

WASM

La mort de JavaScript ?





**Michaël
Fiorito**



michael.fiorito@oniryx.be



**Denis
Voituron**



denis@voituron.net



Le JavaScript

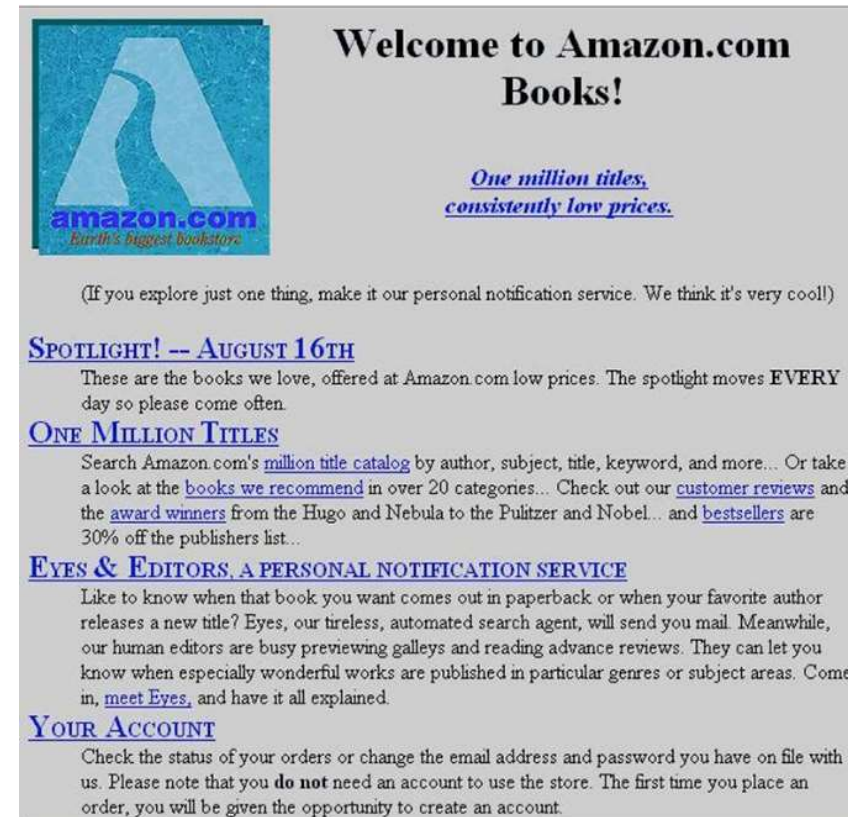
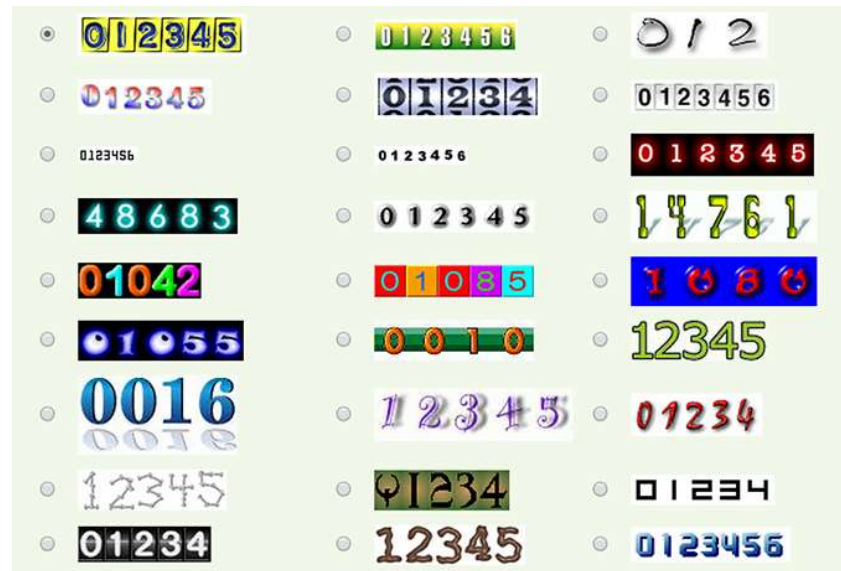


JavaScript (souvent abrégé en « JS ») est un langage de **script** léger, orienté objet, principalement connu comme le langage de script des pages web. Le code JavaScript est **interprété** ou **compilé à la volée** (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un **typage faible et dynamique** qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet.

