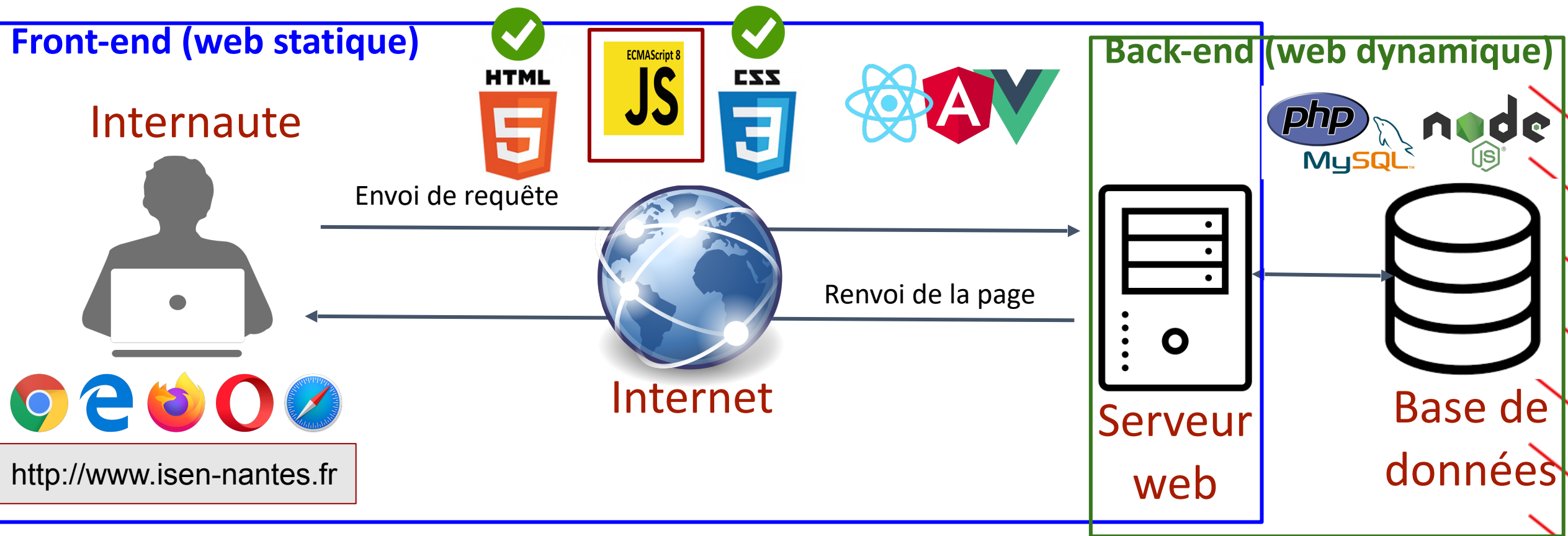


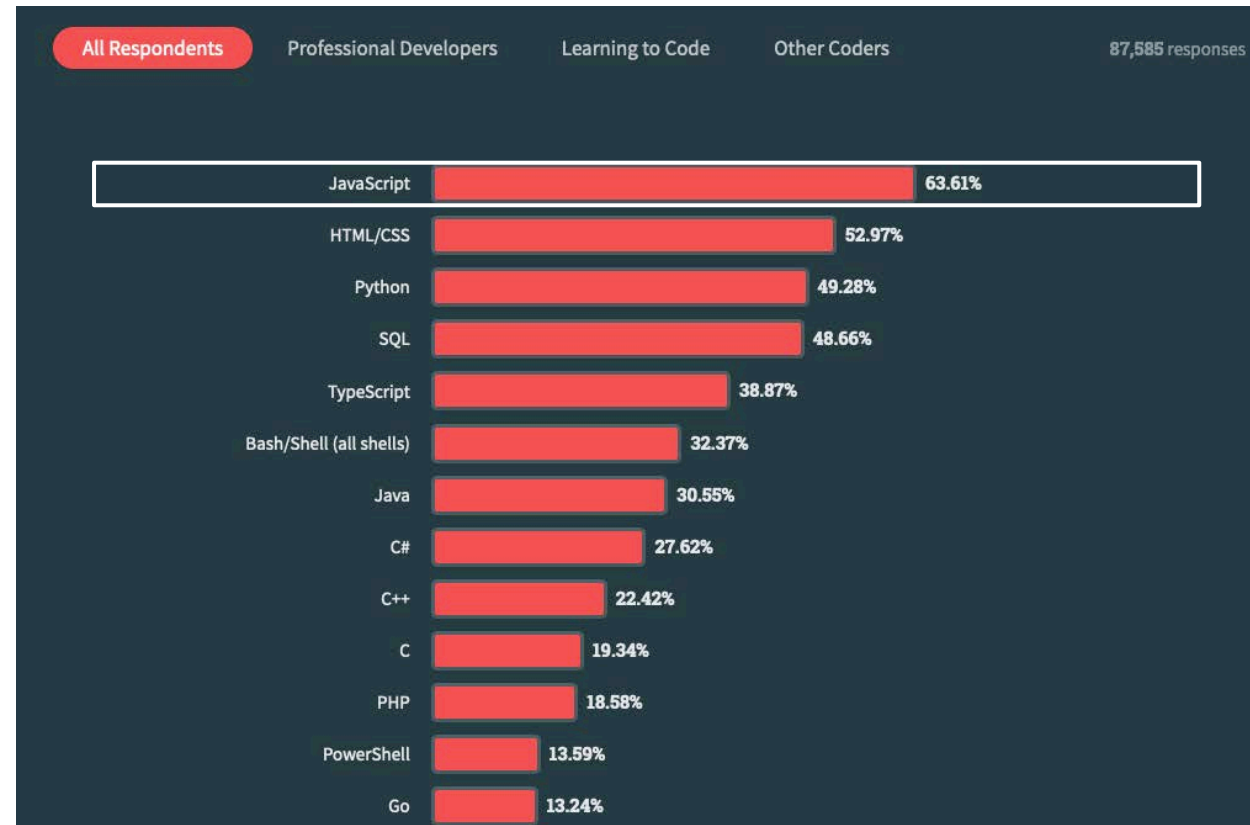
JavaScript - Introduction -

Groupe des étudiants : CIR1



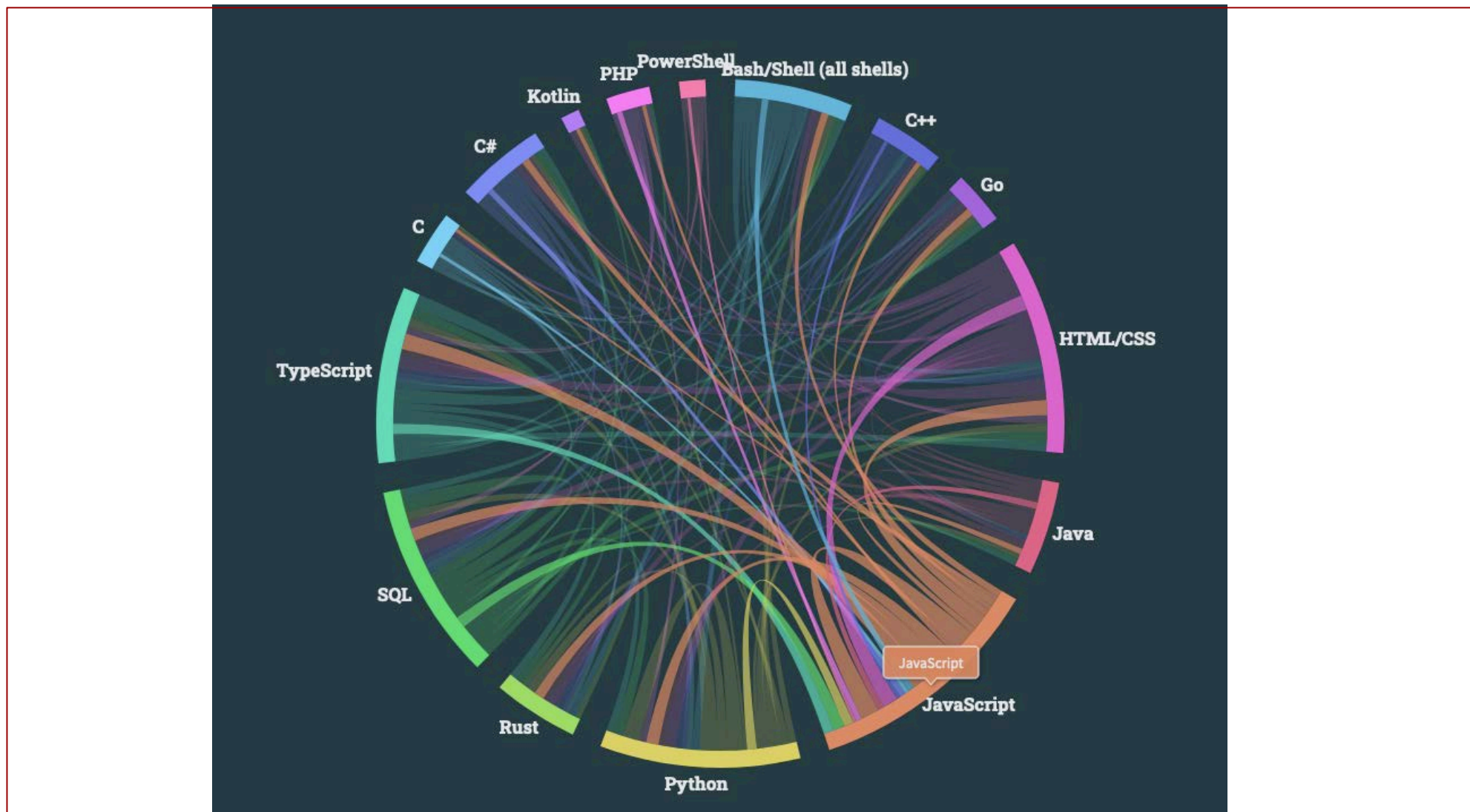


Classement 2023 des langages de programmation



Source : <https://survey.stackoverflow.co/2023/#most-popular-technologies-language>

Lien entre les différentes technologies



Définition

- **JavaScript** (souvent abrégé **JS**) est un langage de programmation web créé en 1995 par Brendan Eich (Netscape)
- C'est un langage orienté client qui ajoute de l'interactivité aux sites web
- Il ne faut pas confondre JS avec Java
JavaScript est un langage de script :
 - Faiblement et dynamiquement typé
 - Orienté objet (basé sur les prototypes)

Java est un langage développé par Oracle :

- compilé, ayant un typage fort, et basé sur des classes.
- utilisé pour faire des applications bureau (JSE), serveur (JEE), ou mobiles (Android)



ECMAScript

- JavaScript est standardisé par Ecma International depuis 1997 sous le nom d'ECMAScript.
- La dernière spécification du langage est [ECMAScript 2023](https://tc39.es/ecma262/2023/), publiée en Juin 2023.
- Il s'agit de la quatorzième édition de la spécification ECMA
<https://tc39.es/ecma262/2023/>



Extensions d'utilisation de JS

- Développement front-end



- Développement back-end



- Programmation mobile hybride (Cross-Platform Mobile App Development)

