 이와 함께 사각형은 오렌지 색이 되는데, 이것은 스타일시트로 색을 지정했기 때문이다.

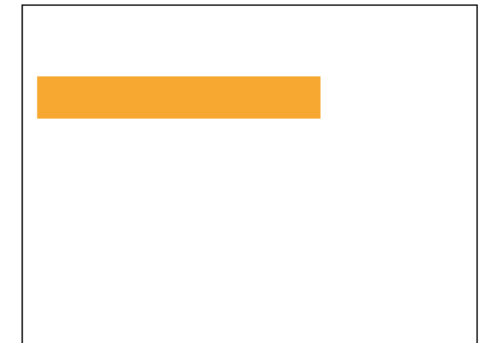
Sample04/index.html

```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  rect { fill : orange; }
</style>
```

Sample04/js/sample.js

```
d3.select("#myGraph") // ID 이름 myGraph를 지정
  .append("svg") // svg 요소 추가
  .append("rect") // rect 요소를 svg 요소에 추가
  .attr("x", "10px") // X 좌표를 10px로 설정
  .attr("y", "50px") // Y 좌표를 50px로 설정
  .attr("width", "200px") // 넓이를 200px로 설정
  .attr("height", "30px") // 높이를 30px로 설정
```

D3.js로 SVG 속성 다루기



(10,50) 위치에 200x30 픽셀 크기의 사각형이 표시된다.

3. 특정 도형 요소 다루기

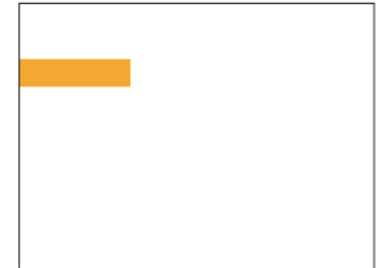
때에 따라서는 특정 rect 요소를 다룰 때가 있다. 이때는 rect 요소에 ID 이름을 지정해두고, d3.select()를 사용하여 ID 이름을 지정한다. 그 다음 attr() 메서드를 이용하여 속성값을 설정한다. 마찬가지로 클래스 이름을 지정해두면 여러 개의 요소를 다룰 수도 있다.

Sample05/index.html

```
<body>
  <h1>D3.js로 SVG 속성 다루기</h1>
  <svg>
    <rect id="myBar" x="0" y="50" width="100" height="25" />
  </svg>
  <script src="js/sample.js"></script>
</body>
```

id="myBar"인 rect 요소를 추가한다.

D3.js로 SVG 속성 다루기

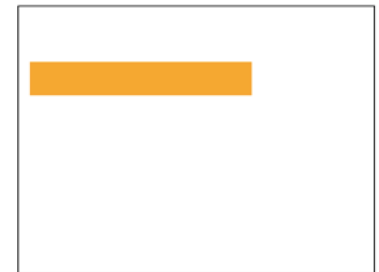


Sample05/js/sample.js

```
d3.select("#myBar") // ID 이름을 myBar로 지정
.attr("x", "10px") // X 좌표를 10px로 설정
.attr("y", "50px") // Y 좌표를 50px로 설정
.attr("width", "200px") // 넓이를 200px로 설정
.attr("height", "30px") // 높이를 30px로 설정
```

id="myBar"인 rect 요소의 속성이 (10,50) 위치에 200x30 픽셀 크기로 변경된다.

D3.js로 SVG 속성 다루기



4. 스타일이나 속성을 한번에 지정하기

`attr()` 메서드와 `style()` 메서드에서는 파라미터에 객체 리터럴 형식({속성 이름: 값, ... })을 지정하면 속성과 값을 한번에 설정할 수 있다. 이와 함께 속성 이름은 “x”와 같이 문자열로 지정할 수도 있다. 스타일시트 속성 이름에 줄표(-)가 포함되었을 때는 문자열로 지정한다.

