

### Étape 1 : Types d'erreurs

1. Créez un fichier **error\_handling.js**.
2. Dans ce fichier, créez une fonction qui génère une division par zéro en essayant de diviser un nombre par 0.
3. Invoquez la fonction et observez l'erreur générée dans la console.

### Étape 2 : Blocs try...catch

1. Modifiez la fonction pour l'entourer d'un bloc **try...catch**.
2. Dans le bloc **catch**, affichez un message d'erreur personnalisé à l'aide de **console.error**.

### Étape 3 : Erreurs personnalisées

1. Créez une nouvelle fonction qui génère une erreur personnalisée si une chaîne de caractères vide est passée en tant qu'argument.
2. Utilisez **throw** pour générer l'erreur personnalisée avec un message approprié.
3. Appelez la fonction avec différents types d'arguments.

### Étape 4 : Expérimentation et utilisation avancée

1. Expérimentez en créant d'autres types d'erreurs, tels que **ReferenceError** et **TypeError**.
2. Utilisez des blocs **try...catch** pour gérer ces erreurs et affichez des messages d'erreur personnalisés.

### Mini projet :

Créer un système de gestion des erreurs robuste pour une application web. L'objectif principal est d'identifier, capturer et gérer les erreurs et les bugs qui se produisent dans l'application, tout en fournissant des informations de débogage utiles pour les développeurs. Le système devra également permettre de suivre l'utilisation de l'application en enregistrant les erreurs dans un journal (log) ou en les envoyant à un serveur pour une analyse ultérieure.