WebAssembly and .NET

Joanna Lamch

2019

- ***Who am I ?***
- Joanna Lamch

  JLamch@gmail.com

  JLamch.net

  ProgramistkaIKot.pl
- Microsoft fangirl
- Developer C#

  .NET Framework 1.1

  15 years (+ overtime)

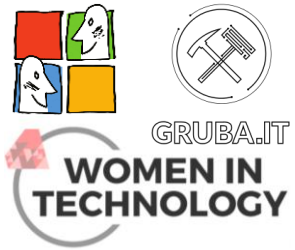
  Xamarin

  SIENN
- Community

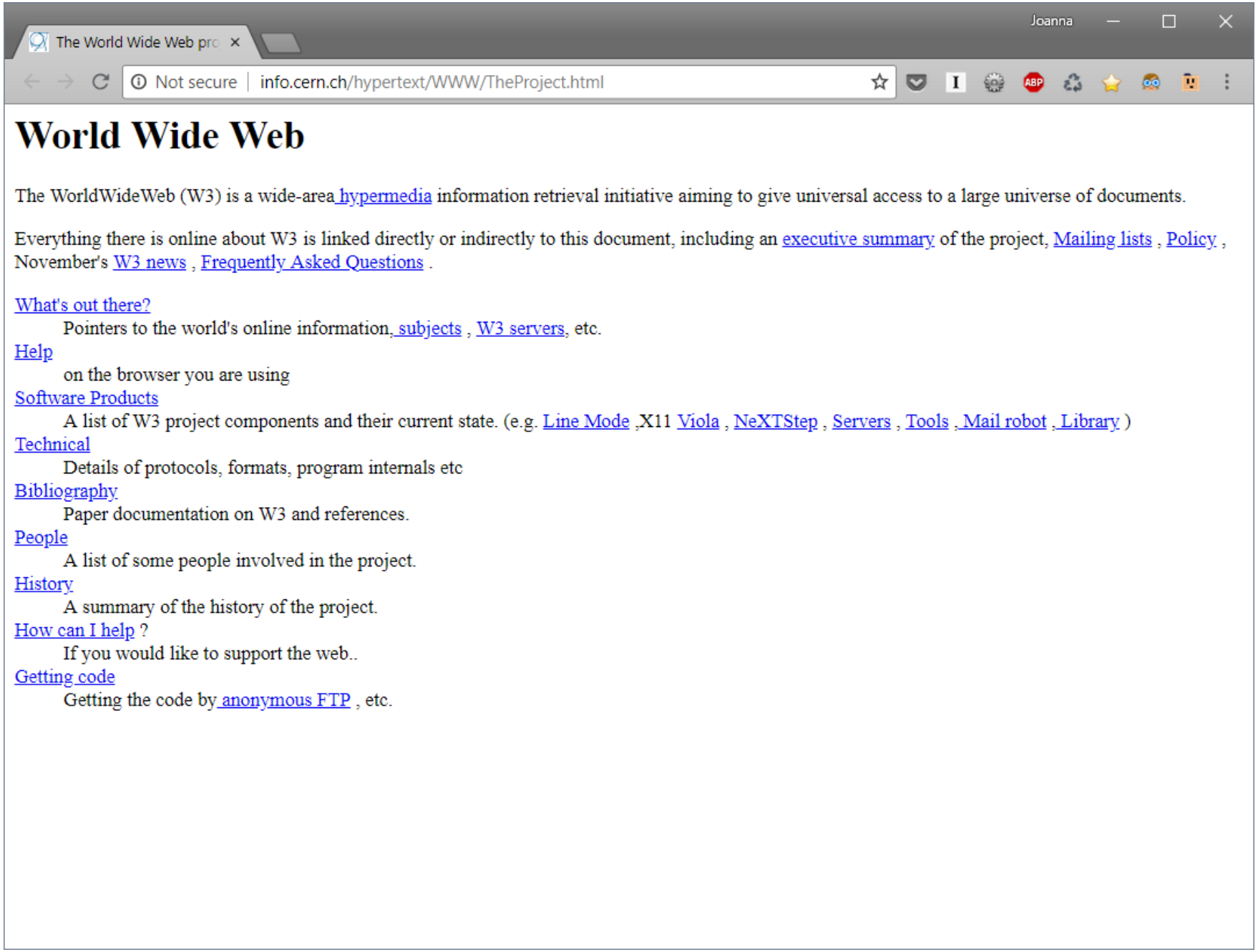
  Śląska Grupa Microsoft

  Women In Technology

  Gruba.IT



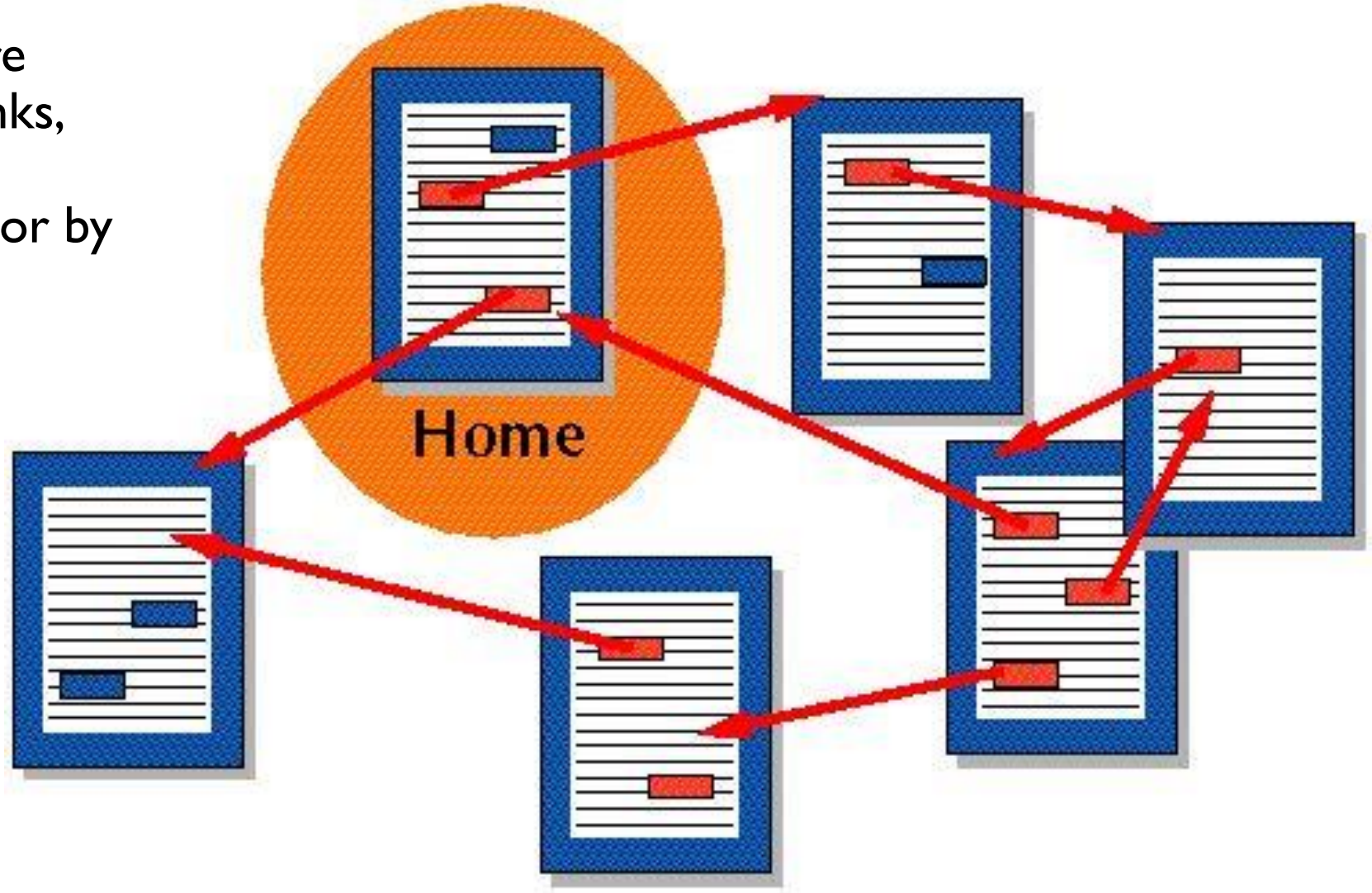
Back in the days...


The World Wide Web pro
Joanna
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

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



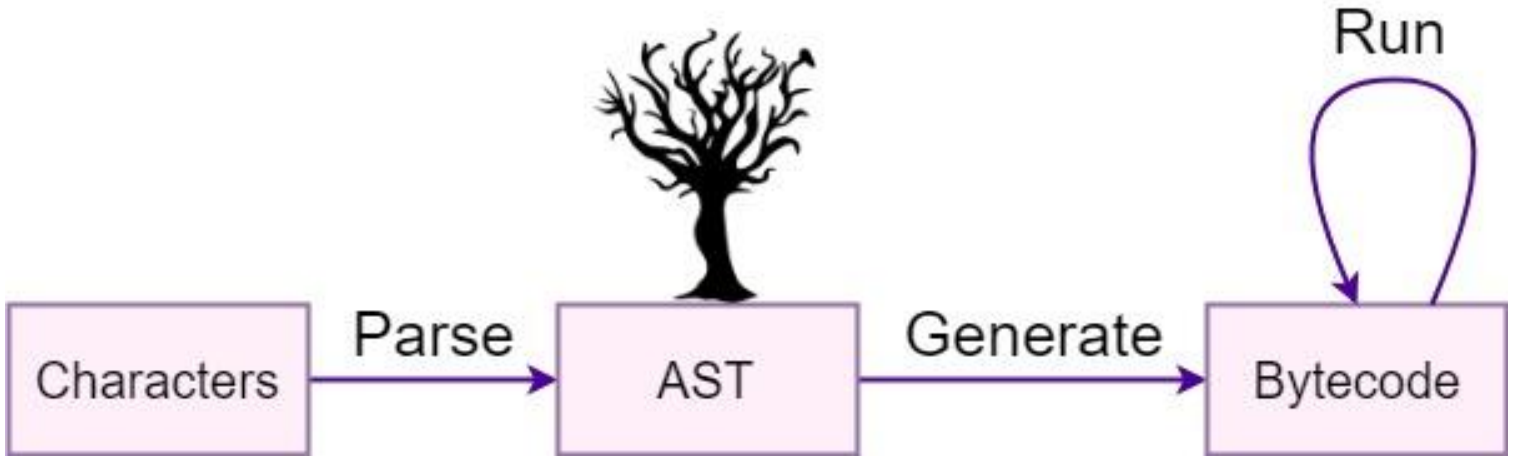
JavaScript

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

Adding interactivity to HTML pages

1995

JavaScript



1995

JavaScript

1995

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

Other plugins

Java Applets [1997]

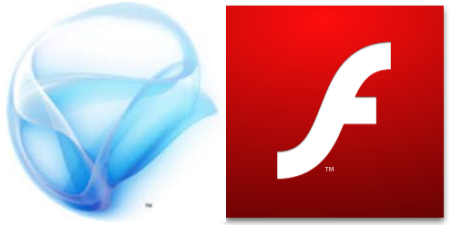
ActiveX [1996]

Flash [1996]

Silverlight [2007]

1996+

Other plugins



Java Applets [1997]

ActiveX [1996]

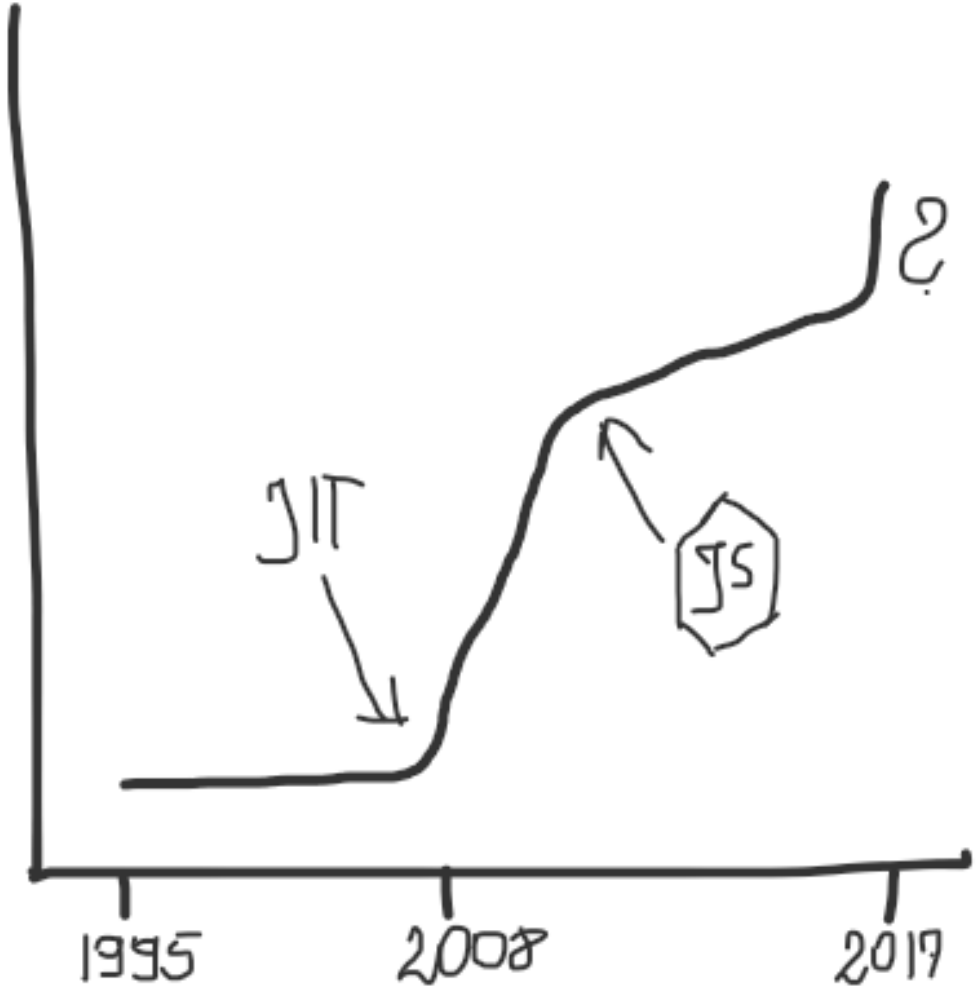
Flash [1996]

Silverlight [2007]

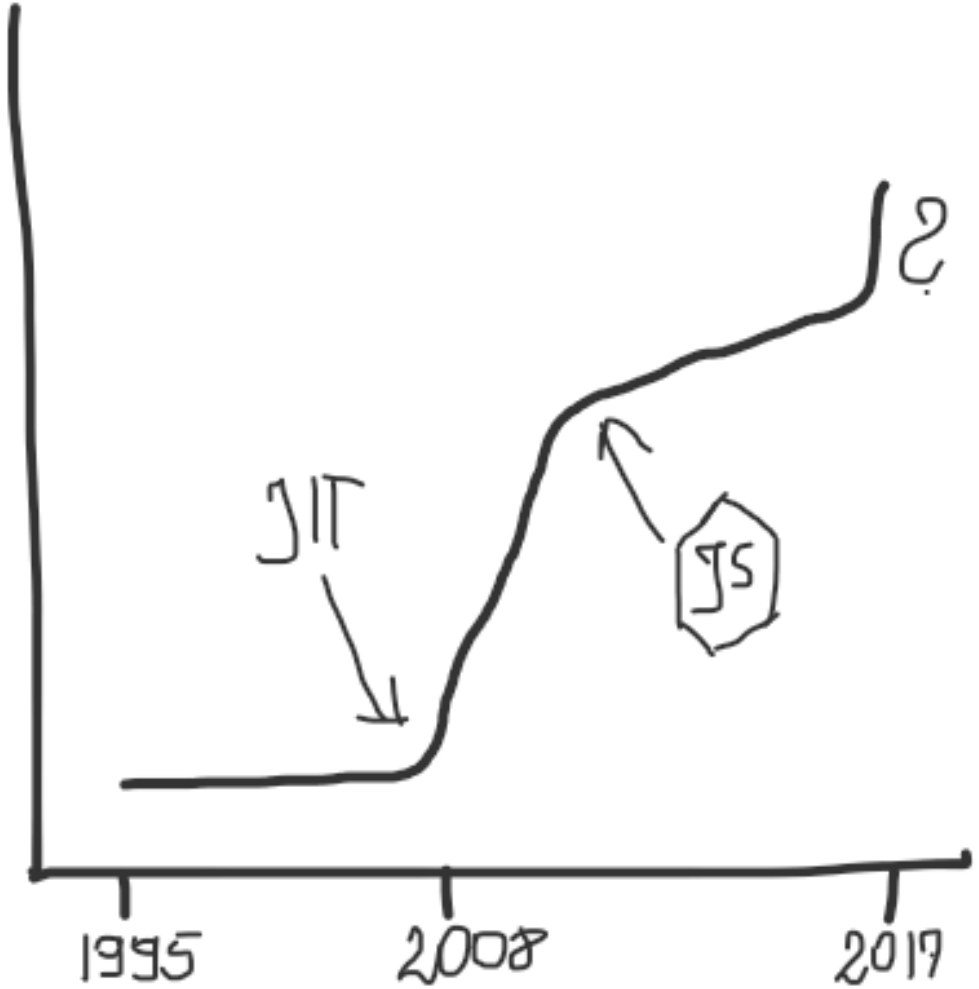
All of them are deprecated
or will be deprecated soon

1996+

Browser performance wars (2008+)

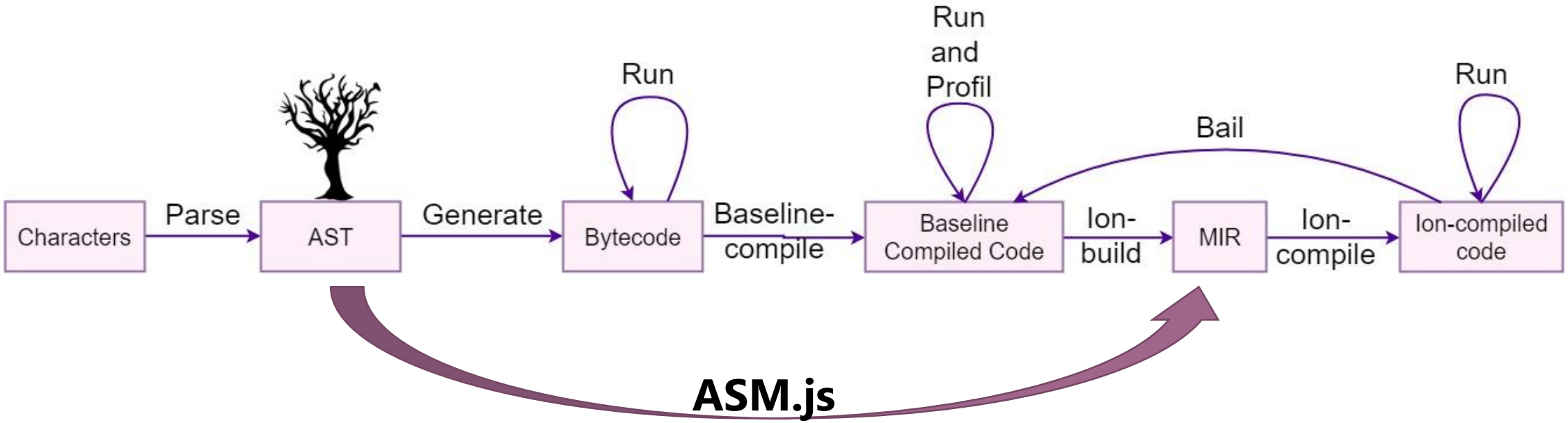


Browser performance wars (2008+)



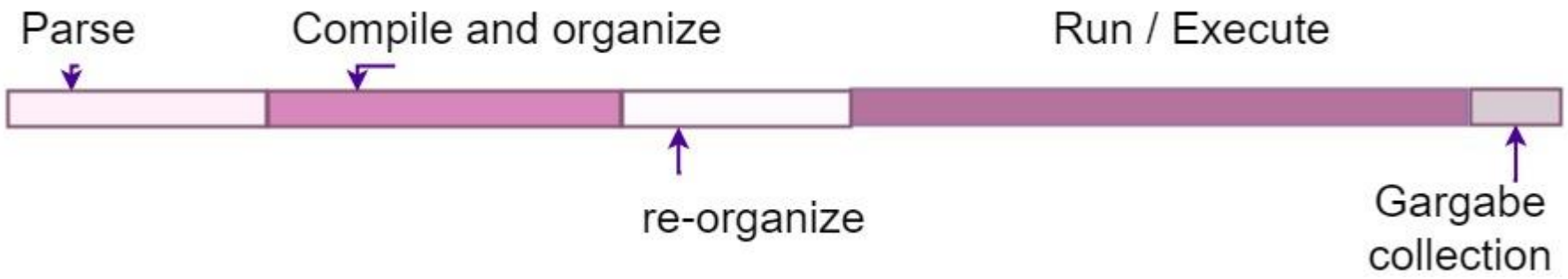
moz://a

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

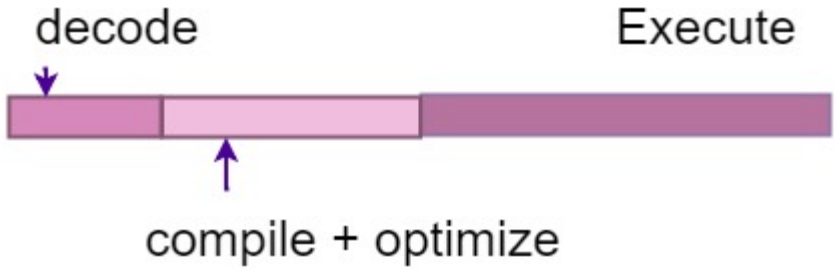
But it’s not very good Assembly Language
(still human readable simple language)

Todays JavaScript

WebAssembly What? Why?