Sample05/js/sample.js

```
d3.select("#myBar")      // ID 이름을 myBar로 지정
  .attr("x", "10px")      // X 좌표를 10px로 설정
  .attr("y", "50px")      // Y 좌표를 50px로 설정
  .attr("width", "200px") // 넓이를 200px로 설정
  .attr("height", "30px") // 높이를 30px로 설정
```

`attr()` 메서드로 속성을 하나씩 지정한다.

Sample05-2/js/sample.js

```
d3.select("#myBar")      // ID 이름에 myBar를 지정
  .attr({
    x : "10px",           // X 좌표를 10px로 설정
    y : "50px",           // Y 좌표를 50px로 설정
    width : "200px",      // 넓이를 200px로 설정
    height : "30px",      // 높이를 30px로 설정
  })
```

객체 리터럴 형식으로 여러 개의 속성을 한번에 지정한다.

5. attr() 메서드의 다른 사용법 : 애니메이션 효과

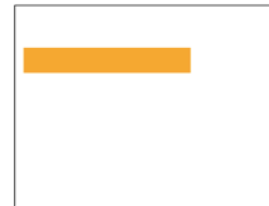
D3.js에서는 그래프를 역동적으로 변화시킬 때도 attr() 메서드가 중요한 역할을 한다. D3.js에서는 transition() 메서드를 메서드 체인에 포함하기만 해도 애니메이션 효과를 줄 수 있다. 이때 애니메이션 시작값과 종료값을 attr() 메서드로 지정한다. 메서드 체인에 transition()과 duration()을 추가하여 3초간 천천히 변화하는 애니메이션 효과를 준다.

Sample06/js/sample.js

```
d3.select("#myBar")
  .attr("x", "10px")
  .attr("y", "50px")
  .attr("width", "200px")
  .attr("height", "30px")
  .transition()
  .duration(3000)
  .attr("width", "50px")
```

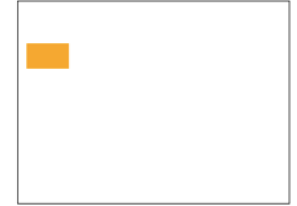
attr() 메서드로 애니메이션의 시작값과 종료값을 지정한다.

D3.js로 SVG 속성 다루기



3초

D3.js로 SVG 속성 다루기



```
// ID 이름을 myBar로 지정
// X 좌표를 10px로 설정
// Y 좌표를 50px로 설정
// 너비를 200px로 설정
// 높이를 30px로 설정
// 애니메이션 설정
// 3초간 변화
// 너비를 50px로 설정
```

Transition()으로 애니메이션 효과를 준다. Duration()으로 3초간 지속하도록 설정한다.

3

도형 스타일 지정

D3.js로 rect 요소의 칠하기나 선의 색을 지정하기
CSS 클래스 이름으로 요소를 지정할 때의 주의점
CSS에서 클래스로 스타일 설정하기
attr() 메서드로 요소의 스타일을 한꺼번에 지정하기

1. D3.js로 rect 요소의 칠하기나 선의 색을 지정하기

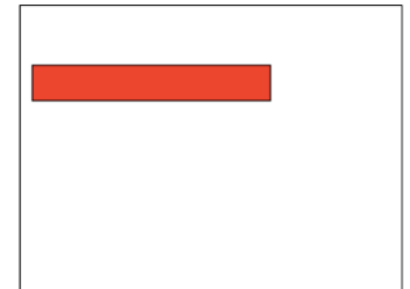
D3.js에서 도형의 속성은 attr() 메서드로 지정하는 반명 도형의 스타일은 style() 메서드를 사용한다. style() 메서드의 첫 번째 파라미터에는 스타일 속성 이름을 지정하고, 두 번째 파라미터에는 설정값을 지정한다. 예를 들어 stroke-width 속성에 10을 설정할 때는 style("stroke-width", 10)과 같이 지정한다. 단위를 붙여 설정할 때는 style("stroke-width", "10px")과 같이 숫자가 아닌 문자열로 지정한다.

