



# Chart

```
/***
USAGE:

Chart(type, data, height, width, xaxis, yaxis, title, min, max, inter

PARAMETERS:

type : str (one of 'circulargauge', 'column', 'multiseriescolumn', 'l
data : list

(optional) height : num (default: 450)
    Height of chart in pixel or percent.
    If value is greater or equal to 1, then value represents pixel, o

(optional) width : num (default: 450)
    Width of chart in pixel or percent.
    If value is greater or equal to 1, then value represents pixel, o

(optional) xaxis : str (default: 'Y-Axis')
    Label for X-Axis.

(optional) yaxis : str (default: 'X-Axis')
    Label for Y-Axis.

(optional) title : str (default: 'Title')
    Label for chart.

(optional) min : num (default: 0)
    Lower bound for Linear Gauge chart type.

(optional) max : num (default: 100)
    Upper bound for Linear Gauge chart type.

(optional) interval : num (default: 10)
    Major interval for axis markers.

(optional) id : str (default: nil)
    ID for chart component. Used for listening for events and interac

VERSIONS:
```

|     |           |         |                                      |
|-----|-----------|---------|--------------------------------------|
| 1.0 | 2-Feb-10  | robertm | initial version                      |
| 1.1 | 4-Mar-10  | steveb  | code clean-up                        |
| 1.2 | 29-Mar-10 | steveb  | better handling of error conditions; |
| 1.3 | 16-Jul-10 | steveb  | fixed improper data handling for 'pi |

\*\*\*/

```
// GET VARIABLES FROM TEMPLATE CALL
var type = string.toLowerCase($type ?? $0 ?? 'circulargauge');
var data = $data ?? $1 ?? 67;
var height = $height ?? $2 ?? 450;
var width = $width ?? $3 ?? 450;
var xaxis = $xaxis ?? $4;
var yaxis = $yaxis ?? $5;
var title = $title ?? $6;
var min = $min ?? $7;
var max = $max ?? $8;
var interval = $interval ?? $9 ?? 10;
var id = $id ?? $10;
var error;

// TODO (steveb): validate the 'data' field
// TODO (steveb): enable/disable animation

// format settings
let settings_xml = <settings>
    <animation enabled="True"/>
</settings>

// format axis
var axes_xml = <axes>
    <x_axis>
        <title enabled=(xaxis is not nil)>
            <text> xaxis </text>
        </title>
        <labels>
            <format> "{%Value}{numDecimals:0}" </format>
        </labels>
    </x_axis>
    <y_axis position=((type == 'bar' || type == 'multiseriesbar') ? "oppo
        <title enabled=(yaxis is not nil)>
            <text> yaxis </text>
        </title>
        <labels>
```

```

        <format> "{%Value}{numDecimals:0}" </format>
    </labels>
    <scale major_interval=(interval) minor_interval=(interval / 4) m...
</y_axis>
</axes>

// set defaults for min-max
let min = min ?? 0;
let max = max ?? 100;

// format data
var data_xml;
if((type != 'pie') && data is map) {
    let data_xml = <data>
        foreach (var series:points in data) {
            <series name=(series)>
                foreach (var p in points) {
                    foreach(var label:value in p) {
                        <point y=(value) name=(label)>
                            <tooltip enabled="true">
                                <format> "%SeriesName} ({%Name}) - {%
...
                            </tooltip>
                        </point>
                    }
                }
            </series>
        }
    </data>;
} else if(data is list) {
    let data_xml = <data>
        <series name="Series 1">
            foreach(var d in data) {
                foreach(var label:value in d) {
                    <point y=(value) name=(label) />
                }
            }
        </series>
    </data>;
}

// CHART BUILDS
var chart;
switch (type) {

