



IBChart 개발자가이드



IBChart 개발자 가이드

초판발행/ 2012년 3월 20일

지은이/ 윤대영

펴낸이/ 정광천

펴낸곳/



(주) 아이비 리더스
Inno Biz Leaders Co., Ltd.

137-878 서울시 구로구 구로3동 한화비즈메트로 1차 1501호

전화/ (02)2621-2080

팩스/ (02)2621-2088

편집/ 윤대영 인쇄/ C&G

※ 기술상담 및 내용문의는(주)아이비리더스로 해주십시오.

<http://www.ibleaders.co.kr>

목 차

Chapter 1. IBChart 개요	10
1. IBChart 개요	12
1.1 IBChart 정의	12
1.2 IBChart 특징	12
1.3 IBChart의 적용 대상 업무	12
1.4 IBChart의 주요 기능	12
Chapter 2. IBChart의 구조	16
2. IBChart의 구조	18
2.1 IBChart의 형태	18
2.2 IBChart의 각 부분 명칭	23
Chapter 3. IBChart의 개발환경 설정	25
3. IBChart의 개발 환경 설정	27
3.1 IBChart의 Server 설치	27
3.1.1 배포파일 및 내용	28
Chapter 4. 개발 절차	32
4. 개발 절차	34
4.1 웹 페이지에 JS 파일 링크 걸기	34
4.2 IBChart 객체를 화면에 추가	34
4.3 초기 항목 설정	35
4.3.1 차트 기본 속성 설정	36
4.3.2 타이틀 및 서브타이틀 속성	39
4.3.3 축(Axis) 속성	40
4.3.4 시리즈(Series) 속성 plot 영역에 표시될 시리즈 객체를 생성하고 데이터를 입력한다.	43

4.3.5 툴팁(ToolTip) 설정 마우스커서가 시리즈 위에 해당 포인트 위에 over 될때 표시될 내용을 정의한다.	45
4.3.6 범례(Legend) 설정	47
4.3.7 차트유형별 설정(PlotOptions) 설정	49
4.3.7.1 꺾은선(Line) 차트 설정	51
4.3.7.2 영역(Area) 차트 설정	53
4.3.7.3 부드러운선(Spline)차트 설정	55
4.3.7.4 산점(Scatter)차트 설정	57
4.3.7.5 막대(Column)차트 설정	59
4.3.7.6 가로막대(Bar)차트 설정	61
4.3.7.7 파이(Pie) 차트 설정	63
4.3.7.8 박스플롯(Box Plot) 차트 설정	65
4.3.7.9 방사형(Polar) 차트 설정	67
4.4 데이터 조회	69
4.5 속성 변경사항 적용하기	70
4.6 데이터 XML 구현하기	71
4.7 데이터 JSON 구현하기	72
Chapter 5. 개발 실습	74
5. 개발 실습	76
5.1 IBChart 객체 생성 및 기본 속성 설정하기	76
5.2 조회하기	77
Chapter 6. 데이터 구조	82
6. 데이터 구조	84
6.1 데이터 XML	85
6.2 데이터 JSON	87
Chapter 7. IBChart 상수	90
7. IBChart 상수	92
7.1 Chart Type	92



7.2 Zoom Type	92
7.3 가로 정렬	93
7.4 세로 정렬	93
7.6 Layout	93
7.7 Dash Style	93
7.8 Symbol	94
7.9 이미지 저장 타입	94
7.10 Stacking	95
7.11 Tick Position	95
7.12 Cursor Type	95
7.13 Axis Title Align	95
 Chapter 8. IBChart Config	 97
 8. IBChart Config	 99
8.1 Config 설정	99
8.2 Property List	100
 Chapter 9. IBChart Events	 102
 9. IBChart Events	 104
9.1 Event 사용	104
9.2 Event List	105
OnClick	106
OnLegendItemClick	107
OnLoad	108
OnPointClick	109
OnRedraw	110
OnSeriesHide	111
OnSeriesMouseOut	112
OnSeriesMouseOver	113
OnSeriesShow	114
OnSearchEnd	115



OnZoomIn	116
OnZoomOut	117
Chapter 10. IBChart Methods	118
10. IBChart Methods	120
10.1 Chart Object	120
AddSeries	121
CreateSeries	122
DoPrint	123
Down2Image	124
Draw	125
DoSearch	126
GetEtcData	127
GetNumberFormat	128
GetSearchData	129
GetSeries	130
GetSeriesCount	131
GetXAxis	132
GetYAxis	133
LoadSearchData	134
RemoveAll	135
RemoveSeries	136
SetBaseOptions	137
SetColors	140
SetCreditsOptions	141
SetLabelsOptions	143
SetLegendOptions	145
SetNumericSymbols	148
SetPaneOptions	149
SetPlotOptions	152
SetOptions	154
SetSeriesOptions	156



SetSubtitleOptions	160
SetTitleOptions	162
SetToolTipOptions	164
SetXAxisOptions	167
SetYAxisOptions	171
UpdateSeries	175
10.2 Series	176
AddPoint	177
AddPoints	178
GetName	180
SetData	181
SetOptions	182
SetVisible	184
10.3 Axis	185
GetLabel	186
GetDataMax	187
GetDataMin	188
GetMax	189
SetOptions	190
Chapter 11. IBChart Property	193
11. IBChart Property	195
11.1 PlotOptions	195
Animation	197
BorderColor	198
BorderRadius	200
BorderWidth	202
Center	204
Color	206
ColorByPoint	208
Colors	210
Cursor	212



DashStyle	213
DataLabels	215
FillColor	218
GroupPadding	220
InnerSize	222
LineColor	224
LineWidth	226
NegativeColor	228
NegativeFillColor	230
PointPadding	232
PointPlacement	234
PointWidth	236
Shadow	238
ShowInLegend	240
Stacking	242
StartAngle	244
Step	246
Tooltip	248
Visible	250
Appendix	252
1. 특수차트	253
1.1 Angular gauges chart	254
1.2 ColumnRange chart	256
1.3 ErrorBar chart	258
1.4 Funnel chart	259
1.5 Bubble chart	261
1.6 HeapMap chart	263
1.7 Prymid chart	266
2. 차트기능	268
2.1 Stack 의 그룹기능	269



2.2 멀티 차트 표현 기능	271
2.3 Secondary Y 축 사용	273
2.4 PlotLine 기능	275
2.5 3D 차트 기능	277
2.5.1 막대(Column) 3D 차트	278
2.5.2 가로막대(Bar) 3D 차트	280
2.5.3 파이(Pie) 3D 차트	282
3. IBSheet 데이터 연동	284



Chapter 1. IBChart 개요



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.

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
- 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 IBChart 의 적용 대상 업무

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

1.4 IBChart 의 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

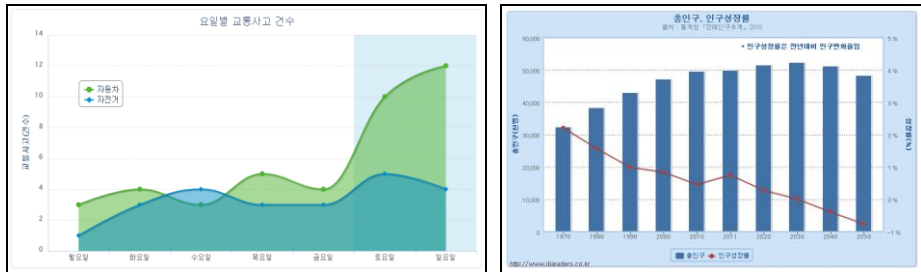
IBChart는 기존 웹 개발자들에게 익숙한 자바스크립트를 이용하여 차트의 속성을 설정하고 제어할 수 있다.

➤ 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

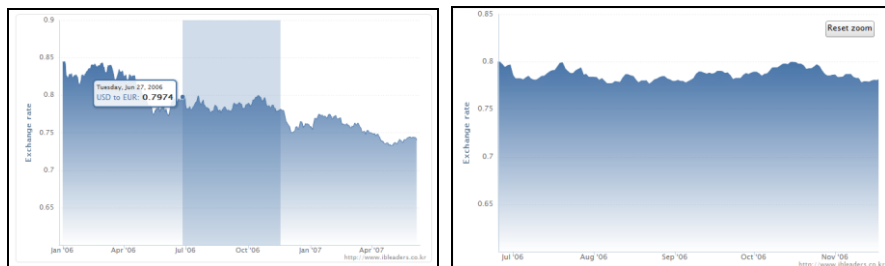
IBChat는 차트 요소의 다양한 디자인 속성 이외에 차트를 조금 더 화려하게 보일 수 있는 특수 효과를 제공한다.



[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



(주) 아이비 리더스

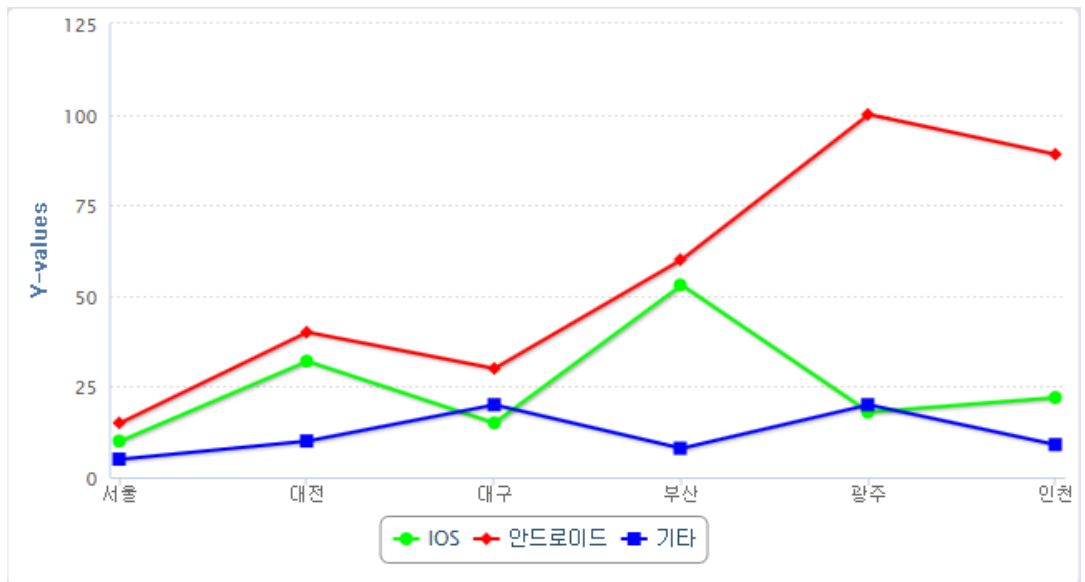
Inno Biz Leaders Co., Ltd.

2. IBChart의 구조

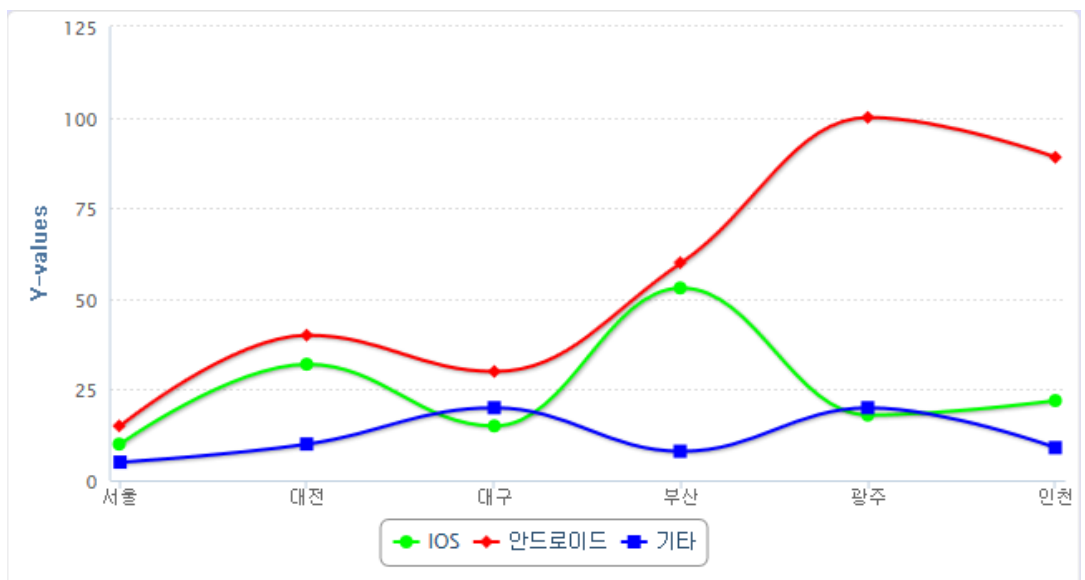
2.1 IBChart 의 형태

➤ IBChart의 형태는 아래와 같이 표현할 수 있다.

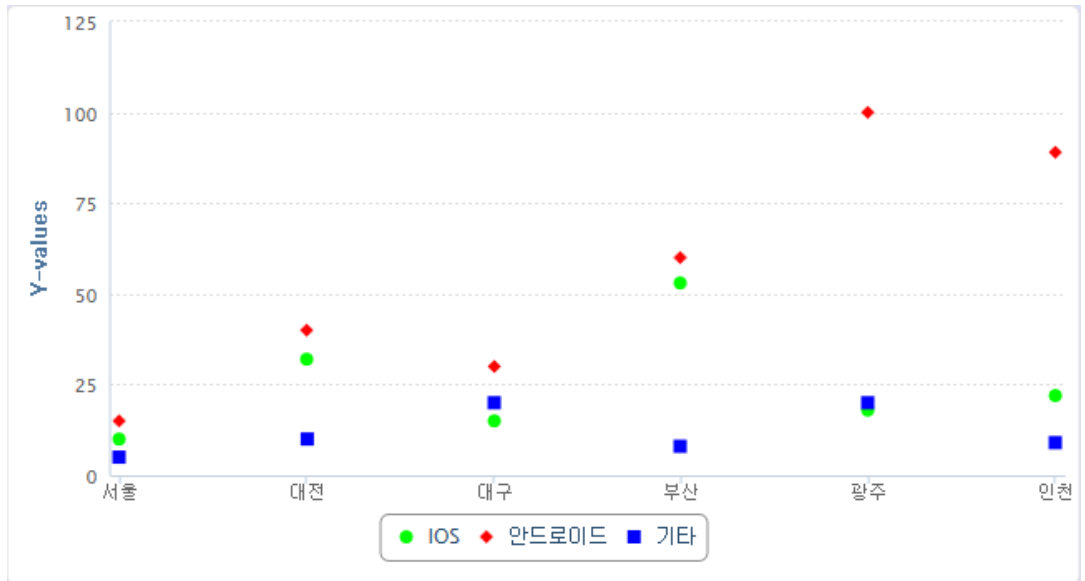
● 꺾은선(Line Type)



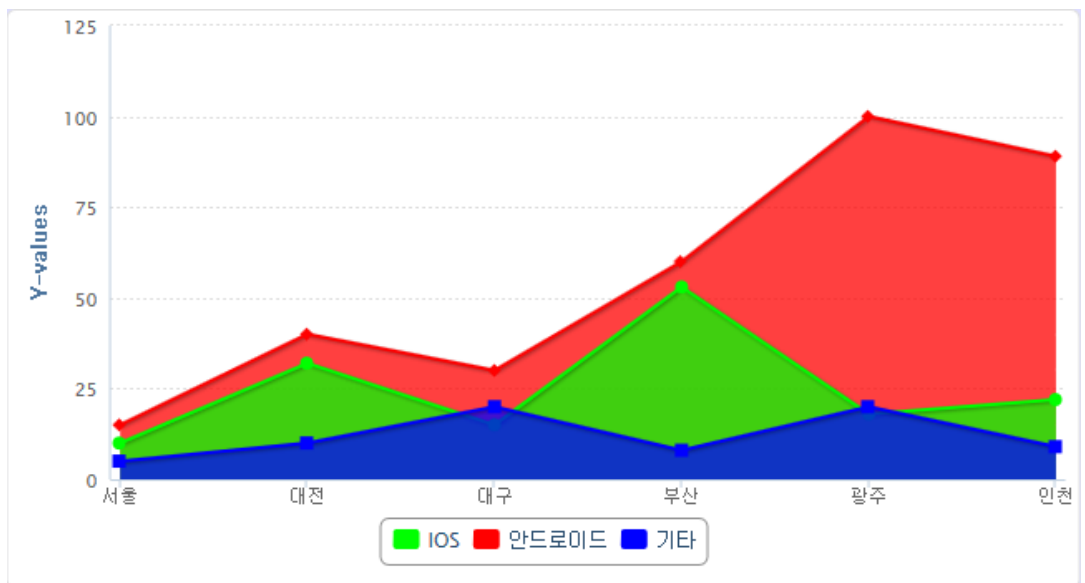
● 부드러운 선(Spline Type)



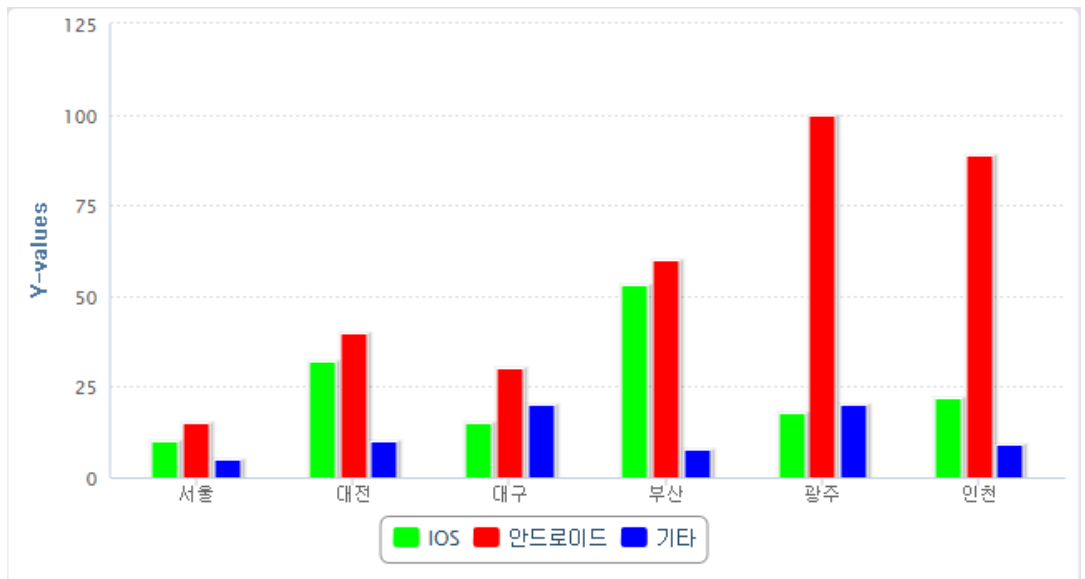
● 산점(Scatter Type)



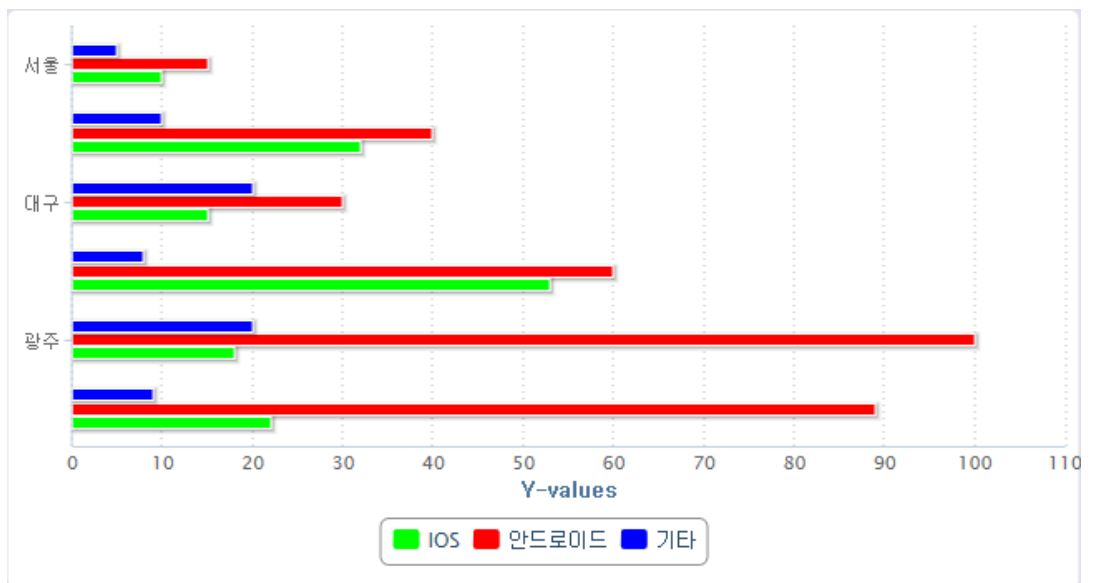
● 영역(Area Type)



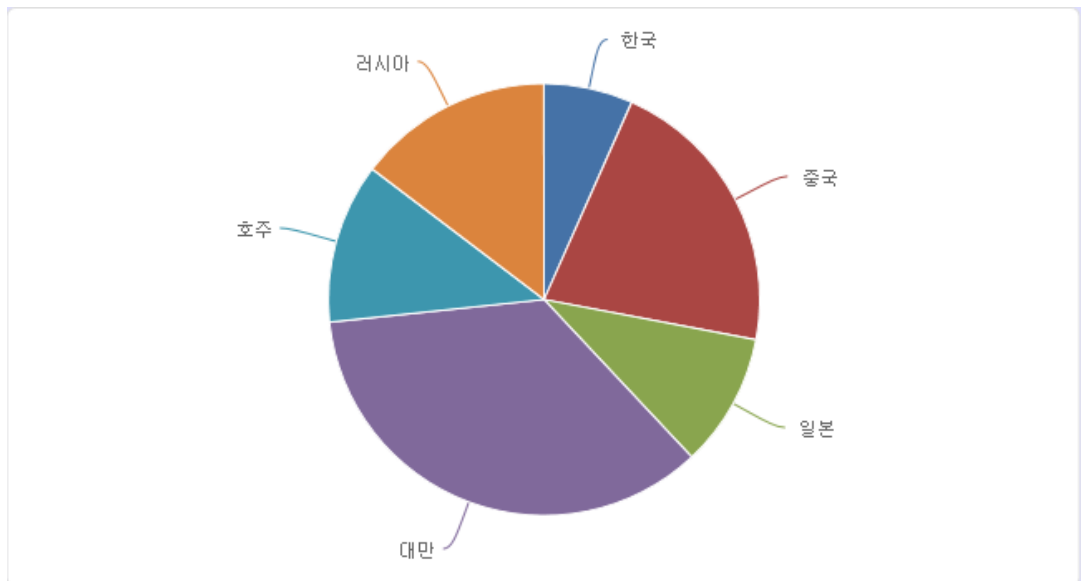
● 막대(Column Type)



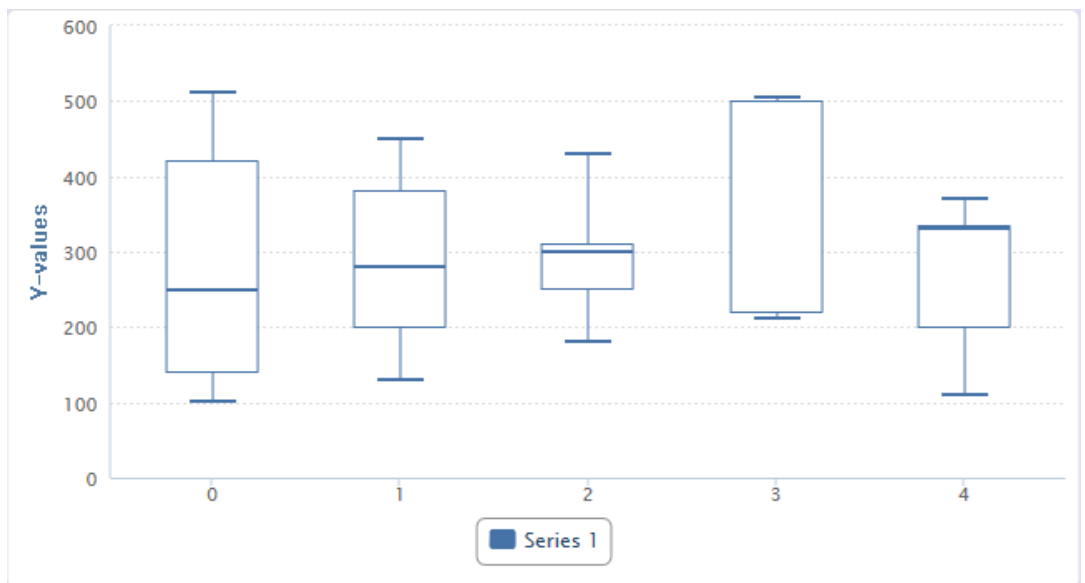
● 가로막대 (Bar Type)



● 파이 (Pie Type)



- 박스플롯(Box Plot Type)



- 방사형(Polar Type)



2.2 IBChart 의 각 부분 명칭



- ① Title : 차트의 제목을 나타내면 기본적으로 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 Axes : 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 웹서버의 특정 폴더에 복사하여 설치한다.

위치는 다음을 기준으로 한다.

```
\Chart\
\Chart\ibchart.js
\Chart\ibchartinfo.js
\Chart\ibchartmsg.xml
\Chart\ibchart.cfg
\Chart\Down2Image.jsp
```

각각 파일의 내용을 설명하자면 아래와 같다.

파일명	내용/역할
ibchart.js	IBChart 엔진 js 파일
ibchartinfo.js	IBChart 생성 함수 및 개발시 유용하게 사용할 수 있는 js함수를 담아둔 파일(지속적 업데이트 예정)
ibchartmsg.xml	IBChart에서 사용하는 메시지 xml 파일 다국어 사용시 해당 파일을 수정하여 지원가능
ibchart.cfg	모든 차크에 전역으로 설정할 값을 정의하기 위한 파일.
Down2Image.jsp	이미지 다운로드 전용함수의 기능 구현을 위한 서버쪽 응용 프로그램 파일 ※ 한글이 깨질경우 jsp 내에서 UTF-8 환경설정 변수값을 변경해야 한다. ※ 이미지 다운로드 jsp 프로그램 내에서 이미지 파일을 서버 디스크에 임시 저장하므로 TempFolder 변수에 서버의 물리적 디렉토리 경로를 설정해야 한다.
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. ibsheet7.jar 파일과 함께 jar 파일이 다소 많은데 본사에 서버모듈 jar 파일 배포시 모두 포함되어 있는 내용이다.

(예)

```
/Tomcat60/lib/ibsheet7.jar
/Tomcat60/lib/batik-anim.jar
/Tomcat60/lib/batik-awt-util.jar
/Tomcat60/lib/batik-bridge.jar
/Tomcat60/lib/batik-codec.jar
/Tomcat60/lib/batik-css.jar
/Tomcat60/lib/batik-dom.jar
/Tomcat60/lib/batik-ext.jar
/Tomcat60/lib/batik-extension.jar
/Tomcat60/lib/batik-gui-util.jar
/Tomcat60/lib/batik-gvt.jar
/Tomcat60/lib/batik-parser.jar
/Tomcat60/lib/batik-script.jar
/Tomcat60/lib/batik-svg-dom.jar
/Tomcat60/lib/batik-svggen.jar
/Tomcat60/lib/batik-swing.jar
/Tomcat60/lib/batik-transcoder.jar
/Tomcat60/lib/batik-util.jar
/Tomcat60/lib/batik-xml.jar
/Tomcat60/lib/cos.jar
/Tomcat60/lib/js.jar
```



/Tomcat60/lib/pdf-transcoder.jar
/Tomcat60/lib/xalan-2.6.0.jar
/Tomcat60/lib/xerces_2_5_0.jar
/Tomcat60/lib/xml-apis-ext.jar
/Tomcat60/lib/xml-apis.jar
/Tomcat60/lib/poi-3.9-20121203.jar
/Tomcat60/lib/poi-ooxml-3.9-20121203.jar
/Tomcat60/lib/poi-ooxml-schemas-3.9-20121203.jar



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.



Chapter 4. 개발 절차



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.

4. 개발 절차

4.1 웹 페이지에 JS 파일 링크 걸기

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

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js" ></script>
<script type="text/javascript" src="/Chart/ibchart.js"></script>
<script type="text/javascript" src="/Chart/ibchartinfo.js"></script>
```

위 세개 파일 중 ibchart.js 파일은 IBChart 엔진 js 파일이다. 그리고 ibchartinfo.js 파일은 차트 생성 함수 및 개발시 유용하게 사용할 수 있는 js함수를 담아둔 파일이다. **jquery.min.js** 파일은 반드시 **ibchart.js** 보다 앞에 링크걸도록 한다.

4.2 IBChart 객체를 화면에 추가

js 파일을 추가한 뒤에 IBChart 객체를 화면에 추가한다.

추가하는 방법은 ibchartinfo.js에 있는 **createIBChart(chartId, width, height)**함수를 이용하여 추가하게 된다.

```
<body>
  <table class='chart_table'>
    <tr>
      <td>
        <script type="text/javascript">
          //IBChart 객체 생성
          createIBChart("myChart", "500px", "400px");
        </script>
      </td>
    </tr>
  </table>
</body>
```

위와 같이 공통 스크립트를 사용하면 모든 화면에 공통적으로 동일한 IBChart 객체를 추가할 수 있다.

4.3 초기 항목 설정

개체가 생성된 후 페이지가 onload 될 때 차트의 초기 항목을 설정한다. 아래와 같이 Body 태그의 onload 이벤트에서 IBChart에 대한 초기 항목을 설정할 함수를 정의한다.

```
<Body onload="기본 속성을 설정할 함수명()">
```

Body 태그의 onload 이벤트에서 정의하는 것은 페이지의 로드가 완료되는 시점에 IBChart의 객체도 생성이 완료되므로 그때 IBChart의 제공 기능들을 사용할 수 있기 때문이다. 만약 객체 생성 전에 IBChart 기능을 사용하려고 한다면 자바스크립트 오류가 발생할 수 있다.

4.3.1 차트 기본 속성 설정

차트의 기본 속성을 공통으로 설정한다. 기본 속성 설정에서는 아래의 상세 설정들을 모두 추가 할 수 있으며, 따로 파일을 만들어 공통으로 관리가 가능하다.

➤ SetOptions 속성

속성	타입	설명
Chart	Object	차트배경 및 형태 등을 설정 (SetBaseOptions 함수 참고)
Legend	Object	범례의 위치 및 형태 설정. (4.3.6 혹은 SetLegendOptions 함수 참고)
PlotOptions	Object	plot영역 (차트세부)설정 설정(4.3.7 혹은 SetPlotOptions 참고)
XAxis	Object	X축을 설정(4.3.3 나 SetXAxisOptions 참고)
YAxis	Object	Y축을 설정(4.3.3 나 SetYAxisOptions 참고)

➤ Example

```
//차트 기본 속성 설정
myChart.SetOptions({
  Chart : {
    BackgroundColor : "#dddddff", //차트 배경색 설정
    Type : "column" //차트 Type 설정
  },
  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,
```

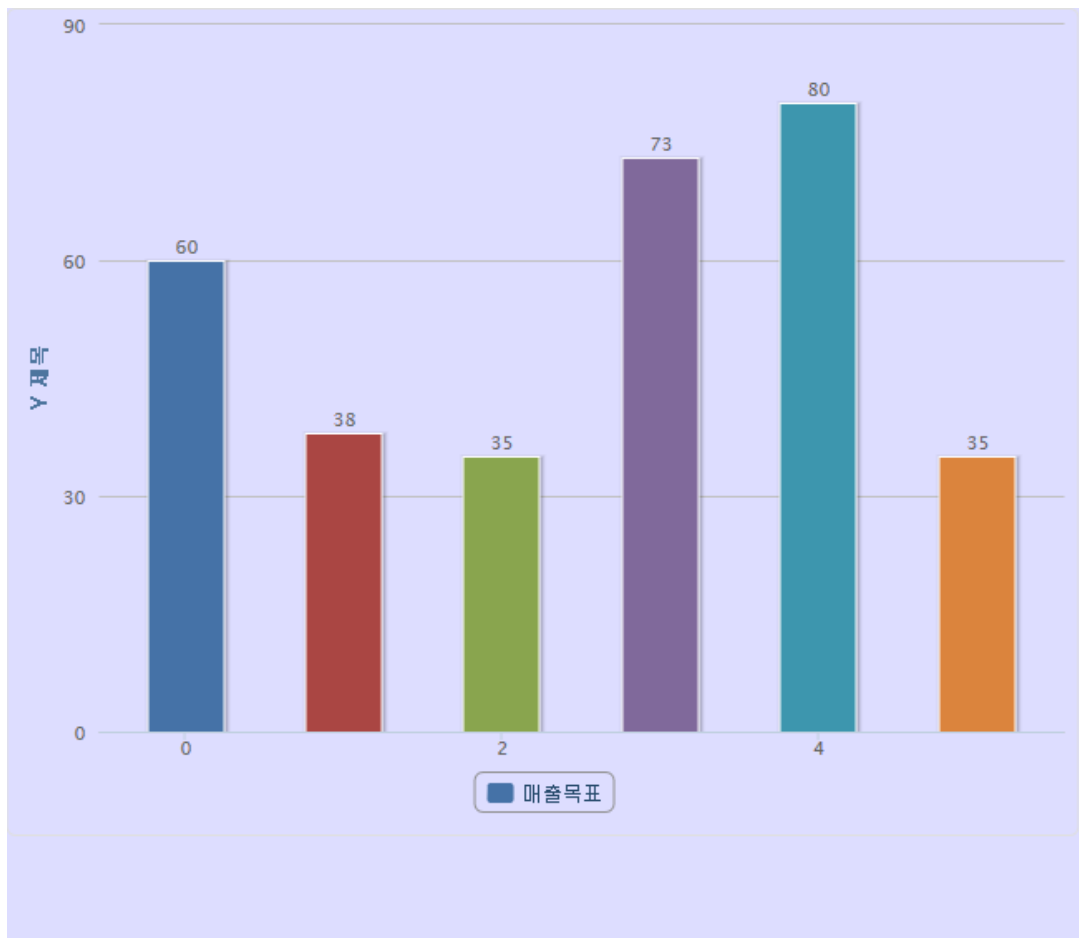


```
        }
    },
    Column : {
        PointPadding : 0.1 // 컬럼간의 간격 설정
    },
},
XAxis : {
    TickInterval : 2, //X축 레이블 간격 설정
    Labels : { //X축 레이블 설정
        Enabled : true
    }
},
YAxis : {
    TickInterval : 30, //Y축 레이블 간격 설정
    Min : 0, //Y축 Min값 설정
    Title : { //Y축 제목 설정
        Text : "Y 제목"
    }
}
});
//Series 설정
myChart.SetSeriesOptions([
    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"}
    ]
], 1);
```



```
myChart.Draw();
```

➤ **Result**



4.3.2 타이틀 및 서브타이틀 속성

타이틀과 서브타이틀을 화면에 추가한다.

추가 옵션(align, float, margin, verticalAlign, x, y)을 이용하여 타이틀과 서브타이틀을 이동할 수 있다. 자세한 내용은 [SetTitleOption\(\)](#), [SetSubtitleOption\(\)](#) 메서드를 참고.

속성	타입	설명
SetTitleOptions	Object	메인타이틀 설정 (Json 형식)
SetSubtitleOptions	Object	서브타이틀 설정 (Json 형식)

➤ Example

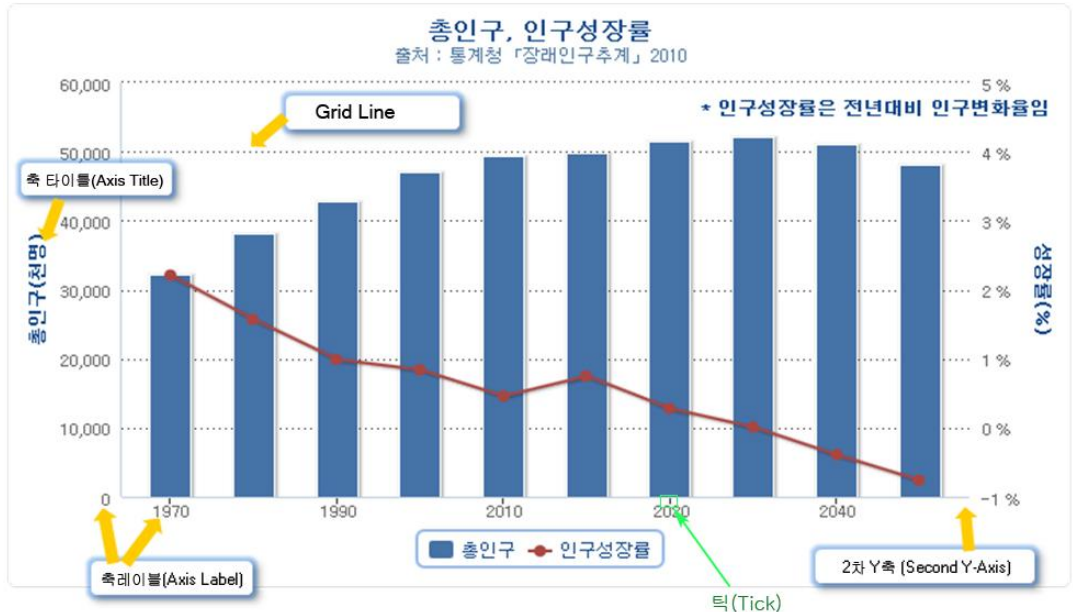
```
myChart.SetTitleOptions({
  text:'cc별 중고자동차 판매량'
},1);
myChart.SetSubTitleOptions({
  text:'2014년 1/4분기'
},1);
myChart.Draw();
```

➤ Result

cc별 중고자동차 판매량
2014년 1/4분기

4.3.3 축(Axis) 속성

X축과 Y축은 파이,방사형 차트를 제외한 대부분의 차트에서 기본적으로 표현된다.



➤ [SetXAxisOptions](#), [SetYAxisOptions](#) 속성

속성	타입	설명
Title	Object	축 타이틀 설정 (Json 형식)
GridLineWidth	int	축 그리드 라인 굵기 설정
TickInterval	int	틱객체 표시 간격 설정
Opposite	Boolean	Second 축 설정
Categories	Object	축레이블 설정

➤ Example

```

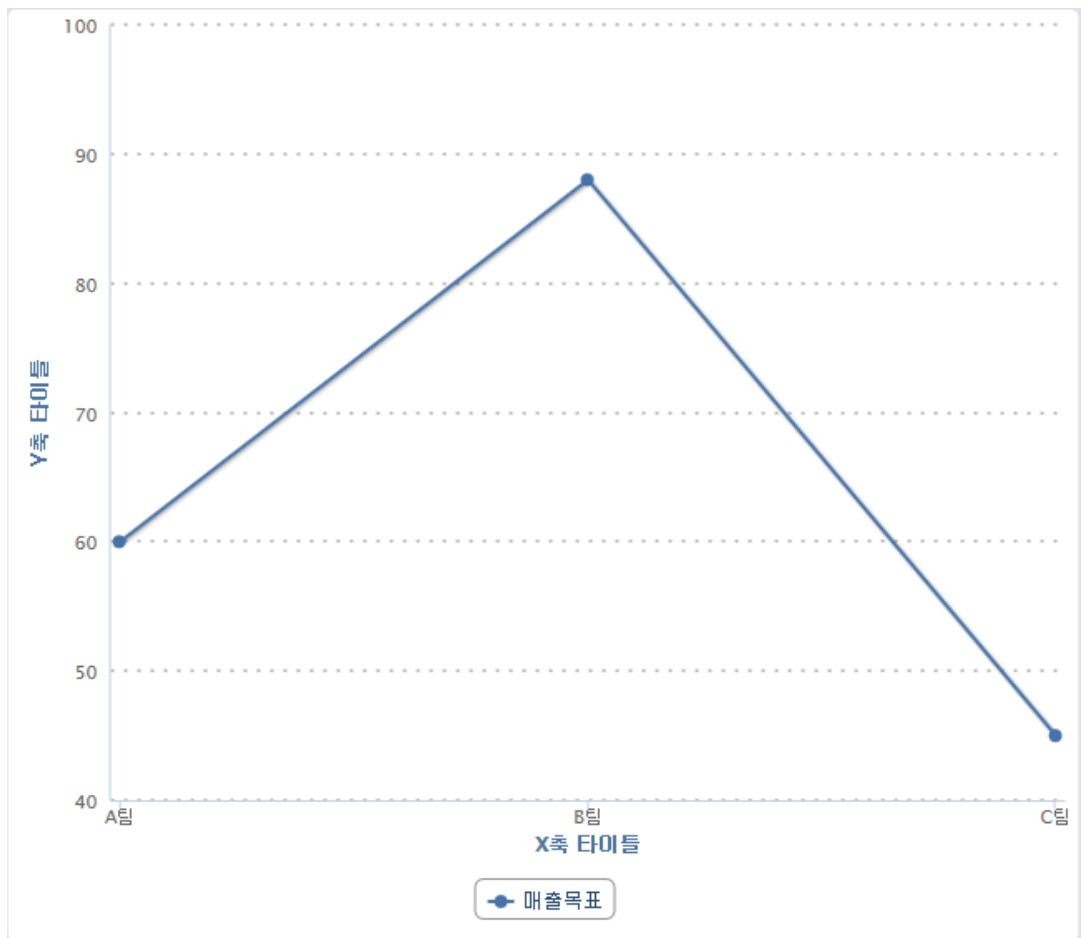
// X축 설정
myChart.SetXAxisOptions({
  Title:{
    Text:"X축 타이틀" // 타이틀 설정
  },
  Categories : ["A팀", "B팀", "C팀"] //X축 레이블 설정

```