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

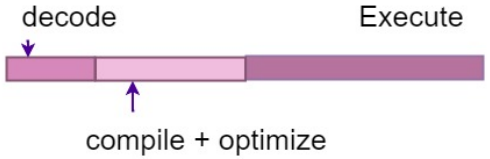
It's a bytecode for web / compilation target ➡ maximized performance



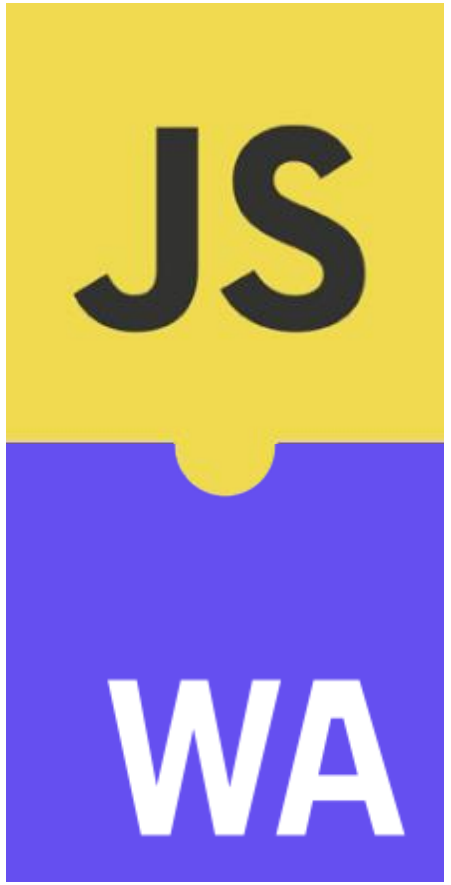
WebAssembly What? Why?

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

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



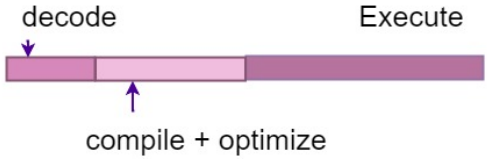
WebAssembly **is not designed to replace** JS, but to coexist
Sandboxed runtime in JS virtual machine
Security it runs locally in JS VM



WebAssembly What? Why?

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

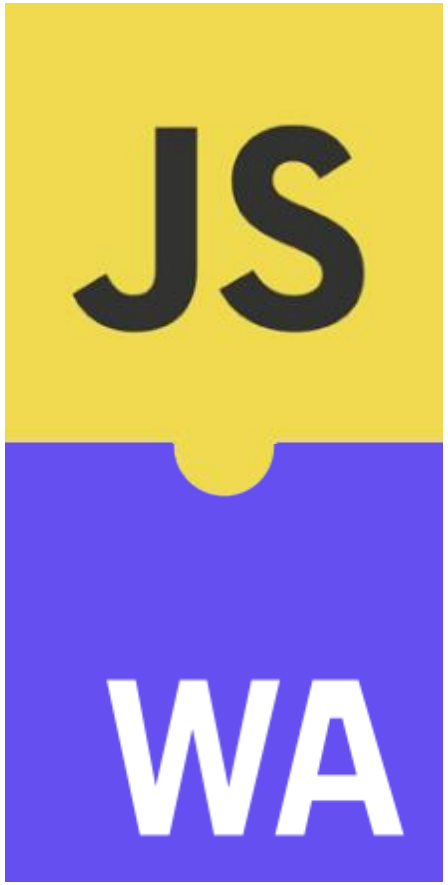
It's a bytecode for web / compilation target ➡ maximized performance



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

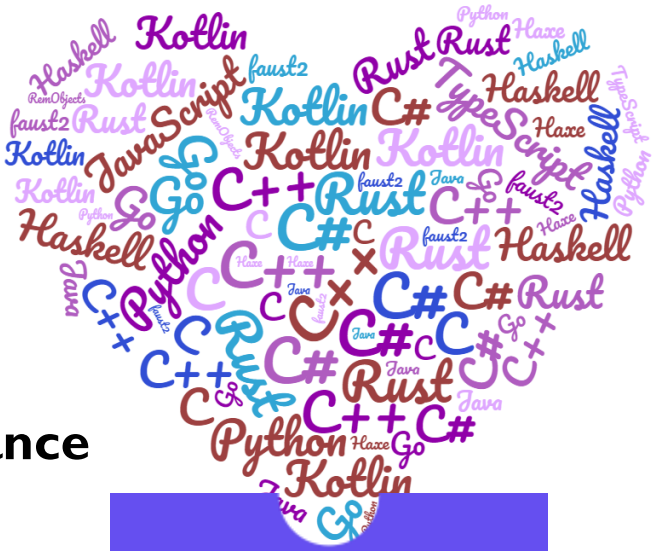
Sandboxed runtime in JS virtual machine

Security it runs locally in JS VM



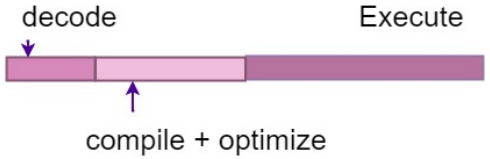
Supported in **all** big browsers – W3C open specification

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

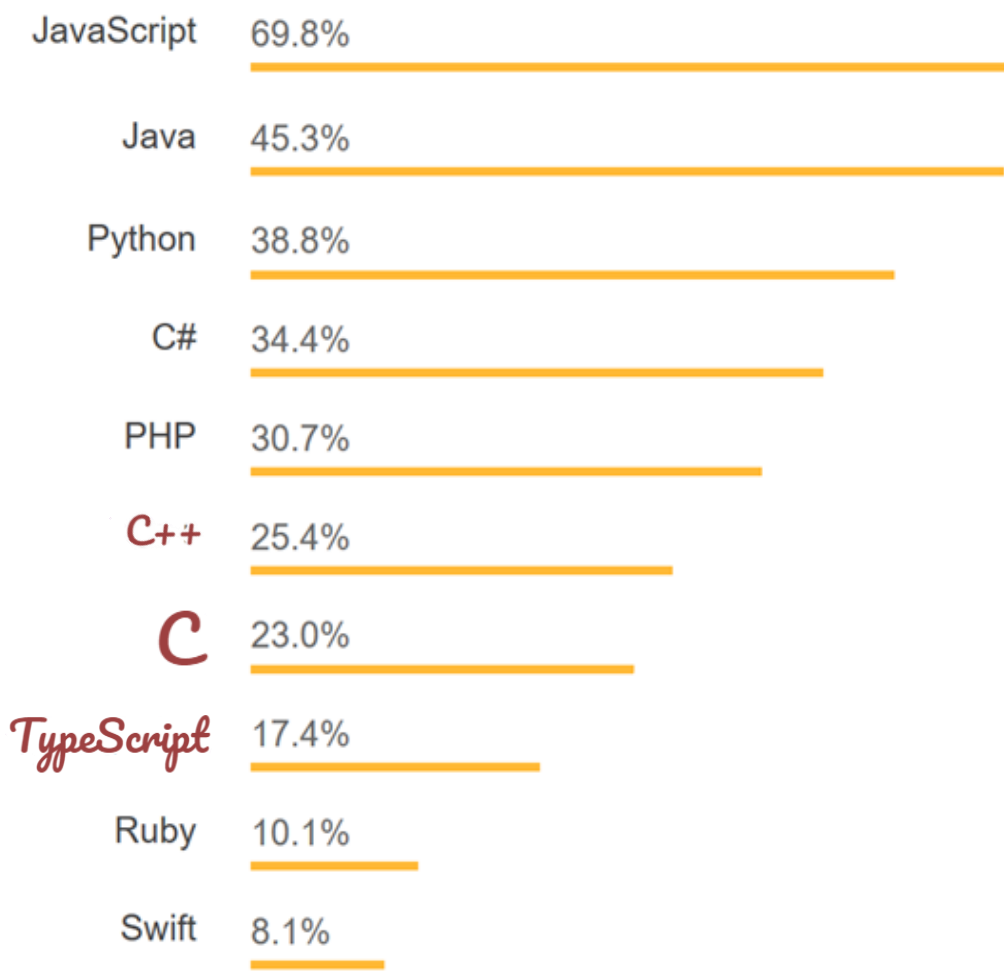
Sandboxed runtime in JS virtual machine

Security it runs locally in JS VM

Supported in **all** big browsers – W3C open specification

Compiled from **other languages**

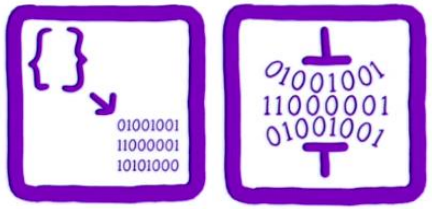
WASM opens possibilities for other languages



WebAssembly current state: MVP

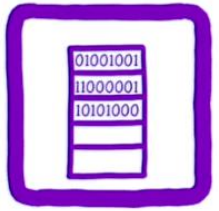
- Compilation target

  Binary so compact



- 4 types, 67 instructions, stack machine

- Linear memory

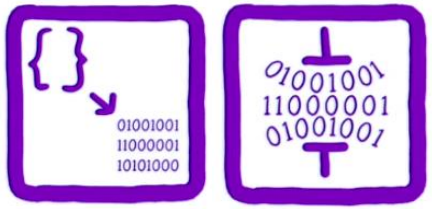


- Fast execution

WebAssembly current state: MVP

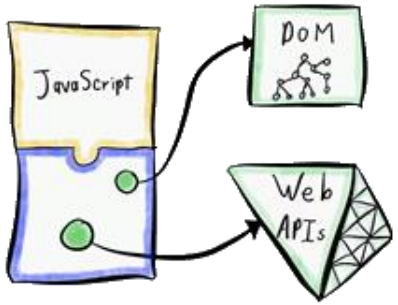
- Compilation target

  Binary so compact

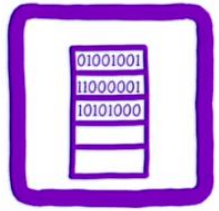


- 4 types, 67 instructions, stack machine

- No Web APIs, DOM access/manipulation

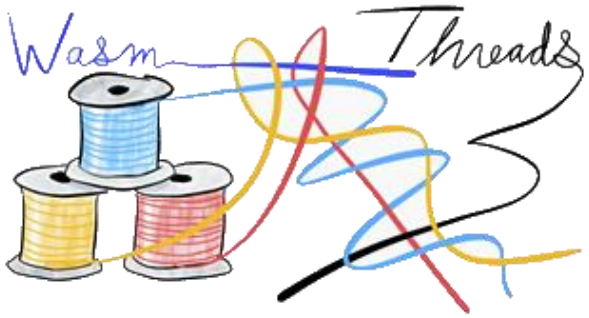


- Linear memory



- Fast execution or not so fast not so fast load

- no Threads



WebAssembly How?

