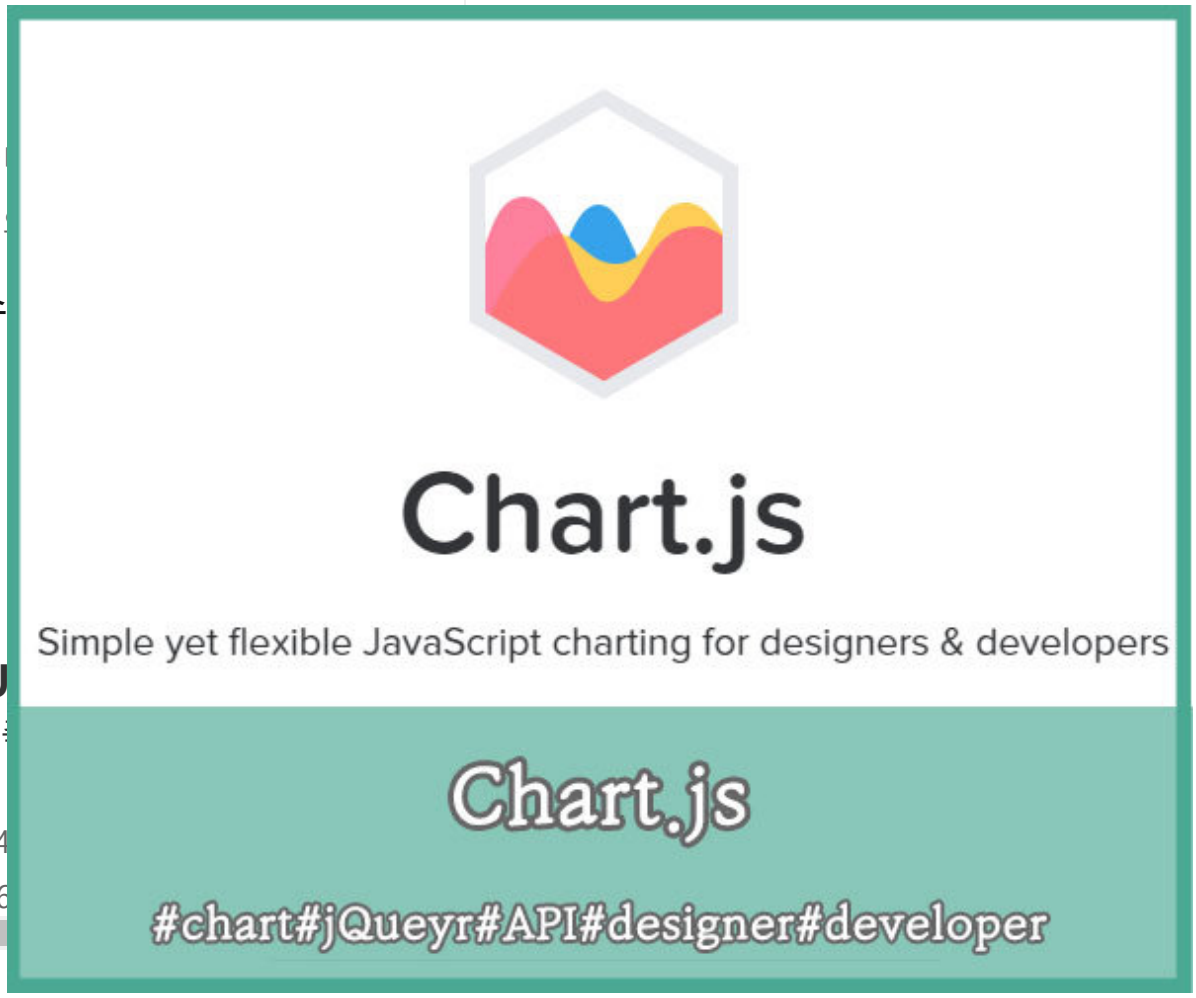
나홀로 쿠킹

COOKING (17)

