WebAssembly

Joanna Lamch

2019 marzec

Who am I ?

Joanna Lamch

- JLamch@gmail.com
- **JLamch.net**
- ProgramistkaIKot.pl

Microsoft fangirl

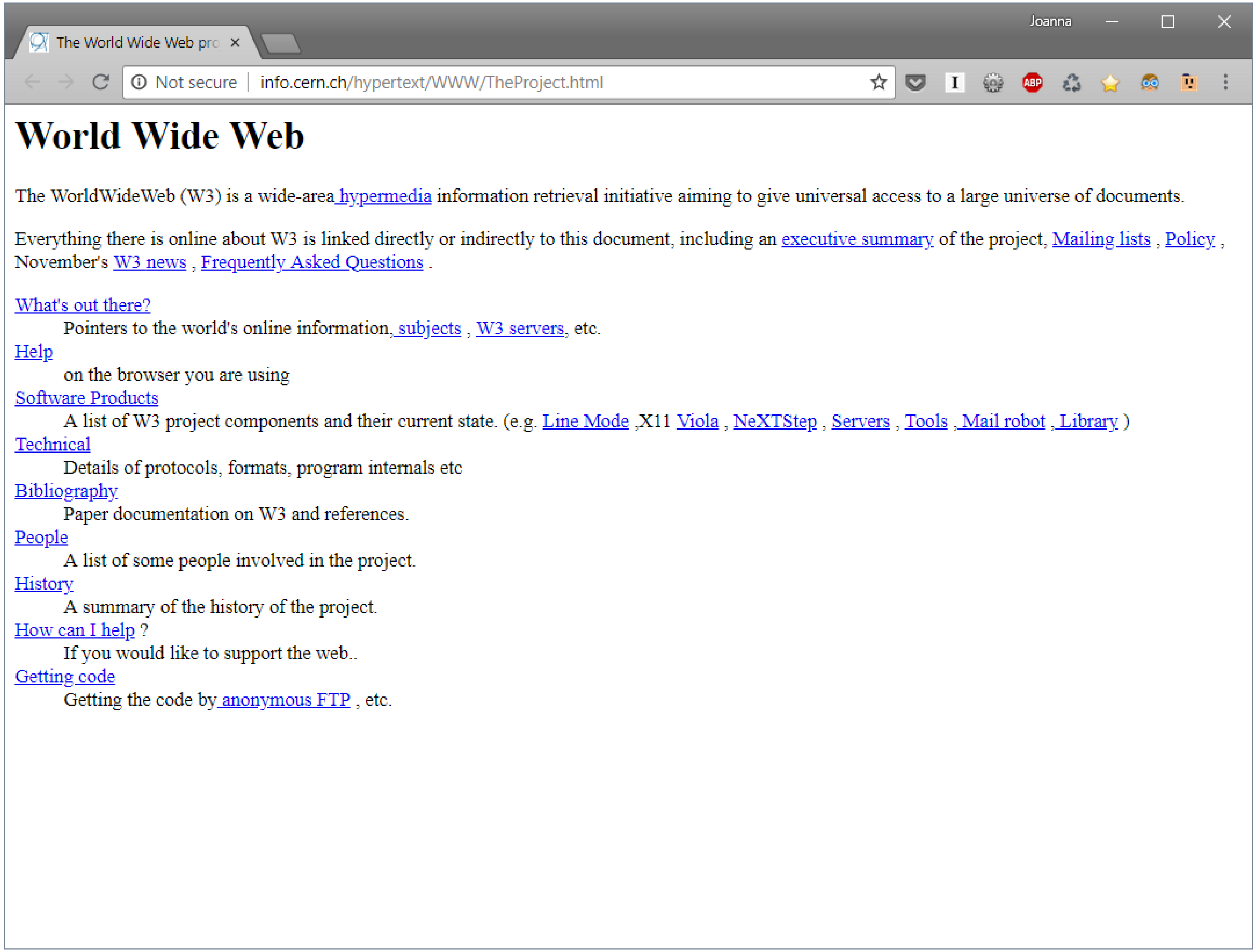
Developer C#

- .NET Framework 1.1
- 15 years (+ overtime)
- Xamarin
- SIENN

Community

- Śląska Grupa Microsoft
- Women In Technology
- Gruba.IT

Back in the days…


The World Wide Web pro

Not secure | info.cern.ch/hypertext/WWW/TheProject.html

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
: Pointers to the world's online information, subjects , W3 servers, etc.

Help
: on the browser you are using

Software Products
: A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical
: Details of protocols, formats, program internals etc

Bibliography
: Paper documentation on W3 and references.

People
: A list of some people involved in the project.

History
: A summary of the history of the project.

How can I help ?
: If you would like to support the web..

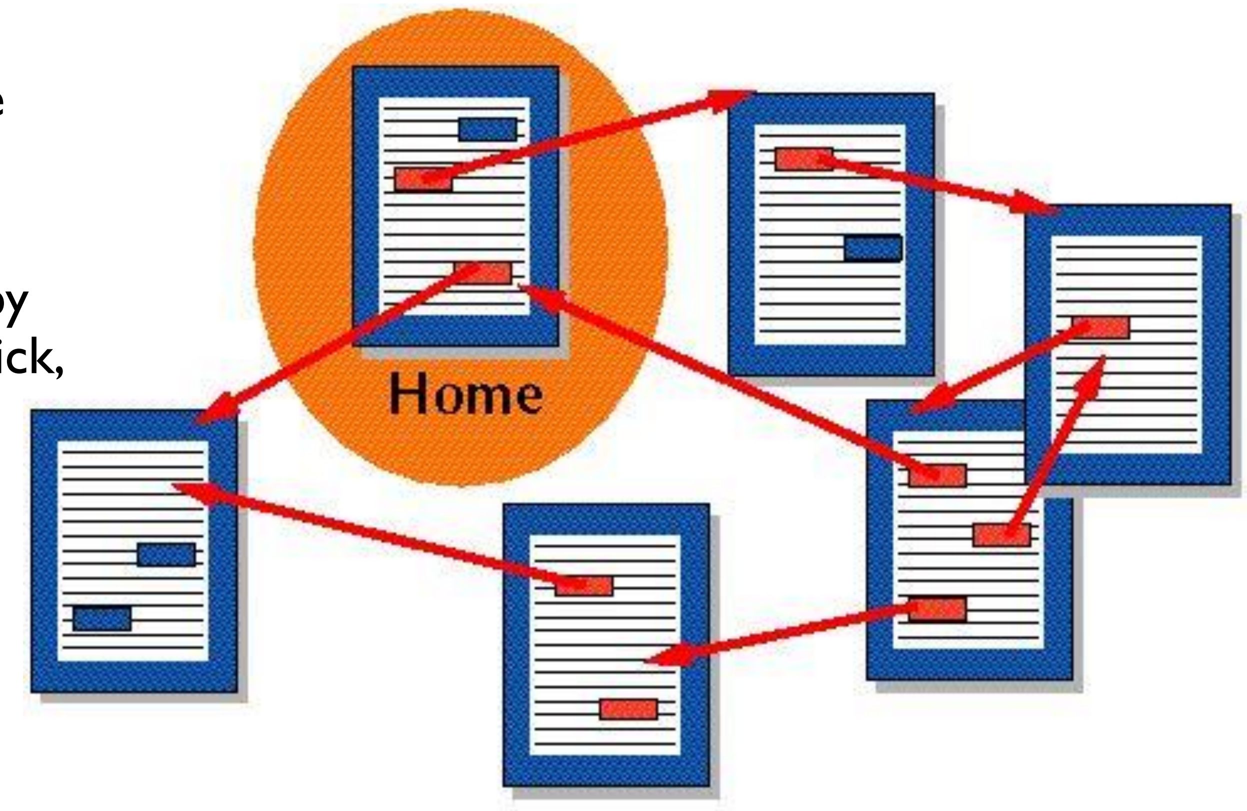
Getting code
: Getting the code by anonymous FTP , etc.

1991

Hypertext, HTML, HTTP

Text displayed on a electronic devices with references (hyperlinks) to other text that the reader can immediately access.

Hypertext documents are interconnected by hyperlinks, which are activated by a mouseClick, keypress set or by touching the screen.