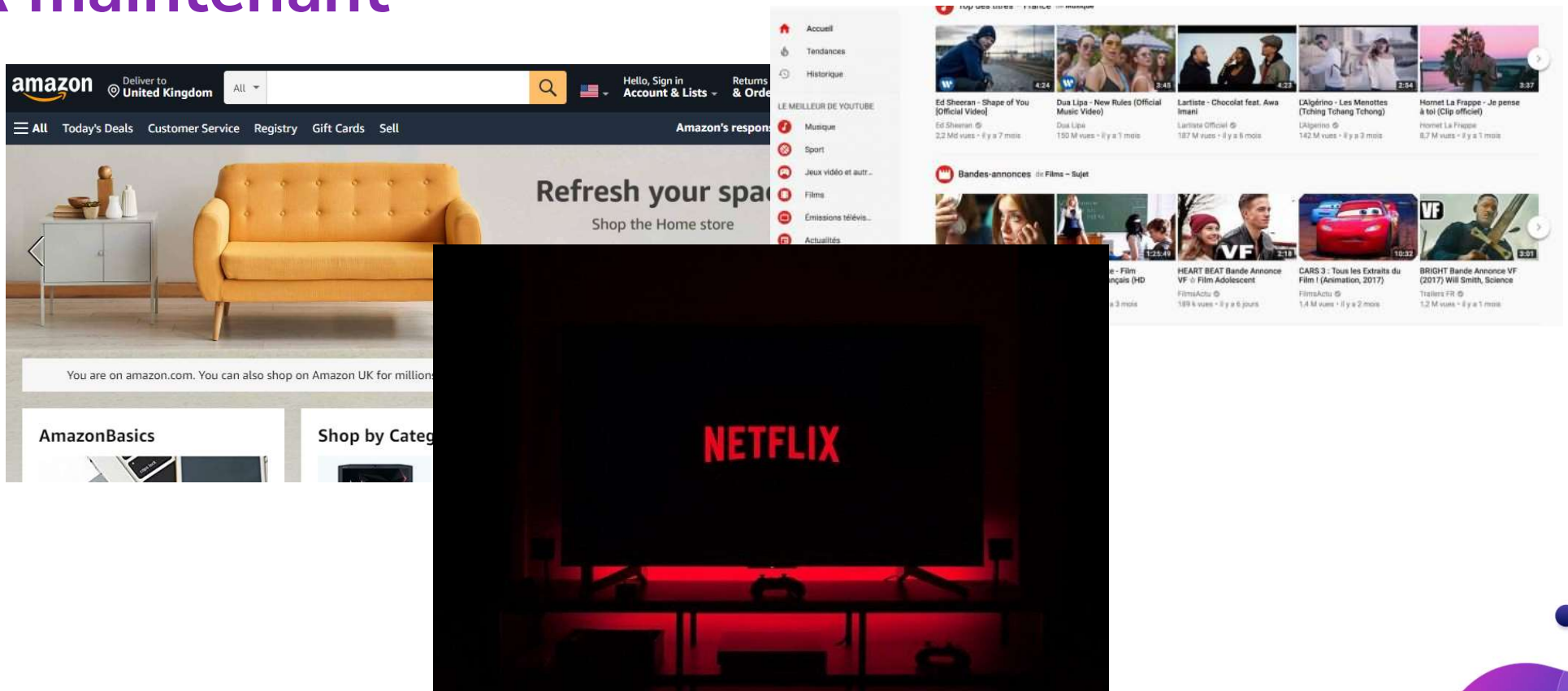
Source: mozilla.org



De 1995 ...

