

# DXChart 개발자 튜토리얼

## (Nexacro용)

---

## 목 차

<b>1. 처음 시작하기</b>	4
(1) Nexacro Project 만들기	4
(2) 차트모듈 설치하기	6
(3) 샘플화면 만들기	9
(4) 차트함수 만들기	13
<b>2. 차트 구성요소</b>	16
(1) 구성요소 용어 설명	16
(2) 데이터 형식	19
1) data Property	19
2) binddataset Property	19
(3) 차트 주요 Properties	20
1) Margin (옵션 - margin 그룹)	20
2) 제목 (옵션 - title 그룹)	23
3) 제목(옵션 -Subtitle)	25
4) X 축 (옵션 - xaxis 그룹)	27
5) Y 축 (옵션 - yaxis 그룹)	37
6) Background 설정하기	50
7) Label 설정하기	55
8) 범례 설정하기	60
9) 툴팁 설정하기	62
<b>3. Effect 설정하기</b>	70
1) Draw	70
2) drawAni	70
3) wave	70

4)	grow.....	71
5)	trace.....	71
6)	unfold (toBottom).....	72
7)	unfoldFromCenterTrace (fromCenter) .....	72
8)	foldtocenter (toCenter) .....	73
10)	implode .....	74
11)	explode.....	74
차트 별 effect 종류 .....		75
<b>4.</b>	<b>Multi Chart 생성하기</b> .....	78
1)	Bar 및 Line 차트 .....	78
2)	Bar 및 Pie 차트.....	81
<b>5.</b>	<b>기타기능</b> .....	83
1)	ConvertDataset .....	83
2)	datachangeevent.....	84

# 1. 처음 시작하기

## (1) Nexacro Project 만들기

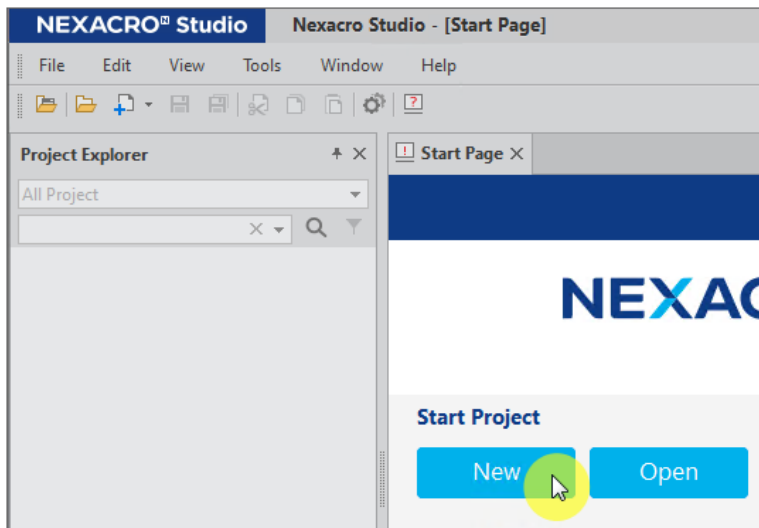
차트를 사용하기 위한 테스트 프로젝트를 생성합니다.

넥사크로스튜디오가 설치되지 않은 경우에는 설치를 먼저 진행해야 합니다.

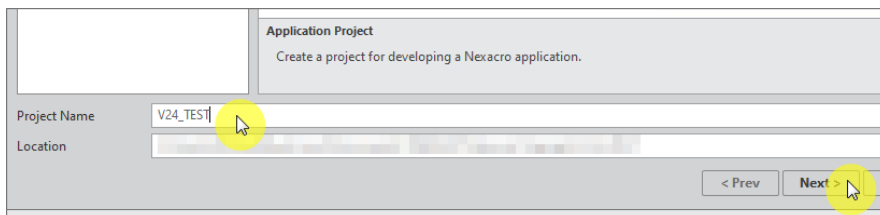
(<http://support.tobesoft.co.kr>에서 다운로드 받아 설치할 수 있습니다.)

1) NexacroStudio를 기동합니다.

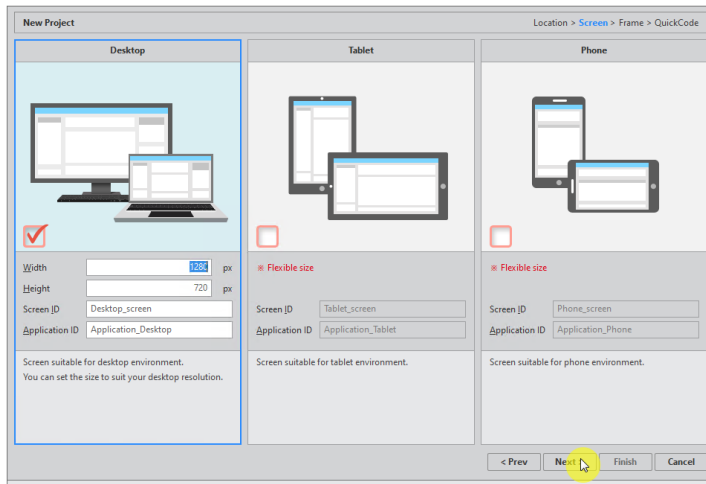
2) 새로운 프로젝트를 생성합니다.



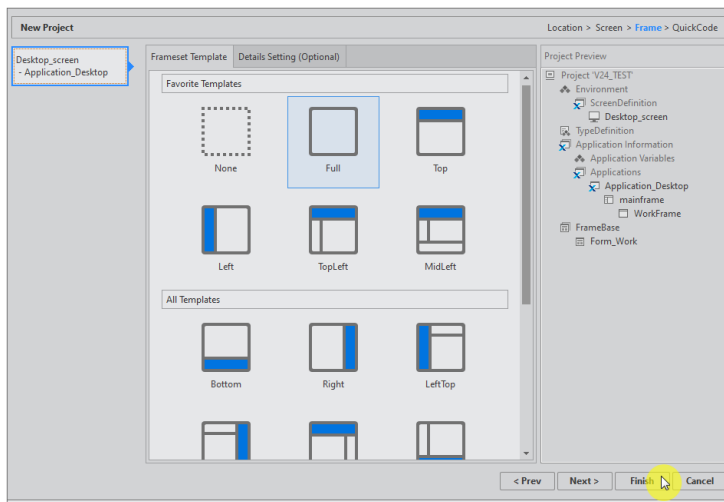
3) 새로운 창이 열리면 Project Name 항목에 프로젝트 이름을 입력하고 [Next] 버튼을 클릭합니다.



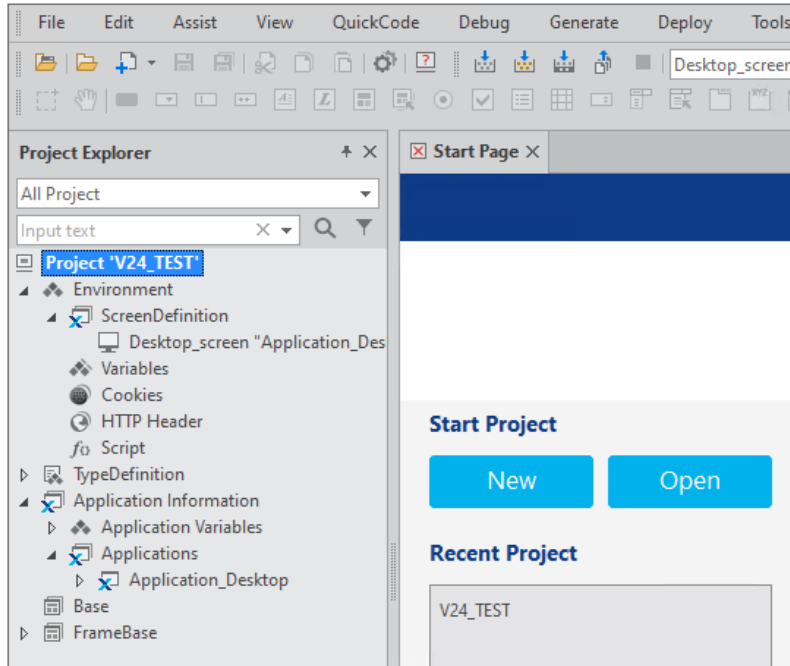
4) Screen 설정값은 기본값으로 놔두고 [Next] 버튼을 클릭합니다.



5) Frame 설정값도 기본값으로 놔두고 [Finish] 버튼을 클릭합니다.



6) Project Explorer 영역에 앞에서 입력한 프로젝트명으로 프로젝트가 생성된 것을 확인합니다.

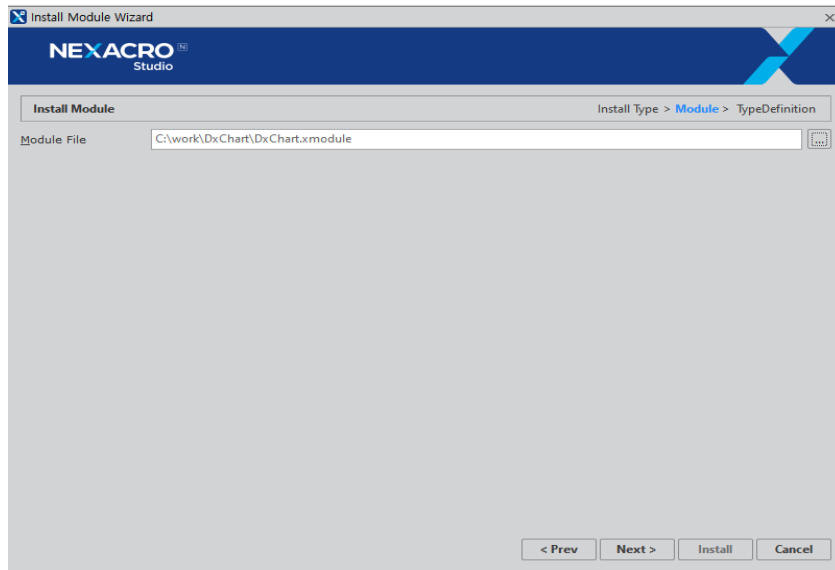


## (2) 차트모듈 설치하기

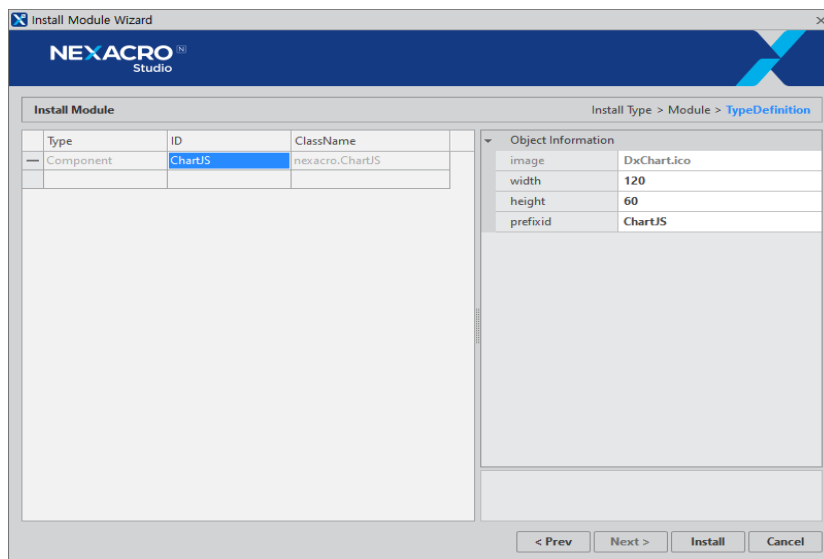
- 1) Nexacro Studio에서 [File]-[install module]을 선택합니다.
- 2) [Module Package(.xmodule)]을 선택하고 [Next]를 클릭합니다.



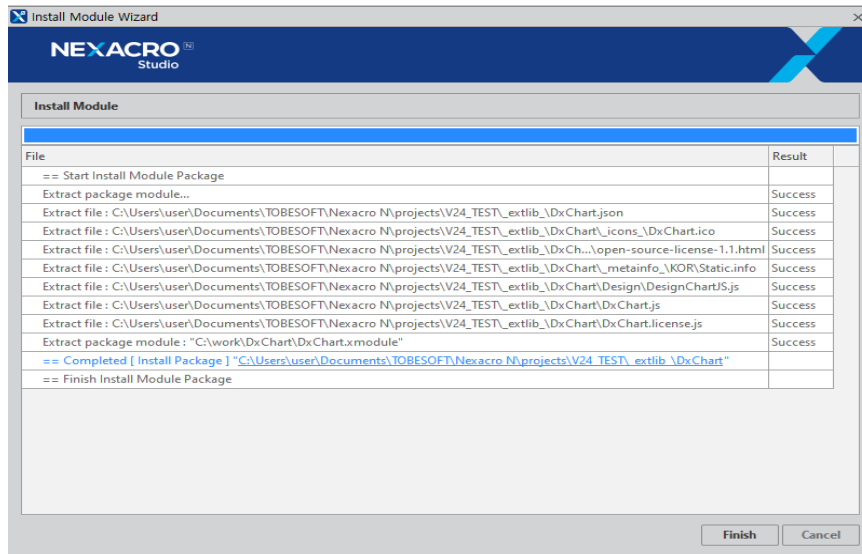
- 3) 준비된 DxChart.xmodule파일을 선택하고 [Next]를 클릭합니다.



4) [Install]버튼을 클릭합니다.

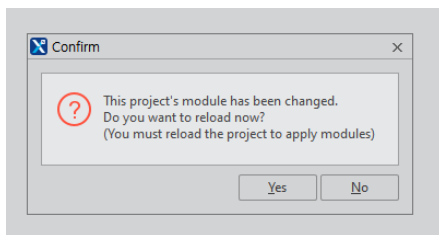


5) [Finish]버튼을 클릭합니다.



6) 설치가 끝나면 프로젝트 파일을 Reload할지를 묻는 Confirm메시지가 표시됩니다.

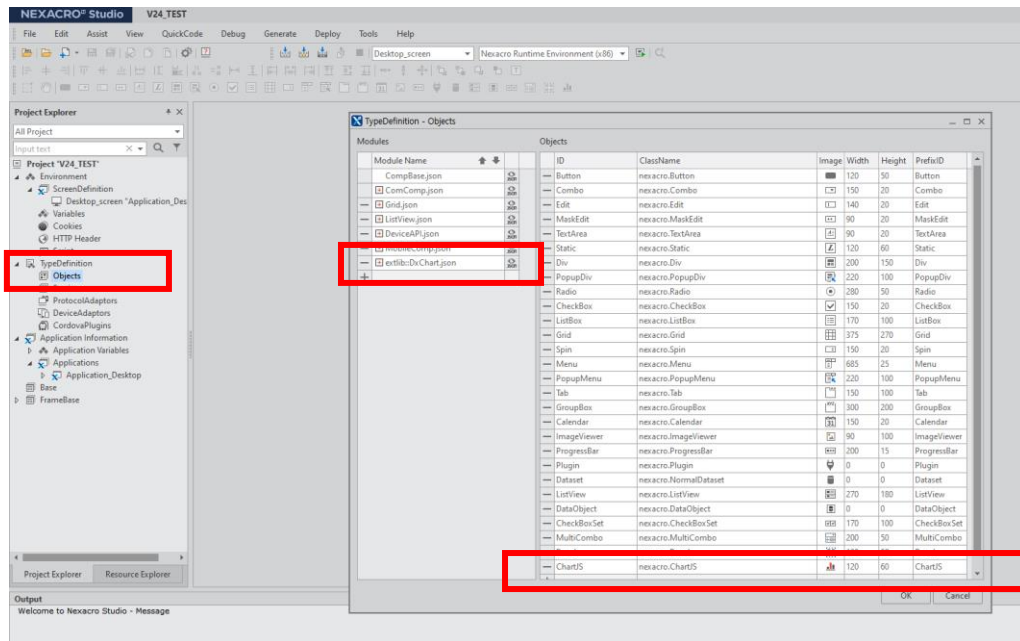
[Yes]를 클릭합니다.



7) DXChart가 설치된 것을 확인합니다.

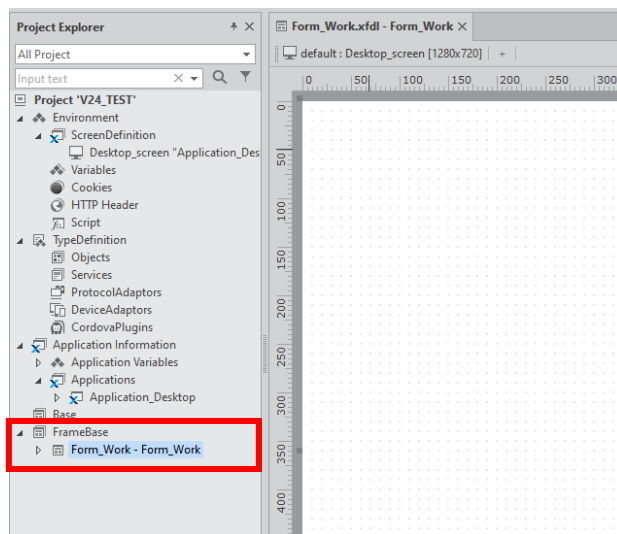
- [Project Explorer]-[TypeDefinition]-[Objects]를 선택하면 [TypeDefinition-Objects]팝업이 표시됩니다.
- [Modlues] 에 "extlib::DxChart.json"이 추가된 것을 확인합니다.
- [Objects]에 "ChartJS"가 추가되어 있는 것을 확인합니다.





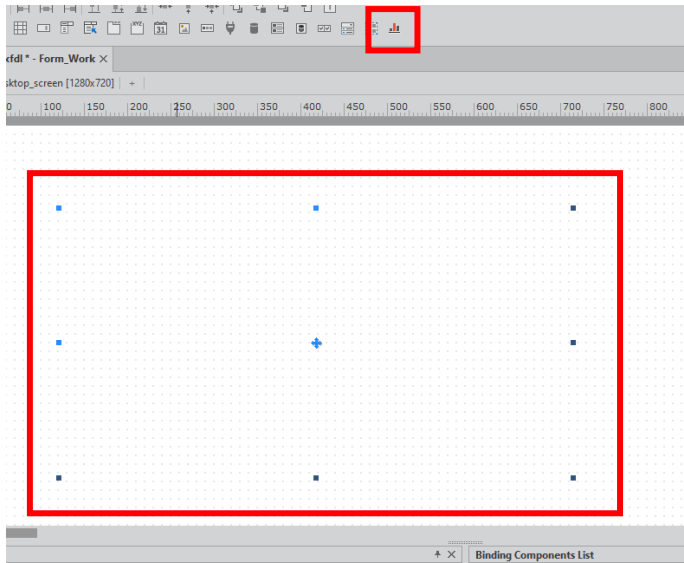
### (3) 샘플화면 만들기

- 1) 작성할 화면을 엽니다. (FrameBase::Frame\_Work.xfdl)



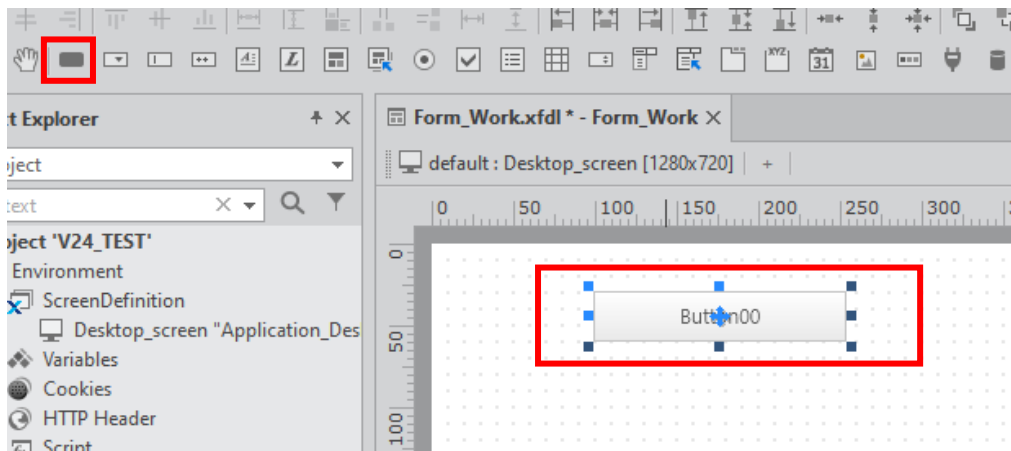
- 2) Chart 컴포넌트를 추가합니다.

컴포넌트의 아이디는 변경을 하지 않으면 기본적으로 "ChartU500"으로 생성됩니다.



(1) 버튼을 추가합니다.

컴포넌트의 아이디는 변경을 하지 않으면 기본적으로 "Button00"으로 생성됩니다.



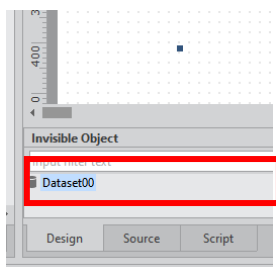
(2) 데이터 생성하기

표시할 데이터를 생성합니다. (데이터셋 사용)

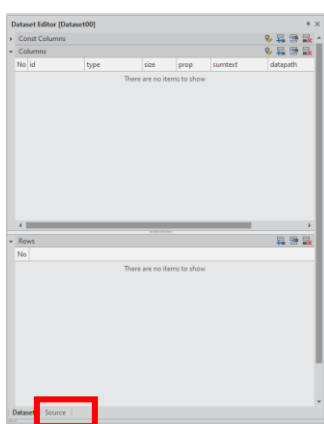
컴포넌트의 아이디는 변경을 하지 않으면 기본적으로 "Dataset00"으로 생성됩니다.



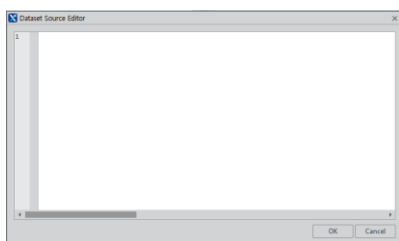
- 1) Dataset을 추가한 후 데이터셋을 더블클릭하면, [Dataset Editor]창이 표시됩니다.



- 2) [Dataset Editor]창 하단의 [Source]탭을 클릭합니다.



- 3) [Dataset Source Editor]팝업이 표시됩니다.



- 4) 아래의 내용을 창에 입력 후 [OK]를 클릭합니다.

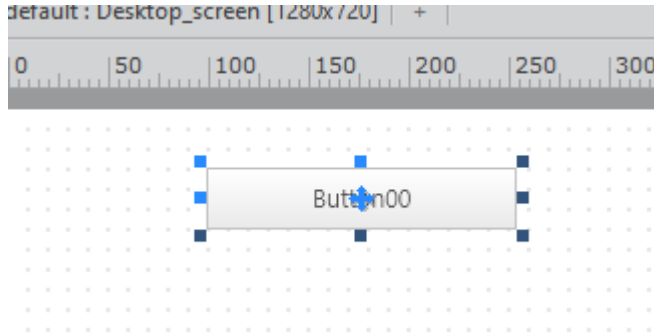


[데이터셋 내용]

```
<ColumnInfo>
  <Column id="indicators" type="STRING" size="256"/>
  <Column id="total" type="BIGDECIMAL" size="256"/>
  <Column id="man" type="BIGDECIMAL" size="256"/>
  <Column id="woman" type="BIGDECIMAL" size="256"/>
</ColumnInfo>
<Rows>
  <Row>
    <Col id="indicators">2015</Col>
    <Col id="total">5203440</Col>
    <Col id="woman">2610477</Col>
    <Col id="man">2592963</Col>
  </Row>
  <Row>
    <Col id="indicators">2016</Col>
    <Col id="total">5397615</Col>
    <Col id="woman">2721754</Col>
    <Col id="man">2675861</Col>
  </Row>
  <Row>
    <Col id="indicators">2017</Col>
    <Col id="total">5618677</Col>
    <Col id="woman">2826828</Col>
    <Col id="man">2791849</Col>
  </Row>
  <Row>
    <Col id="indicators">2018</Col>
    <Col id="total">5848594</Col>
    <Col id="woman">2942274</Col>
    <Col id="man">2906320</Col>
  </Row>
  <Row>
    <Col id="indicators">2019</Col>
    <Col id="total">6147516</Col>
    <Col id="woman">3093783</Col>
    <Col id="man">3053733</Col>
  </Row>
  <Row>
    <Col id="indicators">2020</Col>
    <Col id="total">6643354</Col>
    <Col id="woman">3338956</Col>
    <Col id="man">3304398</Col>
  </Row>
  <Row>
    <Col id="indicators">2021</Col>
    <Col id="total">7165788</Col>
    <Col id="woman">3582018</Col>
    <Col id="man">3583770</Col>
  </Row>
</Rows>
```

#### (4) 차트함수 만들기

생성한 버튼을 더블클릭하고 스크립트를 작성합니다.



[버튼 처리 내용]

```
this.Button00_onclick = function(obj:nexacro.Button,e:nexacro.ClickEventInfo)
{
    var chart = this.ChartJS00;
    var canvas = chart.getCanvas();
    var cvs = canvas.id;
    DxChart.reset(canvas);
    var labels = DxChart.convertDataset(this.Dataset00, ["bind:indecators"])
    var line = new DxChartBar({
        id: cvs,
        elem : canvas,
        binddataset : this.Dataset00,
        data:["bind:total","bind:man","bind:woman"],
        options: {
```

```

        colors: ['#FF6A8E','#42B0FF','#5DCA63'],

        title: { Text : '1인 가구 수', FontStyle: 'bold 12px Tahoma' },

        xaxis: { Use : true, Labels: labels },

        yaxis: { Use : true,

            LabelsCount: 5,

            ScaleMax: 10000000,

            TickmarksCount: 10,

            },

        margin: { Left: 100, Bottom: 50, Inner : 20 },

        tooltips: {

            Data : '%{key}',

            FormattedUnitsPost: '명',

            Effect : 'fade',

            Pointer : true,

            Css: {

                fontSize: '14pt'

            },

            FormattedKeyLabels: ['종합','남자','여자'],

        }

    }

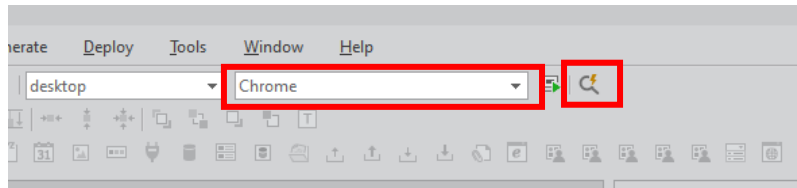
    }).wave();

}

```

### (3) 테스트해보기

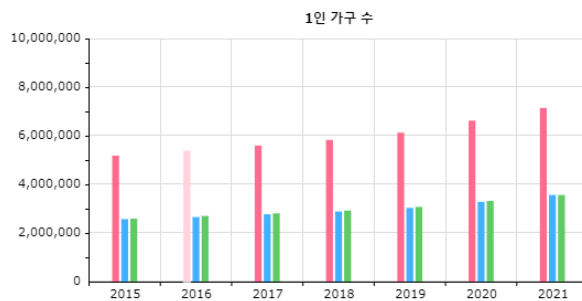
Quick View를 실행합니다.(Browser는 Chrome 또는 Microsoft Edge을 선택합니다.)