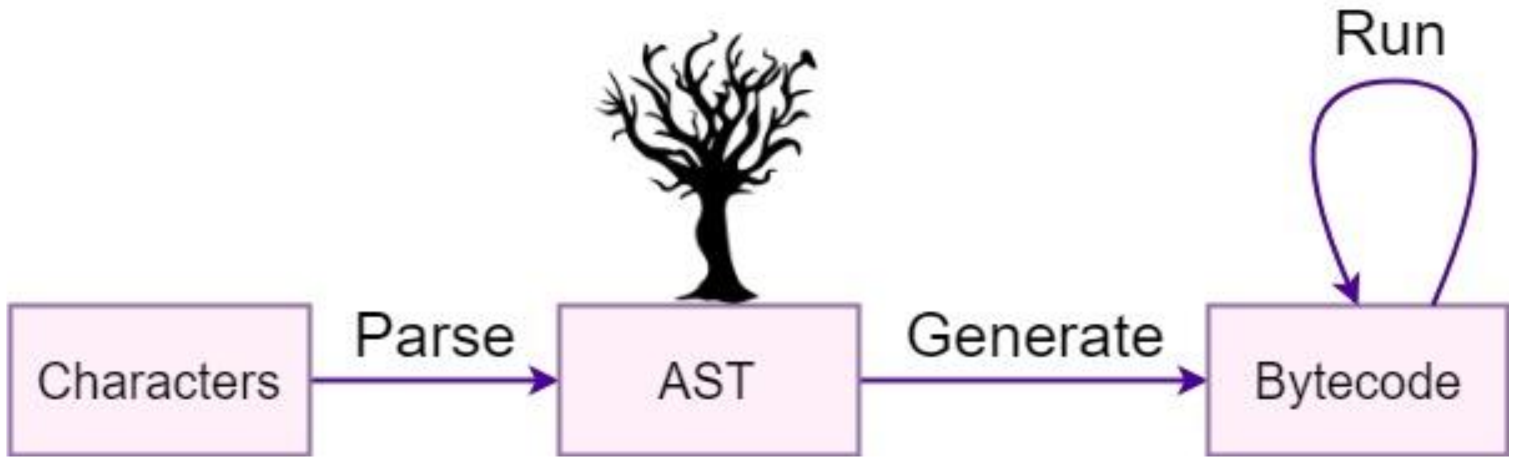
JavaScript

created in 10 days in May 1995, by Brendan Eich for Netscape

Adding interactivity to HTML pages

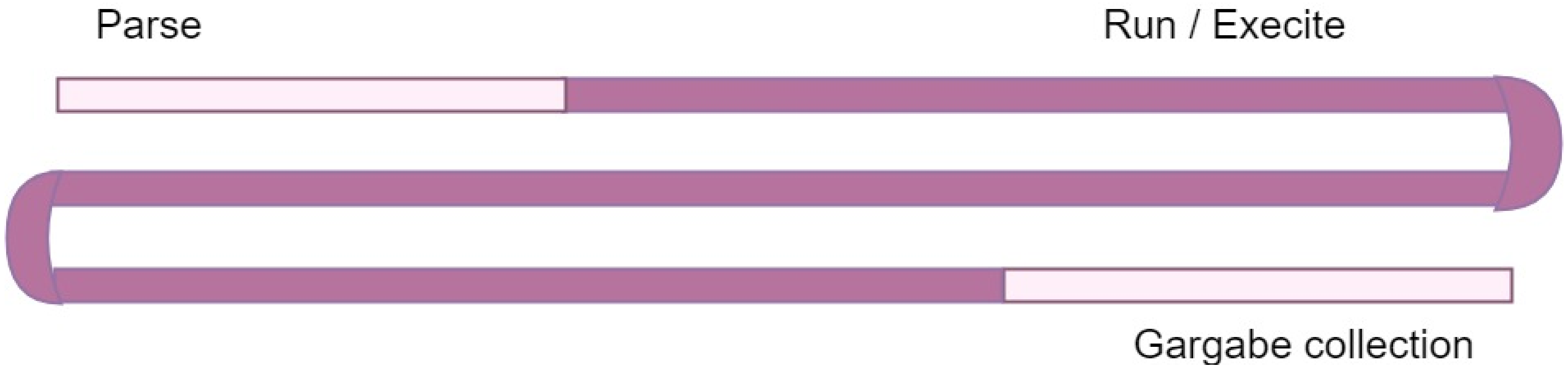
1995

JavaScript



1995

JavaScript



1995

https://hacks.mozilla.org/2017/02/a-cartoon-intro-to-webassembly/

Other plugins

Java Applets [1997]

ActiveX [1996]

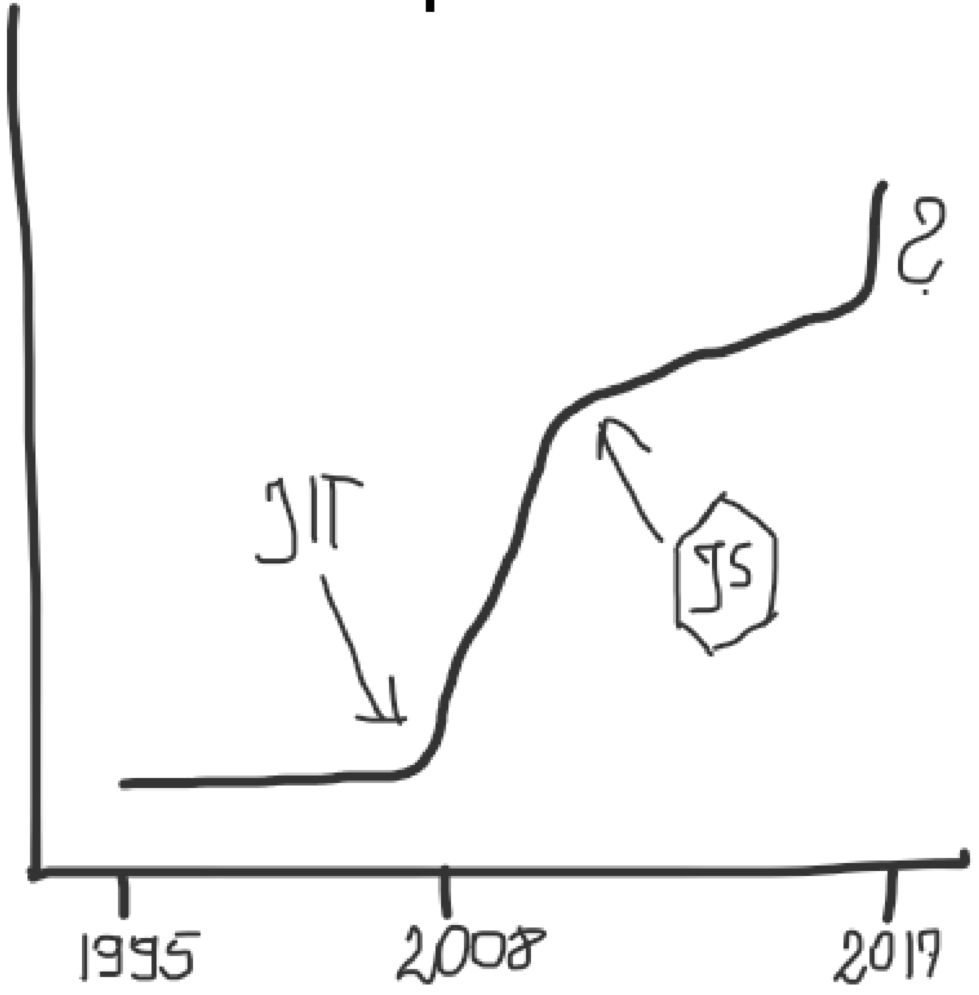
Flash [1996]

Silverlight [2007]

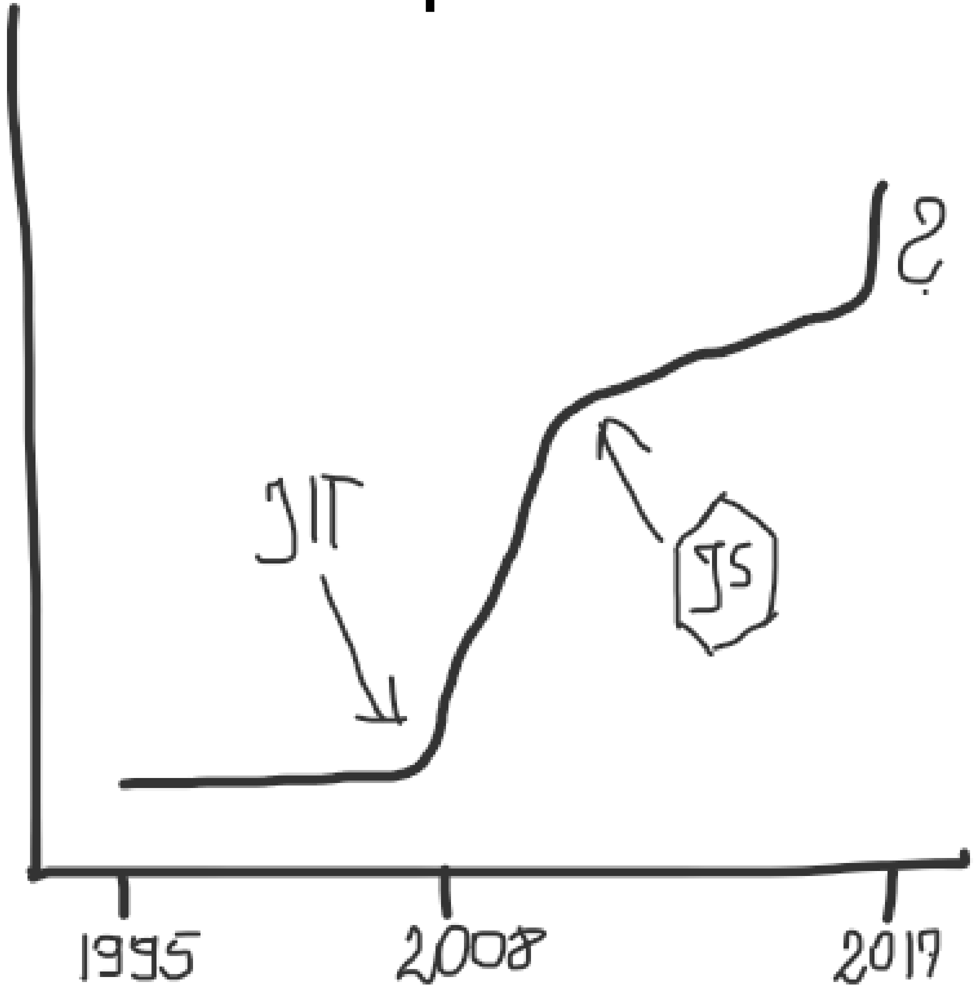
All of them are deprecated
or will be deprecated soon

