

# WebAssembly



Hauptseminar SoSe 2020 (Mobile Technologien)

Alex Ruhl

31.07.2020

# Gliederung

1. Einführung
2. Architektur
3. Konzepte
4. Unterstützte Sprachen
5. Zukünftige Features
6. Code Beispiel
7. Verbreitung
8. Nutzungszwecke
9. Ausblick
10. Fazit

# 1. Einführung

Was ist WebAssembly?

- Binärinstruktionsformat für eine stackbasierte virtuelle Maschine

Wofür ist es gut?

- nahezu native Performance im Web

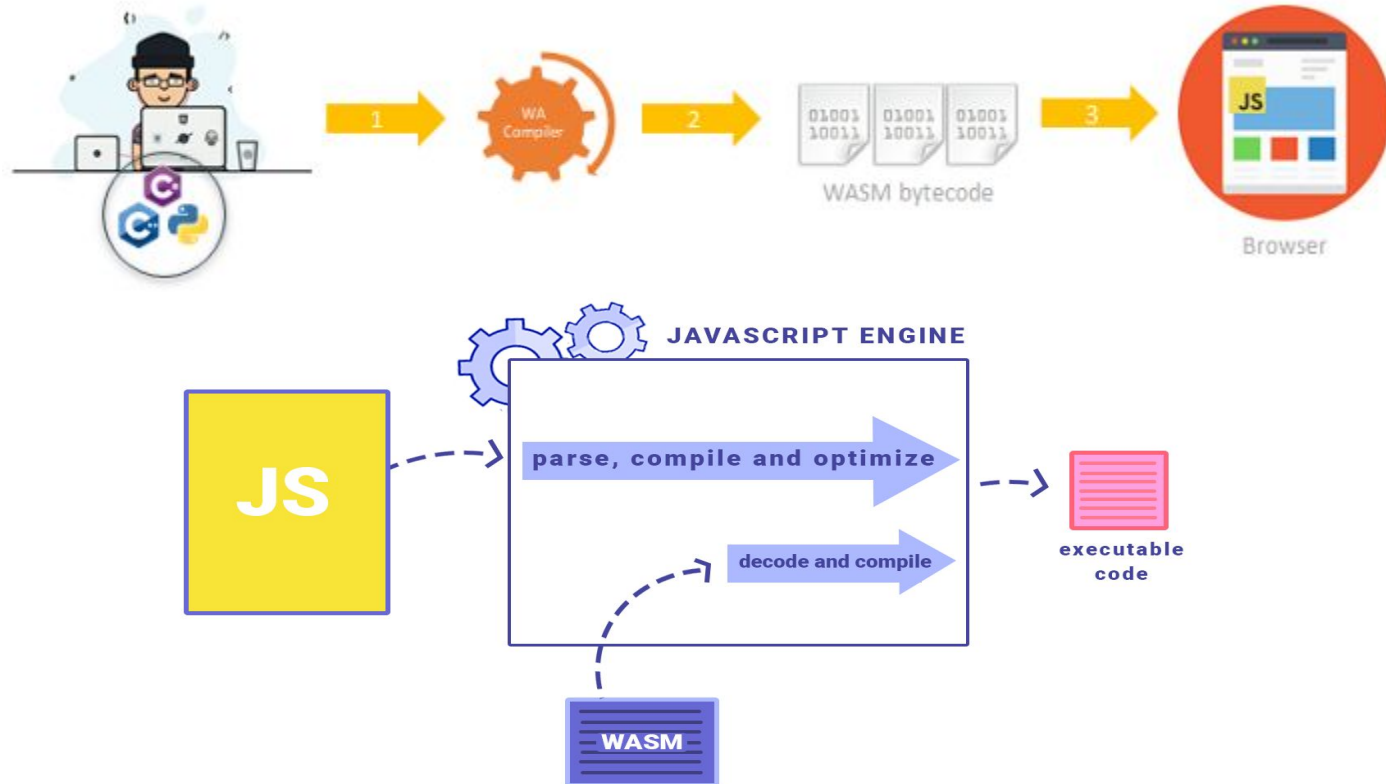
Ist es schon verwendbar?

- bereits von allen großen Browsern unterstützt

## 2. Architektur

- Binärinstruktionsformat
- virtuelle stackbasierte Maschine (VM)
- einfache Typen

## 2. Architektur



### 3. Konzepte

module "example1"

version

1

type section

#0	(i32, i32) → (i32)
#1	() → ()