```

```

// SINGLE-SERIES COLUMN CHART, INCOMING DATA MUST BE FORMATTED AS [{label: "A", value: 1}, {label: "B", value: 2}, {label: "C", value: 3}, {label: "D", value: 4}, {label: "E", value: 5}, {label: "F", value: 6}, {label: "G", value: 7}, {label: "H", value: 8}, {label: "I", value: 9}, {label: "J", value: 10}, {label: "K", value: 11}, {label: "L", value: 12}, {label: "M", value: 13}, {label: "N", value: 14}, {label: "O", value: 15}, {label: "P", value: 16}, {label: "Q", value: 17}, {label: "R", value: 18}, {label: "S", value: 19}, {label: "T", value: 20}, {label: "U", value: 21}, {label: "V", value: 22}, {label: "W", value: 23}, {label: "X", value: 24}, {label: "Y", value: 25}, {label: "Z", value: 26}], incomingType: "column"
// MULTI-SERIES COLUMN CHART, INCOMING DATA MUST BE FORMATTED AS [{series: "A", label: "A", value: 1}, {series: "A", label: "B", value: 2}, {series: "A", label: "C", value: 3}, {series: "A", label: "D", value: 4}, {series: "A", label: "E", value: 5}, {series: "A", label: "F", value: 6}, {series: "A", label: "G", value: 7}, {series: "A", label: "H", value: 8}, {series: "A", label: "I", value: 9}, {series: "A", label: "J", value: 10}, {series: "A", label: "K", value: 11}, {series: "A", label: "L", value: 12}, {series: "A", label: "M", value: 13}, {series: "A", label: "N", value: 14}, {series: "A", label: "O", value: 15}, {series: "A", label: "P", value: 16}, {series: "A", label: "Q", value: 17}, {series: "A", label: "R", value: 18}, {series: "A", label: "S", value: 19}, {series: "A", label: "T", value: 20}, {series: "A", label: "U", value: 21}, {series: "A", label: "V", value: 22}, {series: "A", label: "W", value: 23}, {series: "A", label: "X", value: 24}, {series: "A", label: "Y", value: 25}, {series: "A", label: "Z", value: 26}, {series: "B", label: "A", value: 1}, {series: "B", label: "B", value: 2}, {series: "B", label: "C", value: 3}, {series: "B", label: "D", value: 4}, {series: "B", label: "E", value: 5}, {series: "B", label: "F", value: 6}, {series: "B", label: "G", value: 7}, {series: "B", label: "H", value: 8}, {series: "B", label: "I", value: 9}, {series: "B", label: "J", value: 10}, {series: "B", label: "K", value: 11}, {series: "B", label: "L", value: 12}, {series: "B", label: "M", value: 13}, {series: "B", label: "N", value: 14}, {series: "B", label: "O", value: 15}, {series: "B", label: "P", value: 16}, {series: "B", label: "Q", value: 17}, {series: "B", label: "R", value: 18}, {series: "B", label: "S", value: 19}, {series: "B", label: "T", value: 20}, {series: "B", label: "U", value: 21}, {series: "B", label: "V", value: 22}, {series: "B", label: "W", value: 23}, {series: "B", label: "X", value: 24}, {series: "B", label: "Y", value: 25}, {series: "B", label: "Z", value: 26}], incomingType: "multiseriescolumn"
case 'column':
case 'multiseriescolumn':
case 'bar':
case 'multiseriesbar':


    // determine layout value
    var layout;
    switch(type) {
    case 'column':
    case 'multiseriescolumn':
        let layout = "CategorizedVertical";
    case 'bar':
    case 'multiseriesbar':
        let layout = "CategorizedHorizontal";
    }

    // generate chart xml
    let chart = <anychart>
        settings_xml;
        <charts>
            <chart plot_type=(layout)>
                <data_plot_settings default_series_type="Bar" enable_3d="false">
                    <bar_series group_padding="0.2" >
                        <tooltip_settings enabled="true"/>
                    </bar_series>
                </data_plot_settings>
                <chart_settings>
                    <title enabled=(title is not nil)>
                        <text> title </text>
                    </title>
                </chart_settings>
            </chart>
        </charts>
    
```

// check if we plotting a series of data points

```

if(data is map) {
    <legend enabled="true" position="Bottom" align="Left" style="font-size: 10px;">
        <format> "{%Icon} {%Name}" </format>
        <title enabled="false"/>
        <columns_separators enabled="true"/>
        <background>
            <inside_margin left="10" right="10"/>
        </background>
        <items>
            <item source="Series"/>
        </items>
    </legend>
}