```
}, 1);  
//Y축 설정  
myChart.SetYAxisOptions({  
    Title:{  
        Text:"Y축 타이틀" //타이틀 설정  
    },  
    GridLineWidth : 2, //라인 넓이설정  
    TickInterval : 10 // Tick 간격을 10으로 설정  
}, 1);  
//Series 설정  
myChart.SetSeriesOptions([{  
    Name : "매출목표",  
    Data : [  
        {Y:60, Color : "#4572A7"},  
        {Y:88, Color : "#AA4643"},  
        {Y:45, Color : "#89A54E"},  
    ]  
}], 1);  
myChart.Draw();
```

➤ **Result**



4.3.4 시리즈(Series) 속성

plot 영역에 표시될 시리즈 객체를 생성하고 데이터를 입력한다.

➤ [SetSeriesOptions](#) 속성

속성	타입	설명
Name	String	범례(Legend)에 표시될 시리즈 이름
Data	Object	Series의 Data 설정(Json 배열)
Color	String	각 차트 시리즈의 색상 설정

➤ **Example**

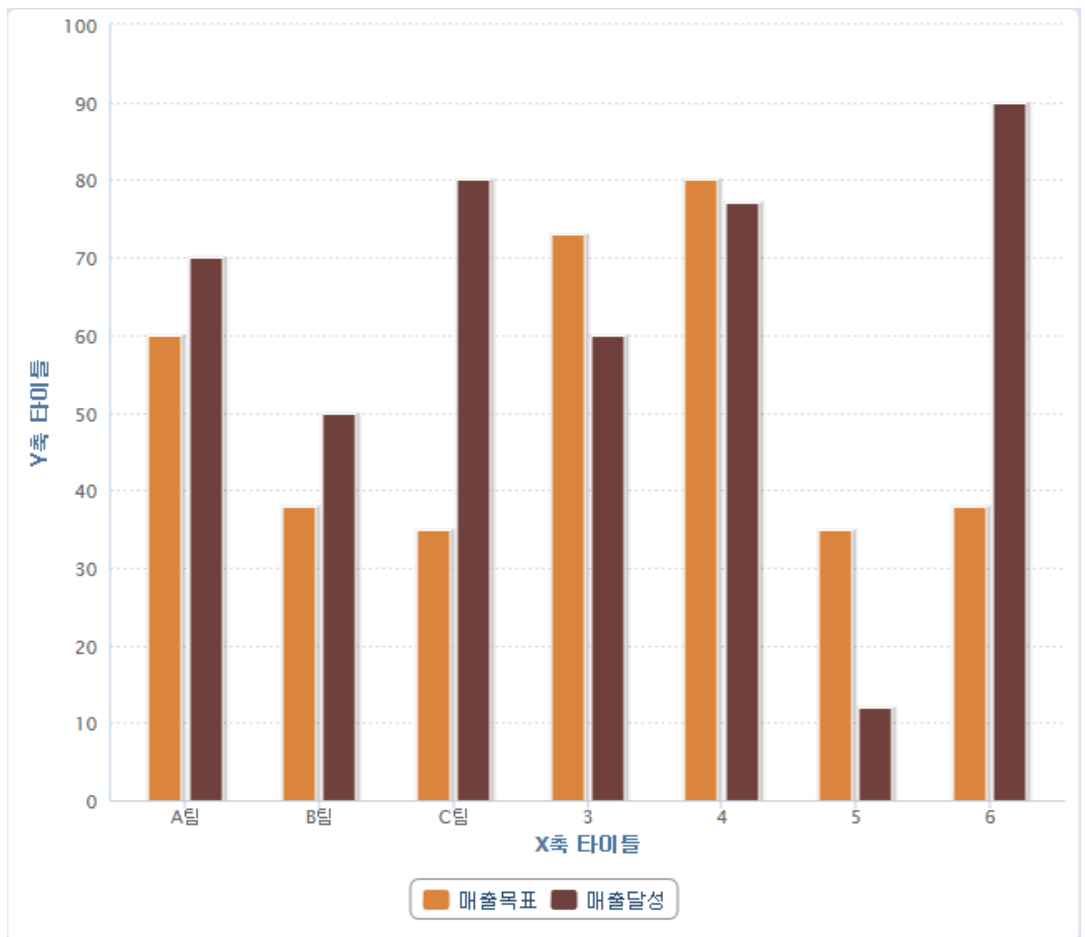
```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"column"
  }
});

//Series 설정
myChart.SetSeriesOptions([
  Name : "매출목표", //Series Name 설정
  Data : [ //Series Data 설정
    {Y:60}, {Y:38}, {Y:35}, {Y:73}, {Y:80}, {Y:35}, {Y:38}
  ],
  Color : "#DB843D" //Series 컬러 설정
},{
  Name : "매출달성",
  Data : [
    {Y:70}, {Y:50}, {Y:80}, {Y:60}, {Y:77}, {Y:12}, {Y:90}
  ],
  Color : "#70403D"
}], 1);

myChart.Draw();
```



➤ Result



4.3.5 툴팁(ToolTip) 설정

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

➤ [SetToolTipOptions](#)

➤ 속성

속성	타입	설명
Enabled	Boolean	Tooltip 사용여부
BackgroundColor	String	배경색상 설정
BorderColor	String	윤곽선 설정
BorderWidth	int	윤곽선 굵기 설정
Formatter	function	툴팁 값에 대한 format 설정

➤ **Example**

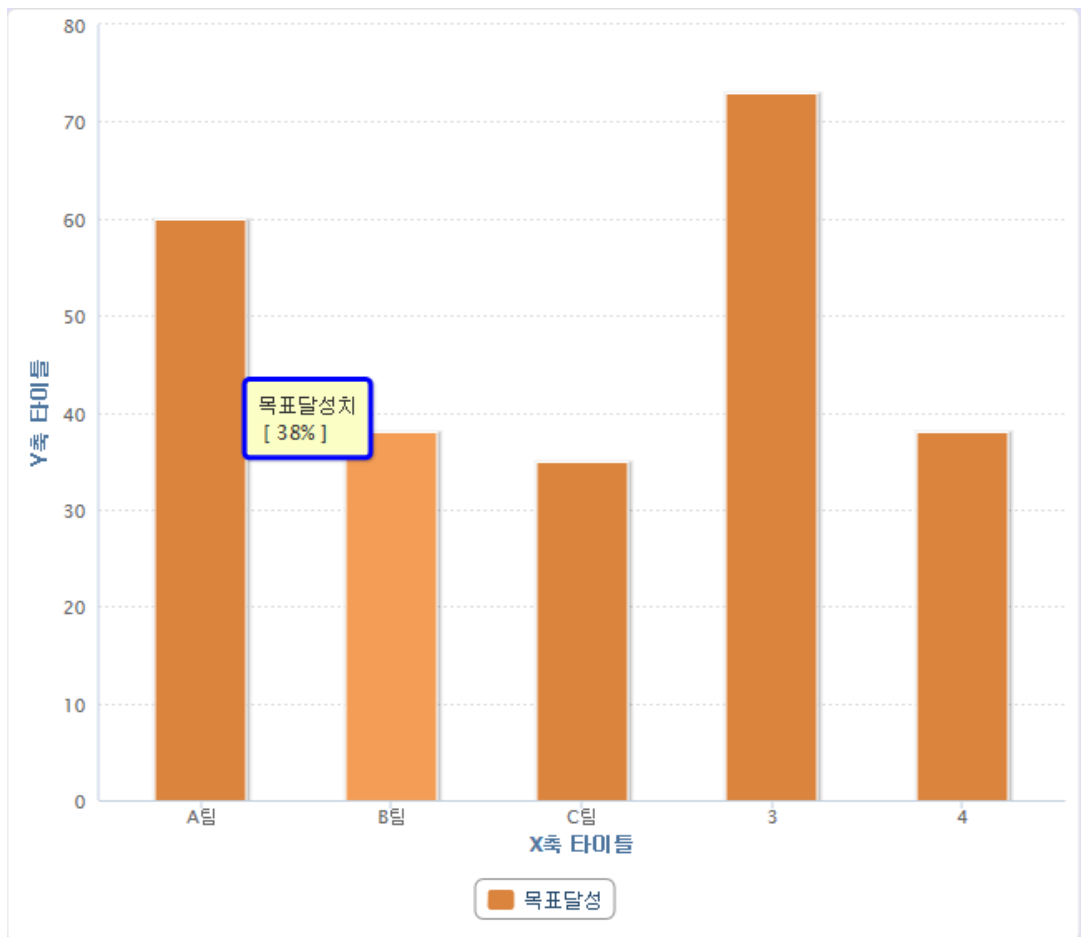
```
//차트 Tooltip 속성 설정
myChart.SetToolTipOptions({
    Enabled :true,    //Tooltip 사용여부
    Formatter : function(){    //Tooltip Formatter 설정
        return '목표달성치<br> [ '+this.y+'% ]';
    },
    BackgroundColor:'#FCFFC5',    //Tooltip 배경색
    BorderColor:'blue',    //Tooltip Border
    BorderWidth:3    //Border 굵기
});

//Series 설정
myChart.SetSeriesOptions([
    Name : "목표달성",
    Data : [
        {Y:60},    {Y:38},
        {Y:35},    {Y:73},
        {Y:38}
    ],
    ],
```

```

Color : "#DB843D"
});
myChart.Draw();
    
```

➤ Result



4.3.6 범례(Legend) 설정

➤ [SetLegendOptions](#) 속성

속성	타입	설명
Enabled	Boolean	Legend 표시여부
Layout	String	Legend Item 표시방법 (horizontal,vertical)
Align	String	Legend의 가로상의 위치(left, right, center)
VerticalAlign	String	Legend의 수직정렬(top, middle, bottom)

➤ **Example**

```
//차트속성
myChart.SetOptions({
    Chart:{
        Type:"column"        //차트종류
    }
});

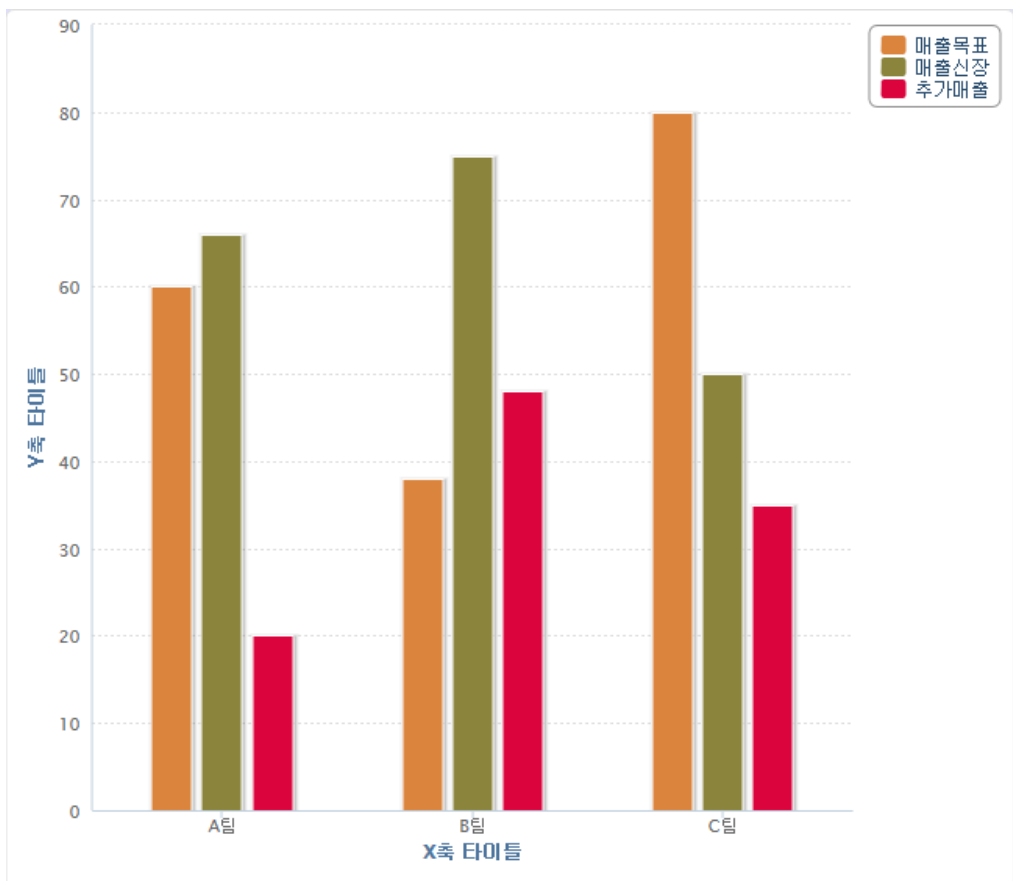
//차트 Legend 설정
myChart.SetLegendOptions({
    Layout:"vertical",
    Align:"right",
    VerticalAlign:"top"
});

//차트 Series 설정
myChart.SetSeriesOptions([
    {
        Name : "매출목표",
        Data : [
            {Y:60},{Y:38},{Y:80}
        ],
        Color : "#DB843D"
    },{
        Name : "매출신장",
        Data : [
```



```
{Y:66},{Y:75},{Y:50}
],
Color : "#8B843D"
},{
Name : "추가매출",
Data : [
{Y:20},{Y:48},{Y:35}
],
Color : "#DB043D"
}], 1);
myChart.Draw();
```

➤ Result



4.3.7 차트유형별 설정(PlotOptions) 설정

➤ [SetPlotOptions](#) 속성

속성	타입	설명
Series	Object	시리즈 속성 설정(Json 형식)
Bar	Object	Bar 차트 속성 설정(Json 형식)
Column	Object	Column 차트 속성 설정(Json 형식)
Pie	Object	Pie 차트 속성 설정(Json 형식)
Line	Object	Line 차트 속성 설정(Json 형식)

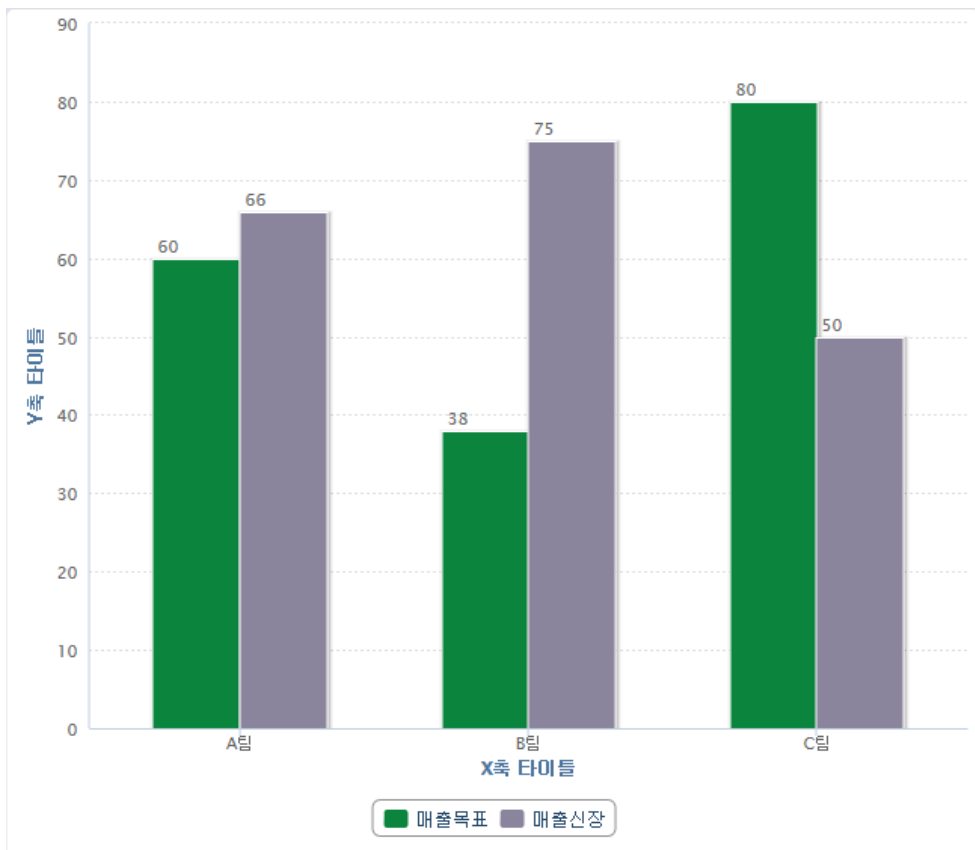
➤ **Example**

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"column"      //차트종류
  }
});
//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Series:{              // Series 속성
    DataLabels:{        //Series 데이터 속성
      Enabled:true,
      Align:"left"
    }
  },
  Column:{              //컬럼 차트 속성
    PointPadding:0.25   //포인트 간격 설정
  }
});
//차트 Series 설정
myChart.SetSeriesOptions([
  Name : "매출목표",
```



```
Data : [  
    {Y:60},{Y:38},{Y:80}  
],  
    Color : "#0B843D"  
},{  
    Name : "매출신장",  
    Data : [  
        {Y:66},{Y:75},{Y:50}  
    ],  
    Color : "#8B849D"  
}], 1);  
myChart.Draw();
```

➤ Result



4.3.7.1 꺾은선(Line) 차트 설정

SetPlotOptions의 Line 속성에서 설정한다.

➤ Line 속성

속성	타입	설명
Cursor	String	라인에 커서 설정
DashStyle	String	라인 DashStyle 설정

➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"line" //차트종류 - 라인차트 설정
  }
});

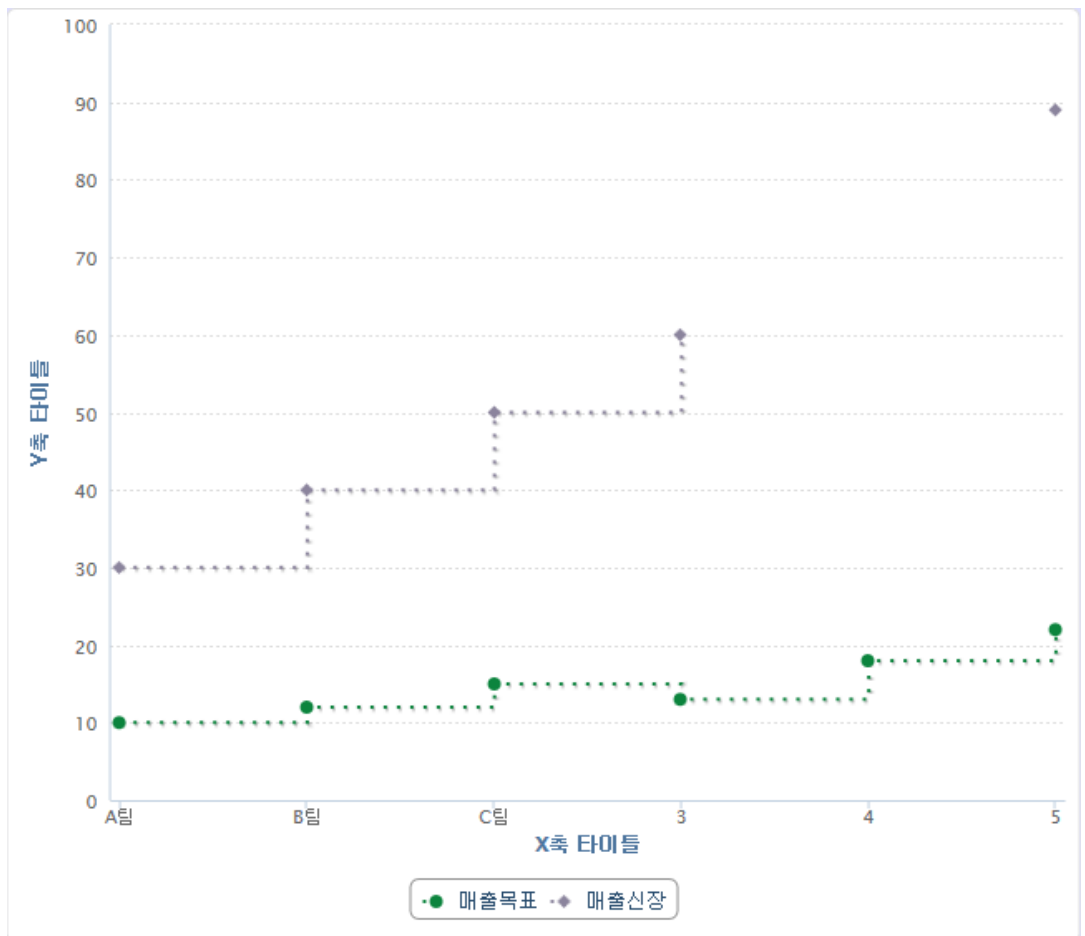
//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Line:{ //컬럼 차트 속성
    Cursor:"pointer", //커서 속성
    DashStyle:"Dot", //선스타일 Dot
    Step:true //Step형태의 라인
  }
});

//차트 Series 설정
myChart.SetSeriesOptions([
  {
    Name : "매출목표",
    Data : [
      {Y:10},{Y:12},{Y:15},{Y:13},{Y:18},{Y:22}
    ],
    Color : "#0B843D"
  },
  {
    Name : "매출신장",
    Data : [
```



```
{Y:30},{Y:40},{Y:50},{Y:60},{Y:null},{Y:89}  
],  
Color : "#8B849D"  
}], 1);  
myChart.Draw();
```

➤ Result



4.3.7.2 영역(Area) 차트 설정

SetPlotOptions의 Area 속성에서 설정한다.

➤ Area 속성

속성	타입	설명
Cursor	String	라인에 커서 설정
DashStyle	String	라인 DashStyle 설정

➤ Example

```
//차트속성
myChart.SetOptions({
    Chart:{
        Type:"area" //area 차트
    }
});

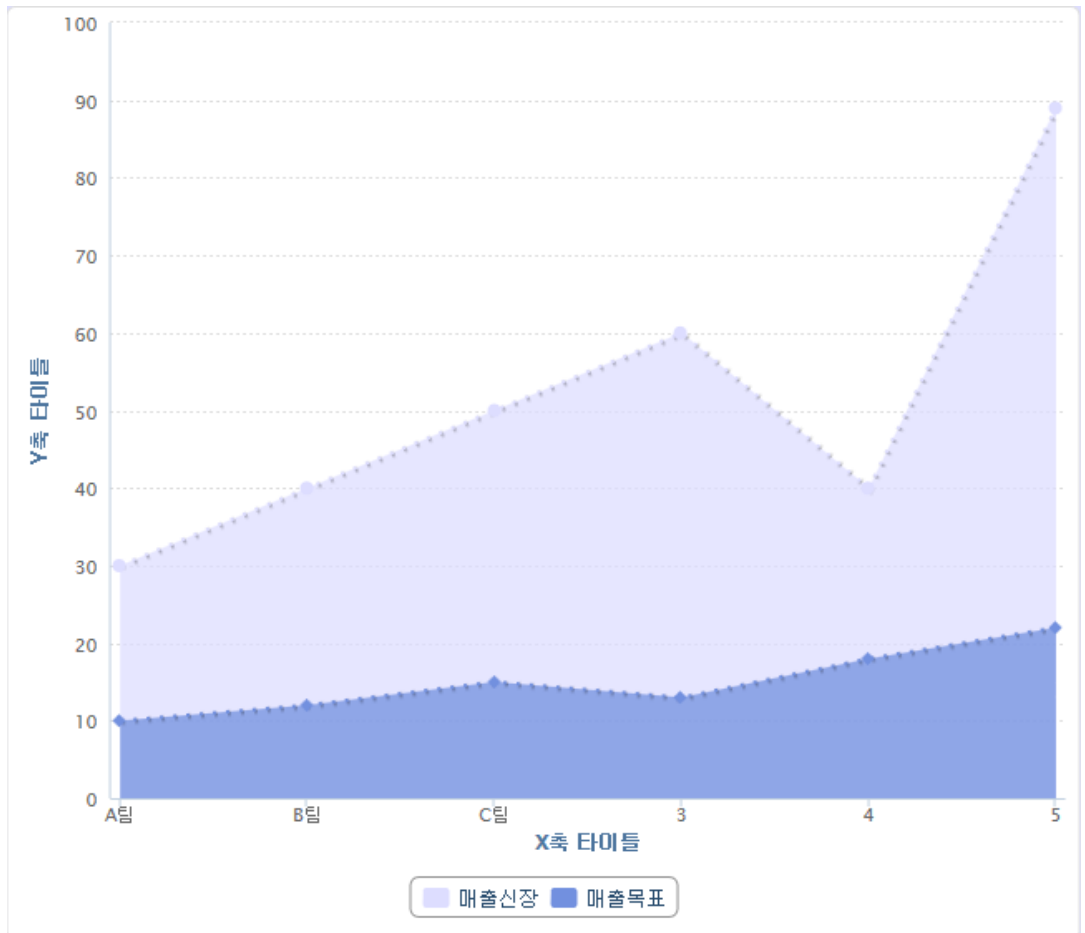
//차트 PlotOptions 속성
myChart.SetPlotOptions({
    Area:{           //컬럼 차트 속성
        Cursor:"pointer",
        DashStyle:"Dot"
    }
});

//차트 Series 설정
myChart.SetSeriesOptions([
    {
        Name : "매출신장",
        Data : [
            {Y:30},{Y:40},{Y:50},{Y:60},{Y:40},{Y:89}
        ],
        Color : "#dddddff"
    },
    {
        Name : "매출목표",
        Data : [
            {Y:10},{Y:12},{Y:15},{Y:13},{Y:18},{Y:22}
        ]
    }
]);
```



```
],  
    Color : "#7290df"  
}], 1);  
myChart.Draw();
```

➤ Result



4.3.7.3 부드러운선(Spline)차트 설정

SetPlotOptions의 Spline 속성에서 설정한다. Line 차트와 속성이 같다. (Step 속성 제외)

➤ Spline 속성

속성	타입	설명
Cursor	String	라인에 커서 설정
DashStyle	String	라인 DashStyle 설정

➤ Example

```
//차트속성
myChart.SetOptions({
    Chart:{
        Type:"spline" //차트종류 - spline 차트 설정
    }
});

//차트 PlotOptions 속성
myChart.SetPlotOptions({
    Spline:{           //컬럼 차트 속성
        Cursor:"pointer", //커서 속성
        DashStyle:"Solid" //선스타일 Dot
    }
});

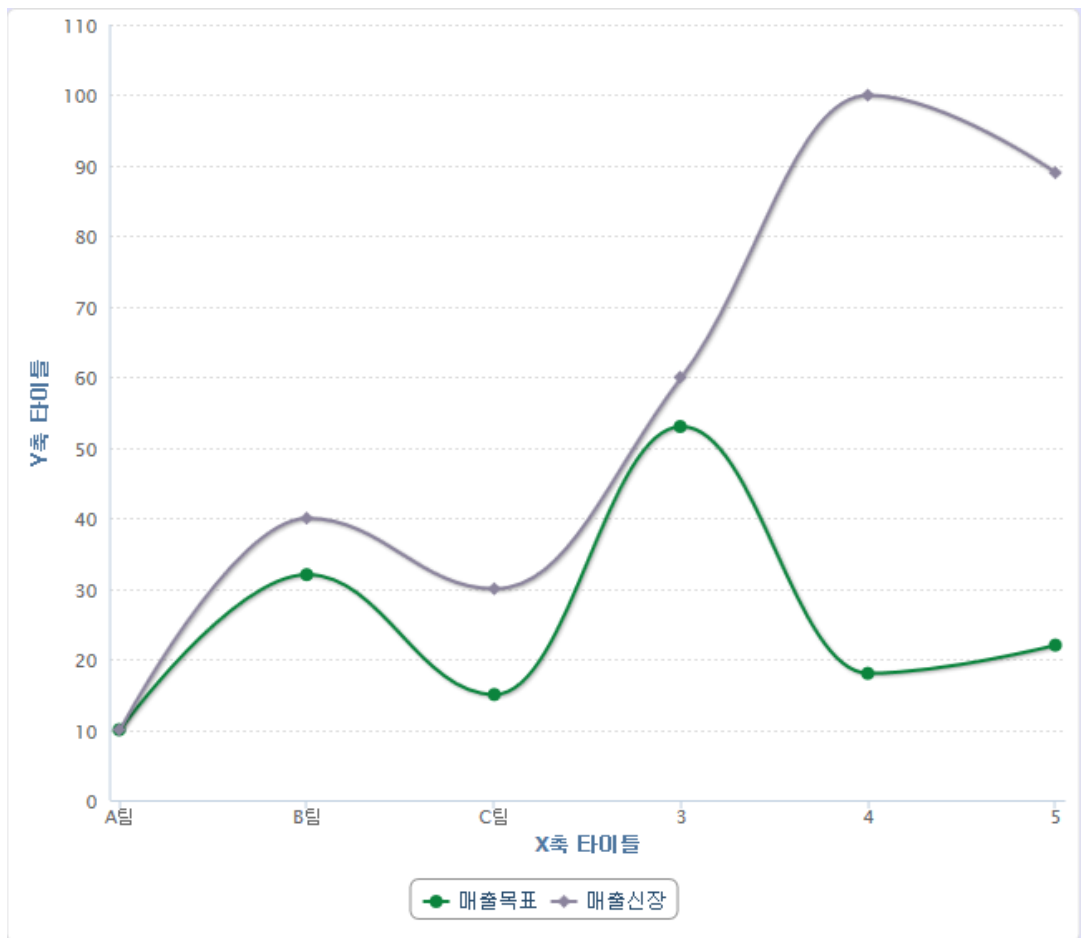
//차트 Series 설정
myChart.SetSeriesOptions([
    {
        Name : "매출목표",
        Data : [
            {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
        ],
        Color : "#0B843D"
    },
    {
        Name : "매출신장",
        Data : [
```

```

        {Y:10},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89}
    ],
    Color : "#8B849D"
}], 1);
myChart.Draw();

```

➤ Result



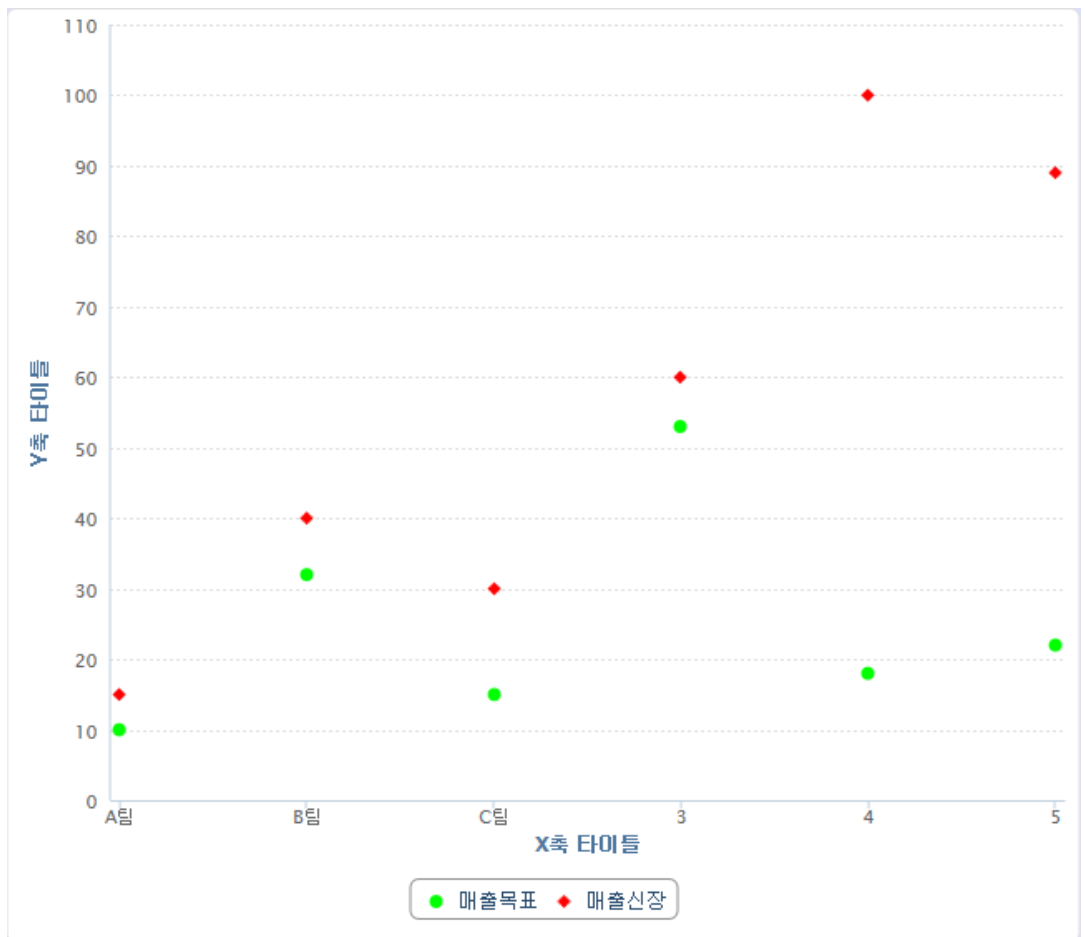


Color : "#FF0000"

```
    }, 1);
```

```
myChart.Draw();
```

➤ Result



4.3.7.5 막대(Column)차트 설정

SetPlotOptions의 Column 속성에서 설정한다.

➤ Column 속성

속성	타입	설명
PointWidth	Number	Point 넓이 설정
DashStyle	String	라인 DashStyle 설정
PointPadding	Number	각 Column의 간격 설정
GroupPadding	Number	각 GroupColumn의 간격 설정

➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"column"      //차트종류 - 컬럼차트 설정
  }
});

//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Column:{             //컬럼 차트 속성
    DashStyle:"Dot",   //선스타일 Dot
    PointPadding:0
  }
});

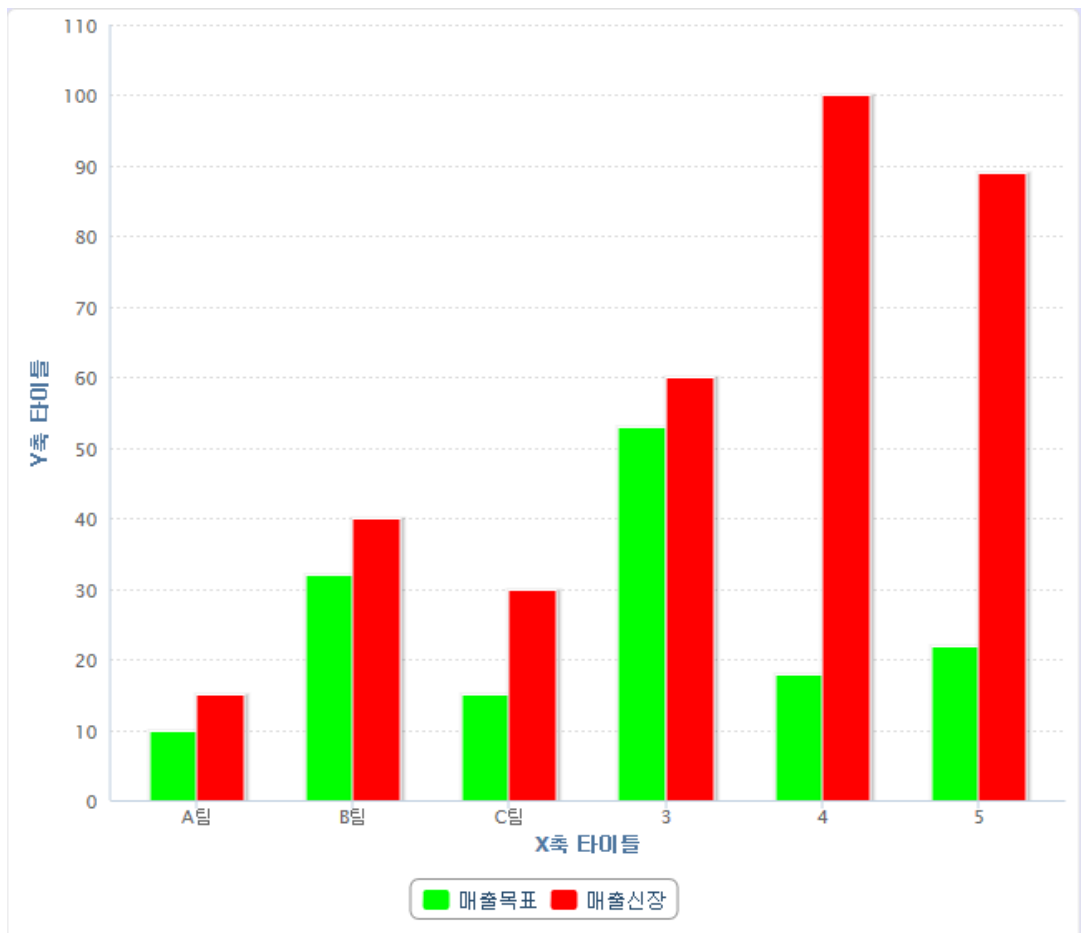
//차트 Series 설정
myChart.SetSeriesOptions([
  Name : "매출목표",
  Data : [
    {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
  ],
  Color : "#00FF00"
},{

```



```
Name : "매출신장",  
Data : [  
    {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89}  
],  
Color : "#FF0000"  
}], 1);  
myChart.Draw();
```

➤ Result



4.3.7.6 가로막대(Bar)차트 설정

SetPlotOptions의 Bar 속성에서 설정한다.

➤ Bar 속성

속성	타입	설명
PointWidth	Number	Point 넓이 설정
DashStyle	String	라인 DashStyle 설정
PointPadding	Number	각 Bar의 간격 설정
GroupPadding	Number	각 GroupBar의 간격 설정

➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"bar"    //차트종류 - bar차트 설정
  }
});

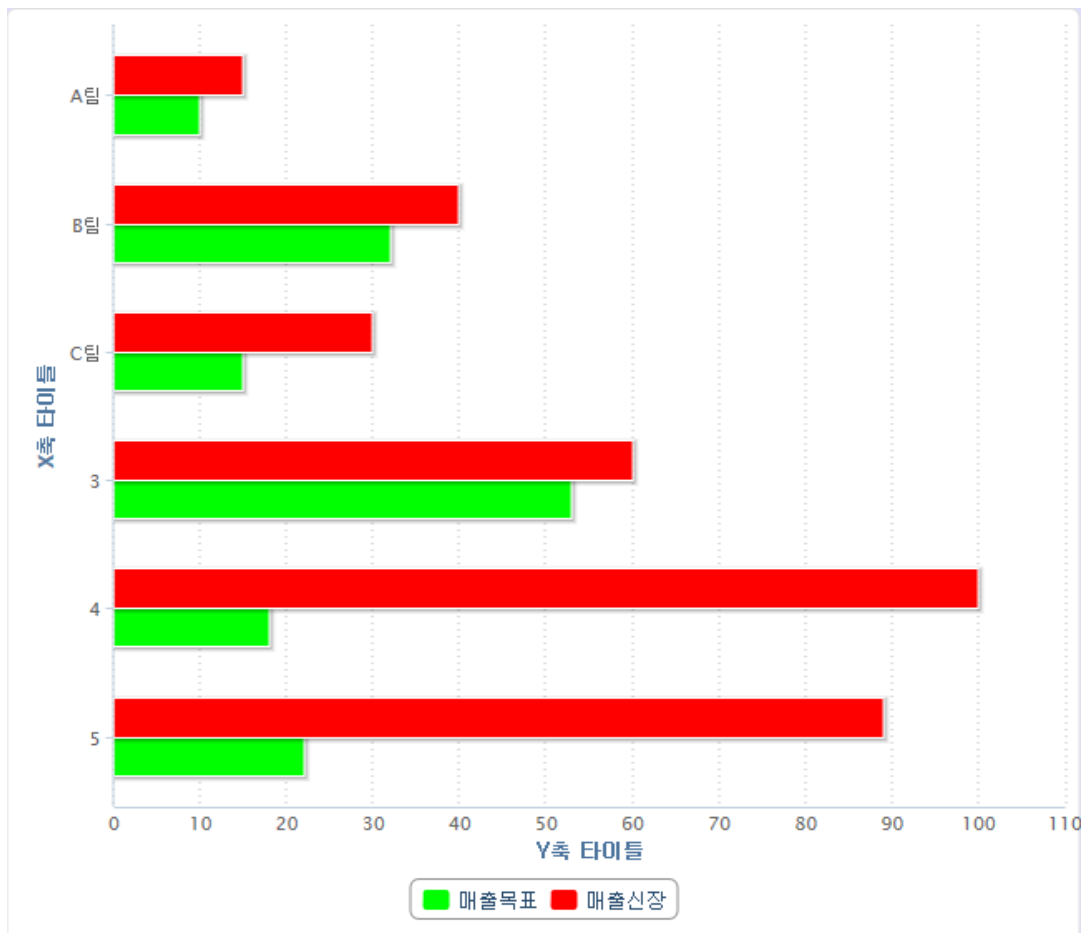
//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Bar:{          //컬럼 차트 속성
    DashStyle:"Dot",    //선스타일 Dot
    PointPadding:0
  }
});

//차트 Series 설정
myChart.SetSeriesOptions([
  Name : "매출목표",
  Data : [
    {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
  ],
  Color : "#00FF00"
},{
  Name : "매출실적",
  Data : [
    {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
  ],
  Color : "#FF0000"
}]
);
```



```
Name : "매출신장",  
Data : [  
    {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89}  
],  
Color : "#FF0000"  
}], 1);  
myChart.Draw();
```

➤ Result



4.3.7.7 파이(Pie) 차트 설정

SetPlotOptions의 Pie 속성에서 설정한다.

➤ Pie 속성

속성	타입	설명
InnerSize	String	Pie차트 안의 공백영역
StartAngle	Number	Pie차트 시작지점

➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"pie" //차트종류 - bar차트 설정
  }
});

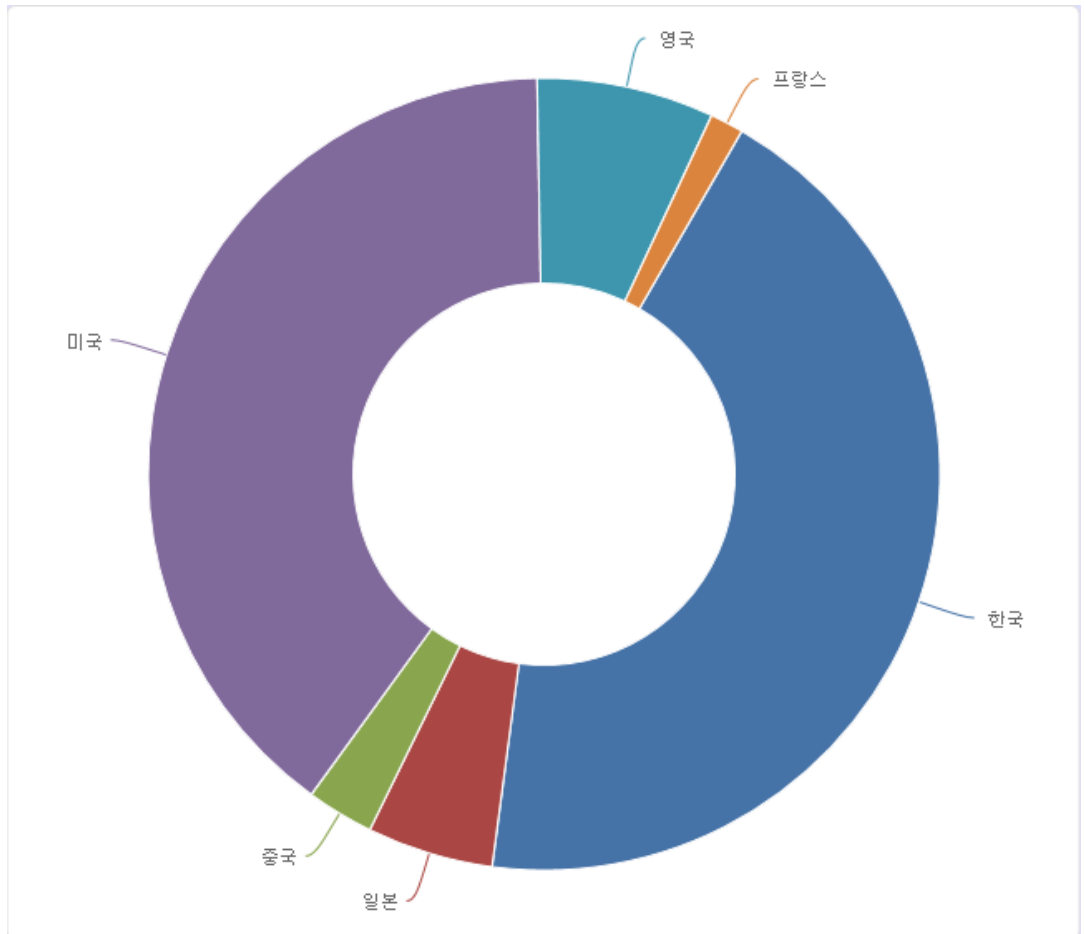
//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Pie:{ //컬럼 차트 속성
    InnerSize:"23",
    StartAngle:30
  }
});

//차트 Series 설정
myChart.SetSeriesOptions([
  {
    Type : "pie",
    Data : [
      ["한국", 636],
      ["일본", 75],
      ["중국", 40],
      ["미국", 577],
      ["영국", 105],
      ["프랑스", 20]
    ]
  }
])
```



```
}], 1);  
myChart.Draw();
```

➤ **Result**



4.3.7.8 박스플롯(Box Plot) 차트 설정

SetPlotOptions의 BoxPlot 속성에서 설정한다.

