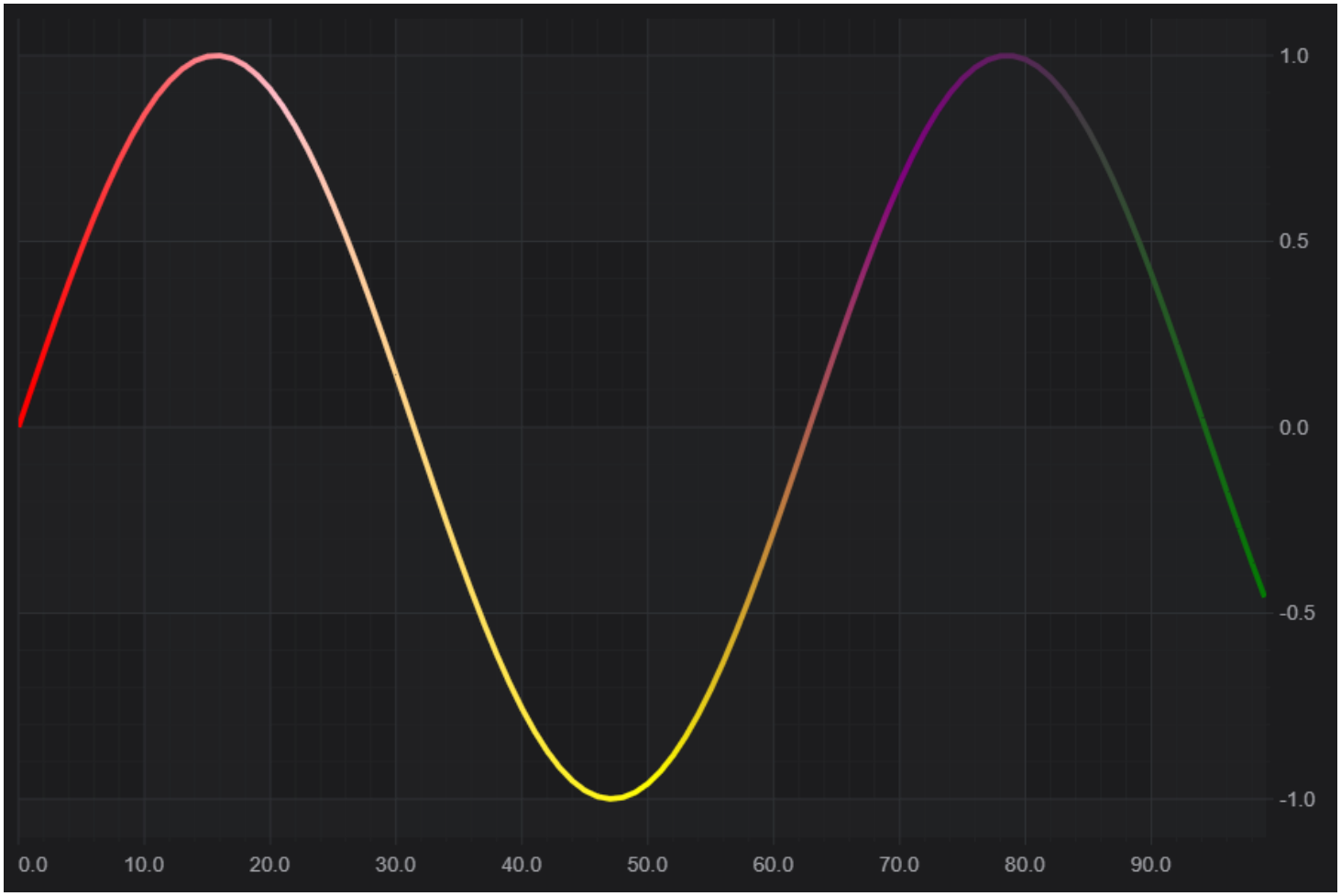
Search for example names and chart types

SciChart.js - High Performance Realtime Javascript Charts Examples Suite

JavaScript Line Chart

Demonstrates how to create a **JavaScript Line Chart** using SciChart.js, High Performance JavaScript Charts



Source Code

View on GitHub

```
import * as React from "react";
import { MouseWheelZoomModifier } from "scichart/Charting/ChartModifiers/MouseWheelZoomModifier";
import { ZoomExtentsModifier } from "scichart/Charting/ChartModifiers/ZoomExtentsModifier";
import { ZoomPanModifier } from "scichart/Charting/ChartModifiers/ZoomPanModifier";
import { PaletteFactory } from "scichart/Charting/Model/PaletteFactory";
import { XyDataSeries } from "scichart/Charting/Model/XyDataSeries";
import { NumericAxis } from "scichart/Charting/Visuals/Axis/NumericAxis";
import { FastLineRenderableSeries } from "scichart/Charting/Visuals/RenderableSeries/FastLineRenderableSeries";
import { SciChartSurface } from "scichart/Charting/Visuals/SciChartSurface";
import { GradientParams } from "scichart/Core/GradientParams";
import { NumberRange } from "scichart/Core/NumberRange";
import { Point } from "scichart/Core/Point";

const divElementId = "chart";
```

```
const divElementId = "chart";

const drawExample = async () => {
    // Create a SciChartSurface
    const { sciChartSurface, wasmContext } = await SciChartSurface.create(divElementId);

    // Create the X,Y Axis
    const xAxis = new NumericAxis(wasmContext);
    sciChartSurface.xAxes.add(xAxis);

    const yAxis = new NumericAxis(wasmContext, { growBy: new NumberRange(0.05, 0.05) });
    sciChartSurface.yAxes.add(yAxis);

    // Create an XyDataSeries as data source
    const xyDataSeries = new XyDataSeries(wasmContext);
    for (let i = 0; i < 100; i++) {
        xyDataSeries.append(i, Math.sin(i * 0.1));
    }

```

// JavaScript Chart Examples

SciChart.js ships with ~40 JavaScript Chart Examples which you can browse, play with, view the source code and see related documentation. All of this is possible with the SciChart.js Examples Suite, which ships as part of the SciChart.js SDK

DOWNLOAD THE SDK

Description

Demonstrates how to create a JavaScript Line Chart. The FastLineRenderableSeries can be used to render an XyDataSeries, XyyDataSeries (uses Y1 only) or OhlcDataSeries (renders Close).

The scatter chart uses the PointMarker API to define the marker shape and size. Point-markers available out of the box include Ellipse (circle), Triangle, Square, Cross and CustomPointMarker, which renders an image.

Tips!

As well as stroke, you can set strokeThickness, isVisible properties to change how the series is rendered.

You can add data-point markers to a line series using the PointMarker API. This is very performant and uses the same WebGL rendering as our Scatter Charts.

Documentation Links

- SciChart.js Documentation Home
- SciChart.js Tutorials
- JavaScript Line Chart Documentation
- Common RenderableSeries Properties

See Also

- The JavaScript Bubble Chart Example
- Styling Point-Markers Example