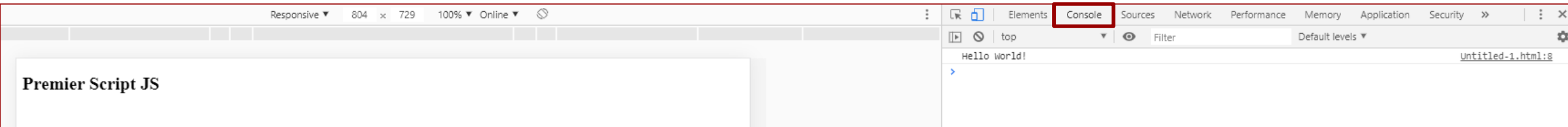


Moteurs JS

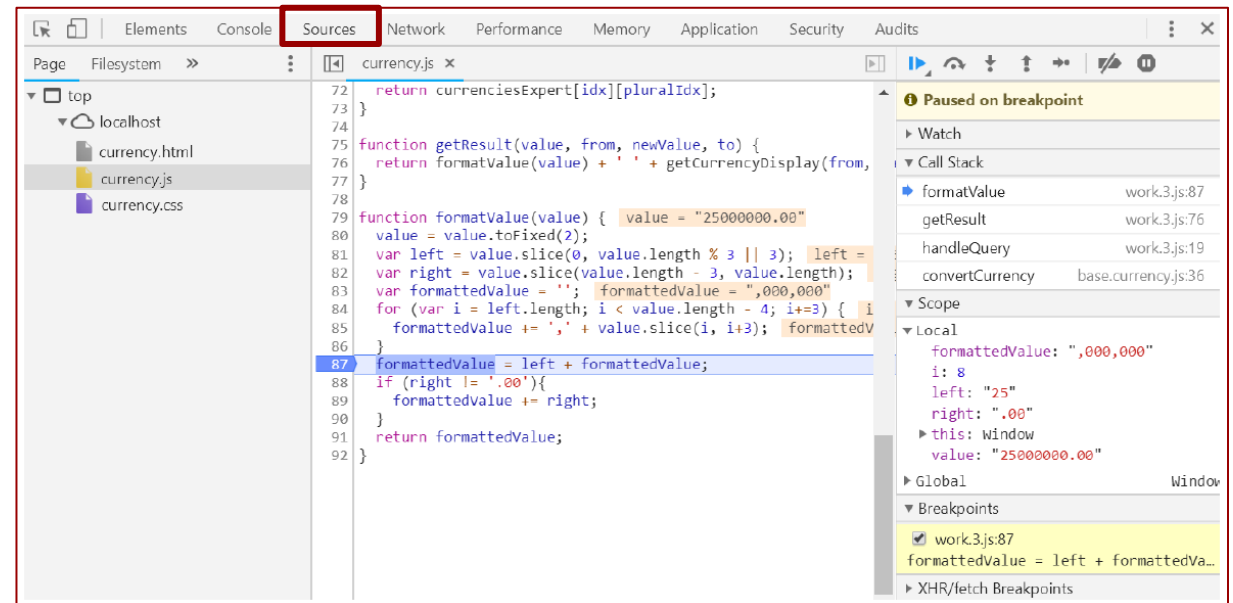
- Un moteur JS permet d'interpréter et d'exécuter un code JS
- Le premier moteur JS est SpiderMonkey.
- Les moteurs JS sont intégrés aux navigateurs :
 - Firefox (Mozilla) : SpiderMonkey (développé en C)
 - **Chrome (Google) : V8 (développé en C++)**
 - Edge (Microsoft) : Chakra (développé en C++)
 - Safari (Apple) : JavaScriptCore (développé en C++)
 - ...

Outils de développement

- Console JavaScript : consultation de la sortie et les erreurs du script
 - Exemple de Google Chrome :



- Débogage JavaScript
 - Exemple de Google Chrome



Outils de développement

- Prototypage en ligne :
 - [JSFiddle](#)
 - [CodePen](#)
 - [W3schools](#)
- Comparaison de différentes implémentations (fonctionnellement équivalentes) :
 - [JSPerf](#)