BoxPlot의 경우 다른 차트와 달리 포인트의 데이터가 5개이다.

예) { Low : 100, Q1 : 200, Median : 300, Q3 : 400, High : 500 }

➤ Data 속성

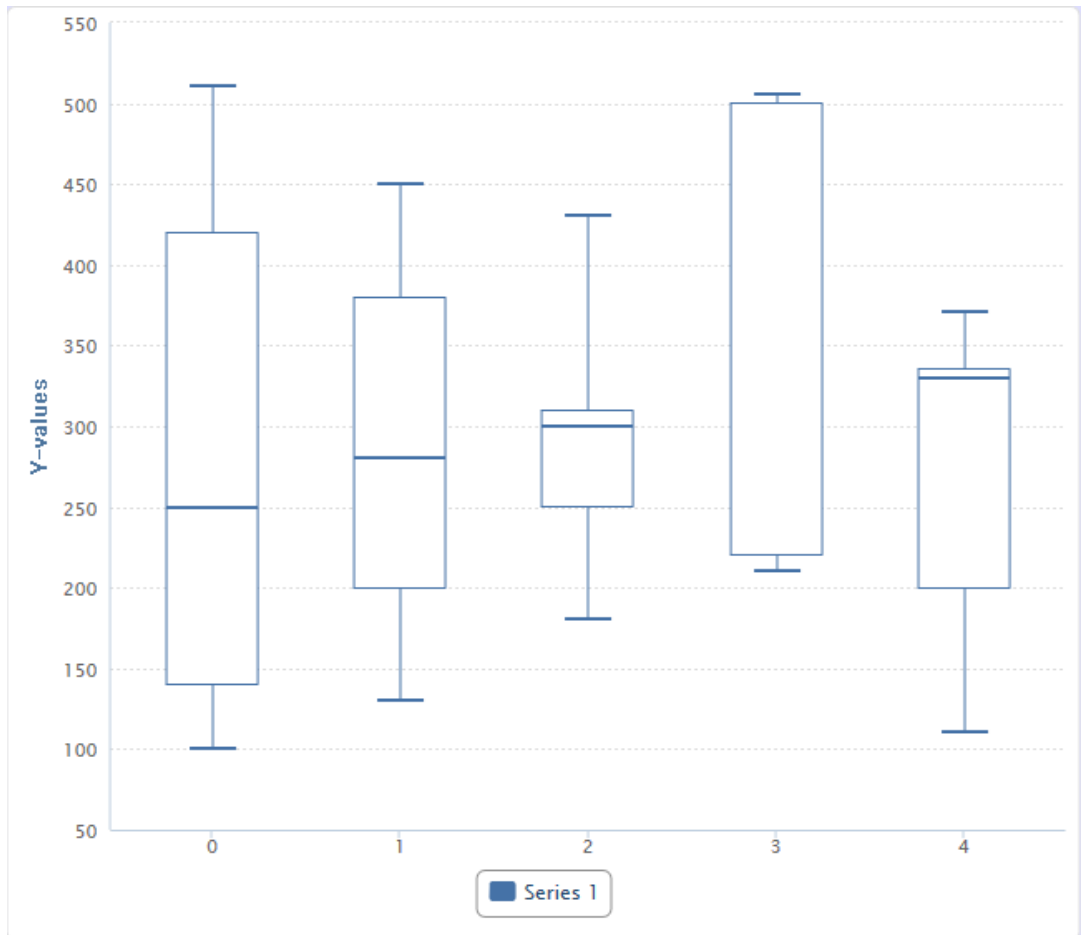
속성	타입	설명
Low	Number	최하위값
Q1	Number	중간박스의 하위값
Median	Number	중간값
Q3	Number	중간박스의 상위값
High	Number	최상위값

➤ Example

```
//차트 Series 설정
myChart.SetSeriesOptions([
    Type : "boxplot",
    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}
    ]
], 1);
myChart.Draw();
```



➤ Result



4.3.7.9 방사형(Polar) 차트 설정

Polar 차트는 기본 차트와는 달리, 차트 기본 속성 설정 부분에서 SetPolar()를 설정하여야 한다.

예) myChart.SetPolar(true), myChart.SetPolar(false)

Line, Spline, Areas, Column의 차트 형태를 모두 표현할수 있다.

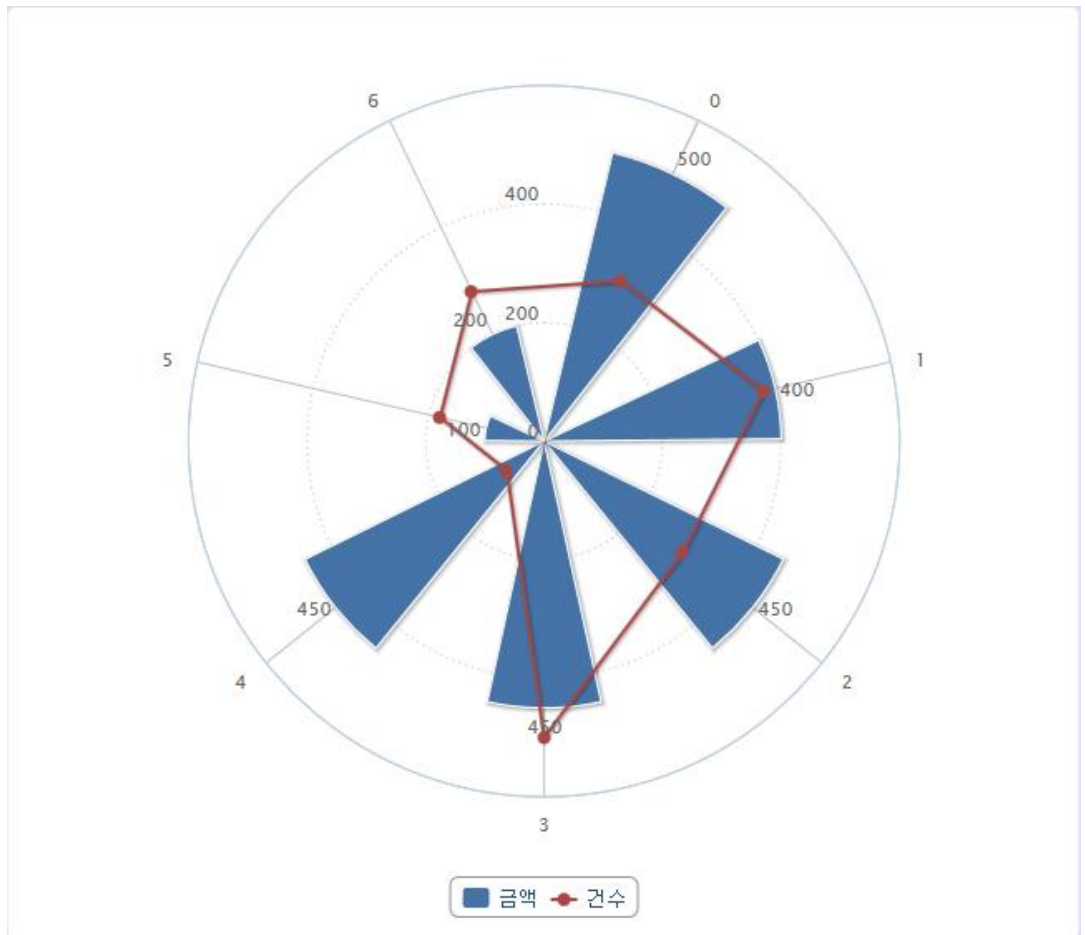
➤ Example

```
//차트 Polar 설정
myChart.SetPolar(true);

//차트 Series 설정
myChart.SetSeriesOptions([
    {
        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]
    }
], 1);

myChart.Draw();
```

➤ **Result**



4.4 데이터 조회

```
IBChart객체.DoSearch(PageUrl, [opt]);
```

Parameter	Type	필수여부	Remark
PageUrl	String	필수	데이터 조회 페이지 URL
opt.Param	Object	선택	조회 조건 QueryString (default:"")
opt.Append	Boolean	선택	조회시 Append 여부(default:false)
opt.Sync	Boolean	선택	조회시 통신 방식(default:false)

Param 인자는 조회 조건을 "조건명1=값1&조건명2=값2" 식으로 QueryString을 구성하여 설정한다.

```
var opt :{
    Param:{
        stockNo : 1234,
        stockNm : "한국"
    },
    Append:false,
    Sync:false
}
myChart.DoSearch("list.jsp", opt);
```

4.5 속성 변경사항 적용하기

IBChart는 차트의 옵션이 변경된 사항이 있을 경우 항상 Draw() 함수를 호출하여 화면에 반영해줘야 한다.

예제 1)

```
function doAction(sAction) {
    switch (sAction) {
        case ":Line":
            myChart.SetOptions({
                Chart:{
                    Type:"line"           //line 차트
                }
            });
            break;
        case "Column":
            myChart.SetOptions({
                Chart:{
                    Type:" column "      // column 차트
                }
            });
            break;
        case "Bar":
            myChart.SetOptions({
                Chart:{
                    Type:" bar "         // bar 차트
                }
            });
            break;
    }
    myChart.Draw(); //속성 재적용
}
```

4.6 데이터 XML 구현하기

조회 함수를 호출하면 인자로 넘긴 URL로 페이지에 연결한다. XML 형태로 구성된 페이지의 내용을 읽어 들어 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 개체 시작	{ IBCHART: {
차트 일반 속성들	BackgroundColor: "#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: "block", 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



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.

5. 개발 실습

샘플 데이터 XML 파일을 이용하여 차트를 생성한다.

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()">
  <script type="text/javascript">
    createIBChart("myChart", "500px", "300px");
  </script>
</body>
```

➤ 기본 속성 설정하기

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js" ></script>
<script type="text/javascript" src="/Chart/ibchart.js"></script>
<script type="text/javascript" src="/Chart/ibchartinfo.js"></script>
<script type="text/javascript">
function LoadPage() {
    myChart.Draw();
}
</script>
```

5.2 조회하기

조회 기능을 처리 하기 위해서는 다음의 3가지 과정을 처리한다.

- 조회 함수 생성하기(필수)
- 조회 XML 생성하기(필수)
- 조회 완료 이벤트 처리하기(선택)

➤ 조회 함수 생성하기(필수)

조회 처리할 조회 함수를 다음과 같이 자바스크립트로 생성한다.

```
function Search() { //조회조건
    var data = {
        Param:{
            sa_name:document.frm.sa_name.value,
            sal:document.frm.sal.value,
            hdate:document.frm.hdate.value
        }
    }
    myChart.DoSearch("test.xml", data);
}
```

조회 조건 인자는 여러 개 조회 조건을 조건명과 조건값을 "="로 연결하고, 여러 개 조건이 사용될 경우 "&"로 연결하여 다음과 같이 구성한다.

조건1=값1&조건2=값2&조건3=값3

➤ 조회 XML 생성하기(필수)

조회 함수에서 설정한 조회 XML 페이지를 구성한다.

다음은 test.xml를 구현한 샘플 페이지이다.



```
<?xml version='1.0'?>
<![CDATA[
  <IBCHART BORDERWIDTH="1" TITLE="국내 OS별 스마트폰 가입자 현황"
  SUBTITLE="(2011년 2월 기준)">
    <DATA>
      <POINTSET AXISLABEL="10년 3월">
        <SERIES SERIESNAME="안드로이드">190</SERIES>
        <SERIES SERIESTYPE="column"
SERIESNAME="iOS">50</SERIES>
        <SERIES SERIESNAME="윈도모바일">85</SERIES>
        <SERIES SERIESNAME="기타">11</SERIES>
      </POINTSET>
      <POINTSET AXISLABEL="10년 6월">
        <SERIES>145</SERIES>
        <SERIES>81</SERIES>
        <SERIES>102</SERIES>
        <SERIES>19</SERIES>
      </POINTSET>
      <POINTSET AXISLABEL="10년 9월">
        <SERIES>204</SERIES>
        <SERIES>115</SERIES>
        <SERIES>101</SERIES>
        <SERIES>21</SERIES>
      </POINTSET>
      <POINTSET AXISLABEL="10년 12월">
        <SERIES>415</SERIES>
        <SERIES>184</SERIES>
        <SERIES>102</SERIES>
        <SERIES>21</SERIES>
      </POINTSET>
      <POINTSET AXISLABEL="11년 1월">
        <SERIES>494</SERIES>
```



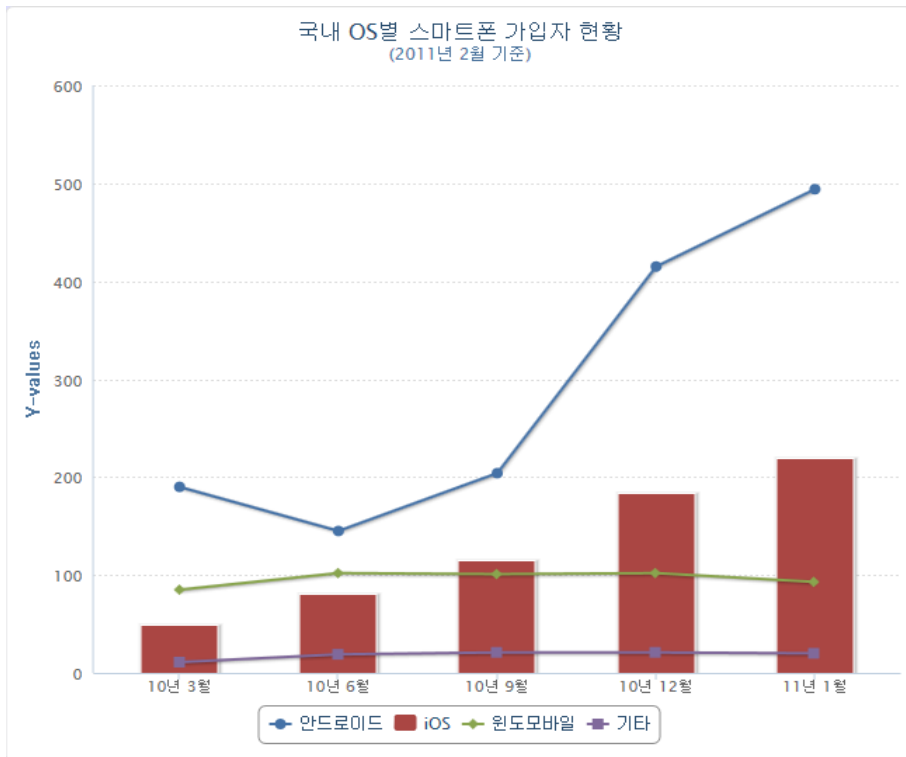
```
<SERIES>219</SERIES>
<SERIES>93</SERIES>
<SERIES>20</SERIES>
</POINTSET>
</DATA>
</IBCHART>
]]>
```

➤ **조회 완료 이벤트 처리하기(선택)**

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

```
function myChart_OnSearchEnd(Code, Msg) {
    alert("조회가 완료되었습니다.");
}
```

위와 같이 조회 기능을 완료하여 처리하면 다음과 같은 결과 화면이 생성된다.





(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.



Chapter 6. 데이터 구조



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.



6. 데이터 구조

IBChart에서 사용되는 데이터 구조를 설명한다.

IBChart는 XML과 JSON 두가지 형식의 자료구조를 사용한다.

각 자료구조의 세부적인 사양은 아래와 같다.

6.1 데이터 XML

조회 함수를 통해 읽어들이는 XML 전체 구조는 다음과 같다.

```
<?xml version='1.0'?>
<![CDATA[
<IBCHART TYPE="" BACKGROUND_COLOR='#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: "열 하나",
        PointColor: "#C6E2FF",
        Value: 120

    },
    {
        SeriesType: "spline",
        SeriesName: "KOREA",
        SeriesColor: "green",
        PointName: "스물 하나",
        PointColor: "#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



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.

7. IBChart 상수

7.1 Chart Type

속성	타입	설명
IBChartType.AREA	String	AREA 차트
IBChartType.AREA_RANGE	String	AREA_RANGE 차트
IBChartType.AREA_SPLINE	String	AREA_SPLINE 차트
IBChartType.BAR	String	BAR 차트
IBChartType.BOXPLOT	String	BOXPLOT 차트
IBChartType.BUBBLE	String	BUBBLE 차트
IBChartType.COLUMN	String	COLUMN 차트
IBChartType.COLUMN_RANGE	String	COLUMN_RANGE 차트
IBChartType.ERRORBAR	String	ERRORBAR 차트
IBChartType.FUNNEL	String	FUNNEL 차트
IBChartType.GAUGE	String	GAUGE 차트
IBChartType.HEATMAP	String	HEATMAP 차트
IBChartType.LINE	String	LINE 차트(기본값)
IBChartType.PIE	String	PIE 차트
IBChartType.PYRAMID	String	PYRAMID 차트
IBChartType.SCATTER	String	SCATTER 차트
IBChartType.SPLINE	String	SPLINE 차트
IBChartType.WATERFALL	String	WATERFALL 차트

7.2 Zoom Type

속성	타입	설명
IBZoomType.X	String	X축 줌
IBZoomType.Y	String	Y축 줌
IBZoomType.X_AND_Y	String	XY축 줌

7.3 가로 정렬

속성	타입	설명
IBAlign.LEFT	String	LEFT 정렬
IBAlign.CENTER	String	CENTER 정렬(기본값)
IBAlign.RIGHT	String	RIGHT 정렬

7.4 세로 정렬

속성	타입	설명
IBVerticalAlign.TOP	String	TOP 정렬
IBVerticalAlign.MIDDLE	String	MIDDLE 정렬
IBVerticalAlign.BOTTOM	String	BOTTOM 정렬(기본값)

7.6 Layout

속성	타입	설명
IBLayout.HORIZONTAL	String	가로로 정렬(기본값)
IBLayout.VERTICAL	String	세로로 정렬

7.7 Dash Style

속성	타입	설명
IBDashStyle.SOLID	String	SOLID
IBDashStyle.SHORT_DASH	String	SHORT_DASH
IBDashStyle.SHORT_DOT	String	SHORT_DOT
IBDashStyle.SHORT_DASH_DOT	String	SHORT_DASH_DOT
IBDashStyle.SHORT_DASH_DOT_DOT	String	SHORT_DASH_DOT_DOT
IBDashStyle.DOT	String	DOT
IBDashStyle.DASH	String	DASH
IBDashStyle.LONG_DASH	String	LONG_DASH
IBDashStyle.DASH_DOT	String	DASH_DOT
IBDashStyle.LONG_DASH_DOT	String	LONG_DASH_DOT

Solid

ShortDash

ShortDot

ShortDashDot

ShortDashDotDot

Dot

Dash

LongDash

DashDot

LongDashDot

LongDashDotDot

7.8 Symbol

속성	타입	설명
IBSymbol.CIRCLE	String	CIRCLE
IBSymbol.SQUARE	String	SQUARE
IBSymbol.DIAMOND	String	DIAMOND
IBSymbol.TRIANGLE	String	TRIANGLE
IBSymbol.TRIANGLE_DOWN	String	TRIANGLE_DOWN

7.9 이미지 저장 타입

속성	타입	설명
IBExportType.PNG	String	image/png(기본값)
IBExportType.JPEG	String	image/jpeg
IBExportType.PDF	String	application/pdf
IBExportType.SVG	String	image/svg+xml
IBExportType.EXCEL	String	application/xls

7.10 Stacking

속성	타입	설명
IBStacking.NORMAL	String	NORMAL(기본값)
IBStacking.PERCENT	String	PERCENT

7.11 Tick Position

속성	타입	설명
IBTickPosition.INSIDE	String	INSIDE(기본값)
IBTickPosition.OUTSIDE	String	OUTSIDE

7.12 Cursor Type

속성	타입	설명
IBCursorType.DEFAULT	String	DEFAULT(기본값)
IBCursorType.POINTER	String	POINTER
IBCursorType.WAIT	String	WAIT
IBCursorType.HELP	String	HELP
IBCursorType.MOVE	String	MOVE
IBCursorType.CROSSHAIR	String	CROSSHAIR

7.13 Axis Title Align

속성	타입	설명
IBAxisTitleAlign.LOW	String	LOW
IBAxisTitleAlign.MIDDLE	String	MIDDLE(기본값)
IBAxisTitleAlign.HIGH	String	HIGH





Chapter 8. IBChart Config



8. IBChart Config

8.1 Config 설정

모든 IBChart에 전역으로 설정 하고자 할 때 ibchart.cfg 파일에 해당 속성을 정의 한다. 파일의 포맷은 xml 형태 이며 Cfg의 Attribute를 통해 설정한다.

```
<CHART>
  <Cfg
    [속성명]=[속성값]
  />
</CHART>
```

예를 들어 SampleProp 속성을 SampleValue 값으로 설정 한다면 아래와 같이 설정 한다.

```
<CHART>
  <Cfg
    Samprop="SampleValue"
  />
</CHART>
```

전역 설정 할 수 있는 속성은 다음과 같다.



8.2 Property List

속성명	설명
LicenseFileName	라이선스 파일명 설정 (Default="license")



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.



Chapter 9. IBChart Events



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.

9. IBChart Events

9.1 Event 사용

IBChart에서 사용자가 원하는 기능을 추가 또는 변경 처리할 수 있도록 이벤트를 제공한다.

```
function 오브젝트ID_이벤트명(파라미터, ...) { }
```

예를 들어 IBChart ObjectID를 "myChart"라 하고, OnLoad 이벤트에 대해 기능 처리를 해야 한다면 다음과 같이 기술할 수 있다.

```
function myChart_OnLoad() {  
    alert("차트가 로드되었습니다.");  
}
```

IBChart가 제공하는 이벤트는 다음과 같다.



9.2 Event List



OnClick

➤ 기능

차트 영역을 클릭 하는 시점에 이 이벤트가 발생한다.

➤ Syntax

Syntax	function 오브젝트ID_OnClick() { }
--------	--------------------------------------

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

<pre>function myChart_OnClick() { alert("차트를 클릭하였습니다."); }</pre>
--

➤ 제공 버전

7.1.0.4	
----------------	--

OnLegendItemClick

➤ 기능

차트의 범례 내의 시리즈를 클릭 하는 시점에 이 이벤트가 발생한다.

➤ Syntax

Syntax	function 오브젝트ID_OnLegendItemClick(Index, Name) { }
--------	---

➤ Parameters

Parameter	Type	설 명
Index	Number	선택한 시리즈의 Index
Name	String	선택한 시리즈의 Name

➤ Example

<pre>function myChart_OnLegendItemClick(Index, Name) { alert(Index + "번째의 시리즈인 " + Name + "이 선택되었습니다."); }</pre>
--

➤ 제공 버전

7.0.0.0



OnLoad

➤ 기능

차트 컨트롤이 로드된 후에 발생한다.

➤ Syntax

Syntax	function 오브젝트ID_OnLoad() { }
--------	-------------------------------------

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

<pre>function myChart_OnLoad() { alert("차트가 로드되었습니다."); }</pre>

➤ 제공 버전

7.0.0.0

OnPointClick

➤ 기능

포인트를 마우스 클릭할 때 발생한다.

포인트는 라인차트의 경우 라인위에 값이 표시된 지점, 막대차트의 경우 막대 하나를 포인트라 지칭한다.

➤ Syntax

Syntax	function 오브젝트ID_OnPointClick(Index, X, Y) {}
--------	---

➤ Parameters

Parameter	Type	설 명
Index	Number	선택한 포인트의 시리즈 Index
X	Number	선택한 포인트의 X 번째 Index
Y	Folat	선택한 포인트의 Y

➤ Example

```
function myChart_OnPointClick(Index, X, Y) {
    alert(Index + "번째 그래프의 " + X + "번 값은 " + Y + "입니다.");
}
```

➤ 제공 버전

7.0.0.0



OnRedraw

➤ 기능

차트가 리사이즈 되어 다시 랜더링 완료 후 발생한다.

➤ Syntax

Syntax	function 오브젝트ID_OnRedraw() {}
--------	--------------------------------------

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

<pre>function myChart_OnRedraw(){ // 비즈니스 로직 구현 }</pre>

➤ 제공 버전

7.0.0.0

OnSeriesHide

➤ 기능

차트의 시리즈를 숨길 때 발생한다.

차트 레전드의 시리즈명을 클릭 할 경우 해당시리즈가 숨겨지는데 이때 발생하는 이벤트이다.

➤ Syntax

Syntax	function 오브젝트ID_OnSeriesHide(Index) { }
--------	--

➤ Parameters

Parameter	Type	설 명
Index	Number	숨긴 시리즈 Index

➤ Example

```
function myChart_OnSeriesHide(Index){
    alert( Index +"번째 그래프를 숨깁니다.");
}
```

➤ 제공 버전

7.0.0.0

OnSeriesMouseOut

➤ 기능

차트의 시리즈에서 마우스가 빠지는 시점에 이 이벤트가 발생한다.
차트내의 같은 시리즈 간의 이동은 시리즈 선상에서의 이동으로 보기 때문에 해당 이벤트가 발생하지 않고 다른시리즈 또는 차트 영역을 벗어나는 경우에 발생 한다. 즉, 해당 시리즈의 툴팁이 사라지는 시점과 이 이벤트가 발생하는 시점은 동일 하다.

➤ Syntax

Syntax	function 오브젝트ID_OnSeriesMouseOut() { }
--------	---

➤ Parameters

Parameter	Type	설 명
		없음

➤ Example

<pre>function myChart_OnSeriesMouseOut(){ // 비즈니스 로직 처리 }</pre>

➤ 제공 버전

7.0.0.0

OnSeriesMouseOver

➤ 기능

차트의 시리즈에서 마우스가 진입하는 시점에 이 이벤트가 발생한다.
이 이벤트는 해당 시리즈의 툴팁이 보이는 시점과 동일 하다.

➤ Syntax

Syntax	function 오브젝트ID_OnSeriesMouseOver() { }
--------	--

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

<pre>function myChart_OnSeriesMouseOver(){ // 비즈니스 로직 처리 }</pre>
--

➤ 제공 버전

7.0.0.0

OnSeriesShow

➤ 기능

차트의 시리즈를 숨김 해제할 때 발생한다.

레전드를 클릭하여 시리즈를 숨겼을 때, 다시 클릭하면 숨김 해제가 되는데 이때 발생하는 이벤트이다.

➤ Syntax

Syntax	function 오브젝트ID_OnSeriesShow(Index) {}
--------	---

➤ Parameters

Parameter	Type	설 명
Index	Number	숨김을 해제한 시리즈 Index

➤ Example

```
function myChart_OnSeriesShow(Index){
    alert( Index +"번째 그래프를 숨김을 해제합니다.");
}
```

➤ 제공 버전

7.0.0.0

OnSearchEnd

➤ 기능

조회가 완료되었을 때 발생한다.

➤ Syntax

Syntax	function 오브젝트ID_OnSearchEnd(Code, Msg) {}
--------	--

➤ Parameters

Parameter	Type	설 명
Code	String	조회 결과 코드
Msg	String	코드에 대한 메시지

➤ Example

<pre>function myChart_OnSearchEnd(Code, Msg){ alert("조회가 완료되었습니다."); }</pre>

➤ 제공 버전

7.0.0.0

OnZoomIn

➤ 기능

ZoomType 을 설정한 경우 마우스 액션으로 차트의 일부분을 확대하는 시점에 발생 한다.

➤ Syntax

Syntax	function 오브젝트ID_OnZoomIn() {}
--------	--------------------------------------

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

```
function myChart_OnZoomIn(){
    alert("ZoomIn");
}
```

➤ 제공 버전

7.0.0.0

OnZoomOut

➤ 기능

차트의 일부분이 확대되어 있는 경우 확대 초기화 버튼을 클릭하는 시점에 발생 한다.

➤ Syntax

Syntax	function 오브젝트ID_OnZoomOut() {}
--------	---------------------------------------

➤ Parameters

Parameter	Type	설 명
없음		

➤ Example

```
function myChart_OnZoomOut(){
    alert("ZoomOut");
}
```

➤ 제공 버전

7.0.0.0



Chapter 10. IBChart Methods



(주) 아이비 리더스

Inno Biz Leaders Co., Ltd.



10. IBChart Methods

10.1 Chart Object

AddSeries

➤ **기능**

시리즈를 차트에 추가한다.

➤ **Syntax**

Syntax	ChartObj. AddSeries (Series)
--------	-------------------------------------

➤ **Info**

Parameter	Type	필수여부	설 명
Series	Object	필수	시리즈 객체

➤ **Example**

```
var series = myChart.CreateSeries();  
myChart.AddSeries(series);
```

➤ **제공 버전**

7.0.0.0



CreateSeries

➤ 기능

시리즈 객체를 생성한다.

➤ Syntax

Syntax	ChartObj. CreateSeries()
--------	---------------------------------

➤ Info

Return	Object, Series 객체		
Parameter	Type	필수여부	설 명
없음			

➤ Example

```
var series = myChart.CreateSeries();  
myChart.AddSeries(series);
```

➤ 제공 버전

7.0.0.0

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	ChartObj. DoPrint()
--------	----------------------------

➤ Info

Return	없음		
Parameter	Type	필수여부	설 명
없음			

➤ Example

```
// 차트 인쇄하기
myChart.DoPrint();
```

➤ 제공 버전

7.0.0.0

Down2Image

➤ 기능

차트 데이터를 이미지로 다운로드한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
FileName	String	이미지 파일명
Type	String	다운로드 형식 (IBExportType 참고)
Width	String	이미지 너비
Url	String	Down2Image.jsp 경로

Type 인자는 IBExportType 상수를 이용하여 설정한다. jpg, png, pdf, svg 형식으로 다운로드할 수 있다. Url 인자는 IBChart에서 제공하는 ibchartimage.jsp 파일의 경로를 설정한다. 해당 경로가 올바르지 않으면 이미지를 다운로드 할 수 없으니 주의한다.

➤ Syntax

Syntax	ChartObj. Down2Image (Param)
--------	-------------------------------------

➤ Info

Parameter	Type	필수여부	설 명
Param	Object	필수	해당 속성 Json Object

➤ Example

```
// 차트 이미지 다운로드
myChart.Down2Image({FileName:"ChartImage", Type:IBExportType.JPEG,
Width:800, Url:"../Chart/ibchartimage.jsp"});
```

➤ 제공 버전

7.0.0.0

Draw

➤ 기능

차트의 최신 상태를 반영하기 위해 차트를 다시 그린다.

차트 옵션이 변경됐을 경우 항상 마지막에 다시 호출해줘야 한다.

➤ Syntax

Syntax	ChartObj. Draw ([CallBack])
--------	------------------------------------

➤ Info

Return	없음		
Parameter	Type	필수여부	설 명
CallBack	Function	선택	Draw 완료시점에 호출할 CallBack 함수
없음			

➤ Example

<pre>// 차트 그리기 myChart.Draw(); // 차트 랜더링 이후 callback 함수 호출 function fnCallBack() { // Draw 완료후 처리할 비즈니스 로직 } myChart.Draw(fnCallBack);</pre>
--

➤ 제공 버전

7.0.0.0

DoSearch

➤ 기능

데이터를 조회한다. XML 또는 JSON 형식으로 조회가 가능하다.

➤ Syntax

Syntax	ChartObj. DoSearch (Url, [opt])
--------	--

➤ Info

Parameter	Type	필수여부	설 명
Url	String	필수	데이터 Json의 URL
opt.Param	Object	선택	조회 조건 QueryString (default:"")
opt.Append	Boolean	선택	조회시 Append 여부(default:false)
opt.Sync	Boolean	선택	조회시 통신 방식(default:false)

➤ Example

```
// 데이터 Json을 조회한다
var data = {
  Param:{
    id:"khlee",
    seq:1
  }
}
myChart.DoSearch("list.jsp", data);
```

➤ 제공 버전

7.0.0.0

GetEtcData

➤ 기능

Xml, JSON 조회시 Etc-Data 에 설정된 내용을 Key 값을 기준으로 해당 Value 값을 가져온다.

➤ Syntax

Syntax	ChartObj. GetEtcData (Key)
--------	-----------------------------------

➤ Info

Return	String, 해당 Value		
Parameter	Type	필수여부	설 명
Key	String	필수	Key 값

➤ Example

```
// ETC DATA 가져오기
var name = myChart.GetEtcData('NAME');
```

➤ 제공 버전

7.0.0.0

GetNumberFormat

➤ 기능

해당 숫자값에 포맷이 적용된 문자열을 가져온다.

➤ Syntax

Syntax	ChartObj. GetNumberFormat (Number, Decimals, [DecimalPoint], [ThousandsSep])
--------	---

➤ Info

Return	String, NumberFormat이 적용된 문자열		
Parameter	Type	필수여부	설 명
Number	Number	필수	숫자
Decimals	Number	필수	소수점 자리수
DecimalPoint	String	선택	소수점 기호, 기본값(.)
ThousandsSep	String	선택	천단위 기호, 기본값(,)

➤ Example

```
function TooltipFormat(){
    return myChart.GetNumberFormat(this.value, 0);
}
```

➤ 제공 버전

7.0.0.0

GetSearchData

➤ 기능

조회 페이지를 호출하여 조회 처리를 완료한 후 생성된 조회결과 데이터를 문자열로 반환한다. 반환된 조회데이터를 LoadSearchData 함수의 인자로 사용하면 결과를 IBChart의 표현 할 수 있다.

➤ Syntax

Syntax	ChartObj. GetSearchData (PageUrl, [Param])
--------	---

➤ Info

Return	String, NumberFormat이 적용된 문자열		
Parameter	Type	필수여부	설 명
PageUrl	String	필수	조회 처리할 페이지 URL
Param	String	선택	조회 조건 Query String, Default=''

➤ Example

```
//조회 데이터 읽어오기
var sXml = myChart.GetSearchData("list.jsp");

//조회 결과 내용을 표현하기
myChart.LoadSearchData(sXml);
```

➤ 제공 버전

7.0.0.0



GetSeries

➤ 기능

특정 시리즈 객체를 가져온다.

➤ Syntax

Syntax	ChartObj. GetSeries (Index)
--------	------------------------------------

➤ Info

Return	Object, 시리즈 객체		
Parameter	Type	필수여부	설 명
Index	Number	필수	시리즈 Index

➤ Example

// series object var series = myChart.GetSeries(1);
--

➤ 제공 버전

7.0.0.0

GetSeriesCount

➤ 기능

현재 차트에 표시된 시리즈 개수를 확인한다.

➤ Syntax

Syntax	ChartObj. GetSeriesCount ()
--------	------------------------------------

➤ Info

Return	Number, 시리즈 개수		
Parameter	Type	필수여부	설 명
없음			

➤ Example

```
// series 개수
var count = myChart.GetSeriesCount();
```

➤ 제공 버전

7.0.0.0

GetXAxis

➤ 기능

이미 생성된 X 축 객체를 가져오거나 새로운 X 축 객체를 생성한다.

➤ Syntax

Syntax	ChartObj. GetXAxis (Index)
--------	-----------------------------------

➤ Info

Return	Object, XAxis Object		
Parameter	Type	필수여부	설 명
Index	Number	필수	축의 Index

➤ Example

```
// 축 Index 설정 및 축 가져오기
var axis = myChart.GetXAxis(0);
```

➤ 제공 버전

7.0.0.0

GetYAxis

➤ 기능

이미 생성된 Y 축 객체를 가져오거나 새로운 Y 축 객체를 생성한다.

Syntax

Syntax	ChartObj. GetYAxis (Index)
--------	-----------------------------------

➤ Info

Return	Object, YAxis Object		
Parameter	Type	필수여부	설 명
Index	Number	필수	축의 Index

➤ Example

```
// 축 Index 설정 및 축 가져오기
var axis = myChart.GetYAxis(0);
```

➤ 제공 버전

7.0.0.0

LoadSearchData

➤ 기능

조회데이터(xml or json)을 함수의 인자로 전달 받아 IBChart에 로딩한다.

Syntax

Syntax	ChartObj. LoadSearchData (Content, [Opt])
--------	--

➤ Info

Return			
Parameter	Type	필수여부	설 명
Content	String	필수	조회XML 또는 조회 JSON 문자열
Opt.Append	Boolean	선택	Append 조회 여부, Default=false

➤ Example

```
//조회 데이터 읽어오기
var sXml = myChart.GetSearchData("list.jsp");

//조회 결과 내용을 표현하기
myChart.LoadSearchData(sXml);
```

➤ 제공 버전

7.0.0.0

RemoveAll

➤ 기능

현재 차트에 표시된 모든 시리즈를 제거한다.

➤ Syntax

Syntax	ChartObj. RemoveAll ()
--------	-------------------------------

➤ Info

Return	Boolean, true(성공) false(실패)		
Parameter	Type	필수여부	설 명
없음			

➤ Example

// 모든 시리즈 제거 var flag = myChart.RemoveAll();

➤ 제공 버전

7.0.0.0

RemoveSeries

➤ 기능

특정 시리즈를 제거한다.

➤ Syntax

Syntax	ChartObj. RemoveSeries (Index, [Draw])
--------	---

➤ Info

Return	Boolean, true(성공) false(실패)		
Parameter	Type	필수여부	설 명
Index	Number	필수	시리즈 Index
Draw	Boolean	선택	Draw 여부

➤ Example

// 첫번째 시리즈 제거 var flag = myChart.RemoveSeries(1);
--

➤ 제공 버전

7.0.0.0

SetBaseOptions

➤ 기능

차트의 기본 속성 설정

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Type	String	차트 유형 선택
Polar	Boolean	Polar 차트 설정 여부
Height	Number	차트 높이 설정
Width	Number	차트 넓이 설정
Inverted	Boolean	x축과 y축을 바꿈
BorderColor	String	차트 테두리 색상 설정
BorderWidth	Number	차트 테두리 선 넓이 설정
BorderRadius	Number	차트 테두리 Radius 설정
Margin	Array	차트 영역 주변 여백설정
ZoomType	String	x축 ,y축에 대한 확대기능 사용 유무
BackgroundColor	String	차트 영역 전체의 배경색
PlotBackgroundColor	String	차트 표현 영역의 배경색
PlotBackgroundImage	String	차트 표현 영역에 보여질 이미지 URL
Options3d	Object	3D 차트 표현 설정 객체

➤ Syntax

Syntax	ChartObj. SetBaseOptions (opt, [merge])
--------	--

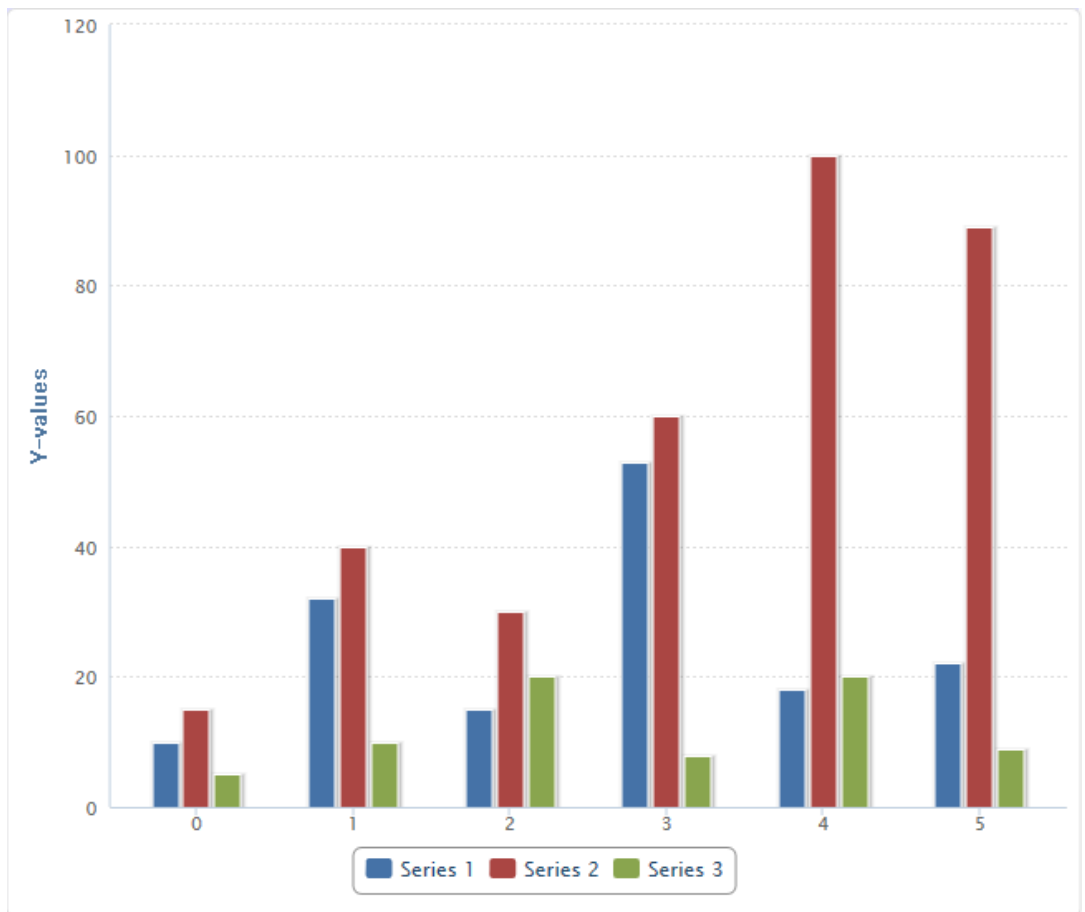
➤ Info

Parameter	Type	필수여부	설 명
opt	Object	필수	속성설정 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)



➤ **Example**

```
myChart.SetBaseOptions({
    Type:'column',
    Height : 550,
    Width : 650,
    margin: [50, 0, 0, 0], //top,right,bottom,left
    ZoomType:'x', // 'x' , 'y' or 'xy'
    PlotBackgroundColor:"#EDEDED"
},0);
myChart.SetSeriesOptions([
    Data : [ {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22} ]
},
    {
        Data : [ {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89} ]
    },
    {
        Data : [ {Y:5},{Y:10},{Y:20},{Y:8},{Y:20},{Y:9} ]
    }
], 1);
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

SetColors

➤ 기능

차트 컨트롤의 시리즈 색상 값을 설정한다.
설정된 순서에 따라 시리즈 색상이 적용된다.