#### [실행 결과]

- 1) 버튼을 클릭합니다.
- 2) 결과를 확인합니다.

Button00



## 2. 차트 구성요소

### (1) 구성요소 용어 설명

차트의 호출형식은 아래와 같이 구성되어 있습니다.

```
var chart = this.chart; ①

var canvas = chart.getCanvas();

var cvs = canvas.id;

DxChart.reset(canvas);

var bar = new DxChartBar({ ②
    id: cvs, ③
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        colors: ['#FF6A8E','#42B0FF','#5DCA63'],
        title: { Text : '1인 가구 수' },
        xaxis: { Use : true, Labels: labels },
        yaxis: { Use : true,
            LabelsCount: 5,
            ScaleMax: 10000000,
            TickmarksCount: 10,
        },
        margin: { Left: 100, Bottom: 50, Inner : 20 },
    }
}).wave(); ④
```



- 1) 차트를 그리기 전에 차트영역을 reset해줍니다.

```
var chart = this.chart;    // Chart Object를 지정합니다.

var canvas = chart.getCanvas(); // canvas를 가져옵니다.

var cvs = canvas.id;      // canvas의 아이디

DxChart.reset(canvas);    // 차트영역(canvas)을 reset합니다.
```

- 2) 차트의 종류(오브젝트)를 지정합니다.

```
var bar = New DxChartBar(config);
```

//config : 차트를 그리기 위한 정보(config)

지정가능한 차트의 종류는 아래와 같습니다.

차트종류		오브젝트명
activity 차트	activity	DxChartActivity
막대 차트	bar	DxChartBar
양극성 차트	bipolar	DxChartBipolar
연료 차트	fuel	DxChartFuel
Funnel 차트	funnel	DxChartFunnel
간트 차트	gantt	DxChartGantt
게이지 차트	gauge	DxChartGauge
가로 막대 차트	hbar	DxChartHbar
히트맵 차트	heatmap	DxChartHeatmap
말발굽 차트	horseshoe	DxChartHorseshoe
수평 진행 차트	hprogress	DxChartHProgress
선 차트	line	DxChartLine
미터 차트	meter	DxChartMeter
계기판 차트	odo	DxChartOdo
조직구조 차트	org	DxChartOrg
파이 차트	pie	DxChartPie
피라미드 차트	pyramid	DxChartPyramid
레이더 차트	radar	DxChartRadar
로즈 차트	rose	DxChartRose
산점도 차트	rscatter	DxChartRscatter
분산형 차트	scatter	DxChartScatter
세그먼트 차트	segment	DxChartSegment
반원형 진행 차트	semicircularprogress	DxChartSemiCircle

온도계 차트	thermometer	DxChartThermometer
트리 차트	tree	DxChartTree
수직 진행 차트	vprogress	DxChartVProgress
워터폴 차트	waterfall	DxChartWaterfall

3) config를 정의합니다.

- Config는 JSON형식으로 구성이 되어 있습니다.
- Config는 아래의 구성요소로 되어 있습니다.

구성요소명	설명	사용예
id	차트 canvas의 아이디	Id : canvas.id
elem	차트 canvas 오브젝트	elem:this.chart00.getCanvas()
binddataset	Nexacro의 데이터셋 오브젝트	Binddataset:
datachangeevent	Bind된 데이터셋의 데이터가 변경이 되면 차트를 다시 그릴지를 확인한다. Use : 사용여부(true/false(default)) Effect : 다시 그릴 때 사용할 effect함수명(String)	datachangeevent : {Use:true,Effect:"wave"}
data	차트에 표시할 데이터 리스트(배열) Binddataset이 세팅된 경우는 "bind"+컬럼명을 지정하면 해당 컬럼의 데이터값이 배열로 세팅됩니다.	-["bind:Col0", "bind:Col1"] -[[10,20,30] , [20,30,40]]
option	차트에 대한 옵션값을 세팅합니다. - background , key , xaxis , yaxis , tooltip 등	{ xaxis : {...}, yaxis : {...}, key : {...}, ... }

## (2) 데이터 형식

차트에 사용되는 데이터의 형식은 2차원 배열을 사용합니다.

### 1) data Property

차트에 사용될 데이터를 지정합니다. 데이터의 지정방식을

bind컬럼지정/CSV배열/2차원배열등로 표현되어 지정 가능합니다.

형식: data : [ [첫번째 차트 데이터] , [두번째 차트 데이터] , [세번째 차트 데이터]]

사용 예 : data : ["bind:Column0","bind:Column1"] (binddataset을 지정한경우)

data: ["123,123,123,123","123,123,123,1234"] (CSV데이터의 배열)

data: [[123,123,123,123],[123,123,123,1234]] (2차원 배열)

### 2) binddataset Property

데이터셋의 오브젝트를 지정합니다.

이 Property를 사용하는 경우, 일부 Property에서는 "bind: 컬럼명"형태로

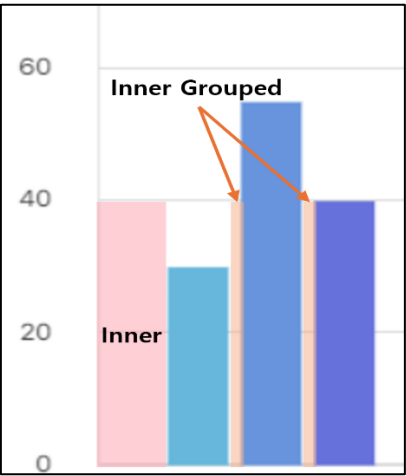
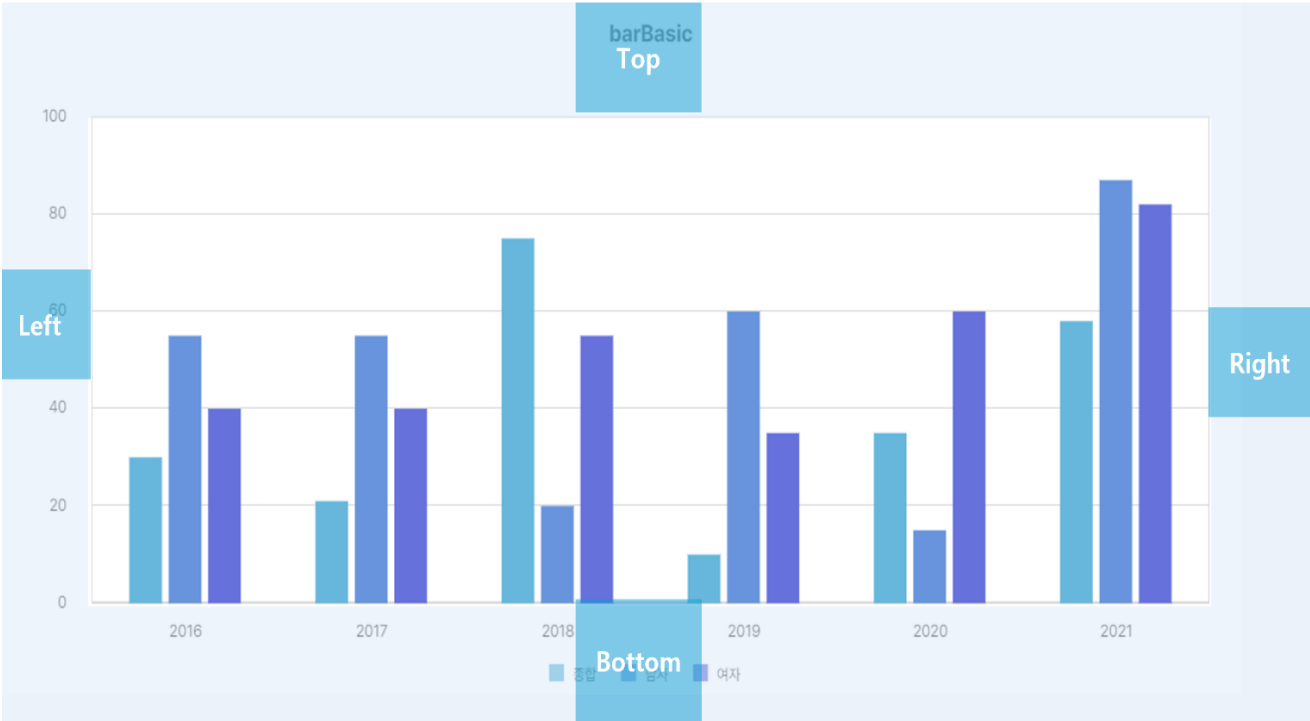
데이터의 리스트(배열)을 지정할 수 있습니다.

(Nexacro 전용 프로퍼티)

사용 예 : binddataset : this.dsList

(3) 차트 주요 Properties

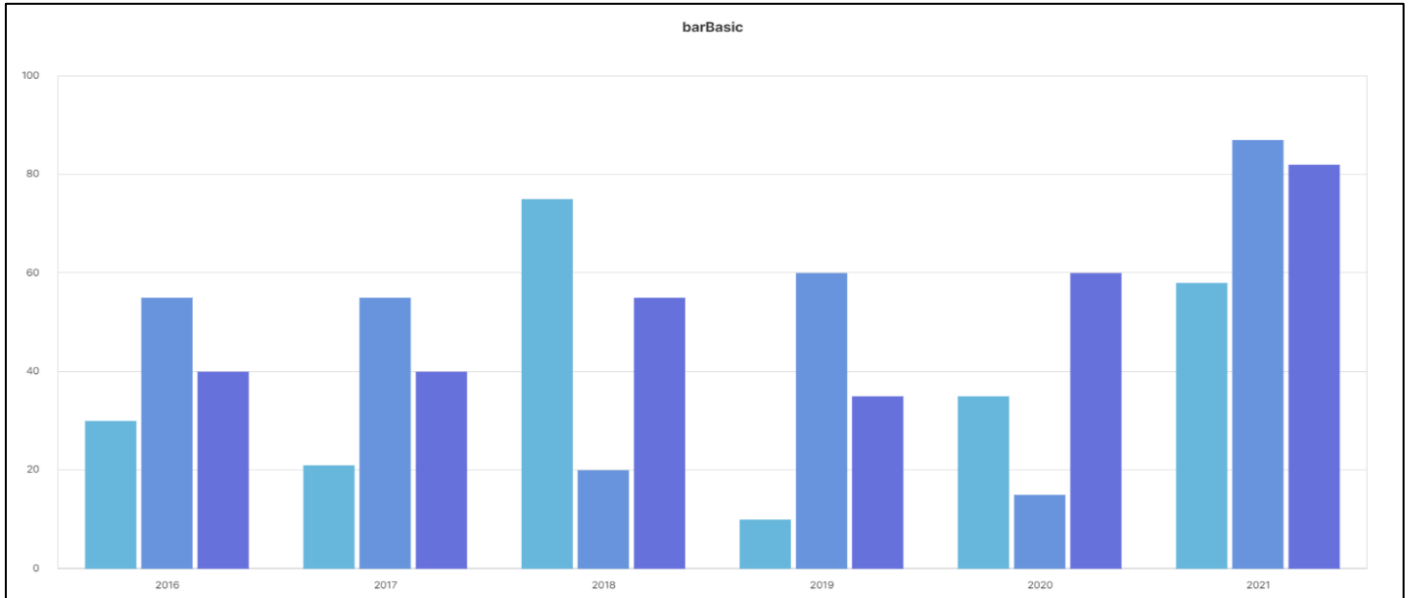
1) Margin (옵션 - margin 그룹)



Sub property	설명	타입	기본값
Bottom	차트의 아래쪽 여백을 설정합니다.	number	35
Inner	차트의 내부 여백을 설정합니다.	number	5
InnerGrouped	그룹화된 차트에서 동일한 그룹에 있는 요소 사이의 여백입니다.	number	1
Left	차트의 왼쪽 여백을 설정합니다.	number	35
Right	차트의 오른쪽 여백을 설정합니다.	number	35
Top	차트의 위쪽 여백을 설정합니다.	number	35

### ① Margin 사용 예제

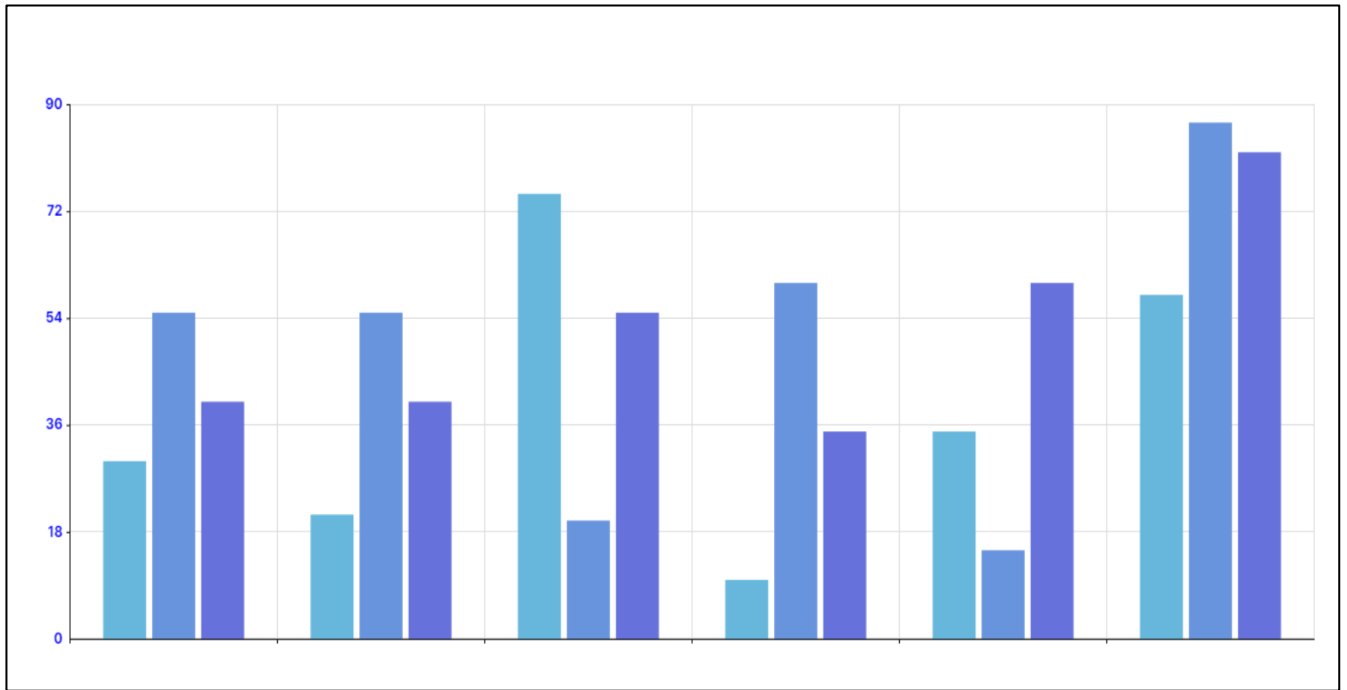
아래는 차트의 Margin Properties 설정하기 튜토리얼 입니다.



```
var bar = new DxChartBar({  
  
    id: cvs,  
  
    elem : canvas,  
  
    binddataset : this.Dataset00,  
  
    data:["bind:total","bind:man","bind:woman"],  
  
    options: {  
  
        margin: {  
  
            Left: 100, // 차트의 왼쪽 여백 값을 설정할수 있습니다.  
  
            Right: 100, // 차트의 오른쪽 여백 값을 설정할수 있습니다.  
  
            Top: 50, // 차트의 위쪽 여백 값을 설정할수 있습니다.  
  
            Bottom: 50, // 차트의 아래쪽 여백 값을 설정할수 있습니다.  
  
            Inner : 20, // 차트의 내부의 여백 값을 설정할수 있습니다.  
  
            InnerGroupued : 20 // 차트의 그룹 사이 여백 값을 설정할수 있습니다.  
  
        },  
  
    }.  
  
}).draw();
```

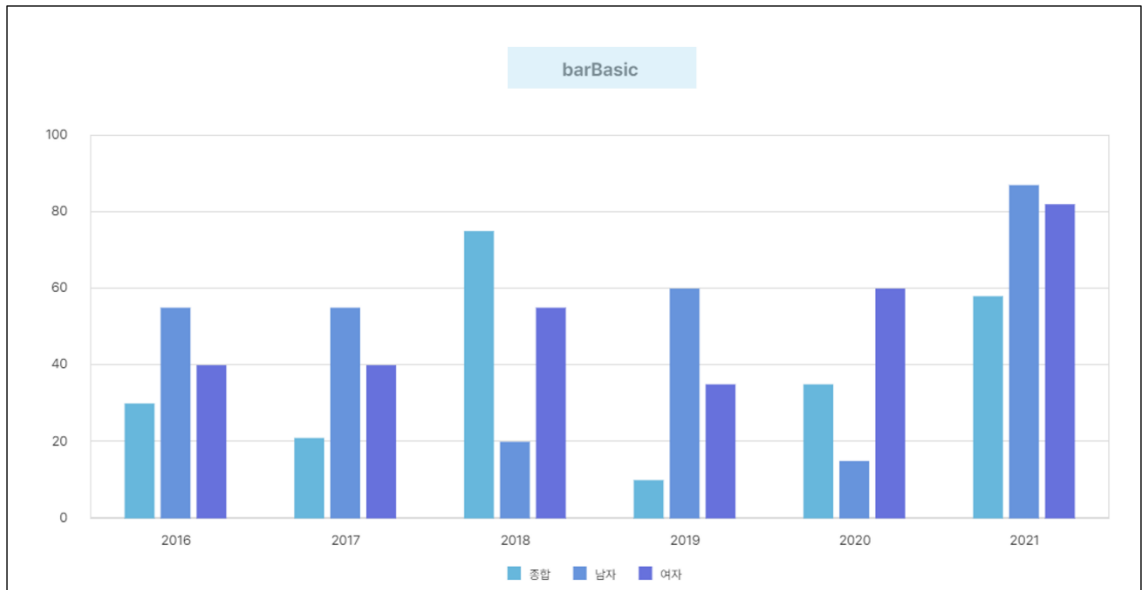
## ② Text Property 예제

차트 내부에 렌더링 되는 모든 Text 값들의 색상, Font Style을 지정할 수 있습니다. title, X축라벨, Y축라벨 등 모든 텍스트 값들을 제어할 수 있습니다.



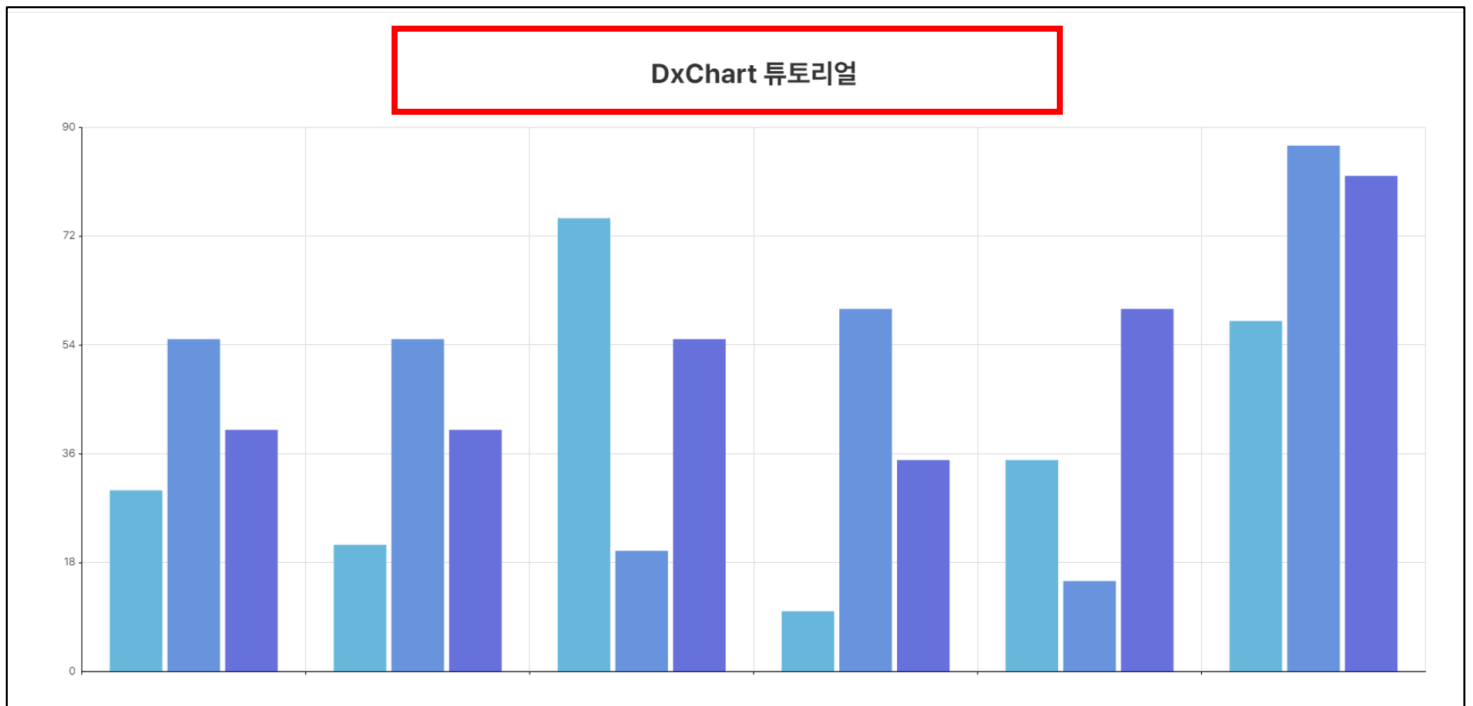
```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        text: {
            Color:'Blue',    // text 색상을 설정할수 있습니다. 위의 예제에선 Y축 라벨 색상입니다.
            FontStyle: '20px bold Pretendard' // Text FontStyle을 설정할 수 있습니다.
        }
    }
}).draw();
```

## 2) 제목 (옵션 – title 그룹)



Sub property	설명	타입	기본값
Color	제목의 색을 지정합니다.	object	null
FontStyle	제목의 글꼴 스타일을 지정합니다.	string	null
Halign	제목의 가로 정렬을 지정합니다. left, right, center 를 사용할 수 있습니다.	object	null
OffsetX	제목의 가로 오프셋을 지정합니다.	number	0
OffsetY	제목의 세로 오프셋을 지정합니다.	number	0
Subtitle	차트의 부제목을 지정합니다. 제목 바로 아래에 생기기 때문에 마진을 주지 않으면 제목이 제대로 보이지 않을 수 있습니다.	string	
SubtitleColor	부제목의 색을 지정합니다.	string	#aaa
SubtitleFontStyle	부제목의 글꼴 스타일을 지정합니다.	string	null
SubtitleOffsetX	부제목의 가로 오프셋을 지정합니다.	number	0
SubtitleOffsetY	부제목의 세로 오프셋을 지정합니다.	number	0
Text	중앙 상단의 제목을 지정합니다.	string	
Valign	제목의 세로 정렬을 지정합니다. top, bottom, center를 사용할 수 있습니다.	object	null
X	제목의 가로 좌표를 지정합니다.	object	null
Y	제목의 세로 좌표를 지정합니다.	object	null