Test runner

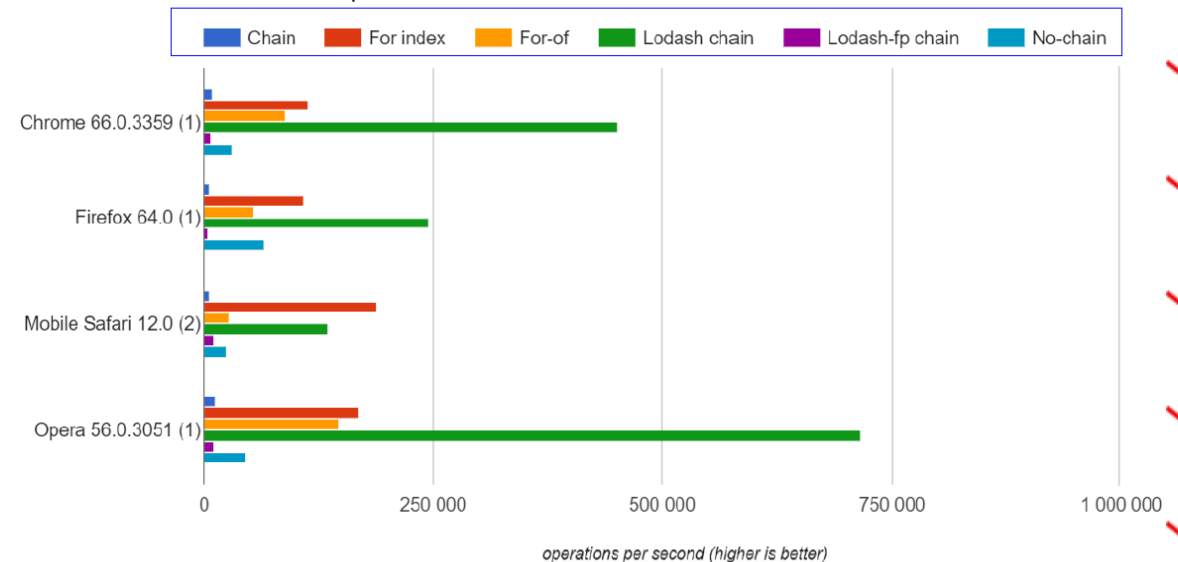
Ready to run.

Run tests

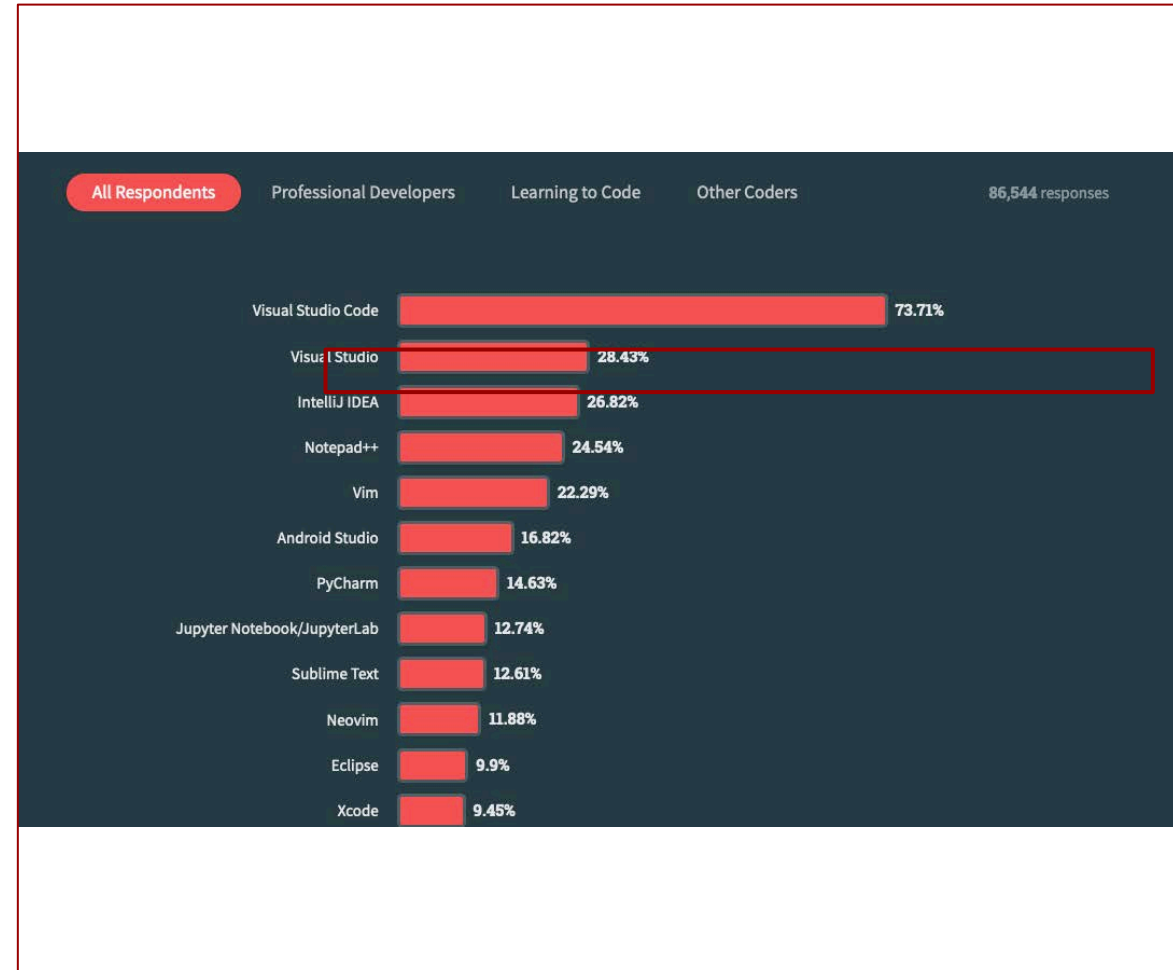
Testing in Chrome 71.0.3578 / Windows 10 0.0.0