executes the **Emscripten** compiler

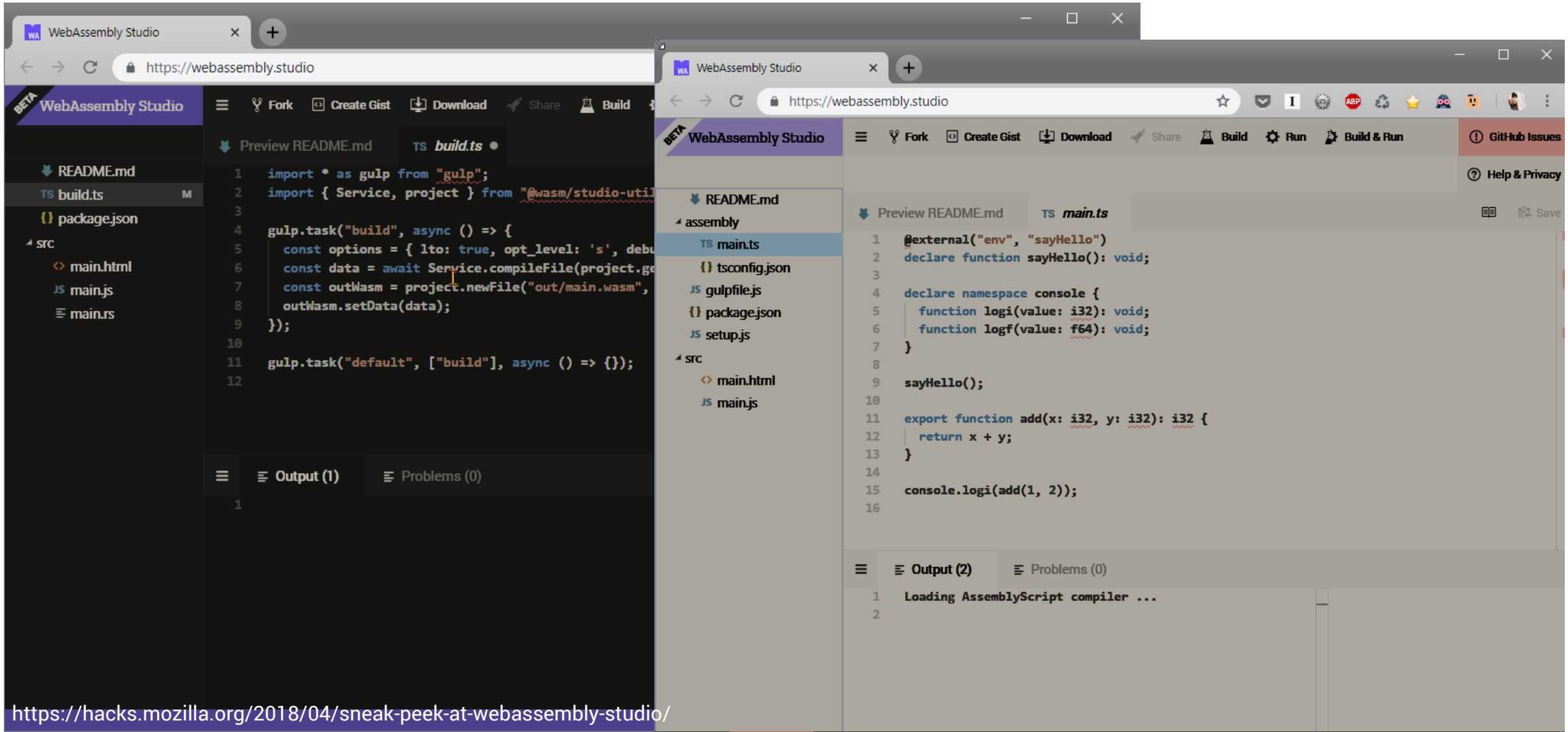
name our output file

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

Emscripten outputs WebAssembly - switch

input file

WebAssembly studio C/Rust/TypeScript



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

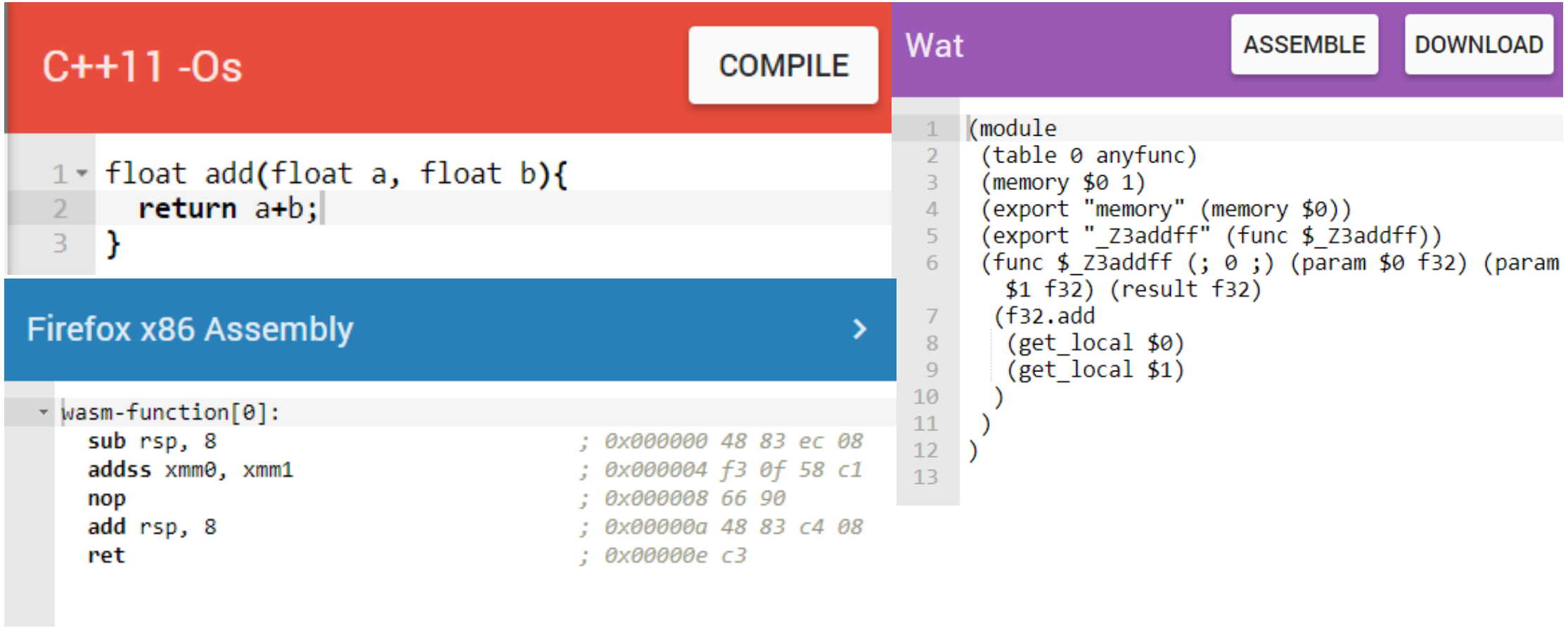
Demo

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

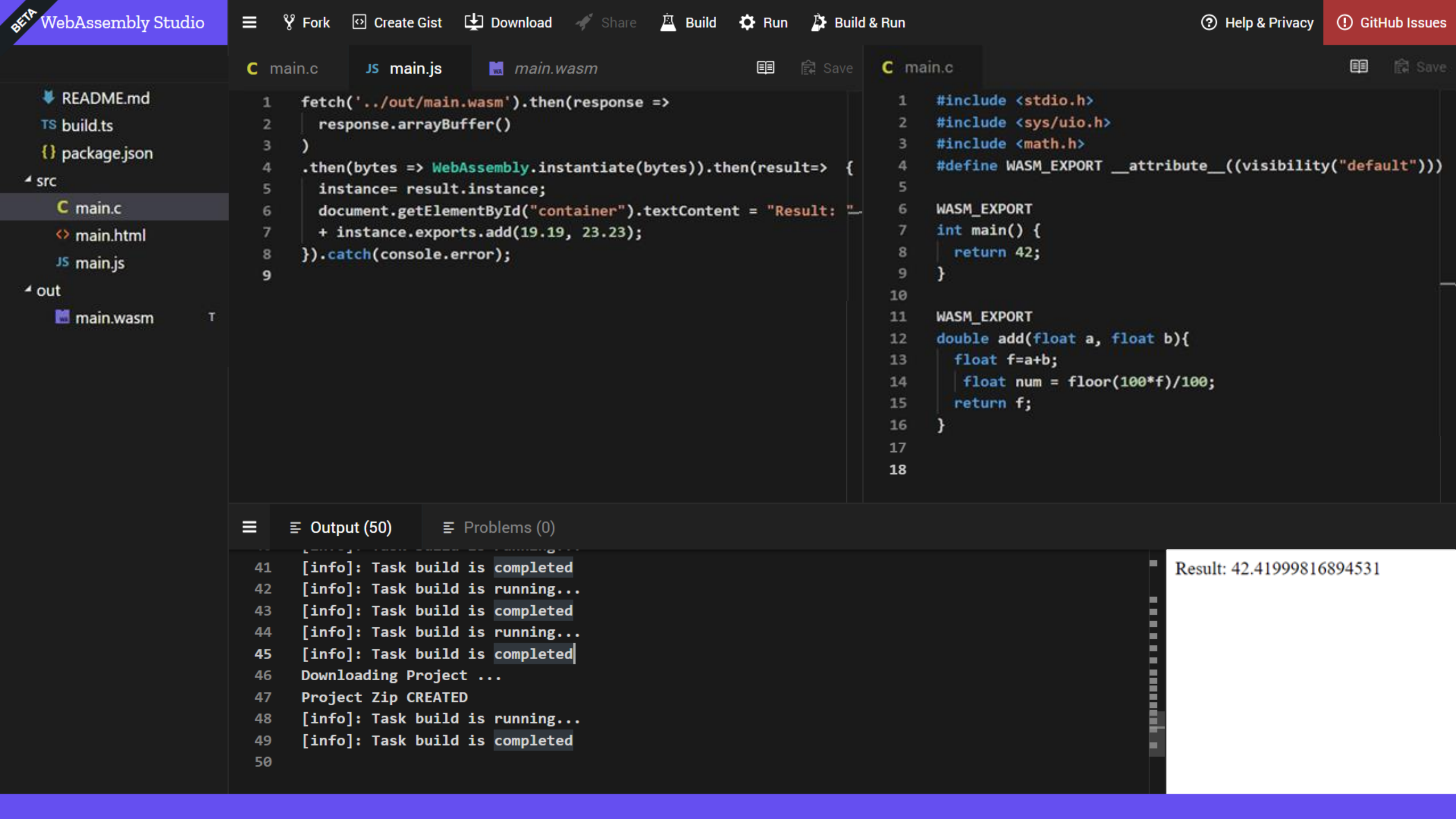
https://wasdk.github.io/WasmFiddle/

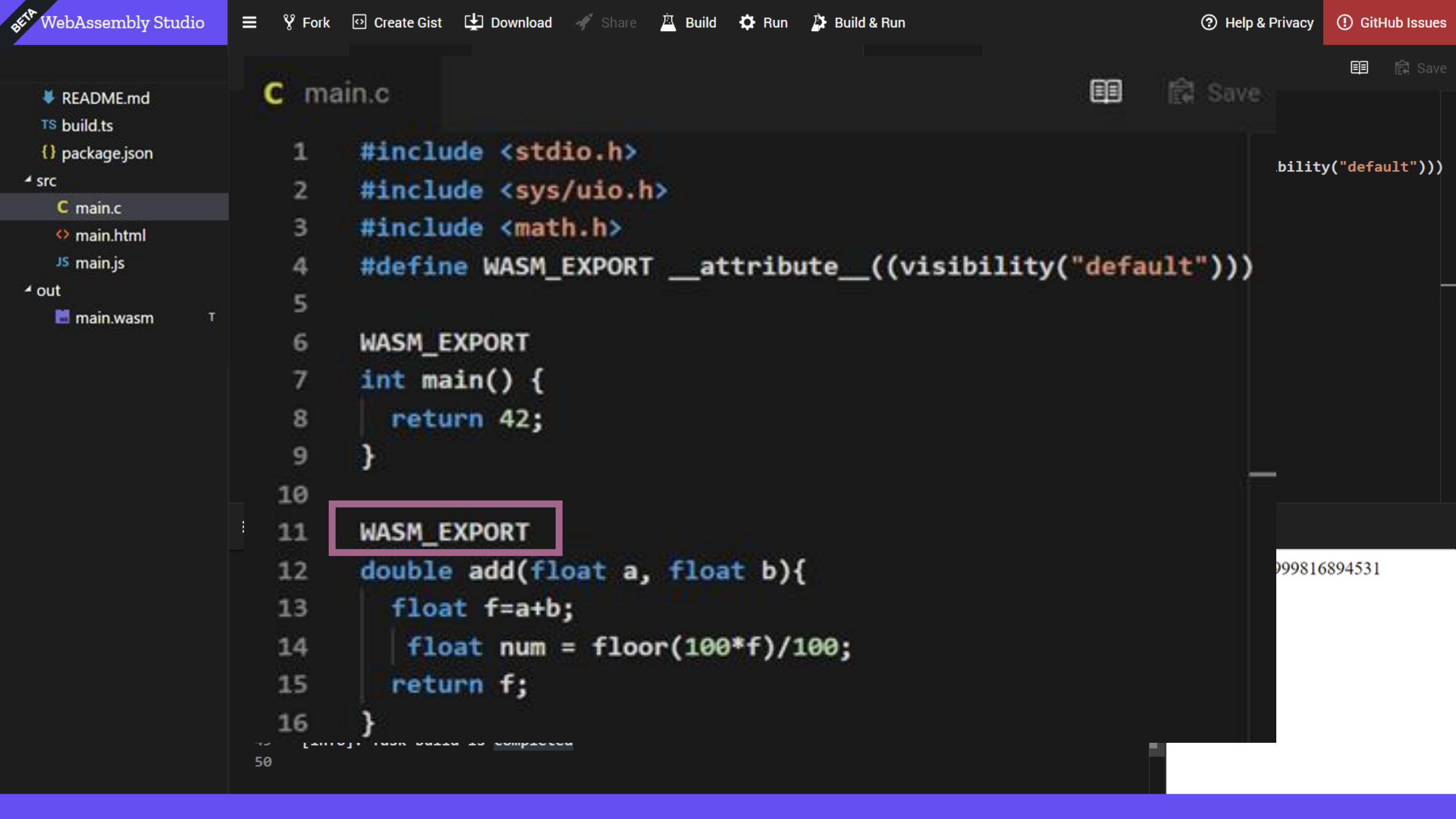
https://webassembly.studio/

WasmExplorer



https://bit.ly/2JkqFYt


BETA
WebAssembly Studio
Fork
Create Gist
Download
Share
Build
Run
Build & Run
Help & Privacy
GitHub Issues
README.md
build.ts
package.json
src
main.c
main.html
main.js
out
main.wasm
main.c
main.js
main.wasm
Save
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);
main.c
Save
#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;
}
Output (50)
Problems (0)
[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


BETA
WebAssembly Studio
Fork
Create Gist
Download
Share
Build
Run
Build & Run
Help & Privacy
GitHub Issues
README.md
build.ts
package.json
src
main.c
main.html
main.js
out
main.wasm
main.c
Save
#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;
}
bility("default")))
999816894531

BETA WebAssembly Studio

Fork | Create Gist | Download | Share | Build | Run | Build & Run | Help & Privacy | GitHub Issues

- README.md
- build.ts
- package.json
- src
  - main.c
  - main.html
  - main.js
- out
  - main.wasm

main.c | main.js | main.wasm | Save | main.c | Save

```
1  fetch('../out/main.wasm').then(response =>
2    response.arrayBuffer()
3  )
4  .then(bytes => WebAssembly.instantiate(bytes)).then(result=>  {
5    instance= result.instance;
6    document.getElementById("container").textContent = "Result: "
7    + instance.exports.add(19.19, 23.23);
8  }).catch(console.error);
9
```

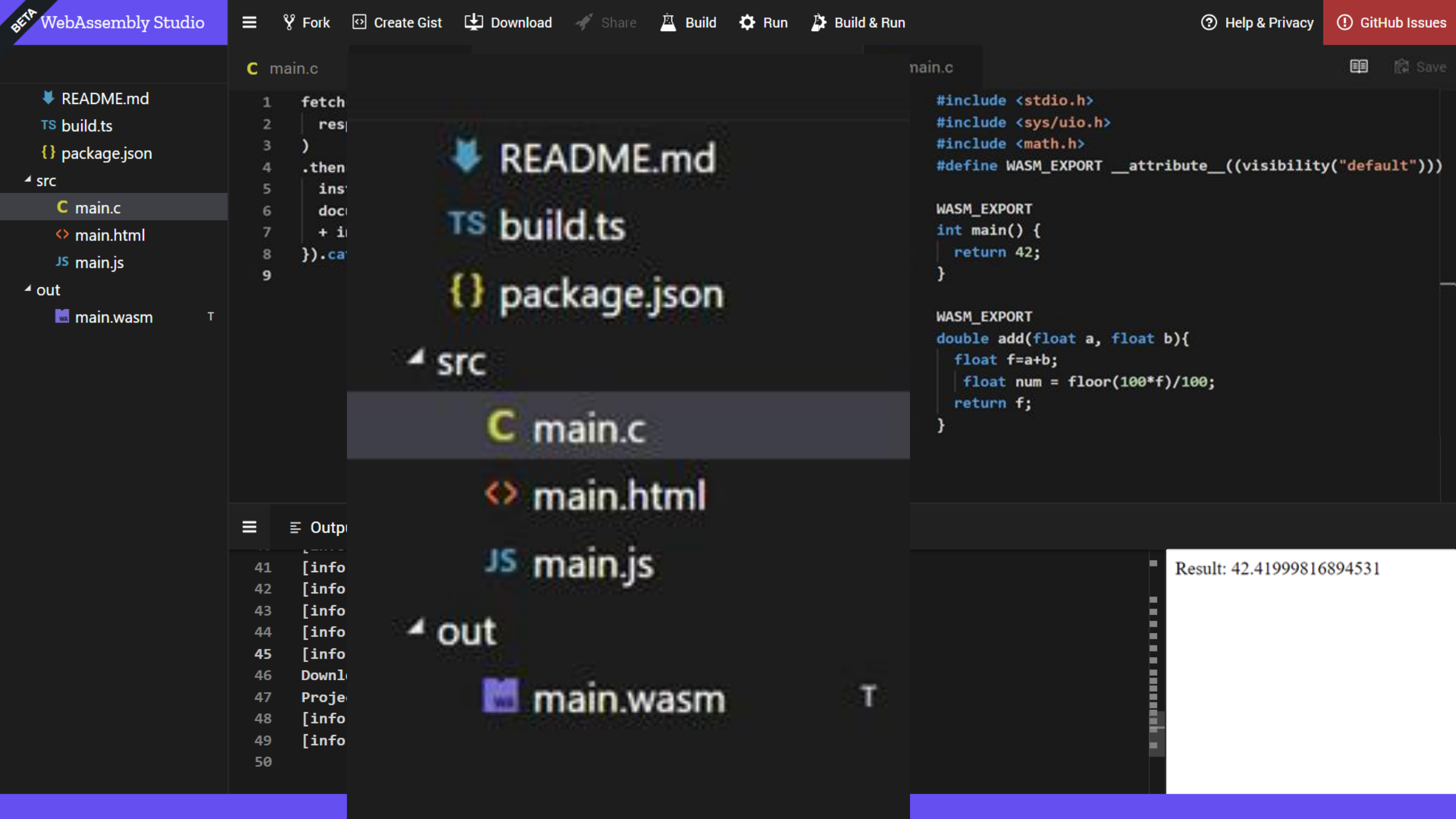
efault")))

Result: 42.41999816894531

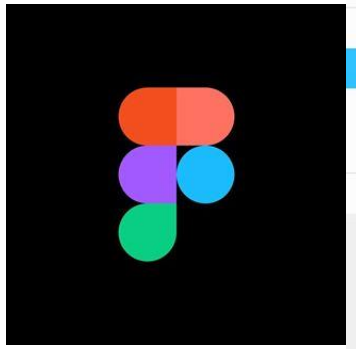
Output (50) | Problem

```
41  [info]: Task build is com
42  [info]: Task build is run
43  [info]: Task build is com
44  [info]: Task build is run
45  [info]: Task build is com
46  Downloading Project ...
47  Project Zip CREATED
48  [info]: Task build is running...
49  [info]: Task build is completed
50
```

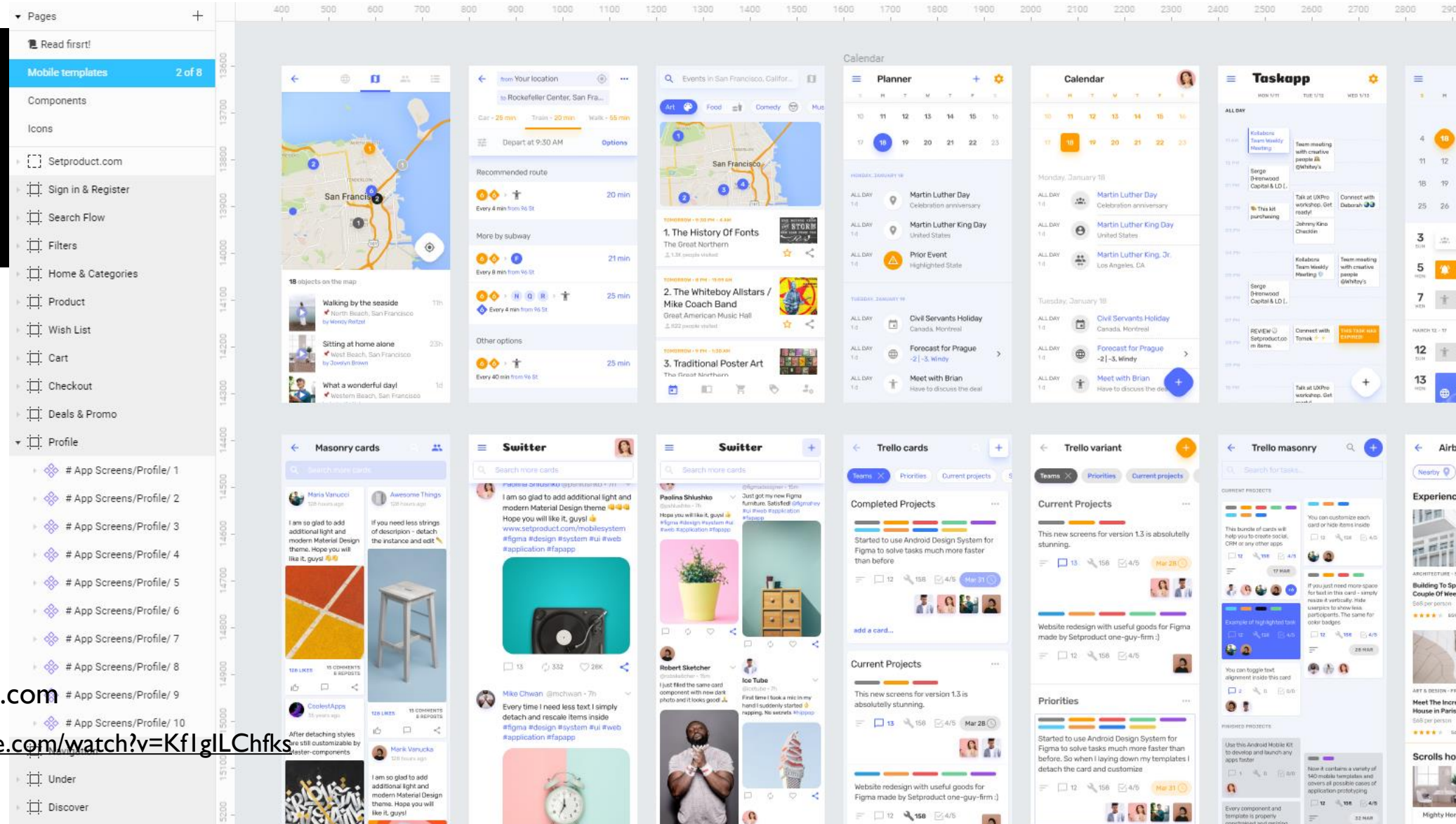
Result: 42.41999816894531


BETA
WebAssembly Studio
Fork
Create Gist
Download
Share
Build
Run
Build & Run
Help & Privacy
GitHub Issues
README.md
build.ts
package.json
src
main.c
main.html
main.js
out
main.wasm
Save
#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;
}
Result: 42.41999816894531