Sample07/js/sample.js

```
d3.select("#myBar")      // ID 이름을 myBar로 지정
  .attr("x", "10px")      // X 좌표를 10px로 설정
  .attr("y", "50px")      // Y 좌표를 50px로 설정
  .attr("width", "200px") // 넓이를 200px로 설정
  .attr("height", "30px") // 높이를 30px로 설정
  .style("fill", "red")    // 스타일의 fill 속성을 red로 설정
  .style("stroke", "black") // 선의 색을 검정으로 설정
```

style() 메서드를 사용하여 칠하기 색을 빨간색으로, 사각형의 주위를 검은 선으로 표시한다.

D3.js로 스타일 다루기



2. CSS 클래스 이름으로 요소를 지정할 때의 주의점

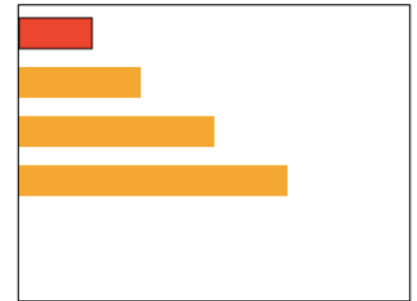
앞의 예제에서는 rect 요소에 ID 이름을 지정했지만, CSS 클래스 이름을 지정할 때는 다음과 같이 된다. 이와 함께 4개의 rect 요소에 같은 CSS 클래스 이름을 지정하고 있지만, 색이 변하는 것은 최초의 rect 요소 뿐이다. 이것은 d3.select()는 CSS 클래스 이름을 지정해도 최초의 요소에만 반영되기 때문이다. 해당하는 모든 CSS 클래스 이름의 요소를 대상으로 할 때는 d3.selectAll() 메서드를 사용한다.

Sample08/index.html

4개의 rect 요소에 "bar"클래스가 적용되었다.

```
<svg>
  <rect class="bar" x="0" y="10" width="60" height="25" />
  <rect class="bar" x="0" y="50" width="100" height="25" />
  <rect class="bar" x="0" y="90" width="160" height="25" />
  <rect class="bar" x="0" y="130" width="220" height="25" />
</svg>
```

D3.js로 스타일 다루기



Sample08/js/sample.js

D3.select() 메서드는 "bar"클래스가 적용된 첫번째 요소를 선택한다.

```
d3.select(".bar") // CSS 클래스 이름을 bar로 지정
  .style("fill", "red") // 스타일 fill 속성에 red를 지정
  .style("stroke", "black") // 선의 색을 검정으로 설정
```

d3.select()는 CSS 클래스의 최초 요소만 반영하므로, 첫번째 사각형에만 속성이 적용된다.
모든 CSS 클래스를 선택할 때는 d3.selectAll() 메서드를 사용한다.

3. CSS에서 클래스로 스타일 설정하기

웹에서는 코드에서 스타일을 직접 다루는 것은 바람직하지 않다고 한다. 이것은 코드 안에 디자인이 포함되면 디자인을 변경할 때 코드를 수정해야 하기 때문이다. 또한 분업이 안 되므로 프로그래머의 부담이 늘어난다는 단점도 있다. 더불어 그래프를 그릴 때 범용적으로 이용할 수 있도록 하려면 프로그램 코드에서 디자인 부분을 뛴 수 있으면 분리하는 편이 좋다.

D3.js에서는 미리 CSS 클래스 이름을 사용하여 표시하는 디자인(색이나 선 굵기 등)을 준비하고 attr() 메서드를 사용하여 CSS 클래스를 지정하도록 한다. 이렇게 해두면 색이나 선 굵기를 바꾸고 싶을 때 코드를 수정하지 않고도 CSS만 변경하면 되기 때문이다.