```

```

                </legend>
            }
            axes_xml;
        </chart_settings>
        data_xml;
    </chart>
</charts>
</anychart>

// MULTI-SERIES LINE CHART, INCOMING DATA MUST BE FORMATTED AS {series1:
case "line":
    let chart = <anychart>
        settings_xml;
    <charts>
        <chart plot_type="CategorizedVertical">
            <chart_settings>
                <title enabled=(title is not nil)>
                    <text> title </text>
                </title>
                <legend enabled="true">
                    <title enabled="false"/>
                </legend>
                axes_xml;
            </chart_settings>
            <data_plot_settings default_series_type="Spline">
                <line_series>
                    <marker_settings>
                        <marker size="8"/>
                        <states>
                            <hover>
                                <marker size="12"/>
                            </hover>
                        </states>
                    </marker_settings>
                    <tooltip_settings enabled="True"/>
                </line_series>
            </data_plot_settings>
            data_xml;
        </chart>
    </charts>
</anychart>

//3D PIE CHART, DATA VARIABLE MUST BE FORMATTED AS {name1:value1, name2: value2}
case 'pie':
    let chart = '<anychart>

```

```
settings_xml;
<charts>
    <chart plot_type="Pie">
        <data_plot_settings enable_3d_mode="true">
            <pie_series>
                <tooltip_settings enabled="true">
                    <format>
                        {Name} : {Value}{numDecimals:0} ({YPercentOfSeries}{numDecimals:0}%)
                    </format>
                </tooltip_settings>
                <label_settings enabled="true">
                    <background enabled="false"/>
                    <position anchor="Center" valign="Center" horizontalAlign="Center" rotation="0" style="font-size: 14px; color: white; opacity: 0.8; font-weight: bold; text-align: center; width: 100%; height: 100%;">
                        <font color="White">
                            <effects>
                                <drop_shadow enabled="true" distance="5px" color="black" opacity="0.5" style="filter: drop-shadow(5px 5px 5px black);"/>
                            </effects>
                        </font>
                        <format>{YPercentOfSeries}{numDecimals:0}%</format>
                    </label_settings>
                </pie_series>
            </data_plot_settings>
        <data>
            <series name="Series 1" type="Pie">
                .. (
                    foreach (var name:y in data) {
                        '<point name="' .. name .. '" y="' .. y .. '" value="' .. value .. '">'
                    }
                ) ..
                '</series>
            </data>
        <chart_settings>
            <title enabled="true" padding="15">
                <text>' .. title .. '</text>
            </title>
            <legend enabled="true" position="Bottom" align="Spread">
                <format>{Icon} {Name} - {YValue}{numDecimals:0}%</format>
                <title enabled="false"/>
                <columns_separator enabled="false"/>
                <background>
                    <inside_margin left="10" right="10"/>
                </background>
                <items>
                    <item source="Points"/>
                </items>
            </legend>
        </chart_settings>
    </chart>
</charts>
```

```

                </legend>
            </chart_settings>
        </chart>
    </charts>
</anychart>';

// PYRAMID/FUNNEL CHART, DATA VARIABLE MUST BE FORMATTED AS {label1:value}
case 'pyramid':
case 'funnel':
    var ispyramid = (type == 'pyramid');
    let chart = <anychart>
        settings_xml;
    <charts>
        <chart plot_type="Funnel">
            <chart_settings>
                <title enabled=(title is not nil)>
                    <text> title </text>
                </title>
                <data_plot_background enabled="false" />
                <legend enabled="false" />
            </chart_settings>
            <data_plot_settings enable_3d_mode="true">
                <funnel_series inverted=(ispyramid) neck_height=(ispyramid ? 10 : 0)>
                    <animation enabled="true" type="Appear" show_mode="OnLoad" />
                    <connector enabled="true" color="Black" opacity="1" />
                    <tooltip_settings enabled="true">
                        if(ispyramid) {
                            <position anchor="CenterRight" padding="10" />
                        }
                        <format> "{%Name} - {%-YValue}{numDecimals:0}</format>
                    </tooltip_settings>
                    <label_settings enabled="true">
                        <animation enabled="true" type="Appear" show_mode="OnLoad" />
                        if(ispyramid) {
                            <position anchor="Center" valign="Center" />
                        } else {
                            <position anchor="center" padding="50"/>
                        }
                        <format> "{%Name} - {%-YValue}{numDecimals:0}</format>
                    <background enabled="true">
                        <corners type="Rounded" all="3"/>
                    </background>
                    <states>
                        <hover>
                            <background>

