WebAssembly: Disassembled

Stefan Schöberl
schoeberl.dev
stefanschoeberl

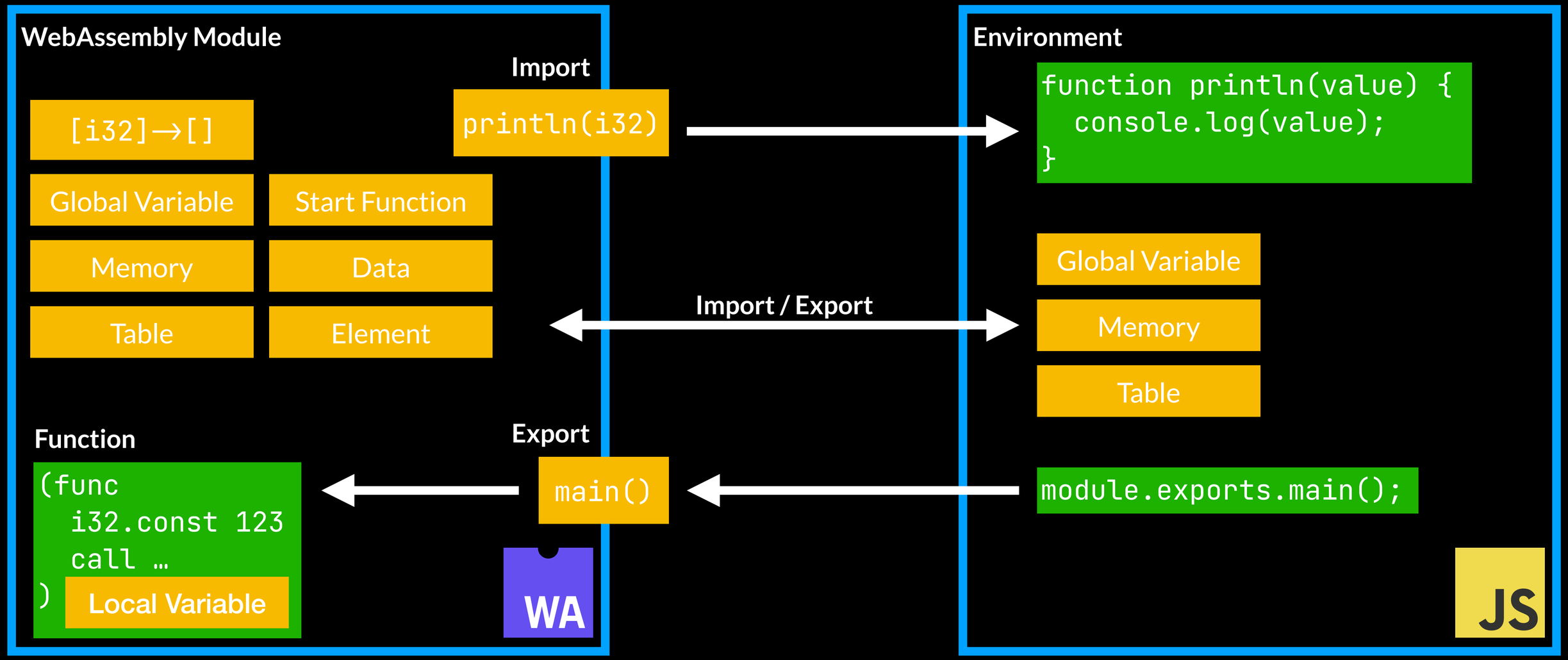
scch { } Software Competence Center Hagenberg

