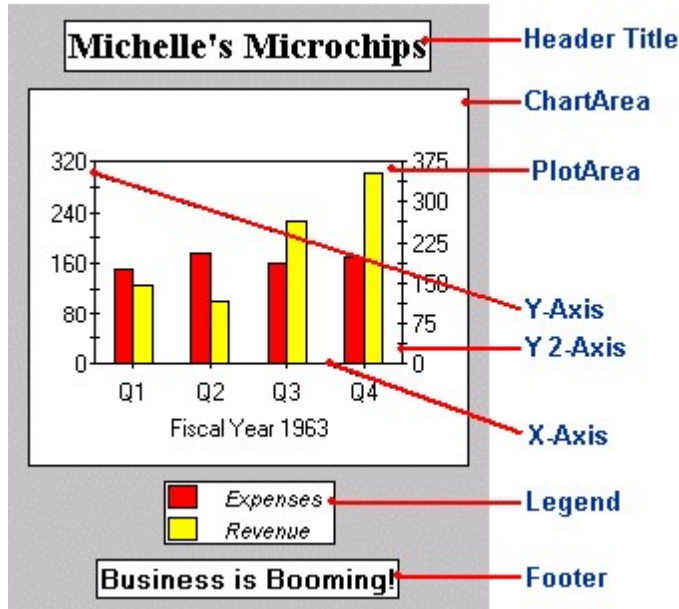


# 图表基础

C1Chart 控件含有一个丰富的对象模型来让您可以创建大量的图表类型, 并控制它们的外观和行为. 本章节通过描述图表中的主要元素和主要属性之间的关联来向您介绍 C1Chart 的对象模型. 在阅读完本章节后, 您将对C1Chart对象模型中的主要元素有一个很好的理解, 这将帮助您更好地使用 C1Chart 来创建您的控件. 后续的章节将会深入到更深的细节中, 覆盖编程模型中的剩余部分.

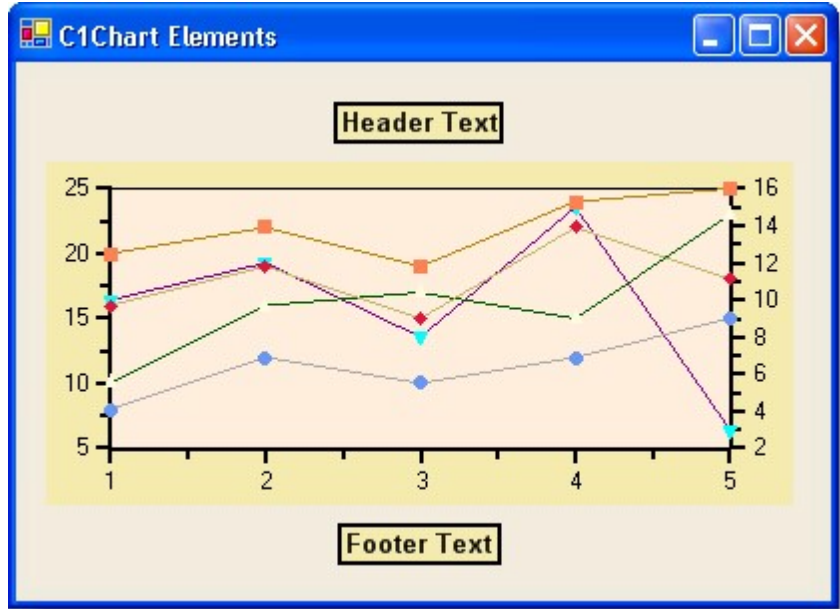
2D 图表技术  
下表展示了用于描述图表元素的术语:



注意: 饼状图表含有除了轴线之外的相同元素.  
后续的主题将会描述在处理图像时的元素, 和每个元素相关的主要属性.

## 定义表头和页尾元素

这里展示了表头和页尾元素是如何展示在图表上的.