## ① Title Property 예제



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        title: { Text : 'DxChart 튜토리얼', // 차트의 title Text값을 설정합니다.  
                FontStyle: '32px bold Pretendard', // title FontStyle을 설정할 수 있습니다.  
                Y :73      // 차트 내 title의 Y 위치값을 설정 할수 있습니다.  
                X: 450 // 차트 내 title의 X 위치값을 설정 할수 있습니다.  
            },  
    },  
}).draw();
```

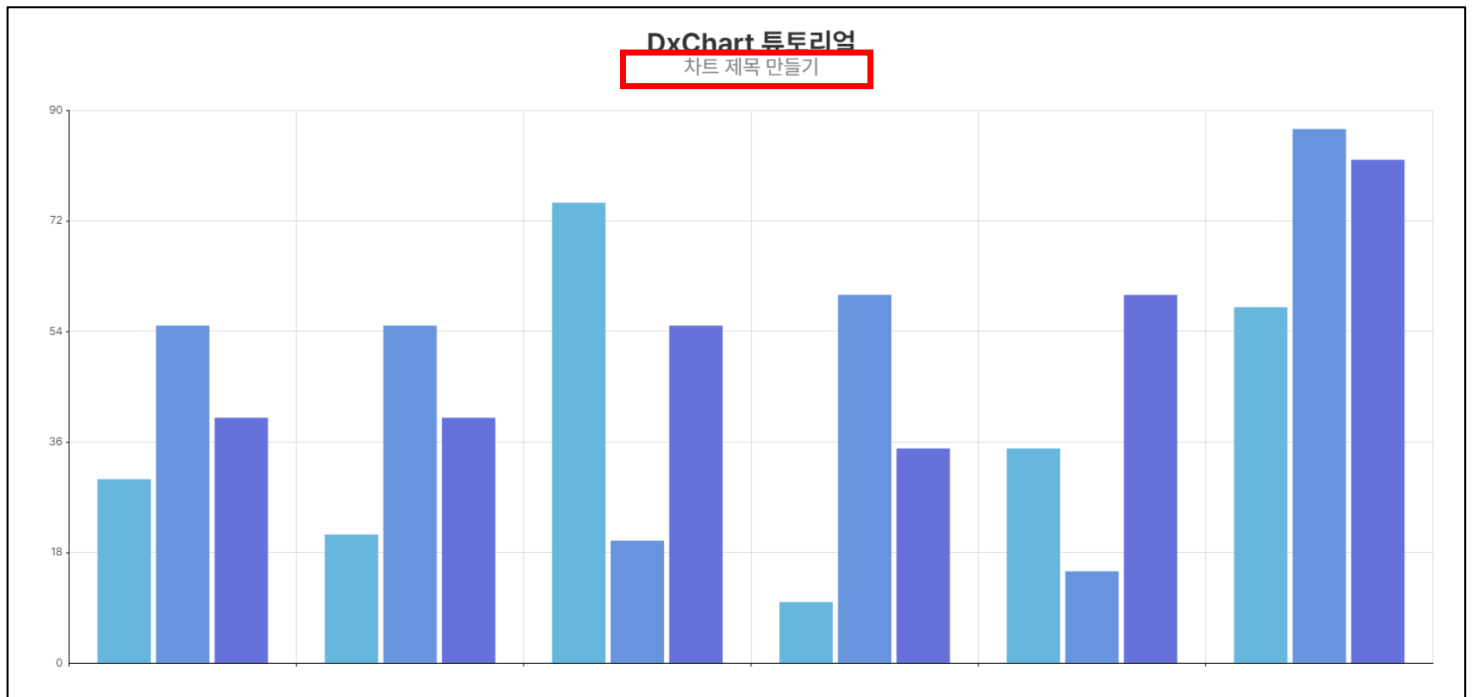


### 3) 제목(옵션 -Subtitle)

DxChart는 Main Title 외에 차트의 부제목 또한 설정할 수 있습니다. 같은 title Properties내에 Subtitle 옵션값으로 설정 할 수 있습니다.

Sub property	설명	타입	기본값
Subtitle	차트의 부제목을 지정합니다. 제목 바로 아래에 생기기 때문에 마진을 주지 않으면 제목이 제대로 보이지 않을 수 있습니다.	string	
SubtitleColor	부제목의 색을 지정합니다.	string	#aaa
SubtitleFontStyle	부제목의 글꼴 스타일을 지정합니다.	string	null
SubtitleOffsetX	부제목의 가로 오프셋을 지정합니다.	number	0
SubtitleOffsetY	부제목의 세로 오프셋을 지정합니다.	number	0

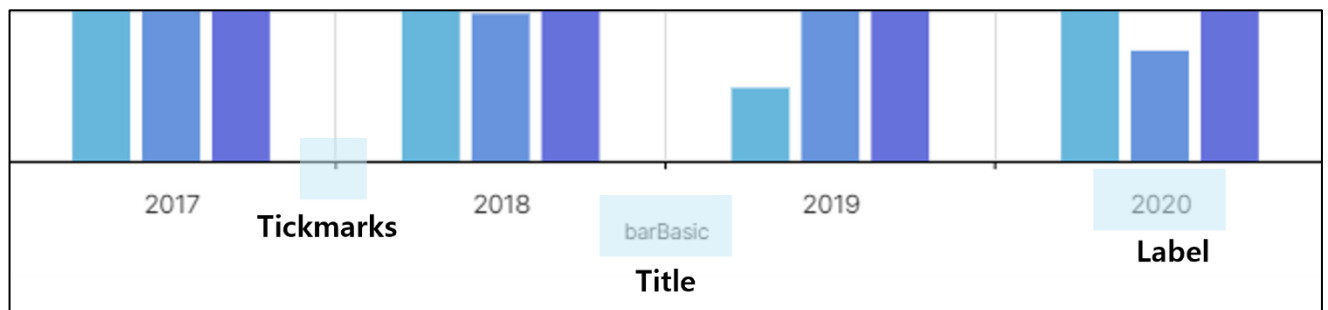
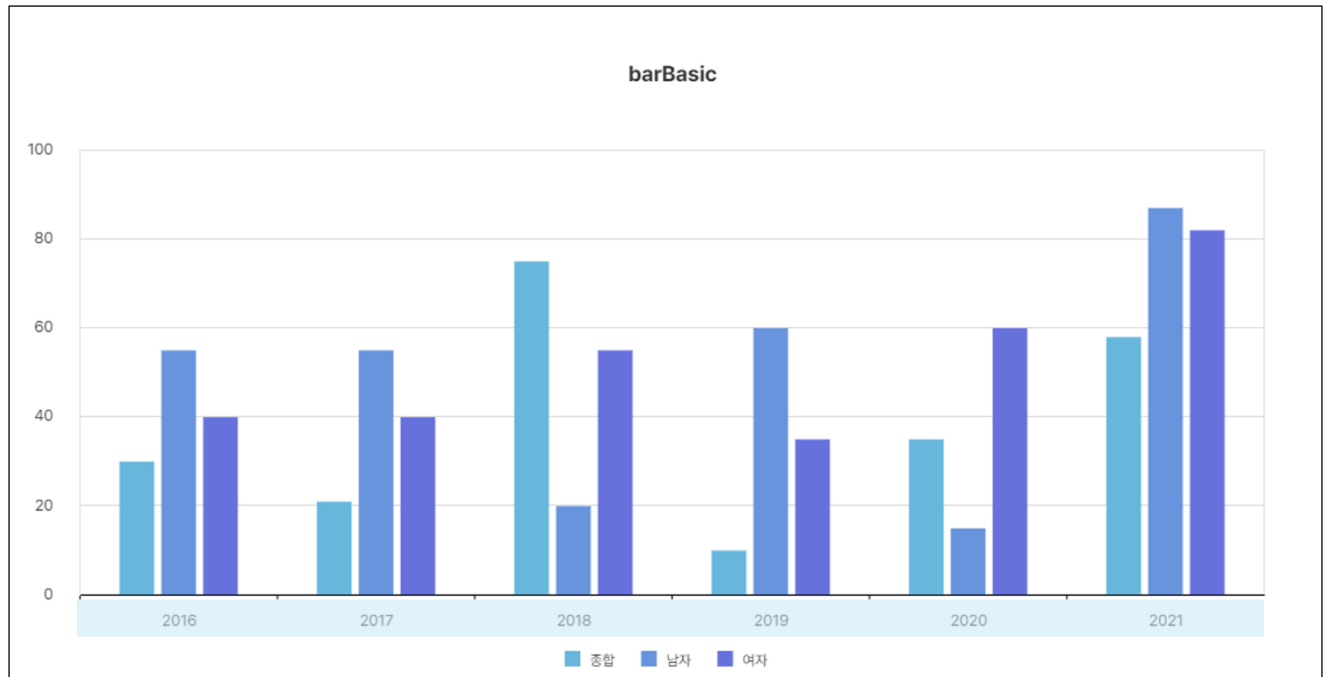
### ① SubTitle Property 예제



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        title: { Text : 'DxChart 튜토리얼', // 차트의 title Text값을 설정합니다.  
                Subtitle: '차트 제목 만들기', // 차트의 부제목을 설정할 수 있습니다.  
                SubtitleFontStyle : '24px Pretendard' // 차트 부제목의 폰트 스타일을 설정 할 수 있습니다.  
                // SubtitleOffsetY: 103 // 차트 부제목의 Offset Y 좌표 위치를 설정 할 수 있습니다. X 좌표 위치값 변경도 가능합니다.  
            },  
    },  
}).draw();
```

#### 4) X축 (옵션 - xaxis 그룹)

차트의 X축에 대한 설정을 하는 Property입니다.

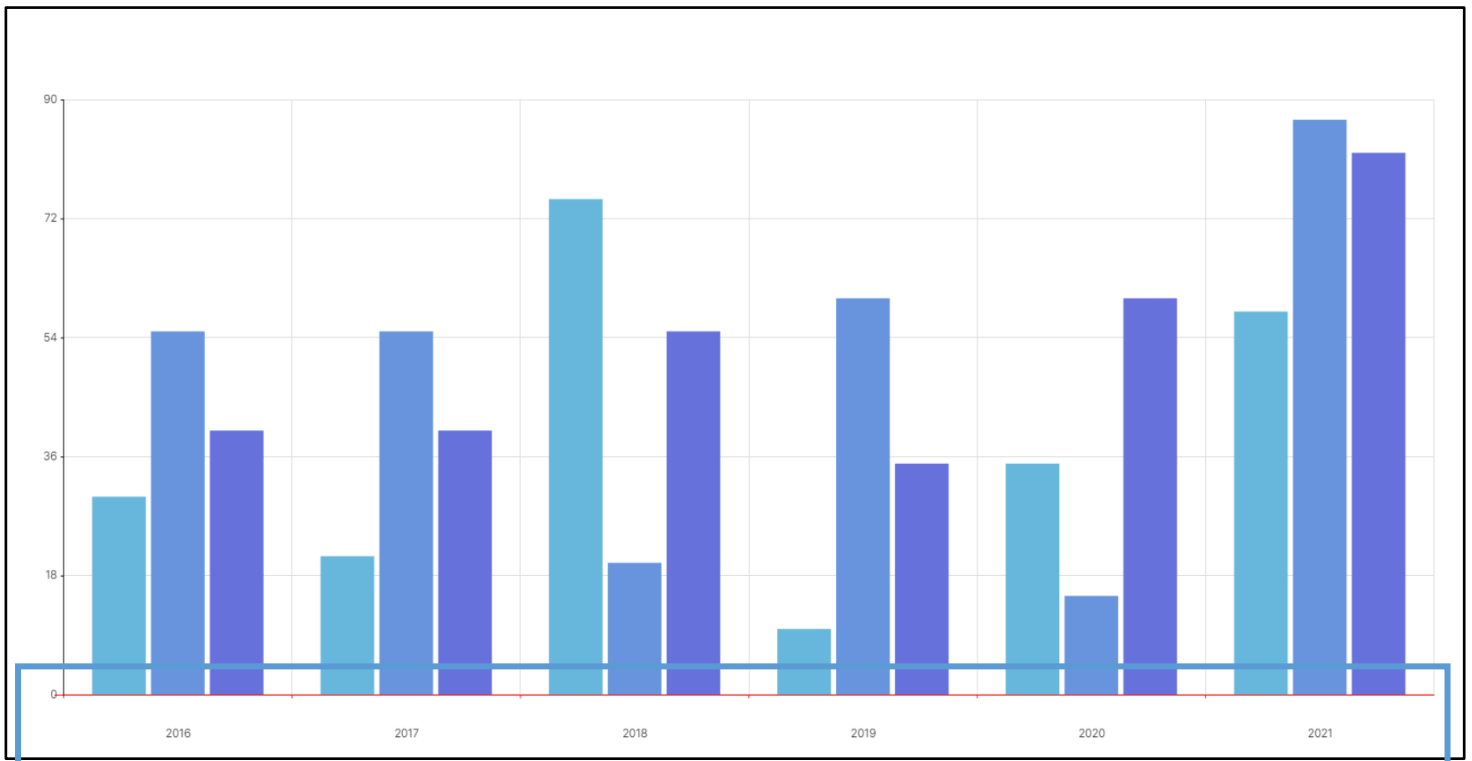


Sub property	설명	타입	기본값
AddLineSize	x축 선의 길이를 지정합니다. 양수이면 원점에서 왼쪽으로 x축 길이가 늘어나며 음수이면 원점에서 오른쪽으로 x축 길이가 줄어듭니다.	number	0
Color	x축 선의 색을 지정합니다.	string	black
Labels	x축에서 사용할 라벨을 지정합니다. 라벨은 html 태그 형식 또는 배열 형태로 사용할 수 있습니다.	object	null
LabelsAngle	x축 라벨의 각도를 지정합니다. 양수일 때 지정한 각도만큼 왼쪽으로 회전하며 음수일 때 오른쪽으로 회전합니다.	number	0

LabelsColor	x축 라벨의 색을 지정합니다.	object	null
LabelsFontStyle	x축 라벨의 글꼴 스타일을 지정합니다.	string	null
LabelsFormattedDecimals	x축에서 %{value_formatted}처럼 형식이 지정된 라벨에서 지정한 값만큼의 소수점을 표시합니다.	number	0
LabelsFormattedPoint	x축에서 %{value_formatted}처럼 형식이 지정된 라벨에서 소수점 표시 방식을 지정합니다.	string	.
LabelsFormattedThousand	x축에서 %{value_formatted}처럼 형식이 지정된 라벨에서 천 단위 구분 기호를 지정합니다.	string	,
LabelsFormattedUnitsPost	x축에서 %{value_formatted}처럼 형식이 지정된 라벨에서 뒤에 표시할 문자를 지정합니다.	string	
LabelsFormattedUnitsPre	x축에서 %{value_formatted}처럼 형식이 지정된 라벨에서 앞에 표시할 문자를 지정합니다.	string	
LabelsHalign	x축 라벨의 가로 정렬을 지정합니다. left, right, center를 사용할 수 있습니다.	object	null
LabelsOffsetX	x축 라벨의 가로 오프셋을 지정합니다.	number	0
LabelsOffsetY	x축 라벨의 세로 오프셋을 지정합니다.	number	0
LabelsPosition	x축 라벨의 포지션을 지정합니다. section, edge를 사용할 수 있습니다. section은 영역쪽에 라벨이 위치하고 edge는 데이터 포인트 점에 라벨이 위치합니다.	string	section
LabelsValign	라벨의 세로 정렬을 지정합니다. top, bottom, center를 사용할 수 있습니다.	object	null
Linewidth	x축의 두께를 지정합니다.	number	1
Position	x축의 위치를 지정합니다. top, bottom, center를 사용할 수 있습니다.	string	bottom
Tickmarks	x축 라벨에 눈금을 표시할 지 여부입니다.	boolean	true
TickmarksCount	x축 라벨의 눈금 개수를 지정합니다.	object	null
TickmarksLastLeft	x축 라벨의 가장 왼쪽 마크를 표시할 지 여부입니다.	object	null
TickmarksLastRight	x축 라벨의 가장 오른쪽 마크를 표시할 지 여부입니다.	object	null
TickmarksLength	x축 라벨의 데이터마다 위치한 마크값의 길이를 지정합니다.	number	
Title	x축의 제목을 지정합니다.	string	
TitleColor	x축 제목의 색상을 지정합니다.	object	null
TitleFontStyle	x축 제목의 글꼴 스타일을 지정합니다.	string	null
TitleHalign	x축 제목의 가로 정렬을 지정합니다. 'left','right','center'를 사용할 수 있습니다.	string	center

TitleOffsetX	x축 제목의 가로 오프셋을 지정합니다.	number	0
TitleOffsetY	x축 제목의 세로 오프셋을 지정합니다.	number	0
TitlePos	x축 제목의 위치를 지정합니다. 양수일 때 아래로 내려가고 음수일 때 위로 올라갑니다.	object	null
TitleValign	x축 제목의 수직 정렬을 지정합니다. top, bottom, center를 사용할 수 있습니다.	string	top
TitleX	x축 제목의 가로 좌표를 지정합니다.	object	null
TitleY	x축 제목의 세로 좌표를 지정합니다.	object	null
Use	x축을 사용할 지 여부입니다.	boolean	true

## ① X축 Property 기본 예제



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        xaxis: {  
            Use : false, // 차트의 X축을 Draw 할지 설정할 수 있습니다.  
            Labels: ["bind:indecators"], // X축 라벨의 데이터 값을 설정할 수 있습니다.  
            // 주로 Nexacro 내 Dataset 바인딩을 통해 데이터 값을 설정 합니다.  
            // Nexacro가 아닌 다른 프레임워크나 실행환경에선 배열값으로 설정할수 있습니다.  
            // ex) Lables :['1','2','3','4','5']  
            AddLineSize : 10, // X축 선의 길이 값을 설정 할 수 있습니다. }  
        }  
    };
```

Color : 'red', // X축의 색상을 설정 할 수 있습니다.

Position: 'bottom' // X축의 위치를 설정 할 수 있습니다. 기본값은 bottom이며,  
top, bottom 두가지 값으로 설정 할 수 있습니다.

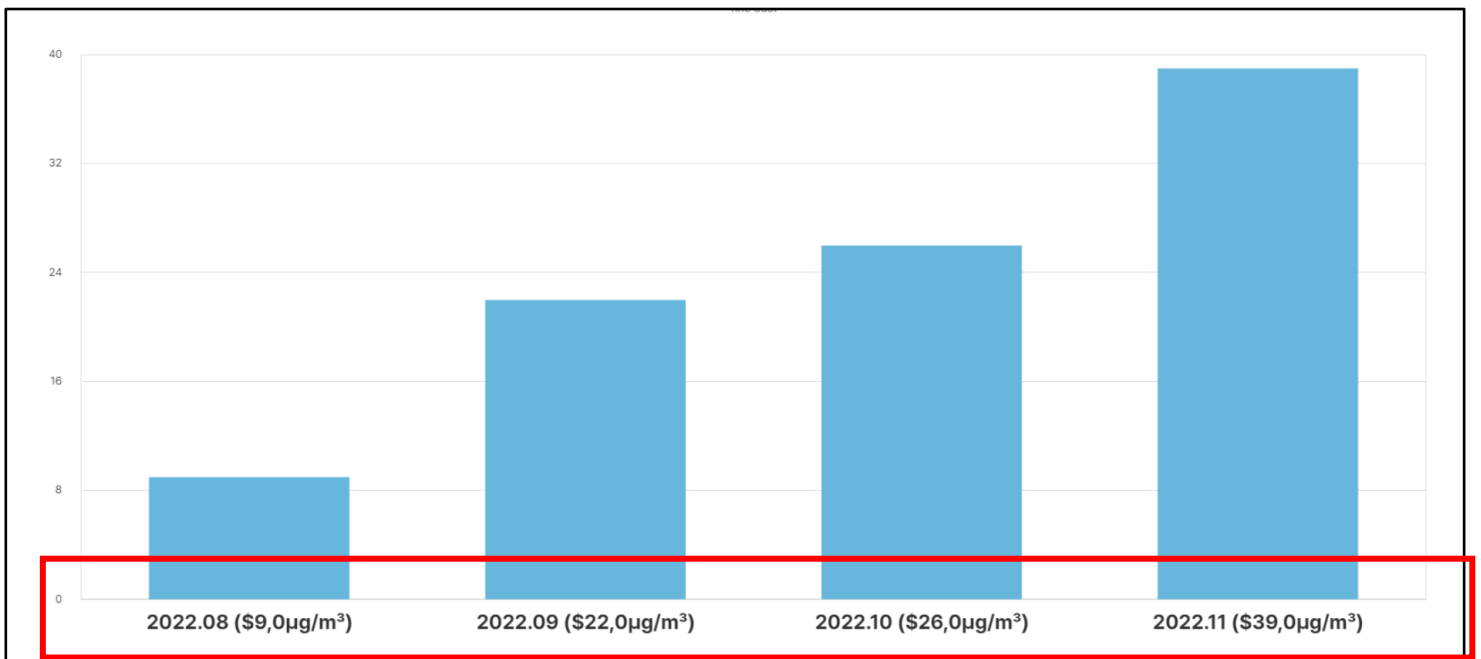
X축의 Position을 top으로 설정 할 시 차트가 상하반전이 되어 렌더링됩니다.

```
}
```

```
}
```

```
}).draw();
```

### ③ X축 Labels 기본 예제



```

var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        names: DxChart.convertDataset(this.Dataset00,['bind:year']),
        // Labels formatted 사용을 위해 names프로퍼티 선언후, 데이터셋바인딩을 해줍니다.
        // 프로퍼티 이름은 자유롭게 사용이 가능합니다.
        xaxis: { Use : true,
            Labels: '%{property:names[%{index}]} (%{value_formatted})',
            // 현재 예제에서는 Labels formatted 기능을 사용하기위해 글로벌 내장 함수 사용을 합니다.
            // Labels formatted 기능을 제외할시, 기존과 동일하게 dataset바인딩이나, 배열 데이터로 선언합니다.
            LineWidth : 5, // 차트의 X축 Line의 두께 값을 설정 할 수 있습니다..
            LabelsAngle : 5, // 차트의 X축 Labels의 각도를 설정 할 수 있습니다. 양수 음수 모두 가능합니다.
            LabelsColor: 'blue', // 차트의 X축 Labels의 색상을 설정할 수 있습니다.
            LabelsOffsetY:15, // 차트의 X축 OffsetY 좌표 값을 설정 할 수 있습니다.
            LabelsFontStyle: '24px bold Pretendard', // X축 Labels의 FontStyle을 설정할 수 있습니다.
            LabelsFormattedDecimals : 1, // X축 Labels의 표시되는 소수점 수를 설정 할 수 있습니다.
            LabelsFormattedPoint: ".", // X축 Labels의 표시되는 소수점 문자를 설정할 수 있습니다.
            LabelsFormattedThousand: ",", //X축 Labels의 표시되는 천단위 구분점을 설정할 수 있습니다.
            LabelsFormattedUnitsPost: " µg/m³", //X축 Labels 뒤에오는 구분문자를 설정할 수 있습니다.
            LabelsFormattedUnitsPre: "$", //X축 Labels 앞에오는 구분문자를 설정할 수 있습니다.
            LabelsHalign : 'center', //X축 Labels의 가로정렬 방향을 설정할수 있습니다. (left,right,center).
            LabelsOffsetX : 0, //X축 Labels의 OffsetX 위치값을 설정할 수 있습니다.
            LabelsOffsetY : 0, //X축 Labels의 OffsetY 위치값을 설정할 수 있습니다.
            LabelsPosition : 'section', //X축 Labels의 Position값을 설정할 수 있습니다. (section, edge)

```



#### ④ 추가 예제 xaxisLabelsFormatter

LabelsFormatter Properties는 차트의 데이터 값이나 라벨 데이터의 명확성을 제시할 수 있습니다. 기존 데이터에 추가적인 String값이나 특정 값을 붙여 줌으로써 해당 데이터의 명료성을 더해 줄 수 있습니다.

xaxisLabelsFormatter 를 만들 때 사용할 수 있는 매크로는 다음과 같습니다.

##### ① 인덱스 매크로

`%{index}`

- 이 매크로는 레이블의 인덱스를 표시합니다. 자바스크립트 배열과 같이 0 부터 시작하는 점을 유의해야 합니다.

##### ② 속성 매크로

`%{property:myProperties[%{index}]}`

- 이 매크로는 템플릿 문자열에서 객체의 속성을 참조할 수 있게 해줍니다. 예를 들어, 위의 이름 속성을 설정하고 템플릿 문자열에서 참조할 수 있습니다. 속성 매크로 사용시, 해당 속성에 사용될 변수를 반드시 할당하고 사용해야 합니다.

##### ③ 값 매크로

`%{value}, %{value_formatted}`

- 인덱스와 속성 매크로가 라벨에 명시될 값을 선언한다면, 값 매크로는 명시될 값 뒤에 `Format value` 를 설정할 수 있습니다. 따라서 `FormattedPost`, `FormattedPre` 프로퍼티와 항상 함께 사용해야 합니다.

Ex) xaxis : {

`Labels: %{index} %{value_formatted} //인덱스매크로`

`Labels: %{property:myProperties[%{index}]} %{value_formatted} // 속성 매크로`

`//속성 매크로 사용시 항상 프로퍼티 오브젝트를 생성해주어야합니다.`

`myProperties : ['1','2','3','4',,,, ] ,`

`myProperties : DxChart.convertDataset(this.Dataset, ['bind:data'])`

}

#### ④ 전역변수 매크로

`%{global:myVariable} %{global:myVariable[%{index}]}`

- 이 매크로는 템플릿 문자열에서 전역 변수를 참조할 수 있게 합니다. 이를 통해 차트 객체의 속성으로 데이터를 설정하는 것보다 쉽게 데이터를 참조할 수 있습니다.

#### ⑤ 함수 호출 매크로

`%{function:myFunction()} %{function:myFunction(%{index})}`

- 이 매크로는 함수를 호출하여 반환 값을 레이블 텍스트로 사용할 수 있게 합니다. 인수를 전달하지 않고 함수를 호출할 수도 있으며, 인덱스를 인수로 전달할 수도 있습니다.

Ex)

```
this.fnDraw = function()
{
  var chart = this.ChartJS00;
  var canvas = chart.getCanvas();
  var cvs = canvas.id;
  DxChart.reset(canvas);
```

```
//전역변수를 선언해줍니다.
myVariable = [.....]
```

```
var bar = new DxChartBar({
  id: cvs,
  elem : canvas,
  binddataset : this.Dataset00,
  data:["bind:val"],
  options: {

    xaxis:{
```

```
      Labels: '%{global:myVariable[%{index}]} (%{value_formatted})',
```

```
      LabelsFormattedUnitsPost: 'µg/m³', //( %{value_formatted}) 값 뒤에 붙을 Format 입니다
```

```
      LabelsFormattedDecimals: 1,
```

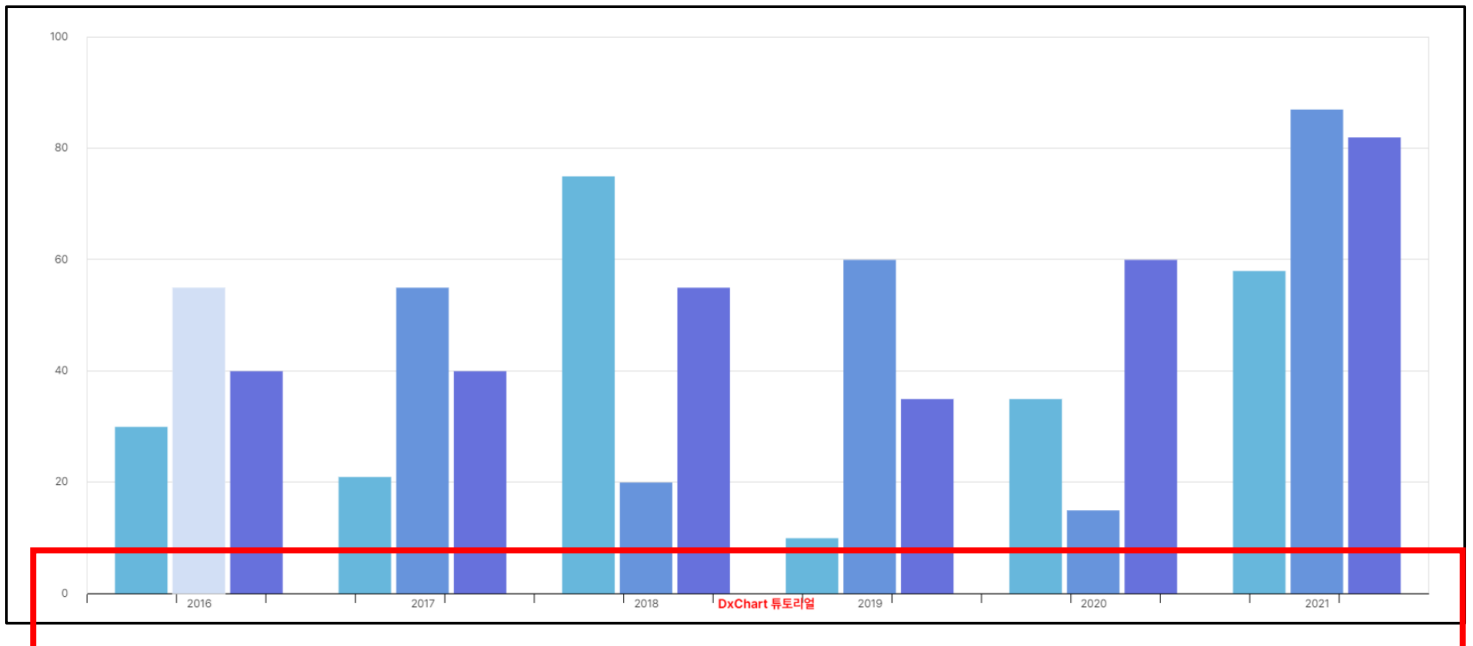
```
      LabelsFormattedPoint: ',',
```

```
      LabelsFormattedThousand: ',,',
```

```
      LabelsFormattedUnitsPre:
```

```
    }
```

⑤ X축 Property 기본 예제( Tickmarks, Title 관련)



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        xaxis: { Use : true,  
            Labels: ['bind:indicators'],  
            Tickmarks:true, // X축 Labels에 눈금 사용 여부를 설정 할 수 있습니다. 기본값은 true입니다.  
            TickmarksCount:15, // X축 Labels에 눈금 갯수를 설정 할 수 있습니다.  
            TickmarksLastLeft:null, // X축 Labels에 눈금 사용 여부를 설정 할 수 있습니다. 기본값은 true입니다.  
            TickmarksLastRight:null, // X축 Labels에 눈금 사용 여부를 설정 할 수 있습니다. 기본값은 true입니다.  
            TickmarksLength:10, // X축 Labels에 눈금의 길이를 설정할 수 있습니다..  
            Title:'DxChart 튜토리얼', // X축 Labels의 제목을 설정할 수 있습니다.
```

TitleColor: 'red', // X축 Labels 제목의 색상값을 설정 할 수 있습니다.

TitleFontStyle:'15px bold Pretendard', // X축 Labels 제목의 폰트 스타일을 설정할 수 있습니다.