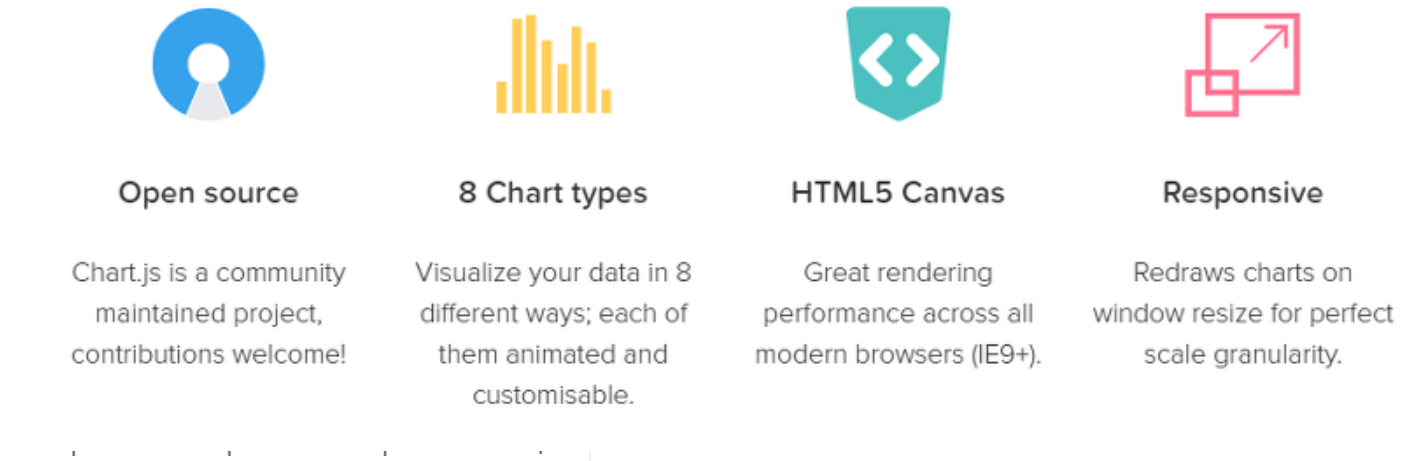
2016. 11. 25. 15:00



[API] Chart.js

차트생성기능을 만들기위해
여러 무료 api를 찾던중
탁월한 결과 기능이
젤 맘에들어서 사용해본
chart.js를 소개하도록 하겠다.

http://www.chartjs.org



오픈소스이고 8가지의 타입이 있고 html5의 canvas를 사용하고 반응형이다.

시작해보자.
일단 chart.js를 사용하려면 GitHub에서 다운받거나 CDN을 이용한다.
나는 걍 CDN을 이용한다.

Creating a Chart

To create a chart, we need to instantiate the `Chart` class. To do this, we need to pass in the node, jQuery instance, or 2d context of the canvas of where we want to draw the chart. Here's an example.

```
<canvas id="myChart" width="400" height="400"></canvas>
```

```
// Any of the following formats may be used
var ctx = document.getElementById("myChart");
var ctx = document.getElementById("myChart").getContext("2d");
var ctx = $("#myChart");
var ctx = "myChart";
```

IT (21)

차트를 만들자.

먼저 canvas를 만들고 너비와 높이를 정한뒤

REVIEW (41)

꼭 id를 넣자.

TRAVEL (0)

Once you have the element or context, you're ready to instantiate a pre-defined chart-type or create your own!

The following example instantiates a bar chart showing the number of votes for different colors and the y-axis starting at 0.

```
<canvas id="myChart" width="400" height="400"></canvas>
<script>
var ctx = document.getElementById("myChart");
var myChart = new Chart(ctx, { //차트 선언
    type: 'bar', //차트타
    data: { //차트 데이터 object배열
        labels: ["Red", "Blue", "Yellow", "Green", "Purple", "Orange"], //라벨 배열
        datasets: [{ //데이터 object배열
            label: '# of Votes',                //차트 라벨명
            data: [12, 19, 3, 5, 2, 3],         //데이터 배열
            backgroundColor: [
                'rgba(255, 99, 132, 0.2)',      /* 차트 배경색
                'rgba(54, 162, 235, 0.2)',      컬러 코드로 입력해도 되지만
                'rgba(255, 206, 86, 0.2)',      차트가 겹치게 될 경우 다른 데이터 값들이
                'rgba(75, 192, 192, 0.2)',      보이지 않기 때문에 RGBA로 투명도 조절을
                'rgba(153, 102, 255, 0.2)',     해주는게 좋다.*/
                'rgba(255, 159, 64, 0.2)'
            ],
            borderColor: [
                'rgba(255,99,132,1)',           /* 차트 테두리색
                'rgba(54, 162, 235, 1)',        테두리에 적용되는 색상이기 때문에 컬러코드로
                'rgba(255, 206, 86, 1)',        사용해도 무방하다.
                'rgba(75, 192, 192, 1)',        RGBA값에서 A값에 1을 줄바에는 차라리 코드값
                'rgba(153, 102, 255, 1)',       을 사용하는게 부담이 덜된다.
                'rgba(255, 159, 64, 1)'         */
            ],
            borderWidth: 1                      //차트 테두리 두께
        }]
    },
    options: {    //옵션 : 추가하고싶은 옵션이 있을때 사용
        scales: {
            yAxes: [{  //y축에 관련된 옵션 (x축일 경우 : xAxes)
                ticks: {
                    beginAtZero:true     //데이터 표기를 '0'부터 표기한다.
                }
            }]
        }
    }
});
</script>
```

기본 사용틀이다.

