

IBChart

개발자가이드

IBChart 개발자 가이드

초판발행/ 2012년 3월 20일

지은이/ 임흥택

펴낸이/ 정광천

펴낸곳/



08394 서울시 구로구 디지털로 242(구로동) 한화비즈메트로 1501호

전화/ (02)2621-2080

팩스/ (02)2621-2088

편집/ 임흥택 인쇄/ C&G

※ 기술상담 및 내용문의는(주)아이비리더스나 아이비시트닷컴으로 해주십시오.
<http://www.ibleaders.co.kr> | <http://www.ibsheet.com>.

목 차

Chapter 1. IBChart 개요	13
1. IBChart 개요	15
1.1 IBChart 정의	15
1.2 IBChart 특징	15
1.3 IBChart의 적용 대상 업무	15
1.4 IBChart의 주요 기능	15
Chapter 2. IBChart의 구조	19
2. IBChart의 구조	21
2.1 IBChart의 차트 유형	21
2.2 IBChart의 각 부분 명칭	26
Chapter 3. IBChart의 개발환경 설정	28
3. IBChart의 개발 환경 설정	30
3.1 IBChart의 Server 설치	30
3.1.1 배포파일 및 내용	31
Chapter 4. 개발 절차	34
4. 개발 절차	36
4.1 웹 페이지에 JS 파일 링크 걸기	36
4.2 차트 객체 생성	37
4.3 차트 구성 옵션 설정	39
4.3.1 차트 기본 속성 설정	42
4.3.2 타이틀 및 서브타이틀 속성	46
4.3.3 축(Axis) 속성	48
4.3.4 시리즈(series) 속성	52
4.3.5 툴팁(tooltip) 설정	55

4.3.6 범례(legend) 설정	58
4.3.7 차트유형별 설정(plotOptions) 설정	61
4.3.7.1 꺾은선(line) 차트 설정	62
4.3.7.2 영역(area) 차트 설정	64
4.3.7.3 부드러운선(spline)차트 설정	66
4.3.7.4 산점(scatter)차트 설정	68
4.3.7.5 막대(column)차트 설정	70
4.3.7.6 가로막대(bar)차트 설정	72
4.3.7.7 파이(pie) 차트 설정	74
4.3.7.8 박스플롯(boxplot) 차트 설정	76
4.3.7.9 방사형(polar) 차트 설정	79
4.3.7.10 워드클라우드(wordcloud) 차트 설정	81
4.4 데이터 조회	84
4.5 속성 변경사항 적용하기	87
4.6 데이터 XML 구현하기	89
4.7 데이터 JSON 구현하기	90
Chapter 5. 개발 실습	93
5. 개발 실습	95
5.1 IBChart 객체 생성 및 기본 속성 설정하기	95
5.2 조회된 데이터 사용	96
Chapter 6. 데이터 구조	98
6. 데이터 구조	100
6.1 데이터 XML	101
6.2 데이터 JSON	103
Chapter 7. IBChart 상수	105
7. IBChart 상수	107
7.1 IBLANG	107
7.1.1 chart	108

7.1.2 alert	109
7.1.3 text	110
Chapter 8. IBChart Events	113
8. IBChart Events	115
8.1 Event 사용	115
8.2 Event List	115
addSeries	116
afterPrint	117
beforePrint	118
click	119
drilldown	121
drillup	123
drillupAll	124
pointClick	125
pointMouseOut	127
pointMouseOver	129
pointSelect	131
pointUnselect	133
pointUpdate	135
redraw	136
searchEnd	137
selection	139
seriesAfterAnimate	140
seriesCheckBoxClick	141
seriesClick	143
seriesHide	145
seriesLegendItemClick	147
seriesMouseOut	149
seriesMouseOver	151
seriesShow	153

Chapter 9. IBChart Methods	155
9. IBChart Methods	157
9.1 Chart Object	157
addSeries	158
doPrint	159
down2Image	160
draw	162
getAlertText	163
getData	164
getId	165
getLang	166
getOptions	167
getSeries	168
getSVGString	169
loadSearchData	170
parseText	172
removeAll	174
removeSeries	175
series	176
setLang	177
setOptions	178
setEventListener	181
setEvents	183
setSize	185
updateWords	186
version	187
9.2 Series	188
addPoint	189
getData	191
hide	192
remove	193

removePoint	194
setData	195
setOptions	196
show	198
Chapter 10. IBChart Property	200
10. IBChart Property	202
10.1 chart	203
type	204
plotBackgroundColor	207
plotBorderColor	209
plotBorderWidth	211
plotShadow	213
inverted	215
polar	217
zoomType	219
backgroundColor	220
borderColor	222
borderWidth	224
borderRadius	226
shadow	228
10.2 pane	230
startAngle	231
endAngle	234
background	237
10.3 plotOptions	240
animation	242
borderColor	243
borderRadius	245
borderWidth	247
center	249
cursor	251

colorIterate	253
colorSort	256
dashStyle	259
dataLabels	261
endAngle	264
fillColor	266
groupPadding	268
innerSize	270
lineColor	272
lineWidth	274
marker	276
negativeColor	278
negativeFillColor	280
orientation	282
pointStart	285
pointPadding	287
pointPlacement	289
pointWidth	291
rotation	294
shadow	297
showInLegend	299
stacking	301
states	303
startAngle	305
scaleType	307
step	310
spiralType	312
placeType	314
valueRange	317
style	320
visible	323
10.4 subtitle	325

text	326
align	327
verticalAlign	329
x	330
y	331
style	332
useHTML	334
10.5 title	335
text	336
align	337
verticalAlign	339
x	340
y	341
style	342
useHTML	344
10.6 legend	345
enabled	347
align	349
verticalAlign	351
backgroundColor	353
borderColor	355
borderWidth	357
borderRadius	359
layout	361
title	363
x	366
y	368
10.7 tooltip	371
pointFormat	389
headerFormat	394
useHTML	396
shared	398

valuePrefix	400
valueSuffix	402
enabled	404
animation	406
backgroundColor	408
borderColor	410
borderWidth	412
borderRadius	414
shadow	416
style	418
10.8 xAxis	420
title	422
categories	425
crosshair	427
reversed	429
showFirstLabel	431
showLastLabel	433
lineWidth	435
lineColor	437
startOnTick	439
endOnTick	441
labels	443
opposite	446
min	448
max	450
tickInterval	452
gridLineWidth	456
gridLineColor	458
gridLineDashStyle	460
plotBands	462
offset	465
minorTickInterval	467

minorGridLineWidth	469
minorGridLineColor	471
minorGridLineDashStyle	473
alternateGridColor	476
10.9 yAxis	478
title	480
categories	483
crosshair	485
reversed	487
showFirstLabel	489
showLastLabel	491
lineWidth	493
lineColor	495
labels	497
opposite	500
min	502
max	504
tickInterval	506
gridLineWidth	508
gridLineColor	510
gridLineDashStyle	512
plotBands	514
offset	517
minorTickInterval	519
minorGridLineWidth	521
minorGridLineColor	523
minorGridLineDashStyle	525
alternateGridColor	527
Appendix	530
1. 특수차트	533
1.1 Angular gauges chart	534

1.2 ColumnRange chart	536
1.3 ErrorBar chart	538
1.4 Funnel chart	540
1.5 Bubble chart	542
1.6 HeapMap chart	544
1.7 Prymid chart	547
1.7 Wordcloud chart	549
2. 차트기능	556
2.1 stack 의 그룹기능	557
2.2 멀티 차트 표현 기능	559
2.3 Secondary Y 축 사용	561
2.4 plotLine 기능	563
3. IBSheet 데이터 연동	566

Chapter 1. IBChart 개요

1. IBChart 개요

1.1 IBChart 정의

IBChart는 웹 개발 환경에 가장 적합하도록 설계된 차트 컴포넌트이다. 개발자들의 노력을 최소화하면서 웹 환경에서 차트를 표현할 수 있도록 하는 것이 일차적인 목적이다. IBChart는 웹 서버 상의 동적 데이터를 조회 가능하게 함으로써 사용자가 선택한 조건에 맞는 차트를 보여줄 수 있다.

1.2 IBChart 특징

- 크로스 브라우징, 크로스 플랫폼을 지원
- 별도의 개발툴이 필요없고 일반적인 Javascript를 통하여 각 기능을 설정함으로 웹개발자(ASP.net, JSP, PHP 등)가 쉽게 이해하고 응용하여 기능 구현이 가능
- 모든 주요 브라우저 및 운영체제에 일관되게 배포 가능
- 개발 생산성을 대폭 향상시킬 수 있는 개발 방안과 환경을 제공
- JSON와 XML을 통한 데이터 통신 방식 사용
- 메시지 파일 형식을 통한 다국어 지원 및 적용 용이
- 개발자에게는 쉬운 개발을 운용자에게는 편리한 운용 방안 제공

1.3 IBChart의 적용 대상 업무

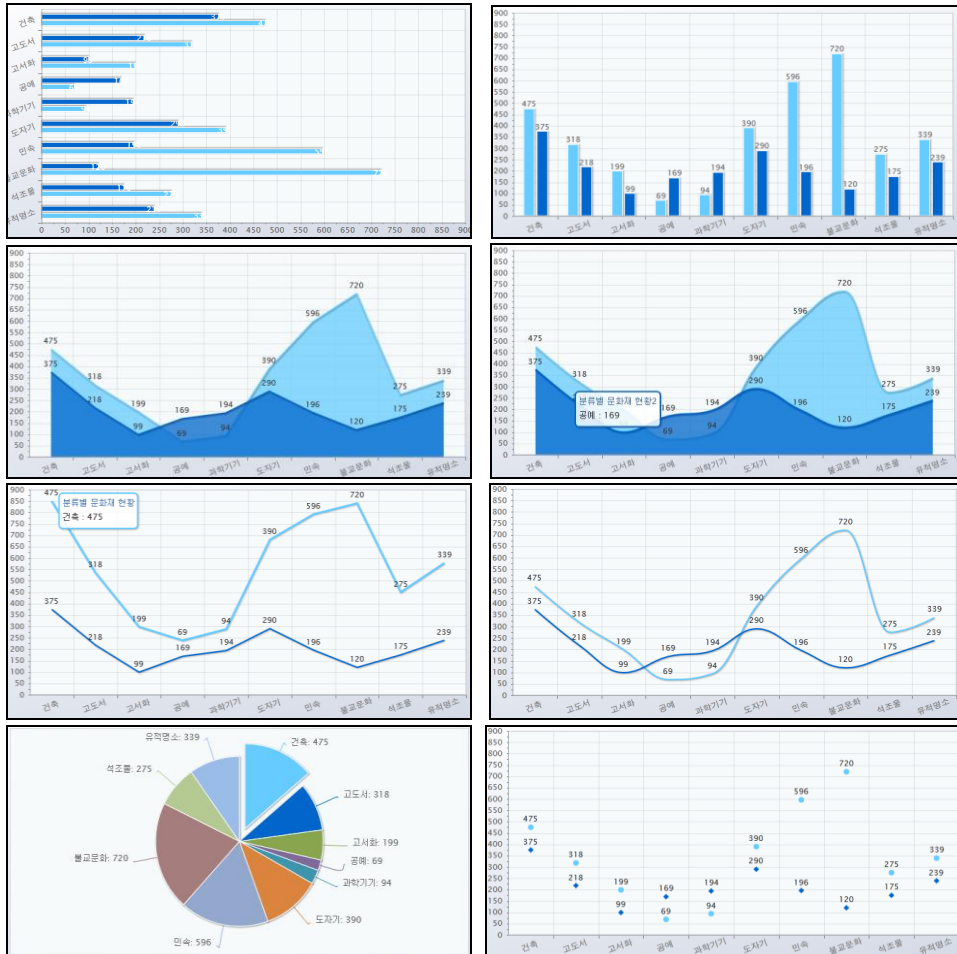
- 조회된 데이터를 그래픽컬하게 보여줄 필요가 있는 업무(데이터 시각화)
- 기존 C/S 환경과 비슷한 구조로 업무 환경을 웹으로 변경하고자 하는 업무

1.4 IBChart의 주요 기능

➤ 다양한 차트 지원

IBChart는 일반적으로 많이 쓰이는, 컬럼 차트, 꺾은선 차트 등을 비롯하여,

바 차트, 영역 차트, 도넛 차트 등을 지원한다. 이러한 다양한 차트를 지원하기 때문에 개발자와 사용자는 원하는 형태로 데이터를 자유롭게 표현할 수 있다.



[그림 1-1 다양한 차트 타입]

➤ **편리한 개발 방식 제공**

IBChart는 기존 웹 개발자들에게 익숙한 자바스크립트를 이용하여 차트의 속성을 설정하고 제어할 수 있다.

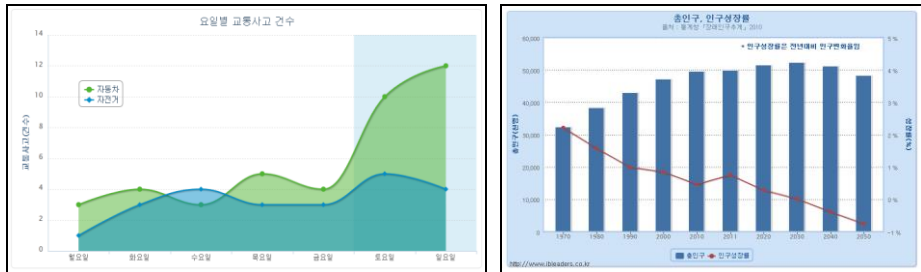
➤ **풍부한 디자인 옵션**

차트내의 개별 속성의 색상 선택 및 그라데이션 등의 지정이 가능하며, 각각

웹 디자인 환경에 맞춰 차트를 표현할 수 있다.

➤ 다양한 특수 효과 지원

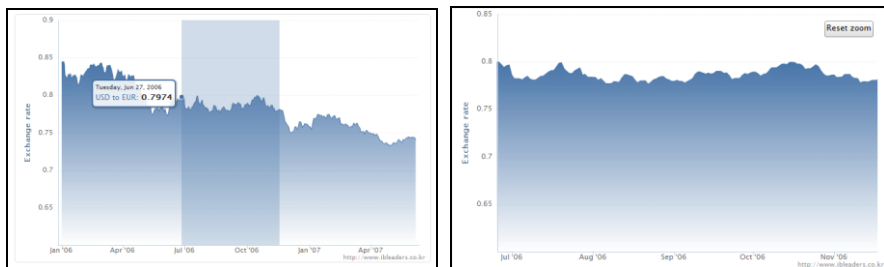
IBChart는 차트 요소의 다양한 디자인 속성 이외에 차트를 조금 더 화려하게 보일 수 있는 특수 효과를 제공한다.



[그림 1-5 영역설정 효과와 그림자 효과]

➤ 차트 확대 기능

수 많은 데이터를 차트로 표현하면, 데이터를 식별하기가 어려워진다. IBChart는 이러한 경우에 사용자가 원하는 차트 영역만큼 차트를 확대할 수 있도록 마우스 드래그를 통한 차트 확대 기능을 제공한다. 또한 확대된 차트는 스크롤바를 통하여 이동하는데, 불편함이 있을 수 있어, 마우스 드래그를 통한 차트 이동 기능을 제공한다.

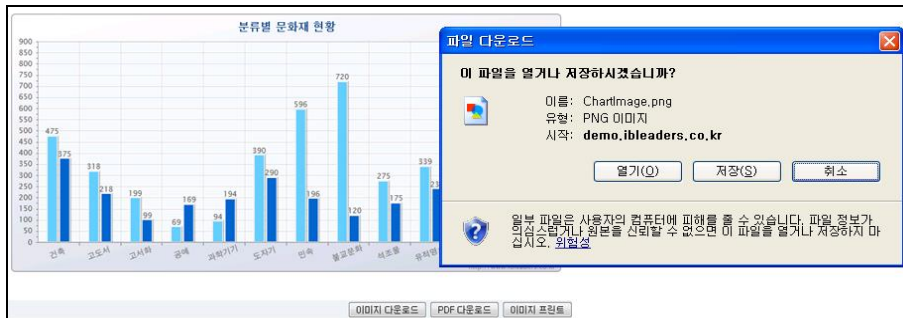


[그림 1-6 마우스로 선택한 영역 확대]

➤ 이미지 파일 저장

IBChart는 현재 화면의 차트를 JPG, PNG 이미지로 저장하는 기능과 PDF, Excel, SVG 문서로 제공하여, 사용자가 원하는 경우에 차트를 이미지로 저장한

후 다른 문서에서 편리하게 삽입, 편집, 응용할 수 있다.



[그림 1-8 이미지 저장 기능]

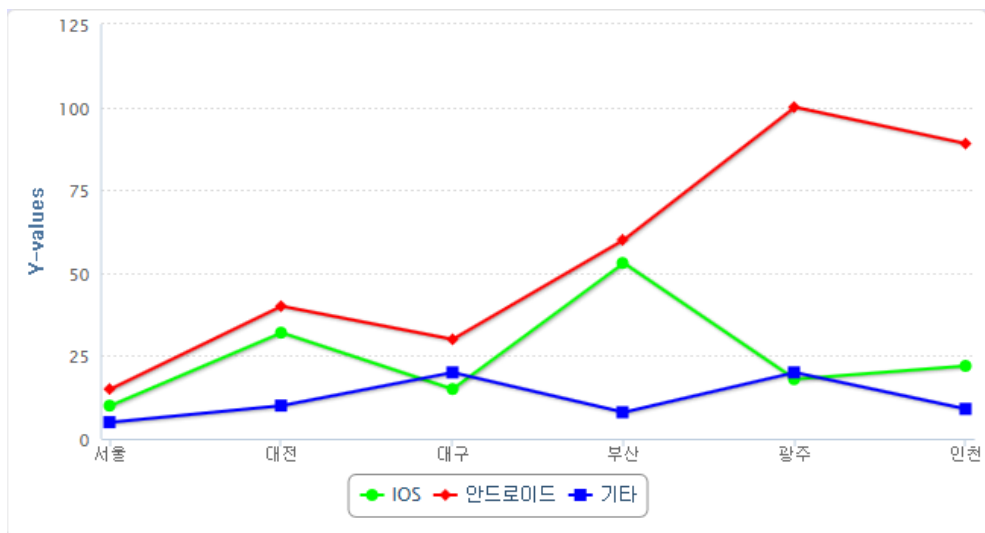
Chapter 2. IBChart의 구조

2. IBChart의 구조

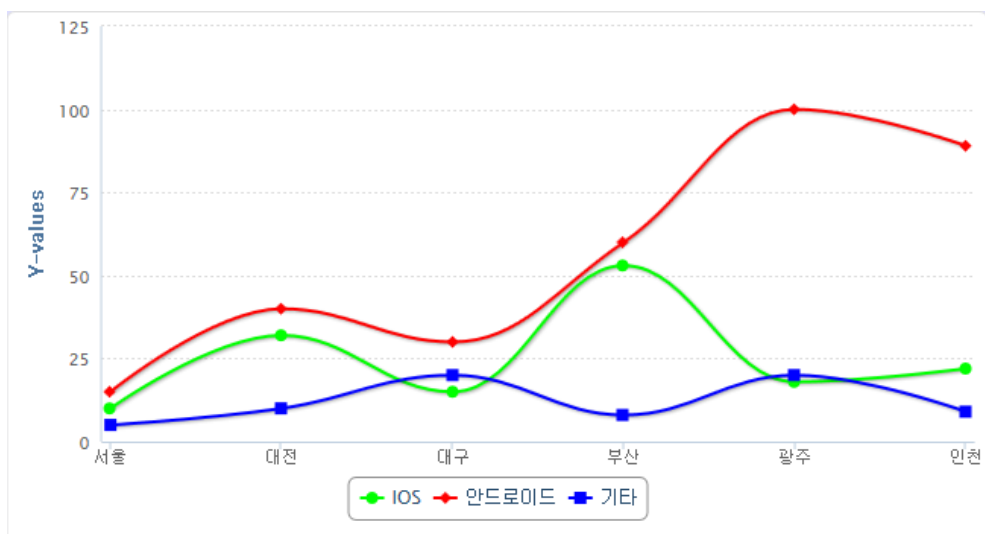
2.1 IBChart의 차트 유형

➤ IBChart의 형태는 아래와 같이 표현할 수 있다.

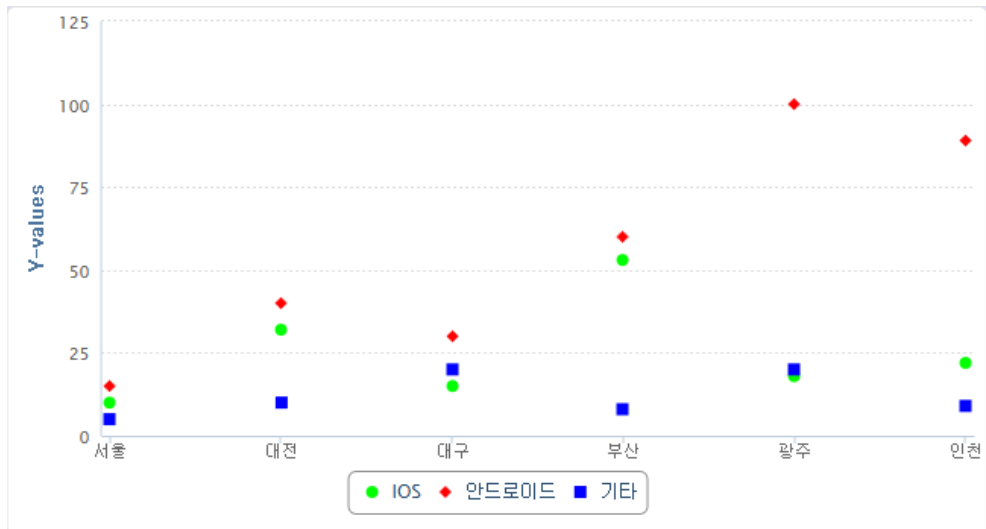
- 꺾은선(Line Type)



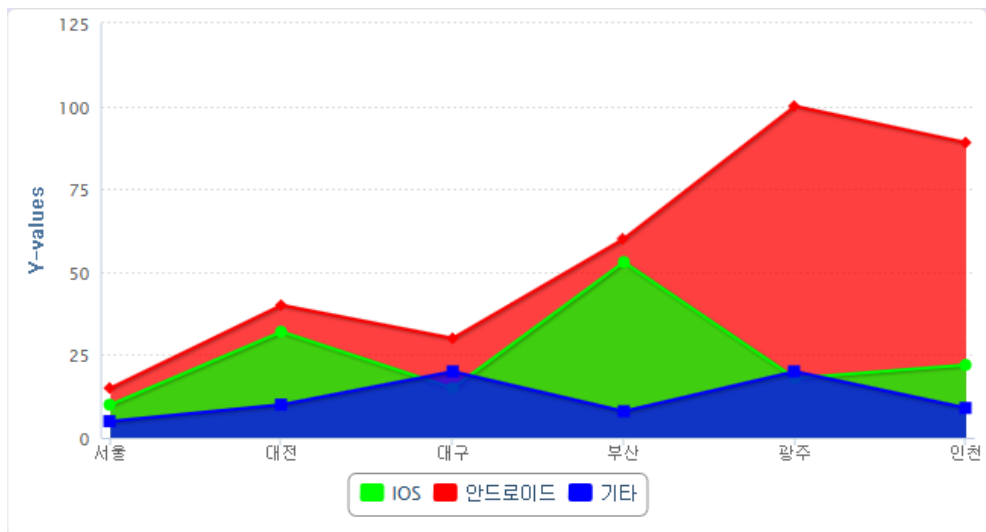
- 부드러운 선(Spline Type)



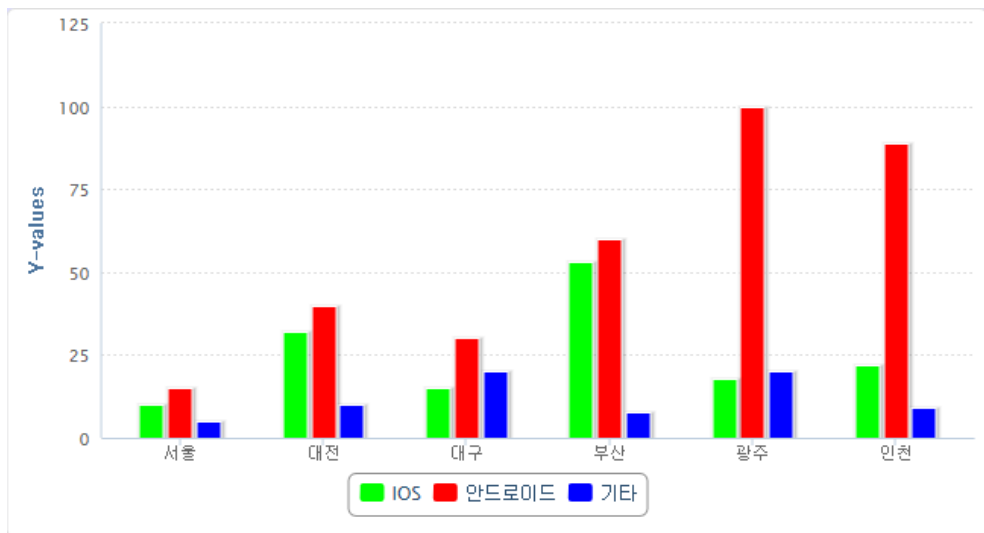
● 산점(Scatter Type)



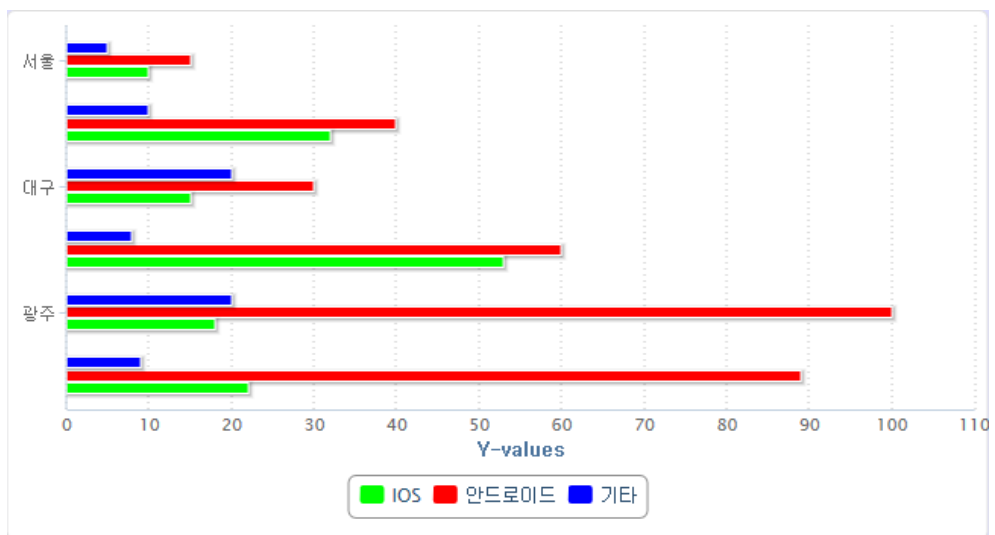
● 영역(Area Type)



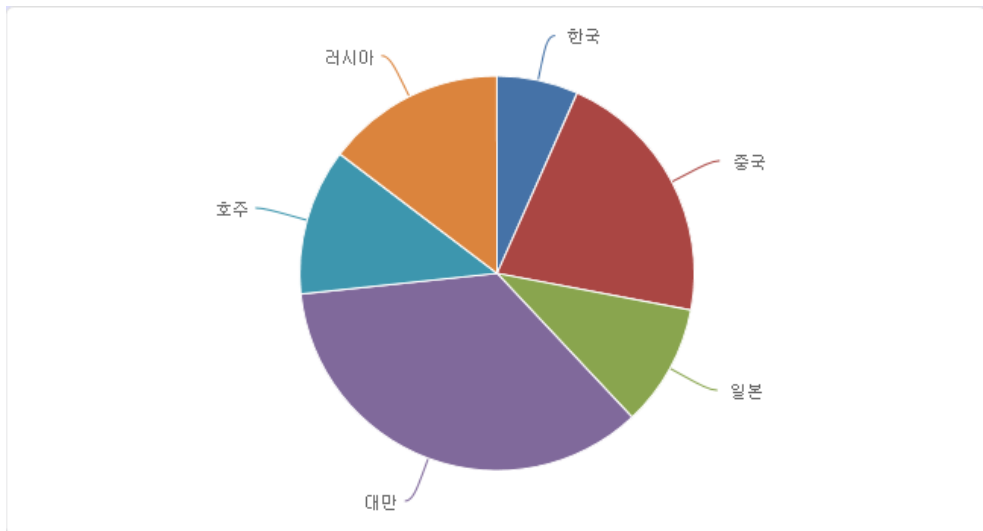
● 막대(Column Type)



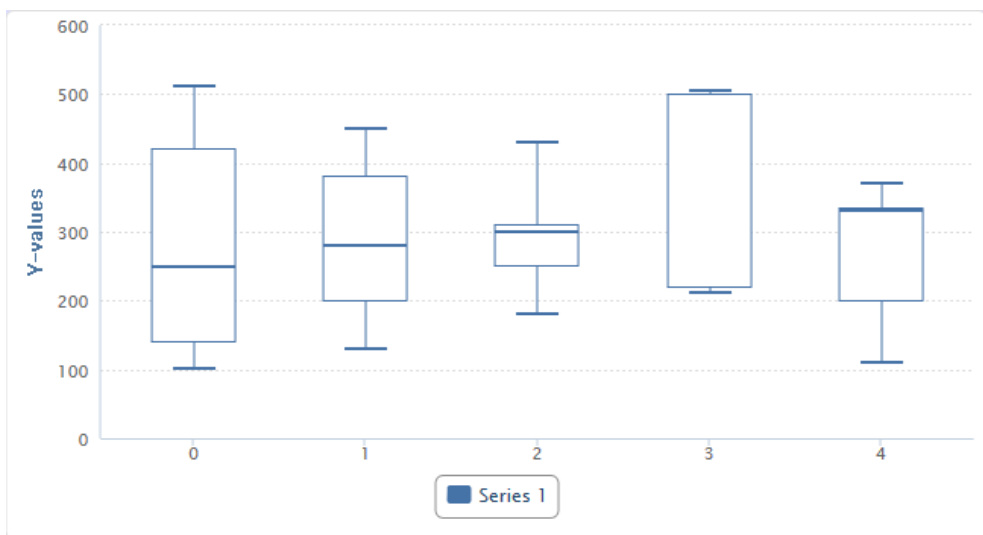
● 가로막대(Bar Type)



- 파이(Pie Type)



- 박스플롯(Box Plot Type)



- 방사형(Polar Type)



- 워드클라우드형(Wordcloud Type)



2.2 IBChart의 각 부분 명칭



- ① title : 차트의 제목을 나타내면 기본값으로 차트의 윗부분에 표현된다.
- ② series : 각 포인트의 집합이다. 차트는 1개 혹은 여러 개의 시리즈로 구성된다.
- ③ tooltip : 시리즈에 마우스 오버시 표현되는 내용이다.
- ④ legend : 범례(차트 시리즈 이름과 색상(모양)을 보여준다)
- ⑤ axis : X축과 Y축을 말한다.

Chapter 3. IBChart의 개발환경 설정

3. IBChart의 개발 환경 설정

3.1 IBChart 의 Server 설치

3.1.1 배포파일 및 내용

IBChart를 이용한 개발 환경을 구축하기 위해서 아래와 같이 제공되는 파일들을 웹사이트의 특정 폴더에 복사하여 설치한다.

위치는 다음을 기준으로 한다.

```

WibchartW
WibchartWibleaders.js
WibchartWibchart.js
WibchartWibchartinfo.js
WibchartWDown2Image.jsp
  
```

각각 파일의 내용을 설명하자면 아래와 같다.

파일명	내용/역할
ibleaders.js	IB Sheet 7 제품군 공통 js 파일
ibchart.js	IBChart 엔진 js 파일
ibchartinfo.js	IBChart 생성 함수 및 개발시 유용하게 사용할 수 있는 js함수를 담아둔 파일(지속적 업데이트 예정)
Down2Image.jsp	<p>이미지 다운로드 전용함수의 기능 구현을 위한 서버쪽 응용 프로그램 파일</p> <p>※ 한글이 깨질경우 jsp 내에서 UTF-8 환경설정 변수값을 변경해야 한다.</p> <p>※ 이미지 다운로드 jsp 프로그램 내에서 이미지 파일을 서버 디스크에 임시 저장하므로 TempFolder 변수에 서버의 물리적 디렉토리 경로를 설정해야 한다.</p>
lbsheet7_1.x.x.jar	다운로드를 위한 jar 파일

이미지 다운로드 사용중 한글이 깨질 경우 웹페이지의 charset의 UTF-8 설정 또는 Down2Image.jsp 의 isUTF8 설정값을 변경하여 그 설정값이 서로 일치해야 한다. 또한 Down2Image.jsp 파일 안에서 이미지 파일을 임시로 저장할 경로를 해당 서버의 환경에 맞게 정확히 지정해야 한다.

이미지 다운로드의 기능을 원활히 지원하기 위해서는 WAS 단에 batik 라이브러리와 ibsheet7.jar 가 설치되어 있어야 한다. 아래와 같이 lib 폴더 안에 파일을 설치한 경우 WAS 서버를 재부팅해야 적용된다.

Chapter 4. 개발 절차

4. 개발 절차

4.1 웹 페이지에 JS 파일 링크 걸기

IBChart 객체를 생성하고, 생성된 객체에 여러가지 기능 동작 수행을 위한 4가지 js 파일을 링크 건다.

```
<script type="text/javascript" src="/highcharts/highcharts.js"> </script>
<script type="text/javascript" src="/chart/ibleaders.js"> </script>
<script type="text/javascript" src="/chart/ibchartinfo.js"> </script>
<script type="text/javascript" src="/chart/ibchart.js"> </script>
```

위 4개 파일 중 highcharts.js 파일은 highsoft의 highcharts이다. IBChart를 이용하려면 highcharts를 사용해야한다. highcharts.js는 기본으로 포함시키고 필요에 따라 highcharts의 module를 추가해서 사용한다. ibchart.js 파일은 IBChart 엔진 js 파일이다. 그리고 ibchartinfo.js 파일은 차트 생성 함수 및 개발 시 유용하게 사용할 수 있는 js함수를 담아둔 파일이다.

4.2 차트 객체 생성

차트 객체 생성은 차트가 만들어질 컨테이너가 구성된 이후에 설정 할 수 있다. body 태그 하위에 컨테이너로 사용할 div 를 생성 하고, body 태그의 onload 이벤트 또는, jQuery 라이브러리를 사용하는 경우 \$(document).ready(); 에서 설정 하여야 한다.

객체 생성은 createIBChart 함수를 이용하여 생성하며, 세부 설명은 아래와 같다.

syntax	createIBChart(cont, id, opt)		
name	type	required	description
cont	string object	Y	컨테이너의 id 또는 DOM 객체
id	string	Y	생성할 차트의 id
opt	object	Y	차트 생성 옵션
opt.width	string	Y	생성할 차트의 너비 (단위:px)
opt.height	string	Y	생성할 차트의 높이 (단위:px)
opt.locale	string	N(kr)	언어 설정
opt.plugin	string	N(hc)	차트 플러그인 설정

차트를 생성 하는 예제는 아래와 같다.

```
<!DOCTYPE html>
<html>
<head>
...
<script type="text/javascript">
  // 일반적으로 사용 시
  function fnLoadPage() {
    // 차트 객체 생성
    createIBChart("cont", "myChart", {
      width: "500px",
      height: "500px"
```

```

    });

}
</script>
</head>
<body onload="fnLoadPage();">
    <div id="cont"></div>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
...
<script type="text/javascript">
    // jQuery 라이브러리 사용시
    $(document).ready(function() {
        // 차트 객체 생성
        createIBChart("cont", "myChart", {
            width: "500px",
            height: "500px"
        });
    });
</script>
</head>
<body>
    <div id="cont"></div>
</body>
</html>

```

4.3 차트 구성 옵션 설정

차트 객체 생성 완료 후 `setOptions` method 를 이용 하여 차트 구성 옵션을 설정한다. 아래의 예제는 사용 위치에 대한 설명이며, 구성 옵션에 대한 세부적인 설정은 이 문서 10장의 설명을 참고 한다.

IBChart객체.**setOptions**(prop , [opt]);

```
예) var prop= {  chart:{ type: "spline" },
                title:{text:"월별 재고량"},
                xAxis:{ .... },
                ....
            };

myChart.setOptions(prop, {append:true,redraw:true});
```

➤ prop Info

속성	타입	설명
chart	object	차트배경 및 형태 등을 설정
title	object	차트 제목 및 제목의 형태 등을 설정
subtitle	object	차트 부제목 및 부제목의 형태 등을 설정
colors	array	시리즈별 적용할 색상의 배열 집합 (배열의 길이보다 시리즈가 더 많을 경우 다시 첫번 첫부터 반복 적용 된다.)
xAxis	object	X축을 설정
yAxis	object	Y축을 설정
plotOptions	object	plot영역 (차트세부)설정 설정
tooltip	object	툴팁 속성 및 형태 등을 설정
pane	object	pane 속성 및 형태 등을 설정 (guage,polar 유형에서만 사용)
legend	object	범례의 위치 및 형태 설정.

➤ **opt Info**

속성	타입	설명
append	boolean	기존설정과 병합(merge)처리 여부. default:false true로 설정하지 않는 경우 이전에 설정했던 내용은 clear하고 새로이 설정된다.
redraw	boolean	설정적용후 다시 그리기. default:true
resetData	boolean	데이터 병합 여부 (속성을 유지하면서 데이터 초기화 여부, ※ `opt.append`값과 `opt.mergeData`값이 `false`이면 데이터 초기화) default:false
mergeData	boolean	데이터를 기존 데이터에 덮어씌울지 여부 (`opt.resetData`값이 `false`일때만 적용됨) default:true
resetColors	boolean	시리즈 색상값(colors) 병합 여부 (속성을 유지하면서 데이터 초기화 여부, ※ `opt.append`값과 `opt.mergeColors`값이 `false`이면 데이터 초기화) default:false
mergeColors	boolean	시리즈 색상값(colors)을 기존 시리즈 색상값에 덮어씌울지 여부(`opt.resetColors`값이 `false`일때만 적용됨) default:true
deepCopy	boolean	깊은 복사(데이터량에 따라 시간이 더 필요할 수 있음) default:false

사용 예제는 아래와 같다

```
<!DOCTYPE html>

<html>
<head>
...
<script type="text/javascript">
    // 일반적으로 사용 시
    function fnLoadPage() {
        // 차트 객체 생성
        createIBChart("cont", "myChart", {
            width:"500px",
            height:"500px"
        });

        // 차트 구성 옵션 설정
        myChart.setOptions({
            ...
        });
    }
</script>
</head>
<body onload="fnLoadPage();" >
    <div id="cont"></div>
</body>
</html>
```

4.3.1 차트 기본 속성 설정

차트 기본 속성 설정은 `setOptions` 의 **chart** 속성을 이용하여 정의 한다.

이 속성에서는 차트 유형, 차트의 배경색, 확대/축소기능 사용여부 등을 정의 할 수 있다.

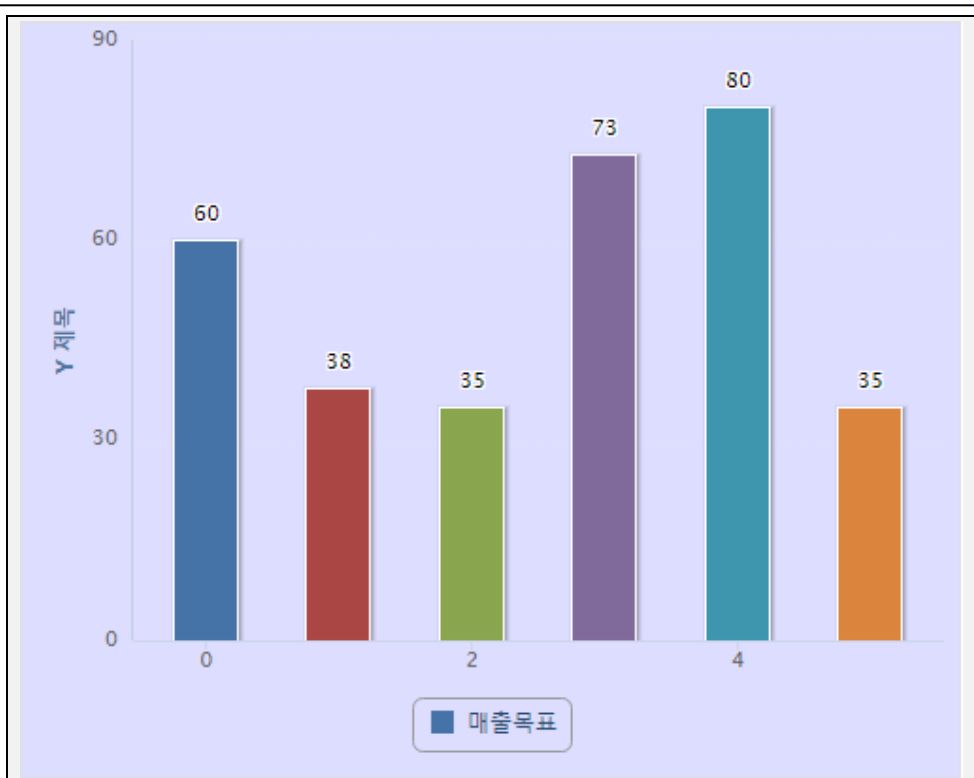
➤ Info

name	type	required	description
type	string	N(line)	차트 유형
<code>plotBackgroundColor</code>	color	N	차트 영역의 배경색상
<code>animation</code>	boolean	N(true)	애니메이션 사용 여부
<code>plotBorderColor</code>	color	N(#CCC)	차트 영역의 border 색상
<code>plotBorderWidth</code>	number	N(0)	차트 영역의 border 너비
<code>plotShadow</code>	boolean	N(false)	차트 영역에 대한 그림자 표시 여부
<code>inverted</code>	boolean	N(false)	X축과 Y축 반전 여부
<code>polar</code>	boolean	N(false)	polar 차트 구성 여부
<code>zoomType</code>	string	N(none)	마우스 드래그시 zoom 적용 방법 설정
<code>backgroundColor</code>	color	N	차트 컨테이너 영역의 배경색상
<code>borderColor</code>	color	N	차트 컨테이너 영역의 border 색상
<code>borderWidth</code>	number	N(0)	차트 컨테이너 영역의 border 너비
<code>borderRadius</code>	number	N(0)	차트 컨테이너 영역에 대한 radius 적용 값
<code>shadow</code>	boolean	N	차트 컨테이너 영역에 대한 그림자 표시 여부

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart : {
    backgroundColor : "#dddddff", //차트 배경색 설정
    type : "column", //차트 Type 설정
    zoomType : "x"
  },
  legend : {
    layout : "vertical", //Legend 모양 설정
    align : "center", //Legend 가로 정렬 설정
    verticalAlign : "bottom" //Legend 세로 정렬 설정
  },
  plotOptions : {
    series : {
      dataLabels : { //시리즈의 데이터 레이블 설정
        enabled : true,
      }
    },
    column : {
      pointPadding : 0.1 // 컬럼간의 간격 설정
    },
  },
  xAxis : {
    tickInterval : 2, //x축 레이블 간격 설정
    labels : { //x축 레이블 설정
      enabled : true
    }
  },
  yAxis : {
    tickInterval : 30, //y축 레이블 간격 설정
    min : 0, //y축 Min값 설정
    title : { //y축 제목 설정
```

```
        text : "Y 제목"
    }
},
series: [{
    name : "매출목표",
    data : [
        {y:60, color : "#4572A7"},
        {y:38, color : "#AA4643"},
        {y:35, color : "#89A54E"},
        {y:73, color : "#80699B"},
        {y:80, color : "#3D96AE"},
        {y:35, color : "#DB843D"}
    ]
}]
}, {
    redraw: true
});
```



4.3.2 타이틀 및 서브타이틀 속성

title 과 **subtitle**을 화면에 추가한다.

추가 옵션(align, float, margin, verticalAlign, x, y)을 이용하여 타이틀과 서브타이틀을 이동할 수 있다. 자세한 내용은 10.4 10.5 장을 참고.

➤ Info

name	type	required	description
text	string	N	제목 문자열
align	string	N(center)	좌우 정렬 값
verticalAlign	string	N	상하 정렬 값
x	number	N(0)	좌우 포지션 이동 값
y	number	N	상하 포지션 이동 값
style	object	N	스타일 설정
style.color	color	N	폰트 색상
style. fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께
useHTML	boolean	N(false)	제목 문자열에 html tag 사용 여부

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: 'cc별 중고자동차 판매량',
    style: {fontSize: "16pt"}
  },
  subtitle: { // subtitle 설정
    text: '2014년 1/4분기'
  }
}, {
  append: true,
  redraw: true
})
```

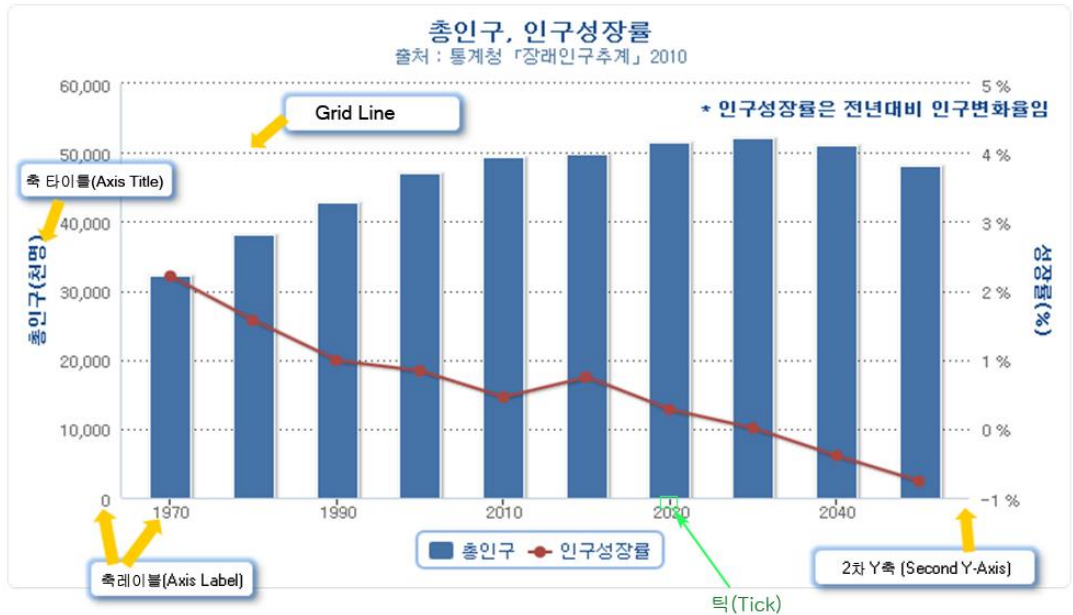
});

cc별 중고자동차 판매량

2014년 1/4분기

4.3.3 축(Axis) 속성

X축(xAxis)과 Y축(yAxis)은 파이,방사형,게이지 차트를 제외한 대부분의 차트에서 기본적으로 표현된다.



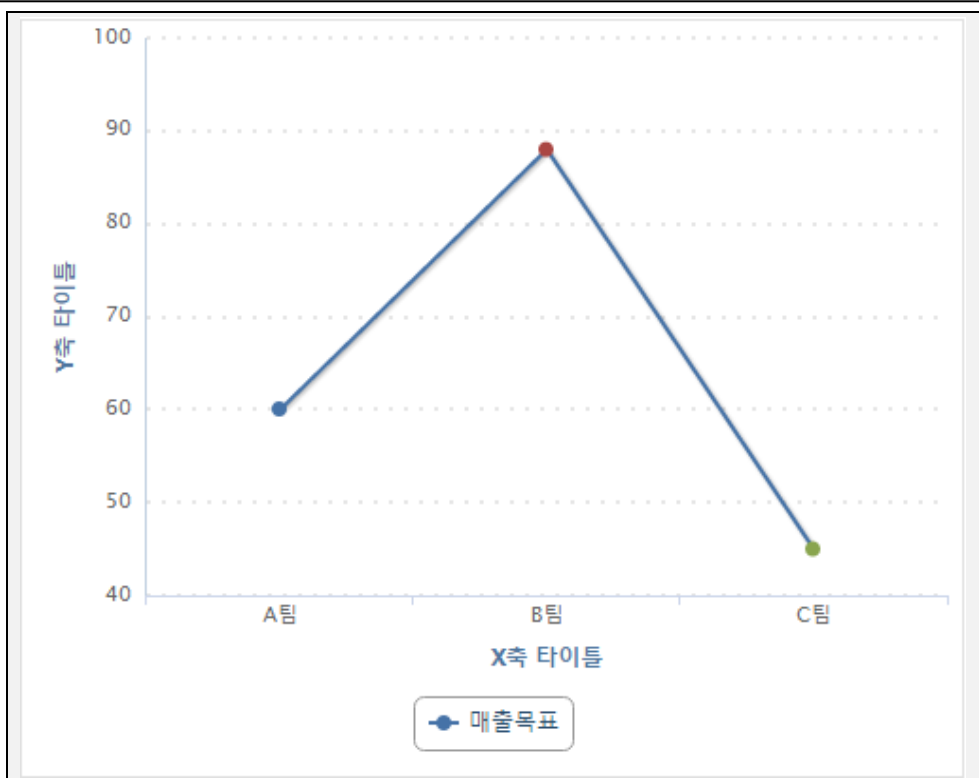
➤ Info

name	type	required	description
title	object	N	축 제목 속성 (JSON 형식)
categories	array	N	축레이블 설정
crosshair	boolean	N(false)	포인트에 마우스 오버시 가이드라인 표시 여부
reversed	boolean	N(false)	축 반전 여부
showFirstLabel	boolean	N(true)	첫번째 축 라벨여부 표시
showLastLabel	boolean	N(true)	마지막 축 라벨여부 표시
lineWidth	number	1	축 라인의 너비
lineColor	color	N(#ccd6eb)	축 라인의 색상
startOnTick	boolean	N(false)	축의 시작점을 강제로 표시할지 여부

endOnTick	boolean	N(false)	축의 마지막점을 강제로 표시할지 여부
labels	object	N	라벨 속성
labels.align	string	N(center)	좌우 정렬 방법
labels.format	string	N({value})	축 라벨의 포맷 문자열 (값은 {value} 문자열을 이용한다.)
labels.formatter	function	N	축 라벨의 포맷을 정의하는 callback 함수
labels.style	object	N	라벨 스타일 속성
labels.useHTML	boolean	N(false)	html tag 사용 여부
labels.rotation	number	N(0)	기울기 (0 ~ 360)
labels.step	number	N	축레이블 표시 간격
labels.staggerLines	number	N	축레이블 지그재그 표시
opposite	boolean	N(false)	x축은 상단, y축은 우측에 축 표시 여부
min	number	N	축의 최소값
max	number	N	축의 최대값
tickInterval	number	N	축에 표시할 점의 간격
gridLineWidth	number	N(0)	그리드 라인의 너비
gridLineColor	color	N(#262626)	그리드 라인의 색상
gridLineDashStyle	string	N(solid)	그리드 라인의 종류
plotBands	object	N	플롯밴드 속성 (특정영역 하이라이팅)
offset	number	N	기본 위치 기준으로 표시할 offset 값
minorTickInterval	number	N	축에 표시할 보조 점의 간격
minorGridLineWidth	string	N	보조 그리드 라인의 너비
minorGridLineColor	color	N	보조 그리드 라인의 색상
minorGridLineDashStyle	string	N	보조 그리드 라인의 종류
alternateGridColor	color	N	그리드 라인의 교차 색상

➤ Example

```
// x,y축 설정
myChart.setOptions({
  xAxis: { // x축 설정
    title:{
      text:"x축 타이틀" // 타이틀 설정
    },
    labels:{rotation:90}, //축레이블 90도 회전
    categories : ["A팀", "B팀", "C팀"] //x축 레이블 설정
  },
  yAxis: { // y축 설정
    title:{
      text:"y축 타이틀" // 타이틀 설정
    },
    gridLineWidth : 2, //라인 넓이설정
    tickInterval : 10 // Tick 간격을 10으로 설정
  },
  series:[{
    name : "매출목표",
    data : [
      {y:60, color : "#4572A7"},
      {y:88, color : "#AA4643"},
      {y:45, color : "#89A54E"},
    ]
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.4 시리즈(series) 속성

plot 영역에 표시될 각 시리즈 객체에 대한 상세 내용을 설정하거나 데이터를 입력한다. ([10.3 표](#)와 유사)

➤ **Info**

name	type	required	description
name	string	N	범례에 표시될 시리즈 이름
data	array<number> array<array> array<object>	Y	series의 data 설정(JSON 배열)
color	string	N	각 차트 시리즈의 색상 설정
type	string	N	시리즈 종류
allowPointSelect	boolean	N(false)	시리즈의 지점 클릭 가능 유무
animation	boolean	N(true)	애니메이션 유무
dataLabels	object	N	시리즈의 각 지점마다 데이터 라벨 표시 유무와 옵션
enableMouseTracking	boolean	N	사용자의 마우스 움직임과 시리즈의 상호작용 유무
negativeColor	color	N	series.threshold의 값 기준으로 시리즈에서 기준 이하의 값의 색상을 설정
cursor	string	N	시리즈 위에 마우스를 올렸을 때 마우스 포인트 모양
selected	boolean	N(false)	시리즈 선택 유무
stack	string number	N	시리즈에서 stacking을 지원할 때 하나의 axis에서 그룹의 이름이나 숫자로 구분해서 표현

shadow	boolean	N(false)	시리즈 그림자 효과 표현 유무
showInLegend	boolean	N(true)	시리즈를 범례에 표시 유무
tooltip	object	N	시리즈별 툴팁 설정
threshold	number	N(0)	series.negativeColor의 기준값
visibel	boolean	N(true)	시리즈 표시 유무
xAxis	number	N(0)	시리즈의 데이터를 표현할 x축의 인덱스
yAxis	number	N(0)	시리즈의 데이터를 표현할 y축의 인덱스 y2축 사용시 설정

➤ Example

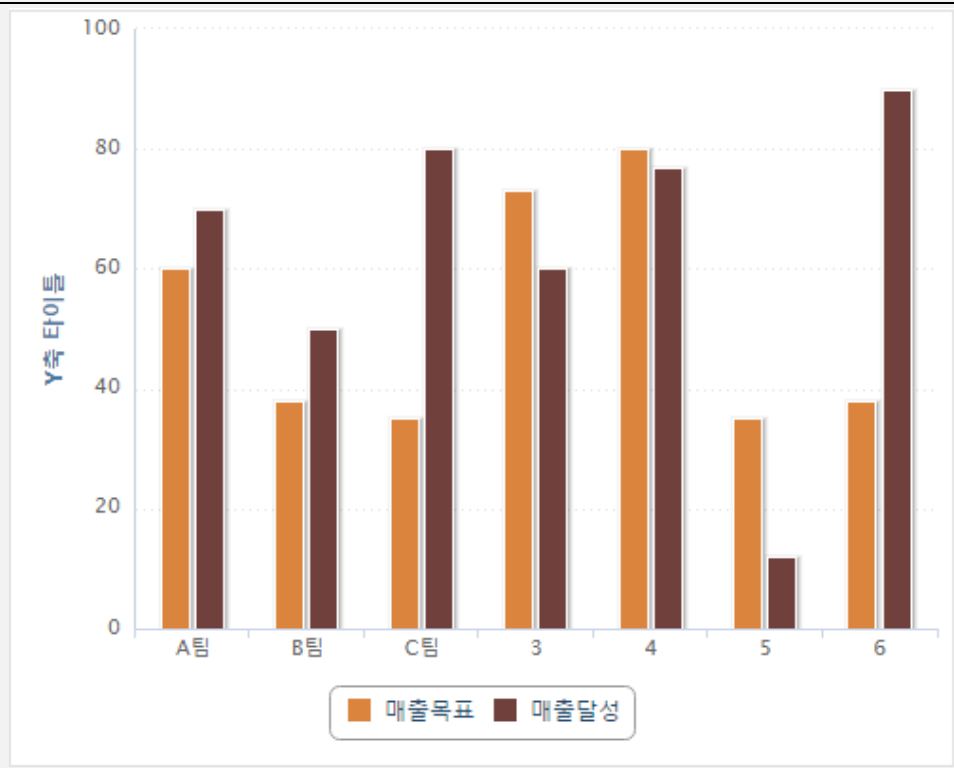
```
//차트속성
myChart.setOptions({
  chart:{
    type:"column"
  },
  xAxis: { // x축 설정
    categories : ["A팀", "B팀", "C팀"] //x축 레이블 설정
  },
  yAxis: {
    title: {
      text: "Y축 타이틀"
    }
  },
  series:[{
    name : "매출목표", //Series Name 설정
    data : [60,38,35,73,80,35,38], //Series Data 설정
    color : "#DB843D" //Series 컬러 설정
  },{
```

```

    name : "매출달성",
    data : [70,50,80,60,77,12,90],
    color : "#70403D"

  }]
}, {
  redraw: true
});

```



4.3.5 툴팁(tooltip) 설정

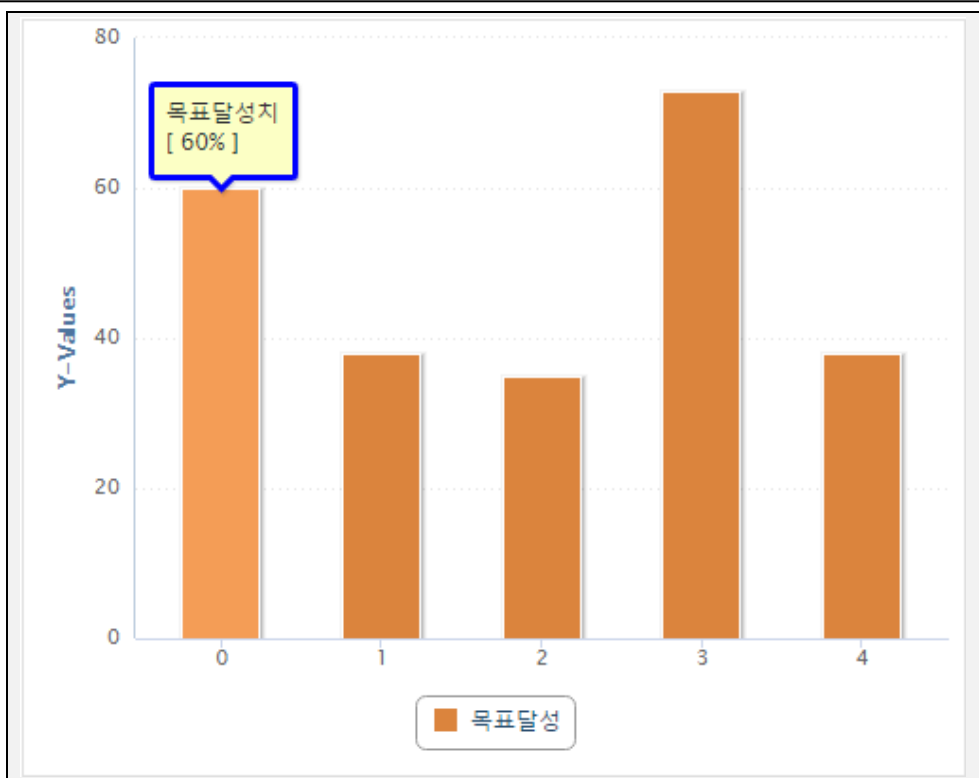
마우스커서가 시리즈 위에 해당 포인트 위에 over 될때 표시될 내용을 정의한다.

➤ Info

name	type	required	description
pointFormat	object	N	포인트 값 출력 포맷
headerFormat	object	N	헤더 출력 포맷
useHTML	boolean	N(false)	html tag 사용 여부
shared	boolean	N(false)	모든 시리즈의 값을 묶어서 출력할지 여부
formatter	function	N	표시할 내용 직접 정의
valuePrefix	string	N	값의 머리 글 문자열
valueSuffix	string	N	값의 꼬리 글 문자열
enabled	boolean	N(true)	툴팁 사용 여부
animation	boolean	N(true)	출력시 애니메이션 효과 사용 여부
backgroundColor	color	N	배경색
borderColor	color	N	border 색상
borderWidth	number	N(1)	border의 너비
borderRadius	number	N(3)	radius 적용 값
shadow	boolean	N(true)	그림자 표시 여부
style	object	N	스타일 속성
style.color	color	N(#333333)	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께
tooltip.formatter	function	N	툴팁 내용변경
tooltip.pointFormat	function	N	툴팁 내용변경
tooltip.headFormat	function	N	툴팁 내용변경

➤ Example

```
//차트 tooltip 속성 설정
myChart.setOptions({
  chart: {
    type: "column" //차트종류
  },
  tooltip: { // tooltip 설정
    enabled :true, //tooltip 사용여부
    formatter : function(){ //tooltip formatter 설정
      return '목표달성치<br> [ '+this.y+'% ]';
    },
    backgroundColor:'#FCFFC5', //tooltip 배경색
    borderColor:'blue', //tooltip border
    borderWidth:3 //border 굵기
  },
  series: [{
    name : "목표달성",
    data : [60,38,35,73,38],
    color : "#DB843D"
  }]
}, {
  redraw: true
});
```

4.3.6 범례(legend) 설정

시리즈에 대한 정보(색, 이름 등)를 박스형태로 보여준다.

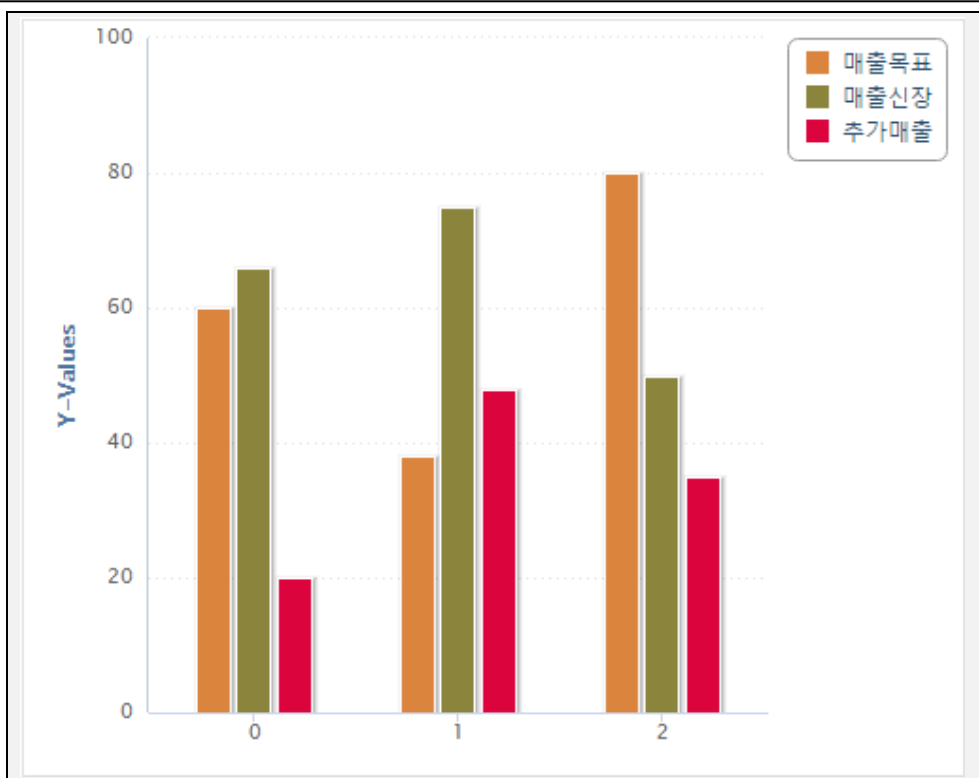
➤ Info

name	type	required	description
enabled	boolean	N(true)	범례 표시 여부
align	string	N(center)	가로 정렬 방법
verticalAlign	string	N(bottom)	세로 정렬 방법
backgroundColor	color	N	배경색
borderColor	color	N(#999999)	border 색상
borderWidth	number	N(1)	border의 너비
borderRadius	number	N(0)	radius 적용 값
layout	string	N(horizontal)	배치 방법 (horizontal, vertical)
title	object	N	제목 속성
title.text	string	N	제목 문자열
title.style	object	N	스타일 속성
x	number	N(0)	좌우 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)
y	number	N(0)	상하 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)

➤ Example

```
//차트속성
myChart.setOptions({
  chart:{
    type:"column" //차트종류
  },
  legend:{ // legend 설정
    layout:"vertical",
```

```
        align:"right",
        verticalAlign:"top"
    },
    series: [{
        name : "매출목표",
        data : [60,38,80],
        color : "#DB843D"
    }, {
        name : "매출신장",
        data : [66,75,50],
        color : "#8B843D"
    }, {
        name : "추가매출",
        data : [20,48,35],
        color : "#DB043D"
    }]
}, {
    redraw: true
});
```



4.3.7 차트유형별 설정(plotOptions) 설정

각 차트 유형별 속성을 설정한다.

➤ plotOptions 속성

속성	타입	설명
series	object	공통 시리즈 속성 설정(JSON 형식) - 모든 시리즈 종류에 적용
line	object	line 차트 속성 설정(JSON 형식)
area	object	area 차트 속성 설정(JSON 형식)
spline	object	spline 차트 속성 설정(JSON 형식)
scatter	object	scatter 차트 속성 설정(JSON 형식)
column	object	column 차트 속성 설정(JSON 형식)
bar	object	bar 차트 속성 설정(JSON 형식)
pie	object	pie 차트 속성 설정(JSON 형식)
boxplot	object	boxplot 차트 속성 설정(JSON 형식)
polar	object	polar 차트 속성 설정(JSON 형식)

4.3.7.1 꺾은선(line) 차트 설정

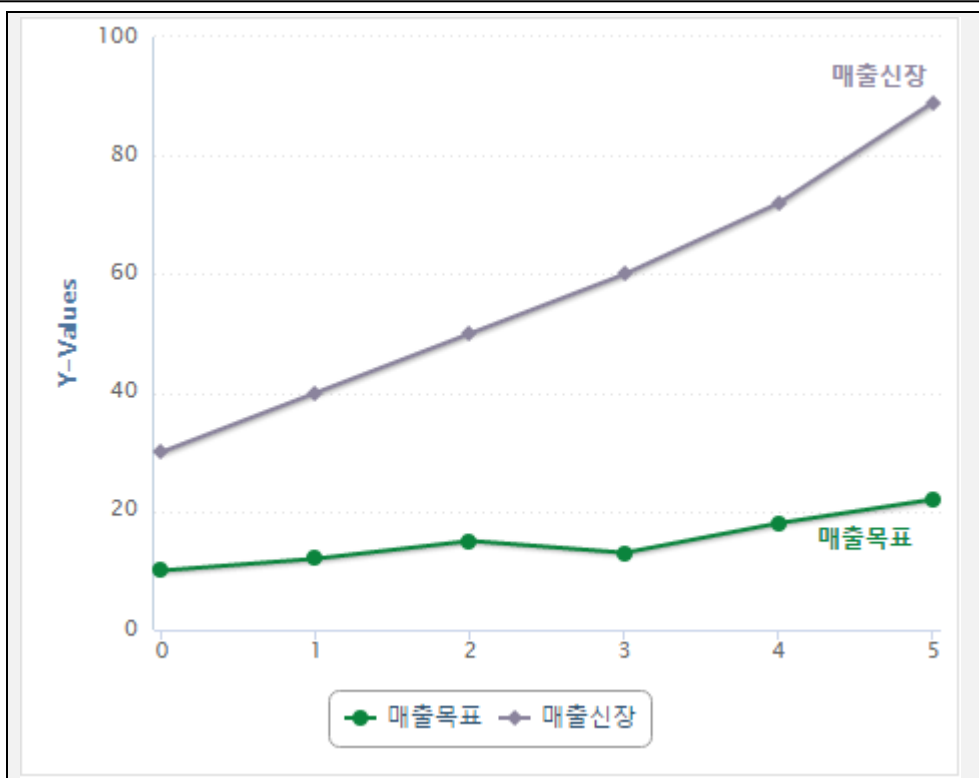
plotOptions의 line 속성에서 설정한다.

- line 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart: {
    type: "line"
  },
  plotOptions: {
    line:{
      cursor:"pointer",
    }
  },
  series: [{
    name : "매출목표",
    data : [10,12,15,13,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [30,40,50,60,null,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.7.2 영역(area) 차트 설정

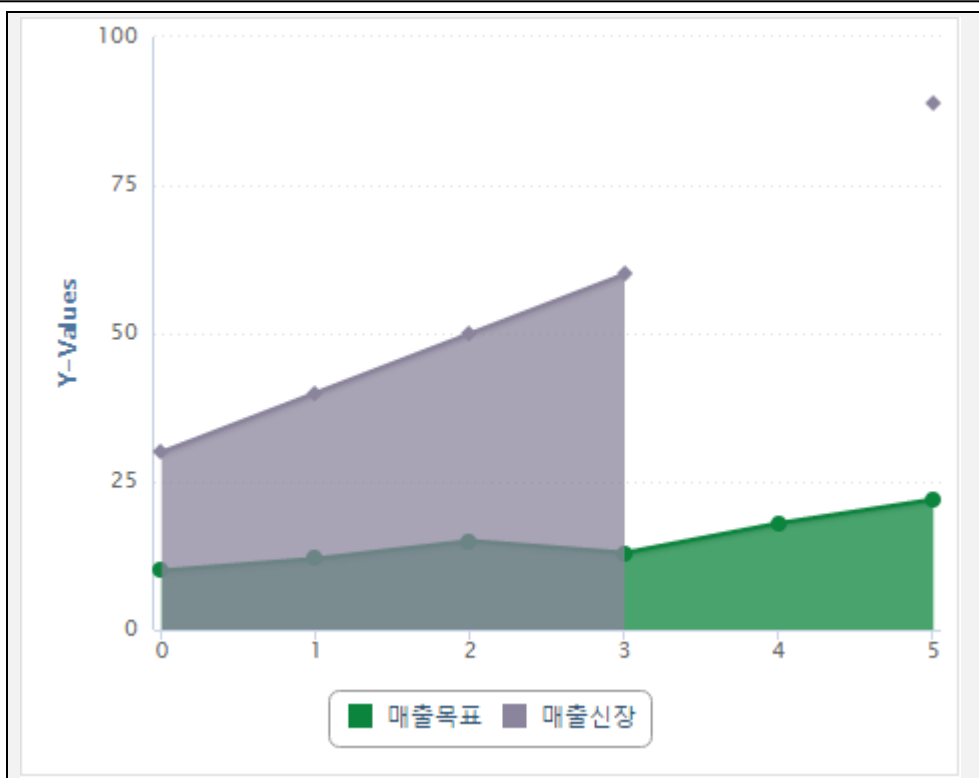
plotOptions의 area 속성에서 설정한다.

- area 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "area"
  },
  plotOptions: {
    line:{
      cursor:"pointer",
      dashStyle:"Dot",
      step:true
    }
  },
  series: [{
    name : "매출목표",
    data : [10,12,15,13,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [30,40,50,60,null,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```

4.3.7.3 부드러운선(spline)차트 설정

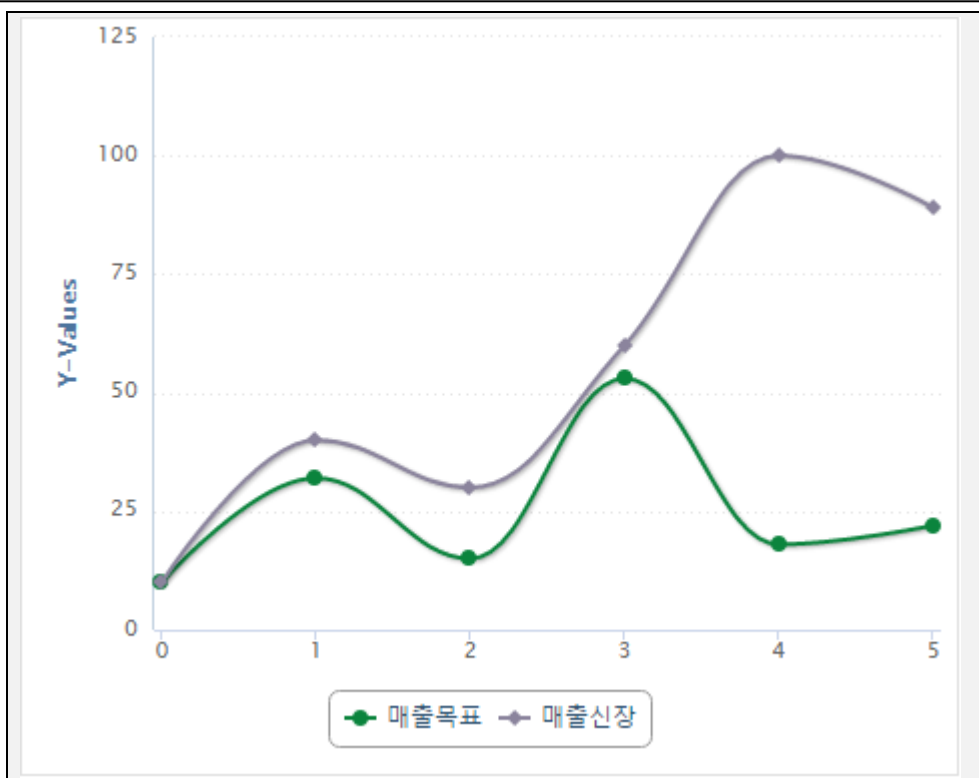
plotOptions의 spline 속성에서 설정한다. line 차트와 속성이 같다. (step 속성 제외)

- spline 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "spline"
  },
  series: [{
    name : "매출목표",
    data : [10,32,15,53,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [10,40,30,60,100,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.7.4 산점(scatter)차트 설정

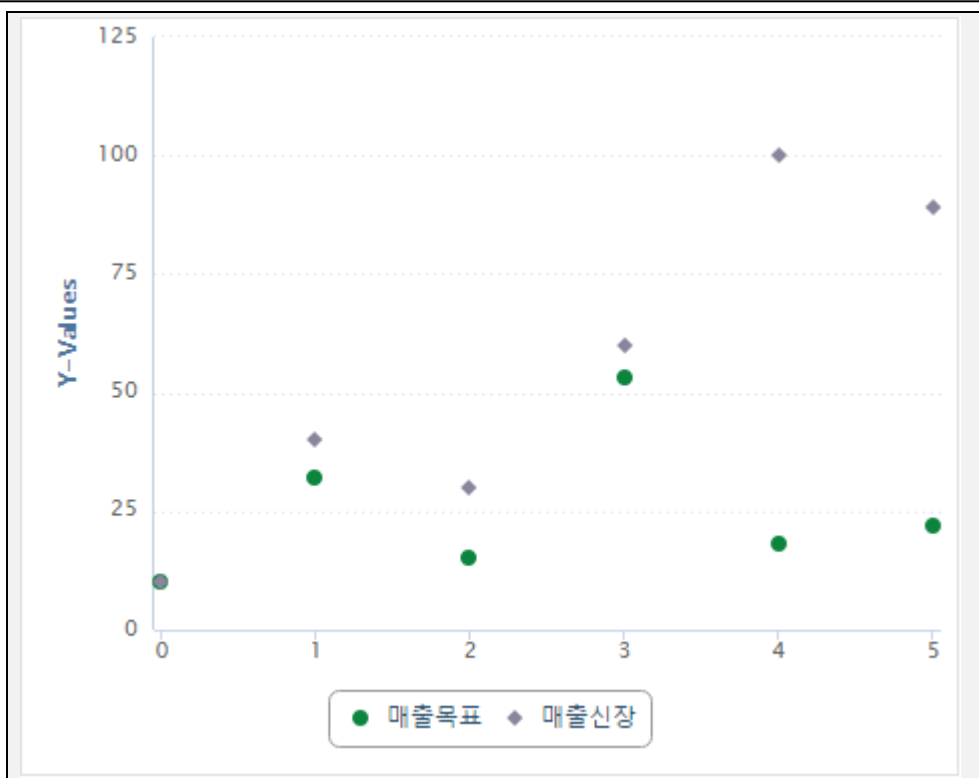
plotOptions의 scatter 속성에서 설정한다.

- scatter 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "scatter"
  },
  series: [{
    name : "매출목표",
    data : [10,32,15,53,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [10,40,30,60,100,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.7.5 막대(column)차트 설정

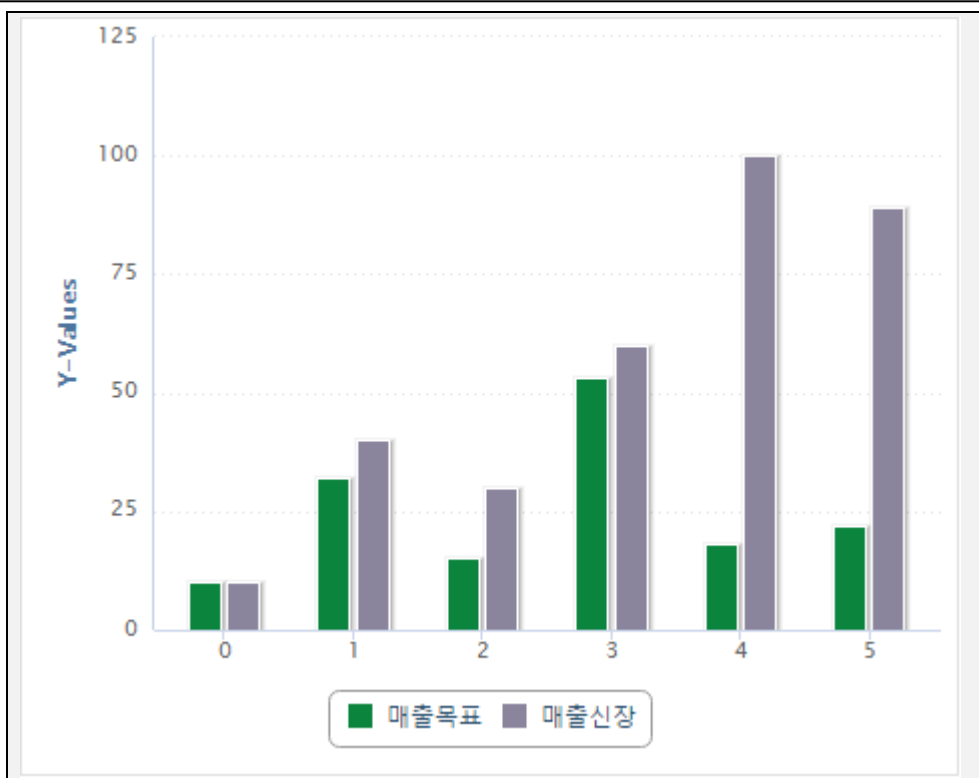
plotOptions의 column 속성에서 설정한다.

- column 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "column"
  },
  series: [{
    name : "매출목표",
    data : [10,32,15,53,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [10,40,30,60,100,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.7.6 가로막대(bar)차트 설정

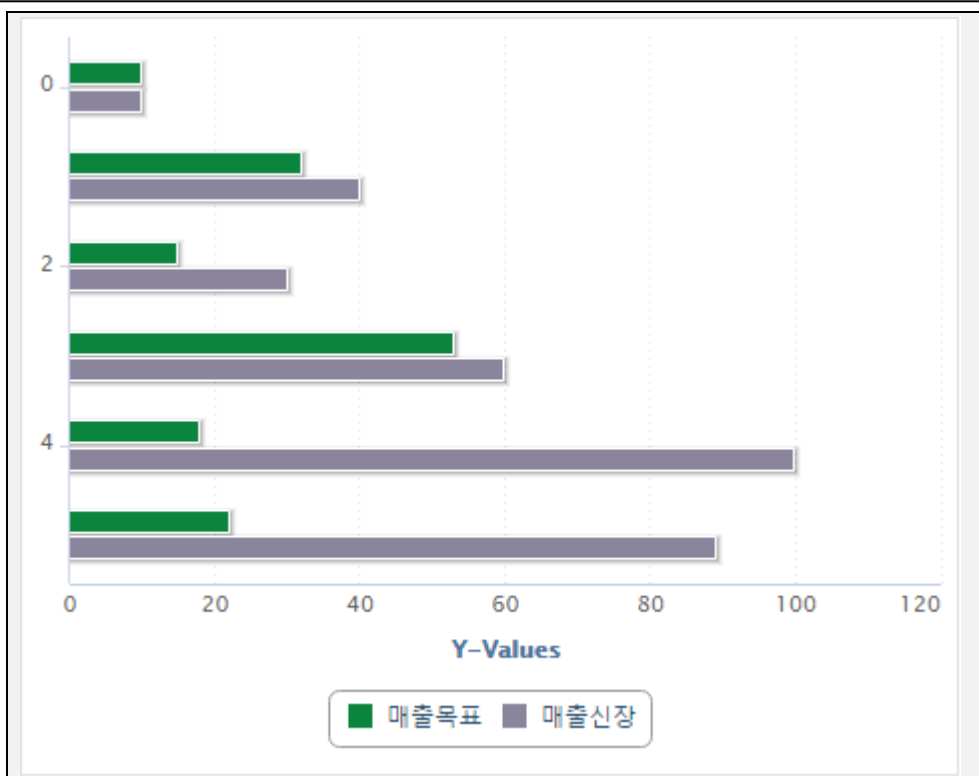
plotOptions의 bar 속성에서 설정한다.

- bar 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "bar"
  },
  series: [{
    name : "매출목표",
    data : [10,32,15,53,18,22],
    color : "#0B843D"
  },{
    name : "매출신장",
    data : [10,40,30,60,100,89],
    color : "#8B849D"
  }]
}, {
  append: true,
  redraw: true
});
```

4.3.7.7 파이(pie) 차트 설정

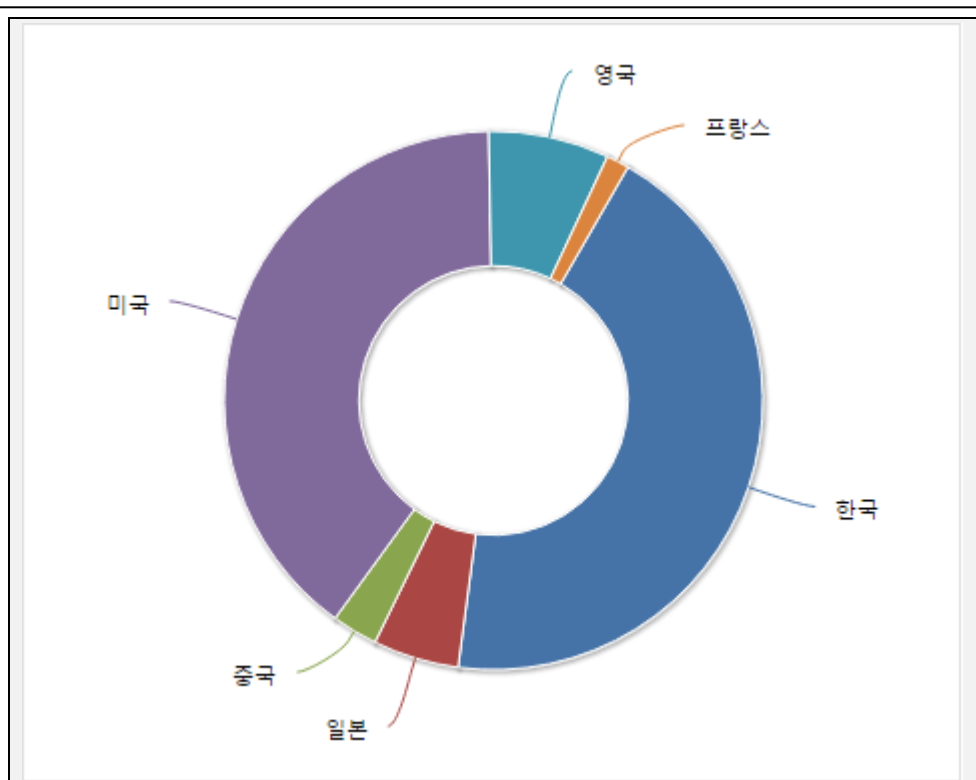
plotOptions의 pie 속성에서 설정한다.

