D3.js

- description : D3.js 관련 내용 기술
- author : eleven
- email : jskim@repia.com
- lastupdate : 2020-07-13

Chart libraries for javascript

1. D3
2. GOOGLE CHART
3. AMCHARTS
4. JSCharting
5. HIGHCHARTS

D3.js란?

> Data - Driven Documents의 약자로, Javascript 기반의 라이브러리 언어이다.
> 대표적인 데이터 시각화 프레임워크이다.
> 사용자가 입력한 시각화 규칙을 바탕으로 데이터를 반영해 결과물을 생성한다.

사용하기

- D3 공식사이트
- CDN 사용

```
<script src="https://d3js.org/d3.v5.min.js"></script>  //d3.js v.5
```

기본 메서드

- d3.select() : 특정 태그 하나를 선택한다.
- d3.selectAll() : 특정 태그 전체를 선택한다.
- .data() : 선택된 태그에 연결할 데이터를 가져온다.
- .enter() : 선택한 태그와 데이터를 연결시킨다. 데이터 개수에 비해 태그가 부족할 경우, 그 수만큼의 빈 태그를 반환한다.
- .append() : 새로운 태그를 추가한다.

예제

- 막대차트 그리기(기본메서드 이해하기)

Chart.js

Chart.js

> CDN

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
```

Chart.js 라이브러리

- 데이터라벨 표시하기

> Line Chart

```
    <div>
        <canvas id="myChart" width="700" height="400"></canvas>   // 차트가 그려질 canvas태그를 만든다.
    </div>

    <script>
        var config = new Chart(document.getElementById("myChart").getContext('2d'),{
            type: 'line',   // 차트의 종류를 선언한다.
            data: {
                labels: ['January', 'February', 'March', 'April', 'May', 'June', 'July'],
                datasets: [{                                // datasets 값의 배열을 추가하면 그림2와 같이 하나의 canvas에 두가지 데이터를 출력할 수 있다.
                    label: 'My First dataset'
                    , backgroundColor: 'rgba(255, 99, 132, 0.2)'
                    , borderColor: 'rgb(255, 99, 132)'
                    , data: [51, 33, 60, 25, 80, 40, 77]
                    , fill: true
                }]
            },
            options: {
                responsive: false,    // 차트의 반응형 여부를 정하는 옵션. false일 때, canvas태그에 사이즈 속성을 주면 해당 값으로 고정형 차트가 출력된다. default = true = 반응형.
                title: {
                    display: true,
                    text: 'Chart.js Line Chart'
```

```
                },
                scales: {
                    xAxes: [{
                        display: true,                  // x축 표시 여부.
                        scaleLabel: {                   // 축 라벨에 대한 옵션.
                            display: true,          // 라벨 표시 여부.
                            labelString: 'Month'    // 라벨 이름.
                        }
                    }],
                    yAxes: [{
                        display: true,
                        scaleLabel: {
                            display: true,
                            labelString: 'Value'
                        },
                        ticks: {
                            beginAtZero: false,    // true일 경우 y축 값이 0부터 시작한다.
                                                                stepSize: 10
// y축 값의 간격을 지정한다.
                        }
                    }]
                }
            }
        });
```

> Radar Chart

```
    <div style="width: 700px; height: 700px;">
        <canvas id="myChart"></canvas>
    </div>

    <script>
        var ctx = document.getElementById("myChart").getContext('2d');
        var radarOptions = {
            maintainAspectRatio: false     //  false일 경우 canvas가 포함된 div의 크기에 맞춰 차트가 생성된다. default = true.
            , title: {
                display: true,
                text: '== RADAR CHART =='
            }
            , scale: {
                angleLines: {
                    display: true  // 각 라벨로부터 중점으로 이어지는 라인 표시.
                }
            }
        }
```

```
        var myChart = new Chart(ctx, {
            type: 'radar',
            data: {
                labels: ["Red", "Blue", "Yellow", "Green", "Purple",
"Orange", "Navy"],
                datasets: [{
                    label: '# first'
                    , data: [51, 33, 60, 25, 80, 40, 77]
                    , fill: true
                    , backgroundColor: 'rgba(255, 99, 132, 0.2)'
                    , borderColor: 'rgb(255, 99, 132)'
                    , borderWidth: 2
                }
                ,{
                    label: '# second'
                    , data: [21, 10, 79, 82, 55, 46, 22]
                    , fill: true
                    , backgroundColor: 'rgba(54, 162, 235, 0.2)'
                    , borderColor: 'rgb(54, 162, 235)'
                    , borderWidth: 2
                }]
            },
            options: radarOptions
        });

    </script>
```

> JSON 데이터 사용하기

```
        <div style="width: 700px; height: 700px;">
        <canvas id="myChart"></canvas>
    </div>

    <script>
                // JSON 데이터를 선언한다.
        var jsonData = [{
                "siteId":"P001"
                ,"siteNm":"한국대학교"
                                ,"stdCount":340
            }
            ,{
                "siteId":"P002"
                ,"siteNm":"러시아대학교"
                                ,"stdCount":275
            }, ...
        ]