TitleHalign: 'center', // X축 Labels에 제목의 가로정렬 값을 설정 할수 있습니다. (left,right,center)

TitleOffsetX: 5, // X축 Labels에 제목의 OffsetX 값을 설정 할수 있습니다.

TitleOffsetY:5, // X축 Labels에 제목의 OffsetY 값을 설정 할수 있습니다.

TitlePos:10, // X축 Labels에 제목의 위치를 지정합니다. 양수일 때 아래로 내려가고 음수일 때 위로 올라갑니다..

TitleValign:'center', // X축 Labels에 제목의 세로정렬 값을 설정 할수 있습니다. (top, bottom, center)

TitleX: 850, // X축 Labels에 제목의 X좌표 값을 설정 할수 있습니다. 양수, 음수 모두 설정 가능합니다

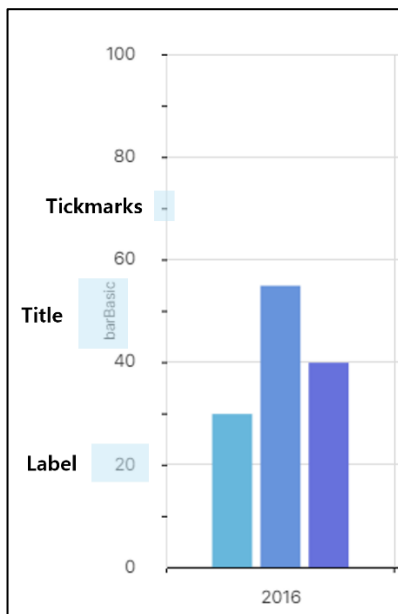
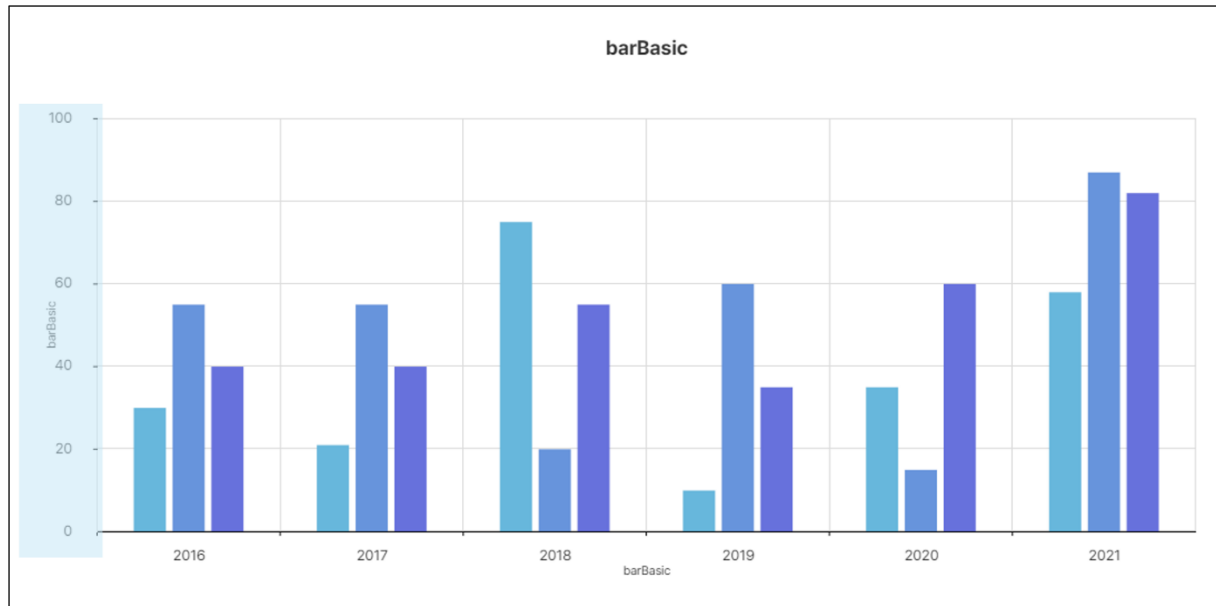
TitleY:790 // X축 Labels에 제목의 Y좌표 값을 설정 할수 있습니다. 양수, 음수 모두 설정 가능합니다

},

}

}).draw();

## 5) Y축 (옵션 – yaxis 그룹)



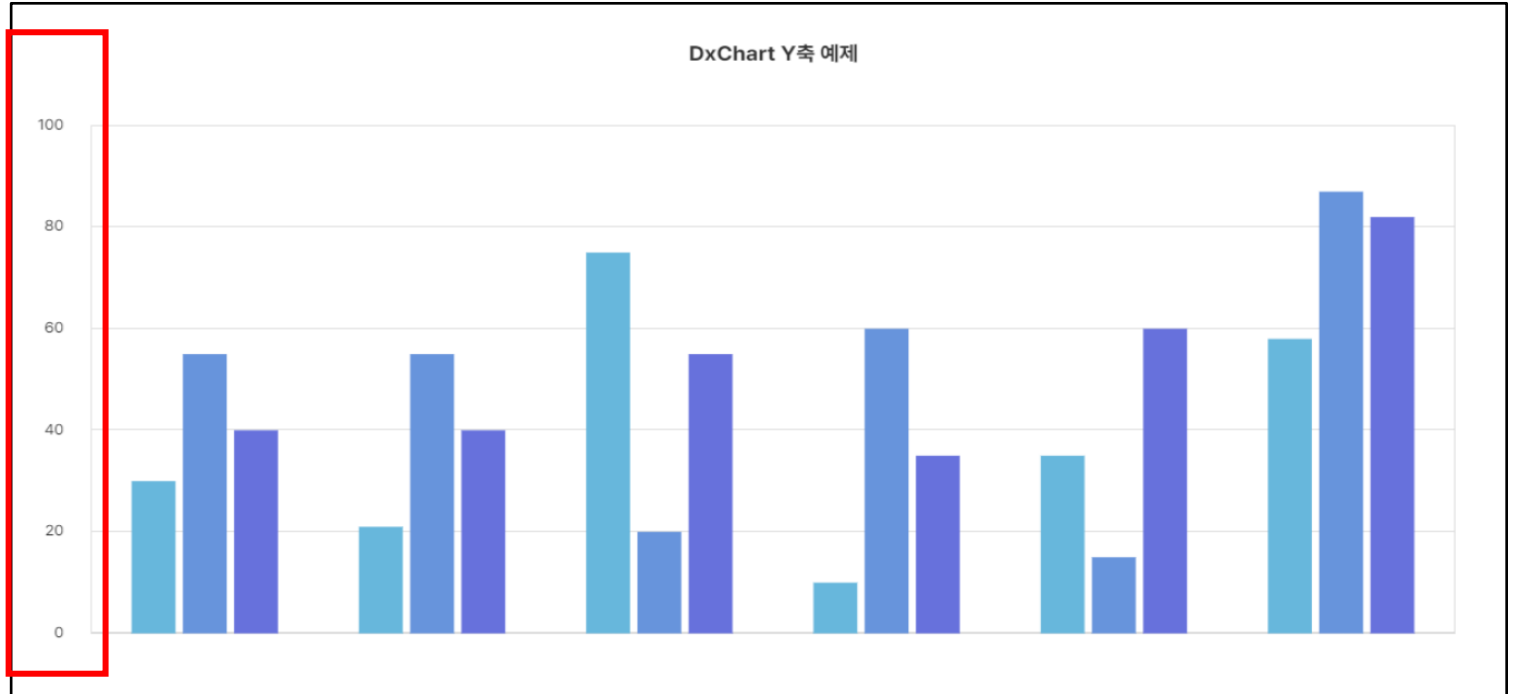
Sub property	설명	타입	기본값
AddLineSize	y축 선의 길이를 지정합니다. 양수이면 원점에서 왼쪽으로 y축 길이가 늘어나며 음수이면 원점에서 오른쪽으로 y축 길이가 줄어듭니다.	number	0
Color	y축 선의 색을 지정합니다.	string	black
LabelsColor	y축 라벨의 색을 지정합니다.	object	null
LabelsCount	y축 라벨의 개수를 지정합니다. 라벨이 많아도 넘	number	5

	어도 지정한 만큼의 개수만 보여줍니다.		
LabelsFontStyle	y축 라벨의 글꼴 스타일을 지정합니다.	string	null
LabelsHalign	y축 라벨의 가로 정렬을 지정합니다. left, right, center를 사용할 수 있습니다.	object	null
LabelsOffsetX	y축 라벨의 가로 오프셋을 지정합니다.	number	0
LabelsOffsetY	y축 라벨의 세로 오프셋을 지정합니다.	number	0
LabelsPosition	y축 라벨의 포지션을 지정합니다. section, edge가 차트 유형별로 적용되어 있으며 수정이 불가능한 차트도 있습니다.	string	edge
LabelsSpecific	y축 라벨 값을 특정한 값으로 지정합니다.	object	null
LabelsValign	라벨의 세로 정렬을 지정합니다. top, bottom, center를 사용할 수 있습니다.	object	null
LineWidth	y축의 두께를 지정합니다.	number	1
Position	y축의 위치를 지정합니다. top, bottom, center를 사용할 수 있습니다.	string	left
Scale	y축 라벨의 규모를 보여줄 지 여부입니다. true면 데이터 값에 따라 y축에 라벨이 표현됩니다.	boolean	true
ScaleDecimals	y축 라벨의 규모를 보여줄 때, 지정한 값만큼의 소수점을 표시합니다.	number	0
ScaleFormatter	y축 라벨 규모의 데이터를 특정한 형식으로 변환 하거나 서식을 적용합니다.	object	null
ScaleInvert	y축 라벨 규모를 반전시킬 지 여부입니다.	boolean	false
ScaleMax	y축 라벨의 규모를 보여줄 때, 라벨의 최대값을 지정합니다.	object	null
ScaleMin	y축 라벨의 규모를 보여줄 때, 라벨의 최소값을 지정합니다.	number	0
ScalePoint	y축 라벨의 규모를 보여줄 때, 소수점 표시 방식을 지정합니다.	string	.
ScaleRound	y축 라벨의 규모를 반올림할 지 여부입니다.	boolean	false
ScaleThousand	y축 라벨의 규모를 보여줄 때, 천 단위 구분 기호를 지정합니다.	string	,
ScaleUnitsPost	y축 라벨의 규모를 보여줄 때, 뒤에 표시할 문자를 지정합니다.	string	
ScaleUnitsPre	y축 라벨의 규모를 보여줄 때, 앞에 표시할 문자를 지정합니다.	string	
Tickmarks	y축 라벨에 눈금을 표시할 지 여부입니다.	boolean	true
TickmarksCount	y축 라벨의 눈금 개수를 지정합니다.	object	null
TickmarksLastBottom	y축 라벨의 가장 아래쪽 마크를 표시할 지 여부입	object	null

	니다.		
TickmarksLastTop	y축 라벨의 가장 위쪽 마크를 표시할 지 여부입니다.	object	null
TickmarksLength	y축 라벨의 데이터마다 위치한 마크값의 길이를 지정합니다.	number	3
Title	y축의 제목을 지정합니다.	string	
TitleColor	y축 제목의 색상을 지정합니다.	object	null
TitleFontStyle	y축 제목의 글꼴 스타일을 지정합니다.	string	null
TitleHalign	y축 제목의 가로 정렬을 지정합니다. 'left','right','center'를 사용할 수 있습니다.	object	null
TitleOffsetX	y축 제목의 가로 오프셋을 지정합니다.	number	0
TitleOffsetY	y축 제목의 세로 오프셋을 지정합니다.	number	0
TitlePos	y축 제목의 위치를 지정합니다. 양수일 때 아래로 내려가고 음수일 때 위로 올라갑니다.	object	null
TitleValign	y축 제목의 수직 정렬을 지정합니다. top, bottom, center를 사용할 수 있습니다.	object	null
TitleX	y축 제목의 가로 좌표를 지정합니다.	object	null
TitleY	y축 제목의 세로 좌표를 지정합니다.	object	null
Use	y축을 사용할 지 여부입니다.	boolean	true

### ① Y축 Property 기본 예제

다음 예제는 Basic한 차트의 Y축 생성 예제입니다.

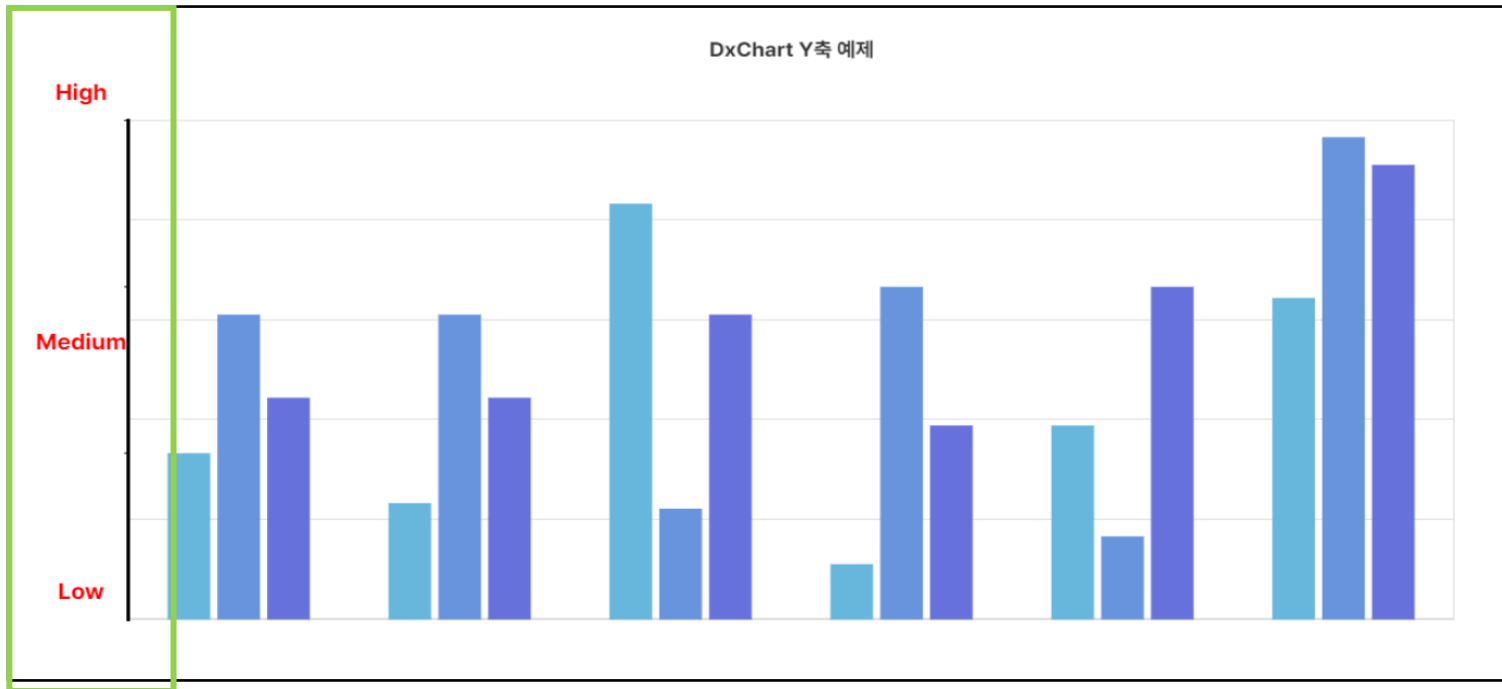


```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        yaxis: {  
            Use : true, // 차트의 Y축을 Draw 할지 설정할 수 있는 옵션입니다. 디폴트 값은 true입니다.  
            ScaleMax: 100, // 차트 Y축 라벨 데이터값의 MAX값을 설정할 수 있습니다.  
            Tickmarks:false, // 차트 Y축의 눈금선 사용 여부를 설정할 수 있습니다. 디폴트값은 true입니다.  
            TickmarksCount: 5, // 차트 Y축의 눈금선 개수 를 설정 할 수 있습니다.  
            Color:"#E2E2E2", // 차트 Y축 라인의 색상을 설정 할 수 있습니다.  
            LabelsOffsetX:-15, //차트의 Y축과 라벨 사이 OffsetX값을 설정할 수 있습니다 양수 음수 둘다 가능합니다..  
        }  
    }  
}).draw();
```



## ⑥ Y축 Property 기본 예제 (Labels 관련 예제)

다음은 차트의 Y축 라벨에 관련된 기본 예제 입니다.



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        yaxis: {  
            Use : true, // 차트의 Y축을 Draw 할지 설정할 수 있는 옵션입니다. 디폴트 값은 true입니다.  
            Tickmarks:true, // 차트 Y축의 눈금선 사용 여부를 설정할 수 있습니다. 디폴트값은 true입니다.  
            TickmarksCount: 3, // 차트 Y축의 눈금선 개수 를 설정 할 수 있습니다.  
            Color:"#E2E2E2", // 차트 Y축 라인의 색상을 설정 할 수 있습니다.  
            LabelsCount: 3, // 차트 Y축 라벨 데이터 의 개수를 설정할 수 있습니다. 기본적으로 바인딩된 데이터 값들의 평균  
                값이 라벨로 설정 됩니다.  
            LabelsFontStyle:'20px bold Pretendard', // 차트 Y축 라벨 데이터의 폰트 스타일을 설정할 수 있습니다.  
            LabelsHalign: 'center', // 차트 Y축 라벨 데이터의 가로정렬을 설정할 수 있습니다.  
            LabelsOffsetX:-35, //차트의 Y축과 라벨 사이 OffsetX값을 설정할 수 있습니다 양수 음수 둘다 가능합니다..  
            LabelsOffsetY: -15, //차트의 Y축과 라벨 사이 OffsetY값을 설정할 수 있습니다 양수 음수 둘다 가능합니다..
```

LabelsPosition: 'edge', //차트의 Y축 라벨 데이터의 포지션을 설정할 수 있습니다. 'section', 'edge'로 설정 가능합니다.

LabelsSpecific:['High','Medium','Low'], // 차트의 Y축 라벨 데이터를 바인딩된 데이터의 평균값이 아닌

특정한 값으로 설정할 수 있습니다. Array 형식으로 설정이 가능합니다.

LabelsValign:'left', //차트의 Y축 라벨데이터의 세로정렬 값을 설정할 수 있습니다.

LineWidth:3, // 차트의 Y축 색상을 설정 할 수 있습니다.

Position:'left' // 차트의 Y축 포지션을 설정할 수 있습니다.

LabelsColor:'red', //차트 Y축 라벨 데이터의 색상을 설정 할 수 있습니다.

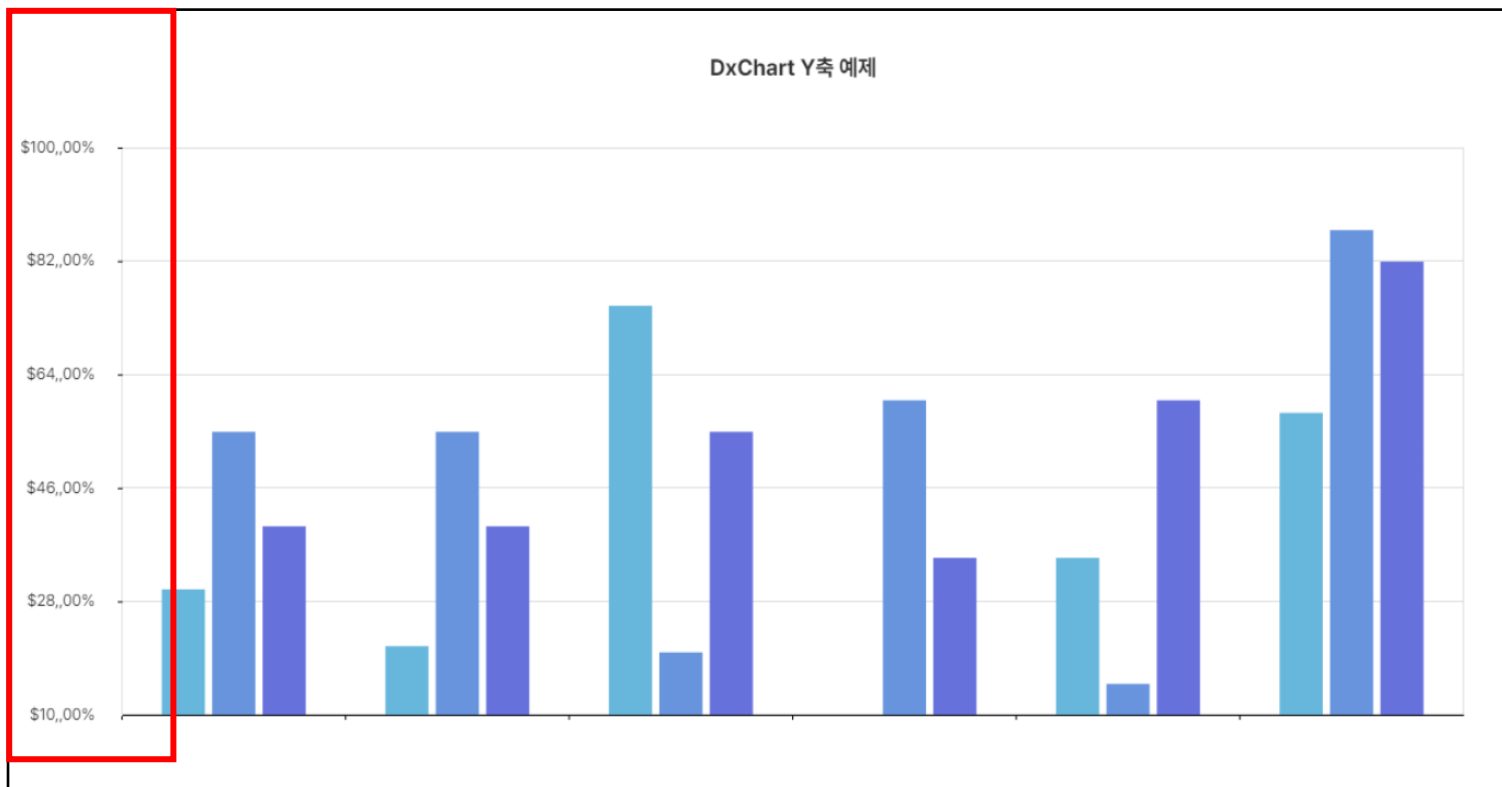
}

}

}).draw();

#### ⑦ Y축 Property 기본 예제 (Scale 관련 예제)

다음은 Y축 Scale에 관련된 기본 예제 입니다. Scale Properties는 기본적으로 Y축 라벨데이터와 연관되어 있습니다. Scale 설정에 따라서 차트의 데이터 값 표현이 달라질 수 있습니다.



```

var bar = new DxChartBar({

    id: cvs,

    elem : canvas,

    binddataset : this.Dataset00,

    data:["bind:total","bind:man","bind:woman"],

    options: {

        yaxis: {

            Use : true,

            Color:"#E2E2E2",

            LabelsOffsetX:-15, //y축과 label 사이 15px 설정

            Scale: true, //차트의 Y축 Scale값 표시 여부를 설정할 수 있습니다.

            ScaleRound: true, //차트의 Y축 Scale값을 반올림 표현을 할지 설정할 수 있습니다.
            예를들어 Scale의 최대값을 59로설정할시, 100으로 반올림하여 표현합니다.

            ScaleMax: 59, //차트의 Y축 Scale의 최대값을 설정할 수 있습니다. 설정값에 따라 차트의 데이터 표현형
            태가 달라지므로, 차트의 형태도 달라질수 있습니다.

            ScaleMin:10, , //차트의 Y축 Scale의 최소값을 설정할 수 있습니다. 설정값에 따라 차트의 데이터 표현형
            태가 달라지므로, 차트의 형태도 달라질수 있습니다.

            ScaleUnitsPost: '%', //차트의 Y축 Scale 값 뒤에 문자를 설정 할 수 있습니다.

            ScaleUnitsPre:'$', //차트의 Y축 Scale 값 앞에 문자를 설정 할 수 있습니다.

            ScaleDecimals: 2, //차트의 Y축 Scale 소수점 자릿 수를 설정 할 수 있습니다..

            ScaleInvert:false, //차트의 Y축 Scale의 최대값과 최소값을 Invert하여 표시할 수 있습니다.

            ScalePoint: ',', //차트의 Y축 Scale 값 소수점 구분점을 설정할 수 있습니다.

            ScaleThousand:'.' //차트의 Y축 Scale 값 천단위 구분점을 설정할 수 있습니다.

