WebAssembly: Bare Metal

{SE, IEM, HCC} Inside - 6. April 2022

Stefan Schöberl

SE Bachelor & Master
FH Hagenberg

Researcher and Senior Software Engineer
SCCH

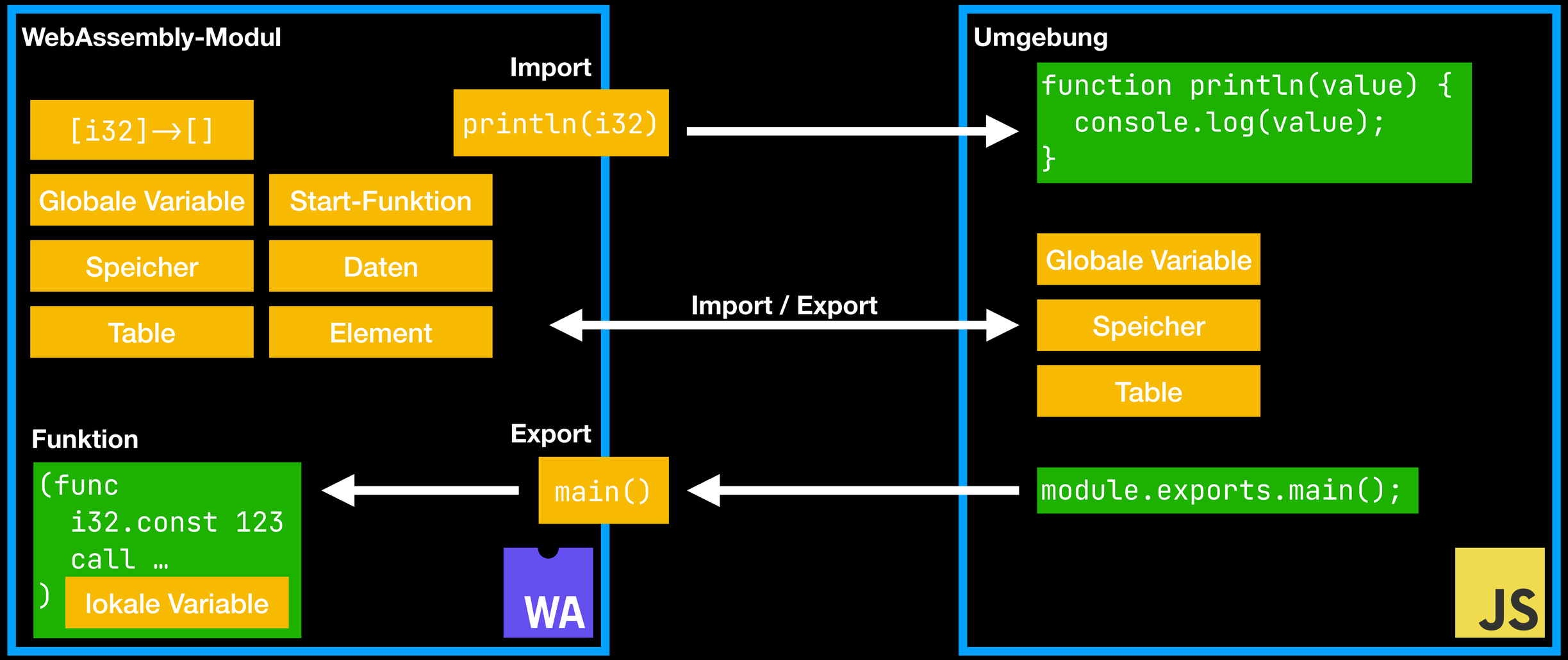
Nebenberuflich Lehrender
FH Hagenberg

Stefan Schöberl