表头和页尾是用来展示图表的描述性信息的. 它们由Header和Footer属性控制. 这两个属性都返回一个 Title 对象, 以下是该对象含有一些的主要属性:

属性	描述
Text	在标题(表头和页尾)上显示的文字.

Style	用来设置标题的字体,特性, 颜色, 边框的属性.
Compass	决定标题的位置.
Visible	决定标题是否可见.

C1Chart 自动处理标题的大小和位置, 基于它们的内容和 Compass 属性的设置. 标题对象位置可用使用 SizeDefault 和 LocationDefault 属性来进行自定义 (如果是负值, 会激活自动的位置和大小设置).

创建表头和页尾元素

图表的表头和页尾元素可以通过它们的Title对象以编程方式创建, 或者可以通过使用图表属性窗口, 图表属性编辑器, 图表智能设计器来在设计时创建.

最简单的创建方式是使用图表智能设计器, 关于使用图表智能设计器创建表头和页尾的更多信息, .

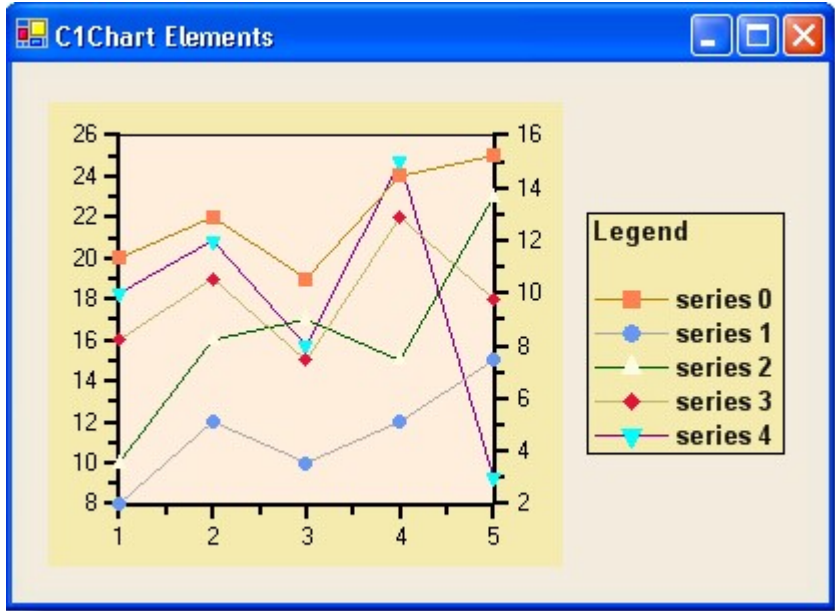
自定义表头和页尾元素

可以通过 Title 的属性来自定义表头和页尾的文字, 对齐, 位置, 边框, 颜色, 和字体. 另外, 可以使用视觉效果设计器来给表头和页尾元素添加光源模式, 阴影, 自定义颜色.

来获取关于增强 2D 图表元素的属性和工具的更多信息.

## 定义图例元素

下图显示了图表中的图例元素.



图例元素显示了图表中每一个数据序列的信息. 图表图例显示了物理颜色和数据序列的映射. 图例由 Legend 属性控制, 该属性返回一个 Legend 对象, 该对象含有以下的主要属性:

属性	描述
Text	显示在图例标题上的文字.
Style	用于设置图例的字体, 定位, 颜色, 和边框信息.
Compass	决定图例的位置.
Visible	决定图例是否可见.
Orientation	决定图例元素是横向还是竖向显示.
Reversed	控制ChartGroup显示在图例上的顺序, 一个ChartGroup元素显示在图例上的顺序是由每一个CahrtGroup对象的LegendReversed属性来决定的.

C1Chart 自动处理图例的大小和位置, 基于它们的内容和 Compass 属性的设置. 标题对象位置可用使用 SizeDefault 和 LocationDefault 属性来自定义 (如果是负值, 会激活自动的位置和大小设置).