Test		Ops/sec
Chain	<pre>const res = arr .filter(e => e && e.user) .map(e => e.user) .map(u => u.wealth 0) .reduce((acc, wealth) => acc + wealth, 0);</pre>	ready
No-chain	<pre>const res = arr.reduce((acc, e) => { const wealth = (e && e.user) ? (e.user.wealth 0) : 0; return acc + wealth; }, 0);</pre>	ready

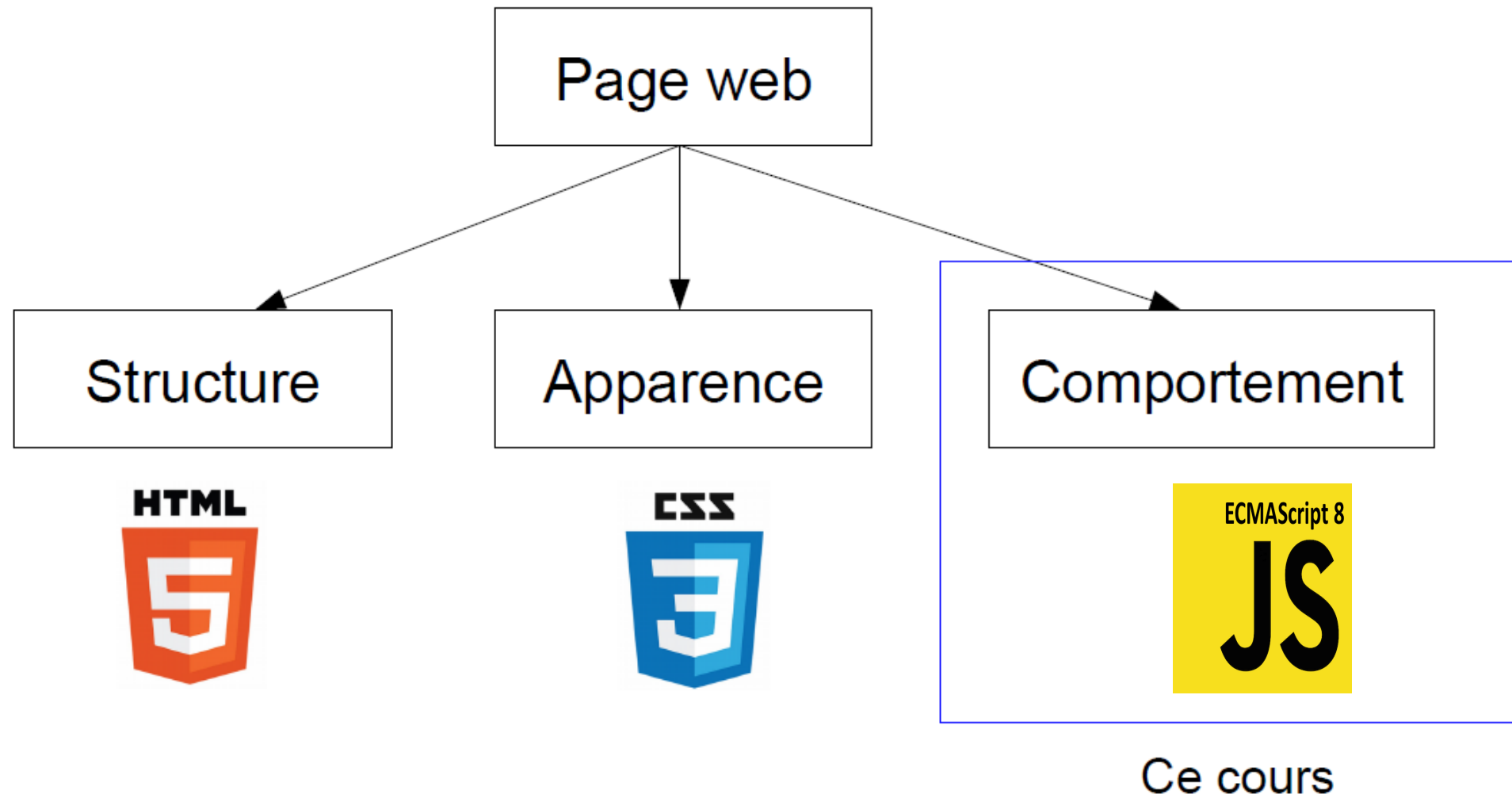
Différentes implémentations



- Environnement de développement intégré (IDE)
 - WebStorm (JetBrains)
- Codes open source :
 - **Github**
- Éditeurs de code :
 - **Visual Studio Code**
 - Atom
 - Sublime Text
 - Brackets
 - ...



Source : <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies>



Où intégrer le code JS dans HTML ?

1. À l'intérieur d'une balise `<script>`
 - Exemple

```
<script>  
  
... code JavaScript  
  
</script>
```

Inconvénient :

- Non respect de la séparation structure/contenu

