

Création d'un environnement de développement

Tags: HTML TP

Editeur de texte

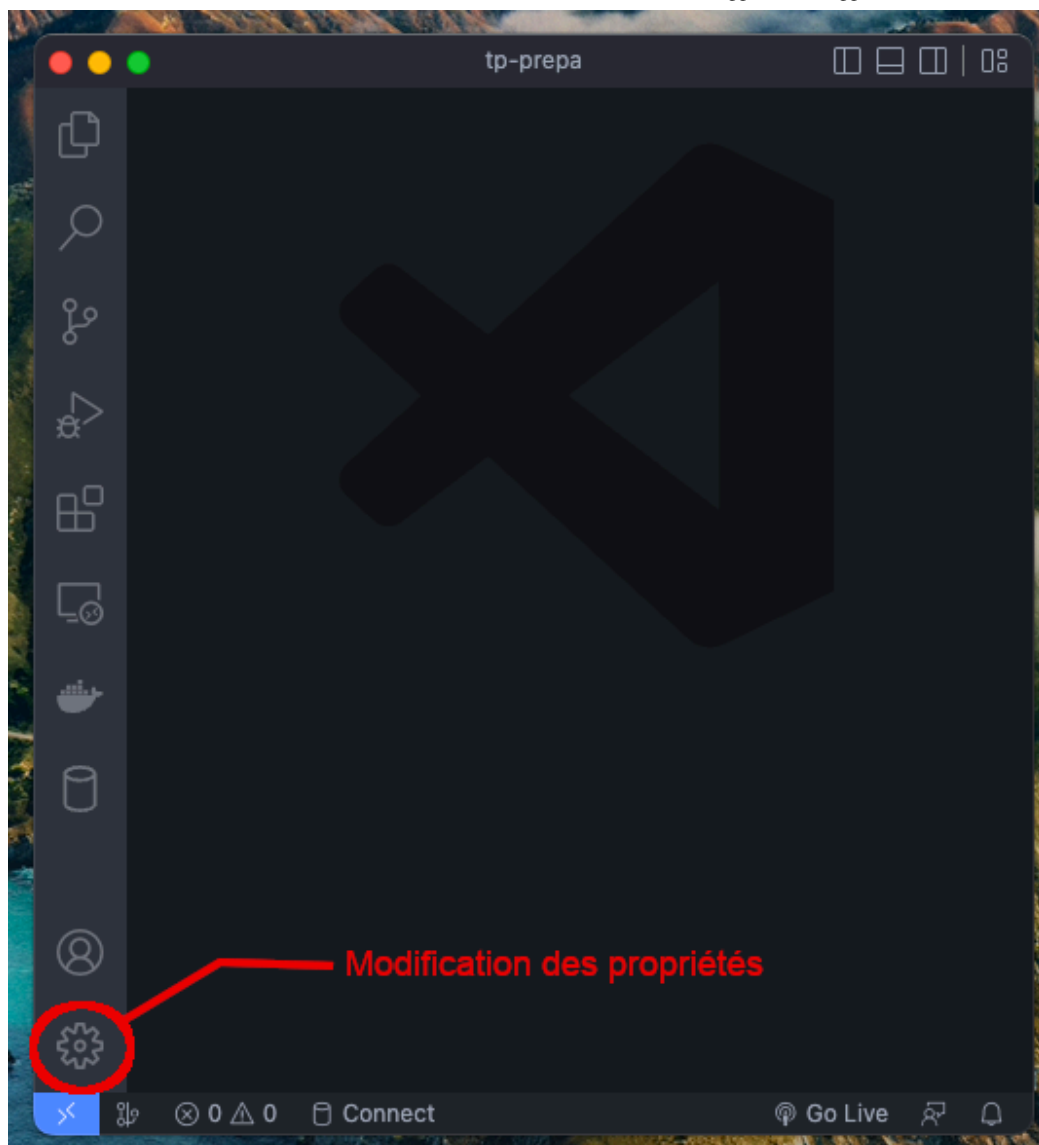
L'éditeur de texte [Visual Studio Code](#) permet entre autres choses de faire du développement d'application Web. Comme tous les outils, il faut une période d'apprentissage pour le maîtriser. C'est un choix que vous devez faire. L'intérêt de VS Code est dans son caractère universel. Vous pourrez l'utiliser dans de nombreux travaux de développements comme [python](#), [Java](#), [PHP](#), [C++](#), [javascript](#) et bien d'autres.