WebAssembly

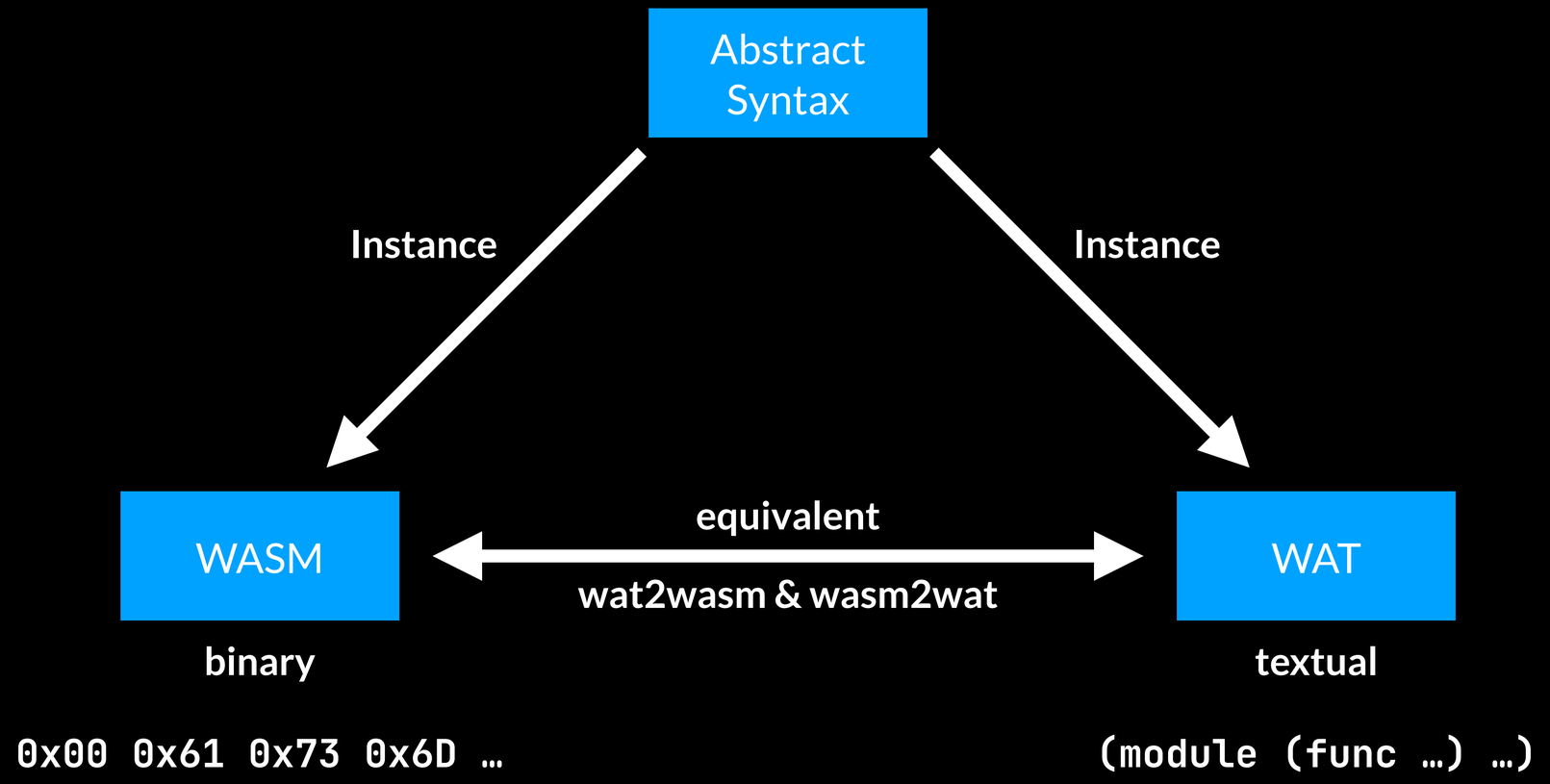
- Complement to JavaScript - not replacement
- Efficiency
- Security model of JavaScript
- Code reusability
- W3C standard
- Ongoing development




Big Picture


WebAssembly Module
Import
[i32]→[]
println(i32)
Global Variable
Start Function
Memory
Data
Table
Element
Function
(func
i32.const 123
call …
)
Local Variable
Export
main()
WA
Environment
function println(value) {
console.log(value);
}
Global Variable
Memory
Table
Import / Export
module.exports.main();
JS