1996+

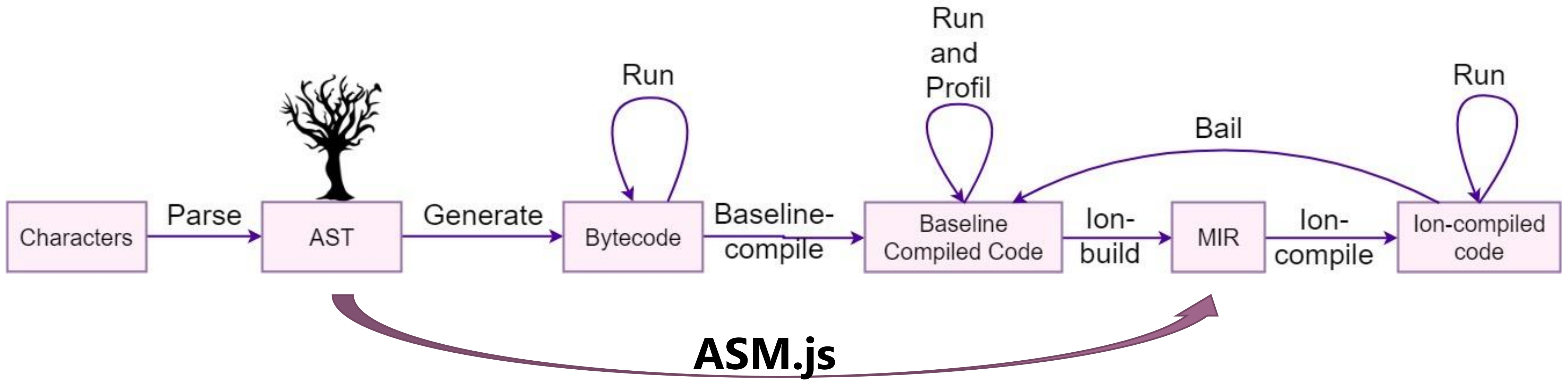
Browser performance wars (2008+)



Browser performance wars (2008+)

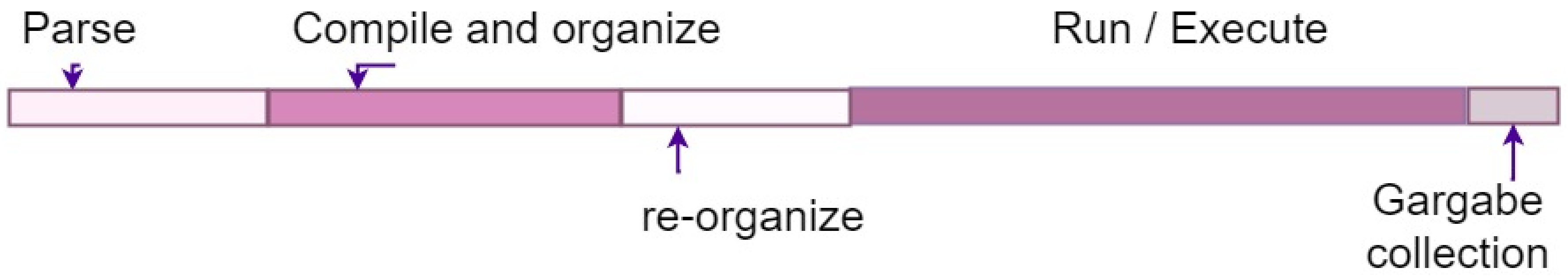


Browser performance wars (2008+)



https://blog.mozilla.org/luke/2014/01/14/asm-js-aot-compilation-and-startup-performance/

Browser performance wars (2008+)



Todays JavaScript

Different way the same language

there is a LOT of JavaScript

JavaScript is the language of the Web

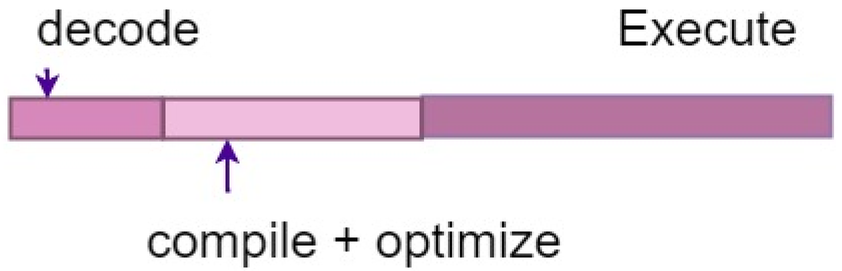
But it’s not very good Assembly Language
(still human readable simple language)

Todays JavaScript

WebAssembly What? Why?

A new low-level **binary** format **for the web** (WASM)

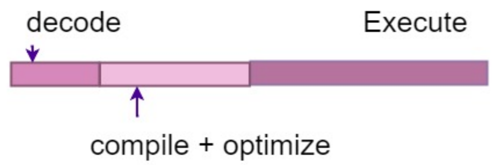
It's a bytecode for web / compilation target ➡ maximized performance



WebAssembly What? Why?

A new low-level **binary** format **for the web** (WASM)

It’s a bytecode for web / compilation target ➡ maximized performance



WebAssembly **is not designed to replace** JS, but to coexist

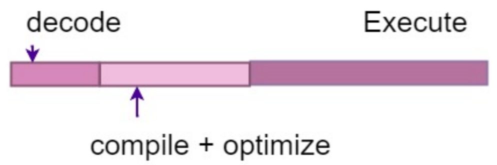
Sandboxed runtime in JS virtual machine

Security it runs locally in JS VM

WebAssembly What? Why?

A new low-level **binary** format **for the web** (WASM)

It’s a bytecode for web / compilation target ➡ maximized performance



WebAssembly **is not designed to replace** JS, but to coexist

Sandboxed runtime in JS virtual machine

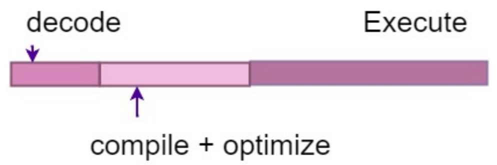
Security it runs locally in JS VM

Supported in **all** big browsers

WebAssembly What? Why?

A new low-level **binary** format **for the web** (WASM)

It's a bytecode for web / compilation target ➡ maximized performance



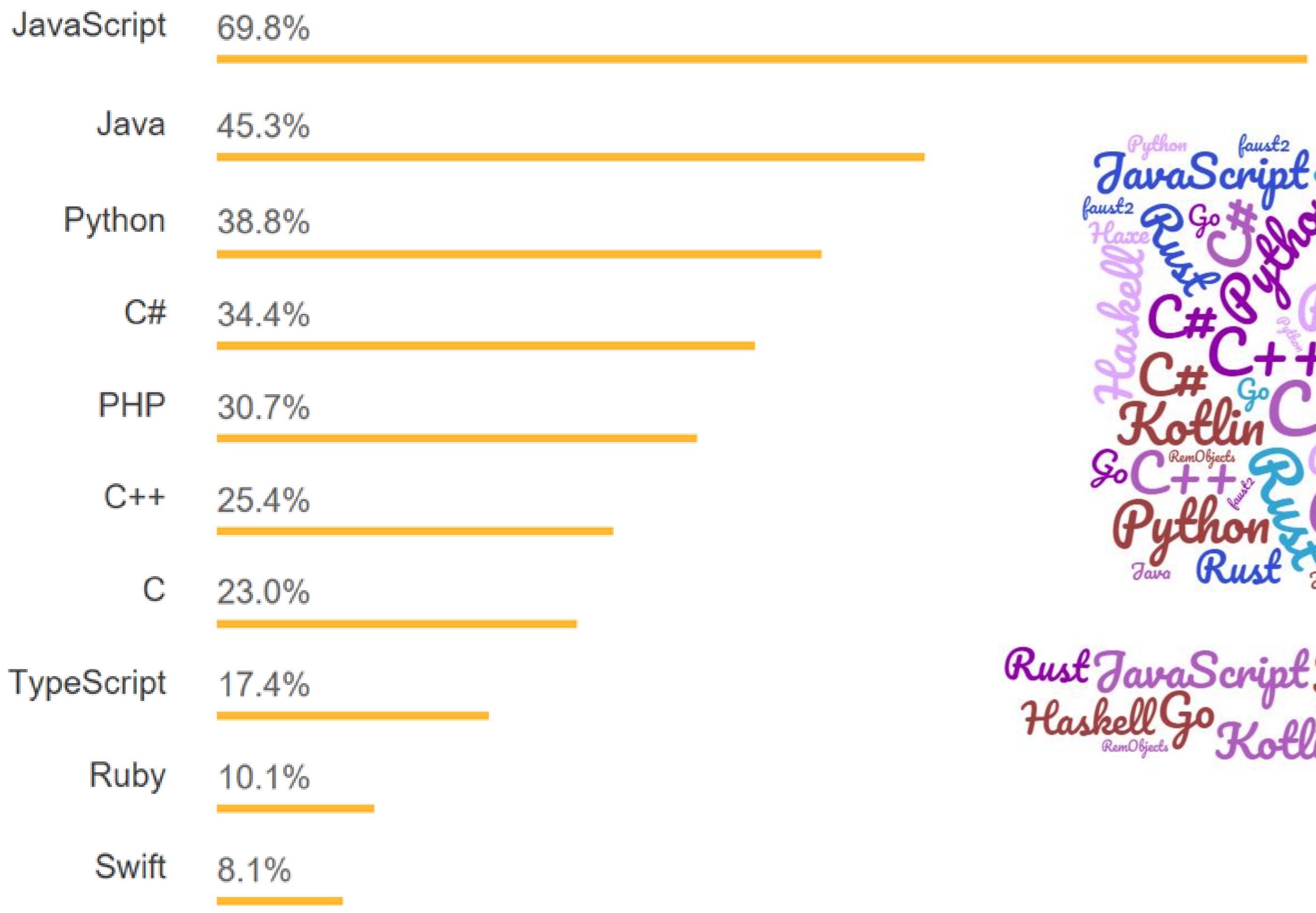
WebAssembly **is not designed to replace** JS, but to coexist