- pie 속성

[10.3. 참조](#)

- **Example**

```
//차트속성
myChart.setOptions({
  chart:{
    type: "pie"
  },
  plotOptions: {
    pie: {
      innerSize: "50%",
      startAngle: 30
    }
  },
  series: [{
    data: ["한국", 636],
      ["일본", 75],
      ["중국", 40],
      ["미국", 577],
      ["영국", 105],
      ["프랑스", 20]]
  }]
}, {
  append: true,
  redraw: true
});
```



4.3.7.8 박스플롯(boxplot) 차트 설정

plotOptions의 boxplot 속성에서 설정한다.

boxplot의 경우 다른 차트와 달리 포인트의 데이터가 5개이다.

예) { low : 100, q1 : 200, median : 300, q3 : 400, high : 500 }

➤ plotbox 속성

[10.3. 참조](#)

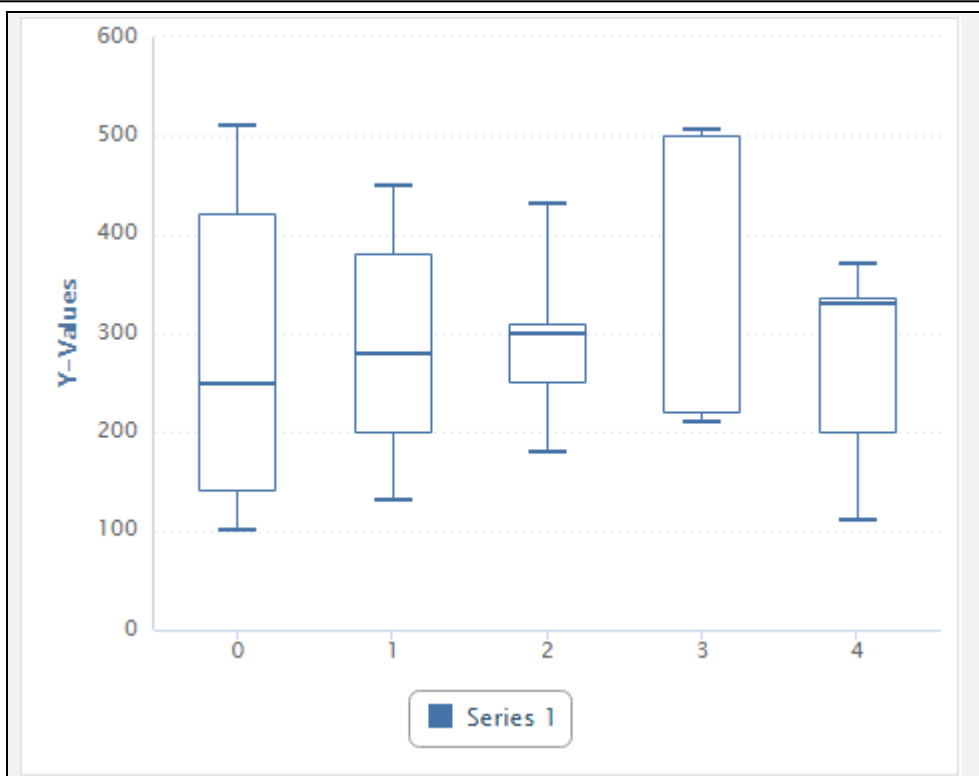
➤ data 속성

속성	타입	설명
low	number	최하위값
q1	number	중간박스의 하위값
median	number	중간값
q3	number	중간박스의 상위값
high	number	최상위값

➤ **Example**

```
// boxplot 차트 series 설정
myChart.setOptions({
  chart:{
    type: "boxplot"
  },
  series: [{
    data: [
      {low:100, q1:140, median:250, q3:420, high: 510},
      {low:130, q1:200, median:280, q3:380, high: 450},
      {low:180, q1:250, median:300, q3:310, high: 430},
      {low:210, q1:220, median:null,q3:500, high: 505},
      {low:110, q1:200, median:330, q3:335, high: 370}
    ]
  }]
}, {
```

```
        append: true,  
        redraw: true  
    });  
  
    myChart.setOptions({  
        chart:{  
            type: "boxplot"  
        },  
        series: [{  
            data: [[100,140,250,420,510],  
                [130,200,280,380,450],  
                [180,250,300,310,430],  
                [210,220,null,500,505],  
                [110,200,330,335,370]]  
        }]  
    }, {  
        append: true,  
        redraw: true  
    });
```



4.3.7.9 방사형(polar) 차트 설정

polar 차트는 기본 차트와는 달리, 차트 기본 속성 설정 부분에서 polar 속성을 true값으로 설정하여야 한다.

line, spline, areas, column의 차트 형태를 모두 표현할수 있다.

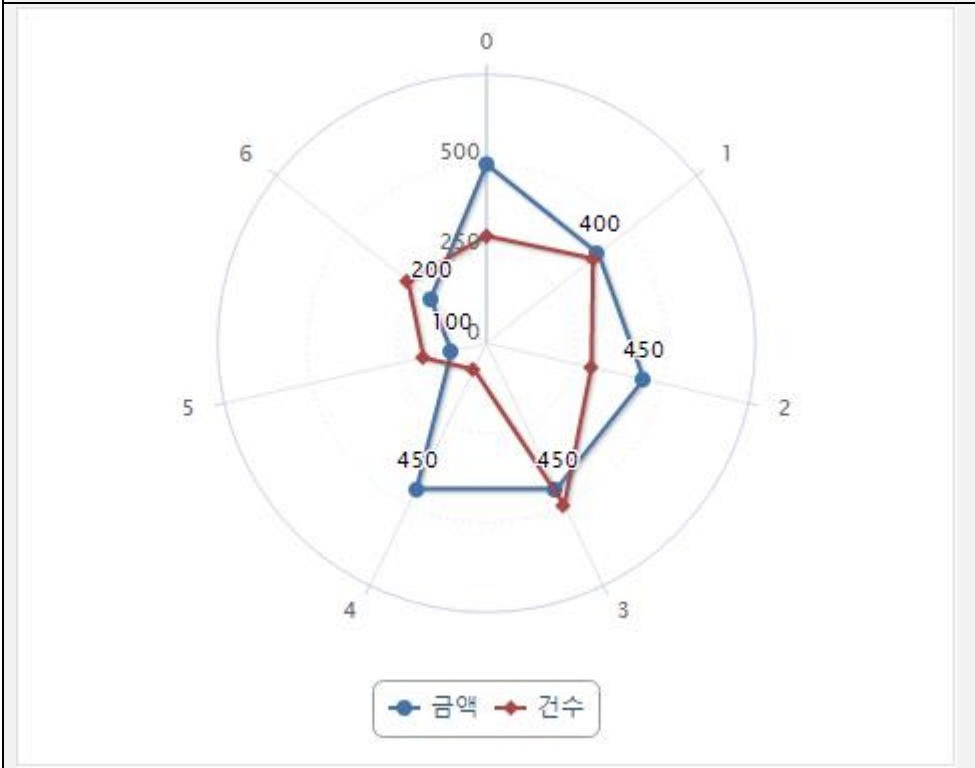
- polar 속성

[10.3. 참조](#)

- **Example**

```
//차트 polar 설정
myChart.setOptions({
  chart:{
    polar: true
  },
  yAxis: {
    title: {
      text:""
    }
  },
  series: [{
    type : "column",
    name : "금액",
    data : [500, 400, 450, 450, 450, 100, 200],
    dataLabels : {
      enabled : true
    }
  }, {
    type : "line",
    name : "건수",
    data : [300, 380, 300, 500, 80, 180, 280]
  }
], {
```

```
append: true,  
redraw: true  
});
```



4.3.7.10 워드클라우드(wordcloud) 차트 설정

wordcloud 차트는 기본 차트와는 달리, 차트 생성할 때 opt.plugin값을 "d3"로 설정해야한다. 또 wordcloud는 다른 차트와 달리 다른 설정을 설정할 수 있다.

- wordcloud 속성

[10.3. 참조](#)

- Example

```
//wordcloud 차트 생성
createIBChart(chart,"myChart", {
    width: "500px",
    height: "500px",
    plugin: "d3"
});

//wordcloud 차트 설정
myChart.setOptions({
    chart: {
        type: "wordcloud"
    },
    plotOptions: {
        wordcloud: {
            rotation: [0,90],
            orientation: 1,
            colorIterate: true,
            colorSort: "none",
            scaleType: "none",
            placeType: "linear",
            spiralType: "rectangular",
            valueRange: [15, 60],
            style: {
                fontFamily: "맑은 고딕",
```

```

        fontWeight: "bold",
        fontPadding: 0.2
      }
    }
  }, {
    append: false
  });

//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 60
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 30
      }]
    }
  ]
});

```

```
        }  
    }  
    }, {  
        append: true  
    });  
}
```

IB Sheet 7
IBLeaders
아이비리더스
IBChart(H) 7

4.4 데이터 조회

➤ Syntax

Syntax	<code>{ibchart}.loadSearchData(prop, [opt])</code>
--------	--

Parameter	Type	필수여부	Remark
prop	object	필수	조회된 데이터
opt.append	boolean	선택	조회시 append 여부(default:false)

```
var data = {
  ibchart: {
    backgroundColor: "white",
    borderWidth: "1",
    borderColor: "#0000FF",
    zoomType: "x",
    title: "타이틀",
    subtitle: "부제목",
    data: [{
      axisLabel: "1/4분기",
      series: [{
        seriesType: "line",
        seriesName: "중국",
        seriesColor: "blue",
        pointName: "하나",
        pointColor: "yellow",
        value: 220
      }, {
        seriesType: "column",
        seriesName: "호주",
        seriesColor: "#DDDDFF",
        pointName: "열하나",
```

```
        pointColor: "#C6E2FF",
        value: 120
    }, {
        seriesType: "spline",
        seriesName: "KOREA",
        seriesColor: "green",
        sointName: "스물하나",
        sointColor: "#EECCFF",
        value: 55
    }
  ], {
    axisLabel: "2/4분기",
    series: [280, 256, 222]
  }
];

myChart.loadSearchData(data, {
  append: true
});
```



4.5 속성 변경사항 적용하기

IBChart는 차트의 옵션이 변경된 사항이 있을 경우 항상 draw() 함수를 호출하거나 setOptions() 함수에서 redraw 옵션을 true를 설정해서 화면에 반영해줘야 한다.

➤ Example

```
function doAction(sAction) {  
    switch (sAction) {  
        case ":Line":  
            myChart.setOptions({  
                chart:{  
                    type:"line" //line 차트  
                }  
            }, {  
                append: true  
            });  
            break;  
        case "Column":  
            myChart.setOptions({  
                chart:{  
                    type:" column " // column 차트  
                }  
            }, {  
                append: true  
            });  
            break;  
    }  
  
    myChart.draw(); //속성 재적용  
}
```

➤ Example

```
function doAction(sAction) {  
    switch (sAction) {  
        case ":Line":  
            myChart.setOptions({  
                chart:{  
                    type:"line" //line 차트  
                }  
            }, {  
                append: true,  
                redraw: true //속성 재적용  
            });  
            break;  
        case "Column":  
            myChart.setOptions({  
                chart:{  
                    type:" column " // column 차트  
                }  
            }, {  
                append: true,  
                redraw: true //속성 재적용  
            });  
            break;  
    }  
}
```


4.6 데이터 XML 구현하기

조회 함수를 호출하면 인자로 넘긴 URL로 페이지에 연결한다. XML 형태로 구성된 페이지의 내용을 읽어 들여 데이터를 차트로 표현한다.

조회 데이터를 담고 있는 페이지는 다음과 같은 구조의 XML로 구성된다.

설명	XML 구성
XML 기본 태그 정의	<?xml version='1.0'?>
대분류 시작	<IBCHART>
ETC 데이터 시작	<ETC-DATA>
ETC 데이터	<ETC KEY="key_name">key_value</ETC>
ETC 데이터 종료	</ETC-DATA>
데이터 시작	<DATA>
시리즈별 첫번째 데이터 집합	<POINTSET>
1시리즈에 1포인트	<SERIES>100</SERIES>
2시리즈에 1포인트	<SERIES>200</SERIES>
첫번째 데이터 집합 종료	</POINTSET>
시리즈별 두번째 데이터 집합	<POINTSET>
1시리즈에 2포인트	<SERIES>150</SERIES>
2시리즈에 2포인트	<SERIES>300</SERIES>
두번째 데이터 집합 종료	</POINTSET>
데이터 종료	</DATA>
대분류 종료	</IBCHART>

4.7 데이터 JSON 구현하기

조회 함수를 호출하면 인자로 넘긴 URL로 페이지에 연결한다. JSON 형태로 구성된 페이지의 내용을 읽어 들여 데이터를 차트로 표현한다.

조회 데이터를 담고 있는 페이지는 다음과 같은 구조의 JSON 으로 구성된다.

반드시 속성명은 대문자로 표시해야 하며, 속성은 " " (이중따옴표) 로 감싸거나 생략해도 상관없다. ETC-DATA 가 아닌 etcData 임을 유의하도록 한다.

설명	XML 구성
JSON 차트 객체 시작	{ ibchart: {
차트 일반 속성들	background-color: "#CCFFCC",
	borderWidth: "1",
	borderColor: "#0000FF",
	zoomType: "x",
	title: "타이틀",
	subtitle: "부제목",
ETC 데이터의 시작	etcData: [
속성 : 값	{key:"sname",value:"홍길동"},
속성 : 값 (마지막 데이터에서는 , 가 없음을 유의한다.)	{key:"age",value:"20"}
ETC 데이터의 끝],
데이터의 시작	{ data: [
시리즈별 첫번째 데이터 집합	{ axisLabel: "1/4분기", series: [
1시리즈에 1포인트	{ seriesType: "line", seriesName: "중국", seriesColor: "blue", pointName: "하나",

	pointColor: "yellow", value: 77 },
2시리즈에 1포인트	{ seriesType: "area", seriesName: "호주", seriesColor: "red", pointName: "열하나", pointColor: "black", value: 120 },
3시리즈에 1포인트 (마지막 데이터에서는 , 가 없음을 유의한다.)	{ seriesType: "spline", seriesName: "KOREA", seriesColor: "green", pointName: "스물하나", pointColor: "#EECCFF", value: 55 }
첫번째 데이터 집합 종료]],
시리즈별 두번째 데이터 집합	{ axisLabel: "2/4분기", series: [
1시리즈에 2포인트	{ value:77},
2시리즈에 2포인트	256,
3시리즈에 2포인트	222,
두번째 데이터 집합 종료 (마지막 데이터에서는 , 가 없음을 유의한다.)]]
데이터의 종료]]
JSON 차트 개체 종료	}}

Chapter 5. 개발 실습

5. 개발 실습

5.1 IBChart 객체 생성 및 기본 속성 설정하기

ibchartinfo.js 내의 createIBChart 함수를 사용하여 IBChart 객체를 생성하고 body의 onLoad 이벤트에서 생성된 차트 객체에 대한 기본 속성을 설정한다.

➤ 객체 생성하기

```
<body onload="LoadPage()" ">  
  
  <div id="chart-div"></div>  
  
</body>
```

➤ 기본 속성 설정하기

```
<script type="text/javascript" src="/Chart/ibchart.js"></script>  
<script type="text/javascript" src="/Chart/ibchartinfo.js"></script>  
<script type="text/javascript">  
function LoadPage() {  
    createIBChart(chart-div, "myChart",{  
        width: "500px",  
        height: "300px"  
    });  
    myChart.draw();  
}  
</script>
```

5.2 조회된 데이터 사용

조회된 데이터를 차트에 동적으로 적용한다. 이를 위한 단계는 아래와 같다.

➤ **조회 JSON(or XML) 생성하기(필수)**

JSON(or XML) 데이터를 구성한다. ([Chapter 6](#). 참조)

➤ **조회 완료 이벤트 처리하기(선택)**

조회가 완료된 후에 어떤 처리가 필요한 경우 완료 이벤트에 처리를 구현한다.

```
myChart.setEventListener("searchEnd", function(event) {  
    console.log("조회가 완료되었습니다.");  
});
```

➤ **조회된 JSON(or XML) 데이터 차트에 적용(필수)**

loadSearchData() 함수를 이용해 조회된 데이터를 차트에 적용시킨다. ([Chapter 4.4](#). 참조)

Chapter 6. 데이터 구조

6. 데이터 구조

IBChart에서 사용되는 데이터 구조를 설명한다.

IBChart는 XML과 JSON 두가지 형식의 자료구조를 사용한다.

각 자료구조의 세부적인 사양은 다음과 같다.

6.1 데이터 XML

조회 함수를 통해 읽어들이는 XML 전체 구조는 다음과 같다.

```
<?xml version='1.0'?>
<![CDATA[
<IBCHART TYPE="" BACKGROUNDCOLOR='#CCFFCC' BORDERWIDTH='1'
BORDERCOLOR='#0000FF' ZOOMTYPE='x' TITLE='타이틀' SUBTITLE='부제목' >
  <ETC-DATA>
    <ETC key="sname">홍길동</ETC>
    <ETC key="age">20</ETC>
  </ETC-DATA>
  <DATA>
    <POINTSET AXISLABEL="1/4분기">
      <SERIES SERIESTYPE="line" SERIESNAME="중국"
SERIESCOLOR="blue" PointLabel="하나" POINTCOLOR="yellow">50</SERIES>
      <SERIES SERIESTYPE="area" SERIESNAME="호주"
SERIESCOLOR="red" PointLabel="열하나" POINTCOLOR="black">48</SERIES>
      <SERIES SERIESTYPE="spline" SERIESNAME="KOREA"
SERIESCOLOR="green" POINTNAME="스물하나"
POINTCOLOR="#EECCFF">60</SERIES>
    </POINTSET>
    <POINTSET AXISLABEL="2/4분기">
      <SERIES POINTNAME="둘" POINTCOLOR="">22</SERIES>
      <SERIES POINTNAME="열둘" POINTCOLOR="">13</SERIES>
      <SERIES POINTNAME="스물둘" POINTCOLOR="">16</SERIES>
    </POINTSET>
  </DATA>
</IBCHART>
]]>
```

위의 조회 XML에서 사용하는 XML요소는 다음과 같이 필수 항목과 선택 항목으로 나누어 진다.

요소명	필수여부	설 명
IBCHART	필수	조회 데이터의 최상위 요소
DATA	필수	조회 데이터의 시작 요소
POINTSET	필수	포인트 집합
SERIES	필수	시리즈 포인트
ETC-DATA	선택	기타 데이터 설정 요소

6.2 데이터 JSON

조회 함수를 통해 읽어들이는 JSON 전체 구조는 다음과 같다.

대소문자 구분을 하며, ETC-DATA가 아닌 etcData 임을 유의한다.

series 속성 안에서는 값만 나열할 수 있고, 값에 대한 속성을 명시할 때에는 value 속성을 사용한다.

```
{
  ibchart: {
    backgroundColor: "white",
    borderWidth: "1",
    borderColor: "#0000FF",
    zoomType: "x",
    title: "타이틀",
    subtitle: "부제목",
    etcData: [
      {key:"sname",value:"홍길동"},
      {key:"age",value:"20"}
    ],
    data: [{
      axisLabel: "1/4분기",
      series: [{
        seriesType: "line",
        seriesName: "중국",
        seriesColor: "blue",
        pointName: "하나",
        pointColor: "yellow",
        value: 220
      }, {
        seriesType: "column",
        seriesName: "호주",
        seriesColor: "#DDDDFF",
```

```

    pointName: "열하나",
    pointColor: "#C6E2FF",
    value: 120
  }, {
    seriesType: "spline",
    seriesName: "KOREA",
    seriesColor: "green",
    sointName: "스물하나",
    sointColor: "#EECCFF",
    Value: 55
  }]
}, {
  axisLabel: "2/4분기",
  series: [280, 256, 222]
}]
}
}

```

위의 조회 JSON에서 사용하는 JSON요소는 다음과 같이 필수 항목과 선택 항목으로 나누어 진다.

요소명	필수여부	설 명
ibchart	필수	조회 데이터의 최상위 요소
data	필수	조회 데이터의 시작 요소
series	필수	시리즈 포인트
etcData	선택	기타 데이터 설정 요소

Chapter 7. IBChart 상수

7. IBChart 상수

7.1 IBLANG

사용자가 언어 설정할 수 있도록 언어 설정 상수를 제공한다. 한국어(언어코드: "kr"), 영어(언어코드: "en")는 기본으로 제공하고 다른 언어는 제공되는 언어 설정을 기준으로 번역하여 추가해서 사용할 수 있다.

속성	타입	설명
IBLANG.언어코드.chart	object	차트 표현에 사용되는 언어 설정
IBLANG.언어코드.alert	object	데이터 오류나 사용자에게 경고 표시를 위한 언어 설정
IBLANG.언어코드.text	object	chart, alert 이외의 자유롭게 사용하는 문자열을 위한 언어 설정

7.1.1 chart

차트 표현에 사용되는 언어 설정이다. 속성명과 데이터 타입을 유지한 수정은 자유롭다.

속성	타입	설명
thousandsSep	string	천단위(1000) 구분자
contextButtonTitle	string	보조메뉴
decimalPoint	string	정수 구분자
downloadJPEG	string	JPEG 다운로드
downloadPDF	string	PDF 다운로드
downloadPNG	string	PNG 다운로드
downloadSVG	string	SVG 다운로드
drillUpText	string	한 단계 위로
invalidDate	string	인식불가 데이터
loading	string	조회중
noData	string	데이터 없음
numericSymbols	string	숫자 축약
printChart	string	프린트
resetZoom	string	줌 아웃
months	string	월
shortMonths	string	짧은 월
weekdays	string	주
shortWeekdays	string	짧은 주

7.1.2 alert

데이터 오류나 사용자에게 경고를 알리는 언어 설정이다. 속성명과 데이터 타입, \${숫자}인자를 유지한 수정은 자유롭다. 단, 속성 하나라도 누락되면 오류가 발생할 수 있다.

속성	타입	설명
createIBChartId	string	차트 아이디 오류
chartProperties	string	차트 속성 오류
chartNoData	string	차트 데이터 없음
chartData	string	차트데이터 오류
plugin	string	플러그인 오류
pluginLoading	string	플러그인 로드 오류
series	string	시리즈 오류
noSeries	string	시리즈 인덱스 오류
eventListenerType	string	이벤트 종류 오류
eventListenerFunc	string	이벤트 핸들러 오류
chartNoIndex	string	빈 차트 시리즈 인덱스 오류
chartIndex	string	차트 시리즈 인덱스 오류
variableRequired	string	필수값 오류
variableInteger	string	정수 오류
variableFloat	string	실수 오류
variableString	string	문자열 오류
variableLessThen	string	A가 B보다 클때 오류
variableGreaterThen	string	A가 B보다 작을때
variableEpuat	string	A와 B가 같지 않을때
noMessage	string	빈 메세지 오류

7.1.3 text

chart, alert 이외의 자유롭게 사용하는 문자열을 위한 언어 설정이다. 자유롭게 수정이 가능하며 getAlertText() 함수를 이용하면 \${숫자}인자를 쉽게 적용할 수 있다.

속성	타입	설명
legend	string	sheet2chart에서 사용될 legendTitle 기본값
productName	string	getAlertText() 함수의 알림 머리말
referenceSite	string	getAlertText() 함수의 알림 꼬리말

Chapter 8. IBChart Events

8. IBChart Events

8.1 Event 사용

IBChart 에서는 다양한 비즈니스 로직 구성을 위해 이벤트를 제공 한다.
이벤트 추가는 `setEventListener` 와 `setEvents` method 를 통해서 추가 할 수 있다.

8.2 Event List

IBChart가 제공하는 이벤트는 다음과 같다.

addSeries

➤ 기능

시리즈 추가 완료 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(add)	이벤트 타입

➤ Example

```
myChart.setEvents({
  addSeries: function(event) {
    console.log("addSeries");
  }
});

myChart.setEventListener("addSeries", function(event){
  console.log("addSeries");
});

myChart.setOptions({
  events : {
    addSeries: function(event) {
      console.log("addSeries");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

afterPrint

➤ 기능

doPrint() 호출 시 프린트 직후에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(print)	이벤트 타입

➤ Example

```
myChart.setEvents({
  afterPrint: function(event) {
    console.log("afterPrint");
  }
});

myChart.setEventListener("afterPrint", function(event){
  console.log("afterPrint");
});

myChart.setOptions({
  events : {
    afterPrint: function(event) {
      console.log("afterPrint");
    }
  }
});
```

➤ 제공 버전

7.3.0.0	
----------------	--

beforePrint

➤ 기능

doPrint() 호출 시 프린트 직전에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(print)	이벤트 타입

➤ Example

```
myChart.setEvents({
  beforePrint: function(event) {
    console.log("beforePrint");
  }
});

myChart.setEventListener("beforePrint", function(event){
  console.log("beforePrint");
});

myChart.setOptions({
  events : {
    beforePrint: function(event) {
      console.log("beforePrint");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

click

➤ 기능

차트 영역을 클릭하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(click)	이벤트 타입
event.x	number	N	클릭 지점의 x좌표 (컨테이너 기준)
event.y	number	N	클릭 지점의 y좌표 (컨테이너 기준)

➤ Example

```
myChart.setEvents({
  addSeries: function(event) {
    console.log("click " + event.x + " " + event.y);
  }
});

myChart.setEventListener("click", function(event){
  console.log("click " + event.x + " " + event.y);
});

myChart.setOptions({
  events : {
    click: function(event) {
      console.log("click " + event.x + " " + event.y);
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

drilldown

➤ 기능

드릴 다운 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(drilldown)	이벤트 타입
event.x	number	N	클릭 지점의 x좌표 (컨테이너 기준)
event.y	number	N	클릭 지점의 y좌표 (컨테이너 기준)

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel 값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  drilldown: function(event) {
    console.log("drilldown: " + this.category);
  }
});

myChart.addEventListener("drilldown", function(event) {
  console.log("drilldown: " + this.category);
});

myChart.setOptions({
  events : {
```

```
drilldown: function(event) {  
    console.log("drilldown: " + this.category);  
}  
}  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

drillup

➤ 기능

드릴 업 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(drillup)	이벤트 타입

➤ Example

```
myChart.setEvents({
  drillup: function(event) {
    console.log("drillup");
  }
});

myChart.setEventListener("drillup", function(event){
  console.log("drillup");
});

myChart.setOptions({
  events : {
    drillup: function(event) {
      console.log("drillup");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

drillupAll

➤ 기능

최상단의 드릴 업 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(drillup)	이벤트 타입

➤ Example

```
myChart.setEvents({
  drillAll: function(event) {
    console.log("drillAll");
  }
});

myChart.addEventListener("drillAll", function(event) {
  console.log("drillAll");
});

myChart.setOptions({
  events : {
    drillAll: function(event) {
      console.log("drillAll");
    }
  }
});
```

➤ 제공 버전

7.3.0.0	
----------------	--

pointClick

➤ 기능

시리즈의 포인트 클릭 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(click)	이벤트 타입
event.x	number	N	클릭 지점의 X좌표 (컨테이너 기준)
event.y	number	N	클릭 지점의 Y좌표 (컨테이너 기준)
event.point	object	N	포인트 정보

■ event.point Info

name	type	required	description
x	number	N(x값)	클릭 지점 X좌표(x축 기준)
y	number	N(y값)	클릭 지점 Y좌표(y축 기준)
name	string	N(axislabel값)	클릭 지점 이름

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  pointClick: function(event) {
    console.log("pointClick: " + this.category);
  }
});
```

```

    }
  });

  myChart.setEventListener("pointClick", function(event) {
    console.log("pointClick: " + this.category);
  });

  myChart.setOptions({
    events : {
      pointClick: function(event) {
        console.log("pointClick: " + this.category);
      }
    }
  })

```

➤ **제공 버전**

7.3.0.0	
----------------	--

pointMouseOut

➤ 기능

시리즈의 포인트에서 마우스 커서가 out 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(mouseout)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel 값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  pointMouseOut: function(event) {
    console.log("pointMouseOut: " + this.category);
  }
});

myChart.addEventListener("pointMouseOut", function(event){
  console.log("pointMouseOut: " + this.category);
});

myChart.setOptions({
  events : {
    pointMouseOut: function(event) {
      console.log("pointMouseOut: " + this.category);
    }
  }
});
```

```
}  
  
}  
  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

pointMouseOver

➤ 기능

시리즈의 포인트에서 마우스 커서가 over 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(mouseover)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel 값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  pointMouseOver: function(event) {
    console.log("pointMouseOver: " + this.category);
  }
});

myChart.setEventListener("pointMouseOver", function(event){
  console.log("pointMouseOver: " + this.category);
});

myChart.setOptions({
  events : {
    pointMouseOver: function(event) {
      console.log("pointMouseOver: " + this.category);
    }
  }
});
```

```
}  
  
}  
  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

pointSelect

➤ 기능

시리즈의 포인트를 선택하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(select)	이벤트 타입

■ event.point Info

name	type	required	description
x	number	N(x값)	클릭 지점 X좌표(x축 기준)
y	number	N(y값)	클릭 지점 Y좌표(y축 기준)
name	string	N(axislabel값)	클릭 지점 이름

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  pointSelect: function(event) {
    console.log("pointSelect: " + this.category);
  }
});
```

```
myChart.addEventListener("pointSelect", function(event){  
    console.log("pointSelect: " + this.category);  
});
```

```
myChart.setOptions({  
    events : {  
        pointSelect: function(event) {  
            console.log("pointSelect: " + this.category);  
        }  
    }  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

pointUnselect

➤ 기능

시리즈의 포인트를 선택 해제하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(unselect)	이벤트 타입

■ event.point Info

name	type	required	description
x	number	N(x값)	클릭 지점 X좌표(x축 기준)
y	number	N(y값)	클릭 지점 Y좌표(y축 기준)
name	string	N(axislabel값)	클릭 지점 이름

➤ Context Info

name	type	required	description
this	object	N	포인트 속성
this.category	string number	N(axislabel값)	지점의 카테고리(axislabel)
this.x	number	N(x값)	지점의 x값
this.y	number	N(y값)	지점의 y값

➤ Example

```
myChart.setEvents({
  pointUnselect: function(event) {
    console.log("pointUnselect: " + this.category);
  }
});
```

```
myChart.setEventListener("pointUnselect", function(event) {  
    console.log("pointUnselect: " + this.category);  
});
```

```
myChart.setOptions({  
    events : {  
        pointUnselect: function(event) {  
            console.log("pointUnselect: " + this.category);  
        }  
    }  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

pointUpdate

➤ 기능

시리즈의 포인트가 갱신되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(update)	이벤트 타입

➤ Example

```
myChart.setEvents({
  pointUpdate: function(event) {
    console.log("pointUpdate");
  }
});

myChart.setEventListener("pointUpdate", function(event){
  console.log("pointUpdate");
});

myChart.setOptions({
  events : {
    pointUpdate: function(event) {
      console.log("pointUpdate");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

redraw

➤ 기능

차트가 redraw 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(redraw)	이벤트 타입

➤ Example

```
myChart.setEvents({
  redraw: function(event) {
    console.log("redraw");
  }
});

myChart.addEventListener("redraw", function(event){
  console.log("redraw");
});

myChart.setOptions({
  events : {
    redraw: function(event) {
      console.log("redraw");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

searchEnd

➤ 기능

차트의 데이터가 조회 완료되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(search)	이벤트 타입
event.code	string	N	에러 코드
event.msg	string	N	메시지

➤ Example

```
myChart.setEvents({
  searchEnd: function(event) {
    console.log("searchEnd");
  }
});

myChart.addEventListener("searchEnd", function(event) {
  console.log("searchEnd");
});

myChart.setOptions({
  events : {
    searchEnd: function(event) {
      console.log("searchEnd");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

selection

➤ 기능

차트에서 마우스 드래그로 영역을 선택하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N()	이벤트 타입

➤ Example

```
myChart.setEvents({
  selection: function(event) {
    console.log("selection");
  }
});

myChart.addEventListener("selection", function(event){
  console.log("selection");
});

myChart.setOptions({
  events : {
    selection: function(event) {
      console.log("selection");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

seriesAfterAnimate

➤ 기능

시리즈의 애니메이션 이후 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N()	이벤트 타입

➤ Example

```
myChart.setEvents({
  seriesAfterAnimate: function(event) {
    console.log("seriesAfterAnimate");
  }
});

myChart.setEventListener("seriesAfterAnimate", function(event){
  console.log("seriesAfterAnimate");
});

myChart.setOptions({
  events : {
    seriesAfterAnimate: function(event) {
      console.log("seriesAfterAnimate");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
----------------	--

seriesCheckBoxClick

➤ 기능

시리즈의 체크박스를 클릭 하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(click)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesCheckBoxClick: function(event) {
    console.log("seriesCheckBoxClick");
  }
});

myChart.setEventListener("seriesCheckBoxClick", function(event){
  console.log("seriesCheckBoxClick");
});
```

```
myChart.setOptions({  
  events : {  
    seriesCheckBoxClick: function(event) {  
      console.log("seriesCheckBoxClick");  
    }  
  }  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesClick

➤ 기능

시리즈를 클릭 하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(click)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesClick: function(event) {
    console.log("seriesClick");
  }
});

myChart.setEventListener("seriesClick", function(event){
  console.log("seriesClick");
});
```

```
myChart.setOptions({
  events : {
    seriesClick: function(event) {
      console.log("seriesClick");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesHide

➤ 기능

시리즈가 숨김 처리 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(hide)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesHide: function(event) {
    console.log("seriesHide");
  }
});

myChart.setEventListener("seriesHide", function(event) {
  console.log("seriesHide");
});
```

```
myChart.setOptions({
  events : {
    seriesHide: function(event) {
      console.log("seriesHide");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesLegendItemClick

➤ 기능

시리즈의 범례를 클릭하는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(click)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesLegendItemClick: function(event) {
    console.log("seriesLegendItemClick");
  }
});

myChart.setEventListener("seriesLegendItemClick",
  function(event) {
    console.log("seriesLegendItemClick");
  });
```

```
myChart.setOptions({
  events : {
    seriesLegendItemClick: function(event) {
      console.log("seriesLegendItemClick");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesMouseOut

➤ 기능

시리즈에서 마우스 커서가 out 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(mouseout)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesMouseOut: function(event) {
    console.log("seriesMouseOut");
  }
});

myChart.setEventListener("seriesMouseOut", function(event){
  console.log("seriesMouseOut");
});
```

```
myChart.setOptions({  
  events : {  
    seriesMouseOut: function(event) {  
      console.log("seriesMouseOut");  
    }  
  }  
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesMouseOver

➤ 기능

시리즈에서 마우스 커서가 over 되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(mouseover)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesMouseOver: function(event) {
    console.log("seriesMouseOver");
  }
});

myChart.setEventListener("seriesMouseOver", function(event){
  console.log("seriesMouseOver");
});
```

```
myChart.setOptions({
  events : {
    seriesMouseOver: function(event) {
      console.log("seriesMouseOver");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
---------	--

seriesShow

➤ 기능

시리즈에서 숨김에서 보임으로 변경되는 시점에 발생하는 이벤트

➤ Info

name	type	required	description
event	object	N	이벤트 인자 객체
event.type	string	N(show)	이벤트 타입

➤ Context Info

name	type	required	description
this	object	N	시리즈 속성
this.name	string	N	시리즈 명
this.index	number	N	시리즈 인덱스
this.data	array	N	시리즈의 데이터
this.xData	array	N	시리즈의 x축 데이터
this.yData	array	N	시리즈의 y축 데이터
this.dataMax	number	N	시리즈의 데이터에서 최대 값
this.dataMin	number	N	시리즈의 데이터에서 최소 값

➤ Example

```
myChart.setEvents({
  seriesShow: function(event) {
    console.log("seriesShow");
  }
});

myChart.setEventListener("seriesShow", function(event) {
  console.log("seriesShow");
});
```

```
myChart.setOptions({
  events : {
    seriesShow: function(event) {
      console.log("seriesShow");
    }
  }
})
```

➤ 제공 버전

7.3.0.0	
---------	--

Chapter 9. IBChart Methods

9. IBChart Methods

9.1 Chart Object

addSeries

➤ 기능

시리즈를 차트에 동적으로 추가 한다.

➤ Syntax

Syntax	<code>{ibchart}.addSeries(prop, [opt])</code>
---------------	---

➤ Info

returns	없음		
name	type	required	description
prop	object	Y	시리즈 속성 및 데이터
opt	object	N	옵션 속성
opt.redraw	boolean	N(true)	설정 후 다시 그릴지 여부

➤ Example

<pre>myChart.addSeries({ data: [10, 20, 30] // 시리즈의 데이터 설정 });</pre>
--

➤ 제공 버전

7.3.0.0	
----------------	--

doPrint

➤ 기능

차트 이미지를 인쇄한다. 인쇄 처리는 브라우저의 인쇄 기능을 따르며 이미지를 인쇄하기 위해서는 브라우저의 페이지 설정에서 배경색 및 이미지 인쇄 옵션을 설정하여야 한다.

➤ Syntax

Syntax	<code>{ibchart}.doPrint()</code>
---------------	----------------------------------

➤ Info

returns	없음		
name	type	required	description
없음			

➤ Example

```
// 차트 인쇄하기
myChart.doPrint();
```

➤ 제공 버전

7.3.0.0	
----------------	--

down2Image

➤ 기능

차트 데이터를 이미지로 다운로드한다.

type 속성 설정에 따라 다양한 이미지 형식의 파일로 다운로드할 수 있다. url 속성은 별도로 제공되는 서버모듈의 Down2Image.jsp 파일의 경로를 설정한다.

해당 경로가 올바르지 않으면 이미지를 다운로드 할 수 없으니 주의한다.

➤ Syntax

Syntax	<code>{ibchart}.down2Image(opt)</code>
---------------	--

➤ Info

returns	없음		
name	type	required	description
opt	object	Y	옵션 속성
opt.fileName	string	N(chart)	파일 명
opt.type	string	N(png)	다운로드 형식
opt.url	string	Y	처리할 서버 url 경로
opt.width	number	N	이미지 너비 (높이는 너비에 비례하여 자동 설정) (단위:px)
opt.sourceWidth	number	N	변환대상 차트의 너비 (단위:px)
opt.sourceHeight	number	N	변환대상 차트의 높이 (단위:px)

➤ Example

```
// 차트 이미지 다운로드
myChart.down2Image({
  fileName: "ChartImage",
  type: "jpg",
  width: 800,
  url: "../Chart/Down2Image.jsp"
});
```


➤ 제공 버전

7.3.0.0	
---------	--

draw

➤ 기능

차트의 최신 상태를 반영하기 위해 차트를 다시 그린다.

➤ Syntax

Syntax	<code>{ibchart}.draw([callback])</code>
--------	---

➤ Info

returns	없음		
name	type	required	description
callback	function	N	처리 후 호출할 callback 함수

➤ Example

```
// 차트 그리기
myChart.draw();

// 차트 렌더링 이후 callback 함수 호출
myChart.draw(function() {
    // TODO: 완료 후 처리할 비즈니스 로직 작성
});
```

➤ 제공 버전

7.3.0.0	
---------	--

getAlertText

➤ 기능

언어설정에 담긴 문자열을 반환한다.

➤ Syntax

Syntax	<code>{ibchart}.getAlertText()</code>
--------	---------------------------------------

➤ Info

returns	string, 문자열		
name	type	required	description
str	string	Y	IBLANG or 사용자 정의 문자열

➤ Example

```
// alert or console
alert(myChart.getAlertText("경고입니다!"));
// "경고입니다!"
alert(myChart.getAlertText("${0}는 ${1}이다.")
    .format("사과", "과일"));
// "사과는 과일이다."
alert(myChart.getAlertText(IBLANG["ko"].alert.chartData));
// "데이터가 잘못되었습니다!"
alert(myChart.getAlertText(IBLANG["ko"].alert.variableEqual)
    .format("front", "rear"));
// "front값은 rear값과 같아야합니다."
```

➤ 제공 버전

7.3.0.0	
---------	--

getData

➤ 기능

차트의 모든 옵션 객체를 반환 한다.

➤ Syntax

Syntax	<code>{ibchart}.getData()</code>
--------	----------------------------------

➤ Info

returns	array.<object>, 모든 데이터 정보		
name	type	required	description
			없음

➤ Example

```
// data object
var data = myChart.getData();
```

➤ 제공 버전

7.3.0.0	
----------------	--

getId

➤ 기능

차트의 id를 반환 한다.

➤ Syntax

Syntax	<code>{ibchart}.getId()</code>
--------	--------------------------------

➤ Info

returns	string, 차트 id		
name	type	required	description
없음			

➤ Example

```
// chart id
var id = myChart.getId();
```

➤ 제공 버전

7.3.0.0	
---------	--

getLang

➤ 기능

차트의 언어권을 반환한다.

➤ Syntax

Syntax	<code>{ibchart}.getLang()</code>
--------	----------------------------------

➤ Info

returns	string, 언어코드		
name	type	required	description
			없음

➤ Example

```
// chart locale
var chartLocale = myChart.getLang(); // "kr"
```

➤ 제공 버전

7.3.0.0	
----------------	--

getOptions

➤ 기능

차트의 모든 옵션 객체를 반환 한다.

➤ Syntax

Syntax	<code>{ibchart}.getOptions()</code>
--------	-------------------------------------

➤ Info

returns	object, 모든 옵션 정보		
name	type	required	description
			없음

➤ Example

```
// option object
var option = myChart.getOptions();
```

➤ 제공 버전

7.3.0.0	
----------------	--

getSeries

➤ 기능

차트의 모든 시리즈 객체를 반환 한다.

➤ Syntax

Syntax	<code>{ibchart}.getSeries()</code>
--------	------------------------------------

➤ Info

returns	array.<object>, 모든 시리즈 정보		
name	type	required	description
			없음

➤ Example

```
// series object
var series = myChart.getSeries();
```

➤ 제공 버전

7.3.0.0	
----------------	--

getSVGString

➤ 기능

차트의 SVG를 문자열로 반환한다.

➤ Syntax

Syntax	<code>{ibchart}.getSVGString()</code>
--------	---------------------------------------

➤ Info

returns	string, 차트의 SVG 문자열		
name	type	required	description
			없음

➤ Example

```
var svg = encodeURIComponent(myChart.getSVGString());

// 서버모듈에 따라 달라질 수 있음
var opt = {
    "ExtendParamMethod": "Post",    // 기본값이 "Get"이므로 "Post"로 변경해서 보내야 파라미터가 끊기지 않고 보내 질 수 있다. (Get 방식은 길이제한이 있음)
    "ExtendParam": "Svg="+svg      // ExtendParam에 추가 할 수 있는 차트 관련 파라미터 Svg, ImageWidth, ChartName, ChartPos(0:시트 뒤에 붙임, 1:시트앞에 붙임)
};

mySheet.Down2Excel(opt); // 시트와 차트를 하나의 엑셀 파일로 다운로드
```

➤ 제공 버전

7.3.1.0	
----------------	--

loadSearchData

➤ 기능

인자로 받은 데이터 객체 또는 문자열을 차트에 로드 한다.

➤ Syntax

Syntax	<code>{ibchart}.loadSearchData (data, [opt])</code>
--------	---

➤ Info

returns	없음		
name	type	required	description
data	object string	Y	로드할 데이터 객체 또는 문자열
opt	object	N	옵션 속성
opt.append	boolean	N(false)	데이터 병합 여부
opt.resetData	boolean	N(true)	데이터 병합 여부 (속성을 유지하면서 데이터 초기화 여부, ※ opt.append값과 opt.mergeData값이 false이면 데이터 초기화)
opt.mergeData	boolean	N(false)	데이터를 기존 데이터에 덮어씌울지 여부(opt.resetData값이 false일 때만 적용됨)
opt.deepCopy	boolean	N(false)	깊은 복사(데이터량에 따라 시간이 더 필요할 수 있음)

➤ **Example**

```
// 데이터를 차트에 로드 처리
myChart.loadSearchData({
  ibchart: {
    data: [{
      series: [1,10,100]
    }, {
      series: [2,20,200]
    }, {
      series: [3,30,300]
    }]
  }
});
```

➤ **제공 버전**

7.3.0.0	
7.3.1.4	resetData, mergeData 옵션 추가

parseText

➤ 기능

문장에서 단어를 추출하고 단어의 빈도수를 계산한다. (빈도수 기준 내림차순 정렬)

➤ Syntax

Syntax	<code>{ibchart}.parseText(text, ignore, max)</code>
--------	---

➤ Info

returns	object, 입력된 문장에서 추출된 단어와 단어의 빈도수를 내림차순으로 반환		
name	type	required	description
text	string	Y	분석할 문장
ignore	string array<string>	N	입력된 문장에서 단어의 빈도수 계산에서 제외할 단어 (RegExp Style String)
max	number	N	입력된 문장에서 상위 n번째까지의 기준

➤ Example

```
// 차트를 다시 그리기
myChart.setOptions({
  title: {
    text: "윤동주 <별 헤는 밤, 1946>"
  },
  series: [{
    name: "단어 사용 빈도",
    // 윤동주 <별 헤는 밤, 1946> 발췌
    data: myChart.parseText("계절이 지나가는 하늘에는 가을로 가득 차 있습니다. 나는 아무 걱정도 없이 가을 속의 별들을 다 헬 듯합니다. 가슴속에 하나
```

둘 새겨지는 별을 이제 다 못 헤는 것은 쉬이 아침이 오는 까닭이요, 내일 밤이 남은 까닭이요, 아직 나의 청춘이 다하지 않은 까닭입니다. 별 하나에 추억과 별 하나에 사랑과 별 하나에 쓸쓸함과 별 하나에 동경과 별 하나에 시와 별 하나에 어머니, 어머니, 어머니, 나는 별 하나에 아름다운 말 한마디씩 불러 봅니다. 소학교 때 책상을 같이 했던 아이들의 이름과, 패, 경, 옥, 이런 이국 소녀들의 이름과, 벌써 아기 어머니 된 계집애들의 이름과, 가난한 이웃 사람들의 이름과, 비둘기, 강아지, 토끼, 노새, 노루, '프랑시스 잠', '라이너 마리아 릴케' 이런 시인의 이름을 불러 봅니다. 이네들은 너무나 멀리 있습니다. 별이 아스라히 멀듯이. 어머니, 그리고 당신은 멀리 북간도에 계십니다. 나는 무엇인지 그리워 이 많은 별빛이 내린 언덕 위에 내 이름자를 써 보고 흙으로 덮어 버리었습니다. 맑은 밤을 새워 우는 별레는 부끄러운 이름을 슬퍼하는 까닭입니다. 그러나 겨울이 지나고 나의 별에도 봄이 오면 무덤 위에 파란 잔디가 피어나듯이 내 이름자 묻힌 언덕 위에도 자랑처럼 풀이 무성할 게외다.", ["이런", "다", "하나", "둘", "하나에"], 20)

```
    }}
  }, {
    append: true, // 이미 설정된 차트 속성은 유지
    resetData: true, // 이미 설정된 데이터는 지운다.
    redraw: true // 차트를 다시 그린다.
  });

// myChart.draw(); // 차트를 다시 그린다. (메소드 별도 호출)
```

➤ 제공 버전

7.3.1.0	
----------------	--

removeAll

➤ 기능

차트에 표시된 모든 속성 및 데이터를 초기화한다.

➤ Syntax

Syntax	<code>{ibchart}.removeAll()</code>
---------------	------------------------------------

➤ Info

returns	boolean, 성공 여부		
name	type	required	description
			없음

➤ Example

```
// 차트 초기화 처리
var res = myChart.removeAll();
```

➤ 제공 버전

7.3.0.0	
----------------	--

removeSeries

➤ **기능**

특정 시리즈를 제거한다.

➤ **Syntax**

Syntax	<code>{ibchart}.removeSeries(index)</code>
---------------	--

➤ **Info**

returns	boolean, 성공 여부		
name	type	required	description
index	number	Y	시리즈 Index

➤ **Example**

```
// 첫번째 시리즈 제거  
myChart.removeSeries(0);
```

➤ **제공 버전**

7.3.0.0	
----------------	--

series

➤ 기능

특정 시리즈의 객체를 반환 한다.

➤ Syntax

Syntax	<code>{ibchart}.series(index)</code>
---------------	--------------------------------------

➤ Info

returns	boolean, 성공 여부		
name	type	required	description
index	number	Y	시리즈 Index

➤ Example

```
// 첫번째 시리즈에 포인트 추가 하기
var series = myChart.series(0);
series.addPoint({
  x: 10,
  y: 200
});
```

➤ 제공 버전

7.3.0.0	
----------------	--

setLang

➤ 기능

차트의 언어권을 설정한다. (한국어, 영어 지원)

ibchartinfo.js > **IBLANG**에 언어 설정 기본 제공

다른 언어는 사용자가 언어설정에서 추가해야 사용가능함.

➤ Syntax

Syntax	{ibchart}.setLang(locale)
---------------	----------------------------------

➤ Info

returns	없음		
name	type	required	description
locale	string	Y	언어코드

➤ Example

// 한국어 myChart.setLang("kr");
// 영어 myChart.setLang("en");
// 일본어 myChart.setLang("jp");
// 중국어 myChart.setLang("cn");

➤ 제공 버전

7.3.0.0	
----------------	--

setOptions

➤ 기능

차트의 모든 속성을 설정한다.

아래 예제와 같이 setOptions 안에 chart, legend, plotOptions, colors, xAxis, yAxis, pane, subtitle, title, tooltip, credits 와 같이 적고 아래에 해당하는 속성을 설정한다. 속성값은 각 명칭별 상세속성 값과 같다.

(각 속성의 세부 속성 설명은 10. IBChart Property 를 참고 한다.)

➤ Syntax

Syntax	<code>{ibchart}.setOptions(prop, [opt])</code>
---------------	--

➤ Info

returns	없음		
name	type	required	description
prop	object	Y	속성 정보
prop.chart	object	N	기본 속성
prop.title	object	N	제목 속성
prop.subtitle	object	N	부제목 속성
prop.colors	array	N	시리즈별 적용할 색상 배열 집합
prop.xAxis	object	N	X축 속성
prop.yAxis	object	N	Y축 속성
prop.plotOptions	object	N	구성 옵션 속성
prop.tooltip	object	N	툴팁 속성
prop.pane	object	N	pane 속성 (gauge, polar only)
prop.legend	object	N	범례 속성
opt	object	N	옵션 속성
opt.append	boolean	N(false)	이전 설정과의 병합 처리 여부
opt.redraw	boolean	N(false)	설정 적용 후 다시 그리기 여부
opt.resetData	boolean	N(false)	데이터 병합 여부 (속성을 유지하면서 데이터 초기화 여부, ※

			opt.append값과 opt.mergeData 값이 false이면 데이터 초기화)
opt.mergeData	boolean	N(true)	데이터를 기존 데이터에 덮어씌울지 여부(opt.resetData값이 false일때만 적용됨)
opt.resetData	boolean	N(false)	시리즈 색상값(colors) 병합 여부 (속성을 유지하면서 데이터 초기화 여부, ※opt.append값과 opt.mergeColors값이 false이면 데이터 초기화)
opt.mergeData	boolean	N(true)	시리즈 색상값(colors)을 기존 시리즈 색상값에 덮어씌울지 여부 (opt.resetColors값이 false일때만 적용됨)
opt.deepCopy	boolean	N(false)	깊은 복사(데이터량에 따라 시간이 더 필요할 수 있음)

➤ Example

```
myChart.setOptions({
  chart : {
    backgroundColor : "#EDED",
    type : "column",
    style : {
      fontSize : "20px"
    }
  },
  legend : {
    layout : "vertical",
    align : "right",
    verticalAlign : "top"
  },
  plotOptions : {
```

```

series : {
  shadow : false,
  dataLabels : {
    enabled : true,
    align : "center"
  }
},
column : {
  pointPadding : 0.02
}
},
xAxis : {
  tickInterval : 1,
  labels : {
    enabled : true
  }
},
yAxis : {
  tickInterval : 100,
  title : {
    text : ""
  }
}
});

```

➤ 제공 버전

7.3.0.0	
7.3.1.4	resetData, mergeData 옵션 추가
7.3.1.12	resetColors, mergeColors 옵션 추가

setEventListener

➤ 기능

특정 이벤트를 추가 한다.

➤ Syntax

Syntax	<code>{ibchart}.setEventListener(eventType, func)</code>
---------------	--

➤ Info

returns	없음		
name	type	required	description
eventType	string	Y	이벤트 종류
func	function	Y	이벤트 핸들러

➤ Enum

■ eventType

value	description
searchEnd	조회 완료 이벤트
addSeries	시리즈 추가 완료 이벤트
afterPrint	프린트 직후 이벤트
beforePrint	프린트 직전 이벤트
click	차트 영역 클릭 이벤트
drilldown	드릴다운 이벤트
drillup	드릴업 이벤트
drillupAll	최상단 드릴업 이벤트
redraw	redraw 이벤트
selection	선택 이벤트
seriesAfterAnimate	시리즈 애니메이션 직후 이벤트
seriesCheckBoxClick	시리즈 체크박스 클릭 이벤트
seriesClick	시리즈 클릭 이벤트
seriesHide	시리즈 숨김 이벤트

seriesLegendItemClick	시리즈 범례 클릭 이벤트
seriesMouseOut	시리즈 마우스 아웃 이벤트
seriesMouseOver	시리즈 마우스 오버 이벤트
seriesShow	시리즈 보임 이벤트
pointClick	포인트 클릭 이벤트
pointMouseOut	포인트 마우스 아웃 이벤트
pointMouseOver	포인트 마우스 오버 이벤트
pointRemove	포인트 지움 이벤트
pointSelect	포인트 선택 이벤트
pointUnselect	포인트 선택 해제 이벤트
pointUpdate	포인트 업데이트 이벤트

➤ **Example**

```
// 조회 완료 이벤트 등록
myChart.setEventListener("searchend", function(event) {
    if(event.code === -1 || event.msg !== "OK") {
        return -1;
    }
    console.log("데이터 " + event.type + " 완료!");
});
```

➤ **제공 버전**

7.3.0.0

setEvents

➤ 기능

차트에서 사용할 이벤트들을 추가 한다

➤ Syntax

Syntax	<code>{ibchart}.setEvents(opt)</code>
---------------	---------------------------------------

➤ Info

returns	없음		
name	type	required	description
opt	array.<object>	Y	이벤트 종류와 핸들러로 구성된 객체

➤ Enum

■ eventType

value	description
addSeries	시리즈 추가 완료 이벤트
afterPrint	프린트 직후 이벤트
beforePrint	프린트 직전 이벤트
click	차트 영역 클릭 이벤트
drilldown	드릴다운 이벤트
drillup	드릴업 이벤트
drillupAll	최상단 드릴업 이벤트
load	차트 완성 이벤트
pointClick	포인트 클릭 이벤트
pointMouseOut	포인트 마우스 아웃 이벤트
pointMouseOver	포인트 마우스 오버 이벤트
pointRemove	포인트 지움 이벤트
pointSelect	포인트 선택 이벤트
pointUnselect	포인트 선택 해제 이벤트

pointUpdate	포인트 업데이트 이벤트
redraw	redraw 이벤트
searchEnd	조회 완료 이벤트
selection	선택 이벤트
seriesAfterAnimate	시리즈 애니메이션 직후 이벤트
seriesCheckBoxClick	시리즈 체크박스 클릭 이벤트
seriesClick	시리즈 클릭 이벤트
seriesHide	시리즈 숨김 이벤트
seriesLegendItemClick	시리즈 범례 클릭 이벤트
seriesMouseOut	시리즈 마우스 아웃 이벤트
seriesMouseOver	시리즈 마우스 오버 이벤트
seriesShow	시리즈 보임 이벤트

➤ **Example**

```
// 조회 완료와 차트영역 클릭 이벤트 등록
myChart.setEvents({
  searchEnd : function(event) {
    if(event.code === -1 || event.msg !== "OK") {
      return -1;
    }
    console.log("데이터 " + event.type + " 완료!");
  },
  click : function(event) {
    console.log("Chart Click!");
  }
});
```

➤ **제공 버전**

7.3.0.0	
----------------	--

setSize

➤ 기능

차트의 너비 및 높이를 동적으로 변경한다.

➤ Syntax

Syntax	<code>{ibchart}.setSize(width, height)</code>
---------------	---

➤ Info

returns	없음		
name	type	required	description
width	number	Y	차트의 너비 (px)
height	number	Y	차트의 높이 (px)

➤ Example

<pre>// 너비 500px, 높이 300px로 차트의 크기를 변경한다. myChart.setSize(500, 300);</pre>
--

➤ 제공 버전

7.3.1.10	setSize 메소드 추가
-----------------	----------------

updateWords

➤ 기능

Wordcloud 차트에 사용되는 단어를 업데이트한다.

➤ Syntax

Syntax	<code>{ibchart}.updateWords(opt)</code>
---------------	---

➤ Info

returns	없음		
name	type	required	description
없음			

➤ Example

```
// plotOptions와 series (단어 데이터) 만 업데이트
myChart.updateWords();
```

➤ 제공 버전

7.3.1.0	
----------------	--

version

➤ 기능

제품의 버전을 반환한다.

➤ Syntax

Syntax	<code>{ibchart}.version(opt)</code>
---------------	-------------------------------------

➤ Info

returns	string, IBChart의 버전		
name	type	required	description
없음			

➤ 제공 버전

7.3.0.0	
----------------	--

9.2 Series

차트의 series method 에서 반환되는 시리즈 객체에서 메소드 체이닝으로 사용 가능한 기능은 다음과 같다

addPoint

➤ 기능

동적으로 시리즈의 포인트를 추가한다.

추가 후 바로 화면에 표시 하지 않을 경우 redraw 인자를 false 로 설정한다.

redraw 인자를 false 로 설정하여 추가된 포인트는 draw 메소드를 통해 표시할 수 있다.

shift 인자를 true 로 설정하는 경우 포인트가 추가되는 시점에 가장 좌측의 포인트는 사라지면서 왼쪽으로부터 오른쪽으로 흐름을 표현할 때 사용 한다.

➤ Syntax

Syntax	{ibchart-series}.addPoint(prop, [opt])
---------------	---

➤ Info

returns	없음		
name	type	required	description
prop	number array object	Y	데이터 포인트 속성 (Y값, X,Y(Z) 의 배열, point 객체)
opt	object	N	옵션 속성
opt.redraw	boolean	N(true)	추가 후 다시 그릴지 여부
opt.shift	boolean	N(false)	포인트에 대한 시프트 처리 여부

➤ Example

```
// 동적으로 포인트를 추가하여 표현 한다.
myChart.series(0).addPoint({
  name: "A",
  x: 2,
  y: 3
});
```

➤ 제공 버전

7.3.0.0	
---------	--

getData

➤ 기능

대상 시리즈의 데이터를 가져온다.

Syntax

Syntax	<code>{ibchart-series}.getData()</code>
---------------	---

➤ Info

returns	array, 시리즈 데이터의 배열 집합		
name	type	required	description
			없음

➤ Example

```
// 첫번째 시리즈의 데이터를 가져온다.
var series = myChart.series(0).getData();
```

➤ 제공 버전

7.3.0.0	
----------------	--

hide

➤ 기능

보이는 시리즈를 감춘다.

➤ Syntax

Syntax	<code>{ibchart-series}.hide()</code>
--------	--------------------------------------

➤ Info

returns	없음		
name	type	required	description
없음			

➤ Example

```
// 첫번째 시리즈를 감춘다.
myChart.series(0).hide();
```

➤ 제공 버전

7.3.0.0	
---------	--

remove

➤ 기능

대상 시리즈를 삭제 한다.

➤ Syntax

Syntax	<code>{ibchart-series}.remove()</code>
---------------	--

➤ Info

returns	array, 시리즈 데이터의 배열 집합		
name	type	required	description
없음			

➤ Example

```
// 첫번째 시리즈를 제거한다.
myChart.series(0).remove();
```

➤ 제공 버전

7.3.0.0	
----------------	--

removePoint

➤ 기능

시리즈의 특정 포인트를 삭제 한다.

➤ Syntax

Syntax	<code>{ibchart-series}.removePoint(index, redraw)</code>
---------------	--

➤ Info

returns	없음		
name	type	required	description
index	number	Y	포인트의 index
redraw	boolean	N(false)	처리 후 다시 그릴지 여부

➤ Example

<pre>// 첫번째 시리즈의 첫번째 포인트 제거 myChart.series(0).removePoint(0, true);</pre>

➤ 제공 버전

7.3.0.0	
----------------	--

setData

➤ 기능

특정 시리즈의 데이터를 설정한다.

➤ Syntax

Syntax	<code>{ibchart-series}.setData(data, [opt])</code>
---------------	--

➤ Info

returns	없음		
name	type	required	description
data	object	Y	데이터 객체
opt	object	N	옵션 속성
opt.append	boolean	N(false)	기존 데이터에 append 처리 여부
opt.redraw	boolean	N(true)	데이터 적용후 다시 그릴지 여부
opt.animation	boolean	N(true)	애니메이션 여부

➤ Example

```
// 첫번째 시리즈의 데이터 재설정 하기
var series = myChart.series(0);
series.setData([20,30,50,80,90,100,20]);
```

➤ 제공 버전

7.3.0.0	
----------------	--

setOptions

➤ 기능

대상 시리즈의 속성을 설정한다.

Parameter	Type	설 명
Data	Object	Series의 데이터 설정 JSON Object
LegendIndex	Number	Series의 Legend의 표시되는 Index 설정
Name	String	Series의 Name 설정
Stack	String	Stack옵션 설정
Type	String	Series의 타입 설정
XAxis	Number	X축을 두개로 사용할 때 축 번호를 설정
YAxis	Number	Y축을 두개로 사용할 때 축 번호를 설정
ZIndex	Number	Series의 zIndex 설정

➤ Syntax

Syntax	<code>{ibchart-series}.setOptions(prop, [opt])</code>
---------------	---

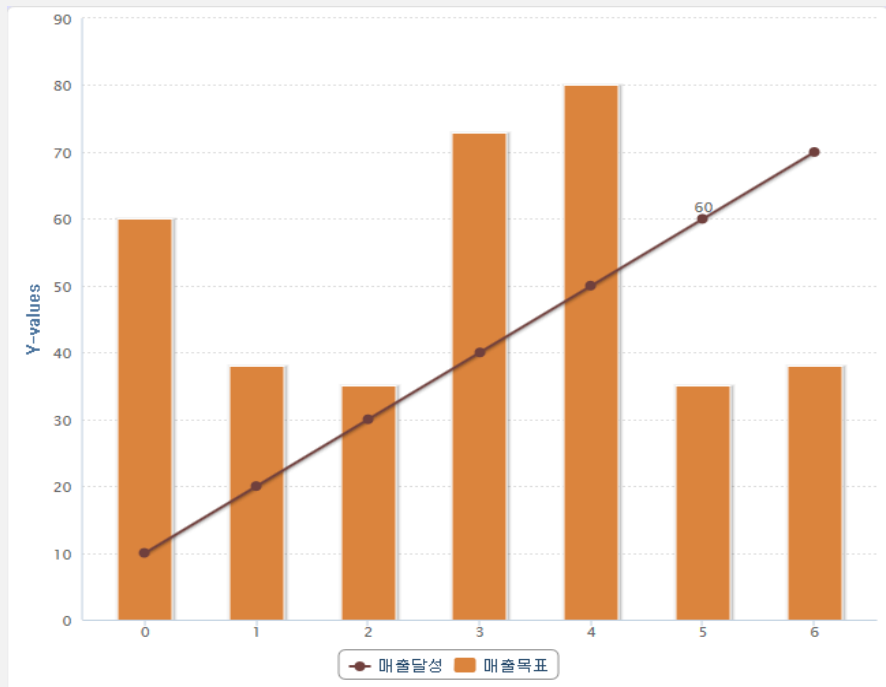
➤ Info

returns	없음		
name	type	required	description
prop	object	Y	시리즈 구성 옵션 속성
opt	object	N	옵션 속성
opt.append	boolean	N(false)	이전 설정과의 병합 처리 여부
opt.redraw	boolean	N(true)	설정 적용 후 다시 그리기 여부

➤ Example

```
var series = myChart.series(1); // 2번째 시리즈를 가져오기
series.SetOptions({
  type:"line",
  data:[10,20,30,40,50,60,70],
```

});



➤ 제공 버전

7.3.0.0

show

➤ 기능

대상 시리즈를 보이게 설정 한다.

➤ Syntax

Syntax	<code>{ibchart-series}.show()</code>
---------------	--------------------------------------

➤ Info

returns	없음		
name	type	required	description
없음			

➤ Example

