

AC14.01 : Exploiter de manière autonome un environnement de développement efficace et productif

Un environnement de développement efficace est essentiel pour travailler rapidement et avec précision, tout en réduisant les erreurs. Visual Studio Code est un outil permettant de coder, avec des fonctionnalités pour travailler de manière productive, comme le remplissage automatique (auto-fill), des thèmes de couleurs et des raccourcis clavier.

Les outils de débogage, comme la console, permettent d'identifier, d'analyser et de corriger les erreurs dans le code. L'utilisation de ces outils est importante pour gagner du temps et permet d'obtenir le résultat final attendu.

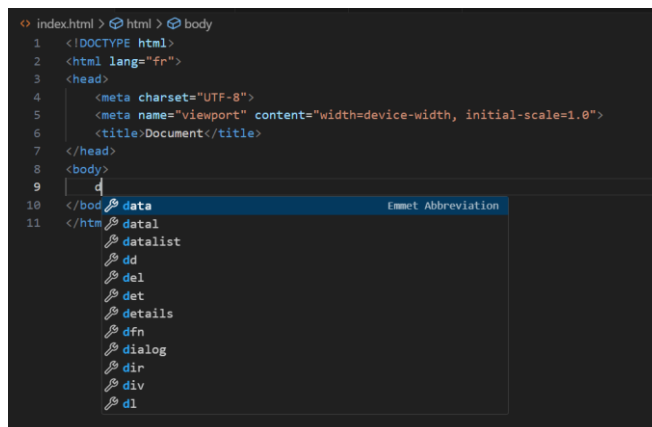
1. Remplissage automatique (Auto-fill)

Justification :

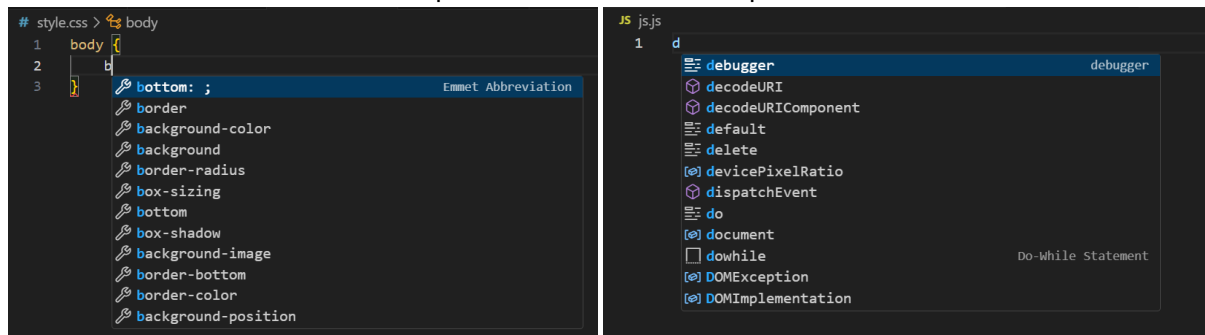
Le remplissage automatique accélère le codage en suggérant des propositions pour les mots ou expressions en cours de saisie. Cela réduit les erreurs de syntaxe et montre une cohérence au code. Il est utilisé pour des langages comme JavaScript, HTML ou CSS. Il permet aussi de gagner du temps en évitant de taper chaque mot en entier.

Exemple :

J'ai voulu écrire ouvrir une balise « div » et pour cela j'ai juste écrit « d » et il m'a proposé plusieurs balises dont la balise « div ».



C'est exactement la même chose pour le CSS et le Javascript :



2. Thèmes de couleurs et raccourcis clavier

Justifications :

- Thèmes de couleurs : L'utilisation de différents thèmes de couleurs permet d'améliorer la lisibilité du code. Les thèmes permettent aussi de mieux distinguer la syntaxe, en

colorant les mots-clés, variables et commentaires différemment. Cela rend le code plus facile à comprendre, ce qui permet également de réduire les erreurs.

- Raccourcis clavier : Les raccourcis clavier permettent d'exécuter des actions rapidement, sans avoir à naviguer dans les menus. Cela permet de gagner du temps et d'améliorer l'efficacité.

Exemples :

- Thèmes de couleurs :

```
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>SAE 105</title>
8 </head>
9
10 <link rel="stylesheet" href="style/style.css">
11
12 <body>
13
14   <header>
15     <a href="index.html" class="acc">Accueil</a>
16     <a href="page2.html" class="som">Sommaire</a>
17   </header>
18
```

```
style > # style.css > ...
1 body{
2   margin: 0;
3   max-height: 100%;
4 }
5
6 header{
7   background-color: black;
8   color: white;
9   display: flex;
10  padding: 20px;
11  font-family: Verdana, Geneva, Tahoma, sans-serif;
12  font-size: 20px;
13 }
```

```
js > JS scripts.js > ...
1 SAE["SAE1.01"].titre
2
3 Object.keys(SAE).forEach(function (numSAE) {
4   let compé = '';
5   SAE[numSAE].compétences.forEach(function (competence) {
6     compé += '<div class="competence">' + competence + '</div>';
7   });
8 }
```

- Raccourcis clavier :

« shift » + « Alt » + « f » : ce raccourci permet de disposer le code correctement.

« ctrl » + « / » : ce raccourci permet de commenter une ligne de code.

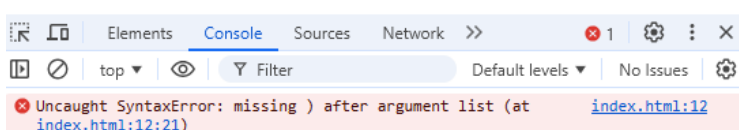
« ctrl » + « z » : ce raccourci permet d'annuler la dernière action.

3. Outils de débogage

Justification :

Les outils de débogage sont essentiels pour trouver et corriger les erreurs dans le code. Ils permettent d'analyser l'exécution du programme, de vérifier les variables et de repérer les erreurs de logique ou de syntaxe. La console des navigateurs web, par exemple, affiche les messages d'erreur, les avertissements et les informations utiles sur l'exécution du code. Ces outils accélèrent le débogage.

Exemple :



Ici il s'agit d'une faute de syntaxe à la ligne 12, il manquait une parenthèse.