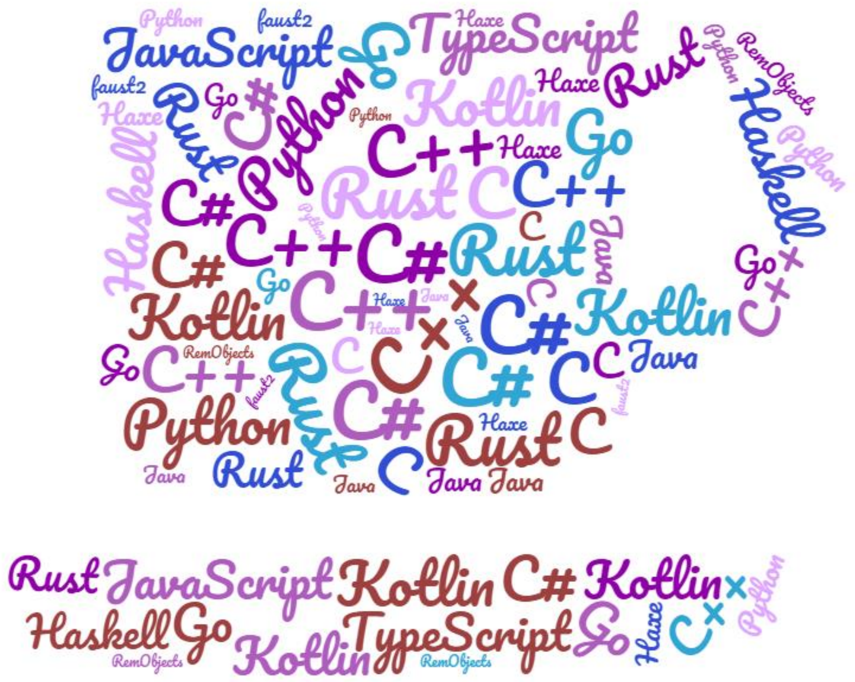
Sandboxed runtime in JS virtual machine
Security it runs locally in JS VM

Supported in **all** big browsers

Compiled from **other languages**

WASM opens possibilities for other languages





WebAssembly current state: MVP

- Compilation target

  Binary so compact

- Linear memory

- 4 types, 67 instructions, stack machine

- Fast execution

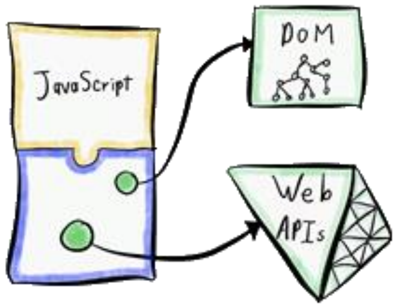
WebAssembly current state: MVP

- Compilation target

  Binary so compact

- 4 types, 67 instructions, stack machine

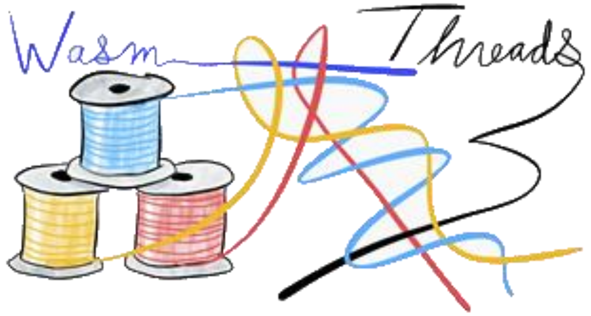
- No Web APIs, DOM access/manipulation



- Linear memory

- Fast execution
  or not so fast
  Not so fast load

- No Threads



WebAssembly How?

executes the **Emscripten** compiler

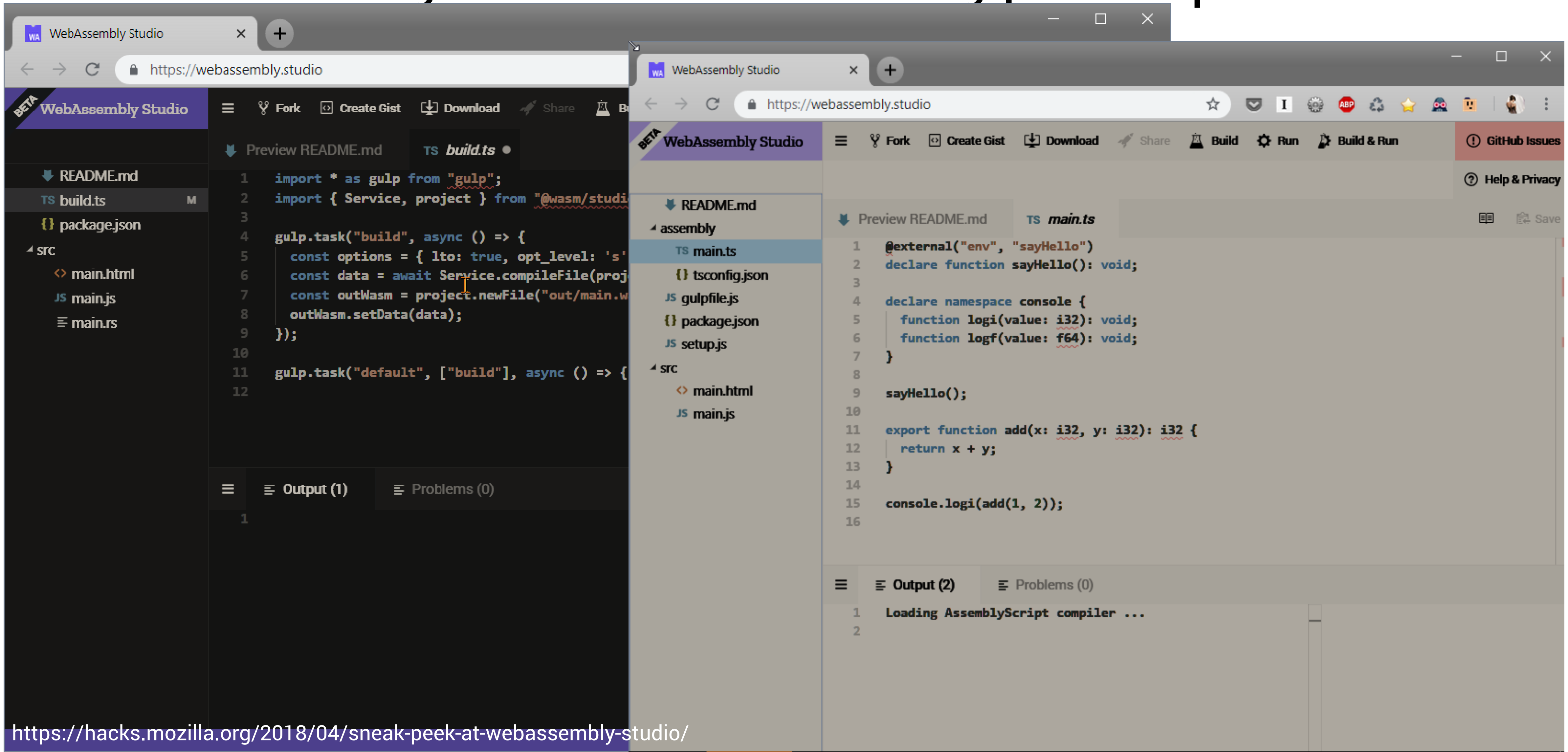
name our output file

```
$emcc -s WASM=1 -o example.js  example.cpp
```

input file

Emscripten outputs WebAssembly - switch

WebAssembly studio C/Rust/TypeScript



```
import * as gulp from "gulp";
import { Service, project } from "@wasm/studi

gulp.task("build", async () => {
  const options = { lto: true, opt_level: 's'
  const data = await Service.compileFile(proj
  const outWasm = project.newFile("out/main.w
  outWasm.setData(data);
});

gulp.task("default", ["build"], async () => {
```

```
@external("env", "sayHello")
declare function sayHello(): void;

declare namespace console {
  function logi(value: i32): void;
  function logf(value: f64): void;
}

sayHello();

export function add(x: i32, y: i32): i32 {
  return x + y;
}

console.logi(add(1, 2));
```

```
Loading AssemblyScript compiler ...
```