Global Font Settings

There are 4 special global settings that can change all of the fonts on the chart. These options are in `Chart.defaults.global`.

Name	Type	Default	Description
defaultFontColor	Color	'#666'	Default font color for all text
defaultFontFamily	String	"'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"	Default font family for all text
defaultFontSize	Number	12	Default font size (in px) for text. Does not apply to radialLinear scale point labels
defaultFontStyle	String	'normal'	Default font style. Does not apply to tooltip title or footer. Does not apply to chart title

REVIEW (41)

차본 폰트 셋팅이다.

TRAVEL (0)

Common Chart Configuration

The following options are applicable to all charts. The can be set on the global configuration, or they can be passed to the chart constructor.

Name	Type	Default	Description
responsive	Boolean	true	Resizes the chart canvas when its container does.
responsiveAnimationDuration	Number	0	Duration in milliseconds it takes to animate to new size after a resize event.
maintainAspectRatio	Boolean	true	Maintain the original canvas aspect ratio (`width / height`) when resizing
events	Array[String]	`["mousemove", "mouseout", "click", "touchstart", "touchmove", "touchend"]`	Events that the chart should listen to for tooltips and hovering
onClick	Function	null	Called if the event is of type 'mouseup' or 'click'. Called in the context of the chart and passed an array of active elements
legendCallback	Function	`function (chart) { }`	Function to generate a legend. Receives the chart object to generate a legend from. Default implementation returns an HTML string.
onResize	Function	null	Called when a resize occurs. Gets passed two arguments: the chart instance and the new size.

공통으로 사용되는 구성이다.

Title Configuration

The title configuration is passed into the `options.title` namespace. The global options for the chart title is defined in `Chart.defaults.global.title`.

Name	Type	Default	Description
display	Boolean	false	Display the title block
position	String	'top'	Position of the title. Possible values are 'top', 'left', 'bottom' and 'right'.
fullWidth	Boolean	true	Marks that this box should take the full width of the canvas (pushing down other boxes)
fontSize	Number	12	Font size inherited from global configuration
fontFamily	String	"'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"	Font family inherited from global configuration
fontColor	Color	"#666"	Font color inherited from global configuration
fontStyle	String	'bold'	Font styling of the title.
padding	Number	10	Number of pixels to add above and below the title text
text	String	''	Title text

Example Usage

The example below would enable a title of 'Custom Chart Title' on the chart that is created.

```
var chartInstance = new Chart(ctx, {
    type: 'line',
    data: data,
    options: {
        title: {
            display: true,
            text: 'Custom Chart Title'
        }
    }
})
```

차트 제목에 관련된 구성요소들인데
일단 정렬 옵션 안주면 안나오고
나오게 하고 싶으면 저 도표에 있는 옵션들로 구성하면 될거 같다.

Colors

When supplying colors to Chart options, you can use a number of formats. You can specify the color as a string in hexadecimal, RGB, or HSL notations. If a color is needed, but not specified, Chart.js will use the global default color. This color is stored at `Chart.defaults.global.defaultColor`. It is initially set to 'rgba(0, 0, 0, 0.1)';

You can also pass a CanvasGradient object. You will need to create this before passing to the chart, but using it you can achieve some interesting effects.

Patterns

An alternative option is to pass a CanvasPattern object. For example, if you wanted to fill a dataset with a pattern from an image you could do the following.

```
var img = new Image();
img.src = 'https://example.com/my_image.png';
img.onload = function() {
    var ctx = document.getElementById('canvas').getContext('2d');   /* canvas에 이미지를 로드시켜서
    var fillPattern = ctx.createPattern(img, 'repeat');               패턴화 한다. */

    var chart = new Chart(ctx, {
        data: {
            labels: ['Item 1', 'Item 2', 'Item 3'],
            datasets: [{
                data: [10, 20, 30],
                backgroundColor: fillPattern    // 데이터의 배경으로 사용할 수 있다.
            }]
        }
    })
}
```

Using pattern fills for data graphics can help viewers with vision deficiencies (e.g. color-blindness or partial sight) to more easily understand your data.

