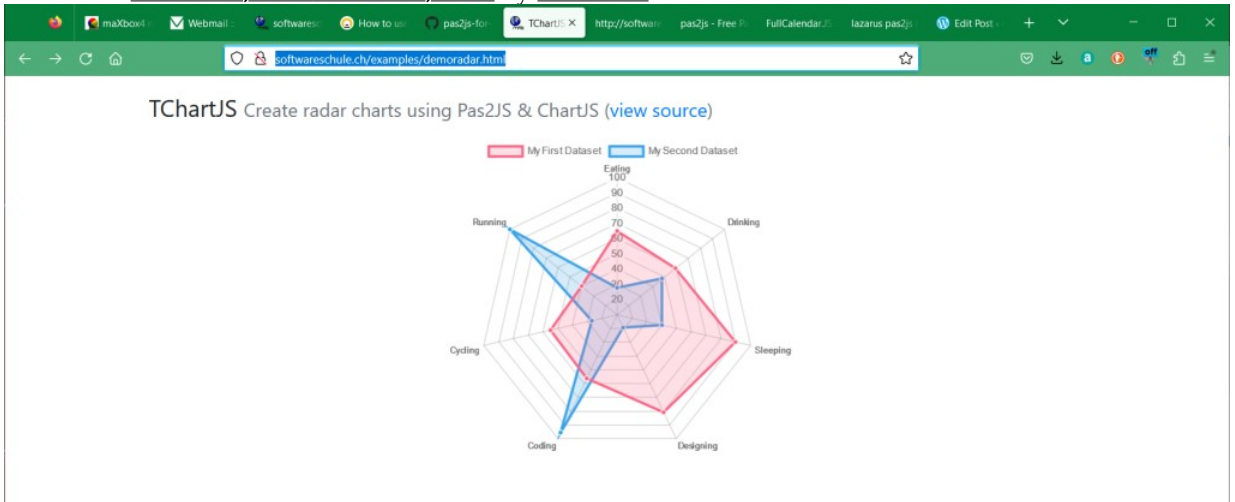


maXbox

pas2js

Posted on **March 19, 2023** **March 19, 2023** by **maxbox4**



Pas2js is an open source [Pascal \(https://wiki.freepascal.org/Pascal\)](https://wiki.freepascal.org/Pascal) to [JavaScript \(https://wiki.freepascal.org/JavaScript\)](https://wiki.freepascal.org/JavaScript) transpiler. It parses Object Pascal or maXbox files and emits JavaScript. It takes Delphi/Lazarus projects and modules (.DPR, .LPR, .PAS, .PP) and converts them to JavaScript (.JS). The JavaScript is currently of level ECMAScript 5 and should run in any browser or in Node.js (target "nodejs"). It is available in 5 forms:

- as a library
- as a command-line program
- as a webservice
- as a node.js program
- as a program running in the browser.

It transpiles from actual Pascal source, it has no intermediate .ppu files. That means all sources must always be available.

So my first example is a radarchart from ChartJS which I build in Lazarus:

```

1  program demoradar;
2
3  {$MODE OBJFPC}
4  {$MODESWITCH EXTERNALCLASS}
5
6  uses
7      ChartJS;
8
9  var
10     config: TChartConfiguration;
11     dataset1, dataset2: TChartRadarDataset;
12 begin
13     config := TChartConfiguration.new;
14     config.type_ := 'radar';
15     config.data := TChartData.new;
16     config.data.labels := ['Eating', 'Drinking', 'Sleeping', 'Designing',
17         'Coding', 'Cycling', 'Running'];
18
19     dataset1 := TChartRadarDataset.new;
20     dataset1.label_ := 'My First Dataset';
21     dataset1.data := [65, 59, 90, 81, 56, 55, 40];
22     dataset1.fill := True;
23     dataset1.backgroundColor := 'rgba(255, 99, 132, 0.2)';
24     dataset1.borderColor := 'rgb(255, 99, 132)';
25     dataset1.pointBackgroundColor := 'rgb(255, 99, 132)';
26     dataset1.pointBorderColor := '#fff';
27     dataset1.pointHoverBackgroundColor := '#fff';
28     dataset1.pointHoverBorderColor := 'rgb(255, 99, 132)';