Sample09/index.html

```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  .bar { fill : orange; }
  .bar_note { fill : red; }
</style>
```

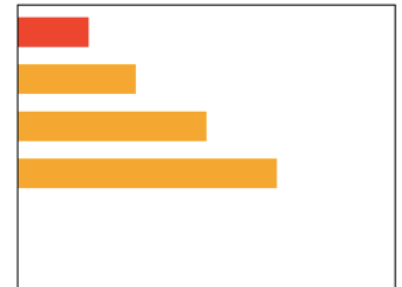
svg 요소에 bar_note 클래스를 지정한다.

Sample09/js/sample.js

```
d3.select(".bar") // CSS 클래스 이름에 bar를 지정
  .attr("class", "bar_note") // 요소의 class 속성에 bar_note를 지정
```

선택된 요소의 속성에 bar_note 클래스를 지정한다.

D3.js로 스타일 다루기



4. attr() 메서드로 요소의 스타일을 한꺼번에 지정하기

attr() 메서드를 사용하여 스타일을 한꺼번에 지정할 때는 style() 속성에 스타일을 설정하면 된다.

Sample10/index.html

```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  .bar { fill : orange; }
</style>
```

Sample10/js/sample.js

```
d3.select(".bar")          // CSS 클래스 이름에 bar를 지정
.attr("style", "fill:red;stroke:black") // 스타일 일괄 지정
```

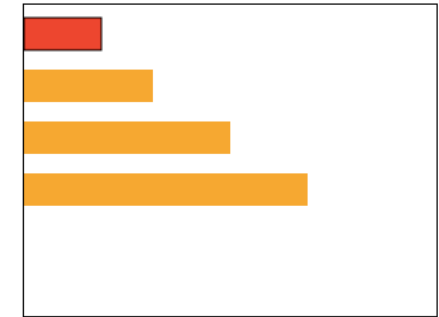
d3.select()로 선택된 첫번째 요소의 스타일을 빨간색으로 칠하고 검정색으로 선을 지정한다.

Sample08/js/sample.js

```
d3.select(".bar")          // CSS 클래스 이름을 bar로 지정
.style("fill", "red")       // 스타일 fill 속성에 red를 지정
.style("stroke", "black")   // 선의 색을 검정으로 설정
```

Style() 메서드를 사용하여 하나씩 지정할 수도 있다.

attr() 메서드로 한꺼번에 지정



4

여러 개의 도형 다루기

여러 개의 도형 다루기 : `selectAll()`

여러 개의 도형 스타일 지정하기 :
`style()`

특정 순서의 요소를 다루기 :
`function(d,i)`

1. 여러 개의 도형 다루기 : selectAll()

D3.js로 여러 개의 도형을 다루려면 selectAll() 메서드를 사용한다. selectAll() 메서드의 파라미터에는 요소 이름이나 CSS 클래스 이름을 지정한다. 해당하는 요소가 있다면 해당 요소들 모두를 반환값으로 돌려준다. 메서드 체인을 사용했을 때는 해당하는 요소 모두에 대해 처리가 이루어진다.

앞에서 특정 CSS 클래스 이름을 가진 rect 요소의 스타일을 변경할 때 맨 처음의 rect 요소만 변경된 것은 select() 메서드를 사용했기 때문이다. 이 때 selectAll() 메서드를 사용하면 해당하는 CSS 클래스 이름을 가진 모든 rect 요소의 스타일이 변경된다. selectAll() 메서드를 사용했을 때는 여러 개의 요소가 반환되지만, 스타일 등을 설정하려고 반복 처리를 기술하지 않아도 되므로 D3.js의 코드가 간단해진다.

Sample11/index.html

```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  .bar { fill : orange; }
  .bar_note { fill : red; }
</style>
```

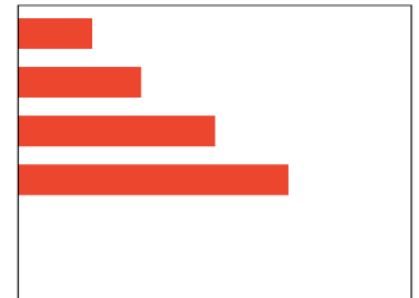
bar_note 클래스의 속성을 지정한다.

