

Module : JavaScript (M105) TP N° 5

Année de Formation 2023/2024

Filière : Développement digital

Groupe : DEV 102 - DEV 103

Niveau : 1ère année

Exercice 1 :

1. Créez une fonction **simulerTraitementAsynchrone** qui simule une opération **asynchrone** en utilisant **setTimeout**. La fonction doit prendre deux paramètres : un message et un délai (en millisecondes).
2. Utilisez un callback pour gérer le résultat de l'opération asynchrone. Le callback doit être appelé avec le message modifié (par exemple, en ajoutant "Traitement terminé: " au début du message).
3. Ensuite, créez une version de la fonction **simulerTraitementAsynchrone** qui renvoie une **promesse** au lieu d'utiliser un callback.

Exercice 2 :

1. Créez une fonction appelée **verifierNombrePair** qui prend un nombre en tant qu'argument.
2. À l'intérieur de la fonction, créez une promesse qui vérifie si le nombre est pair.
3. Si le nombre est pair, résolvez la promesse avec le message "Le nombre est pair".
4. Si le nombre est impair, rejetez la promesse avec le message "Le nombre est impair".

Exercice 3 :

1. Créez une fonction appelée **telechargerFichier** qui simule le téléchargement d'un fichier en utilisant une promesse.
2. La fonction doit prendre le nom du fichier en tant qu'argument.
3. Si le fichier est téléchargé avec succès, résolvez la promesse avec le message "Téléchargement réussi : {nom du fichier}".

4. Si une erreur survient pendant le téléchargement (simulez cela en rejetant la promesse), rejetez la promesse avec le message "Échec du téléchargement : {nom du fichier}".

Exercice 4 :

1. Créez une fonction appelée **diviserSécurisée** qui prend deux nombres en tant qu'arguments.
2. À l'intérieur de la fonction, divisez le premier nombre par le deuxième.
3. Utilisez la gestion des exceptions pour gérer les cas où le deuxième nombre est zéro.
4. Si la division est réussie, retournez le résultat de la division.
5. Si le deuxième nombre est zéro, lancez une exception avec le message "Division par zéro".

Exercice 5 :

1. Créez une fonction appelée **calculerInverse** qui prend un nombre en tant qu'argument.
2. À l'intérieur de la fonction, utilisez la gestion des exceptions pour calculer l'inverse du nombre.
3. Si le nombre est zéro, lancez une exception avec le message "Division par zéro".
4. Si le nombre est non numérique, lancez une exception avec le message "La valeur n'est pas un nombre".
5. Si tout est correct, retournez l'inverse du nombre.