



Welcome home !

Règlement intérieur

Poster une issue

JavaScript

Environnement de dev

👋 Semaine 1

Terminal

Variables

Conditions

Boucles

👋 Semaine 2

Manipuler des chaînes de caractères

Fonctions

🎥 Vidéos

Introduction et concept

Notations possibles

Déclaration et Appel

Ajouter des paramètres à vos fonctions

Valeur de retour

Tableaux

👋 Semaine 3

Déroulement d'un programme

Les booléens

Les valeurs truthy et falsy

Notion de scope

Conditions d'existence

Conditions imbriquées

👋 Semaine 4

Objets

Requêtes à API

En JavaScript, vous avez la possibilité de rendre vos fonctions plus souples, et de faire en sorte de les utiliser dans différents contextes, en leur fournissant des informations complémentaires.

Vous pourrez donner à votre fonctions des informations au moment où vous l'appeler pour l'exécuter, et son exécution dépendra des arguments que vous lui aurez fourni !

Utiliser un paramètre

Comme nous le disions, les paramètres sont des valeurs que vous pouvez passer à une fonction. Ces paramètres permettent à la fonction d'accepter des entrées spécifiques et de générer une sortie basée sur ces entrées.

L'utilisation de paramètres dans une fonction peut rendre votre code plus flexible et réutilisable. En effet, les paramètres permettent à une même fonction d'être appelée avec différentes entrées, sans avoir à écrire une nouvelle fonction pour chaque ensemble d'entrées.

Pour illustrer cela, reprenons l'exemple de la fonction `displayMessage`. Cette fonction ne prenait aucun paramètre et affichait simplement un message "Bonjour!" dans la console.

Si vous voulez modifier cette fonction pour qu'elle affiche un message spécifique en fonction de ce vous lui fournissez, vous pouvez ajouter un paramètre :

```
const displayMessage = (message) => {  
  console.log(message);  
}
```