```

Through external class definitions, the compiler can use JavaScript classes:

- All classes available in the JavaScript runtime, and in the browser are available through import units (comparable to the windows or Unix units for the native compiler).
- For Node.js, basic support for the nodejs runtime environment is available.
- An import unit for jQuery is available (libjquery)
- a converter from maXbox to lpr project files

<https://wiki.freepascal.org/pas2js> (<https://wiki.freepascal.org/pas2js>)

For the generated code to work, a small JavaScript file is needed: rtl.js. It defines an object rtl. This object will start the Object Pascal code if you include a call to rtl.run() in the [HTML](https://wiki.freepascal.org/index.php?title=HTML&action=edit&redlink=1) (<https://wiki.freepascal.org/index.php?title=HTML&action=edit&redlink=1>) page. Then I pass the file above to the transpiler:

pas2js can automatically include this file (rtl.js) in the generated output, like this:



```
pas2js -Jc -Jirtl.js -Tbrowser demoradar.lpr
```

```

>>> C:\Program Files\Streaming\maxbox4\examples\pas2js-windows-
2.2.0\pas2js-windows-2.2.0\bin\i386-win32>pas2js -Jc -Jirtl.js -Tbrowser
demoradar.lpr
Pas2JS Compiler version 2.2.0 [2022/02/22] for Win32 i386
Copyright (c) 2021 Free Pascal team.
Info: 8699 lines in 6 files compiled, 0.2 secs

```

The pas2js compiler and RTL are – naturally – open source and can be downloaded and used freely. And I got my output as a javascript file demoradar.js

```
1  var pas = { $libimports: {} };
2
3  var rtl = {
4
5      version: 20200,
6
7      quiet: false,
8      debug_load_units: false,
9      debug_rtti: false,
10
11     $res : {},
12
13     debug: function(){
14         if (rtl.quiet || !console || !console.log) return;
15         console.log(arguments);
16     },
17
18     error: function(s){
19         rtl.debug('Error: ',s);
20         throw s;
21     },
22
23     warn: function(s){
24         rtl.debug('Warn: ',s);
25     },
26
27     checkVersion: function(v){
28         if (rtl.version != v) throw "expected rtl version "+v+", but found "+rt
29     },
30
31     hiInt: Math.pow(2,53),
32
33     hasString: function(s){
34         return rtl.isString(s) && (s.length>0);
35     },
36
37     isArray: function(a) {
38         return Array.isArray(a);
39     },
40
41     isFunction: function(f){
42         return typeof(f)=="function";
43     }, and much more.....
```

This is then integrated in a html file:

```

1  <!doctype html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, shr
6      <meta name="description" content="Example showing how to use TChartJS">
7      <meta name="author" content="silvioprogram">
8      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstr
9      <script src="https://cdn.jsdelivr.net/npm/chart.js/2.7.3/Char
10     <script src="demoradar.js"></script>
11     <title>TChartJS example</title>
12     <style>
13       .title {
14         margin: 20px 0 20px 0
15       }
16     </style>
17   </head>
18   <body>
19     <div class="container">
20       <h3 class="title">
21         TChartJS <small class="text-muted">Create radar charts using Pas2JS
22       </h3>
23       <canvas id="myChart" height="100"></canvas>
24     </div>
25     <script>
26       rtl.run();
27     </script>
28   </body>
29 </html>

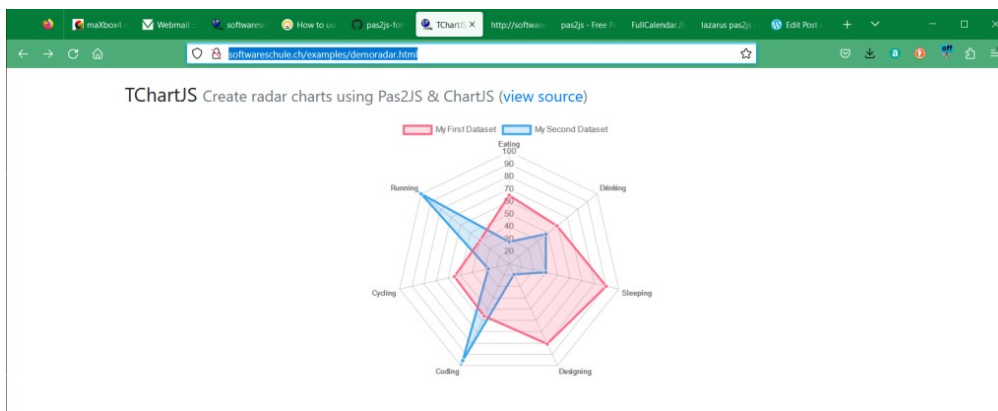
```