WASM in real world



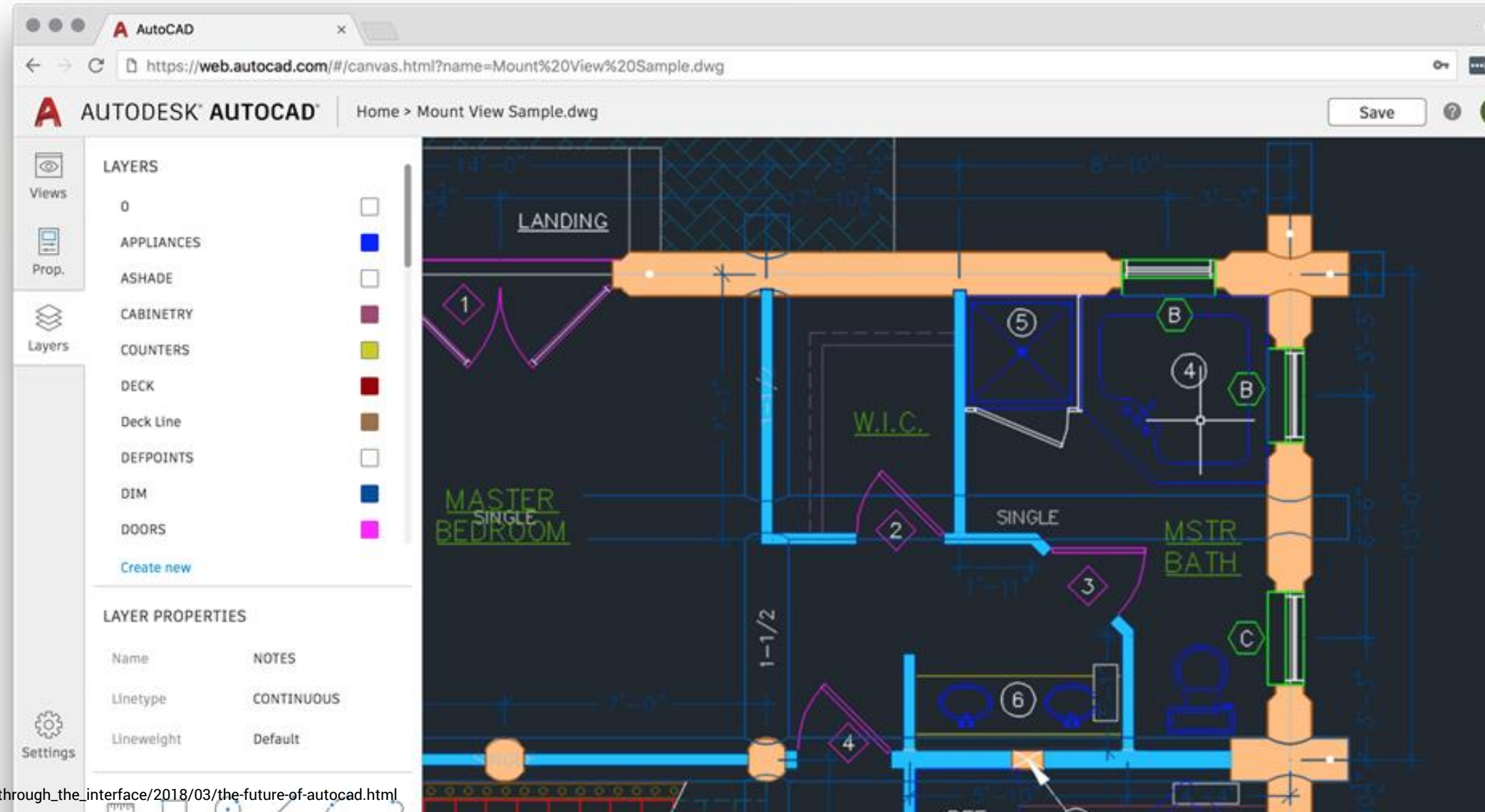
Figma



https://www.figma.com

https://www.youtube.com/watch?v=Kf1gILChfks

WASM in real world

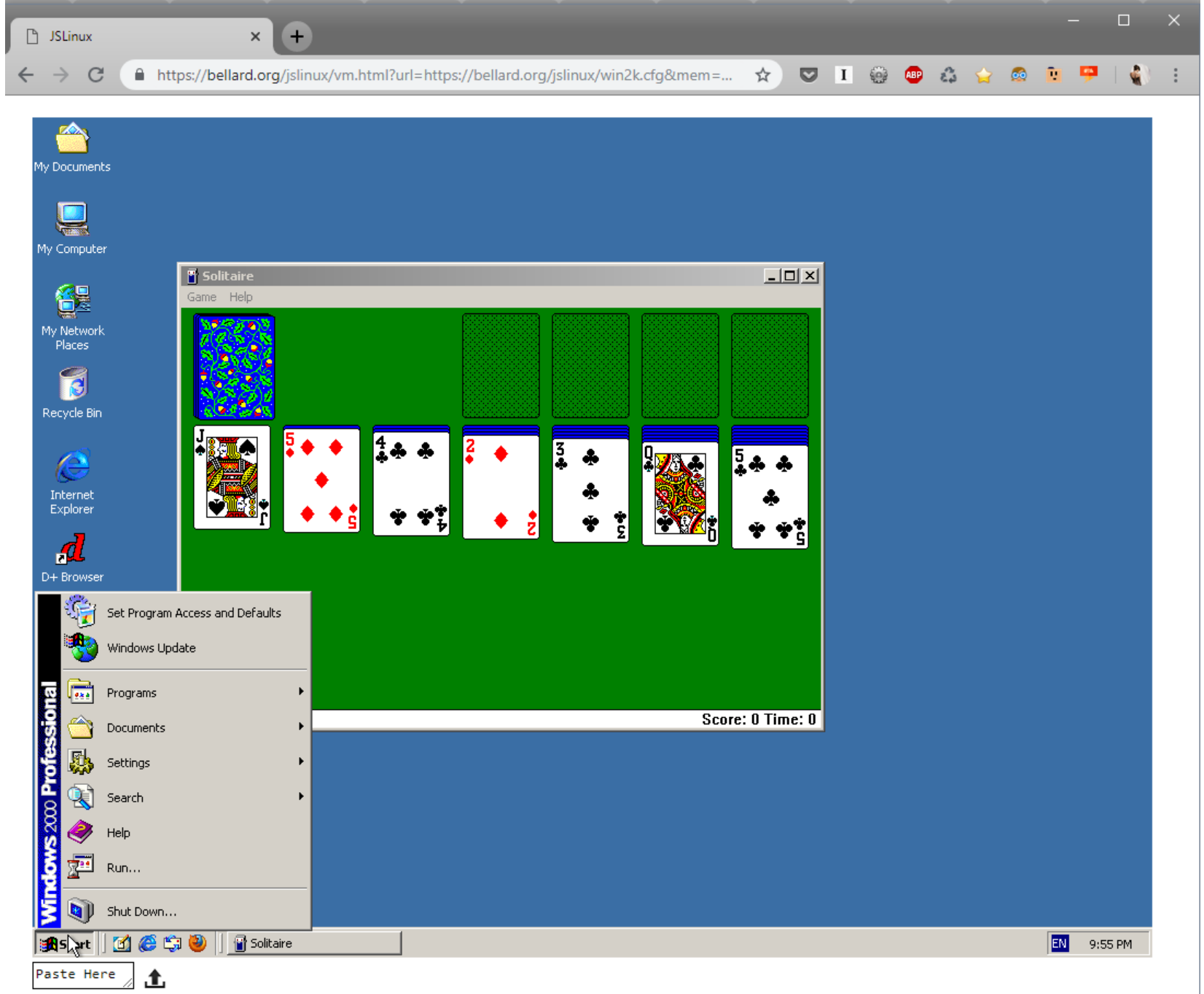


http://through-the-interface.typepad.com/through_the_interface/2018/03/the-future-of-autocad.html

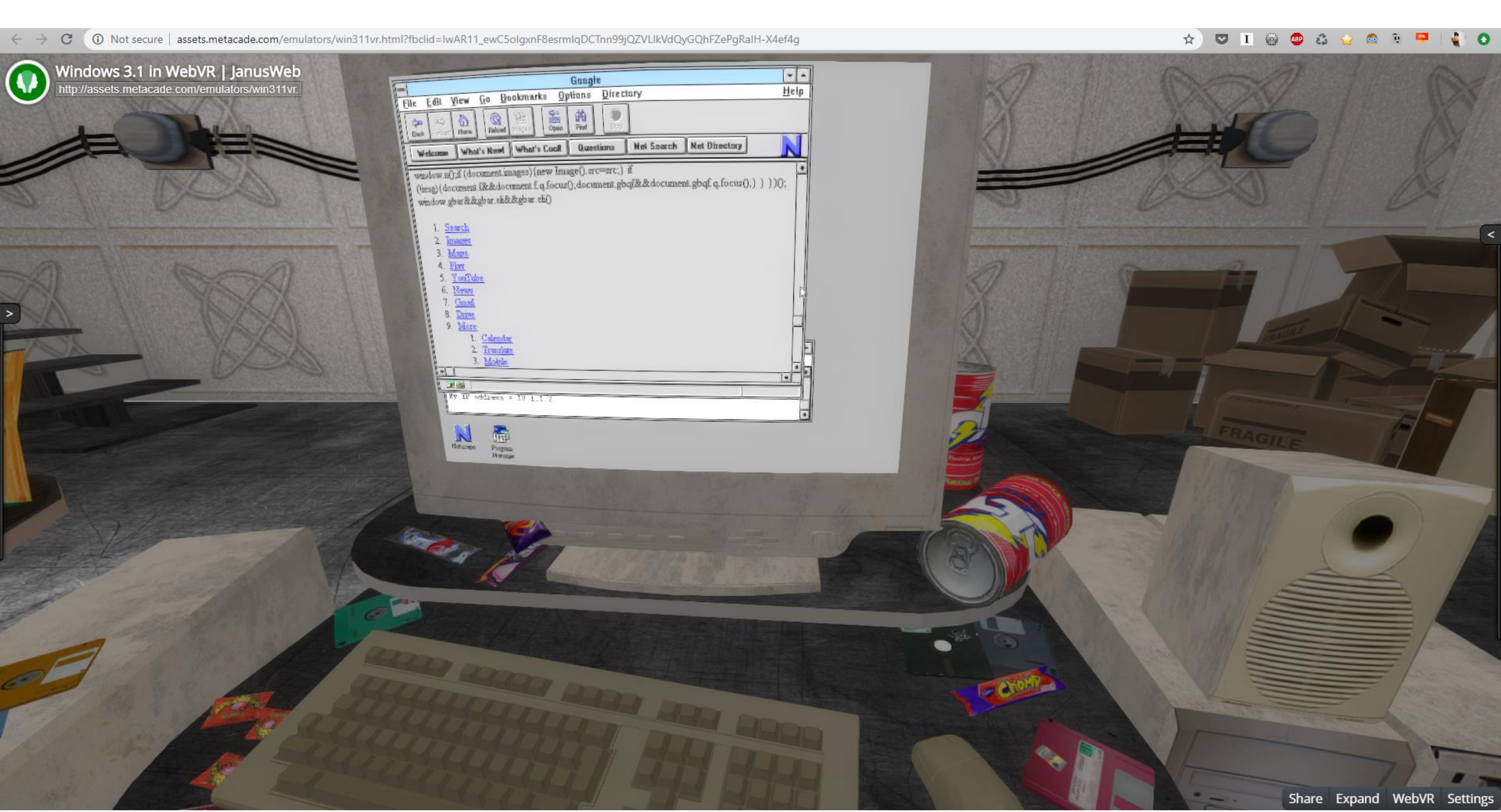
D3wasm - An experimental port of id Tech 4 engine to WebAssembly

Online demonstration running **Doom 3 Demo**

http://www.continuation-labs.com/projects/d3wasm/?fbclid=IwAR2V9OqEDgu3bu-vMNlxcZCUOm0HQAlv6ys-jcZGSMQY56saD8FYrHdVx_s

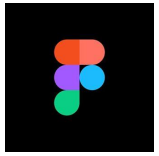


https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/win2k.cfg&mem=192&graphic=1&w=1024&h=768



http://assets.metacade.com/emulators/win311vr.html?fbclid=IwAR11_ewC5oIgxnF8esrmlqDCTnn99jQZVLlkVdQyGQhFZePgRaIH-X4ef4g

- WASM in real world



Figma

https://www.figma.com/

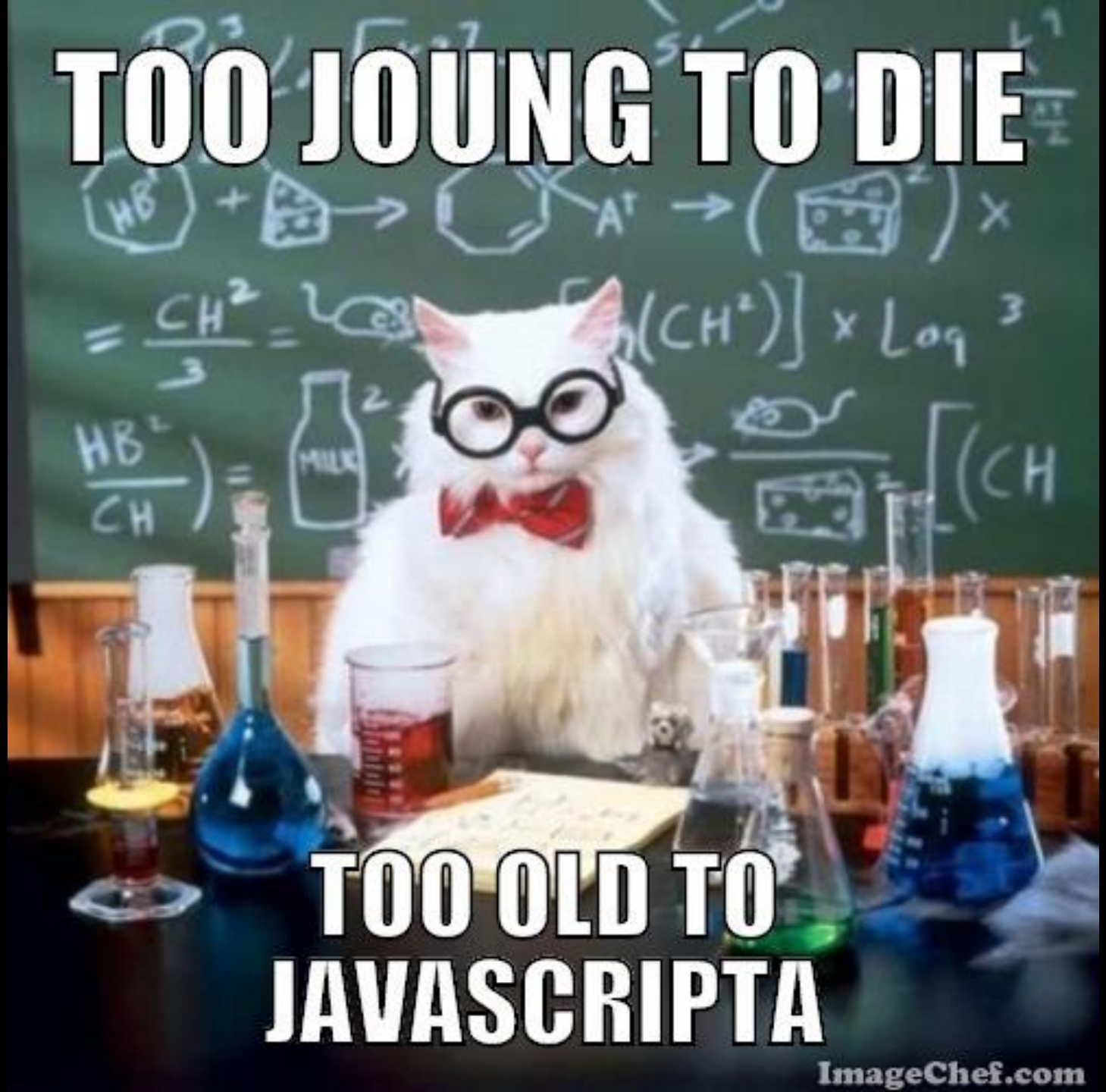
https://www.youtube.com/watch?v=Kf1gILChfks

AUTODESK® AUTOCAD®

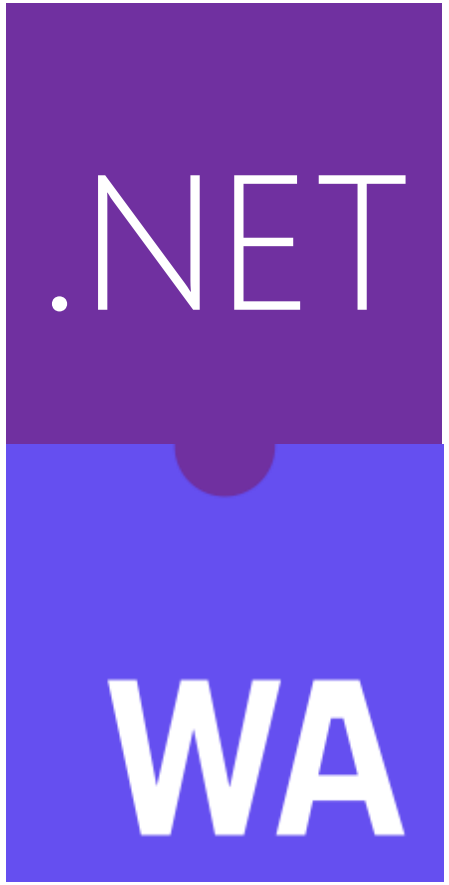
D3WASM

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_ewC5oIgxnF8esrmlqDCTnn99jQZVLlkVdQyGQhFZePgRaIH-X4ef4g


TOO JOUNG TO DIE
TOO OLD TO
JAVASCRIPTA
ImageChef.com

Why use .NET for browser apps?



Stable, mature, productive: **.NET Standard, MSBuild**

Fast, scalable, reliable: **.NET Core for backend services**

Modern languages: **Innovations in C#, F#, Razor**

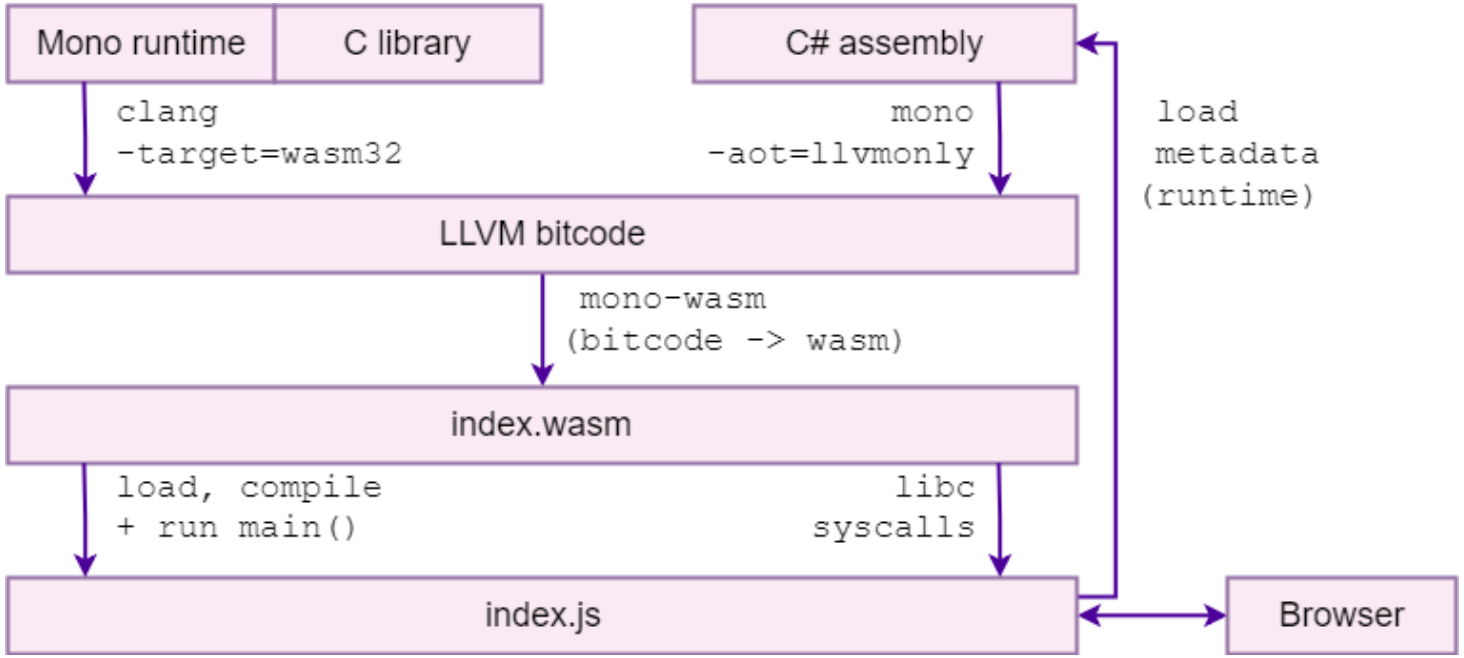
First-rate dev tools: **Visual Studio, IntelliSense**

Solid foundation to build on

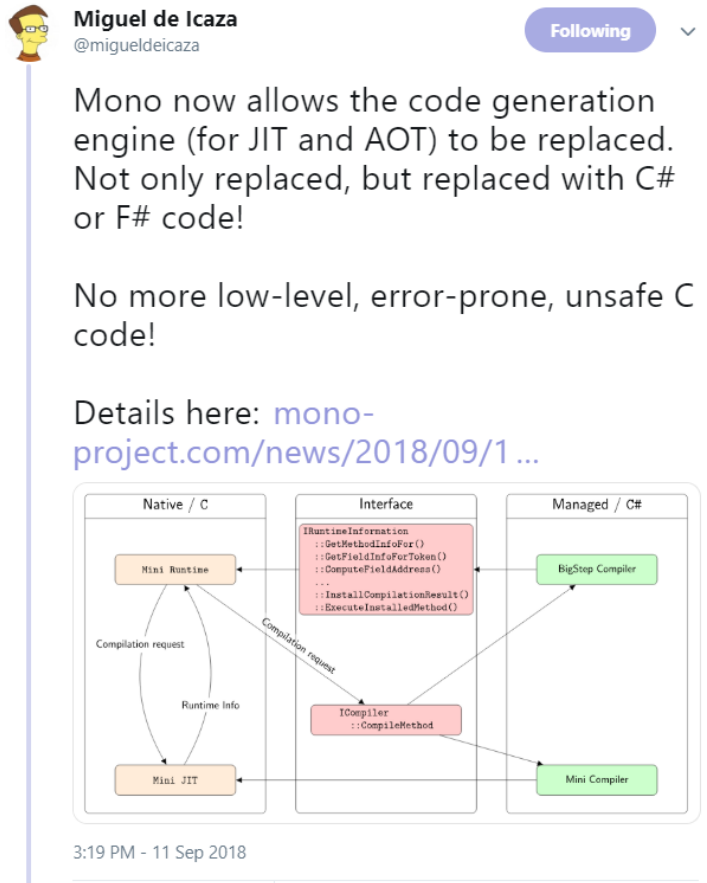
Leverage existing skills and knowledge

Because We can 

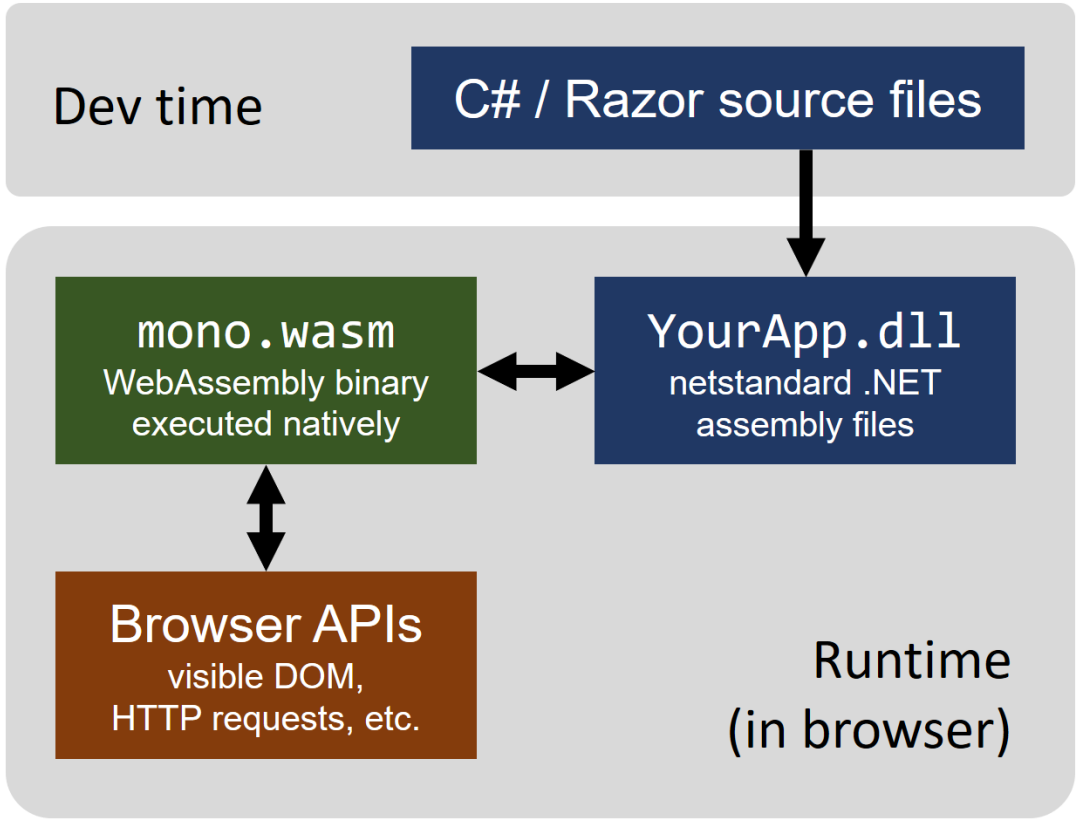
C# in wasm ? Mono !

