

# What's new with JavaScript in GNOME

## The 2021 edition

**Philip Chimento**

💬 pchimento • 🚩 🍅 ptomato

🐦 @therealptomato

GUADEC Online, July 22, 2021

Photo by [Lucas Pezeta](#) from [Pexels](#)



# Introduction: What we will talk about

- Same thing we talk about every year!
- **What's new** in GJS?
- **What's next** in GJS?
- We need some **help!** Here's what you can do

# Introduction

- Presentation is full of [links](#) if you want to click through later
- If you want to follow along: [ptomato.name/talks/guadec2021](https://ptomato.name/talks/guadec2021)

# What's new in JavaScript land for GNOME 40 and 41?

# Crash fixes

- ④① 10 crashing bugs fixed in the past year
  - But also, 10 crashing bugs reported in the past year
  - [Total of 9](#) currently open
  - 2 older than 1 year
- ④① Many refactors adding type safety (🎩 Marco Trevisan)

# Performance improvements ④1

- Memory usage improvements (🎩 Marco Trevisan)
  - [!625](#) for objects
  - [!519](#) for functions

# ES Modules ④0

```
// Old, but still works, not deprecated:  
const System = imports.system; // built-in module  
imports.gi.versions.Gtk = '3.0';  
const {Gtk} = imports.gi; // G-I binding  
imports.searchPath.unshift('resource:///com/example/MyApp');  
const {MyClass, myFunction} = imports.src.myModule;  
    // imports src/myModule.js from search path  
  
// New 👍  
import System from 'system'; // built-in module  
import Gtk from 'gi://Gtk?version=3.0'; // G-I binding  
import {MyClass, myFunction} from './src/myModule.js';  
    // imports path relative to currently executing file
```

# ES Modules ④0

- Run your main file with `-m`
  - or `gjs_context_eval_module()` if using the GObject-based API
- ES modules can import legacy modules, but not vice versa
  - To port your code, start at the top level and work downwards
- 🎩 Evan Welsh



# ES Modules ④0

- ES modules are always in [strict mode](#)
- Importing is compile-time!
  - To import conditionally, use dynamic async `import( )`

*// NOT POSSIBLE* ❌

```
if (usingSomeLibrary)
  import SomeLibrary from 'gi://SomeLibrary';
```

*// Possible* ✅

```
if (usingSomeLibrary)
  const SomeLibrary = await import('gi://SomeLibrary');
```

# Debugger improvements ④0

- Backtrace command that prints all the locals, like GDB
- Program listing
- Show line along with frame
- 🎩 Nasah Kuma

# Debugger improvements ④0

```
function myFunction(a, b, c) {  
    const d = Math.random();  
    debugger;  
}  
myFunction(3, 2, 1);
```

# Debugger improvements ④④

```
db> bt full
#0      myFunction(3, 2, 1) at bug.js:3:4
d = 0.7590159046381023
#1      toplevel at bug.js:5:10
```

```
db> list
1      function myFunction(a, b, c) {
2          const d = Math.random();
*3          debugger;
4      }
5      myFunction(3, 2, 1);
```

```
db> frame 1
#1      toplevel at bug.js:5:10
5      myFunction(3, 2, 1);
```

# JS Object GObject parameters ④0

```
Properties: {  
  myArray: GObject.ParamSpec.jsobject('my-array', 'My array',  
    'You can now use JS objects as the value of GObject properties',  
    GObject.ParamFlags.READWRITE),  
},
```

