

Apprendre le JavaScript - Techniques de débogage dans VS Code

Introduction

Le débogage est le processus de recherche et de correction d'erreurs dans un programme. Les erreurs peuvent être des erreurs de syntaxe, des erreurs de logique ou des erreurs d'exécution. Le débogage est une compétence essentielle pour tout développeur.

Techniques de débogage

1. Reproduire l'erreur

La première étape pour déboguer un programme est de reproduire l'erreur. Cela signifie que vous devez comprendre comment l'erreur se produit et dans quelles conditions elle se produit.

Est-ce que l'erreur se produit à chaque fois que vous exécutez le programme ? Est-ce que l'erreur se produit dans des conditions spécifiques ? Est-ce que l'erreur se produit après une certaine action de l'utilisateur ?

2. Vérifiez s'il y a une erreur dans la console

Si vous avez une erreur dans votre code, vous verrez un message d'erreur dans la console. Cela vous donnera des informations sur l'endroit où se trouve l'erreur et ce qui la cause. La ligne est indiquée dans le message d'erreur.

Vous pouvez la chercher dans un moteur de recherche pour trouver une solution.

Types d'erreurs courantes

- Vous avez mal écrit un mot-clé, une variable ou une fonction. Elle n'est pas reconnue par le programme.

L'erreur est "Uncaught ReferenceError: ... is not defined".

- Vous avez déclaré une variable mais vous ne l'avez pas initialisée avec let ou const.

L'erreur est "Uncaught ReferenceError: ... is not defined".

- Vous avez oublié une parenthèse, un crochet ou une accolade.

L'erreur est "Uncaught SyntaxError: ...".

- Vous avez déclaré deux fois la même variable.

L'erreur est "Uncaught SyntaxError: Identifier '...' has already been declared".

- Vous essayez de réaffecter une valeur à une constante.

L'erreur est "Uncaught TypeError: Assignment to constant variable".

- Vous avez sélectionné un élément qui n'existe pas dans le DOM et vous appelez une méthode sur cet élément. Revérifiez ce que vous avez mis dans le `querySelector`.

L'erreur est "Uncaught TypeError: Cannot read property '...' of null".

- Vous avez sélectionné plusieurs éléments HTML avec `querySelectorAll`. Il s'agit d'une `NodeList`, pas d'un élément HTML unique. Vous devez utiliser une boucle pour parcourir les éléments.

L'erreur est "Uncaught TypeError: ... is not a function".

- Votre événement se déclenche mais la fonction associée ne s'exécute pas. Vérifiez que vous avez bien passé la fonction en paramètre de la méthode `addEventListener`.

L'erreur est "Uncaught TypeError: ... is not a function".

- Votre événement se déclenche plusieurs fois. Vous avez mis un `addEventListener` dans une boucle sur le même élément.
- Le mot `NaN` apparaît dans votre programme. Cela signifie que vous avez fait une opération mathématique avec une valeur qui n'est pas un nombre. `NaN` = "Not a number". Soit vous avez oublié de convertir une chaîne de caractères en nombre, soit vous avez fait une opération mathématique avec une valeur qui n'est pas un nombre.

3. Commenter le code et le décommenter au fur et à mesure

Si vous n'avez pas de message d'erreur ou si vous n'arrivez pas à trouver, une autre technique de débogage consiste à commenter le code qui ne fonctionne pas et à le décommenter au fur et à mesure que vous le corrigez. Trouvez exactement la ligne qui cause l'erreur.

Est-ce que l'erreur se reproduit après avoir commenté une partie du code ? Est-ce que l'erreur disparaît après avoir décommenté une partie du code ?

4. Afficher des messages dans la console

Comme vu précédemment, la méthode la plus simple pour déboguer un programme est d'afficher des messages dans la console. Cela permet de vérifier les valeurs des variables à différents points du programme.

Affichez les tableaux pour voir leur contenu avec `console.table()`.

Affichez les objets pour voir leur contenu avec `console.dir()`.

5. Utiliser des points d'arrêt dans le navigateur

Les erreurs de logique et les erreurs d'exécution sont plus difficiles à trouver. Pour cela, vous pouvez utiliser des points d'arrêt dans le code.

En mettant le mot-clé `debugger` dans le code, le navigateur s'arrêtera à cet endroit lors de l'exécution du programme. Cela permet d'inspecter les valeurs des variables dans l'onglet "Sources" des outils de

développement du navigateur. Suivez alors les instructions pas à pas pour voir comment les valeurs des variables changent.

6. Utiliser l'onglet "Écouteurs d'événements" dans les outils de développement de Chrome

L'onglet "Écouteurs d'événements" dans les outils de développement de Chrome vous permet de voir tous les événements qui sont écoutés sur une page web. Cela peut vous aider à comprendre pourquoi un événement ne se déclenche pas.

7. Utiliser des extensions dans VS Code

Il existe des extensions dans VS Code qui peuvent vous aider à déboguer votre programme. Par exemple, l'extension "Debugger for Chrome" vous permet de déboguer votre programme JavaScript directement dans VS Code.

<https://raygun.com/learn/javascript-debugging-tips> <https://www.freecodecamp.org/news/how-to-debug-javascript-tips-and-tools/>