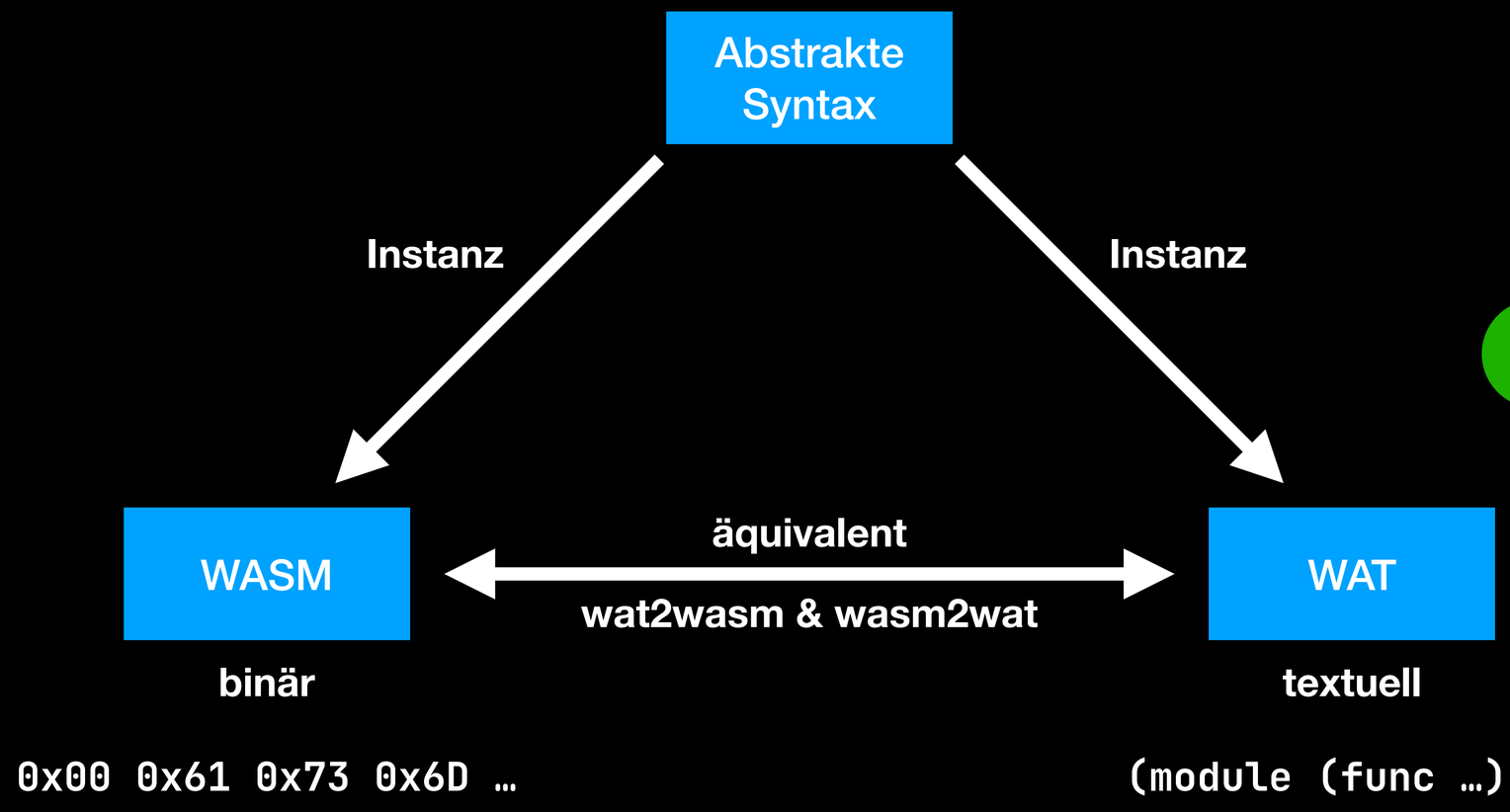
WebAssembly

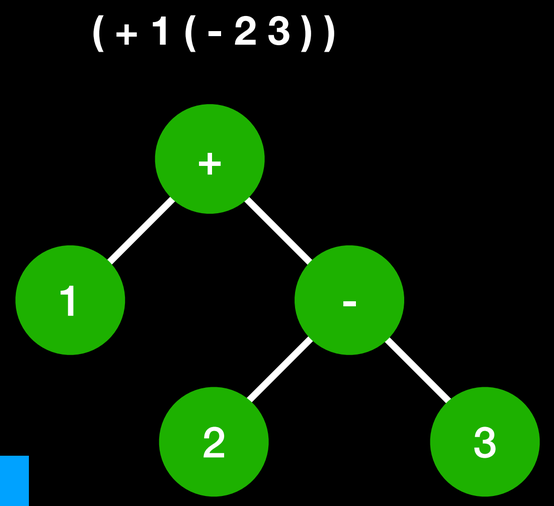
- Alternative zu JavaScript
- Effizienz
- NICHT Ersatz von JavaScript
- Sicherheitsmodell von JavaScript
- Code-Wiederverwendung
- W3C-Standard (Dezember 2019)