 Marco Trevisan

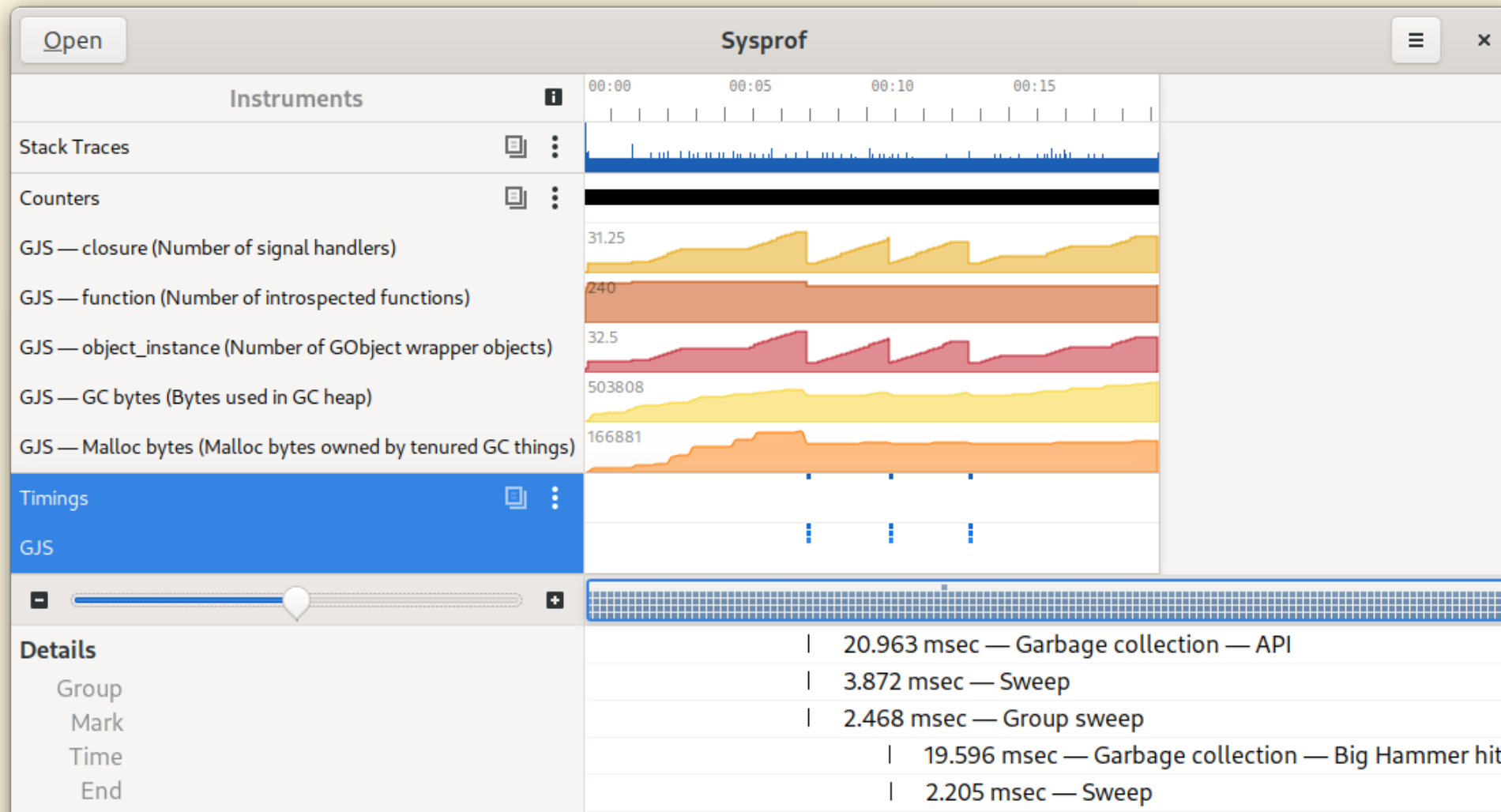
# Text encoding/decoding ④1

```
// Old, but still works, not deprecated:
const ByteArray = imports.byteArray;
const pizza = ByteArray.toString(Uint8Array.of(0xf0, 0x9f, 0x8d, 0x95) /*, 'utf-8' */);
const pizzaBytes = ByteArray.fromString('🍕' /*, 'utf-8' */);

// New 👍
const decoder = new TextDecoder(/* 'utf-8' */);
const pizza = decoder.decode(Uint8Array.of(0xf0, 0x9f, 0x8d, 0x95));
const encoder = new TextEncoder();
const pizzaBytes = encoder.encode('🍕');
```

 Evan Welsh

# Memory profiling ④1



# Console API ④1

```
// Old, but still works, not deprecated:  
log('A message from your program');  
  
// New, nice if you're used to browsers and Node:  
console.log('A message from your program');
```

 Evan Welsh



# Documentation and Examples

-  Extensions Rebooted

**What is upcoming in JavaScript land for  
GNOME?**

# Native async operations ④① ④②

## Annotations:

```
<method name="load_contents_async" c:identifier="g_file_load_contents_async"  
  glib:finish-func="g_file_load_contents_finish"  
  glib:sync-func="g_file_load_contents">
```

```
// Opt-in per method no longer needed:  
// Gio._promisify(Gio._LocalFilePrototype, 'load_contents_async', 'load_contents_finish');  
const [contents] = await file.load_contents_async(/* cancel = */ null);
```

# Next JS engine upgrade (Firefox 91) ④2

**#private** class fields

```
class MyClass {  
  #value;  
  
  someMethod( ) {  
    doSomethingWith(this.#value);  
  }  
}
```

**Note:** Doesn't yet integrate with GObject classes.

# Next JS engine upgrade (Firefox 91) ④2

`at()`: Python-style indexing for [arrays](#), [strings](#), and [typed arrays](#)

```
const arr = [1, 2, 3, 4, 5, 6, 7];  
arr.at(0)    // 1  
arr.at(-1)   // 7  
arr.at(999)  // undefined  
'my string'.at(-7) // ''  
  
arr[arr.length - 1] // no longer necessary!
```

# Next JS engine upgrade (Firefox 91) ④2

`Promise.any()` : First successful sub-promise

```
const cancellable = new Gio.Cancellable();
const fastestServer = await Promise.any([
  checkServer(EAST, cancellable),
  checkServer(WEST, cancellable),
  waitSeconds(30),
]);
if (!fastestServer) {
  cancellable.cancel();
  notify('No server found within 30 seconds');
}
```

# Next JS engine upgrade (Firefox 91) ④2

`??=`, `&&=`, `||=` operators

Short-circuiting assignment operators

- `a ??= b` - assign `a = b` if `a` is `null` or `undefined`
- `a ||= b` - assign `a = b` if `a` is falsey
- `a &&= b` - assign `a = b` if `a` is truthy

**How can *you* help?**



# Help define best practices for the GNOME JS ecosystem

- [Sample app](#) has been updated
- For an experiment with even newer stuff, see the [GJS Bloatpad](#)
- [Writing native Linux desktop apps with JavaScript](#) at LAS 2021
- Talk to us in [#javascript:gnome.org](#) about things confusing you!

# The Big Hammer, why isn't it removed?

Well-defined problems are nearly all solved. Squishy problems remain.

We need:

- ④② better memory accounting
- to quantify acceptable memory usage for Shell, & tune GC accordingly.
- to figure out some solution that fits both Shell and apps
- a communication plan

# Thanks

GJS contributors from 40 and 41

# License

Presentation licensed under Creative Commons BY-NC-ND 4.0

A close-up photograph of a ginger cat with white markings on its face, resting its head on the keyboard of a silver laptop. The cat's eyes are partially closed, and its whiskers are visible. The laptop screen is open to the right, showing a blurred image. The word "Questions?" is overlaid in white text on the cat's face.

# Questions?