Module Structure



Stack machine

$a = 4 + 2 * b$

Stack

ALU

Program

const 4
const 2
load b
mul
add
store a

Variables

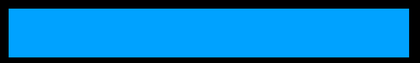
a	0
b	3

$a = 4 + 2 * b$

Stack

ALU

Program



```
const 4
```

const 4
const 2
load b
mul
add
store a

Variables

a	0
b	3

$a = 4 + 2 * b$

Stack

ALU

```
const 4
```

Program

```
const 4
const 2
load b
mul
add
store a
```

Variables

a	0
b	3

$a = 4 + 2 * b$

Stack

ALU

```
const 2
```

Variables

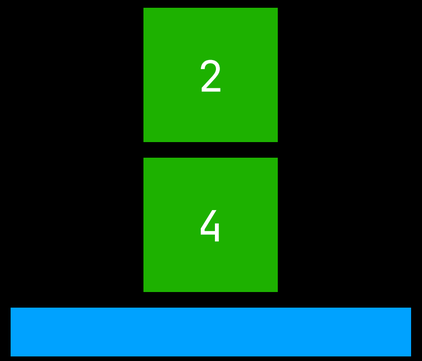
a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack



ALU

```
const 2
```

Variables

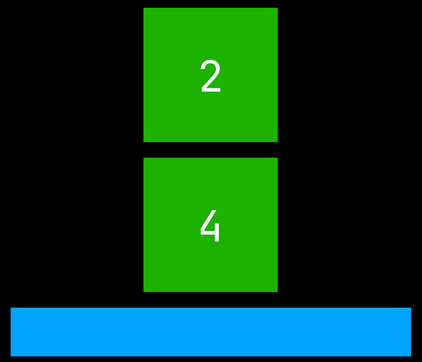
a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack