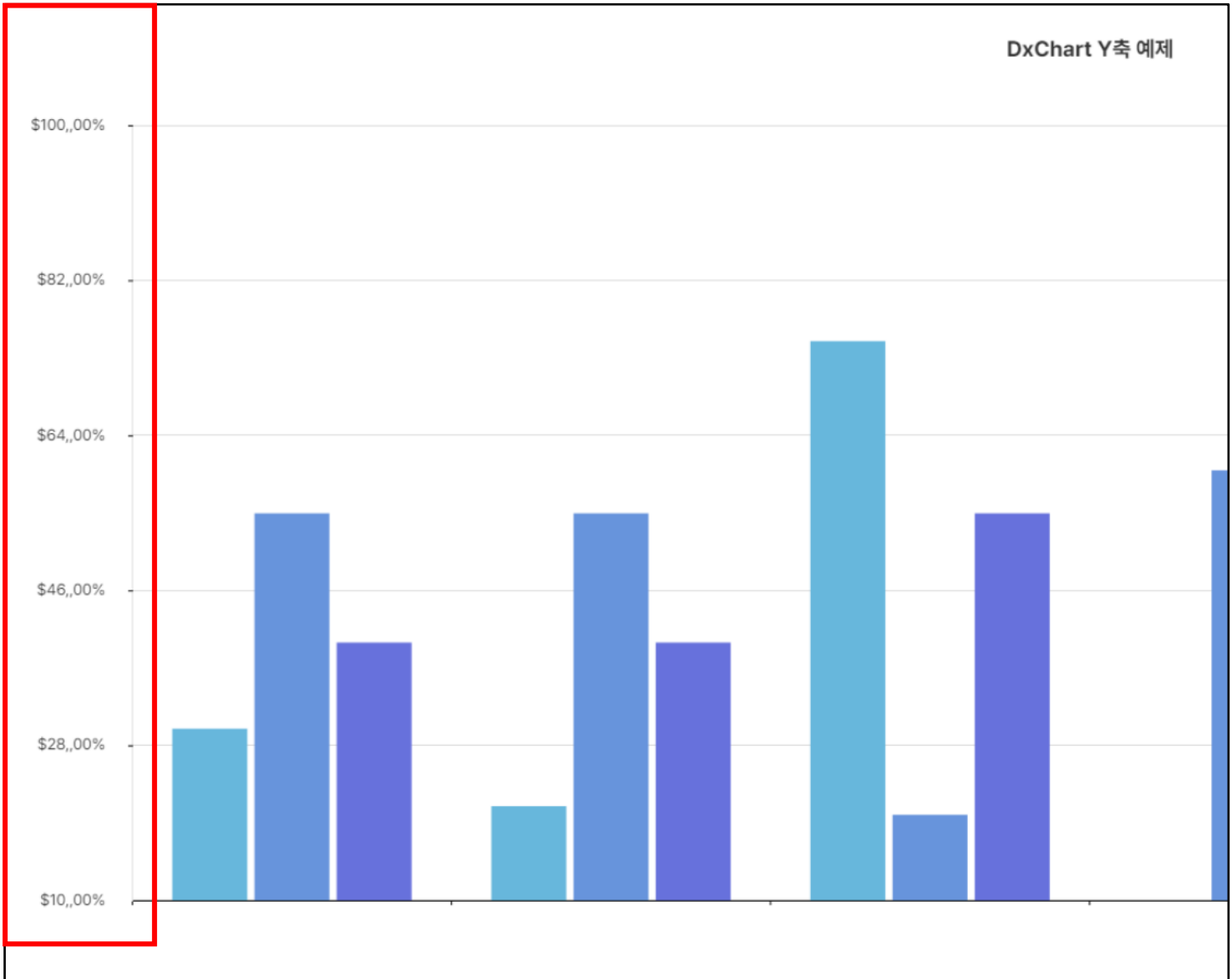
        }

    }
}

```

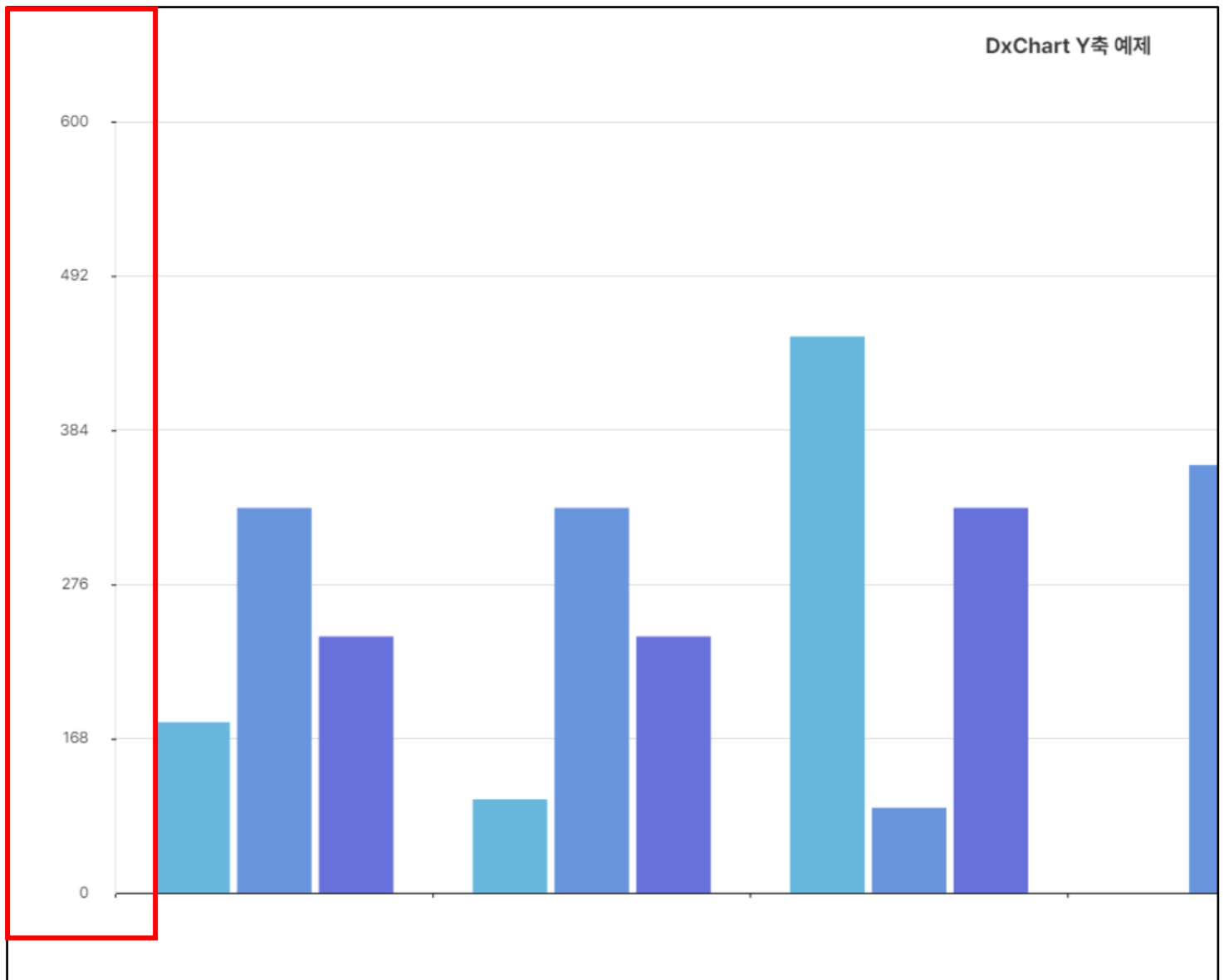
⑧ Y축 Property ScaleFormatter예제 (Scale 관련 예제)

ScaleFormatter는 Y축 Scale 데이터 값을 바인딩된 데이터값과 다르게 표현할수 있는 Properties 입니다. 옵션 객체 안에서 함수형태로 선언하여 사용할 수 있습니다.



앞선 예제의 Y축 Scale값입니다. ScaleFormatter 프로퍼티를 통해서 해당 값을 다른값으로 표현할수 있습니다.

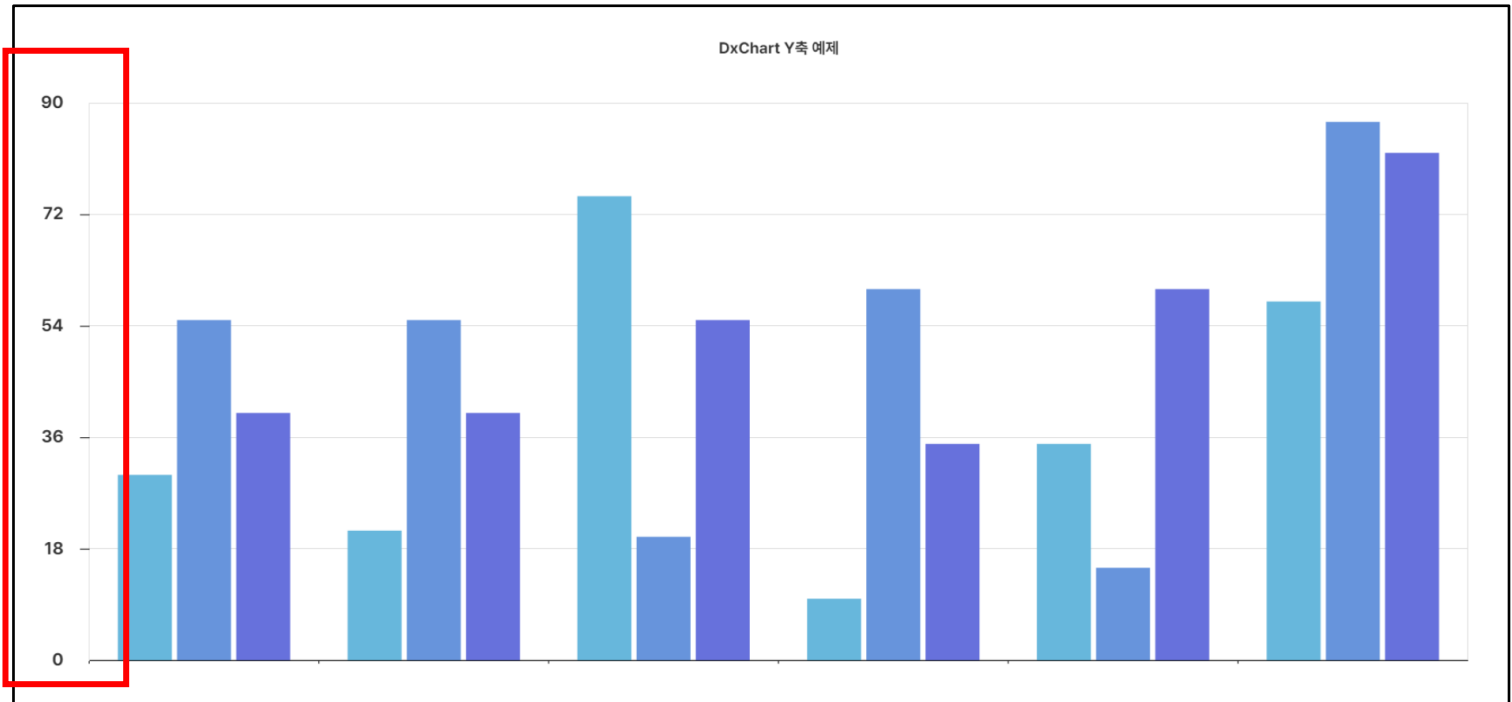
```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        //기본적으로 ScaleFormatter는 yaxis에 관련된 프로퍼티이기 때문에 yaxis 객체안에 선언합니다.  
        yaxis: {  
            ScaleFormatter : function (e)  
            {  
                //yaxis의 Scale값은 바인딩된 데이터 값이기 때문에, 기본적으로 Number형식을 따릅니다  
                var num = Number(e.number) * 6; //매개변수 e는 바인딩된 데이터값이 넘어오게되며, 설정하고  
                //싶은 값을 변수에 담아줍니다.  
                return String(num) //설정된 값을 String으로 변환해 반환합니다.,  
            }  
        }  
    }  
}).draw();
```



기존 Sclae값에 \*6을 한 데이터를 반환한 결과입니다. SclaeFormmater를 설정하게되면, 기존에 설정했던 Scale 구분점 값이나, Post, Pre값들은 초기화 되게 됩니다.

이와 같이 Scale 데이터 값을 특정 값으로 표현할 수 있습니다.

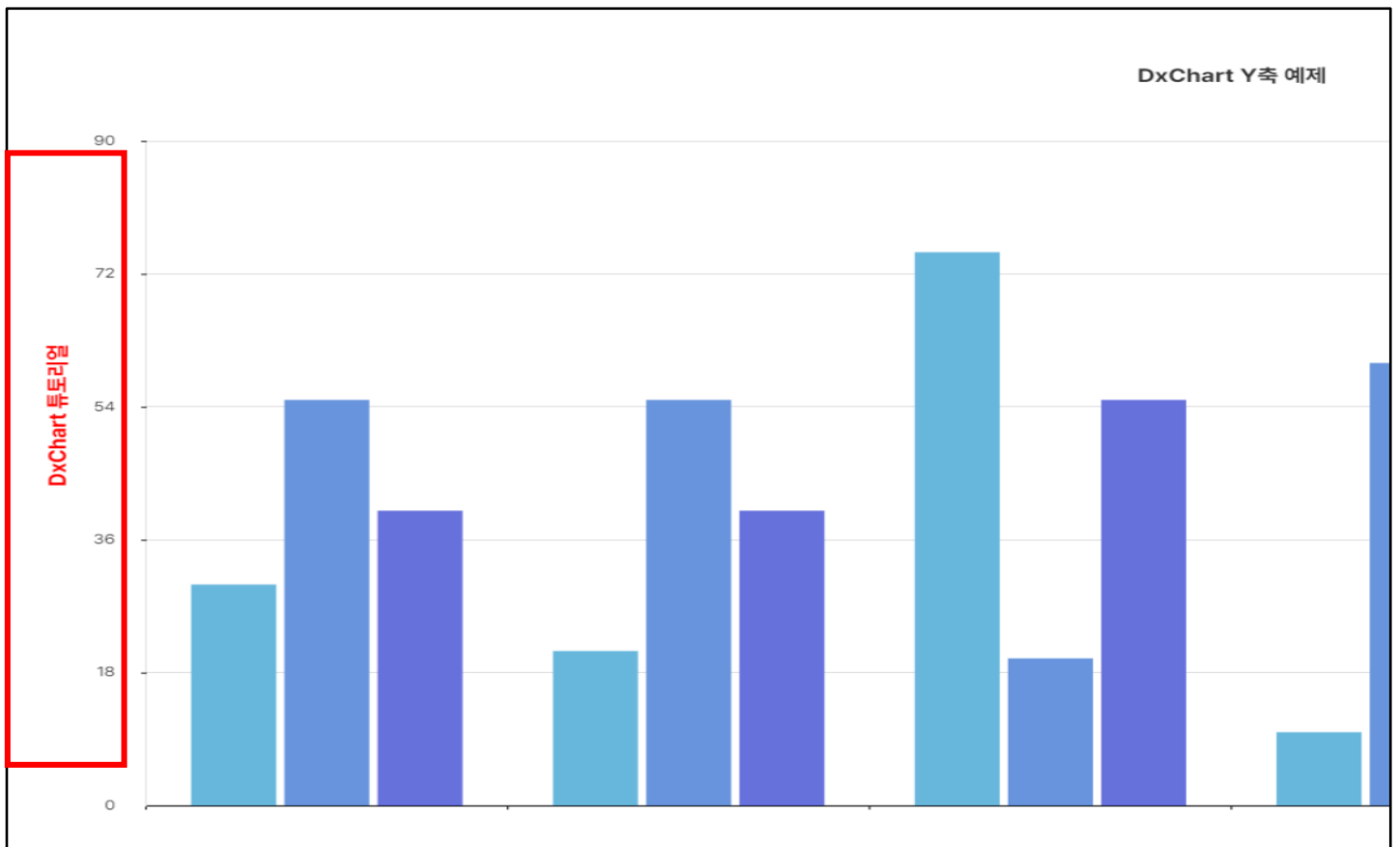
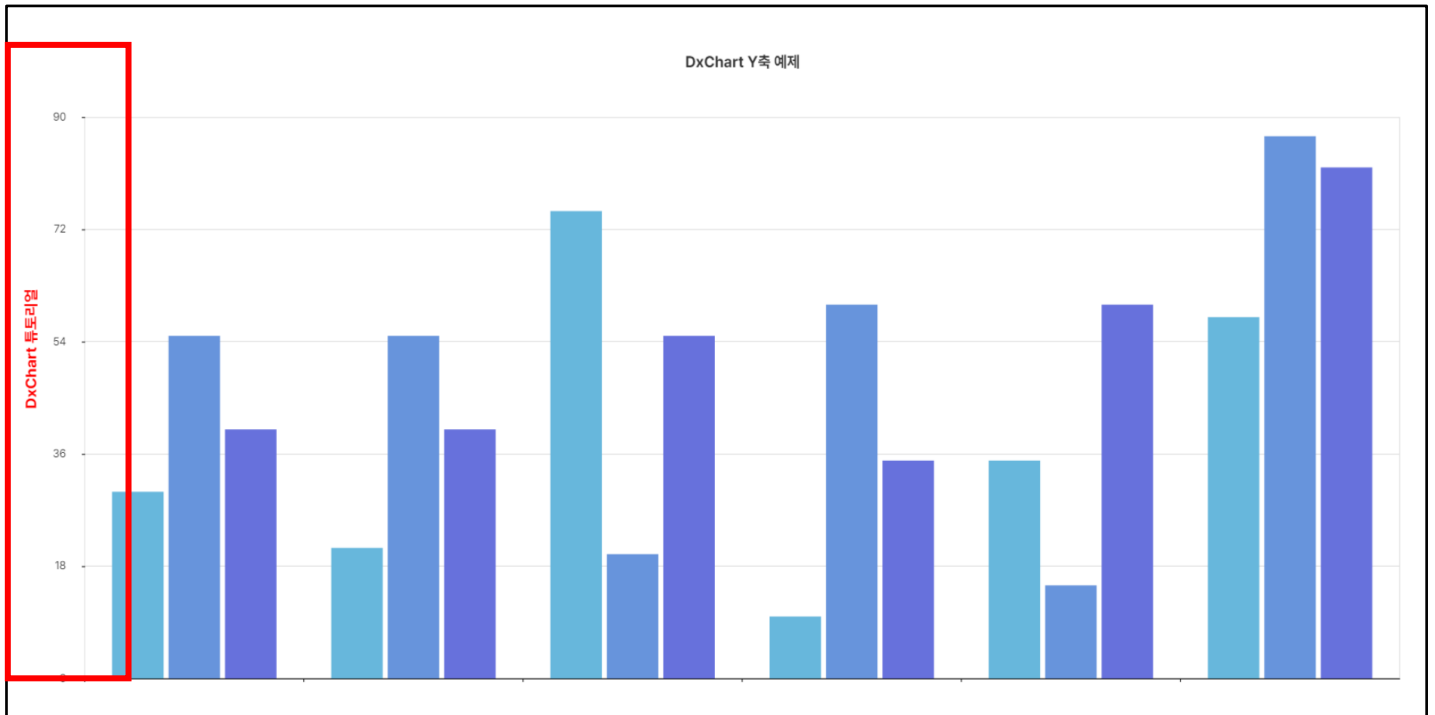
⑨ Y축 Property 기본 예제 (Tickmarks 관련 예제)



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00,  
    data:["bind:total","bind:man","bind:woman"],  
    options: {  
        yaxis: { //기본적인 설정은 Basic예제와 동일합니다  
            Tickmarks: true, //차트 Y축의 눈금선 표시여부를 설정할 수 있습니다.  
            TickmarksCount: 5, //차트 Y축의 눈금선 표시여부를 설정할 수 있습니다.  
            TickmarksLastBottom:false, //차트 Y축의 눈금선 표시여부를 설정할 수 있습니다.  
            TickmarksLastTop:false, //차트 Y축의 눈금선 표시여부를 설정할 수 있습니다.  
            TickmarksLength: 10, //차트 Y축의 눈금선 표시여부를 설정할 수 있습니다.  
            LabelsFontStyle: '20px bold Pretendard'  
        }  
    }  
}).draw();
```

⑩ Y축 Property 기본 예제 (title 관련 예제)

다음은 Y축 title에 관한 기본 예제입니다.





```

var bar = new DxChartBar({

    id: cvs,

    elem : canvas,

    binddataset : this.Dataset00,

    data:["bind:total","bind:man","bind:woman"],

    options: {

        yaxix: { //기본적인 설정은 Basic예제와 동일합니다

            Use : true,

            Color:"#E2E2E2",

            LabelsOffsetX:-15, //y축과 label 사이 15px 설정

            Title:'DxChart 튜토리얼', //차트 Y축 타이틀을 설정할 수 있습니다.

            TitleColor:'red', //차트 Y축 타이틀을 색상을 설정할 수 있습니다.

            TitleFontStyle:'18px bold Pretendard', //차트 Y축 타이틀 폰트 스타일을 설정할 수 있습니다.

            TitleHalign:'center', //차트 Y축 타이틀 가로정렬을 설정할수 있습니다. 'left','right','cetner'.

            TitleOffsetX:80, //차트 Y축 타이틀 OffsetX 값을 설정할 수 있습니다.

            TitleOffsetY:-20, //차트 Y축 타이틀 OffsetY 값을 설정할 수 있습니다.

            TitlePos:-20, //차트 Y축 타이틀을 설정할 수 있습니다.

            TitleValign:'cetner', //차트 Y축 타이틀 세로정렬 값 을 설정할 수 있습니다.

            TitleX:50, //차트 Y축 타이틀 X좌표값 을 설정할 수 있습니다.

            TitleY:400 //차트 Y축 타이틀 Y좌표값 을 설정할 수 있습니다.

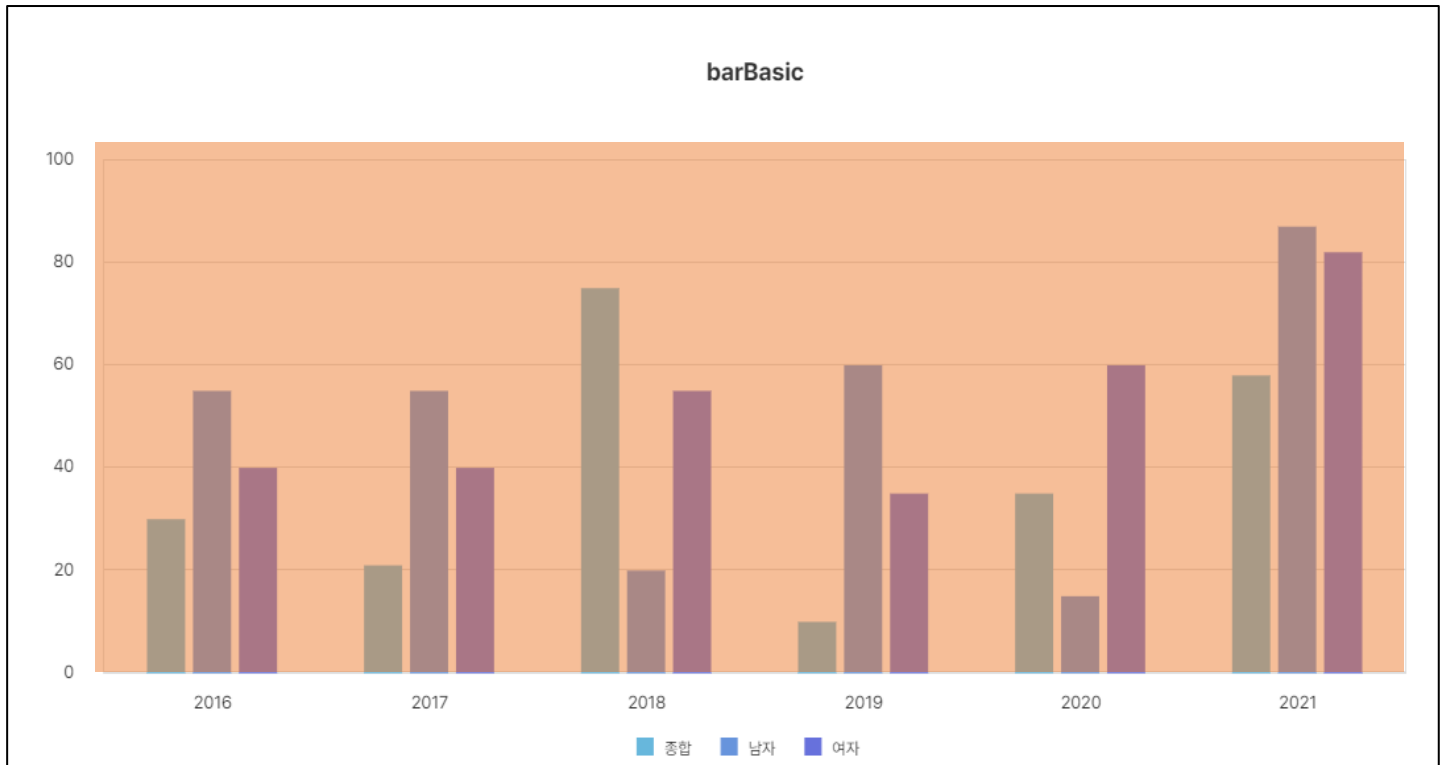
        }

    }

}).draw();

```

## 6) Background 설정하기



차트의 Background를 설정합니다.

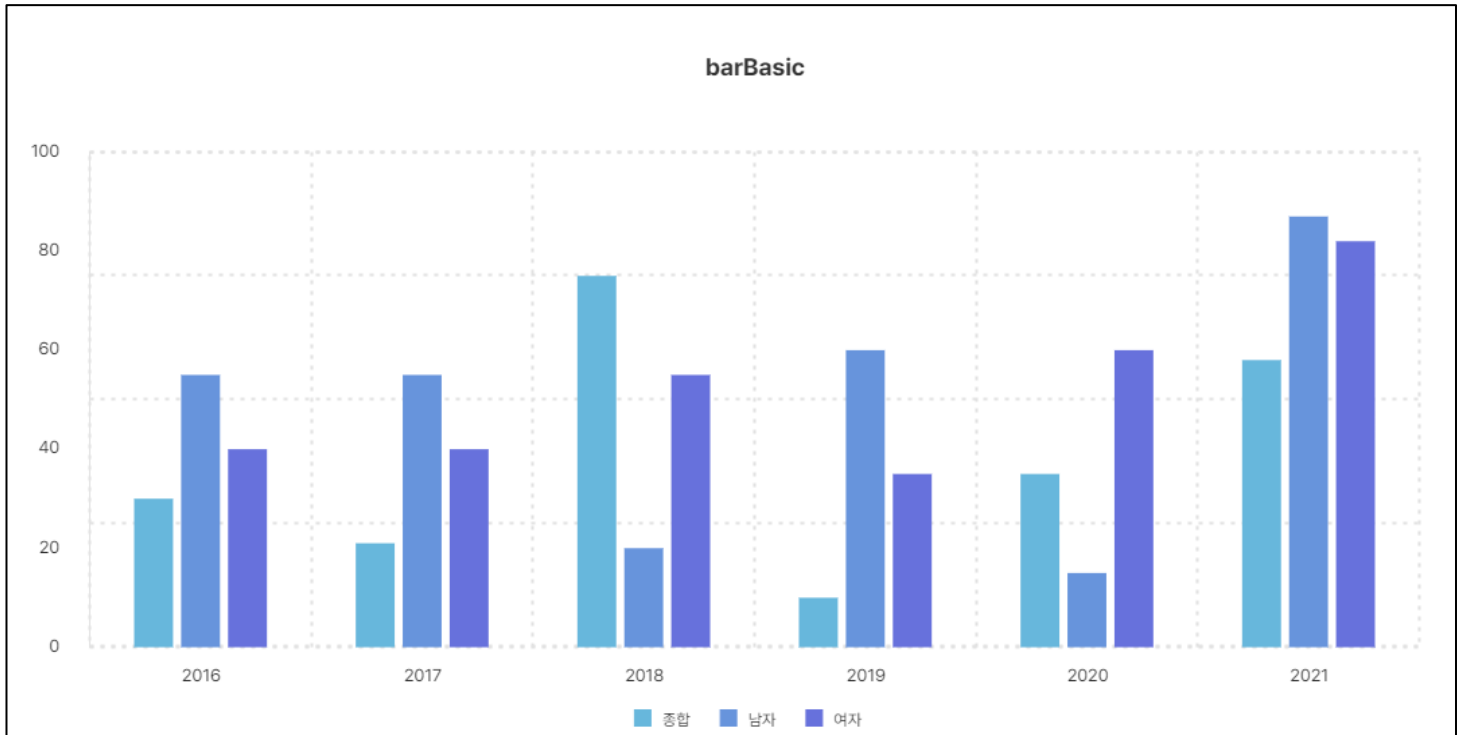
### Properties

Sub property	설명	타입	기본값
BarsColor1	배경에서 막대의 가로 색상을 지정합니다. 지정한 색이 홀수 번째에 반복됩니다.	string	rgba(0,0,0,0)
BarsColor2	배경에서 막대의 가로 색상을 지정합니다. 지정한 색이 짝수 번째에 반복됩니다.	string	rgba(0,0,0,0)
BarsCount	배경의 막대 개수를 지정합니다.	object	null
Color	차트의 여백을 제외한 배경 색을 지정합니다.	object	null
Grid	배경에 그리드를 표시할 지 여부입니다.	number	1
GridAutofit	배경 그리드의 픽셀 너비/높이를 지정하는 대신 자동 맞춤을 사용할 지 여부입니다.	boolean	TRUE
GridAutofitAlign	배경 그리드 autofit을 사용했을 때 정렬을 사용할 지 여부입니다.	boolean	TRUE
GridBorder	배경 그리드 주위에 테두리를 그릴지 여부입니다.	boolean	TRUE
GridColor	배경 그리드의 색을 지정합니다.	string	#dddddd
GridDashed	배경 그리드를 대시(-) 형태로 표현할 지 여부입니다.	boolean	FALSE
GridDotted	배경 그리드를 점(.) 형태로 표현할 지 여부입니다.	boolean	FALSE
GridHlines	배경 그리드의 가로 선을 그릴지 여부입니다.	boolean	TRUE

