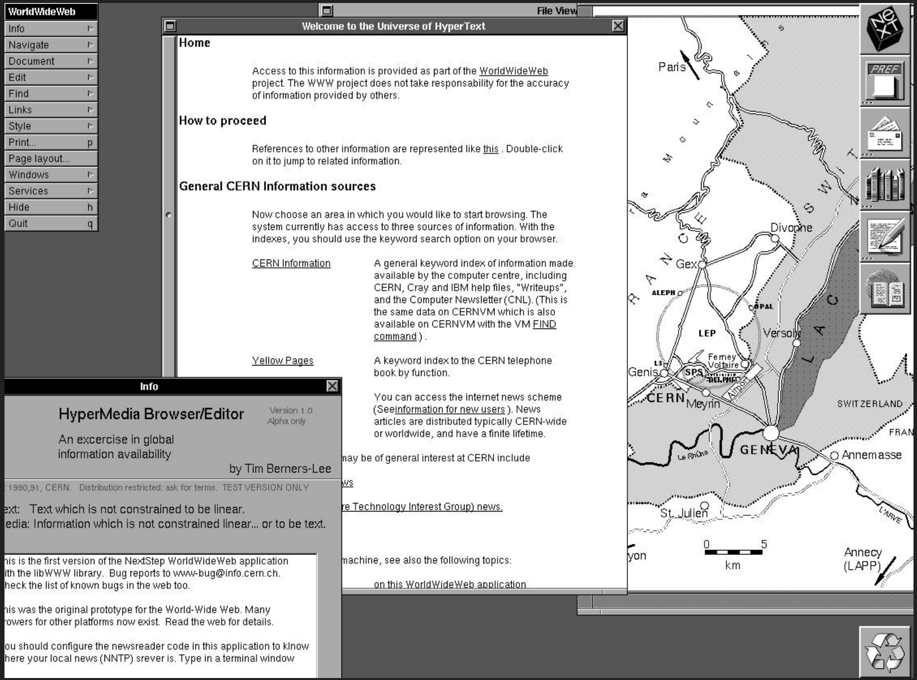
WASM for dummies

A very short primer in a series of acts

Divya_Mohan02

#FOSDEMFlashbacks



Welcome to the Universe of HyperText

Home

Access to this information is provided as part of the WorldWideWeb project. The WWW project does not take responsability for the accuracy of information provided by others.

How to proceed

References to other information are represented like this . Double-click on it to jump to related information.

General CERN Information sources

Now choose an area in which you would like to start browsing. The system currently has access to three sources of information. With the indexes, you should use the keyword search option on your browser.

CERN Information	A general keyword index of information made available by the computer centre, including CERN, Cray and IBM help files, "Writeups", and the Computer Newsletter (CNL). (This is the same data on CERNVM which is also available on CERNVM with the VM FIND command) .
Yellow Pages	A keyword index to the CERN telephone book by function.
	You can access the internet news scheme (Seeinformation for new users). News articles are distributed typically CERN-wide or worldwide, and have a finite lifetime.

may be of general interest at CERN include

ws

re Technology Interest Group) news.

machine, see also the following topics:

on this WorldWideWeb application

Info

HyperMedia Browser/Editor

Version 1.0 Alpha only

An excercise in global information availability

by Tim Berners-Lee

1990,91, CERN. Distribution restricted: ask for terms. TEST VERSION ONLY

ext: Text which is not constrained to be linear.
edia: Information which is not constrained linear... or to be text.

his is the first version of the NextStep WorldWideWeb application ith the libWWW library. Bug reports to www-bug@info.cern.ch. heck the list of known bugs in the web too.

his was the original prototype for the World-Wide Web. Many rowers for other platforms now exist. Read the web for details.

ou should configure the newsreader code in this application to know here your local news (NNTP) srever is. Type in a terminal window

The very first web browser, Credits: CERN

Divya_Mohan02

ACT I: JavaScript

JavaScript: The Gen Z (undefeated) compilation target

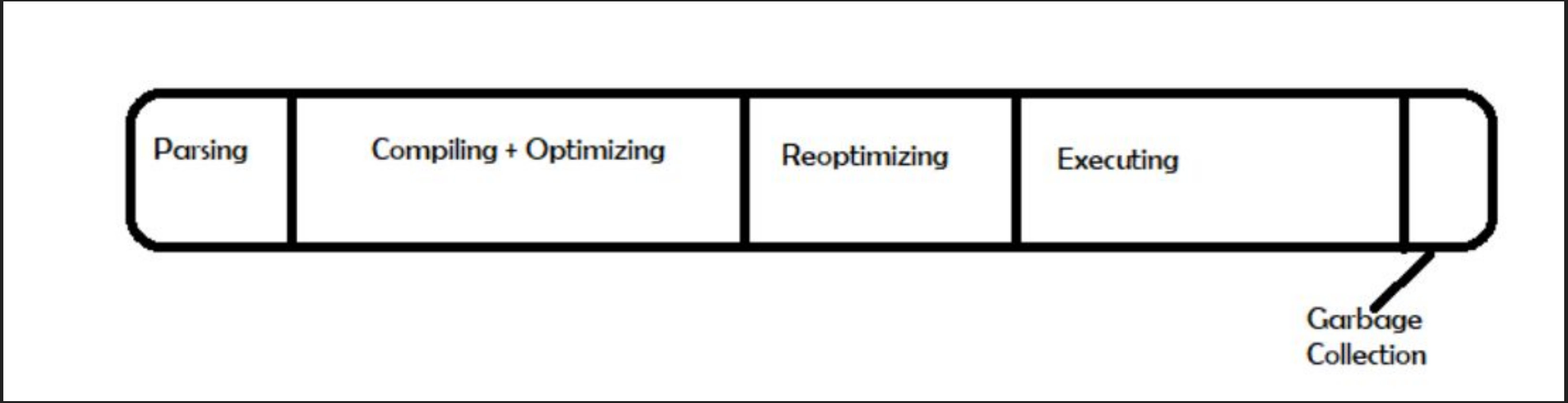
- Developed in 1995 by Brendan Eich
- Low entry level barrier leading to ubiquity
- Undefeated champion of compilation targets ever since
- Non-exhaustive list of competitors
  - ActiveX
  - Adobe Flash
  - JavaBeans