```
// 첫번째 시리즈를 보이게 한다.
myChart.series(0).show();
```

➤ 제공 버전

7.3.0.0	
----------------	--

Chapter 10. IBChart Property

10. IBChart Property

차트 구성시 setOptions 에서 사용하는 속성들에 대한 세부 속성은 다음과 같다.

속성	타입	설명
chart	object	차트배경 및 형태 등을 설정
title	object	차트 제목 및 제목의 형태 등을 설정
subtitle	object	차트 부제목 및 부제목의 형태 등을 설정
colors	array	시리즈별 적용할 색상의 배열 집합 (배열의 길이보다 시리즈가 더 많을 경우 다시 첫번 첫부터 반복 적용 된다.)
xAxis	object	X축을 설정
yAxis	object	Y축을 설정
plotOptions	object	plot영역 (차트세부)설정 설정
tooltip	object	툴팁 속성 및 형태 등을 설정
pane	object	pane 속성 및 형태 등을 설정 (guage,polar 유형에서만 사용)
legend	object	범례의 위치 및 형태 설정.

10.1 chart

차트의 기본 옵션을 설정 한다.

차트 기본 속성 설정은 setOptions 의 chart 속성을 이용하여 정의 한다.

이 속성에서는 차트 유형, 차트의 배경색 등을 정의 할 수 있다.

➤ Info

name	type	required	description
type	string	N(line)	차트 유형
plotBackgroundColor	color	N	차트 영역의 배경색상
plotBorderColor	color	N(#cccccc)	차트 영역의 border 색상
plotBorderWidth	number	N(0)	차트 영역의 border 너비
plotShadow	boolean	N(false)	차트 영역에 대한 그림자 표시 여부
inverted	boolean	N(false)	X축과 Y축 반전 여부
polar	boolean	N(false)	polar 차트 구성 여부
zoomType	string	N(none)	마우스 드래그시 zoom 적용 방법 설정
backgroundColor	color	N	차트 컨테이너 영역의 배경 색상
borderColor	color	N	차트 컨테이너 영역의 border 색상
borderWidth	number	N(0)	차트 컨테이너 영역의 border 너비
borderRadius	number	N(0)	차트 컨테이너 영역에 대한 radius 적용 값
shadow	boolean	N	차트 컨테이너 영역에 대한 그림자 표시 여부

type

➤ 기능

차트 유형을 설정 한다.

➤ Info

name	type	required	description
type	string	N(line)	차트 유형

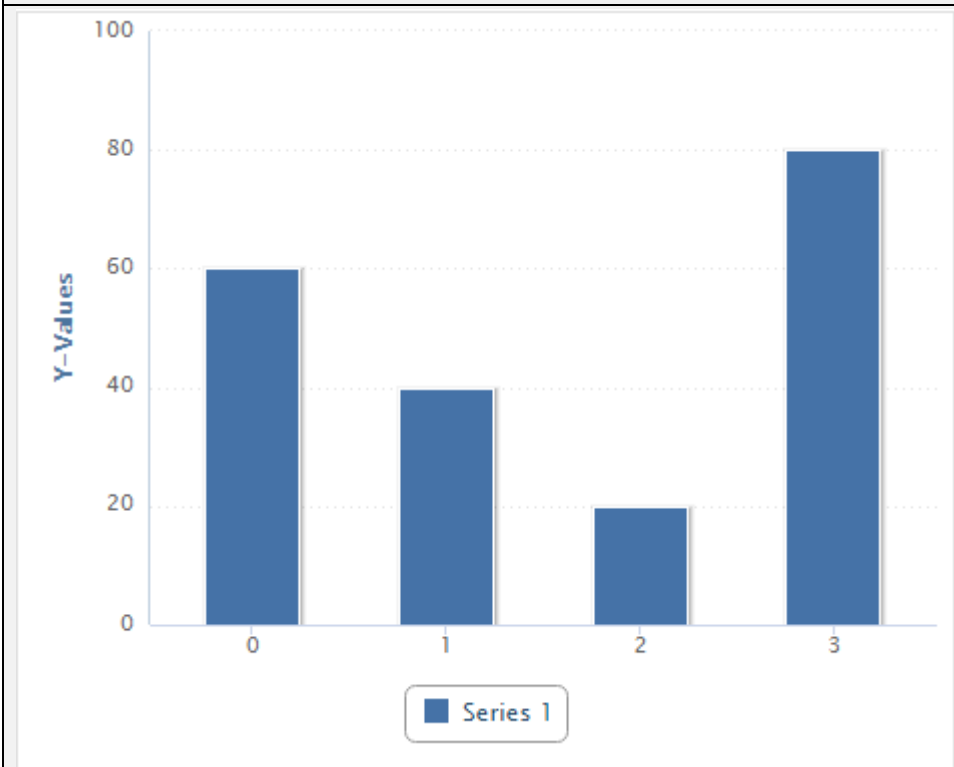
➤ Enum

■ type

value	description
area	영역 차트
arearange	영역 범위 차트
bar	가로 막대 차트
boxplot	박스플롯 차트
bubble	버블 차트
column	세로 막대 차트
columnrange	세로 막대 범위 차트
errorbar	오차 막대 차트
funnel	깔대기 차트 (highcharts/modules/funnel.js 필요)
gauge	게이지 차트 (highcharts/highcharts-more.js 필요)
heatmap	히트맵 차트 (highcharts/modules/heatmap.js 필요)
line	꺾은선 차트
pie	파이 차트
pyramid	원뿔형 차트
scatter	산점 차트
spline	부드러운선 차트
waterfall	폭포형 차트
wordcloud	워드클라우드

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    type: 'column'
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

7.3.1.0

워드클라우드 차트 추가

plotBackgroundColor

➤ 기능

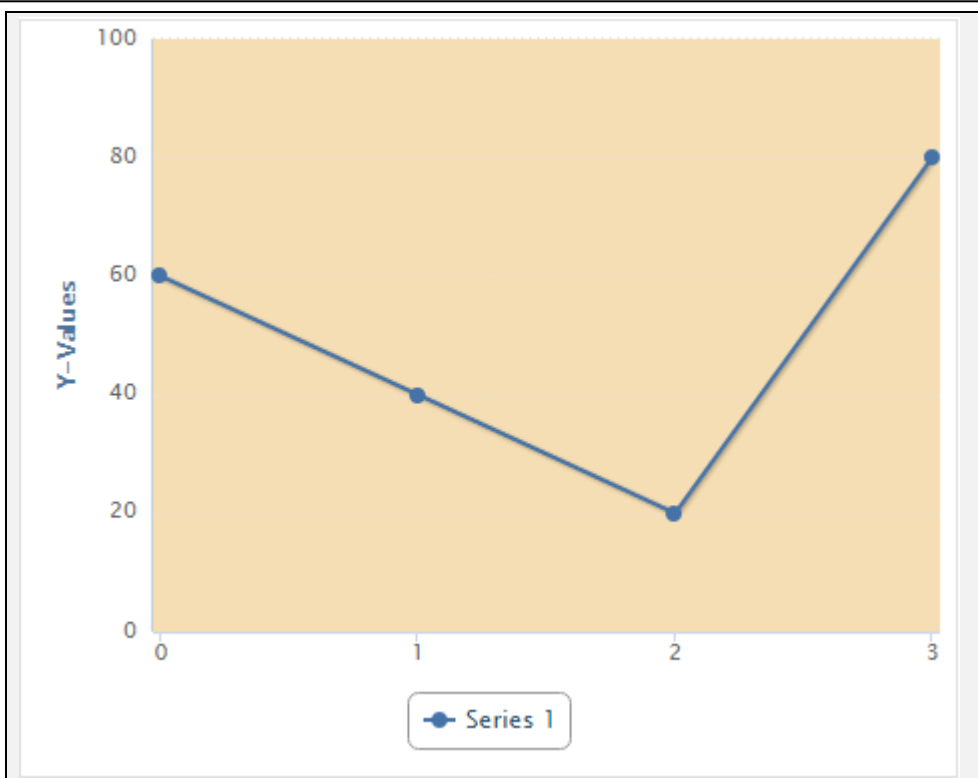
차트 영역의 배경색상을 설정 한다.

➤ Info

name	type	required	description
plotBackgroundColor	color	N	차트 영역의 배경색상

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    plotBackgroundColor: "#F5DEB3"
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

plotBorderColor

➤ 기능

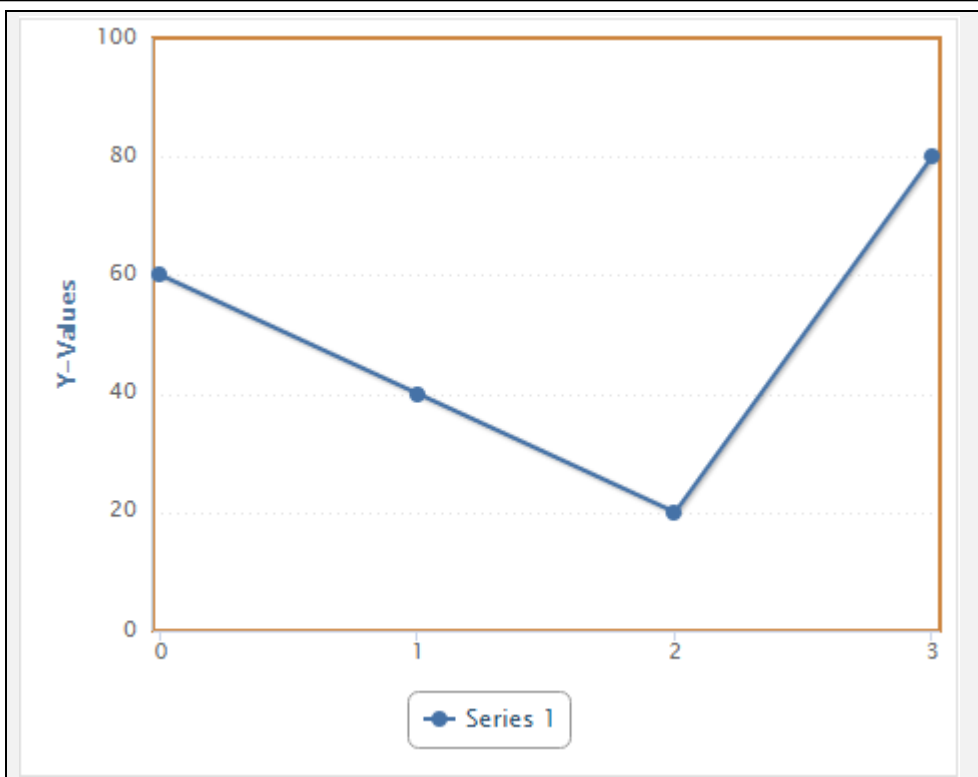
차트 영역의 border 색상을 설정 한다.

➤ Info

name	type	required	description
plotBorderColor	color	N(#cccccc)	차트 영역의 border 색상

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    plotBorderColor : "#CD853F",
    plotBorderWidth: 2
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

plotBorderWidth

➤ 기능

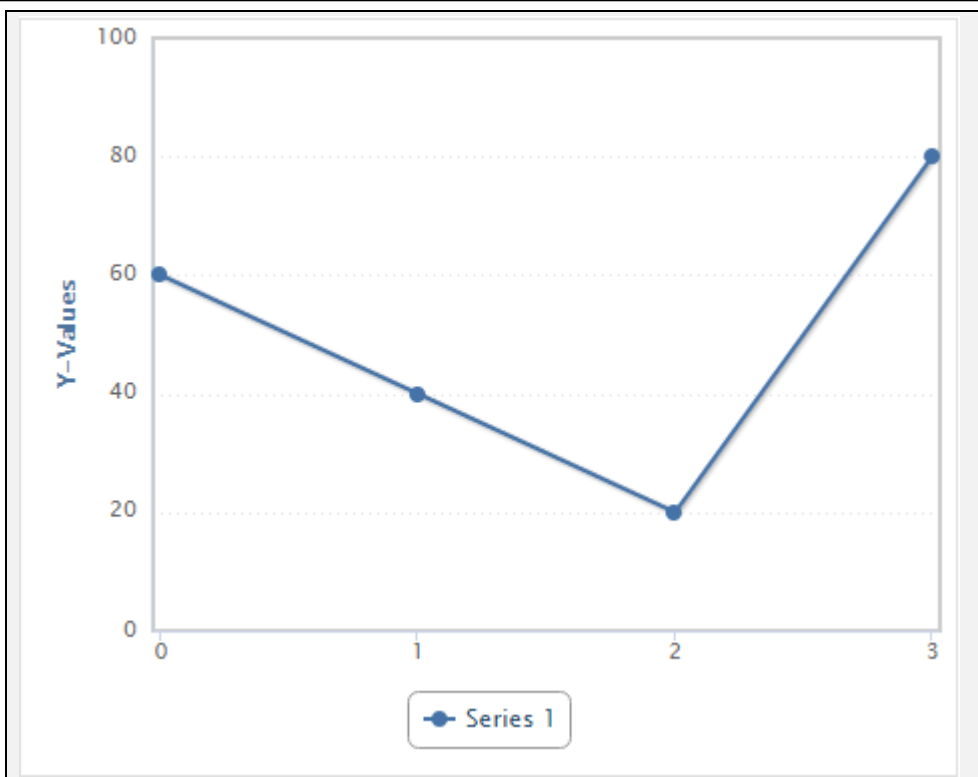
차트 영역의 border 너비를 설정 한다.

➤ Info

name	type	required	description
plotBorderWidth	number	N(0)	차트 영역의 border 너비

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    plotBorderWidth: 2
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

plotShadow

➤ 기능

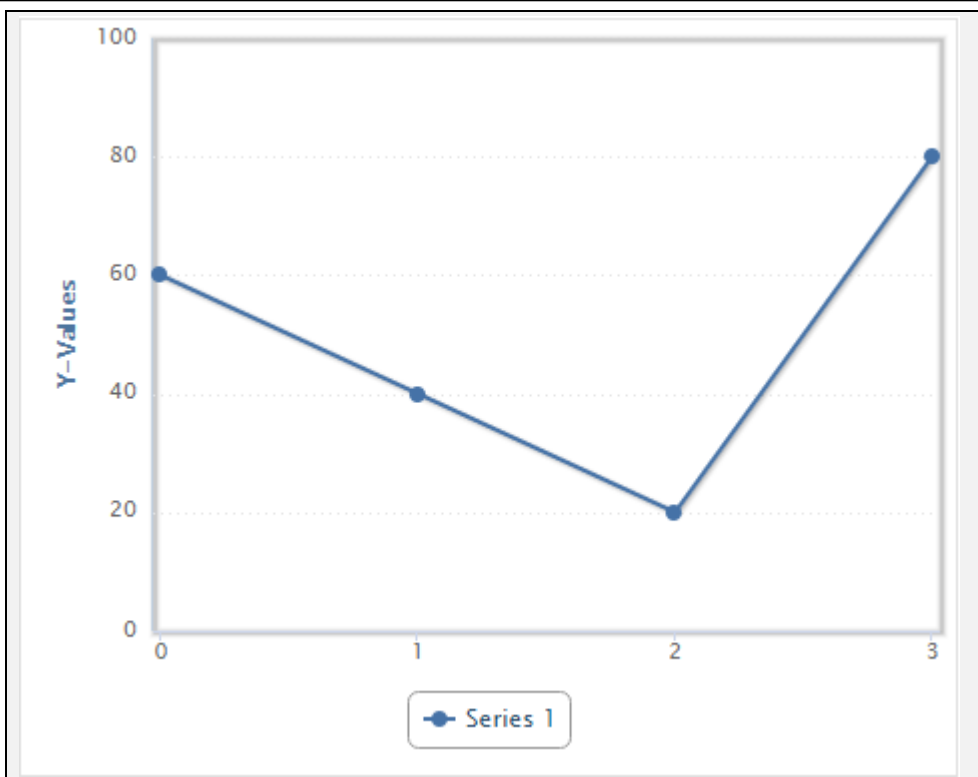
Plot 영역의 그림자 효과 여부를 설정 한다.

➤ Info

name	type	required	description
plotShadow	boolean	N(false)	차트 영역에 대한 그림자 표시 여부

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    plotShadow: true
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

inverted

➤ 기능

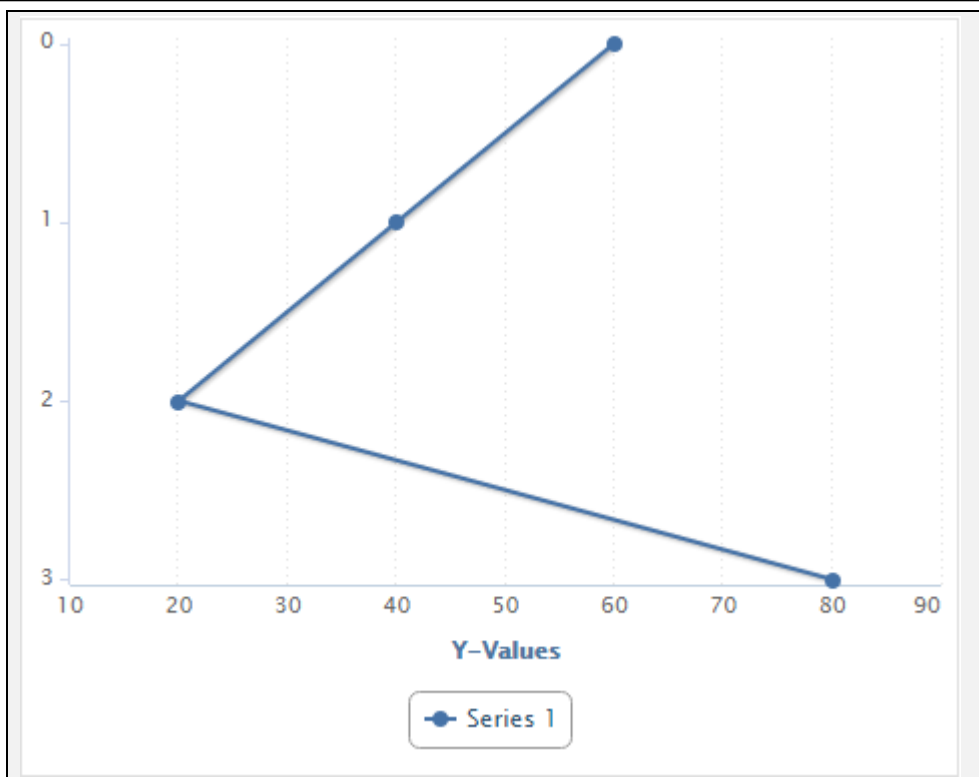
X 축과 Y 축 반전 여부를 설정 한다.

➤ Info

name	type	required	description
inverted	boolean	N(false)	X축과 Y축 반전 여부

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    inverted: true
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

polar

➤ 기능

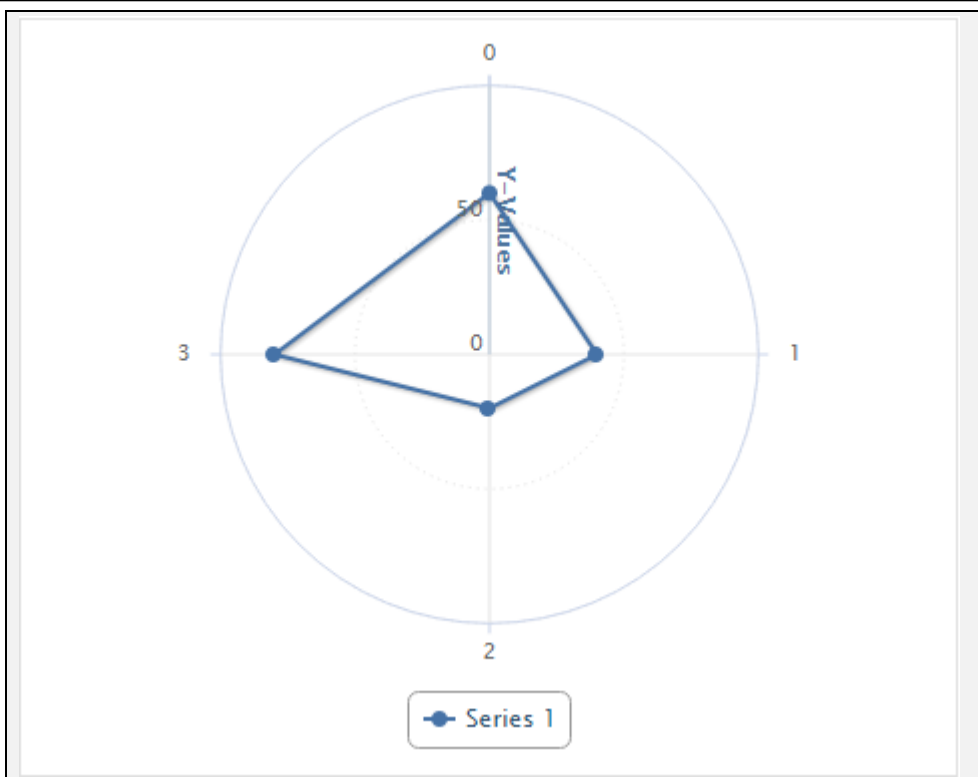
polar 차트 구성 여부를 설정 한다. (highcharts/highcharts-more.js 필요)

➤ Info

name	type	required	description
polar	boolean	N(false)	polar 차트 구성 여부

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    polar: true
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

zoomType

➤ 기능

마우스 드래그시 zoom 적용 방법을 설정 한다.

➤ Info

name	type	required	description
zoomType	string	N(none)	마우스 드래그시 zoom 적용 방법 설정

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart : {
    zoomType: "x"
  }
}, {
  redraw: true,
  append: true
});
```

➤ Enum

■ zoomType

value	description
none	사용안함
x	X축으로만 zoom 처리
y	Y축으로만 zoom 처리
xy	X축, Y축 양방향 zoom 처리

➤ 제공 버전

7.3.0.0	
---------	--

backgroundColor

➤ 기능

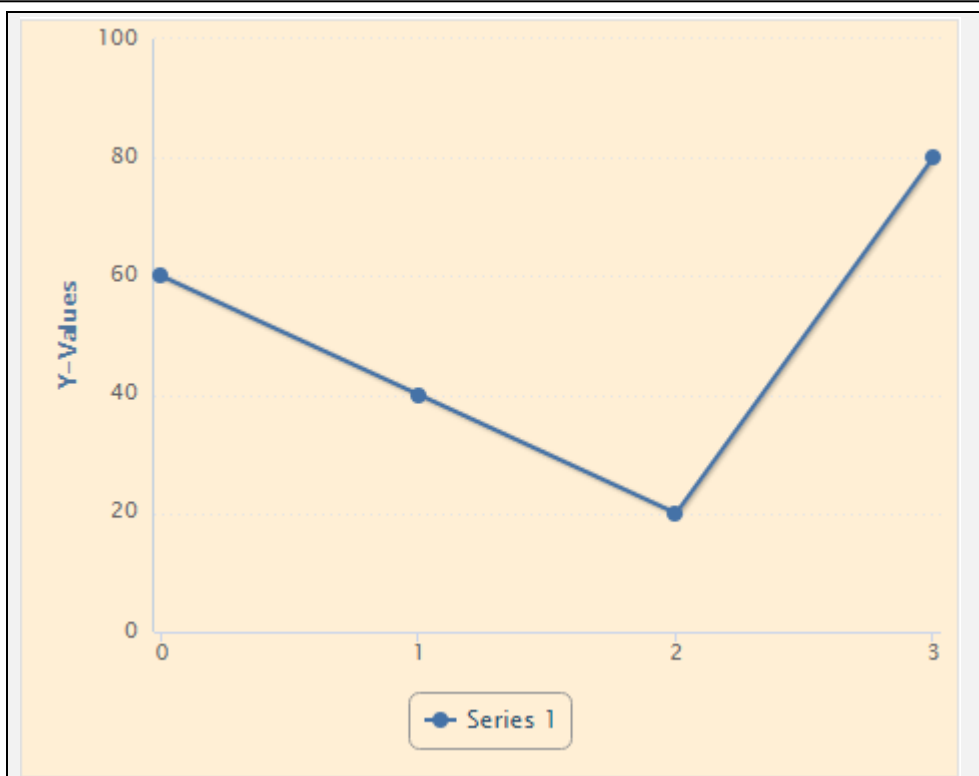
차트 컨테이너 영역의 배경색상을 설정 한다.

➤ Info

name	type	required	description
backgroundColor	color	N	차트 컨테이너 영역의 배경색상

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    backgroundColor: "#FFEFD5"
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

borderColor

➤ 기능

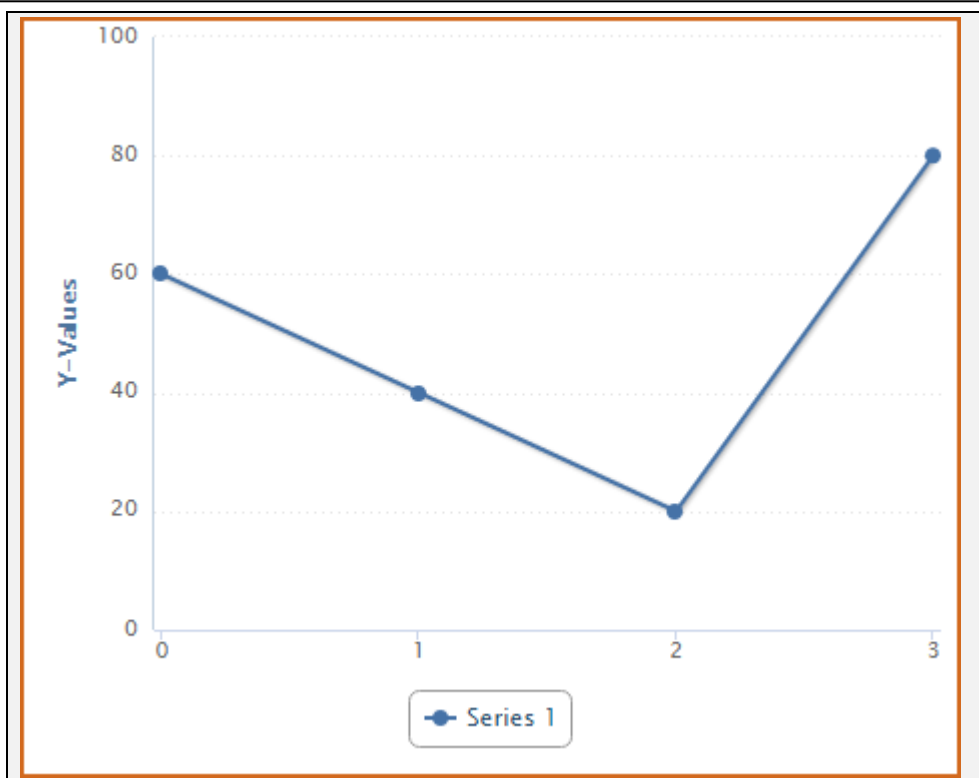
차트 컨테이너 영역의 border 색상을 설정 한다.

➤ Info

name	type	required	description
borderColor	color	N	차트 컨테이너 영역의 border 색상

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    borderColor: "#D2691E",
    borderWidth: 2
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

borderWidth

➤ 기능

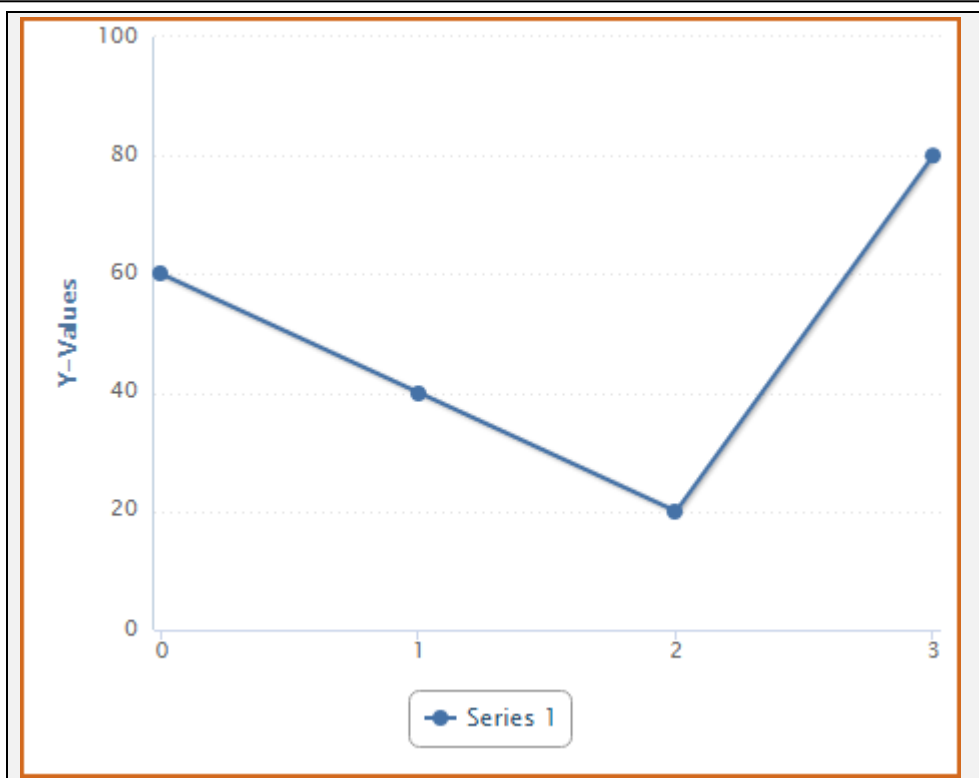
차트 컨테이너 영역의 border 너비를 설정 한다.

➤ Info

name	type	required	description
borderWidth	number	N(0)	차트 컨테이너 영역의 border 너비

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    borderColor: "#D2691E",
    borderWidth: 2
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ **제공 버전**

7.3.0.0	
---------	--

borderRadius

➤ 기능

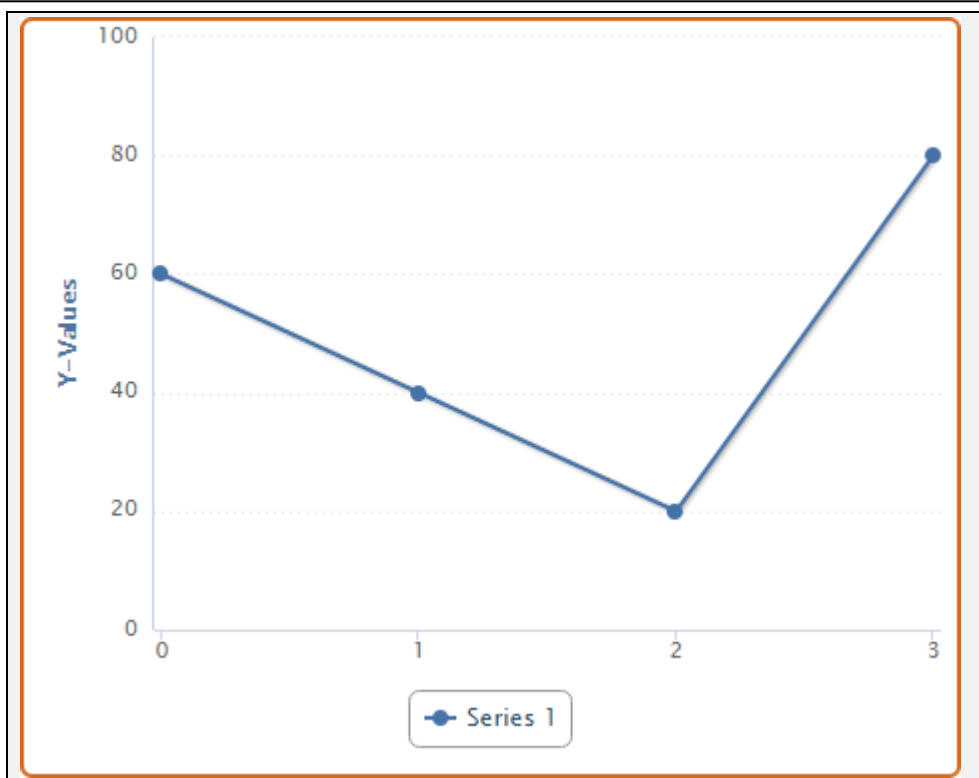
차트 컨테이너 영역에 대한 radius 값을 설정 한다.

➤ Info

name	type	required	description
borderRadius	number	N(0)	차트 컨테이너 영역에 대한 radius 값

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    borderColor: "#D2691E",
    borderWidth: 2,
    borderRadius: 5
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

shadow

➤ 기능

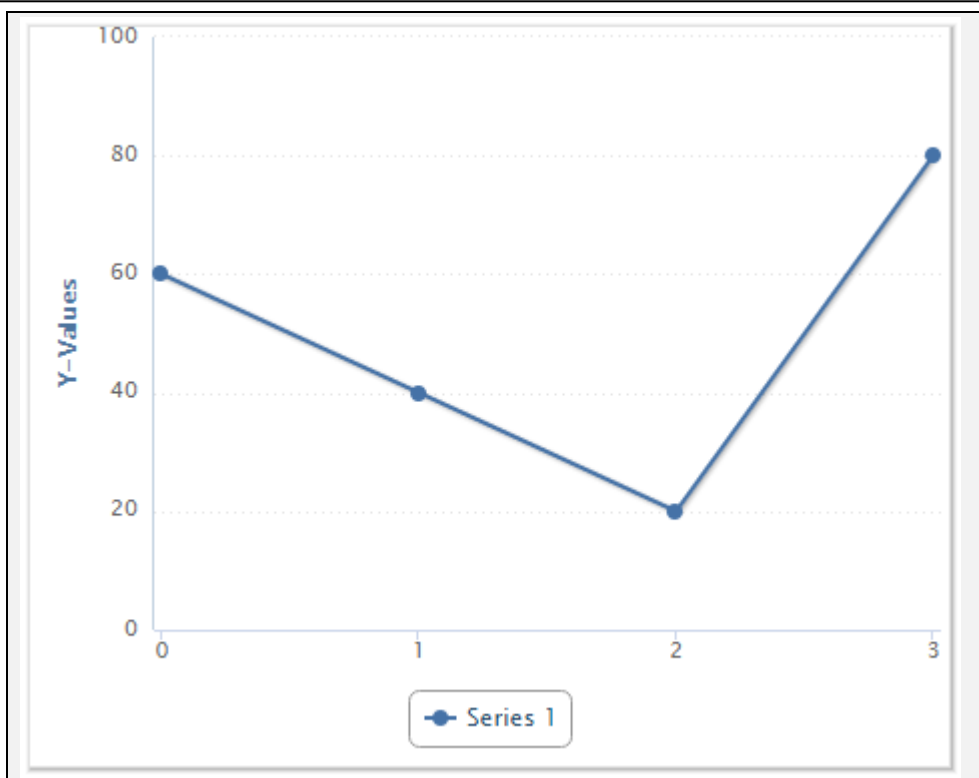
차트 컨테이너 영역에 대한 그림자 표시 여부를 설정 한다.

➤ Info

name	type	required	description
shadow	boolean	N	차트 컨테이너 영역에 대한 그림자 표시 여부

➤ Example

```
//차트 기본 속성 설정
myChart.setOptions({
  chart: {
    shadow: true
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

10.2 pane

pane에 대한 구성 옵션을 설정 한다. (gauge, polar)

gauge 차트와 polar 차트는 [10.1 chart>type>gauge](#)에서 처럼 js파일을 추가로 로드 해주어야한다.

➤ Info

name	type	required	description
startAngle	number	N	앵글의 시작 점
endAngle	number	N	앵글의 마지막 점
background	object	N	배경 속성
background. backgroundColor	color	N(#cccccc)	배경 색상
background. borderColor	color	N	border 색상
background. borderWidth	number	N	border의 너비
background. innerRadius	number string	N	pane 안쪽의 radius 적용 값 (숫자 또는 퍼센트 값으로 설정)
background. outerRadius	number string	N	pane 바깥쪽의 radius 적용 값 (숫자 또는 퍼센트 값으로 설정)

startAngle

➤ 기능

앵글의 시작 점을 설정 한다.

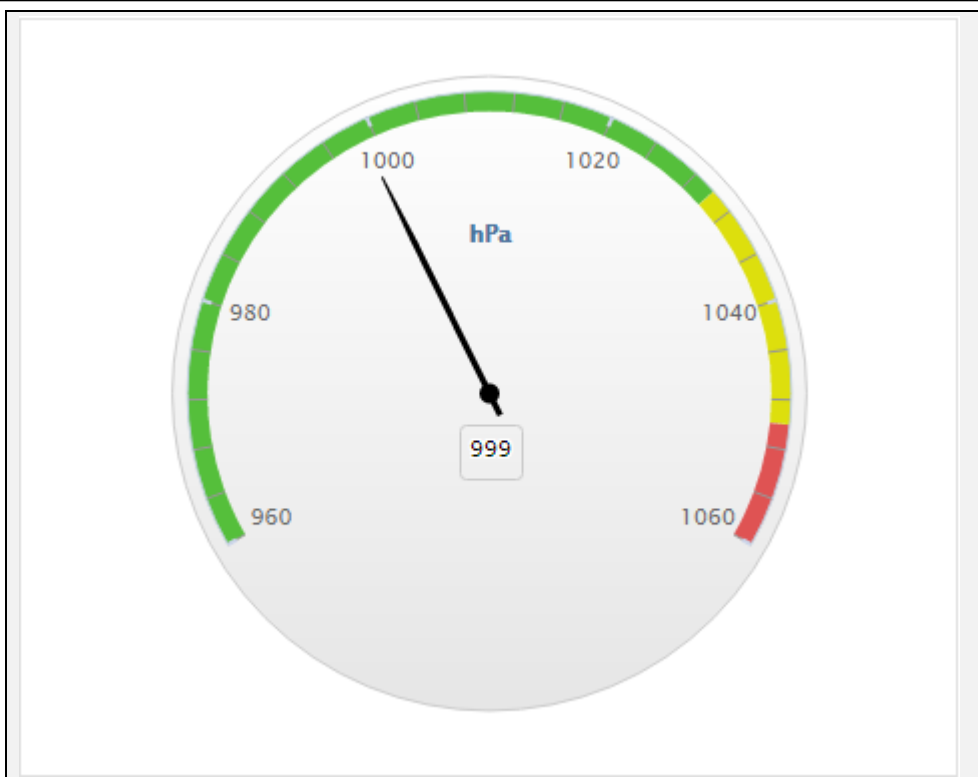
➤ Info

name	type	required	description
startAngle	number	N	앵글의 시작 점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "gauge"
  },
  pane: {
    startAngle: -120,
    endAngle: 120,
  },
  yAxis: {
    min: 960,
    max: 1060,
    gridLineWidth: 0,
    plotBands: [{
      from: 960,
      to: 1030,
      color: '#55BF3B' // green
    }, {
      from: 1030,
      to: 1050,
      color: '#DDDF0D' // yellow
    }, {
      from: 1050,
```

```
        to: 1060,  
        color: '#DF5353' // red  
    }],  
    title: {  
        text: "hPa"  
    }  
},  
tooltip: {  
    pointFormat: "{point.y} hPa"  
},  
series: [{  
    name: "hPa",  
    data: [999]  
}]  
}, {  
    append: true,  
    redraw: true  
});
```

➤ 제공 버전

7.3.0.0

endAngle

➤ 기능

앵글의 마지막 점을 설정 한다.

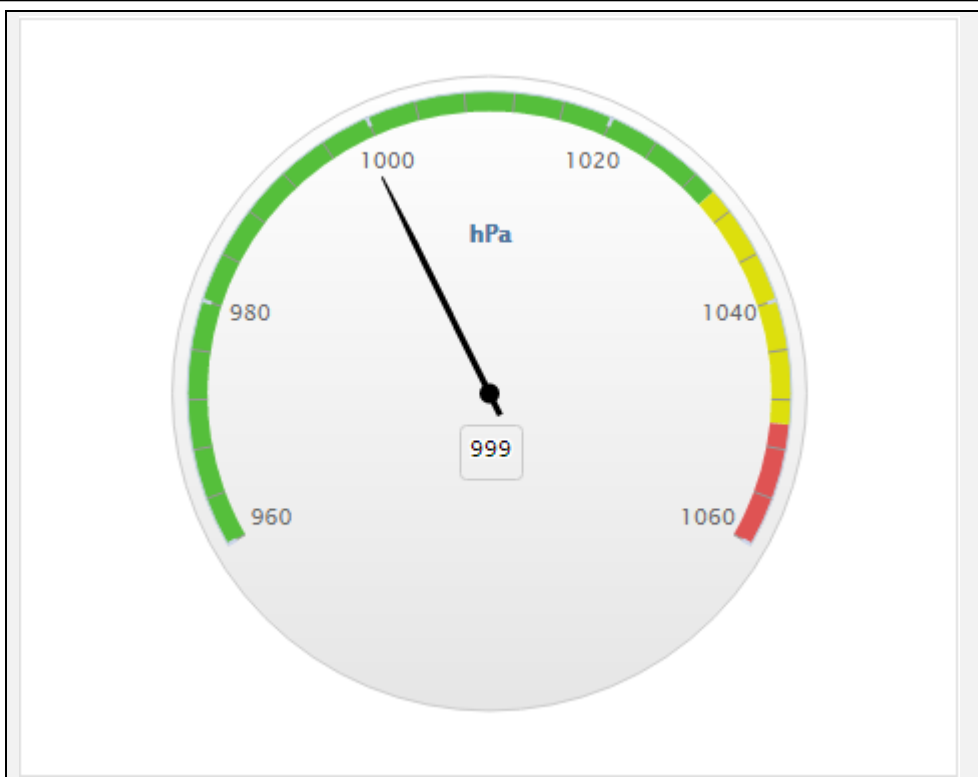
➤ Info

name	type	required	description
endAngle	number	N	앵글의 마지막 점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "gauge"
  },
  pane: {
    startAngle: -120,
    endAngle: 120,
  },
  yAxis: {
    min: 960,
    max: 1060,
    gridLineWidth: 0,
    plotBands: [{
      from: 960,
      to: 1030,
      color: '#55BF3B' // green
    }, {
      from: 1030,
      to: 1050,
      color: '#DDDF0D' // yellow
    }, {
      from: 1050,
```

```
        to: 1060,  
        color: '#DF5353' // red  
    }],  
    title: {  
        text: "hPa"  
    }  
},  
tooltip: {  
    pointFormat: "{point.y} hPa"  
},  
series: [{  
    name: "hPa",  
    data: [999]  
}]  
}, {  
    append: true,  
    redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

background

➤ 기능

배경 속성을 설정 한다.

➤ Info

name	type	required	description
background	object	N	배경 속성
background. backgroundColor	color	N(#cccccc)	배경 색상
background. borderColor	color	N	border 색상
background. borderWidth	number	N	border의 너비
background. innerRadius	number string	N	pane 안쪽의 radius 적용 값 (숫자 또는 퍼센트 값으로 설정)
background. outerRadius	number string	N	pane 바깥쪽의 radius 적용 값 (숫자 또는 퍼센트 값으로 설정)

➤ Example

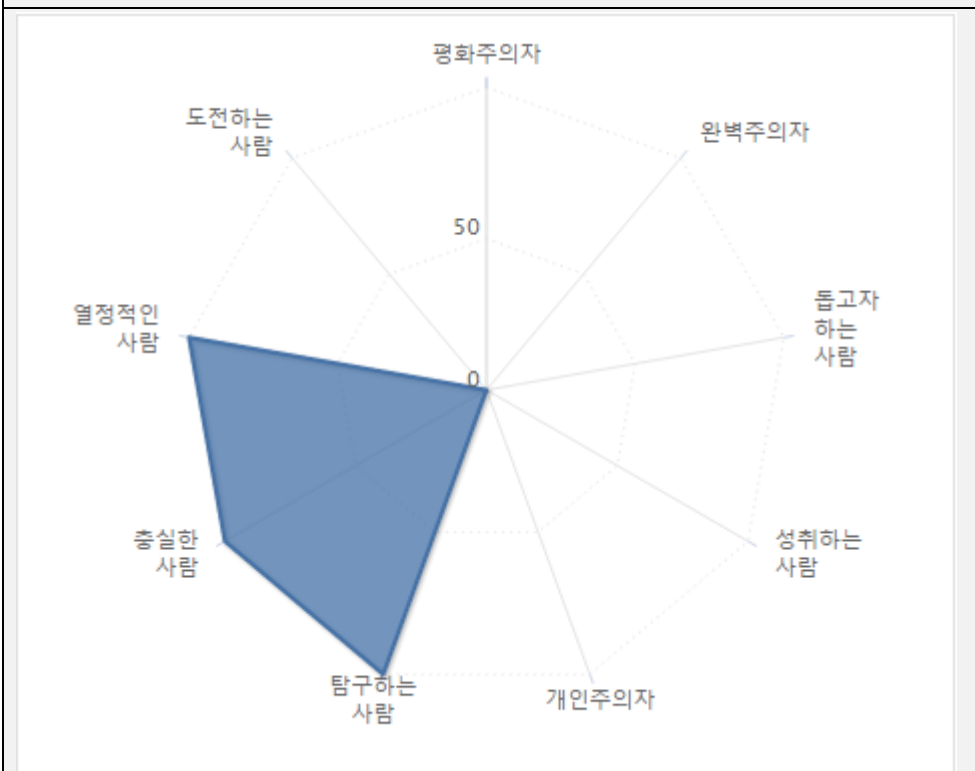
```
myChart.setOptions({
  chart: {
    polar: true,
    type: 'area'
  },
  xAxis: {
    categories: ["평화주의자", "완벽주의자", "돕고자 하는 사람", "성취하는 사람", "개인의주의자", "탐구하는 사람", "충실한 사람", "열정적인 사람", "도전하는 사람"],
```

```
    tickmarkPlacement: 'on',
    lineWidth: 0
  },
  yAxis: {
    title: {
      text : ""
    },
    gridLineInterpolation: 'polygon',
    lineWidth: 0,
    max: 100,
    min: 0
  },
  pane: [{
    background: null,
  }],
  tooltip: {
    enabled: false
  },
  legend: {
    enabled: false
  },
  plotOptions: {
    series: {
      marker: {
        enabled: false,
        states: {
          hover: {
            enabled: false
          },
          select: {
            enabled: false
          }
        }
      }
    }
  }
}
```

```

    }
  }
},
series: [{
  data: [0, 0, 0, 0, 0, 100, 100, 100, 0],
  pointPlacement: 'on'
}]
}, {
  append: true,
  redraw: true
});

```



➤ **제공 버전**

7.3.0.0

10.3 plotOptions

차트의 시리즈에 대한 구성 옵션을 설정 한다.

설정 시 **series** 속성 하위로 설정하게 되면 모든 시리즈에 적용되고 차트 유형 속성 하위로 설정하면 해당 차트 유형에만 구성 옵션이 적용 된다. 그리고 **series**의 속성은 모든 차트 유형의 속성보다 높은 우선 순위를 갖는다.

차트 유형별 지원하는 속성 정보는 아래의 표를 참고 한다.

속성	area	line	scatter	spline	bar	column	pie	wordcloud
animation	○	○	○	○	○	○	○	○
borderColor					○	○	○	
borderRadius					○	○		
borderWidth					○	○	○	
center							○	
cursor	○	○	○	○	○	○	○	
colorIterate								○
colorSort								○
dashStyle	○	○	○	○				
dataLabels	○	○	○	○	○	○	○	
endAngle							○	
fillColor	○							
groupPadding					○	○		
innerSize							○	
lineColor	○							
lineWidth	○	○		○				
marker	○	○	○	○				
negativeColor	○	○		○	○	○		
negativeFillColor	○							
orientation								○
pointStart	○	○	○	○	○	○		

pointPadding					○	○		
pointPlacement	○	○		○				
pointWidth					○	○		
rotation								○
shadow	○	○	○	○	○	○	○	
showInLegend	○	○	○	○	○	○		
stacking	○	○		○	○	○		
states	○	○	○	○	○	○	○	
startAngle							○	
scaleType								○
step		○						
spiralType								○
placeType								○
style								○
visible	○	○	○	○	○	○	○	

animation

➤ 기능

애니메이션 효과 사용 여부를 설정 한다.

➤ Info

name	type	required	description
animation	boolean	N(true)	애니메이션 효과 사용여부

➤ Example

```
myChart.setOptions( {
    series: {
        animation: false
    }
});
```

➤ 제공 버전

7.3.0.0	
---------	--

borderColor

➤ 기능

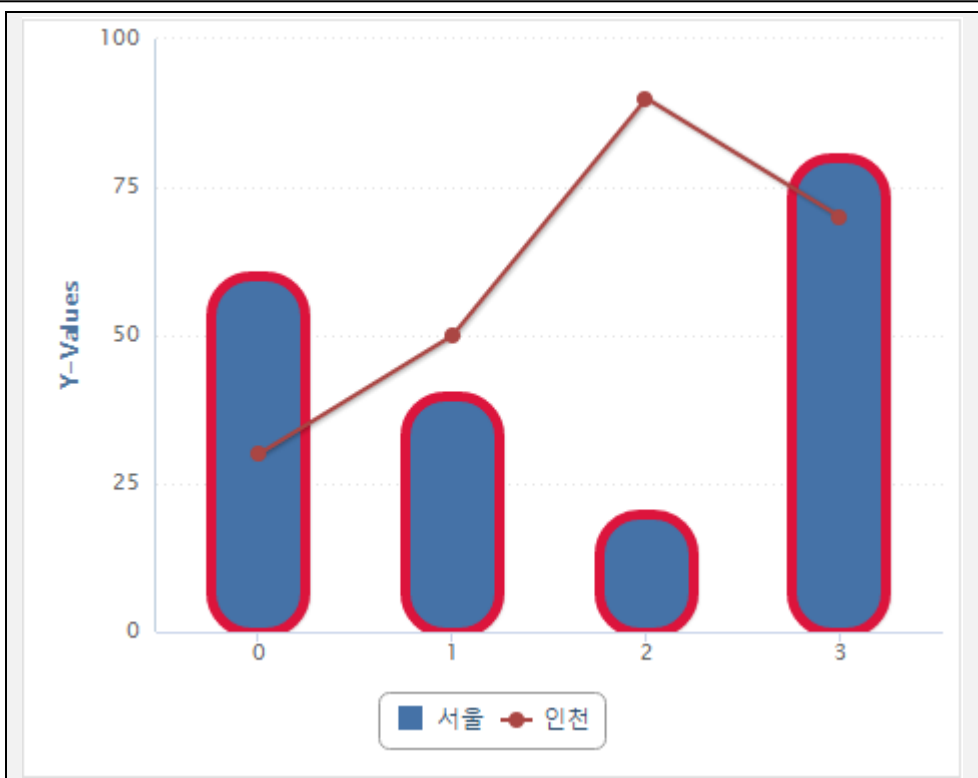
border 색상을 설정 한다.

➤ Info

name	type	required	description
borderColor	color	N(#ffffff)	border 색상

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      borderColor : "#dc143c",
      borderWidth: 5,
      borderRadius: 20
    }
  },
  series: [{
    type: "column",
    name: "서울",
    data: [60, 40, 20, 80]
  }, {
    name: "인천",
    data: [30, 50, 90, 70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

borderRadius

➤ 기능

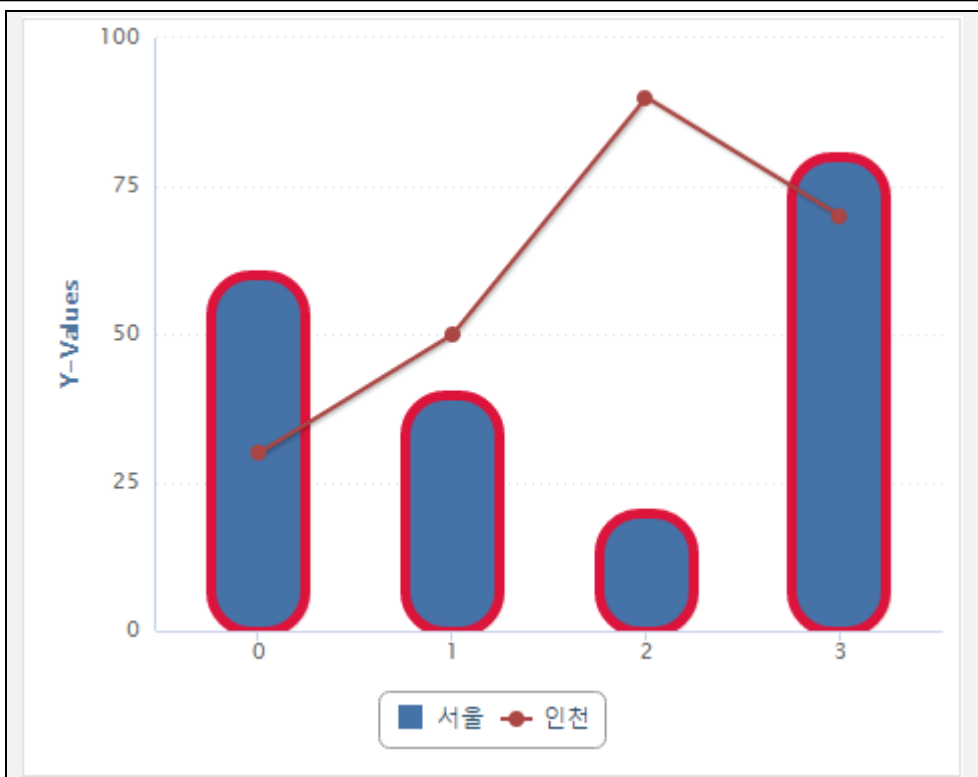
border 의 radius 적용 값을 설정 한다.

➤ Info

name	type	required	description
borderRadius	number	N(0)	border의 radius 적용 값

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      borderColor : "#dc143c",
      borderWidth: 5,
      borderRadius: 20
    }
  },
  series: [{
    type: "column",
    name: "서울",
    data: [60, 40, 20, 80]
  }, {
    name: "인천",
    data: [30, 50, 90, 70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

borderWidth

➤ 기능

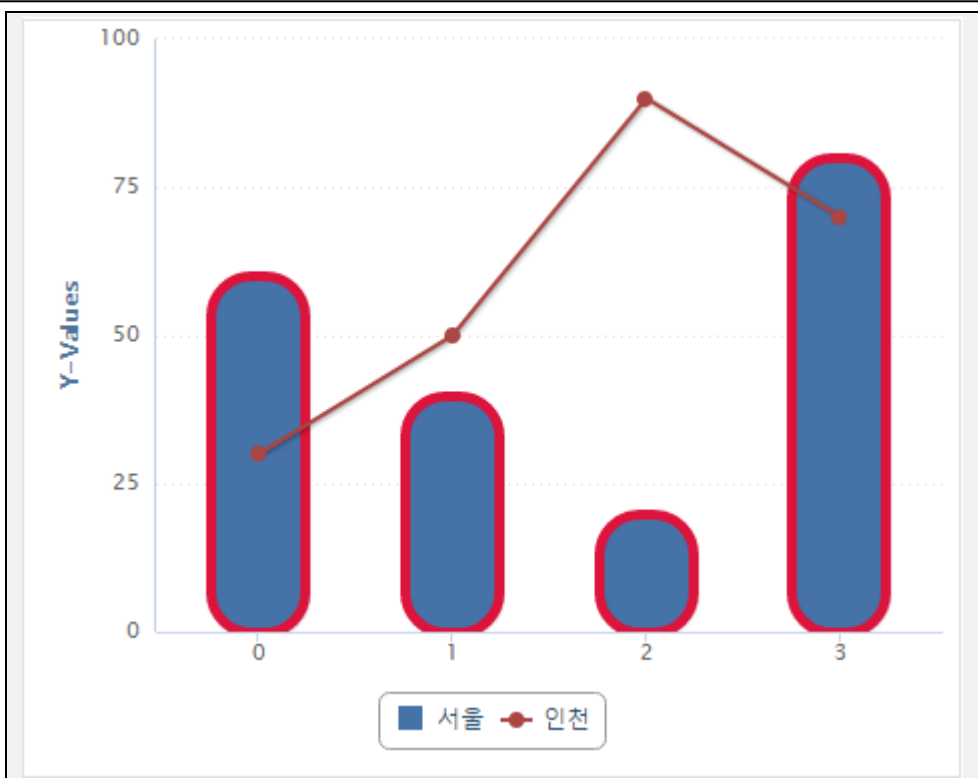
border 의 너비를 설정 한다.

➤ Info

name	type	required	description
borderWidth	number	N(1)	border의 너비

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      borderColor : "#dc143c",
      borderWidth: 5,
      borderRadius: 20
    }
  },
  series: [{
    type: "column",
    name: "서울",
    data: [60, 40, 20, 80]
  }, {
    name: "인천",
    data: [30, 50, 90, 70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

center

➤ 기능

pie 와 같은 유형의 차트에 대한 중심 위치를 설정 한다.

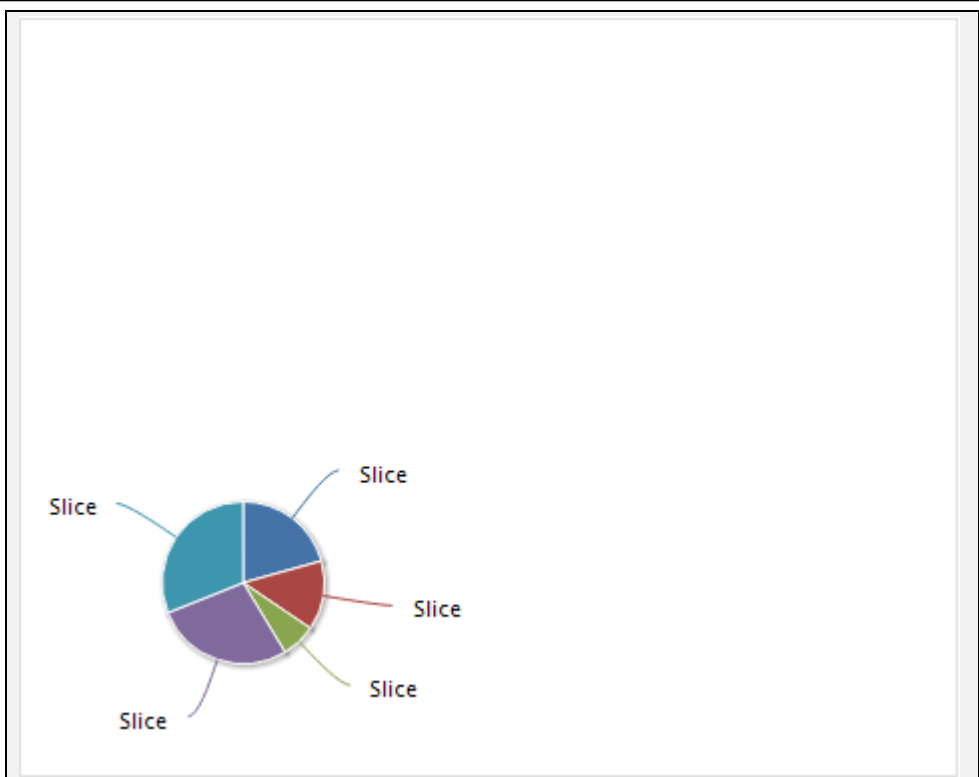
배열내의 값은 숫자 또는 퍼센트형태의 값으로 설정이 가능 하다.

➤ Info

name	type	required	description
center	array	N	차트의 중심 위치

➤ Example

```
myChart.setOptions({
  chart: {
    type: "pie"
  },
  plotOptions: {
    pie: {
      center: ["20%", "80%"]
    }
  },
  series: [{
    name : "서울",
    data : [60,40,20,80,90]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
----------------	--

cursor

➤ 기능

마우스 오버 시 커서의 모양을 설정 한다.

➤ Info

name	type	required	description
cursor	string	N(default)	커서 모양

➤ Enum

■ cursor

value	description
cursor	기본 모양
pointer	포인터 모양

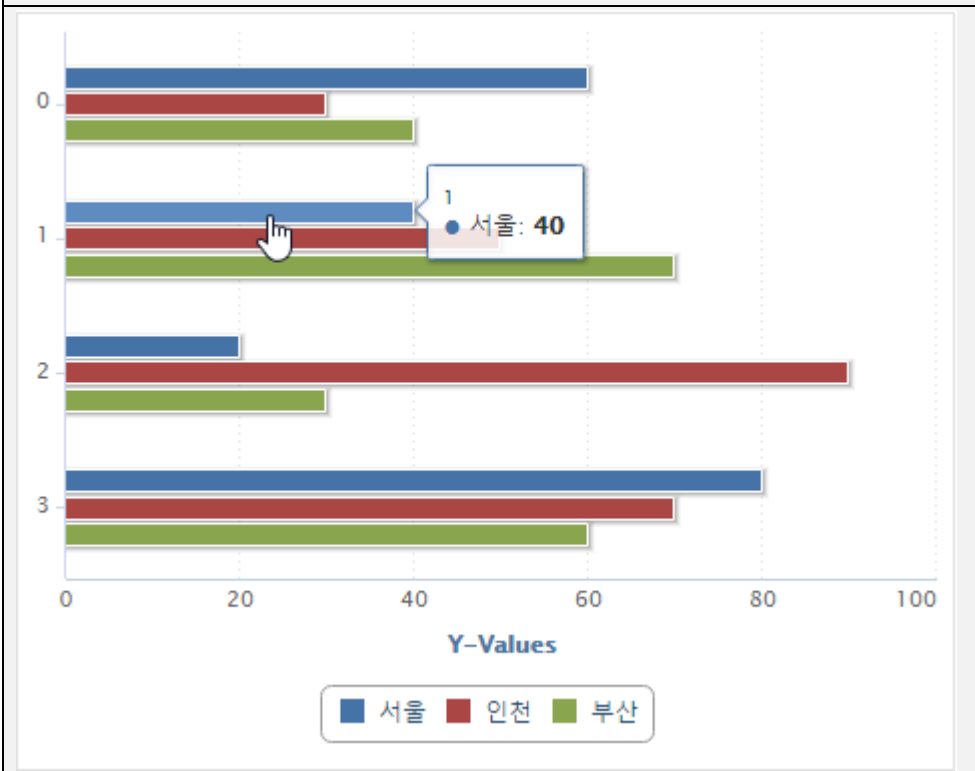
➤ Example

```
myChart.setOptions({
  chart: {
    type: "bar"
  },
  plotOptions: {
    bar: {
      cursor : "pointer"
    }
  },
  series: [{
    name : "서울",
    data : [60,40,20,80]
  }, {
    name : "인천",
    data : [30,50,90,70]
```

```

}, {
  name : "부산",
  data : [40, 70, 30, 60]
}]
}, {
  append: true,
  redraw: true
});

```



➤ **제공 버전**

7.3.0.0	
---------	--

colorIterate

➤ 기능

Wordcloud 차트의 색 반복 여부를 설정 한다.

➤ Info

name	type	required	description
colorIterate	boolean	N(true)	색상 반복 여부

➤ Enum

■ colorIterate

value	description
true	설정된 colors의 값으로 색을 반복해서 설정
false	설정된 colors 값이 이외는 차트 draw()와 down2Image()할 때마다 바뀜

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  colors: ["#0d0", "#fc0"],
  plotOptions: {
    wordcloud: {
      colorIterate: true
    }
  }
}, {
  append: false
});
```

```
//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 60
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 30
      }]
    }
  ]
}, {
  append: true
});
```

IB Sheet 7
IBLeaders
아이비리더스
IBChart(H) 7

➤ **제공 버전**

7.3.1.0

wordcloud 차트에만 적용가능

colorSort

➤ 기능

Wordcloud 차트의 색 지정 방식

➤ Info

name	type	required	description
colorSort	string	N("none")	색지정 방식

➤ Enum

■ colorSort

value	description
"none"	데이터 입력 순서
"rank"	입력된 데이터의 최대값부터 내림차순

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      colorSort: "rank"
    }
  }
}, {
  append: false
});

//wordcloud 데이터 로드
```



```
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 30
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 20
      }]
    }
  ]
}, {
  append: true
});
```

IB Sheet 7
아이비리더스
IBLeaders
IBChart(H) 7

➤ 제공 버전

7.3.1.0

wordcloud 차트에만 적용가능

dashStyle

➤ 기능

차트 선의 종류를 설정 한다..

➤ Info

name	type	required	description
dashStyle	string	N(solid)	선의 종류

➤ Enum

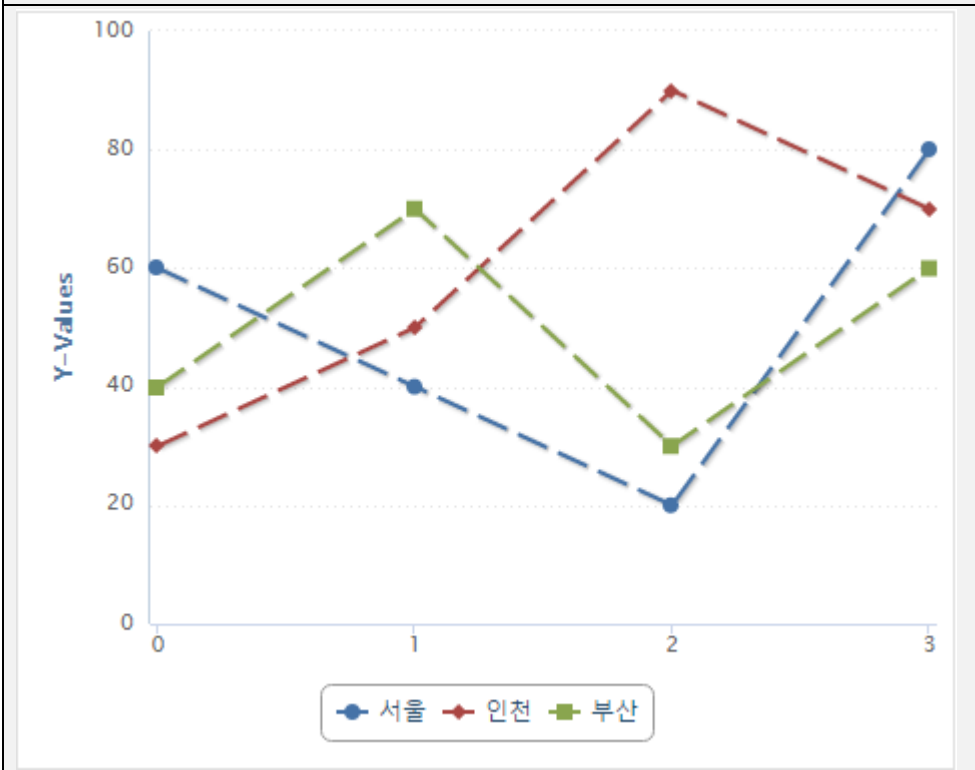
■ dashStyle

value	description
solid	실선
dash	대쉬선
longdash	긴 대쉬선
dot	점선

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      dashStyle: "longDash"
    }
  },
  series: [{
    name : "서울",
    data : [60,40,20,80]
  }, {
    name : "인천",
    data : [30,50,90,70]
  }, {
```

```
name : "부산",
data : [40,70,30,60]
}]
}, {
append: true,
redraw: true
});
```



➤ 제공 버전

7.3.0.0

dataLabels

➤ 기능

데이터 라벨의 속성을 설정 한다.

➤ Info

name	type	required	description
dataLabels	object	N	데이터 라벨 속성
dataLabels.align	string	N(center)	정렬 방법
dataLabels.color	color	N	색상
dataLabels.distance	number	N(30)	끝지점 부터의 표시 위치
dataLabels.enabled	boolean	N(false)	표시 여부
dataLabels.format	string	N({y})	출력 포맷 Ex) format: '{y} mm'
dataLabels.formatter	function	N	출력 포맷을 정의하는 callback Ex) this.y 현재 y값
dataLabels.rotation	number	N(0)	기울기
dataLabels.style	object	N	스타일 속성
dataLabels.useHTML	boolean	N	HTML 사용 여부
dataLabels.x	number	N	좌우 포지션 이동 값
dataLabels.y	number	N	상하 포지션 이동 값
dataLabels.inside	boolean	N	상자 내부의 레이블을 정렬여부 True일 경우 내부의 정렬

■ dataLabels.style Info

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기 (단위:px)
fontWeight	string	N(bold)	폰트 bold 여부

➤ Enum

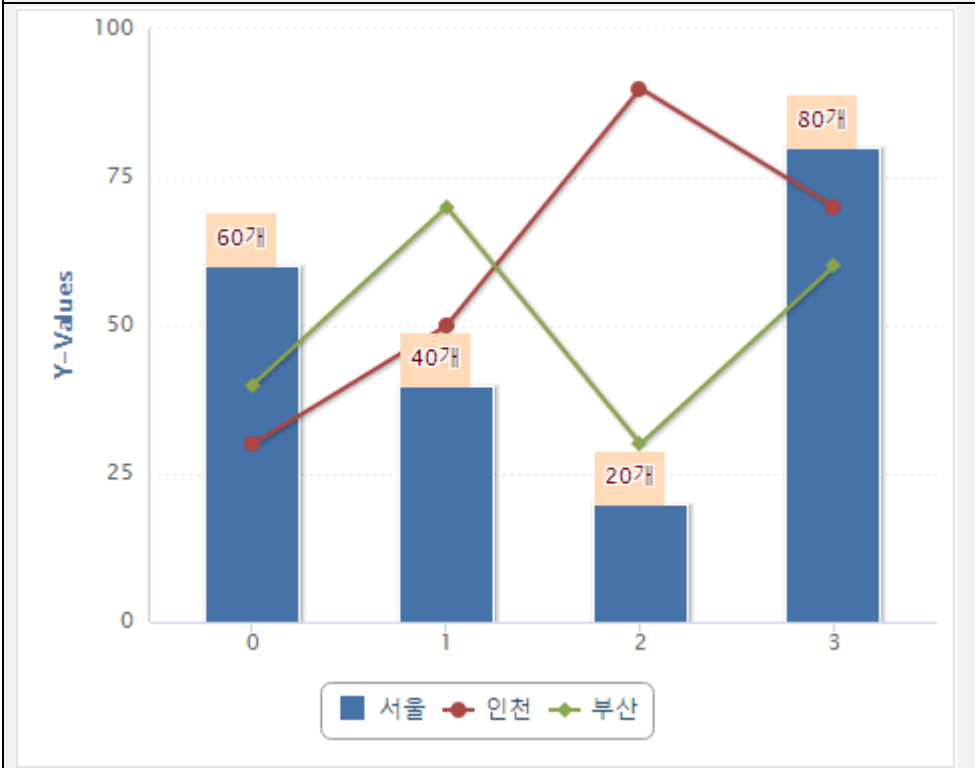
■ align

value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      dataLabels : {
        enabled:true,
        align:"left",
        backgroundColor:"#ffdab9",
        color:"#800000",
        formatter:function(){
          return this.y + "개";
        }
      }
    }
  },
  series: [{
    type: "column",
    name : "서울",
    data : [60,40,20,80]
  }, {
    name : "인천",
    data : [30,50,90,70]
  }, {
    name : "부산",
    data : [40,70,30,60]
  }]
});
```

```
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0

endAngle

➤ 기능

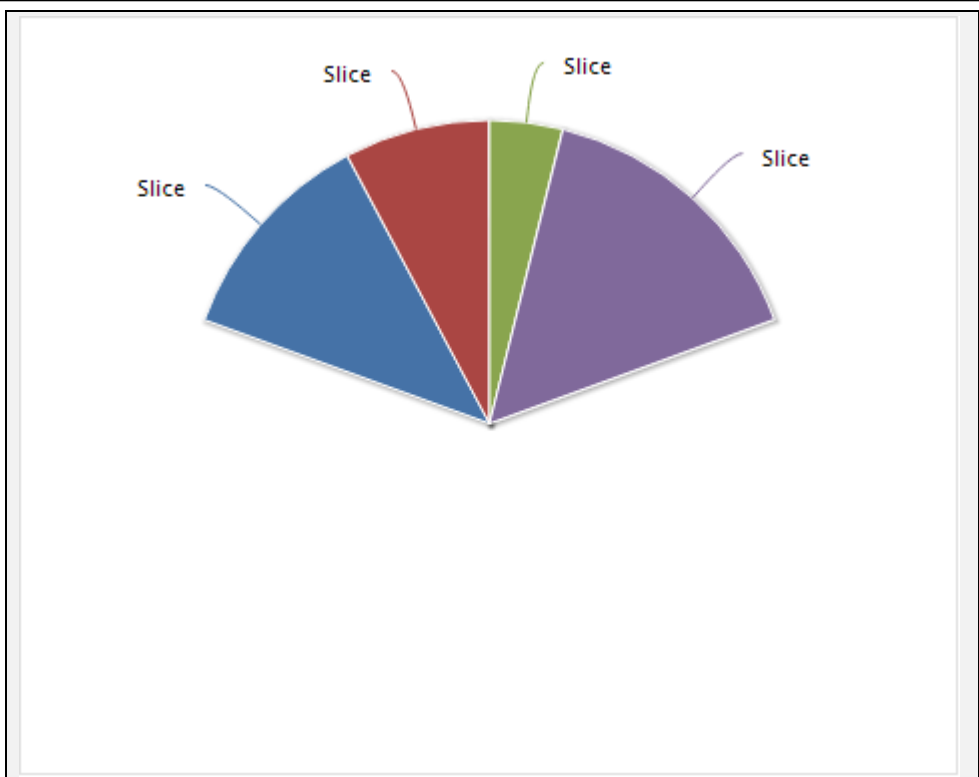
앵글의 마지막 점을 설정 한다.

➤ Info

name	type	required	description
endAngle	number	N	앵글의 마지막 점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "pie"
  },
  plotOptions: {
    pie: {
      startAngle:-70,
      endAngle:70
    }
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ **제공 버전**

7.3.0.0	
---------	--

fillColor

➤ 기능

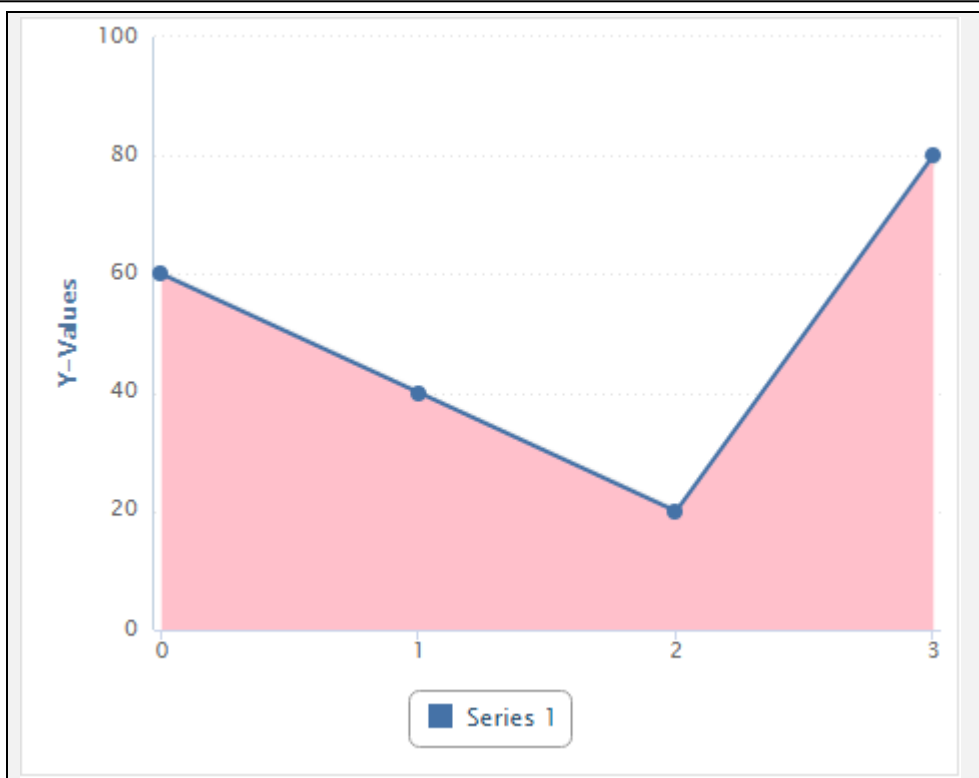
시리즈 영역의 배경색을 설정 한다.