GridHlinesCount	배경 그리드의 가로 선 수를 지정합니다.	number	5
GridHsize	배경 그리드의 가로 크기를 지정합니다. backgroundGridAutofit 옵션이 false일 때만 동작합니다.	number	25
GridLineWidth	배경 그리드 선의 두께를 지정합니다. 소수점도 허용됩니다.	number	1
GridThreedYaxis	3D 차트를 그렸을 때 배경 그리드가 Y축으로 확장할 지 여부입니다.	boolean	TRUE
GridVlines	배경 그리드의 세로 선을 그릴지 여부입니다.	boolean	TRUE
GridVsize	배경 그리드의 세로 크기를 지정합니다. backgroundGridAutofit 옵션이 false일 때만 동작합니다.	number	25
Hbars	배경 그리드의 특정 위치에 특정 크기의 가로 막대를 지정합니다. 첫 번째 값은 데이터 범위 내에 있어야 표시됩니다. 두 번째 값은 크기를 지정하며 마지막 값은 색을 지정합니다. []안에 ,로 구분하여 여러 막대를 여러 위치에 그릴 수 있습니다. background : {Hbars : [[70, 10, 'yellow'], [120, 20, 'blue']]}	object	[]
Image	배경에 사용할 이미지를 지정합니다.	object	null
ImageAlign	배경 이미지를 정렬을 지정합니다. top, bottom, left, right 등을 기본적으로 사용 가능하며 top left, bottom right 등 응용도 가능합니다.	object	null
ImageAlpha	배경 이미지의 투명도를 지정합니다. 0~1까지 지정할 수 있으며 작아질수록 투명해집니다.	number	1
ImageH	배경 이미지의 높이를 지정합니다.	object	null
ImageStretch	배경 이미지가 차트에 맞게 확장될 지 여부입니다.	boolean	TRUE
ImageW	배경 이미지의 길이를 지정합니다.	object	null
ImageX	배경 이미지의 가로 좌표를 지정합니다.	object	null
ImageY	배경 이미지의 세로 좌표를 지정합니다.	object	null

### ① Background Grid 속성

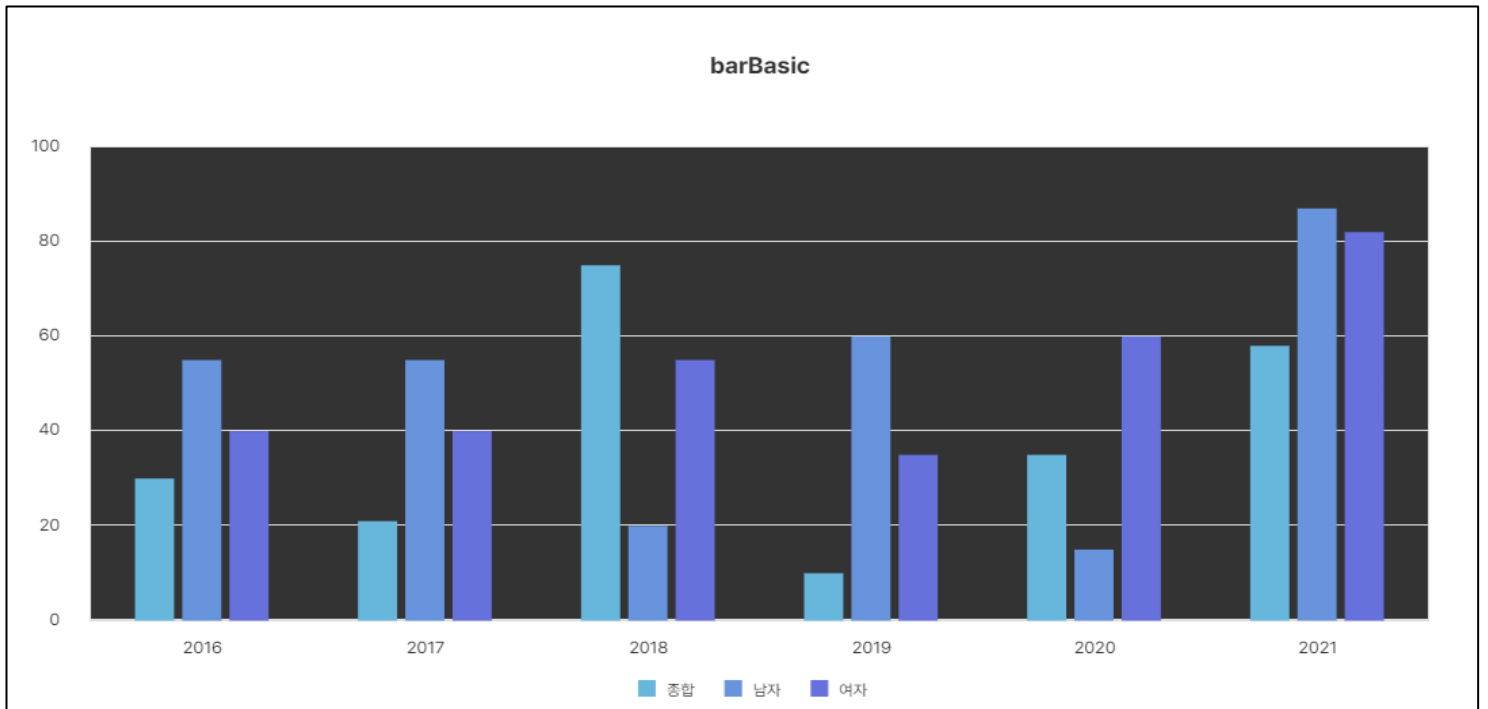
Background Grid 속성을 설정할 수 있습니다.



```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        //기본적인 설정은 Basic예제와 동일합니다
        background:{
            Grid: true           //Grid 옵션을 설정합니다.
            ,GridColor:"#E2E2E2" //Grid의 색상을 설정합니다.
            ,GridDashed:true     //Grid의 점 및 선 형태를 설정합니다.
            ,GridDotted:false    //Grid의 점 및 선 형태를 설정합니다.
            ,GridHlinesCount:4   //Grid의 가로 선의 개수를 지정합니다.
            ,GridLineWidth:2     //Grid의 선의 굵기를 지정합니다.
        }
    }
}).draw();
```

## ② Background 색상 설정하기

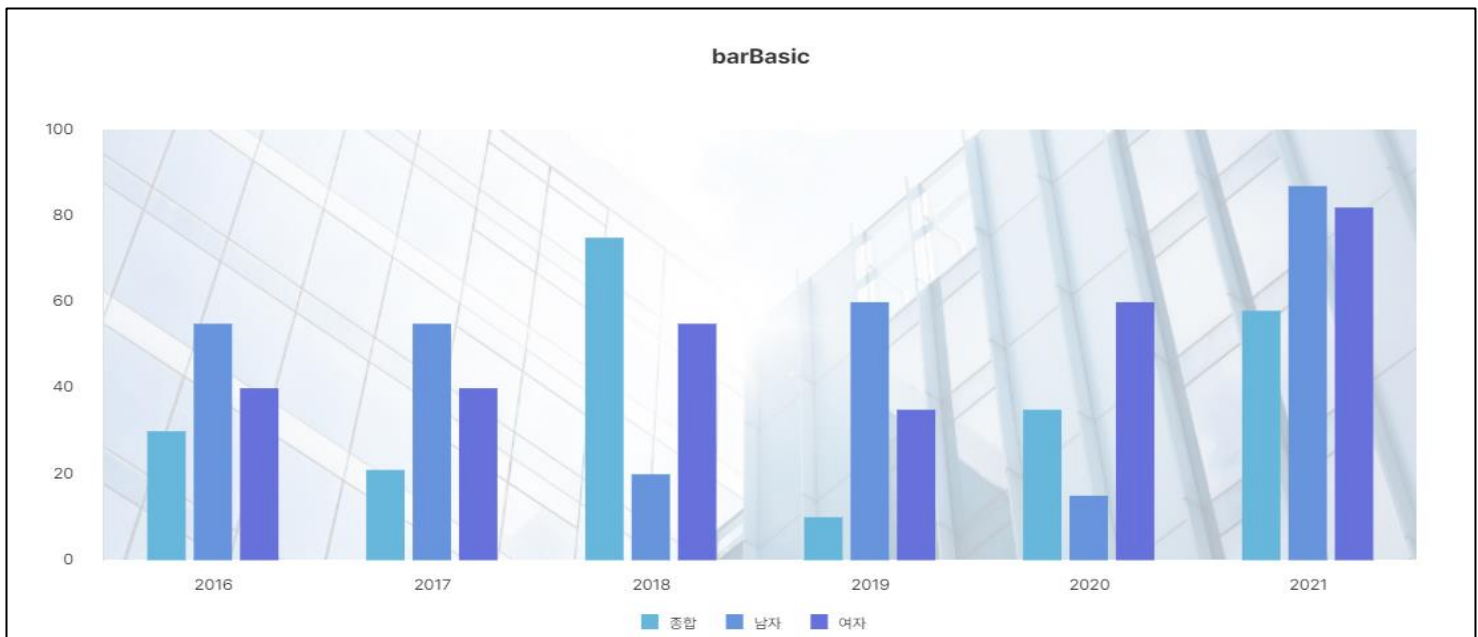
Background 색상 지정 가능합니다.



```
var bar = new DxChartBar({
  id: cvs,
  elem : canvas,
  binddataset : this.Dataset00,
  data:["bind:total","bind:man","bind:woman"],
  options: {
    //기본적인 설정은 Basic예제와 동일합니다
    background:{
      Color:"#333" //BackGroud Color를 설정합니다.
      ,GridVlines: false //Grid의 수직 선을 미설정합니다
      ,GridBorder: true //Grid Border를 설정합니다..
    }
  }
}).draw();
```

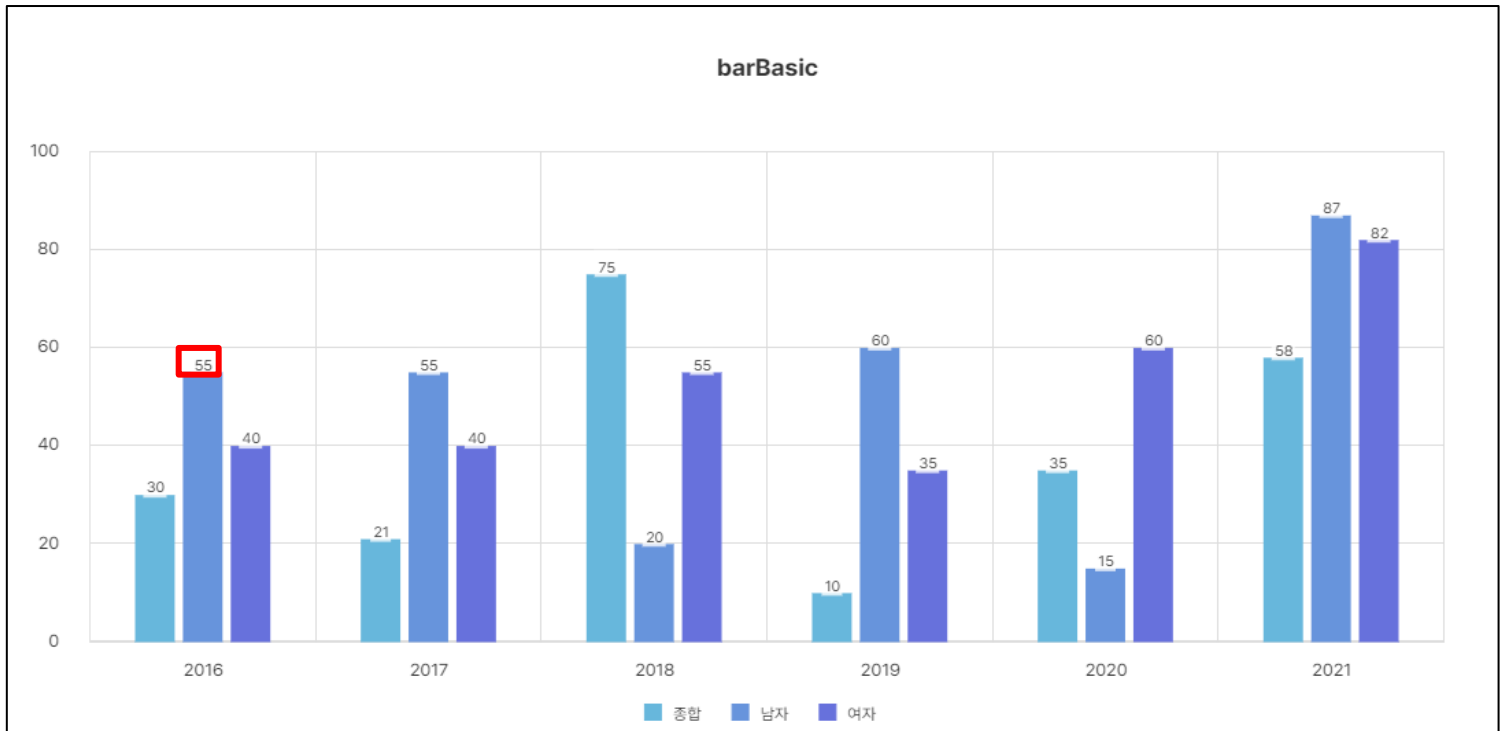
### ③ Background Image 설정하기

Background Image 설정이 가능합니다.



```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        //기본적인 설정은 Basic예제와 동일합니다
        background:{
            Image: 'images::barBg1.png' //이미지 경로를 설정합니다.
            , ImageAlpha: 0.3           //이미지의 알파 값을 설정합니다.
            , Grid: false                //Grid는 사용안함으로 설정합니다.
        }
    }
}).draw();
```

## 7) Label 설정하기



## Properties

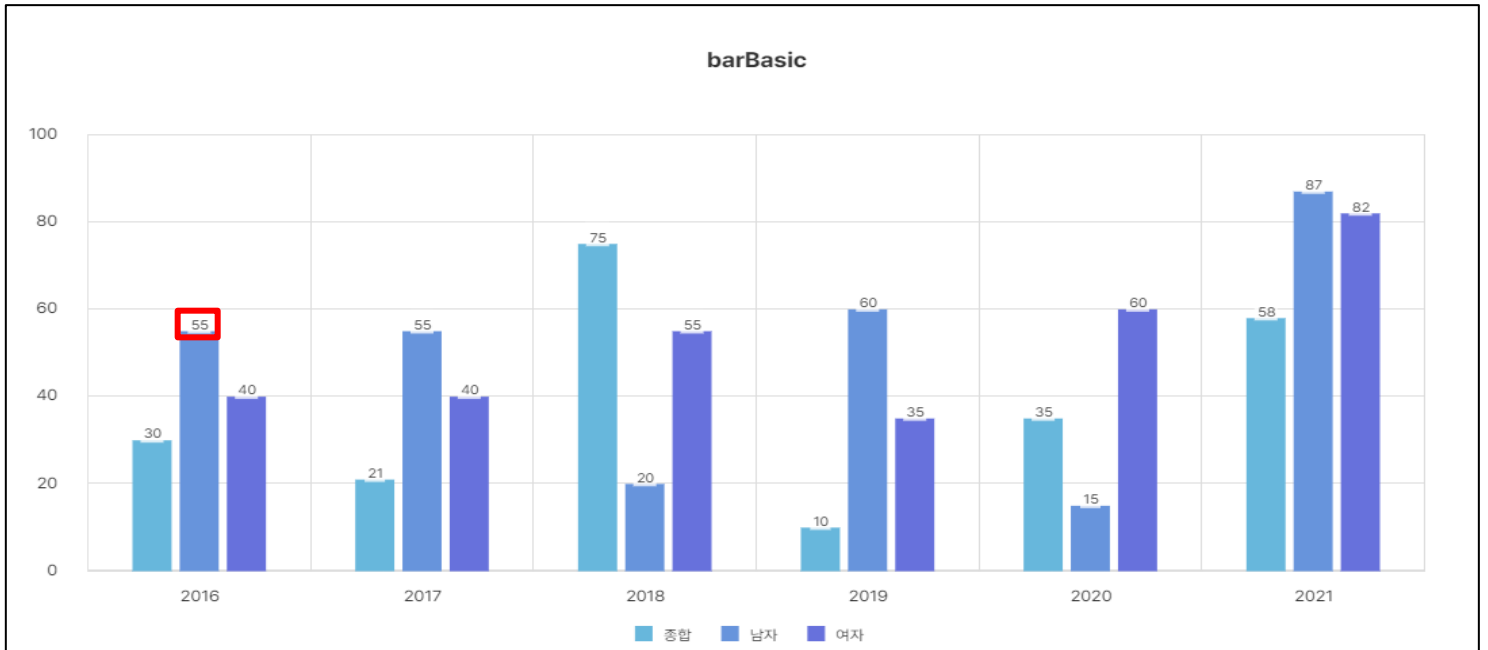
Sub property	설명	타입	기본값
Above	차트 요소 위에 라벨을 표시할 지 여부입니다.	boolean	FALSE
AboveAngle	차트 요소 위에 라벨을 표시했을 때, 라벨의 각도를 지정합니다.	object	null
AboveBackground	차트 요소 위에 라벨을 표시했을 때, 라벨의 배경 색을 지정합니다.	string	rgba(0,0,0,0)
AboveBorderStyle	차트 요소 위에 라벨을 표시했을 때, 라벨의 테두리 스타일을 지정합니다.	string	null
AboveColor	차트 요소 위에 라벨을 표시했을 때, 라벨 텍스트의 색을 지정합니다.	object	null
AboveDecimals	차트 요소 위에 라벨을 표시했을 때, 지정한 값만큼의 소수점을 표시합니다.	number	0
AboveFontStyle	차트 요소 위에 라벨을 표시했을 때, 라벨의 글꼴 스타일을 지정합니다.	string	null
AboveFormatter	차트 요소 위에 라벨을 표시했을 때, 라벨의 데이터를 특정한 형식으로 변환하거나 서식을 적용합니다.	object	null
AboveOffset	차트 요소 위에 라벨을 표시했을 때, 라벨의 오프셋을 지정합니다. 이 오프셋을 지정하면 세로	object	null

	오프셋을 지정한 것과 동일합니다.		
AboveOffsetX	차트 요소 위에 라벨을 표시했을 때, 라벨의 가로 오프셋을 지정합니다.	number	0
AboveOffsetY	차트 요소 위에 라벨을 표시했을 때, 라벨의 세로 오프셋을 지정합니다.	number	0
AbovePoint	차트 요소 위에 라벨을 표시했을 때, 라벨의 소수점이 나올 자리의 구분 기호를 지정합니다.	string	.
AboveSpecific	차트 요소 위에 라벨을 표시했을 때, 라벨 값을 특정한 값으로 지정합니다.	object	null
AboveThousand	차트 요소 위에 라벨을 표시했을 때, 라벨의 천 단위 구분 기호를 지정합니다.	string	,
AboveUnitsPost	차트 요소 위에 라벨을 표시했을 때, 라벨의 데이터 뒤에 표시할 문자를 지정합니다.	string	
AboveUnitsPre	차트 요소 위에 라벨을 표시했을 때, 라벨의 데이터 앞에 표시할 문자를 지정합니다.	string	
Ingraph	차트 내부에 그려지는 라벨을 사용할 지 여부입니다.	object	null
IngraphColor	차트 내부 라벨의 색을 지정합니다.	object	null
IngraphFontStyle	차트 내부 라벨의 글꼴 스타일을 지정합니다.	string	null
IngraphOffsetX	차트 내부 라벨의 가로 오프셋을 지정합니다.	number	0
IngraphOffsetY	차트 내부 라벨의 세로 오프셋을 지정합니다.	number	0



### ① Label Above 설정하기

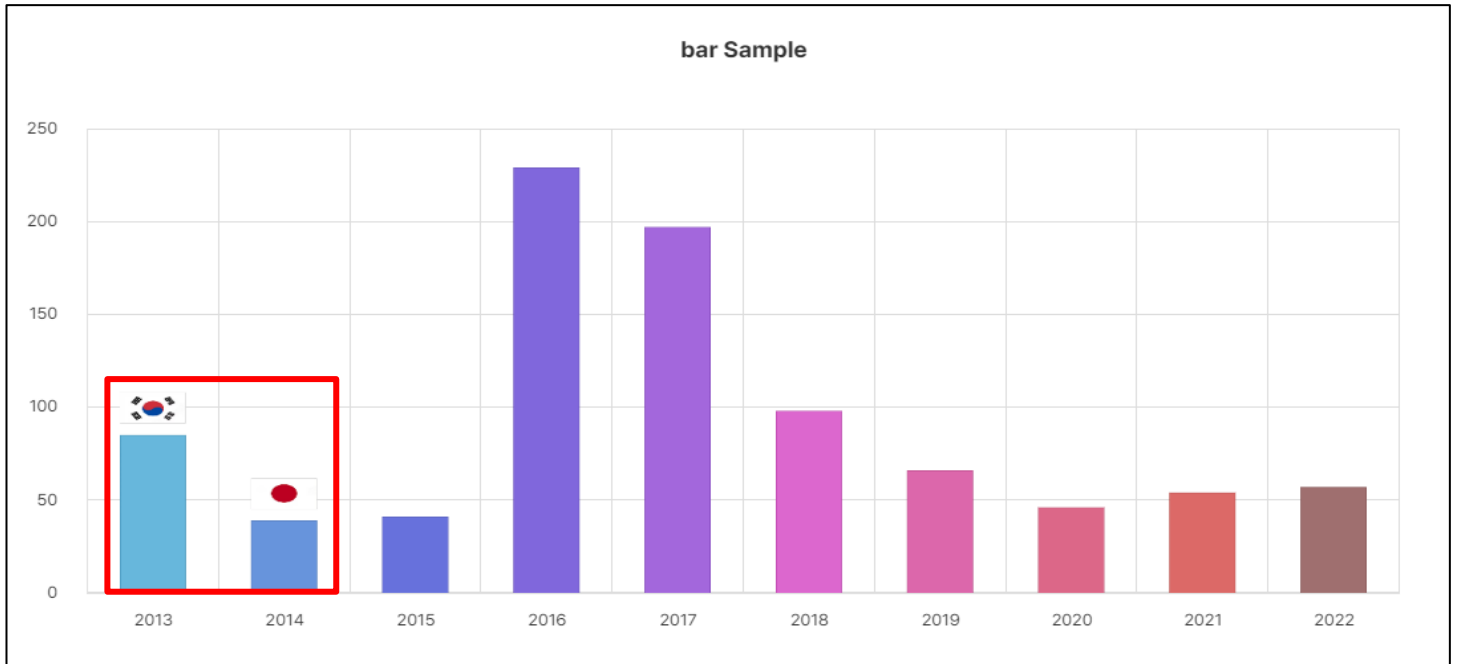
Label를 해당 차트 위에 표시할 수 있습니다.



```
var bar = new DxChartBar({
  id: cvs,
  elem : canvas,
  binddataset : this.Dataset00,
  data:["bind:total","bind:man","bind:woman"],
  options: {
    //기본적인 설정은 Basic예제와 동일합니다
    labels:{
      Above:true //labelsAbove 설정 합니다.
      AboveBackground: 'rgba(255,255,255,0.75)' //labelsAbove의 Background 색상을 설정합니다.
    }
  }
}).draw();
```

## ② Lable Above Image 설정하기

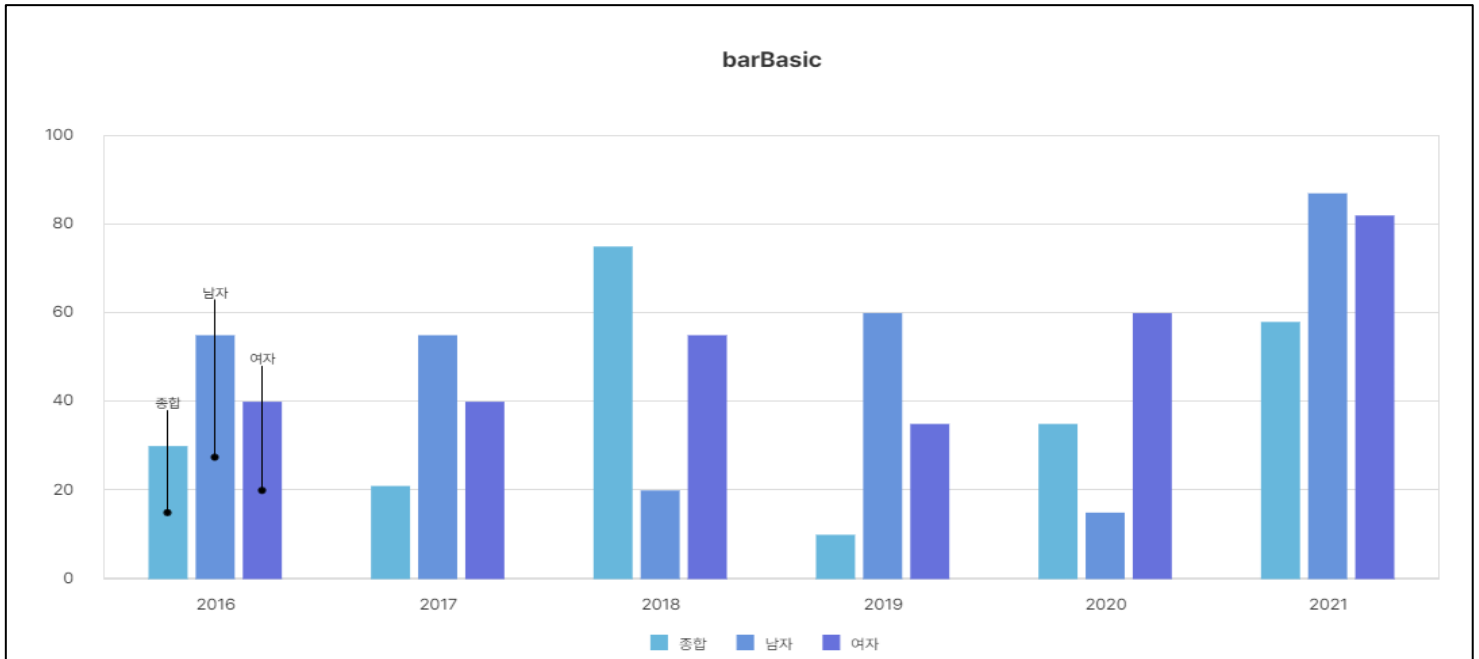
Lable의 이미지를 포함하여 차트 위에 표시할 수 있습니다. (\*Single bar만 지원합니다)



```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:val"],
    options: {
        xaxis:{Labels: ['bind:year'],Color:"#E2E2E2",LabelsOffsetY:5,Tickmarks:false}, //Xaxis를 설정합니다.
        yaxis:{Color:"#E2E2E2",LabelsOffsetX:-15,Tickmarks:false}, //Yaxis를 설정합니다.
        margin:{ //margin은 Basic예제와 동일합니다 },
        title:{Text:"bar Sample",Y :73,SubtitleOffsetY:8 }, //Title를 설정합니다.
        colorsSequential: true, // colorsSequential true 설정 시 default color 순차적으로 보여줍니다.
        labels:{
            Above: false, //labels Above를 미설정합니다.
            AboveImage : true, //labels Image를 설정합니다.
            AboveImageGap : 10, //labels Above 이미지 사이 gap을 설정합니다.
            AboveImageUrl :[ 'images::nation/korea.png','images::nation/japan.png'], //labels 이미지 url를 설정합니다.
            AboveImageWidth : 100, //labels Above 이미지 너비를 설정합니다.
            AboveImageHeight : 50, //labels Above 이미지 높이 설정합니다.
            AboveImageLineWidth : 0}, //labels Above 이미지 테두리 선의 굵기를 설정합니다.
        }
    }
}).draw();
```

