

```
/**
```

```
USAGE:
```

```
Chart(type, data, height, width, xaxis, yaxis, title, min, max,  
interval, id)
```

```
PARAMETERS:
```

```
type : str (one of 'circulargauge', 'column', 'multiseriescolumn',  
'lineargauge', 'line', 'bar', 'multiseriesbar', 'pie', 'pyramid', or  
'funnel')
```

```
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,  
otherwise the value is a relative percentage.
```

```
(optional) width : num (default: 450)
```

```
Width of chart in pixel or percent.
```

```
If value is greater or equal to 1, then value represents pixel,  
otherwise the value is a relative percentage.
```

```
(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  
interacting with chart through JavaScript.
```

```
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;
new default look for progress gauge
1.3      16-Jul-10     steveb      fixed improper data handling for
'pie' type. support for min,max

```

```

*** /

```

```

// 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') ?
"opposite" : nil)>
    <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)
minimum=(min) maximum=(max) />

```

```

    </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}) -
{%Value}" </format>
              </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
[{{label1:value1}}, {{label2:value2}}, {{label3:value3}}]
// MULTI-SERIES COLUMN CHART, INCOMING DATA MUST BE FORMATTED AS
{series1:[{{label1:value1}}, {{label2:value2}}, {{label3:value3}}],
series2:[{{label1:value1}}, {{label2:value2}}, {{label3:value3}}],
series3:[{{label1:value1}}, {{label2:value2}}, {{label3:value3}}]}
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_mode="true" z_aspect="0.25">
                    <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>

                    // check if we plotting a series of data points
                    if(data is map) {
                        <legend enabled="true" position="Bottom"
align="Spread" ignore_auto_item="true" padding="15">
                            <format> "{%Icon} {%Name}" </format>
                            <title enabled="false"/>
                            <columns_separator enabled="true"/>
                            <background>
                                <inside_margin left="10" right="10"/>
                            </background>
                            <items>
                                <item source="Series"/>
                            </items>
                        </legend>
                    }
                    axes_xml;
                </chart_settings>
                data_xml;
            </chart>

```

```

    </charts>
  </anychart>;

// MULTI-SERIES LINE CHART, INCOMING DATA MUST BE FORMATTED AS
{series1:[{label1:value1}, {label2:value2}, {label3:value3},
{label4:value4}], series2:[{label1:value1}, {label2:value2},
{label3:value3}, {label4:value4}], series3:[{label1:value1},
{label2:value2}, {label3:value3}, {label4:value4}]}
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, name3:value3}
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"
halign="Center" padding="20"/>
        <font color="White">
            <effects>
                <drop_shadow enabled="true"
distance="2" opacity="0.5" blur_x="2" blur_y="2"/>
            </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 ..
'">'
        }
    ) ..
    '</series>'
</data>
<chart_settings>
    <title enabled="true" padding="15">
        <text>' .. title .. '</text>'
    </title>
    <legend enabled="true" position="Bottom"
align="Spread" ignore_auto_item="true" padding="15">
        <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>

```

```

</anychart>';

// PYRAMID/FUNNEL CHART, DATA VARIABLE MUST BE FORMATTED AS
{label1:value1, label2:value2, label3:value3}
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 ? 0 : nil) fit_aspect="1" min_width=(ispyramid ? 0
: nil) padding=(ispyramid ? 0 : nil) mode="Square">
            <animation enabled="true" type="Appear"
show_mode="Smoothed" start_time="0.3" duration="1.3"
interpolation_type="Cubic"/>
            <connector enabled="true" color="Black"
opacity="0.4"/>
            <tooltip_settings enabled="true">
              if(ispyramid) {
                <position anchor="CenterRight"
padding="10" valign="Center" halign="right"/>
              }
              <format> "{%Name} - {%YValue}{numDecimals:0}"
</format>
            </tooltip_settings>
            <label_settings enabled="true">
              <animation enabled="true" type="Appear"
show_mode="Smoothed" start_time="0.3" duration="1.3"
interpolation_type="Cubic"/>
              if(ispyramid) {
                <position anchor="Center" valign="Center"
halign="Center" padding="10"/>
              } 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(%Color)" thickness="2"/>
                </background>
            </hover>
            <pushed>
                <background>
                    <border type="Solid"
color="#494949" thickness="2" opacity="0.7"/>
                </background>
            </pushed>
            <selected_hover>
                <background>
                    <border type="Solid"
color="DarkColor(%Color)" thickness="2" opacity="0.7"/>
                </background>
            </selected_hover>
            <selected_normal>
                <background>
                    <border type="Solid"
color="DarkColor(%Color)" thickness="2"/>
                </background>
            </selected_normal>
        </states>
    </label_settings>
    <funnel_style>
        <border color="Black" opacity="0.05"/>
        <states>
            <hover>
                <fill color="%Color"/>
                <hatch_fill enabled="true"
type="Percent50" color="White" opacity="0.3"/>
            </hover>
            <selected_hover>
                <fill color="%Color"/>
                <hatch_fill type="Checkerboard"
color="#404040" opacity="0.1"/>
            </selected_hover>
            <selected_normal>
                <fill color="%Color"/>
                <hatch_fill type="Checkerboard"
color="Black" opacity="0.1"/>
            </selected_normal>
        </states>
    </funnel_style>
    <marker_settings enabled="true">
        <marker type="None" anchor="Center"
v_align="Center" h_align="Center" size="12"/>