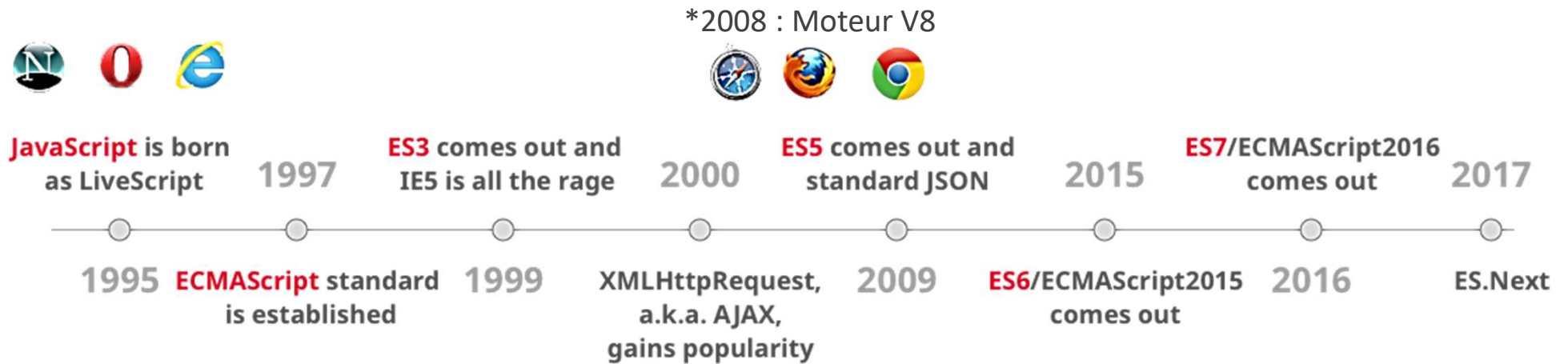


A maintenant

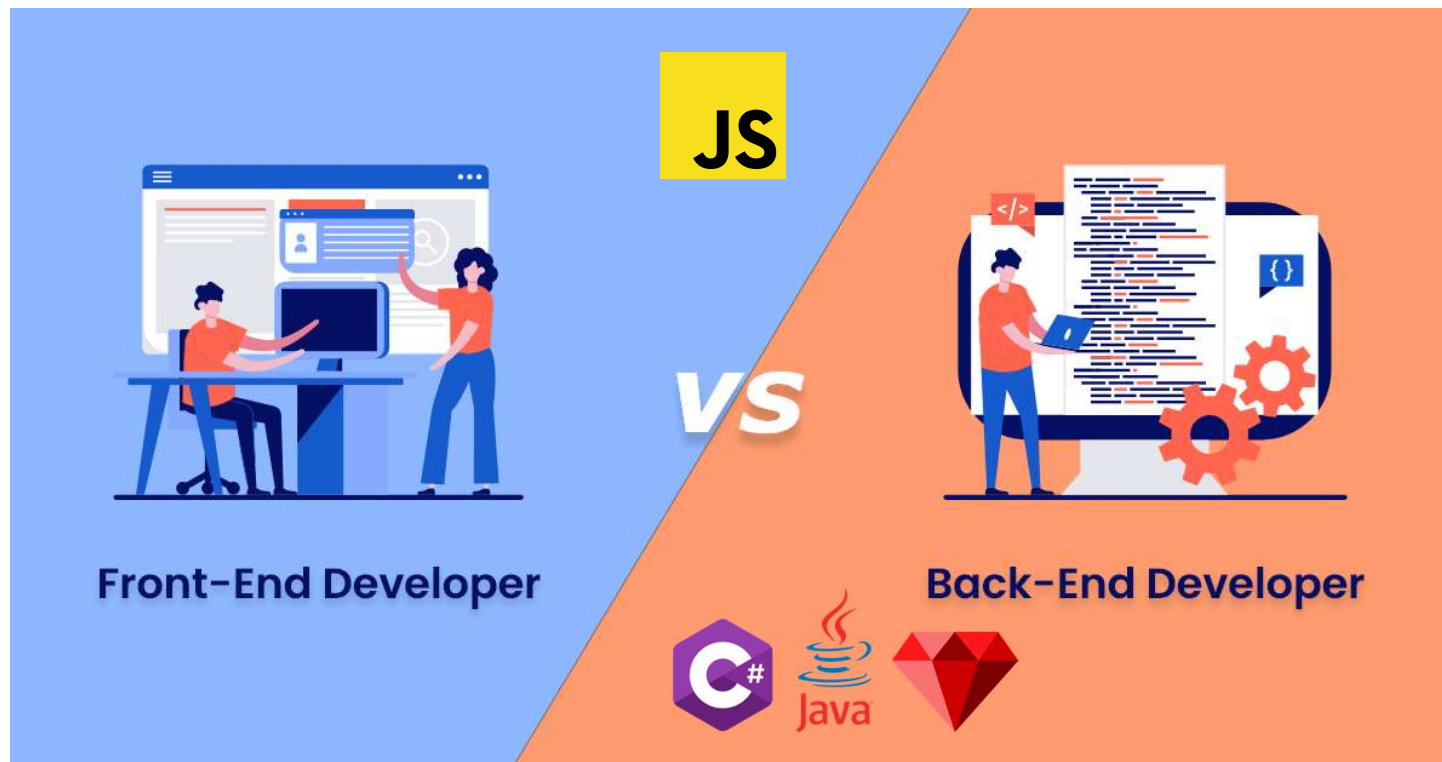


(et oui... les compteurs de « visites » sont morts et enterrés ☺)