➤ Info

name	type	required	description
fillColor	color	N	시리즈 영역의 배경색

➤ Example

```
myChart.setOptions({
  chart: {
    type: "area"
  },
  plotOptions: {
    area: {
      fillColor: "#ffc0cb"
    }
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

groupPadding

➤ 기능

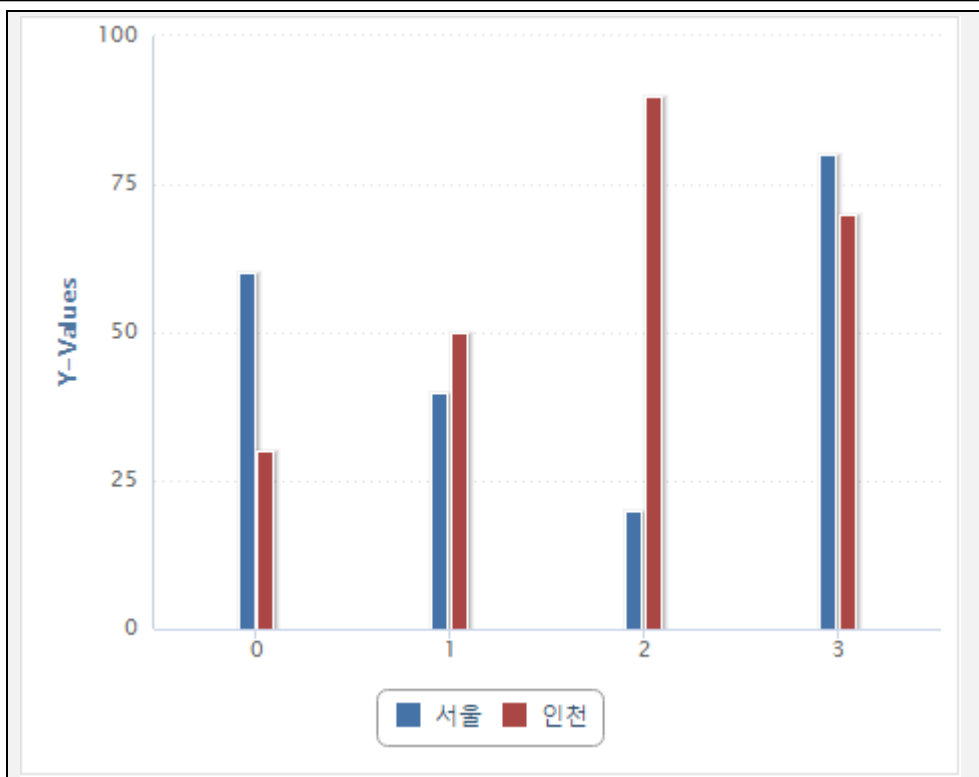
그룹 간 padding 적용 값을 설정 한다.

➤ Info

name	type	required	description
groupPadding	number	N(0.2)	그룹 간 padding 적용 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      groupPadding : 0.4
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

innerSize

➤ 기능

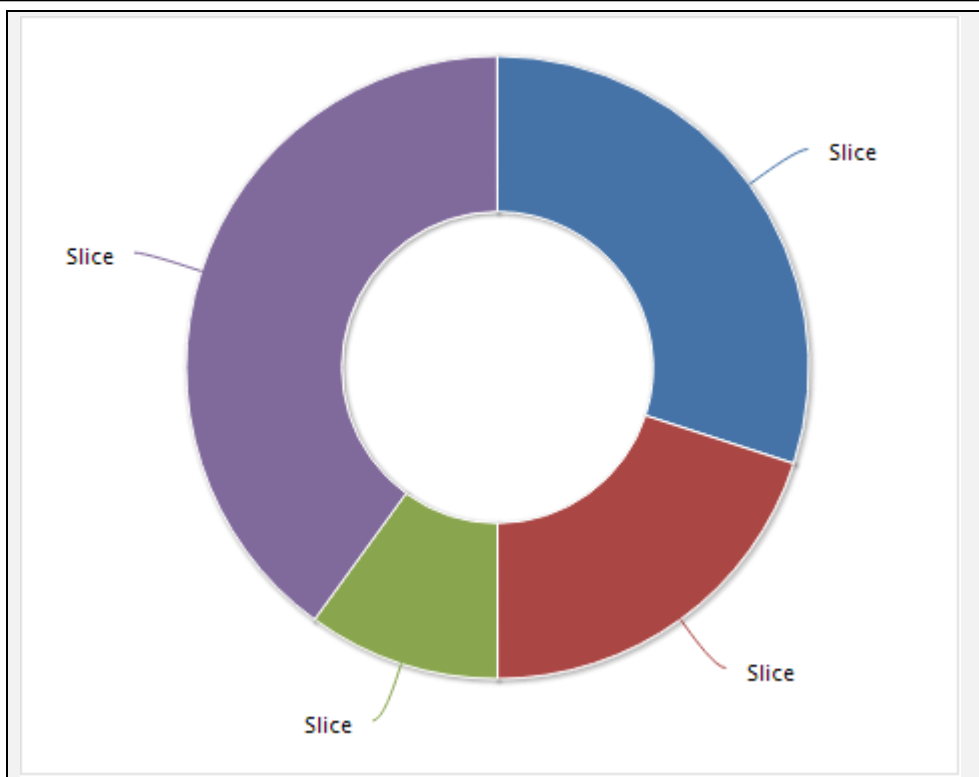
pie 와 같은 유형차트 내부 원의 크기를 설정 한다.

➤ Info

name	type	required	description
innerSize	number string	N(0)	차트의 내부 원의 크기

➤ Example

```
myChart.setOptions({
  chart: {
    type: "pie"
  },
  plotOptions: {
    pie: {
      innerSize : "50%"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

lineColor

➤ 기능

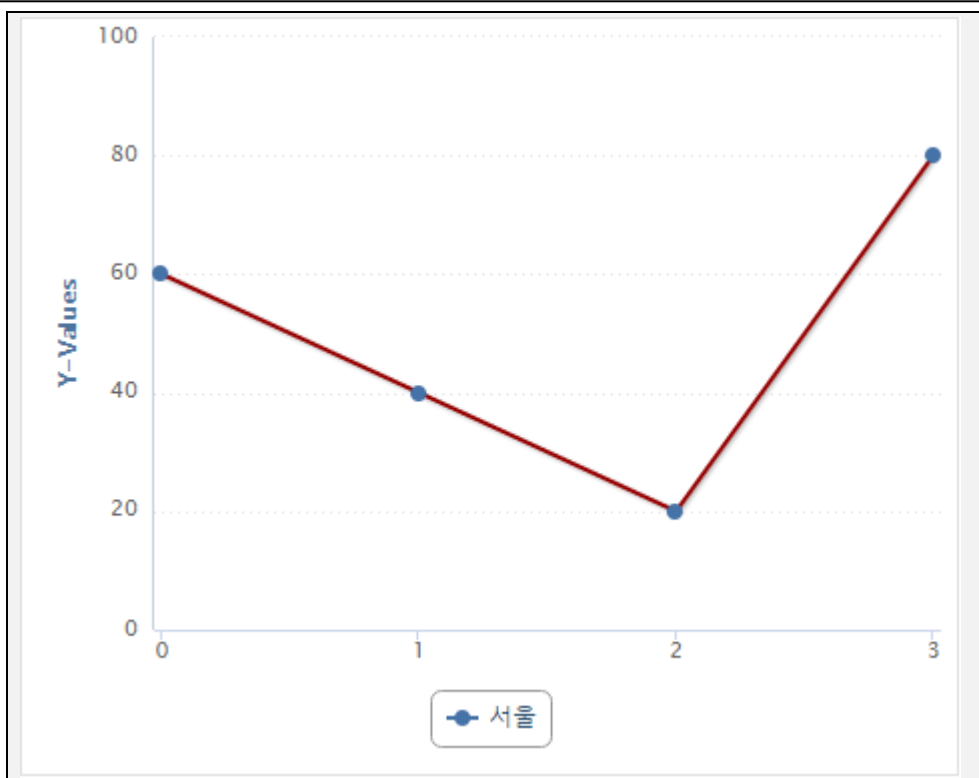
시리즈 선의 색상을 설정 한다.

➤ Info

name	type	required	description
lineColor	color	N	시리즈 선의 색상

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      lineColor: "#990000"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0	
---------	--

lineWidth

➤ 기능

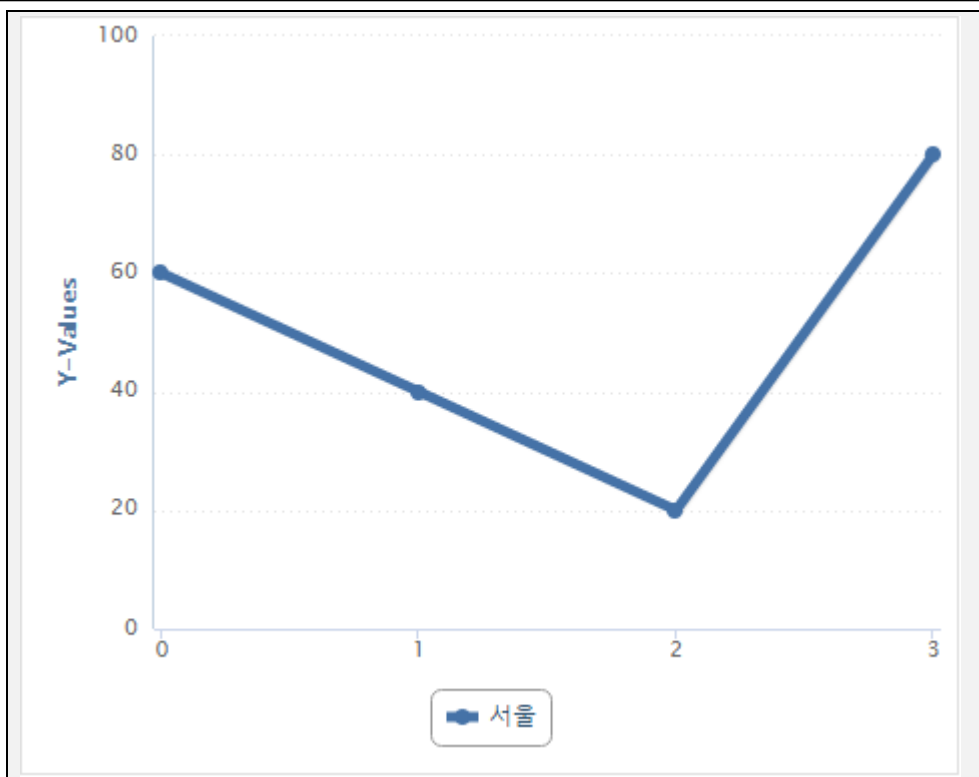
시리즈 선의 너비를 설정 한다.

➤ Info

name	type	required	description
linewidth	number	N(2)	시리즈 선의 너비

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      linewidth:5
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

marker

➤ 기능

포인트의 마커 속성을 설정 한다.

➤ Info

name	type	required	description
marker	object	N	포인트 마커 속성
marker.enabled	boolean	N	사용 여부
marker.radius	number	N(4)	radius 적용 값
marker.states	object	N	상태 속성
marker.states.hover	object	N	마우스 hover 시 속성
marker.states.select	object	N	포인트 선택시 속성

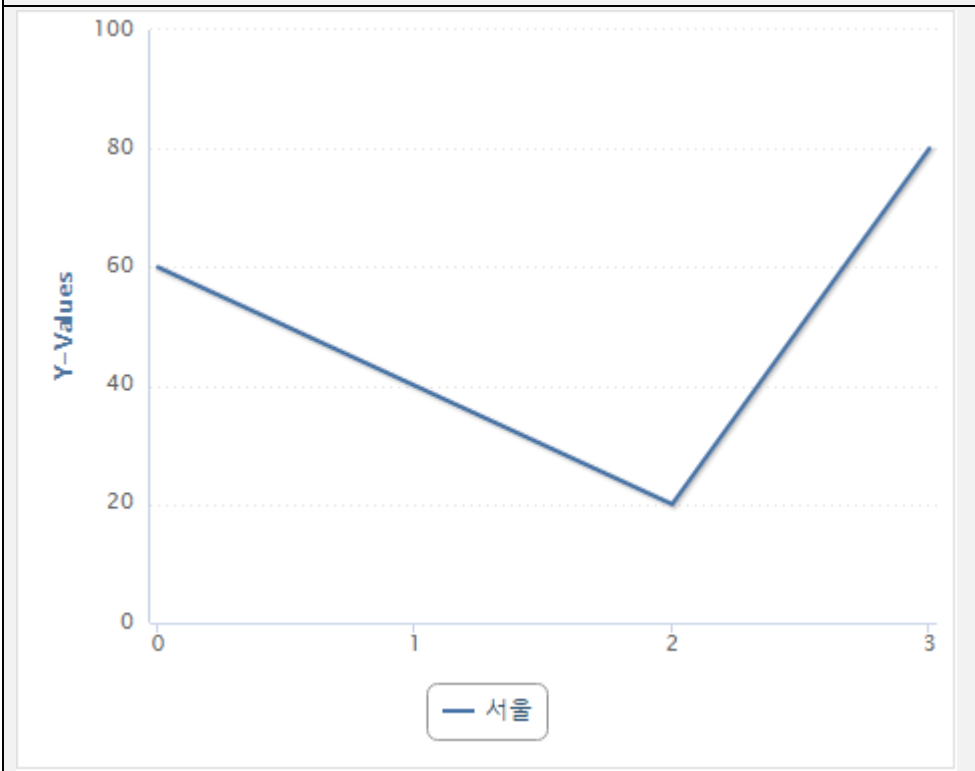
■ marker.states.hover / markerstates.select Info

name	type	required	description
enabled	boolean	N(true)	사용 여부
lineColor	color	N	라인 색상
linewidth	number	N(0)	라인 너비
fillColor	color	N	마커 색상

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      marker: {
        enabled: false
      }
    }
  },
  series: [{
```

```
name: "서울",
data : [60,40,20,80]
}]
}, {
append: true,
redraw: true
});
```



➤ **제공 버전**

7.3.0.0

negativeColor

➤ 기능

임계값 아래의 점 색상을 설정 한다.

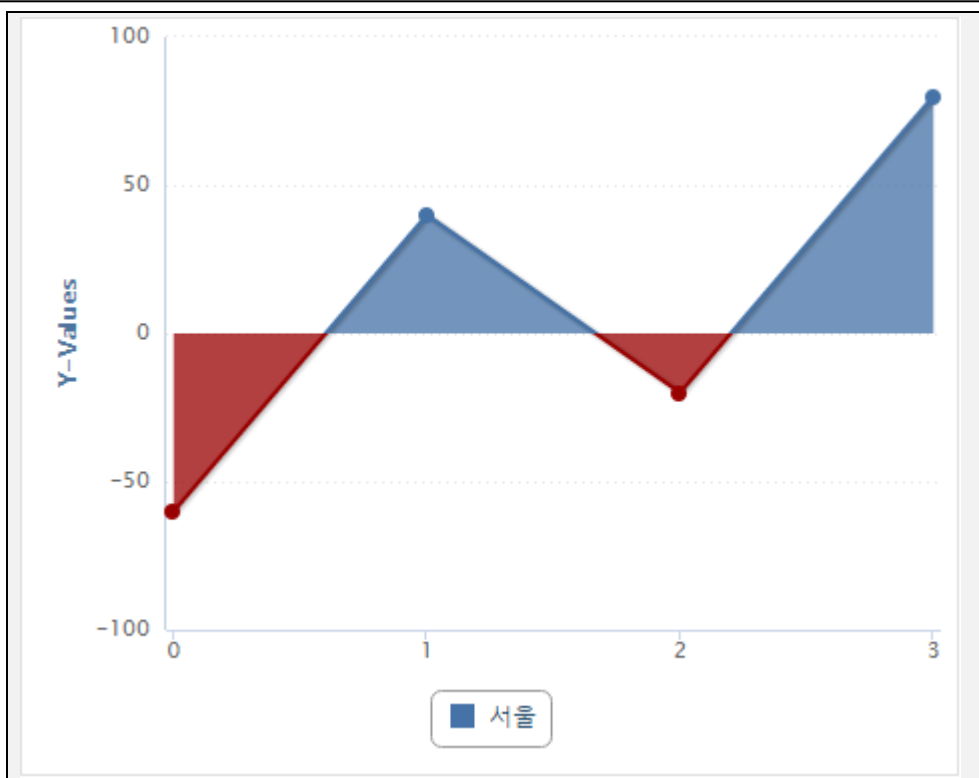
어떤 기준값(예: 0)보다 아래로 내려갔을 경우 색상을 설정하는 기능

➤ Info

name	type	required	description
negativeColor	color	N	임계값 아래의 점 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "area"
  },
  plotOptions: {
    area: {
      negativeColor: "#990000"
    }
  },
  series: [{
    name: "서울",
    data : [-60,40,-20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

negativeFillColor

➤ 기능

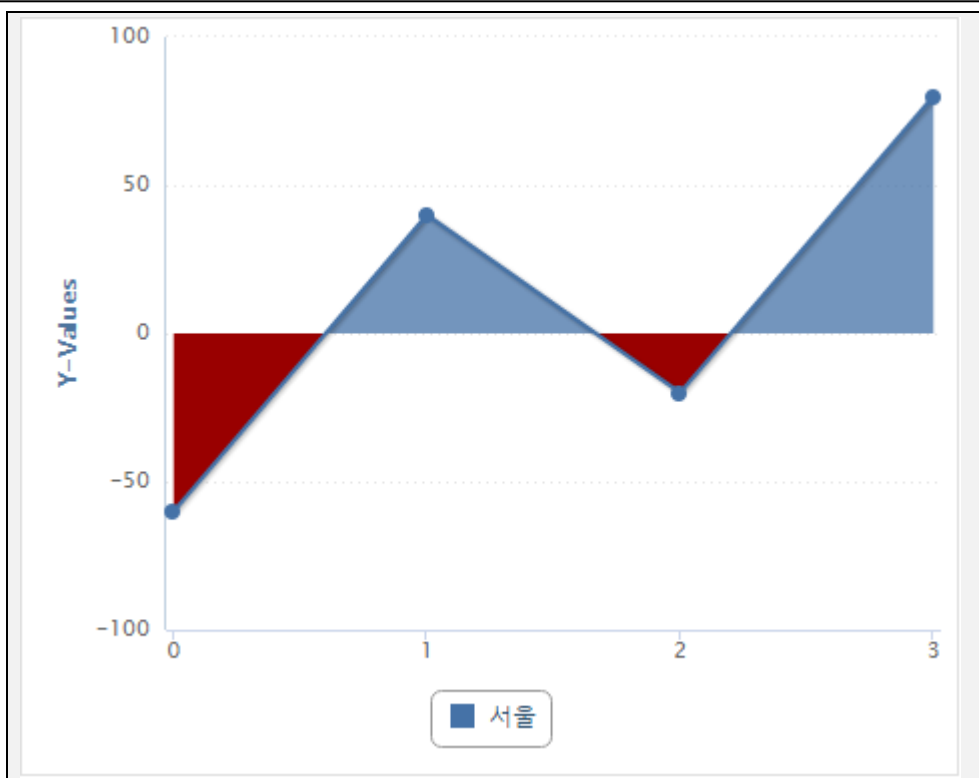
임계값 아래의 배경 색상을 설정 한다.

➤ Info

name	type	required	description
negativeFillColor	color	N	임계값 아래의 배경 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "area"
  },
  plotOptions: {
    area: {
      negativeFillColor: "#990000"
    }
  },
  series: [{
    name: "서울",
    data : [-60,40,-20,80]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0

orientation

➤ 기능

Wordcloud 차트의 단어 회전 방향의 수를 설정한다.

➤ Info

name	type	required	description
orientation	number	N(1)	회전 방향 수

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      orientation: 2
    }
  }
}, {
  append: false
});

//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 60
      }
    ]
  }
}
```

```
        ]],
      }, {
        "series": [{
          "pointName": "아이비리더스",
          "value": 50
        }],
      }, {
        "series": [{
          "pointName": "IB Sheet 7",
          "value": 40
        }],
      }, {
        "series": [{
          "pointName": "IBChart (H) 7",
          "value": 30
        }],
      }
    ]
  }, {
    append: true
  });
```



➤ **제공 버전**

7.3.1.0	wordcloud 차트에만 적용가능
----------------	---------------------

pointStart

➤ 기능

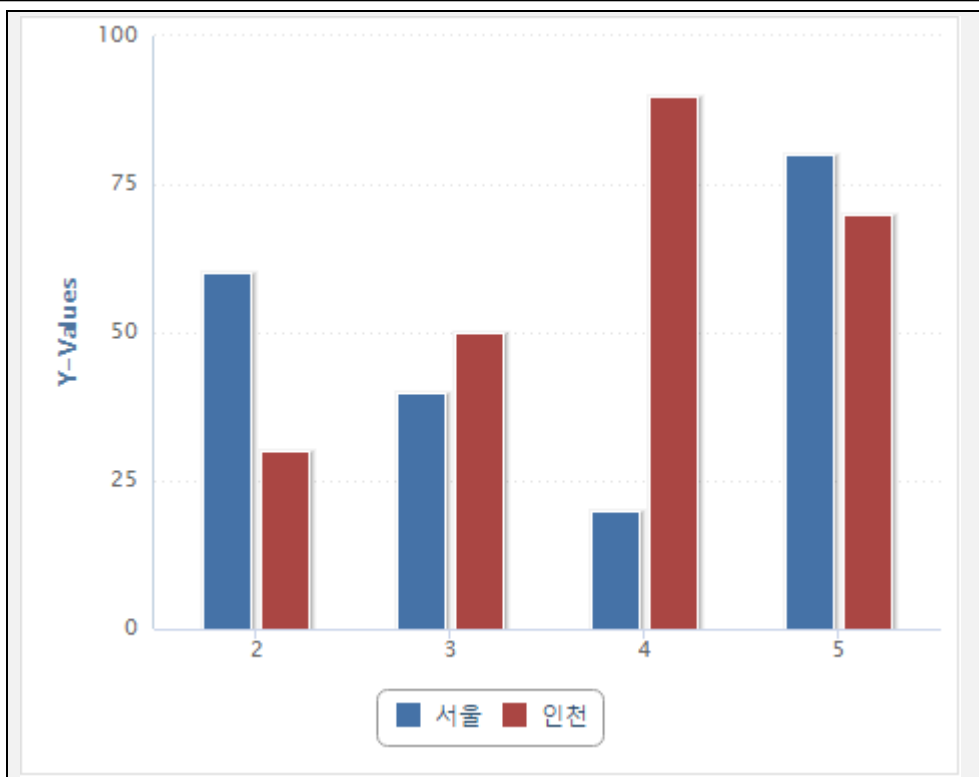
포인트 시작 점을 설정 한다.

➤ Info

name	type	required	description
pointStart	number	N(0)	포인트 시작 점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      pointStart: 2
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

pointPadding

➤ 기능

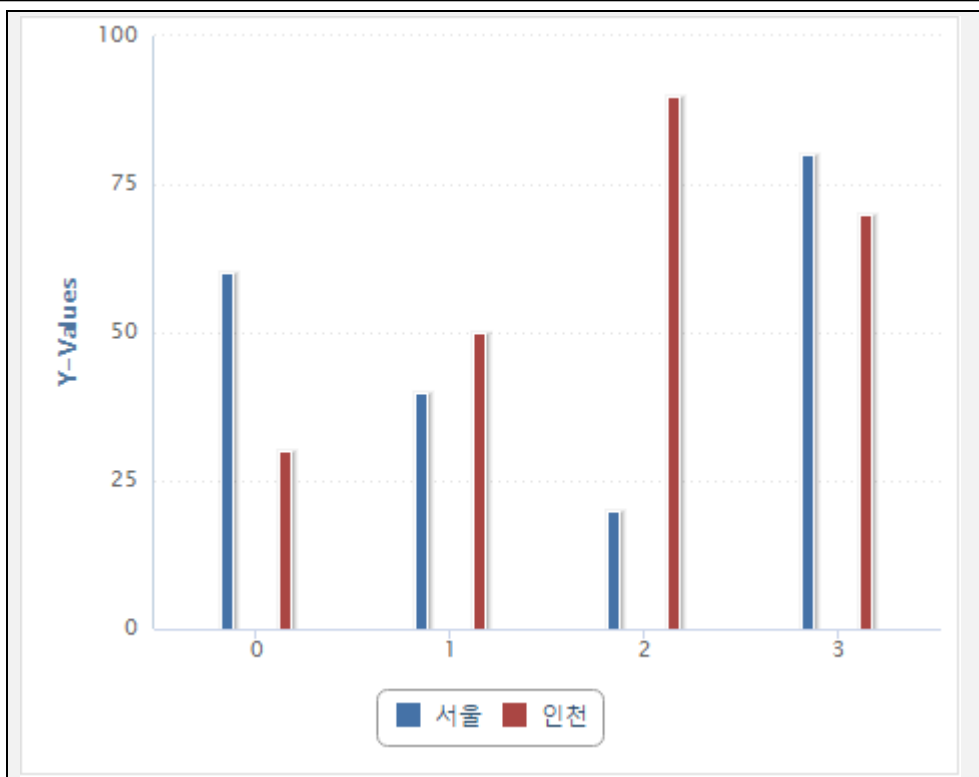
포인트 간 padding 적용 값을 설정 한다.

➤ Info

name	type	required	description
pointPadding	number	N(0.1)	포인트 간 padding 적용 값

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  plotOptions: {  
    column: {  
      pointPadding: 0.4  
    }  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

pointPlacement

➤ 기능

포인트의 위치를 설정 한다.

➤ Info

name	type	required	description
pointPlacement	string	N	포인트 위치

➤ Enum

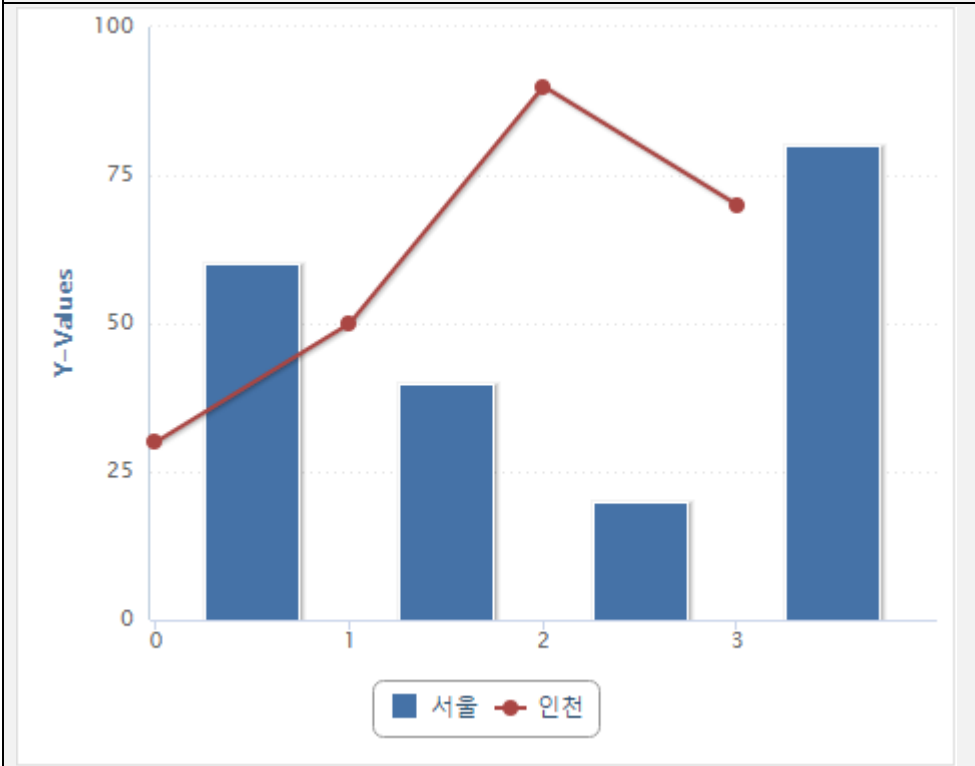
■ pointPlacement

value	description
on	축의 점 위치
between	축의 점간 사이 위치

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      pointPlacement: "between" //컬럼차트 속성은 포인트와 포인트 사이
    },
    line : {
      pointPlacement: "on" //라인차트 속성은 x축 포인트 위에
    }
  },
  series: [{
    type: "column",
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```

```
data : [30,50,90,70]
}}
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

pointWidth

➤ 기능

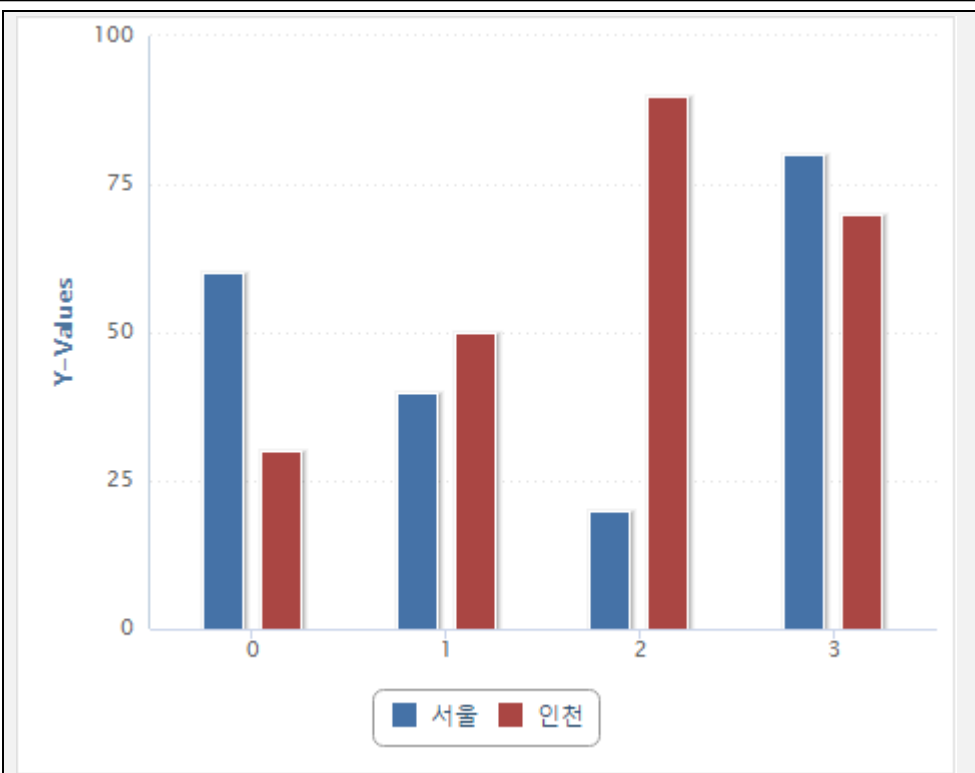
포인트의 너비를 설정 한다.

➤ Info

name	type	required	description
pointWidth	number	N	포인트 너비

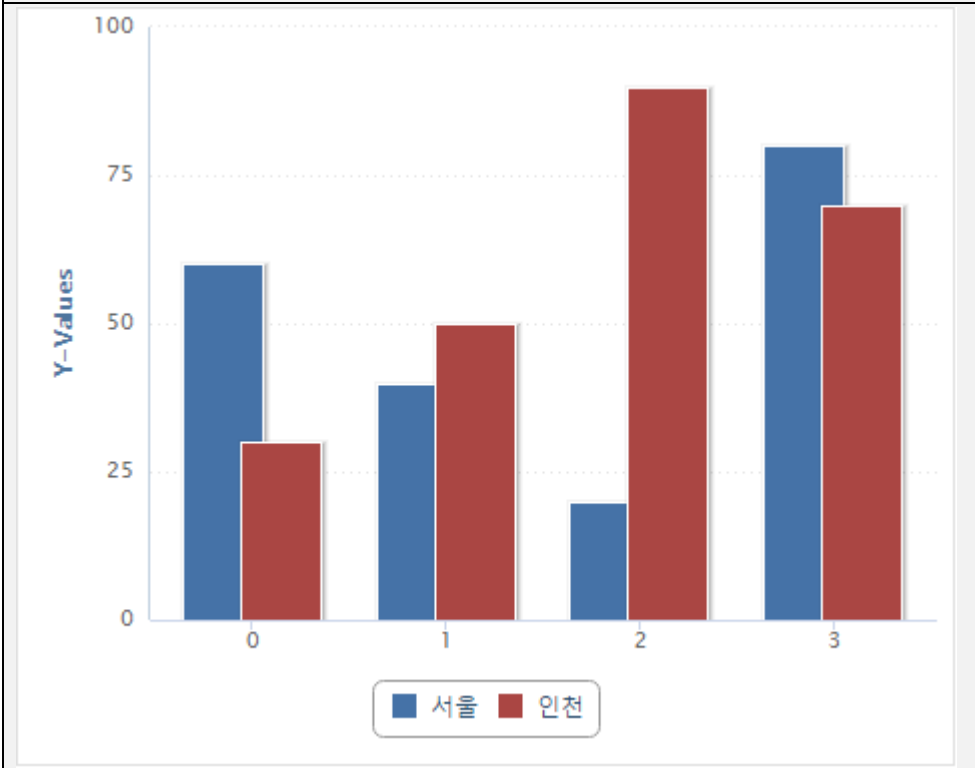
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      pointWidth: 20
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  }]
}, {
  append: true,
  redraw: true
});
```



```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      pointWidth: 40
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }]
})
```

```
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0

rotation

➤ 기능

Wordcloud 차트의 단어 회전 범위를 설정한다.

➤ Info

name	type	required	description
rotation	array<number>	N([0,90])	회전 각도 범위의 시작과 끝을 설정

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      orientation: 2,
      rotation: [-45,0]
    }
  }
}, {
  append: false
});

//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
```

```
        "pointName": "IBLeaders",
        "value": 60
    }], {
        "series": [{
            "pointName": "아이비리더스",
            "value": 50
        }], {
        "series": [{
            "pointName": "IB Sheet 7",
            "value": 40
        }], {
        "series": [{
            "pointName": "IBChart (H) 7",
            "value": 30
        }], {
        "append": true
    });
```



➤ 제공 버전

7.3.1.0	wordcloud 차트에만 적용가능
---------	---------------------

shadow

➤ 기능

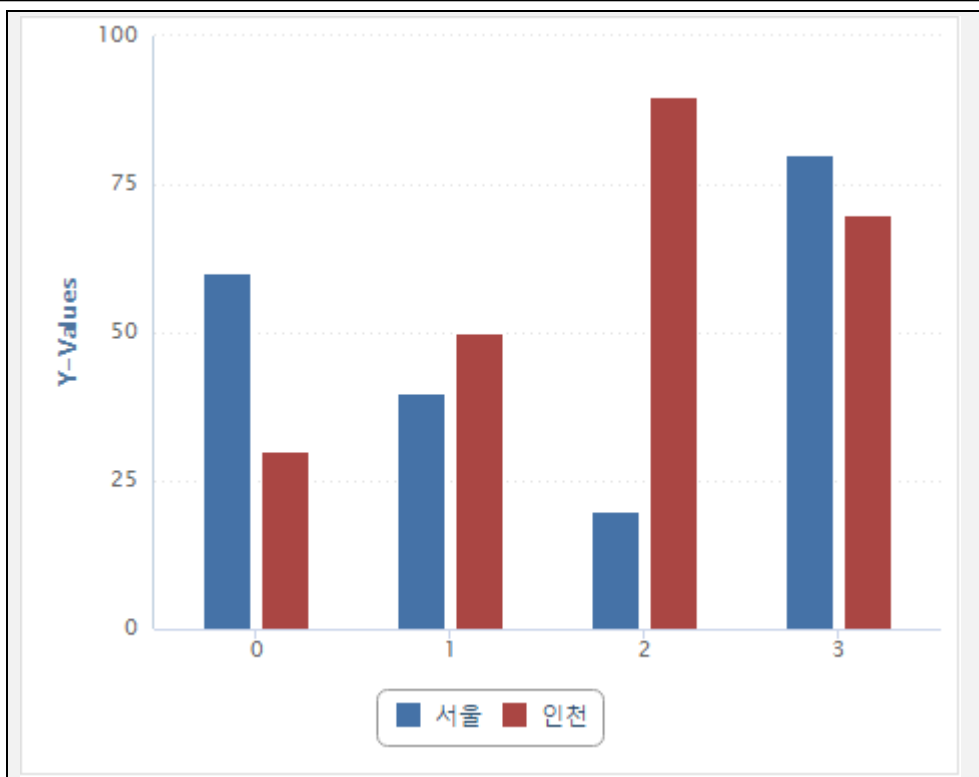
시리즈의 그림자 표시 여부를 설정 한다.

➤ Info

name	type	required	description
shadow	boolean	N(true)	그림자 표시 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    series: {
      shadow: false
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

showInLegend

➤ 기능

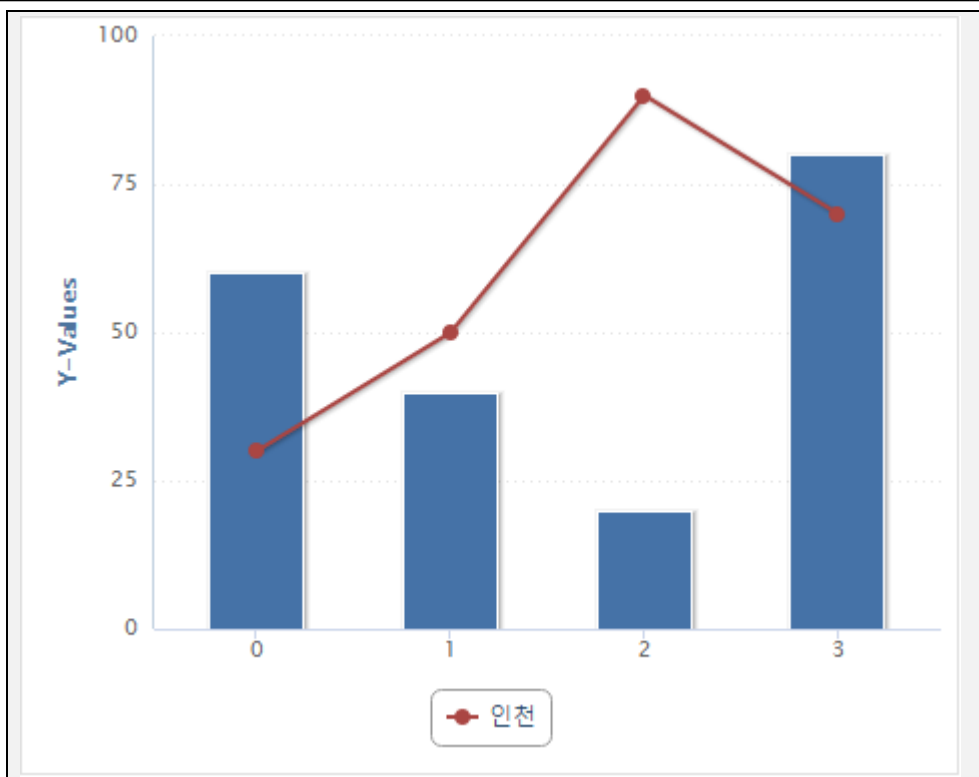
해당 시리즈 정보를 범례에 포함 할지 여부를 설정 한다.

➤ Info

name	type	required	description
showInLegend	boolean	N(true)	범례 포함 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      showInLegend: false
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    type: "line",
    name: "인천",
    data : [30,50,90,70]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

stacking

➤ 기능

시리즈의 데이터 stack 방법을 설정 한다.

➤ Info

name	type	required	description
stacking	string	N	stack 방법

➤ Enum

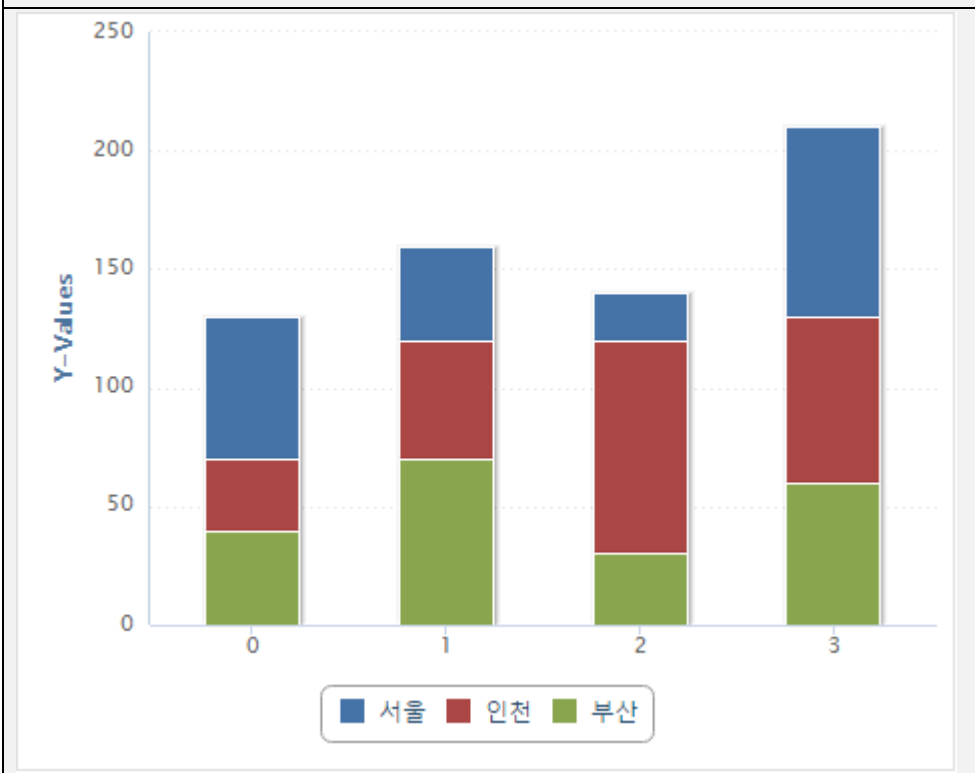
■ stacking

value	description
null	사용 안함
normal	일반 stack 방법
percent	비율 stack 방법

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  plotOptions: {
    column: {
      stacking: "normal"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```

```
data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
----------------	--

states

➤ 기능

시리즈 선의 상태 속성을 설정 한다.

➤ Info

name	type	required	description
states	object	N	시리즈의 상태 속성
states.hover	object	N	마우스 hover 시 속성

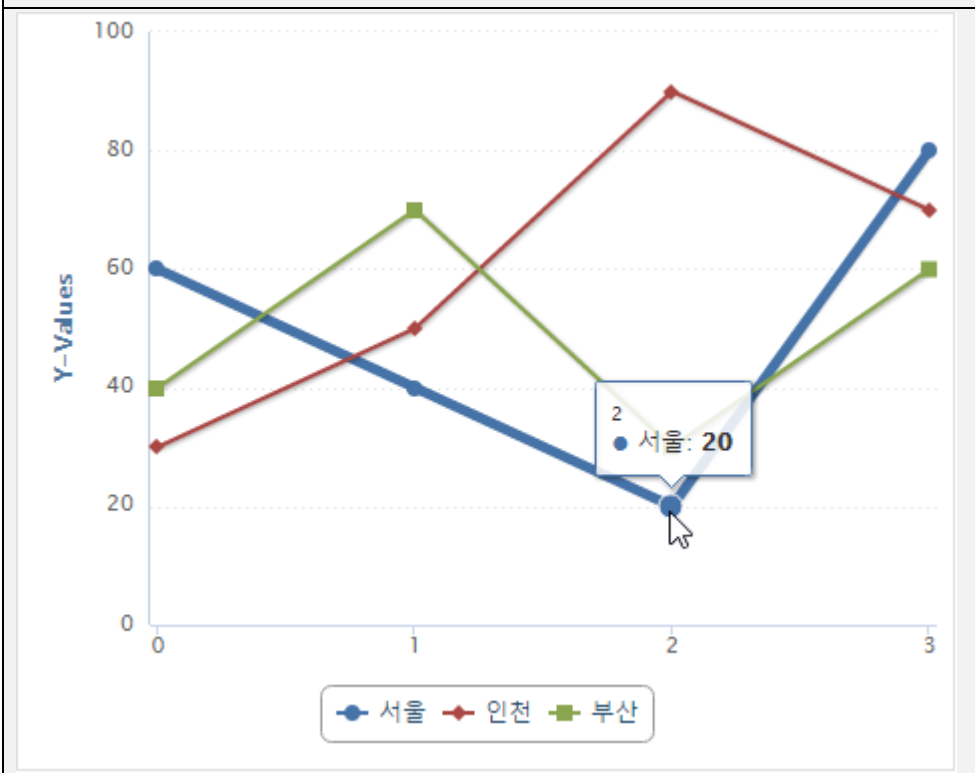
■ states.hover Info

name	type	required	description
enabled	boolean	N(true)	사용 여부
lineWidth	number	N(2)	라인 너비

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      states: {
        hover: {
          lineWidth: 5
        }
      }
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```

```
data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

startAngle

➤ 기능

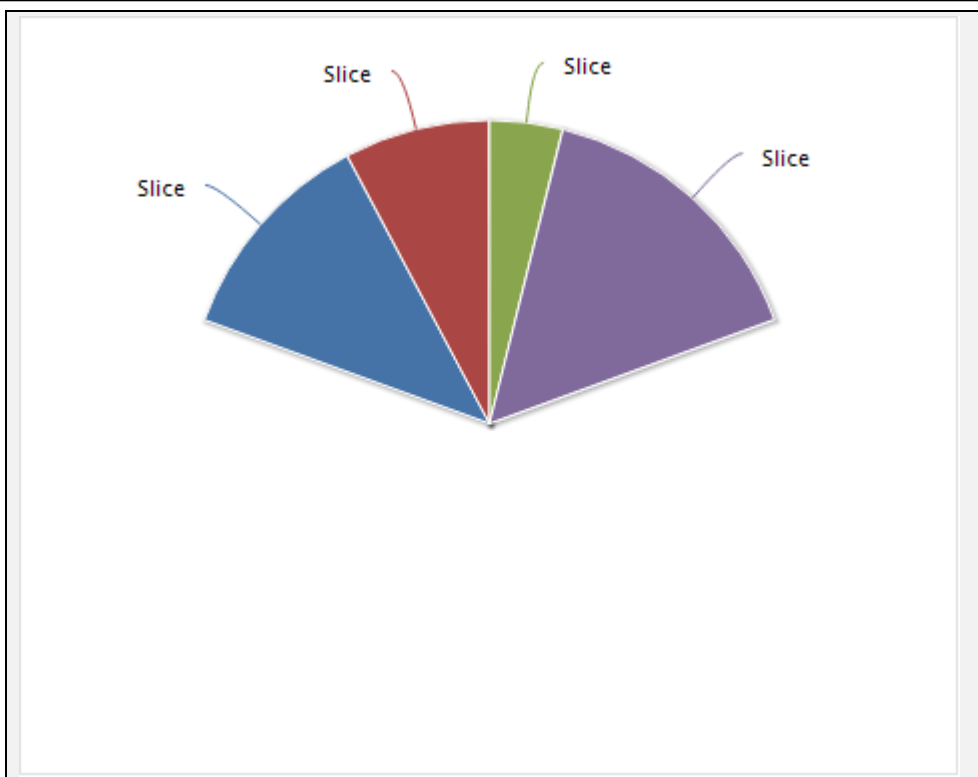
pie 와 같은 유형의 차트에 대한 앵글의 시작 점을 설정 한다.

➤ Info

name	type	required	description
startAngle	number	N	앵글의 시작 점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "pie"
  },
  plotOptions: {
    pie: {
      startAngle: -70,
      endAngle: 70
    }
  },
  series: [{
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

scaleType

➤ 기능

Wordcloud 차트의 입력된 데이터를 정규화하는 방식을 설정한다.

➤ Info

name	type	required	description
scaleType	string	N("linear")	입력데이터 정규화 방식

➤ Enum

■ scaleType

value	description
"none"	사용안함.
"log"	로그형
"sqrt"	제곱형
"linear"	선형

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      scaleType: "none"
    }
  }
}, {
  append: false
});
```

```
//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 16
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 15
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 14
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 13
      }]
    }
  ]
}, {
  append: true
});
```

아이비리더스
IBLeaders
IB Sheet 7
IBChart(H) 7

➤ **제공 버전**

7.3.1.0

wordcloud 차트에만 적용가능

step

➤ 기능

포인트간 선에 대한 단계 표현 방법을 설정 한다.

➤ Info

name	type	required	description
step	string boolean	N(left false)	단계 표현 방법

➤ Enum

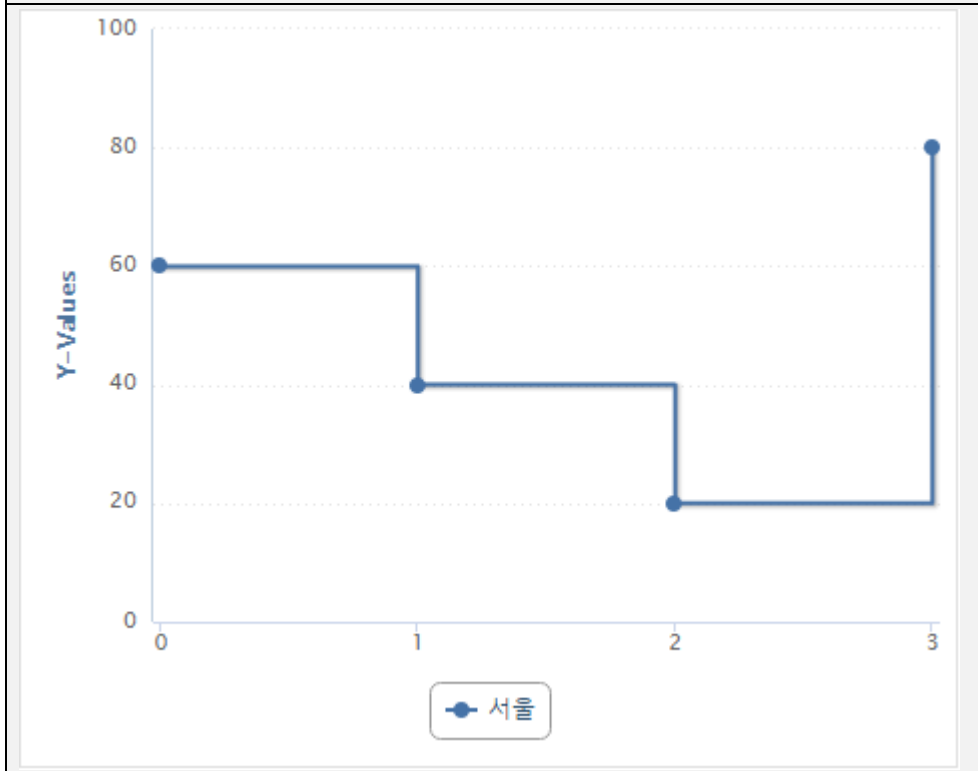
■ step

value	description
null	사용 안함
left	좌측 단계 표현 방법
center	중간 단계 표현 방법
right	우측 단계 표현 방법

➤ Example

```
myChart.setOptions({
  plotOptions: {
    line: {
      step: true
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }]
}, {
  append: true,
  redraw: true
})
```

```
});
```



➤ **제공 버전**

7.3.0.0

spiralType

➤ 기능

Wordcloud 차트의 단어 배치 종류를 설정한다.

➤ Info

name	type	required	description
spiralType	string	N("archimedean")	단어 배치 종류

➤ Enum

■ spiralType

value	description
"rectangular"	사각형으로 배치
"archimedean"	원형으로 배치

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      spiralType: "rectangular"
    }
  }
}, {
  append: false
});

//wordcloud 데이터 로드
```



```
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 60
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 30
      }]
    }
  ]
}, {
  append: true
});
```

➤ **제공 버전**

7.3.1.0

wordcloud 차트에만 적용가능

placeType

➤ 기능

Wordcloud 차트의 입력된 데이터를 spiralType에 맞게 위치하는 방식을 설정한다.

➤ Info

name	type	required	description
placeType	string	N("linear")	단어 위치 방식

➤ Enum

■ placeType

value	description
"linear"	선형(내림차순)
"random"	임의 위치(차트 draw()와 down2Image() 할때마다 위치가 바뀜)

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      placeType: "random"
    }
  }
}, {
  append: false
});
```

```
//wordcloud 데이터 로드
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 60
      }]
    }, {
      "series": [{
        "pointName": "아이비리더스",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "IB Sheet 7",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "IBChart(H) 7",
        "value": 30
      }]
    }
  ]
}, {
  append: true
});
```

IBLeaders

IB Sheet 7

아이비리더스

IBChart(H) 7

➤ 제공 버전

7.3.1.0

wordcloud 차트에만 적용가능

valueRange

➤ 기능

Wordcloud 차트의 입력된 데이터(글자)의 크기를 설정한다.

➤ Info

name	type	required	description
valueRange	array<number> string	N([15,60])	단어 최소/최대 크기 지정

➤ Enum

■ valueRange

value	description
[15,60]	고정형 ([최소 크기, 최대 크기]를 숫자로 지정, ※ 차트의 크기에 따라 최대값을 바꿔줘야함 - 최소값과 최대값이 차트의 크기에 비례해 큰 값이 설정될 경우나 최소값과 최대값의 차이가 단어의 갯수에 비해 작을 경우 성능이 저하될 수 있음)
"responsive"	반응형 (차트의 크기에 따라 글자 크기를 변경하고 줌인/줌아웃)

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      valueRange: [15,60]
    }
  }
})
```

```
}, {  
    append: false  
});  
  
//wordcloud 데이터 로드  
myChart.loadSearchData({  
    "ibchart": {  
        "data": [{  
            "series": [{  
                "seriesName": "WordCloud",  
                "pointName": "IBLeaders",  
                "value": 60  
            }]  
        }, {  
            "series": [{  
                "pointName": "아이비리더스",  
                "value": 50  
            }]  
        }, {  
            "series": [{  
                "pointName": "IB Sheet 7",  
                "value": 40  
            }]  
        }, {  
            "series": [{  
                "pointName": "IBChart(H) 7",  
                "value": 30  
            }]  
        }  
    }  
}, {  
    append: true  
});
```

```
});
```

IBLeaders
IB Sheet 7
아이비리더스
IBChart(H) 7

➤ 제공 버전

7.3.1.6

wordcloud 차트에만 적용가능

style

➤ 기능

wordcloud 차트의 단어 스타일을 설정 한다.

➤ Info

name	type	required	description
style.fontFamily	string	N	폰트 종류 (클라이언트 사용자에게 있는 폰트만 유효함(웹폰트 포함), Export 시 서버에 해당 폰트가 있어야 동일하게 출력이 가능함)
style. fontPadding	number	N(1)	단어간 여백
style.fontWeight	string	N ("bold")	폰트 두께

➤ Enum

■ Style.fontWeight

value	description
"normal"	폰트 두께를 얇게 설정
"bold"	폰트 두께를 굵게 설정

➤ Example

```
//wordcloud 차트 설정
myChart.setOptions({
  chart: {
    type: "wordcloud"
  },
  plotOptions: {
    wordcloud: {
      style: {
        fontFamily: "맑은 고딕",
```



```

        fontWeight: "",
        fontPadding: 0.2
    }
}

}, {
    append: false
});

//wordcloud 데이터 로드
myChart.loadSearchData({
    "ibchart": {
        "data": [{
            "series": [{
                "seriesName": "WordCloud",
                "pointName": "IBLeaders",
                "value": 60
            }]
        }, {
            "series": [{
                "pointName": "아이비리더스",
                "value": 50
            }]
        }, {
            "series": [{
                "pointName": "IB Sheet 7",
                "value": 40
            }]
        }, {
            "series": [{
                "pointName": "IBChart(H) 7",
                "value": 30
            }]
        }
    ]
});

```

```

    ]]
  ]]
}
}, {
  append: true
});

```

아이비리더스
IBLeaders
IB Sheet 7
IBChart(H) 7

➤ **제공 버전**

7.3.1.0

wordcloud 차트에만 적용가능

visible

➤ 기능

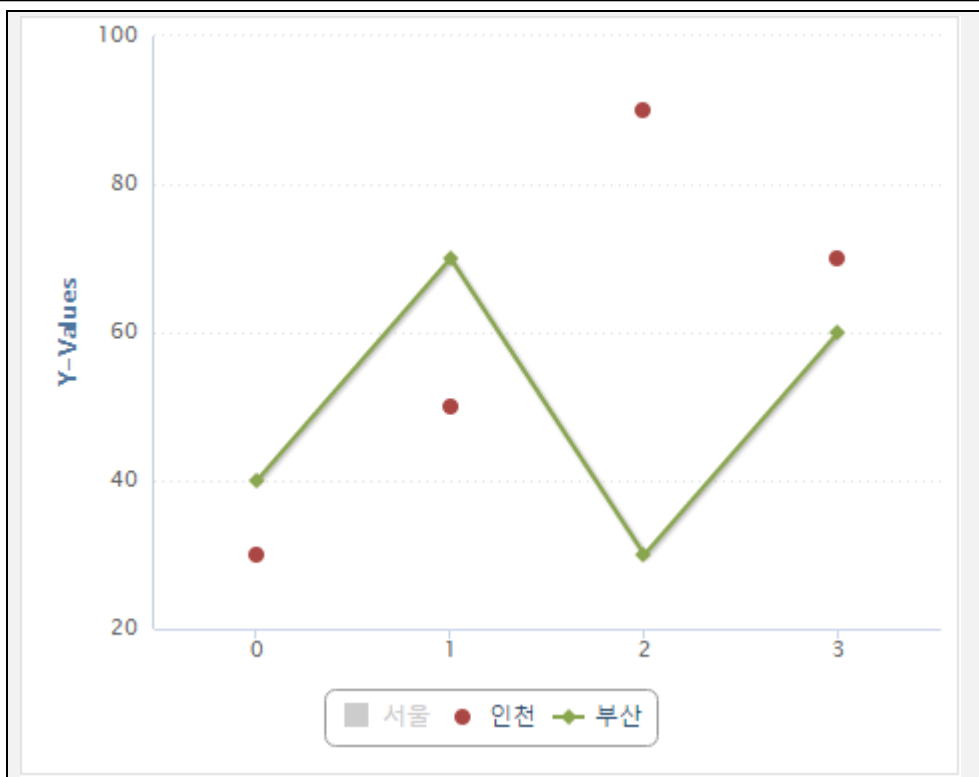
시리즈의 보임/숨김 여부를 설정 한다.

➤ Info

name	type	required	description
visible	boolean	N(true)	보임/숨김 여부

➤ Example

```
myChart.setOptions({
  plotOptions: {
    column: {
      visible: false
    }
  },
  series: [{
    type: "column",
    name: "서울",
    data : [60,40,20,80]
  }, {
    type: "scatter",
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

10.4 subtitle

부제목에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
text	string	N	제목 문자열
align	string	N(center)	좌우 정렬 값
verticalAlign	string	N	상하 정렬 값
x	number	N(0)	좌우 포지션 이동 값
y	number	N	상하 포지션 이동 값
style	object	N	스타일 설정
style.color	color	N	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께
useHTML	boolean	N(false)	제목 문자열에 html tag 사용 여부

text

➤ 기능

제목 문자열을 설정 한다.

➤ Info

name	type	required	description
text	string	N	제목 문자열

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0	
---------	--

align

➤ 기능

좌우 정렬을 설정 한다.

➤ Info

name	type	required	description
align	string	N(center)	좌우 정렬 값

➤ Enum

■ align

value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기",
    align: "left"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량

2014년 1/4분기

➤ 제공 버전

7.3.0.0

verticalAlign

➤ 기능

상하 정렬을 설정 한다.

➤ Info

name	type	required	description
verticalAlign	string	N	상하 정렬 값

■ verticalAlign

value	description
top	상단 정렬
middle	중앙 정렬
bottom	하단 정렬

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기",
    verticalAlign: "middle"
  }
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0	
----------------	--

x

➤ 기능

좌우 포지션 이동 값을 설정 한다.

➤ Info

name	type	required	description
x	number	N(0)	좌우 포지션 이동 값

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기",
    x: 40
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

y

➤ 기능

상하 포지션 이동 값을설정 한다.

➤ Info

name	type	required	description
y	number	N	상하 포지션 이동 값

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기",
    y: 25
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

style

➤ 기능

스타일을 설정 한다.

➤ Info

name	type	required	description
style.color	color	N	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기",
    style: {
      color: "red",
      fontSize: "1.2em",
      fontWeight: "bold"
    }
  }, {
    append: true,
    redraw: true
  });
```

cc별 중고자동차 판매량

2014년 1/4분기

➤ 제공 버전

7.3.0.0	
---------	--

useHTML

➤ 기능

제목 문자열에 html tag 사용 여부를 설정 한다.

➤ Info

name	type	required	description
useHTML	boolean	N(false)	제목 문자열에 html tag 사용 여부

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "<span style='font-size: 1.2em; font-weight: bold;
'>2014년</span> 1/4분기",
    useHTML: true
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량

2014년 1/4분기

➤ 제공 버전

7.3.0.0	
---------	--

10.5 title

제목에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
text	string	N	제목 문자열
align	string	N(center)	좌우 정렬 값
verticalAlign	string	N	상하 정렬 값
x	number	N(0)	좌우 포지션 이동 값
y	number	N	상하 포지션 이동 값
style	object	N	스타일 설정
style.color	color	N	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께
style.fontFamily	string	N	폰트 글꼴
useHTML	boolean	N(false)	제목 문자열에 html tag 사용 여부

text

➤ 기능

제목 문자열을 설정 한다.

➤ Info

name	type	required	description
text	string	N	제목 문자열

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량"
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

align

➤ 기능

좌우 정렬을 설정 한다.

➤ Info

name	type	required	description
align	string	N(center)	좌우 정렬 값

➤ Enum

■ align

value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text:"cc별 중고자동차 판매량",
    align: "right"
  },
  subtitle: { // subtitle 설정
    text:"2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량

2014년 1/4분기

➤ 제공 버전

7.3.0.0

verticalAlign

➤ 기능

상하 정렬을 설정 한다.

➤ Info

name	type	required	description
verticalAlign	string	N	상하 정렬 값

■ verticalAlign

value	description
top	상단 정렬
middle	중앙 정렬
bottom	하단 정렬

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text:"cc별 중고자동차 판매량",
    verticalAlign: "middle"
  },
  subtitle: { // subtitle 설정
    text:"2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0	
----------------	--

x

➤ 기능

좌우 포지션 이동 값을 설정 한다.

➤ Info

name	type	required	description
x	number	N(0)	좌우 포지션 이동 값

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량",
    x: 40
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

y

➤ 기능

상하 포지션 이동 값을설정 한다.

➤ Info

name	type	required	description
y	number	N	상하 포지션 이동 값

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text:"cc별 중고자동차 판매량",
    y: 25
  },
  subtitle: { // subtitle 설정
    text:"2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

style

➤ 기능

스타일을 설정 한다.

➤ Info

name	type	required	description
style.color	color	N	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께
style.fontFamily	string	N	폰트 글꼴

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text: "cc별 중고자동차 판매량",
    style: {
      color: "#1E90FF",
      fontSize: "1.5em",
      fontWeight: "bold",
      fontFamily: "serif"
    }
  },
  subtitle: { // subtitle 설정
    text: "2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량

2014년 1/4분기

➤ **제공 버전**

7.3.0.0

useHTML

➤ 기능

제목 문자열에 html tag 사용 여부를 설정 한다.

➤ Info

name	type	required	description
useHTML	boolean	N(false)	제목 문자열에 html tag 사용 여부

➤ Example

```
myChart.setOptions({
  title: { // title 설정
    text:"<span style='font-size: 1.2em; font-weight: bold; font-family: serif;'>cc별</span> 중고자동차 판매량",
    useHTML: true
  },
  subtitle: { // subtitle 설정
    text:"2014년 1/4분기"
  }
}, {
  append: true,
  redraw: true
});
```

cc별 중고자동차 판매량
2014년 1/4분기

➤ 제공 버전

7.3.0.0

10.6 legend

범례에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
enabled	boolean	N(true)	범례 표시 여부
align	string	N(center)	가로 정렬 방법
verticalAlign	string	N(bottom)	세로 정렬 방법
backgroundColor	color	N	배경색
borderColor	color	N(#999999)	border 색상
borderWidth	number	N(1)	border의 너비
borderRadius	number	N(0)	radius 적용 값
layout	string	N(horizontal)	배치 방법
title	object	N	제목 속성
title.text	string	N	제목 문자열
title.style	object	N	스타일 속성
x	number	N(0)	좌우 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)
y	number	N(0)	상하 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)
floating	boolean	N(false)	플로팅영역 여부
reversed	boolean	N(false)	시리즈 항목 역순
itemStyle.color	string	N	시리즈 text 색상
itemStyle.fontWeight	string	N	시리즈 text 굵기
itemStyle.fontSize	String	N	시리즈 text 크기
itemWidth	number	N(0)	시리즈 아이템 별 너비
itemDistance	Number	N(20)	시리즈 아이템 별 간격
itemHoverStyle.color	Color	N(#000000)	시리즈 아이템 마우스 호버 시 색상

enabled

➤ 기능

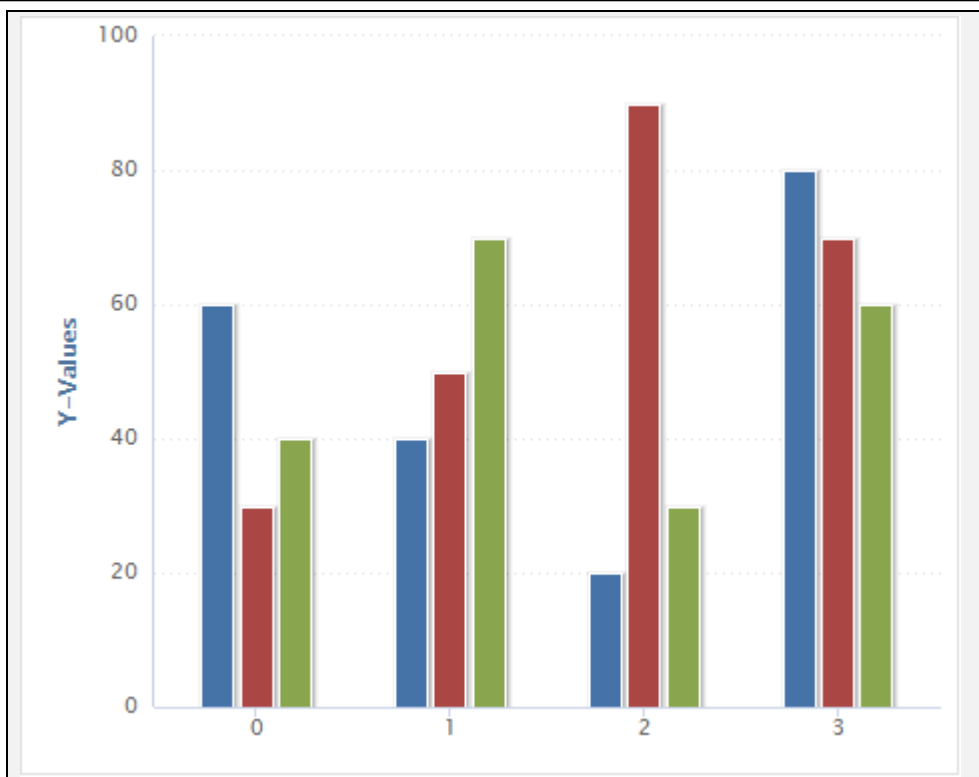
범례 표시 여부를 설정 한다.

➤ Info

name	type	required	description
enabled	boolean	N(true)	범례 표시 여부

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  legend: {  
    enabled: false  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

align

➤ 기능

가로 정렬 방법을 설정 한다.

➤ Info

name	type	required	description
align	string	N(center)	가로 정렬 방법

➤ Enum

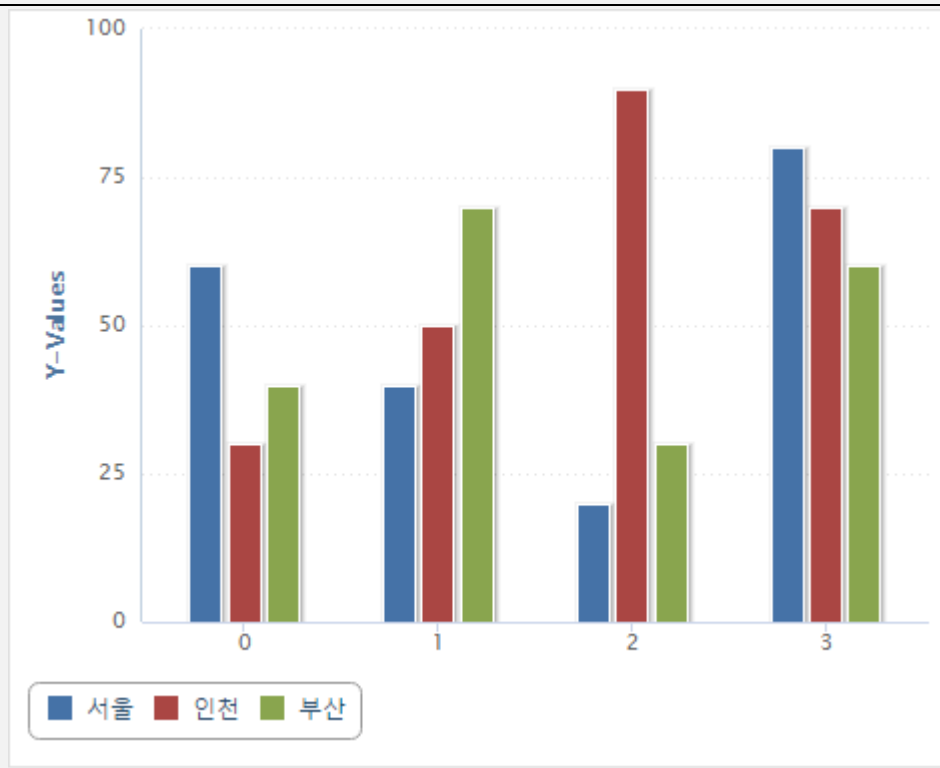
■ align

value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    align: "left"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
```

```
name: "부산",
data : [40,70,30,60]
}]
}, {
append: true,
redraw: true
});
```



➤ **제공 버전**

7.3.0.0

verticalAlign

➤ 기능

세로 정렬 방법을 설정 한다.

➤ Info

name	type	required	description
verticalAlign	string	N(bottom)	세로 정렬 방법

➤ Enum

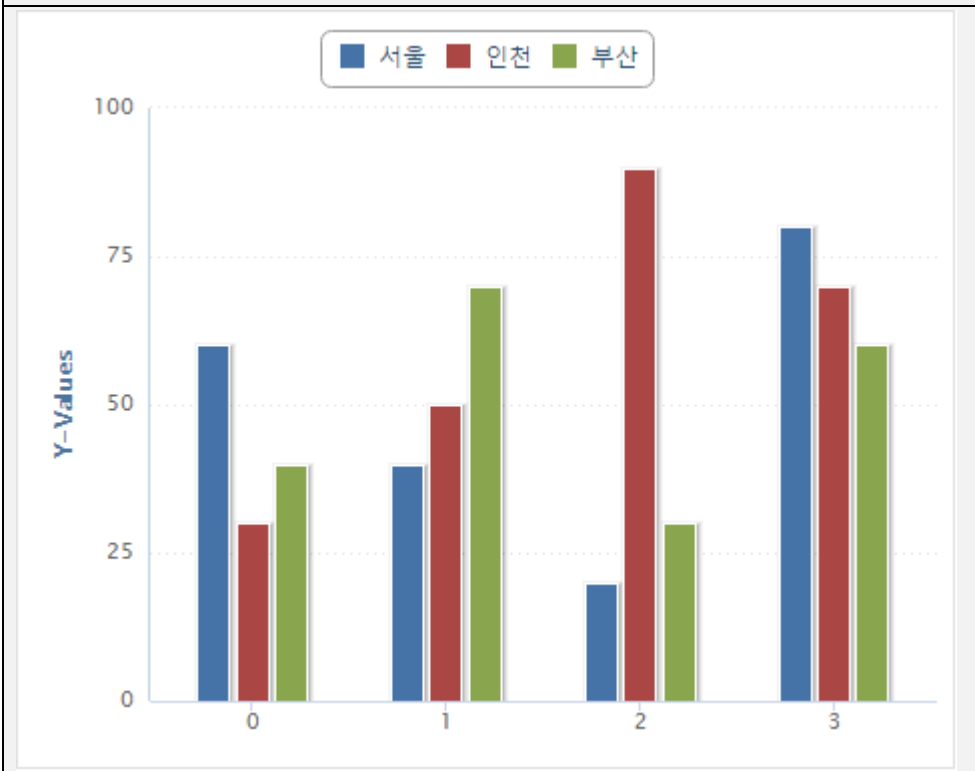
■ verticalAlign

value	description
top	상단 정렬
middle	중앙 정렬
bottom	하단 정렬

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    verticalAlign: "top"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
```

```
name: "부산",
data : [40,70,30,60]
}]
}, {
append: true,
redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

backgroundColor

➤ 기능

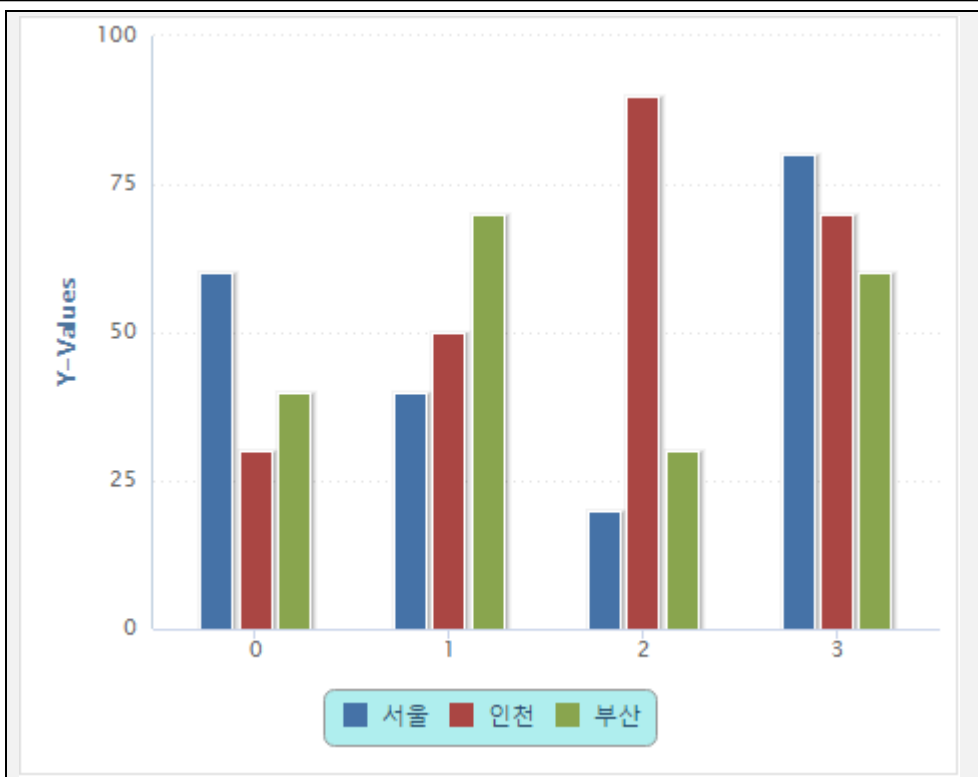
배경색을 설정 한다.

➤ Info

name	type	required	description
backgroundColor	color	N	배경색

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  legend: {  
    backgroundColor: "#AFEEEE"  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

borderColor

➤ 기능

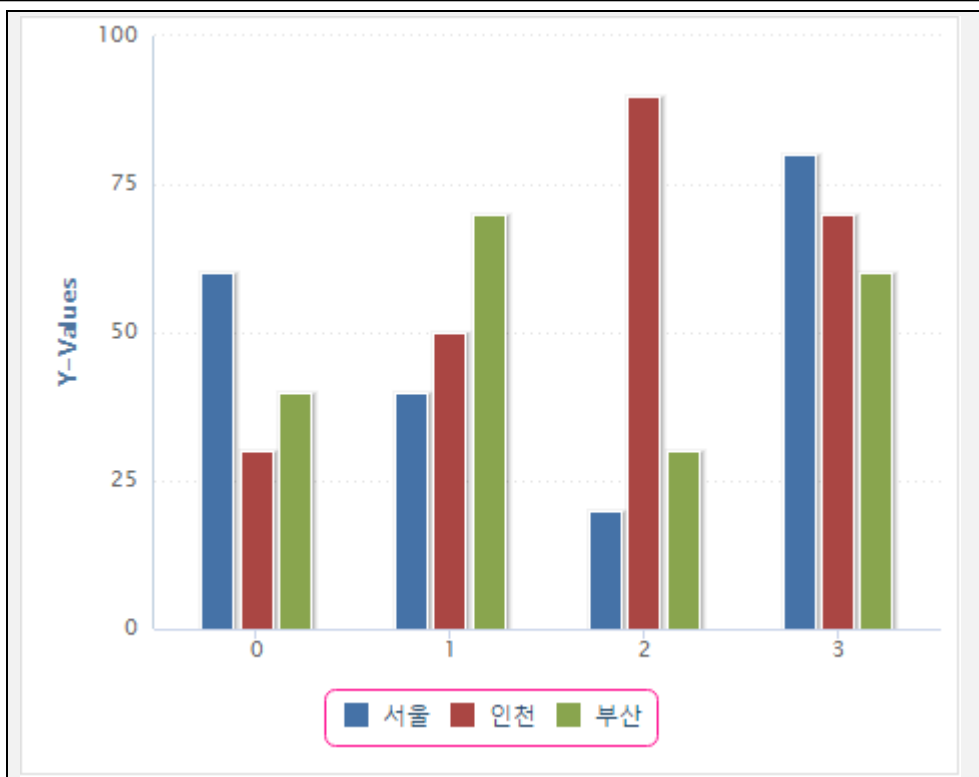
border 색상을 설정 한다.

➤ Info

name	type	required	description
borderColor	color	N(#999999)	border 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    borderColor: "#FF1493"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

borderWidth

➤ 기능

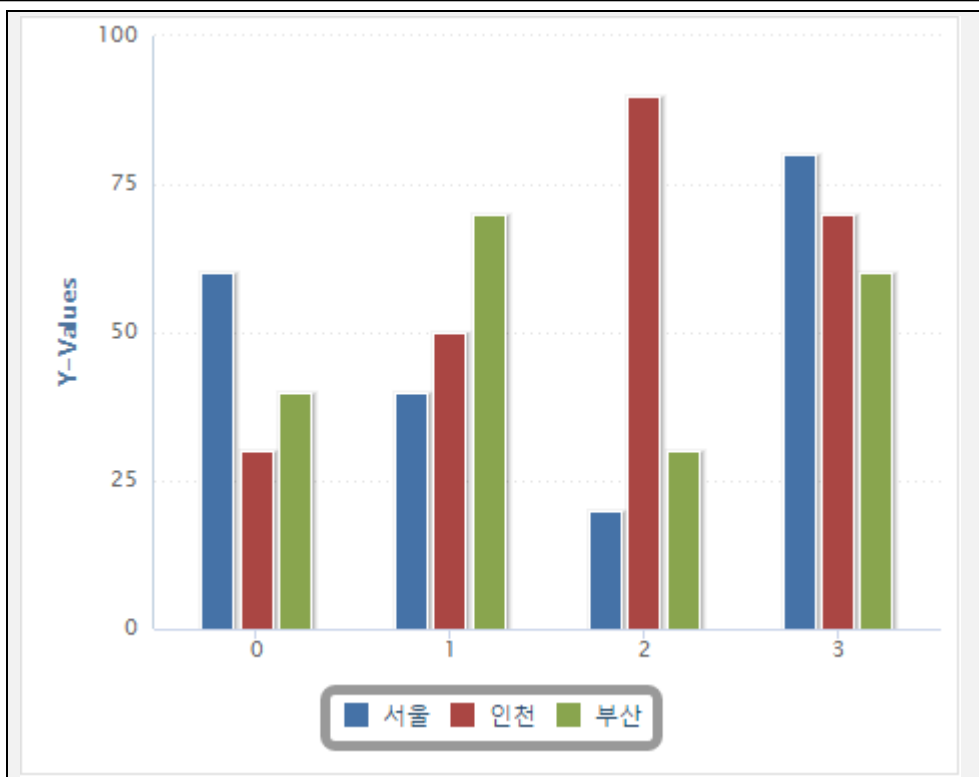
border 의 너비를 설정 한다.

➤ Info

name	type	required	description
borderWidth	number	N(1)	border의 너비

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  legend: {  
    borderWidth: 5  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

borderRadius

➤ 기능

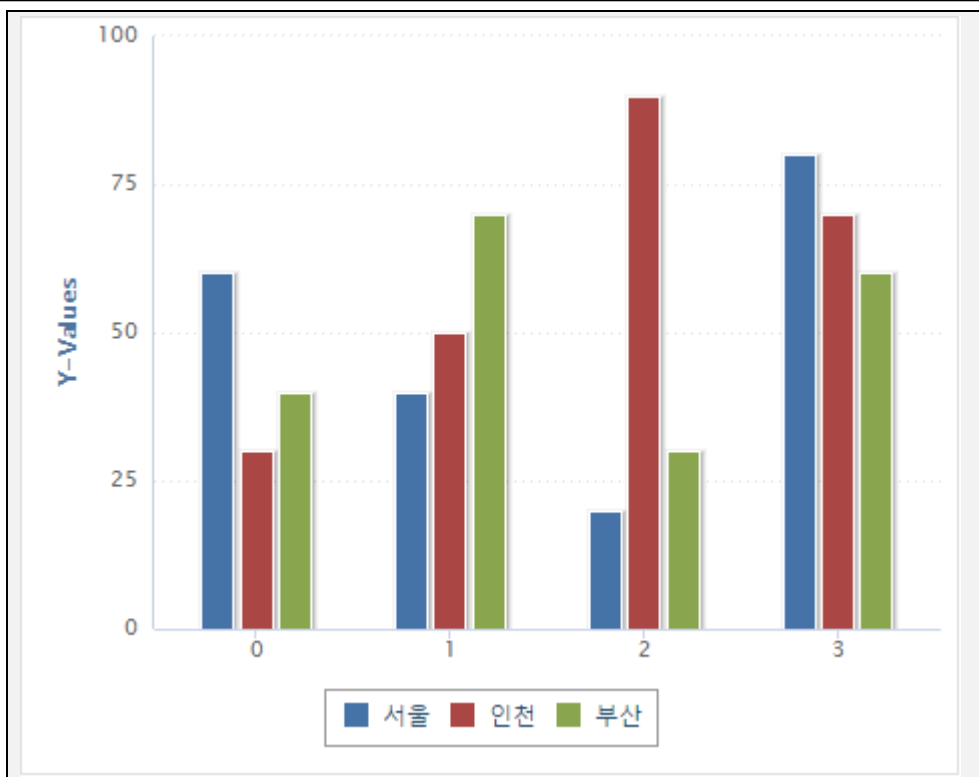
radius 적용 값을 설정 한다.

➤ Info

name	type	required	description
borderRadius	number	N(5)	radius 적용 값

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  legend: {  
    borderRadius: 0  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

layout

➤ 기능

배치 방법을 설정 한다.

➤ Info

name	type	required	description
layout	string	N(horizontal)	배치 방법

➤ Enum

■ layout

value	description
horizontal	가로 나열 배치
vertical	세로 나열 배치

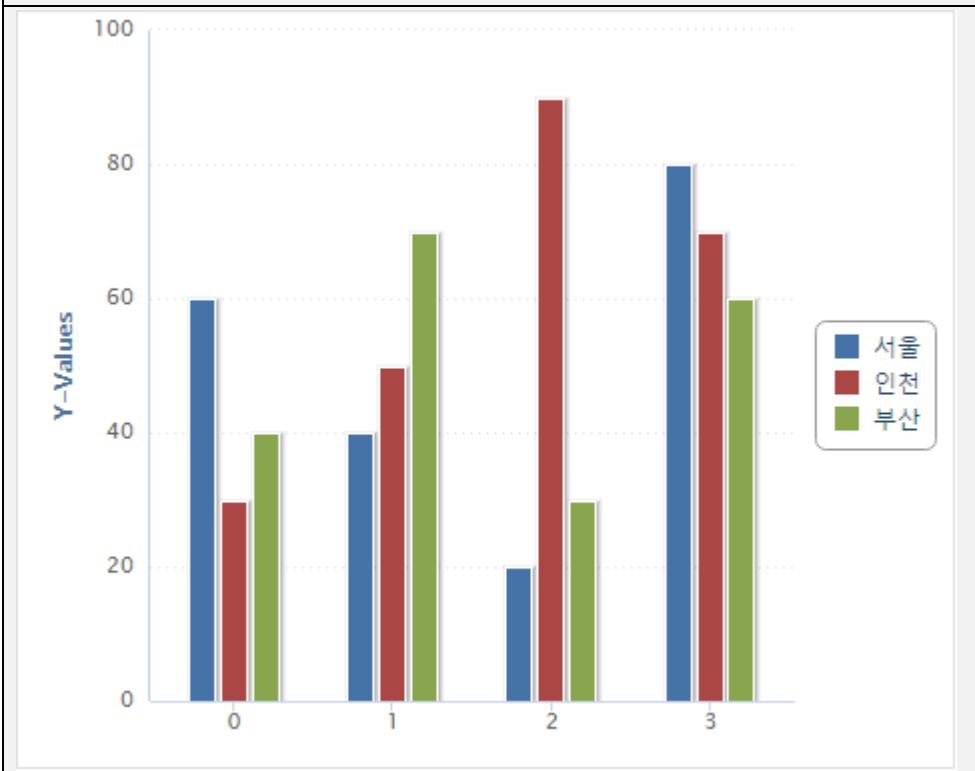
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    layout: "vertical",
    align: "right",
    verticalAlign: "middle"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
```

```

    }, {
      name: "부산",
      data : [40,70,30,60]
    }
  ], {
    append: true,
    redraw: true
  });

```



➤ **제공 버전**

7.3.0.0	
----------------	--

title

➤ 기능

애니메이션 효과 사용 여부를 설정 한다.