➤ Syntax

Syntax	ChartObj. SetColors (Colors)
--------	-------------------------------------

➤ Info

Parameter	Type	필수여부	설 명
Colors	Array	필수	색상 배열값

➤ Example

```
// 기본 색상값 설정
var arr = new Array("#FF0000", "#00FF00");
myChart.SetColors(arr);
```

➤ 제공 버전

7.0.0.0

SetCreditsOptions

➤ 기능

차트의 Credits 속성을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	Credits 사용여부, 기본값 true
Text	String	Credits Text
Href	String	Credits URL
Position	Object	Credits 위치 설정

➤ Syntax

Syntax	ChartObj. SetCreditsOptions (Credits, [merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
Credits	Object	필수	Credits 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

```
// 차트에 Credits 설정
myChart.SetCreditsOptions({
  Enabled:true,
  Href:"http://www.ibleaders.co.kr",
  Position : {
    align: 'center',
    x: 0,
    verticalAlign: 'bottom',
    y: -25
  }
})
```



```
}  
});
```

Mychart

Hcharts.com

- 제공 버전
7.0.0.0

SetLabelsOptions

➤ 기능

차트 컨트롤의 영역에 HTML 레이블을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Items	Object	Label 설정(Json Object)
Style	Object	Label 스타일 설정(CSS Object)

Items는 Json 배열의 형태로 설정하며 속성은 아래와 같다.

Parameter	Type	설 명
Html	Text	Label 텍스트 설정
Style	Object	Label 스타일 설정(CSS Object)

➤ Syntax

Syntax	ChartObj. SetLabelsOptions (LabelItem, [merge]);
--------	---

➤ Info

Parameter	Type	필수여부	설 명
LabelItem	Object	필수	LabelItem 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

<pre>myChart.SetLabelsOptions({ Items:[{ Html:"IBChart Labels", Style:{ Left:"100px",</pre>



```
        Top : "100px",
        Color: "red"
    }
}, {
    Html: "Test Sample",
    Style: {
        Left: "100px",
        Top : "120px",
        Color: "blue"
    }
}],
Style: {
    FontSize: "40px"
}
});
```

➤ 제공 버전

7.0.0.0

SetLegendOptions

➤ 기능

범례를 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Title	Object	범례 타이틀
Enabled	Boolean	사용여부, 기본값 true (Pie타입은 false)
Layout	String	레이아웃 (horizontal, vertical)
Align	String	가로 정렬 (left, center, right)
VerticalAlign	String	세로 정렬 (top, middle, bottom)
Margin	Number	여백
X	Number	X 위치
Y	Number	Y 위치
Floating	Boolean	플로팅 여부(default:false)
BackgroundColor	String	배경 색상
BorderColor	String	외곽선 색상
BorderWidth	Number	외곽선 두께
BorderRadius	Number	외곽선 모서리 부분의 둥근 정도의 크기
Reversed	Boolean	항목을 역순으로 표시함

타이틀은 JSON 객체로 내부적으로 Text 와 Style 속성을 갖는다.

Parameter	Type	설 명
Text	String	범례 타이틀 텍스트
Style	Object	범례 타이틀에 적용할 css. Color나 FontWeight 등이 사용 가능하다.

BackgroundColor 와 BorderColor 인자는 WebColor 값으로 설정한다.

➤ Syntax

Syntax	ChartObj. SetLegendOptions (Legend, [Merge])
--------	---

➤ **Info**

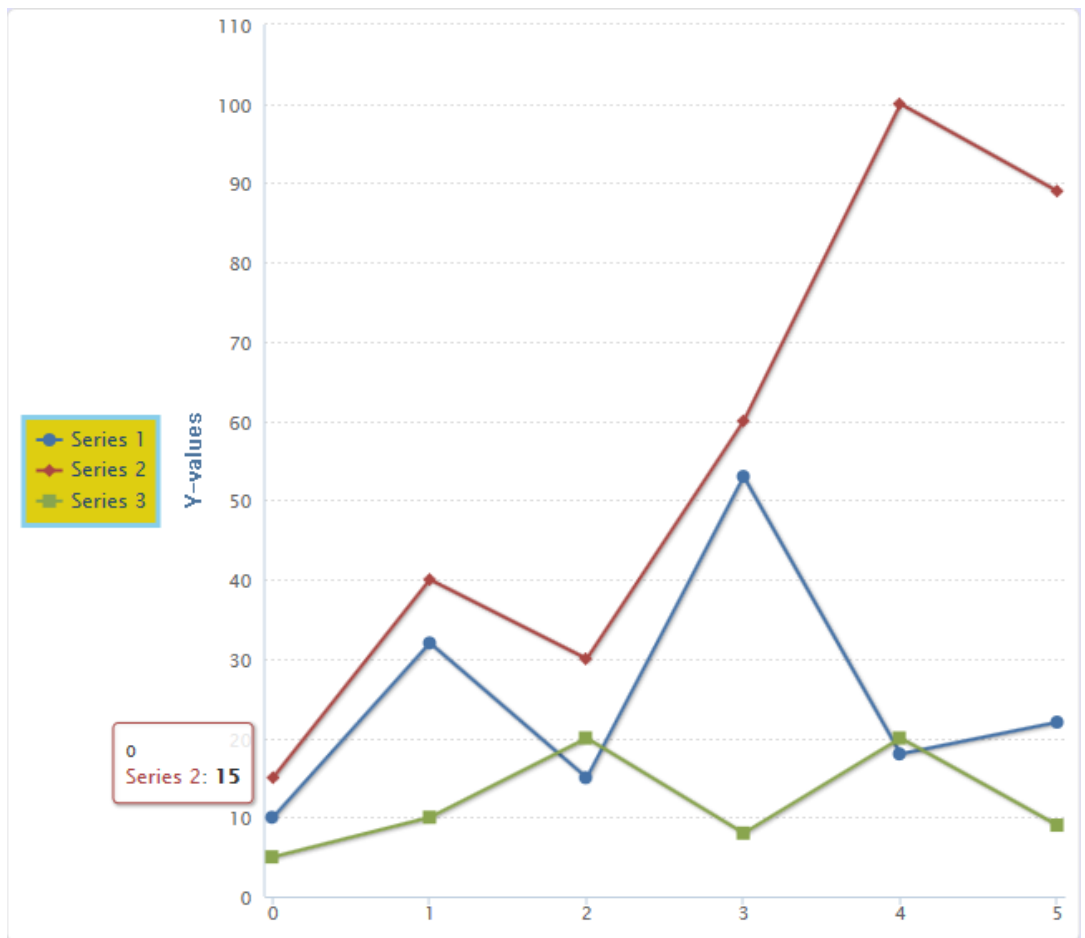
Parameter	Type	필수여부	설 명
Legend	Object	필수	Legend 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부(default:0)

➤ **Example**

```
//범례설정
myChart.SetLegendOptions({
    Enabled:true,
    Layout:"vertical",
    Align:"left",
    VerticalAlign:"middle",
    BorderColor:"skyblue",
    BorderWidth:3,
    BorderRadius:0,
    BackgroundColor:"#DECE11"
});

myChart.SetSeriesOptions([
    {
        Data : [ {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22} ]
    },{
        Data : [ {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89} ]
    },{
        Data : [ {Y:5},{Y:10},{Y:20},{Y:8},{Y:20},{Y:9} ]
    }
], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

SetNumericSymbols

➤ 기능

축 레이블의 숫자에 붙는 단위에 대하여 재설정한다.

X 축과 Y 축에서 1000이 넘을 경우 1k 1000000 이 넘을 경우 1M 으로 표시되는데 이러한 단위를 k, M, G, T, P 순서로 표시된다.

이 단위를 없애거나 다른 문자열로 표기하고 싶을때에 재정의한다.

➤ Syntax

Syntax	ChartObj.SetNumericSymbols(Symbols)
--------	-------------------------------------

➤ Info

Parameter	Type	필수여부	설 명
Symbols	Array	필수	천단위 문자의 Array

➤ Example

```
//천단위문자를 재정의함
myChart.SetNumericSymbols(["K","M","G","T"]); // 소문자 k 대신 대문자 K 로
재설정함.

myChart.SetNumericSymbols(["", "", "", "", "", ""]); // 단위 문자를 모두 없앴. (1
과 1000과 1000000 이 모두 1 로 표시됨)
```

➤ 제공 버전

7.0.0.0

SetPaneOptions

➤ 기능

게이지 차트의 여러가지 원판들을 표현한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

속성	Type	필수여부	설 명
Background	Object	선택	게이지 차트 배경 설정 Json Object
EndAngle	Number	선택	원의 종료 지점
StartAngle	Number	선택	원의 시작 지점

Background 의 속성에서 설정 가능한 속성은 다음과 같다.

속성	Type	필수여부	설 명
InnerRadius	String	선택	원판의 안쪽 반지름 %로 표현 100%인 경우 게이지차트의 크기 에 해당되는 지점이 원판의 안쪽 지점이 된다.
OuterRadius	String	선택	원판의 바깥쪽 반지름 %로 표현 100%인 경우 게이지차트의 크기 에 해당되는 지점이 원판의 바깥 쪽 지점이 된다.
BorderWidth	Number	선택	원판의 외곽선 두께
BorderColor	String	선택	원판의 외곽선 색상
BackgroundColor	String	선택	원판의 배경색 색상

해당 메소드는 게이지 차트의 원판들을 설정하는 기능이므로 InnerRadius 와 OuterRaidus 를 반드시 설정해야 한다.

여러 개의 원판을 표현해야 하므로 예제처럼 배열 형태로 입력해야 한다.

➤ Syntax

Syntax	ChartObj. SetPaneOptions (Panes, [merge])
--------	--

➤ Info

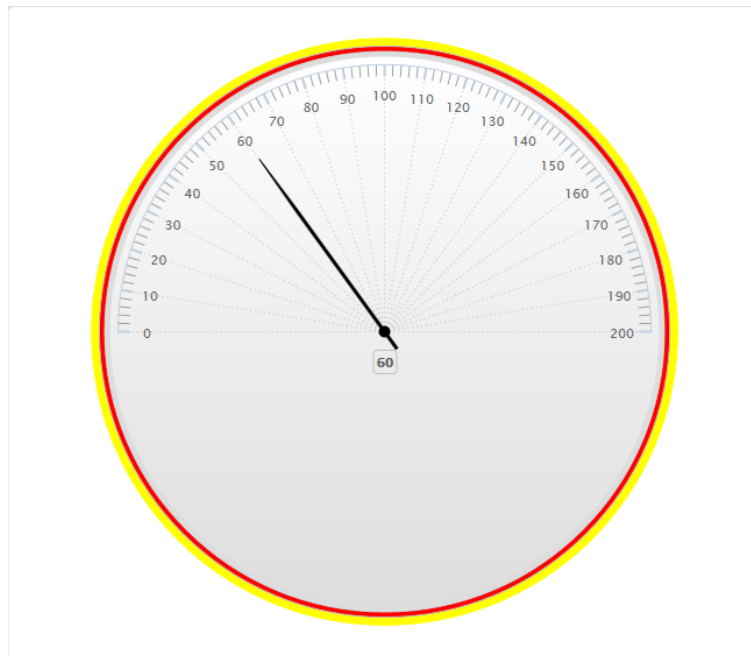
Parameter	Type	필수여부	설 명
Panes	Object	필수	JSON Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

```
// 게이지 차트에서 두개의 서로 다른 크기의 원판을 그린다.
myChart.SetPaneOptions({
    StartAngle: -90,
    EndAngle: 90,
    Background: [{
        BackgroundColor:"yellow",
        BorderWidth: 0,
        OuterRadius: '110%'
    }, {
        BackgroundColor:"red",
        BorderWidth: 1,
        OuterRadius: '107%'
    }, {
        // default background
    }, {
        BackgroundColor: '#DDD',
        BorderWidth: 0,
        OuterRadius: '105%',
        InnerRadius: '103%'
    }
}]
```

```

});
myChart.SetBaseOptions({
    Type:"gauge"
});
myChart.SetYAxisOptions({
    Max:200,
    Min:0,
    TickInterval : 10 // Tick 간격을 10 으로 설정
}, 1);
myChart.SetSeriesOptions([
    Data : [
        {Y:60}
    ]
}], 1);
myChart.Draw();
  
```



➤ 제공 버전
7.0.0.0

SetPlotOptions

➤ 기능

차트의 유형별 세부내용을 설정 한다.

차트 유형별 설정할수 있는 속성은 11.1 장의 [PlotOptions](#) 을 참고.

➤ Syntax

Syntax	ChartObj. SetPlotOptions (PlotOptions,[merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
PlotOptions	Object	필수	차트의 유형별 설정값. Series,Pie,Line,Bar,Column등 유형에 따라 기본적인 설정을 정의할 수 있다.
merge	Boolean	선택	이전 설정과의 병합 처리 여 부(default:0)

➤ Example

```
myChart.SetPlotOptions({
    //각 시리즈에 대한 기본 설정
    Series : {
        Shadow : false,
        //각 포인트 상단에 데이터레이블을 표시한다.
        DataLabels : {
            Enabled : true,
            Align : "center"
        }
    },
    //막대 그래프에 대한 기본 디자인 설정
    Column : {
        PointPadding : 0.02
    }
})
```




```
    },  
    //꺾은선 그래프에 대한 기본 디자인 설정  
    Line:{  
        //계단형 그래프를 만듦  
        Step:true,  
        //선의 굵기  
        LineWidth:2px  
    }  
},0);
```

➤ 제공 버전

7.0.0.0

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.

➤ Syntax

Syntax	ChartObj. SetOptions (Options, [merge])
--------	--

➤ Info

Parameter	Type	필수여부	설 명
Options	Object	필수	차트 영역의 속성값.
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

```
myChart.SetOptions({
  Chart : {
    BackgroundColor : "#EDEDED",
    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

7.0.0.0

SetSeriesOptions

➤ 기능

차트 시리즈의 속성을 설정한다.

시리즈의 속성값은 Json 형태로 설정하며 속성값은 다음과 같다.

Parameter	Type	설 명
Data	Object	Series의 데이터 설정 Json Object
Index	Number	Series의 Index 설정
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

시리즈를 설정할때에는 Data 영역에는 Json Object 형태로 몇가지 설정을 더 할 수 있는데, Data의 속성값은 다음과 같다.

Parameter	Type	설 명
Color	String	시리즈 포인트의 색상 설정
DataLabels	Object	시리즈 포인트의 레이블 설정 Json Object
Id	String	포인트의 Id 설정
Name	String	포인트 Name 설정
Stack	String	Stack옵션 설정
X	Number	포인트의 X 값
Y	Number	포인트의 Y 값

DataLabels도 JsonObject 형태로 설정할수 있는데, 사용되는 속성값은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	데이터 레이블 설정
Align	String	데이터 레이블 정렬 설정
VerticalAlign	String	데이터 레이블 수직 정렬 설정
Crop	Boolean	데이터 레이블이 차트 영역을 넘은경우 제거여부
BackgroundColor	String	데이터 레이블 배경색상 설정
BorderColor	String	데이터 레이블의 테두리 색상 설정
BorderRadius	Number	데이터 레이블의 테두리 둥글기 설정
BorderWidth	Number	데이터 레이블의 테두리 굵기 설정
Color	String	데이터 레이블의 텍스트 색상 설정
Formatter	Function	데이터 레이블의 형식 설정
Style	Object	데이터 레이블의 스타일을 지정한다. CSS Object ex) { color:'#3E576F' }

➤ **Syntax**

Syntax	ChartObj. SetSeriesOptions (SeriesOptions, [merge])
--------	--

➤ **Info**

Parameter	Type	필수여부	설 명
SeriesOptions	Object	필수	Series Options Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부(default:0)

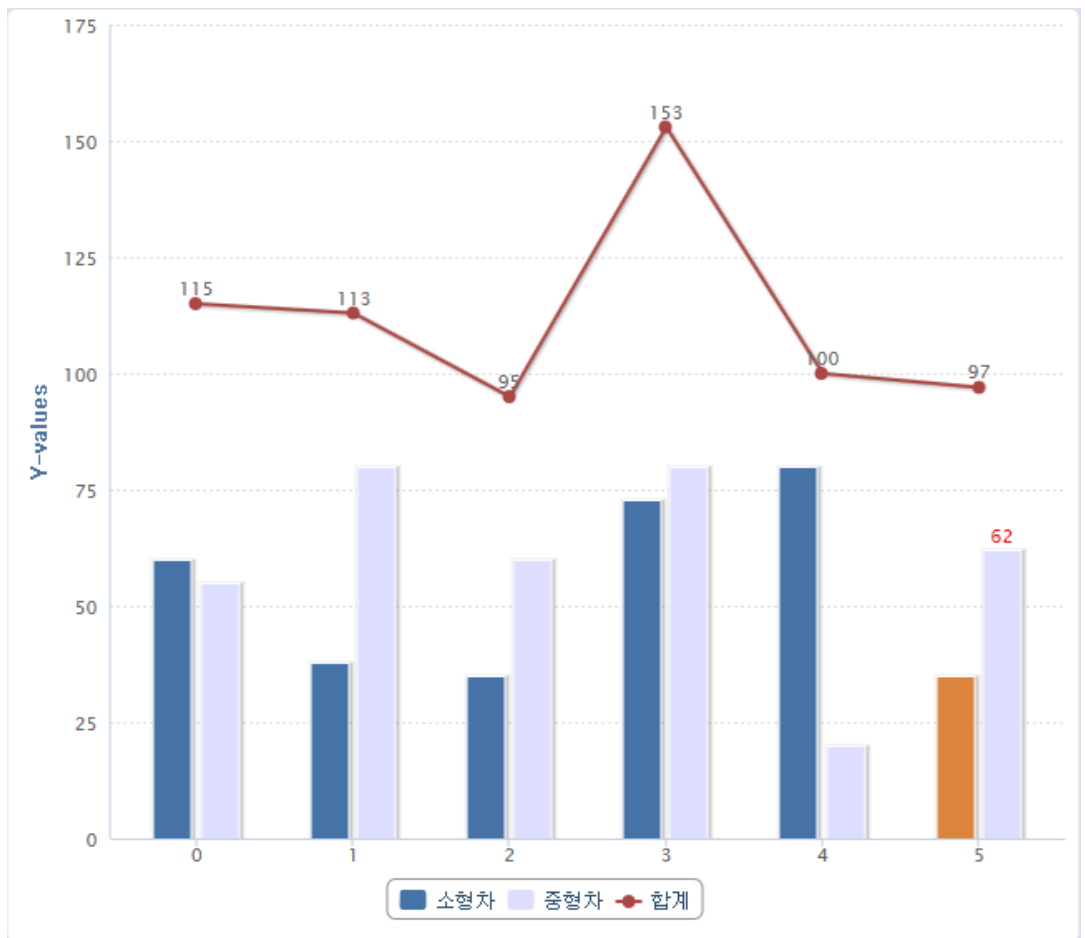
➤ **Example**

<pre>myChart.SetSeriesOptions([Name : "소형차", Type : "column", Data : [// 첫번째 시리즈 설정, 첫번째 시리즈 마지막 포인트에 색상 {Y:60}, {Y:38}, {Y:35}, {Y:73}, {Y:80}, {Y:35, Color : "#DB843D"}],[// 두번째 시리즈 설정, 마지막 포인트에 데이터 레이블 표시</pre>



```
Name : "중형차",
Data : [55,80,60,80,20,{
    Y: 62,
    DataLabels:{
        Enabled:true,
        Color:"red"
    }
}],
Type : "column",
Color : "#DDDDFF"
},{ // 3번째 시리즈 생성
    Name : "합계",
    Data : [115,113,95,153,100,97],
    DataLabels:{Enabled:true}
}], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

SetSubtitleOptions

➤ 기능

차트에 부제목을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Text	String	서브타이틀 텍스트
Align	String	서브타이틀 가로 정렬 (left, right, center)
VerticalAlign	String	서브타이틀 세로 정렬 (top, middle, bottom)
X	Number	X 위치 세부조정(픽셀 단위)
Y	Number	Y 위치 세부조정(픽셀 단위)
Floating	Boolean	플로팅 여부
Style	Object	부제목의 스타일을 지정한다. CSS Object ex) { color:'#3E576F' }

➤ Syntax

Syntax	ChartObj. SetSubtitleOptions (SubTitle, [merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
SubTitle	Object	필수	SubTitle 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

```
myChart.SetSubtitleOptions({
  Text : "부제목",
  Align : "center",
  VerticalAlign : "middle",
  Style:{
```




```
color:"red",  
fontWeight: 'bold'  
}  
));
```

부제목

- 제공 버전
7.0.0.0

SetTitleOptions

➤ 기능

차트에 제목을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Text	String	타이틀 텍스트
Align	String	타이틀 가로 정렬 (left, right, center)
VerticalAlign	String	타이틀 세로 정렬 (top, middle, bottom)
X	Number	X 위치 세부조정(픽셀 단위)
Y	Number	Y 위치 세부조정(픽셀 단위)
Floating	Boolean	플로팅 여부(차트안에 다른 객체와 위치조정 여부) default:false
Style	Object	제목의 스타일을 지정한다. CSS Object ex) { color:'#3E576F' }

➤ Syntax

Syntax	ChartObj. SetTitleOptions (Title, [merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
Title	Object	필수	SubTitle 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ Example

```
myChart.SetTitleOptions({
  Text : "제목",
  Align : "center",
  VerticalAlign : "top",
```



```
Style:{  
    Color:"blue",  
    FontWeight: 'bold'  
}  
});  
  
myChart.Draw();
```

제목

- 제공 버전
7.0.0.0

SetToolTipOptions

➤ 기능

툴팁을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	툴팁의 사용여부
Style	Object	툴팁의 스타일을 지정한다. CSS Object ex) { color:'#3E576F' }
Shared	Boolean	같은 포인트의 툴팁 공유여부(true로 설정시 시리즈와 무관하게 한 포인트에서 동일하게 툴팁이 표시됨)
BorderColor	Text	테두리 색상
BackgroundColor	Text	배경 색상
BorderWidth	Number	테두리 굵기
Formatter	Function	툴팁 표현 정의 함수 this.x (x축 레이블) this.y (y축 레이블) this.series.name (범례에 표시된 시리즈명, Shared가 true인 경우에는 사용불가) this.percent (포인트에 대한 퍼센트값[파이차트나 100%누적차트에서만 사용가능]) this.total (포인트에 전체 값[누적차트에서만 사용가능])

➤ Syntax

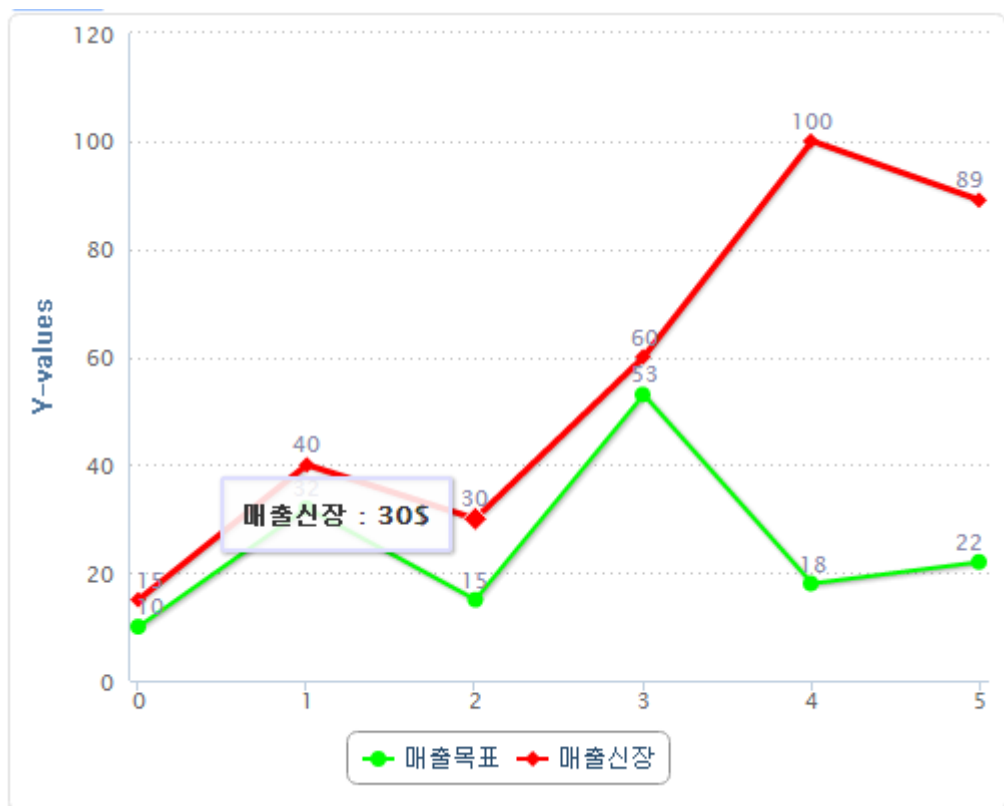
Syntax	ChartObj. SetToolTipOptions (Tooltip, [merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
Tooltip	Object	필수	Tooltip 속성 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ **Example**

```
// 툴팁 설정
myChart.SetToolTipOptions({
  Enabled:true,
  Style: {
    padding: 10,
    fontWeight: 'bold',
    fontSize : '18px'
  },
  BorderWidth:2,
  BorderRadius:1,
  BorderColor:"#DDDDFF",
  Formatter:function(){
    return this.series.name+" : "+this.y+"$";
  }
});
```



- 제공 버전
7.0.0.0

SetXAxisOptions

➤ 기능

차트의 XAxis 속성을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Categories	Object	축레이블 값 설정 Text Array
Labels	Object	축레이블의 세부설정 Json Object
LineColor	Text	X축 선의 색상
LineWidth	Number	X축 선의 굵기
GridLineColor	String	X축 그리드 선 색상
GridLineDashStyle	String	X축 그리드 선 모양(점선, 직선등)
GridLineWidth	Number	X축 그리드 굵기
Offset	Number	축과 차트 데이터(plot) 영역과의 간격
TickInterval	Number	표시되는 X축의 간격표시
Title	Object	X축 타이틀 Json Object

레이블(Labels) 설정값은 Json 형태로 설정하고, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	X축 레이블 사용여부
Align	String	X축 레이블 정렬 (left, center, right)
Formatter	Function	X축 레이블 표현 함수
Rotation	Number	X축 레이블의 회전
StaggerLines	Number	X축이 가로일 때 축간 간격에 따라 줄바꿈
Style	Object	X축 레이블의 스타일 설정 CSS Object

XAxis의 설정값중 Title은 Json 형태로 설정하고, 설정가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	X축 타이틀 사용여부
Align	String	X축 타이틀 정렬 (left, center, right)
Rotation	Number	X축 타이틀 의 회전
Text	String	X축 타이틀 설정
Style	Object	X축 타이틀 스타일 설정 CSS Object

➤ **Syntax**

Syntax	ChartObj. SetXAxisLabelsText (Options, [merge])
--------	--

➤ **Info**

Parameter	Type	필수여부	설 명
Options	Object	필수	X축 옵션 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

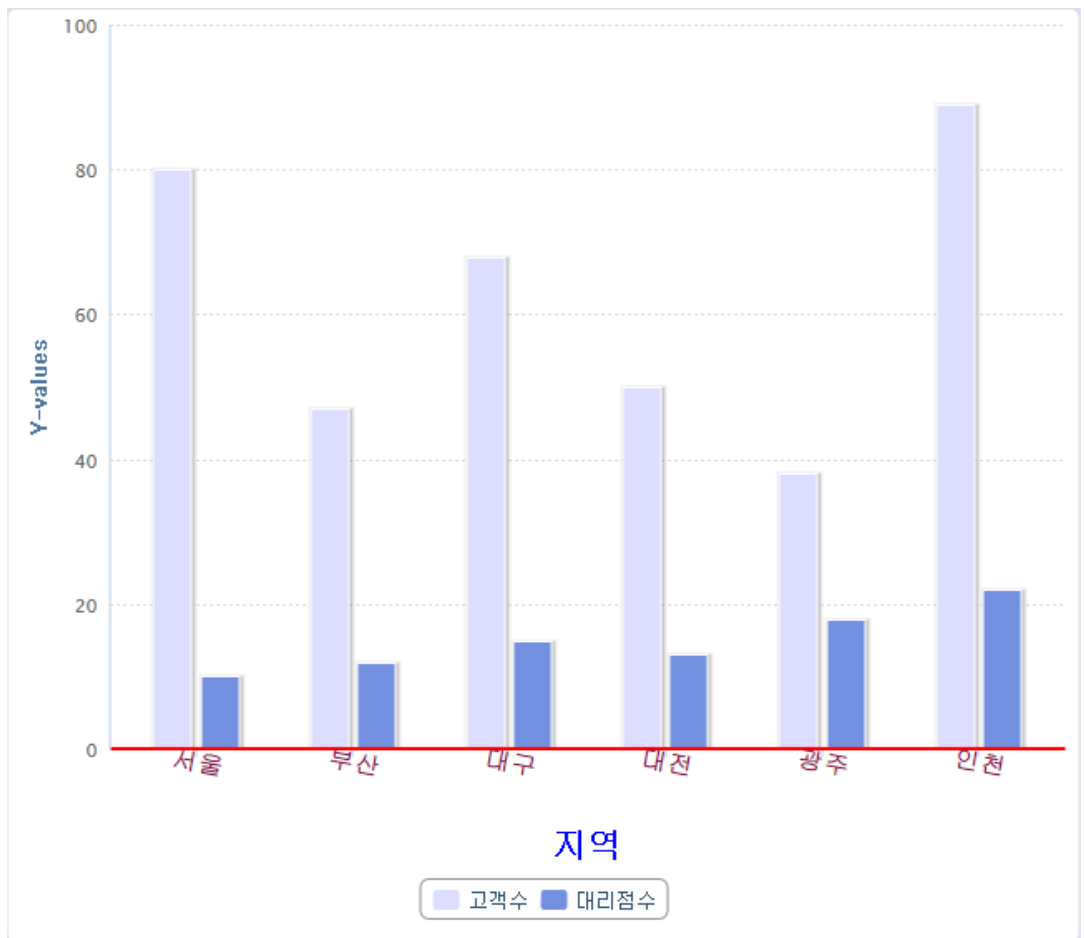
➤ **Example**

```
myChart.SetXAxisOptions({ //X축 속성
  Categories:['서울','부산','대구','대전','광주','인천'],
  LineColor:"red",
  LineWidth:2,
  Labels:{
    Enabled:true,
    Align:"center",
    Rotation:10,
    Style:{
      Color:"#8B0A50",
      FontWeight:'bold',
      FontSize:'15px'
    }
  },
  Title:{
```




```
Enabled:true,
Text:'지역',
Style:{
  FontSize:'20px',
  Color:'blue'
}
});
myChart.SetSeriesOptions([
  Name : "고객수",
  Type : "column",
  Data : [
    {Y:80},{Y:47},{Y:68},
    {Y:50},{Y:38},{Y:89}
  ],
  Color : "#dddddff"
},{
  Name : "대리점수",
  Type : "column",
  Data : [
    {Y:10},{Y:12},{Y:15},
    {Y:13},{Y:18},{Y:22}
  ],
  Color : "#7290df"
}], 1);

myChart.Draw();
```



- 제공 버전
7.0.0.0

SetYAxisOptions

➤ 기능

차트의 YAxis 속성을 설정한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Labels	Object	레이블 설정 Json Object
LineColor	Text	Y축 라인 색상
LineWidth	Number	Y축 라인 넓이
GridLineColor	String	Y축 그리드 색상
GridLineDashStyle	String	Y축 그리드 선 스타일
GridLineWidth	Number	Y축 그리드 넓이
MinorGridLineColor	String	Y축 Minor 그리드 색상
MinorGridLineDashStyle	String	Y축 Minor 그리드 선 스타일
MinorGridLineWidth	Number	Y축 Minor 그리드 넓이
Offset	Number	축과 차트 데이터 영역과의 간격
TickInterval	Number	표시되는 Y축의 간격표시
MinorTickInterval	Number	TickInterval 간격 사이의 작은 간격표시
Title	Object	Y축 타이틀 Json Object
Opposite	Boolean	반대편 Y축 표시
Max	Number	Y축 최대값 설정
Min	Number	Y축 최소값 설정

레이블 설정값은 Json 형태로 설정하고, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	Y축 레이블 사용여부
Align	String	Y축 레이블 정렬 (left, center, right)
Formatter	Function	Y축 레이블 표현 함수

Rotation	Number	Y축 레이블의 회전
StaggerLines	Number	Y축이 가로일 때 축간 간격에 따라 줄바꿈
Style	Object	Y축 레이블의 스타일 설정 CSS Object

YAxis의 설정값중 Title은 Json 형태로 설정하고, 설정가능한 속성은 다음과 같다.

Parameter	Type	설 명
Enabled	Boolean	Y축 타이틀 사용여부
Align	String	Y축 타이틀 정렬 (left, center, right)
Rotation	Number	Y축 타이틀 의 회전 (default 270)
Text	String	Y축 타이틀 설정
Style	Object	Y축 타이틀 스타일 설정 CSS Object
Margin	Number	Y축 타이틀 margin 설정

➤ **Syntax**

Syntax	ChartObj. SetYAxisLabelsText (Options, [merge])
--------	--

➤ **Info**

Parameter	Type	필수여부	설 명
Options	Object	필수	X축 옵션 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default:0)

➤ **Example**

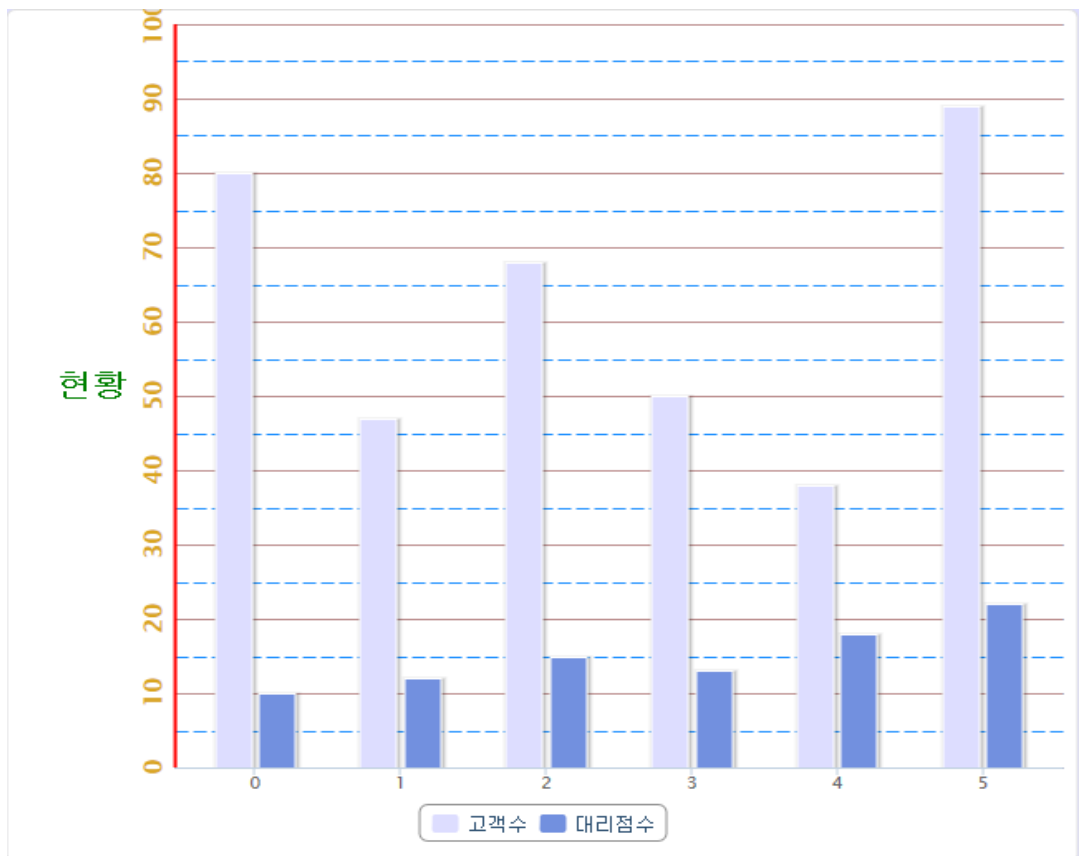
```
myChart.SetYAxisOptions({
  LineColor:"red",
  LineWidth:2,
  TickInterval:10,
  MinorTickInterval:5,
  GridLineDashStyle:"solid",
  GridLineColor:"#bc8f8f",
  MinorGridLineDashStyle:"longdash",
```



```
MinorGridLineColor:"#1e90ff",
Labels:{
  Enabled:true,
  Align:"center",
  Rotation:270,
  Style:{
    Color:"#daa520",
    FontWeight:'bold',
    FontSize:'15px'
  }
},
Title:{
  Enabled:true,
  Text:'현황',
  Style:{
    FontSize:'20px',
    Color:'green'
  },
  Margin:50,
  Rotation:0
}
});
myChart.SetSeriesOptions([
  Name : "고객수",
  Type : "column",
  Data : [
    {Y:80},{Y:47},{Y:68},
    {Y:50},{Y:38},{Y:89}
  ],
  Color : "#ddddff"
},{
  Name : "대리점수",
```



```
Type : "column",  
Data : [  
    {Y:10},{Y:12},{Y:15},  
    {Y:13},{Y:18},{Y:22}  
    ],  
    Color : "#7290df"  
    }, 1);  
  
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

UpdateSeries

➤ 기능

해당 시리즈가 변경되는 경우 해당 시리즈를 업데이트할 때 사용한다.

➤ Syntax

Syntax	ChartObj. UpdateSeries (Series, Index)
--------	---

➤ Info

Return	없음		
Parameter	Type	필수여부	설 명
Series	Object	필수	Series Object
Index	Number	필수	Series Index

➤ Example

```
// 시리즈 업데이트
var series = myChart.GetSeries(1);
series.SetOptions({          //시리즈의 속성을 변경
    Type:"line",
    Data:[10,20,30,40,50,
           {Y:60,DataLabels:{Enabled:true}},70],
    LegendIndex:-1
},1);
myChart.UpdateSeries(series,1);
```

➤ 제공 버전

7.0.0.0



10.2 Series

시리즈에 대한 옵션을 설정한다.

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	SeriesObj. AddPoint (Value, Redraw, Shift)
--------	---

➤ Info

Parameter	Type	필수여부	설 명
Value	Number	필수	Y 값
Redraw	Boolean	선택	추가후 표시 여부 (Default : true)
Shift	Boolean	선택	시프트 처리 여부 (Default : false)

➤ Example

<pre>// 동적으로 포인트를 추가하여 표현 한다. var series = myChart.GetSeries(0); series.AddPoints(30);</pre>
--

➤ 제공 버전

7.2.0.2	
----------------	--

AddPoints

➤ 기능

시리즈에 포인트를 추가한다.

설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

Parameter	Type	설 명
Y	Number	Y 값
X	Number	X 값
Name	String	포인트 이름
Sliced	Boolean	오프셋 표시여부 (Pie에서 사용)
Color	String	포인트 색상
DialRadius	Number	중심을 기준으로 바늘의 최대 길이. 100% 로 입력시 게이지안에 꽂찰 정도의 길이임 % 단위로 입력.
DialWidth	Number	바늘의 폭. 픽셀단위
DialLength	String	바늘의 뾰족한 정도. 100% 입력시 막대 모양, 1% 입력시 뾰족한 바늘 모양. % 단위로 입력.
DialRearLength	String	바늘의 뒷부분(중심기준)의 길이. 100%로 입력시, DialRadius 의 범위 내에서 바늘 뒷부분이 꽂참. % 단위로 입력.
DialBackColor	String	게이지 차트의 바늘 색상 #형태로 입력.

해당 메소드는 포인트를 추가하기 위해 사용하는 메소드로 Y, X 값이 필수로 설정되어야 한다.

Sliced 인자는 Pie 타입에서만 설정가능한 옵션이다.

➤ **Syntax**

Syntax	SeriesObj. AddPoints (Points)
--------	--------------------------------------

➤ **Info**

Parameter	Type	필수여부	설 명
Points	Object	필수	Points 속성 Json Object

➤ **Example**

```
// 새로운 시리즈에 포인트를 추가한다.
var series = myChart.CreateSeries();
var points = [
  {Y: 40, X: 0, Name:"하나", Color: "#FFFFCC"}
  ,{Y: 140, X: 1, Name:"둘", Color: "#FF00CC"}
  ,{Y: 240, X: 2, Name:"셋", Color: "#00FFCC"}
  ,{Y: 340, X: 3, Name:"넷", Color: "#0000CC"}
  ,{Y: 440, X: 4, Name:"다섯", Color: "#FF0000"}
  ,{Y: 540, X: 5, Name:"여섯", Color: "#0000CC"}
  ,{Y: 640, X: 6, Name:"일곱", Color: "#665212"}
];
series.AddPoints(points);
myChart.AddSeries(series);
```

➤ **제공 버전**

7.0.0.0



GetName

➤ 기능

시리즈 이름을 확인한다.

➤ Syntax

Syntax	SeriesObj. GetName ()
--------	------------------------------

➤ Info

Return	String, 시리즈 명		
Parameter	Type	필수여부	설 명
없음			

➤ Example

```
// 0번째 시리즈 이름을 확인 한다.  
var series = myChart.GetSeries(0);  
var name = series.GetName();  
alert("0 번째 시리즈 명은 " + name + " 입니다.");
```

➤ 제공 버전

7.0.0.0

SetData

➤ 기능

특정 시리즈의 데이터를 설정한다.

➤ Syntax

Syntax	SeriesObj. SetData (Data, [redraw])
--------	--

➤ Info

Parameter	Type	필수여부	설 명
Data	Object	필수	데이터 배열
redraw	Boolean	선택	렌더링 처리 여부 (default : true)

➤ Example

```
// 시리즈 가져오기 ( chartObj.GetSeries(index) )
var series = myChart.GetSeries(1);
series.SetData( [20,30,50,80,90,100,20] );
```

➤ 제공 버전

7.0.0.0

SetOptions

➤ 기능

특정 시리즈의 속성을 설정한다.

시리즈의 속성을 변경한 후, 변경된 내용을 적용하고 싶다면, 반드시 `chartObj.updateSeries(seriesObj, seriesIndex)` 메소드와 `chartObj.Draw()`를 호출하여야 한다.