Using the Patternomaly library you can generate patterns to fill datasets.

```
var chartData = {
    datasets: [{
        data: [45, 25, 20, 10],
        backgroundColor: [
            pattern.draw('square', '#ff6384'),
            pattern.draw('circle', '#36a2eb'),       /* 배경에 모양과 색상을 정해
            pattern.draw('diamond', '#cc65fe'),         서 패턴을 넣을 수 있다. */
            pattern.draw('triangle', '#ffce56'),
        ]
    }],
    labels: ['Red', 'Blue', 'Purple', 'Yellow']
};
```

ABOUT ME …을 수 있는데

유튜브를 좋아하는 개발자 일상 블로그

Today 145   Yesterday 244

Total 116,936

Advanced usage

Prototype Methods

For each chart, there are a set of global prototype methods on the shared ChartType which you may find useful. These are available on all charts created with Chart.js, but for the examples, let's use a line chart we've made.

```
// For example:
var myLineChart = new Chart(ctx, config);  //전역변수
```

.destroy() //생성 된 모든 차트 인스턴스를 제거

Use this to destroy any chart instances that are created. This will clean up any references stored to the chart object within Chart.js, along with any associated event listeners attached by Chart.js. This must be called before the canvas is reused for a new chart.

```
// Destroys a specific chart instance
myLineChart.destroy();
```

/* 있던 캔버스에 새 차트를 만들려면 이 메서드를 호출 */

텍스트를 입력해주세요.

.update(duration, lazy)

Triggers an update of the chart. This can be safely called after replacing the entire data object. This will update all scales, legends, and then re-render the chart.

//말그대로 업데이트.

```
// duration is the time for the animation of the redraw in milliseconds
// lazy is a boolean. if true, the animation can be interrupted by other animations
myLineChart.data.datasets[0].data[2] = 50; // Would update the first dataset's value of 'March' to be 50
myLineChart.update(); // Calling update now animates the position of March from 90 to 50.
```

.reset() //차트 초기화

Reset the chart to it's state before the initial animation. A new animation can then be triggered using update.

```
myLineChart.reset();
```

디 디 디 니

차트타입 몇가지만 소개 하도록 하겠다.

ABOUT ME 에 대해서 타입만 설명 하도록 하겠음.