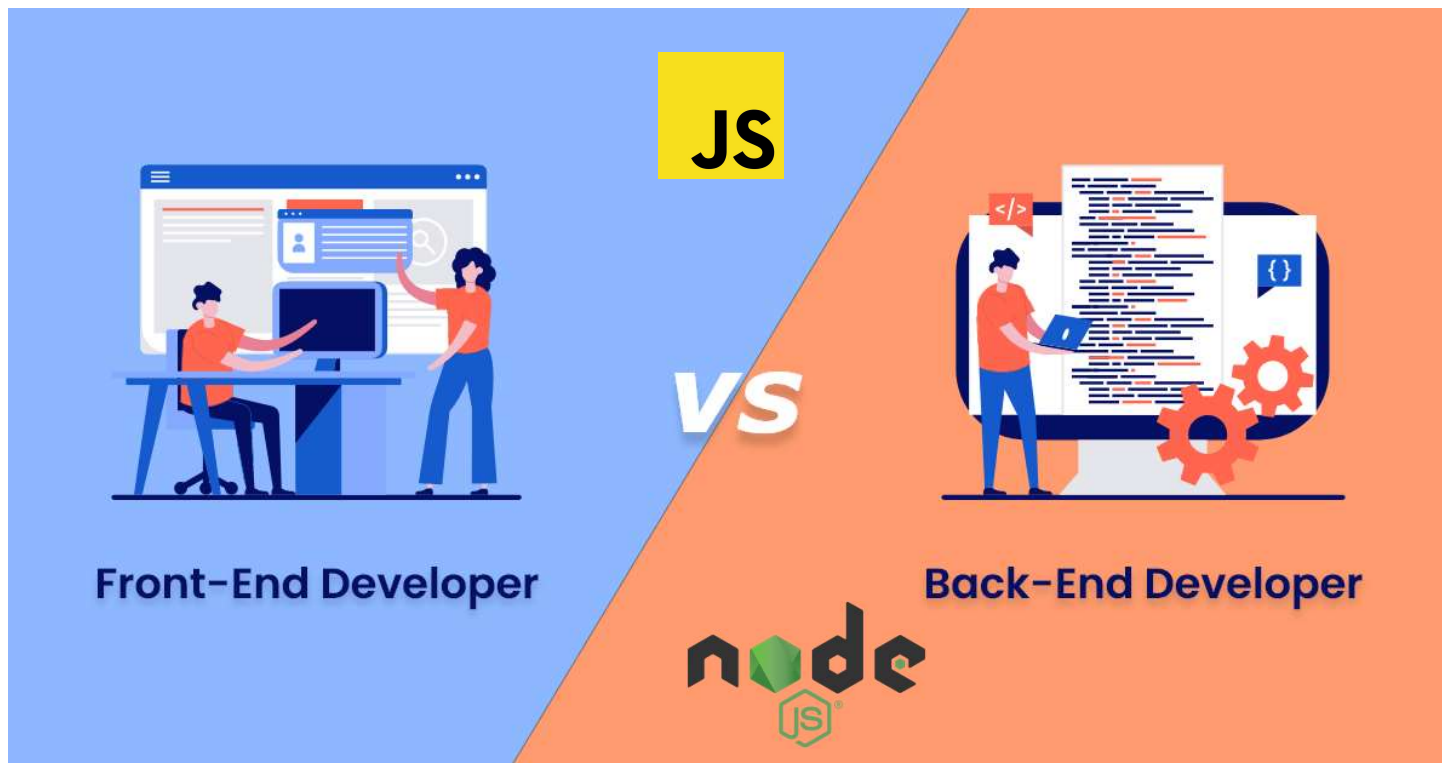




D'abord comme FrontEnd



Ensuite comme BackEnd

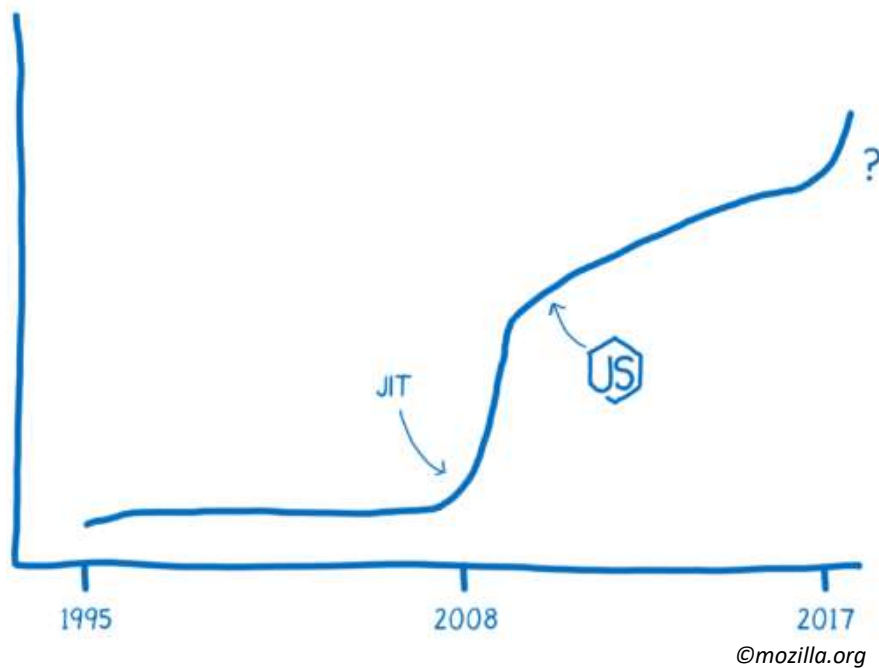


A still from the movie Toy Story showing Woody and Buzz Lightyear. Woody is on the left, looking concerned. Buzz is on the right, in his green and white space ranger suit, with his arms outstretched in a dramatic gesture. The background is a simple room with a door and some stars on the wall.

JAVASCRIPT

JAVASCRIPT EVERYWHERE

Et maintenant ?



- Limité par sa conception (Interprété / JIT)
- Pas de réutilisation de code existant (C, C++, C#, ...)
- Performances « moyennes »
- Sécurité gérée du côté client
- Inconsistances dues au navigateur

Des tentatives de solution



WebAssembly



Le "**wasm**" est un format **portable**, efficace en termes de **taille** et de **temps** de chargement, adapté à la compilation sur le **web**.





