D3 예제 : 막대차트 그리기

```
<body>
   <svg width = "500" height = "500"></svg>
      <script>
         const dataFile = [10, 20, 30, 40, 50];    // 데이터배열 선언.
         const svg = d3.select("svg");             // DOM에서 <svg>요소를 찾는다.

         svg.selectAll('rect')      // <rect>요소를 모두 찾는다. 현재 DOM안에 <rect>는 하나도 없기 때문에 빈 값을 반환한다.
            .data(dataFile)         // dataFile배열의 개수를 계산하고, 이후 이어지는 메서드체인을 그 수만큼 반복한다.
            .enter()                // select 또는 selectAll에서 반환된 값이 있는지 확인하고, 있다면 각각에 데이터와 연결한다. 데이터가 연결될 값이 부족하다면, 빈 값을 생성해 데이터와 연결한다.
            .append('rect')         // enter에서 반환 된 값에 <rect>요소를 준다.
            .attr('fill', 'green')                  // 위에서 반환된 요소에 속성(fill)과 값(green)을 준다. 현재는 <rect fill="green" /> .
            .attr('height', (d) => {return d})      // 속성 뒤의 값이 들어간 자리에 있는 코드는 익명함수를 간소화해 표현한 코드이다. 괄호안의 'd'는 데이터배열의 값을 가진 임의의 변수이다.   'height'속성의 값에 d를 준다.
            .attr('width', 40)
            .attr("x", (d, i) => {return 50 * i})  // x속성은 svg태그 공간 안에서 왼쪽아래의 한 점을 가리키고, x속성의 값은 한 점으로부터의 거리(가로)를 나타난다. 괄호안의 'i'는 데이터배열의 인덱스 값을 가진 임의의 변수이다.
            .attr("y", (d, i) => {return 100 - dataFile[i]});
      </script>
</body>
```

```
<svg width="500" height="500">
  <rect fill="green" height="10" width="40" x="0" y="90"></rect>
  <rect fill="green" height="20" width="40" x="50" y="80"></rect>
  <rect fill="green" height="30" width="40" x="100" y="70"></rect>
  <rect fill="green" height="40" width="40" x="150" y="60"></rect>
  <rect fill="green" height="50" width="40" x="200" y="50"></rect>
</svg>
```



eleven, d3, 차트그리기, 막대차트

From:
http://125.132.25.164/dokuwiki/ - 문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:d3:%EB%A7%89%EB%8C%80%EC%B0%A8%ED%8A%B8_%EA%B7%B8%EB%A6%AC%EA%B8%B0&rev=1594859002

Last update: 2022/03/10 19:52