

Documentation sur "use strict" en JavaScript

Qu'est-ce que "use strict" ?

La directive "use strict" permet d'activer le mode strict en JavaScript, introduit avec *ECMAScript 5*.

Ce mode restreint certains comportements du langage pour rendre le code plus sûr, plus prévisible et faciliter la détection d'erreurs courantes lors du développement.

Comment activer le mode strict ?

- **Au niveau du script entier :**

Placez "use strict"; ou 'use strict'; tout en haut du fichier JavaScript, avant tout autre code exécutable (les commentaires sont autorisés avant).

```
"use strict";  
// Tout le code ici est en mode strict  
let x = 1;
```

- **Au niveau d'une fonction :**

Placez "use strict"; au début du corps de la fonction pour n'activer le mode strict que dans cette fonction.

```
function test() {  
  "use strict";  
  // Cette fonction est en mode strict  
}
```

Points importants sur l'utilisation

- "use strict" doit être la première instruction exécutable du script ou de la fonction.
- Il n'existe pas de directive pour désactiver le mode strict une fois activé dans un fichier ou une fonction.
- Le mode strict ne s'applique pas aux blocs de code {} isolés (comme dans les boucles ou les conditions), uniquement aux scripts ou fonctions.

Comportements modifiés en mode strict

- **Interdiction des variables non déclarées :**

Attribuer une valeur à une variable non déclarée génère une erreur.

- **this dans les fonctions :**

Dans une fonction classique appelée sans contexte, `this` vaut `undefined` au lieu de référencer l'objet global.

- **Écriture sur des propriétés en lecture seule ou non extensibles :**

Provoque une erreur au lieu d'être silencieusement ignorée.

- **Suppression de variables, fonctions ou arguments non supprimables :**

Génère une erreur.

- **Paramètres dupliqués dans une fonction :**

Provoque une erreur de syntaxe.

- **Certaines syntaxes interdites :**

Par exemple, l'utilisation de `with` est interdite.

Exemple d'erreur courante évitée

```
"use strict";  
x = 3.14; // Erreur : x n'est pas déclaré
```

Compatibilité

- Tous les navigateurs modernes supportent `"use strict"` (sauf IE9 et versions antérieures).

Résumé des avantages

- Détection plus rapide des erreurs (notamment les fautes de frappe).
- Code plus sécurisé et prévisible.
- Préparation à des évolutions du langage.