https://hacks.mozilla.org/2018/04/sneak-peek-at-webassembly-studio/

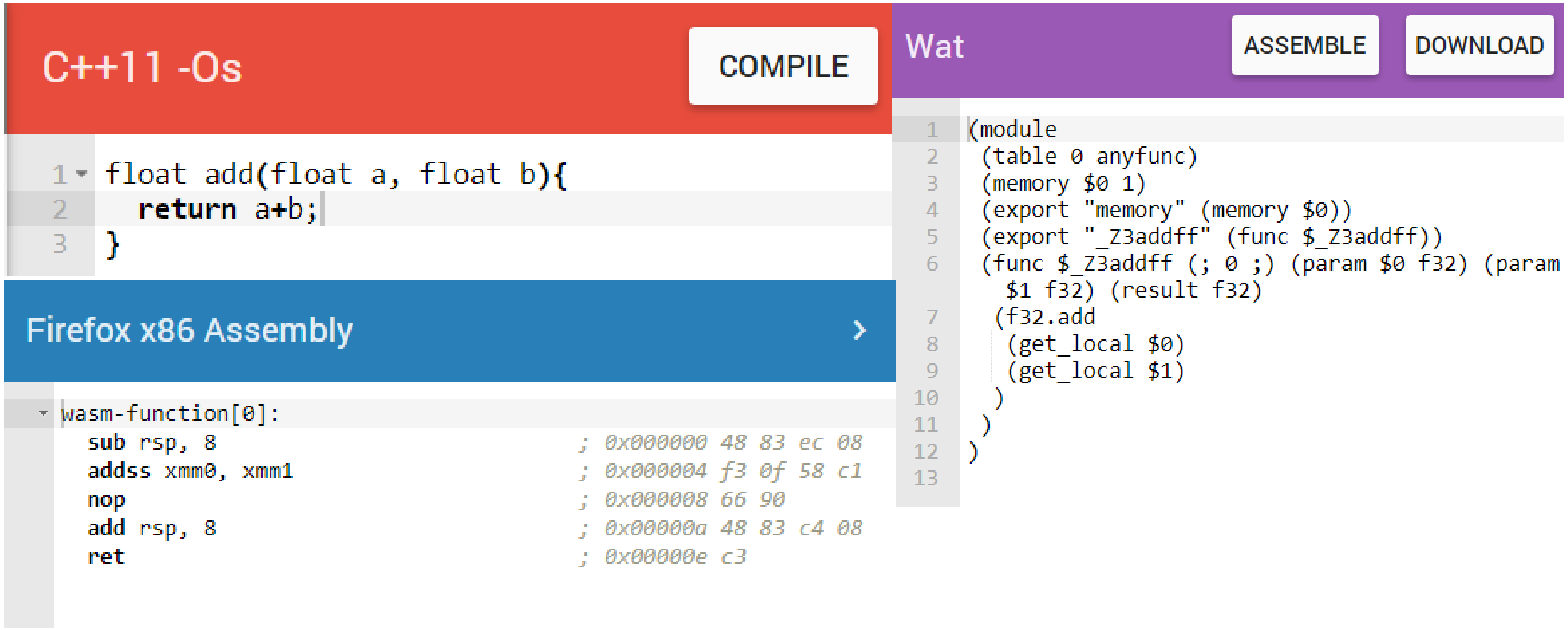
Demo

http://mbebenita.github.io/WasmExplorer/

https://wasdk.github.io/WasmFiddle/

https://webassembly.studio/

WasmExplorer

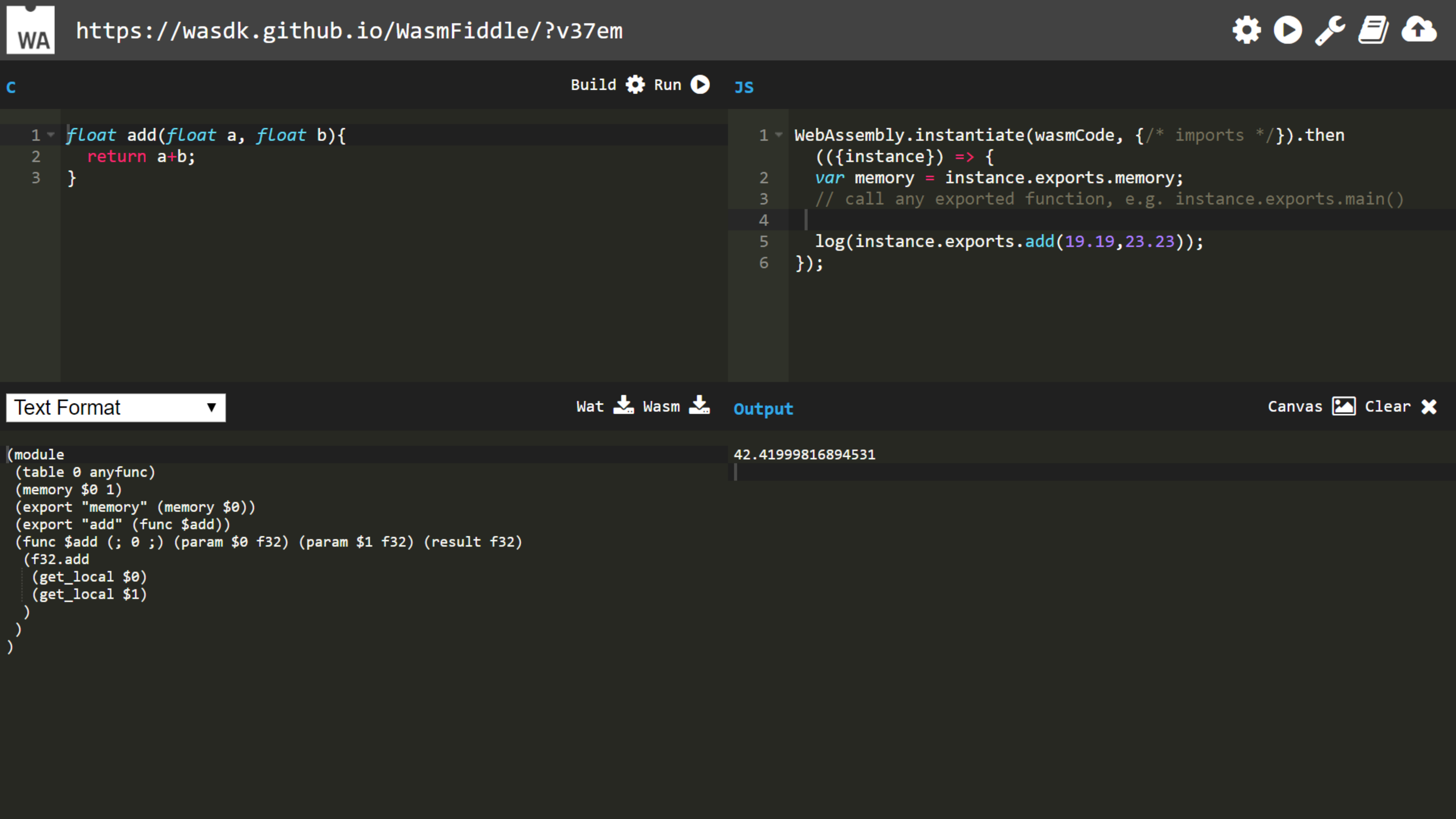


```
float add(float a, float b){
  return a+b;
}
```

```
wasm-function[0]:
  sub rsp, 8                        ; 0x000000 48 83 ec 08
  addss xmm0, xmm1                  ; 0x000004 f3 0f 58 c1
  nop                               ; 0x000008 66 90
  add rsp, 8                        ; 0x00000a 48 83 c4 08
  ret                               ; 0x00000e c3
```

```
(module
 (table 0 anyfunc)
 (memory $0 1)
 (export "memory" (memory $0))
 (export "_Z3addff" (func $_Z3addff))
 (func $_Z3addff (; 0 ;) (param $0 f32) (param
   $1 f32) (result f32)
  (f32.add
   (get_local $0)
   (get_local $1)
  )
 )
)
```