Où intégrer le code JS dans HTML ?

1. À l'intérieur d'une balise `<script>`

2. Dans un script externe `".js"` en utilisant l'attribut `src` de la balise `<script>`

- Exemples :

a. Script local `<script src="scripts/math.js"></script>`

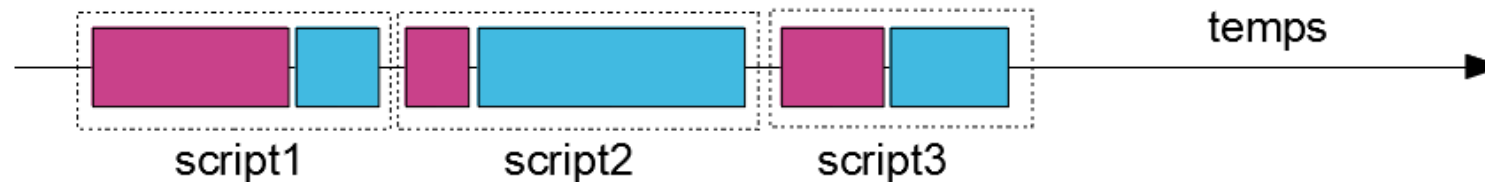
b. Script distant :

```
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
```

Chargement des scripts

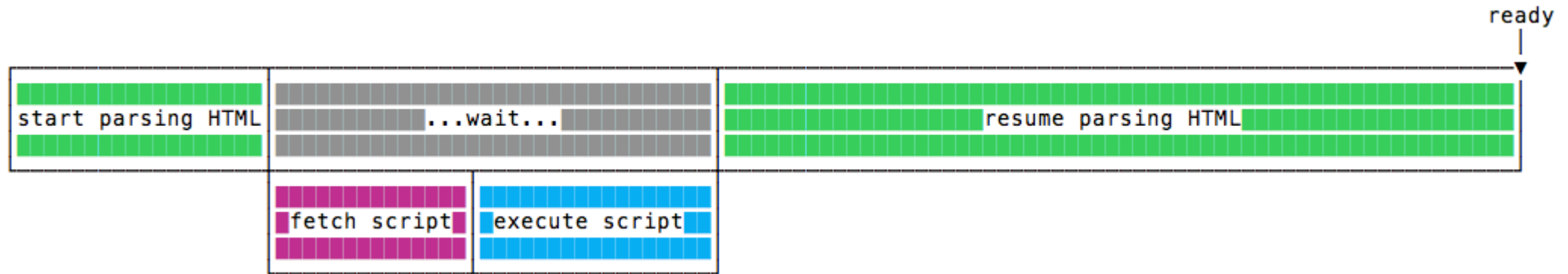
- Le chargement d'un script se décompose en 2 phases :
 - a. La récupération du script
 - b. L'exécution du script
- Par défaut, les scripts sont chargés de manière **synchrone**, ce qui implique:
 - a. qu'ils sont chargés dans l'ordre des balises
 - b. que le chargement de chaque script est bloquant
- Exemple :

```
<script src="path/to/script1.js"></script>  
<script src="path/to/script2.js"></script>  
<script src="path/to/script3.js"></script>
```