Il propose des composants complémentaires qui ajoutent des capacités et des facilités qui augmentent votre productivité. En voici une liste non exhaustive :

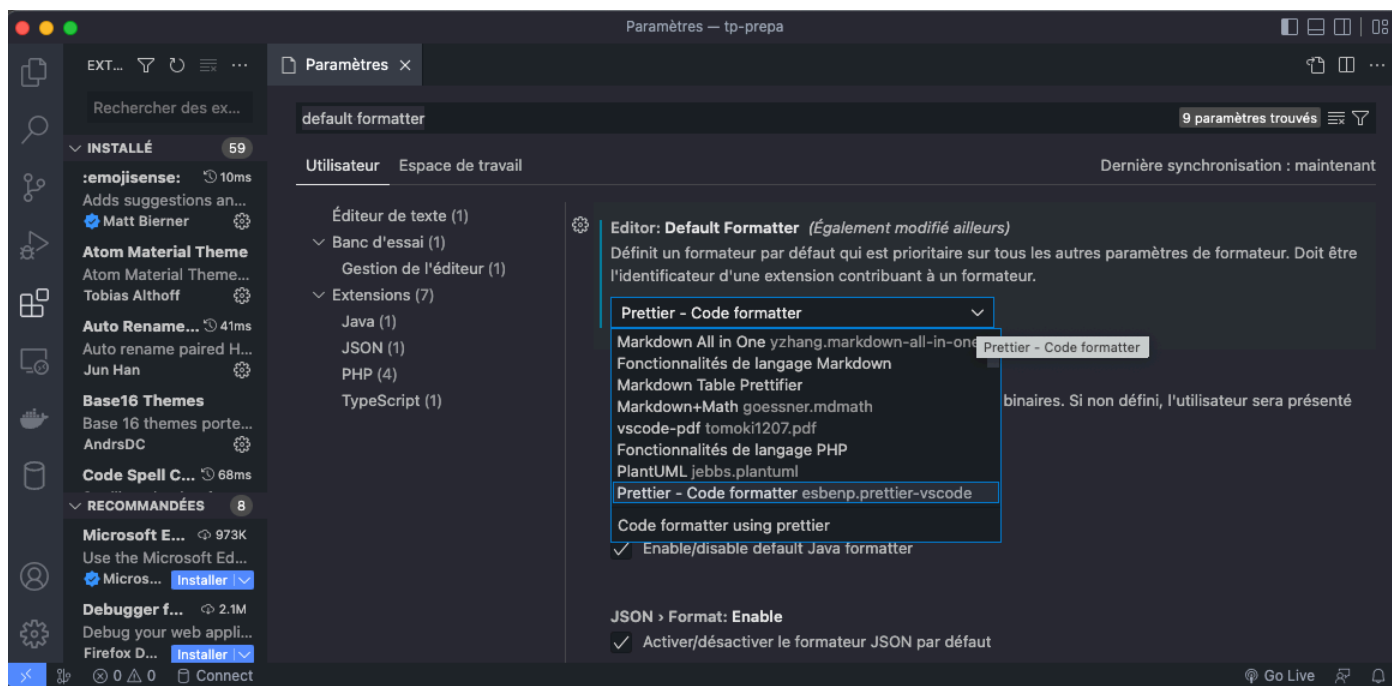
- [Indent Rainbow](#) : ajoute de la couleur à l'indentation.
- [AutoClose Tag](#) [Auto Rename Tag](#) : Propose les balises, éléments en fonction du début de la frappe.
- [VS Code Icons](#) : Ajoute des petites images en fonction de l'extension du fichier.
- [HTML Boilerplate](#) : Création d'une page HTML 5 vide.
- [Prettier](#) : Permet le formatage du document.
- [Markdown All in One](#) : Utilisation du langage de création de documentation.
- [Markdown Preview Enhanced](#) : Affichage du résultat après traduction.
- [GitLens](#) : Utilisation de `git`, le gestionnaire de version, dans votre éditeur.
- [Image preview](#) : Affichage de l'image et des informations.
- [Color Highlight](#) : Configuration des couleurs en CSS.
- [Live Server](#) : Serveur Web de test.
- Et bien d'autres voir l'article [VS Code Extensions For Web Dev Productivity](#).