시리즈의 설정 값은 JSON 타입으로 설정하며, 설정 가능한 속성은 다음과 같다.

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>SeriesObj.SetOptions(Options, [merge])</code>
--------	---

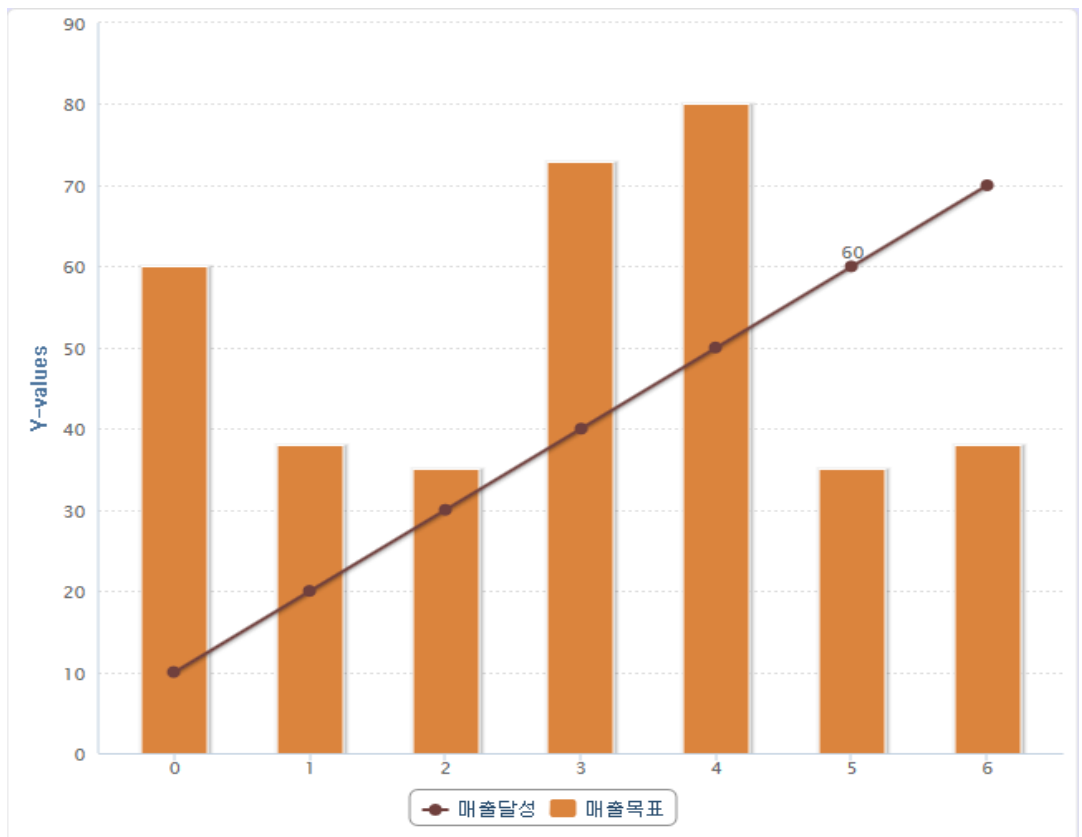
➤ Info

Parameter	Type	필수여부	설 명
Options	Object	필수	시리즈 속성 객체 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default : 0)

➤ Example

```

//시리즈 가져오기
var series = myChart.GetSeries(1); // 2번째 시리즈를 가져오기
series.SetOptions({
  Type:"line",      //시리즈의 속성을 Line 차트 유형
  Data:[10,20,30,40,50,
        {Y:60,DataLabels:{Enabled:true}},70],    //시리즈의 데이터를 설정
  LegendIndex:-1    //시리즈 레전드를 1번째 시리즈 보다 앞에 놓기
},1);
myChart.UpdateSeries(series, 1); //변경된 시리즈를 적용하기
myChart.Draw();      //다시 redraw
  
```



➤ 제공 버전
7.0.0.0

SetVisible

➤ 기능

특정 시리즈의 숨김여부를 설정한다.

➤ Syntax

Syntax	SeriesObj. SetVisible (Visible, [redraw])
--------	--

➤ Info

Parameter	Type	필수여부	설 명
Visible	Boolean	필수	보임/숨김 여부
redraw	Boolean	선택	렌더링 처리 여부 (default : true)

➤ Example

```
// 시리즈 가져오기 ( chartObj.GetSeries(index) )
var series = myChart.GetSeries(1);
series.SetVisible(false, 1);
```

➤ 제공 버전

7.0.0.0

10.3 Axis

X, Y 축에 대해 정의한다. `SetStackLabels` 함수는 YAxis 만 지원한다.

GetLabel

➤ 기능

X 축, Y 축의 인덱스별 설정되어 있는 레이블 텍스트를 확인한다.
설정이 없는 경우는 인덱스를 반환 한다.

➤ Syntax

Syntax	AxisObj. GetLabel (Index)
--------	----------------------------------

➤ Info

Parameter	Type	필수여부	설 명
Index	Integer	필수	확인하고자 하는 Axis의 인덱스

➤ Example

```
// 0번째 X축의 인덱스가 2인 레이블값을 확인한다.
var axis = myChart.GetXAxis(0);
var labelText = axis.GetLabel(2);
```

➤ 제공 버전

7.0.0.0



GetDataMax

➤ 기능

Y 축 데이터값에서 최대값을 확인한다.

➤ Syntax

Syntax	AxisObj. GetDataMax()
--------	------------------------------

➤ Info

Parameter	Type	필수여부	설 명
없음			

➤ Example

// 0번째 Y축의 최대값을 확인 한다. var axis = myChart.GetYAxis(0); var max = axis.GetDataMax();

➤ 제공 버전

7.1.0.6	
---------	--



GetDataMin

➤ 기능

Y 축 데이터값에서 최소값을 확인한다.

➤ Syntax

Syntax	AxisObj. GetDataMin()
--------	------------------------------

➤ Info

Parameter	Type	필수여부	설 명
없음			

➤ Example

```
// 0번째 Y축의 최대값을 확인 한다.  
var axis = myChart.GetYAxis(0);  
var max = axis.GetDataMin();
```

➤ 제공 버전

7.1.0.6	
---------	--



GetMax

➤ 기능

X 축, Y 축에 표시되어 있는 최대값을 확인한다.

➤ Syntax

Syntax	AxisObj. GetMax()
--------	--------------------------

➤ Info

Parameter	Type	필수여부	설 명
없음			

➤ Example

```
// 0번째 Y축의 최대값을 확인 한다.  
var axis = myChart.GetYAxis(0);  
var max = axis.GetMax();
```

➤ 제공 버전

7.0.0.0

SetOptions

➤ 기능

특정 Axis 의 속성을 설정한다.

Axis 의 속성은 각 X 축, Y 축에 따라서 SetXAxisOptions, SetYAxisOptions 속성과 같다.

➤ Syntax

Syntax	AxisObj. SetOptions (Options, [merge])
--------	---

➤ Info

Parameter	Type	필수여부	설 명
Options	Object	필수	특정 Axis의 속성 객체 Json Object
merge	Boolean	선택	이전 설정과의 병합 처리 여부 (default : 0)

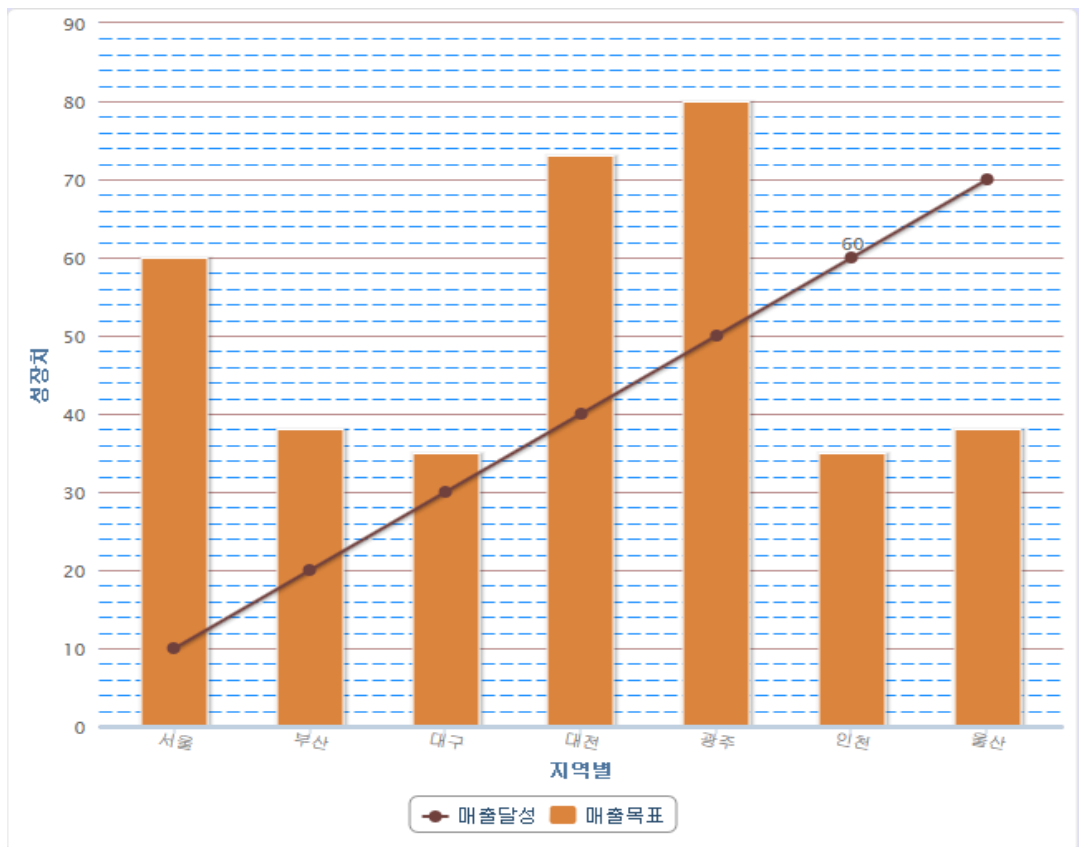
➤ Example

```
var xaxis = myChart.GetXAxis(0);    //X축 가져오기
xaxis.SetOptions({                  //X(0)축 속성변경
    Categories:['서울','부산','대구','대전','광주','인천','울산'],
    LineWidth:3,
    Labels:{
        Rotation:10
    },
    Title:{
        Text:"지역별"
    }
});

var yaxis = myChart.GetYAxis(0);    //Y축 가져오기
yaxis.SetOptions({                  //Y(0)축 속성변경
    TickInterval:10,
```

```
MinorTickInterval:2,
GridLineDashStyle:"solid",
GridLineColor:"#bc8f8f",
MinorGridLineDashStyle:"longdash",
MinorGridLineColor:"#1e90ff",
Title:{
  Text:"성장치"
}
});

myChart.Draw();
```



➤ 제공 버전
7.0.0.0





Chapter 11. IBChart Property



11. IBChart Property

11.1 PlotOptions

charbObj.SetPlotOptions 에서 사용가능한 차트별 속성 표이다.

차트별 사용 가능 여부는 아래의 표와 같다.

전체 리스트 업	Area	AreaSpline	Line	Scatter	Spline	Bar	Column	Pie
Animation	○	○	○	○	○	○	○	○
BorderColor						○	○	○
BorderRadius						○	○	
BorderWidth						○	○	○
Center								○
Color	○	○	○	○	○	○	○	
ColorByPoint						○	○	
Colors						○	○	○
Cursor	○	○	○	○	○	○	○	○
DashStyle	○	○	○	○	○			
DataLabels	○	○	○	○	○	○	○	○
FillColor	○	○						
GroupPadding						○	○	
InnerSize								○
LineColor	○	○						
LineWidth	○	○	○		○			
NegativeColor	○	○	○		○	○	○	
NegativeFillColor	○	○						
PointPadding						○	○	
PointPlacement	○	○	○		○			
PointWidth						○	○	
Shadow	○	○	○	○	○	○	○	○
ShowInLegend	○	○	○	○	○	○	○	



Stacking	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
StartAngle								<input type="radio"/>
Step			<input type="radio"/>					
ToolTip	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Animation

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, pie, arearange, areasplinerange, gauge, series

➤ 기능

애니메이션 기능 추가 설정

➤ Syntax

Syntax	Animation : BooleanValue
--------	---------------------------------

➤ Info

Parameter	Type	필수여부	설 명
BooleanValue	Boolean	필수	애니메이션 설정여부

➤ Example

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

➤ 제공 버전

7.0.0.0

BorderColor

➤ 지원타입

bar, column, pie

➤ 기능

차트 데이터의 테두리 색상을 설정한다.

➤ Syntax

Syntax	BorderColor : ColorValue
--------	--------------------------

➤ Info

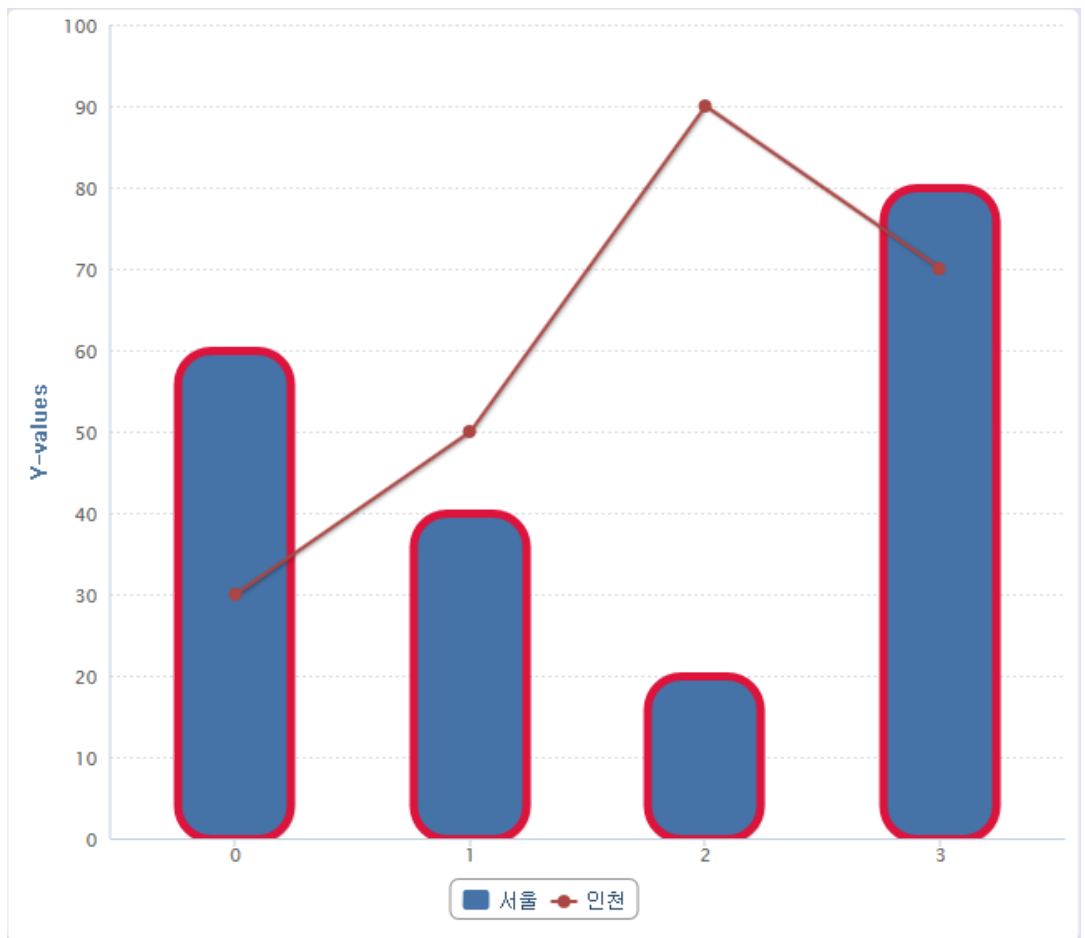
Parameter	Type	필수여부	설 명
ColorValue	String	필수	색상 설정

➤ Example

```
myChart.SetPlotOptions( {
column: {
    BorderColor : "#dc143c",
    BorderWidth:5,
    BorderRadius:20
}});

myChart.SetSeriesOptions([
    {
        Type:"column", Name : "서울",    Data : [60,40,20,80]
    },{
        Type:"line", Name : "인천",    Data : [30,50,90,70]
    }
], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

BorderRadius

➤ 지원타입

bar, column

➤ 기능

차트 데이터 테두리의 Radius 설정한다.

➤ Syntax

Syntax	BorderRadius : RadiusValue
--------	-----------------------------------

➤ Info

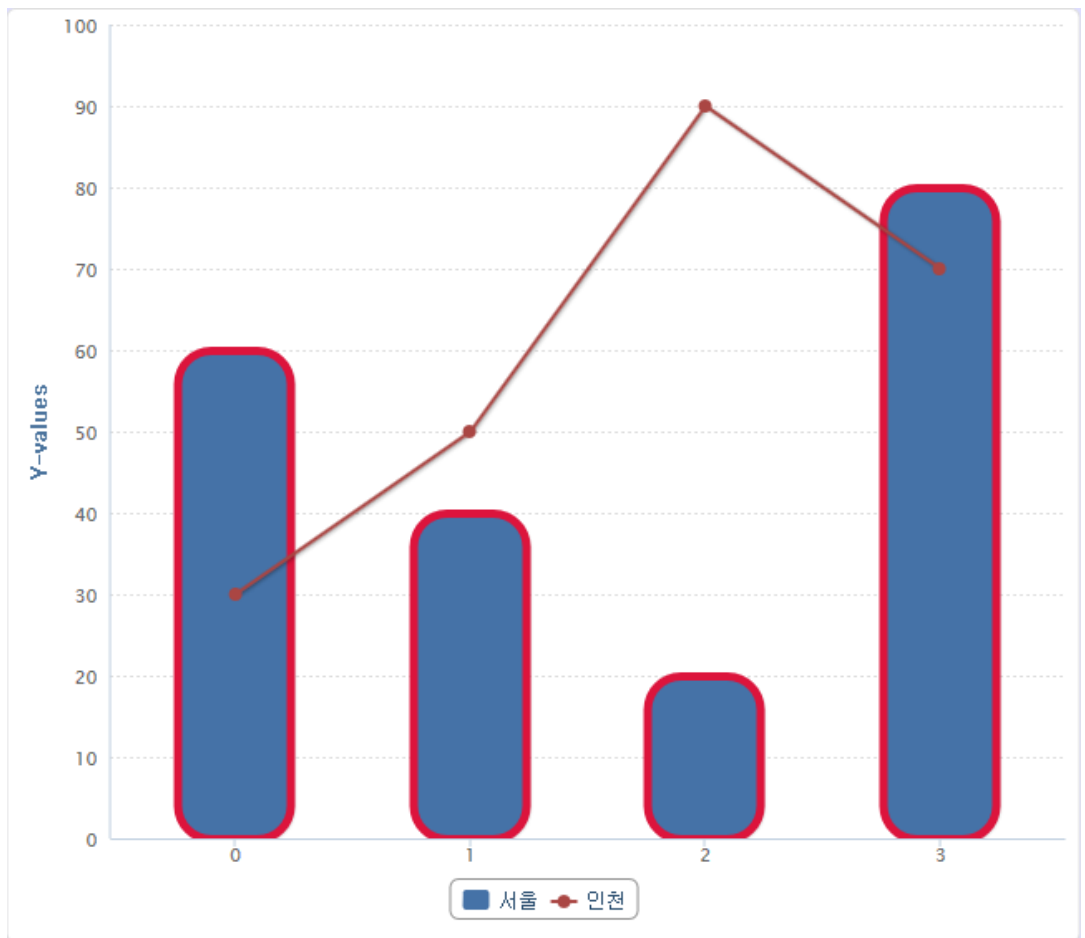
Parameter	Type	필수여부	설 명
RadiusValue	Number	필수	Radius 값 설정

➤ Example

```
myChart.SetPlotOptions( {
column: {
    BorderColor : "#dc143c",
    BorderWidth:5,
    BorderRadius:20
}});

myChart.SetSeriesOptions([
    {
        Type:"column", Name : "서울",    Data : [60,40,20,80]
    },{
        Type:"line", Name : "인천",    Data : [30,50,90,70]
    }], 1);

myChart.Draw();
```

➤ 제공 버전
7.0.0.0



BorderWidth

➤ 지원타입

bar, column, pie

➤ 기능

- 차트 데이터 테두리의 넓이를 설정한다.

➤ Syntax

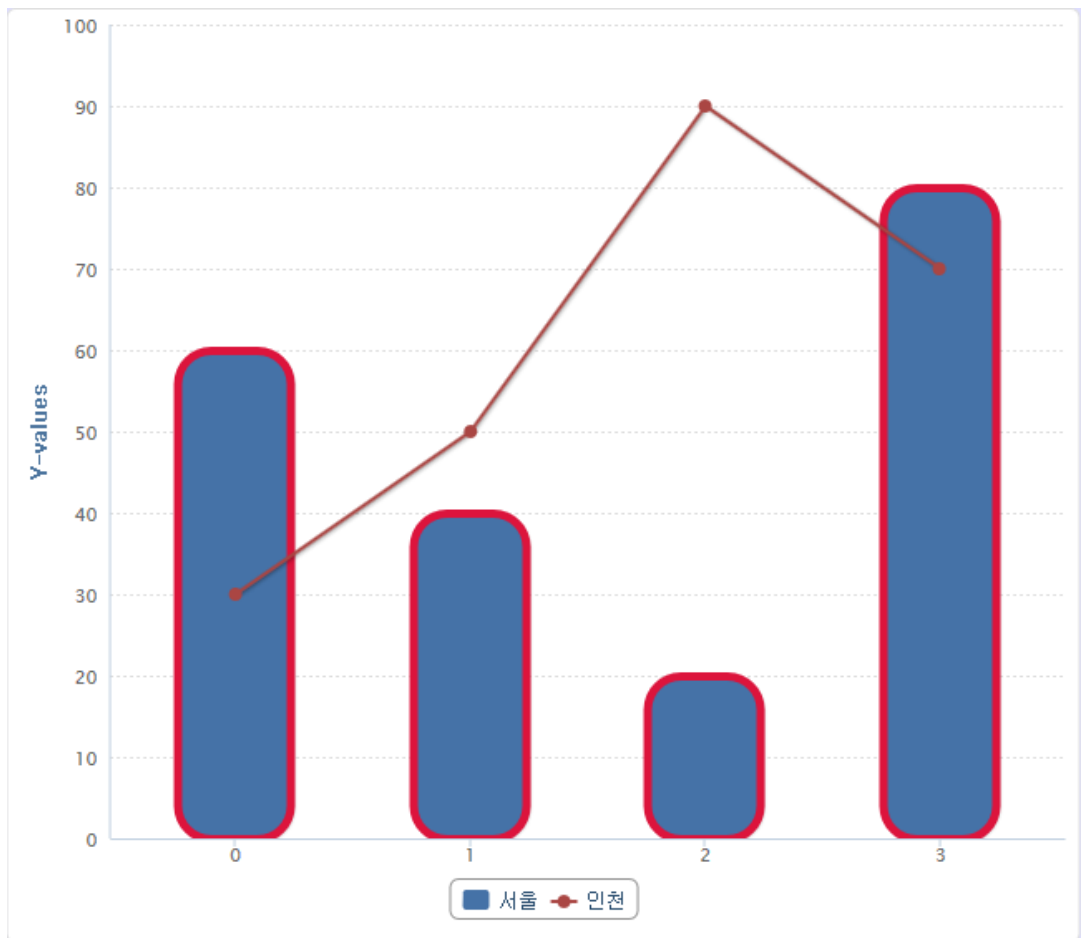
Syntax	BorderWidth : WidthValue
--------	---------------------------------

➤ Info

Parameter	Type	필수여부	설 명
WidthValue	Number	필수	넓이 값 설정

➤ Example

```
myChart.SetPlotOptions( {  
  column: {  
    BorderColor : "#dc143c",  
    BorderWidth:5,  
    BorderRadius:20  
  });  
  
myChart.SetSeriesOptions([[  
  Type:"column", Name : "서울",   Data : [60,40,20,80]  
],{  
  Type:"line", Name : "인천",   Data : [30,50,90,70]  
}], 1);  
  
myChart.Draw();
```



➤ 제공 버전
7.0.0.0



Center

➤ 지원타입

pie

➤ 기능

파이차트 중심의 위치를 설정한다.

값은 % 또는 Pixel value 로 가능하다.

➤ Syntax

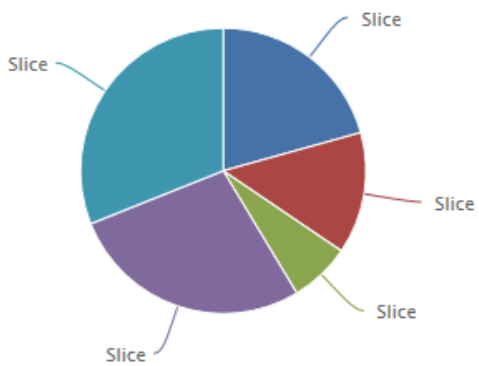
Syntax	Center : PositionValue
--------	------------------------

➤ Info

Parameter	Type	필수여부	설 명
PositionValue	Object	필수	위치 값 설정 [X, Y] (String Array)

➤ Example

```
myChart.SetPlotOptions( {  
  pie: {  
    Center:["20%", "80%"]  
  }  
});  
  
myChart.SetSeriesOptions([ {  
  Type:"pie", Name : "서울",   Data : [60,40,20,80,90]  
}, 1);  
  
myChart.Draw();
```



- 제공 버전
7.0.0.0

Color

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series

➤ 기능

단일 색상을 설정한다.

➤ Syntax

Syntax	Color : ColorValue
--------	---------------------------

➤ Info

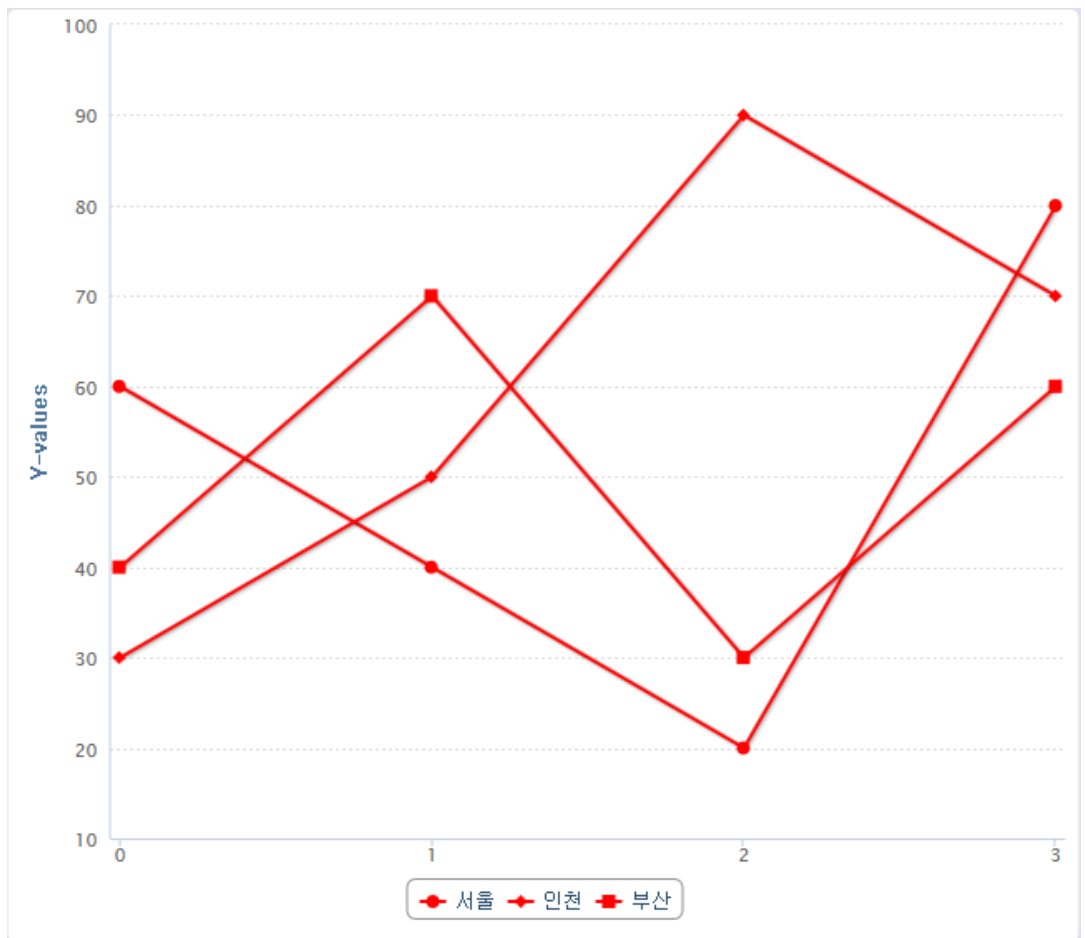
Parameter	Type	필수여부	설 명
ColorValue	String	필수	색상값

➤ Example

```
myChart.SetPlotOptions( {
  series: { //시리즈의 색상을 red로
    Color : "red"
  }
});

myChart.SetSeriesOptions([
  {
    Name : "서울",    Data : [60,40,20,80]
  },
  {
    Name : "인천",    Data : [30,50,90,70]
  },
  {
    Name : "부산",    Data : [40,70,30,60]
  }
], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

ColorByPoint

➤ 지원타입

bar, column

➤ 기능

막대차트에서 포인트별로 색상을 따로 설정할 때 설정한다.

➤ Syntax

Syntax	ColorByPoint : PointValue
--------	----------------------------------

➤ Info

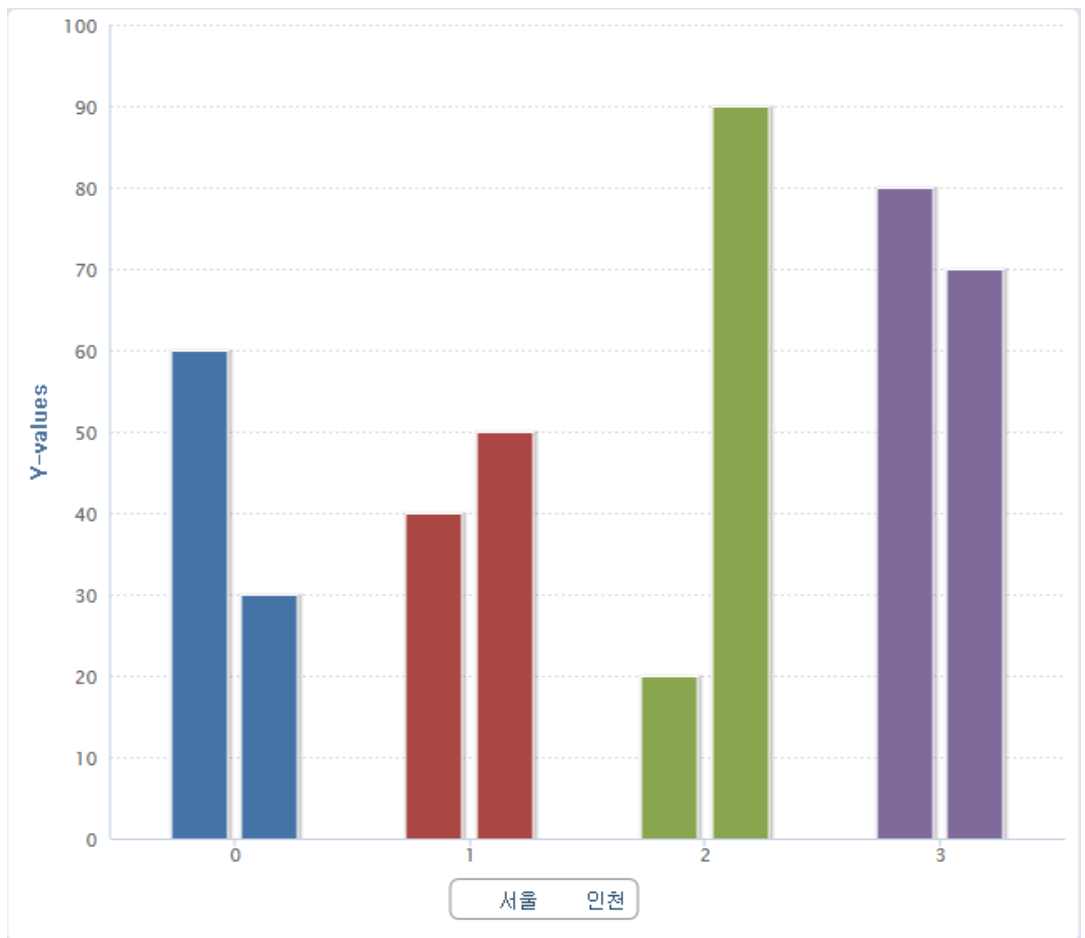
Parameter	Type	필수여부	설 명
PointValue	Boolean	필수	포인트별 색상 설정 여부

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    ColorByPoint : true
  });

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"column", Name : "인천",   Data : [30,50,90,70]
  }], 1);

myChart.Draw();
```

➤ 제공 버전
 7.0.0.0

Colors

➤ 지원타입

bar, column, pie

➤ 기능

포인트별로 차트의 색상을 지정한다. Pie 차트의 경우에는 하나의 시리즈
이므로 제약이 없지만, Bar 차트와 Column 차트의 경우 ColorByPoint 속성이
True 인 경우에만 사용할수 있다.

➤ Syntax

Syntax	Colors : ColorValue
--------	----------------------------

➤ Info

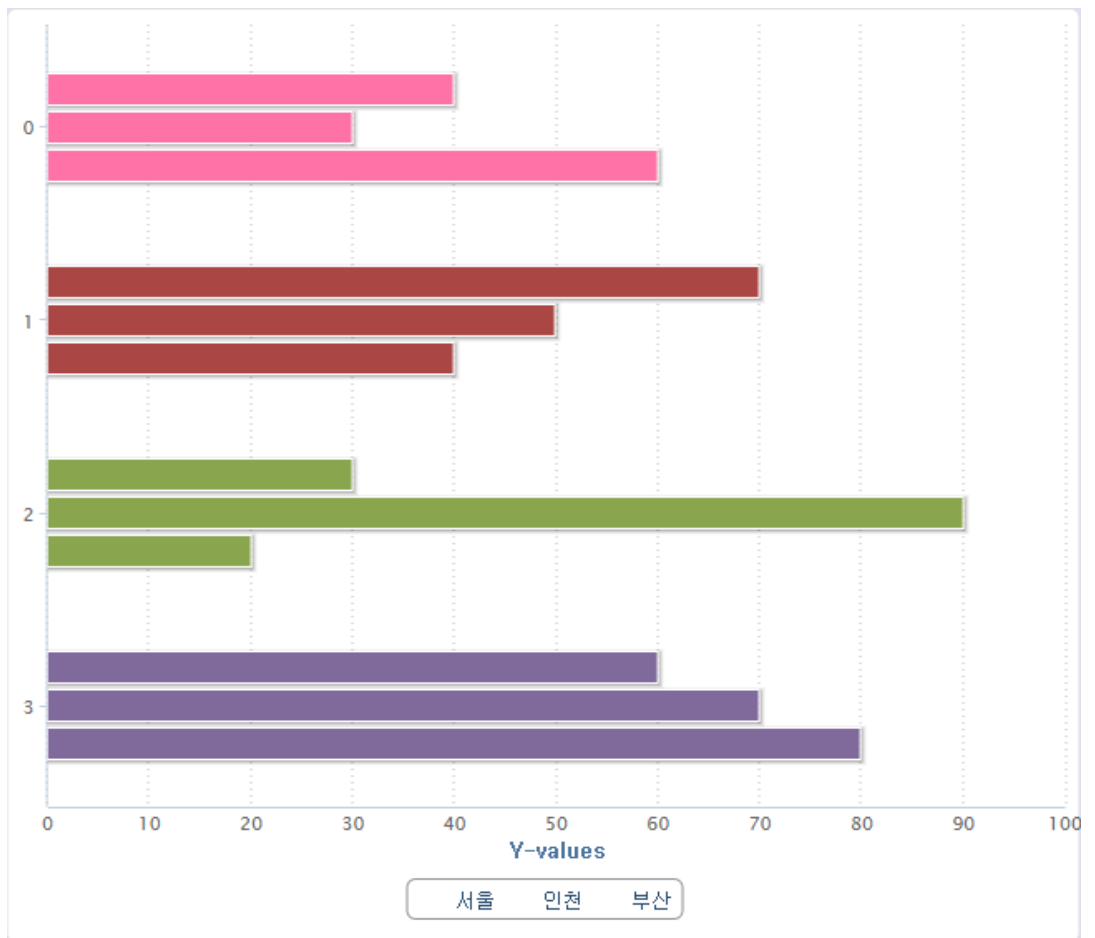
Parameter	Type	필수여부	설 명
ColorValue	Object	필수	차트의 색상순서 Text Array

➤ Example

```
myChart.SetPlotOptions( {
  bar: {
    ColorByPoint:true, //Colors 사용
    Colors: ['#ff72A7','#AA4643','#89A54E','#80699B']
  });

  myChart.SetSeriesOptions([
    {
      Type:"bar", Name : "서울",   Data : [60,40,20,80]
    },
    {
      Type:"bar", Name : "인천",   Data : [30,50,90,70]
    },
    {
      Type:"bar", Name : "부산",   Data : [40,70,30,60]}], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

Cursor

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series, pie

➤ 기능

마우스 오버시 시리즈 및 차트의 Cursor 여부를 설정한다.

➤ Syntax

Syntax	Cursor : CursorValue
--------	-----------------------------

➤ Info

Parameter	Type	필수여부	설 명
CursorValue	String	필수	커서 설정 ("pointer")

➤ Example

```
myChart.SetPlotOptions( {
  bar: {
    Cursor : "pointer"
  });

myChart.SetSeriesOptions([
  {
    Type:"bar", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"bar", Name : "인천",   Data : [30,50,90,70]
  },{
    Type:"bar", Name : "부산",   Data : [40,70,30,60]}], 1);

myChart.Draw();
```

➤ 제공 버전

7.0.0.0

DashStyle

➤ 지원타입

area, areaspline, line, scatter, spline, series

➤ 기능

차트의 DashStyle 을 설정한다.