import section

#0	"half" from "example2" of type 0
----	----------------------------------

function section

#1	type 1
#2	type 0

export section

"double" of type 0

start section

function #0

code section

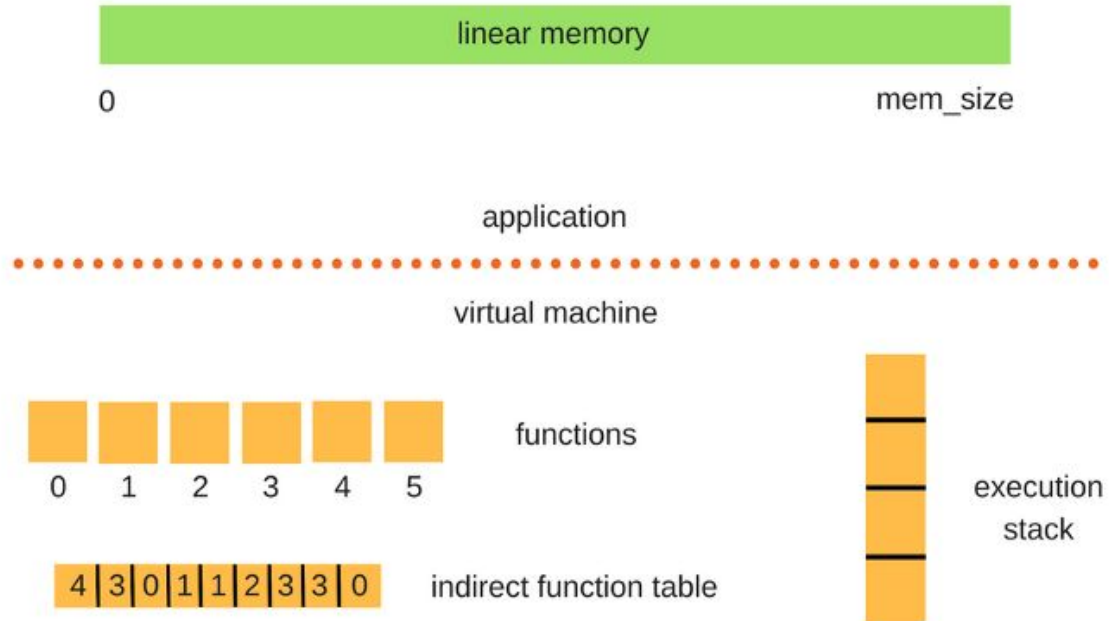
#0	i32.const 2 i32.load 2 0 0 call 1
#1	load_local 0 load_local 1 i32.mul

data section

0x5 0x0 0x0 0x0

...

### 3. Konzepte



## 4. Unterstützte Sprachen

- Anfangs nur C und C++ Dateien mit emscripten
- Momentan sind 17 Sprachen voll unterstützt
- WebAssembly verfügt über keine Garbage Collection
  - einige Sprachen sind deswegen nicht komplett unterstützt



## 5. Zukünftige Features

- Threads
  - vorgeschlagen und spezifiziert
- Garbage Collection
  - bereits vorgeschlagen

## 6. Code Beispiel

sub.c	sub.wat
<pre>int sub(int x, int y) {     return x - y; }</pre>	<pre>(module   (table 0 anyfunc)   (memory \$0 1)   (export "memory" (memory \$0))   (export "sub" (func \$sub))   (func \$sub (; 0 ;) (param \$0 i32) (param \$1 i32) (result i32)     (i32.sub       (get_local \$0)       (get_local \$1)     )   ) )</pre>

## 6. Code Beispiel

example.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <script>
      WebAssembly.instantiateStreaming(fetch('sub.wasm'))
        .then(prog => {
          console.log(prog.instance.exports.sub(5, 2));
        });
    </script>
  </body>
</html>
```

## 7. Verbreitung

- Daten aus dem Jahr 2019 von den Alexa Top 1 Millionen Webseiten
  - insgesamt nur 1639 Internetseiten erfasst die 1950 WebAssembly-Module verwendet haben
- große Projekte setzen auf WebAssembly
  - z. B. AutoCAD, QT und Google Earth

## 8. Nutzungszwecke

- rechenintensive Aufgaben
- bestehende Codebasis im Browser verwendbar machen
  
- auf den Alexa Top 1 Millionen Webseiten jedoch häufig für böartige Zwecke verwendet

## 9. Ausblick

- bereits offizieller Webstandard
- harmonie zwischen JavaScript und WebAssembly
- weitere Features wie GC und Threads werden WebAssembly noch attraktiver machen
- vermutlich werden die Nutzungszahlen in den nächsten Jahren ansteigen

# 10. Fazit

- sehr gute Spezifikation
- sehr performant
- gut durchdachte Konzepte
- es fehlen nur noch entsprechende Nutzungszahlen
- leichte Einarbeitung aber umfangreich

# Quellen

- [https://cdn-images-1.medium.com/max/1600/0\\*xU7akQpF9KctXbQA.png](https://cdn-images-1.medium.com/max/1600/0*xU7akQpF9KctXbQA.png)
- [https://miro.medium.com/max/700/1\\*Bj1Chj-JQumttK7sIKf0BQ.png](https://miro.medium.com/max/700/1*Bj1Chj-JQumttK7sIKf0BQ.png)
- [https://miro.medium.com/max/700/0\\*QphcOVaiVC2YL7Jd](https://miro.medium.com/max/700/0*QphcOVaiVC2YL7Jd).
- <https://github.com/WebAssembly/proposals>
- <https://rsms.me/res/wasm/overview.svg>
- <https://www.sec.cs.tu-bs.de/pubs/2019a-dimva.pdf>
- <https://madewithwebassembly.com/all-projects/>
- <https://webassembly.org/>