请注意Legend属性并不控制每一个图例元素的文字, 它由每一个序列附件的标签决定, 例如, 标签含有文字, LegendEntry 属性决定序列标签是否显示在图例上.

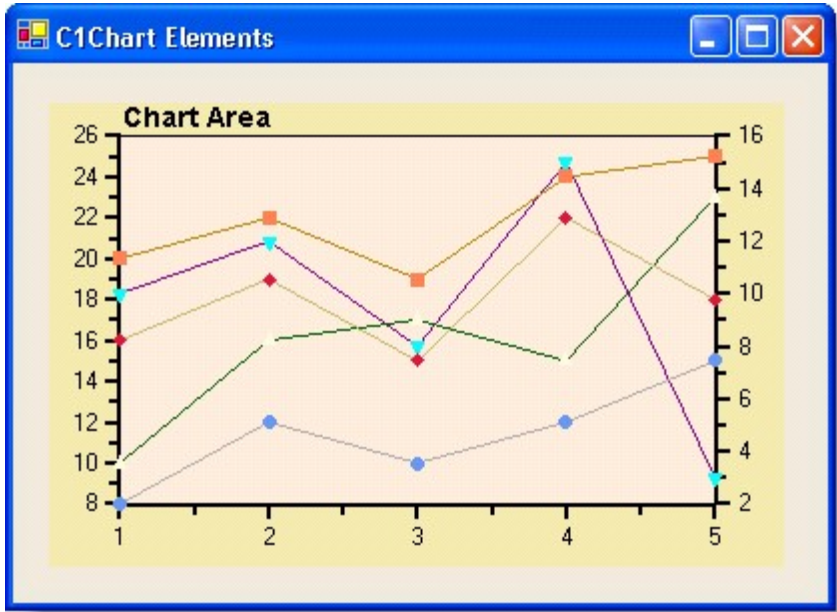
创建图例元素

图表的图例元素可以通过它们的legend对象以编程方式创建, 或者可以通过使用图表属性窗口, 图表属性编辑器, 图表智能设计器来在设计时创建.

最简单的创建方式是使用图表智能设计器,

# 定义图表区域元素

下图显示了 C1Chart 中的图表区域元素。

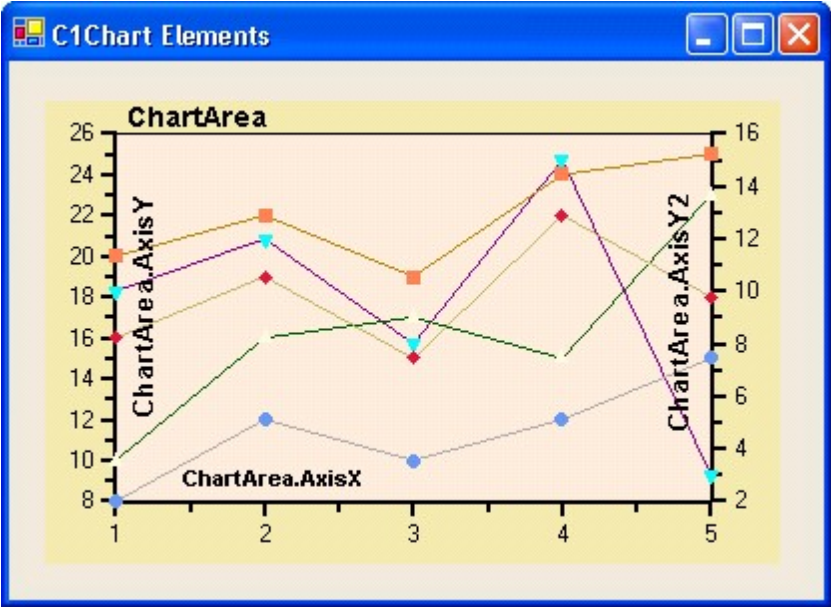


ChartArea对象代表那些包含数据的图表区域(不包括标题和图例,但是包括轴线),ChartArea 属性返回一个 Area 对象,该对象含有以下主要的属性:

属性	描述
AxisX, AxisY, AxisY2	这些属性中的每一个都会返回一个 Axis 对象,该对象让您能够自定义轴线的外观. Axis 对象会在下一个章节轴线对象 (64页)中进行解释,并且会在 <a href="#">轴线</a> (205页)的主题中做进一步的介绍.
Inverted	允许您以 90 度旋转图表(当需要创建水平柱状图时将很有用),更多的信息,请参见 <a href="#">翻转</a> 和 <a href="#">旋转图表轴线</a> (219 页).
Margins	返回一个Margin对象,该对象让您指定图表区域和绘制区域的距离. 轴线标签在这个区域被显示.
PlotArea	返回一个 PlotArea 对象,该对象控制轴线内的区域的外观. 关于此的更多信息,请参见 <a href="#">绘制区域</a> (227 页).
Style	包括设置图表区域的颜色和边框的属性.

## 轴线对象

下图展示了 C1Chart 的 X 和 Y 轴线:



多数的图表含有两个轴线, X 和 Y. 例外的是饼状图(没有轴线), 和含有第二个 Y 轴线的图表 (Y2, 所以含有三个轴线). 这些轴线被ChartArea的子属性代表: ChartArea.AxisX, ChartArea.AxisY, 和AxisY2. 这些属性中的每一个都返回一个 Axis 对象, 该对象含有以下的主要属性:

- 布局和样式属性

下面描述了在C1Chart控件中轴线的布局和式样属性.

属性	描述
Compass	允许您设置轴线的位置. 例如, 您可能想在数据的上面而不是下面显示X轴线. 关于此的更多信息, 请参见 <a href="#">轴线位置</a> (205页).
Font	设置在轴线上显示值时使用的字体. 关于更多信息, 请参见 <a href="#">轴线外观</a> .
ForeColor	设置用于展示轴线, 刻度线, 和值元素时使用的的颜色. 关于更多信息, 请参见 <a href="#">轴线外观</a> .
Reversed	允许您翻转轴线的方向. 例如, 您可以从上到下显示Y的值, 关于此的更多信息, 请参见 <a href="#">翻转和旋转图表轴线</a> (219页).
Text	设置一个紧挨着轴线显示的字符串 (典型的应用场景是描述被轴线叙述的变量和单位). 关于此的更多信息, 请参见 <a href="#">轴线标题和旋转</a> (206页).
Rotation	设置文字字符串的方向.

- 注解属性

下面描述了在C1Chart控件中轴线的注解属性.

属性	描述
AnnoFormat	一个预先定义的格式集合, 用来格式化紧挨着轴线显示的值.
AnnoFormatString	.NET格式化字符串是用来当AnnoFormat属性被设置为
	"NumericManual" 或者 "DateManual" 时用来格式化紧挨着轴 "NumericManual" 或者 "DateManual" 并且 AnnoFormatString为空时, 图表使用一个算法来寻址"最优"格式.
AnnoMethod	决定哪些值紧挨着轴线显示. 当设置为"Values"时, 显示真实
ValueLabels	一个文字/值的集合, 当AnnoMethod设置为"ValueLabels"时, 轴线上显示产品名称), 关于此的更多信息, 请参见 <a href="#">值标签注</a> (222页).
AnnotationRotation	运行您旋转值, 以让它们占用轴线上更少的空间. 关于此的更