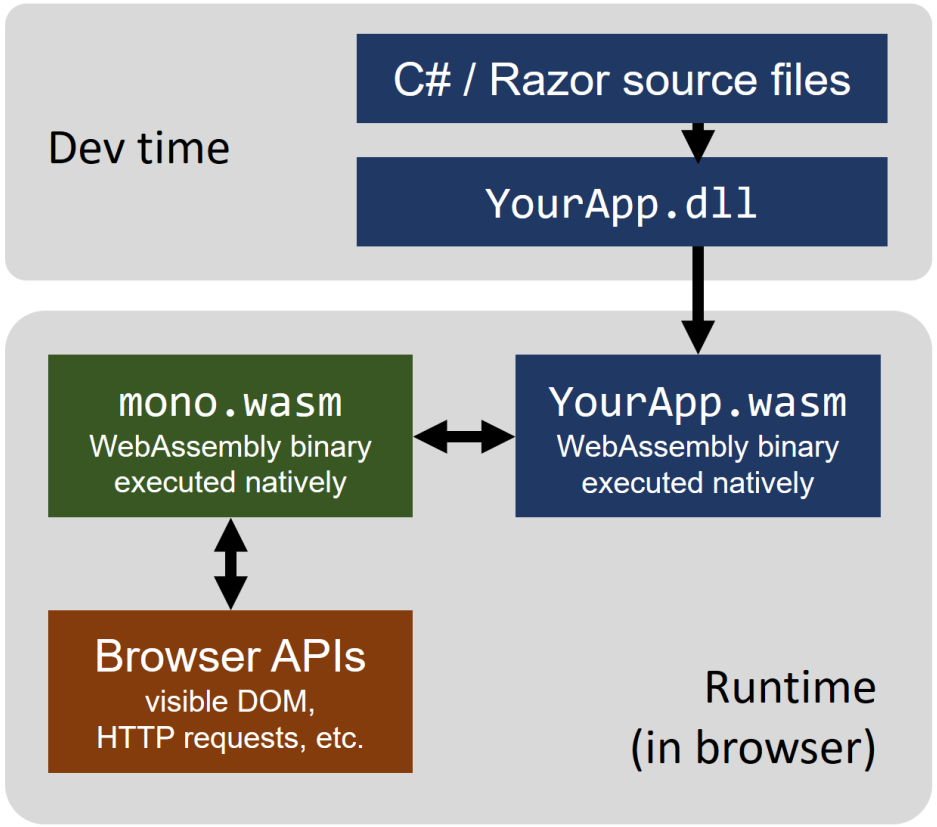


https://github.com/migueldeicaza/mono-wasm
https://www.mono-project.com/news/2018/09/11/csharp-jit/
https://twitter.com/migueldeicaza/status/1039639597435641856



Interpreter mode vs AOT compiled





XAML for desktop app

XAML standard is ready

XAML for desktop app

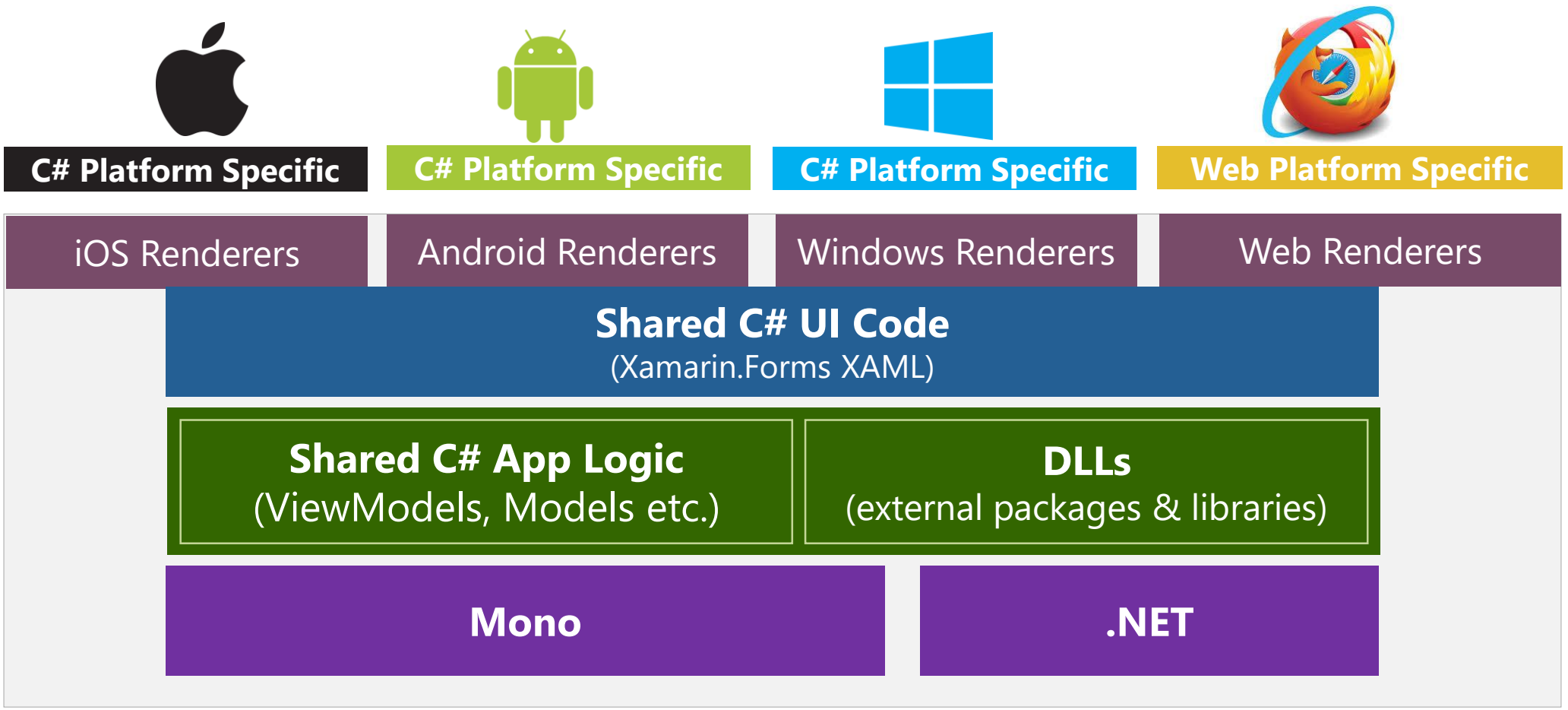
XAML standard is ready and dead

XAML for desktop app

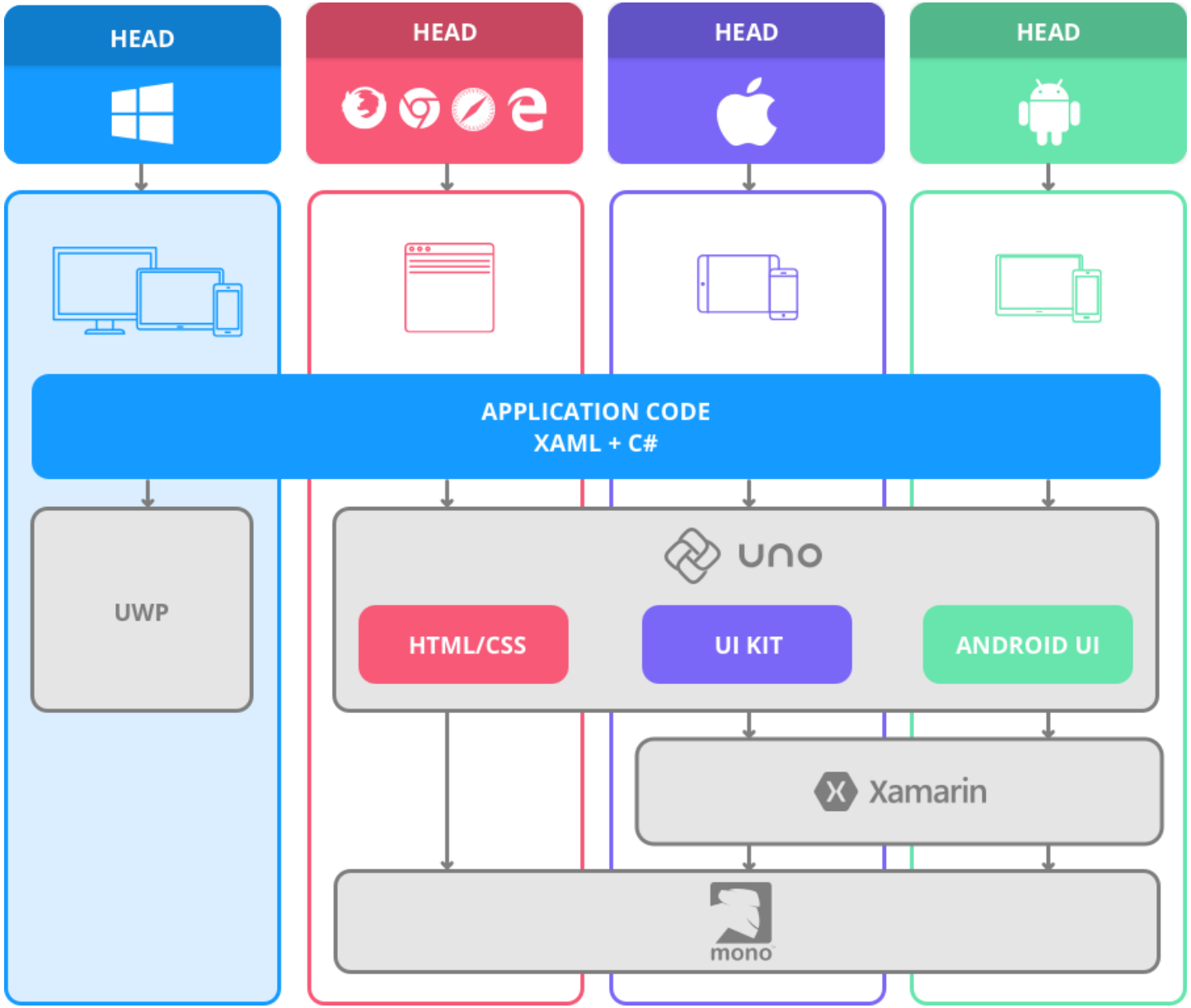
XAML standard is ready and dead

- Xamarin.Forms
- UWP
- Silverlight
- WPF

Ooui


C# Platform Specific
C# Platform Specific
C# Platform Specific
Web Platform Specific
iOS Renderers
Android Renderers
Windows Renderers
Web Renderers
Shared C# UI Code
(Xamarin.Forms XAML)
Shared C# App Logic
(ViewModels, Models etc.)
DLLs
(external packages & libraries)
Mono
.NET

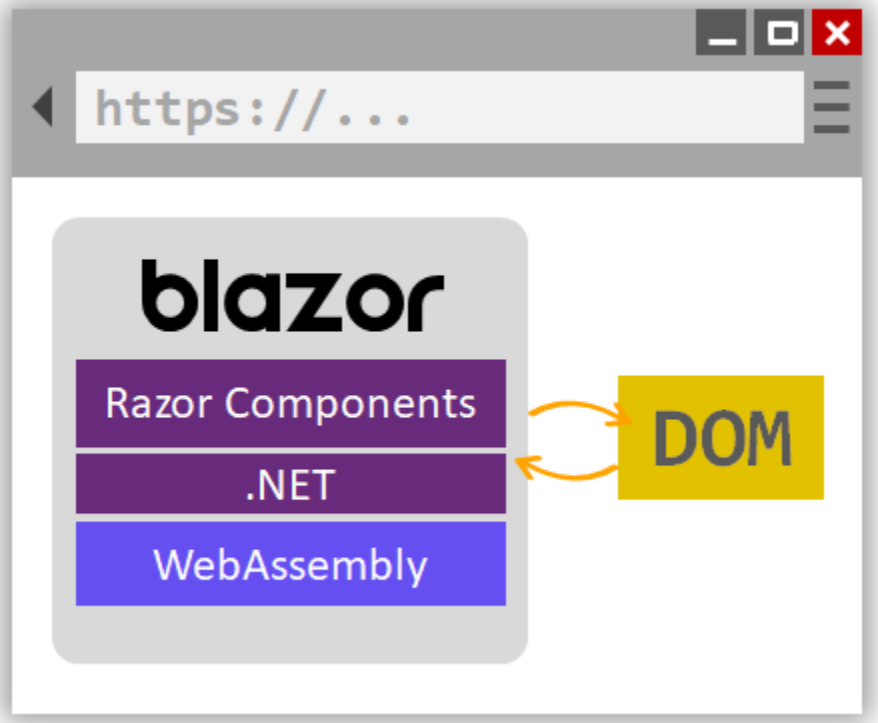
UNO


HEAD
HEAD
HEAD
HEAD
APPLICATION CODE
XAML + C#
UWP
uno
HTML/CSS
UI KIT
ANDROID UI
Xamarin
mono

FrogUI

https://www.lesarde.com/

Blazor? Razor components ?



Blazor

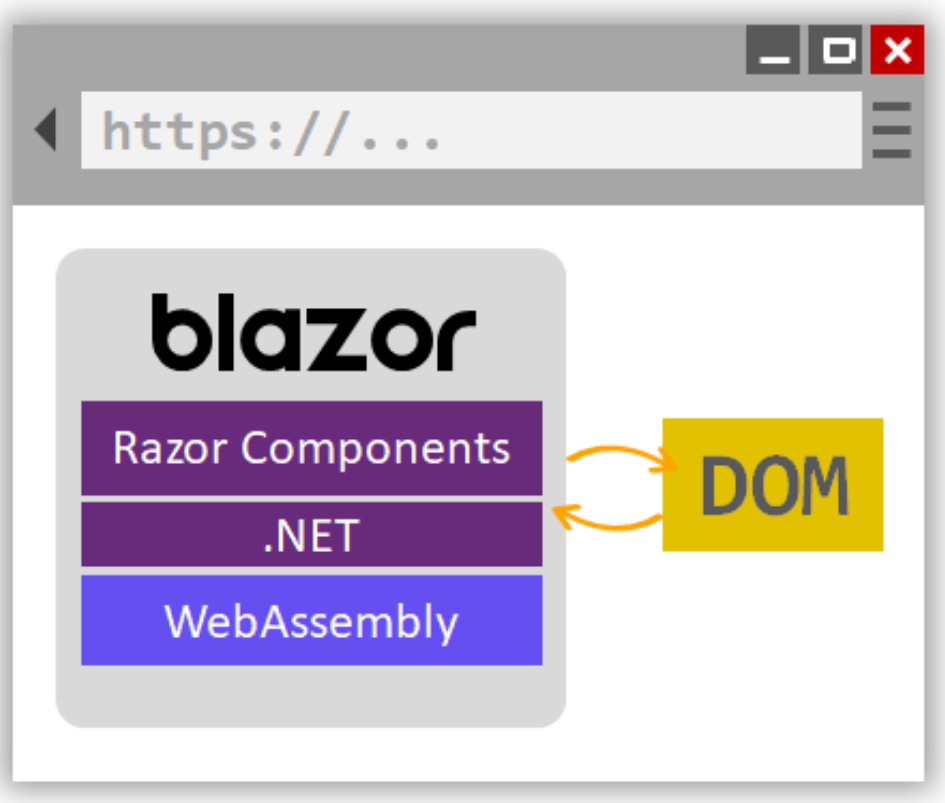
For information about using Blazor, see blazor.net.

Sources moved

Almost all the sources for Blazor and the Razor Components programming model have moved here in the central ASP.NET Core repo. We are also in the process of migrating open issues from here to there.