ALU

```
load b
```

Variables

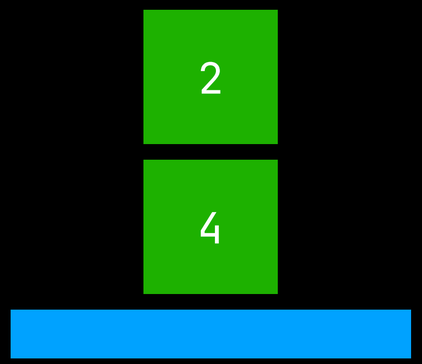
a	0
b	3

Program

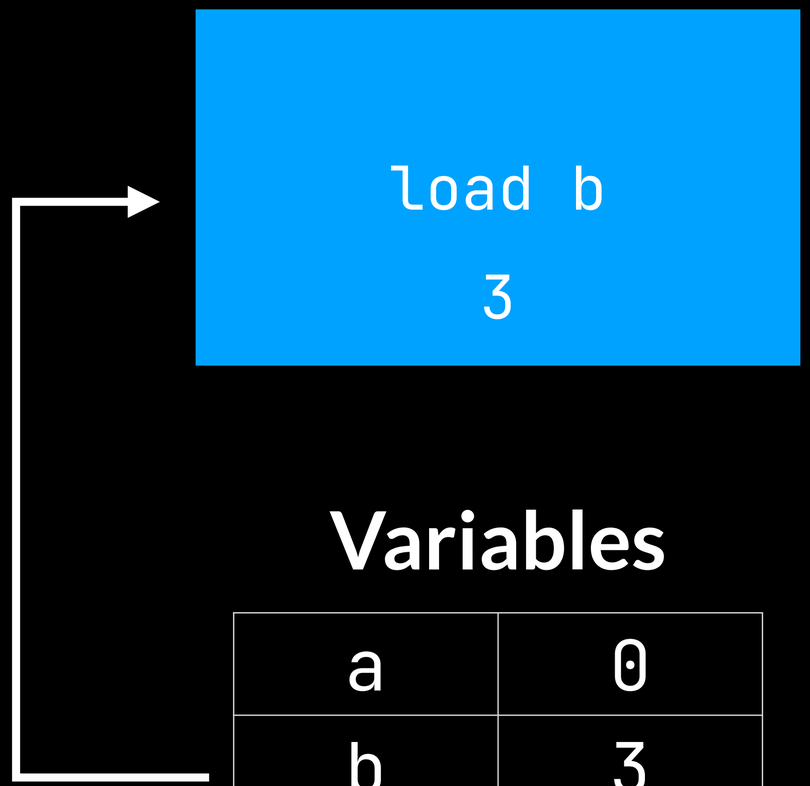
```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack



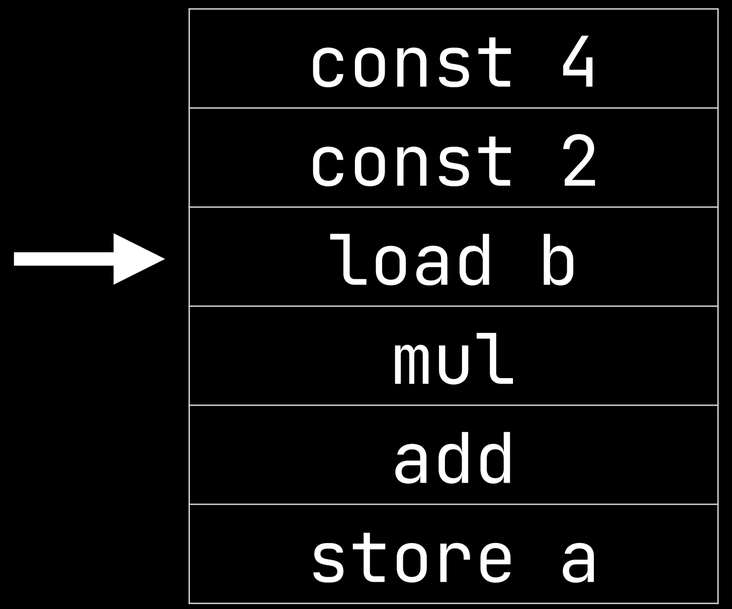
ALU



Variables

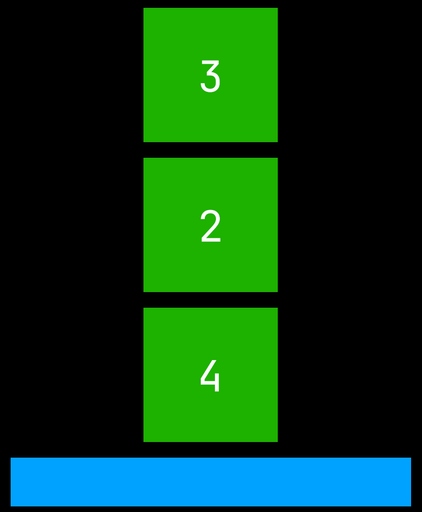
a	0
b	3

Program



$a = 4 + 2 * b$

Stack



ALU

```
load b
```

Variables

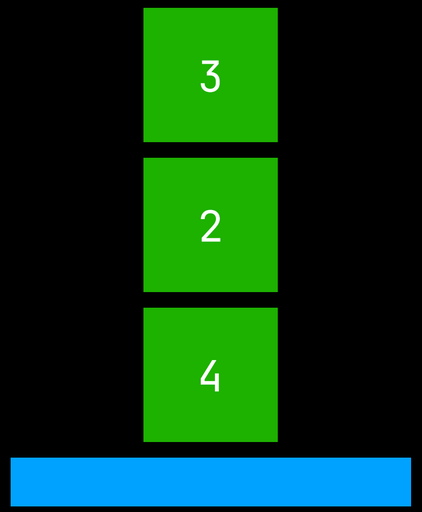
a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack



ALU

mul

Variables

a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack

ALU

