

Max Kleiner

Posted on 31. Okt.

Edit Manage Stats



1

JavaScript Loader

#webdev #javascript #maxbox #pascal

Loaders are transformations that are applied to the source code of a module or a script. They allow you to pre-process files or html with javascript inside as you import or "load" them. Thus, loaders are kind of like "tasks" in other build tools and provide a powerful way to handle front-end build steps.

The idea is to download a JS programm and run it native locally in a runtime browser (webview2) with some modifications in it, based on a script you deliver before. I call this a multiple deployment as a fixture:

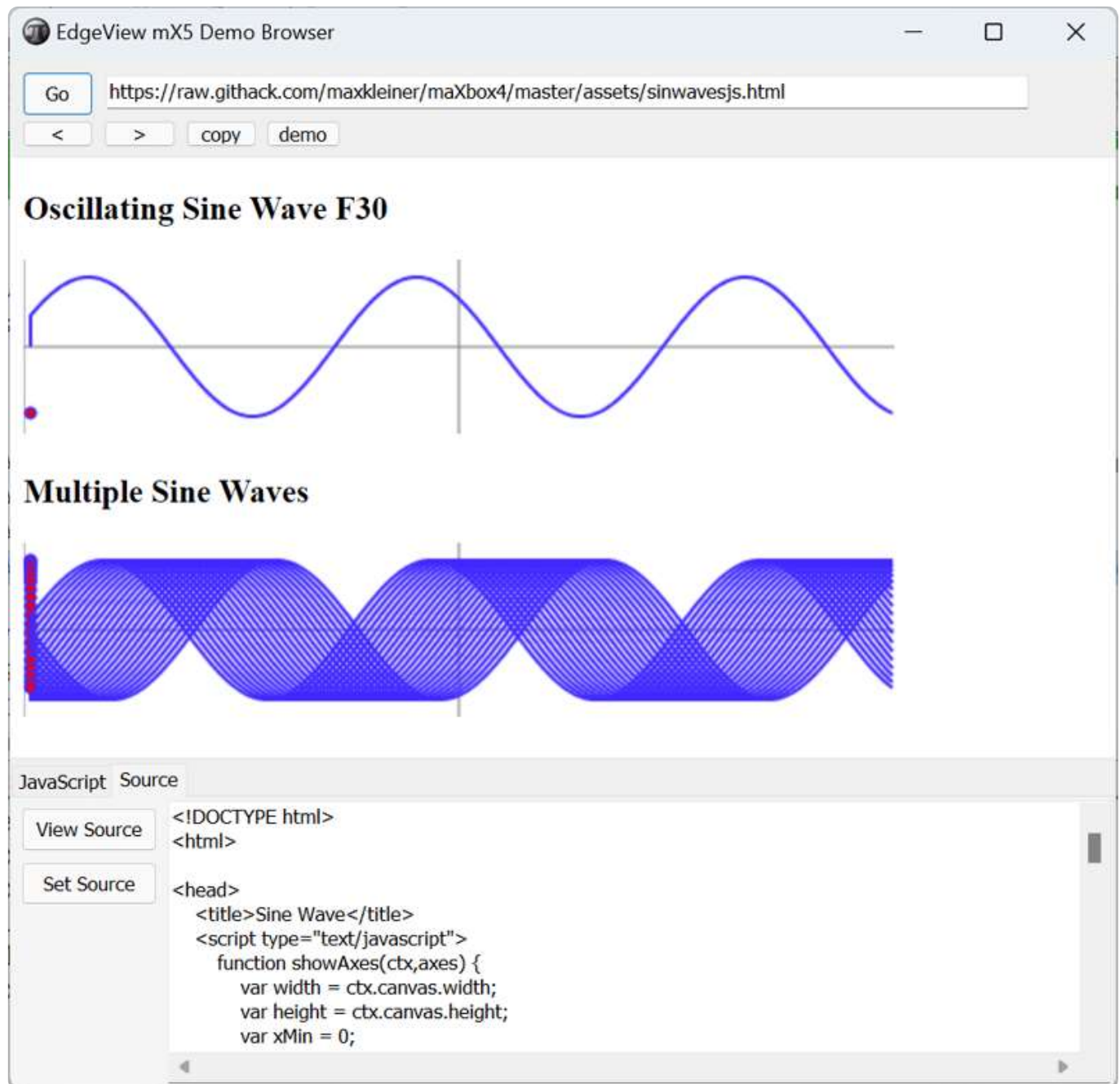
```

const
  SINWAVES2URL= 'https://raw.githack.com/maxkleiner/maXbox4/master/assets/sinwavesjs
aMS:= TMemoryStream.Create;
try
  HttpGet(SINWAVES2URL, amS)
  writeln('urlcontent size: '+itoa((ams.size)));
  aMS.Seek(0, 0);
  memoHTML.lines.loadfromstream(aMS);
  //javascript fixture_
  memoHTML.text:= StringReplace(memoHtml.text,'frequency = 20;','frequency = 30;',[
  memoHTML.text:= StringReplace(memoHtml.text,'Sine Wave</h3>','Sine Wave F30</h3>'

```

```
    navigatetoString(memoHTML.text);  
  finally  
    aMS.Free  
  end;
```

For example, you can use loaders to tell a website to load a CSS with javascript file and to modify parameters in JavaScript before you run it in a local browser of webview2.