```

```
        <border type="Solid" color="DarkColor(Black)" width="1px" style="border-radius: 10px;"/>
    </background>
</hover>
<pushed>
    <background>
        <border type="Solid" color="#494949" width="1px" style="border-radius: 10px;"/>
    </background>
</pushed>
<selected_hover>
    <background>
        <border type="Solid" color="DarkColor(Yellow)" width="1px" style="border-radius: 10px;"/>
    </background>
</selected_hover>
<selected_normal>
    <background>
        <border type="Solid" color="DarkColor(Yellow)" width="1px" style="border-radius: 10px;"/>
    </background>
</selected_normal>
</states>
</label_settings>
<funnel_style>
    <border color="Black" opacity="0.05"/>
    <states>
        <hover>
            <fill color="%Color"/>
            <hatch_fill enabled="true" type="Perforated" color="White" size="10px" style="border-radius: 10px;"/>
        </hover>
        <selected_hover>
            <fill color="%Color"/>
            <hatch_fill type="Checkerboard" color="White" style="border-radius: 10px;"/>
        </selected_hover>
        <selected_normal>
            <fill color="%Color"/>
            <hatch_fill type="Checkerboard" color="White" style="border-radius: 10px;"/>
        </selected_normal>
    </states>
</funnel_style>
<marker_settings enabled="true">
    <marker type="None" anchor="Center" v_align="Bottom" style="border-radius: 10px;"/>
    <fill color="Yellow"/>
    <border color="DarkColor(Yellow)"/>
    <states>
        <hover>
            <marker type="Star5"/>
        </hover>
        <selected>
            <marker type="Cross"/>
        </selected>
    </states>
</marker_settings>
```

```

        <pushed>
            <marker type="Star5" size="8"/>
        </pushed>
        <selected_hover>
            <marker type="Star5" size="14"/>
        </selected_hover>
        <selected_normal>
            <marker type="Star5"/>
        </selected_normal>
    </states>
    </marker_settings>
</funnel_series>
</data_plot_settings>
data_xml;
</chart>
</charts>
</anychart>

// CIRCULAR GAUGE CHART, DATA VARIABLE MUST BE A NUMBER
case 'circulargauge':
    let chart = <anychart>
        settings_xml;
        <margin all="0"/>
        <gauges>
            <gauge>
                <chart_settings>
                    <title enabled=(title is not nil)>
                        <text> title </text>
                    </title>
                    <chart_background>
                        <border enabled="false"/>
                    </chart_background>
                </chart_settings>
                <circular name="data">
                    <axis radius="37" start_angle="85" sweep_angle="190">
                        <labels align="Outside" padding="6">
                            <format> "{%Value}{numDecimals:0}" </format>
                        </labels>
                        <scale_bar>
                            <fill color="#292929"/>
                        </scale_bar>
                        <major_tickmark align="Center" length="10" padding="2" />
                        <minor_tickmark enabled="false"/>
                    <color_ranges>
                        <color_range start=(min) end=(max) align="In">