```
mul
2 * 3 = 6
```

Program

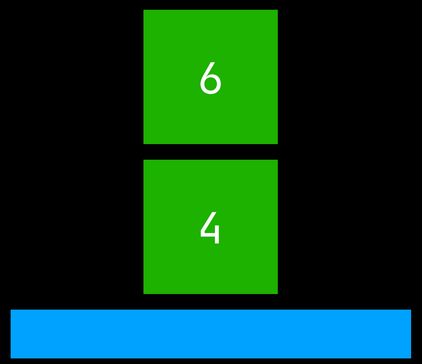
```
const 4
const 2
load b
mul
add
store a
```

Variables

a	0
b	3

$a = 4 + 2 * b$

Stack



ALU

mul

Program

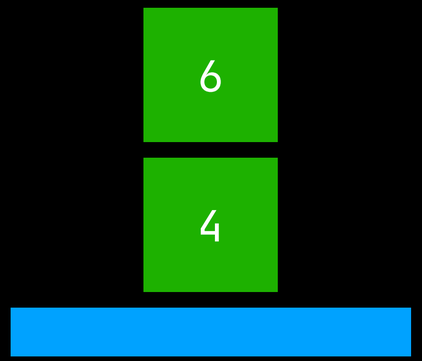
const 4
const 2
load b
mul
add
store a

Variables

a	0
b	3

$a = 4 + 2 * b$

Stack



ALU

add

Variables

a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack

ALU

```
add
4 + 6 = 10
```

Program

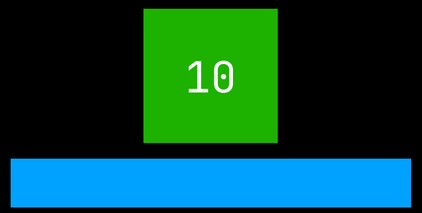
const 4
const 2
load b
mul
add
store a

Variables

a	0
b	3

$a = 4 + 2 * b$

Stack



ALU

add

Variables

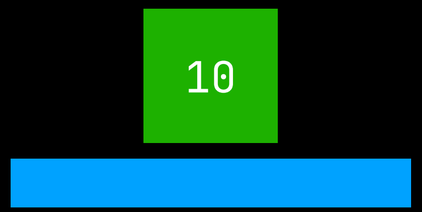
a	0
b	3

Program

const 4
const 2
load b
mul
add
store a

$a = 4 + 2 * b$

Stack



ALU

```
store a
```

Variables

a	0
b	3

Program

```
const 4
const 2
load b
mul
add
store a
```

$a = 4 + 2 * b$

Stack

ALU

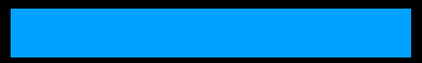
```
store a
10
```

Program