➤ Syntax

Syntax	DashStyle : DashValue
--------	-----------------------

➤ Info

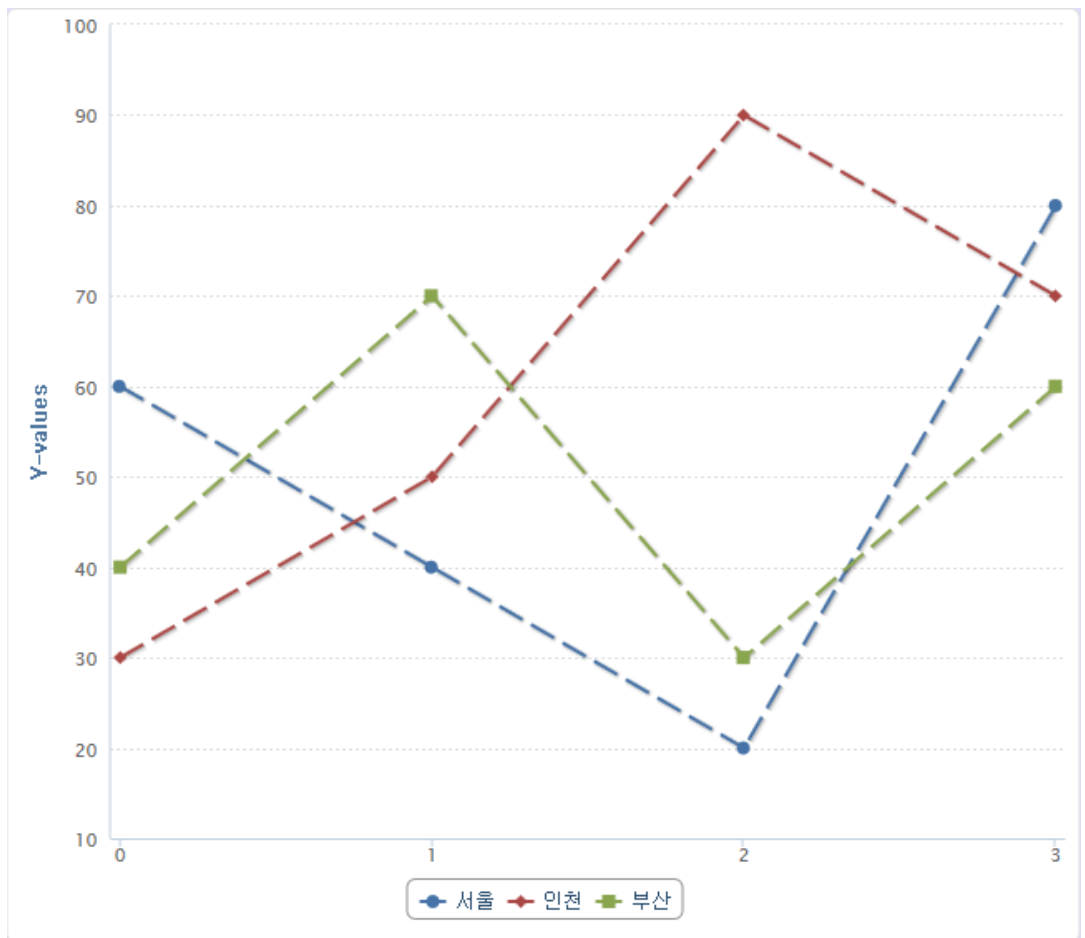
Parameter	Type	필수여부	설 명
DashValue	String	필수	대쉬 타입 설정 (밑에 참고)
DashStyle	Solid, ShortDash, ShortDot, ShortDashDot, ShortDashDotDot, Dot, Dash, LongDash, DashDot, LongDashDot, LongDashDotDot		

➤ Example

```
myChart.SetPlotOptions( {
    line: {
        DashStyle:"LongDash"
    }
});

myChart.SetSeriesOptions([
    {
        Type:"line", Name : "서울",    Data : [60,40,20,80]
    },
    {
        Type:"line", Name : "인천",    Data : [30,50,90,70]
    },
    {
        Type:"line", Name : "부산",    Data : [40,70,30,60]}], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

DataLabels

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series, pie

➤ 기능

데이터 레이블을 설정한다.

속성은 Json Object 형태로 표현하며, 속성은 아래와 같다.

Parameter	Type	설 명
Enabled	Boolean	데이터 레이블 설정
Align	String	데이터 레이블 정렬 설정
VerticalAlign	String	데이터 레이블 수직 정렬 설정
Crop	Boolean	데이터 레이블이 차트 영역을 넘은경우 제거여부
BackgroundColor	String	데이터 레이블 배경색상 설정
BorderColor	String	데이터 레이블의 테두리 색상 설정
BorderRadius	Number	데이터 레이블의 테두리 둥글기 설정
BorderWidth	Number	데이터 레이블의 테두리 굵기 설정
Color	String	데이터 레이블의 텍스트 색상 설정
Formatter	Function	데이터 레이블의 형식 설정
Style	Object	데이터 레이블의 스타일을 지정한다. CSS Object ex) { color:'#3E576F' }

➤ Syntax

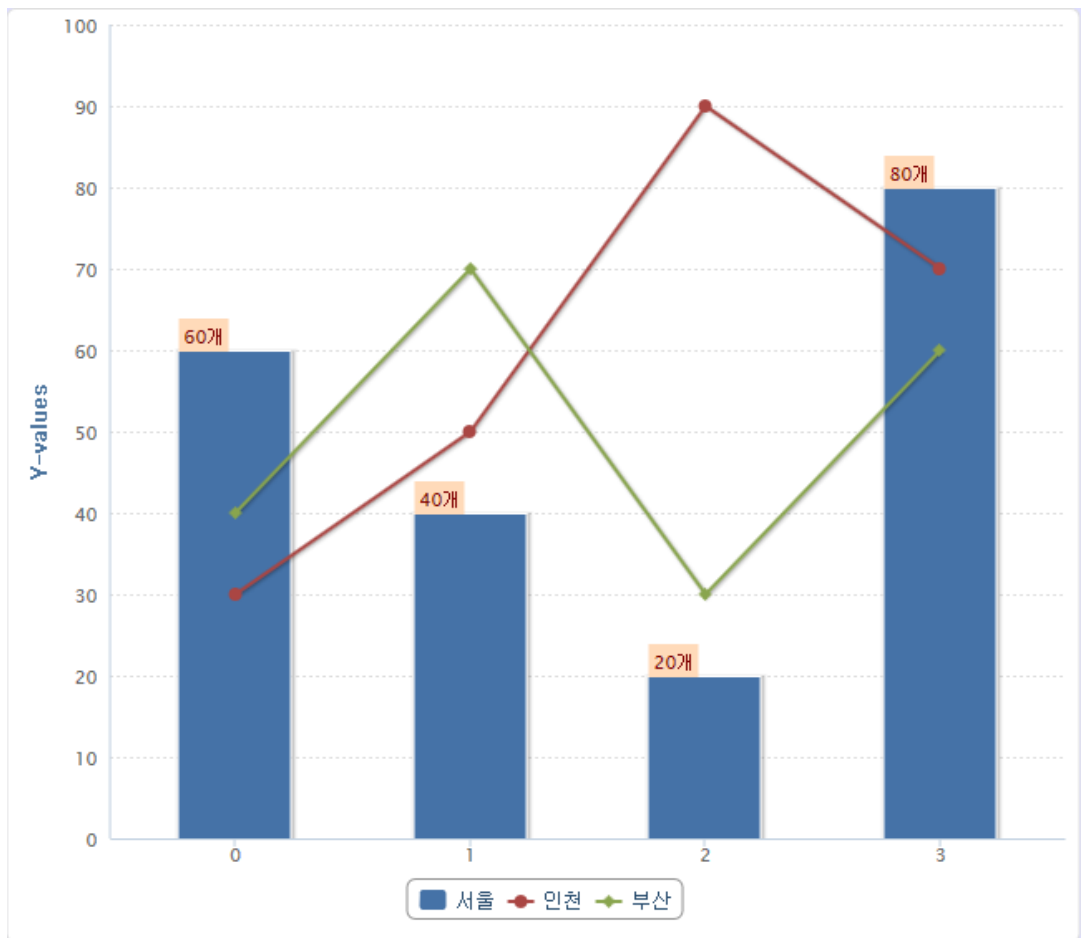
Syntax	DataLabels : LabelValue
--------	--------------------------------

➤ Info

Parameter	Type	필수여부	설 명
LabelValue	Object	필수	데이터 레이블 설정 Json Object

➤ **Example**

```
myChart.SetPlotOptions( {  
column: {    // 컬럼 차트에만 데이터 레이블을 설정  
    DataLabels : {  
        Enabled:true,  
        Align:"left",  
        BackgroundColor:"#ffdab9",  
        Color:"#800000",  
        Formatter:function(){  
            return this.y + "개";  
        }  
    }  
}  
});  
  
myChart.SetSeriesOptions([[  
    Type:"column", Name : "서울",    Data : [60,40,20,80]  
},{  
    Type:"line", Name : "인천",    Data : [30,50,90,70]  
},{  
    Type:"line", Name : "부산",    Data : [40,70,30,60]  
}], 1);  
  
myChart.Draw();
```

➤ 제공 버전
 7.0.0.0

FillColor

➤ 지원타입

area, areaspline, arearange, areasplinerange

➤ 기능

영역 차트에서 영역 부분을 채우는 색상을 정의한다.

➤ Syntax

Syntax	FillColor : fillColor
--------	-----------------------

➤ Info

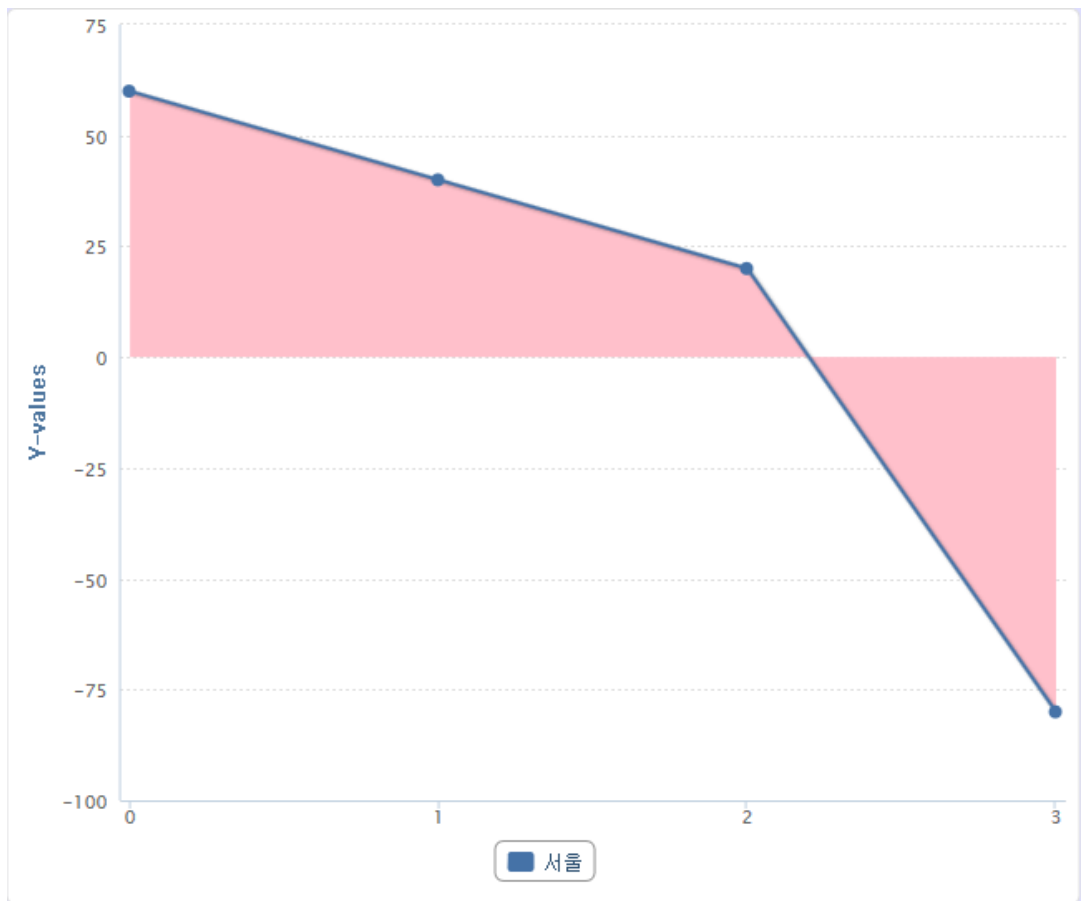
Parameter	Type	필수여부	설 명
fillColor	String	필수	영역 차트의 색상 설정

➤ Example

```
myChart.SetPlotOptions( {
  area: {
    FillColor:"#ffc0cb"
  });

  myChart.SetSeriesOptions([
    {
      Type:"area", Name : "서울", Data : [60,40,20,-80]
    }, 1);

  myChart.Draw();
```



➤ 제공 버전
7.0.0.0

GroupPadding

➤ **지원타입**

bar, column

➤ **기능**

X 축의 그룹들의 Padding 값을 설정한다.

➤ **Syntax**

Syntax	GroupPadding : PaddingValue
--------	------------------------------------

➤ **Info**

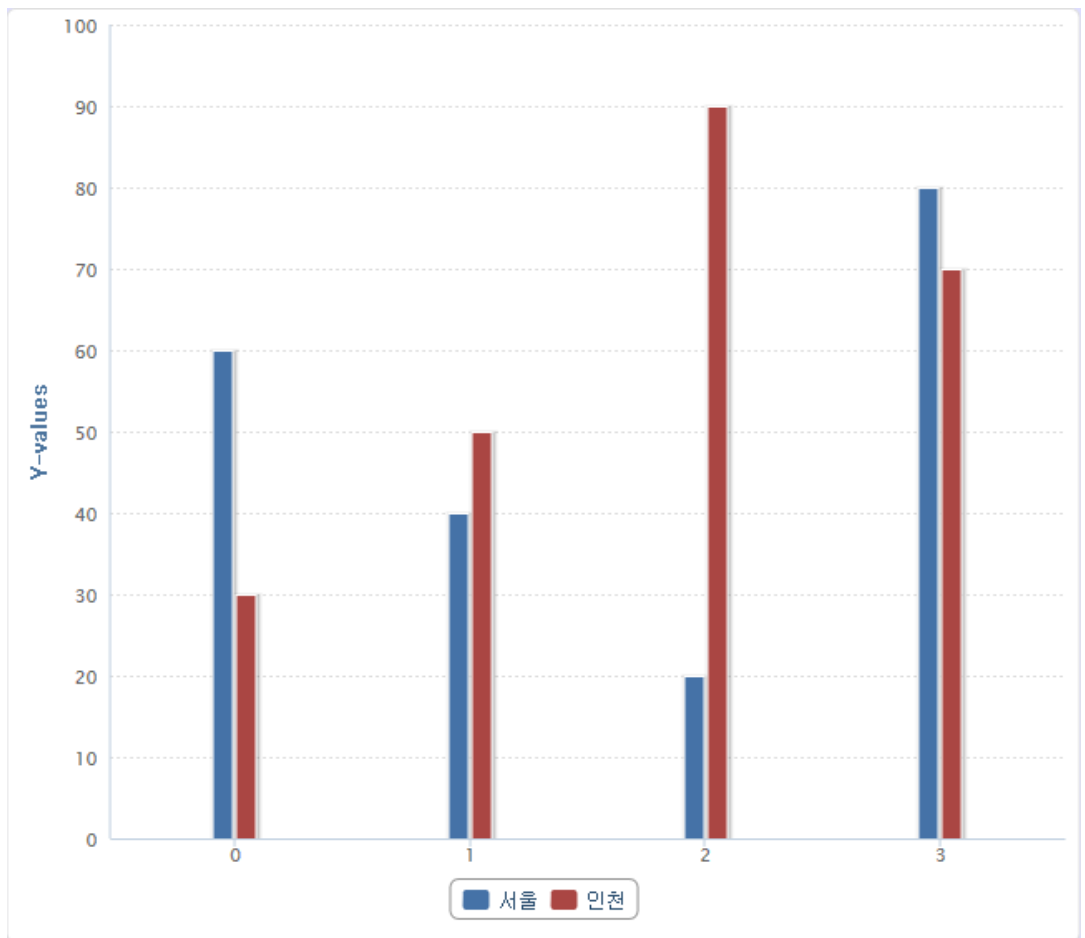
Parameter	Type	필수여부	설 명
PaddingValue	Number	필수	Padding 값 설정 (default : 0.2)

➤ **Example**

```
myChart.SetPlotOptions( {
  column: {
    GroupPadding : 0.4
  });

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"column", Name : "인천",   Data : [30,50,90,70]
  }], 1);

myChart.Draw();
```



➤ 제공 버전

7.0.0.0

InnerSize

➤ 지원타입

pie

➤ 기능

파이차트 가운데 영역에 공간을 만든다. 도넛 차트를 만들기 위해 사용한다.
값은 % 또는 Pixel value 로 가능하다.

➤ Syntax

Syntax	InnerSize : InnerValue
--------	-------------------------------

➤ Info

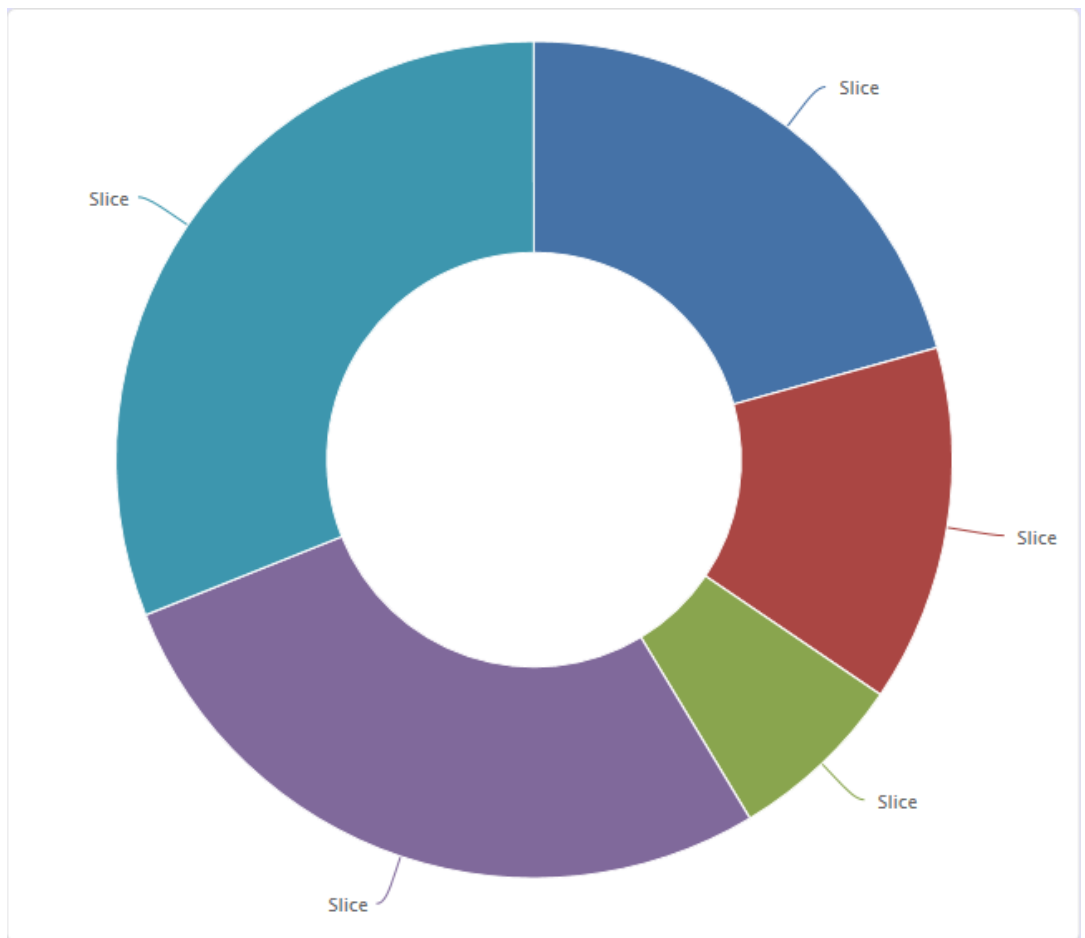
Parameter	Type	필수여부	설 명
InnerValue	String/Number	필수	가운데 영역 설정

➤ Example

```
myChart.SetPlotOptions( {
  pie: {
    InnerSize:"50%"
  }
});

myChart.SetSeriesOptions([
  {
    Type:"pie", Name : "서울", Data : [60,40,20,80,90]
  }, 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

LineColor

➤ 지원타입

area, areaspline, arearange, areasplinerange

➤ 기능

차트의 선색상을 지정한다. Color 속성으로 지정 가능하나 영역 차트의 경우 FillColor 와 구분하기 위해 사용한다.

➤ Syntax

Syntax	LineColor : ColorValue
--------	------------------------

➤ Info

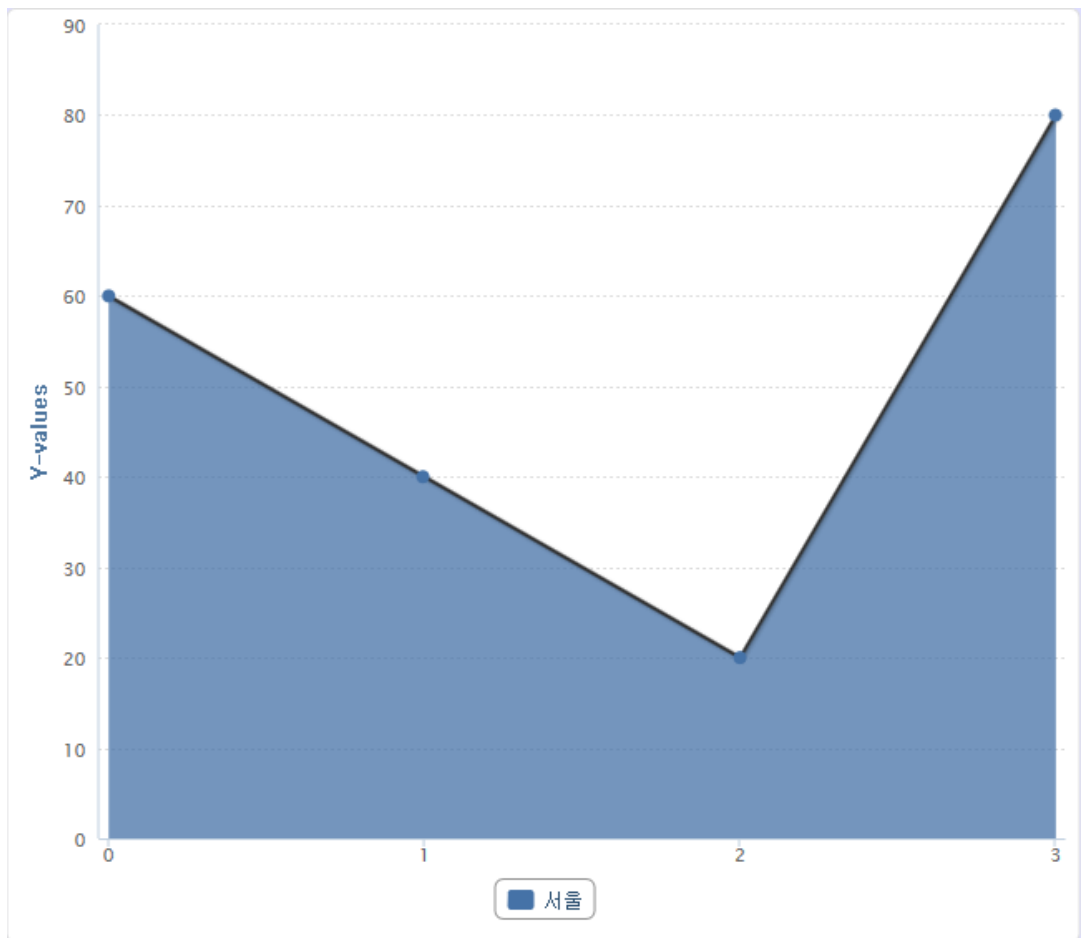
Parameter	Type	필수여부	설 명
ColorValue	String	필수	라인 색상값

➤ Example

```
myChart.SetPlotOptions( {
  area: {
    LineColor:"#303030"
  });

myChart.SetSeriesOptions([
  {
    Type:"area", Name : "서울", Data : [60,40,20,80]
  }, 1);

myChart.Draw();
```

➤ 제공 버전
7.0.0.0

LineWidth

➤ 지원타입

area, areaspline, line, spline, arearange, areasplinerange, series

➤ 기능

선의 넓이를 설정한다.

➤ Syntax

Syntax	LineWidth : WidthValue
--------	------------------------

➤ Info

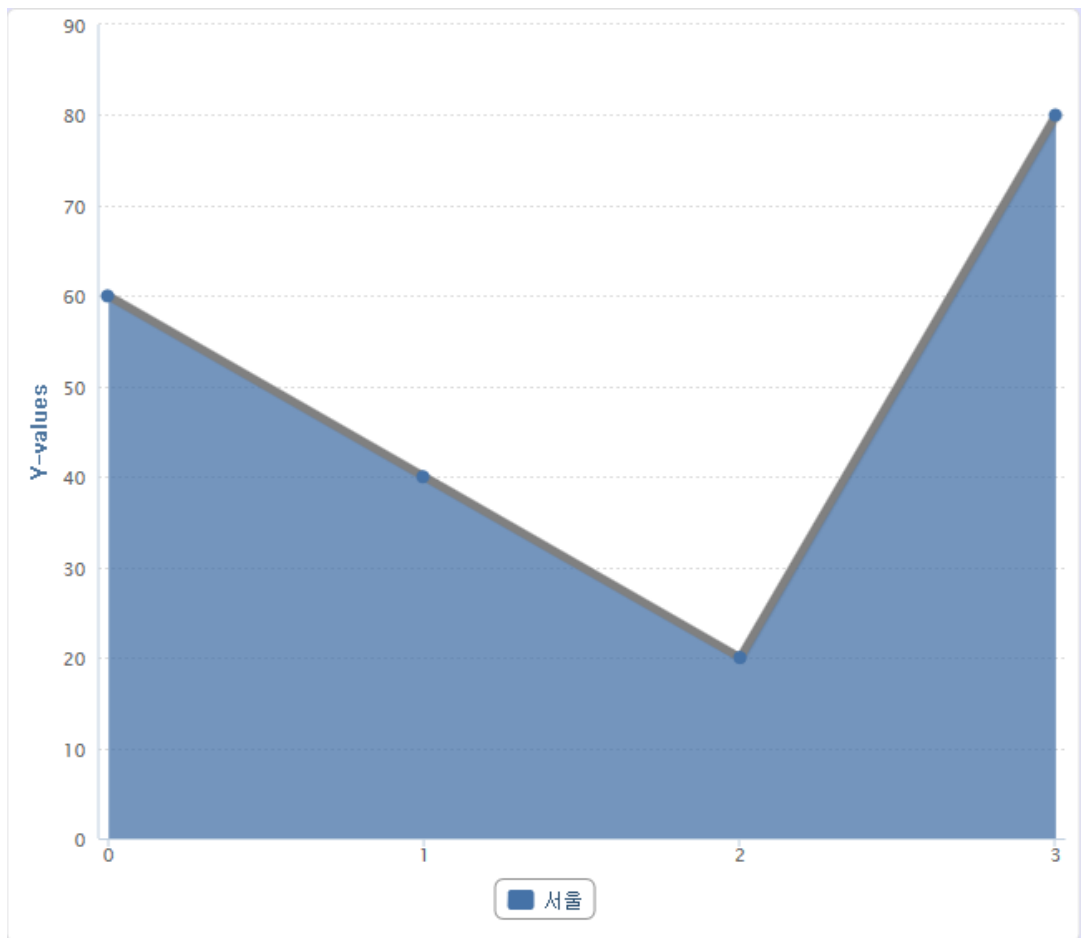
Parameter	Type	필수여부	설 명
WidthValue	Number	필수	선의 넓이 설정

➤ Example

```
myChart.SetPlotOptions( {
  area: {
    LineColor:"gray",
    LineWidth:5
  }
});

myChart.SetSeriesOptions([
  {
    Type:"area", Name : "서울", Data : [60,40,20,80]
  }, 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

NegativeColor

➤ 지원타입

area, areaspline, line, spline, bar, column, arearange, areasplinerange, gauge, series

➤ 기능

음수값의 색상을 설정한다.

➤ Syntax

Syntax	NegativeColor : ColorValue
--------	-----------------------------------

➤ Info

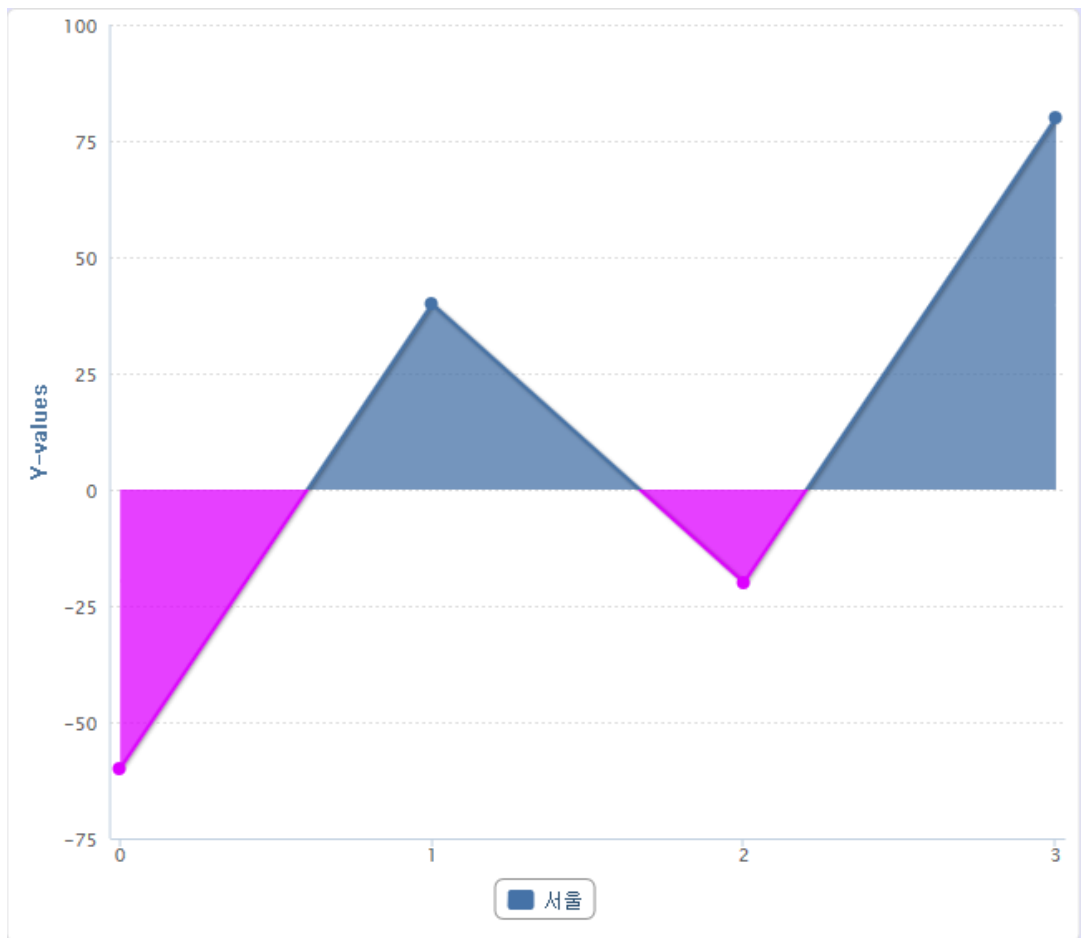
Parameter	Type	필수여부	설 명
ColorValue	String	필수	음수값 색상 설정

➤ Example

```
myChart.SetPlotOptions( {
  area: {
    NegativeColor:"# DD00FF "
  });

myChart.SetSeriesOptions([
  {
    Type:"area", Name : "서울", Data : [-60,40,-20,80]
  }, 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

NegativeFillColor

➤ 지원타입

area, areaspline, arearange, areasplinerange

➤ 기능

영역 차트에서 음수영역을 채우는 색상을 설정

➤ Syntax

Syntax	NegativeFillColor : FillColor
--------	--------------------------------------

➤ Info

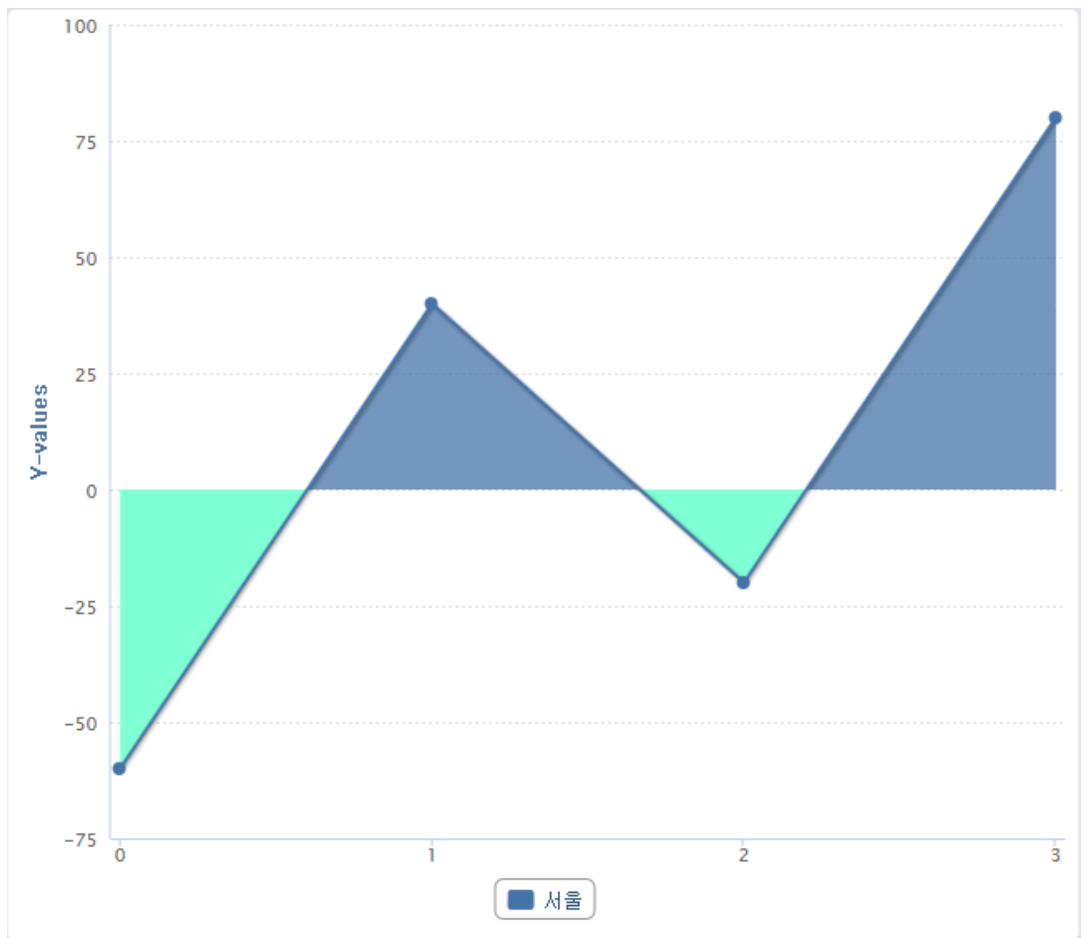
Parameter	Type	필수여부	설 명
FillColor	String	필수	색상 설정

➤ Example

```
myChart.SetPlotOptions( {
  area: {
    NegativeFillColor: "#7ffd4"
  });

  myChart.SetSeriesOptions([ {
    Type: "area", Name : "서울", Data : [-60,40,-20,80]
  }, 1);

  myChart.Draw();
```



➤ 제공 버전
7.0.0.0

PointPadding

➤ 지원타입

bar, column

➤ 기능

그룹안에 포인트별 간격을 설정한다.

➤ Syntax

Syntax	PointPadding : PaddingValue
--------	------------------------------------

➤ Info

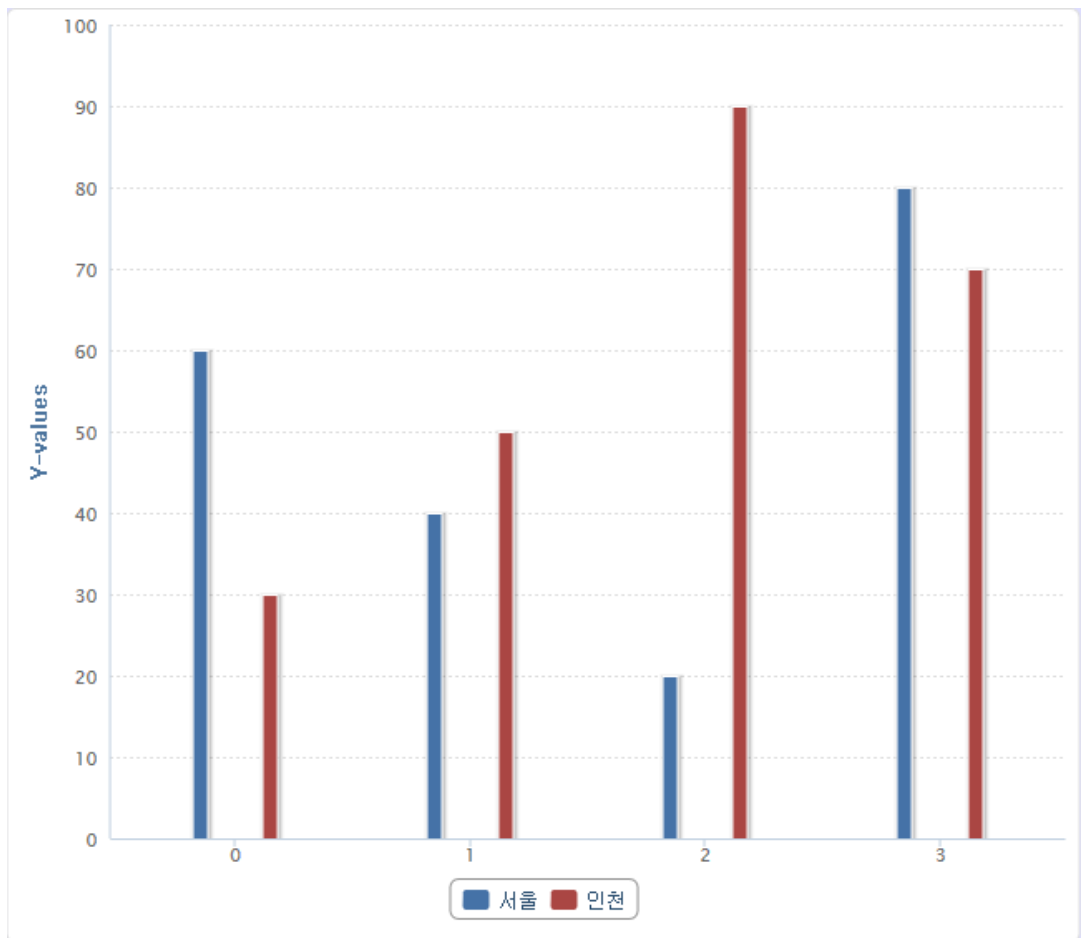
Parameter	Type	필수여부	설 명
PaddingValue	Number	필수	Padding 값 설정 (default : 0.1)

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    PointPadding : 0.4
  });

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"column", Name : "인천",   Data : [30,50,90,70]
  }], 1);

myChart.Draw();
```

➤ 제공 버전

7.0.0.0

PointPlacement

➤ 지원타입

area, areaspline, line, spline, bar, column, arearange, areasplinerange, series

➤ 기능

포인트의 위치를 설정한다.

On 속성은 X 축 포인트 위에, Between 속성은 X 축 포인트와 포인트 사이에 위치한다.

➤ Syntax

Syntax	PointPlacement : PointPlaceValue
--------	---

➤ Info

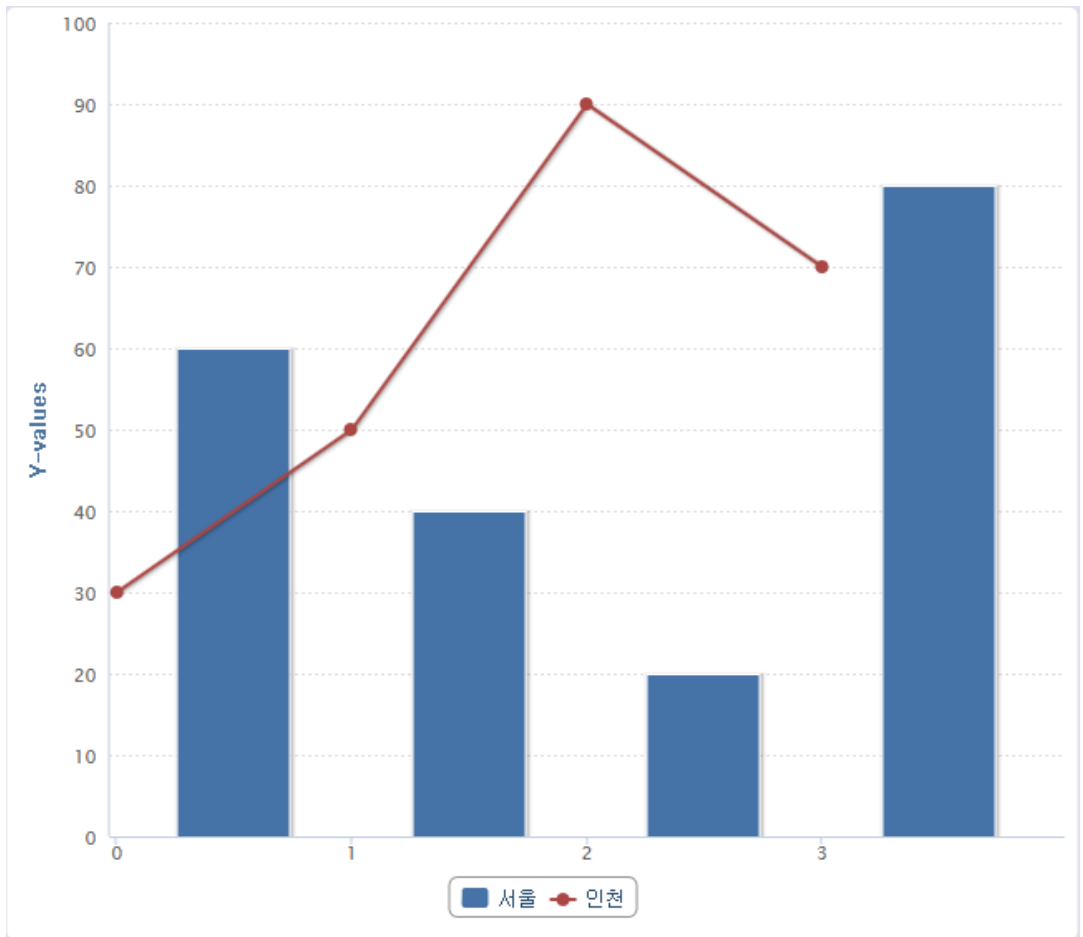
Parameter	Type	필수여부	설 명
PointPlaceValue	String	필수	위치 설정 (on, between)

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    PointPlacement:"between"    //컬럼차트 속성은 포인트와 포인트 사이
  },
  line : {
    PointPlacement:"on"        //라인차트 속성은 x축 포인트 위에
  }
});

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },
  {
    Type:"line", Name : "인천",   Data : [30,50,90,70]
  }
], 1);
```

```
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

PointWidth

➤ 지원타입

bar, column

➤ 기능

포인트의 넓이를 설정한다.

➤ Syntax

Syntax	PointWidth : WidthValue
--------	-------------------------

➤ Info

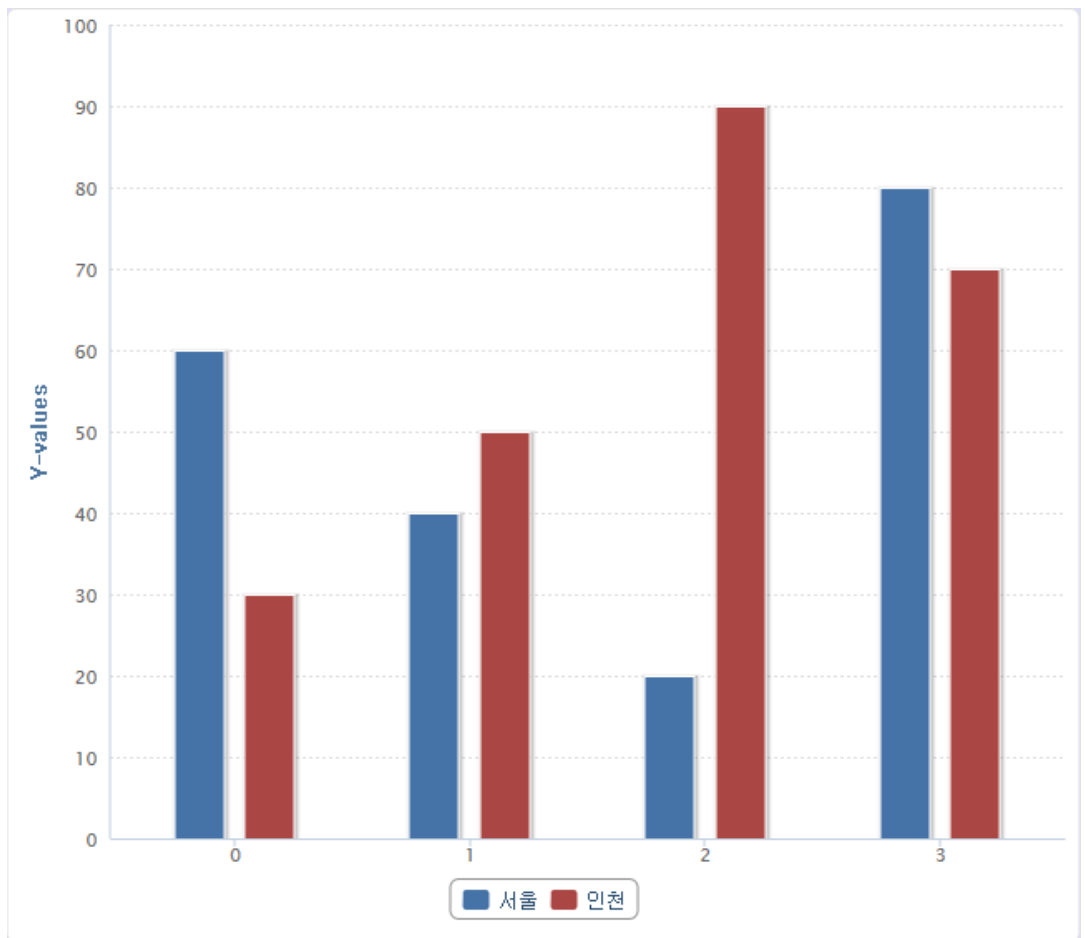
Parameter	Type	필수여부	설 명
WidthValue	Number	필수	포인트 넓이 값 설정

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    PointWidth:30
  });

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"column", Name : "인천",   Data : [30,50,90,70]
  }], 1);

myChart.Draw();
```



➤ 제공 버전

7.0.0.0

Shadow

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, series, pie

➤ 기능

그림자 설정여부.