https://bit.ly/2JkqFYt

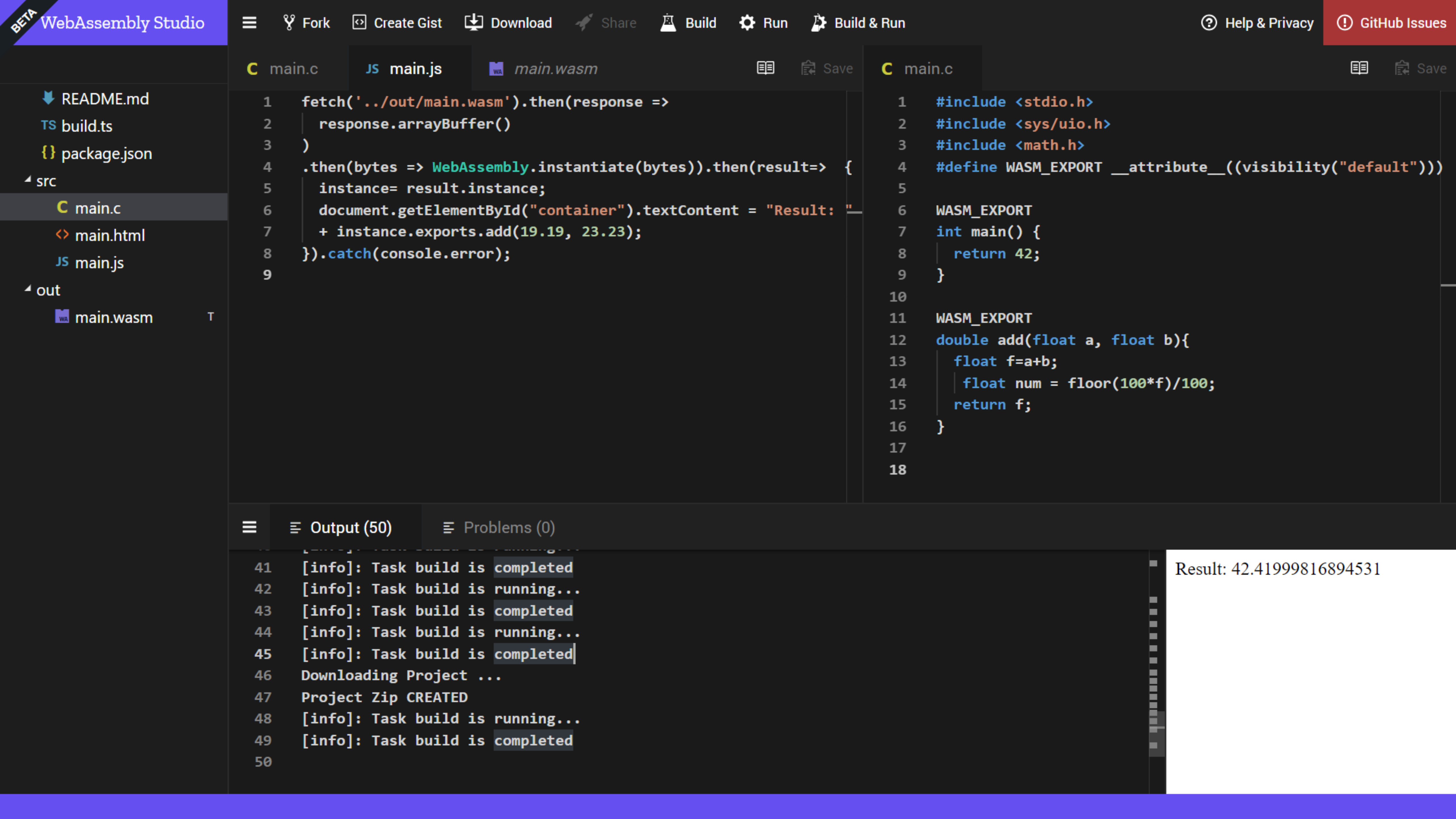


```c
float add(float a, float b){
  return a+b;
}
```

```js
WebAssembly.instantiate(wasmCode, {/* imports */}).then
  (({instance}) => {
  var memory = instance.exports.memory;
  // call any exported function, e.g. instance.exports.main()

  log(instance.exports.add(19.19,23.23));
});
```

```
(module
 (table 0 anyfunc)
 (memory $0 1)
 (export "memory" (memory $0))
 (export "add" (func $add))
 (func $add (; 0 ;) (param $0 f32) (param $1 f32) (result f32)
  (f32.add
   (get_local $0)
   (get_local $1)
  )
 )
)
```

```
42.41999816894531
```



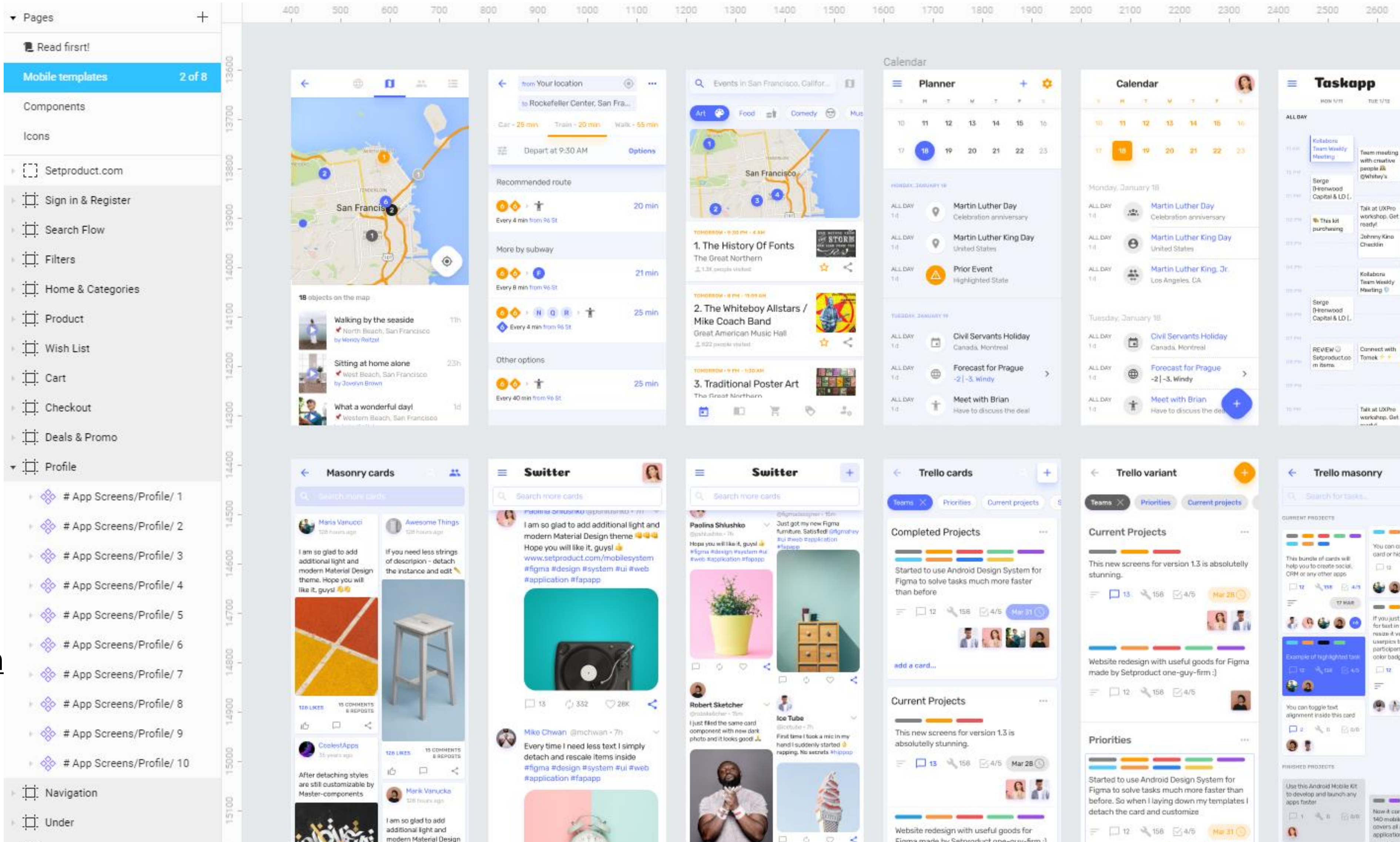
```javascript
fetch('../out/main.wasm').then(response =>
  response.arrayBuffer()
)
.then(bytes => WebAssembly.instantiate(bytes)).then(result=>  {
  instance= result.instance;
  document.getElementById("container").textContent = "Result: "
  + instance.exports.add(19.19, 23.23);
}).catch(console.error);
```

```c
#include <stdio.h>
#include <sys/uio.h>
#include <math.h>
#define WASM_EXPORT __attribute__((visibility("default")))

WASM_EXPORT
int main() {
  return 42;
}

WASM_EXPORT
double add(float a, float b){
  float f=a+b;
   float num = floor(100*f)/100;
  return f;
}
```

```
[info]: Task build is completed
[info]: Task build is running...
[info]: Task build is completed
[info]: Task build is running...
[info]: Task build is completed
Downloading Project ...
Project Zip CREATED
[info]: Task build is running...
[info]: Task build is completed
```