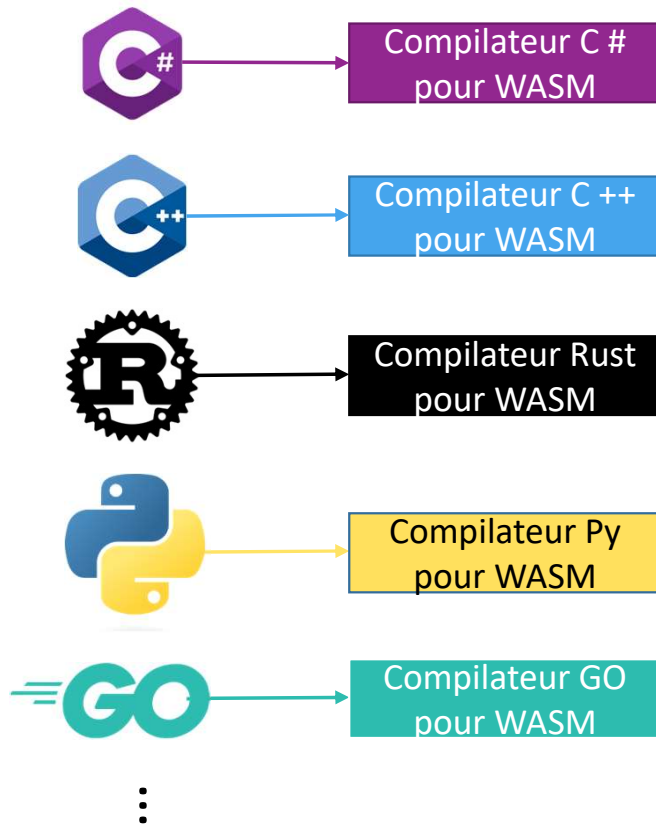
Navigateurs supportés

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
			2-46													
4-50	12-14		¹ 47-51	10-37												
² 51-56	³ 15	3.1-10.1	⁴ 52	² 38-43			3.2-10.3	4-6.4								
57-106	16-106	11-16.0	53-105	44-90	6-10		11-16.0	7.2-17.0		12-12.1		2.1-4.4.4				
107	107	16.1	106	91	11	107	16.1	18.0	all	64	13.4	107	106	13.1	13.18	¹ 2.5
108-110		16.2-TP	107-108													

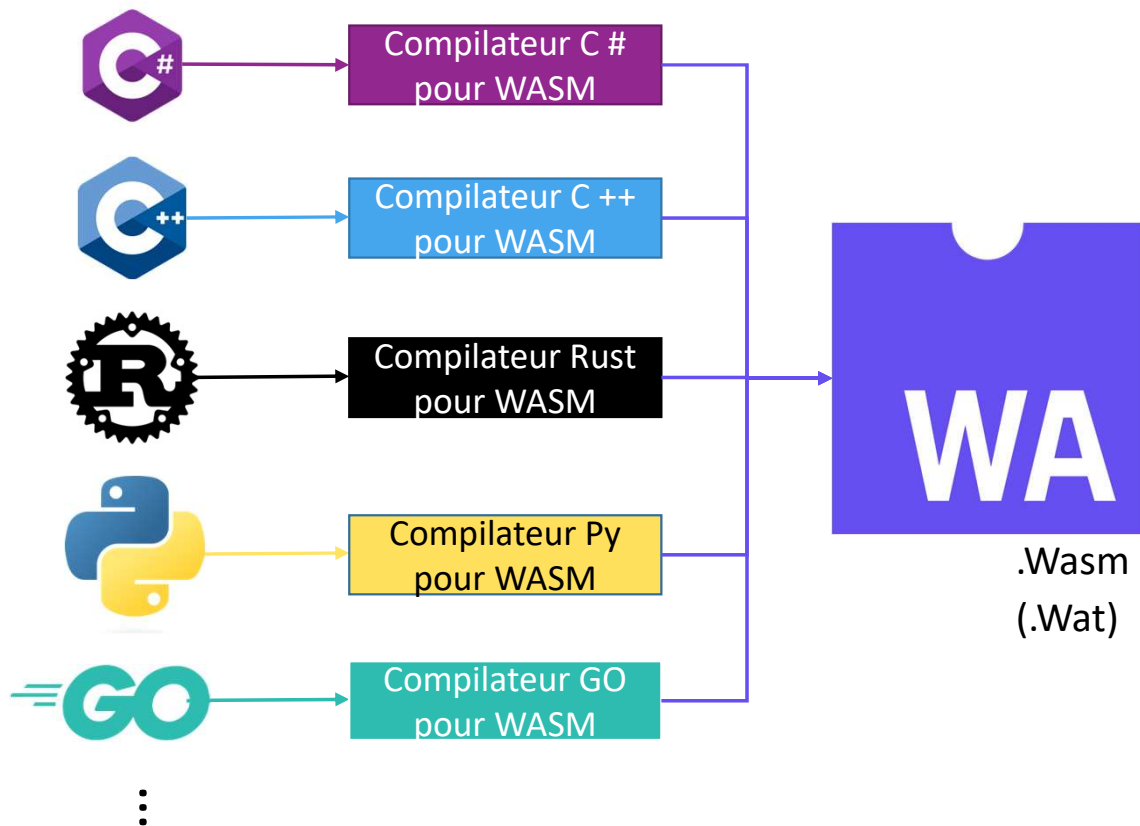
Global usage: 96.2% of all users



Comment ça marche ?