요리를 좋아하는 개발자 일상 블로그

Today 145 Yesterday 244

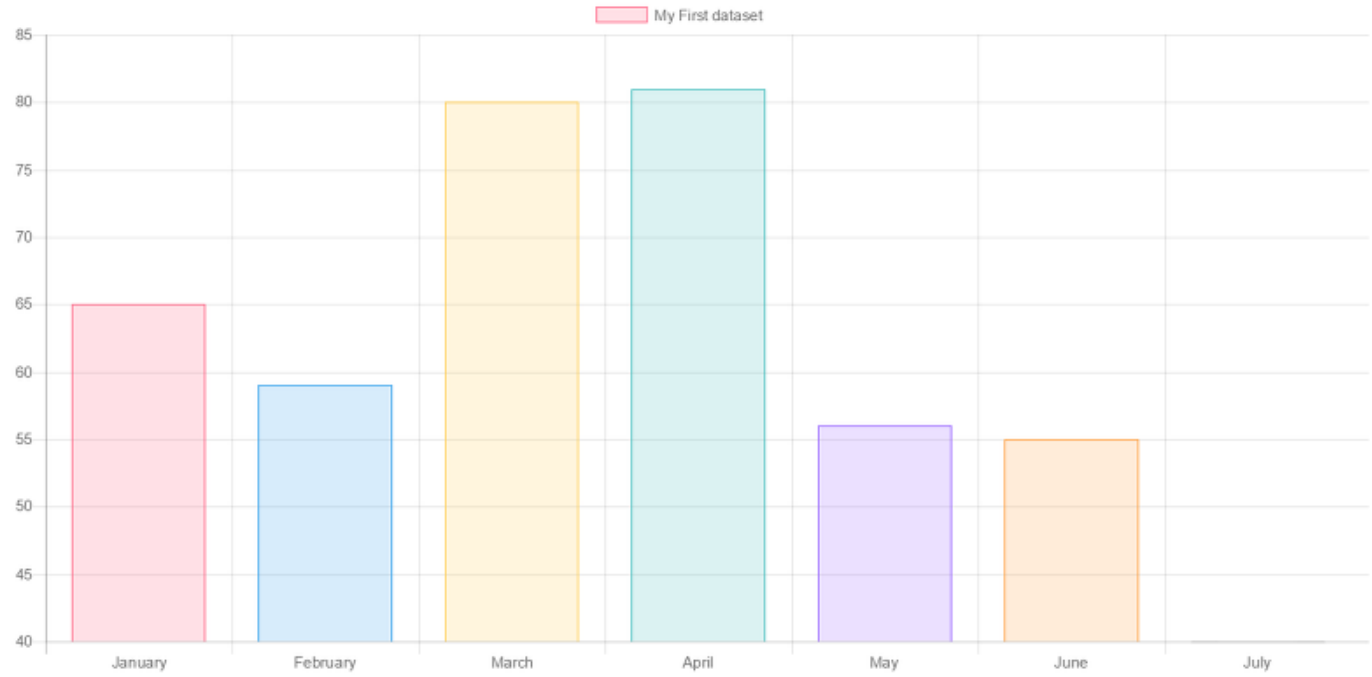
Total 116,936

Bar Chart

Introduction

A bar chart is a way of showing data as bars.

It is sometimes used to show trend data, and the comparison of multiple data sets side by side.



바 차트는 익숙한 형태
사용법도 쉬움..

[MySQL] 집탱구리 [퓨 집탱구리 [jQuery] 트리... 집탱구리 [JSON] Jso... 집탱구리

Example Usage

```
var myBarChart = new Chart(ctx, {
    type: 'bar',
    data: data,
    options: options
});
```

Or if you want horizontal bars.

```
var myBarChart = new Chart(ctx, {
    type: 'horizontalBar',
    data: data,
    options: options
});
```

기본틀에서 벗어나지 않음

An example data object using these attributes is shown below.

```
var data = {
    labels: ["January", "February", "March", "April", "May", "June", "July"],
    datasets: [
        {
            label: "My First dataset",
            backgroundColor: [
                'rgba(255, 99, 132, 0.2)',
                'rgba(54, 162, 235, 0.2)',
                'rgba(255, 206, 86, 0.2)',
                'rgba(75, 192, 192, 0.2)',
                'rgba(153, 102, 255, 0.2)',
                'rgba(255, 159, 64, 0.2)'
            ],
            borderColor: [
                'rgba(255,99,132,1)',
                'rgba(54, 162, 235, 1)',
                'rgba(255, 206, 86, 1)',
                'rgba(75, 192, 192, 1)',
                'rgba(153, 102, 255, 1)',
                'rgba(255, 159, 64, 1)'
            ],
            borderWidth: 1,
            data: [65, 59, 80, 81, 56, 55, 40],
        }
    ]
};
```

안에 들어가는 데이터도 크게 복잡하지 않음.

인기포스트

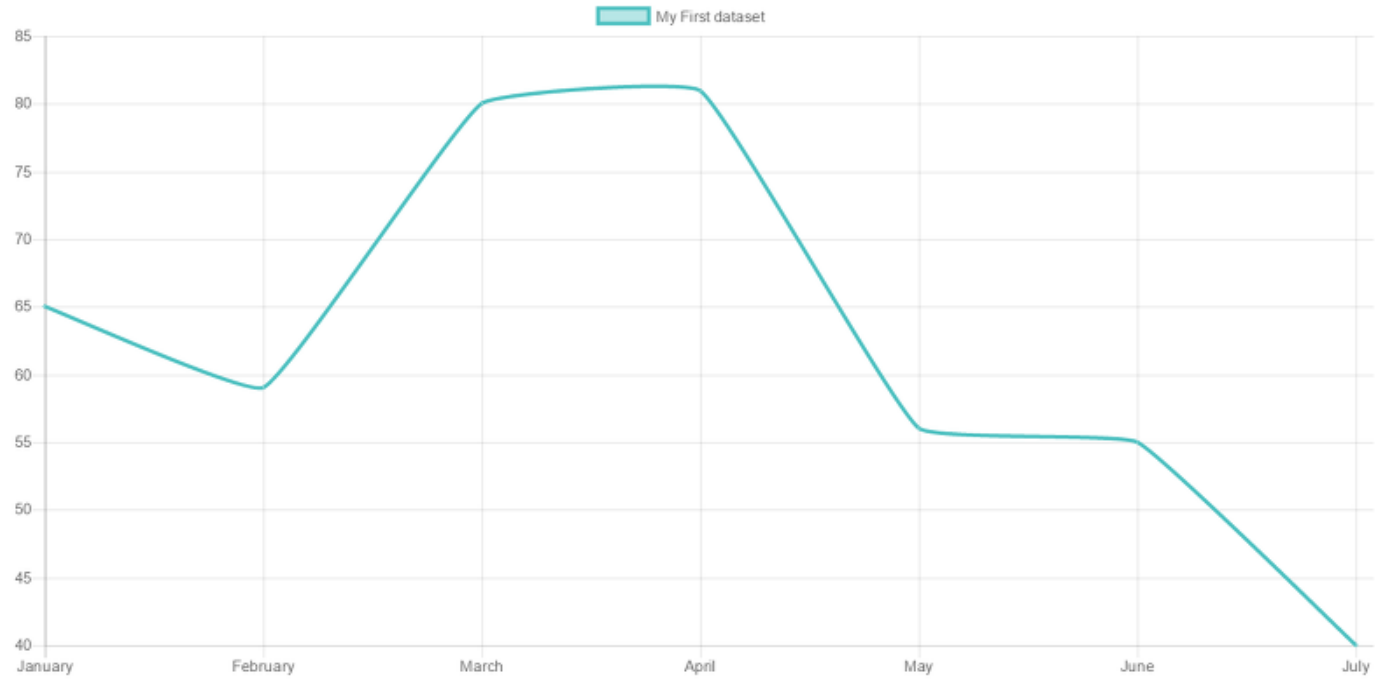
Line타입도 만만찮게 쉬움.