Laufzeitsystem



Modul-Format





aktuelle Einschränkungen

- kein Garbage Collector
- kein direkter Zugriff auf JS-Objekte und DOM
- kein Exception-Handling
- kein Multithreading
- 32 Bit Adressraum
- 4 Datentypen

https://webassembly.org/docs/future-features

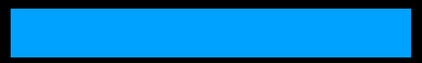
Stackmachine

a = 4 + 2 * b

Stack

ALU

Programm



Variablen

a	0
b	3

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

const 4

Programm

const 4
const 2
load b
mul
add
store a

Variablen

a	0
b	3

a = 4 + 2 * b

Stack

ALU

Variablen

a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

Variablen

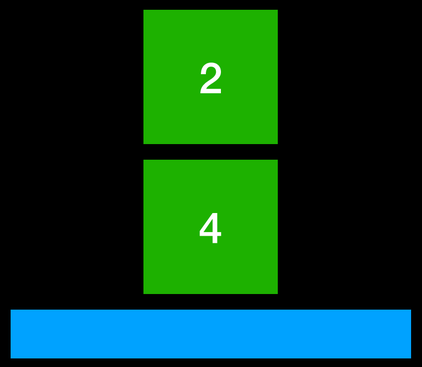
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

const 2

Variablen

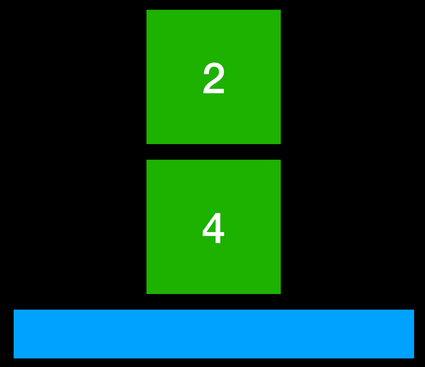
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

load b

Variablen

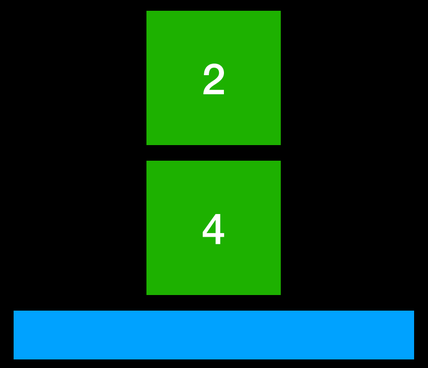
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

Variablen

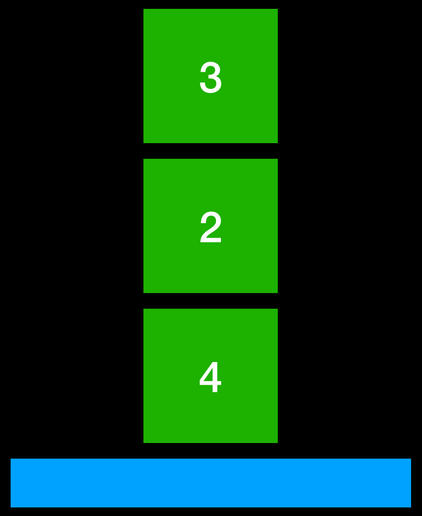
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

load b

Variablen

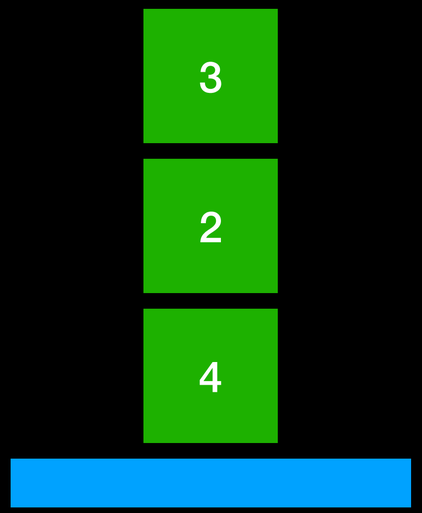
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

Variablen

a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

mul

2 * 3 = 6

Variablen

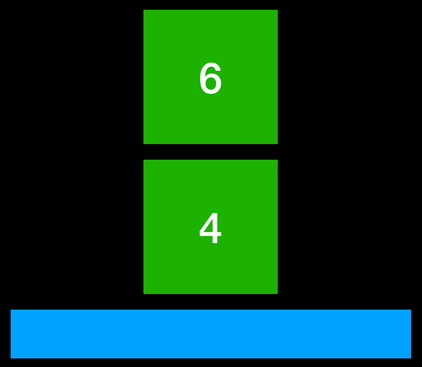
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