The `crossorigin="anonymous">` **anonymous** means:

Request uses CORS headers and credentials flag is set to 'same-origin'. There is no exchange of **user credentials** via cookies, client-side SSL certificates or HTTP authentication, unless destination is the same origin.

which you can see and call at:

<http://softwareschule.ch/examples/demoradar.html> (<http://softwareschule.ch/examples/demoradar.html>)



the example of <http://softwareschule.ch/examples/demoradar.html> (<http://softwareschule.ch/examples/demoradar.html>)

Chart.js provides a set of frequently used chart types, plugins, and customization options. In addition to a reasonable set of **built-in chart types** (<https://www.chartjs.org/docs/latest/charts/area.html>), you can use additional community-maintained **chart types** (<https://github.com/chartjs/awesome#charts>) (<https://github.com/chartjs/awesome#charts>). On top of that, it's possible to combine several chart types into a **mixed chart** (<https://www.chartjs.org/docs/latest/charts/mixed.html>) (essentially, blending multiple chart types into one on the same canvas).

Releases of pas2js

The releases contain binaries for **Windows (32 and 64bit), Linux (64 bit) and macOS**.

Installation procedure:

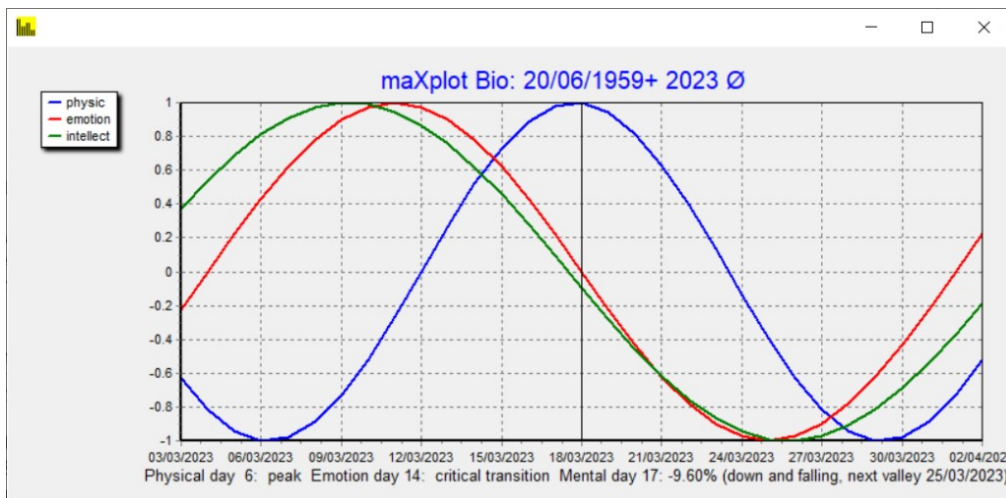
1. Download pas2js from:

- <https://downloads.freepascal.org/fpc/contrib/pas2js/> (<https://downloads.freepascal.org/fpc/contrib/pas2js/>)

Every version has a directory with the version number. A list of changes can be found on the changelog page **Pas2JS Version Changes** (https://wiki.freepascal.org/Pas2JS_Version_Changes)

2. Unpack it in folder of your choice. The example below uses **C:\lararus\pas2js**. The release contains three folders:

- bin – contains the compiler as executable (pas2js or pas2js.exe), a pas2js.cfg, a library and some utilities.
- demo – lots of examples
- packages – the Pascal units of the RTL and other packages.
- tools – html2form – HTML to pascal code converter program
- utils – A script to create a pas2js.cfg: createconfig.pp



from maXbox4 bio-script

https://wiki.freepascal.org/lazarus_pas2js_integration (https://wiki.freepascal.org/lazarus_pas2js_integration)