➤ Info

name	type	required	description
title	object	N	제목 속성
title.text	string	N	제목 문자열
title.style	object	N	스타일 속성

■ title.style

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기
fontWeight	string	N	폰트 두께

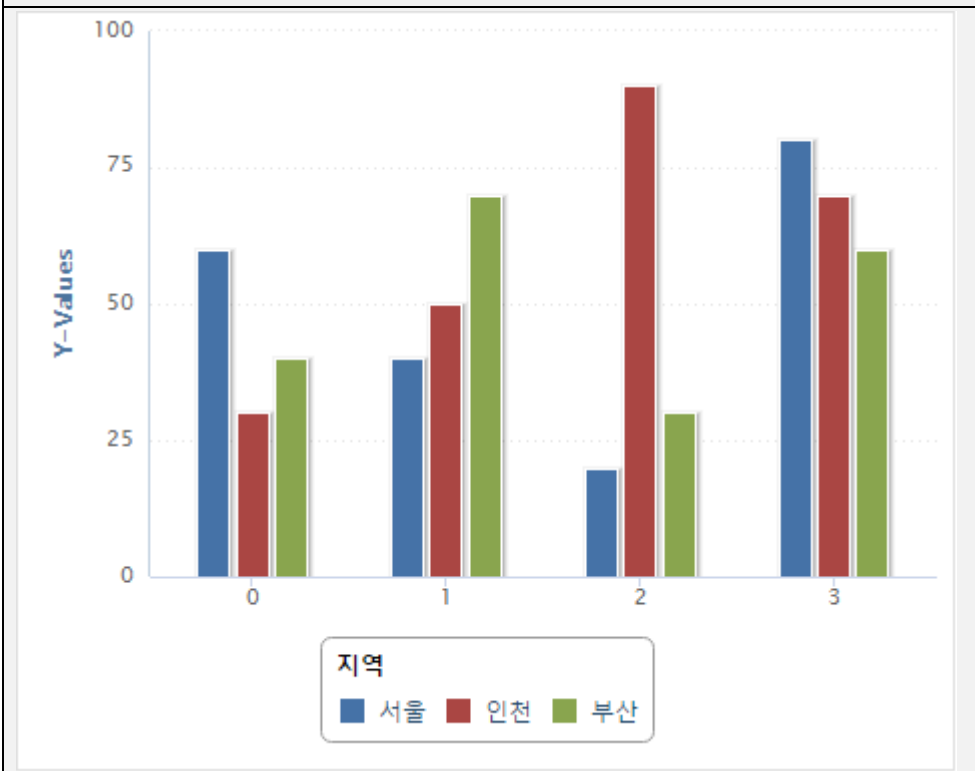
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    title: {
      text: "지역",
      style: {
        fontWeight: "bold"
      }
    }
  },
  series: [{
```

```

name: "서울",
data : [60,40,20,80]
},{
name: "인천",
data : [30,50,90,70]
},{
name: "부산",
data : [40,70,30,60]
}]
}, {
append: true,
redraw: true
});

```



➤ **제공 버전**

7.3.0.0

x

➤ 기능

애니메이션 효과 사용 여부를 설정 한다.

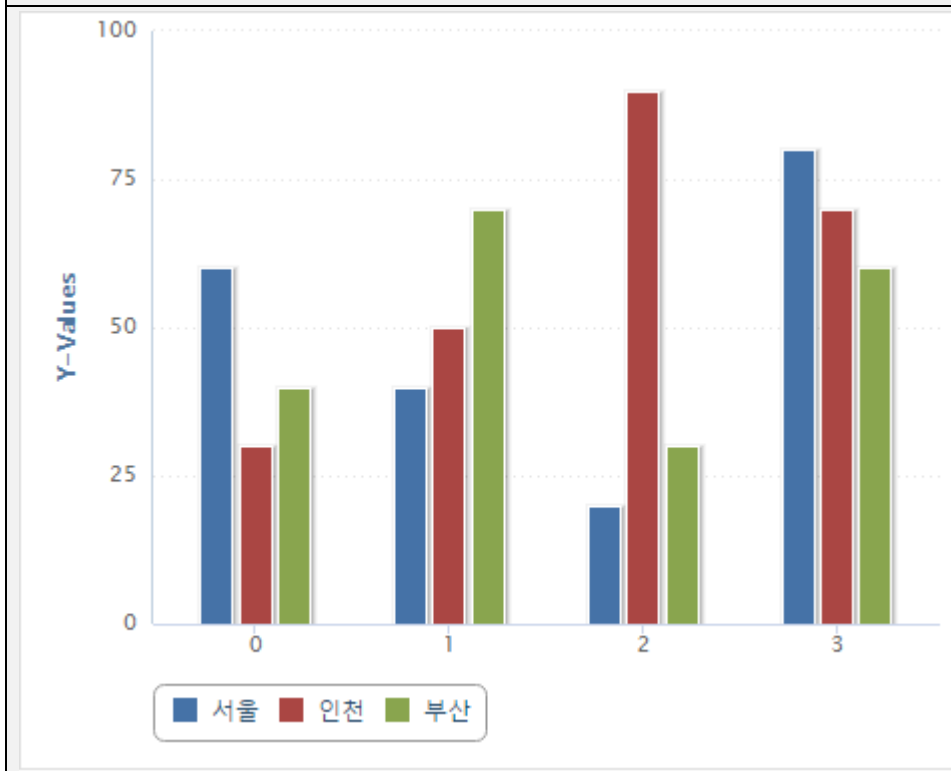
➤ Info

name	type	required	description
x	number	N(0)	좌우 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    align: "left",
    x: 57
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
})
```

});



➤ **제공 버전**

7.3.0.0

y

➤ 기능

상하 포지션 이동 값을 설정 한다.

➤ Info

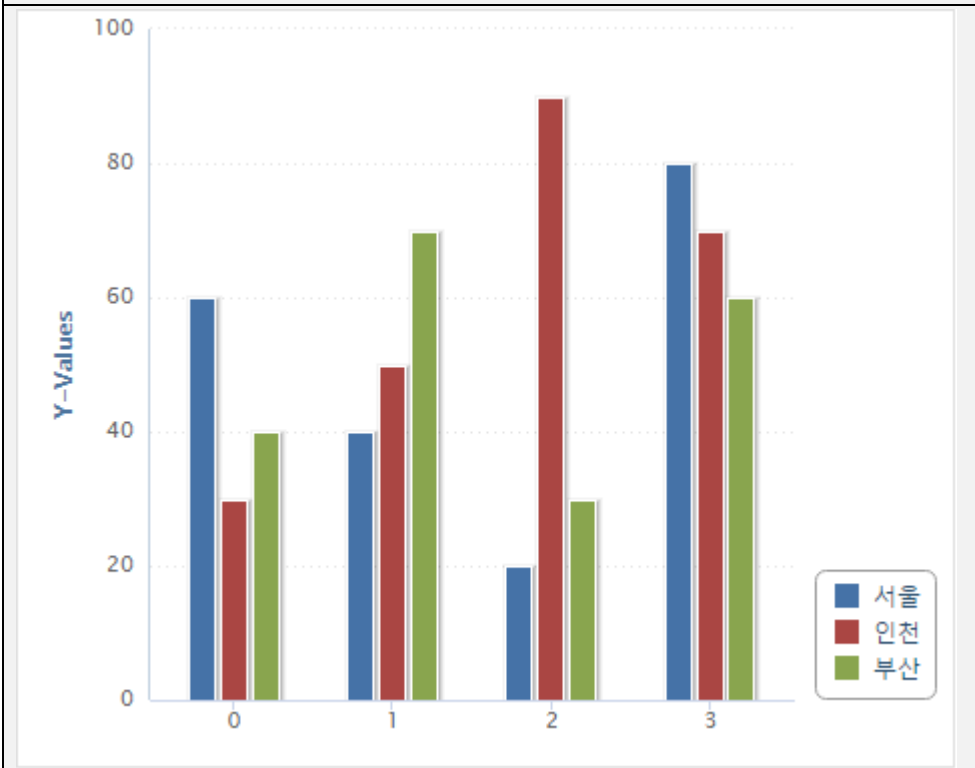
name	type	required	description
y	number	N(0)	상하 포지션 이동 값 (정렬 기준으로 기본값은 0 이다)

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    align: "right",
    verticalAlign: "bottom",
    layout: "vertical",
    y: -20
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
```



```
append: true,  
redraw: true  
});
```



➤ **제공 버전**

7.3.0.0

floating

➤ 기능

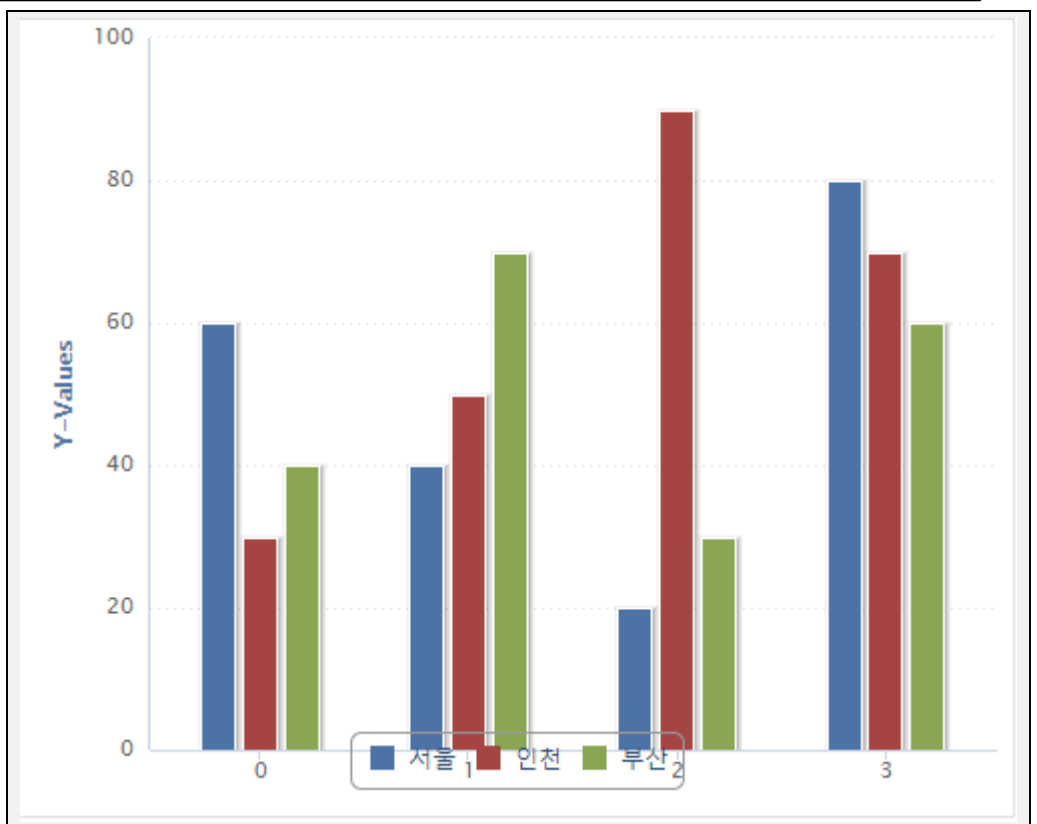
플로팅 여부 옵션을 설정 한다.

➤ Info

name	type	required	description
floating	boolean	N(false)	플로팅 여부(기본값 false)

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    floating: "true"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



reversed

➤ 기능

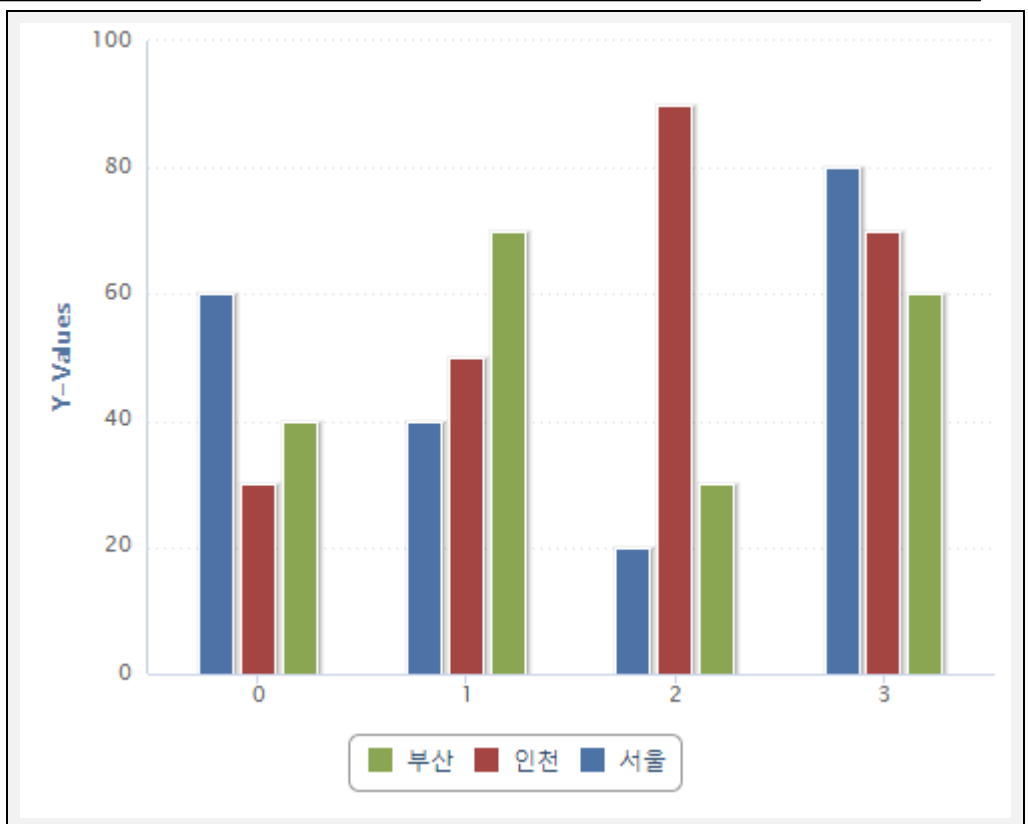
시리즈 항목 역순

➤ Info

name	type	required	description
reversed	boolean	N(false)	시리즈항목역순 여부(기본값 false)

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    reversed: "true"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



itemStyle

➤ 기능

시리즈 style을 설정한다.

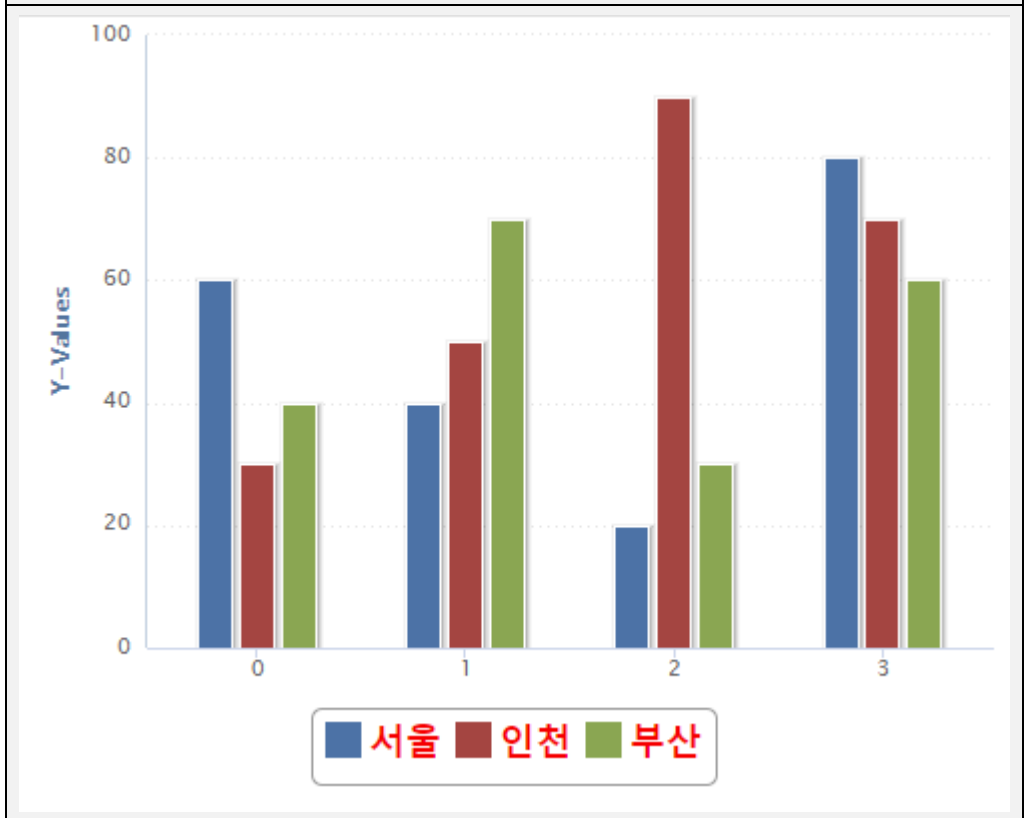
➤ Info

name	type	required	description
itemStyle.color	color	N	시리즈 text 색상
itemStyle.fontWeight	String	N	시리즈 text 굵기
itemStyle.fontSize	String	N	시리즈 text 크기

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    itemStyle: {
      color: "#FF0000",
      fontWeight: "bold",
      fontSize: "1.5em"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
```

```
data : [40,70,30,60]
}}
}, {
  append: true,
  redraw: true
});
```



itemWidth

➤ 기능

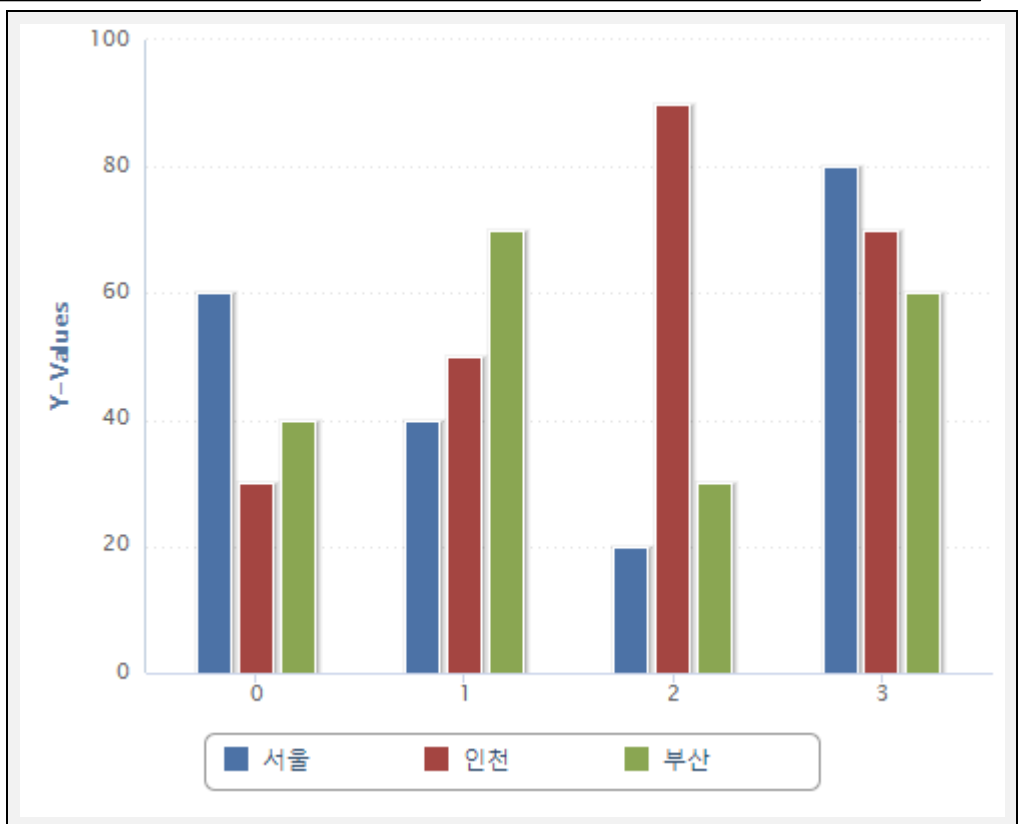
시리즈 아이템 별 너비를 설정한다.

➤ Info

name	type	required	description
itemWidth	number	N(0)	시리즈 아이템 별 너비 설정

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    itemWidth: 100
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



itemDistance

➤ 기능

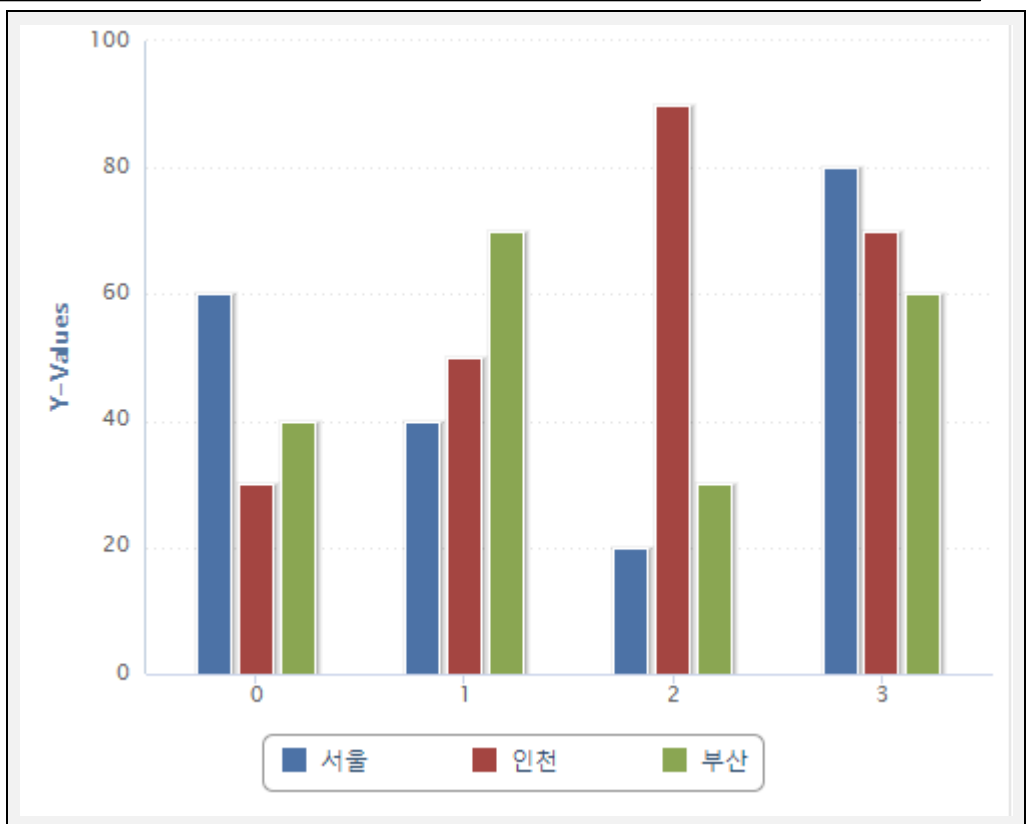
시리즈 아이템 별 너비를 설정한다.

➤ Info

name	type	required	description
itemWidth	number	N(0)	시리즈 아이템 별 너비 설정

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    itemDistance: 50
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



legend.itemHoverStyle

➤ 기능

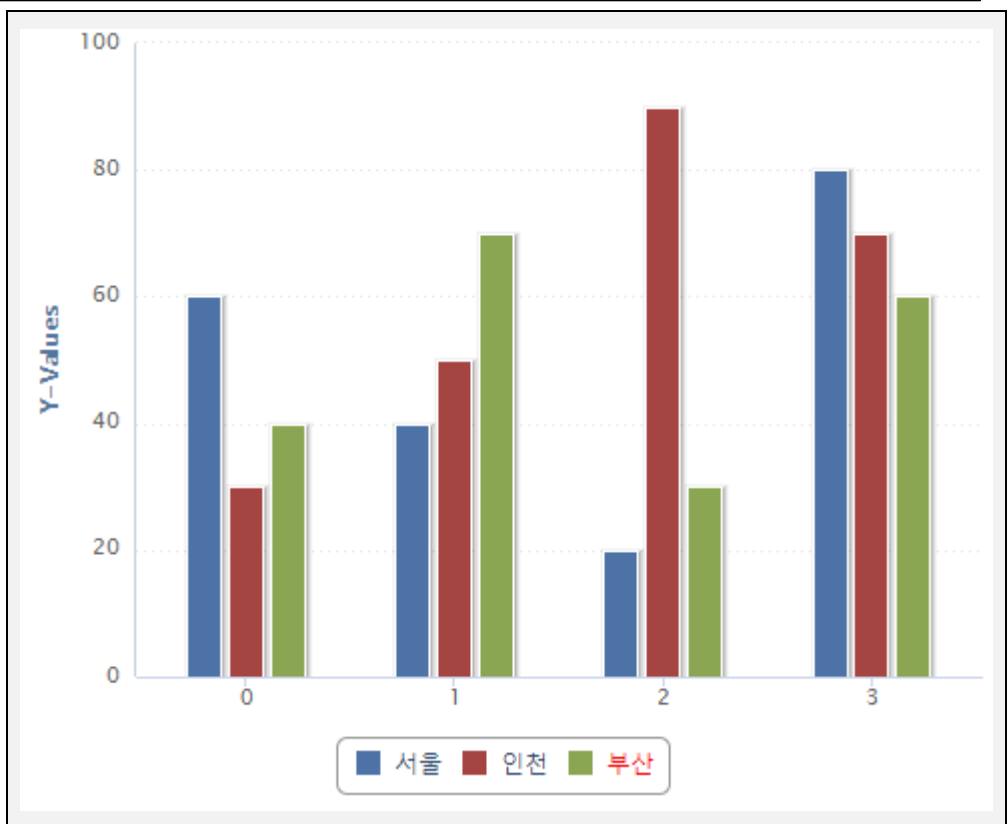
시리즈 아이템 마우스 호버시 스타일

➤ Info

name	type	required	description
itemHoverStyle.color	string	N(#000000)	마우스 호버시 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  legend: {
    itemHoverStyle: {
      color: "#FF0000"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



10.7 tooltip

툴팁에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
formatter	object	N	툴팁의 텍스트를 포맷하는 콜백
pointFormat	object	N	포인트 값 출력 포맷
headerFormat	object	N	헤더 출력 포맷
useHTML	boolean	N(false)	html tag 사용 여부
shared	boolean	N(false)	모든 시리즈의 값을 묶어서 출력할지 여부

valuePrefix	string	N	값의 머리 글 문자열
valueSuffix	string	N	값의 꼬리 글 문자열
enabled	boolean	N(true)	툴팁 사용 여부
animation	boolean	N(true)	출력시 애니메이션 효과 사용 여부
backgroundColor	color	N	배경색
borderColor	color	N	border 색상
borderWidth	number	N(1)	border의 너비
borderRadius	number	N(3)	radius 적용 값
shadow	boolean	N(true)	그림자 표시 여부
style	object	N	스타일 속성
style.color	color	N(#333333)	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께

formatter

➤ 기능

툴팁의 텍스트를 포맷하는 콜백 기능

➤ Info

name	type	required	description
formatter	object	N	툴팁의 텍스트를 포맷

➤ Enum

■ pointFormat

value	description
{ this.percentage }	백분율
{ this.point }	포인트 개체 사용 할 수 있음. Ex) this.point.name
{ this.series }	시리즈 개체 사용 할 수 있음. Ex) this.series.name
{ this.total }	총 가치 x값
{ this.x }	현재 x값
{ this.y }	현재 y값

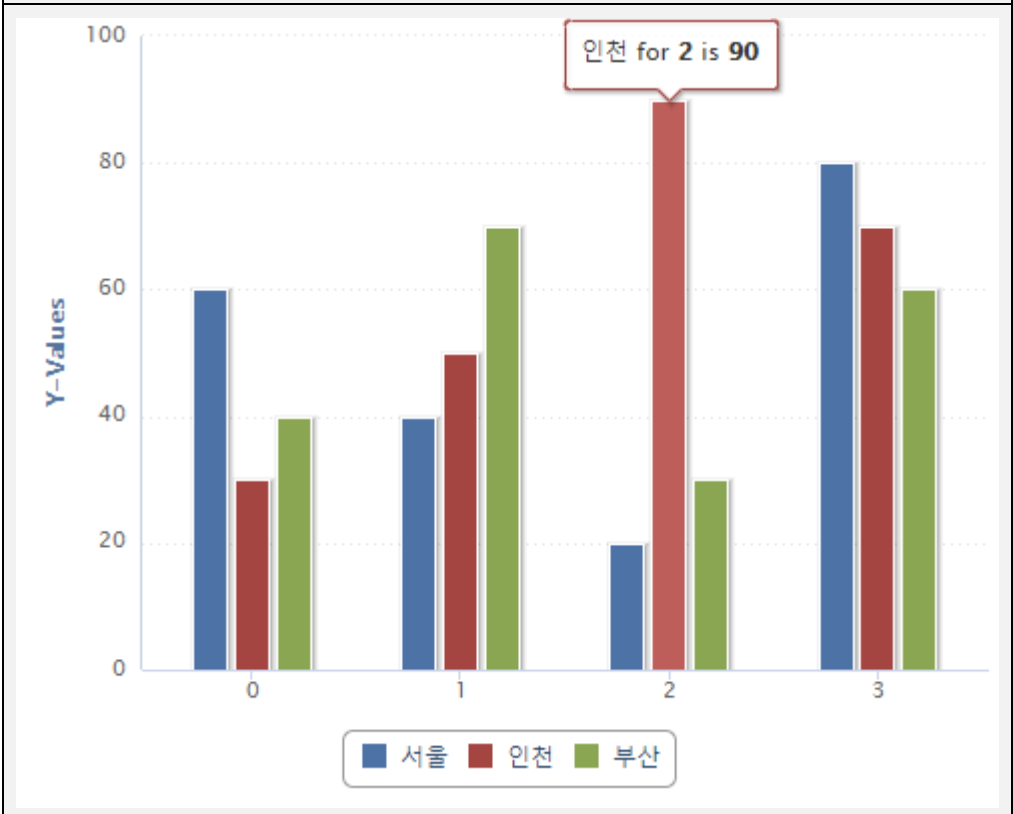
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    formatter: function () {
      return this.series.name + ' for <b>' + this.x +
        '</b> is <b>' + this.y + '</b>';
    }
  }
});
```

```

},
series: [{
  name: "서울",
  data : [60,40,20,80]
},{
  name: "인천",
  data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});

```



➤ 제공 버전

7.3.0.0	
---------	--

pointFormat

➤ 기능

포인트 값 출력 포맷을 설정 한다.

➤ Info

name	type	required	description
pointFormat	object	N	포인트 값 출력 포맷

➤ Enum

■ pointFormat

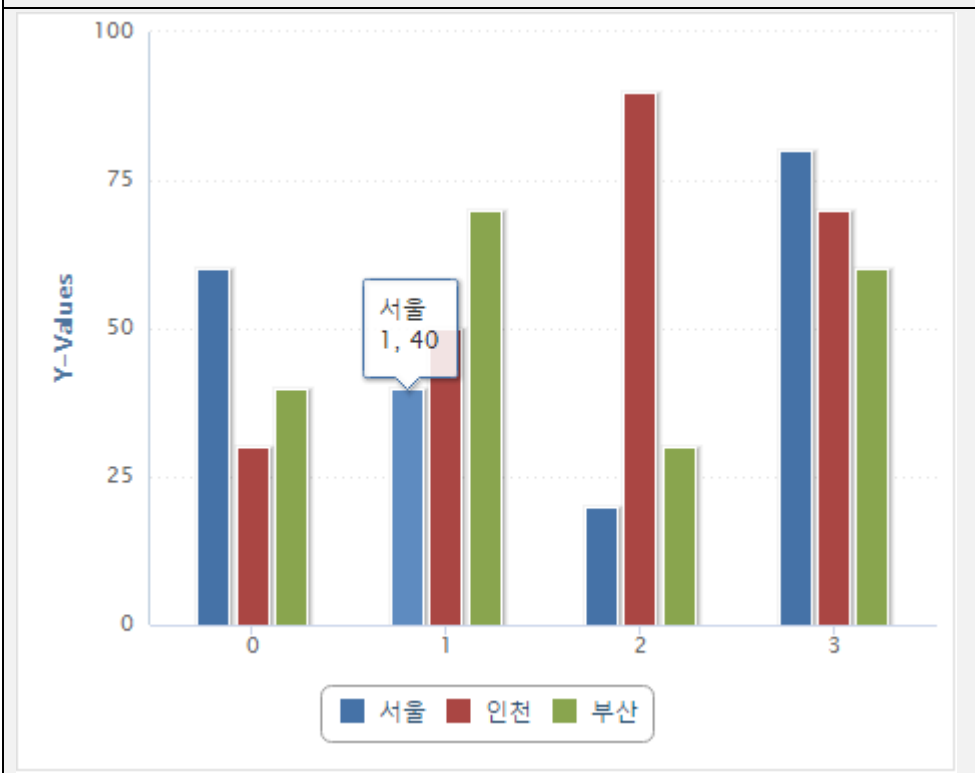
value	description
{series.name}	시리즈명
{series.color}	시리즈 색상
{point.x}	X축 값
{point.y}	y축 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    headerFormat: "{series.name}<br/>",
    pointFormat: "{point.x}, {point.y}"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```



```
data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

headerFormat

➤ 기능

헤더 출력 포맷을 설정 한다.

➤ Info

name	type	required	description
headerFormat	object	N	헤더 출력 포맷

➤ Enum

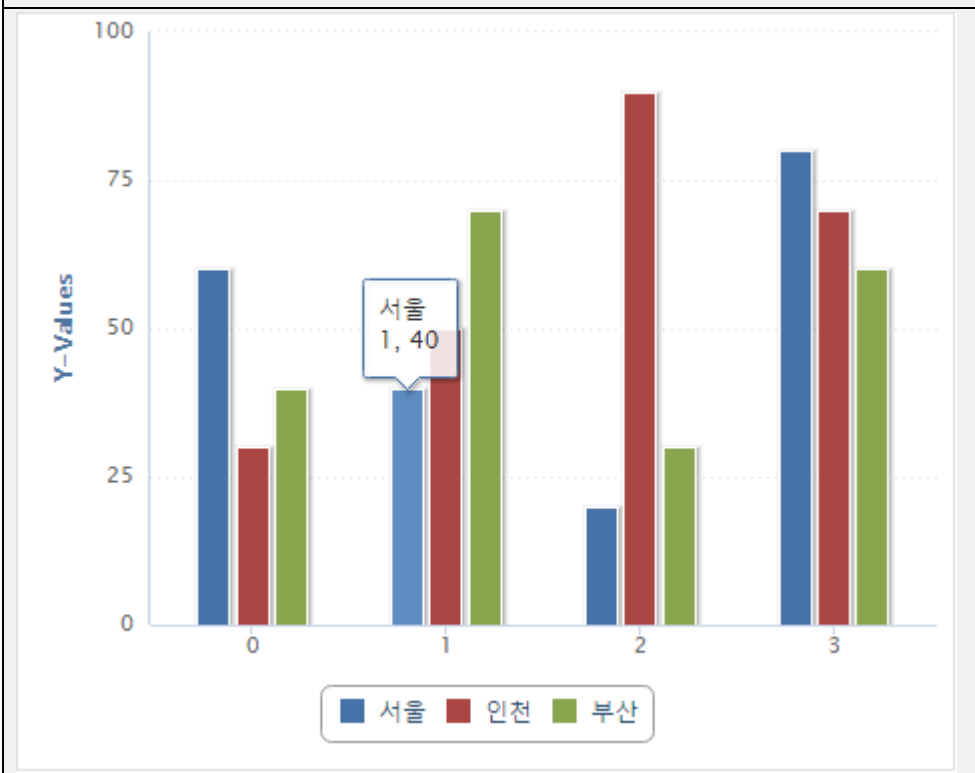
■ headerFormat

value	description
{series.name}	시리즈명
{series.color}	시리즈 색상
{point.x}	X축 값
{point.y}	y축 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    headerFormat: "{series.name}<br/>",
    pointFormat: "{point.x}, {point.y}"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```

```
data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

useHTML

➤ 기능

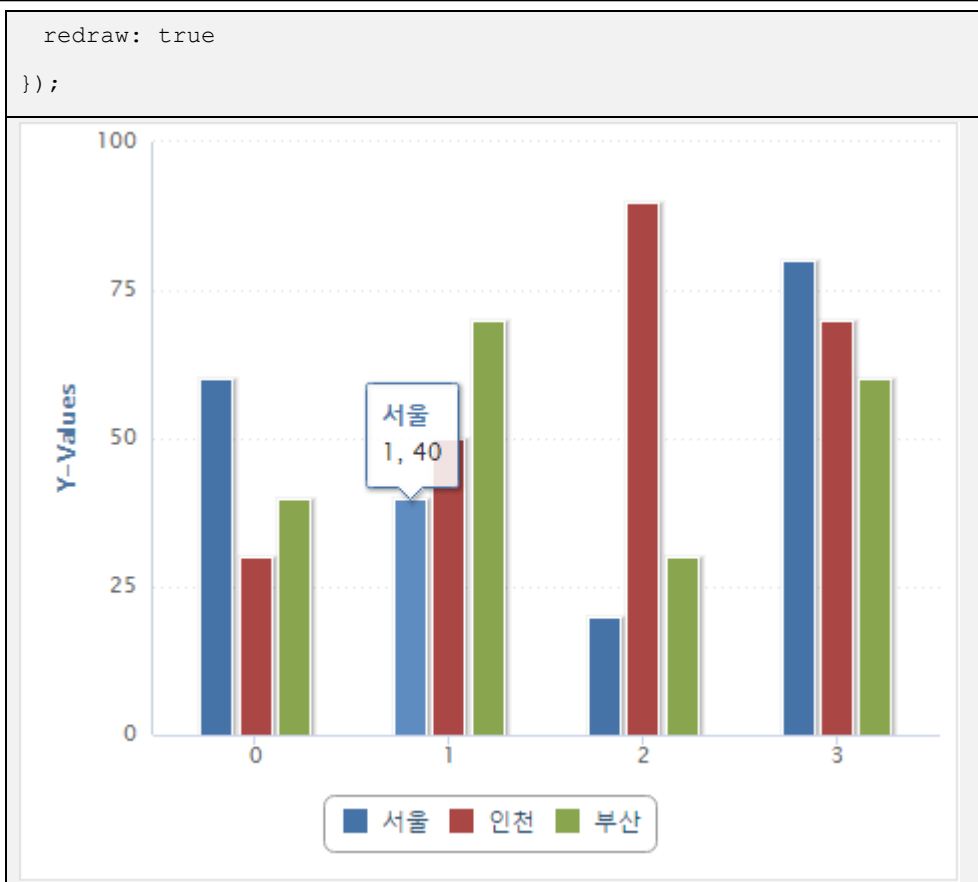
html tag 사용 여부를 설정 한다.

➤ Info

name	type	required	description
useHTML	boolean	N(false)	html tag 사용 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    headerFormat: "<span style='color: {series.color}; font-weight: bold'>{series.name}</span><br/>",
    pointFormat: "{point.x}, {point.y}",
    useHTML: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
```



➤ **제공 버전**

7.3.0.0	
---------	--

shared

➤ 기능

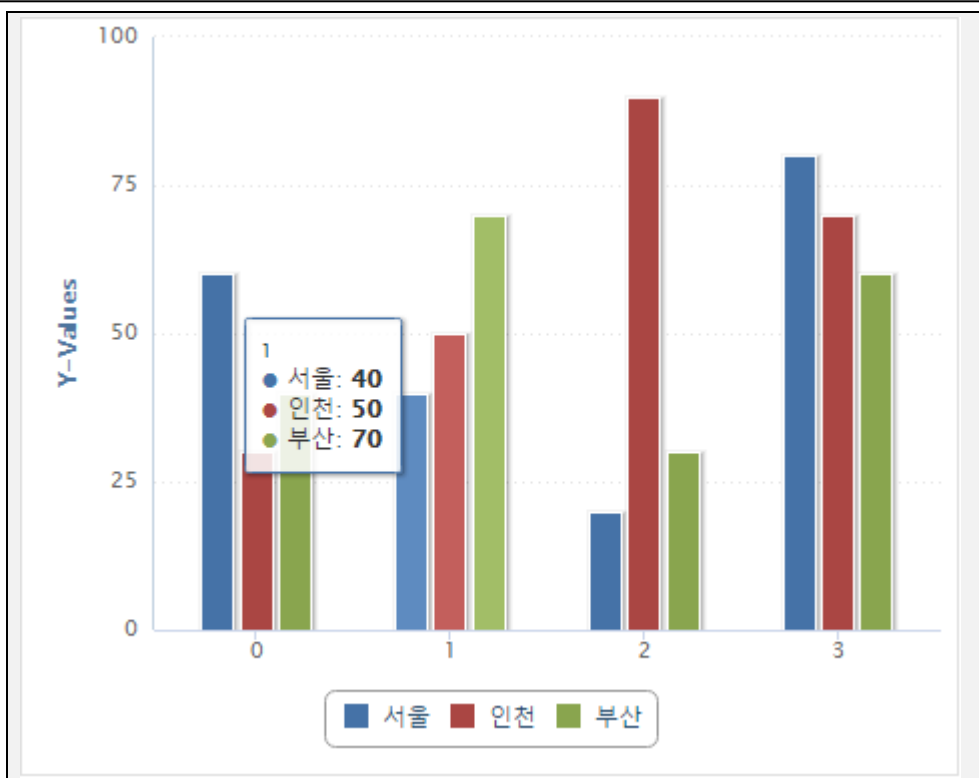
모든 시리즈의 값을 묶어서 출력할지 여부를 설정 한다.

➤ Info

name	type	required	description
shared	boolean	N(false)	모든 시리즈의 값을 묶어서 출력할지 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    shared: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

valuePrefix

➤ 기능

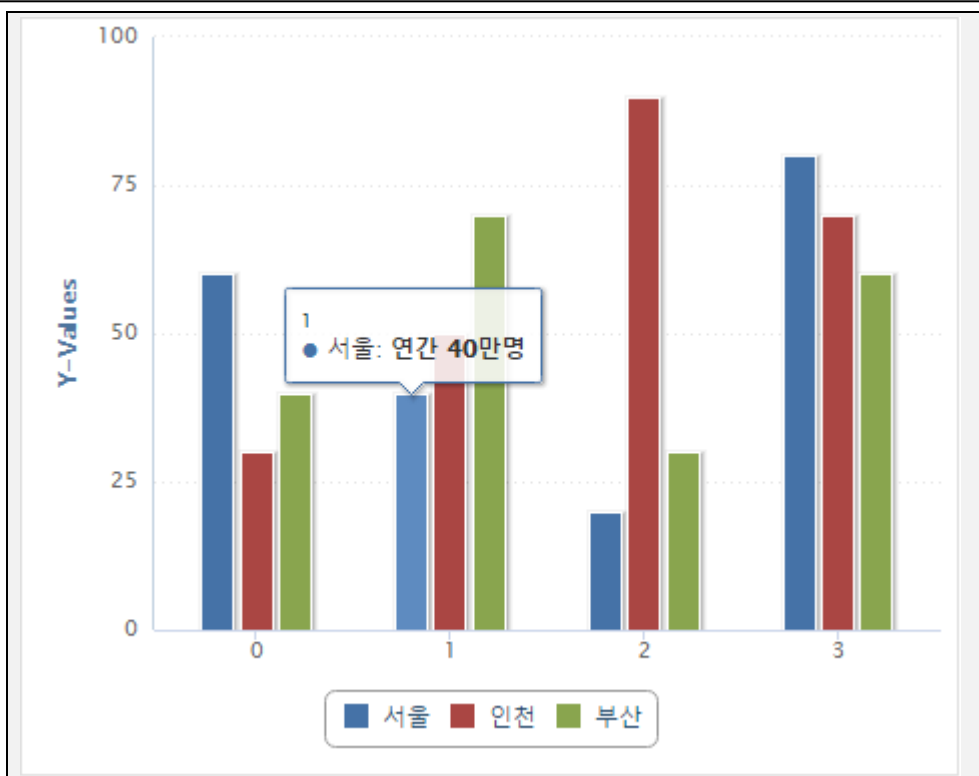
값의 머리 글 문자열을 설정 한다.

➤ Info

name	type	required	description
valuePrefix	string	N	값의 머리 글 문자열

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    valuePrefix: "연간 ",
    valueSuffix: "만명"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0

valueSuffix

➤ 기능

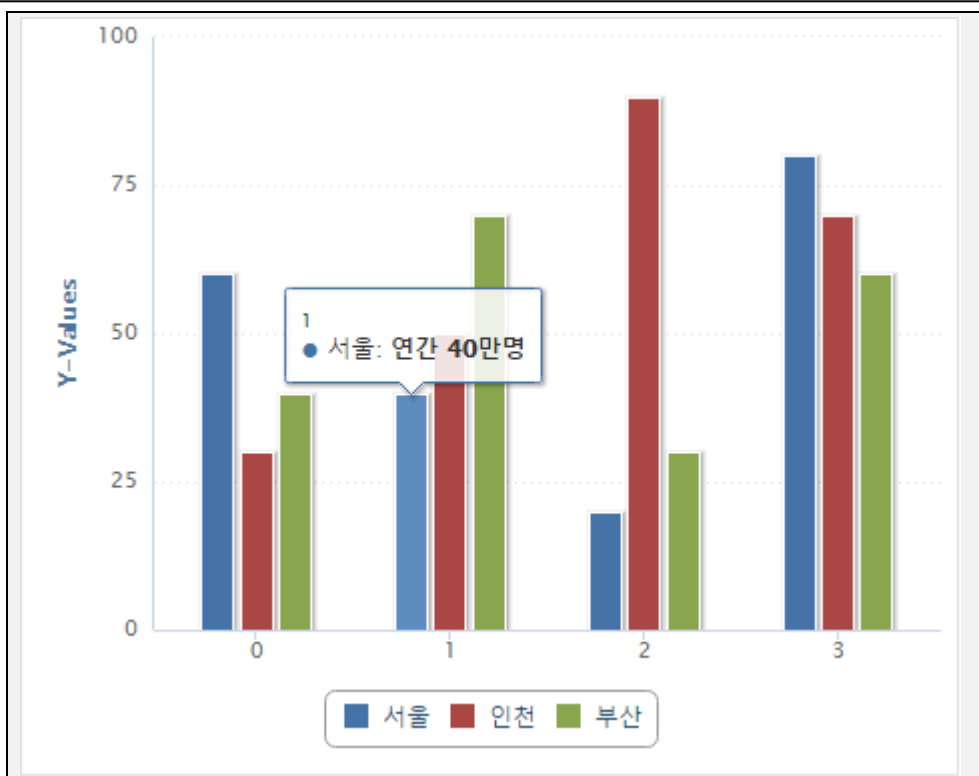
값의 꼬리 글 문자열을 설정 한다.

➤ Info

name	type	required	description
valueSuffix	string	N	값의 꼬리 글 문자열

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    valuePrefix: "연간 ",
    valueSuffix: "만명"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

enabled

➤ 기능

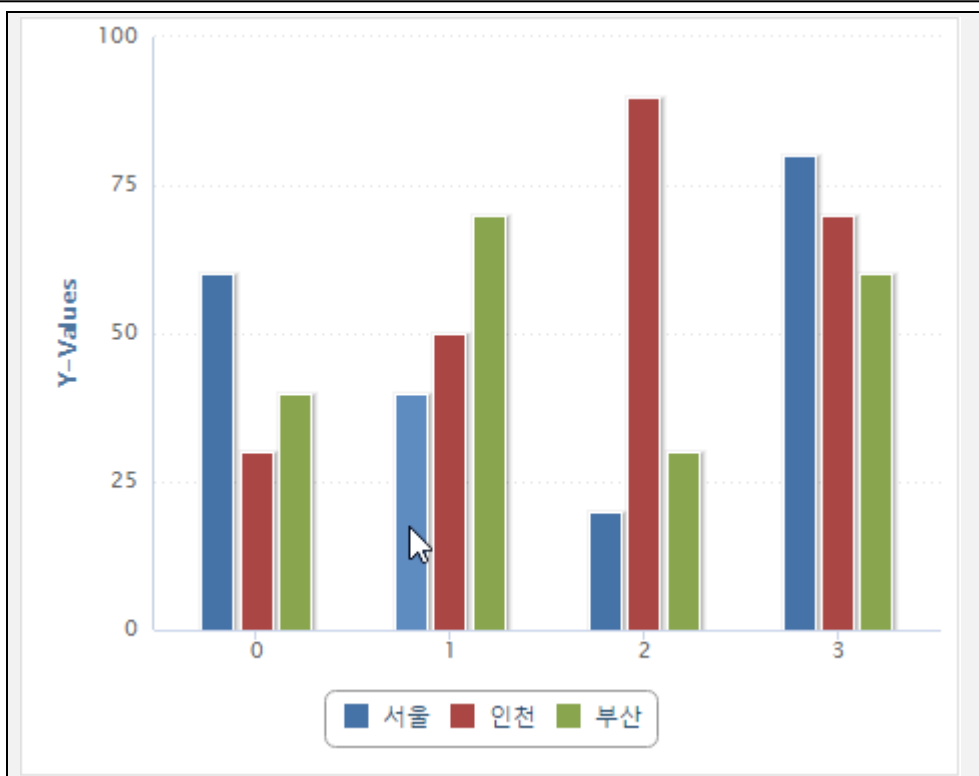
툴팁 사용 여부를 설정 한다.

➤ Info

name	type	required	description
enabled	boolean	N(true)	툴팁 사용 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    enabled: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

animation

➤ 기능

출력시 애니메이션 효과 사용 여부를 설정 한다.

➤ Info

name	type	required	description
animation	boolean	N(true)	출력시 애니메이션 효과 사용 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    animation: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0	
---------	--

backgroundColor

➤ 기능

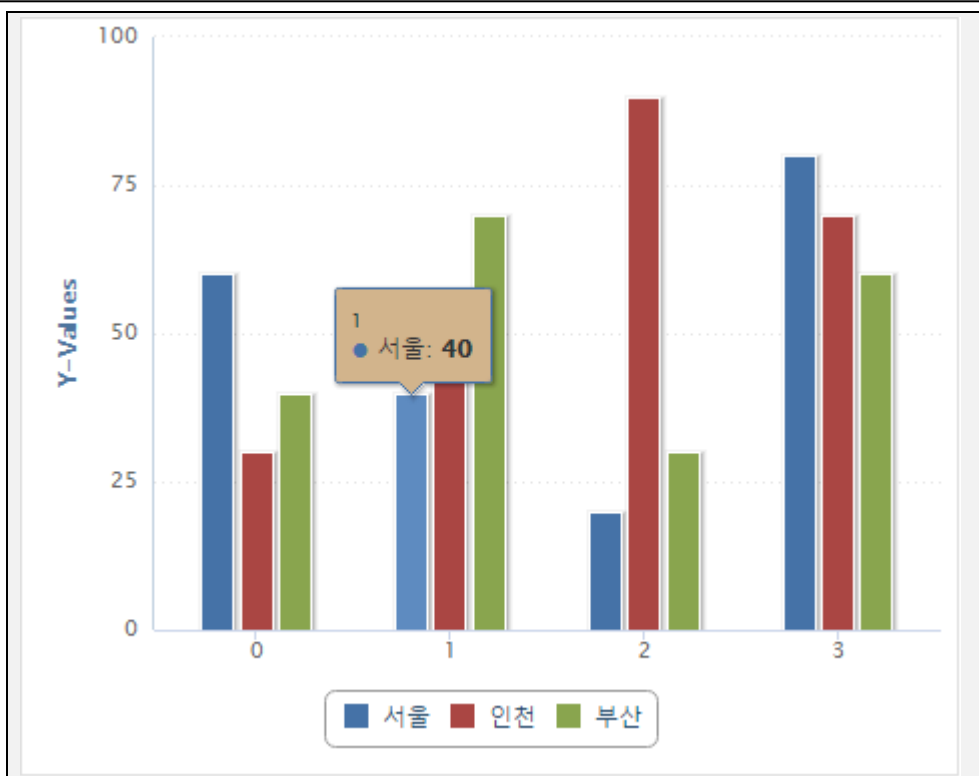
배경색을 설정 한다.

➤ Info

name	type	required	description
backgroundColor	color	N	배경색

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    backgroundColor: "#D2B48C"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0

borderColor

➤ 기능

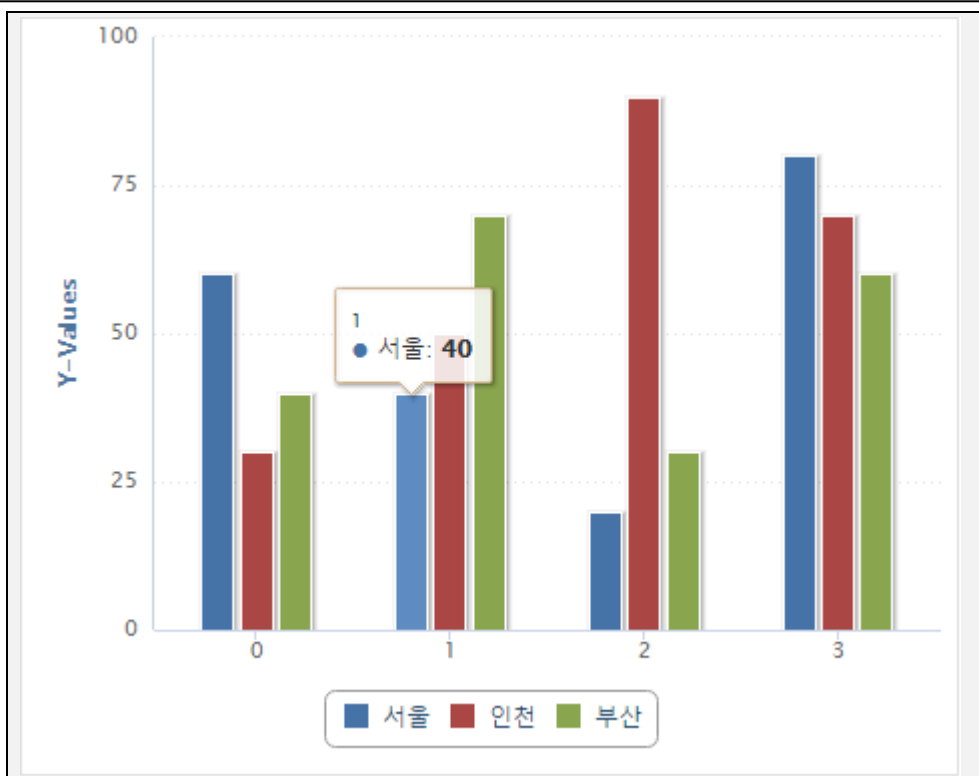
border 색상을 설정 한다.

➤ Info

name	type	required	description
borderColor	color	N	border 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    borderColor: "#D2B48C"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

borderWidth

➤ 기능

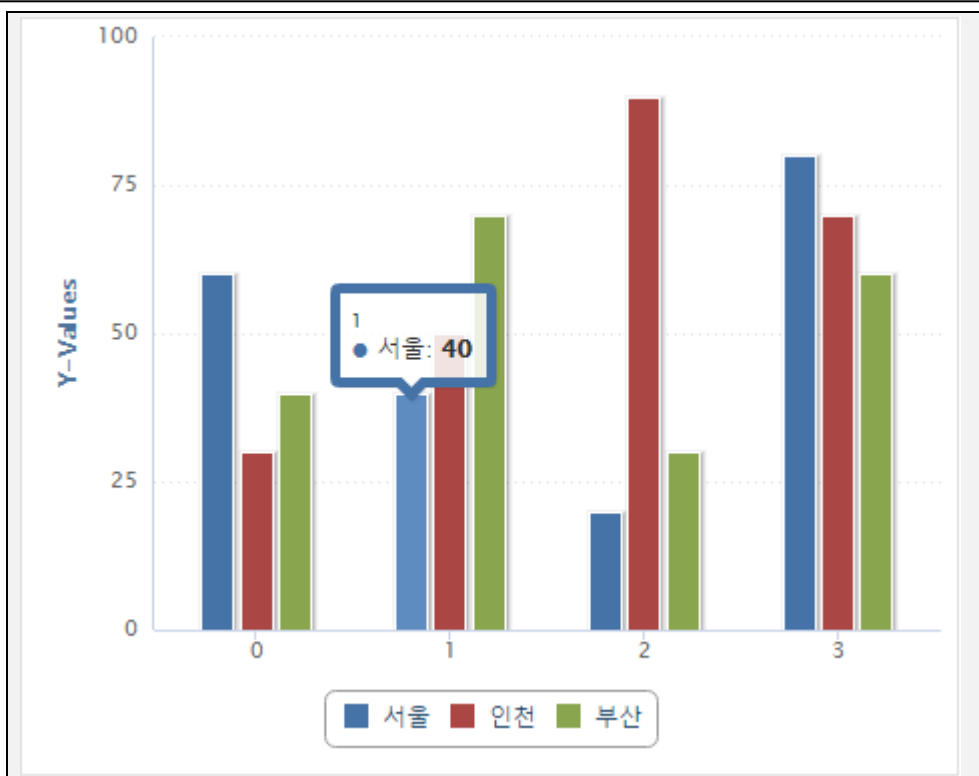
border 의 너비를 설정 한다.

➤ Info

name	type	required	description
borderWidth	number	N(1)	border의 너비

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    borderWidth: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

borderRadius

➤ 기능

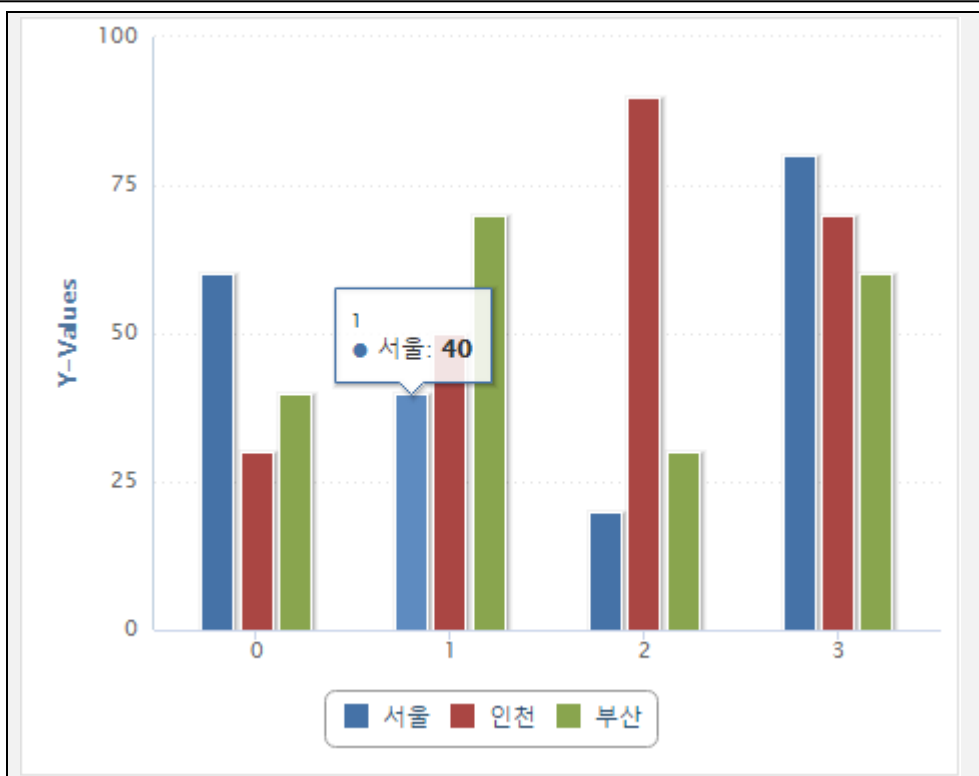
radius 적용 값을 설정 한다.

➤ Info

name	type	required	description
borderRadius	number	N(3)	radius 적용 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    borderRadius: 0
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

shadow

➤ 기능

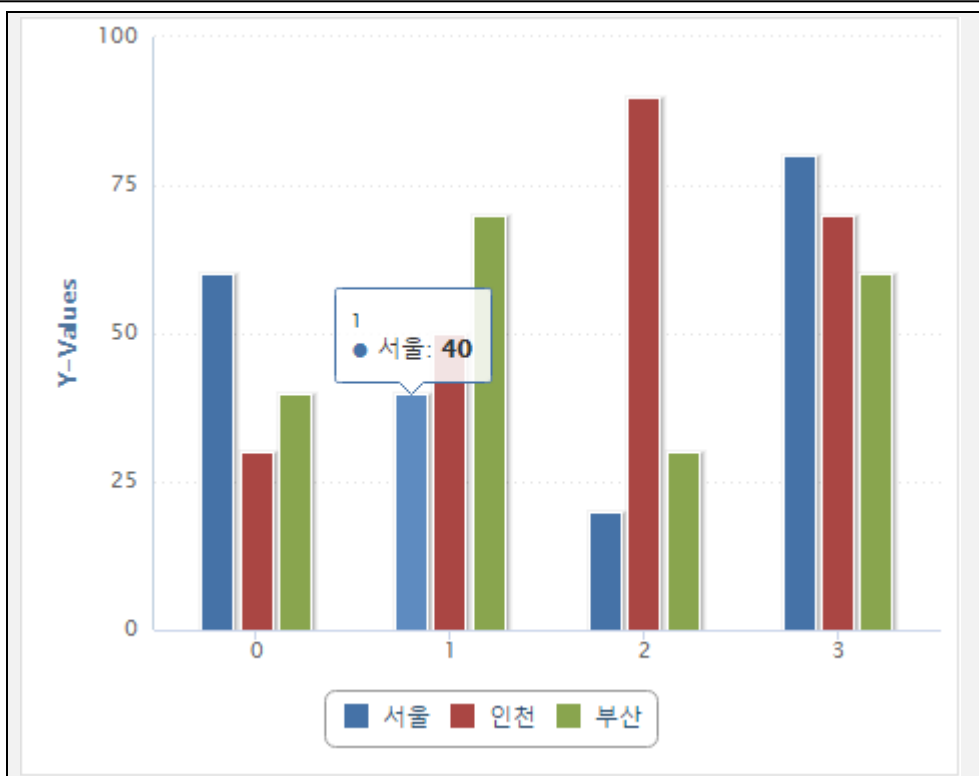
그림자 표시 여부를 설정 한다.

➤ Info

name	type	required	description
shadow	boolean	N(true)	그림자 표시 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    shadow: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0

style

➤ 기능

스타일 속성을 설정 한다.

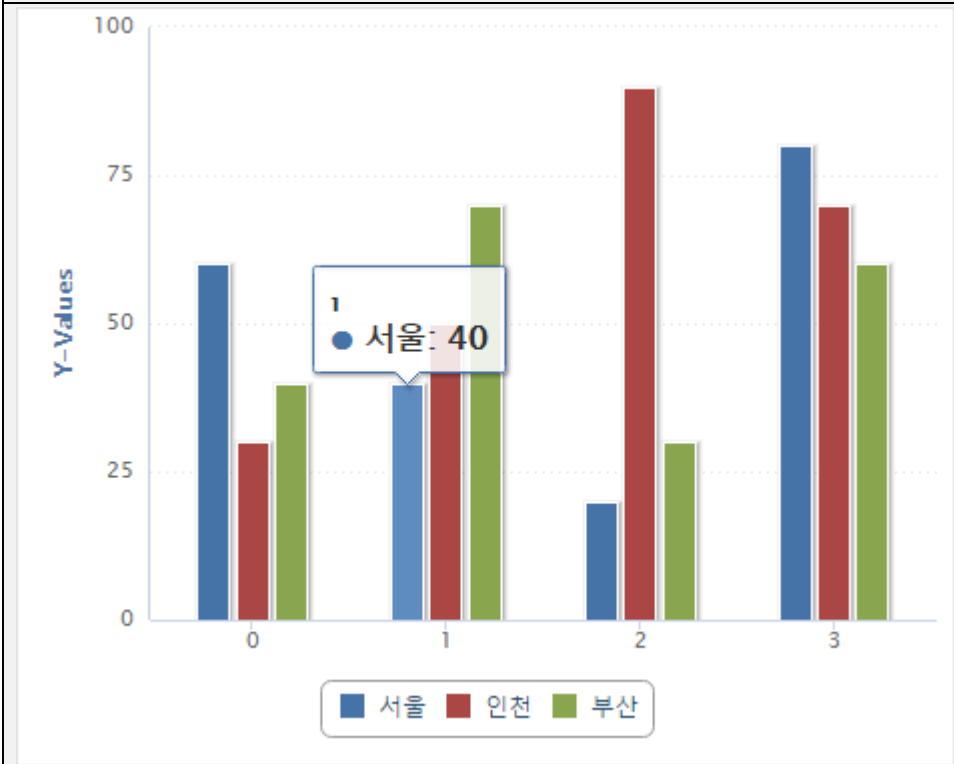
➤ Info

name	type	required	description
style	object	N	스타일 속성
style.color	color	N(#333333)	폰트 색상
style.fontSize	string	N	폰트 크기
style.fontWeight	string	N	폰트 두께

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  tooltip: {
    style: {
      fontSize: "1.3em",
      fontWeight: "bold"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
```

```
data : [40,70,30,60]
}}
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

10.8 xAxis

x축에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
title	object	N	축 제목 속성 (JSON 형식)
categories	array	N	축레이블 설정
crosshair	boolean	N(false)	포인트에 마우스 오버시 가이드라인 표시 여부
reversed	boolean	N(false)	축 반전 여부
showFirstLabel	boolean	N(true)	첫번째 축 라벨여부 표시
showLastLabel	boolean	N(true)	마지막 축 라벨여부 표시
lineWidth	number	1	축 라인의 너비
lineColor	color	N(#ccd6eb)	축 라인의 색상
startOnTick	boolean	N(false)	축의 시작점을 강제로 표시할지 여부
endOnTick	boolean	N(false)	축의 마지막점을 강제로 표시할지 여부
labels	object	N	라벨 속성
labels.align	string	N(center)	좌우 정렬 방법
labels.format	string	N({value})	축 라벨의 포맷 문자열 (값은 {value} 문자열을 이용한다.)
labels.formatter	function	N	축 라벨의 포맷을 정의하는 callback 함수
labels.style	object	N	라벨 스타일 속성
labels.useHTML	boolean	N(false)	html tag 사용 여부
labels.rotation	number	N(0)	기울기 (0 ~ 360)
labels.step	number	N	표시 단계

labels.enabled	boolean	N(true)	레이블 사용 여부
labels.staggerLines	number	N	레이블 다단
opposite	boolean	N(false)	상단에 축 표시 여부
min	number	N	축의 최소값
max	number	N	축의 최대값
tickWidth	number	N	축에 표시한 틱의 너비
tickInterval	number	N	축에 표시할 점의 간격
gridLineWidth	number	N(0)	그리드 라인의 너비
gridLineColor	color	N(#262626)	그리드 라인의 색상
gridLineDashStyle	string	N(solid)	그리드 라인의 종류
plotBands	object	N	플롯밴드 속성 (특정영역 하이라이팅)
offset	number	N	기본 위치 기준으로 표시할 offset 값
minorTickInterval	number string	N	축에 표시할 보조 점의 간격
minorGridLineWidth	string	N	보조 그리드 라인의 너비
minorGridLineColor	color	N	보조 그리드 라인의 색상
minorGridLineDashStyle	string	N	보조 그리드 라인의 종류
alternateGridColor	color	N	그리드 라인의 교차 색상

title

➤ 기능

축 제목 속성을 설정 한다.