This is in preparation for shipping Razor Components as a built-in feature of ASP.NET Core 3.0. *Note: client-side Blazor remains experimental while we continue to work on making the WebAssembly runtime complete.*

Blazor? Razor components? It's complicated



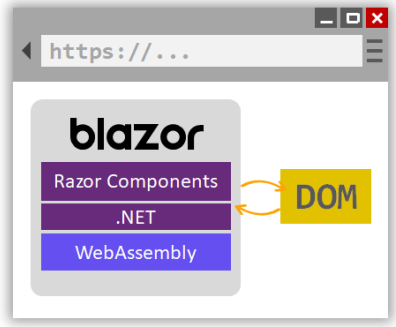
This is in preparation for shipping Razor Components as a built-in feature of ASP.NET Core 3.0. *Note: client-side Blazor remains experimental while we continue to work on making the WebAssembly runtime complete.*

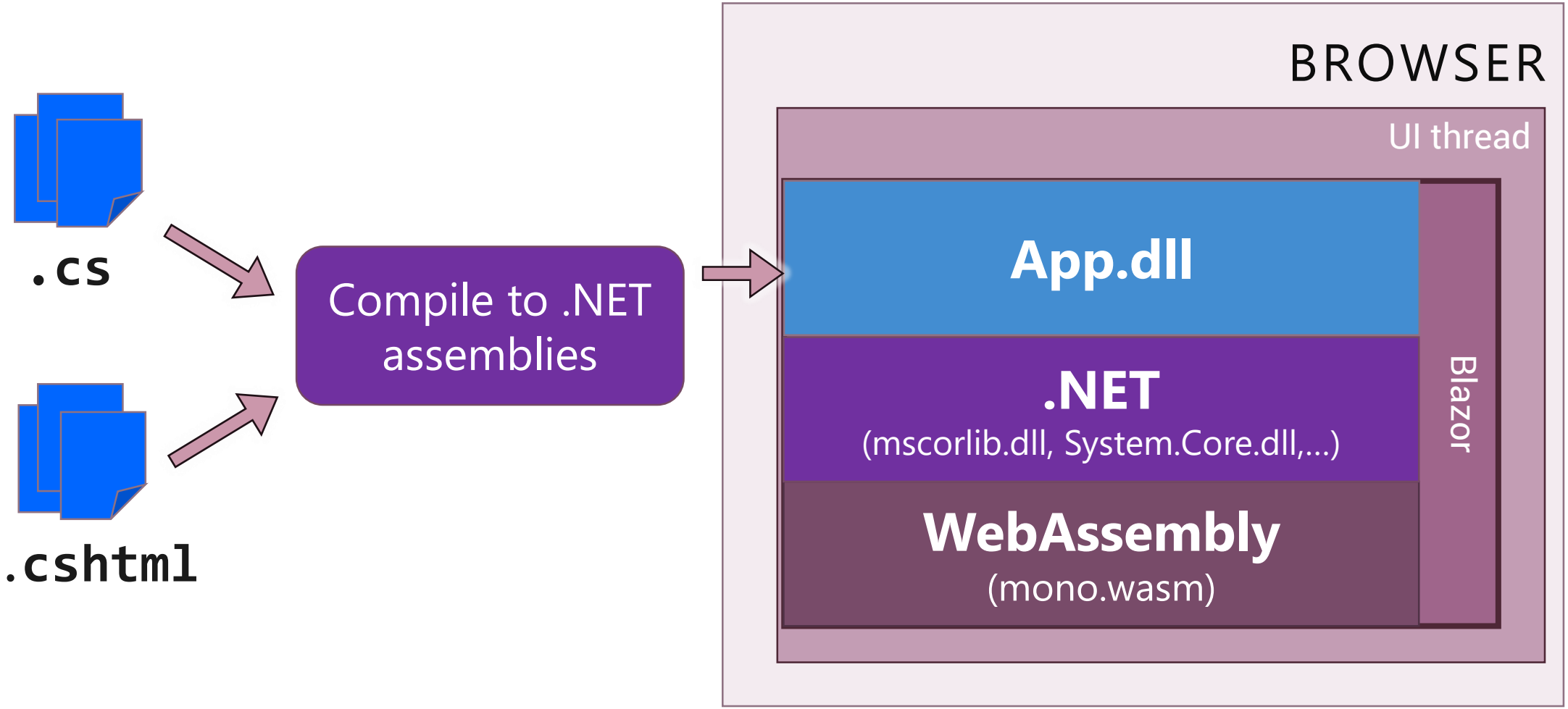
https://github.com/aspnet/AspNetCore/issues/8931

Blazor new project

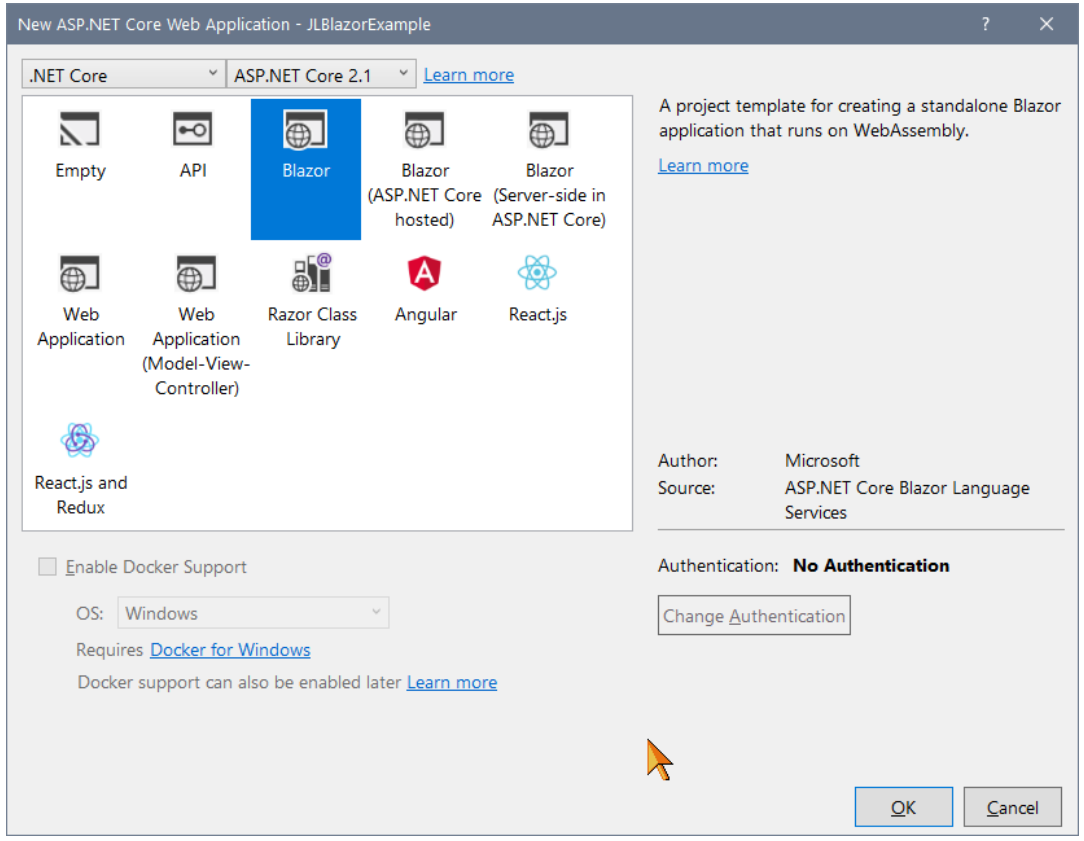
```
Blazor (hosted in ASP.NET server)                blazorhosted          [C#]
Blazor Library                                   blazorlib             [C#]
Blazor (standalone)                              blazor                [C#]
Razor Class Library                              razorclasslib         [C#]
ibrary
ASP.NET Core Web App (Razor Components)          razorcomponents       [C#]
```

Blazor client-side

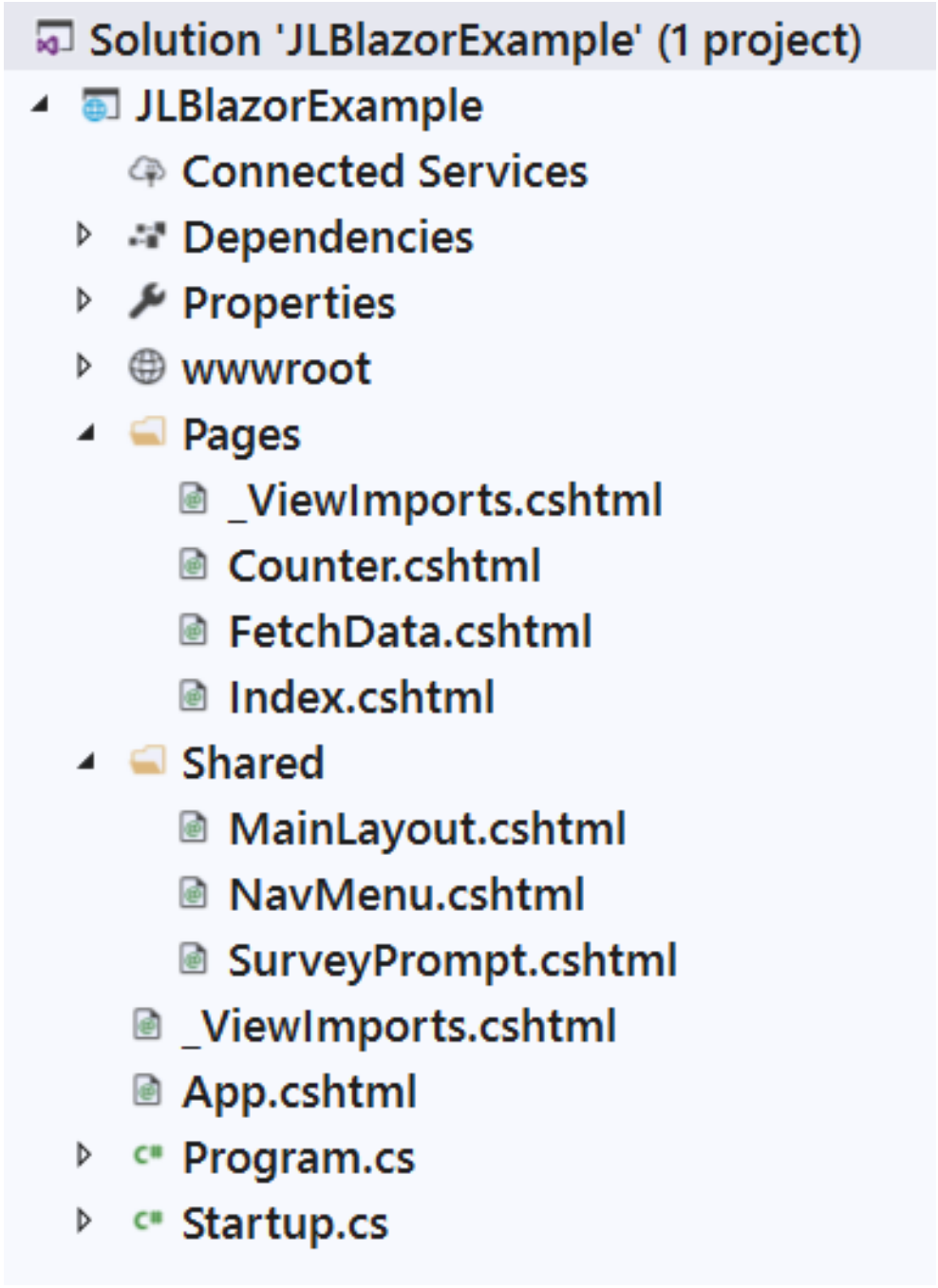




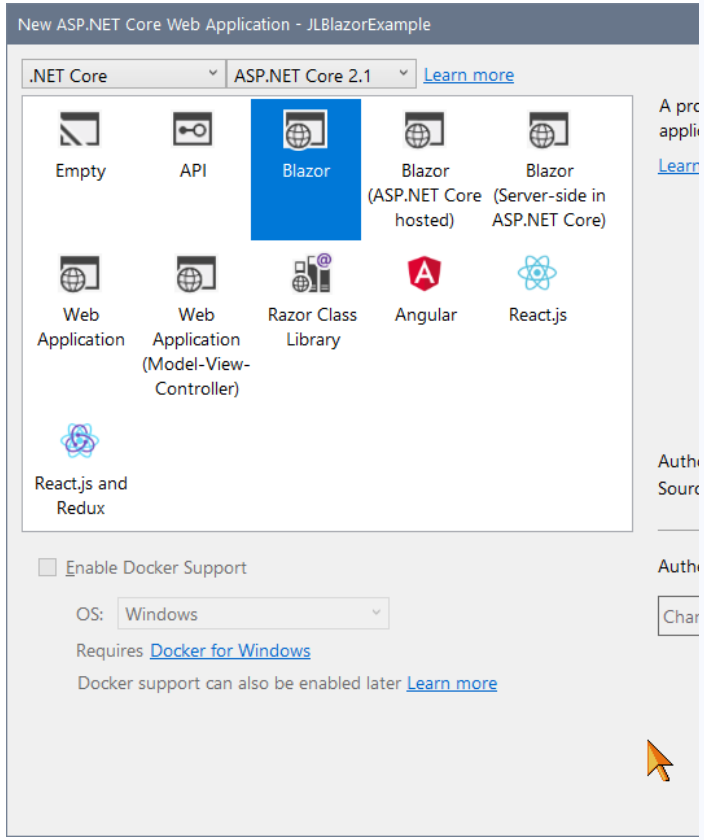
Blazor client-side




```
C:\Projects> dotnet new blazor
```



Blazor client



Properties

wwwroot

Pages

_ViewImports.cshtml

Counter.cshtml

FetchData.cshtml

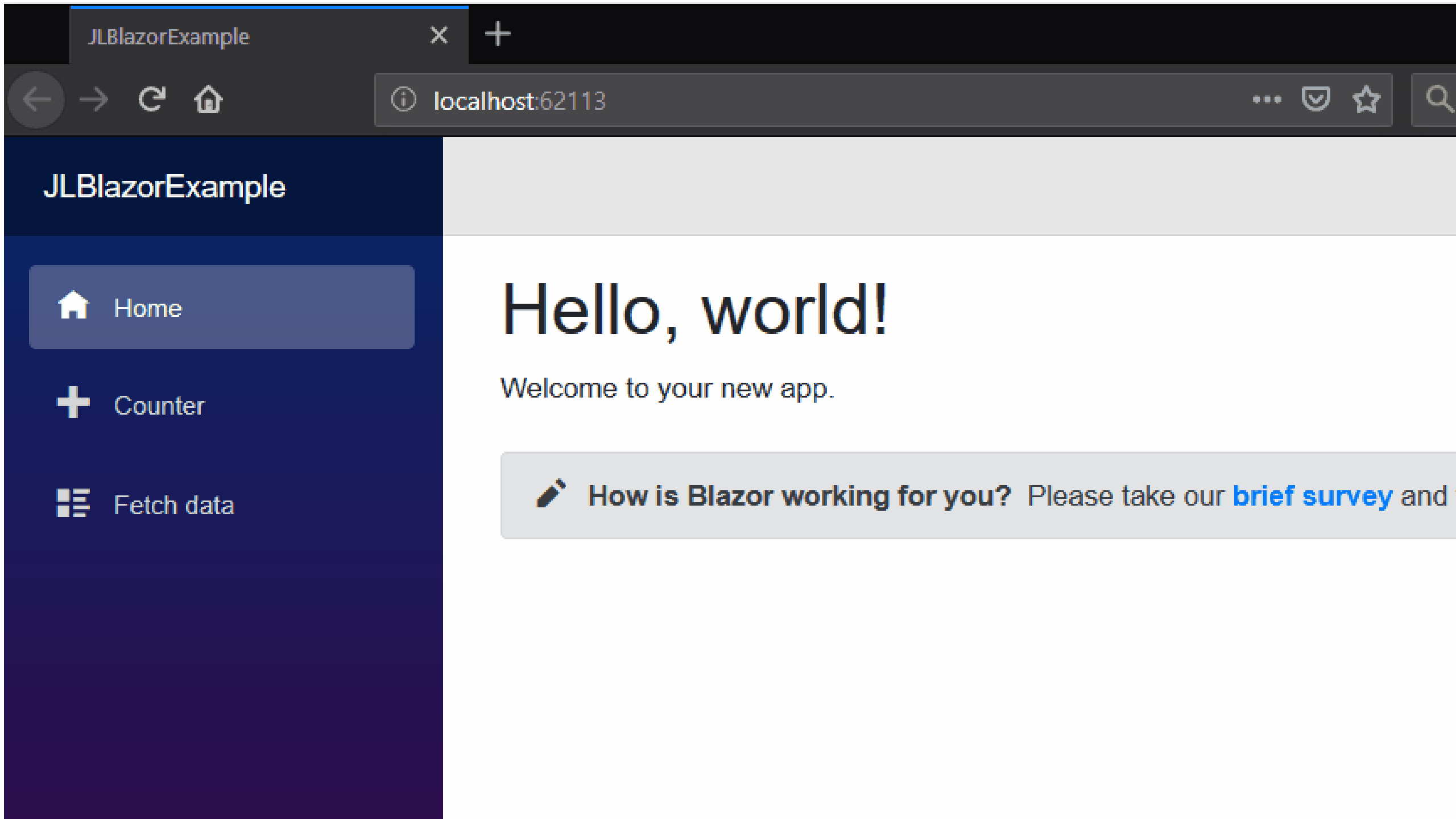
Index.cshtml

Shared

MainLayout.cshtml

NavMenu.cshtml

SurveyPrompt.cshtml


JLBlazorExample
localhost:62113
JLBlazorExample
Home
Counter
Fetch data
Hello, world!
Welcome to your new app.
How is Blazor working for you? Please take our brief survey and