[MySQL] 문자... [퓨 ... [jQuery] 트리... [JSON] Jso...

Line Chart

Introduction

A line chart is a way of plotting data points on a line. Often, it is used to show trend data, and the comparison of two data sets.



bar차트와 다르지 않은 사용법이다.

Example Usage

```
var myLineChart = new Chart(ctx, {
    type: 'line',
    data: data,
    options: options
});
```

Alternatively a line chart can be created using syntax similar to the v1.0 syntax

```
var myLineChart = Chart.Line(ctx, {
    data: data,
    options: options
});
```

REVIEW (41)

라인 타입에는 각 데이터 마다 포인터가 있어서
거기에 대한 옵션도 추가 할 수 있다.

TRAVEL (0)

An example data object using these attributes is shown below.

```
var data = {
    labels: ["January", "February", "March", "April", "May", "June", "July"],
    datasets: [
        {
            label: "My First dataset",
            fill: false,
            lineTension: 0.1,
            backgroundColor: "rgba(75,192,192,0.4)",
            borderColor: "rgba(75,192,192,1)",
            borderCapStyle: 'butt',
            borderDash: [],
            borderDashOffset: 0.0,
            borderJoinStyle: 'miter',
            pointBorderColor: "rgba(75,192,192,1)",
            pointBackgroundColor: "#fff",
            pointBorderWidth: 1,
            pointHoverRadius: 5,
            pointHoverBackgroundColor: "rgba(75,192,192,1)",
            pointHoverBorderColor: "rgba(220,220,220,1)",
            pointHoverBorderWidth: 2,
            pointRadius: 1,
            pointHitRadius: 10,
            data: [65, 59, 80, 81, 56, 55, 40],
            spanGaps: false,
        }
    ]
};
```

파이 & 도넛 타입은 같은 사용법으로
타입만 다르게 써주면 된다.

Pie & Doughnut Charts

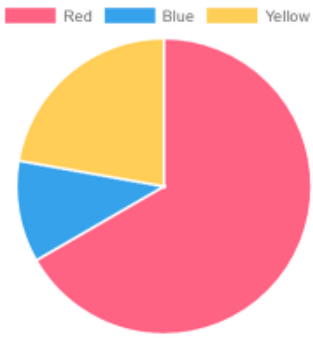
Introduction

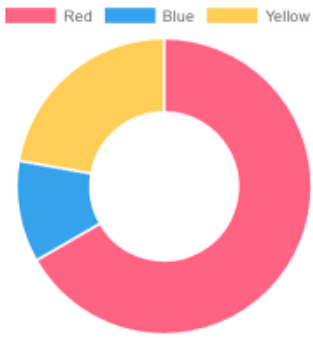
Pie and doughnut charts are probably the most commonly used charts there are. They are divided into segments, the arc of each segment shows the proportional value of each piece of data.

They are excellent at showing the relational proportions between data.

Pie and doughnut charts are effectively the same class in Chart.js, but have one different default value - their `cutoutPercentage`. This equates what percentage of the inner should be cut out. This defaults to 0 for pie charts, and 50 for doughnuts.

They are also registered under two aliases in the `Chart` core. Other than their different default value, and different alias, they are exactly the same.