Result: 42.41999816894531

WASM in real world

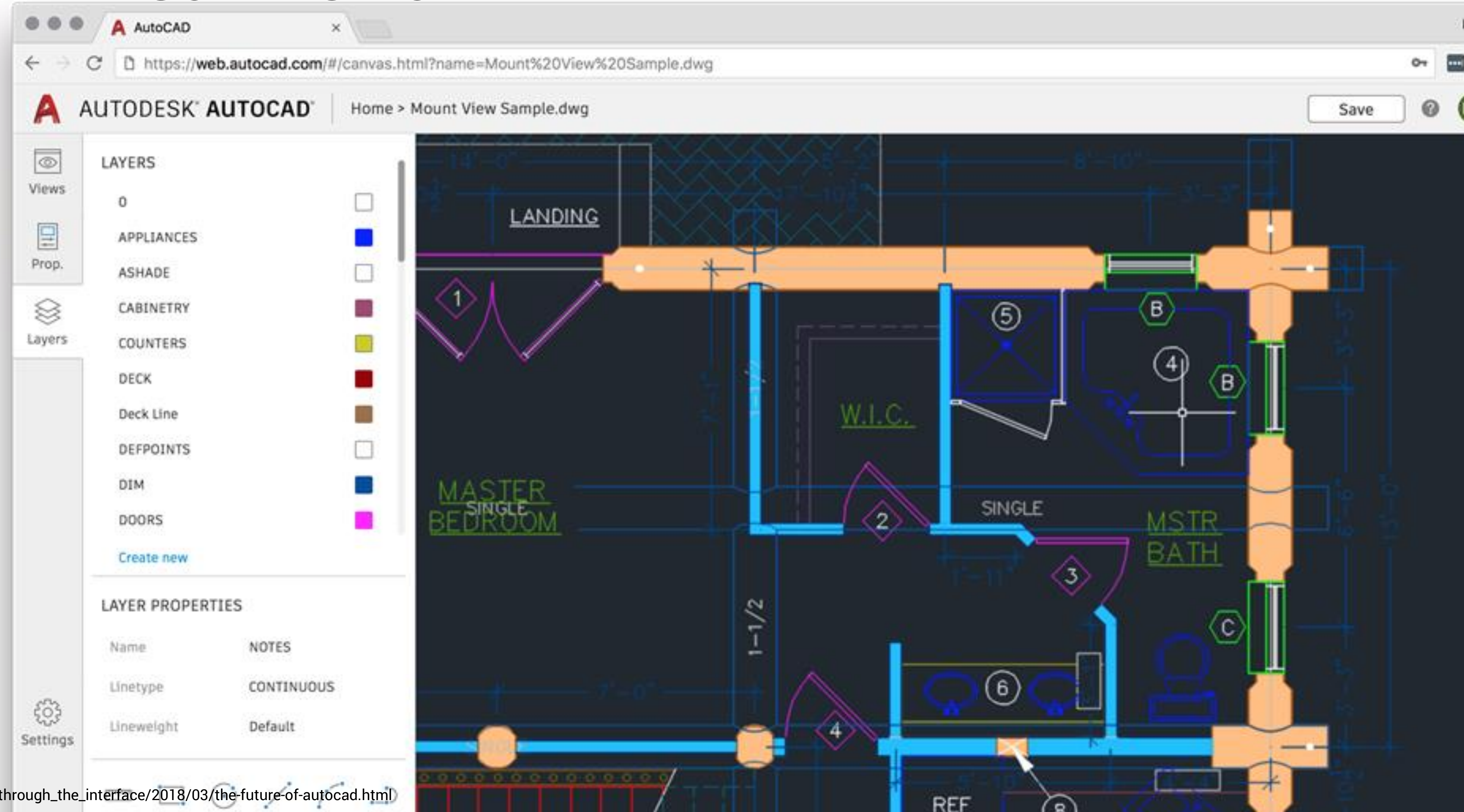


Figma

https://www.figma.com

[https://www.youtube.com watch?v=Kf1gILChfks](https://www.youtube.com/watch?v=Kf1gILChfks)

WASM in real world



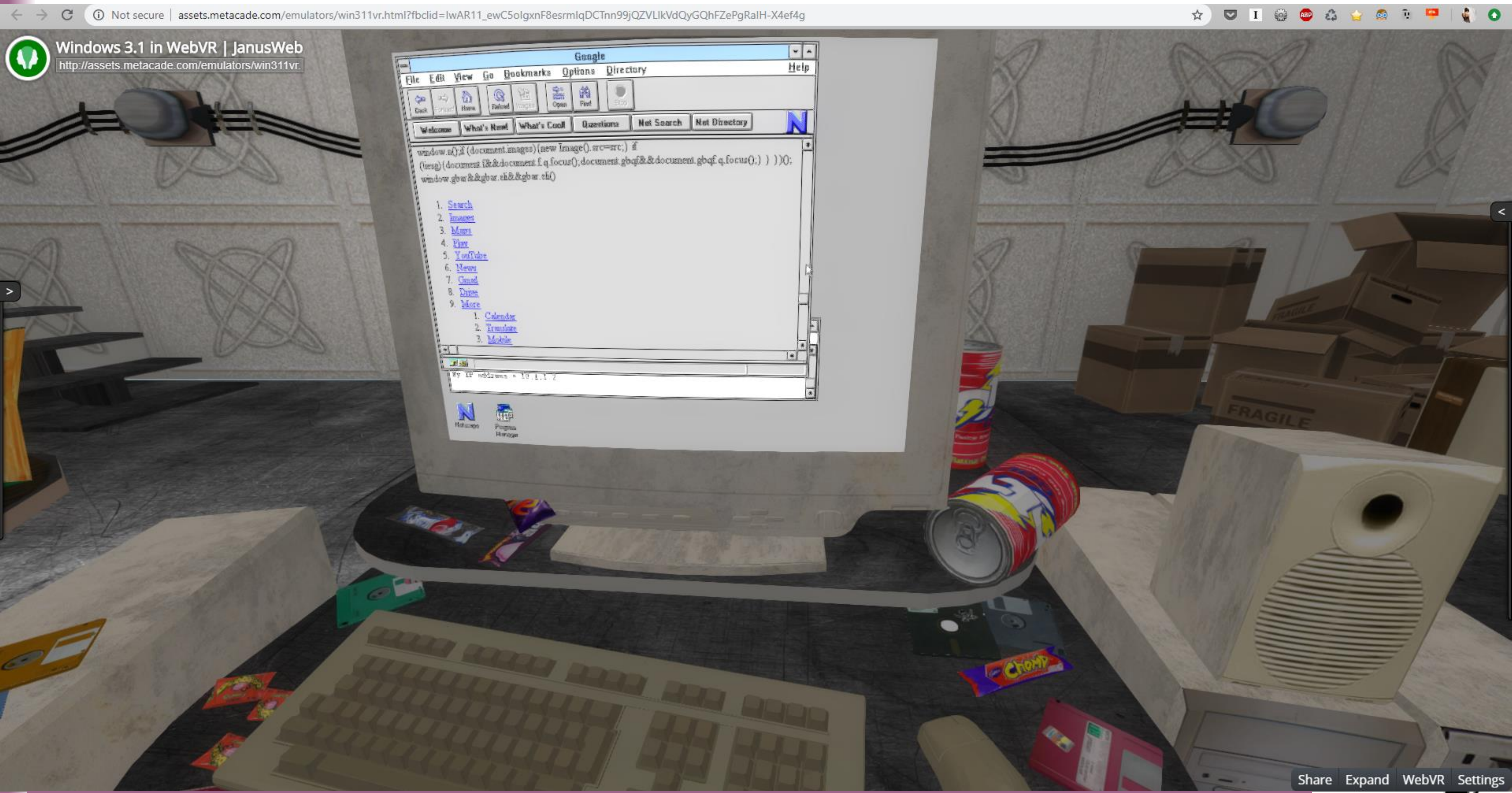
LAYERS

0	⬜
APPLIANCES	🟦
ASHADE	⬜
CABINETRY	🟪
COUNTERS	🟨
DECK	🟥
Deck Line	🟫
DEFPOINTS	⬜
DIM	🟦
DOORS	🟪

LAYER PROPERTIES

Name	NOTES
Linetype	CONTINUOUS
Lineweight	Default

http://through-the-interface.typepad.com/through_the_interface/2018/03/the-future-of-autocad.html


Not secure | assets.metacade.com/emulators/win311vr.html?fbclid=IwAR11_ewC5olgxnF8esrmlqDCTnn99jQZVLlkVdQyGQhFZePgRalH-X4ef4g
Windows 3.1 in WebVR | JanusWeb
http://assets.metacade.com/emulators/win311vr.
Google
File Edit View Go Bookmarks Options Directory Help
Welcome What's New! What's Cool! Questions Net Search Net Directory
Netscape
Program Manager
FRAGILE
Chomp
Share Expand WebVR Settings

```
window.n();if (document.images) (new Image().src=src;) if
(iesg) (document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();) } })();
window.gbar&&gbar.eli&&gbar.eli()
```

1. Search
2. Images
3. Maps
4. Play
5. YouTube
6. News
7. Gmail
8. Drive
9. More
   1. Calendar
   2. Translate
   3. Mobile

```
My IP address = 10.1.1.2
```

http://assets.metacade.com/emulators/win311vr.html?fbclid=IwAR11_ewC5olgxnF8esrmlqDCTnn99jQZVLlkVdQyGQhFZePgRalH-X4ef4g

WASM in real world

Figma

https://www.figma.com/

https://www.youtube.com/watch?v=Kf1glLChfks

AUTODESK® AUTOCAD®

D3WASM

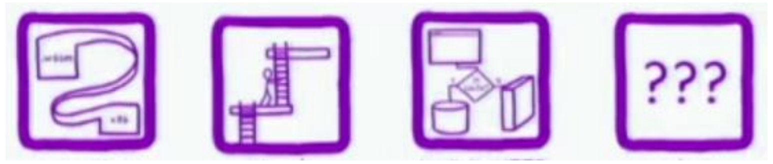
https://s3.amazonaws.com/mozilla-games/ZenGarden/EpicZenGarden.html?fbclid=IwAR0_uAensGfTj1Mzp4wXgVoxZjquxRo_uu2YD8yDuleTpPohaXyilDd82X8
https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/win2k.cfg&mem=192&graphic=1&w=1024&h=768
https://aesalazar.github.io/AsteroidsWasm/
http://sqliteefcore-wasm.platform.uno/
https://raytracer-mono-aot.platform.uno/
http://www.continuation-labs.com/projects/d3wasm/?fbclid=IwAR2V9OqEDgu3bu-vMNlxcZCUOm0HQAlv6ys-jcZGSMQY56saD8FYrHdVx_s

http://assets.metacade.com/emulators/win311vr.html?fbclid=IwAR11_ewC5oIgxnF8esrmlqDCTnn99jQZVLIkVdQyGQhFZePgRaIH-X4ef4g

What WebAssembly are next to?

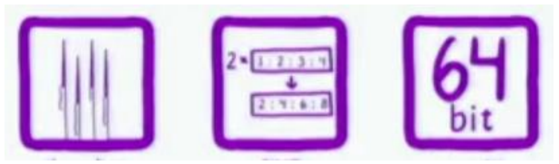
Loadtime improvements

- streaming compilation
- tiered compilation
- implicit HTTP caching
- other improvements



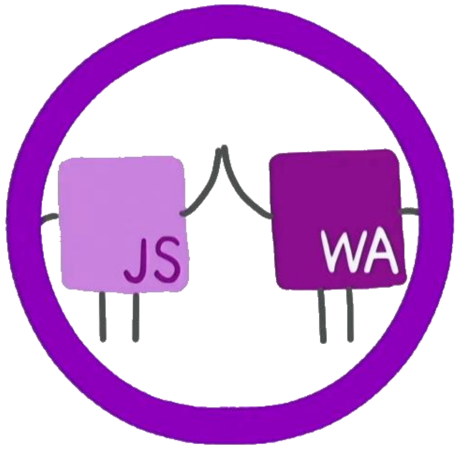
Making use of modern Hardware

- threading
- SIMD
- wasm 64bit

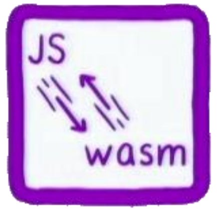


https://github.com/WebAssembly/design/blob/master/FutureFeatures.md

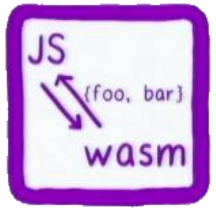
What WebAssembly are next to?