Blazor: Sample

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

Blazor: Routing

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

Blazor: Razor template

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

Blazor: C#

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

Blazor: One way data binding

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

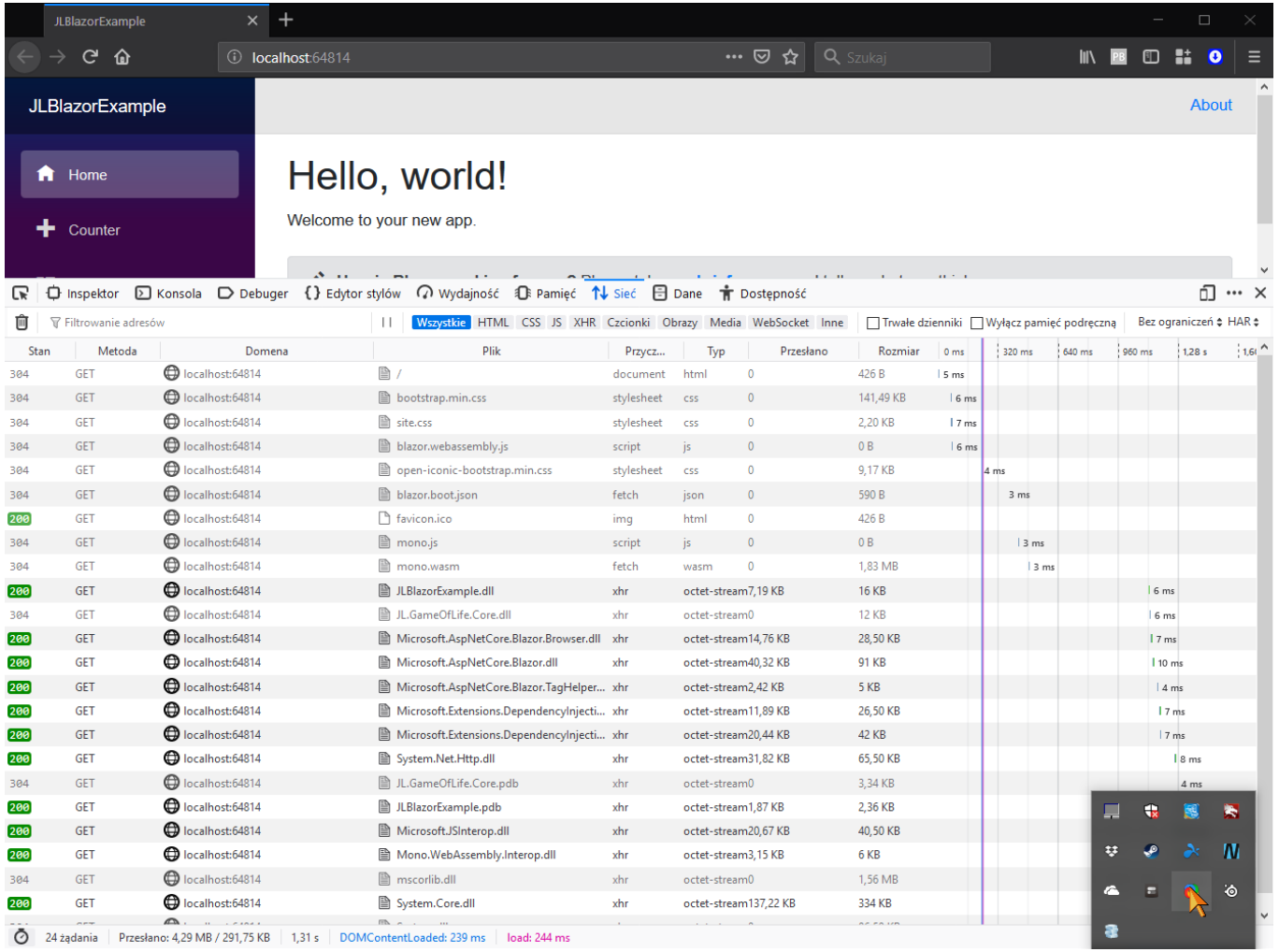
Blazor: Two way data binding

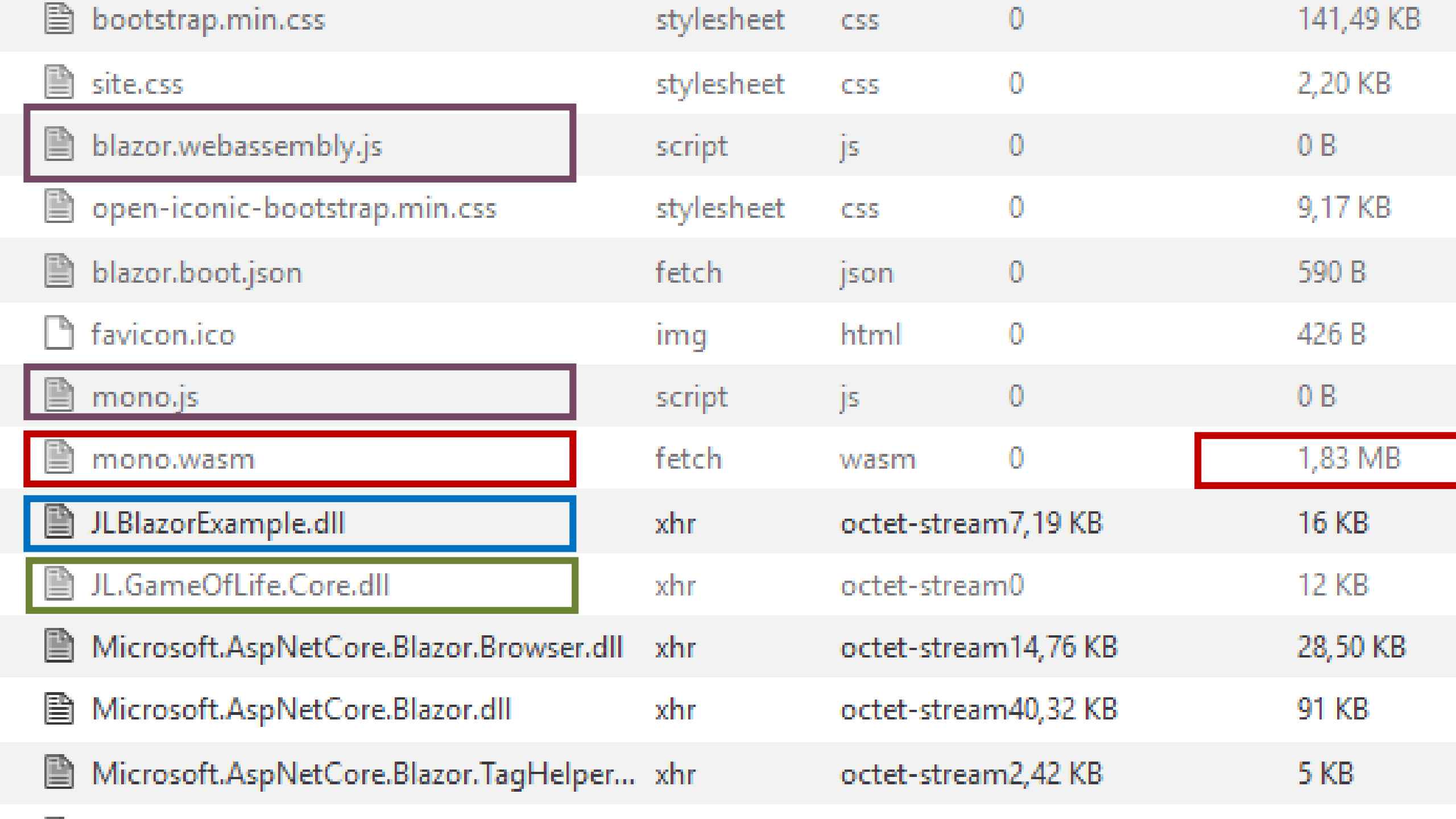
```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>

<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

Blazor: Event binding

```
@page "/counter"
@inject HttpClient

<h1>Counter</h1>
<input type="number" bind="IncrementAmount" />
<p>Current count: @currentCount</p>
<button class="btn btn-primary" onclick="@IncrementCount">Click me</button>

@functions {
    int currentCount = 0;
    int IncrementAmount { get; set; } = 1;

    void IncrementCount()
    {
        currentCount += IncrementAmount;
    }
}
```

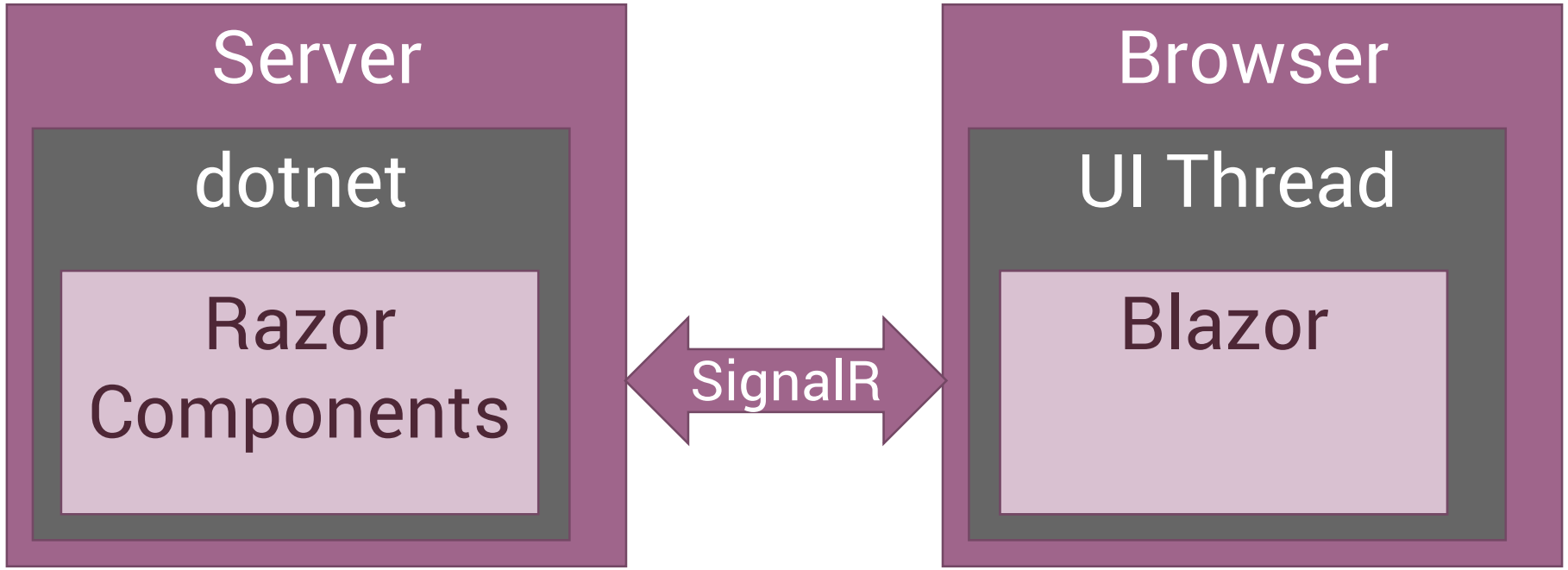
Blazor sample in browser



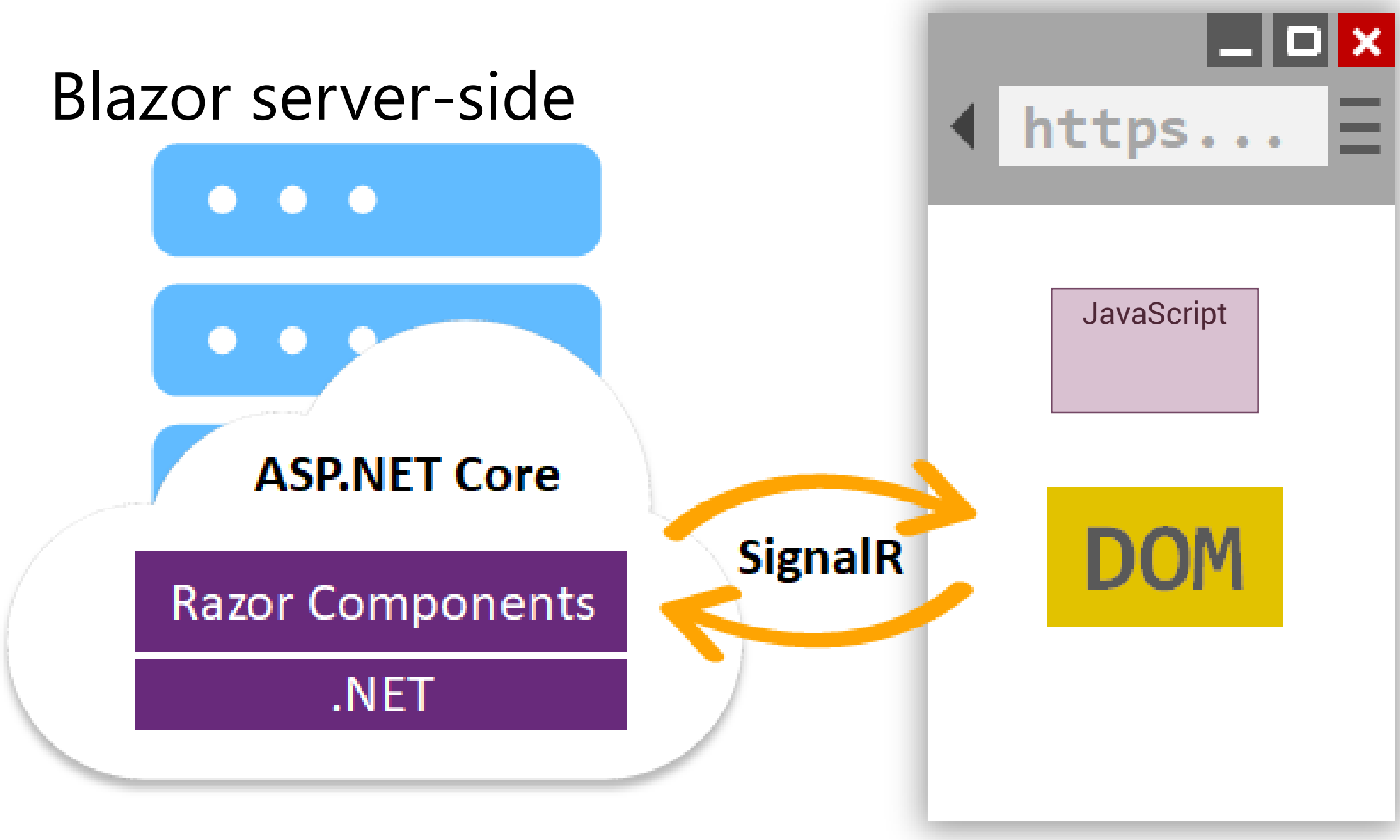


bootstrap.min.css	stylesheet	css	0	141,49 KB
site.css	stylesheet	css	0	2,20 KB
blazor.webassembly.js	script	js	0	0 B
open-iconic-bootstrap.min.css	stylesheet	css	0	9,17 KB
blazor.boot.json	fetch	json	0	590 B
favicon.ico	img	html	0	426 B
mono.js	script	js	0	0 B
mono.wasm	fetch	wasm	0	1,83 MB
JLBlazorExample.dll	xhr	octet-stream	7,19 KB	16 KB
JL.GameOfLife.Core.dll	xhr	octet-stream	0	12 KB
Microsoft.AspNetCore.Blazor.Browser.dll	xhr	octet-stream	14,76 KB	28,50 KB
Microsoft.AspNetCore.Blazor.dll	xhr	octet-stream	40,32 KB	91 KB
Microsoft.AspNetCore.Blazor.TagHelper...	xhr	octet-stream	2,42 KB	5 KB

Let’s switch elements places

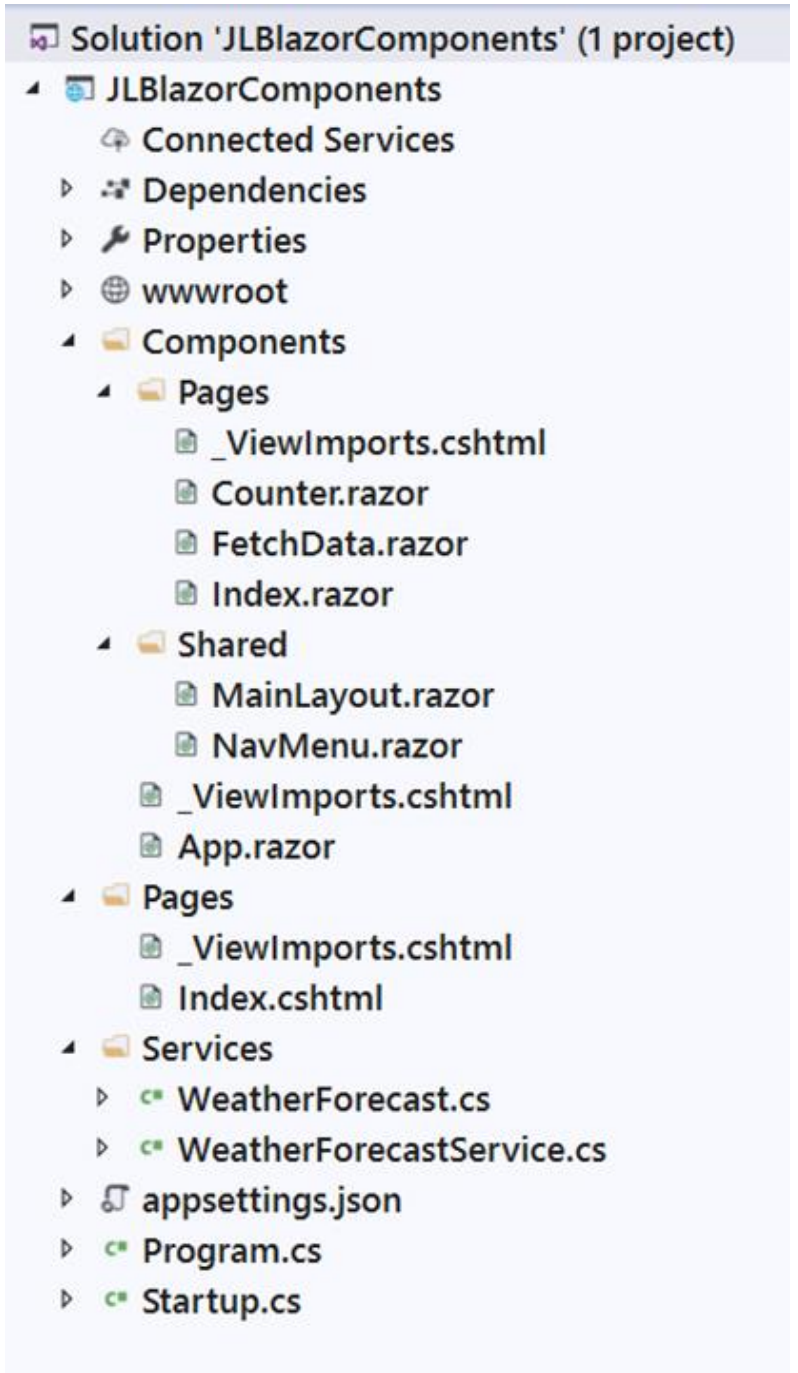


Blazor server-side

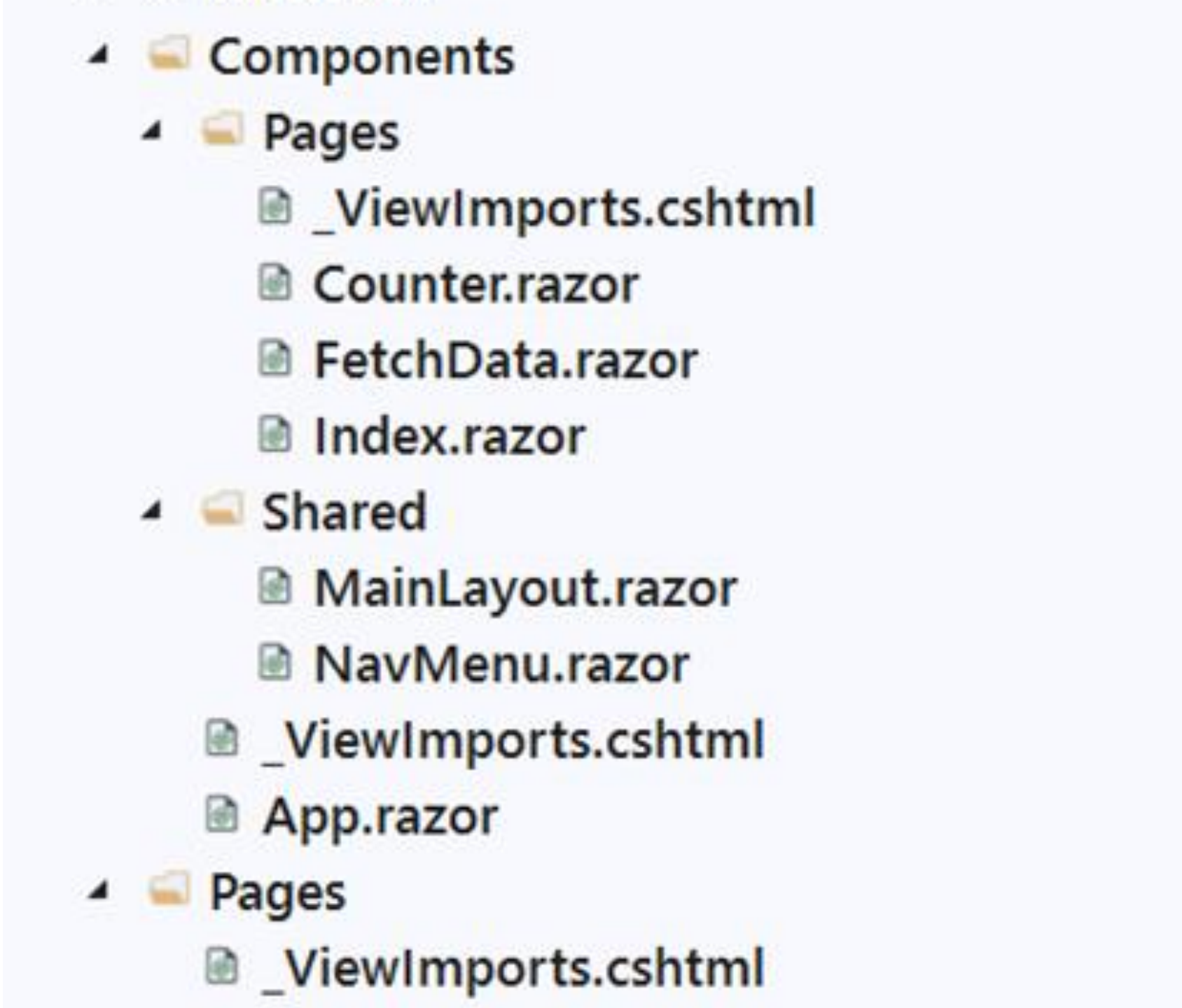


Blazor server-side

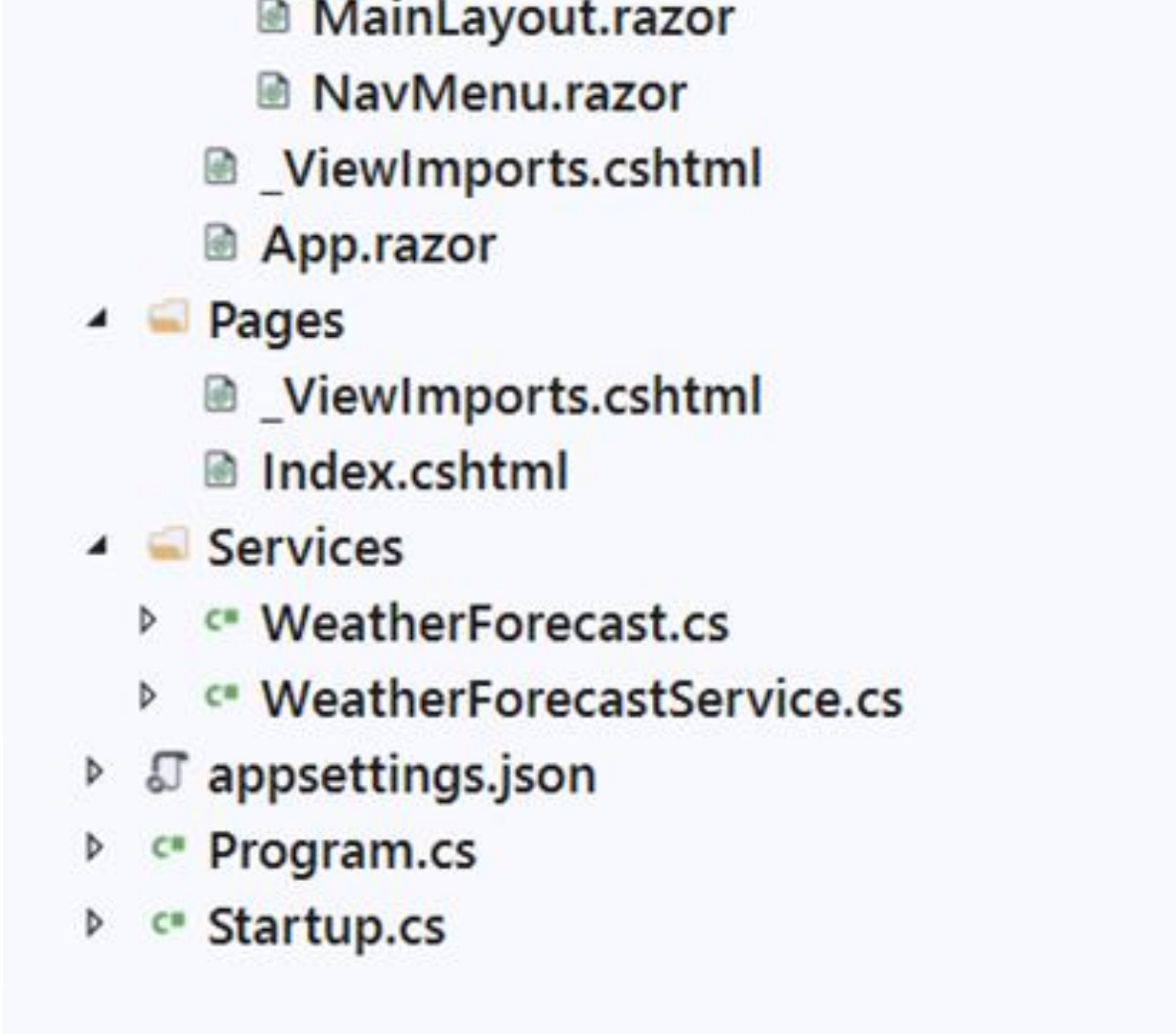
```
C:\Projects> dotnet new razorcomponents -n JLBlazorComponents
```



Blazor server-side

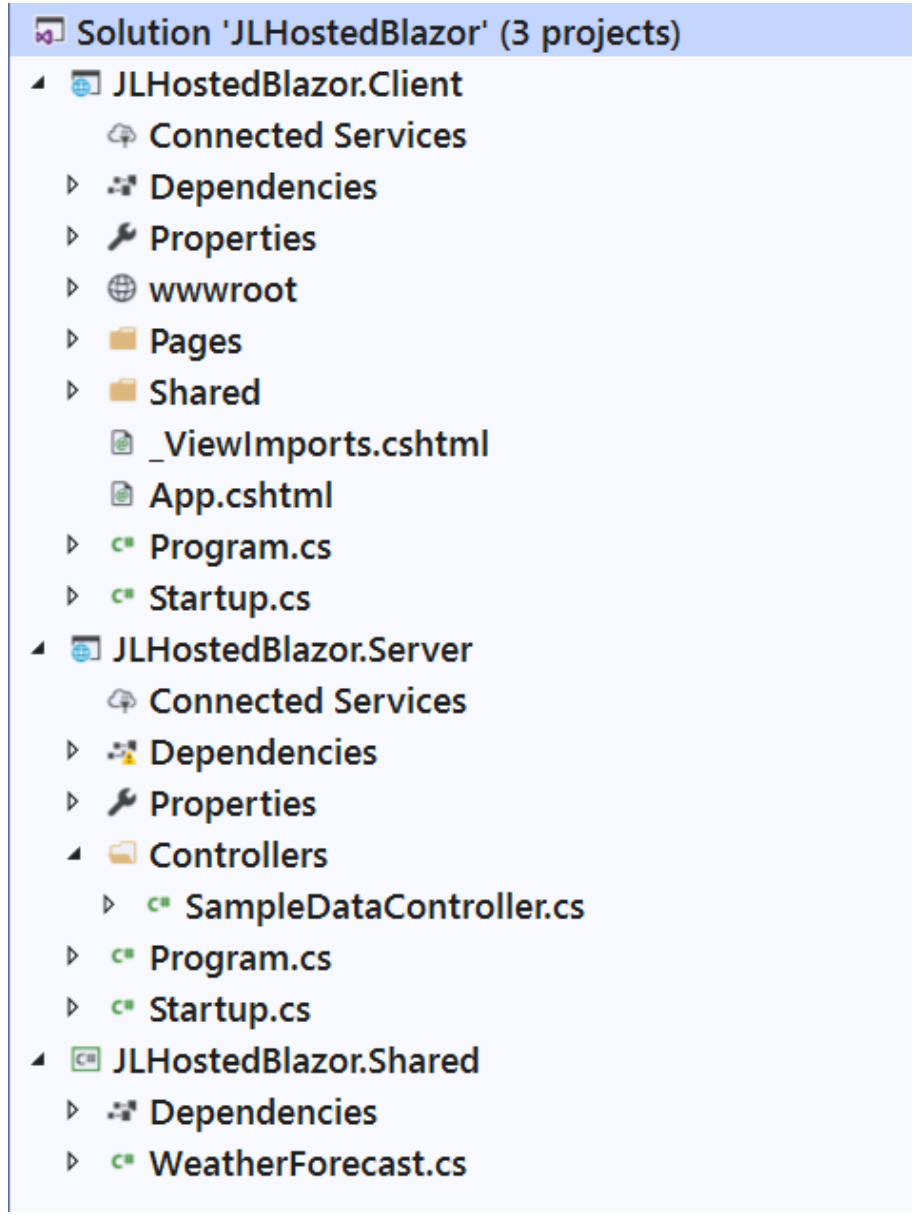


Blazor server-side

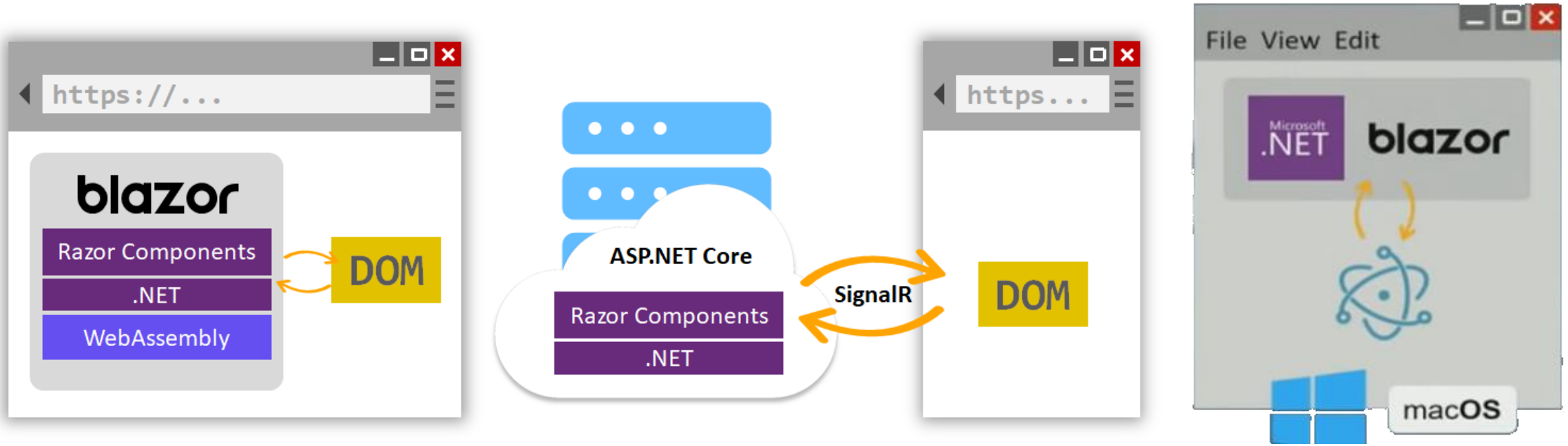


Blazor hosted