Example Usage

```
// For a pie chart
var myPieChart = new Chart(ctx,{
    type: 'pie',
    data: data,
    options: options
});
```

```
// And for a doughnut chart
var myDoughnutChart = new Chart(ctx, {
    type: 'doughnut',
    data: data,
    options: options
});
```

An example data object using these attributes is shown below.

```
var data = {
    labels: [
        "Red",
        "Blue",
        "Yellow"
    ],
    datasets: [
        {
            data: [300, 50, 100],
            backgroundColor: [
                "#FF6384",
                "#36A2EB",
                "#FFCE56"
            ],
            hoverBackgroundColor: [
                "#FF6384",
                "#36A2EB",
                "#FFCE56"
            ]
        }]
};
```

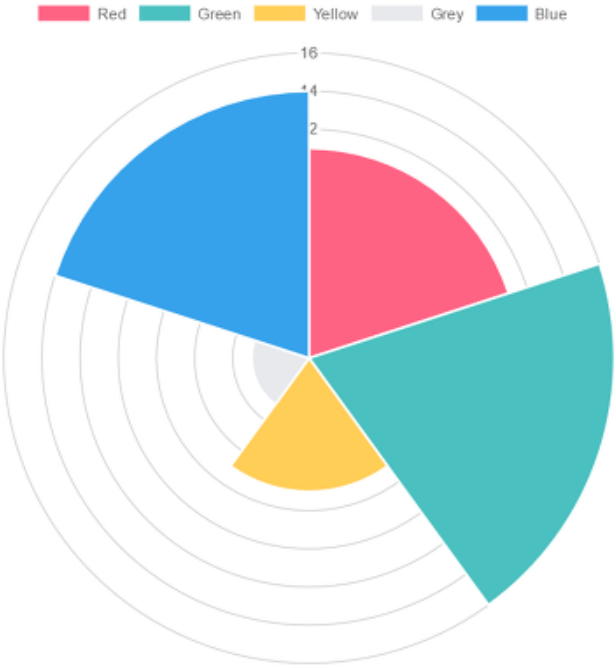
폴라타입도 비슷하다.

Polar Area Chart

Introduction

Polar area charts are similar to pie charts, but each segment has the same angle - the radius of the segment differs depending on the value.

This type of chart is often useful when we want to show a comparison data similar to a pie chart, but also show a scale of values for context.



Example Usage

```
new Chart(ctx, {
    data: data,
    type: 'polarArea',
    options: options
});
```

An example data object using these attributes is shown below.

```
var data = {
    datasets: [{
        data: [
            11,
            16,
            7,
            3,
            14
        ],
        backgroundColor: [
            "#FF6384",
            "#4BC0C0",
            "#FFCE56",
            "#E7E9ED",
            "#36A2EB"
        ],
        label: 'My dataset' // for legend
    }],
    labels: [
        "Red",
        "Green",
        "Yellow",
        "Grey",
        "Blue"
    ]
};
```

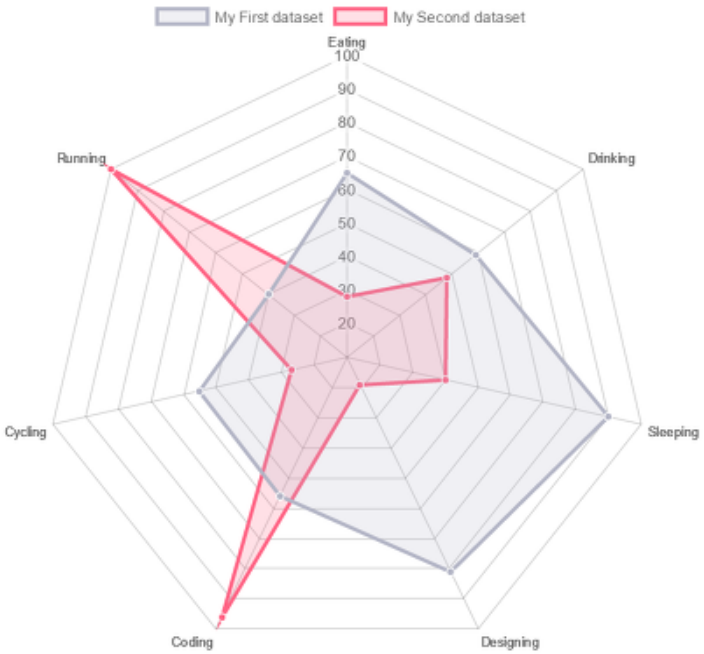
레이더 타입은 조금 다르게
각각의 라벨별로 데이터 배열이 있어야된다.