                // 사용할 value를 key값으로 가져온다.
```

```
        var labels = jsonData.map(function(e) {
            return e.siteNm;          // return "한국대학교", "러시아대학교"
        });

        var config = new
Chart(document.getElementById("myChart").getContext('2d'),{
            type: 'line',
            data: {
                labels: labels,    // 데이터가 담긴 변수를 라벨로 사용한다.
                datasets: [{
                    label: 'My First dataset'
                    , backgroundColor: 'rgba(255, 99, 132, 0.2)'
                    , borderColor: 'rgb(255, 99, 132)'
                    , data: [51, 33, 60, 25, 80, 40, 77]
                    , fill: true
                }]
            },
                        // 이하 생략
```

> Doughnut Chart

```
    <div>
        <canvas id="myChart" width="900px" height="500px"></canvas>
    </div>

    <script>
        var ctx = document.getElementById('myChart').getContext('2d');

        var jsonData = [{
            "departId":"001"
            ,"departNm":"기계공학과"
            ,"count":48
            }
            ,{
            "departId":"002"
            ,"departNm":"정보통신공학과"
            ,"count":6
            }
            ,...
                ]

        var data = jsonData.map(function(e) {
            return e.count;
        });
        var label = jsonData.map(function(e) {
            return e.departNm;
        });
```

```
        var config = {
            type: 'doughnut',
            data: {
                datasets: [{
                    data: data,
                    backgroundColor: ['#ff6384', '#ff9f40', '#ffcd56',
'#4bc0c0', '#36a2eb',]
                }],
                labels: label
            },
            options: {
                rotation: 1 * Math.PI,         // rotation 과 circumference 옵
션에 1 * Math.PI 값을 주면 반쪽 도넛차트가 된다. 생략 시 기본적인 도넛차트.
                circumference: 1 * Math.PI,
                cutoutPercentage: 60,          // 차트 가운데 빈 공간의 크기를 지정한
다. 0일 경우 파이차트와 같이 빈 공간이 없게된다.
                responsive: false,
                legend: {
                    position: 'top',       // 범례의 위치를 지정한다.
'top','left','bottom','right'.
                    labels: {
                        fontSize: 16, // 범례 폰트의 사이즈.
                        boxWidth: 50, // 범례 컬러박스의 사이즈.
                                                padding: 15    // 범례 사이의 간
격.
                    }
                },
                title: {
                    display: true,
                    text: '학과별 역량평가 참여 통계',
                    fontSize: 25           // 차트 제목 폰트의 사이즈.
                },
                animation: {
                    animateScale: true,    // true일 경우 차트 생성시 중앙에서부터
커지는 애니메이션.
                    animateRotate: true    // true일 경우 차트 생성시 회전 애니메이
션.
                }
            }
        };

        var doughnut = new Chart(ctx, config);
    </script>
```

> Bubble Chart

```
        <div>
```

```
        <canvas id="myChart"></canvas>
    </div>

    <script>

    var options = {
            maintainAspectRatio: false,
            type: 'bubble',
            options: {
                responsive: true,
                aspectRatio: 3,     // 캔버스의 가로:세로 비율을 나타내는 값. 1일 경우 정사각형의 차트가 그려진다. canvas태그 또는 css로 크키가 지정될 경우 무시된다.
                title: {
                    display: true,
                    text: 'BUBBLE CHART',
                    fontSize: 20
                },
            },
            data: {
                datasets: [                             // dataset 하나가 버블 하나를 나타낸다.
                    {
                        label: 'Tony',
                        data: [
                            {
                                x: 3,   // x축 위치.
                                y: 5,   // y축 위치.
                                r: 20   // 버블 크기.
                            }
                        ],
                        backgroundColor:"rgba(255, 99, 132, 0.5)",
                        hoverBackgroundColor: "rgba(255, 99, 132, 2)"   // 버블에 마우스오버 했을때의 컬러 지정.
                    },
                    {
                        label: 'Captain',
                        data: [
                            {
                                x: 3.7,
                                y: 4,
                                r: 90
                            }
                        ],
                        backgroundColor:"rgba(255, 19, 200, 0.5)",
                        hoverBackgroundColor: "rgba(255, 19, 200, 2)"
                    },
                    {
                        label: 'Thor',
                        data: [
                            {
                                x: 2.1,
```

```
                                y: 6,
                                r: 18
                            }
                        ],
                        backgroundColor:"rgba(255, 99, 132, 0.5)",
                        hoverBackgroundColor: "rgba(255, 99, 132, 2)"
                    }, ...
                ]
            }
    }

            var ctx = document.getElementById('myChart').getContext('2d');
            new Chart(ctx, options);

    </script>
```

주요 사이트

직무지식맵
울산R&D

Ref

D3.js 기초

eleven, 주레피, D3.js, d3, chart.js, 현재시간비교