```
C:\Projects> dotnet new
blazorhosted -n JLHostedBlazor
```



Blazor: possibilities



https://www.youtube.com/watch?v=qrf3OrLHeFI
Steve Sanderson - Blazor, a New .NET Single Page Application Framework | Øredev 2018

What’s the future?

What WebAssembly are next to?

Loadtime improvements

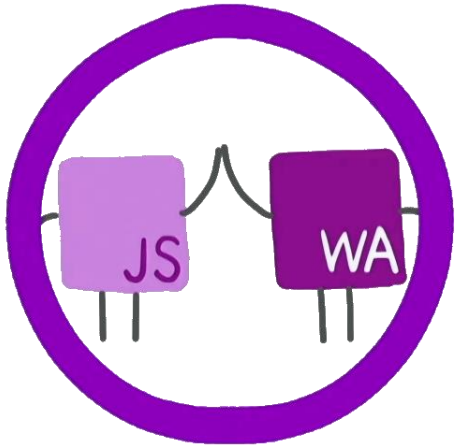
- streaming compilation
- tiered compilation
- implicit HTTP caching
- other improvements

Making use of modern Hardware

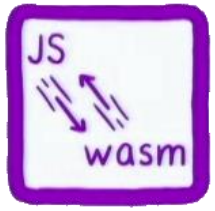
- threading
- SIMD
- wasm 64bit

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

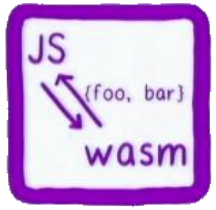
What WebAssembly are next to?



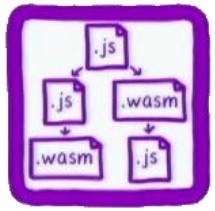
- fast calls
  JS <~> wasm



- esy and fast
  data exchange



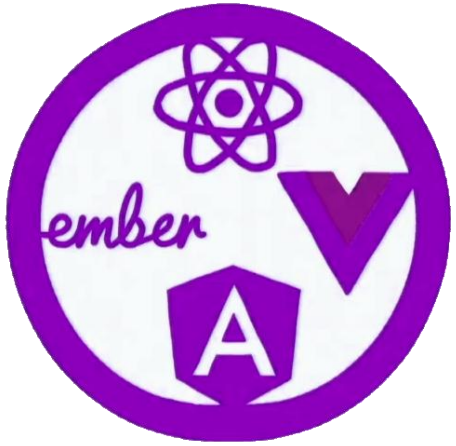
- ES module
  integration



- toolchain integration
  like npm or webpack

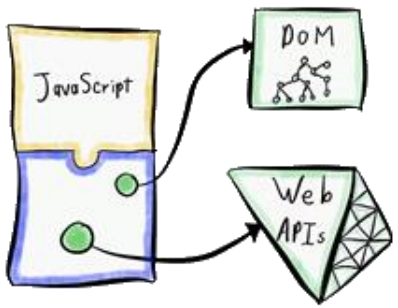
- backward compatibility

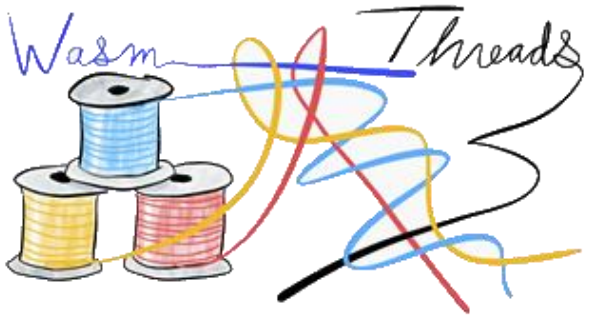
What WebAssembly are next to?



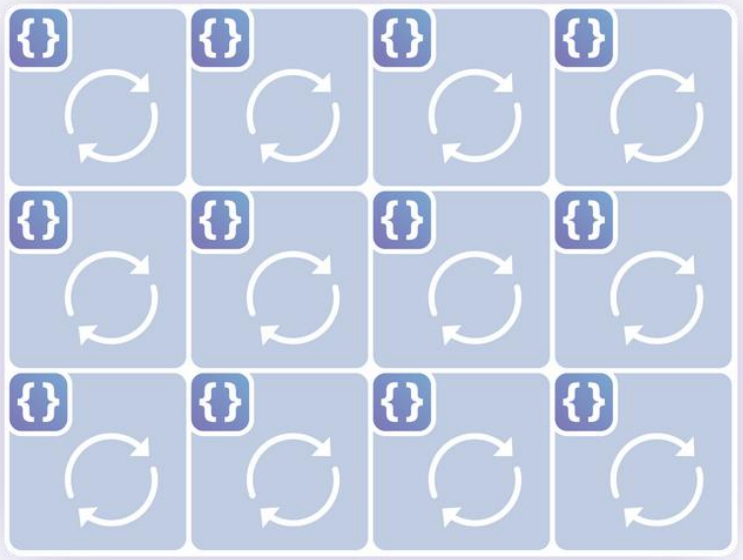
Hight level language features

- GC integration
- Exception handling
- Debugging
- Tail cals

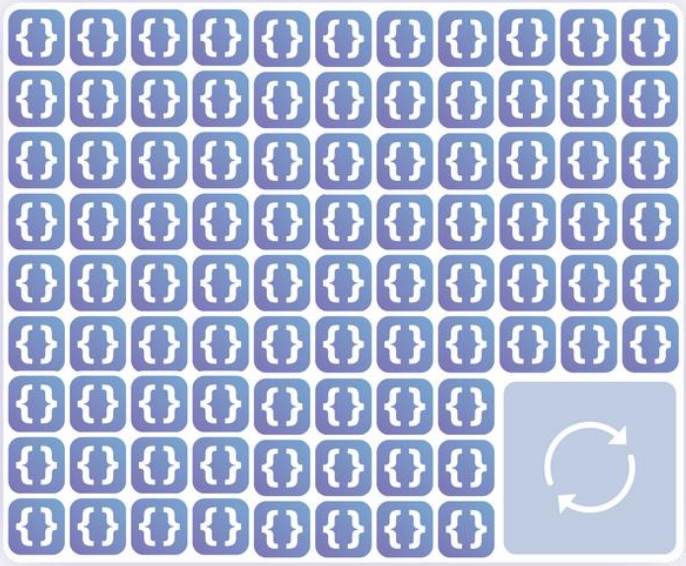




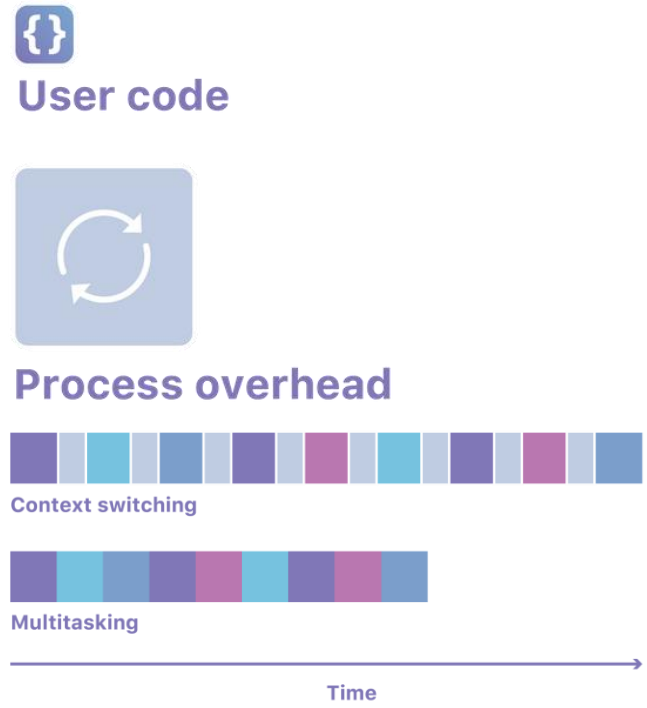
CloudFlare



Virtual machine

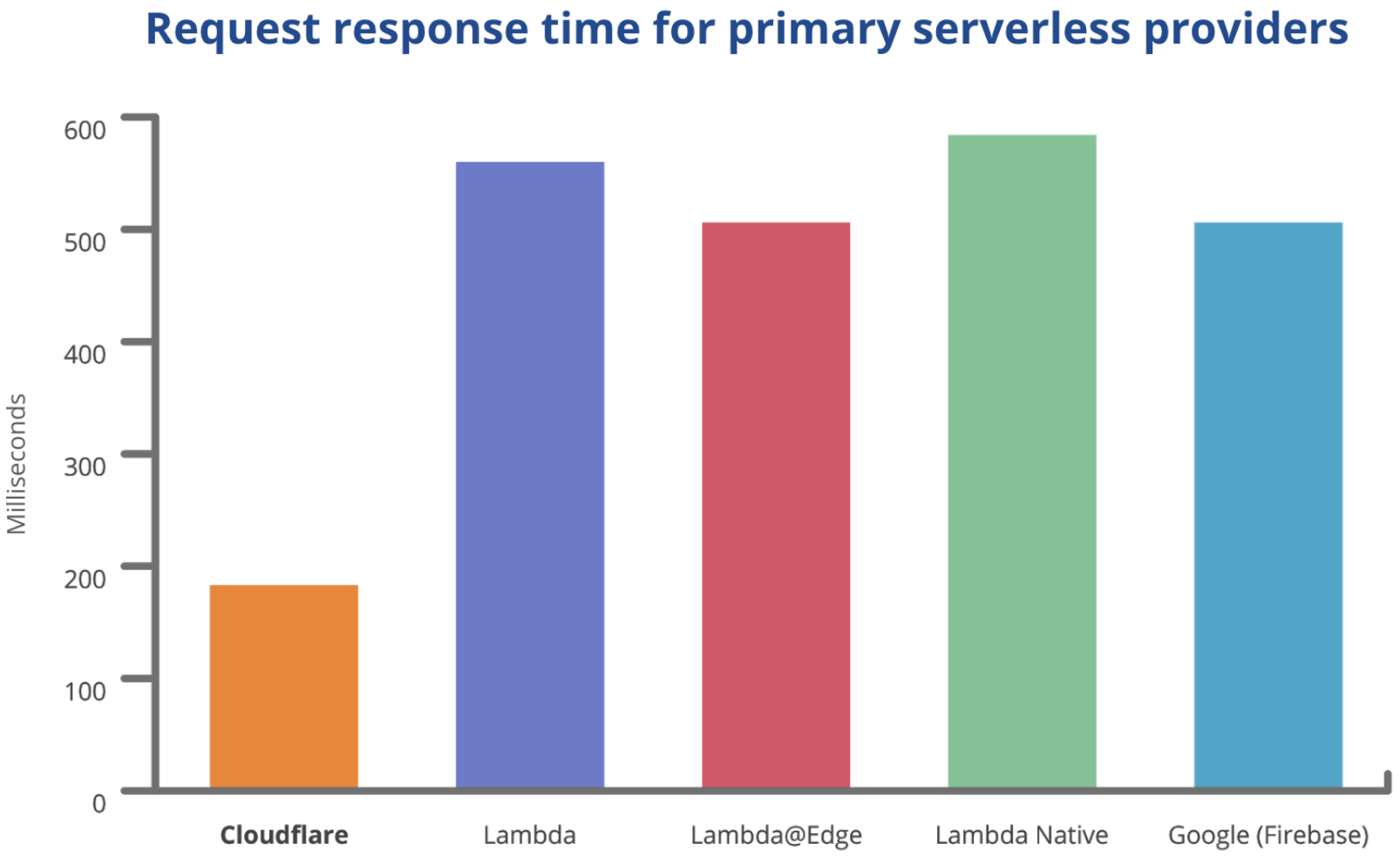


Isolate model



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

CloudFlare no more cold starts?



So WebAssembly?

Possibilities

Flexibility

Diversity

Cross-platform

Promise of bright tomorrow for todays projects

Questions??

Thank you ☺

Joanna Lamch

JLamch@gmail.com

JLamch.net

ProgramistkaIKot.pl

eventory

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

• 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

- 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.