

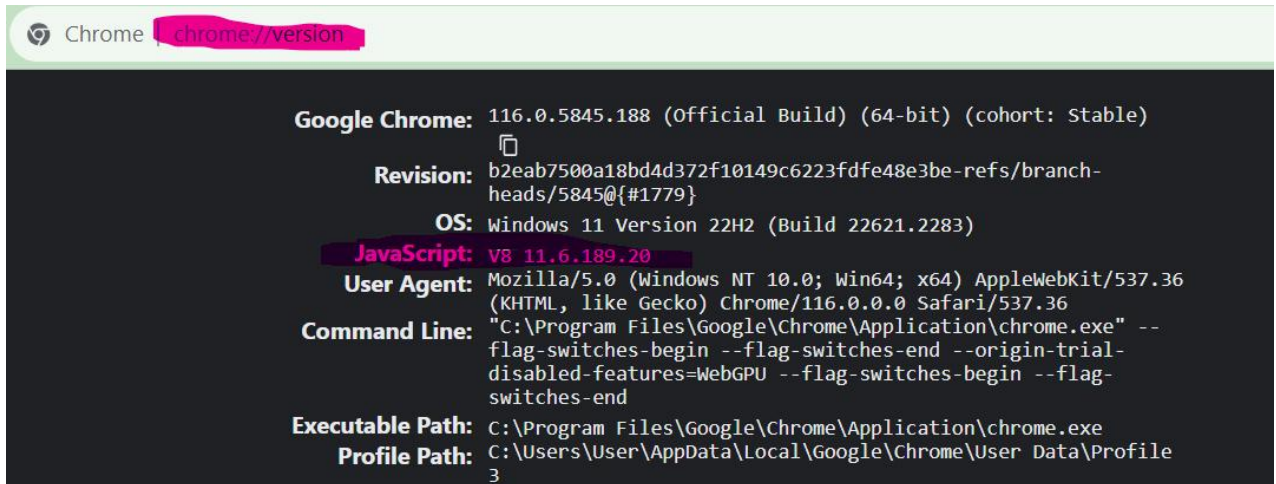
Section : Web et Mobile / S2

Matière : Interfaces utilisateur interactives (JavaScript)

TP1 : Préparation de l'environnement de développement JavaScript

1- Identifier la version moteur JavaScript

- Lancer chrome.
- Vérifier la version du moteur JS en tapant `chrome://version`



2- Exécution d'un programme JS :

Plusieurs méthodes existent pour exécuter un programme Js :

i. En utilisant la console du navigateur :

Donc on peut exécuter js directement sur chrome (ou n'importe quel autre navigateur).

- Taper **F12**. La console s'affichera.
- Taper votre premier script JS : `alert('Hello World !')` ; taper entrée.

ii. Exécuter JS avec Node.js en ligne de commande

- Aller sur nodejs.org et télécharger la dernière version de node.js puis l'installer.

Node.js ne possède pas d'interface graphique.

Pour l'utiliser il faut lancer un terminal (PowerShell sous Windows).

- Taper directement : `node` puis entrée.

Vous aurez le `prompt`. Vous pouvez commencer à coder du JS.

- Taper `console.log('Hello World !')` ;
- Taper `alert("bonjour");` => c'est quoi le problème ?

Ça ne reste toujours pas pratique. L'idéal est de travailler avec node.js sous Vscode.

NB : cette solution est l'idéale dans le cas où on exécute du JS sans manipuler le HTML.



iii. Utiliser un éditeur de texte VSCode avec node.js

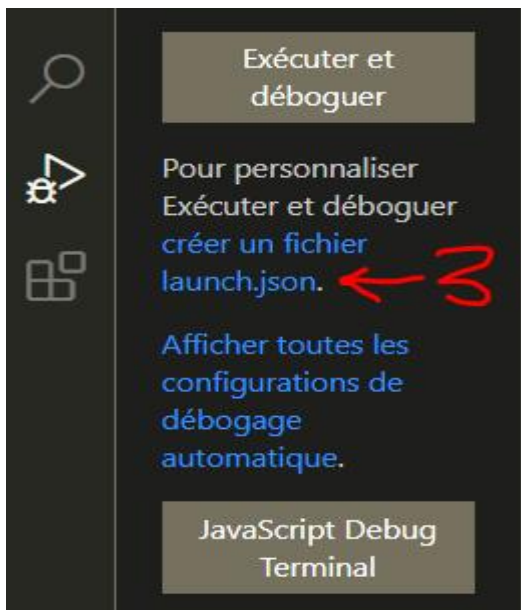
- Lancer VSCode, écrire votre programme javascript.

Pour l'exécuter aller sur le menu **Exécuter**->**Exécuter sans débogage** (ctrl+f5)-> un menu déroulant va être affiché-> choisissez **Node.js**.