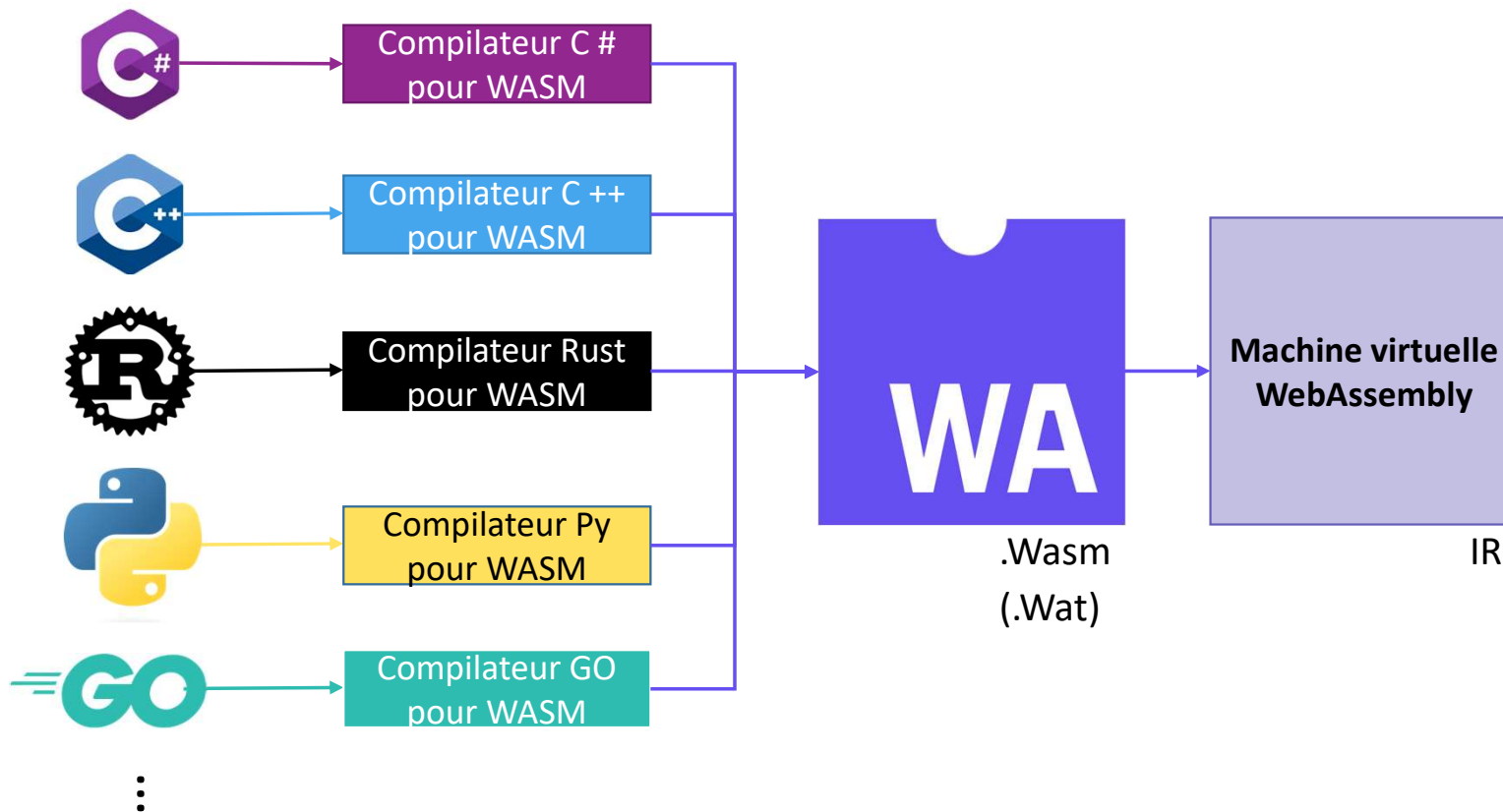
Comment ça marche ?