Configuration de l'éditeur

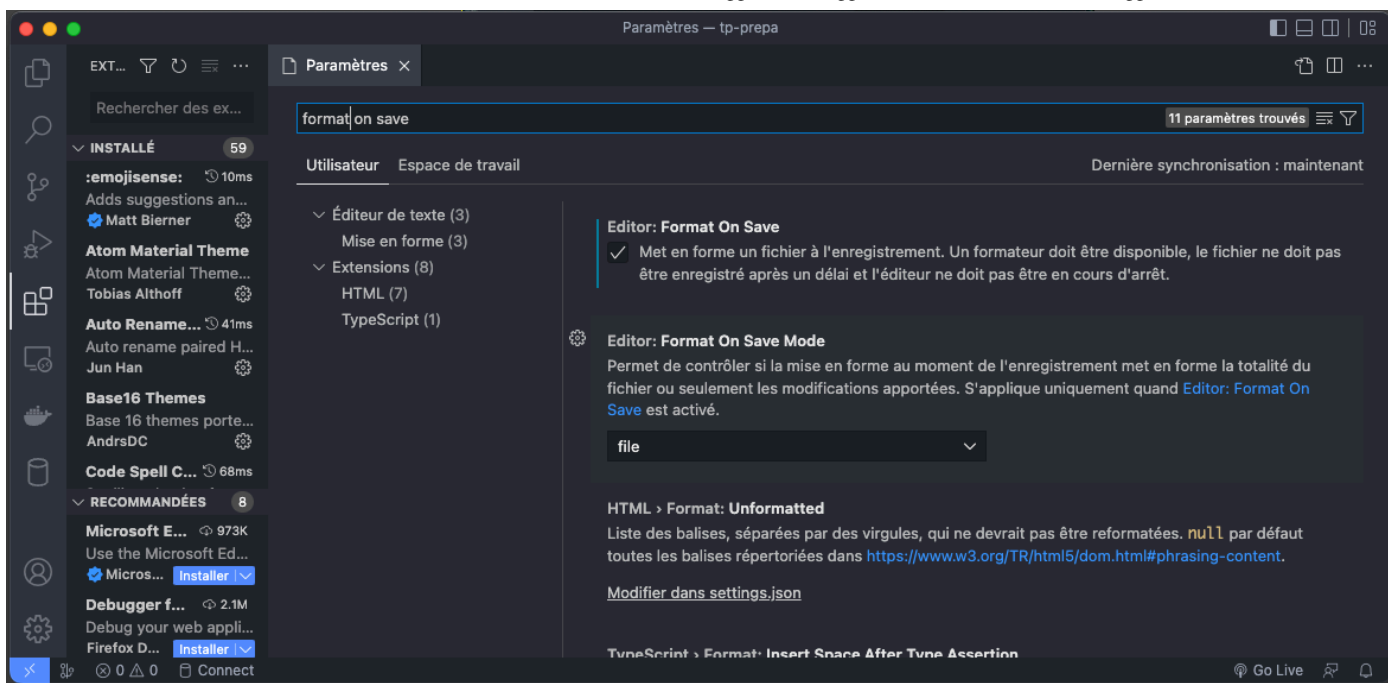
Nous allons modifier quelques propriétés de l'éditeur. Pour cela il faut cliquer sur la roue dentée (voir figure plus bas) et dans le menu choisir **Paramètres**.