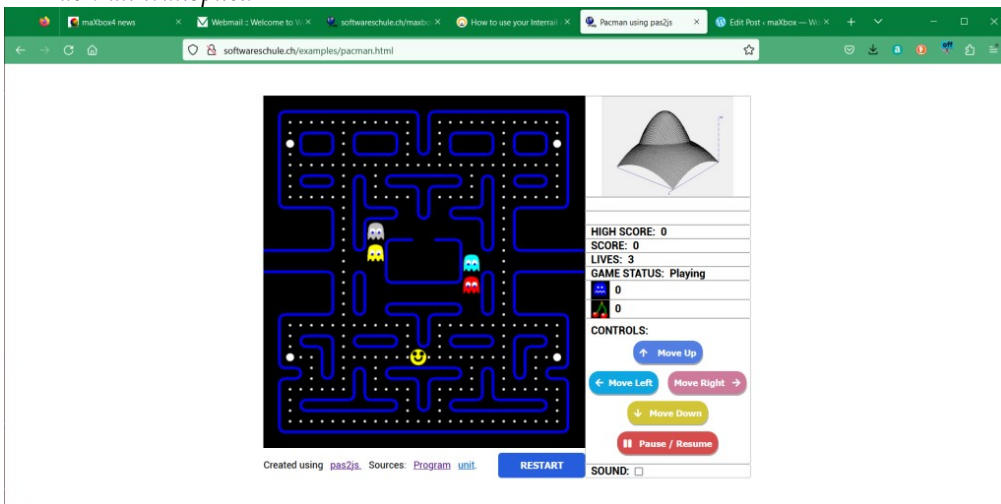


```
\maxbox4\examples\pas2js-windows-2.2.0\pas2js-windows-2.2.0\bin\i386-
win32>pas2js -Jc -Jrtl.js -Tbrowser ..\..\demo\pacman\pacman.lpr
Pas2JS Compiler version 2.2.0 [2022/02/22] for Win32 i386
Copyright (c) 2021 Free Pascal team.
```

```
\pacman.lpr(6,15)Hint:Unit"JS"notused in pacman
\pacman.lpr(6,38)Hint:Unit"Web"notused in pacman
\packages\fcl-base\browserapp.pas(32,9) Hint: Unit
"Rtl.BrowserLoadHelper" not used in browserapp
\maxbox4\examples\pas2js-windows-2.2.0\pas2js-windows-2.2.0\packages
\rtl\Rtl.BrowserLoadHelper.pas(46,32) Hint: Parameter"response"not
used
\pacman\upacman.pp(6,34)Hint:Unit"JS"notused in upacman
\upacman.pp(1403,23) Hint:Function result does not seem to be set
\upacman.pp(1414,23) Hint:Function result does not seem to be set
\upacman.pp(90,30)Hint:Parameter"Event"not used
\upacman.pp(91,35)Hint:Parameter"Sender"not used
\upacman.pp(92,33)Hint:Parameter"aEvent"not used
\upacman.pp(144,28) Hint:Private method "TPacman.DoRestartClick"is
never used
\upacman.pp(255,11) Hint:Local constant"WallSet"notused
\maxbox4\examples\pas2js-windows-2.2.0\pas2js-windows-2.2.0\demo
\pacman\upacman.pp(309,3)Hint:Localvariable"n"not used
Info: 30569 lines in 14 files compiled, 0.3 secs
```

C:\Program Files\Streaming\maxbox4\examples\pas2js-windows-2.2.0\pas2js-windows-2.2.0\bin\i386-win32>

PacMan transpiled



Pac the Man from maXbox to JS

Posted in [Lazarus](#), [maXbox](#), [Multilanguage](#) Tagged [crossorigin](#), [javascript](#), [transpiler](#) [1 Comment](#)

One thought on “pas2js”

1.

maxbox4 says:

March 19, 2023 at 4:13 pm [Edit](#)

<http://softwareschule.ch/examples/pacman.html>

[REPLY](#) ▶

[Create a free website or blog at WordPress.com.](#)