```

```
<fill type="Gradient">
    <gradient>
        <key color="Red"/>
        <key color="Yellow"/>
        <key color="Green"/>
    </gradient>
</fill>
<border enabled="true" color="Black" opacity="0.7" style="Solid" width="1px" />
</color_range>
</color_ranges>
</axis>
<frame>
    <inner_stroke enabled="false"/>
    <outer_stroke enabled="false"/>
    <background>
        <fill type="Gradient">
            <gradient angle="45">
                <key color="#FDFDFD"/>
                <key color="#F7F3F4"/>
            </gradient>
        </fill>
        <border enabled="true" color="#A9A9A9" style="Solid" width="1px" />
    </background>
    <effects enabled="false"/>
</frame>
<pointers>
    <pointer value=(data) name="value">
        <label enabled="true" under_pointers="true">
            <position placement_mode="ByPoint" x="50" y="50" />
            <format> "{%Value}{numDecimals:0}%" </format>
            <background enabled="false"/>
        </label>
        <needle_pointer_style thickness="7" point_type="triangle">
            <fill color="Rgb(230,230,230)"/>
            <border color="Black" opacity="0.7" style="Solid" width="1px" />
            <effects enabled="true">
                <bevel enabled="true" distance="2" style="Outer" />
                <drop_shadow enabled="true" distance="5" style="Outer" />
            </effects>
            <cap>
                <background>
                    <fill type="Gradient">
                        <gradient type="Linear" angle="45">
                            <key color="#D3D3D3"/>
                            <key color="#6F6F6F"/>
                        </gradient>
                    </fill>
                </background>
            </cap>
        </needle_pointer_style>
    </pointer>
</pointers>
```

```

                </gradient>
            </fill>
            <border color="Black" opacity="0.8" width="1px" style="border-radius: 50%; border-style: solid; border-width: 1px; border-color: black; border-top: none; border-left: none; border-bottom: none; border-right: none; margin: 0 auto; width: 100px; height: 100px; position: relative; z-index: 1;>
            <background>
                <effects enabled="true">
                    <bevel enabled="true" distance="2" style="border-radius: 50%; border-style: solid; border-width: 1px; border-color: black; border-top: none; border-left: none; border-bottom: none; border-right: none; margin: 0 auto; width: 100px; height: 100px; position: absolute; z-index: 0; top: -10px; left: -10px; border-radius: 50%;>
                    <drop_shadow enabled="true" distance="2" style="border-radius: 50%; border-style: solid; border-width: 1px; border-color: black; border-top: none; border-left: none; border-bottom: none; border-right: none; margin: 0 auto; width: 100px; height: 100px; position: absolute; z-index: 0; top: -10px; left: -10px; border-radius: 50%;>
                </effects>
            </background>
            <pointer>
                <cap style="border-radius: 50%; border-style: solid; border-width: 1px; border-color: black; border-top: none; border-left: none; border-bottom: none; border-right: none; margin: 0 auto; width: 100px; height: 100px; position: absolute; z-index: 1; top: 0px; left: 0px; border-radius: 50%;>
                <needle_pointer_style>
                    <animation enabled="true" start_time="0" duration="1000" style="border-radius: 50%; border-style: solid; border-width: 1px; border-color: black; border-top: none; border-left: none; border-bottom: none; border-right: none; margin: 0 auto; width: 100px; height: 100px; position: absolute; z-index: 1; top: 0px; left: 0px; border-radius: 50%;>
                </needle_pointer_style>
            </pointer>
        </pointers>
    </circular>
</gauge>
</gauges>
</anychart>

// LINEAR GAUGE, DATA VARIABLE MUST BE A NUMBER
case 'lineargauge':
    let chart = <anychart>
        settings_xml;
        <margin all="0"/>
        <gauges>
            <gauge>
                <chart_settings>
                    <title>
                        <text> title </text>
                    </title>
                    <chart_background>
                        <border enabled="false"/>
                    </chart_background>
                </chart_settings>
                <linear name="data">
                    <axis size="0" position="50">
                        <scale minimum=(min) maximum=(max) major_intervals="10" style="border: 1px solid black; width: 100%; height: 10px; position: relative; z-index: 1;>
                            <scale_bar enabled="false"/>
                            <labels padding="5"/>
                            <color_ranges>
                                <color_range start=(min) end=(max) align="Outer">
                                    <fill type="Gradient">
                                        <gradient angle="90">
                                            <key color="Red"/>
                                            <key color="Yellow"/>
                                            <key color="Green"/>
                                        </gradient>
                                    </fill>
                                </color_range>
                            </color_ranges>
                        </scale>
                    </axis>
                </linear>
            </gauge>
        </gauges>
    </anychart>

```

```
        </gradient>
    </fill>
    <border enabled="true" type="Solid" colo
</color_range>

        </color_ranges>
    </axis>
    <pointers>
        <pointer type="Marker" value=(data) name="value"
            <tooltip enabled="true"/>
            <marker_pointer_style align="Outside" padding="10px" font_size="12px" font_weight="bold" color="black" border_color="black" border_width="1px" border_radius="5px" background_color="white" border_type="Solid" border_type="Solid" border_type="Solid" border_type="Solid"/>
            <animation enabled="true" start_time="0" duration="1000" easing="ease-in-out">
                <label enabled="true">
                    <position placement_mode="ByAnchor" vertical="bottom" horizontal="left" offset_x="10px" offset_y="10px" alignment="right" baseline="top" style="font-size: 14px; font-weight: bold; color: black; border: 1px solid black; padding: 5px; background-color: white; border-radius: 5px; margin-bottom: 5px; position: absolute; z-index: 1000; opacity: 0.8; transition: all 0.3s ease-in-out; transform: rotate(-15deg);>
                    <format> "{%Value}{numDecimals:0}%" </format>
                    <background enabled="false"/>
                </label>
            </pointer>
        </pointers>
    </linear>
</gauge>
</gauges>
</anychart>

default:
    if(!error) {
        let error = "Invalid chart type selected (did not recognize '"
    }
}

// check if there was an error
if(error) {
    <p style="color: red"> error </p>
} else {
    anychart(chart, width, height, id);
}
```