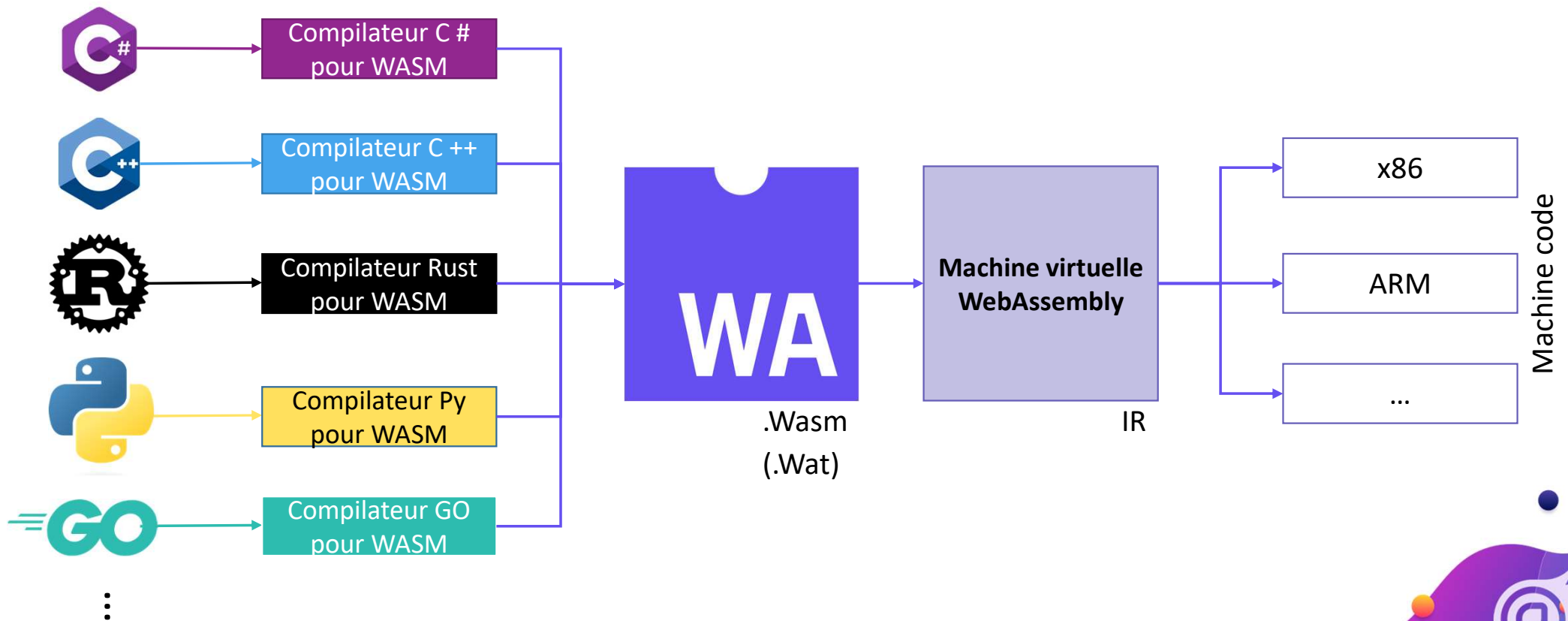
```
(module
  (table 0 anyfunc)
  (memory $0 1)
  (export "memory" (memory $0))
  (export "factorial" (func $factorial))
  (func $factorial (; 0 ;) (param $0 i32) (result i32)
    (local $1 i32)
    (local $2 i32)
    (set_local $2
      (i32.const 1)
    )
    (block $label$0
      (br_if $label$0
        (i32.lt_s
          (get_local $0)
          (i32.const 2)
        )
      )
      (set_local $2
        (i32.const 1)
      )
    )
    (loop $label$1
      (set_local $2
        (i32.mul
          (get_local $0)
          (get_local $2)
        )
      )
      (set_local $1
        (i32.gt_s
          (get_local $0)
          (i32.const 2)
        )
      )
      (set_local $0
        (i32.add
          (get_local $0)
          (i32.const -1)
        )
      )
      (br_if $label$1
        (get_local $1)
      )
    )
    (get_local $2)
  )
)
```

.wat

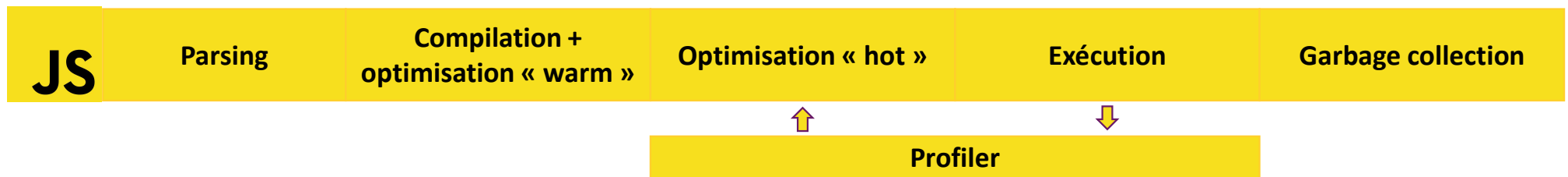
Comment ça marche ?



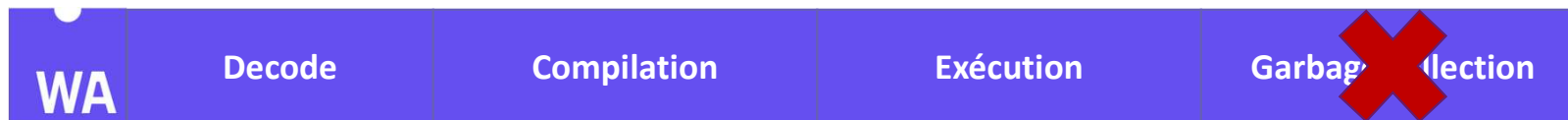
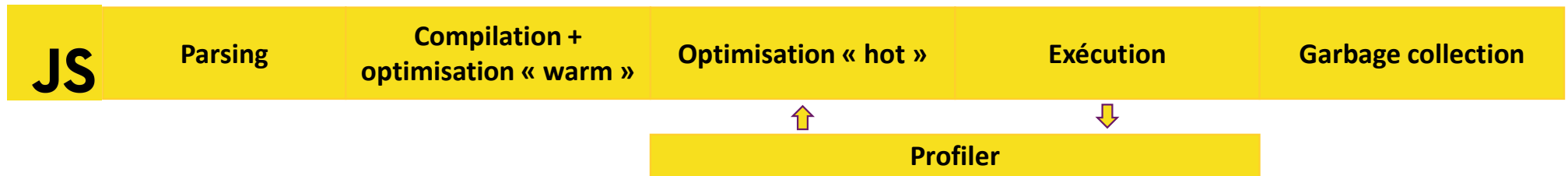
Comment ça marche ?



Performant ?



Performant ?



