



WebAssembly



emscripten



MeetUp C++

Autore:

Lorenzo Veneri

Twitter: @ellorenztw

Sito: <https://lorenzoveneri.azurewebsites.net/>

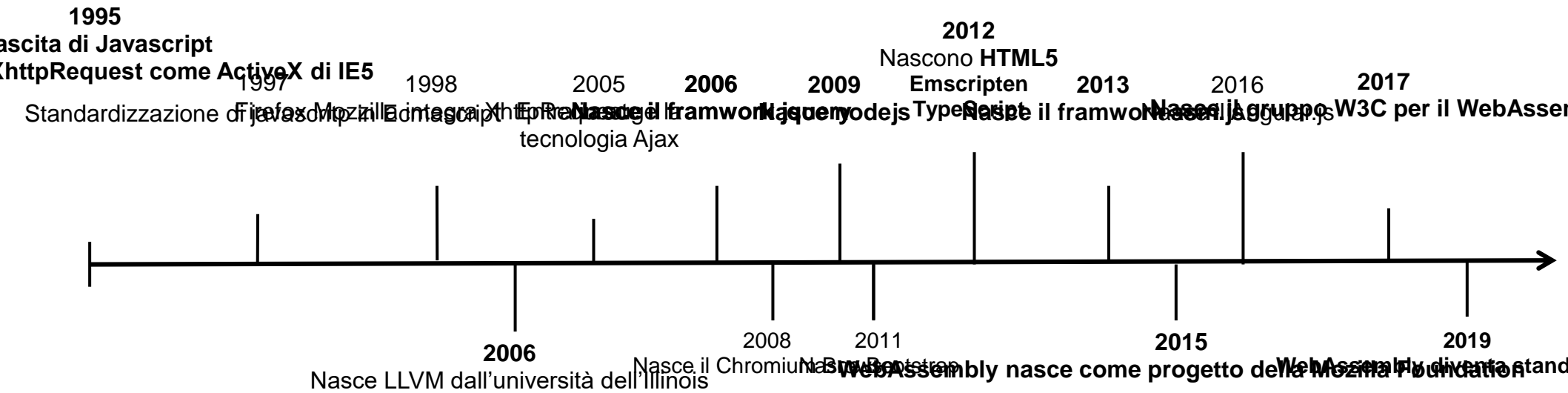
Nascita del WebAssembly

Con l'avvento del XMLHttpRequest, jquery, nodejs e dei framework javascript lato client per far sentire

Partendo da Asm.js e emscripten, che permetteva di compilare del codice c/c++ in javascript o

Nel 2019 il WebAssembly è stato riconosciuto come standard W3C.

Cronostoria per di javascript fino al WebAssmebly



Cos'è il WebAssembly

WebAssembly is a new type of code that can be run in modern web browsers — it is a low-level assembly-like language with a com

fonte: <https://developer.mozilla>

WebAssembly è un nuovo tipo di codice che può essere eseguito nel moderni browser, è un

Obbiettivi del WebAssembly

WebAssembly is being created as an open standard inside the W3C WebAssembly Community Group with the following goals:

- ✓ *Be fast, efficient, and portable — WebAssembly code can be executed at near-native speed across different platforms by*
- ✓ *Be readable and debuggable — WebAssembly is a low-level assembly language, but it does have a human-readable text*
- ✓ *Keep secure — WebAssembly is specified to be run in a safe, sandboxed execution environment. Like other web code, it*
- ✓ *Don't break the web — WebAssembly is designed so that it plays nicely with other web technologies and maintains back*

Fonte: <https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts>

Tecnologia WebAssembly

Il WebAssembly inizialmente è stato sviluppato con tecnologie legate all'ambiente linux e linguaggi come c,c++ e rust, tuttavia sono o
I programmi vengono tradotti e compilati in un assembly a basso livello

Codice sorgente in linguaggio C

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

Linear assembly bytecode (Rappresentazione intermedia)

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

Codifica binaria Wasm (byte esadecimali)

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

S-Expression

La rappresentazione del codice wasm può essere effettuata tramite un linguaggio definito s-expression

Codice sorgente in linguaggio C

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

Rappresentazione s-expression:

```
(module  
  (import "math" "exp" (func $exp (param f64) (result f64)))  
  (func (export "doubleExp") (param $0 f64) (result f64)  
    (f64.mul  
      (call $exp  
        (get_local $0))  
        (f64.const 2))))
```

Caratteristiche

- Nel Web Assembly standard non è prevista l'interazione con l'oggetto DOM.
- Si possono importare le funzioni Javascript del browser all'interno
- Si possono esportare oggetti e strutture per l'interazione con gli script del browser
- Per l'utilizzo nel browser come libreria wasm occorre avere del codice javascript di correlazione
- Nativamente vengono supportati i seguenti tipi di dati
 - Double
 - Int
 - Char

Interazione tra javascript e codice wasm

Wasm e javascript interagiscono attraverso un oggetto di memoria condivisa chiamato "memory".
64Kb (65536 bytes), è possibile settare la cache di memoria aumentando il numero di pagine.

All'interno della cache vengono inseriti i puntatori alle stringhe che vengono trattati come array.



Cos'è Emscripten:

Emscripten è una Low Level Virtual Machine (LLVM) open source per compilare codice C.

Oltre a questo permette anche la creazione dei file .wasm che sono alla base dei WebAssembly.

Le licenze usate sono:

MIT e University of Illinois/NCSA Open Source License.



- ✓ Emscripten permette di compilare codice c/c++ in javascript e in web assembly (wasm)
- ✓ Permette l'ottimizzazione del codice
- ✓ Dispone di una serie di API che permettono l'interazione con
 - ✓ Emscripten.h – permette l'interazione con il browser
 - ✓ Html5.h – per l'interazione con gli eventi html5 in codice nativo
 - ✓ preamble.js – Serve per permettere l'interazione tra il codice c e il codice javascript
 - ✓ File System Api – Permette l'interazione del codice con il



- › Per effettuare il collegamento tra c++ e javascript si possono usare
 - › WebIDL
 - › Embind
- › La versione attuale non supporta per il c++ i

Riferimenti

- ✓ <https://developer.mozilla.org/en-US/docs/WebAssembly>
- ✓ <https://it.wikipedia.org/wiki/WebAssembly>
- ✓ <https://webassembly.org/>
- ✓ <http://asmjs.org/>
- ✓ https://emscripten.org/docs/introducing_emscripten/about_emscripten.html
- ✓ <https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/>
- ✓ [https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts#How does WebAssembly fit into the web platform](https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts#How_does_WebAssembly_fit_into_the_web_platform)
- ✓ <https://wasdk.github.io/WasmFiddle/>
- ✓ <https://webassembly.org/docs/faq/#will-webassembly-support-view-source-on-the-web>
- ✓ <https://kripken.github.io/talks/wasm.html#/>



Lorenzo Veneri

MCSA (Microsoft Certificated Solution Associated) Web Development
Analista sviluppatore presso X DataNet

Twitter: @ellorenztw

Telegram: ellorenz