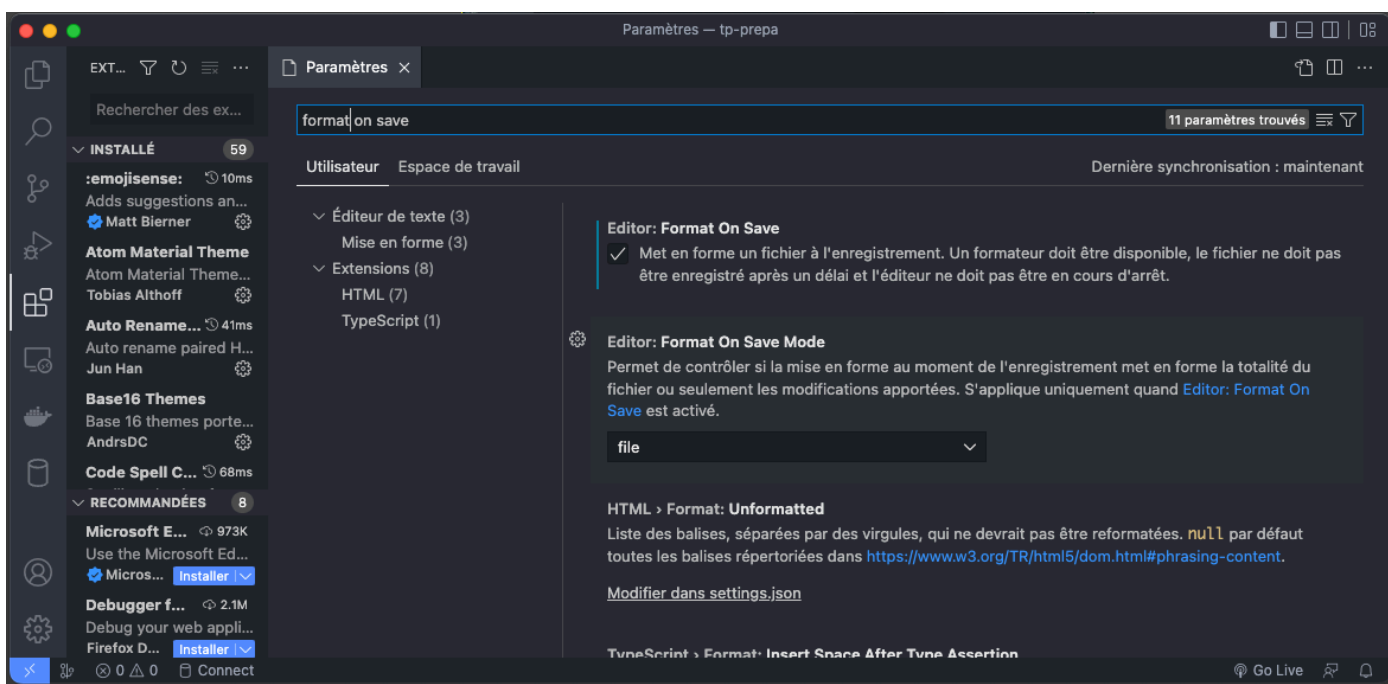
Dans la barre de recherche de l'éditeur saisir "default formatter" et sélectionner "Prettier - Code formatter"



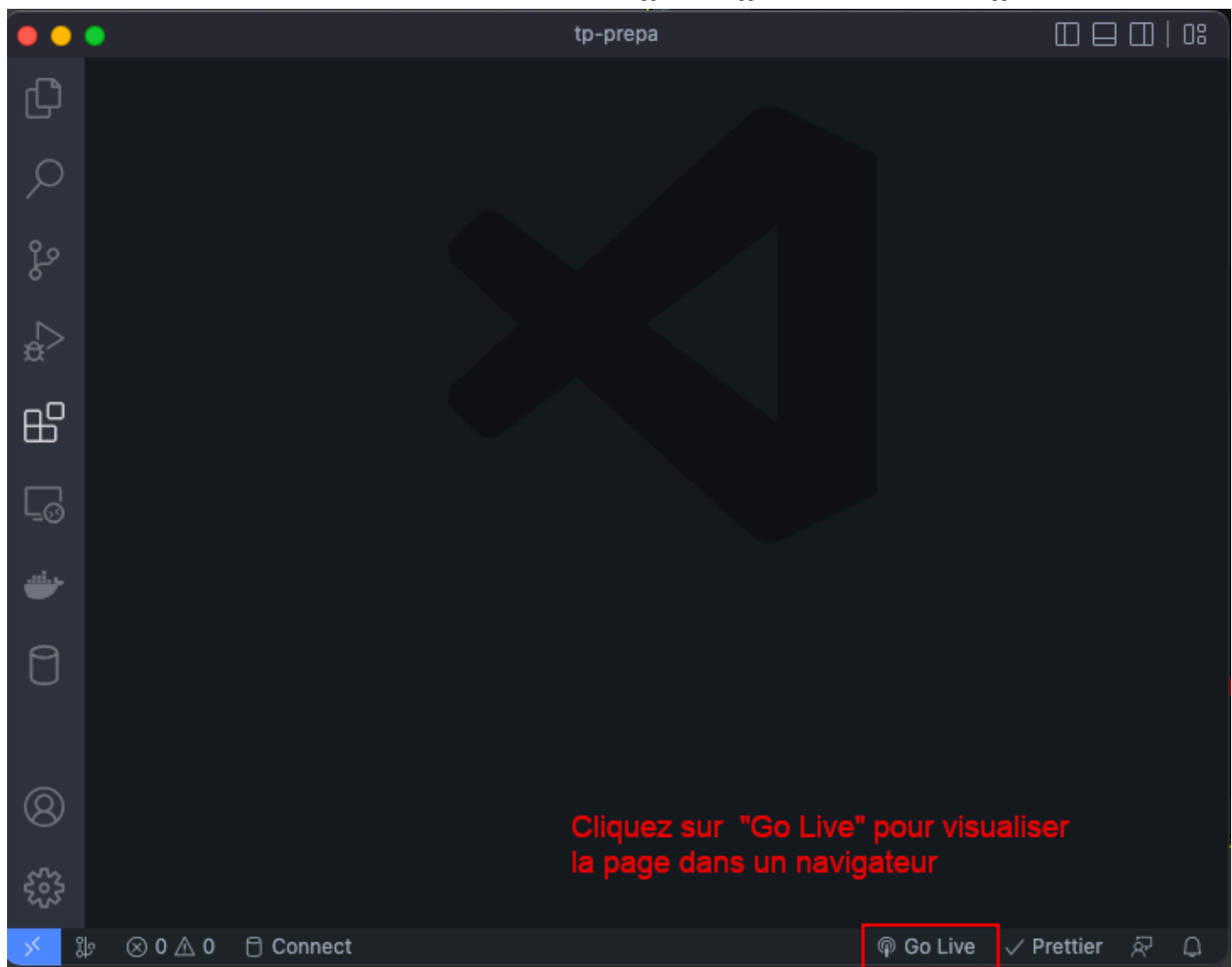
Dans la barre de recherche de l'éditeur saisir "format on save" et validez l'entrée.