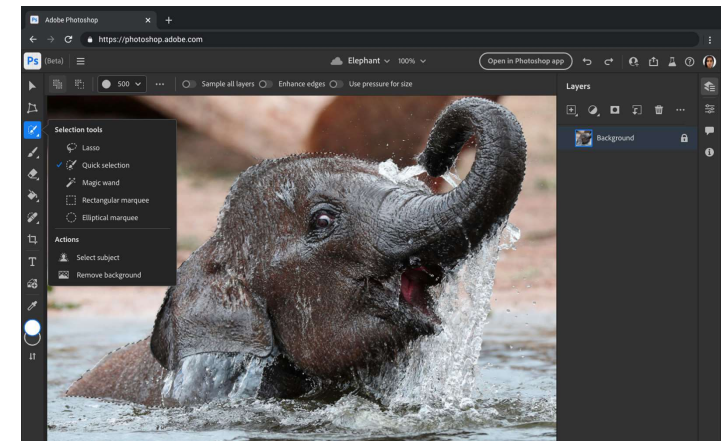
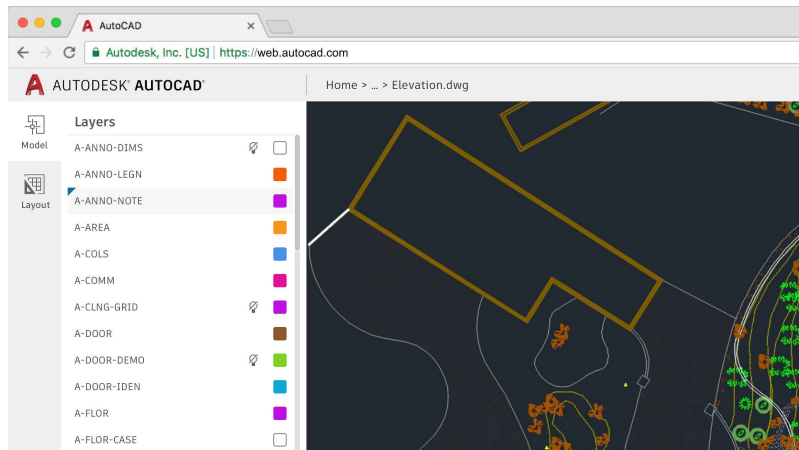
Démo



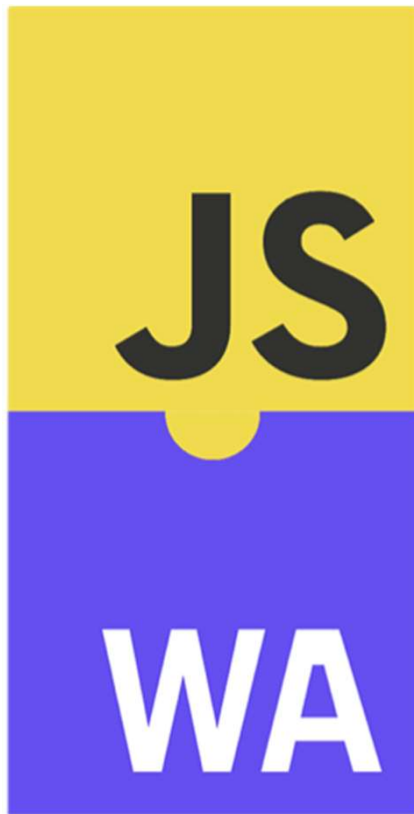
emscripten



Mieux qu'un "Hello world"



JS et WA : Complémentaires



JavaScript (souvent abrégé en « JS ») est un langage de **script** léger, orienté objet, principalement connu comme le langage de script des pages web. Le code JavaScript est **interprété** ou **compilé à la volée** (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un **typage faible et dynamique** qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet.

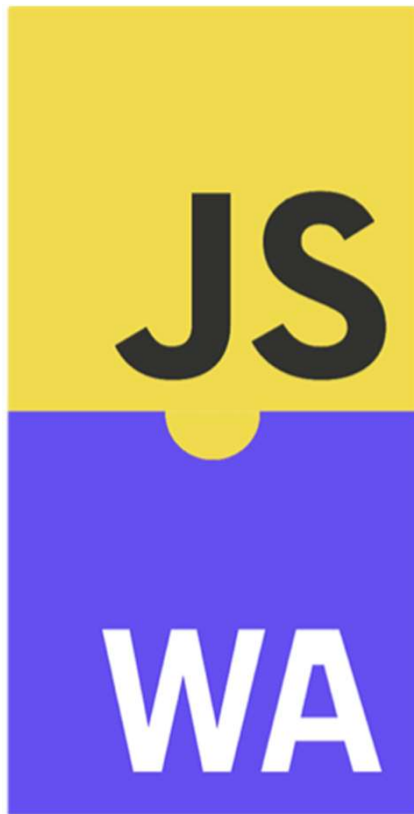
Source: mozilla.org

WebAssembly permet d'exécuter du code écrit depuis **différents langages** sur le Web avec des performances similaires aux applications **natives**. Il est conçu pour être utilisé **de pair avec JavaScript** : on peut charger des modules WebAssembly au sein d'une application JavaScript et partager des fonctionnalités entre les deux. **Cela permet de tirer parti des performances de WebAssembly et de la flexibilité de JavaScript.**

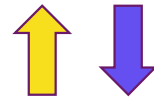
Source: mozilla.org



JS et WA : Complémentaires



- Flexibilité
- Simplicité
- Grande communauté



- Performant
- Sécurisé (sandbox)
- Langages déjà connus (C#, C++, ...)