Remarque : à chaque fois que vous voulez exécuter un script Js, le menu déroulant s'affichera pour pouvoir choisir l'environnement Node.js

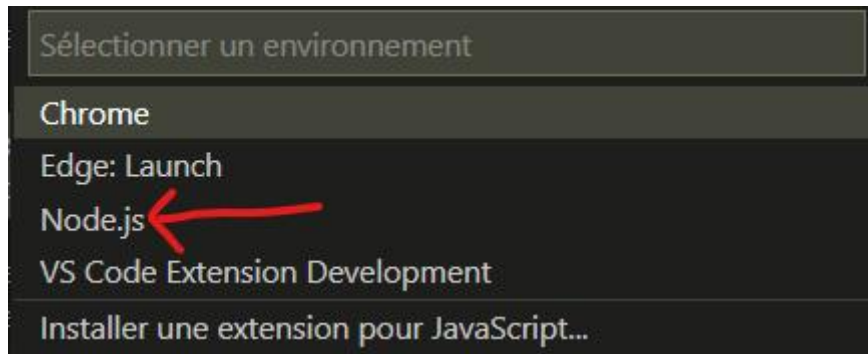
- **Solution** : Pour éviter de devoir sélectionner tout le temps Node.js à chaque exécution du programme, suivez les instructions suivantes :
Supposons que vous ayez un **dossier** nommé **Test-js** contenant un **fichier** **test.js**.
- Ouvrez le dossier **Test-JS** dans VSCode (cliquez bien sur le dossier et pas sur le fichier test.js !)

- a) À gauche dans la barre latérale, sélectionnez l'icône débogage (avec l'icône lecture et le petit insecte)



- b) À gauche, cliquez sur le lien "créer un fichier launch.json",

il va vous proposer un choix, sélectionnez **Node.js**. (si deux Node.js s'affichent, sélectionner le deuxième Node.js)



c) Il va ouvrir le fichier `.vscode/launch.json`.

- Si la propriété `"program"` vaut ceci : `"program": "${workspaceFolder}/test.js "`
Modifiez-la pour mettre à la place : `"program": "${file} "`

- Si la propriété `"type"` vaut ceci : `"type": "pwa-node"`
Modifiez-la pour mettre à la place : `"type": "node"`

d) Sauvegardez votre fichier `launch.json` et c'est bon ! Vous pouvez fermer ce fichier. La prochaine fois que vous cliquerez sur « Exécuter sans débogage » le menu déroulant ne s'affichera plus et l'exécution sera faite immédiatement.

e) Tester.

Si vous faites une fausse manipulation, ou que vous choisissiez autre chose que Node.js par erreur, vous pouvez supprimer le fichier `.vscode/launch.json` ou bien remplacer son contenu par ce qui suit :

```
// ----- Copiez à partir de la ligne ci-dessous -----
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "request": "launch",
      "name": "Déboguer",
      "skipFiles": [
        "<node_internals>/**"
      ],
      "program": "${file}"
    }
  ]
}
// ----- Copiez bien la dernière accolade juste au-dessus -----
```

3- Formater le code

- Assurer vous de l'installation de l'extension **Prettier code formater**
- Aller sur **Preferences -> Setting ->** rechercher :
 - **default formatter ->** choisissez Prettier
 - **format onSave->** cocher la case

4- Créer des snippets

Aller sur **File->Preferences->configure user snippets -> new global snippets file ->** saisissez un nom de fichier -> décommenter le code d'un snippet et modifier comme suit :

Exemple de snippet pour la commande console.log

```
// Example:
{
  "Print to console": {
    "scope": "javascript,typescript",
    "prefix": "cl",
    "body": ["console.log('$1');"],
    "description": "Log output to console"
  }
}
```

Tester.

Si l'auto-complétions est désactivée alors supprimer le \$1 qui permet d'avoir le curseur entre les parenthèses mais désactive l'auto-complétions

5- Emplacement du script JavaScript dans un fichier HTML

- a) Créer la page HTML suivante :

Ceci est une simple page HTML.

Elle est aussi basique que possible. Elle contient :

- Un titre H1
- Deux paragraphes
- Une liste non ordonnée



b) Soit le code JavaScript suivant : `alert (" Bonjour de JavaScript")`

Nous allons tester les différents cas possibles pour l'insertion de ce code dans un fichier HTML.

Donnez vos remarques pour chacun des cas suivants :

☞ Placer le code dans une balise **script**, **après** la balise de **titre** du document.
Exécuter. Remarques ?

☞ Déplacer la balise **script** **après** le premier **paragraphe**.
Exécuter. Remarques ?

☞ Déplacer la balise **script** juste **avant** la fin du **body**
Exécuter. Remarques ?

☞ Insérer le code JavaScript dans un fichier Js séparé puis le lier avec la page HTML et l'insérer avant la balise fermante de body.