```
const 4
const 2
load b
mul
add
store a
```

Variables

a	0
b	3

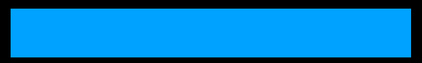


$a = 4 + 2 * b$

Stack

ALU

Program



Variables

a	0
b	3

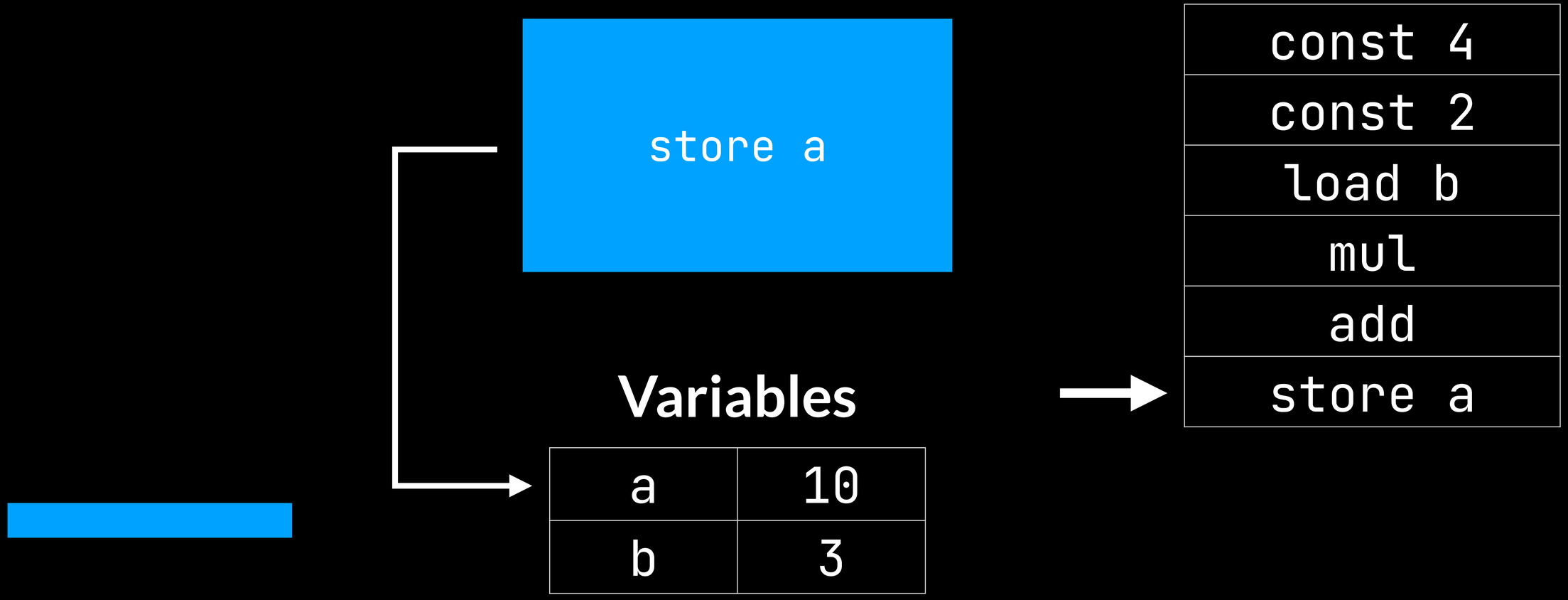
const 4
const 2
load b
mul
add
store a

$a = 4 + 2 * b$

Stack

ALU

Program



$a = 4 + 2 * b$

Stack

ALU

Program

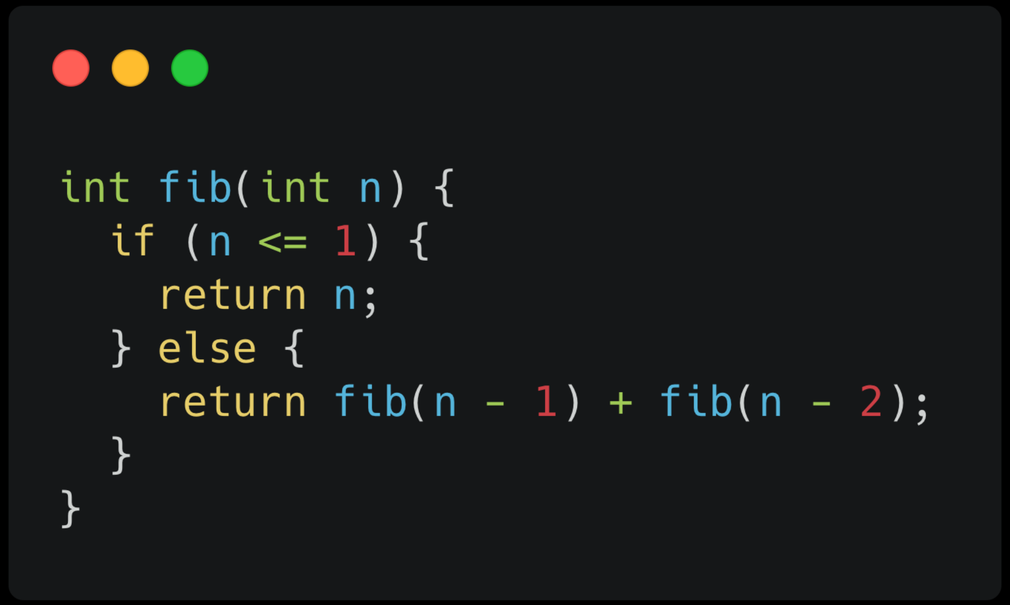
const 4
const 2
load b
mul
add
store a

Variables

a	10
b	3

Fibonacci numbers

Fibonacci numbers



