

# WebAssembly

Morgan Crociati

# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>Résumé</b>	<b>3 - 6</b>
<b>Origine</b>	<b>7 - 10</b>
<b>Fonctionnement</b>	<b>11 - 12</b>
<b>Démonstration</b>	<b>13</b>
<b>Conclusion</b>	<b>14 - 17</b>

# Résumé

# Résumé

Web + Assembleur

2017



*source: fr.wikipedia.org*



*source: fr.wikipedia.org*

# Résumé

WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

*source: webassembly.org*



*source: google.fr*



Microsoft

*source: microsoft.fr*



*source: apple.fr*



*source: intel.fr*



Red Hat

*source: redhat.com*

# Résumé

Efficient and fast

Safe

Part of the open web platform

Portable

Open and debuggable

# Origine

# Origine

2012

Alon Zakai



**emscripten**

*source: commons.wikimedia.org*

**moz://a**

*source: mozilla.org*



2013

**asm.js**

an extraordinarily optimizable, low-level subset of JavaScript

*source: asmjs.org*

```
function strlen(ptr) { // calculate length of C string
  ptr = ptr|0;
  var curr = 0;
  curr = ptr;
  while (MEM8[curr]|0 != 0) {
    curr = (curr + 1)|0;
  }
  return (curr - ptr)|0;
}
```

*source: Alon Zakai, emscripten talk, asmjs.org*

**moz://a**

*source: mozilla.org*

# Origine



*source: fr.wikipedia.org*



*source: fr.wikipedia.org*



*source: fr.wikipedia.org*



*source: mozilla.org*



*source: fr.wikipedia.org*

## 2015

# Fonctionnement

# Fonctionnement

**.wat**

**.wasm**

**Bytecode**



# Démonstration

# Conclusion

# Conclusion

C/C++

Rust

AssemblyScript

C#

F#

Go

Kotlin

Swift

D

Pascal

Zig

GPU

Multithreading

Unification des bibliothèques

Garbage Collector

Exception handling

# Conclusion

**HTML**



**CSS**



**WA**

**JS**



# Question