Dans la barre de recherche de l'éditeur saisir "auto save" et sélectionner "onFocusSave"



Pour visualiser votre web en cours de modification, il suffit de lancer le serveur.



Mise en situation

Dans la suite du module, vous allez devoir créer des pages web qui associent des balises HTML, du code CSS et du code Javascript. Comme indiqué précédemment, il est important de faire le choix d'un environnement de développement qui contient :

- Un éditeur de texte (Environnement de Développement Intégré **EDI** ou **IDE** en anglais pour *Integrated Development Environment*)
- Un mécanisme de visualisation
- Un environnement de *debug*

Nous vous proposons d'utiliser *VS Code* comme éditeur de texte, votre navigateur comme outil de visualisation et le mécanisme de debug associé à *VS Code*.

Pour cela il faut :

- Récupérer *VS Code* (déjà disponible sur les machines de TP, ou dans l'installation de la rentrée),
- Installer les plugins proposés précédemment,
- Configurer l'outil de visualisation pour qu'il utilise votre navigateur,

- Configurer l'outil de debug.

Exercice 1

Question 1.1

Installez un environnement de développement qui vous permet simplement de créer une page html et qui vous permet de visualiser le résultat dans un navigateur.

Question 1.2

1. Créer un répertoire `DevWeb/Code/TP_prepa` qui sera le répertoire racine du projet pour la séance de TP. Dans le répertoire créez :
 - un répertoire `images` qui contiendra les images
 - un répertoire `styles` qui contiendra les fichiers de style
 - un répertoire `scripts` qui contiendra les sources javascript
2. Créez une page web `index.html` avec :
 - Le doctype HTML 5.
 - Précisez que le codage des caractères du document utilise l'encodage UTF-8.
 - Le titre "Travaux Pratiques - Préparation".
 - Le contenu de la page est en français
 - Ajouter une balise `h1` avec le contenu "Travaux Pratiques - Préparation"
 - Ajouter une balise `ul` qui contient 5 item avec un contenu aléatoire
 - Ajouter 10 paragraphes avec du texte aléatoire

Info Pratique

Il existe par défaut dans VS Code un mode [Emmet](#) qui permet de créer très rapidement et simplement ce type de contenu. Par exemple pour créer une liste non ordonnée avec un contenu aléatoire on utilisera :

```
ul>li*5>lorem10
```

Prenez un peu de temps pour comprendre et utiliser la [syntaxe des commandes Emmet](#). Vous en aurez encore besoin pour créer des maquettes de sites rapidement.