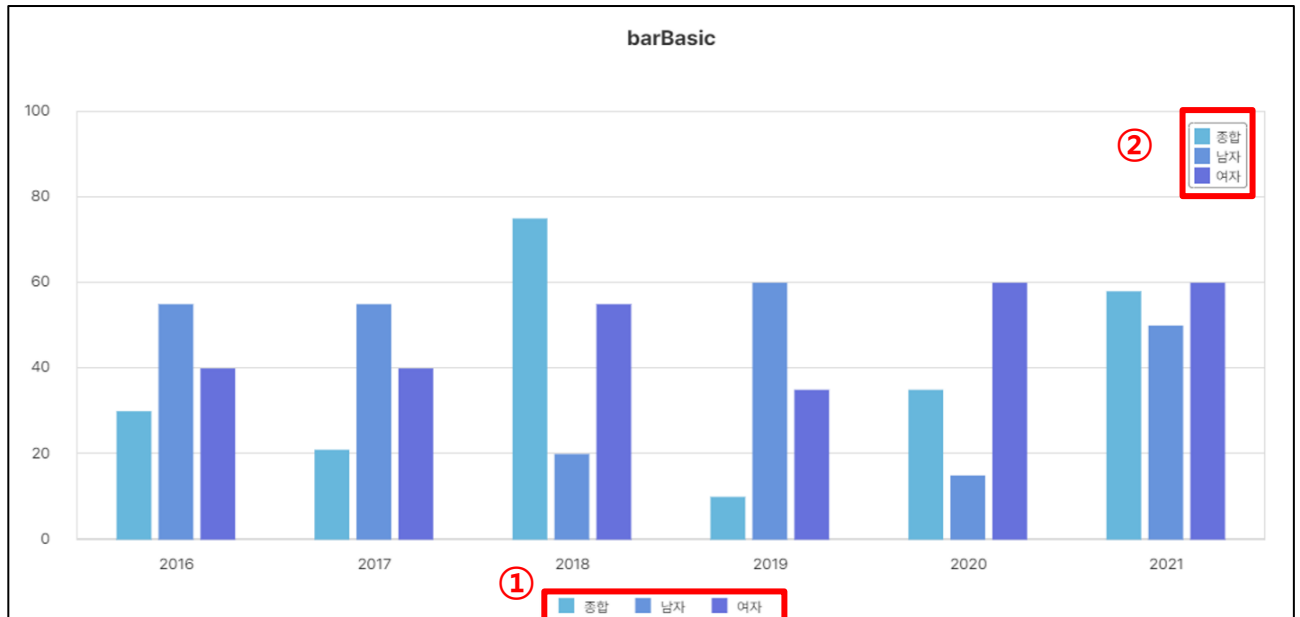
### ③ Lable Ingraph 설정하기

Lable 차트 안에 표시할 수 있습니다.



```
var bar = new DxChartBar({
  id: cvs,
  elem : canvas,
  binddataset : this.Dataset00,
  data:["bind:total","bind:man","bind:woman"],
  options: {
    //기본적인 설정은 Basic예제와 동일합니다
    labels:{
      Ingraph : ['총합','남자','여자'], //Ingraph 데이터를 설정합니다.
      IngraphLabelTop : true,          //Ingraph 레이블 위치를 설정합니다.( Top false 시 차트 안에서 표시)
      IngraphTextBound:false,         //IngraphTextBound 데이터를 설정합니다.
    }
  }
}).draw();
```

## 8) 범례 설정하기



차트의 범례를 설정합니다.

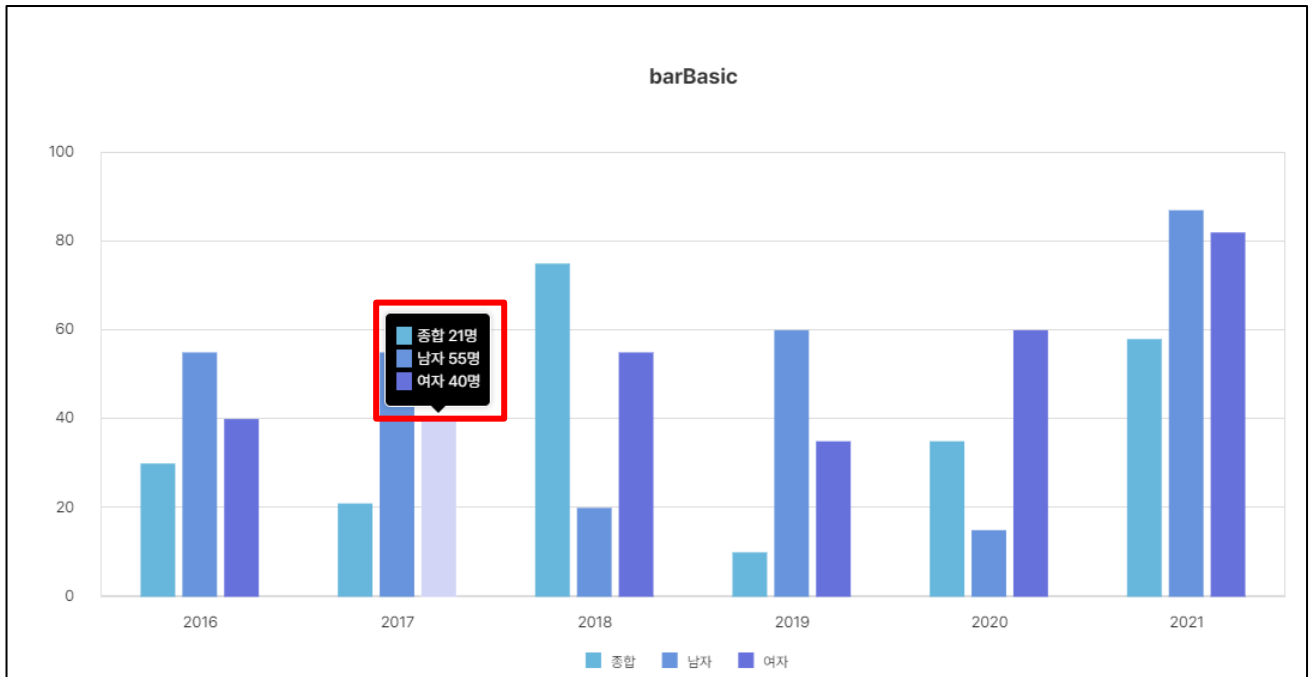
### Properties

Sub property	설명	타입	기본값
Background	범례의 배경 색을 지정합니다.	string	white
Colors	범례의 아이콘 색을 지정합니다. 지정하지 않으면 요소의 색을 따라갑니다. 데이터셋 수만큼 지정할 수 있습니다. 데이터셋이 2개일때의 예시가 있습니다.	object	null
ColorShape	범례의 아이콘 모양을 지정합니다. 기본은 square이며, circle, triangle, line 등을 사용할 수 있습니다.	string	square
Data	범례에 사용 될 라벨 데이터를 지정합니다. binddataset을 사용하기 때문에 배열 안에 해당 데이터셋의 컬럼 값을 사용합니다. 또는 임의로 지정할 수 있습니다.	object	null
Halign	범례의 세로 정렬을 지정합니다. left와 right를 사용할 수 있습니다.	object	null
Interactive	범례의 라벨을 클릭했을 때 상호작용할 지 여부입니다. 범례 라벨 클릭 시 해당 데이터 영역이 강조됩니다.	boolean	FALSE
InteractiveHighlightChartFill	범례 키에 해당하는 데이터 영역을 지정한 색으로 채웁니다.	string	rgba(255,255,255,0.7)

InteractiveHighlightChartStroke	범례 키에 해당하는 데이터 영역의 테두리를 지정한 색으로 채웁니다.	string	rgba(255,0,0,0.3)
InteractiveHighlightLabel	범례의 라벨을 강조하는 색을 지정합니다.	string	rgba(255,0,0,0.2)
LabelsColor	범례의 라벨의 색을 지정합니다.	object	null
LabelsFontStyle	범례의 라벨의 폰트 스타일을 지정합니다.	string	12px Tahoma
LabelsOffsetX	범례 라벨의 가로 오프셋을 지정합니다.	number	0
LabelsOffsetY	범례 라벨의 세로 오프셋을 지정합니다.	number	0
LineWidth	범례의 테두리 선의 두께를 지정합니다.	number	1
Position	범례의 표현 방식을 지정합니다. graph, gutter, margin을 사용할 수 있습니다.	string	graph
PositionMarginBoxed	범례의 Position이 'margin' 일 때, 범례를 박스 안에 표시할 지 여부입니다.	boolean	FALSE
PositionMarginHSpace	범례의 Position이 'margin' 일 때, 범례의 데이터 사이의 간격을 지정합니다.	number	0
PositionX	범례의 가로 좌표를 지정할 수 있습니다.	object	null
PositionY	범례의 세로 좌표를 지정할 수 있습니다.	object	null
Rounded	범례의 모서리를 둥글게 할 지 여부입니다. keyPosition이 margin이나 gutter에서만 동작합니다.	boolean	TRUE
Shadow	범례에 음영을 표시할 지 여부입니다.	boolean	FALSE
ShadowStyle	범례 음영의 스타일을 지정합니다. 'offsetX offsetY blur color' 로 구성되어 있습니다.	string	2px 2px 3px #666666

① 범례 예제	② 범례 예제
<pre>// Basic예제와 동일합니다 options: {   key: {     Data : ['종합','남자','여자']     // Position을 margin으로 설정합니다.     ,Position : 'margin'     ,PositionMarginHSpace: 15 // 범례 안 수평 여백     // Y축 위치 설정     ,PositionY:(canvas.offsetHeight - 20) - 15   } },</pre>	<pre>// Basic예제와 동일합니다 options: {   key: {     Data : ['종합','남자','여자']     // Position을 'graph'으로 설정합니다.     ,Position : 'graph'     // 범례의 테두리 색상을 설정합니다.     ,BorderColor:"#E2E2E2"   } },</pre>

## 9) 툴팁 설정하기



Sub property	설명	타입	기본값
Css	툴팁의 CSS를 지정합니다. 지정할 수 있는 CSS는 color, backgroundColor, fontSize, fontFamily 등이 있습니다.	object	null
Data	툴팁에 보여줄 데이터를 지정합니다. '%{key}', '%{value}',html 태그와 %prop, %property 등을 사용할 수 있습니다.	object	null
Effect	툴팁의 효과를 지정합니다. slide, fade를 사용할 수 있습니다.	string	slide
Event	툴팁을 발생시키는 이벤트를 지정합니다. onclick, onmousemove 를 사용할 수 있습니다.	string	onmousemove
FormattedDecimals	%{value_formatted}처럼 형식이 지정된 툴팁을 사용할 때 소수점 이하 자릿수를 지정합니다.	number	0
FormattedKeyColors	%{key}처럼 형식이 지정된 툴팁을 사용할 때 %{key}부분에 표시되는 색상을 지정할 수 있습니다.	object	null
FormattedKeyColorsShape	%{Key}처럼 형식이 지정된 툴팁을 사용할 때 %{key}부분에 표시되는 모양을 지정할 수 있습니다. circle, square를 사용	string	square

	가능합니다.		
FormattedKeyLabels	%{key}처럼 형식이 지정된 툴팁을 사용할 때 %{key}부분에 표시되는 라벨을 지정합니다.	object	[]
FormattedListItems	Data에 %{list}를 사용할 때, 리스트로 표시될 아이템을 지정합니다. 2차원 배열을 사용하거나 DxChart.convertDataset을 사용할 수 있습니다.	object	null
FormattedListType	Data에 %{list}를 사용할 때, 리스트에서 표시될 아이템의 정렬을 지정합니다. ul,ol이 사용 가능합니다.	string	ul
FormattedPoint	%{value_formatted}처럼 형식이 지정된 툴팁에서 소수점 표시 방식을 지정합니다.	string	.
FormattedTableData	툴팁의 형식을 테이블의 형태로 지정합니다. 3차원 배열을 사용해야 합니다. 데이터 예시는 첫번째 데이터셋의 첫번째 데이터 툴팁입니다. 같은 방법으로 n번째 테이블 툴팁을 지정할 수 있으며, 첫 번째 데이터셋의 데이터 개수만큼 배열이 작성된 후 추가 배열부터는 첫번째 데이터셋의 두번째 데이터 툴팁으로 넘어갑니다.	object	null
FormattedTableHeaders	툴팁 테이블을 사용할 때, 테이블의 헤더를 지정할 수 있습니다.	object	null
FormattedThousand	%{value_formatted}처럼 형식이 지정된 툴팁에서 천 단위 구분 기호를 지정합니다.	string	,
FormattedUnitsPost	%{value_formatted}처럼 형식이 지정된 툴팁에서 뒤에 표시할 문자를 지정합니다.	string	
FormattedUnitsPre	%{value_formatted}처럼 형식이 지정된 툴팁에서 앞에 표시할 문자를 지정합니다.	string	
Highlight	툴팁이 표시될 때 강조 효과를 사용할 지 여부입니다.	boolean	TRUE
HotspotXonly	데이터 포인트의 Y좌표가 0인 X축에 가까이 가면 해당 데이터 포인트에 툴팁이	boolean	FALSE

	틀 지 여부입니다.		
--	------------	--	--

Tooltip의 데이터를 표시하기 위해 사용하는 macro종류

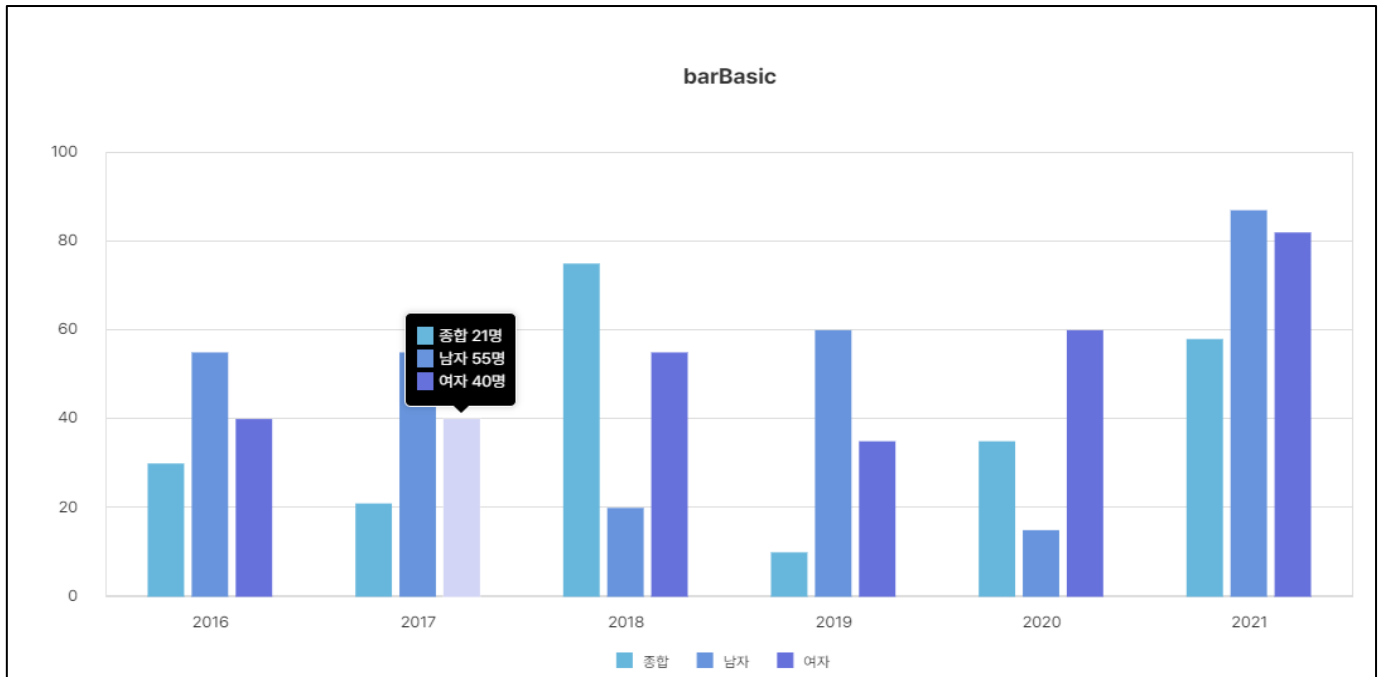
macro	설명
%{index}	여러 개의 차트를 사용할 경우, 차트의 index (0부터 시작)
%{dataset}	하나의 차트에서 선택/지정할 데이터의 index (0부터 시작)
%{dataset2}	bioplar차트에서 사용 왼쪽/오른쪽별로 번호를 부여 <ul style="list-style-type: none"> <li>dataset은 왼쪽부터 0~n으로 번호 부여한 후 오른쪽은 n+1~2n으로 번호를 부여되지만, dataset2는 왼쪽부터 0~n으로 번호 부여한 후 오른쪽은 다시 0~n으로 번호 부여</li> </ul>
%{seq} %{sequential_index} }	차트가 여러 개인 경우, 전체를 하나의 index로 처리 할 경우의 데이터index (0부터 시작)
%{property:title}	차트의 타이틀 텍스트 (option의 title에 해당 )
%{property:axisLabels[%{index}]}	선택한 X축 라벨을 표시합니다.
%{global:contries} %{global:contries[%{index}]} %{global:contries[%{dataset}]}	global로 정의한 데이터를 표시. (nexacro에서는 사용할 수 없습니다.%{property:xxx}를 사용하여 같은 효과를 낼 수 있습니다.)
%{value} %{value2} %{value_formatted}	선택된 데이터 값 vaue : 원래의 값 그대로를 표시합니다.(123456) value_formatted : 숫자인 경우 format(123,456 형식)을 적용하여 표시합니다.
%{percent} %{percent_formatted}	100% 차트인 경우, 해당 값에 대한 %비율을 표시
%{side} %{side_title}	bipolar차트인 경우 , 오른쪽/왼쪽의 데이터여부를 표시 side : 0(왼쪽) / 1(오른쪽) side_title : "left"(왼쪽) , "right"(오른쪽)
%{start}	Gantt차트에서 사용하는 데이터값



%{end} %{duration} %{complete} %{label}	start : Data의 start값 end : Data의 end 값 duration : Data의 duration 값 complete : Data의 complete 값 label : data의 label값  사용예) binddataset : this.Dataset00, data: 'bind:start=start,bind:end=end,bind:complete=complete,bind:label=label,bind:linewidth=linewidth,bind:color=color,bind:bgcolor=background', option : { tooltips : {Data : '#{label} START: #{start} END: #{end} COMPLETE: #{complete}'} } }
#{table}	HTML의 table태그의 값 FormattedTableData에 설정된 데이터를 표시합니다. 해당 형식은 nexacro인 경우는 2차원 array형태, HTML인 경우 table tag의 아이디
#{group} #{group2}	#{dataset}#{dataset2}와 동일
#{list}	HTML의 list태그의 값 FormattedListItem에 설정된 데이터를 표시합니다. 해당 형식은 nexacro인 경우는 2차원 array형태, HTML인 경우 list tag의 아이디  사용예) tooltips:{Data:'<h2>Results</h2> #{list}', Effect: 'none', FormattedListType: 'ol', FormattedListItems: [ ['Barry','Larry','Harry'], ['Bill','Lil','Jill'], ['Jack','Zak','Quack'] ], Event:"onclick" },
#{function:xxx()}	지정한 함수를 실행하고 그 결과를 표시합니다.

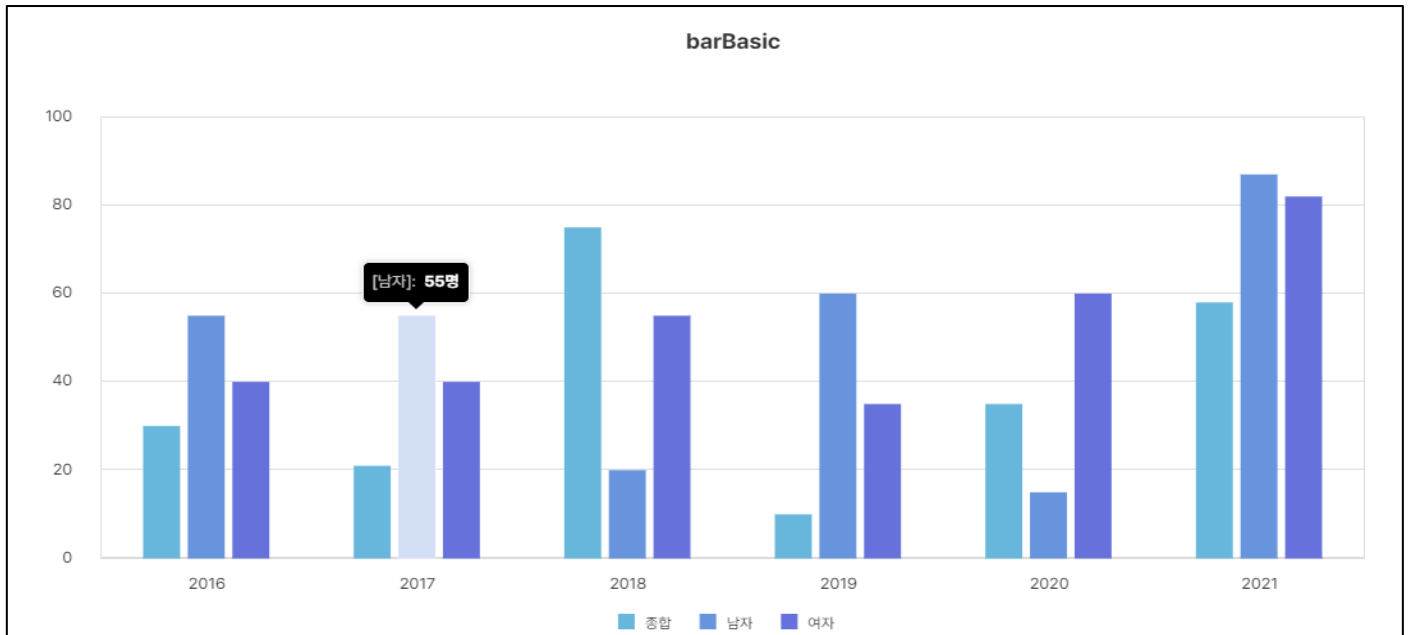
	<ul style="list-style-type: none"> <li>● 함수는 반드시 리턴값이 있어야 하며, String,number값이어야 합니다.</li> <li>● 사용할 수 있는 함수 <ul style="list-style-type: none"> <li>- javascript 고유 함수</li> <li>- DxChart오브젝트의 메소드 함수</li> <li>- Nexacro의 사용자 함수</li> </ul> </li> </ul>
%{custom}	<p>FormattedCustom에 지정된 함수를 실행합니다. 파라미터는 specific이며, 각 차트마다 다르게 세팅이 됩니다.</p> <p>사용예)</p> <pre> tooltips:{     Data:'%{custom}',/'%{value}',     Css: {fontSize: '14pt'},     FormattedCustom : function(specific) {     var item = specific.object.getItem(specific.index);     return {         label : item.name + " Airport&lt;br&gt;"             + "Code:" + item.abbrev     };     } } </pre>

## ① 툴팁 유형 1



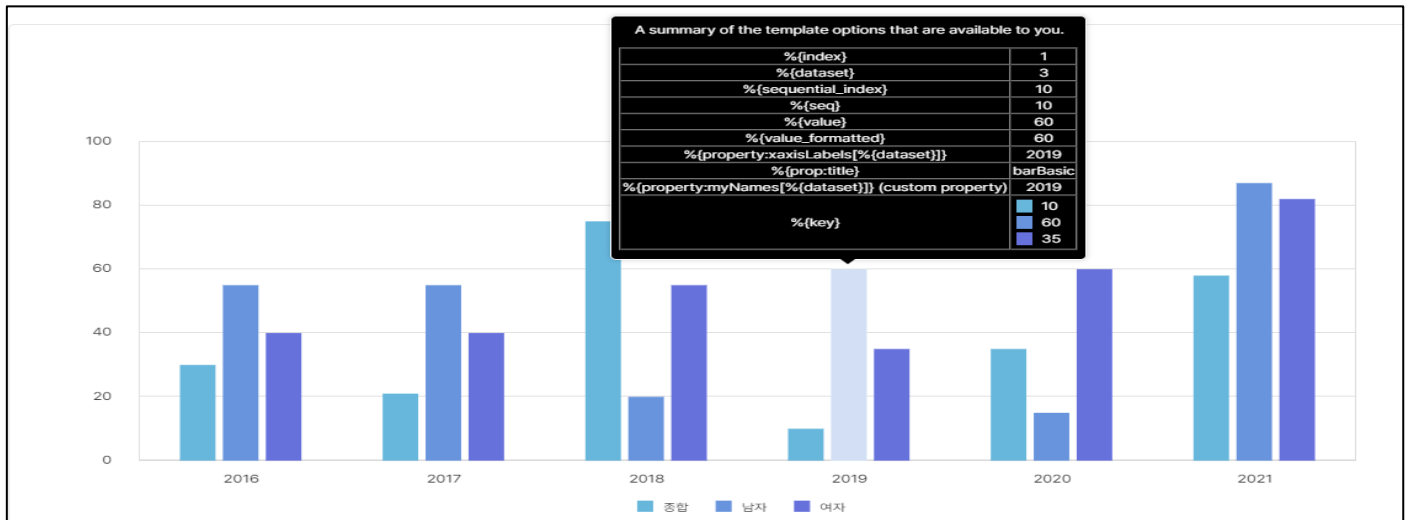
```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        //기본적인 설정은 Basic예제와 동일합니다
        tooltips: {
            Data : '%{key}', // Data 값을 "%{key}" 설정 시 범례 형태로 표시됩니다.
            FormattedUnitsPost: '명', // 뒤에 붙일 문자열을 설정합니다.
            Effect : 'fade', // Tooltips의 효과를 설정합니다.
            Pointer : true, // Tooltips의 포인터를 설정합니다.
            FormattedKeyLabels: ['총합','남자','여자'], // Tooltips의 key 레이블을 추가합니다.설정합니다.
        },
    }
}).draw();
```

## ② 툴팁 유형 2



```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        //기본적인 설정은 Basic예제와 동일합니다
        tooltips: {
            // Html 태그를 삽입하고 매크로를 통해 표시합니다
            // Gender는 해당 key 값, value는 해당 데이터 값입니다.
            Data:'<span style="color:#ccc">[%{property:gender[%{index}]}]:</span> <b> %{value}명 </b>',
            gender : ['종합','남자','여자'], // 사용자가 설정한 property로 tooltip 데이터를 호출합니다.
        },
    }
})
}).draw();
```

### ① 툴팁 유형3



```
var strHtml = "";
strHtml += 'A summary of the template options that are available to you.<p />';
strHtml += '<table border="1" align="center" cellspacing="0">';
strHtml += '<tr><td>%%{index}</td><td>%{index}</td></tr>';
strHtml += '<tr><td>%%{dataset}</td><td>%{dataset}</td></tr>';
strHtml += '<tr><td>%%{sequential_index}</td><td>%{sequential_index}</td></tr>';
strHtml += '<tr><td>%%{seq}</td><td>%{seq}</td></tr>';
strHtml += '<tr><td>%%{value}</td><td>%{value}</td></tr>';
strHtml += '<tr><td>%%{value_formatted}</td><td>%{value_formatted}</td></tr>';
strHtml += '<tr><td>%%{property:axisLabels[%{dataset}]}</td></tr>';
strHtml += '<td>%{property:axisLabels[%{dataset}]}</td></tr>';
strHtml += '<tr><td>%%{prop:title}</td><td>%{prop:titleText}</td></tr>';
strHtml += '<tr><td>%{property:myNames[%{dataset}]} (custom property)</td></tr>';
strHtml += '<td>%{property:myNames[%{dataset}]}</td></tr>';
strHtml += '<tr><td>%%{key}</td><td>%{key}</td></tr></table>';

var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    options: {
        //기본적인 설정은 Basic예제와 동일합니다
        tooltips: { Data : strHtml, // tooltips의 선언한 html Data를 설정합니다.
        // 사용자가 선언한 property로도 사용가능합니다.
        myNames:DxChart.convertDataset(this.Dataset00,["bind:indicators"]),
        }
    }
}).draw();
```

### 3. Effect 설정하기

차트를 그릴 때 애니메이션의 종류를 지정합니다.