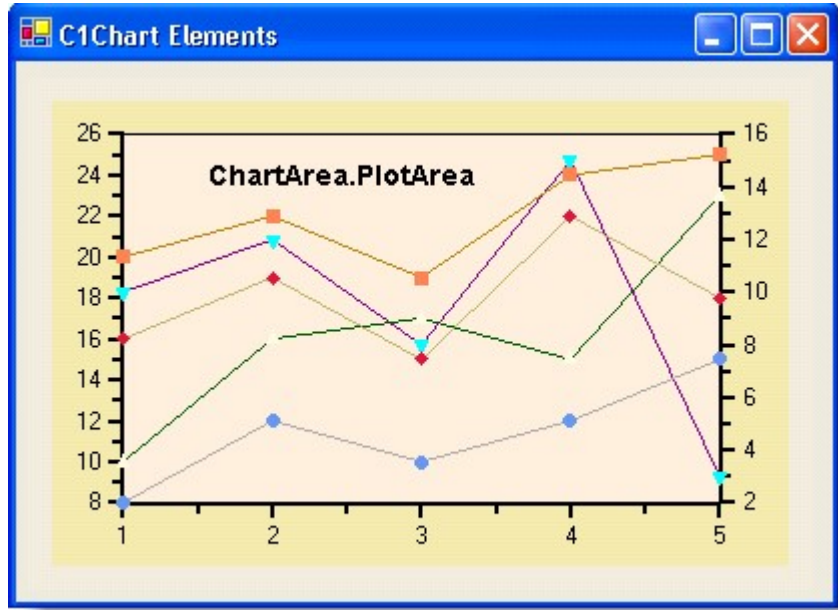
- 缩放, 刻度线和表格线属性

下面描述了在C1Chart控件中轴线的缩放, 刻度线, 网格线样式和功能属性.

属性	描述
AutoMin, AutoMax	决定了轴线的最大值和最小值是否应该被自动计算, 关于此的更多信息, 请参见 <a href="#">轴线边界</a> (209页).
Min, Max	设置轴线的最大值和最小值 (当AutoMin和AutoMax被设置为 False时). 关于此的更多信息, 请参见 <a href="#">轴线边界</a> (209页).
AutoMajor, AutoMinor	决定了主要和次要刻度线之间的间距是否应该被自动计算.
UnitMajor, UnitMinor	设置主要和次要刻度线之间的间距 (当AutoMajor和 AutoMinor属性被设置为False时).
GridMajor, GridMinor	返回一个ChartGridStyle对象, 该对象含有控制在主要和次要刻度线间绘制垂直网格线的属性, 关于此的更多信息, 请参见 <a href="#">轴线网格线</a> (208页).
AutoOrigin	决定X轴是否应该根据Y轴自动计算位置.
Origin	设定X轴相对于Y轴的位置, X轴一般放置在图表的底部, 这个属性允许您定位它, 以使得它在一个给定的坐标上穿过Y轴.
TickFactorMajor	获取或者设置主要刻度线的整因子.
TickFactorMinor	获取或者设置次要刻度线的整因子.
TickGauge	获取或者设置主要刻度线之间以刻度线描述的大致距离数.

绘制区域对象

下图展示了C1Chart中的绘制区域对象



PlotArea 对象代表了用于显示数据序列的图表区域部分. PlotArea 类中最经常被使用的用于自定义该对象的属性如下:

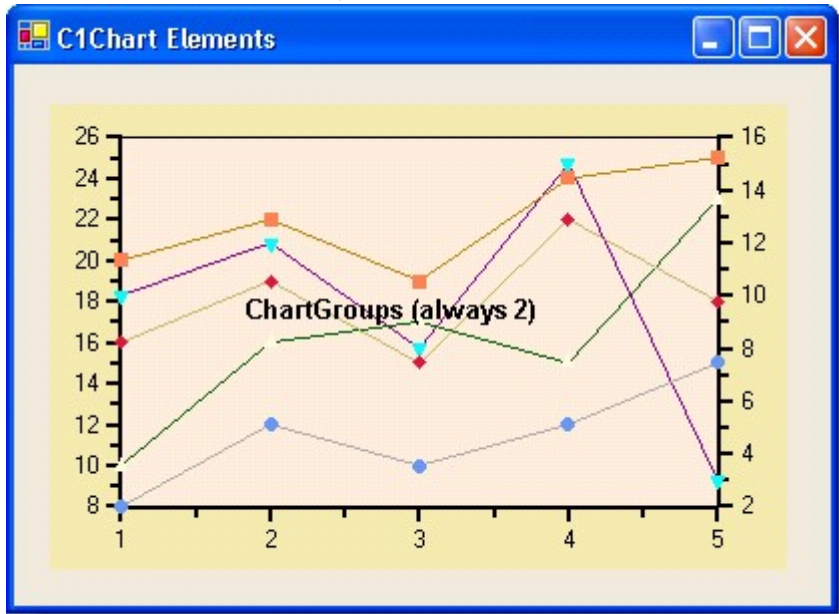
属性	描述
BackColor	设置绘制区域的背景色.
Boxed	决定绘制区域是否被一个实线边框环绕.
ForeColor	设置环绕绘制区域的边框的颜色 (当Boxed属性设置为 True 时).
View3D	返回一个View3D对象, 该对象允许您给2D图表添加 3D 效果.

关于自定义绘制区域的更多信息, 请参见[绘制区域](#) (227 页).



# 定义图表组对象

下图展示了 C1Chart 中的图表组元素。



这个属性返回一个含有两个图表组对象的集合 (通常是 Group0 和 Group1, 您不能在这个集合中进行添加或者删除动作). 这些对象决定了图表类型并且包含将被用于绘制的数据. 第一个组 (Group0) 含有针对主Y轴线上被绘制的序列, 第二个组 (Group1) 可以为空, 或者它可以含有针对次要Y轴线上被绘制的序列. 因为这里含有两个图表组对象, 而且图表类型跟图表组有关系, 您可以在同一个图表中混合使用两种图表类型. 例如, 您可以创建即作为条形图显示的序列, 同时还含有作为线性图显示的序列.

ChartGroup 对象含有以下的主要属性:

属性	描述
ChartData	返回一个ChartData对象, 该对象含有一个持有该组中所有数据的SeriesList属性. 该属性在 <a href="#">图表数据序列对象</a> (68页) 的主题中有更详细的描述.
ChartType	决定在绘制该组时使用的图表类型 (这里含有很多种类的图表, 您可以给每一个图表组使用不同的类型). 关于可用的2D图表类型 <a href="#">特定的2D类型</a> (80页).
ShowOutline	决定用来显示数据的图形化的元素 (图形图, 区域, 扇形扇区) 是否应该使用由ForeColor属性指定的颜色来重点突出显示
Stacked	决定在序列被绘制时数据是否应该堆叠 (通过添加Y值).

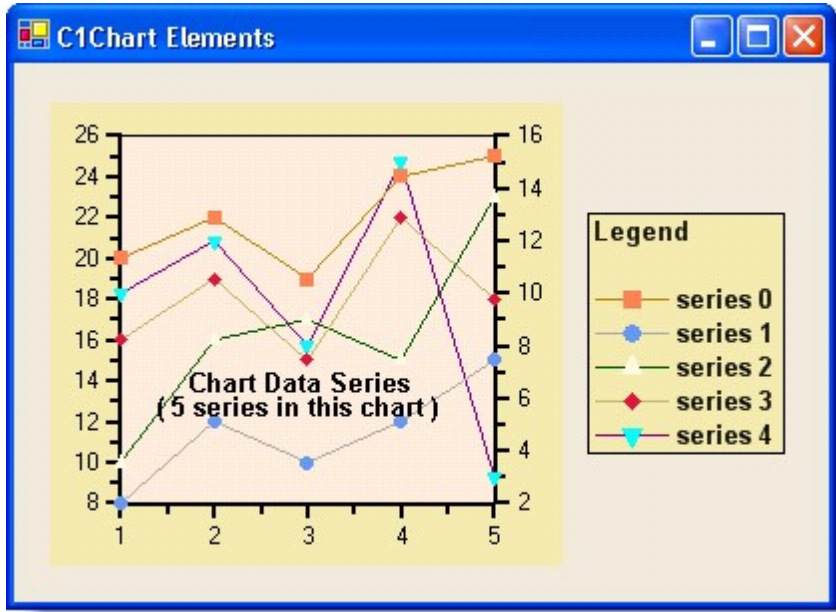
## 图表数据对象

ChartData 对象含有一些的主要属性:

属性	描述
FunctionsList	FunctionsList属性用来获取附加在当前ChartData对象上的函数集合对象. 关于FunctionsList属性的更多信息, 请参见定义 <a href="#">ChartData对象</a> (158页).
Hole	获取或者设置数据洞的值, 关于此的更多信息, 请参见 <a href="#">指定的数据洞</a> (167页).
PointStylesList	获取附加在当前ChartData对象上的PointStylesCollection对象, 关于点样式的更多信息, 请参见 <a href="#">使用点样式进行工作</a> (182页).
SeriesList	获取附加在当前ChartData对象上的DataSeriesCollection对象, 下一个主题, <a href="#">图表数据序列对象</a> (68页), 将提供关于ChartData.SeriesList属性的更多信息.
TrendsList	获取附加在当前ChartData对象上的TrendLinesCollection对象, 关于渐近线的更多信息, 请参见 <a href="#">使用渐近线进行工作</a> (174页).

## 图表数据序列对象

下图展示了在 C1Chart 中显示五个图表数据序列的示例：



从一个开发人员的角度来看, 这是在C1Chart中最重要的一个属性. 所有其它的属性都可以在设计时使用图表向导, 图表属性设计器, 或者从预先定义好的图表布局文件中加载. 在最常见的情况下, 被绘制的数据时通过代码添加的, 为了做这个工作, 您需要使用 `SeriesList` 属性. `SeriesList` 属性返回一个 `ChartDataSeriesCollection` 对象, 该对象允许您在图表上添加或者删除序列, 或者可以获取一个单独的序列. `Adding Data Series`这个示例向您展示了以编程的方式添加数据序列的方法. `ChartDataSeries` 对象含有以下的主要属性:

属性	描述
Display	决定了序列是否可见, 和如何展示缺失值(数据洞).
Label	含有展示在图例上的文字(当 <code>LegendEntry</code> 属性设置为 <code>True</code> 时).
LegendEntry	决定了序列标签是否应该被显示在图例上.
LineStyle	包含了以下的属性:用来决定序列显示时的颜色, 厚度, 和样式(颜色被用在线条, 区域, 条形图, 和扇形切片上), 关于此的更多信息, 请参见 <a href="#">序列的线条和符号样式</a> (240 页).
SymbolStyle	包含了以下的属性:用来决定在序列上标记数据点时使用的形状, 大小, 和符号的颜色. 关于此的更多信息, 请参见 <a href="#">序列的线条和符号样式</a> (240 页).
PointData	返回一个 <code>ChartDataArray</code> 对象, 该对象用来获取或者设置序列上每一个数据点的 X, Y 坐标.
X, Y	返回一个 <code>ChartDataArray</code> 对象, 该对象用来获取或者设置序列上每一个数据点单独的坐标. 一些图表类型含有附加的数据数组. (例如, <code>HiLoOpenClose</code> 还含有 <code>Y1</code> , <code>Y2</code> , 和 <code>Y3</code> ).

`PointData`, `X`, 和 `Y` 属性允许您获取或者设置用于展示每一个单独序列的数据. 大多数的图表类型需要您为每一个点都提供 `X` 和 `Y` 值. 例外是饼状图(它不需要 `X` 值)和一些特殊的图表, 例如 `Bubble`, `HiLo`, `Gantt`, 和 `HiLoOpenClose`, 它们都需要额外的 `Y` 值.