

Web Assembly

Sam Sartor

February 14, 2018

Mines Linux Users Group



Client Code

Interactive Websites

Is it possible to make websites interactive?

We would have to include some instructions along with the document. We need some kind of scripting language for webpages.





JavaScript

On a dark and stormy night ...

The year is 1995. Netscape Communications Corporation is dying. In a frantic attempt to one-up Microsoft, the company decides to embed a scripting language into the Netscape browser.

They give **Brendan Eich** 10 days to make a prototype.

Eich dreams of Scheme. He wants his new language to be elegant, fast, and pure. **But it is too late.** The lawyers at Netscape have made a deal with Sun Microsystems.

It will be known as **JavaScript**.

!![]+!![]



This page uses 800 thousand lines of JavaScript

Multiproject Workspace

label cart

DASHBOARD HELP & UPDATES LISA DEMO

SHOP WEB Current/Backlog

16 Aug - Current Pts: 0 of 20

- Admin should be able to import multiple new products from CSV file (USA C, CE) **Accept** **Reject**
- Admin should be able to create new product (LD, CE) **Deliver**
- Admin should be able to login (USA C) **Finish**
- Shopper should see list of products, with primary photo as thumbnail shopping **Finish**
- Shopper should be able to add product to shopping cart (USA C) cart, shopping **Finish**
- Shopper should be able to click on a product, and see all product details, including photos shopping **Finish**
- Admin should be able to upload multiple product photos and mark one as the primary (USA C) admin, blocked **Finish**
- Admin should be able to upload product photo (CE, USA C) admin **Finish**

Basic Admin Features

- another bug created by groovy (CE) **Accept** **Reject**
- Shopper should be able to compare products (CE, USA C) shopping **Accept** **Reject**

17 25 Aug Pts: 10

- Inventory status endpoint api, inventory **Start**
- Product browsing should be paginated, with 10 products per page (USA C) needs discussion, shopping **Start**
- Shopper should be able to view contents of shopping cart cart, shopping **Start**
- Shopper should be able to remove product from shopping cart cart, shopping **Start**
- Cart manipulation should be AJAXY cart, shopping **Start**
- Some product photos not scaled properly when browsing products shopping **Start**

Initial demo to investors

18 Aug - Current Pts: 0 of 10

- Shopper should see a sample shopping cart (USA C, CE, LD) cart **Finish**
- Some product photos not scaled properly when browsing products (USA C) shopping **Finish**
- Shopper should be able to recommend a product to a friend shopping **Start**
- configure solr for full text searching search **Start**
- Shopper should be able to search for product search, shopping **Start**
- Initial demo to investors **Finish**
- Shopper should be able to enter credit card information and shipping address checkout, shopping **Start**
- Integrate with payment gateway checkout, shopping **Start**
- When shopper submits order, authorize total product amount from payment gateway checkout, needs discussion, shopping **Start**

11 25 Aug Pts: 10

- If system fails to authorize payment amount, display error message to shopper checkout, shopping **Start**
- If authorization is successful, show order number and confirmation message to shopper checkout, shopping **Start**
- Send notification email of order placement to admin admin, checkout, shopping **Start**
- Shopper should be able to check status of order by entering name and order number orders **Start**
- Shopper should be able to ask question about order orders **Start**
- Admin can review all order questions and send responses to shoppers admin, orders **Start**
- Set up Engine Yard production environment deployment **Start**

Beta launch

1 Sep Pts: 4

- Shopper should be able to recommend a product to a friend shopping **Start**
- Shopper should be able to search for product search, shopping **Start**
- Shopper should be able to review products needs design, shopping **Start**
- Shopper should be able to reset forgotten password **Start**

MY SAMPLE PROJECT Current/Backlog

18 Aug - Current Pts: 0 of 10

- Shopper should see a sample shopping cart (USA C, CE, LD) cart **Finish**
- Some product photos not scaled properly when browsing products (USA C) shopping **Finish**
- Shopper should be able to recommend a product to a friend shopping **Start**
- configure solr for full text searching search **Start**
- Shopper should be able to search for product search, shopping **Start**
- Initial demo to investors **Finish**
- Shopper should be able to enter credit card information and shipping address checkout, shopping **Start**
- Integrate with payment gateway checkout, shopping **Start**
- When shopper submits order, authorize total product amount from payment gateway checkout, needs discussion, shopping **Start**

11 25 Aug Pts: 10

- If system fails to authorize payment amount, display error message to shopper checkout, shopping **Start**
- If authorization is successful, show order number and confirmation message to shopper checkout, shopping **Start**
- Send notification email of order placement to admin admin, checkout, shopping **Start**
- Shopper should be able to check status of order by entering name and order number orders **Start**
- Shopper should be able to ask question about order orders **Start**
- Admin can review all order questions and send responses to shoppers admin, orders **Start**
- Set up Engine Yard production environment deployment **Start**