➤ Syntax

Syntax	Shadow : ShadowValue
--------	-----------------------------

➤ Info

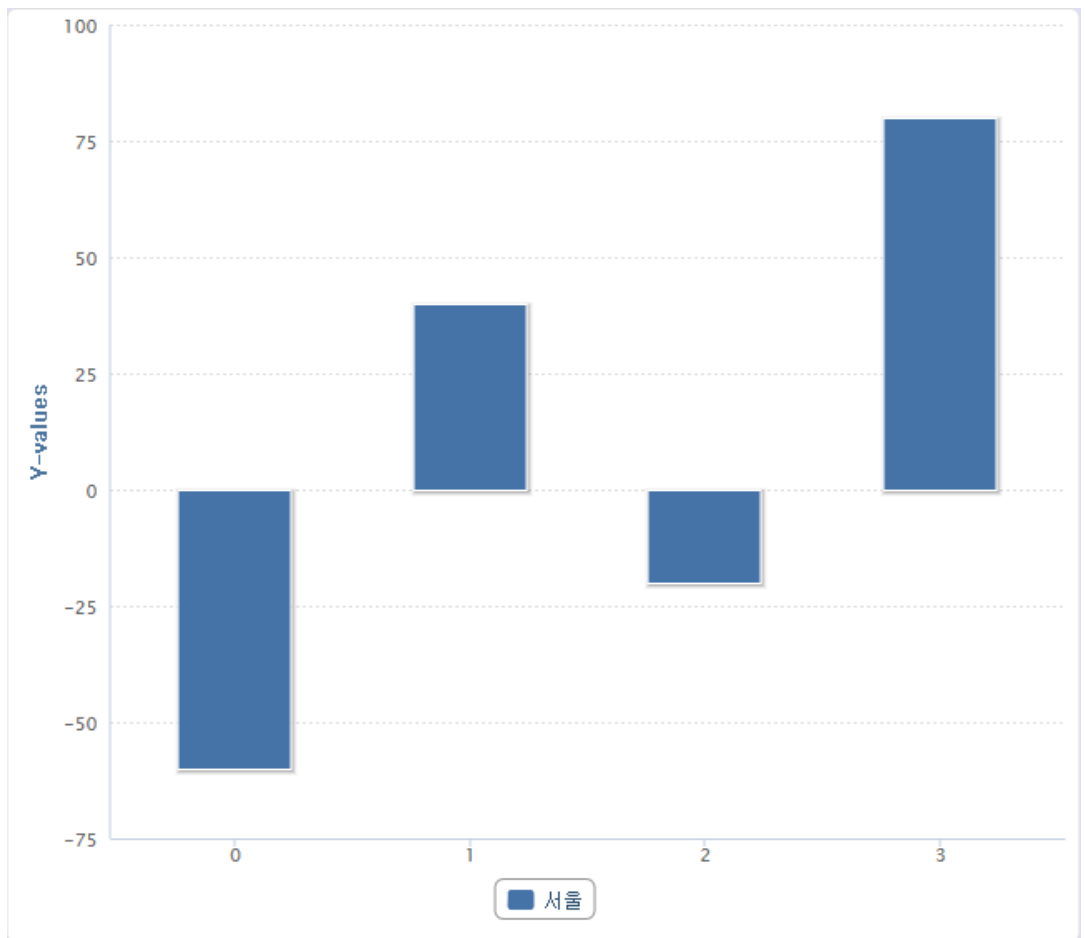
Parameter	Type	필수여부	설 명
ShadowValue	Boolean	필수	그림자 설정여부

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    Shadow :true
  }
});

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [-60,40,-20,80]
  }, 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

ShowInLegend

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series

➤ 기능

마우스 오버시 시리즈 및 차트의 Cursor 여부를 설정한다.

➤ Syntax

Syntax	ShowInLegend : ShowValue
--------	---------------------------------

➤ Info

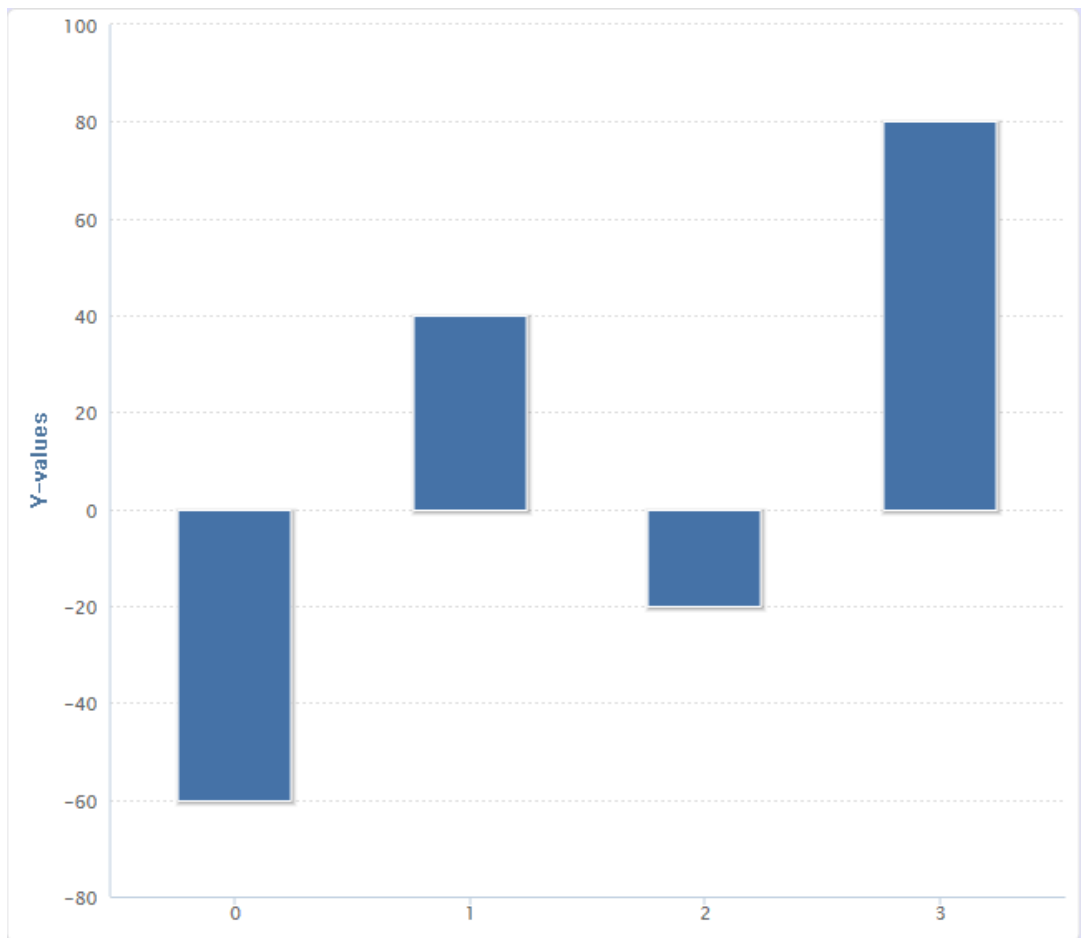
Parameter	Type	필수여부	설 명
ShowValue	Boolean	필수	레전드 표시 여부 설정(default:true)

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    ShowInLegend:false
  }
});

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [-60,40,-20,80]
  }, 1);

myChart.Draw();
```

- 제공 버전
7.0.0.0

Stacking

➤ 지원타입

area, areaspline, line, spline, bar, column, arearange, areasplinerange, series

➤ 기능

Stacking 여부를 설정한다.

한 포인트에 여러가지 시리즈의 값을 쌓아서 표현하는 방식을 Stacking 이라 한다.

➤ Syntax

Syntax	Stacking : StackValue
--------	-----------------------

➤ Info

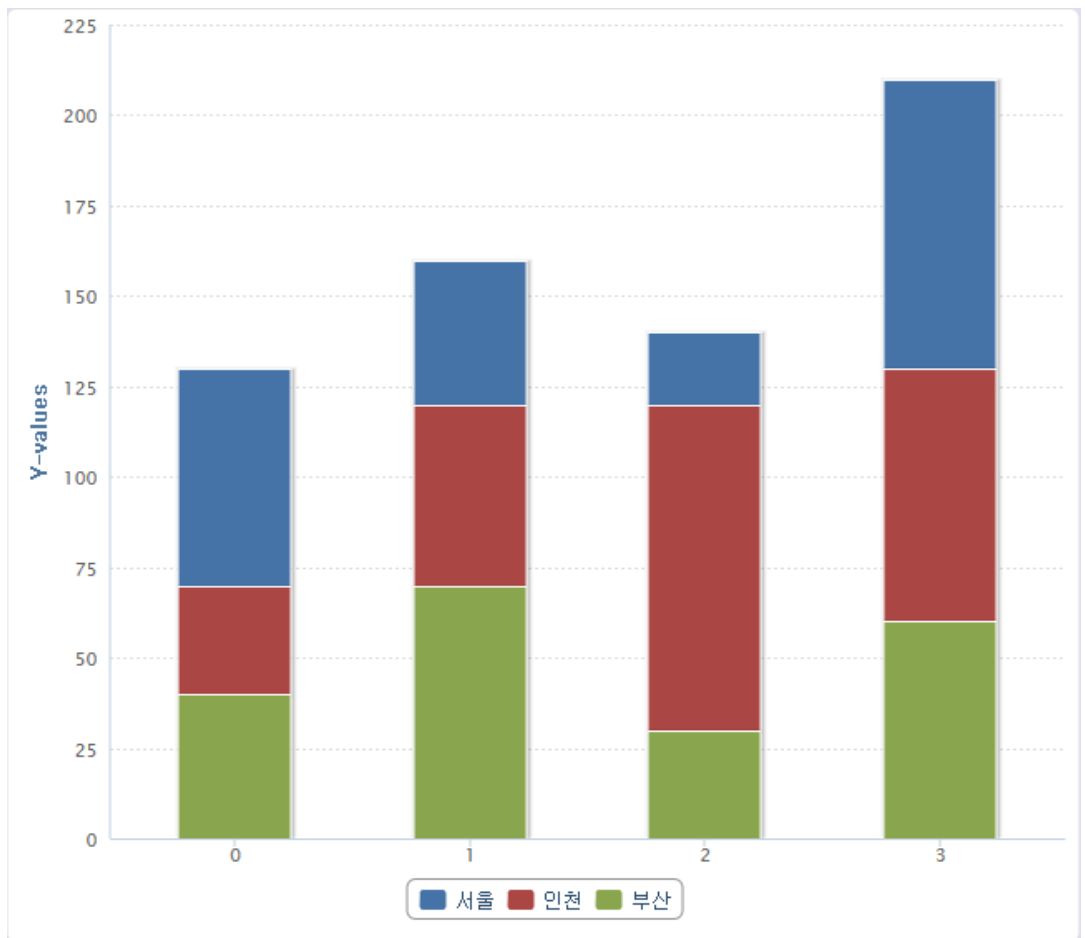
Parameter	Type	필수여부	설 명
StackValue	String	필수	Stacking Style (normal, percent)

➤ Example

```
myChart.SetPlotOptions( {
  column: {
    Stacking : "normal"
  });

myChart.SetSeriesOptions([
  {
    Type:"column", Name : "서울",   Data : [60,40,20,80]
  },{
    Type:"column", Name : "인천",   Data : [30,50,90,70]
  },{
    Type:"column", Name : "부산",   Data : [40,70,30,60]]], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0



StartAngle

➤ 지원타입

pie

➤ 기능

시작지점을 설정한다.

➤ Syntax

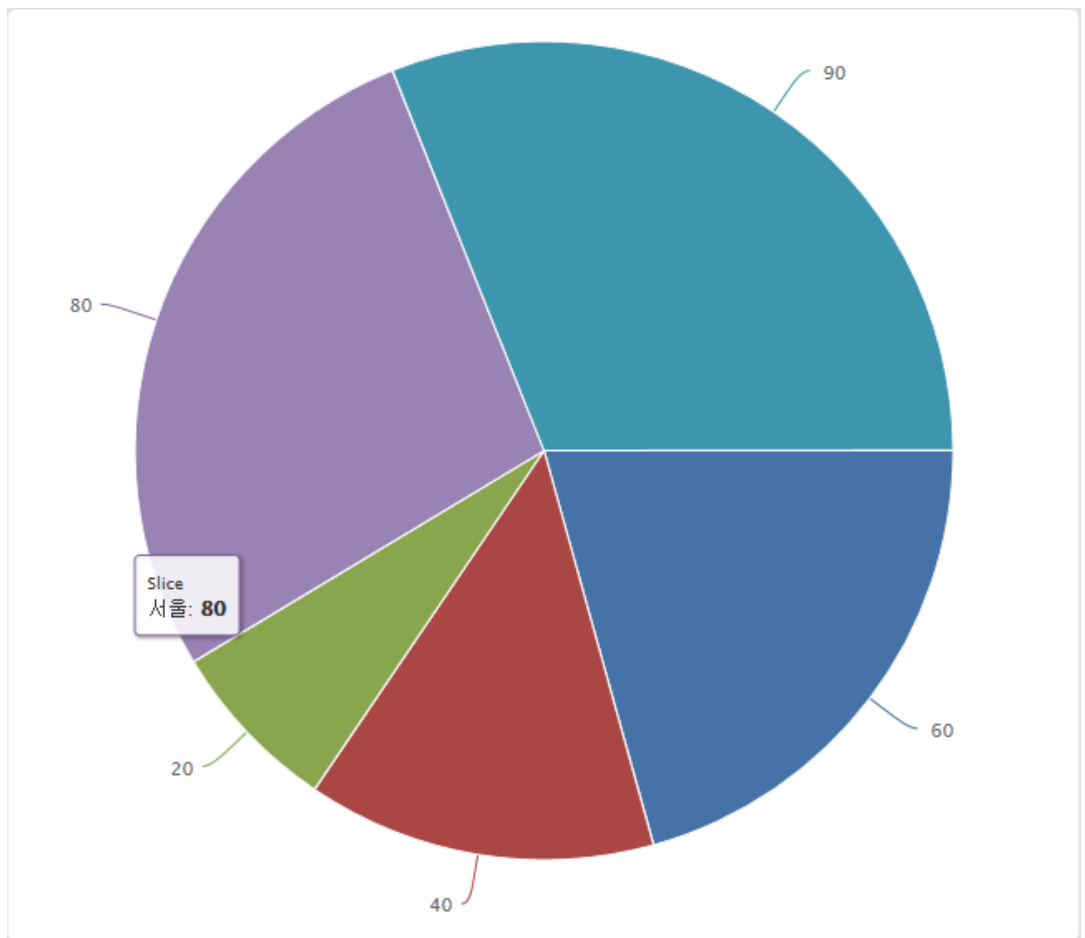
Syntax	StartAngle : AngleValue
--------	--------------------------------

➤ Info

Parameter	Type	필수여부	설 명
AngleValue	Number	필수	StartAngle 값 설정

➤ Example

```
myChart.SetPlotOptions( {  
  pie: {  
    StartAngle:90,  
    DataLabels:{  
      Formatter:function(){  
        return this.y;  
      }  
    }  
  }  
});  
  
myChart.SetSeriesOptions([[  
  Type:"pie", Name : "서울",   Data : [60,40,20,80,90]  
]], 1);  
  
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

Step

➤ 지원타입

line

➤ 기능

라인차트에서만 적용되며, 라인형태를 계단식으로 표현할지 여부를 선택한다.

➤ Syntax

Syntax	Step : StepValue
--------	------------------

➤ Info

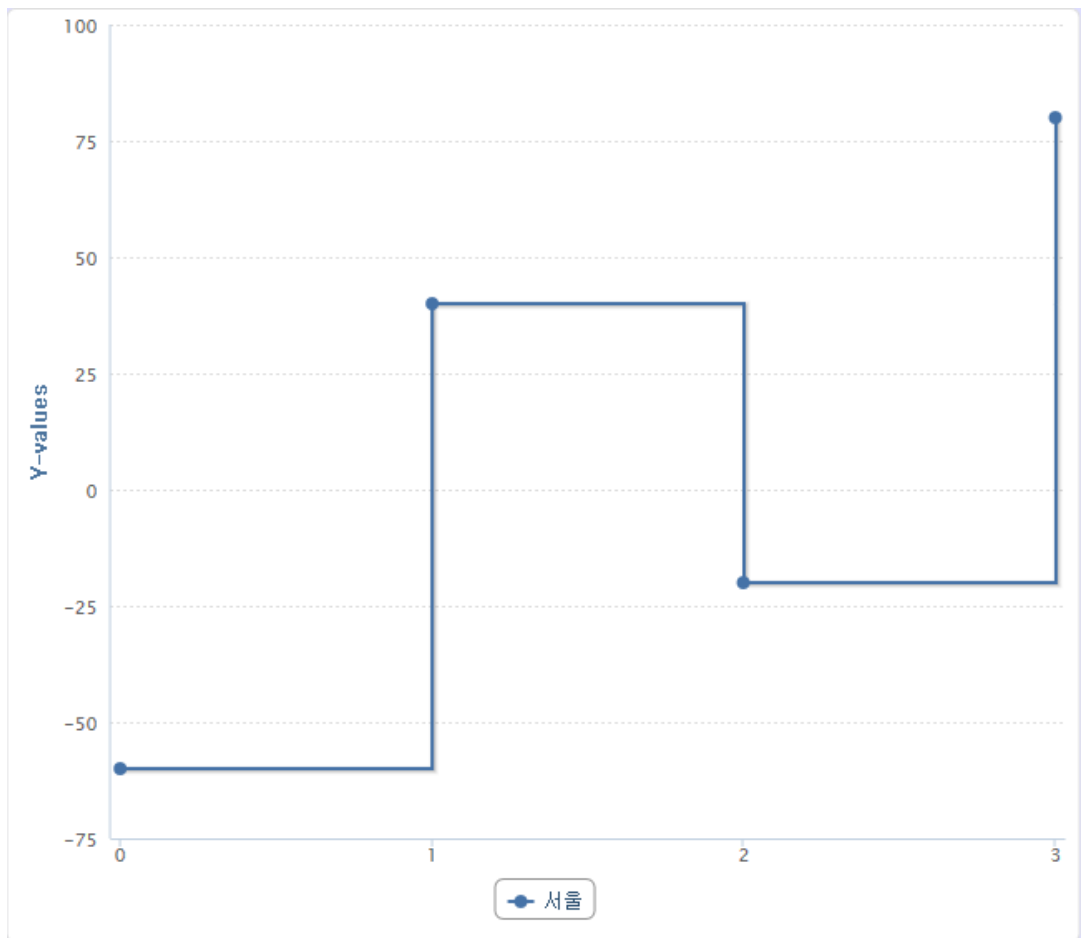
Parameter	Type	필수여부	설 명
StepValue	Boolean	필수	계단식 표현 여부

➤ Example

```
myChart.SetPlotOptions( {
  line: {
    Step:true
  });

myChart.SetSeriesOptions([
  {
    Type:"line", Name : "서울",    Data : [-60,40,-20,80]
  }, 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0

Tooltip

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series, pie

➤ 기능

마우스 오버시 표시되는 툴팁 설정.

전체적인 **SetToolTipOptions** 에서 적용된 부분 외에 시리즈 또는 특정 유형의 차트에 적용하고 싶으부분을 설정한다.

속성은 Json Object 형태로 표현하며, 속성은 아래와 같다.

Parameter	Type	설 명
HeaderFormat	String	툴팁의 헤더를 설정한다. point.key, series.name 등 point와 series의 속성을 사용 할 수 있다. (HTML 가능)
PointFormat	String	포인트의 값을 설정한다. point.x, point.y, series.name 등 속성이 사용가능하고 Prefix, Suffix를 적용 할 수 있다. (HTML 가능)
FooterFormat	String	툴팁 포맷에 마지막 텍스트에 추가된다.
ValuePrefix	String	Point 앞에 추가하는 텍스트
ValueSuffix	String	Point 뒤에 추가하는 텍스트

➤ Syntax

Syntax	Tooltip : TooltipObject
--------	--------------------------------

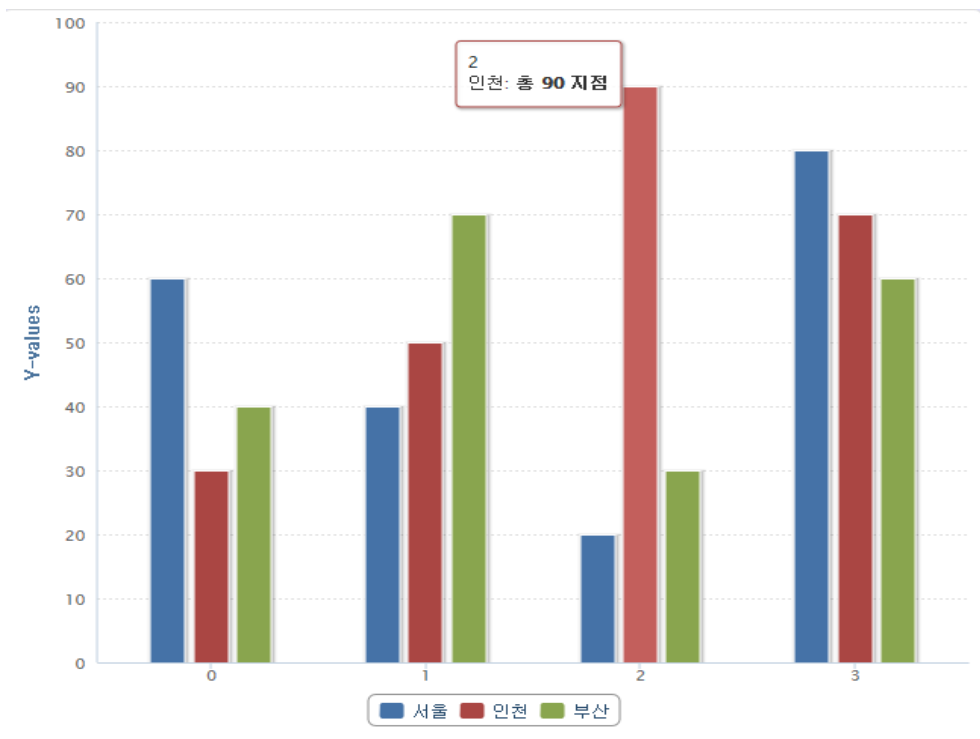
➤ Info

Parameter	Type	필수여부	설 명
TooltipObject	Object	필수	툴팁 속성 설정 Json Object

➤ Example

```

myChart.SetPlotOptions( {
  column: {
    Tooltip : {
      HeaderFormat: '<small>{point.key}</small><br><table>',
      PointFormat: '{series.name}: <b>{point.y}</b><br></table>',
      ValueSuffix : ' 지점',
      ValuePrefix : '총 '}});
myChart.SetSeriesOptions([
  Type:"column", Name : "서울",   Data : [60,40,20,80]],{
  Type:"column", Name : "인천",   Data : [30,50,90,70]],{
  Type:"column", Name : "부산",   Data : [40,70,30,60]], 1);
myChart.Draw();
  
```



➤ 제공 버전

7.0.0.0

Visible

➤ 지원타입

area, areaspline, line, scatter, spline, bar, column, arearange, areasplinerange, gauge, series, pie

➤ 기능

화면에 표시 여부를 설정한다.

➤ Syntax

Syntax	Visible : VisibleValue
--------	-------------------------------

➤ Info

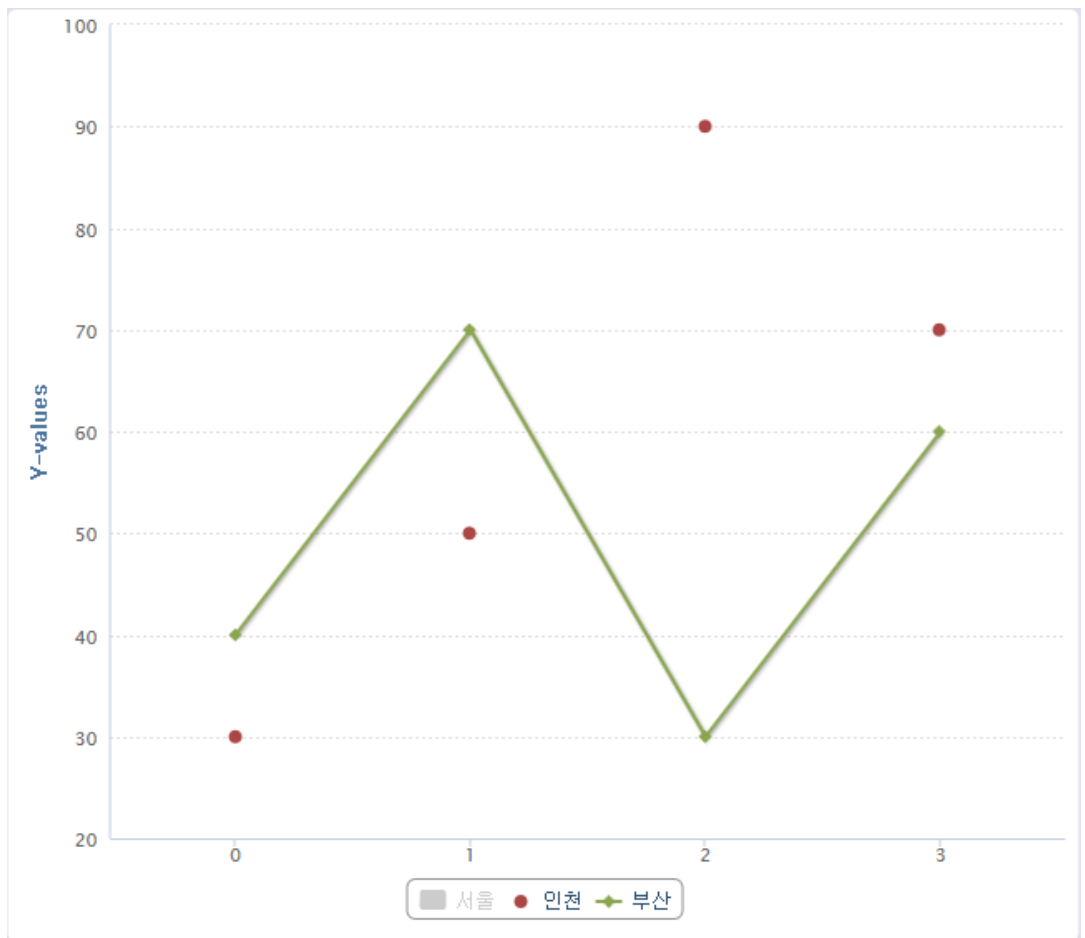
Parameter	Type	필수여부	설 명
VisibleValue	Boolean	필수	표시 여부 설정

➤ Example

```
myChart.SetPlotOptions( {
column: {
Visible:false //컬럼 차트 생략
}});

myChart.SetSeriesOptions([
{
Type:"column", Name : "서울", Data : [60,40,20,80]
},{
Type:"scatter", Name : "인천", Data : [30,50,90,70]
},{
Type:"line", Name : "부산", Data : [40,70,30,60]
}], 1);

myChart.Draw();
```



➤ 제공 버전
7.0.0.0



Appendix



1. 특수차트

1.1 Angular gauges chart

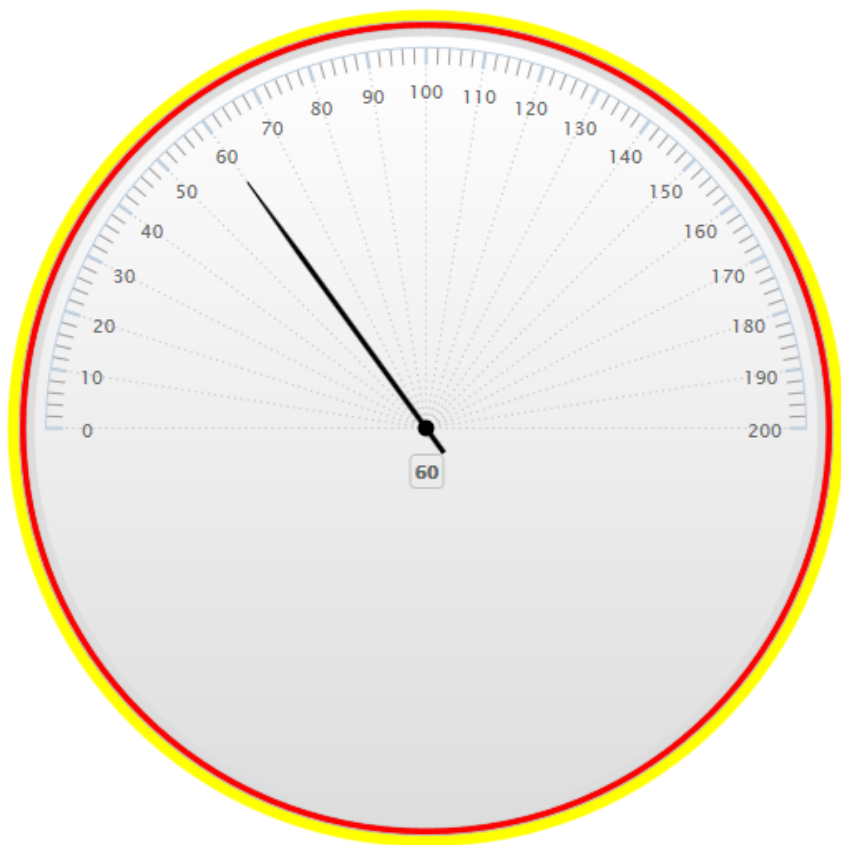
➤ 기능

Gauge 차트를 표현한다.

➤ Example

```
myChart.SetPaneOptions({
    StartAngle: -90,
    EndAngle: 90,
    Background: [{
        BackgroundColor:"yellow",
        BorderWidth: 0,
        OuterRadius: '110%'
    }, {
        BackgroundColor:"red",
        BorderWidth: 1,
        OuterRadius: '107%'
    }, {
        // default background
    }, {
        BackgroundColor: '#DDD',
        BorderWidth: 0,
        OuterRadius: '105%',
        InnerRadius: '103%'
    }
]);
myChart.SetBaseOptions({
    Type:"gauge"
});
myChart.SetYAxisOptions({
    Max:200,
    Min:0,
```

```
TickInterval : 10 // Tick 간격을 10으로 설정
}, 1);
myChart.SetSeriesOptions([
    Data : [
        {Y:60}
    ]
}], 1);
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

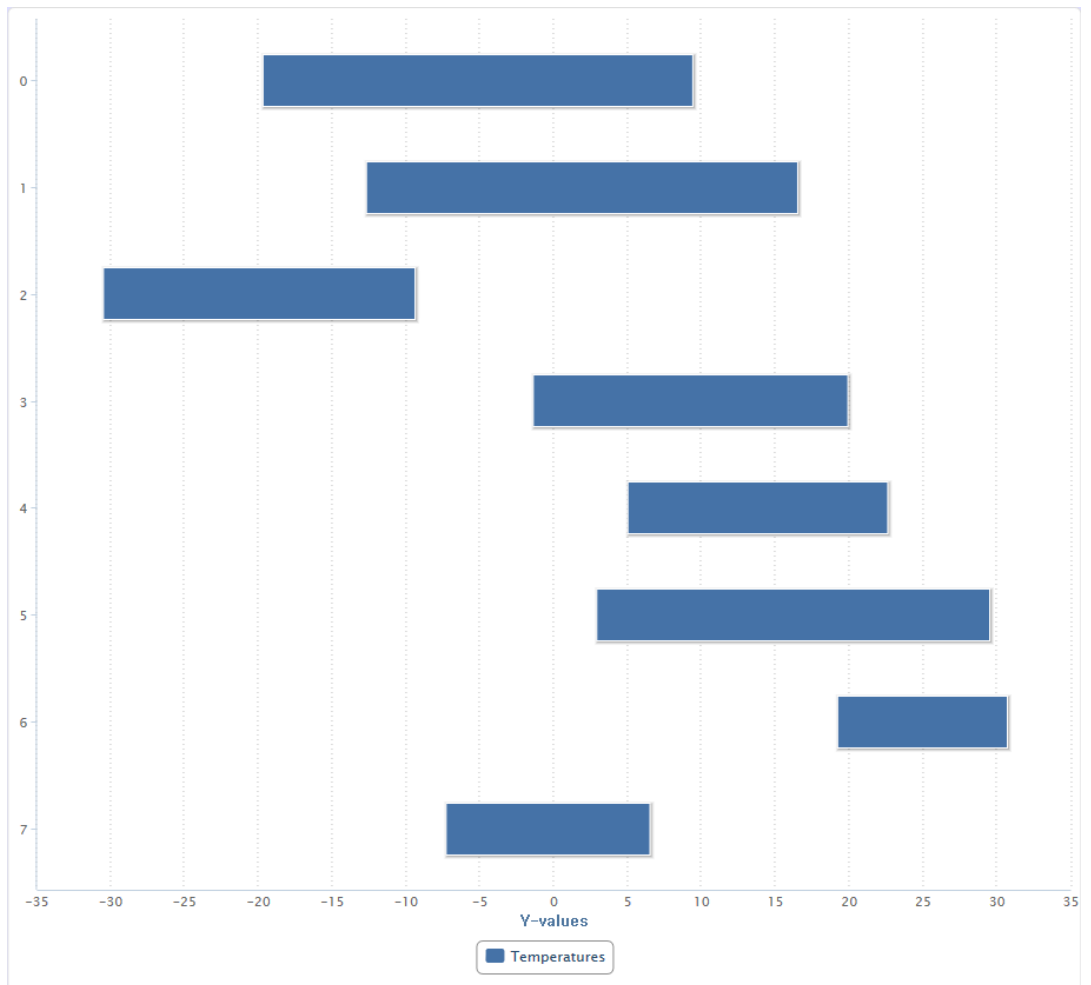
1.2 ColumnRange chart

➤ 기능

ColumnRange 차트를 표현한다.

➤ Example

```
myChart.SetOptions({
  chart:{
    Type:'columnrange',
    Inverted:true
  }
});
myChart.SetSeriesOptions([{
  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]
  ]
}]);
myChart.Draw();
```

➤ 제공 버전
7.0.0.0

1.3 ErrorBar chart

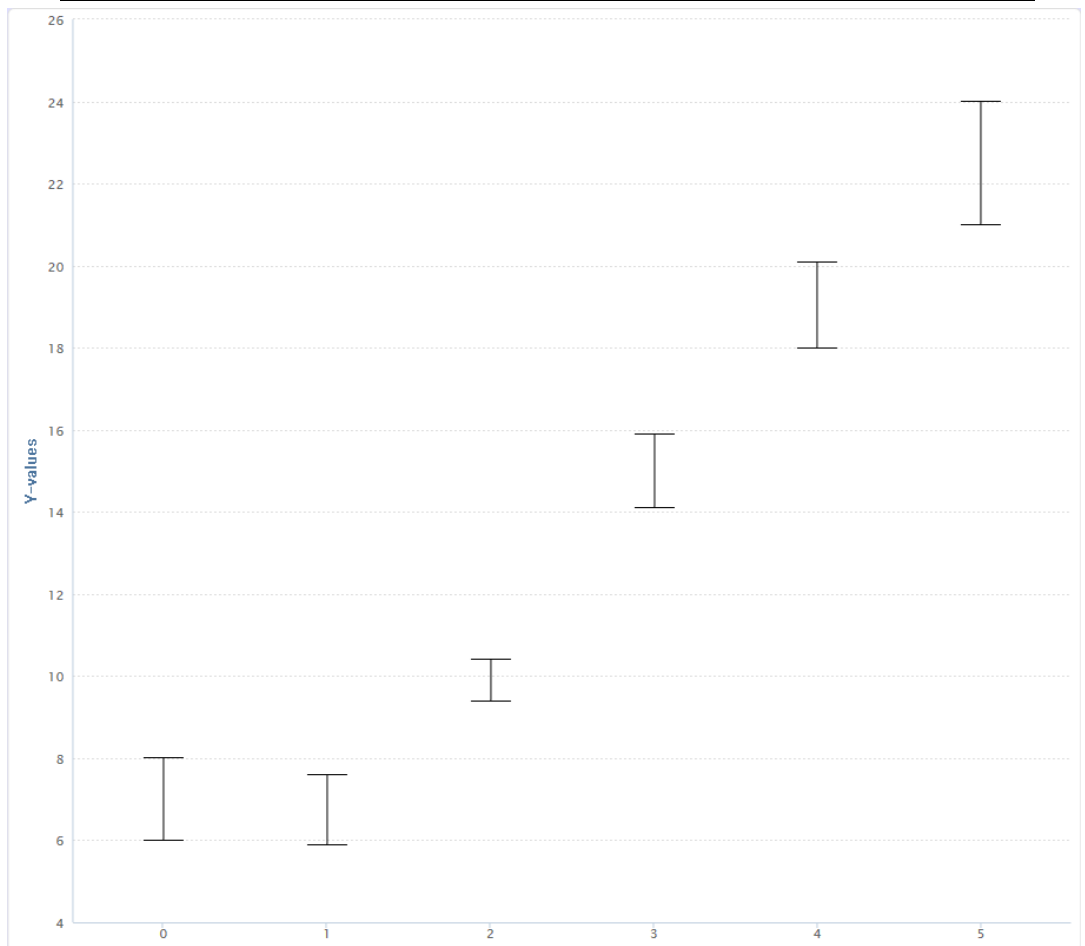
➤ 기능

ErrorBar 차트를 표현한다.

➤ Example

```

myChart.SetSeriesOptions([
    type: 'errorbar',
    data: [[6, 8], [5.9, 7.6], [9.4, 10.4], [14.1, 15.9], [18.0, 20.1], [21.0, 24.0]]]);
myChart.Draw();
    
```



➤ 제공 버전

7.0.0.0

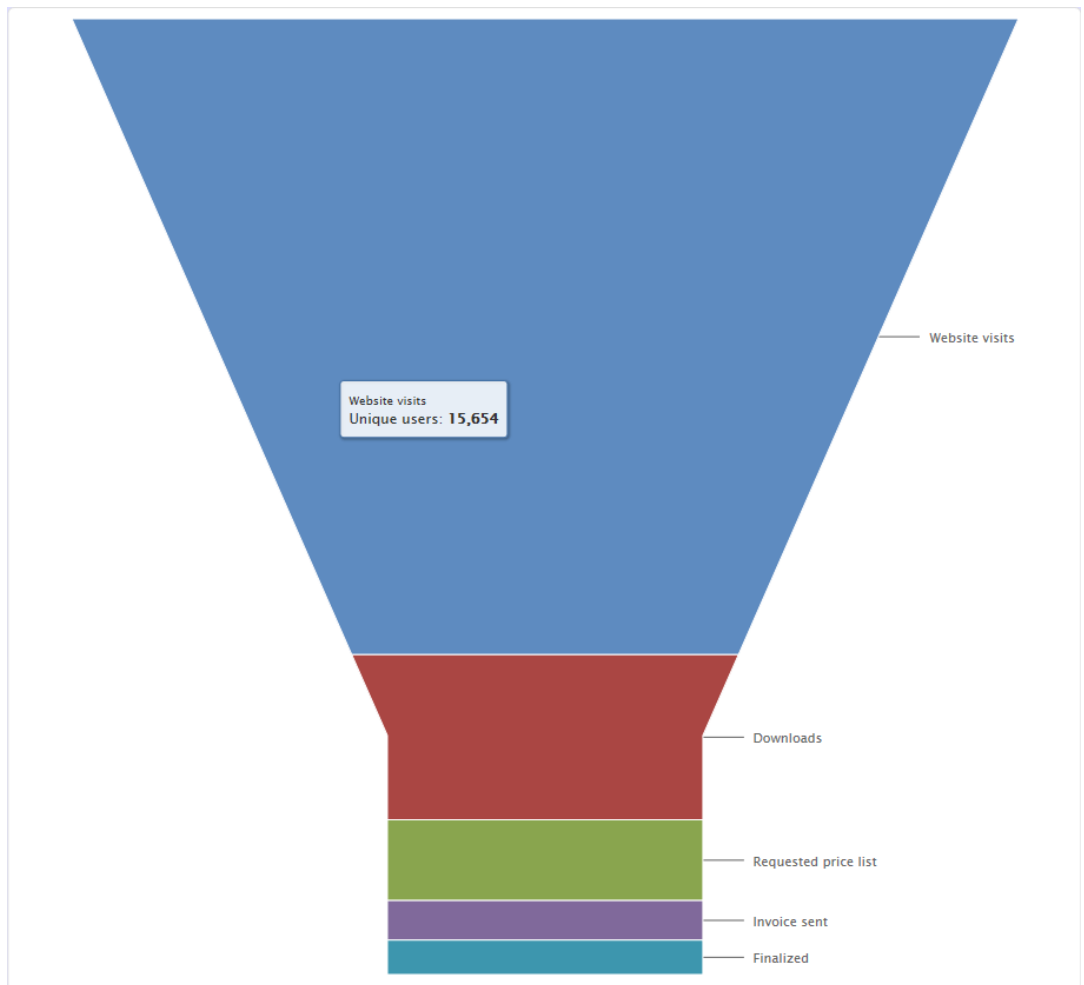
1.4 Funnel chart

➤ 기능

깔때기형 차트를 표현한다.

➤ Example

```
myChart.SetOptions({
  chart:{
    Type:"funnel"
  }
});
myChart.SetSeriesOptions([{
  name: 'Unique users',
  data: [
    ['Website visits', 15654],
    ['Downloads', 4064],
    ['Requested price list', 1987],
    ['Invoice sent', 976],
    ['Finalized', 846]
  ]
}]);
myChart.Draw();
```



- 제공 버전
7.0.0.0

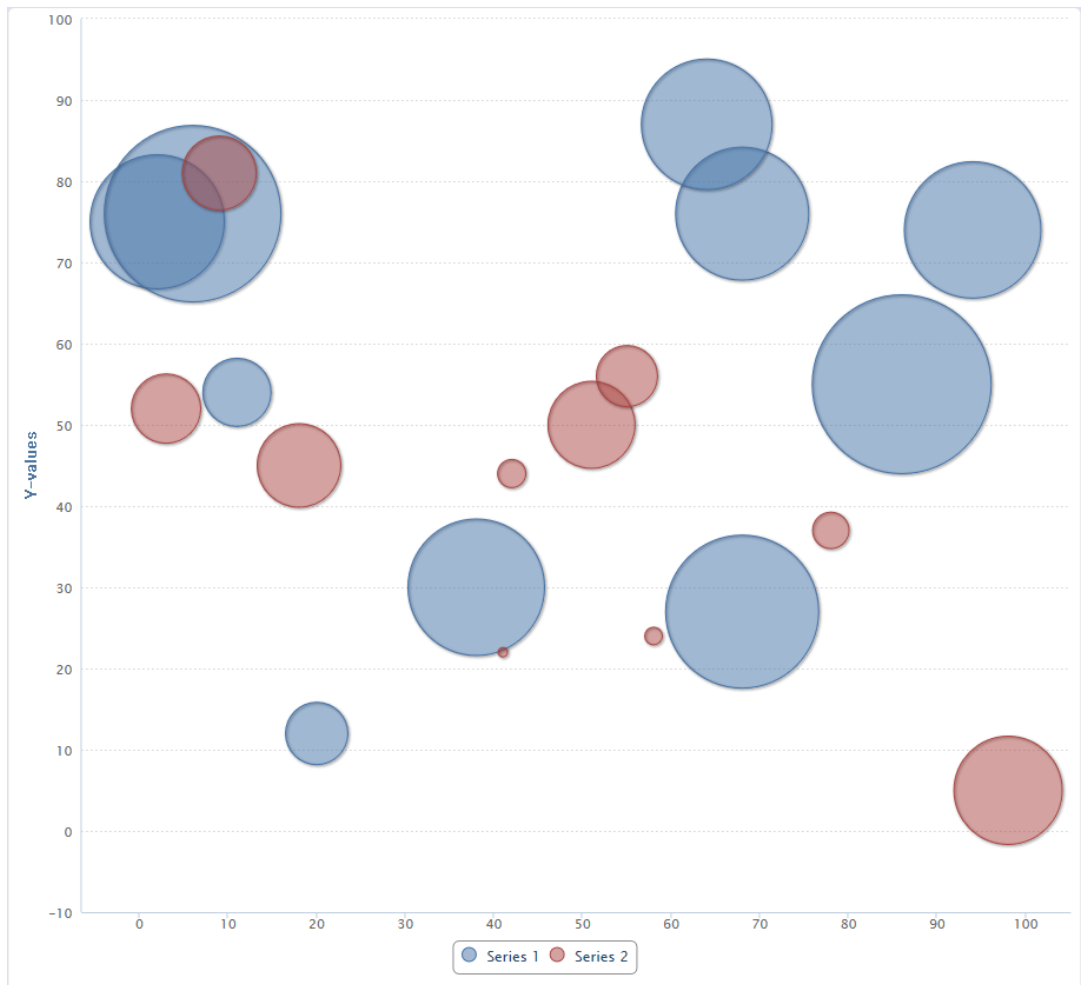
1.5 Bubble chart

➤ 기능

거품형 차트를 표현한다.

➤ Example

```
myChart.SetOptions({
  chart:{
    Type:"bubble"
  }
});
myChart.SetSeriesOptions([[
  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]]
}]);
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

1.6 HeapMap chart

➤ 기능

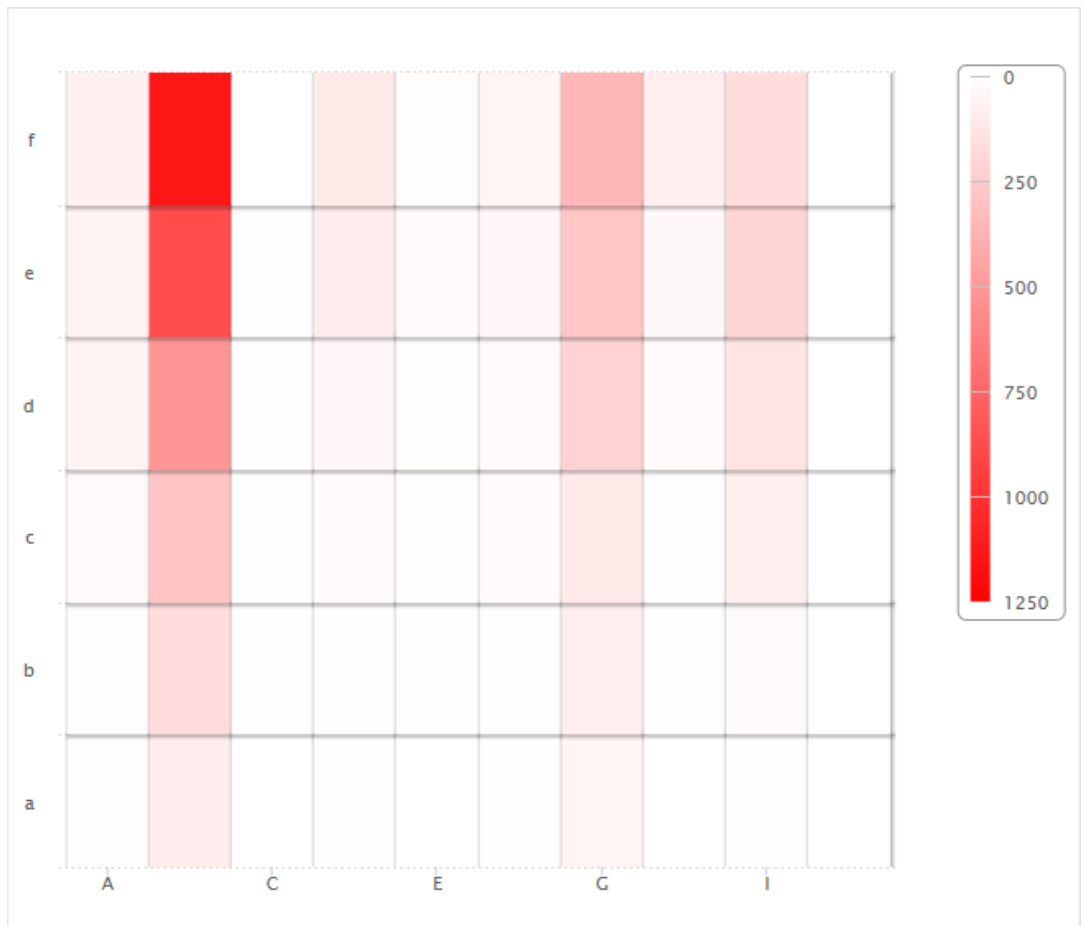
히트맵형 차트를 표현한다.