Variablen

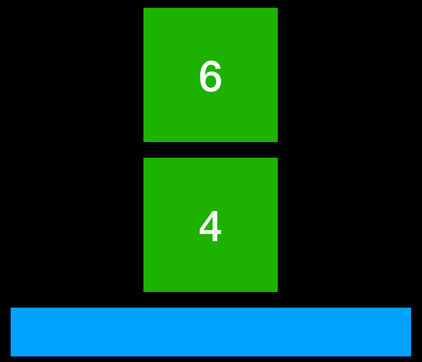
a	0
b	3

Programm

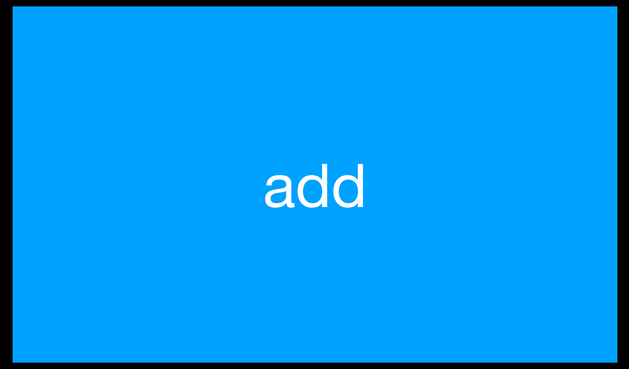
const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU



Variablen

a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

add

4 + 6 = 10

Programm

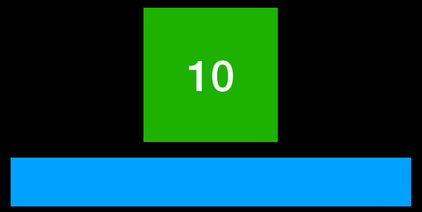
const 4
const 2
load b
mul
add
store a

Variablen

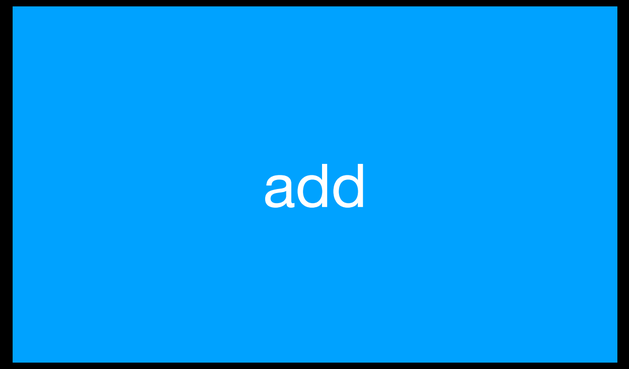
a	0
b	3

a = 4 + 2 * b

Stack



ALU



Variablen

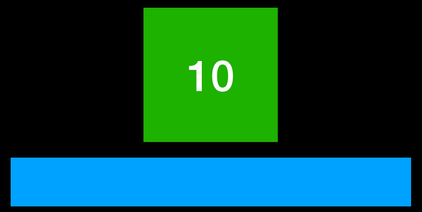
a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack



ALU

store a

Variablen

a	0
b	3

Programm

const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

store a

10

Programm

const 4
const 2
load b
mul
add
store a

Variablen

a	0
b	3

a = 4 + 2 * b

Stack

ALU

store a

10

Variablen

a	0
b	3

Programm

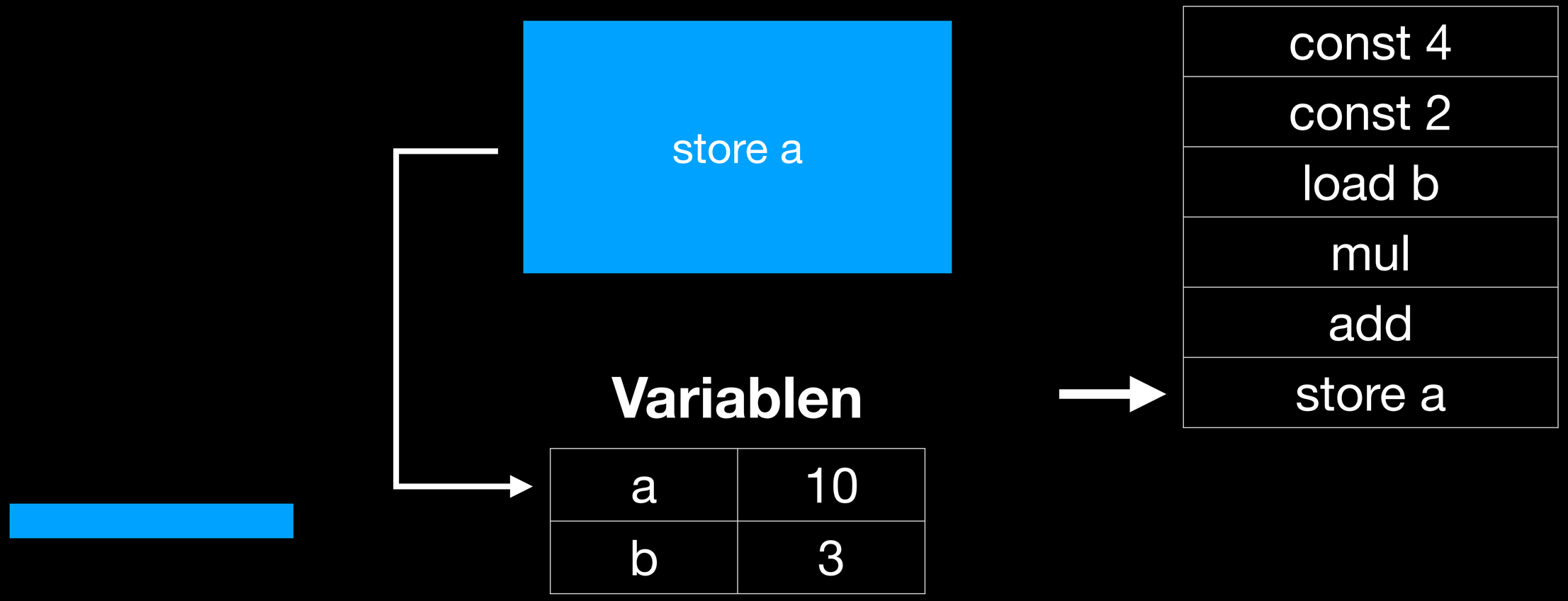
const 4
const 2
load b
mul
add
store a

a = 4 + 2 * b

Stack

ALU

Programm



Live-Demo
01-variables

Instruktionen

nop	[ ] → [ ]
i32.const *i32*	[ ] → [i32]
i32.add	[i32 i32] → [i32]
if [$t^?$]	[i32] → [t*]
local.get *x*	[ ] → [i32]
local.set *x*	[i32] → [ ]
call *x*	[t_1*] → [t_2*]
i32.load *memarg*	[i32] → [i32]
i32.store *memarg*	[i32 i32] → [ ]

https://webassembly.github.io/spec/core/appendix/index-instructions.html

Fibonacci

Fibonacci

```
int fib(int n) {
  if (n <= 1) {
    return n;
  } else {
    return fib(n - 1) + fib(n - 2);
  }
}
```

0 1 1 2 3 5 8 13 21 ...

Live-Demo

02-fibonacci

Schleifen

Schleifen

```
while (i < 10) {
  println(i);
  i = i + 1;
}
```

```
begin:
if (!(i < 10)) {
  goto end;
}

println(i);
i = i + 1;

goto begin;
end:
```

Schleifen

```
begin:
if (!(i < 10)) {
  goto end;
}

println(i);
i = i +1;

goto begin;
end:
```

```
block {
  loop {
    if (!(i < 10)) {
      goto 1;
    }

    println(i);
    i = i + 1;
    goto 0;
  }
}
```

Schleifen

```
block {
  loop {
    if (!(i < 10)) {
      goto 1;
    }

    println(i);
    i = i + 1;
    goto 0;
  }
}
```

```
block
  loop
    local.get $i     ;; i < 10
    i32.const 10
    i32.lt_s

    i32.eqz
    br_if 1

    local.get $i     ;; println(i)
    call $println

    local.get $i     ;; i = i + 1;
    i32.const 1
    i32.add
    local.set $i

    br 0
  end
end
```

Maximum im Array

Maximum im Array

```
int findMax(int[] array, int length) {
  int max = array[0];

  int i = 1;
  while (i < length) {
    if (array[i] > max) {
      max = array[i];
    }
    i = i + 1;
  }

  return max;
}
```

Live-Demo

03-maximum

schoeberl.dev

Grafiken

- https://commons.wikimedia.org/wiki/File:Web_Assembly_Logo.svg
- https://commons.wikimedia.org/wiki/File:Safari_browser_logo.svg
- https://de.wikipedia.org/wiki/Datei:Microsoft_Edge_logo_(2019).svg
- https://commons.wikimedia.org/wiki/File:Firefox_logo,_2019.svg
- https://commons.wikimedia.org/wiki/File:Google_Chrome_icon_(September_2014).svg
- https://commons.wikimedia.org/wiki/File:JavaScript-logo.png
- https://carbon.now.sh (Codeformatierung)