Beta launch

1 Sep Pts: 9

- Shopper should be able to recommend a product to a friend shopping **Start**
- Shopper should be able to search for product search, shopping **Start**
- Shopper should be able to review products needs design, shopping **Start**
- Shopper should be able to reset forgotten password **Start**

labelcart

show 1 done story

- My Sample Project Shopper should see a sample shopping cart (USA C, CE, LD) cart **Finish**
- Shop Mobile Shopper should be able to select item and quantity (USA C) cart, requires web story **Deliver**
- Shop Mobile Shopper should be able to view items and quantity cart, shopping **Finish**
- Shop Mobile Shopper should be able to delete previously selected items cart, shopping **Start**
- Shop Web Shopper should be able to add product to shopping cart (USA C) cart, shopping **Finish**
- Shop Web Shopper should be able to view contents of shopping cart cart, shopping **Start**
- Shop Web Shopper should be able to remove product from shopping cart cart, shopping **Start**
- Shop Web Cart manipulation should be AJAXY cart, shopping **Start**



Emscripten

We need C

- People start writing games in JavaScript
- Why can't we use the Unity game engine?
- It's written in C
- Browsers don't run C
- Browsers only run JavaScript
- What if we compiled C to JavaScript?



Introducing...



emscripten

It compiles things to JavaScript!

asm.js Example

C

```
int factorial(int n) {  
    if (n == 0)  
        return 1;  
    else  
        return n *  
        ↪ factorial(n-1);  
}
```

asm.js

```
function __Z9factoriali($0) {  
    $0 = $0|0;  
    var $1 = 0, $10 = 0, $2 = 0, $3 = 0, $4 = 0, $5 = 0, $6  
    ↪ = 0, $7 = 0, $8 = 0, $9 = 0, label = 0, sp = 0;  
    sp = STACKTOP;  
    STACKTOP = STACKTOP + 16|0; if ((STACKTOP|0) >=  
    ↪ (STACK_MAX|0)) abortStackOverflow(16|0);  
    $2 = $0;  
    $3 = $2;  
    $4 = ($3|0)==(0);  
    if ($4) {  
        $1 = 1;  
    } else {  
        $5 = $2;  
        $6 = $2;  
        $7 = (($6) - 1)|0;  
        $8 = (__Z9factoriali($7)|0);  
        $9 = Math_imul($5, $8)|0;  
        $1 = $9;  
    }  
    $10 = $1;  
    STACKTOP = sp;return ($10|0);  
}
```

That was a Bad Idea

JavaScript was designed for...

- ✓ Crazy people
- Humans
- ✗ Computers

Why compile low-level → high-level?

Why don't we have *machine code for the web*?

```
80483b4: 55      push
80483b5: 89 e5    mov
80483b7: 83 e4    and
80483ba: 83 ec    sub
80483bd: c7 44    movl
80483c4: 00
80483c5: eb 11    jmp
80483c7: c7 04    movl
80483ce: e8 1d    call
80483d3: 83 44    addl
80483d8: 83 7c    cmpl
80483dd: 7e e8    jle
80483df: b8 00    mov
80483e4: c9      leave
80483e5: c3      ret
80483e6: 90      nop
80483e7: 90      nop
80483e8: 90      nop
80483e9: 90      nop
80483ea: 90      nop
```



Web Assembly

Which Direction?



emscripten can compile your code to *Web Assembly*!

WASM Example

C

```
int factorial(int n) {  
    if (n == 0)  
        return 1;  
    else  
        return n *  
            ↪ factorial(n-1);  
}
```

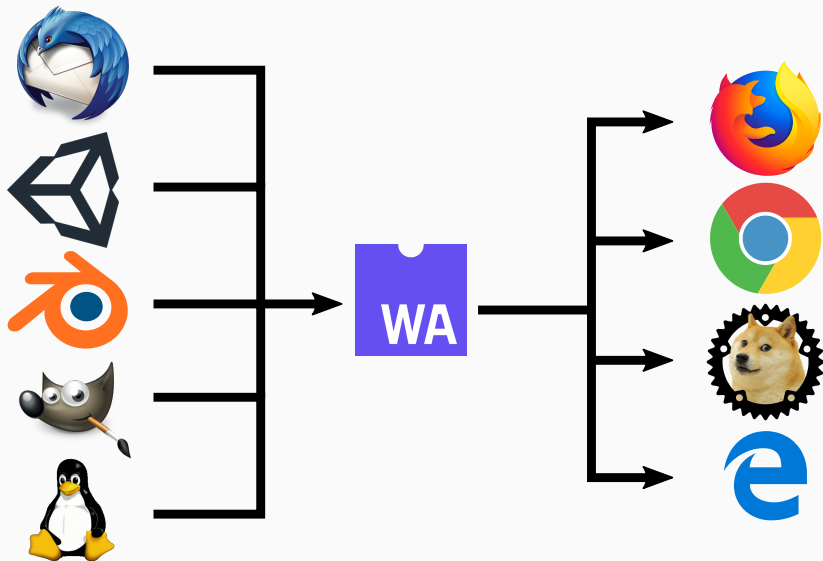
WASM Text

```
get_local 0  
i64.eqz  
if i64  
    i64.const 1  
else  
    get_local 0  
    get_local 0  
    i64.const 1  
    i64.sub  
    call 0  
    i64.mul  
end
```