➤ 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
}
});
myChart.SetSeriesOptions([
    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]]
]);
myChart.Draw();
```

➤ 제공 버전
7.0.0.0

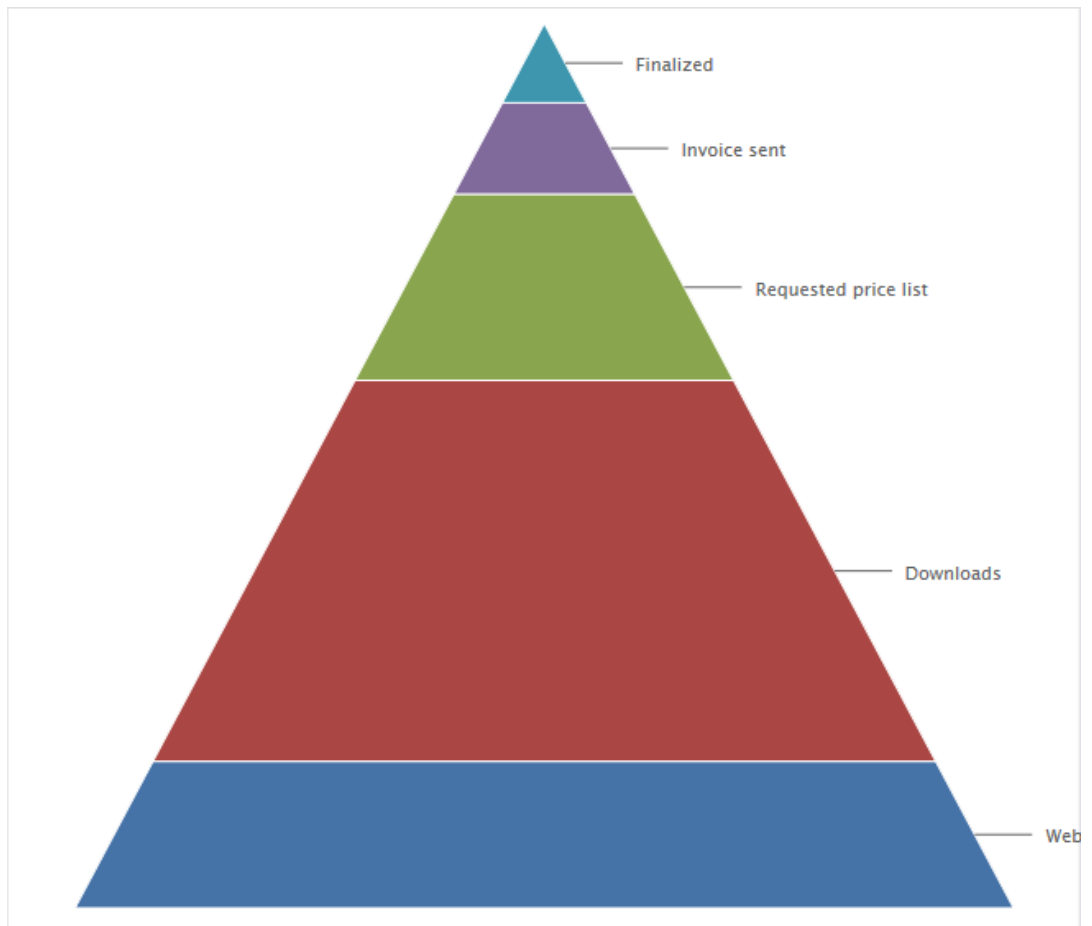
1.7 Prymid chart

➤ 기능

피라미드형 차트를 표현한다.

➤ Example

```
myChart.SetOptions({
  chart:{
    Type:"pyramid"
  }
});
myChart.SetSeriesOptions([{
  name: 'Unique users',
  data: [
    ['Website visits', 1565],
    ['Downloads', 4064],
    ['Requested price list', 1987],
    ['Invoice sent', 976],
    ['Finalized', 846]
  ]
}]);
myChart.Draw();
```



- 제공 버전
7.0.0.0



2. 차트기능

2.1 Stack 의 그룹기능

➤ 기능

Stacking 기능을 이용하면서 그룹을 지어 그룹별로 Stacking 기능을 이용 할 수 있다.

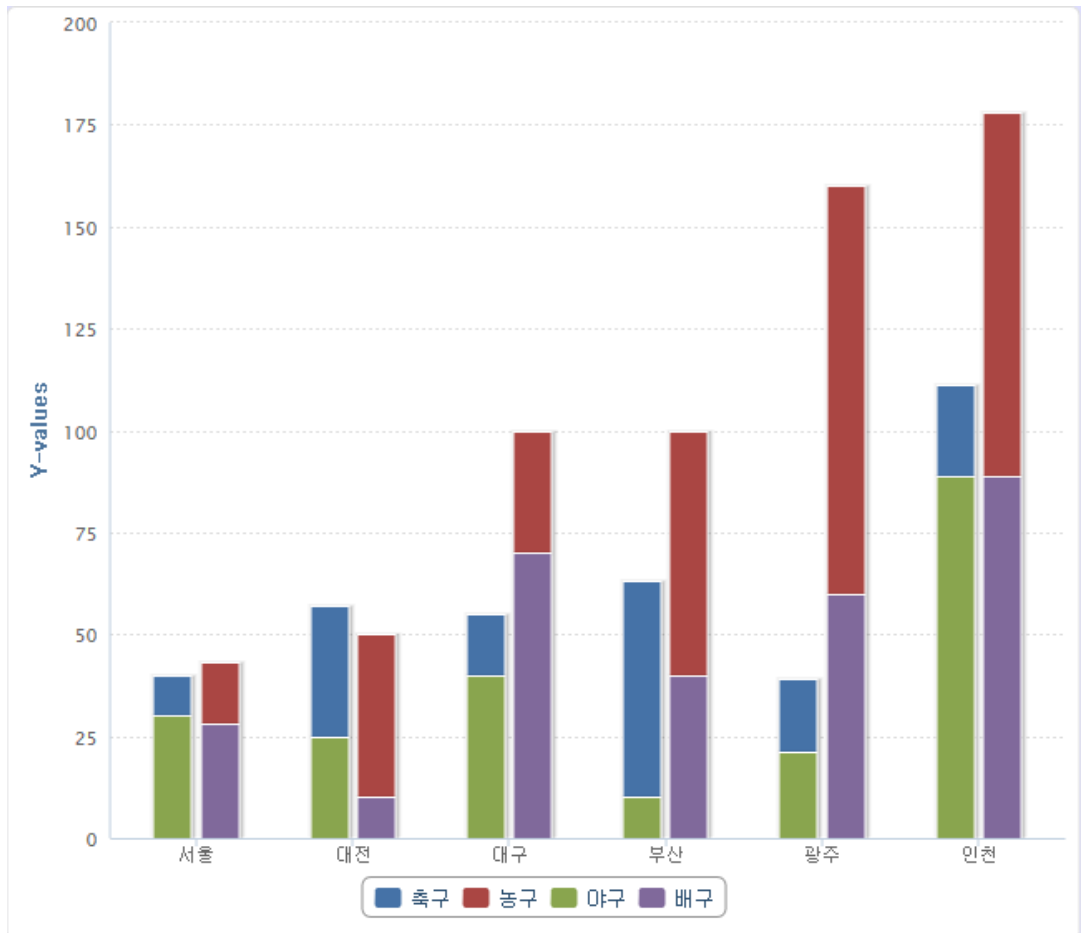
➤ Example

```
myChart.SetSeriesOptions([
    {
        Name : "축구",
        Stack:"summer", //Stacking 기능에서 Stack 이름을 주어 그룹을 형성
        Data : [ {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22} ]
    },
    {
        Name : "농구",
        Stack:"winter",
        Data : [ {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89} ]
    },
    {
        Name : "야구",
        Stack:"summer",
        Data : [ {Y:30},{Y:25},{Y:40},{Y:10},{Y:21},{Y:89} ]
    },
    {
        Name : "배구",
        Stack:"winter",
        Data : [ {Y:28},{Y:10},{Y:70},{Y:40},{Y:60},{Y:89} ]
    }
]), 1);

myChart.SetPlotOptions({
    Column:{
        Stacking:true
    }
});

myChart.SetXAxisOptions({
    Categories:["서울","대전","대구","부산","광주","인천"]
});
```

```
});  
myChart.SetBaseOptions({  
    Type:"column"  
});  
myChart.Draw();
```



➤ 제공 버전
7.0.0.0

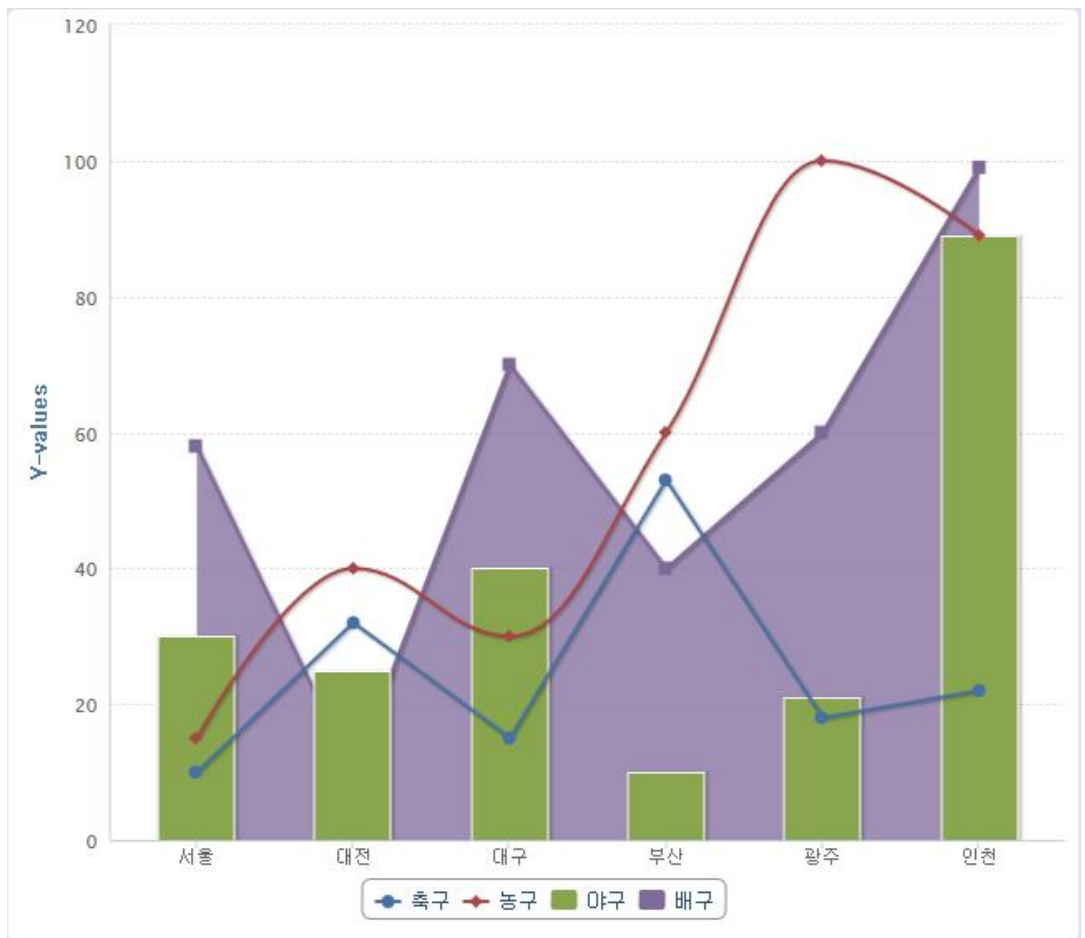
2.2 멀티 차트 표현 기능

➤ 기능

여러가지의 형태를 한 차트에 표현 할 수 있다.

➤ Example

```
myChart.SetSeriesOptions([{\n    Name : "축구",\n    Type:"line",\n    zIndex:3,\n    Data : [ {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22} ]\n},{\n    Name : "농구",\n    Type:"spline",\n    zIndex:4,\n    Data : [ {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89} ]\n},{\n    Name : "야구",\n    Type:"column",\n    zIndex:2,\n    Data : [ {Y:30},{Y:25},{Y:40},{Y:10},{Y:21},{Y:89} ]\n},{\n    Name : "배구",\n    Type:"area",\n    zIndex:1,\n    Data : [ {Y:58},{Y:10},{Y:70},{Y:40},{Y:60},{Y:99} ]\n}], 1);\n\nmyChart.Draw();
```



➤ 제공 버전
7.0.0.0

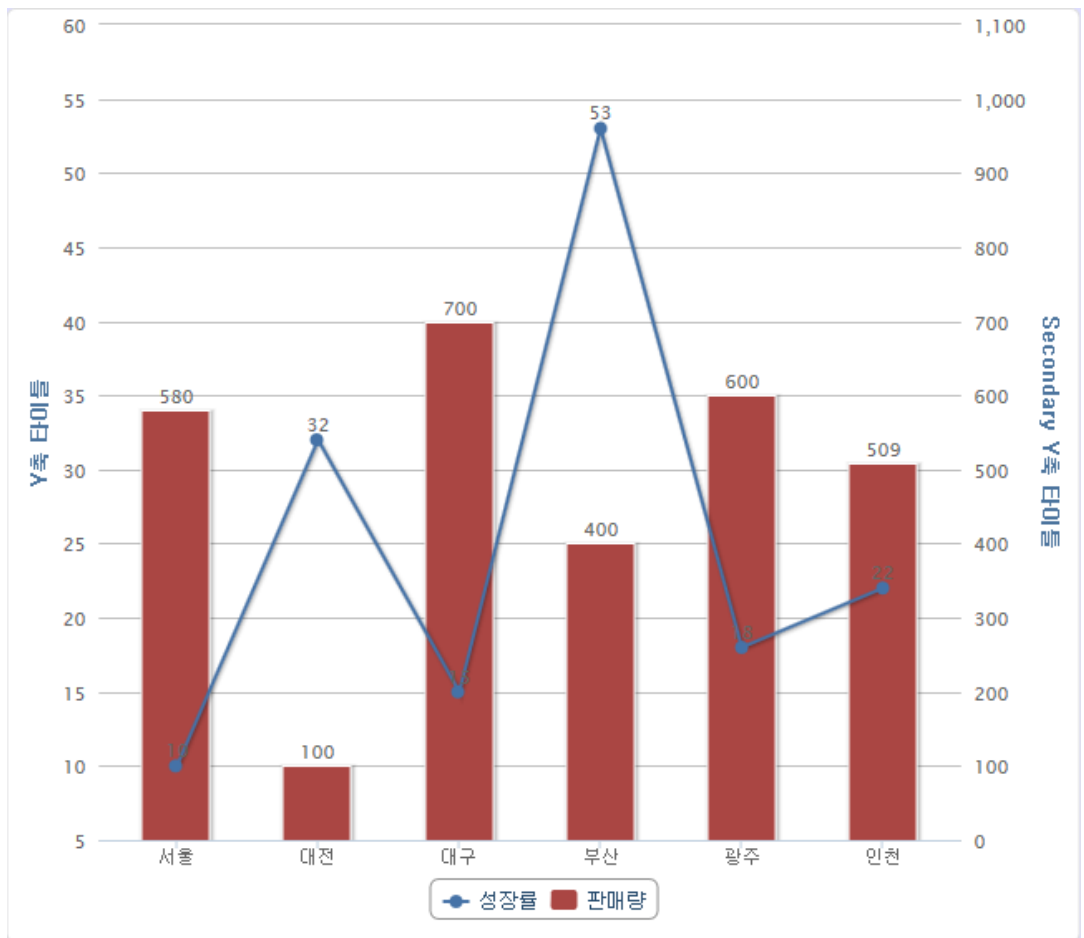
2.3 Secondary Y 축 사용

➤ 기능

두개의 Y 축을 사용한다.

➤ Example

```
myChart.SetSeriesOptions([{\n    Name : "성장률",\n    Type:"line", zIndex:3,\n    Data : [ {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}      ],\n    DataLabels:{ Enabled:true },\n},{\n    Name : "판매량",\n    Type:"column", zIndex:1,\n    Data : [ {Y:580},{Y:100},{Y:700},{Y:400},{Y:600},{Y:509}      ],\n    DataLabels:{ Enabled:true },\n    YAxis:1\n}], 1);\nmyChart.SetYAxisOptions([{\n    Title:{\n        Text:"Y축 타이틀"\n    },\n    Opposite:0\n},{\n    Title:{\n        Text:"Secondary Y축 타이틀"\n    },\n    Opposite:1\n}]);\nmyChart.Draw();
```



➤ 제공 버전

7.0.0.0

2.4 PlotLine 기능

➤ 기능

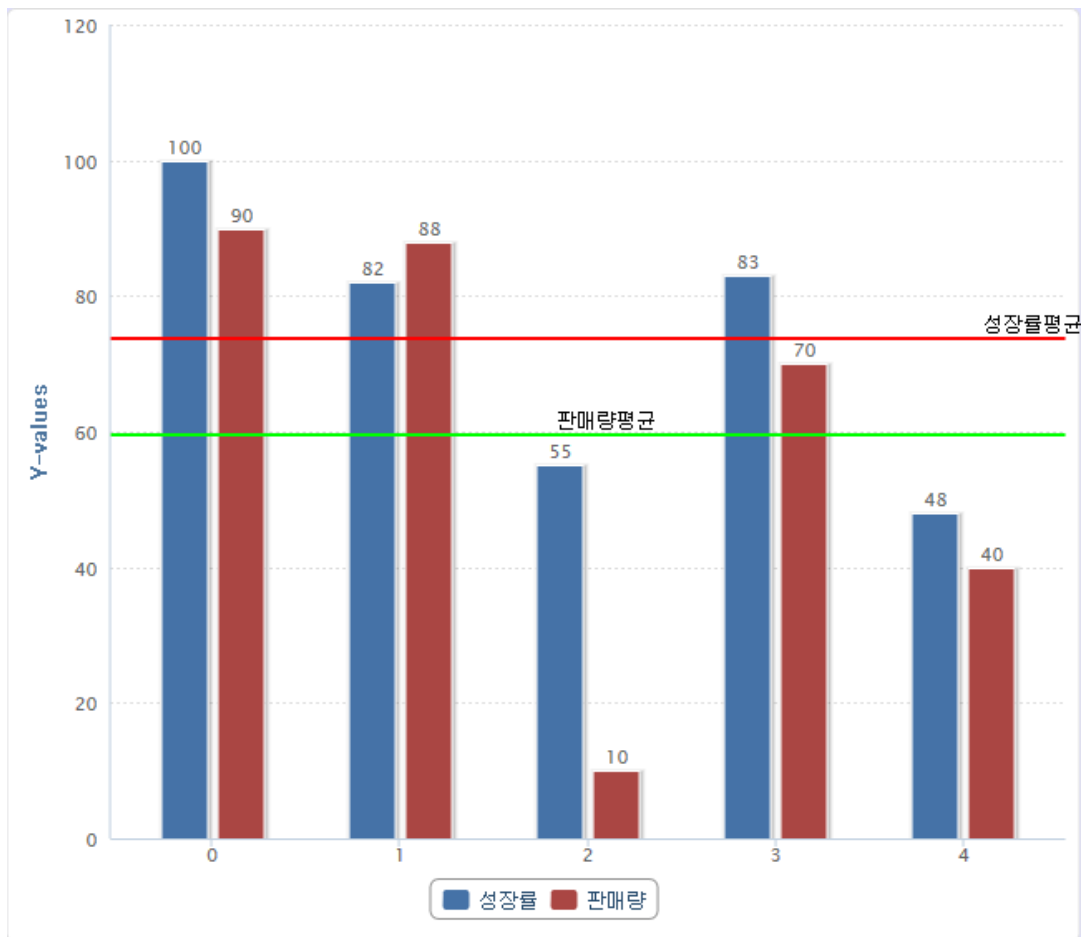
차트 중간에 선을 표현 할수 있다.

➤ Example

```
myChart.SetSeriesOptions([{  
    Name : "성장률",  
    Type:"column",  
    Data : [ {Y:100},{Y:82},{Y:55},{Y:83},{Y:48}],  
    DataLabels:{ Enabled:true }  
}],{  
    Name : "판매량",  
    Type:"column",  
    Data:[{Y:90},{Y:88},{Y:10},{Y:70},{Y:40}],  
    DataLabels:{ Enabled:true },  
}], 1);  
myChart.SetYAxisOptions({  
    PlotLines : [{ //plotline 옵션 설정  
        color: '#FF0000',  
        Value:73.6,  
        Width:2,  
        Label:{  
            Text:"성장률평균",  
            Align:"right"  
        },  
        ZIndex:10  
    }],  
    {  
        color: '#00FF00',  
        Value:59.6,  
        Width:2,  
        Label:{
```



```
Text:"판매량평균",  
Align:"center"  
},  
ZIndex:10  
}  
});  
myChart.Draw();
```



➤ 제공 버전

7.0.0.0

2.5 3D 차트 기능

➤ 기능

막대(Column), 가로막대(Bar), 파이(Pie) 차트를 3D 형태로 표현 한다.

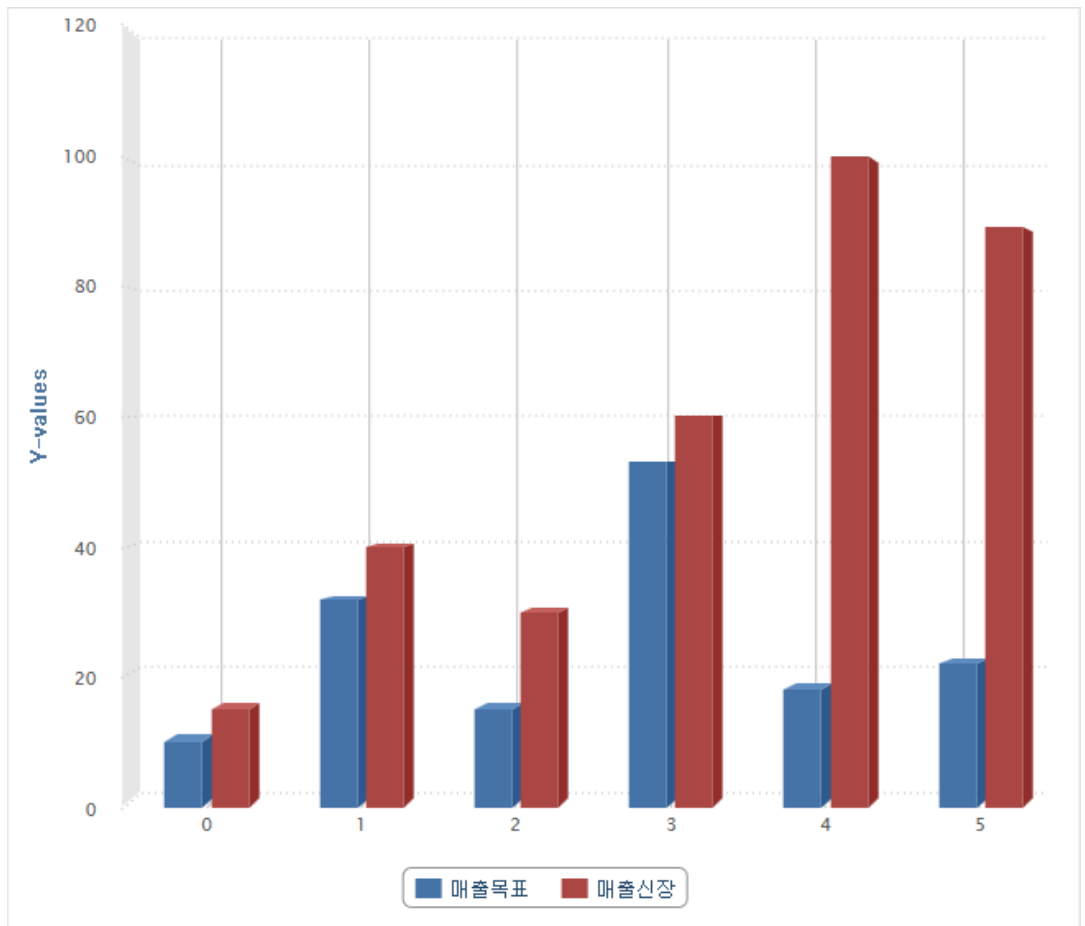
설정은 SetOptions 메소드의 Chart 속성 하위 Options3d 속성을 이용하여 3D 표현 여부를 설정 한다.

Parameter	Type	설 명
Enabled	Boolean	3D 표현 여부
Depth	Number	차트의 3D 깊이

2.5.1 막대(Column) 3D 차트

➤ **Example**

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"column",
    Options3d: {
      Enabled : true,
      Depth : 40
    }
  }
});
//차트 Series 설정
myChart.SetSeriesOptions([
  {
    Name : "매출목표",
    Data : [
      {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
    ]
  },
  {
    Name : "매출신장",
    Data : [
      {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89}
    ]
  }
], 1);
myChart.Draw();
```



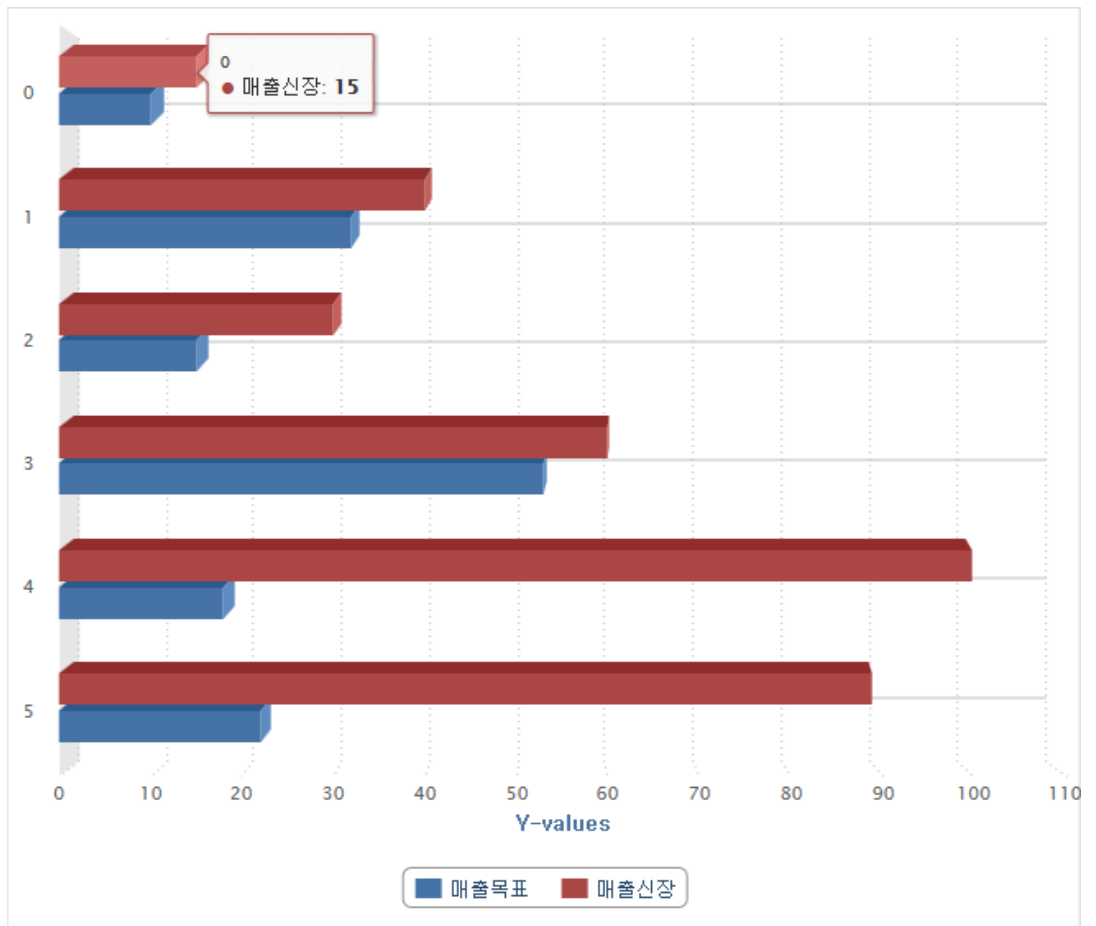
➤ 제공 버전

7.2.0.3

2.5.2 가로막대(Bar) 3D 차트

➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"bar",
    Options3d: {
      Enabled : true,
      Depth : 40
    }
  }
});
//차트 Series 설정
myChart.SetSeriesOptions([
  {
    Name : "매출목표",
    Data : [
      {Y:10},{Y:32},{Y:15},{Y:53},{Y:18},{Y:22}
    ]
  },{
    Name : "매출신장",
    Data : [
      {Y:15},{Y:40},{Y:30},{Y:60},{Y:100},{Y:89}
    ]
  }
], 1);
myChart.Draw();
```

➤ 제공 버전

7.2.0.3

2.5.3 파이(Pie) 3D 차트

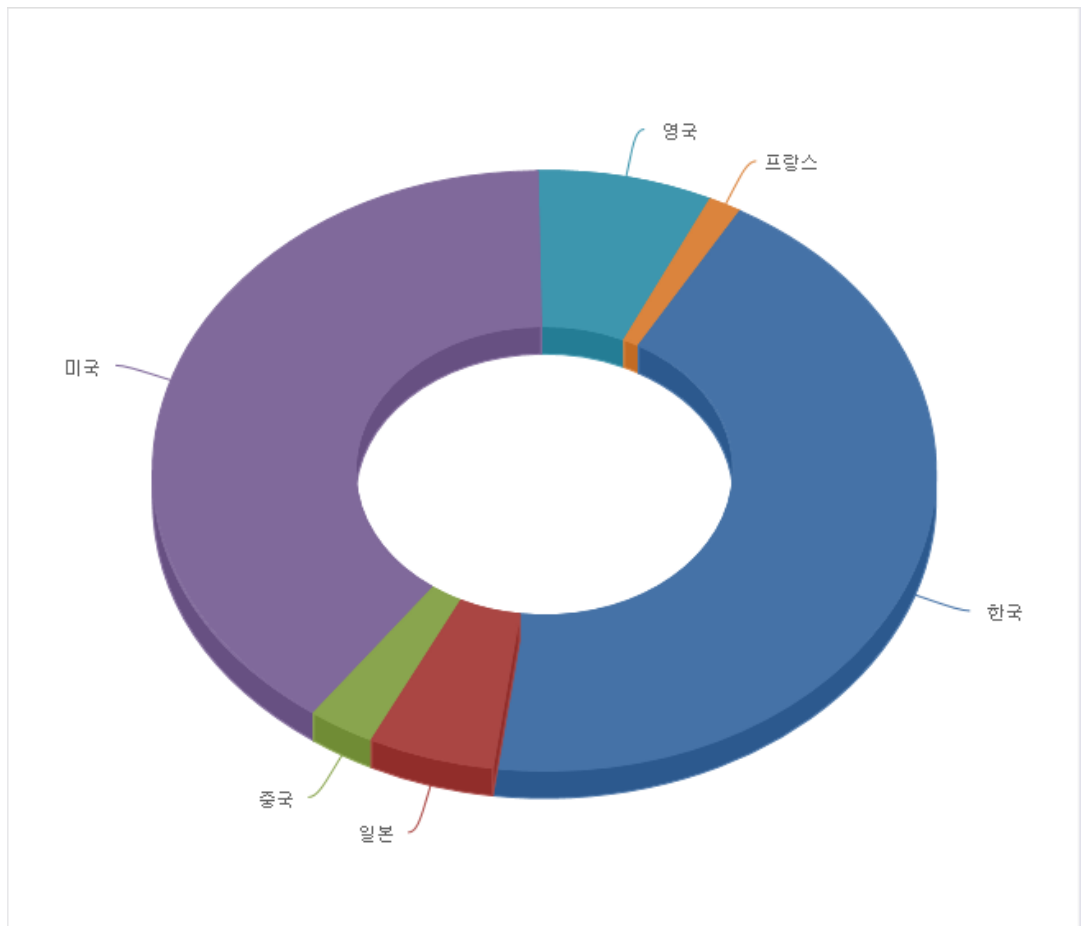
➤ Example

```
//차트속성
myChart.SetOptions({
  Chart:{
    Type:"pie",
    Options3d: {
      Enabled : true,
      Alpha : 40
    }
  }
});

//차트 PlotOptions 속성
myChart.SetPlotOptions({
  Pie:{
    InnerSize:"23",
    StartAngle:30,
    Depth : 35
  }
});

//차트 Series 설정
myChart.SetSeriesOptions([
  Data : [
    ["한국", 636], ["일본", 75], ["중국", 40],
    ["미국", 577], ["영국", 105], ["프랑스", 20]
  ]
], 1);

myChart.Draw();
```



➤ 제공 버전
7.2.0.3

3. IBSheet 데이터 연동

IBSheet 에 조회된 데이터를 쉽게 차트로 표현할 수 있는 기능으로 `SetOptions`

메소드의 `Data` 속성을 이용하여 설정 한다

기본적으로 시트에서의 컬럼은 시리즈, 행은 데이터로 표현된다.

따라서 `HeaderRow` 에 설정된 행의 셀 값은 시리즈 명으로, `HeaderCol` 에 설정된 컬럼의 셀 값은 X 축 명으로 처리 된다.

`SwitchRowCol` 속성을 `true` 로 설정되면 `Row` 와 `Col` 이 스위치되어 처리된다.

`Rows` 와 `Cols` 속성 설정시 `Index` 와 `Index` 의 범위를 설정할 수 있는데, 여기서 `Index` 의 범위는 "-" 이용하여 설정 한다. 예를들어 `Rows` 속성 설정시 `Index` 가 1 인 행과 5 부터 10 까지의 행을 설정하고자 하는 경우 `Rows : "1|5-10"` 과 같이 설정 한다.

Parameter	Type	설 명
Sheet	String	대상 시트의 Id
HeaderRow	Number	헤더로 설정할 행의 Index (Default : 헤더의 마지막 행)
HeaderCol	Number/ String	헤더로 설정할 컬럼의 Index 또는 SaveName (Default : 첫번째 컬럼의 Index)
Rows	String	대상 행의 Index 또는 Index의 범위를 구분자 " "로 연결한 문자열 (Default : 모든 데이터 행)
Cols	String	대상 컬럼의 Index, SaveName 또는 Index의 범위를 구분자 " "로 연결한 문자열 (Default : 모든 컬럼)
SwitchRowCol	Boolean	컬럼과 행의 크로스 처리 여부 (Default : false)

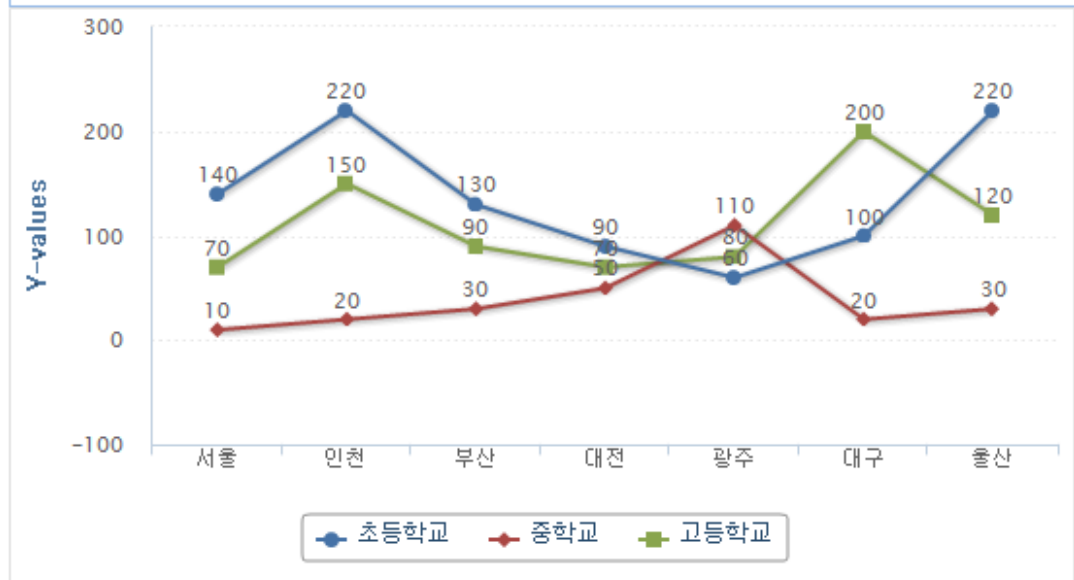


➤ **Example**

```
myChart.SetOptions({
    Data:{
        Sheet:"sheet",
        HeaderCol:0,
        Rows:"2-4",
        Cols:"1-7",
        SwitchRowCol:true
    },
    XAxis:{
        Categories:true    //X축 라벨
    },
    PlotOptions:{
        Series:{
            DataLabels:{
                Enabled:true    //데이터값 표시
            }
        }
    }
});
myChart.Draw();
```



구분	서울	인천	부산	대전	광주	대구	울산
유치원	100	120	130	150	160	200	300
초등학교	140	220	130	90	60	100	220
중학교	10	20	30	50	110	20	30
고등학교	70	150	90	70	80	200	120
기타	1,000	430	230	550	150	1,220	330



➤ 제공 버전

7.2.0.3