Divya_Mohan02

Cons

- Not designed to be a compilation target
- Weakly typed
- Eventually faster
    - We’ll cover this in the next slide!

About time



Credit: https://blog.devgenius.io/a-primer-on-webassembly-834150fdd7ae

Divya_Mohan02

ACT II: asm.js

Asm.js: The origin story

- Subset of JavaScript
- Spec describes sandboxed VMs for memory-unsafe languages
- Low-level efficient compiler target
- Implemented by Mozilla

Cons

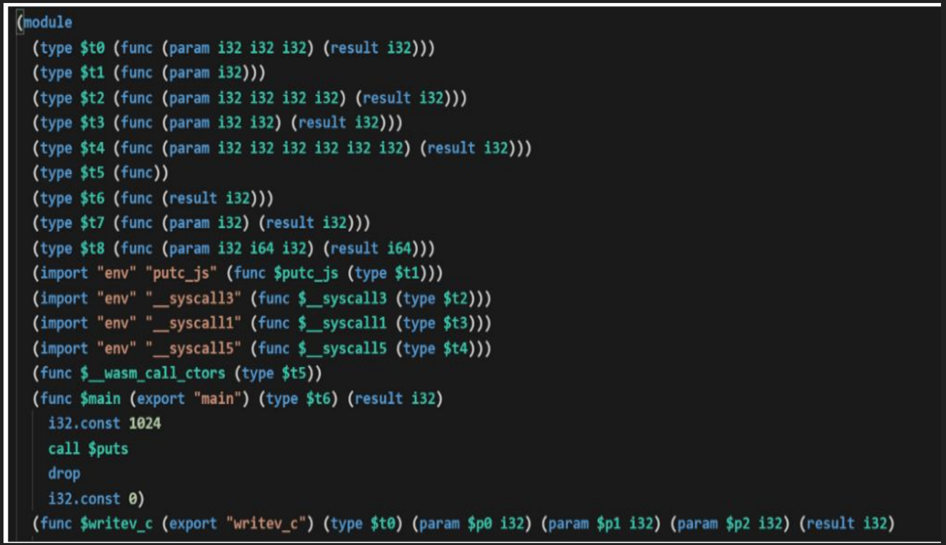
- Not standardized
  - Informal spec
  - Vendor implementations were customized
- Still limited to things that were expressible in JavaScript

ACT III: WebAssembly

Divya_Mohan02

What is it?

- Binary instruction format for stack-based virtual machines
- Designed to be a portable compilation target
    - On the web
    - Off it, as well!
- Strongly typed



```
(module
  (type $t0 (func (param i32 i32 i32) (result i32)))
  (type $t1 (func (param i32)))
  (type $t2 (func (param i32 i32 i32 i32) (result i32)))
  (type $t3 (func (param i32 i32) (result i32)))
  (type $t4 (func (param i32 i32 i32 i32 i32 i32) (result i32)))
  (type $t5 (func))
  (type $t6 (func (result i32)))
  (type $t7 (func (param i32) (result i32)))
  (type $t8 (func (param i32 i64 i32) (result i64)))
  (import "env" "putc_js" (func $putc_js (type $t1)))
  (import "env" "__syscall3" (func $__syscall3 (type $t2)))
  (import "env" "__syscall1" (func $__syscall1 (type $t3)))
  (import "env" "__syscall5" (func $__syscall5 (type $t4)))
  (func $__wasm_call_ctors (type $t5))
  (func $main (export "main") (type $t6) (result i32)
    i32.const 1024
    call $puts
    drop
    i32.const 0)
  (func $writev_c (export "writev_c") (type $t0) (param $p0 i32) (param $p1 i32) (param $p2 i32) (result i32)
```

Divya_Mohan02

Will it replace JavaScript?

Divya_Mohan02

Demo time!

Why is it better?

- Designed to be a compilation target
    - Allows for more languages to be brought into the Web
- Standardized across the four major browsers
- Faster than JavaScript
- Predictable performance across application

Where are we now?

- Still at v1
- MVP released in 2017
    - A lot of things are still in the pipeline for implementation
- Experimental projects/offerings popping up for:
    - Server-side implementation (like NodeJs for JavaScript)
    - Combining cloud native tech to form,
        - Application frameworks running on Kubernetes (e.g. Atmo)
        - Scheduling wasm modules on cloud native frameworks (e.g. Krustlet)

Questions?

Divya_Mohan02