WASM Binary

```
20 00  
50  
04 7E  
42 01  
05  
20 00  
20 00  
42 01  
7D  
10 00  
7E  
0B
```


Software in the Browser





**JAVASCRIPT
DISABLED**

Practical WASM



C/C++

We can use the emscripten sdk to compile C/C++ to JavaScript and WASM.

Emscripten can build a test HTML webpage, JavaScript file, or a pure wasm library.

```
$ emcc demo.c -s WASM=1 -o demo.html
```

```
$ emcc demo.c -s WASM=1 -o demo.js
```

```
$ emcc demo.c -s WASM=1 -s SIDE_MODULE=1 -o demo.wasm
```



Rust

The Rust Language

Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

```
fn main() {  
    let number = 13;  
    let fact = match number {  
        1 => "is one",  
        2 | 3 | 5 | 7 => "is prime",  
        13...19 => "is a teen",  
        _ => "ain't special",  
    };  
  
    println!("{}", number, fact);  
}
```

Rust Toolchains

Rust can be easily compiled to JavaScript and WebAssembly!

You can install a WASM toolchain by running

```
$ rustup toolchain add wasm32-*-*
```

The available toolchains are:

<code>wasm32-unknown-unknown</code>	compile to pure WASM
<code>wasm32-unknown-emsripten</code>	compile to a WASM executable
<code>asmjs-unknown-emsripten</code>	compile to JavaScript

Rust comes with a build system called **Cargo**.

```
$ cargo run
  Finished helloworld [unoptimized + debuginfo] target(s) in 0.0 secs
  Running `target/debug/helloworld`
Hello, world!
```

Cargo can target WASM!

```
$ cargo build --target wasm32-unknown-emscripen
  Compiling helloworld v0.1.0 (file:///home/sam/Code/helloworld)
  Finished dev [unoptimized + debuginfo] target(s) in 1.26 sec
$ cd target/wasm32-unknown-emscripen/debug/
$ node helloworld.js
Hello, world!
```




stdweb

Interact

The whole point is to make interactive websites. How can we edit HTML through Rust? How do we make Rust web development feasible?

Introducing...



Communicate

stdweb provides JavaScript's abilities within Rust:

```
let print_hello = |name: String| {  
    println!( "Hello, {}!", name );  
};  
  
js! {  
    var print_hello = @{print_hello};  
    print_hello("Bob");  
}
```

We can connect to buttons and other page elements:

```
let button = document().query_selector("#button").unwrap();  
button.add_event_listener(move |_: ClickEvent|  
    js!{ @{button}.style = "display: none;" }  
);
```

We can install the cargo-web tool to make Rust WASM even easier!

```
$ cargo web start
  Finished release [optimized] target(s) in 0.0 secs
  You can access the web server at `http://127.0.0.1:8000`.
```



parcel

Bundeling JS and Rust

We can use Parcel to bundle JS and WASM into the same project.

In `main.js`:

```
import {add} from './add.rs';  
console.log(add(2, 3));
```

In `add.rs`:


```
#[no_mangle]  
pub fn add(a: i32, b: i32) -> i32 {  
    a + b  
}
```



Putting it all Together

Conclusion

- Faster websites
- Written in any language
- With desktop libraries
- With native graphics
- With modern tools



```
2084  
15CA ↓  
17B4  
si, 7160  
di, di  
es, [760  
bx, 800F  
cx, cx  
bp, 0001  
dx, dx
```

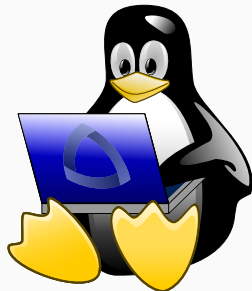



Questions?

Copyright Notice

This presentation was from the **Mines Linux Users Group**. A mostly-complete archive of our presentations can be found online at <https://lug.mines.edu>.

Individual authors may have certain copyright or licensing restrictions on their presentations. Please be certain to contact the original author to obtain permission to reuse or distribute these slides.



Colorado School of Mines
Linux Users Group