➤ Info

name	type	required	description
title	object	N	축 제목 속성 (JSON 형식)
title.text	string	N	제목 문자열
title.align	string	N(center)	좌우 정렬 값
title.x	number	N(0)	좌우 포지션 이동 값
title.y	number	N	상하 포지션 이동 값
title.style	object	N	스타일 설정
title.style.color	color	N	폰트 색상
title.style.fontSize	string	N	폰트 크기
title.style.fontWeight	string	N	폰트 두께
title.rotation	number	N(0)	타이틀 회전
title.enabled	boolean	N(true)	타이틀 사용여부(default: true)

➤ Enum

■ title.align

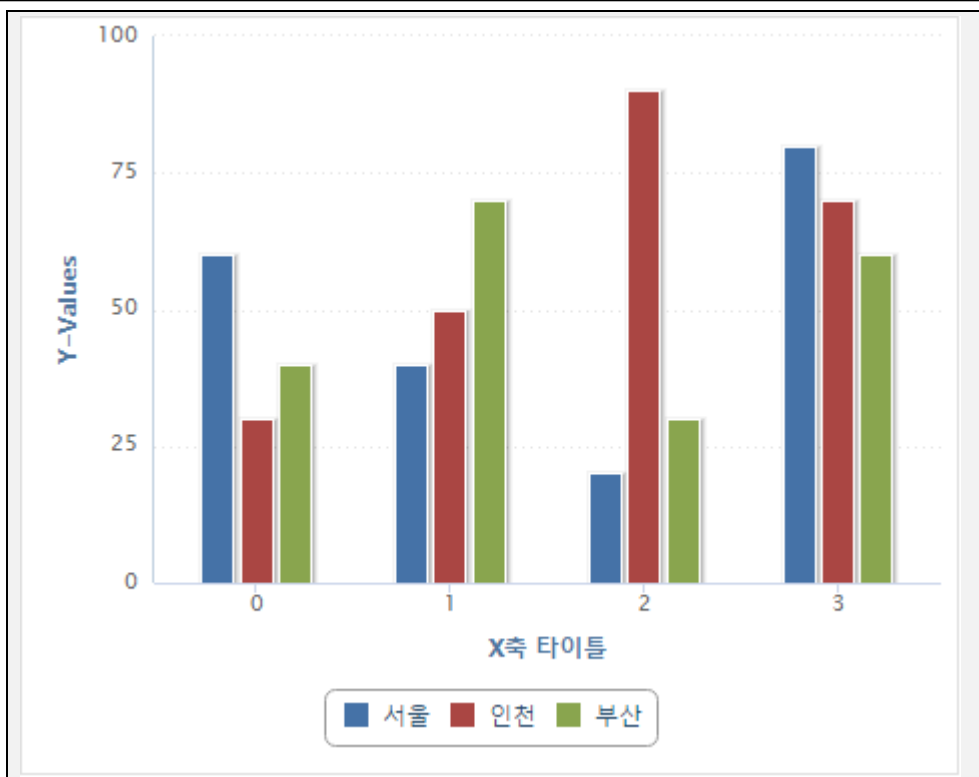
value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },

```

```
xAxis: {  
  title: {  
    text: "x축 타이틀"  
  }  
},  
series: [{  
  name: "서울",  
  data : [60,40,20,80]  
},{  
  name: "인천",  
  data : [30,50,90,70]  
},{  
  name: "부산",  
  data : [40,70,30,60]  
}]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

categories

➤ 기능

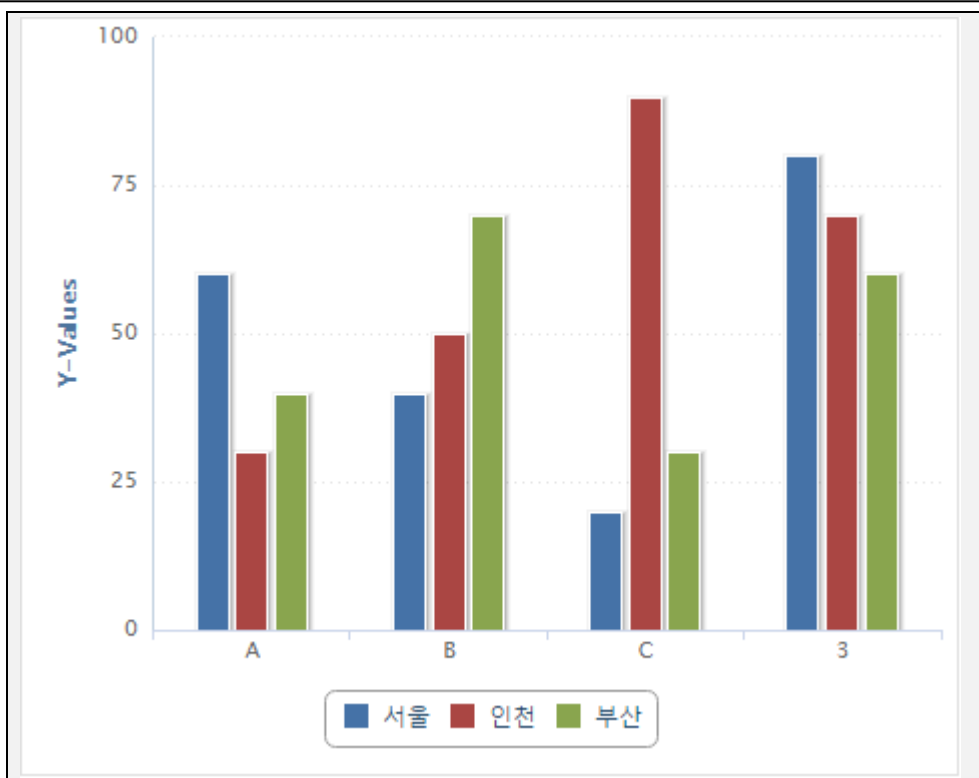
축 레이블을 설정 한다.

➤ Info

name	type	required	description
categories	array	N	축 레이블 설정

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    categories: ["A", "B", "C"]
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

crosshair

➤ 기능

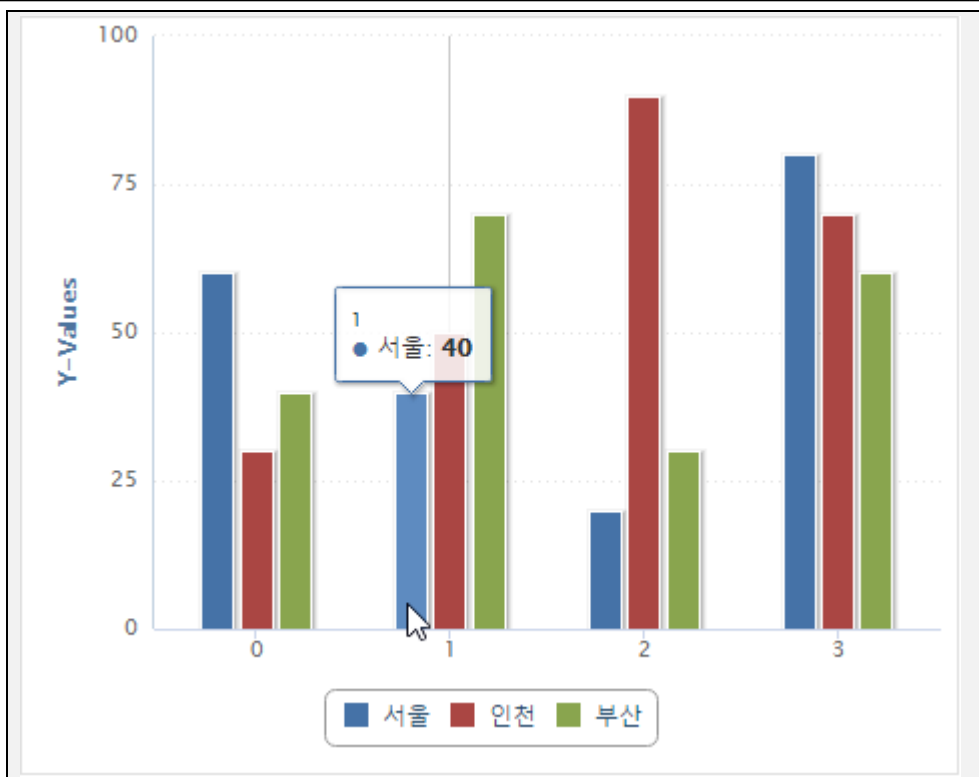
포인트에 마우스 오버시 가이드라인 표시 여부를 설정 한다.

➤ Info

name	type	required	description
crosshair	boolean	N(false)	포인트에 마우스 오버시 가이드라인 표시 여부

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    crosshair: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

reversed

➤ 기능

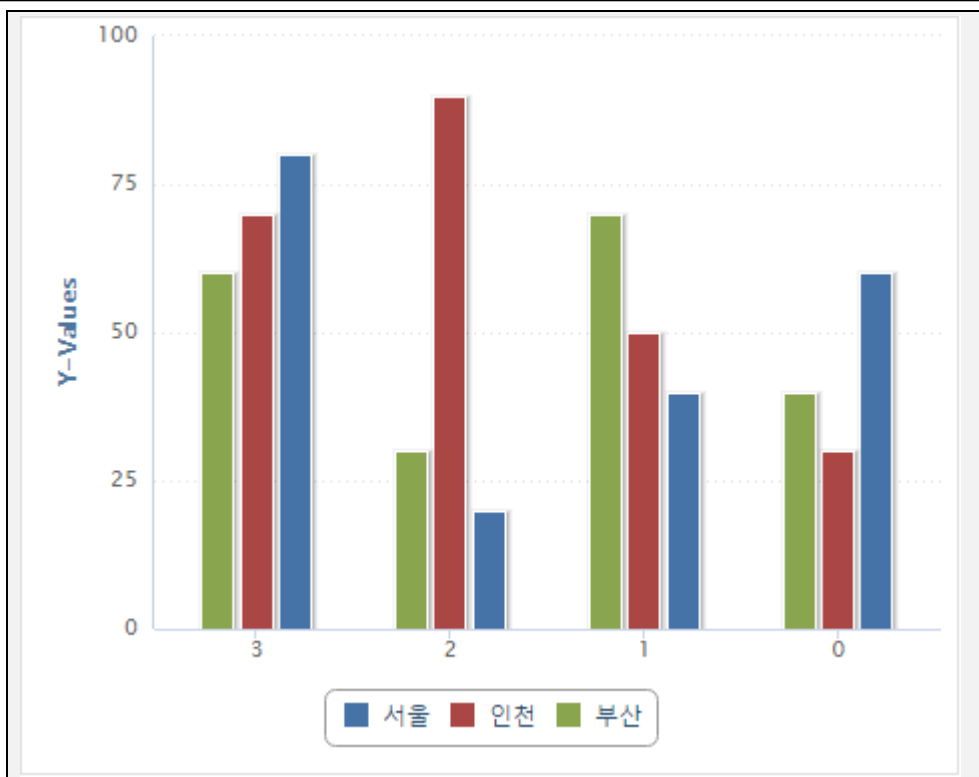
축 반전 여부를 설정 한다.

➤ Info

name	type	required	description
reversed	boolean	N(false)	축 반전 여부

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    reversed: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

showFirstLabel

➤ 기능

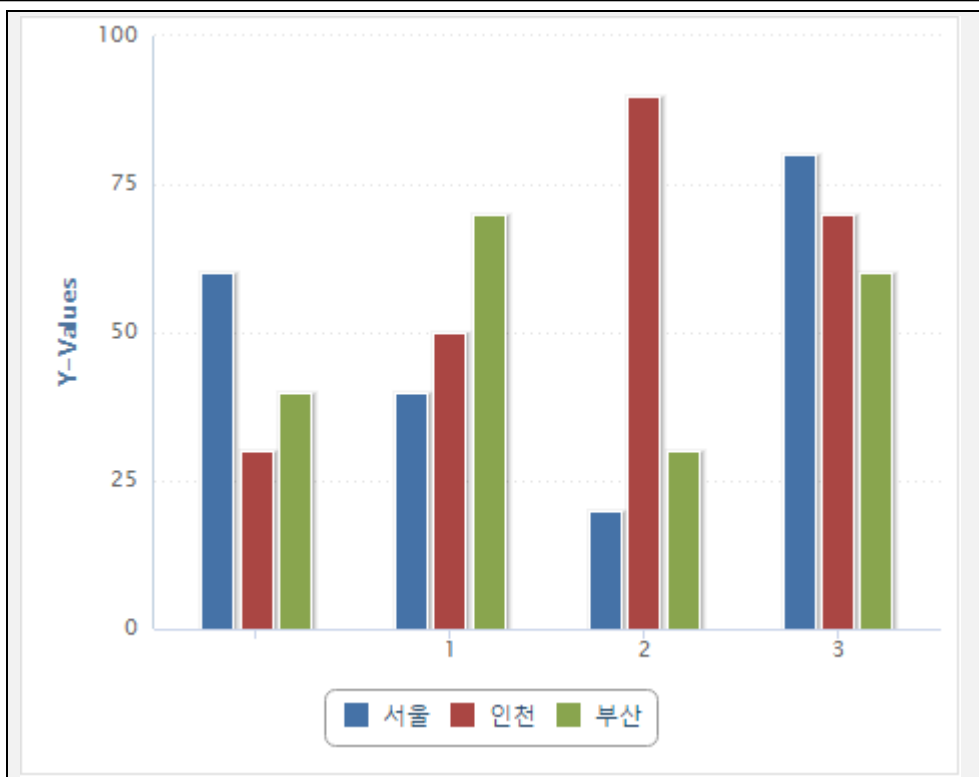
첫번째 축 라벨여부 표시 여부를 설정 한다.

➤ Info

name	type	required	description
showFirstLabel	boolean	N(true)	첫번째 축 라벨여부 표시

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    showFirstLabel: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

showLastLabel

➤ 기능

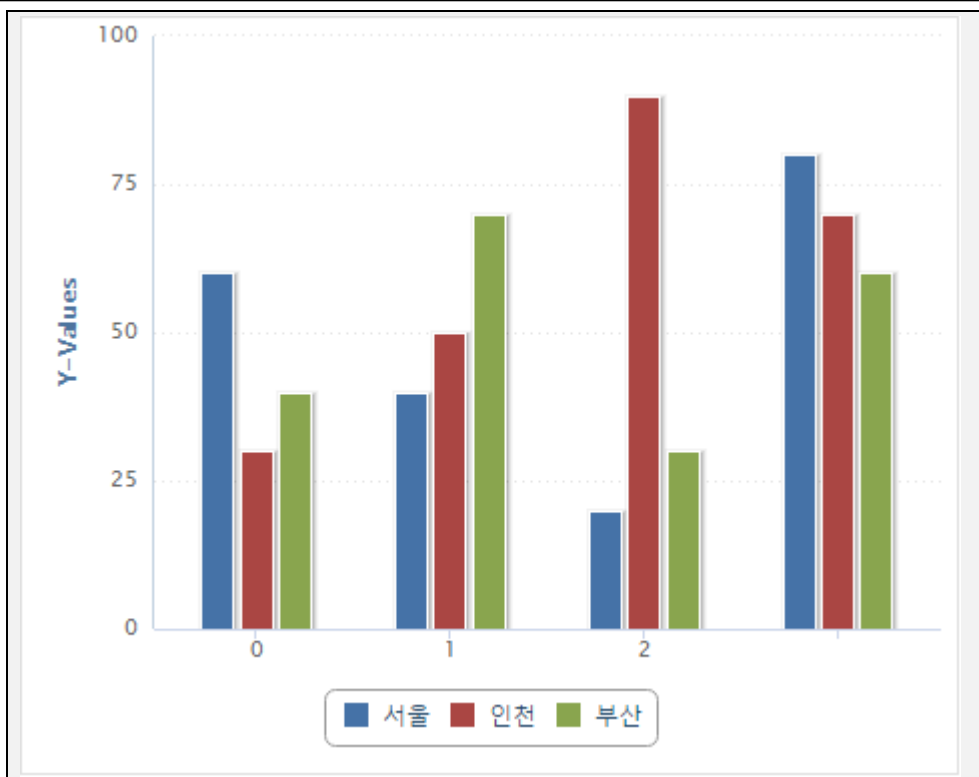
마지막 축 라벨여부 표시 여부를 설정 한다.

➤ Info

name	type	required	description
showLastLabel	boolean	N(true)	마지막 축 라벨여부 표시

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    showLastLabel: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

lineWidth

➤ 기능

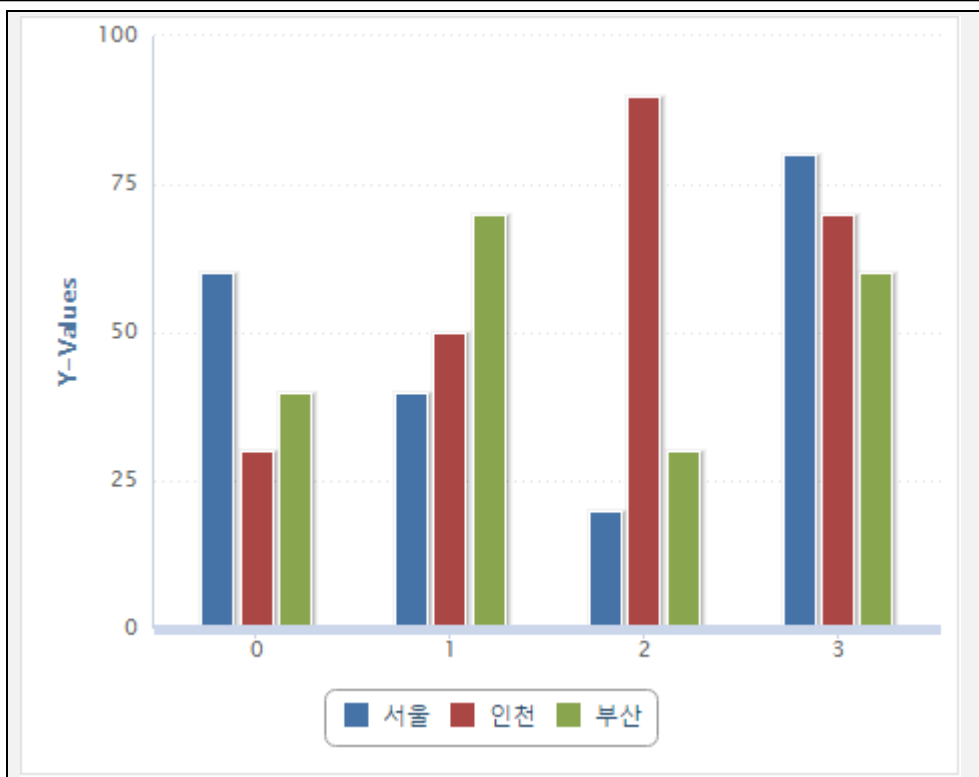
축 라인의 너비를 설정 한다.

➤ Info

name	type	required	description
lineWidth	number	1	축 라인의 너비

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  xAxis: {
    lineWidth: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

lineColor

➤ 기능

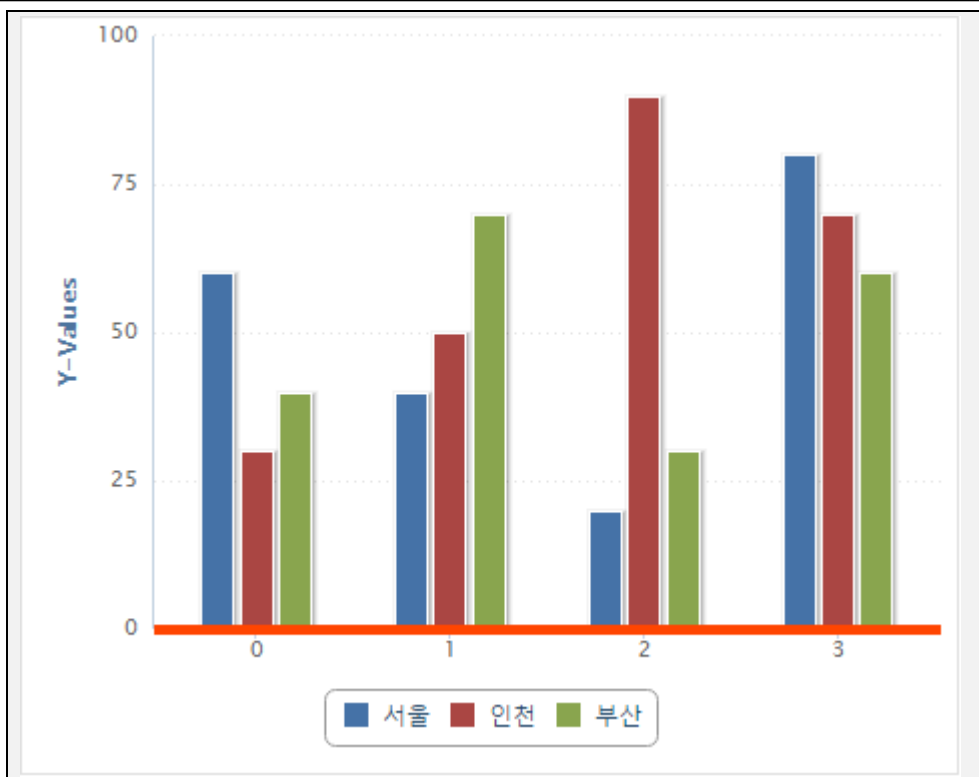
축 라인의 색상을 설정 한다.

➤ Info

name	type	required	description
lineColor	color	N(#ccd6eb)	축 라인의 색상

➤ Example

```
myChart.setOptions({  
  chart:{  
    type: "column"  
  },  
  xAxis: {  
    lineWidth: 5,  
    lineColor: "#FF4500"  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

startOnTick

➤ 기능

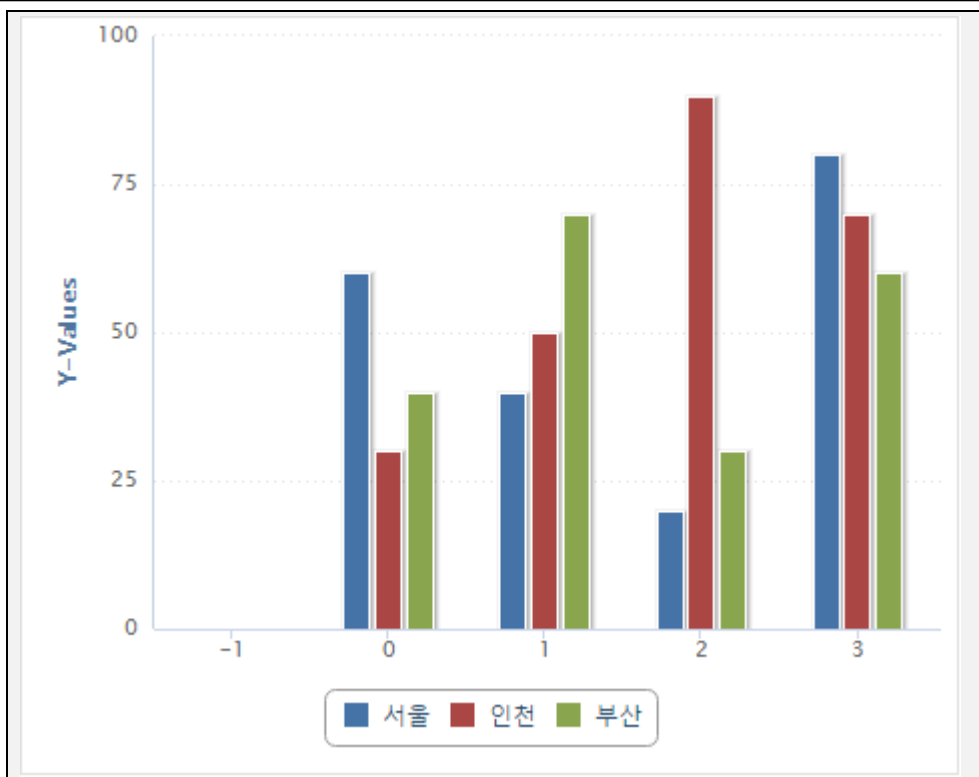
축의 시작점 다음에 강제로 축의 지점을 강제로 표시할지 여부를 설정 한다.

➤ Info

name	type	required	description
startOnTick	boolean	N(false)	축의 시작점을 강제로 표시할지 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    startOnTick: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

endOnTick

➤ 기능

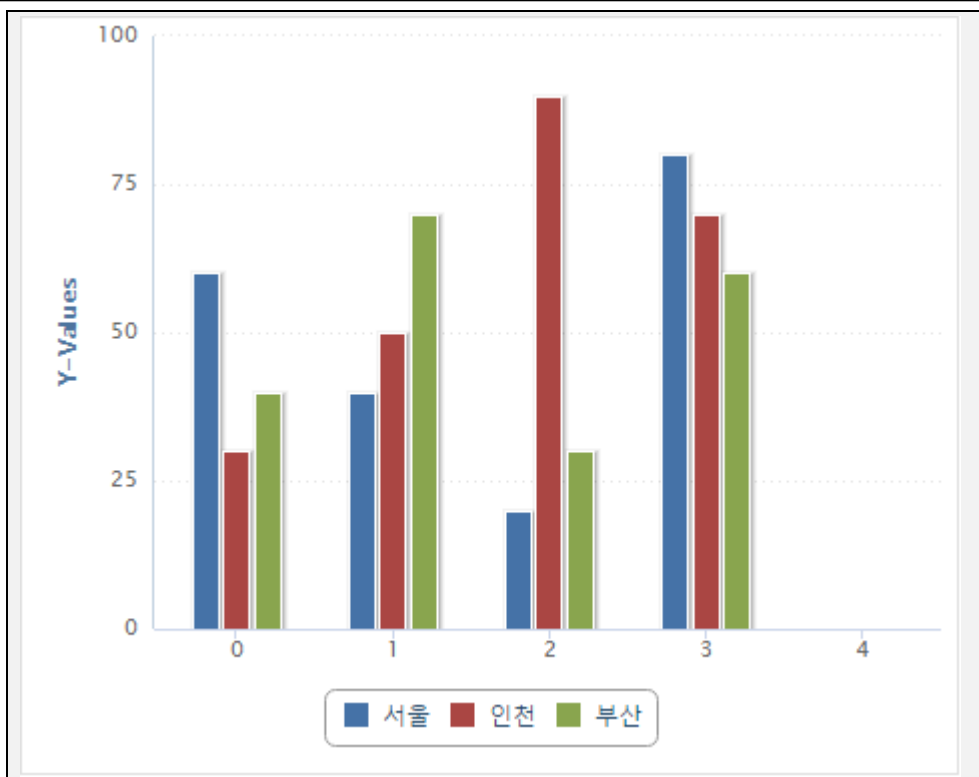
축의 마지막점 다음에 강제로 축의 지점을 표시할지 여부를 설정 한다.

➤ Info

name	type	required	description
endOnTick	boolean	N(false)	축의 마지막점을 강제로 표시할지 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    endOnTick: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

labels

➤ 기능

라벨 속성을 설정 한다.

➤ Info

name	type	required	description
labels	object	N	라벨 속성
labels.align	string	N(center)	좌우 정렬 방법
labels.format	string	N({value})	축 라벨의 포맷 문자열 (값은 {value} 문자열을 이용 한다.) value는 data를 의미.
labels.formatter	function	N	축 라벨의 포맷을 정의하는 callback 함수 (값은 {this.value}) this.value는 현재 data를 의미.
labels.style	object	N	라벨 스타일 속성
labels.useHTML	boolean	N(false)	html tag 사용 여부
labels.rotation	number	N(0)	기울기 (0 ~ 360)
labels.step	number	N	표시 단계
labels.enabled	boolean	N(true)	라벨 사용 여부
labels.staggerLines	number	N	레이블 다단

■ labels.style Info

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기
fontWeight	string	N	폰트 두께

➤ Enum

■ Labels.align

value	description
-------	-------------

center	중앙 정렬
left	좌측 정렬
right	우측 정렬

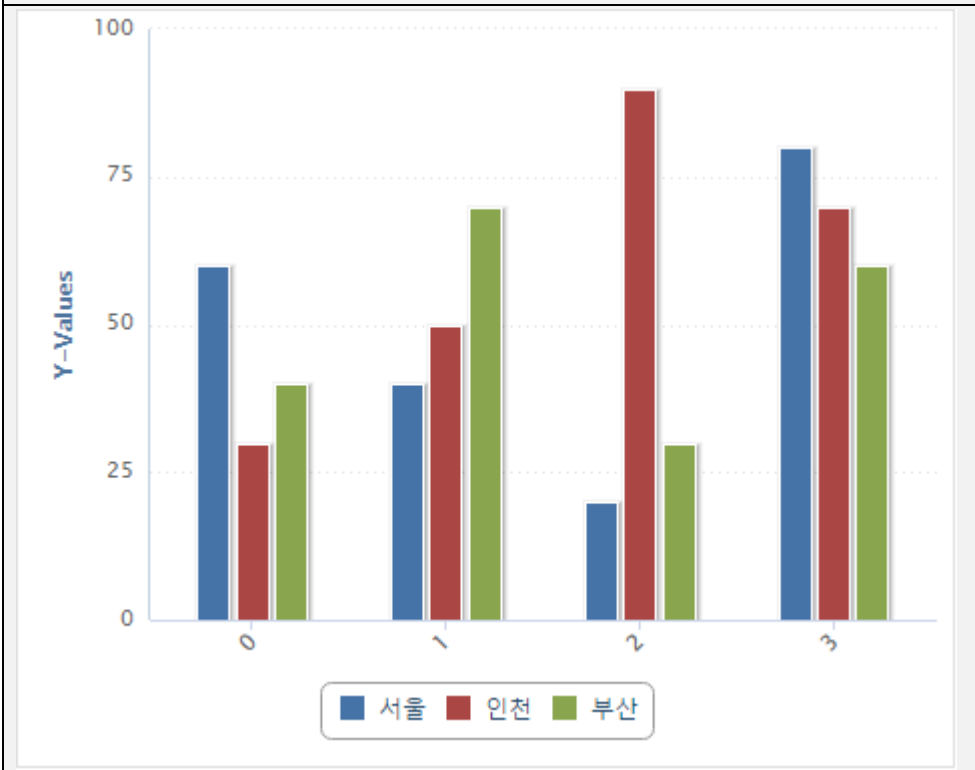
■ labels.format

value	description
{value}	x축 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    labels: {
      style: {
        fontWeight: "bold"
      },
      rotation: -45
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
```

```
append: true,  
redraw: true  
});
```



➤ **제공 버전**

7.3.0.0

opposite

➤ 기능

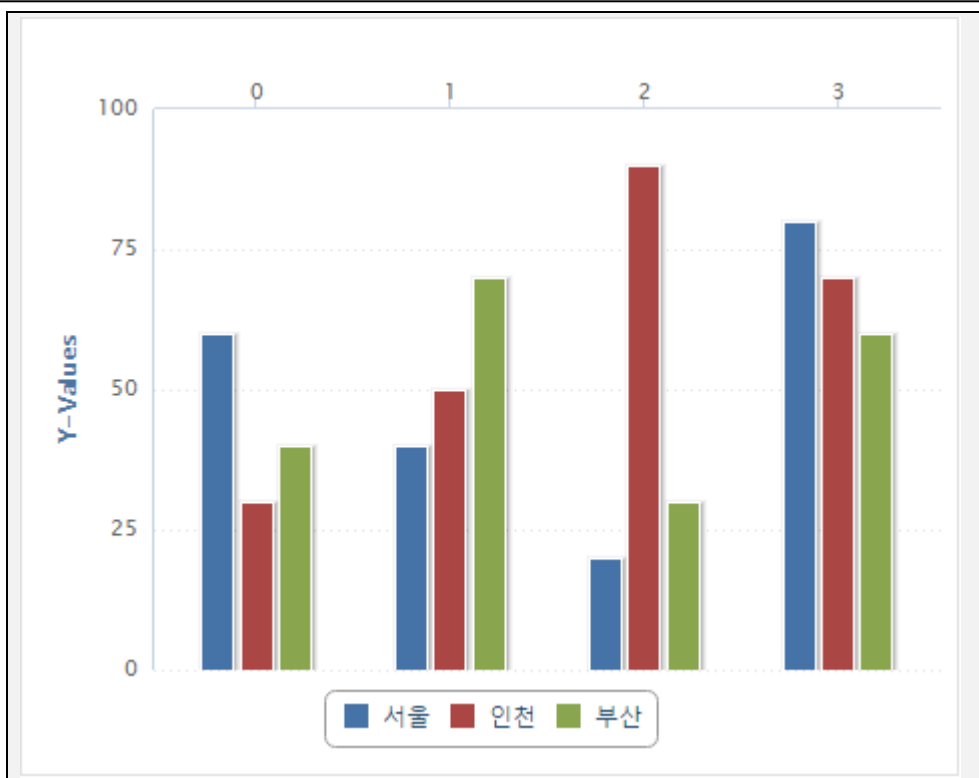
차트 상단에 축 표시 여부를 설정 한다.

➤ Info

name	type	required	description
opposite	boolean	N(false)	상단에 축 표시 여부

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    opposite: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

min

➤ 기능

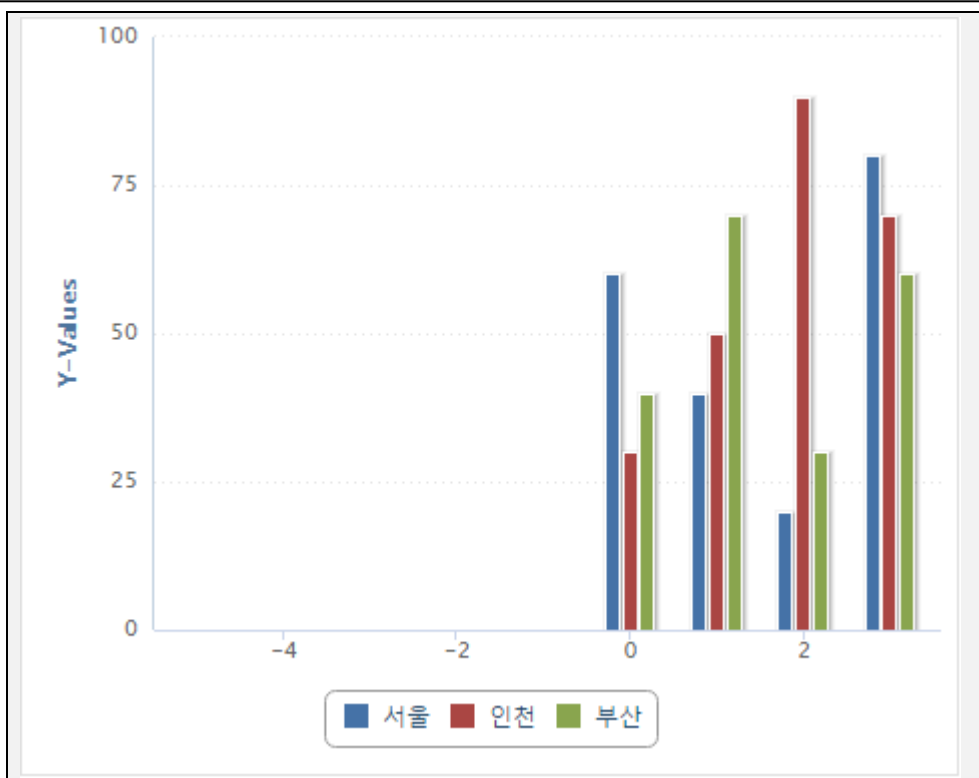
축의 최소값을 설정 한다.

➤ Info

name	type	required	description
min	number	N	축의 최소값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    min: -5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0

max

➤ 기능

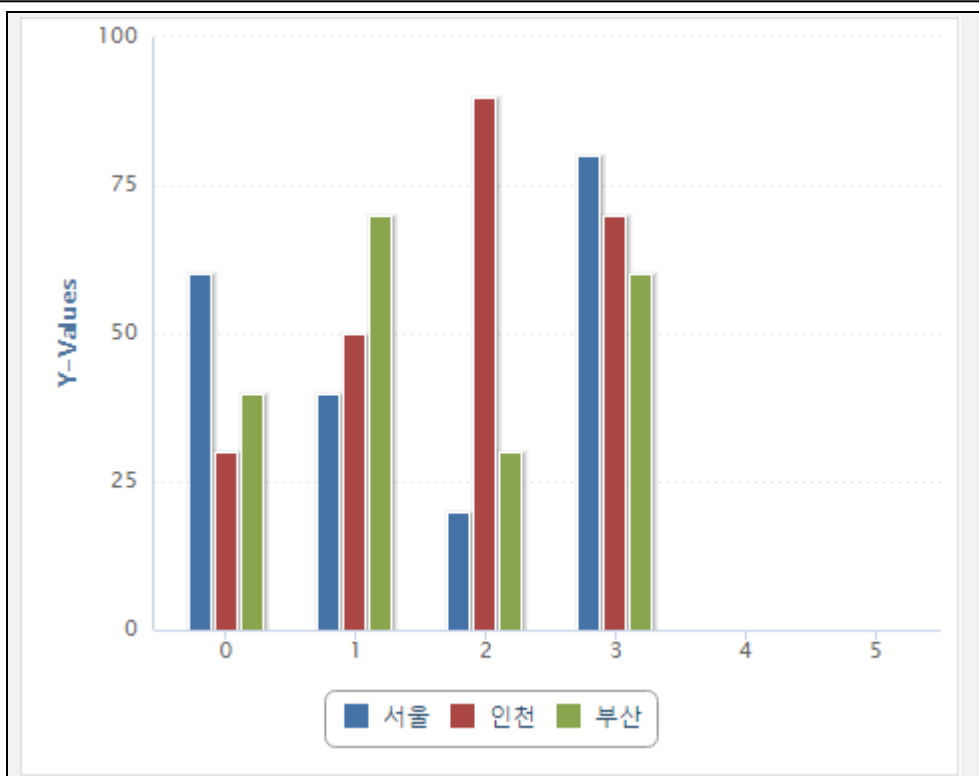
축의 최대값을 설정 한다.

➤ Info

name	type	required	description
max	number	N	축의 최대값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    max: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

tickWidth

➤ 기능

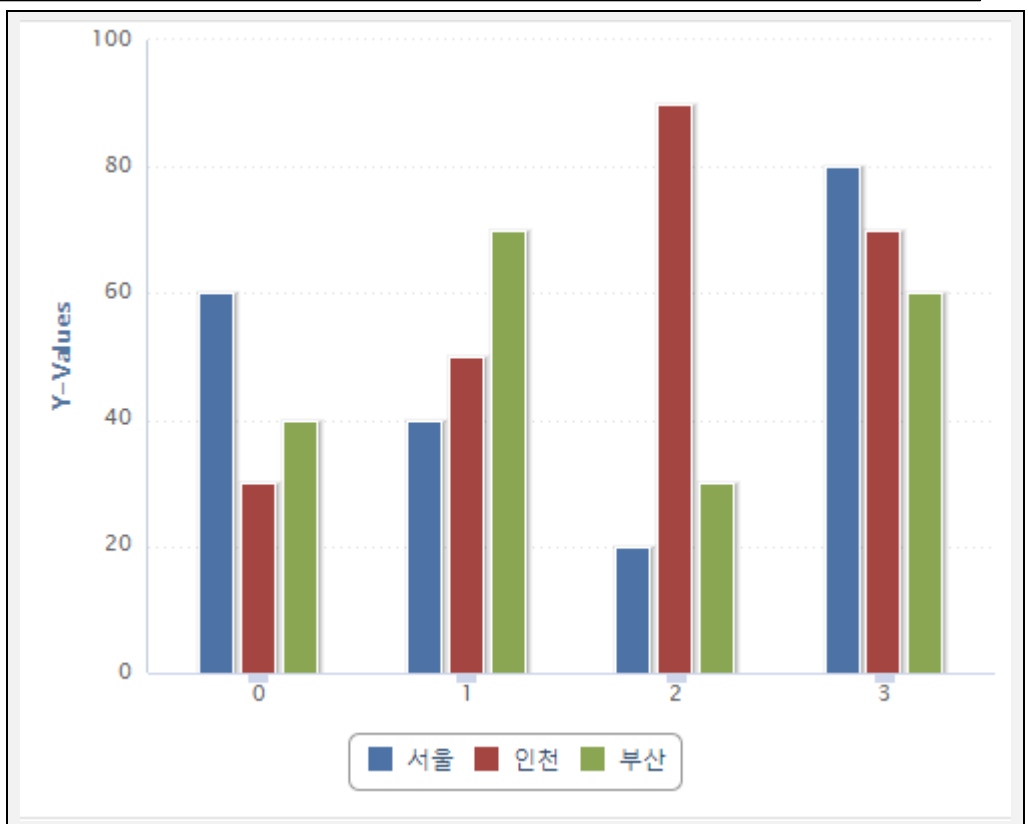
축에 표시할 점의 간격을 설정 한다.

➤ Info

name	type	required	description
tickInterval	number	N	축에 표시할 점의 간격

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    tickWidth: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

tickInterval

➤ 기능

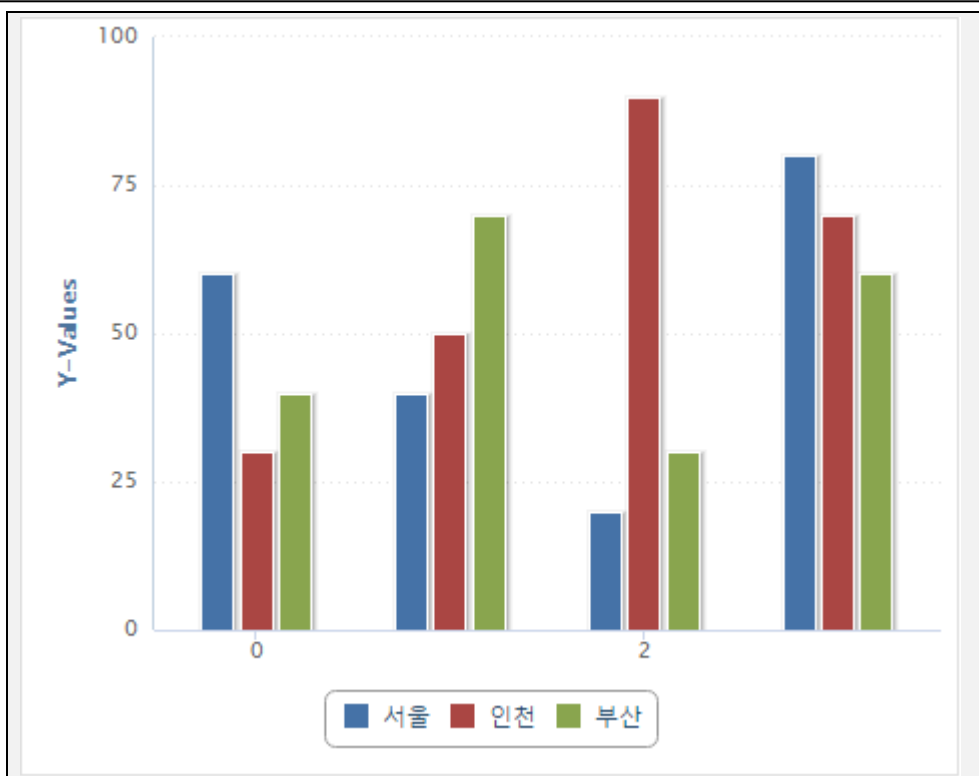
축에 표시할 점의 간격을 설정 한다.

➤ Info

name	type	required	description
tickInterval	number	N	축에 표시할 점의 간격

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    tickInterval: 2
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

gridLineWidth

➤ 기능

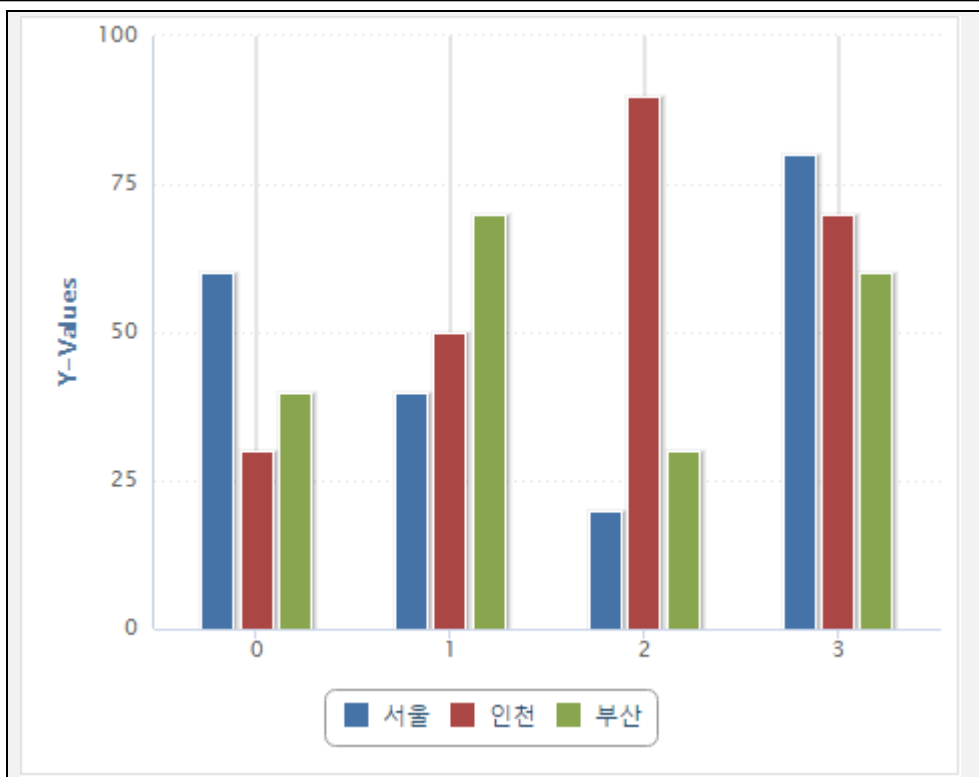
그리드 라인의 너비를 설정 한다.

➤ Info

name	type	required	description
gridLineWidth	number	N(0)	그리드 라인의 너비

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    gridLineWidth: 2
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ **제공 버전**

7.3.0.0	
---------	--

gridLineColor

➤ 기능

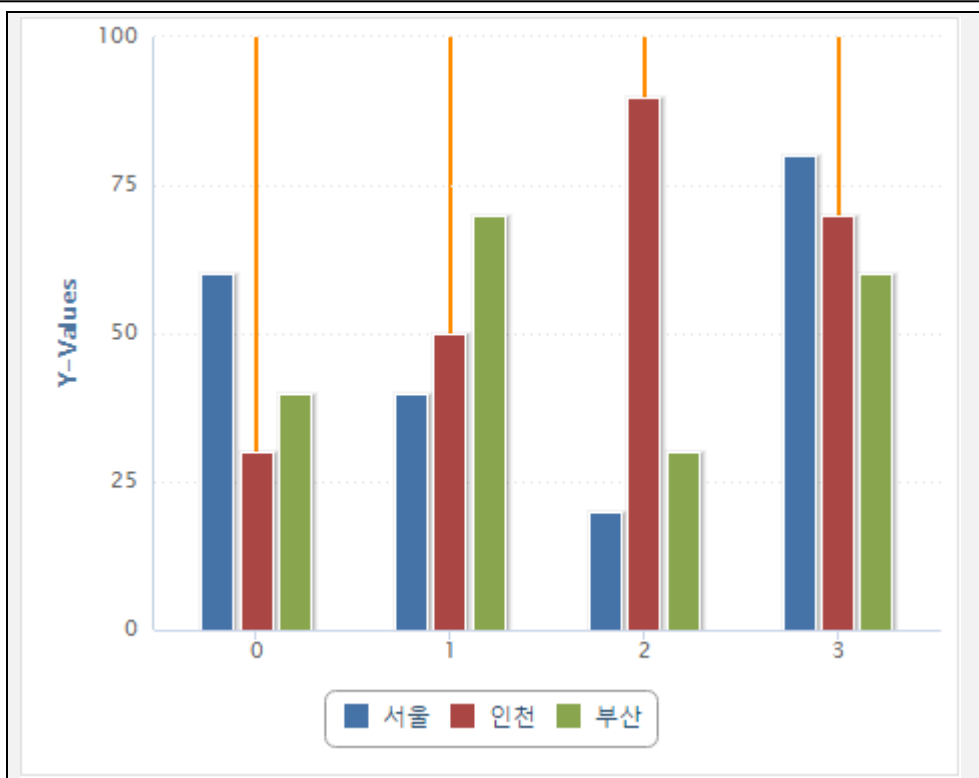
그리드 라인의 색상을 설정 한다.

➤ Info

name	type	required	description
gridLineColor	color	N(#262626)	그리드 라인의 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    gridLineWidth: 2,
    gridLineColor: "#FF8C00"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

gridLineDashStyle

➤ 기능

그리드 라인의 종류를 설정 한다.

➤ Info

name	type	required	description
gridLineDashStyle	string	N(solid)	그리드 라인의 종류

➤ Enum

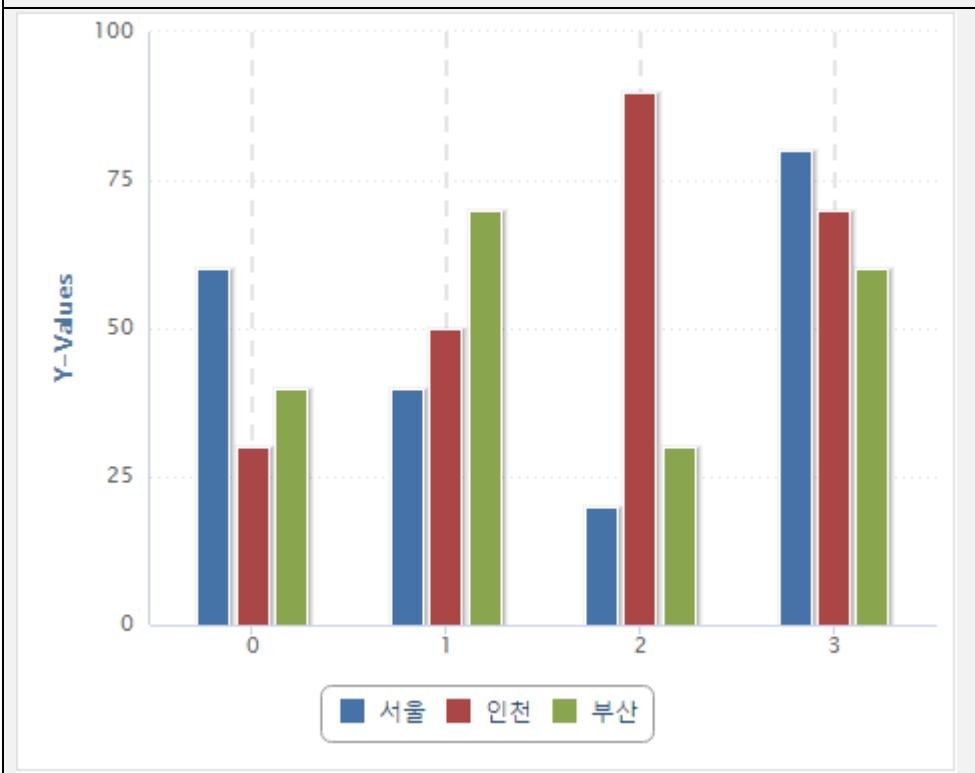
■ gridLineDashStyle

value	description
solid	실선
dash	짧은 점선
longdash	긴 점선
dot	점

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    gridLineWidth: 2,
    gridLineDashStyle: "dash"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
```

```
data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

plotBands

➤ 기능

플롯밴드 속성 (특정영역 하이라이팅)을 설정 한다.

➤ Info

name	type	required	description
plotBands	object	N	플롯밴드 속성
from	number	Y	시작점
to	number	Y	종료점
color	color	Y	색상
label	object	N	라벨 속성

■ plotBands.label Info

name	type	required	description
text	string	N	라벨 문자열
textAlign	string	N	라벨 문자열 정렬 방법
align	string	N(center)	좌우 정렬 방법
rotation	number	N(0)	기울기
useHTML	boolean	N(false)	html tag 사용 여부
style	object	N	스타일 속성

■ plotBands.label.style Info

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기
fontWeight	string	N	폰트 두께

➤ Enum

■ plotBands.label.align / plotBands.label.textAlign

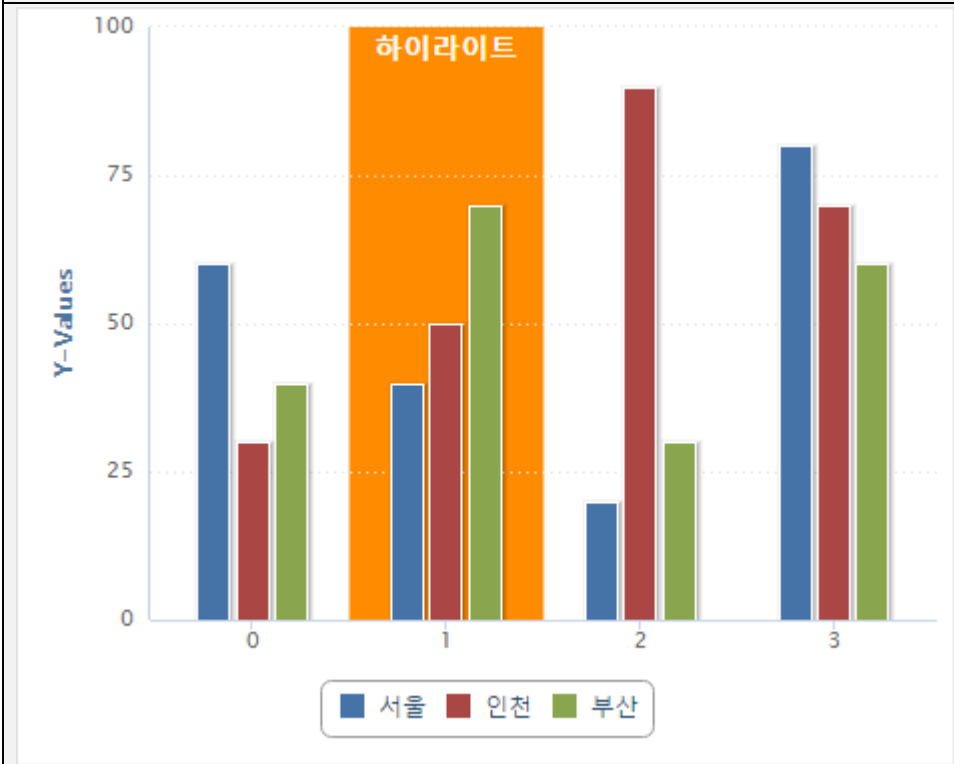
value	description
-------	-------------

center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ **Example**

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    plotBands: {
      from : 0.5,
      to: 1.5,
      color: "#FF8C00",
      label: {
        text: "하이라이트",
        style: {
          color: "#FFFFFF",
          fontWeight: "bold",
          fontSize: "1.2em"
        }
      }
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
```

```
data : [40,70,30,60]
}}
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

offset

➤ 기능

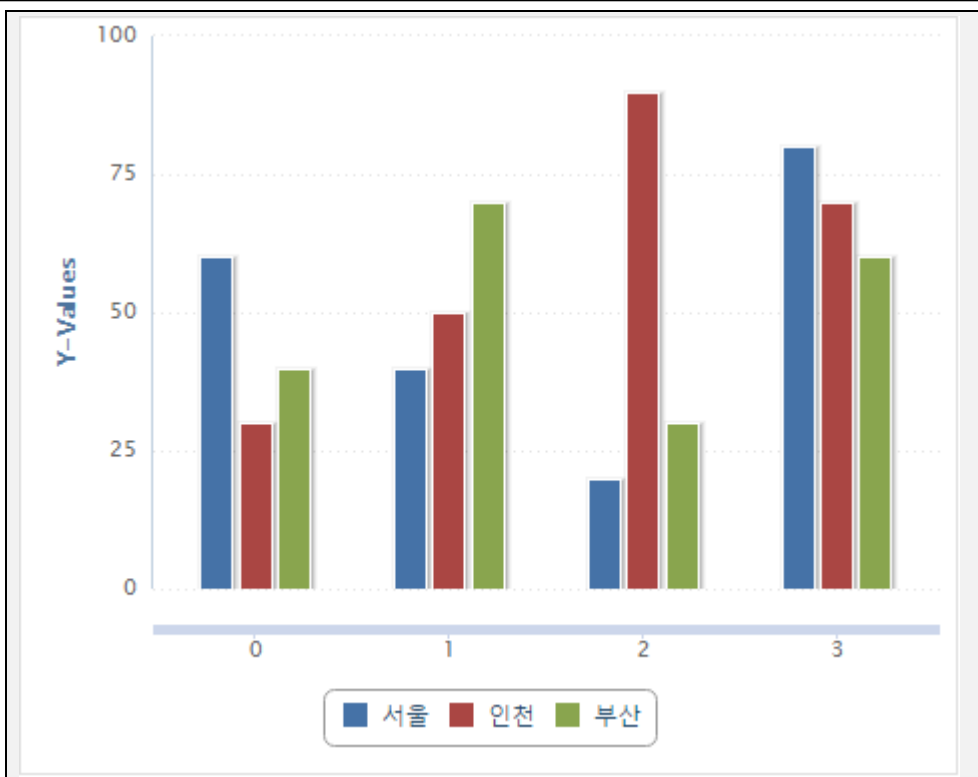
기본 위치 기준으로 표시할 offset 값을 설정 한다.

➤ Info

name	type	required	description
offset	number	N	기본 위치 기준으로 표시할 offset 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    offset: 20,
    lineWidth: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

minorTickInterval

➤ 기능

축에 표시할 보조 점의 간격을 설정 한다.

➤ Info

name	type	required	description
minorTickInterval	number string	N	축에 표시할 보조 점의 간격

➤ Enum

■ minorTickInterval

value	description
auto	간격을 자동으로 설정한다.

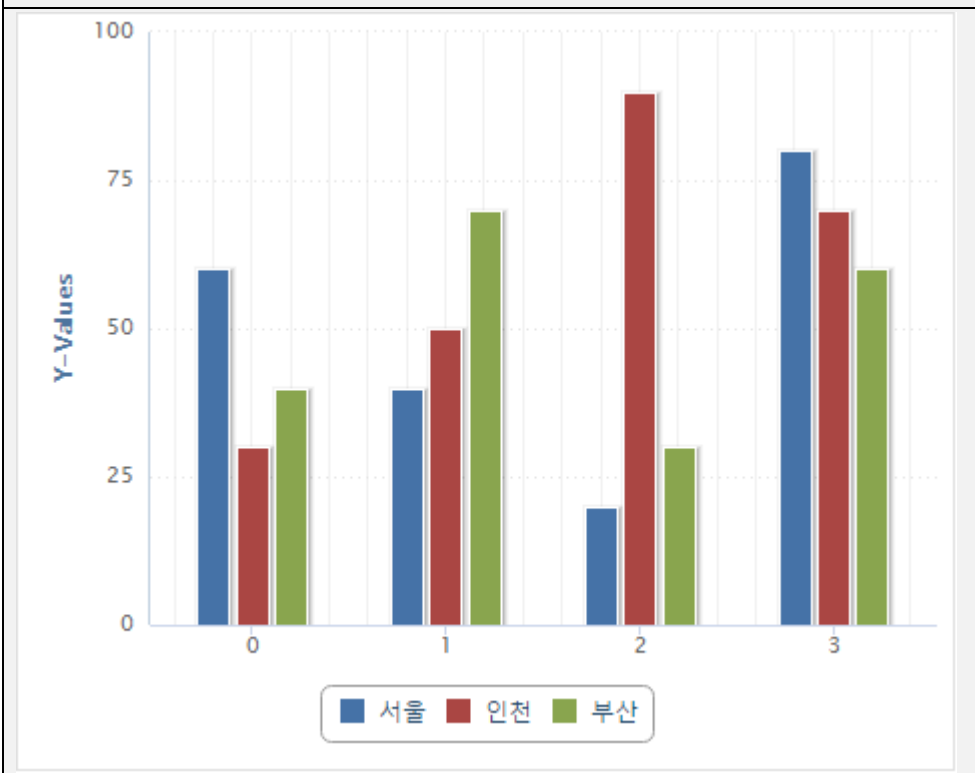
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    minorTickInterval: 0.5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
```

```

    ]]
  }, {
    append: true,
    redraw: true
  });

```



➤ **제공 버전**

7.3.0.0	
---------	--

minorGridLineWidth

➤ 기능

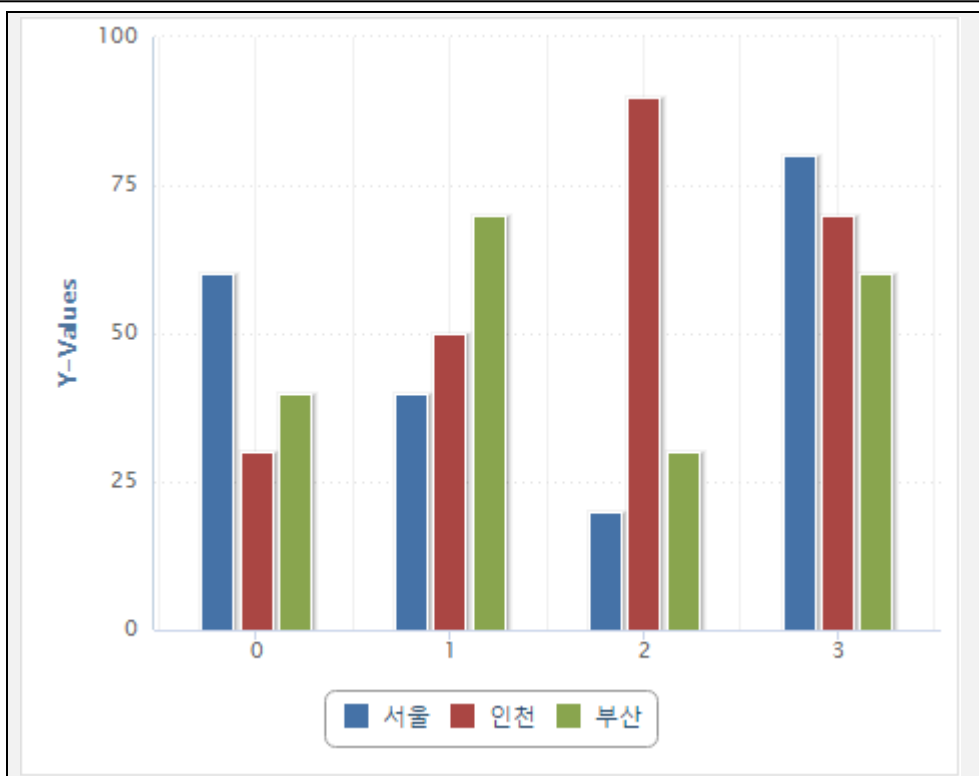
보조 그리드 라인의 너비를 설정 한다.

➤ Info

name	type	required	description
minorGridLineWidth	string	N	보조 그리드 라인의 너비

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    minorGridLineWidth: 5,
    minorTickInterval: 0.5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

minorGridLineColor

➤ 기능

보조 그리드 라인의 색상을 설정 한다.

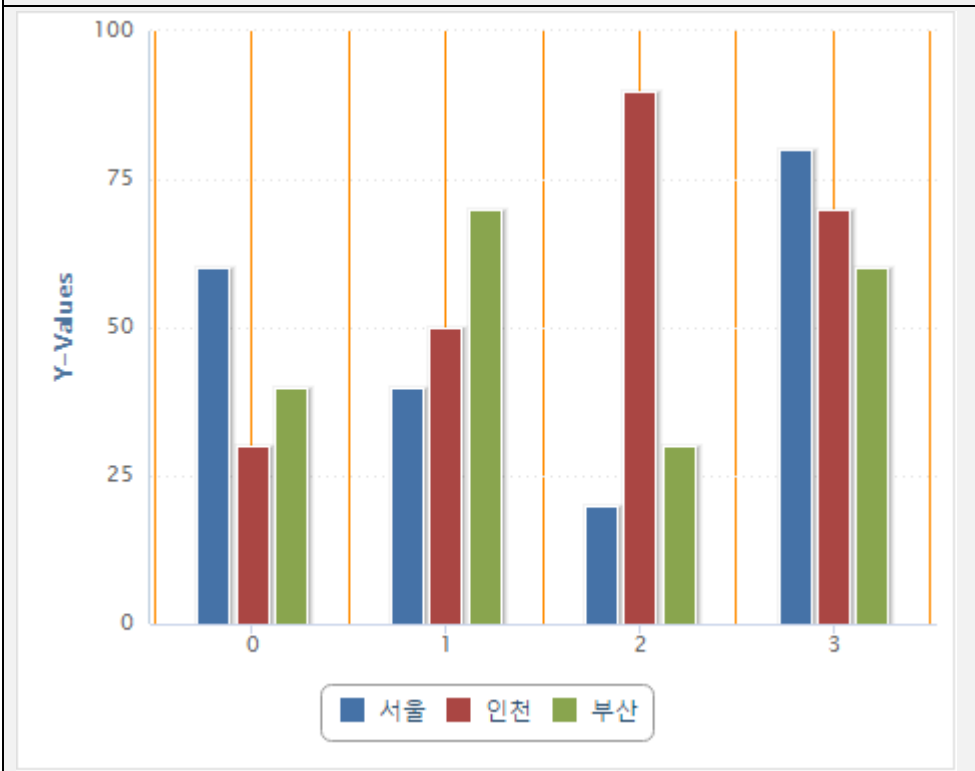
➤ Info

name	type	required	description
minorGridLineColor	color	N	보조 그리드 라인의 색상

➤ Example

```
//
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    minorGridLineColor: '#FF8C00',
    minorGridLineWidth: 5,
    minorTickInterval: 0.5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
```

```
redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

minorGridLineDashStyle

➤ 기능

보조 그리드 라인의 종류를 설정 한다.