```



```

        <fill color="Yellow"/>
        <border color="DarkColor(Yellow)"/>
        <states>
            <hover>
                <marker type="Star5"/>
            </hover>
            <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"
size="3">
                        <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="0"/>

```

```

        <minor_tickmark enabled="false"/>
        <color_ranges>
            <color_range start=(min) end=(max)
align="Inside" start_size="15" end_size="15" padding="6">
                <fill type="Gradient">
                    <gradient>
                        <key color="Red"/>
                        <key color="Yellow"/>
                        <key color="Green"/>
                    </gradient>
                </fill>
                <border enabled="true" color="Black"
opacity="0.4"/>
            </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"/>
        </background>
        <effects enabled="false"/>
    </frame>
    <pointers>
        <pointer value=(data) name="value">
            <label enabled="true" under_pointers="true">
                <position placement_mode="ByPoint" x="50"
y="60"/>
                <format> "{%Value}{numDecimals:0}%"
</format>
                <background enabled="false"/>
            </label>
            <needle_pointer_style thickness="7"
point_thickness="5" point_radius="3">
                <fill color="Rgb(230,230,230)"/>
                <border color="Black" opacity="0.7"/>
                <effects enabled="true">
                    <bevel enabled="true" distance="2"
shadow_opacity="0.6" highlight_opacity="0.6"/>
                    <drop_shadow enabled="true"
distance="1" blur_x="1" blur_y="1" opacity="0.4"/>
                </effects>
            </needle_pointer_style>
        </pointer>
    </pointers>
    <cap>

```

```

        <background>
            <fill type="Gradient">
                <gradient type="Linear"
angle="45">
                    <key color="#D3D3D3"/>
                    <key color="#6F6F6F"/>
                </gradient>
            </fill>
            <border color="Black"
opacity="0.9"/>
        </background>
        <effects enabled="true">
            <bevel enabled="true"
distance="2" shadow_opacity="0.6" highlight_opacity="0.6"/>
            <drop_shadow enabled="true"
distance="1.5" blur_x="2" blur_y="2" opacity="0.4"/>
        </effects>
        </cap>
        </needle_pointer_style>
        <animation enabled="true" start_time="0"
duration="0.7" interpolation_type="Sine"/>
    </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_interval=(interval) minor_interval=(interval / 4) />
                        <scale_bar enabled="false"/>
                        <labels padding="5"/>
                        <color_ranges>

```

```

        <color_range start=(min) end=(max)
align="Outside" padding="0" start_size="8" end_size="8">
        <fill type="Gradient">
            <gradient angle="90">
                <key color="Red"/>
                <key color="Yellow"/>
                <key color="Green"/>
            </gradient>
        </fill>
        <border enabled="true" type="Solid"
color="Black" opacity="0.4"/>
        </color_range>

        </color_ranges>
    </axis>
    <pointers>
        <pointer type="Marker" value=(data) name="value"
color="#4662B0">
            <tooltip enabled="true"/>
            <marker_pointer_style align="Outside"
padding="5" width="10" height="10"/>
            <animation enabled="true" start_time="0"
duration="1" interpolation_type="Elastic"/>
            <label enabled="true">
                <position placement_mode="ByAnchor"
valign="Center" halign="Right" padding="45"/>
                <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 '" ..
type .. "')";
    }
}

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

```