

Apprendre le JavaScript - La persistance des données

Un des grands défis de la programmation est de conserver les données d'une exécution à l'autre ou d'une page à l'autre.

Pour pallier à ce problème, il est possible de stocker les données dans un fichier ou une variable globale à l'aide d'un objet ou d'un tableau. Cependant, dès que le programme se termine, les données modifiées sont perdues et remise à la valeur initiale.

Généralement, pour conserver les données, il est possible de les stocker dans une base de données ou dans le stockage local du navigateur.

Utiliser une variable globale

Une variable globale est une variable qui est accessible de partout dans le programme. Cela permet de stocker des données et de les conserver d'une exécution à l'autre. Cependant, dès que le programme se termine, les données sont perdues. Cela permet de stocker des données de configuration, des données de session ou des données de l'utilisateur.

Il n'est généralement pas recommandé d'utiliser des variables globales pour stocker des données. En effet, cela peut rendre le programme difficile à comprendre et à maintenir. Il est préférable d'utiliser des variables locales pour stocker des données.

Stocker les données dans un fichier

Il est possible de placer des données dans un tableau ou un objet et de les stocker dans un fichier. Cela permet de les lire et de les afficher dans le programme. Il est plus complexe d'écrire le fichier et de le mettre à jour. Ce n'est pas nécessairement une bonne pratique car cela peut rendre le programme difficile à comprendre et à maintenir.

Stocker les données dans une base de données

Une base de données est un système de stockage qui permet de stocker des données de manière structurée. Cela permet de conserver les données d'une exécution à l'autre. Il est possible de stocker des données sous forme de tables, de lignes et de colonnes. Cela permet de stocker des données de configuration, des données de session ou des données de l'utilisateur.

Vous verrez comment interagir avec une base de données dans le cours 31F. Pour l'instant, nous allons voir comment stocker des données dans le stockage local du navigateur.

Stockage local - LocalStorage

Le stockage local est une fonctionnalité du navigateur qui permet de stocker des données dans le navigateur. Cela permet de conserver les données d'une exécution à l'autre. Il est possible de stocker des données sous

forme de clé-valeur. Les données sont stockées dans le navigateur et sont accessibles de partout dans le programme.

Si la fenêtre du navigateur est fermée, les données sont conservées. Cela permet de stocker des données de configuration, des données de session ou des données de l'utilisateur. Cependant, si l'utilisateur efface les données de navigation, les données stockées sont perdues.

Ajouter des données

On peut ajouter des données dans le stockage local avec la méthode `setItem()`. Il est possible de stocker des données sous forme de chaîne de caractères. Pour stocker des objets, il est nécessaire de les convertir en chaîne de caractères avec `JSON.stringify()`.

`JSON.stringify()` permet de convertir un objet en chaîne de caractères. C'est une façon fiable de passer de la donnée complexe vers une base de données ou le stockage local.

```
// Stockage local
localStorage.setItem("nom", "John");
localStorage.setItem("age", 30);
let utilisateur = { nom: "John", age: 30 };
localStorage.setItem("utilisateur", JSON.stringify(utilisateur));
```

Le format JSON

Le format JSON (JavaScript Object Notation) est un format de données textuelles qui permet de représenter des objets JavaScript. C'est un format simple et léger qui est facile à lire et à écrire. Il est utilisé pour échanger des données entre un serveur et un client, pour stocker des données dans une base de données ou pour stocker des données dans le stockage local. Il est basé sur la syntaxe des objets JavaScript.

- Les objets sont délimités par des accolades `{}`.
- Les propriétés sont séparées par des virgules `,`, sauf la dernière.
- Les propriétés sont des paires clé-valeur.
- Les clés sont des chaînes de caractères.
- Les valeurs peuvent être des chaînes de caractères, des nombres, des booléens, des tableaux ou des objets.
- Les tableaux sont délimités par des crochets `[]`.

```
{
  "nom": "John",
  "age": 30,
  "utilisateur": {
    "nom": "John",
    "age": 30
  }
}
```

Récupérer des données

On peut récupérer des données du stockage local avec la méthode `getItem()`. Il est possible de récupérer des données sous forme de chaîne de caractères. Pour récupérer des objets, il est nécessaire de les convertir en objet avec `JSON.parse()`.

`JSON.parse()` permet de convertir une chaîne de caractères en objet. C'est l'opération inverse de `JSON.stringify()`. Cela permet de récupérer des données complexes stockées dans une base de données ou le stockage local. Attention, si la chaîne de caractères n'est pas un objet, cela génère une erreur.

```
// Stockage local
let nom = localStorage.getItem("nom");
let age = localStorage.getItem("age");
let utilisateur = JSON.parse(localStorage.getItem("utilisateur"));
```

Supprimer des données

On peut supprimer des données du stockage local avec la méthode `removeItem()`. On passe en paramètre la clé de la donnée à supprimer.

```
// Stockage local
localStorage.removeItem("nom");
localStorage.removeItem("age");
localStorage.removeItem("utilisateur");
```

Effacer toutes les données

Finalement, on peut effacer toutes les données du stockage local avec la méthode `clear()`. Cela permet de supprimer toutes les données du stockage local.

```
// Stockage local
localStorage.clear();
```