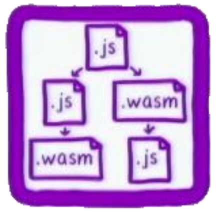
- fast calls
  JS <~> wasm



- esy and fast
  data exchange



- ES module
  integration



- toolchain integration
  like npm or webpack

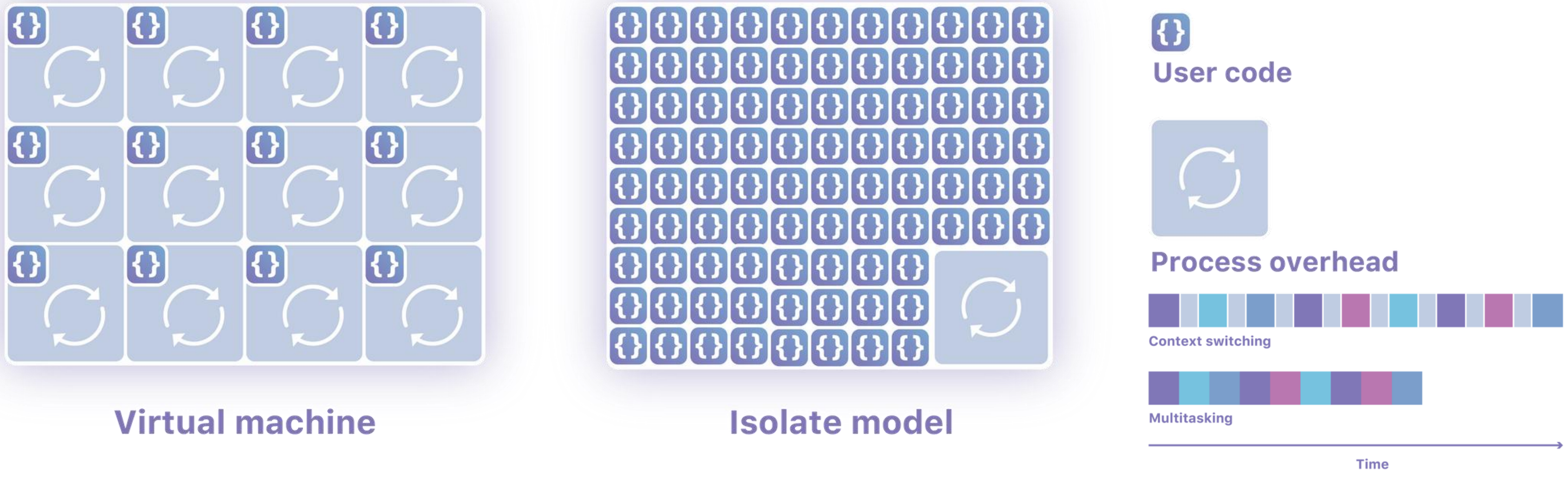
- backward compatibility

What WebAssembly are next to?

Hight level language features

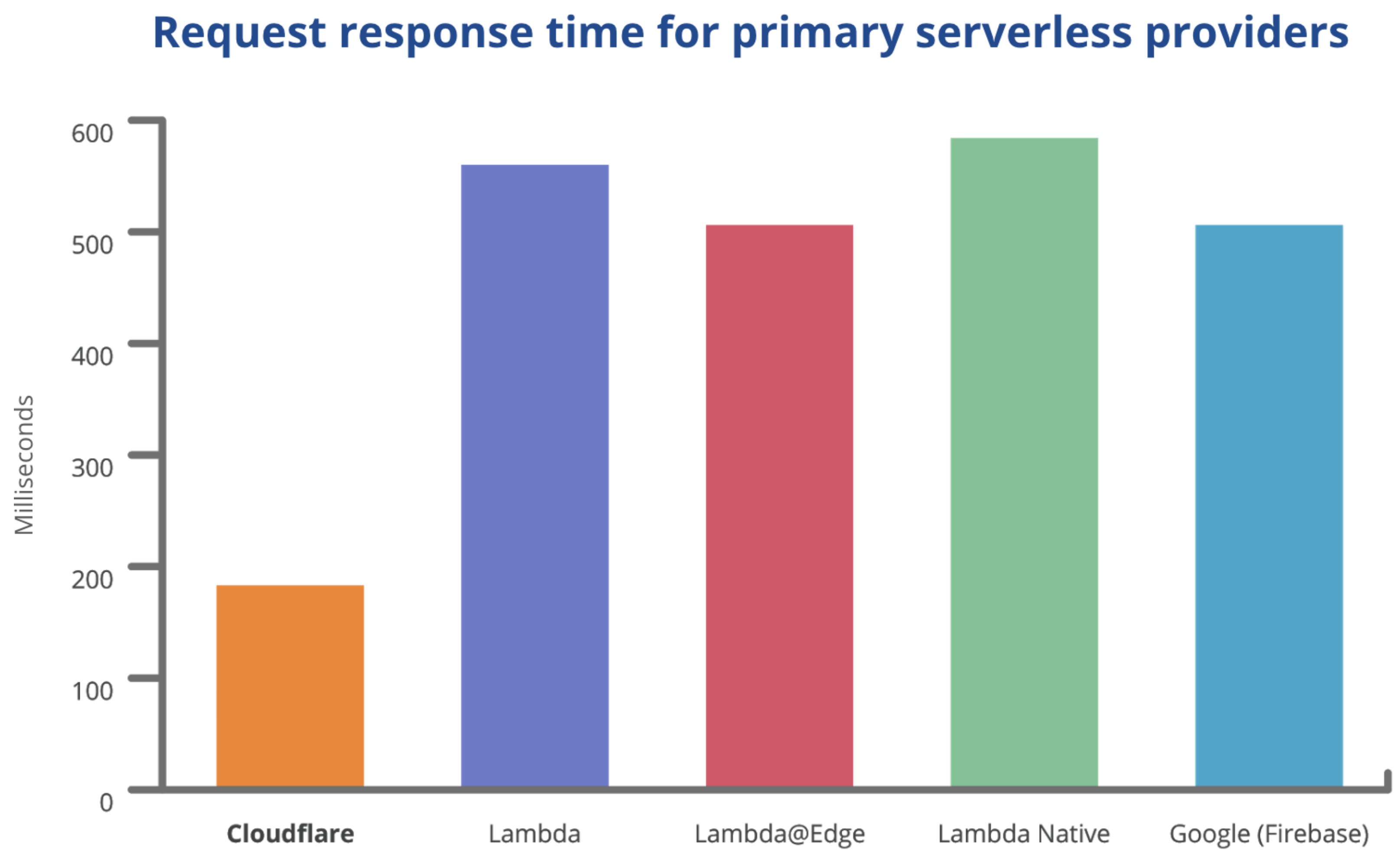
- GC integration
- Exception handling
- Debugging
- Tail cals

CloudFlare



https://blog.cloudflare.com/cloud-computing-without-containers/
https://www.youtube.com/watch?v=A9SydP1CcZU

CloudFlare no more cold starts?



WebAssembly

Questions?

Joanna Lamch

JLamch@gmail.com

JLamch.net

Ankieta

Działanie pociąga za sobą koszty i ryzyko, ale o wiele mniejsze niż te, które wiążą się z wygodną bezczynnością

Będzie mi miło móc ulepszyć moją prezentację dzięki Twoim komentarzom, dlatego proszę Cię o wypełnienie ankiety, bądź kontakt mailowy.

Bibliografia

https://webassembly.org/docs/high-level-goals/

https://www.smashingmagazine.com/2017/05/abridged-cartoon-introduction-webassembly/

https://www.youtube.com/watch?v=HktWin_LPf4&feature=youtu.be

https://www.youtube.com/watch?v=pBYqen3B2gc

https://www.youtube.com/watch?v=BnYq7JapeDA
https://www.youtube.com/watch?v=kS29TT4wk44&feature=youtu.be

https://github.com/mbasso/awesome-wasm

https://github.com/migueldeicaza/mono-wasm

https://superkotlin.com/kotlin-and-webassembly/

https://medium.com/@mumarov/how-to-get-started-with-kotlin-native-and-web-assembly-baa2813f0d9

https://github.com/DenisKolodin/yew

https://www.mergeconflict.fm/89

https://dotnetrocks.com/?show=1539

https://dotnetrocks.com/?show=1540

https://dotnetrocks.com/?show=1537

https://www.hanselman.com/blog/NETAndWebAssemblyIsThisTheFutureOfTheFrontend.aspx
https://hacks.mozilla.org/2018/04/sneak-peek-at-webassembly-studio/

https://github.com/migueldeicaza/mono-wasm?WT.mc_id=-blog-scottha

https://blog.scottlogic.com/ceberhardt/

https://blog.logrocket.com/working-with-the-blazor-javascript-interop-3c2a8d0eb56c
https://s3.amazonaws.com/mozilla-games/ZenGarden/EpicZenGarden.html
https://blog.logrocket.com/working-with-the-blazor-javascript-interop-3c2a8d0eb56c