➤ Info

name	type	required	description
minorGridLineDashStyle	string	N	보조 그리드 라인의 종류

➤ Enum

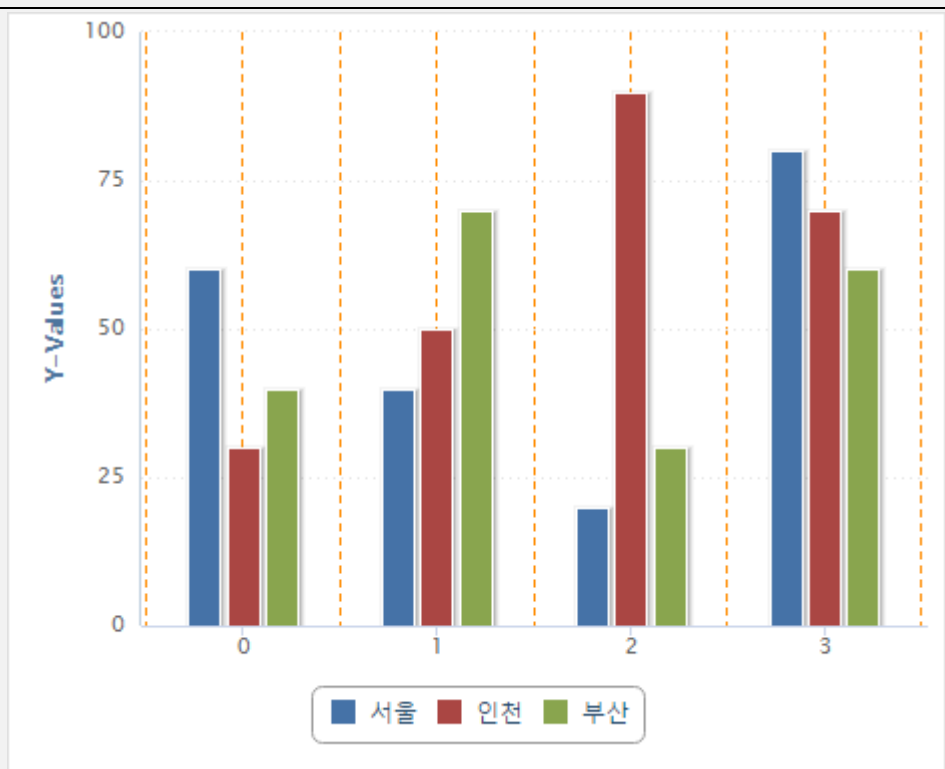
■ gridLineDashStyle

value	description
solid	실선
dash	짧은 점선
longdash	긴 점선
dot	점

➤ Example

```
//
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    minorGridLineColor: '#FF8C00',
    minorGridLineWidth: 5,
    minorGridLineDashStyle: "dash",
    minorTickInterval: 0.5
  },
  series: [{
    name: "서울",
```

```
data : [60,40,20,80]
},{
  name: "인천",
  data : [30,50,90,70]
},{
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

alternateGridColor

➤ 기능

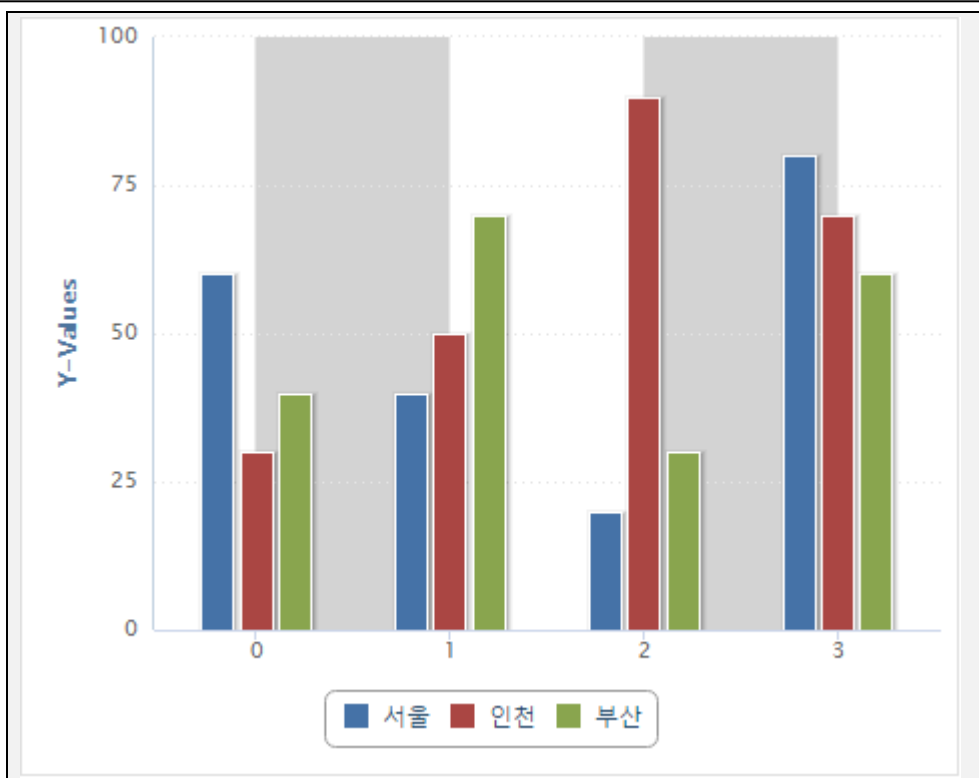
그리드 라인의 교차 색상을 설정 한다.

➤ Info

name	type	required	description
alternateGridColor	color	N	그리드 라인의 교차 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  xAxis: {
    alternateGridColor: "#D3D3D3"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

10.9 yAxis

Y축에 대한 구성 옵션을 설정 한다.

➤ Info

name	type	required	description
title	object	N	축 제목 속성 (JSON 형식)
categories	array	N	축레이블 설정
crosshair	boolean	N(false)	포인트에 마우스 오버시 가이드라인 표시 여부
reversed	boolean	N(false)	축 반전 여부
showFirstLabel	boolean	N(true)	첫번째 축 라벨여부 표시
showLastLabel	boolean	N(true)	마지막 축 라벨여부 표시
lineWidth	number	1	축 라인의 너비
lineColor	color	N(#ccd6eb)	축 라인의 색상
labels	object	N	라벨 속성
labels.align	string	N(center)	좌우 정렬 방법
labels.format	string	N({value})	축 라벨의 포맷 문자열 (값은 {value} 문자열을 이용한다.)
labels.formatter	function	N	축 라벨의 포맷을 정의하는 callback 함수
labels.style	object	N	라벨 스타일 속성
labels.useHTML	boolean	N(false)	html tag 사용 여부
labels.rotation	number	N(0)	기울기 (0 ~ 360)
labels.step	number	N	표시 단계
labels.enabled	boolean	N(true)	레이블 사용 여부
opposite	boolean	N(false)	우측에 축 표시 여부
min	number	N	축의 최소값
max	number	N	축의 최대값

tickInterval	number	N	축에 표시할 점의 간격
gridLineWidth	number	N(0)	그리드 라인의 너비
gridLineColor	color	N(#262626)	그리드 라인의 색상
gridLineDashStyle	string	N(solid)	그리드 라인의 종류
plotBands	object	N	플롯밴드 속성 (특정영역 하이라이팅)
offset	number	N	기본 위치 기준으로 표시할 offset 값
minorTickInterval	number string	N	축에 표시할 보조 점의 간격
minorGridLineWidth	string	N	보조 그리드 라인의 너비
minorGridLineColor	color	N	보조 그리드 라인의 색상
minorGridLineDashStyle	string	N	보조 그리드 라인의 종류
alternateGridColor	color	N	그리드 라인의 교차 색상
stackLabels	object	N	누적 레이블 사용여부

title

➤ 기능

축 제목 속성을 설정 한다.

➤ Info

name	type	required	description
title	object	N	축 제목 속성 (JSON 형식)
title.text	string	N	제목 문자열
title.align	string	N(center)	좌우 정렬 값
title.x	number	N(0)	좌우 포지션 이동 값
title.y	number	N	상하 포지션 이동 값
title.style	object	N	스타일 설정
title.style.color	color	N	폰트 색상
title.style.fontSize	string	N	폰트 크기
title.style.fontWeight	string	N	폰트 두께
title.rotation	number	N(0)	타이틀 회전
title.enabled	boolean	N(true)	타이틀 사용여부(default: true)

➤ Enum

■ title.align

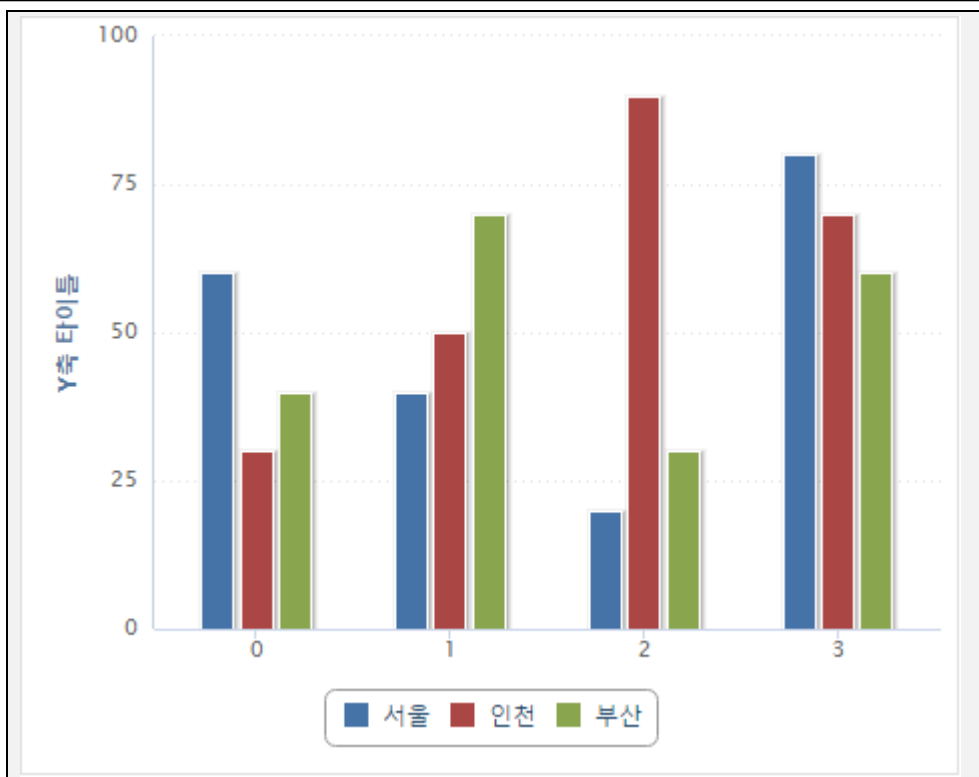
value	description
center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
```



```
yAxis: {  
  title: {  
    text: "Y축 타이틀"  
  }  
},  
series: [{  
  name: "서울",  
  data : [60,40,20,80]  
},{  
  name: "인천",  
  data : [30,50,90,70]  
},{  
  name: "부산",  
  data : [40,70,30,60]  
}]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

categories

➤ 기능

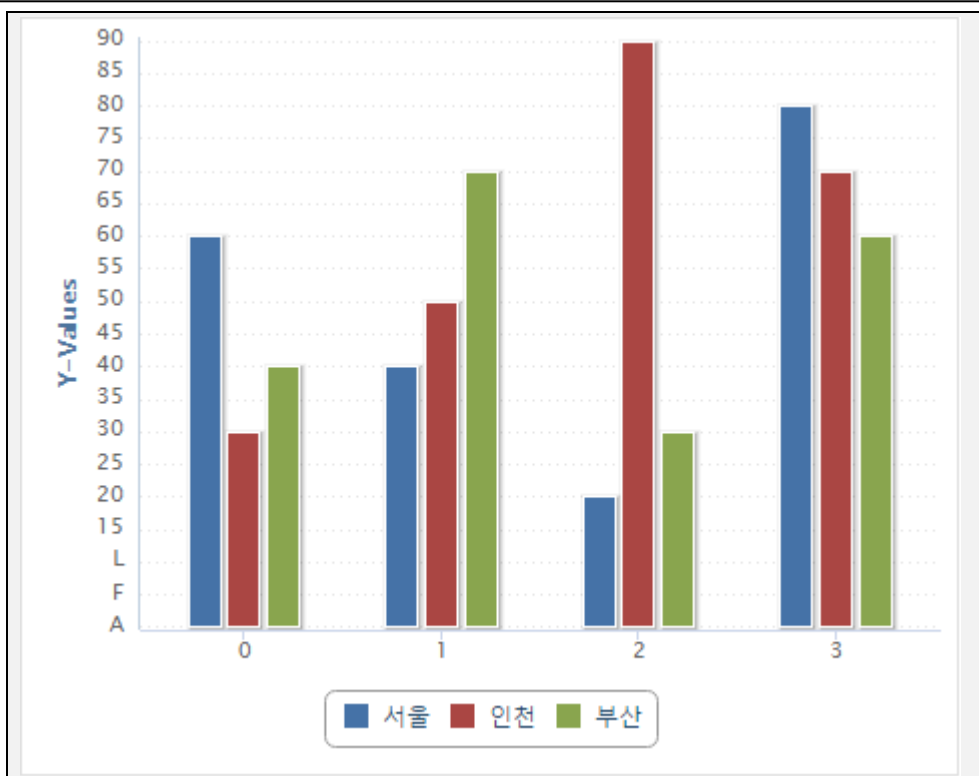
축 레이블을 설정 한다.

➤ Info

name	type	required	description
categories	array	N	축 레이블 설정

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    categories: ["A","B","C","D","E","F","H","I","J","K","L","N"]
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

crosshair

➤ 기능

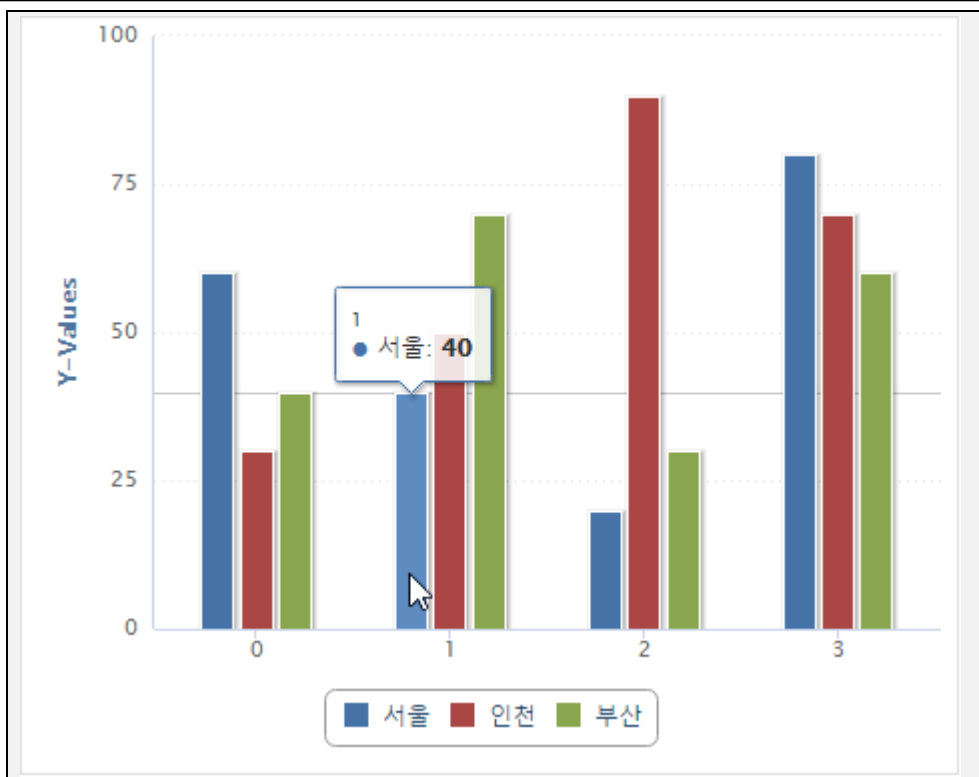
포인트에 마우스 오버시 가이드라인 표시 여부를 설정 한다.

➤ Info

name	type	required	description
crosshair	boolean	N(false)	포인트에 마우스 오버시 가이드라인 표시 여부

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    crosshair: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

reversed

➤ 기능

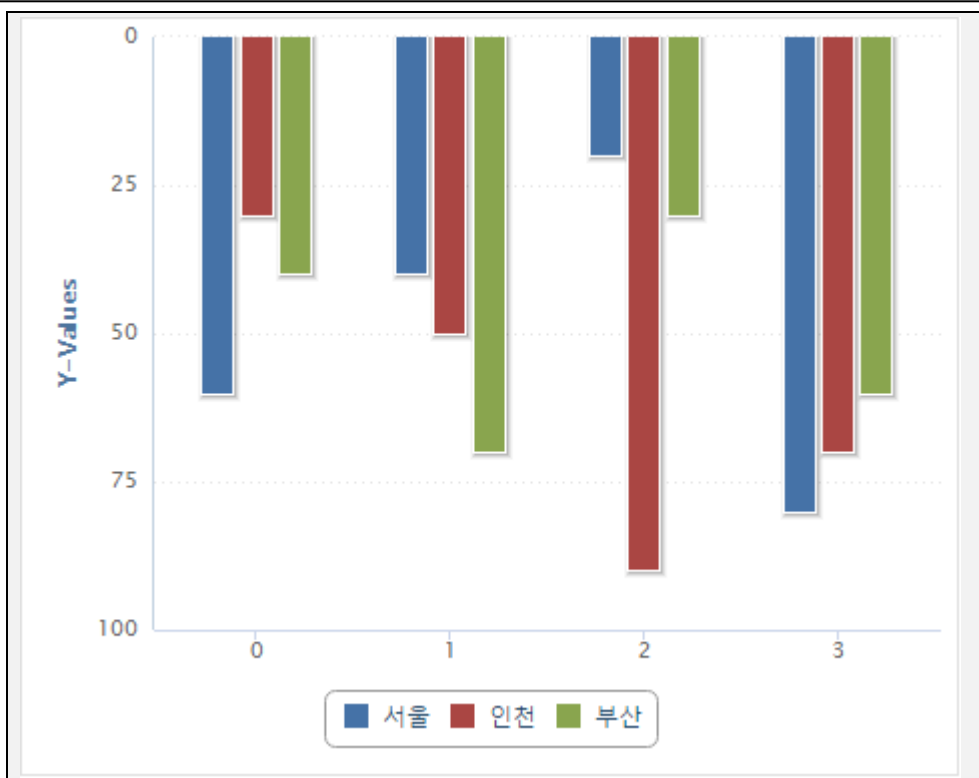
축 반전 여부를 설정 한다.

➤ Info

name	type	required	description
reversed	boolean	N(false)	축 반전 여부

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    reversed: true
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

showFirstLabel

➤ 기능

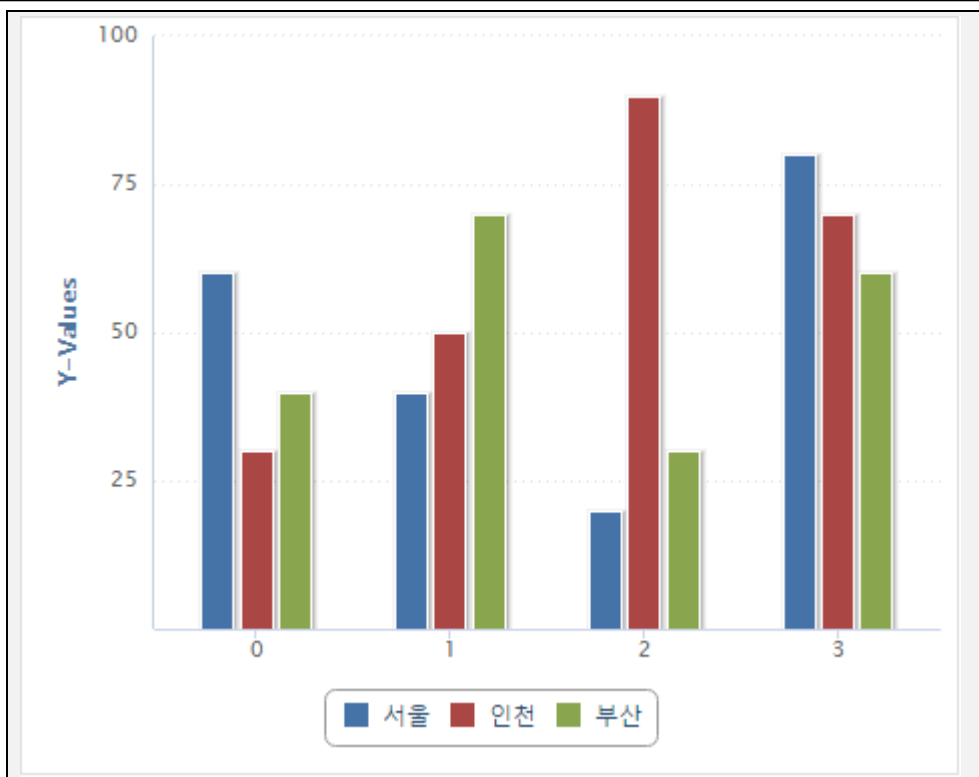
첫번째 축 라벨여부 표시 여부를 설정 한다.

➤ Info

name	type	required	description
showFirstLabel	boolean	N(true)	첫번째 축 라벨여부 표시

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    showFirstLabel: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

showLastLabel

➤ 기능

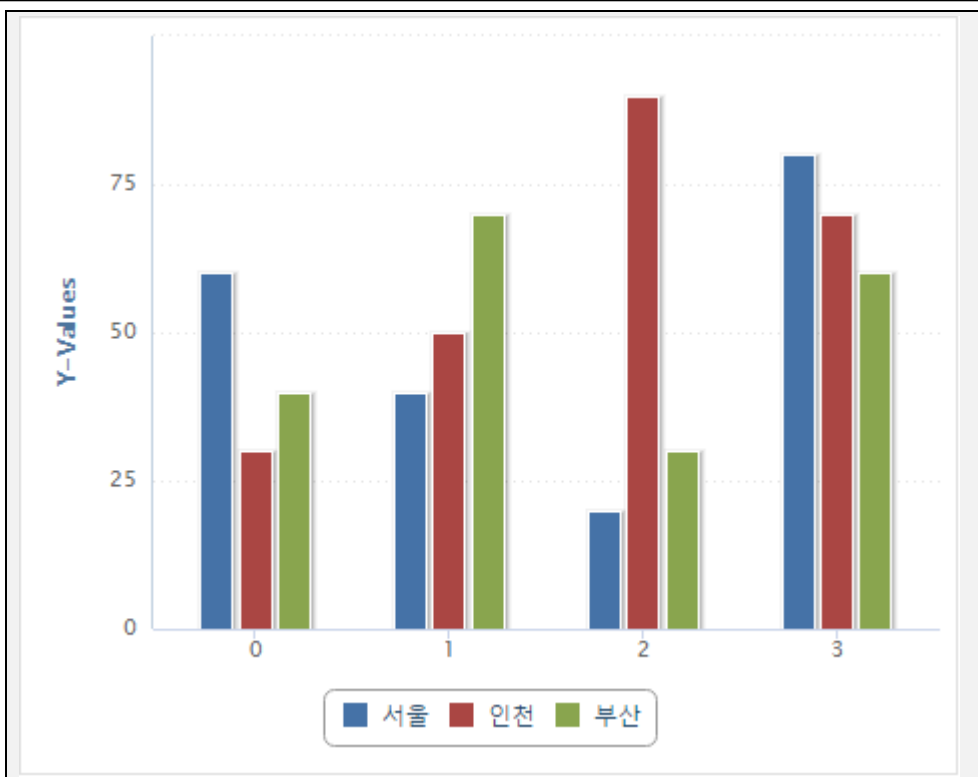
마지막 축 라벨여부 표시 여부를 설정 한다.

➤ Info

name	type	required	description
showLastLabel	boolean	N(true)	마지막 축 라벨여부 표시

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    showLastLabel: false
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

lineWidth

➤ 기능

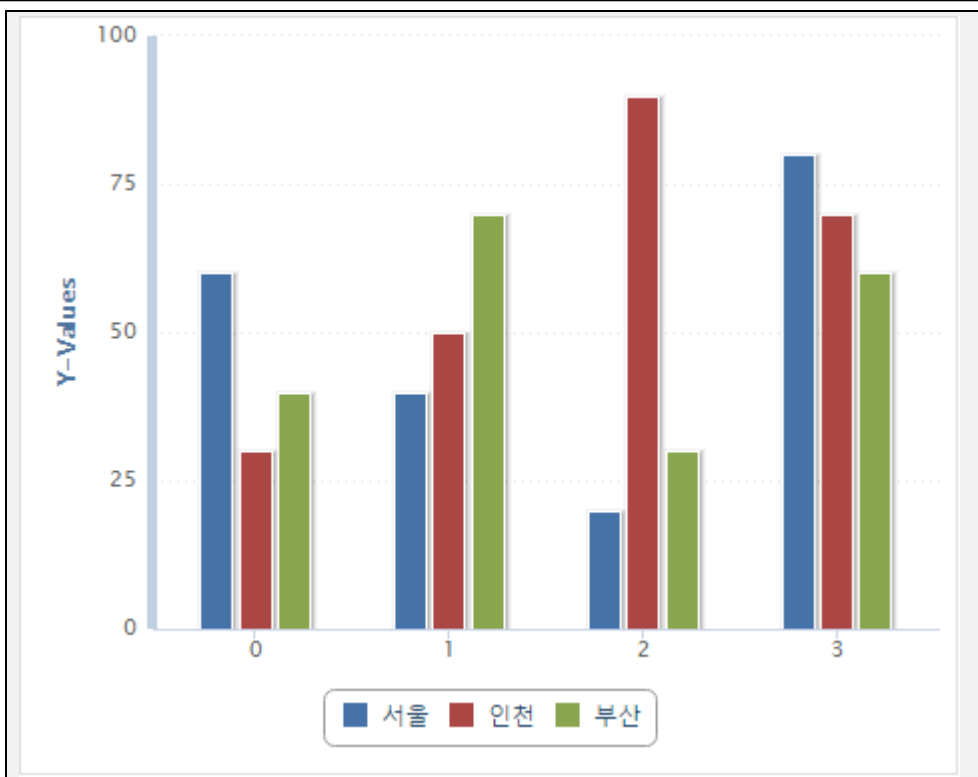
축 라인의 너비를 설정 한다.

➤ Info

name	type	required	description
lineWidth	number	1	축 라인의 너비

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    lineWidth: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  },{
    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

lineColor

➤ 기능

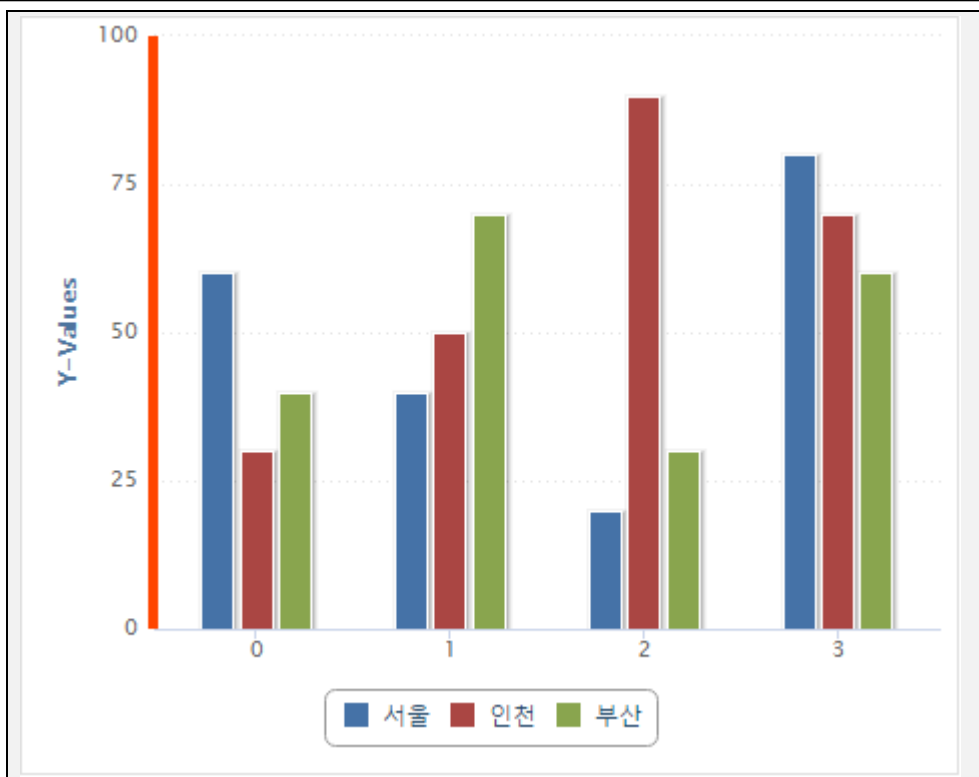
축 라인의 색상을 설정 한다.

➤ Info

name	type	required	description
lineColor	color	N(#ccd6eb)	축 라인의 색상

➤ Example

```
myChart.setOptions({
  chart:{
    type: "column"
  },
  yAxis: {
    lineWidth: 5,
    lineColor: "#FF4500"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

labels

➤ 기능

라벨 속성을 설정 한다.

➤ Info

name	type	required	description
labels	object	N	라벨 속성
labels.align	string	N(center)	좌우 정렬 방법
labels.format	string	N({value})	축 라벨의 포맷 문자열 (값은 {value} 문자열을 이용 한다.) value는 data를 의미.
labels.formatter	function	N	축 라벨의 포맷을 정의하는 callback 함수 (값은 {this.value}) this.value는 현재 data를 의미.
labels.style	object	N	라벨 스타일 속성
labels.useHTML	boolean	N(false)	html tag 사용 여부
labels.rotation	number	N(0)	기울기 (0 ~ 360)
labels.step	number	N	표시 단계
labels.enabled	boolean	N(true)	레이블 사용 여부

■ labels.style Info

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기
fontWeight	string	N	폰트 두께

➤ Enum

■ Labels.align

value	description
center	중앙 정렬

left	좌측 정렬
right	우측 정렬

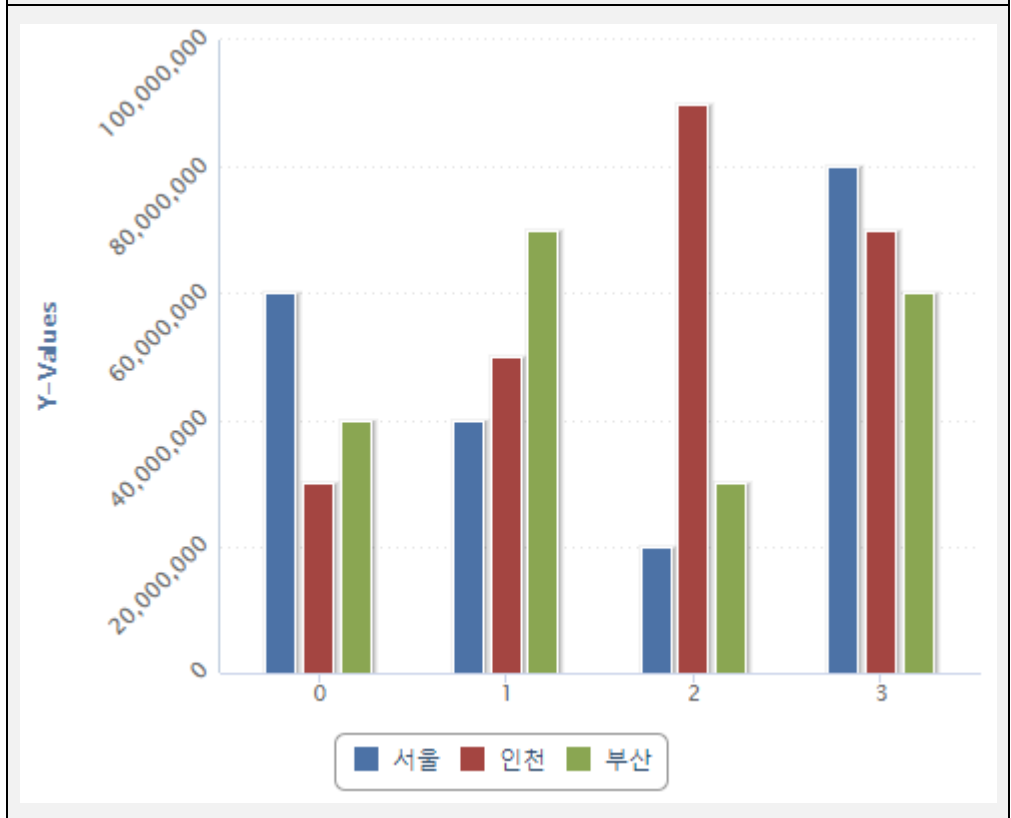
■ labels.format

value	description
{value}	x축 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    labels: {
      style: {
        fontWeight: "bold"
      },
      rotation: -45,
      formatter: function () { //천단위 콤마 찍기
        var regexp = /\B(?=(\d{3})+(?!\d))/g;
        return this.value.toString().replace(regexp, ',');
      }
    }
  },
  series: [{
    name: "서울",
    data : [60000000,40000000,20000000,80000000]
  },{
    name: "인천",
    data : [30000000,50000000,90000000,70000000]
  },{
    name: "부산",
```

```
data : [40000000,70000000,30000000,60000000]
}}
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

opposite

➤ 기능

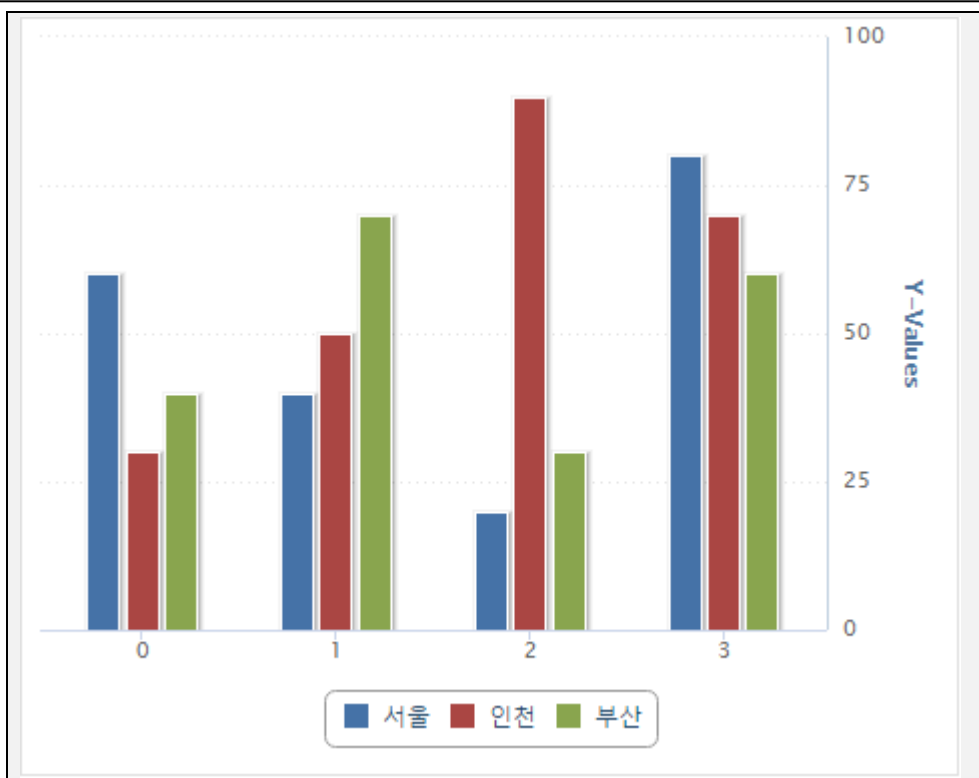
차트 우측에 축 표시 여부를 설정 한다.

➤ Info

name	type	required	description
opposite	boolean	N(false)	우측에 축 표시 여부

➤ Example

```
myChart.setOptions({  
  chart: {  
    type: "column"  
  },  
  yAxis: {  
    opposite: true  
  },  
  series: [{  
    name: "서울",  
    data : [60,40,20,80]  
  }, {  
    name: "인천",  
    data : [30,50,90,70]  
  }, {  
    name: "부산",  
    data : [40,70,30,60]  
  }]  
}, {  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

min

➤ 기능

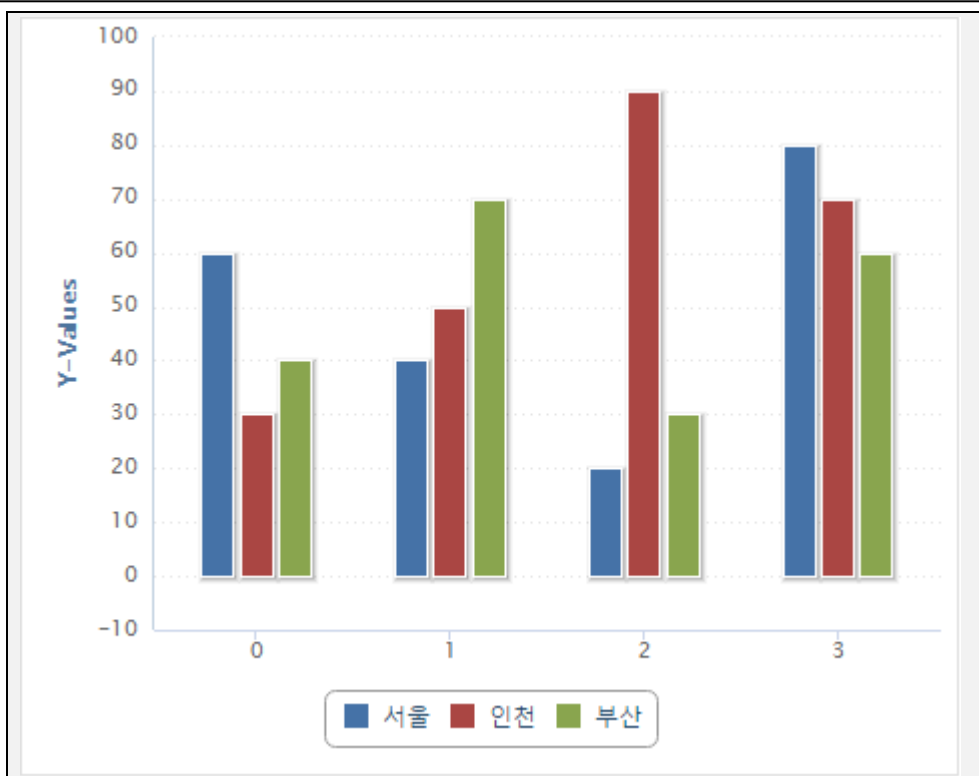
축의 최소값을 설정 한다.

➤ Info

name	type	required	description
min	number	N	축의 최소값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    tickInterval: 10,
    min: -10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

max

➤ 기능

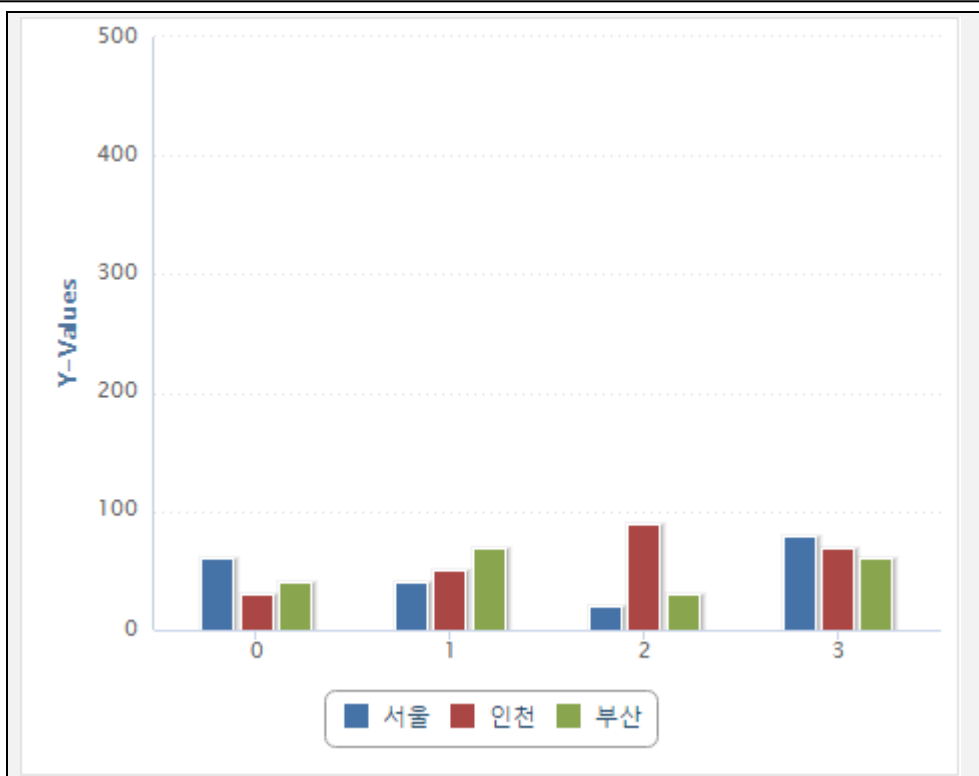
축의 최대값을 설정 한다.

➤ Info

name	type	required	description
max	number	N	축의 최대값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    max: 500
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```

➤ **제공 버전**

7.3.0.0	
---------	--

tickInterval

➤ 기능

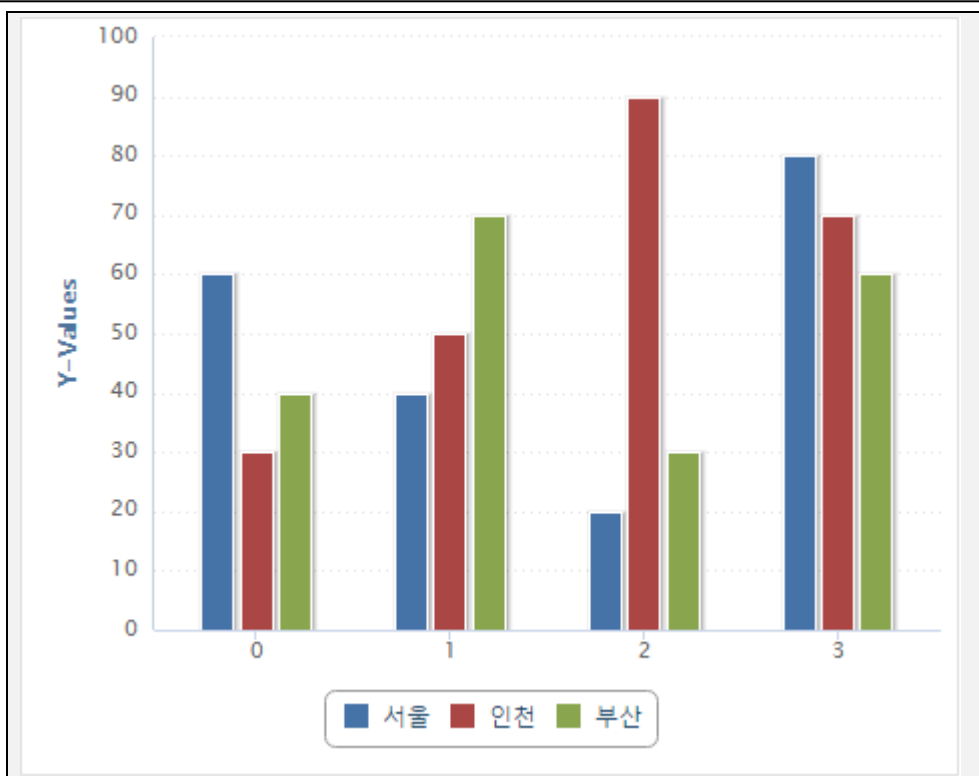
축에 표시할 점의 간격을 설정 한다.

➤ Info

name	type	required	description
tickInterval	number	N	축에 표시할 점의 간격

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    tickInterval: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

gridLineWidth

➤ 기능

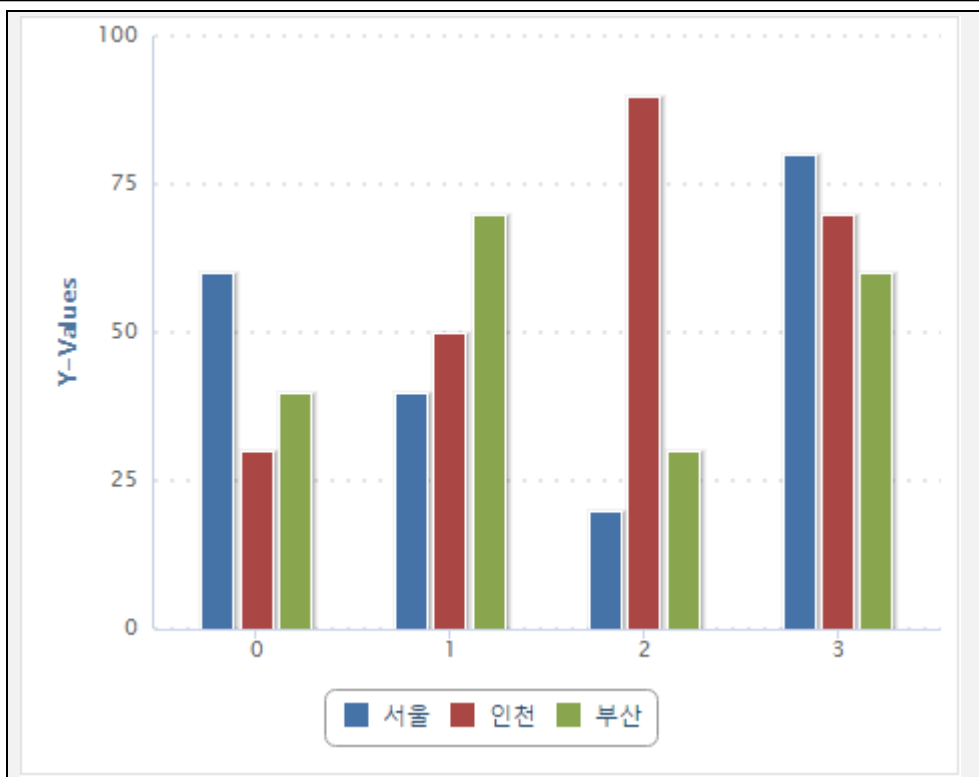
그리드 라인의 너비를 설정 한다.

➤ Info

name	type	required	description
gridLineWidth	number	N(0)	그리드 라인의 너비

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineWidth: 2
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

gridLineColor

➤ 기능

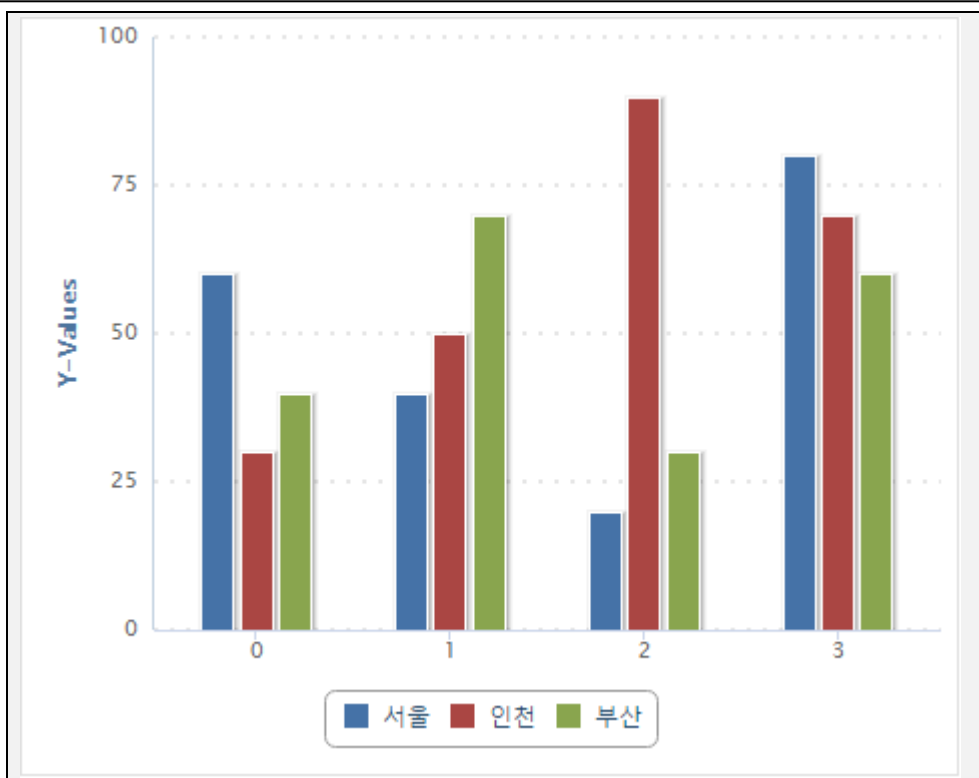
그리드 라인의 색상을 설정 한다.

➤ Info

name	type	required	description
gridLineColor	color	N(#262626)	그리드 라인의 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineColor: "#FF8C00"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

gridLineDashStyle

➤ 기능

그리드 라인의 종류를 설정 한다.

➤ Info

name	type	required	description
gridLineDashStyle	string	N(solid)	그리드 라인의 종류

➤ Enum

■ gridLineDashStyle

value	description
solid	실선
dash	짧은 점선
longdash	긴 점선
dot	점

➤ Example

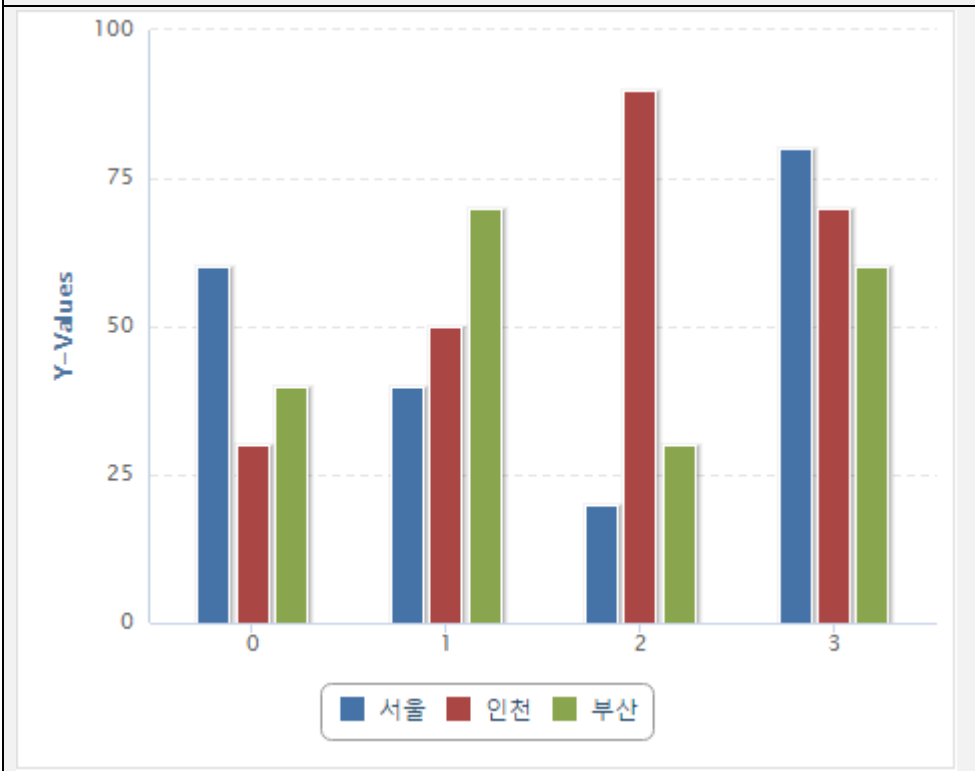
```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineDashStyle: "dash"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }
}]
```



```

}, {
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});

```



➤ **제공 버전**

7.3.0.0	
---------	--

plotBands

➤ 기능

플롯밴드 속성 (특정영역 하이라이팅)을 설정 한다.

➤ Info

name	type	required	description
plotBands	object	N	플롯밴드 속성
from	number	Y	시작점
to	number	Y	종료점
color	color	Y	색상
label	object	N	라벨 속성

■ plotBands.label Info

name	type	required	description
text	string	N	라벨 문자열
textAlign	string	N	라벨 문자열 정렬 방법
align	string	N(left)	상하 정렬 방법
rotation	number	N(0)	기울기
useHTML	boolean	N(false)	html tag 사용 여부
style	object	N	스타일 속성

■ plotBands.label.style Info

name	type	required	description
color	color	N	폰트 색상
fontSize	string	N	폰트 크기
fontWeight	string	N	폰트 두께

➤ Enum

■ plotBands.label.align / plotBands.label.textAlign

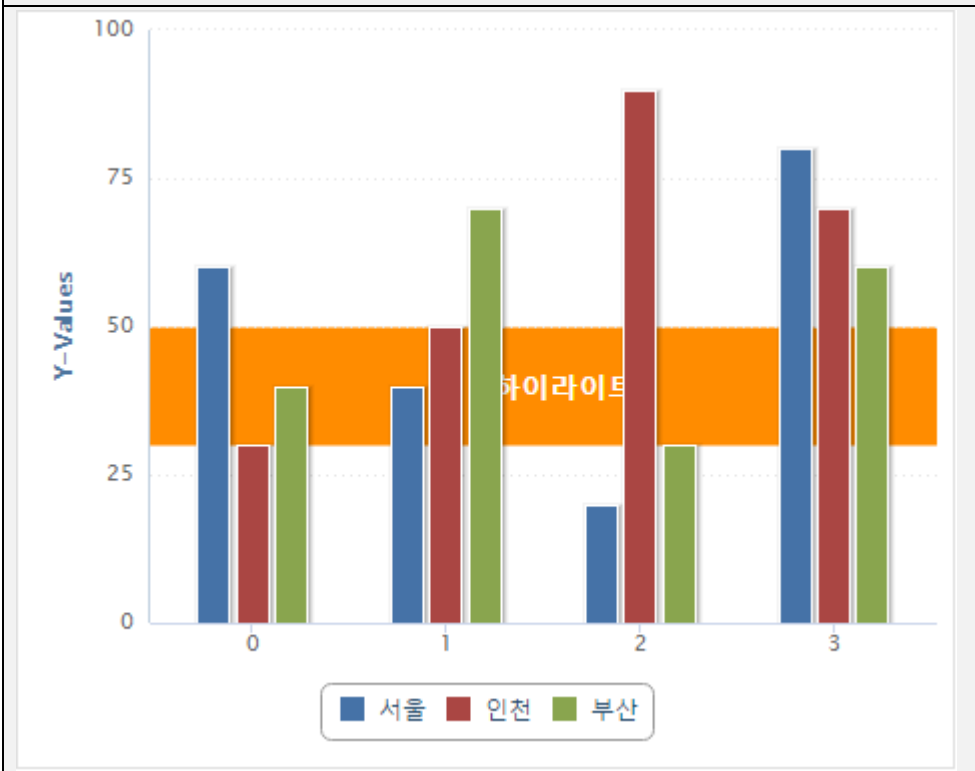
value	description
-------	-------------

center	중앙 정렬
left	좌측 정렬
right	우측 정렬

➤ **Example**

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    plotBands: {
      from : 30,
      to: 50,
      color: "#FF8C00",
      label: {
        text: "하이라이트",
        align: "center",
        style: {
          color: "#FFFFFF",
          fontWeight: "bold",
          fontSize: "1.2em"
        }
      }
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
```

```
name: "부산",
data : [40,70,30,60]
}]
}, {
append: true,
redraw: true
});
```



➤ **제공 버전**

7.3.0.0

offset

➤ 기능

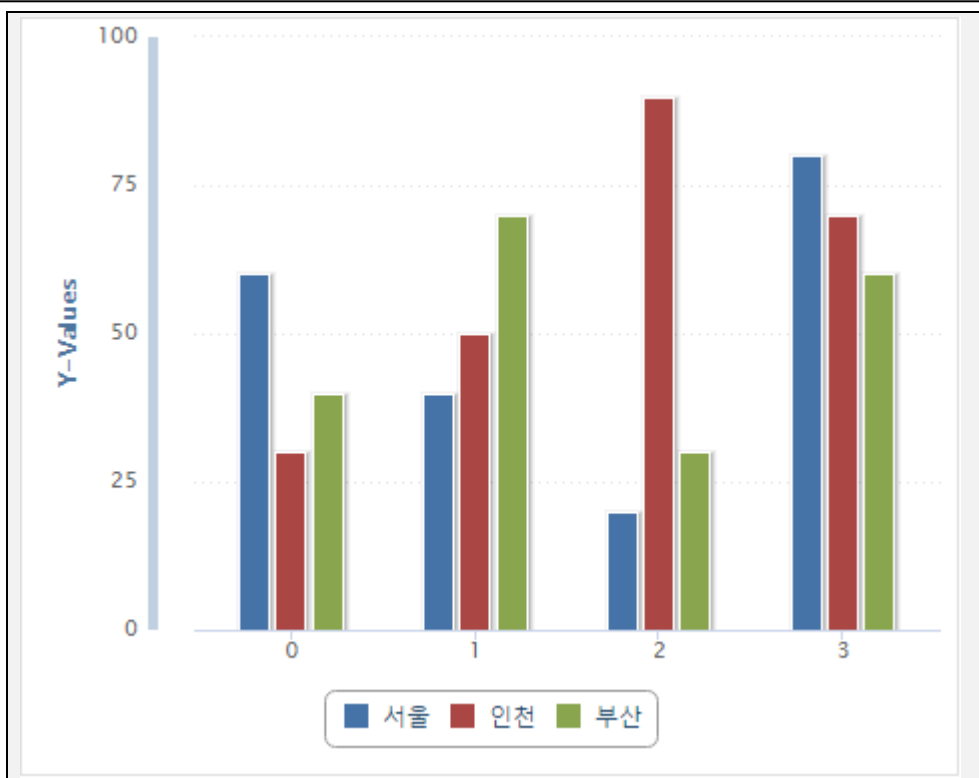
기본 위치 기준으로 표시할 offset 값을 설정 한다.

➤ Info

name	type	required	description
offset	number	N	기본 위치 기준으로 표시할 offset 값

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    offset: 20,
    lineWidth: 5
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

minorTickInterval

➤ 기능

축에 표시할 보조 점의 간격을 설정 한다.

➤ Info

name	type	required	description
minorTickInterval	number string	N	축에 표시할 보조 점의 간격

➤ Enum

■ minorTickInterval

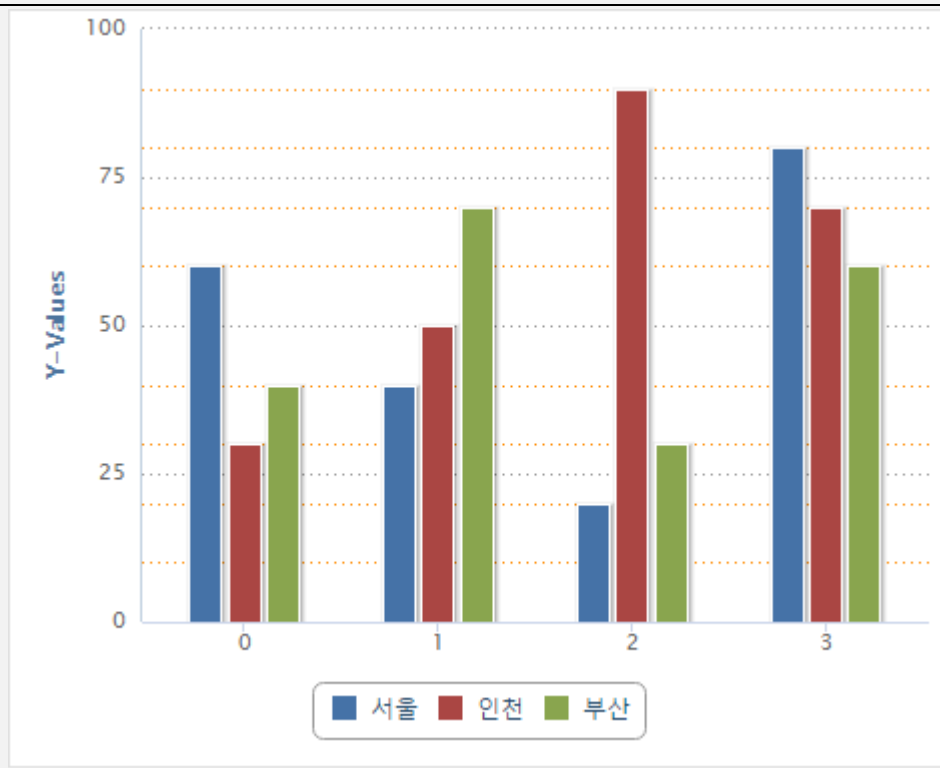
value	description
auto	간격을 자동으로 설정한다.

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineColor: "#999999",
    minorGridLineColor: '#FF8C00',
    minorTickInterval: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
```

```
name: "부산",
data : [40,70,30,60]

}]
}, {
append: true,
redraw: true
});
```



➤ **제공 버전**

7.3.0.0

minorGridLineWidth

➤ 기능

보조 그리드 라인의 너비를 설정 한다.

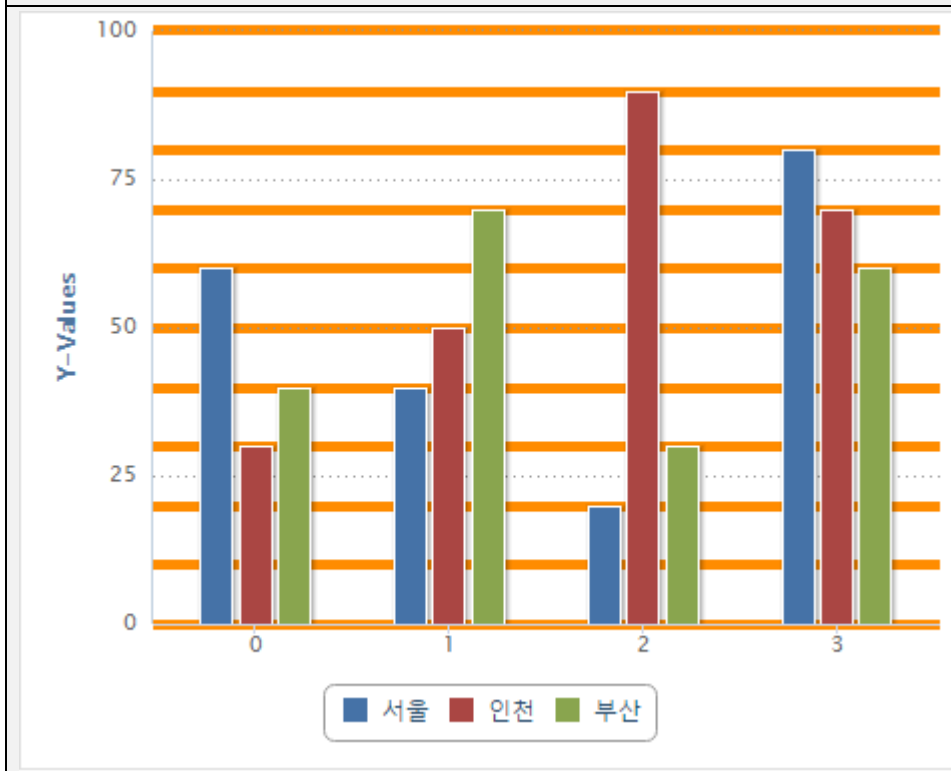
➤ Info

name	type	required	description
minorGridLineWidth	string	N	보조 그리드 라인의 너비

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineColor: "#999999",
    minorGridLineWidth: 5,
    minorTickInterval: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
})
```

});



➤ **제공 버전**

7.3.0.0

minorGridLineColor

➤ 기능

보조 그리드 라인의 색상을 설정 한다.

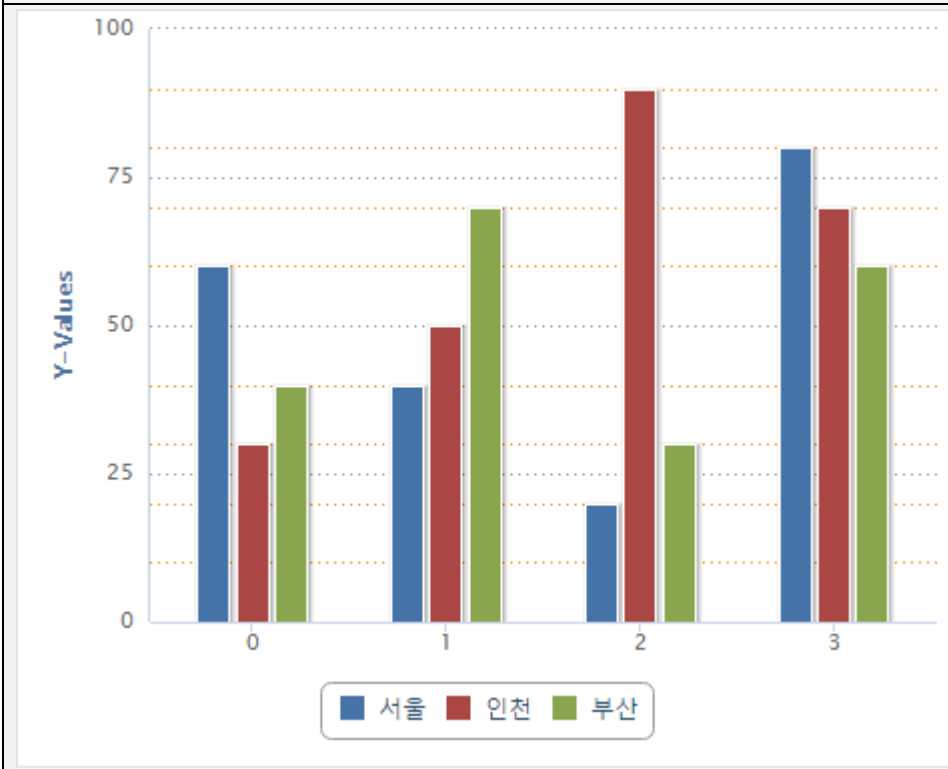
➤ Info

name	type	required	description
minorGridLineColor	color	N	보조 그리드 라인의 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineColor: "#999999",
    minorGridLineColor: '#FF8C00',
    minorTickInterval: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
})
```

});



➤ **제공 버전**

7.3.0.0

minorGridLineDashStyle

➤ 기능

보조 그리드 라인의 종류를 설정 한다.

➤ Info

name	type	required	description
minorGridLineDashStyle	string	N	보조 그리드 라인의 종류

➤ Enum

■ gridLineDashStyle

value	description
solid	실선
dash	짧은 점선
longdash	긴 점선
dot	점

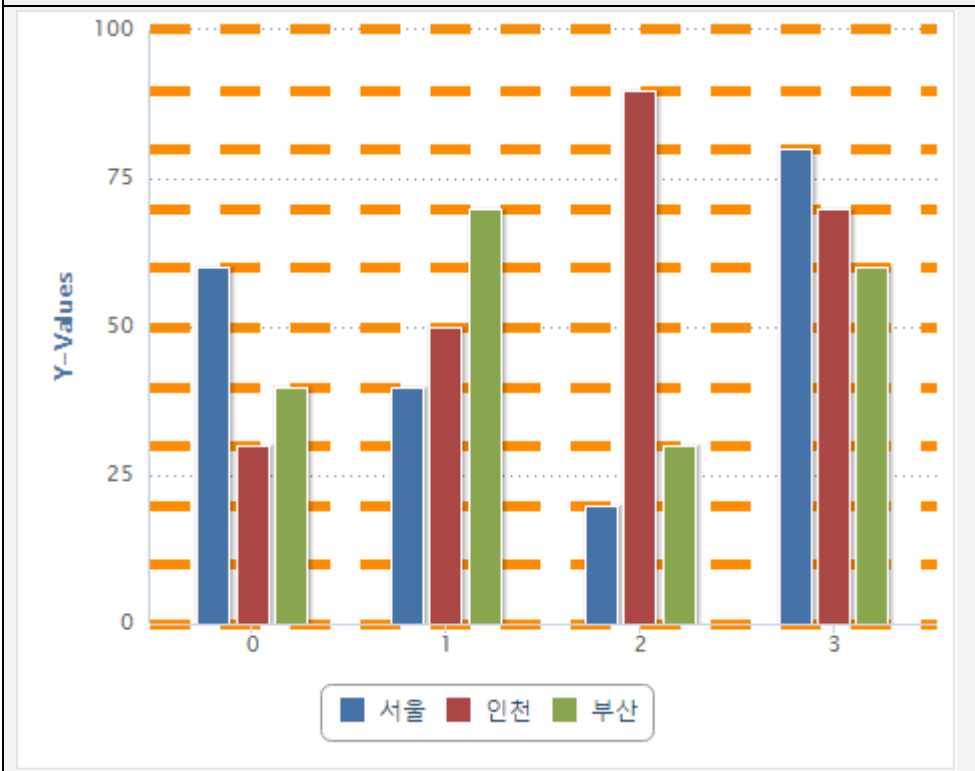
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    gridLineColor: "#999999",
    minorGridLineDashStyle: "dash",
    minorGridLineColor: '#FF8C00',
    minorTickInterval: 10
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
```

```

}, {
  name: "인천",
  data : [30,50,90,70]
}, {
  name: "부산",
  data : [40,70,30,60]
}]
}, {
  append: true,
  redraw: true
});

```



➤ 제공 버전

7.3.0.0

alternateGridColor

➤ 기능

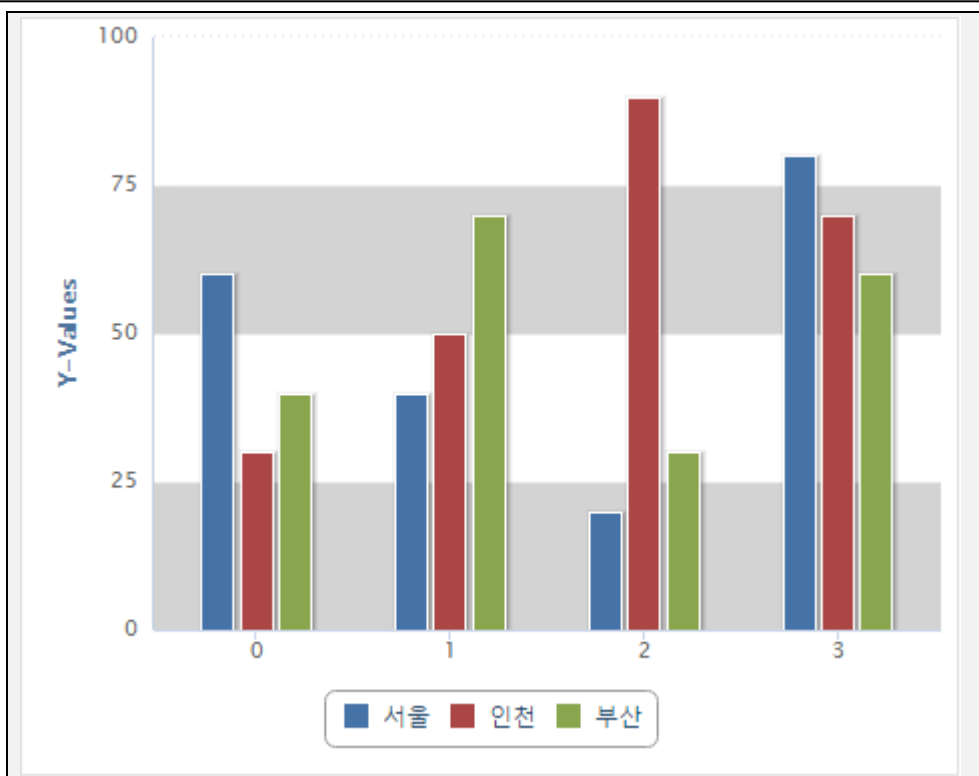
그리드 라인의 교차 색상을 설정 한다.

➤ Info

name	type	required	description
alternateGridColor	color	N	그리드 라인의 교차 색상

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    alternateGridColor: "#D3D3D3"
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

stackLabels

➤ 기능

누적형 레이블 사용여부(합계값 표시)를 표시한다.

➤ Info

name	type	required	description
enabled	boolean	N	누적형 레이블 사용여부 true일 시 사용
align	string	N	누적형 레이블 위치 (left,right,center)

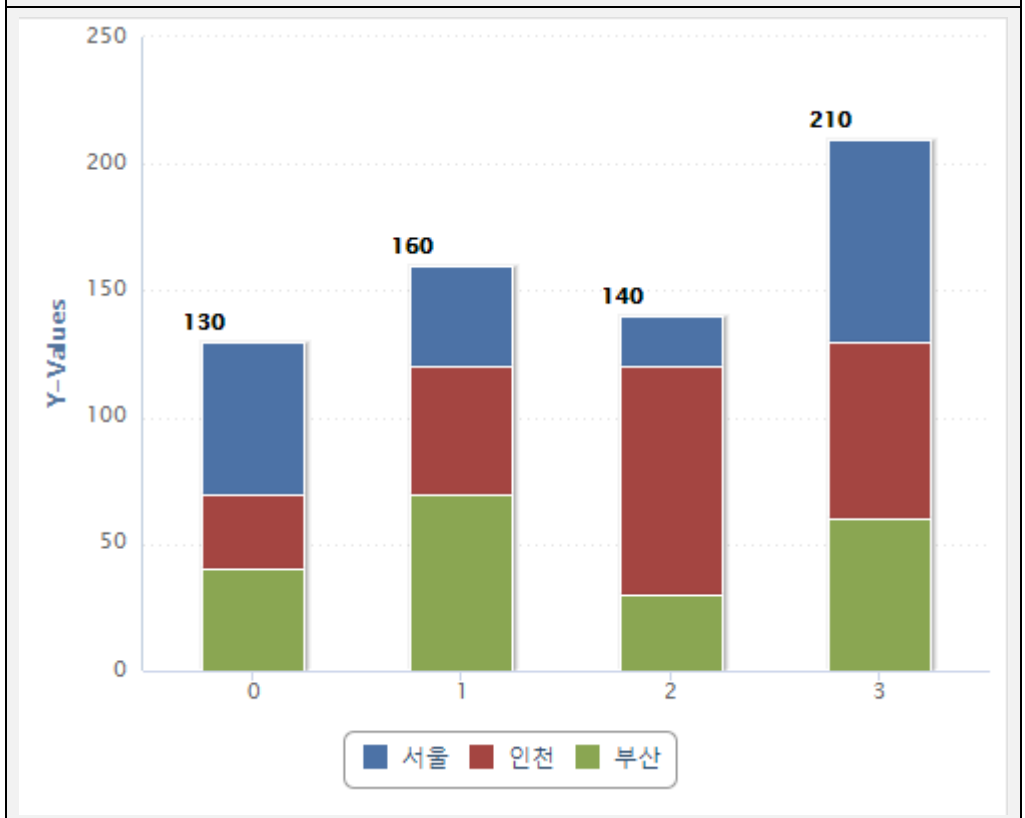
➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    stackLabels: {
      enabled: true,
      align: "left"
    }
  },
  plotOptions: {
    column: {
      stacking: "normal"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
```

```

    name: "인천",
    data : [30,50,90,70]
  },{
    name: "부산",
    data : [40,70,30,60]
  }
], {
  append: true,
  redraw: true
});

```



➤ 제공 버전

7.3.0.0

Appendix

1. 특수차트

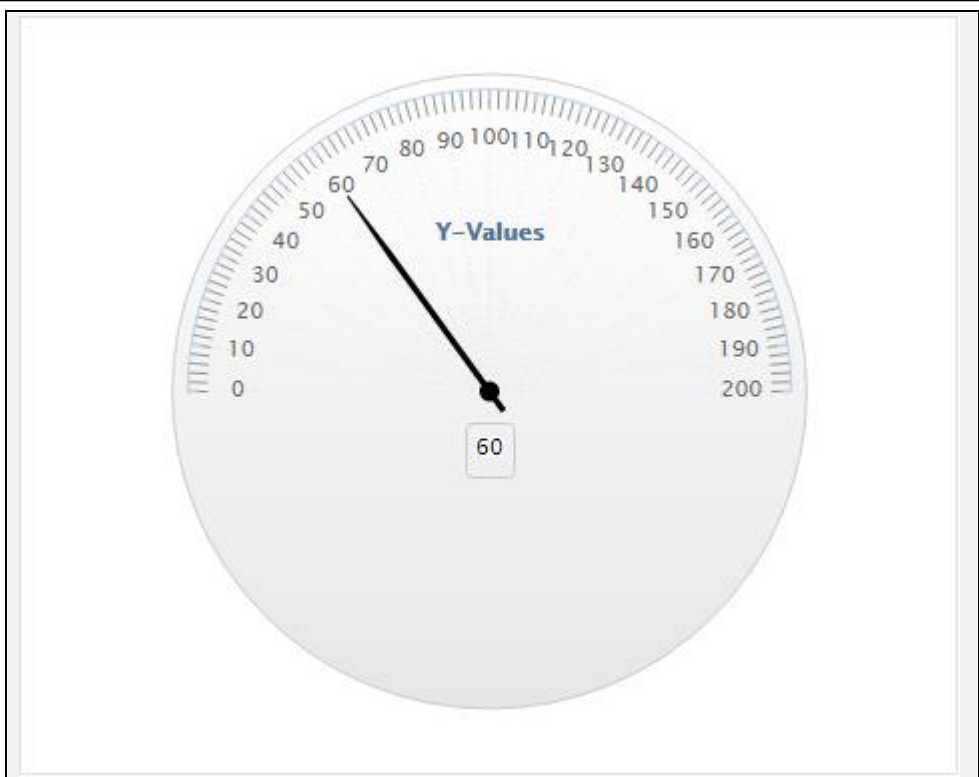
1.1 Angular gauges chart

➤ **기능**

gauge 차트를 표현한다. (highcharts/highcharts-more.js 필요)

➤ **Example**

```
myChart.setOptions({
  chart: {
    type: "gauge"
  },
  pane: {
    startAngle: -90,
    endAngle: 90
  },
  yAxis: {
    max:200,
    min:0,
    tickInterval : 10
  },
  series: [{
    data : [60]
  }]
},{
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

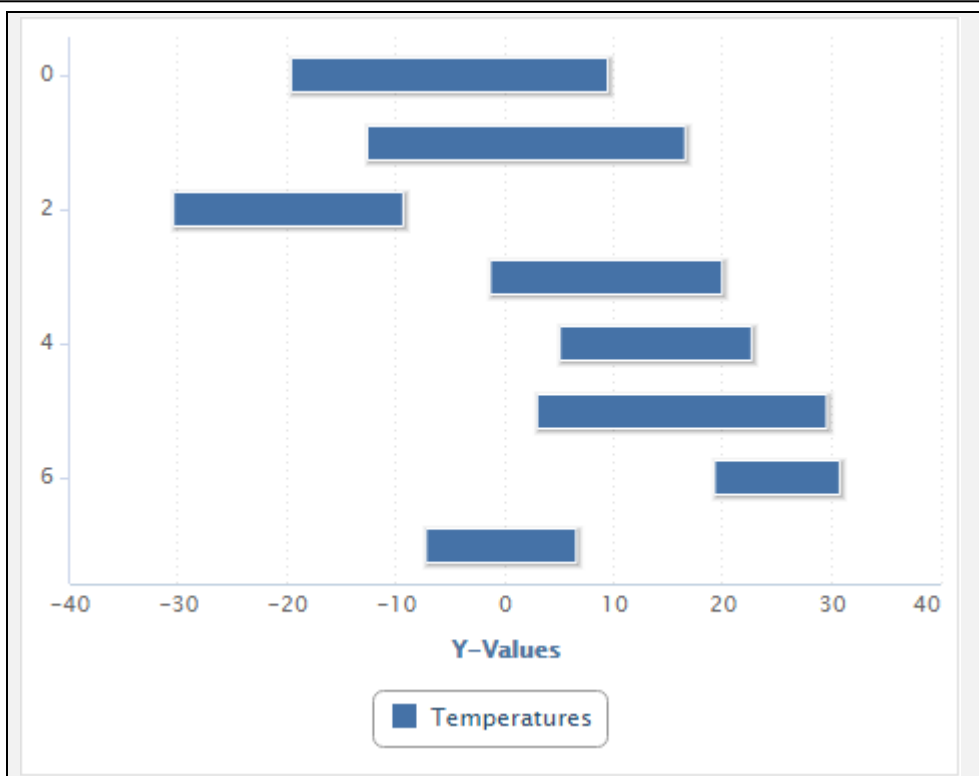
1.2 ColumnRange chart

➤ 기능

ColumnRange 차트를 표현한다. (highcharts/highcharts-more.js 필요)

➤ Example

```
myChart.setOptions({
  chart:{
    type:'columnrange',
    inverted:true
  },
  series: [{
    name: 'Temperatures',
    data: [
      [-19.7, 9.4],
      [-12.7, 16.5],
      [-30.5, -9.4],
      [-1.4, 19.9],
      [5.0, 22.6],
      [2.9, 29.5],
      [19.2, 30.7],
      [-7.3, 6.5]
    ]
  }]
},{
  append: true,
  redraw: true
});
```

➤ 제공 버전

7.3.0.0	
---------	--

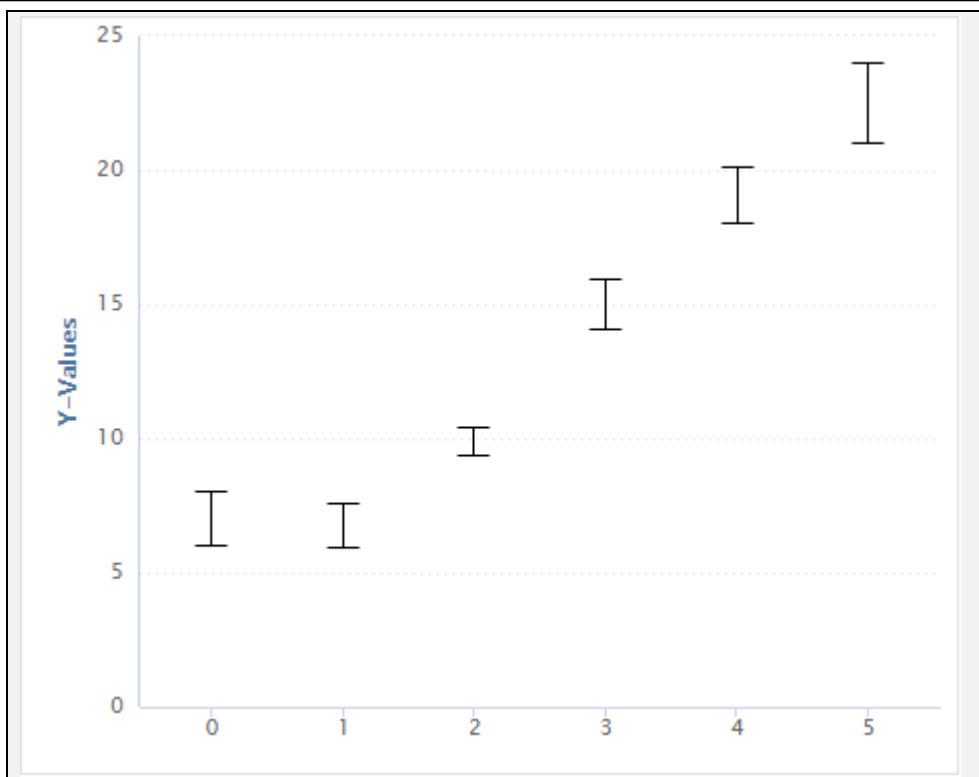
1.3 ErrorBar chart

➤ **기능**

ErrorBar 차트를 표현한다. (highcharts/highcharts-more.js 필요)

➤ **Example**

```
myChart.setOptions({
  chart:{
    type:'errorbar'
  },
  series: [{
    name: 'Temperatures',
    data: [[6, 8], [5.9, 7.6], [9.4, 10.4], [14.1, 15.9], [18.0,
20.1], [21.0, 24.0]]
  }]
},{
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

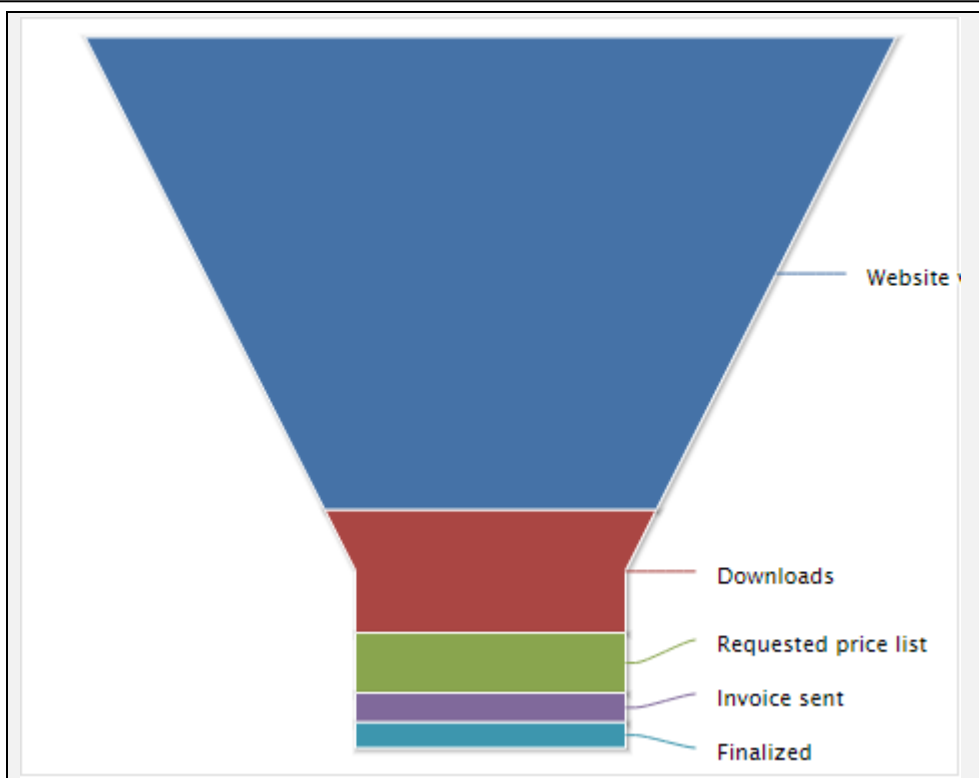
1.4 Funnel chart

➤ 기능

깔때기형 차트를 표현한다. (highcharts/modules/funnel.js 필요)

➤ Example