Loaders can be chained. Each loader in the chain applies transformations to the processed resource for example load the url as a stream in a memo with lines, modify two parameters (frequency and title in our example) and run it with `navigatetoString(memoHTML.text);` on a browser:

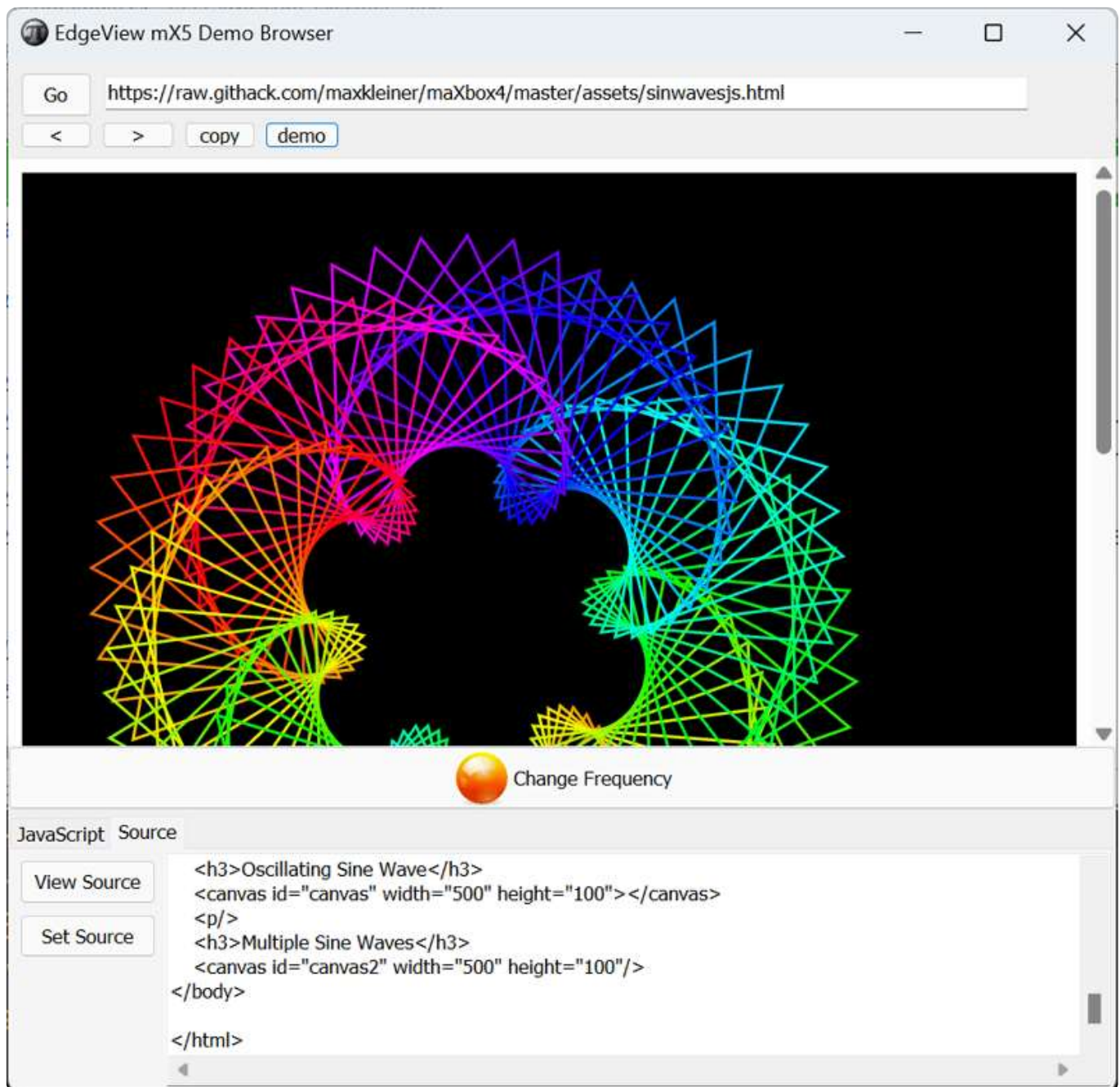
```
with TEdgeViewForm.create(self) do begin
    PageControl1.ActivePageIndex := 1;
    edit1.text:= SINWAVES2URL;
    aMS:= TMemoryStream.Create;
    try
        HttpGet(SINWAVES2URL, amS)
        writeln('urlcontent size: '+itoa((ams.size)));
        aMS.Seek(0, 0);
        memoHTML.lines.loadfromstream(aMS);
        //javascript fixture_
        memoHTML.text:= StringReplace(memoHtml.text, 'frequency = 20;', 'frequency = 30;');
        memoHTML.text:= StringReplace(memoHtml.text, 'Sine Wave</h3>', 'Sine Wave F30</h3>');
        navigatetoString(memoHTML.text);
    finally
        aMS.Free
    end;
    showmodal
    free;
end;
```

The script is available at:

[javascript_execute2_5_script](#)

When distributing your application, there are a few ways you can ensure the **WebView2** Runtime is on client machines.