1) Draw

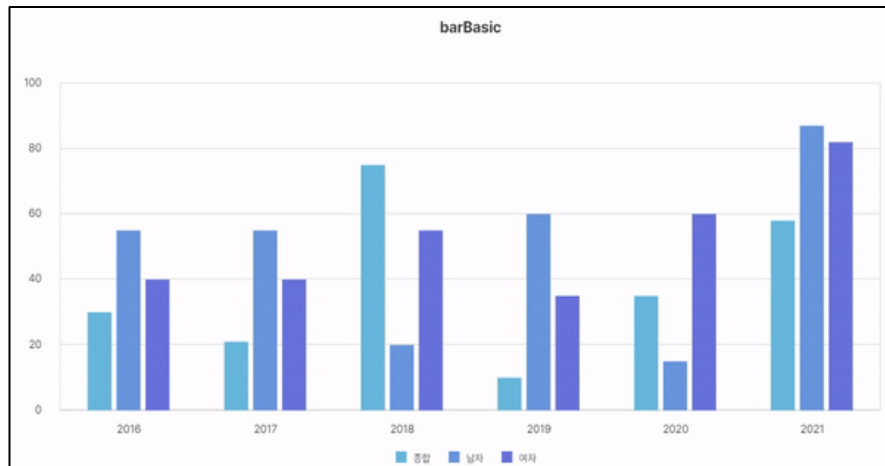
애니메이션을 하지 않고, 차트를 그립니다.

2) drawAni

애니메이션을 지정합니다. 첫번째 인수는 애니메이션의 종류를 나타내고 두번째 인수 이후는 해당 애니메이션의 인수를 지정합니다.

사용 예) wave 애니메이션 : `chart.drawAni("wave");`

3) wave

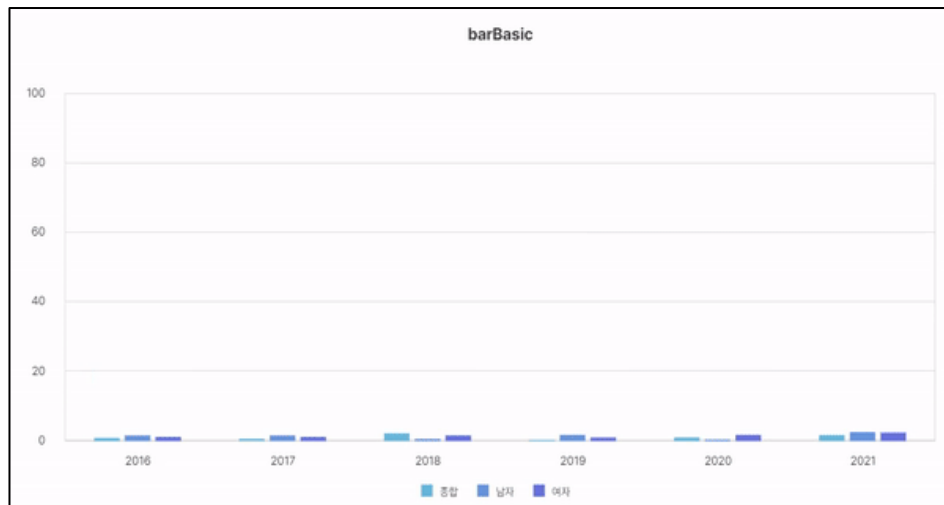


왼쪽에서 오른쪽으로 아래쪽에서 위쪽으로 애니메이션이 동시에 일어납니다.

기본적인 애니메이션 초당 Frame은 90입니다.

```
var bar= New DxChartBar(config);  
bar.wave ();
```

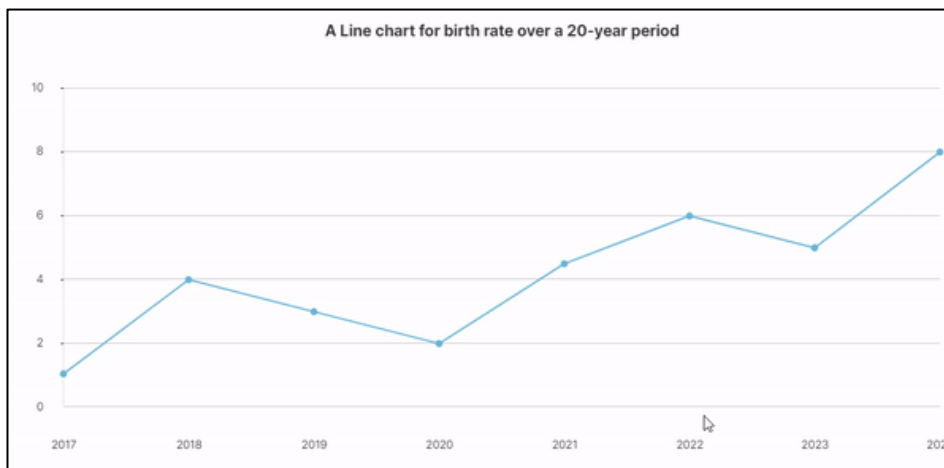
#### 4) grow



아래쪽 또는 위쪽으로 애니메이션이 일어납니다. 기본적인 애니메이션 초당 Frame은 60입니다.

```
var bar= New DxChartBar(config);
bar.grow ();
```

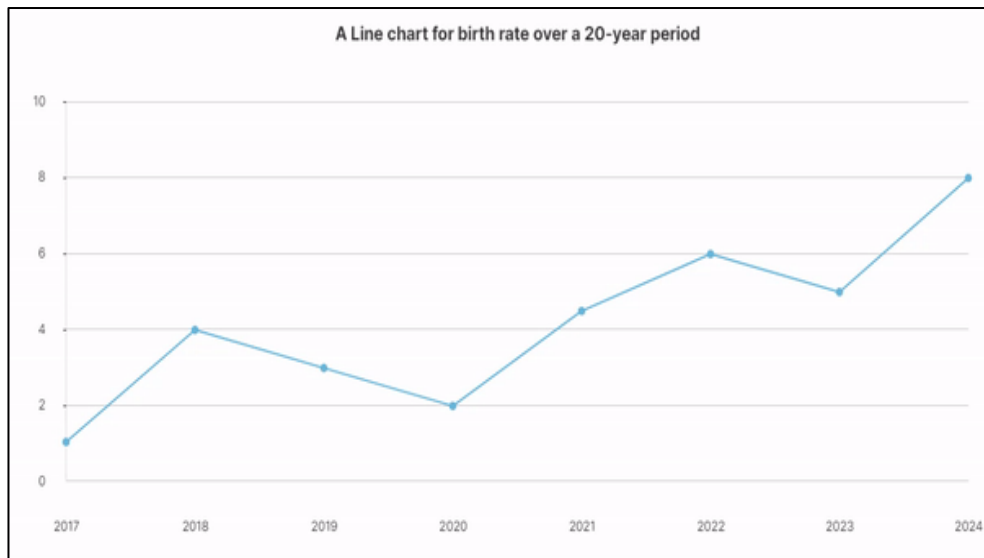
#### 5) trace



왼쪽에서 오른쪽으로 애니메이션이 일어납니다. (적용 대상 차트 : line , scatter , rader)

```
var line= New DxChartLine(config);
line.trace();
```

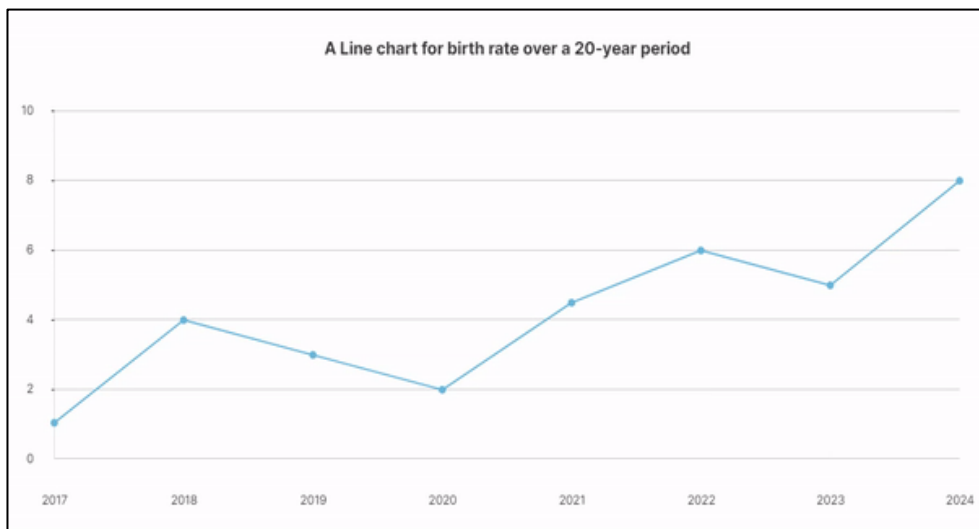
6) unfold (toBottom)



아래쪽으로 애니메이션이 일어납니다. (적용 대상 차트 : line)

```
var line= New DxChartLine(config);  
line. unfold();
```

7) unfoldFromCenterTrace (fromCenter)

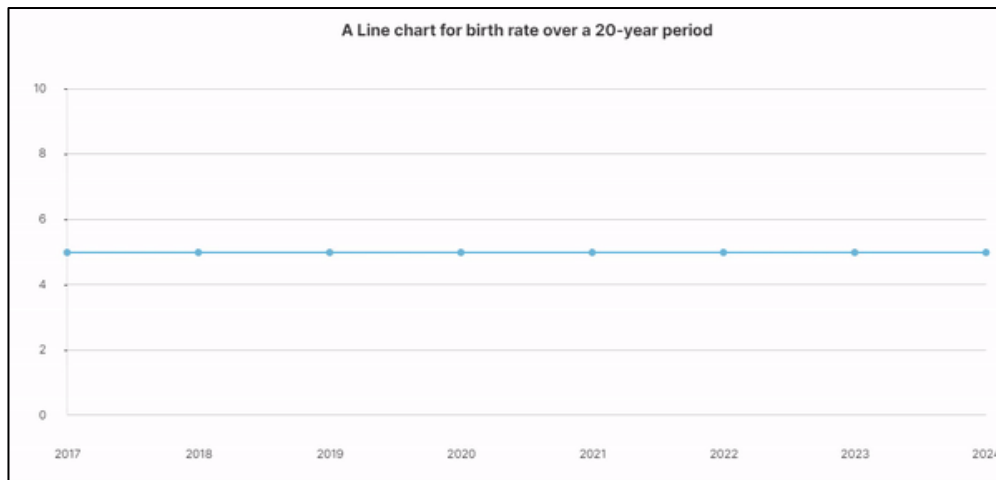


차트의 중앙에서부터 원래의 값으로 이동하는 애니메이션이 일어납니다. (적용 대상 차트 : line)

```
var line= New DxChartLine(config);  
line. unfoldFromCenterTrace();
```



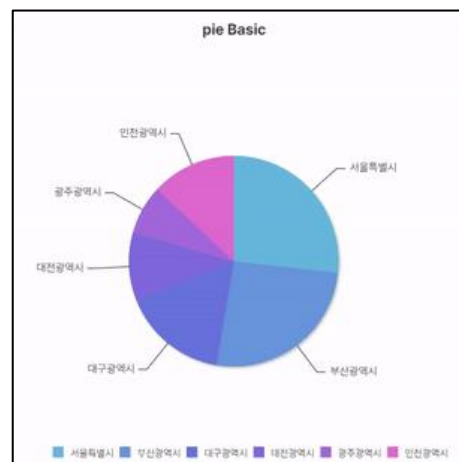
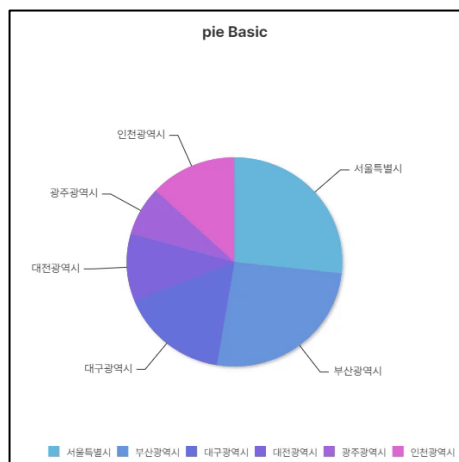
## 8) foldtocenter (toCenter)



차트의 원래의 값에서부터 중앙으로 이동하는 애니메이션이 일어납니다. (적용 대상 차트 : line)

```
var line= New DxChartLine(config);
line.foldtocenter();
```

## 9) roundRobin 및 roundRobinSequential



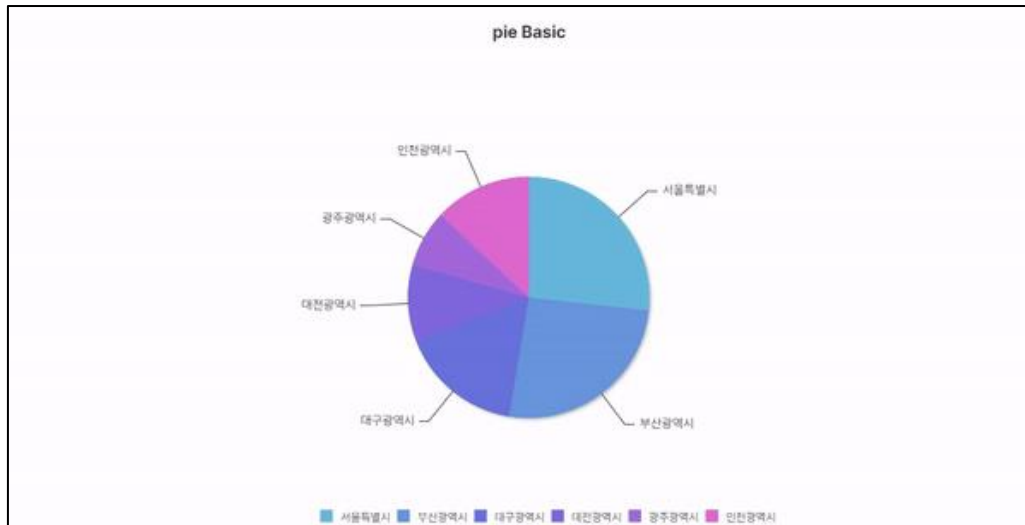
시계방향으로 돌아가면서 차트를 그리는 애니메이션이 일어납니다.

roundRobin은 각각의 값이 조금씩 늘어나면서 파이의 각도가 점점 늘어나지만, roundRobinSequential은 하나의 값이 점점 늘어나면서 하나의 값이 전부 그려지면 다음 값이 늘어나면서 파이의 각도를 증가합니다.

적용 대상 차트 : pie , pieSegment, rose

```
var pie= New DxChartPie(config);
pie.roundRobin();
```

## 10) implode



파이의 각 부분이 바깥으로부터 중심으로 모여 드는 애니메이션이 일어납니다.

적용 대상 차트 : pie , pieSegment, rose

```
var pie= New DxChartPie(config);  
pie.implode();
```

## 11) explode



파이의 각 부분이 중심으로부터 바깥으로 퍼져 나가는 애니메이션이 일어납니다.

적용 대상 차트 : pie , pieSegment, rose , scatter

```
var pie= New DxChartPie(config);  
pie.explode();
```

### 차트 별 effect 종류

차트종류	effect종류	drawAni()의 첫번째 파라미터	Default
activity	grow	grow	grow
bar	wave	wave	grow
	grow	grow	
bipolar	wave	wave	grow
	grow	grow	
fuel	grow	grow	grow
funnel	-	-	draw
gauge	grow	grow	grow
gantt	grow	grow	grow
hbar	grow	grow	grow
	wave	wave	
horseshoe	grow	grow	grow
hprogress	grow	grow	grow
line	unfold	unfold (toBottom)	wave(toTop)
	Unfoldfromcentertrace	Unfoldfromcentertrace (fromCenter)	
	foldtocenter	foldtocenter(toCenter)	
	trace	trace(toRight)	
	wave	wave(toTop)	
meter	grow	grow	grow
odo	grow	grow	grow
pie	grow	grow	grow

	explode	explode	
	roundrobin	roundRobin	
	roundRobinSequential	roundRobinSequential	
	implode	implode	
pieSegment	grow	grow	grow
	explode	explode	
	roundrobin	roundRobin	
	roundRobinSequential	roundRobinSequential	
	implode	implode	
radar	grow	grow	grow
	trace	trace	
rose	grow	grow	grow
	roundRobin	roundRobin	
	implode	implode	
	explode	explode	
rscatter	-		draw
scatter	trace	trace	trace
	explode	explode	
semicircularprogress	grow	grow	grow
segment	grow	grow	grow
thermometer	grow	grow	grow
vprogress	grow	grow	grow
waterfall	grow	grow	grow
org	grow	grow	grow
tree	grow	grow	grow

heatmap	grow	grow	grow
sunburst	grow	grow	grow
pyramid	-		draw
map	-		draw

## 4. Multi Chart 생성하기

### 1) Bar 및 Line 차트



```
var bar = new DxChartBar({  
    id: cvs,  
    elem : canvas,  
    binddataset : this.Dataset00_00,  
    data:["bind:seoul","bind:busan","bind:daegu"],  
    options: {  
        margin:{Left:105,Right:55,Top:135,Inner:30,Bottom:82},  
        xaxis:{  
            Use:false  
            ,Labels: ['bind:year']  
            ,ScaleZerostart: true  
            ,Color:"#E2E2E2"  
            ,LabelsOffsetY:5  
            ,Tickmarks:false
```

```

        },
        colorsStroke: 'white',
        linewidth: 2,
        shadow:{Offsetx: 1,Offsety: 0,Blur: 1,,},
        background:{GridVlines: false,GridBorder: false},
        yaxis:{
            Use:false
            ,Color:"#E2E2E2"
            ,LabelsOffsetX:-15
            ,Tickmarks:false
            ,ScaleMax:50
        },
        title: {Text:'barCombinedLine',Y :73},
        combinedEffect: 'wave',
        //combinedEffectCallback: function () {$a("Finished the bar effect!")},
        combinedEffectOptions: '{frames: 90}',
        tooltips: {
            Data : '%{value_formatted}',
            Effect : 'fade',
            Pointer : false,
            FormattedUnitsPost : '%',
        }
    }
});

var line = new DxChartLine({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00_00,

```

```

data: ['bind:total'],

options: {

    colors:["#6794DC"],

    spline: true,

    tickmarksStyle: 'filledcircle',

    yAxisScaleMax: 40,

    shadow: false,

    xAxis: false,

    yAxis: false,

    combinedEffect: 'trace',

    combinedEffectOptions: '{frames: 90}',

    tooltips: {

        Data : '%{value}',

        Effect : 'fade',

        Pointer : true,

    }

}

});

var combo = new DxChart.CombinedChart(bar, line);

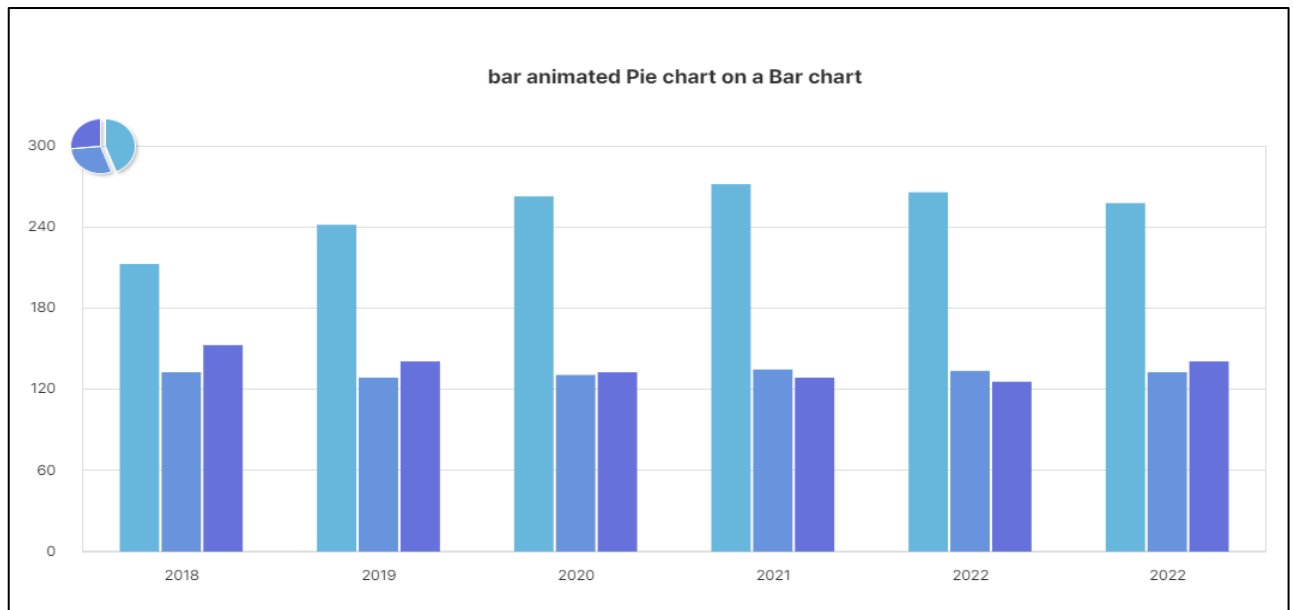
combo.draw();

```

1. 각각 bar와 Line의 object를 생성합니다.
2. DxChart.CombinedChart를 통해 margin 값 계산 후 함께 그려집니다.



## 2) Bar 및 Pie 차트



```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    binddataset : this.Dataset00,
    data:["bind:child","bind:middle","bind:high"],
    options: {
        margin:{Left:105,Right:55,Top:135,Inner:30,Bottom:82},
        title:{Text: 'bar animated Pie chart on a Bar chart',Y :73},
        xaxis:{
            Labels: ['bind:year']
            ,ScaleMax:200
            ,Color:"#E2E2E2"
            ,LabelsOffsetY:5
            ,Tickmarks:false
        },
        yaxis:{
            Use:false
        }
    }
});
```

```

        ,Color:"#E2E2E2"

        ,LabelsOffsetX:-15

        ,Tickmarks:false

    },

    colorsStroke: 'rgba(0,0,0,0)',

    background:{GridVlines: false,},

}

}).draw();

var pie = new DxChartPie({

    id: cvs,

    elem : canvas,

    binddataset : this.Dataset00,

    data:["bind:pieValue"],

    options: {

        center:{x: 120, y: bar.get('marginTop')},

        radius: 25,

        exploded: [5],

        colorsStroke: 'rgba(0,0,0,0)',

        colorsStroke: '#fff',

        shadow: false,

        linewidth: 1

    }

}).roundRobin({frames: 60});

```

1. 각각 bar와 pie의 object를 생성합니다.
2. DxChart.CombinedChart를 사용하지 않고 margin 값으로 설정하여 함께 생성하였습니다.

# 5. 기타기능

## 1) convertDataset

DataSet을 배열로 반환하는 메서드입니다. bind로 개발 시 어려울 때 유용하게 사용가능합니다.

함수명	Argument	Type	설명
ConvertDataset	dataset	Object	Dataset object
	arrCol	Array	변환할 데이터셋 컬럼 명
	nDiv	Number	(옵션) 반복 행 설정
	Option	String	(옵션) "linear" 설정 시 컬럼 별 데이터 생성

예시) Dataset00

Columns					
No	id	type	size	prop	sumtext
1	Column0	STRING	256		
2	Column1	STRING	256		
3	Column2	STRING	256		
Rows					
No	Column0	Column1	Column2		
1	가	나	다		
2	라	마	바		
3	사	아	자		

예시 1) linear 미설정 시

```
var arrData = DxChart.convertDataset(this.Dataset00,["bind:Column0","bind:Column1","bind:Column2"]);
```

Columns					
No	id	type	size	prop	sumtext
1	Column0	STRING	256		
2	Column1	STRING	256		
3	Column2	STRING	256		
Rows					
No	Column0	Column1	Column2		
1	가	나	다		
2	라	마	바		
3	사	아	자		

결과값

arrData =[[가,나,다],[라,마,바],[사,아,자]]

예시 2) linear 설정 시

```
var arrData = DxChart.convertDataset(this. Dataset00,["bind:Column0","bind:Column1"," bind:Column2"],null,"linear");
```

No	id	type	size	prop	sumtext
1	Column0	STRING	256		
2	Column1	STRING	256		
3	Column2	STRING	256		

No	Column0	Column1	Column2
1	가	나	다
2	라	마	바
3	사	아	자

결과값

arrData =[[가,라,사],[나,마,아],[다,바,자]]

2) datachangeevent

기존 Chart에 Binddding 되어있는 Dataset의 값이 변경 되었을 때 차트에 바로 적용해주는 속성입니다.

property	Sub Property	설명	타입	기본값
datachangeevent	Use	Datachangeevent를 사용할 지 여부입니다.	Boolean	null
	Effect	Datachangeevent 발생 시 Chart의 Effect를 설정합니다.	string	draw

```
var bar = new DxChartBar({
    id: cvs,
    elem : canvas,
    bindddataset : this.Dataset00,
    data:["bind:total","bind:man","bind:woman"],
    datachangeevent:{
        Use:true // datachangeevent 사용 설정합니다.
        ,Effect:"draw" // datachangeevent 발생 시 Effect를 설정합니다.
    },
    options: { // 기존 예제와 동일합니다 }
}).wave();
```