```
myChart.setOptions({  
  chart:{  
    type:'funnel'  
  },  
  series: [{  
    name: 'Unique users',  
    data: [  
      ['Website visits', 15654],  
      ['Downloads', 4064],  
      ['Requested price list', 1987],  
      ['Invoice sent', 976],  
      ['Finalized', 846]  
    ]  
  }]  
},{  
  append: true,  
  redraw: true  
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

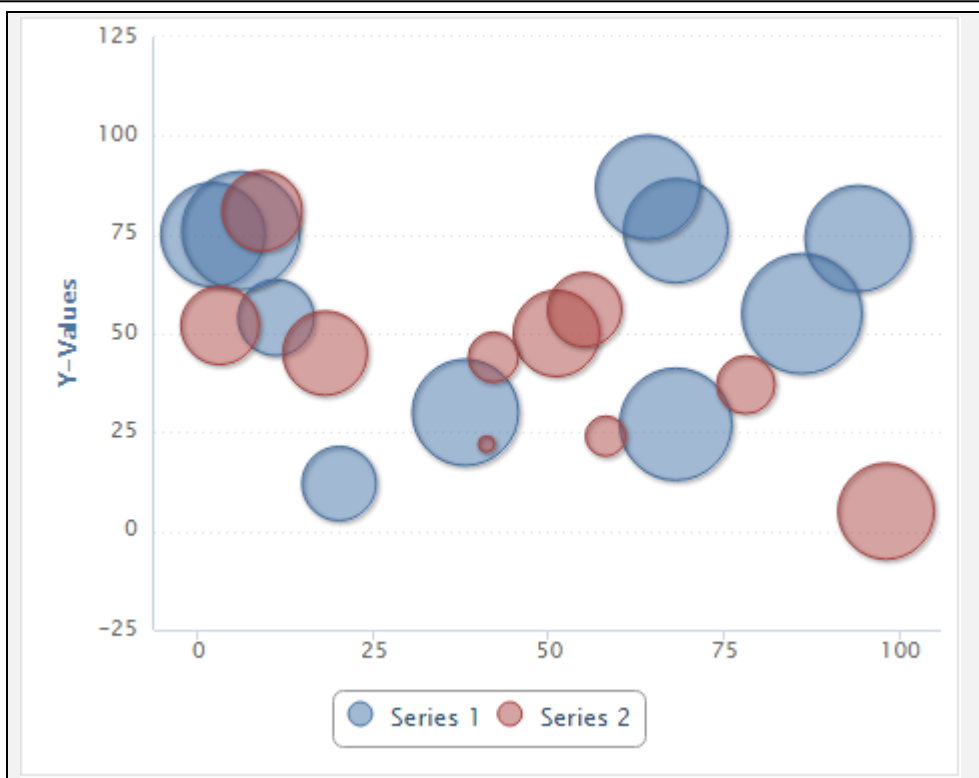
1.5 Bubble chart

➤ 기능

거품형 차트를 표현한다. (highcharts/highcharts-more.js 필요)

➤ Example

```
myChart.setOptions({
  chart:{
    type:'bubble'
  },
  series: [{
    data: [[94,74,60], [2,75,59], [20,12,4], [68,76,58],
[11,54,8],[6,76,91], [64,87,56], [86,55,93], [38,30,60],
[68,27,73]]
  },{
    data: [[9,81,13], [98,5,39], [51,50,23], [41,22,-36], [58,24,-
30],[78, 37, -16], [55, 56, 3], [18, 45, 20], [42, 44, -22], [3,
52, 9]]
  }]
},{
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

1.6 HeapMap chart

➤ 기능

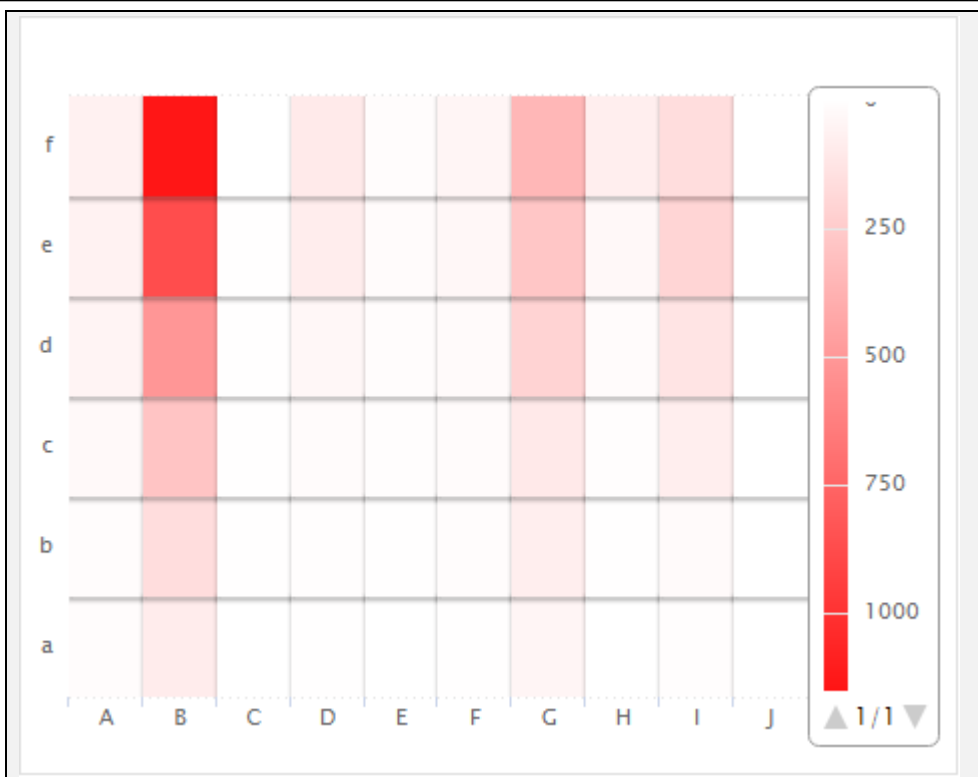
히트맵형 차트를 표현한다. (highcharts/modules/heatmap.js 필요)

➤ Example

```
myChart.setOptions({
  chart:{
    type:'heatmap',
    marginTop: 40,
    marginBottom: 40
  },
  xAxis: {
    categories: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I',
    'J'],
    lineWidth: 0
  },
  yAxis: {
    categories: ['a', 'b', 'c', 'd', 'e', 'f'],
    lineWidth: 0,
    title: ""
  },
  colorAxis: {
    min: 0,
    minColor: '#FFFFFF',
    maxColor: '#ff0000'
  },
  legend: {
    align: 'right',
    layout: 'vertical',
    margin: 0,
```



```
verticalAlign: 'top',
y: 25,
symbolHeight: 320
},
series: [{
  data: [[0,0,17], [0,1,17], [0,2,29], [0,3,54], [0,4,63],
[0,5,68], [1,0,92], [1,1,165],
[1,2,287], [1,3,517], [1,4,873], [1,5,1140], [2,0,1],
[2,1,4], [2,2,3], [2,3,4],
[2,4,4], [2,5,2], [3,0,4], [3,1,8], [3,2,19], [3,3,39],
[3,4,89], [3,5,101], [4,0,1],
[4,1,5], [4,2,10], [4,3,15], [4,4,22], [4,5,17], [5,0,6],
[5,1,13], [5,2,19], [5,3,26],
[5,4,40], [5,5,51], [6,0,50], [6,1,83], [6,2,108],
[6,3,218], [6,4,278], [6,5,352],
[7,0,2], [7,1,5], [7,2,10], [7,3,19], [7,4,35], [7,5,84],
[8,0,10], [8,1,26], [8,2,84],
[8,3,130], [8,4,206], [8,5,169], [9,0,0], [9,1,0], [9,2,0],
[9,3,0], [9,4,1], [9,5,1]]
}]
},{
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0

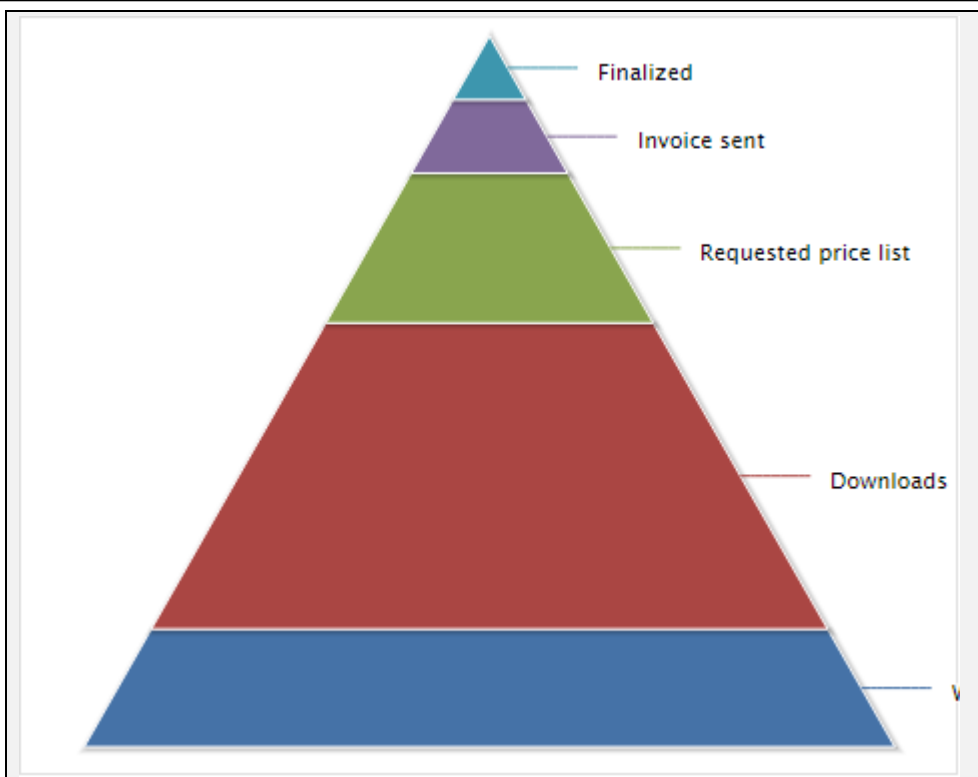
1.7 Prymid chart

➤ **기능**

피라미드형 차트를 표현한다. (highcharts/modules/funnel.js 필요)

➤ **Example**

```
myChart.setOptions({
  chart:{
    type:'pyramid'
  },
  series: [{
    name: 'Unique users',
    data: [
      ['Website visits', 1565],
      ['Downloads', 4064],
      ['Requested price list', 1987],
      ['Invoice sent', 976],
      ['Finalized', 846]
    ]
  }]
},{
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
----------------	--

1.8 Wordcloud chart

➤ 기능

워드클라우드 차트는 가중치(값)에 따라 단어의 크기, 위치, 색을 지정하고 단어를 시각적으로 표현하는 차트이다.

➤ 제한사항

단일 시리즈이며 그 한 개의 시리즈에 단어와 가중치 데이터로 구성되어있다. 차트 생성할때, `opt.plugin`값을 `d3`로 설정하고 사용해야한다. `opt.width`와 `opt.height`값은 `300px`이상으로만 설정해야한다. (※ 너무 작을경우 차트가 표현될 수 없음) 또, `%`로 설정되었을때 부모의 `너비`/`높이`가 `300px` 밑으로 내려가도 안된다.

➤ Example

```
//wordcloud 차트 생성
createIBChart(chart,"myChart", {
    width: "500px",
    height: "500px",
    plugin: "d3"
});

//wordcloud 차트 설정
myChart.setOptions({
    chart: {
        type: "wordcloud"
    }
}, {
    append: false
});

//wordcloud 데이터 로드
```

```
myChart.loadSearchData({
  "ibchart": {
    "data": [{
      "series": [{
        "seriesName": "WordCloud",
        "pointName": "IBLeaders",
        "value": 50
      }]
    }, {
      "series": [{
        "pointName": "javascript",
        "value": 40
      }]
    }, {
      "series": [{
        "pointName": "D3.js",
        "value": 15
      }]
    }, {
      "series": [{
        "pointName": "웹표준",
        "value": 30
      }]
    }, {
      "series": [{
        "pointName": "CrossBrowsing",
        "value": 30
      }]
    }, {
      "series": [{
        "pointName": "Front-end",
        "value": 30
      }]
    }
  ]
})
```

```
    ]]  
  }, {  
    "series": [{  
      "pointName": "Back-end",  
      "value": 30  
    }]  
  }, {  
    "series": [{  
      "pointName": "ECMAScript",  
      "value": 20  
    }]  
  }, {  
    "series": [{  
      "pointName": "Web",  
      "value": 15  
    }]  
  }, {  
    "series": [{  
      "pointName": "C++",  
      "value": 20  
    }]  
  }, {  
    "series": [{  
      "pointName": "C#",  
      "value": 25  
    }]  
  }, {  
    "series": [{  
      "pointName": "JAVA",  
      "value": 38  
    }]  
  }, {
```

```
        "series": [{
            "pointName": "HTML",
            "value": 40
        }]
    }, {
        "series": [{
            "pointName": "AngularJS",
            "value": 30
        }]
    }, {
        "series": [{
            "pointName": "jQuery",
            "value": 35
        }]
    }, {
        "series": [{
            "pointName": "CSS",
            "value": 32
        }]
    }, {
        "series": [{
            "pointName": "솔루션",
            "value": 45
        }]
    }, {
        "series": [{
            "pointName": "Solution",
            "value": 45
        }]
    }, {
        "series": [{
            "pointName": "Language",
```



```
        "value": 15
      }
    ], {
      "series": [{
        "pointName": "Support",
        "value": 15
      }
    ]
  }, {
    "series": [{
      "pointName": "IBSheet 7",
      "value": 25
    }
  ]
}, {
  "series": [{
    "pointName": "IBSheet (H) 7",
    "value": 24
  }
]
}, {
  "series": [{
    "pointName": "IBChart (H) 7",
    "value": 20
  }
]
}, {
  "series": [{
    "pointName": "IBMultiCombo (H) 7",
    "value": 17
  }
]
}, {
  "series": [{
    "pointName": "IBUpload (H) 7",
    "value": 17
  }
]
```

```

    }, {
      "series": [{
        "pointName": "IBMDITab(H) 7",
        "value": 17
      }]
    }, {
      "series": [{
        "pointName": "IBMaskEdit(H) 7",
        "value": 17
      }]
    }
  ], {
    append: true
  });

```



A word cloud visualization featuring various web and programming technologies. The words are arranged in a circular pattern, with some appearing more prominently than others. The colors of the words vary, including shades of blue, green, orange, and purple. The words include:

- IBSheet(H) 7
- CrossBrowsing
- Back-end
- HTML
- IBMDITab(H) 7
- D3.js
- 웹표준 (Web Standard)
- 솔루션 (Solution)
- JAVA
- IBMaskEdit(H) 7
- Language Support
- IBLeaders
- CSS
- Solution
- C#
- IBSheet 7
- Web
- javascript
- C++
- IBChart(H) 7
- AngularJS
- jQuery
- ECMAScript
- Front-end
- IBUpload(H) 7
- IBMultiCombo(H) 7

➤ 제공 버전

7.3.1.0	wordcloud 차트에만 적용가능
---------	---------------------

2. 차트기능

2.1 stack 의 그룹기능

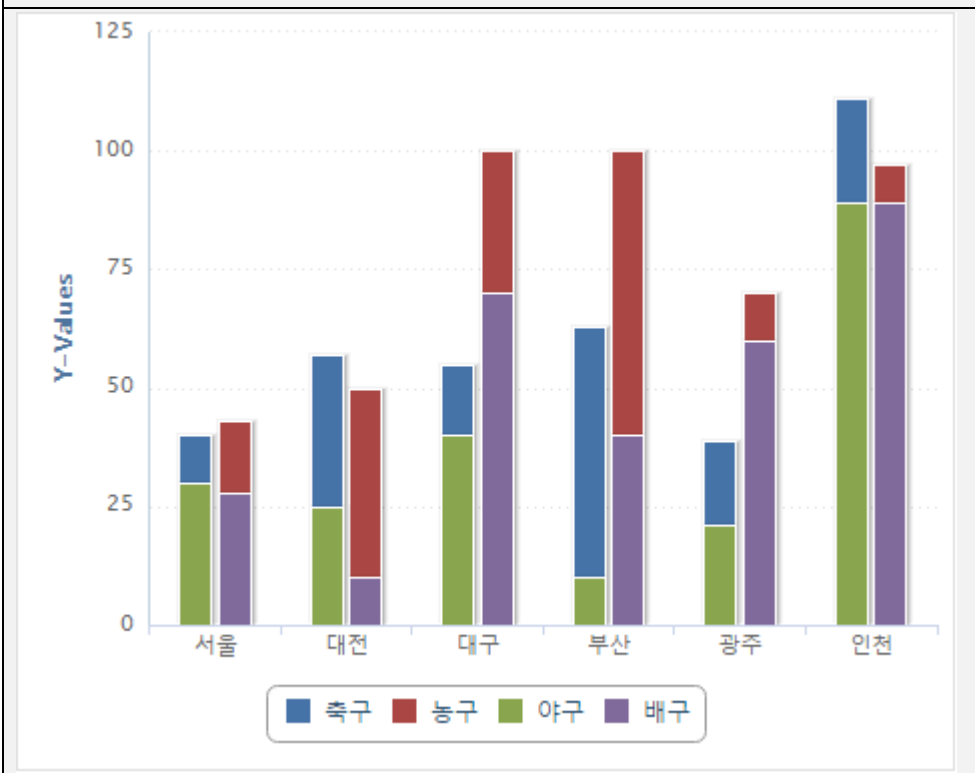
➤ 기능

Stacking 기능을 이용하면서 그룹을 지어 그룹별로 Stacking 기능을 이용 할 수 있다.

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  xAxis: {
    categories:["서울","대전","대구","부산","광주","인천"]
  },
  plotOptions:{
    column: {
      stacking: "normal"
    }
  },
  series: [{
    name : "축구",
    stack:"summer", //steking 기능에서 stack 이름을 주어 그룹을 형성
    data : [10,32,15,53,18,22]
  },{
    name : "농구",
    stack:"winter",
    data : [15,40,30,60,10,8]
  },{
    name : "야구",
    stack:"summer",
    data : [30,25,40,10,21,89]
```

```
}, {
  name : "배구",
  stack: "winter",
  data : [28, 10, 70, 40, 60, 89]
}]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
----------------	--

2.2 멀티 차트 표현 기능

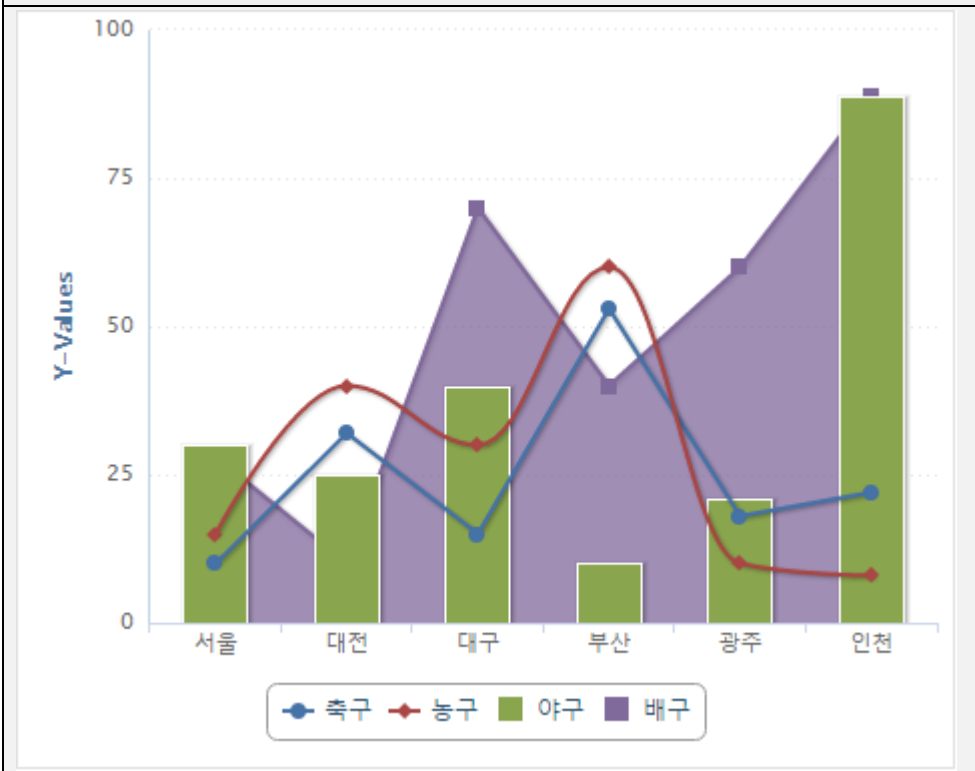
➤ 기능

여러가지의 형태를 한 차트에 표현 할 수 있다.

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  xAxis: {
    categories:["서울", "대전", "대구", "부산", "광주", "인천"]
  },
  series: [{
    type: "line",
    zIndex: 3, // z-layer index
    name : "축구",
    data : [10,32,15,53,18,22]
  },{
    type: "spline",
    zIndex: 4,
    name : "농구",
    data : [15,40,30,60,10,8]
  },{
    type: "column",
    zIndex: 2,
    name : "야구",
    data : [30,25,40,10,21,89]
  },{
    type: "area",
    zIndex: 1,
```

```
name : "배구",
data : [28,10,70,40,60,89]
}]
},{
append: true,
redraw: true
});
```



➤ 제공 버전

7.3.0.0

2.3 Secondary Y 축 사용

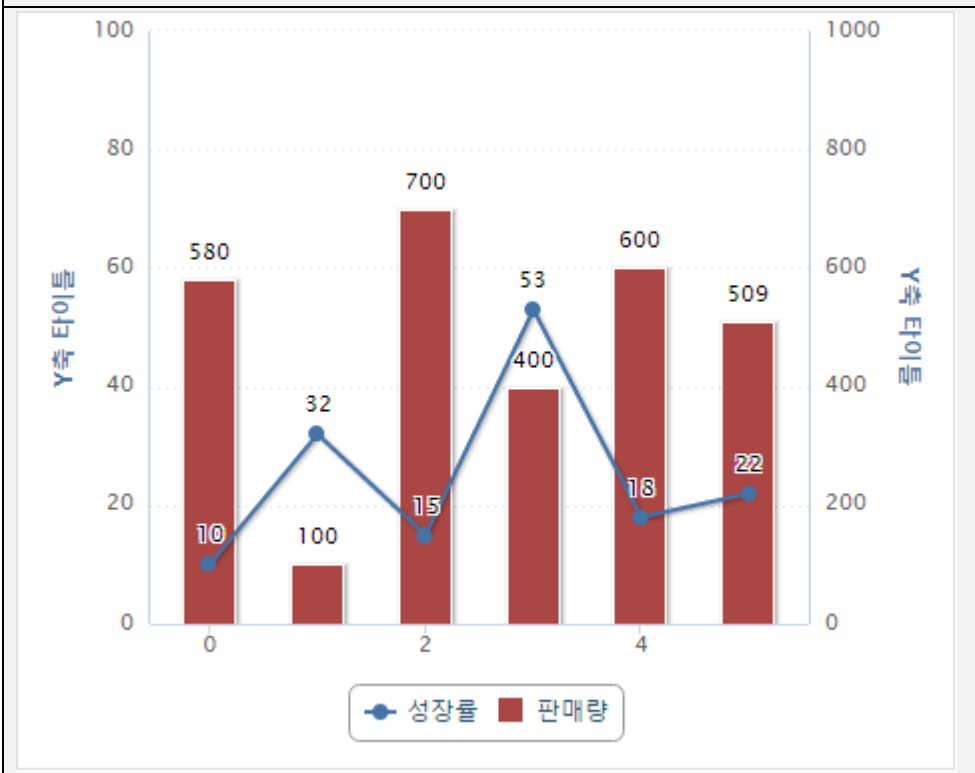
➤ 기능

두개의 Y 축을 사용한다.

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  yAxis: [{
    title:{
      text: "Y축 타이틀"
    }
  },{
    title:{
      text: "Y축 타이틀"
    },
    opposite: true
  }],
  series: [{
    name : "성장률",
    type:"line",
    zIndex:3,
    data : [10,32,15,53,18,22],
    dataLabels:{
      enabled:true
    }
  },{
    name : "판매량",
    zIndex:1,
```

```
data : [580,100,700,400,600,509],
dataLabels:{
  enabled:true
},
yAxis:1
}]
},{
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0

2.4 plotLine 기능

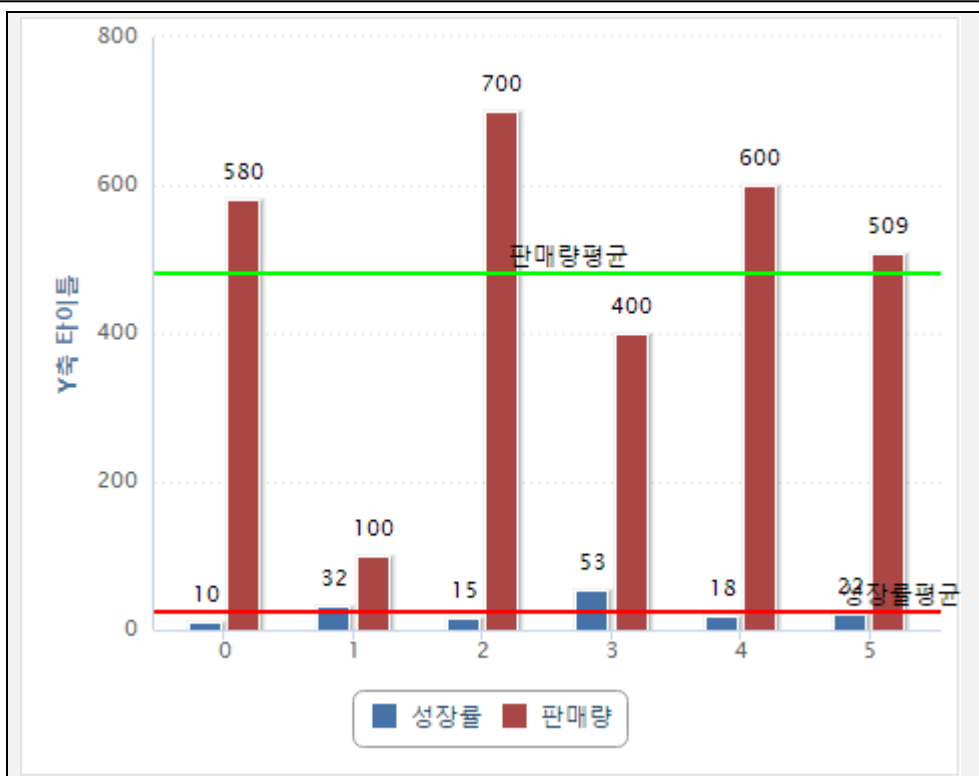
➤ 기능

차트 중간에 선을 표현 할수 있다.

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  yAxis: {
    title:{
      text: "Y축 타이틀"
    },
    plotLines : [{
      color: '#FF0000',
      value:25,
      width:2,
      label:{
        text:"성장률평균",
        align:"right"
      },
    },
    {
      color: '#00FF00',
      value:481.5,
      width:2,
      label:{
        text:"판매량평균",
        align:"center"
      },
    },
  ],
}
```

```
        zIndex:10
      }]
    },
    series: [{
      name : "성장률",
      zIndex:3,
      data : [10,32,15,53,18,22],
      dataLabels:{
        enabled:true
      }
    },{
      name : "판매량",
      zIndex:1,
      data : [580,100,700,400,600,509],
      dataLabels:{
        enabled:true
      }
    }]
  },{
    append: true,
    redraw: true
  });
```



➤ 제공 버전

7.3.0.0

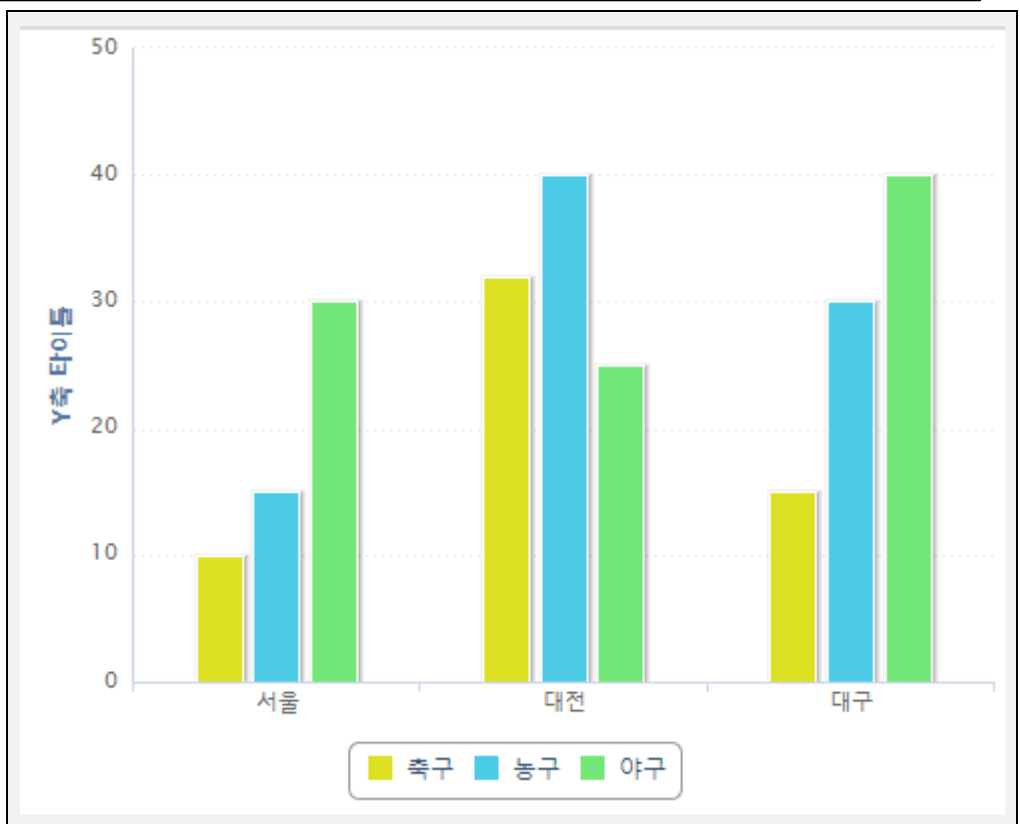
2.5 colors 기능

➤ 기능

차트 별로 color 를 지정 할 수 있다.

➤ Example

```
myChart.setOptions({
  colors: ['#DDDF00', '#24CBE5', '#64E572', '#FF9655', '#FFF263'],
  chart:{
    type:'column'
  },
  yAxis: {
    title:{
      text: "Y축 타이틀"
    },
  },
  xAxis: {
    categories: ["서울", "대전", "대구"]
  },
  series: [{
    name : "축구",
    data : [10,32,15],
  }, {
    name : "농구",
    data : [15,40,30],
  }, {
    name : "야구",
    data : [30,25,40],
  }]
}, {
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

2.6 차트 크기 동적 변경기능

➤ 기능

차트 크기를 동적으로 변경 할 수 있다.

➤ Example

```
myChart.setOptions({
  colors: ['#DDDF00', '#24CBE5', '#64E572', '#FF9655', '#FFF263'],
  chart:{
    type:'column'
  },
  yAxis: {
    title:{
      text: "Y축 타이틀"
    },
  },
  xAxis: {
    categories: ["서울", "대전", "대구"]
  },
  series: [{
    name : "축구",
    data : [10,32,15],
  }, {
    name : "농구",
    data : [15,40,30],
  }, {
    name : "야구",
    data : [30,25,40],
  }]
}, {
  append: true,
  redraw: true
});
```



```
myChart.setSize(300,300);
```



➤ 제공 버전

7.3.0.0

2.7 시리즈 관련기능(pointPadding, pointWidth)

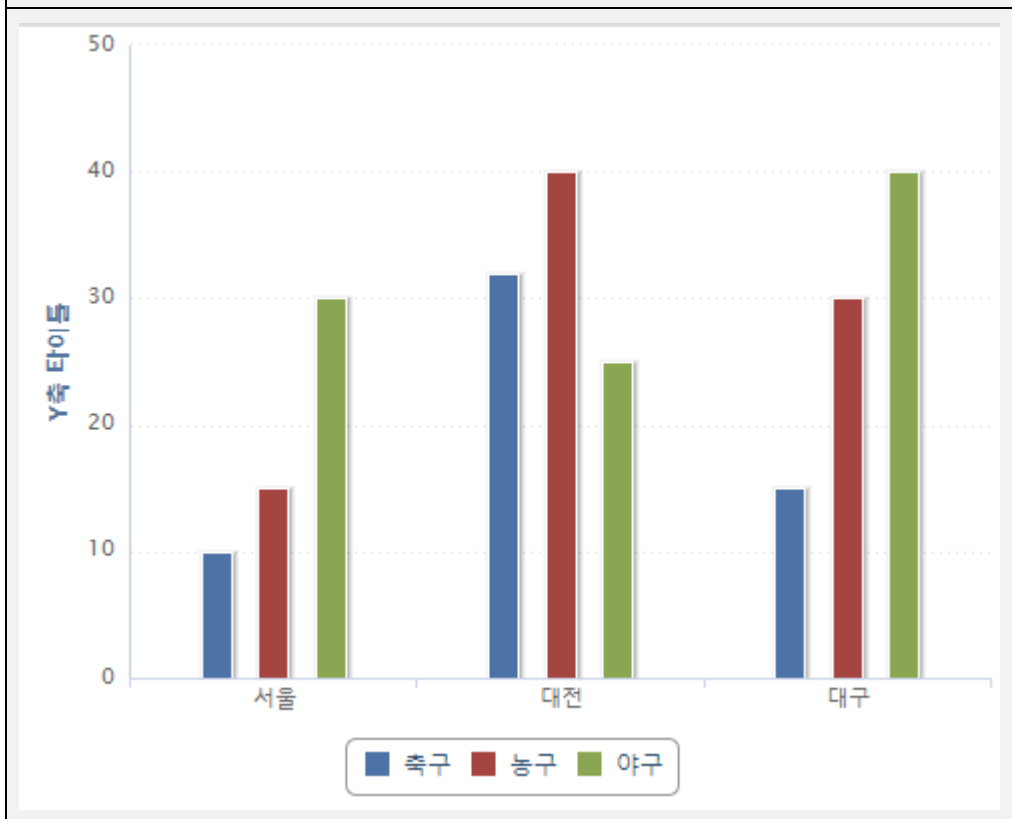
➤ 기능

시리즈 관련기능(pointPadding, pointWidth) 구현

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  yAxis: {
    title:{
      text: "Y축 타이틀"
    },
  },
  xAxis: {
    categories: ["서울", "대전", "대구"]
  },
  plotOptions: {
    series: {
      pointWidth: 15, //포인트의 너비를 설정한다.
      pointPadding: 0.3 //포인트 간 padding 적용 값을 설정한다.
    }
  }
  series: [{
    name : "축구",
    data : [10,32,15],
  }, {
    name : "농구",
    data : [15,40,30],
  }, {
    name : "야구",
    data : [30,25,40],
```

```
    }]  
  }, {  
    append: true,  
    redraw: true  
  });
```



➤ **제공 버전**

7.3.0.0	
----------------	--

2.8 시리즈 관련기능(pointColor)

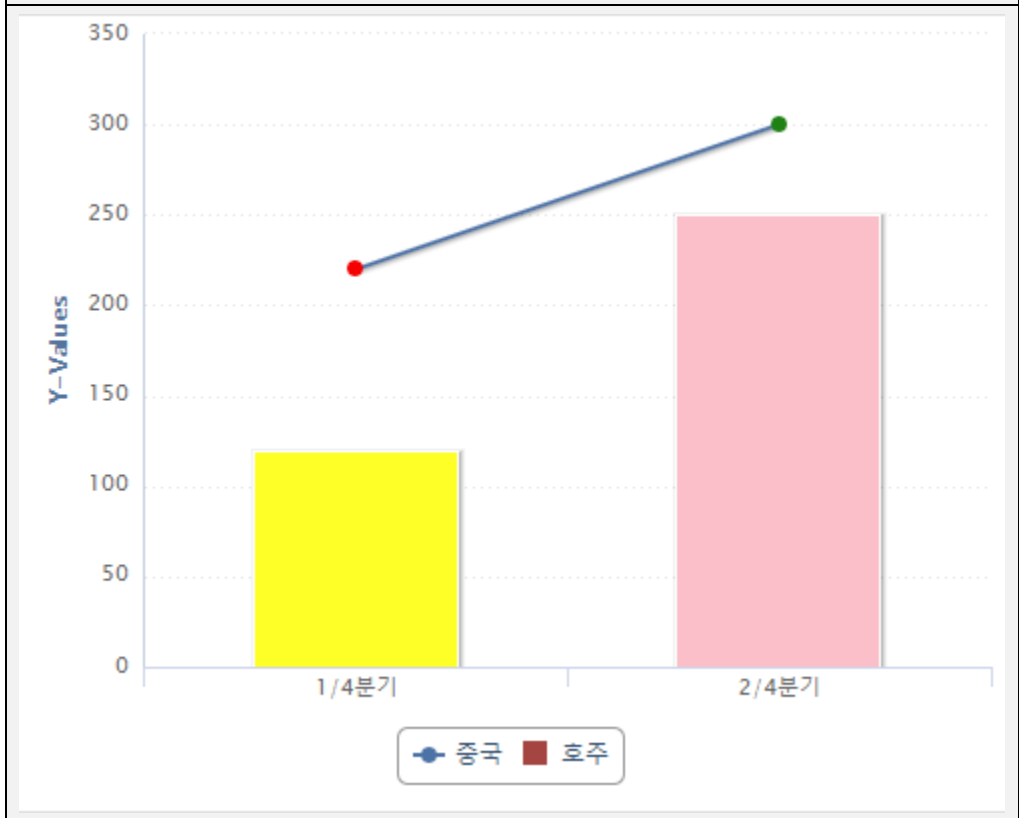
➤ 기능

시리즈 관련기능(pointColor) 구현

➤ Example

```
var data = {  
  ibchart: {  
    data: [{  
      axisLabel: "1/4분기",  
      series: [{  
        seriesType: "line",  
        seriesName: "중국",  
        pointName: "하나",  
        pointColor: "red",  
        value: 220  
      }, {  
        seriesType: "column",  
        seriesName: "호주",  
        pointName: "둘",  
        pointColor: "yellow",  
        value: 120  
      }]  
    }, {  
      axisLabel: "2/4분기",  
      series: [{  
        pointColor: "green",  
        value: 300  
      }, {  
        pointColor: "pink",  
        value: 250  
      }]  
    }  
  ]  
}
```

```
    }  
  }  
};  
myChart.loadSearchData(data, {  
  append: true  
});
```



➤ 제공 버전

7.3.0.0	
---------	--

2.9 시리즈 관련기능(막대 그래데이션)

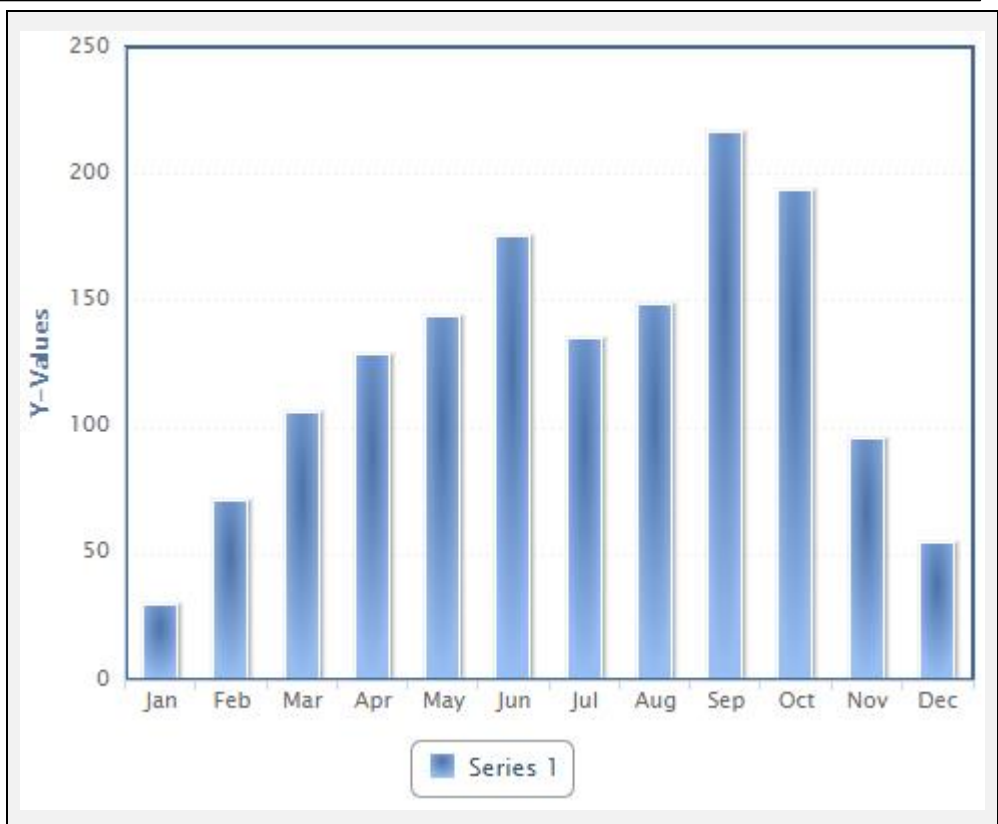
➤ 기능

시리즈 관련기능(막대 그래데이션) 구현.

➤ Example

```
var LightenColor = function (color, percent) {  
    var num = parseInt(color.replace("#", ""), 16),  
        amt = Math.round(2.55 * percent),  
        R = (num >> 16) + amt,  
        B = (num >> 8 & 0x00FF) + amt,  
        G = (num & 0x0000FF) + amt;  
  
    return "#" + (0x1000000 + (R < 255 ? R < 1 ? 0 : R : 255) *  
        0x10000 + (B < 255 ? B < 1 ? 0 : B : 255) * 0x100 +  
        (G < 255 ? G < 1 ? 0 : G : 255)).toString(16).slice(1);  
};  
  
var defaultColors = myChart.getOptions().colors; //기본 컬러 값  
var gradientColors = [];  
  
for(var i = 0; i < defaultColors.length; i++){  
    gradientColors[i] = {  
        radialGradient: {  
            cx: 0.5,  
            cy: 0.3,  
            r: 0.7  
        },  
        // 그래데이션 단계  
        stop: [  
            [0, defaultColors[i]],  
            [1, LightenColor(defaultColors[i], 30)] //brighton  
        ]  
    }  
}
```

```
}  
myChart.setOptions({  
    colors: gradientColors,  
    chart: {  
        type: 'column',  
        plotBorderColor: '#346691',  
        plotBorderWidth: 2  
    },  
    xAxis: {  
        categories: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul',  
'Aug', 'Sep', 'Oct', 'Nov', 'Dec']  
    },  
    series: [{  
        data: [29.9, 71.5, 106.4, 129.2, 144.0, 176.0, 135.6, 148.5,  
216.4, 194.1, 95.6, 54.4]  
    }]  
}, {  
    append: true,  
    redraw: true  
}  
);
```



➤ **제공 버전**

7.3.0.0	
----------------	--

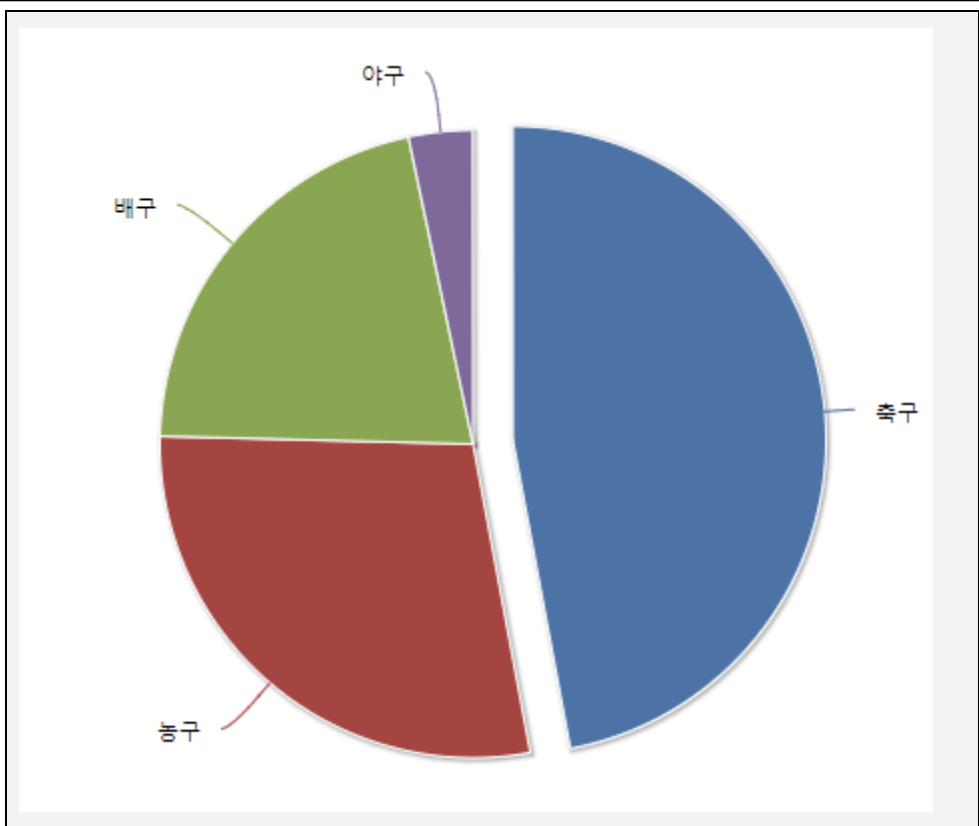
2.10 시리즈 관련 기능(slicedOffset)

➤ 기능

시리즈 관련기능(slicedOffset) 구현, Pie 이격 거리

➤ Example

```
myChart.setOptions({
  chart:{
    type:'pie'
  },
  plotOptions: {
    series: { //series 대신 pie를 넣어도 상관없음.
      slicedOffset: 20
    }
  },
  series: [{
    data : [{
      name: "축구",
      y: 44.2,
      selected: true,
      sliced: true
    },
    ["농구",26.6],
    ["배구",20],
    ["야구",3.1],
  ]
  }]
},{
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

2.11 도넛 안에 타이틀 넣는 기능

➤ 기능

시리즈 관련기능(slicedOffset) 구현, Pie 이격 거리

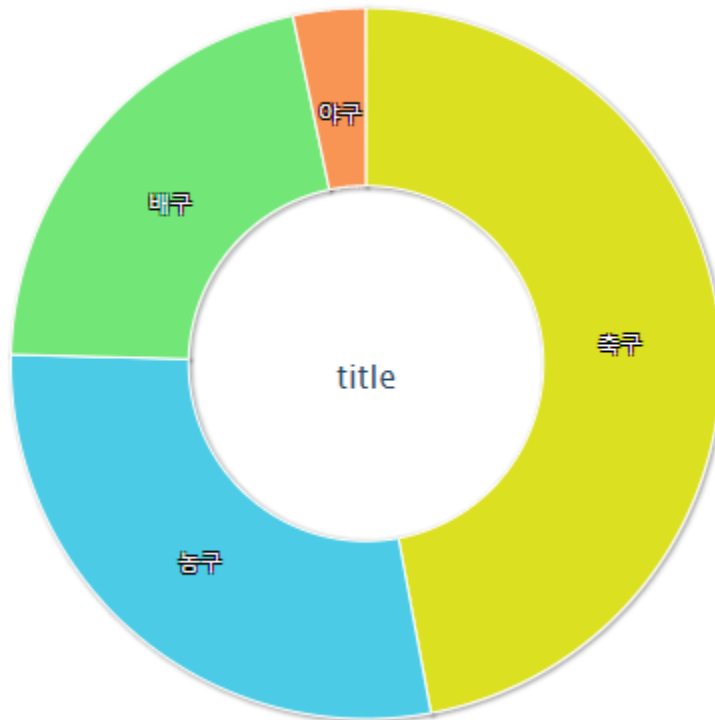
➤ Example

```
myChart.setOptions({
  colors: ['#DDDF00', '#24CBE5', '#64E572', '#FF9655', '#FFF263']
  chart:{
    type:'pie'
  },
  title:{
    text: "title",
    align: "center",
    verticalAlign: "middle"
  },
  plotOptions: {
    pie:{
      dataLabels: {
        enabled: true,
        distance: -50,
      }
    }
  },
  series: [{
    innerSize: "50%",
    data : [
      ["축구", 44]
      ["농구", 26],
      ["배구", 20],
      ["야구", 3],
    ]
  }
}]
```

```

    ]]
  }, {
    append: true,
    redraw: true
  });

```



➤ 제공 버전

7.3.0.0

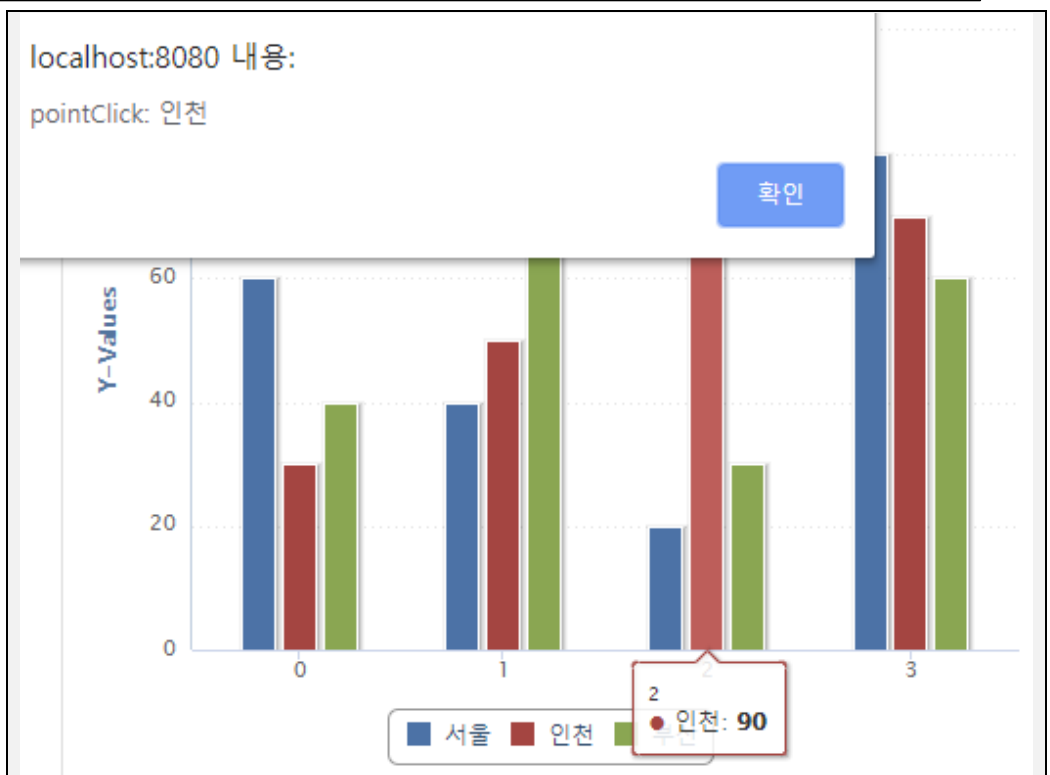
2.12 포인트 클릭 이벤트

➤ 기능

포인트 클릭 이벤트 기능입니다.

➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  series: [{
    name: "서울",
    data: [60, 40, 20, 80]
  }, {
    name: "인천",
    data: [30, 50, 90, 70]
  }, {
    name: "부산",
    data: [40, 70, 30, 60]
  }],
  events: {
    pointClick: function() {
      alert("pointClick: "+ this.series.name)
    }
  },
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

2.13 legend 관련 이벤트

➤ 기능

legend 관련 이벤트 기능입니다. (seriesLegendItemClick())

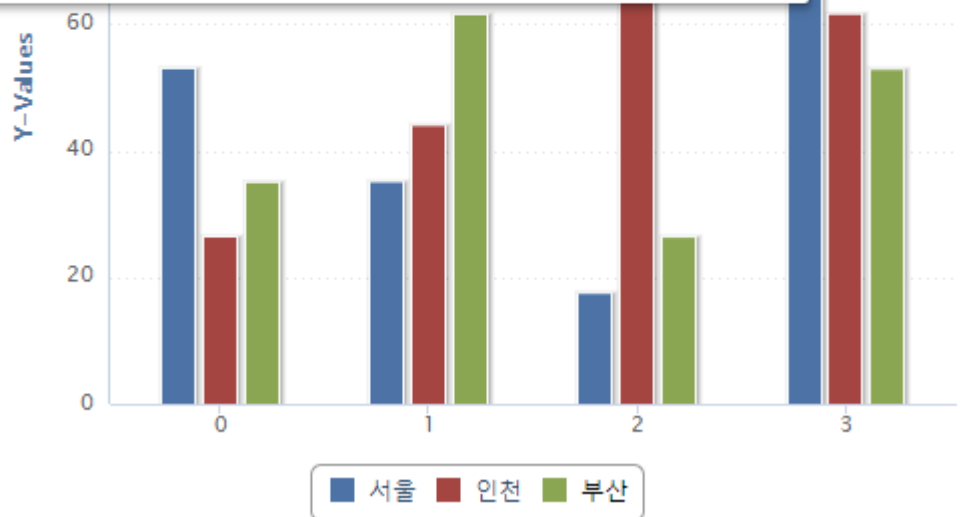
➤ Example

```
myChart.setOptions({
  chart:{
    type:'column'
  },
  series: [{
    name: "서울",
    data: [60, 40, 20, 80]
  }, {
    name: "인천",
    data: [30, 50, 90, 70]
  }, {
    name: "부산",
    data: [40, 70, 30, 60]
  }],
  events: {
    seriesLegendItemClick: function() {
      alert("seriesLegendItemClick함수발생");
    }
  },
  append: true,
  redraw: true
});
```


localhost:8080 내용:

seriesLegendItemClick함수발생

확인



➤ 제공 버전

7.3.0.0

2.14 stacking 에 대한 sum 값 표시기능

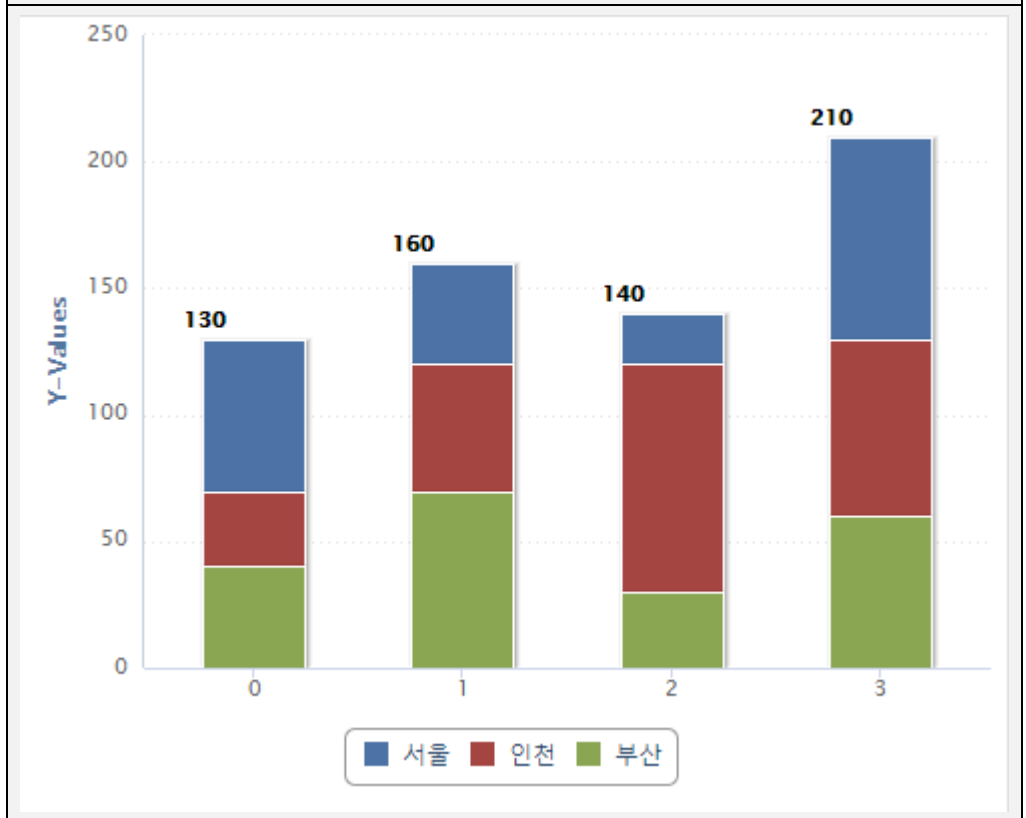
➤ 기능

누적형 레이블 사용여부(합계값 표시)를 표시한다.

➤ Example

```
myChart.setOptions({
  chart: {
    type: "column"
  },
  yAxis: {
    stackLabels: {
      enabled: true,
      align: "left"
    }
  },
  plotOptions: {
    column: {
      stacking: "normal"
    }
  },
  series: [{
    name: "서울",
    data : [60,40,20,80]
  }, {
    name: "인천",
    data : [30,50,90,70]
  }, {
    name: "부산",
    data : [40,70,30,60]
  }]
}, {
```

```
append: true,  
redraw: true  
});
```



➤ 제공 버전

7.3.0.0

2.15 label, title 에 html 사용여부기능

➤ 기능

Html 사용여부기능, useHTML: true 일 경우 사용가능하다.

➤ Example

```
myChart.setOptions({
  title: {
    text: "html적용x"
  },
  subtitle: {
    text: "<span style='font-size: 32px; font-weight: bold; color:red; background:yellow;'>html 적용</span>",
    useHTML: true //true 일 경우 html 코드 사용 가능.
  }
}, {
  append: true,
  redraw: true
});
```

html적용x

html 적용

➤ 제공 버전

7.3.0.0

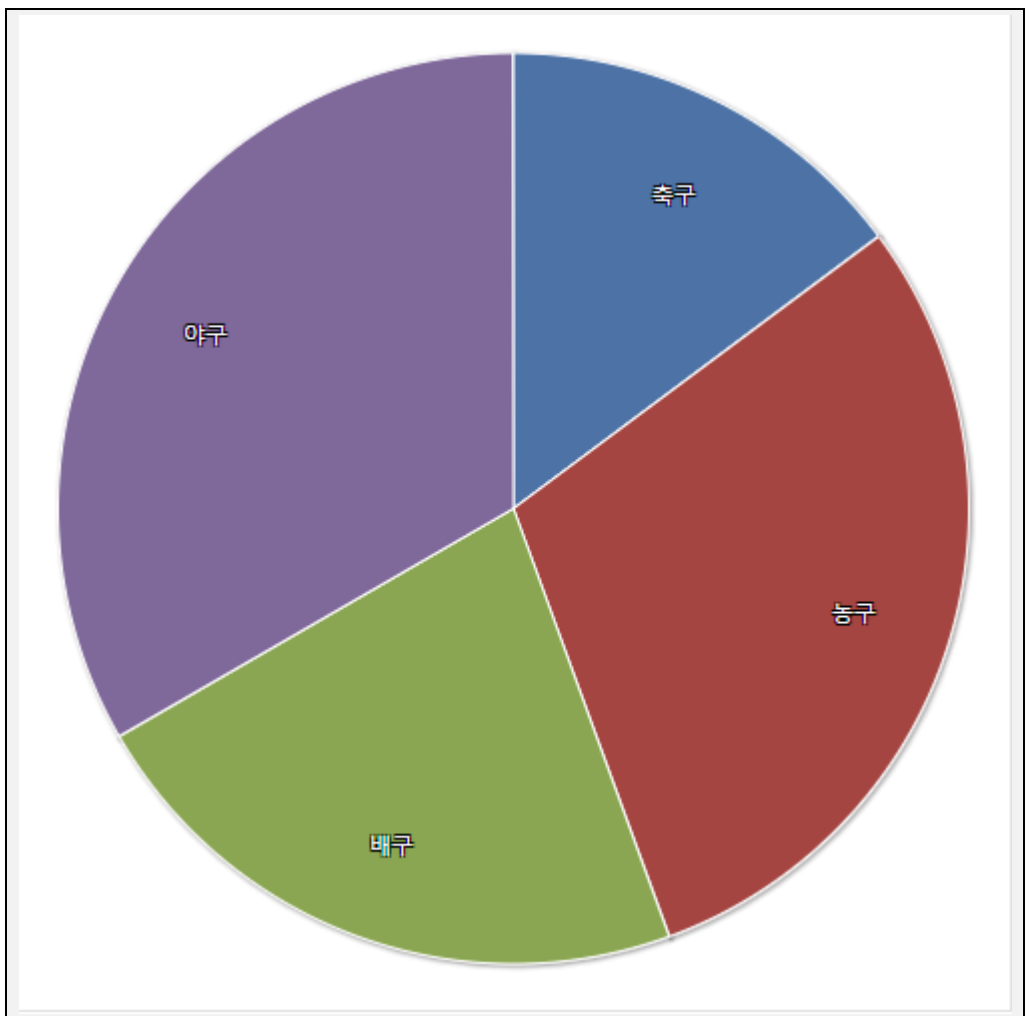
2.16 파이 차트 크기 고정

➤ 기능

파이 차트 크기 고정기능입니다. (plotOptions.pie.size)

➤ Example

```
myChart.setOptions({
  chart:{
    type:'pie'
  },
  plotOptions: {
    pie: {
      dataLabels: {
        enabled: true,
        distance: -50
      },
      size: "100%"
    }
  },
  series: [{
    data : [
      ["축구", 20],
      ["농구", 40],
      ["배구", 30],
      ["야구", 45]
    ]
  }]
},{
  append: true,
  redraw: true
});
```



➤ **제공 버전**

7.3.0.0	
---------	--

2.17 1000 개이상 데이터 표시 방법

➤ 기능

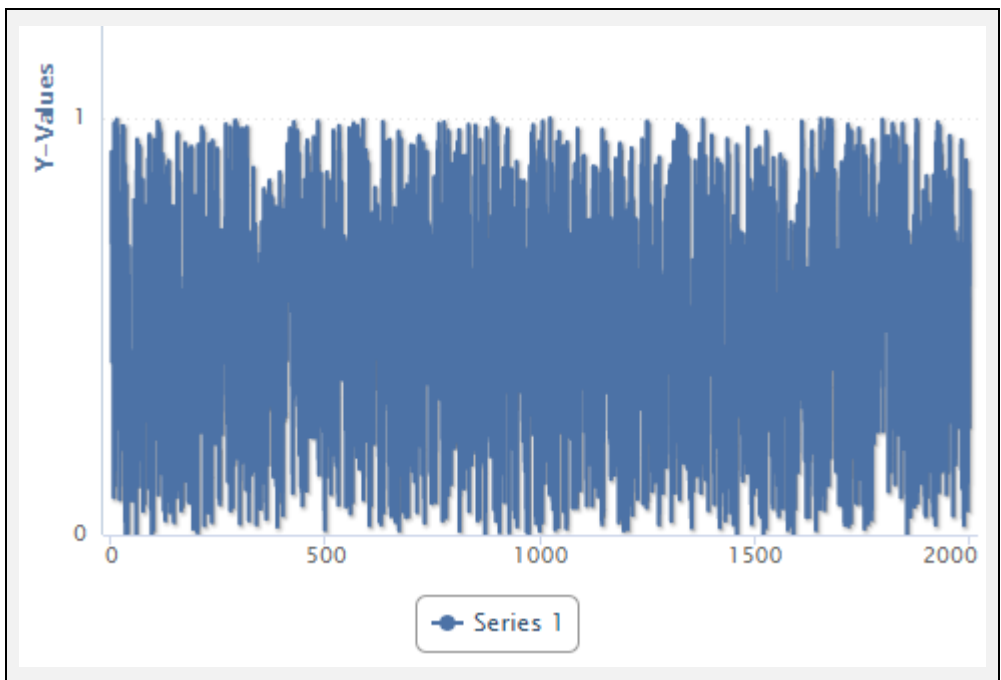
1000 개 이상 데이터 표시 방법입니다. 1000 개부터 렌더링하기 위해서는 turboThreshold 를 설정 해주어야 합니다.

➤ Info

name	type	required	description
turboThreshold	number	N(1000)	0 일 경우 사용하지 않음. default: 1000

➤ Example

```
myChart.setOptions({
  chart:{
    type:'line'
  },
  plotOptions: {
    series: {
      turboThreshold: 2001 //default: 1000
    },
  },
  series: [{
    data : [...]
  }]
},{
  append: true,
  redraw: true
});
```



➤ 제공 버전

7.3.0.0	
---------	--

2.18 작은 비율을 갖는 시리즈를 합쳐서 표시하는 방법

➤ 기능

작은 비율을 갖는 시리즈를 합쳐서 표시하는 방법
searchEnd 이벤트를 이용한 방법.

➤ Example

```
var data = {  
  ibchart: {  
    data: [{  
      axisLabel: "축구",  
      series: [{  
        seriesType: "pie",  
        value: 100  
      }]  
    }, {  
      axisLabel: "야구",  
      series: [{  
        value: 50  
      }]  
    }, {  
      axisLabel: "농구",  
      series: [{  
        value: 10  
      }]  
    }, {  
      axisLabel: "배구",  
      series: [{  
        value: 10  
      }]  
    }, {  
      series: [{
```

```
        value: 5
    }]
}, {
    series: [{
        value: 6
    }]
}, {
    series: [{
        value: 8
    }]
}]
}
};

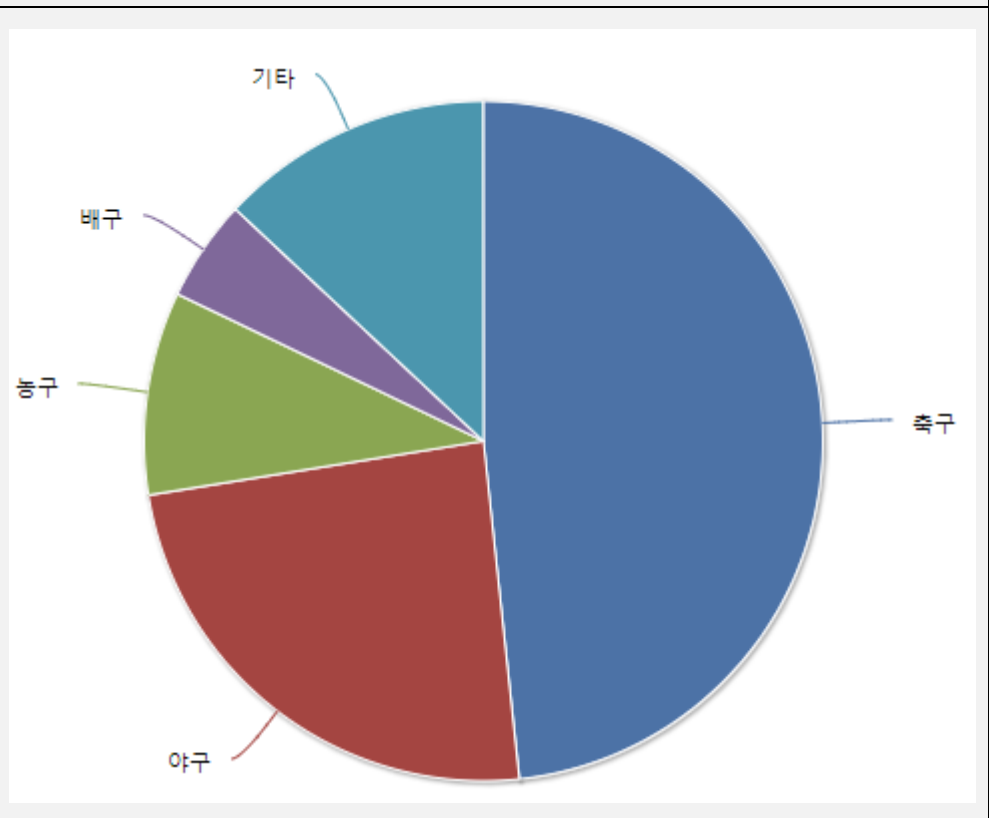
myChart.setEvents({
    searchEnd: function(event){
        var minsum = 0; //작은 비율을 합칠 변수
        for(var i=0; i < data.ibchart.data.length; i++){
            if(data.ibchart.data[i].series[0].value < 10){// 값이 10 미만
                minsum += data.ibchart.data[i].series[0].value;
            }
        }
        myChart.setOptions({
            series: [{
                data: [
                    ["축구", 100],
                    ["야구", 50],
                    ["농구", 20],
                    ["배구", 10],
                    ["기타", minsum]
                ]
            }]
        })
    }
});
```

```

    }, {
      append: true,
      redraw: true
    });
  }
});

myChart.loadSearchData(data, {
  append: true
});

```



➤ **제공 버전**

7.3.0.0	
---------	--

3. IBSheet 데이터 연동

IBSheet 에 조회된 데이터를 쉽게 차트로 표현할 수 있는 기능으로 sheet2chart 메소드를 이용하여 설정 한다
 기본적으로 시트에서의 컬럼은 시리즈, 행은 데이터로 표현된다.

➤ Syntax

Syntax	sheet2chart(sheet, chart, params)
--------	-----------------------------------

➤ Info

returns	없음		
name	type	required	description
sheet	object	Y	시트 객체
chart	object	Y	차트 객체
params	object		조회 옵션
params.datadirection	string	Y	데이터를 만들기 위해 시트의 내용을 읽어들이는 방향
params.datarowcol	string	Y	데이터를 읽어들이는 열 혹은 행, 읽어들이는 방향이 vertical인 경우에는 열을, horizontal인 경우에는 행을 구분자 " " 연결하여 넣는다.
params.legendseriesname	string	Y	범례에 나올 시리즈별 이름(datarowcol의 개수와 동일해야 함.)
params.legendtitle	string	Y(범례)	범례 타이틀
params.startdatarowcol	number	Y(시작행 시작열)	데이터가 될 시작 행 혹은 열
params.enddatarowcol	number	Y(마지막	데이터가 될 마지막 행

		행 마지막 열)	혹은 열
params.labelrowcol	number	Y	x축 레이블이 될 행이나 열 하나만 선택 한다.
params.startlabelrowcol	number	Y	x축 레이블이 될 첫번째 행이나 열
params.endlabelrowcol	number	Y	x축 레이블이 될 마지막 행이나 열
params.labeldirection	string	Y	x축 레이블을 읽어 들이 는 방향
params.seriestype	string array <string>	Y	시리즈 유형

➤ **Enum**

■ **params.datadirection / params.labeldirection**

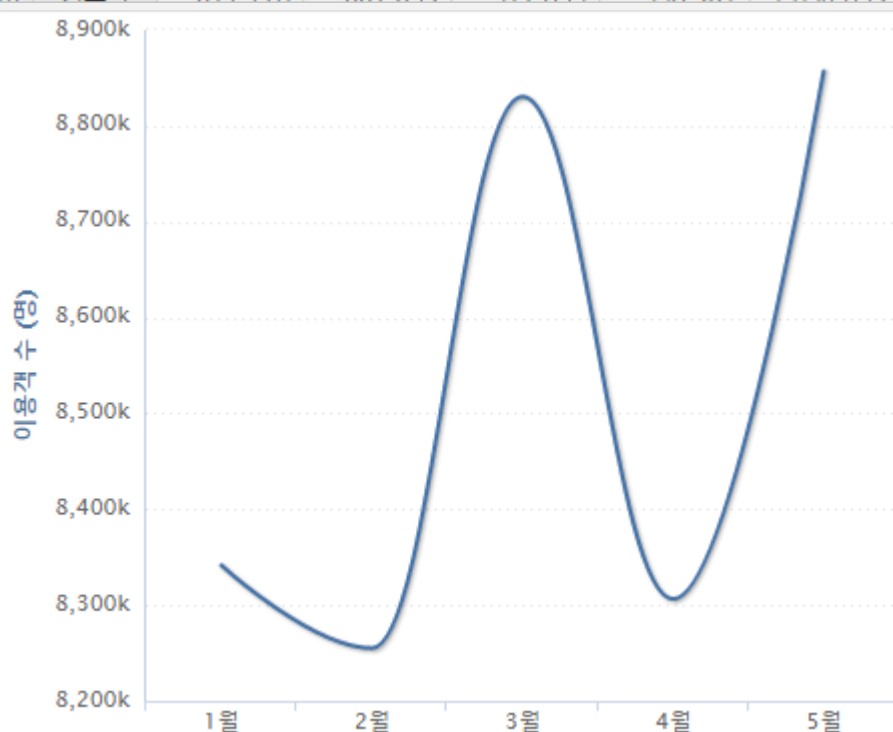
value	description
vertical	세로로 데이터 조회
horizontal	가로로 데이터 조회

➤ **Example**

```
myChart.setOptions({
    xAxis:{
        categories:true    //X축 라벨
    },
    plotOptions:{
        series:{
            dataLabels:{
                enabled:true    //데이터값 표시
            }
        }
    }
});

// IBSheet 연동
myChart.sheet2chart(mySheet, myChart, {
    datadirection: "horizontal",
    datarowcol: "1",
    legendseriesname: "" + mySheet18.GetRowData(1).sLineNm, // 시리즈
    //의 이름을 sheet의 첫번째행에서 sLineNm의 값을 가져온다.
    legendtitle: "범례",
    startdatarowcol: 2,
    enddatarowcol: 5,
    labelrowcol: 0,
    startlabelrowcol: 2,
    endlabelrowcol: 5,
    labeldirection: "horizontal",
    seriestype: ["spline"]
});
```

NO	노선명	1월	2월	3월	4월	5월
1	1호선	8,341,518	8,255,019	8,830,598	8,306,279	8,856,979
2	2호선	41,414,219	42,585,260	46,188,844	43,828,003	46,251,012
3	3호선	14,661,057	15,119,468	16,547,899	15,623,045	16,595,063
4	4호선	16,402,892	16,762,228	18,931,723	17,958,485	19,240,358
5	5호선	15,534,075	16,002,917	17,586,360	17,084,626	17,854,814
6	6호선	8,409,574	8,699,279	10,071,527	9,644,924	10,368,294
7	7호선	16,873,347	17,260,174	19,304,031	18,509,664	19,527,496
8	8호선	4,294,660	4,394,818	4,917,996	4,729,974	4,945,049
9	9호선	5,810,404	5,940,340	6,471,277	6,536,712	6,600,419
10	경부선	12,970,184	12,961,037	15,569,139	14,917,784	15,776,927
11	경원선	5,469,726	5,499,194	6,600,102	6,484,753	7,008,388
12	경의선	1,355,189	1,379,565	1,650,290	1,615,118	1,758,368
13	경인선	10,527,213	10,704,784	12,483,585	11,945,683	12,683,979
14	경춘선	877,726	882,853	972,695	956,381	1,094,053



➤ 제공 버전

7.3.0.0	deprecated
---------	------------