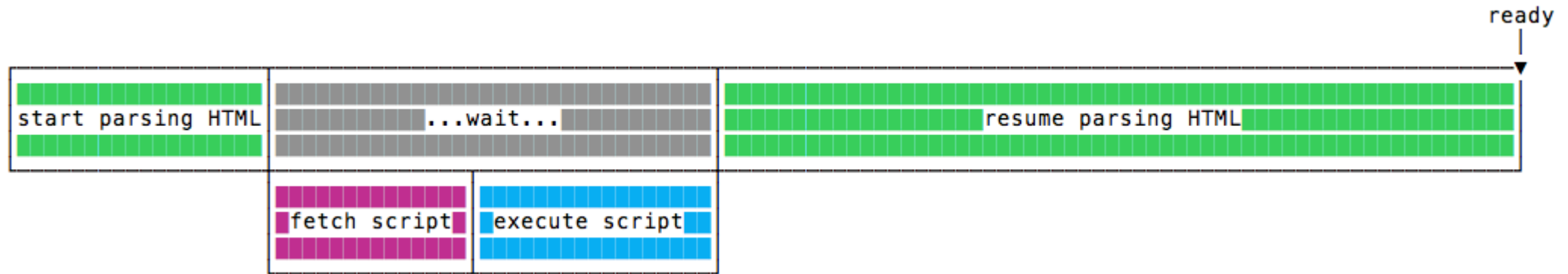
Où placer les balises `<script>` ?

- On peut placer les scripts dans la partie
 - a. `<head>` : le navigateur charge les scripts avant le corps de la page HTML ⇒ l'utilisateur voit une page vide pendant le temps du chargement des scripts

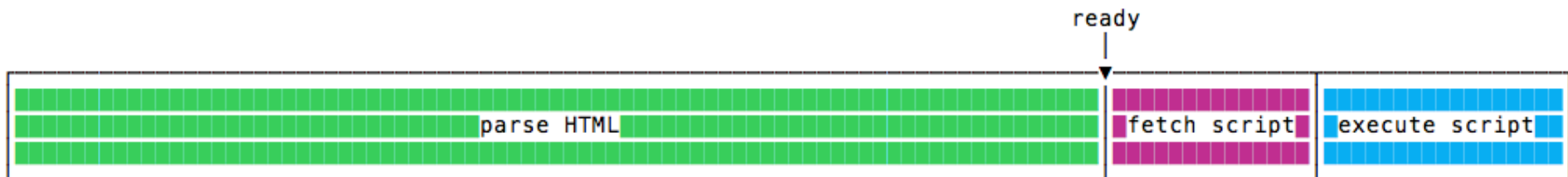


Où placer les balises `<script>` ?

- On peut placer les scripts dans la partie
 - a. `<head>` : le navigateur charge les scripts avant le corps de la page HTML ⇒ l'utilisateur voit une page vide pendant le temps du chargement des scripts



- b. `<body>` : permet l'affichage de la page HTML en attendant le chargement des scripts



Exemple

hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>JS HelloWorld</title>
  </head>

  <body>
    <h1>Open your console!</h1>
    <script src="hello.js"></
script>
  </body>
</html>
```

hello.js

```
console.log("Hello, World!");
```

Console
Google Chrome