Question 1.3

1. Vérifiez le résultat en utilisant le serveur web de test.
2. Testez votre page web sur un site de validation comme celui de [W3C](#).

Nous allons ajouter des images dans la page. Pour créer des pages avec des figures dans la préparation d'une maquette, il existe des sites qui proposent des images aléatoires libres de droit d'utilisation. Voici une liste non exhaustive de sites qui proposent des images :

- [Koben](#) Site d'information.
- [Placeholder](#) propose des rectangles qui représentent la taille de l'image.
- [Lorem ipsum](#) Autre photos.

Question 1.4

1. Ajoutez une image dans votre page en utilisant une référence extérieure
2. Ajoutez une image dans votre page en utilisant une référence relative (dans le répertoire `images`)
3. Ajoutez un article qui contient un entête (`header`) avec un titre et un sous-titre suivi de 4 sections contenant chacune 2 paragraphes, une liste ordonnée avec 5 éléments aléatoires.

Question 1.5

1. Lancez le serveur web de test et vérifiez le résultat.
2. Testez votre page web sur un site de validation comme celui de [W3C](#).

Nous allons utiliser un fichier de style pour vérifier que l'environnement de développement fonctionne.

Voici un petit exemple de fichier de style :

```
body {  
    background-color: khaki;  
}
```

```
h1 {  
  border-bottom: 1px brown solid ;  
}  
  
img {  
  max-width: 100%;  
  height: auto;  
  border-radius: 8px;  
  box-shadow: 0 13px 27px -5px hsl(240deg 30% 28% / 25%), 0 8px 16px -8px hsl(0deg 0% 0% / 10%);  
}
```

Question 1.6

1. Ajoutez le contenu du listing précédent dans un fichier `style.css` du répertoire `css`.
2. Utilisez le fichier de style `style.css` créé précédemment dans votre fichier `index.html`.
3. Vérifiez le résultat en utilisant l'outil de debug.
4. Testez votre page web sur un site de validation comme celui de [W3C](#).
5. Testez le code css avec un site de validation comme celui de [W3C](#).

Nous allons maintenant ajouter un fichier javascript dans la page `index.html`.

HTML 5 propose des API qui permettent de dialoguer avec un environnement dans un cadre normé. C'est le cas pour le langage javascript.

Il y a plusieurs possibilités pour inclure du code javascript dans une page HTML5 :

- Dans la partie entête avec la balise script :

```
<script src="js/global.js"></script>
```

solution préférable ou

```
<script>  
alert("JavaScript c'est surpuissant !");  
</script>
```

- Dans le corps de la page html qu'il faut en général éviter.

Le lien entre le code HTML et un programme se fait en utilisant le DOM (*Document Object Model*) qui est une représentation dans la mémoire de la page HTML5 interprétée par le navigateur.

Le programme javascript est chargé dès qu'il apparait dans la page. Cela peut provoquer des lenteurs à l'affichage dans le navigateur. C'est pourquoi il est important d'associer avec la balise script l'attribut `async` ou `defer`.

- `async` : le code javascript est exécuté dès qu'il est disponible (sans vérifier que le reste du contenu est disponible) et ne bloque pas le chargement du reste de la page.
- `defer` : le code javascript est chargé dans l'ordre d'apparition dans le document et le navigateur exécutera le code après l'analyse du reste du code.

En conclusion l'attribut `defer` est le plus sûr, mais peut retarder l'exécution du code. S'il n'y a pas de dépendance entre morceaux de code `async` sera moins contraignant.

```
<script src="js/global.js" defer></script>
```

Question 1.7

1. Ajoutez le paragraphe suivant dans votre page

```
<p id="nom">Saisir votre nom : </p>
```

Soit le code javascript suivant :

```
let para = document.getElementById("nom");

para.addEventListener("click", updateName);

function updateName() {
  let name = prompt("Saisir un nouveau nom");
  para.textContent = "Collaborateur : " + name;
}
```

2. Placez le code dans l'entête de votre page HTML et rechargez la page. Que se passe-t-il ?

3. Placez le code juste avant la fermeture de la balise `body` de votre page HTML et rechargez la page. Que se passe-t-il ?

4. Supprimez le code précédent.

5. Créez un fichier `script.js` dans le répertoire `scripts` et importez-le dans votre page. Que se passe-t-il ?
6. Que faut-il utiliser comme attribut pour que cela fonctionne ?