The Bootstrapper evergreen is a tiny installer that downloads the Evergreen Runtime matching device architecture and installs it locally. There is also a Link that allows you to programmatically download the Bootstrapper.



One of the main benefits of using the Edge WebView2 Runtime is its support for a latest web standards such as **HTML5**, **CSS3**, and JavaScript ECMAScript 2020. This means that developers can take full advantage of this web technologies to create high-quality user experiences independent from the standard browser.

Unlike Electron and CEF (Chromium Embedded Framework), it's installed on the OS for use by any app that needs it, so apps no longer need to include it in their installer (but can if you want to). It will become pervasive on Windows through Win 10 updates and inclusion in Win 11.

Top comments (1) ◇



Max Kleiner • 1. Nov.



Evergreen Standalone Installer

A full-blown installer that can install the Evergreen Runtime in offline environment. Available for x86/x64/ARM64.

[Code of Conduct](#) • [Report abuse](#)

_SurveyJS

PROMOTED



```
package.json M SurveyCreatorComponent.jsx U X
src > SurveyCreatorComponent.jsx > ...
1  import React from "react";
2  import { SurveyCreator, SurveyCreatorComponent } from "survey-creator-react";
3  import "survey-core/survey.i18n.js";
4  import "survey-creator-core/survey-creator-core.i18n.js";
5  import "survey-core/defaultV2.css";
6  import "survey-creator-core/survey-creator-core.css";
7  import "../index.css";
8
9  const surveyJSON = {
10   elements: [
11     //...
12     {
13       type: "text",
14       name: "question1",
15       title: "My first question",
16     },
17     //...
18   ],
19 };
20
21 function SurveyCreatorRenderComponent() {
22   const options = {
23     showLogicTab: true,
24   };
25   const creator = new SurveyCreator(options);
26   creator.JSON = surveyJSON;
27   return <SurveyCreatorComponent creator={creator} />;
28 }
29
30 export default SurveyCreatorRenderComponent;
```

PROBLEMS OUTPUT TERMINAL PORTS 1

→ workspace git:(master) X npm i survey-creator-react

JavaScript Form Builder UI Component

Generate dynamic JSON-driven forms directly in your JavaScript app (Angular, React, Vue.js, jQuery) with a fully customizable drag-and-drop form builder.

Easily integrate with any backend system and retain full ownership over your data, with no user or form submission limits.

[Learn more](#)**Max Kleiner**

The Art of Coding: maXbox is a script tool engine, compiler and source lib all in one exe to design and code your scripts in a shellbook! Pure Code for Object Scripting. Principle is simplicity and re

JOINED

7. Aug. 2024

More from **Max Kleiner**

maXbox5

#webdev #deeplearning #opensource #datascience

Dragon Curve 2

#javascript #maxbox #challenge #fractal

Dragon Curve

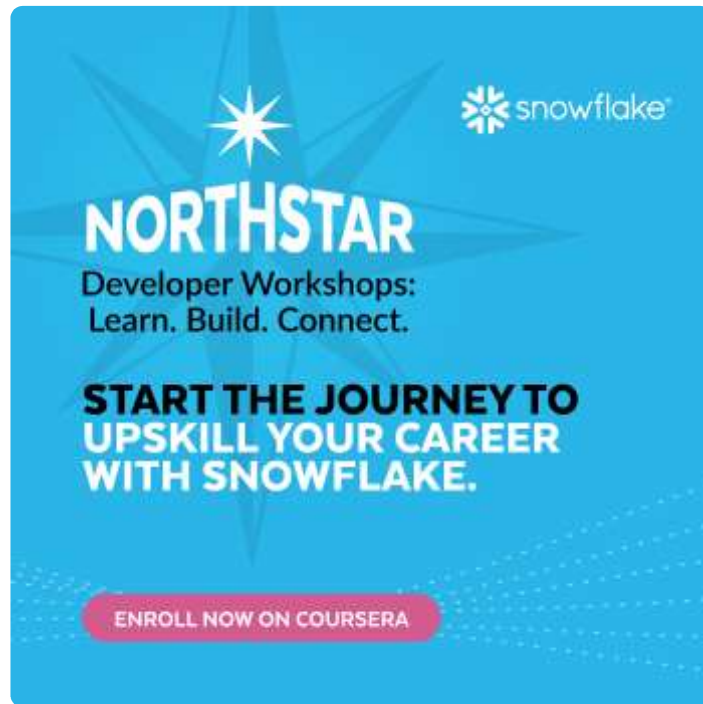
#script #maxbox #graphics #multilanguage



Snowflake

PROMOTED





[Intro to Snowflake for Devs, Data Scientists & Data Engineers](#)

Learn Snowflake's capabilities in AI, machine learning, and app development.

[Learn more](#)