```
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 Coding
01-fibonacci

Loops

Loops

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

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

println(i);
i = i + 1;

goto begin;
end:
```

Loops

```
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;
  }
}
```

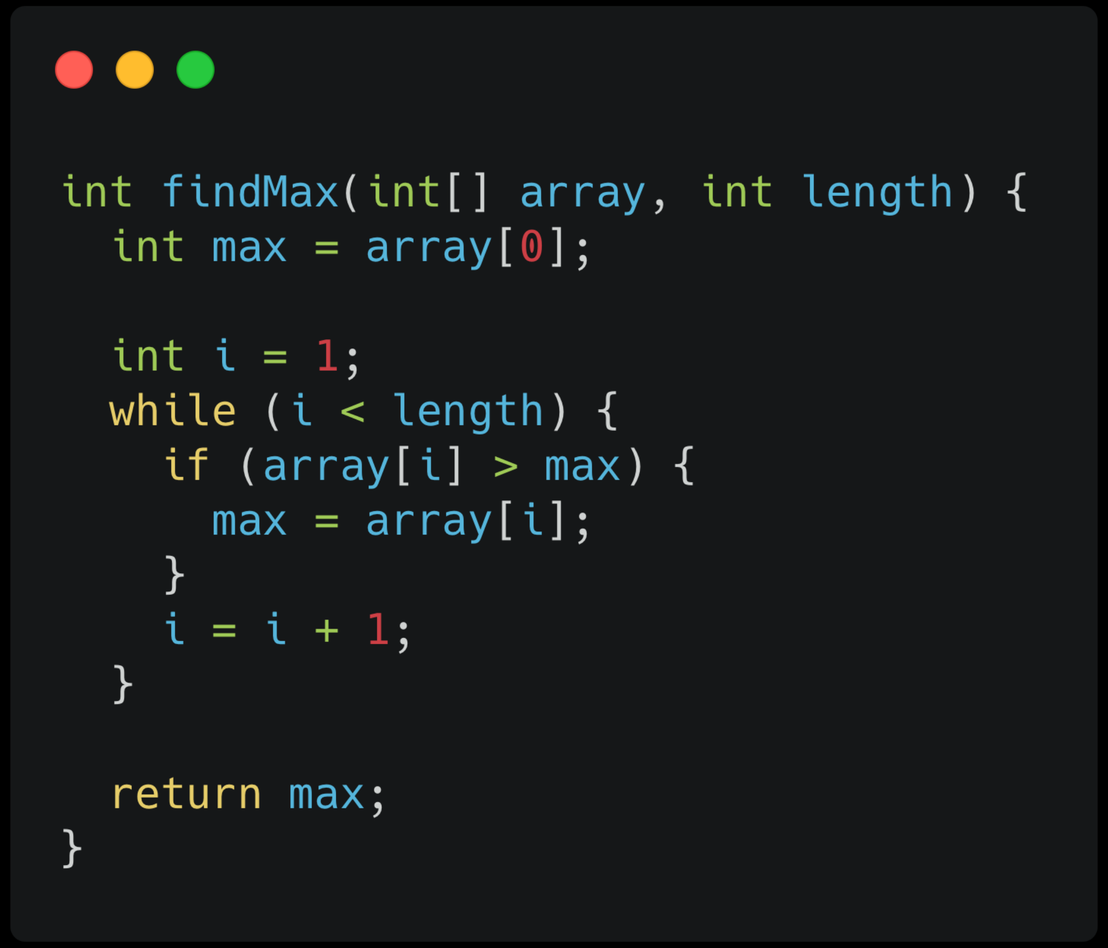
Loops

```
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 value in an array

Maximum value in an 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 Coding
02-maximum

WebAssembly: Disassembled

Stefan Schöberl
schoeberl.dev
stefanschoeberl

scch { } Software Competence Center Hagenberg

Images

- 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 (Code formatting)