Radar Chart

Introduction

A radar chart is a way of showing multiple data points and the variation between them.

They are often useful for comparing the points of two or more different data sets.



Example Usage

```
var myRadarChart = new Chart(ctx, {
    type: 'radar',
    data: data,
    options: options
});
```

요티을 좋아하는 개발자 환영 블로그

An example data object using these attributes is shown below.

```
var data = {
    labels: ["Eating", "Drinking", "Sleeping", "Designing", "Coding", "Cycling", "Running"],
    datasets: [
        {
            label: "My First dataset",
            backgroundColor: "rgba(179,181,198,0.2)",
            borderColor: "rgba(179,181,198,1)",
            pointBackgroundColor: "rgba(179,181,198,1)",
            pointBorderColor: "#fff",
            pointHoverBackgroundColor: "#fff",
            pointHoverBorderColor: "rgba(179,181,198,1)",
            data: [65, 59, 90, 81, 56, 55, 40]
        },
        {
            label: "My Second dataset",
            backgroundColor: "rgba(255,99,132,0.2)",
            borderColor: "rgba(255,99,132,1)",
            pointBackgroundColor: "rgba(255,99,132,1)",
            pointBorderColor: "#fff",
            pointHoverBackgroundColor: "#fff",
            pointHoverBorderColor: "rgba(255,99,132,1)",
            data: [28, 48, 40, 19, 96, 27, 100]
        }
    ]
};
```

홈
생각보다 설명을 자세히 쓸수가 없구먼...

태그
데이터 한두개 정도로 만드는건 쉽지만
방명록일 때 좀 손이 많이간다.
그리고 프로그램을 짜다보면
같은 데이터로 다른 타입의 차트를 생성하고싶을 때가 있는데

IT (21)
Bar,Line 타입

Pie,Douhnut,Polar타입
REVIEW (41)
Radar타입

TRAVEL (0)
이렇게 3가지로 분류해서 데이터를 조물딱 거리면 된다.
국내 (0)
나는 배열을 가지고 놀기엔 하수라 고생좀 했지만....
해외 (0)

첨엔 이러걸 시키다니 욕나왔지만
연관포스트
하다보니 재밌기도 하고
소스 손봐서 다시 만들고 싶다.
[MySQL] 문자... 집탱구
[뷰 ... 집탱구
[jQuery] 트리... 집탱구
[JSON] Jso... 집탱구
소스넘 더럽.....ㅠㅠ

Total 116,936

TRAVEL (0)

구독하기

국내 (0)

해외 (0)

'IT' 카테고리의 다른 글

인기포스트

[유투브]유투브 썸네일 추출하기 (1) 2017.01.16

[구글맵API]구글맵 API사용하기 (0) 2016.12.19

[jQuery]파일용량 계산하기 (0) 2016.12.14

[PHP]배열 키값 재정의 (0) 2016.12.13

[API]차트생성 - Chart.js (0) 2016.11.25

[OVEN]프로토타이핑 툴 오븐!(beta) (0) 2016.11.25

[MySQL] 문자... 집탱구리

[퓨... 집탱구리

[jQuery] 드리... 집탱구리

[JSON] Jso... 집탱구리

ABOUT ME

요리를 좋아하는 개발자 일상 블로그

TAG API , chartjs , JavaScript , jQuery , 오픈소스 , 차트 , 차트 만들기 , 캔버스

Today 145 Yesterday 244

Total 116,936

관련글

관련글 더보기

[구글맵API]구글맵 API사용하기 2016.12.19

[jQuery]파일용량 계산하기 2016.12.14

[PHP]배열 키값 재정의 2016.12.13

[OVEN]프로토타이핑 툴 오븐!... 2016.11.25

댓글 0

댓글을 입력해주세요.

이름

비밀번호

비밀글

등록

COOKING (17)

IT (21)

REVIEW (41)

TRAVEL (0)

국내 (0)

해외 (0)

‹ 이전 1 … 73 74 75 76 77 78 79 80 다음 ›

인기포스트

[MySQL] 문자열 자르기, 합치.. 집탱구리

[퓨어썸샤워기]필터 달린 샤워기 집탱구리

[jQuery]트리구조 플러그인 -.. 집탱구리

[JSON] Json Parser Online -.. 집탱구리

ABOUT ME

요리를 좋아하는 개발자 일상 블로그

Today 145 Yesterday 244

Total 116,936

LINK

Suvely

자과차 향기나는 이야기

문화공간

yoovely

랭킹 코리아

멀리서보이는 경치

집탱구리의 코드

ADMIN