Sample11/js/sample.js

```
d3.selectAll(".bar")    // CSS 클래스 이름에 bar를 지정
    .attr("class", "bar_note")    // 요소의 class 속성에 bar_note를 지정
```

bar클래스의 요소를 모두 선택한다.
여기에 bar_note 클래스의 속성을 지정하여 모든 사각형이 빨간색으로 바뀐다.

요소를 한꺼번에 지정



2. 여러 개의 도형 스타일 지정하기 : style()

attr() 을 사용하여 CSS 클래스 이름을 바꿀 수도 있지만, style() 메서드를 사용하여 스타일을 지정할 수도 있다. 이때도 반복 처리를 수행하는 명령 등은 필요없으며 단순히 설정하고자 하는 스타일을 메서드 체인으로 연결하여 쓰면 된다.

Sample12/index.html

```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  .bar { fill : orange; }
</style>
```

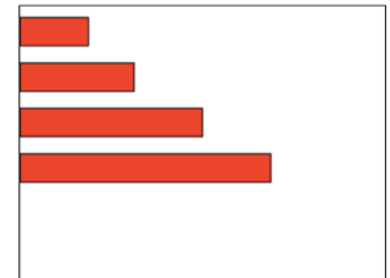
bar 클래스의 속성을 지정한다.

Sample12/js/sample.js

```
d3.selectAll(".bar") // CSS 클래스 이름에 bar를 지정
  .style("fill", "red") // 스타일 fill 속성에 red를 지정
  .style("stroke", "black") // 선의 색을 검은색으로 설정
```

bar클래스의 요소를 모두 선택한다.
style()을 메서드 체인으로 연결하여 모든 사각형이 빨간색이 되고 검은색 테두리를 그린다.

D3.js로 여러 개의 도형 다루기



3. 특정 순서의 요소를 다루기 : function(d,i)

select() 메서드로는 하나의 요소를, selectAll() 메서드로는 여러 개의 요소를 다룰 수 있다. 특정 위치(순서)에 있는 rect 요소를 다루려면, attr() 메서드나 style() 메서드의 두 번째 파라미터에 함수를 지정하면 반복 처리에 개입할 수 있다. 함수는 요소를 하나씩 처리할 때마다 호출된다.

이처럼 파라미터에 함수를 지정한다는 것이 D3.js의 가장 중요한 특징이며, 이 구조에 의해 D3.js에서 그래프 데이터에 따른 처리가 가능해지게 된다.

Sample13/index.html

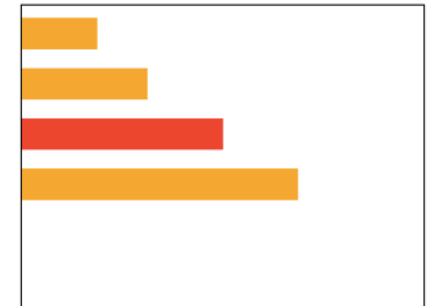
```
<style>
  svg { width: 320px; height: 240px; border: 1px solid black; }
  .bar { fill : orange; }
</style>
```

Sample13/js/sample.js

```
d3.selectAll(".bar")    // CSS 클래스 이름에 bar를 지정
  .style("fill", function(d,i){ // 2번째의 파라미터에 함수를 지정
    if(i == 2){          // 순서를 조사
      return "red";      // 3번째라면 빨간색을 나타내는 문자를 반환
    }
  })
```

Style() 메서드의 2번째 파라미터에 함수를 지정한다. 이 함수에는 2개의 파라미터를 전달하는데, 첫번째 파라미터에 데이터를, 두번째 파라미터에는 반복 순서(요소의 순서)를 전달한다. 반복 순서는 0을 시작으로 1번 반복할 때마다 1씩 증가한다.

특정 순서의 도형 다루기



Q&A

e-mail : dataexperiencedesign@gmail.com

web : www.cognitumlab.com