

Apprendre JavaScript - Objets

Table des matières

- [Introduction](#)
- [Création d'un objet](#)
- [Contexte d'utilisation](#)
- [Créer un tableau d'objets](#)
- [Accéder à une propriété](#)
 - [Meilleures pratiques](#)
- [Ajouter ou modifier une propriété existante](#)
- [Supprimer une propriété](#)
- [Parcourir un objet](#)
 - [méthode `Object.keys`](#)
 - [Méthode `Object.values`](#)
 - [Méthode `Object.entries`](#)

Introduction

Les objets sont des collections de propriétés, et une propriété est une association entre un nom (ou une clé) et une valeur. Une valeur d'une propriété peut être une fonction, auquel cas la propriété est connue sous le nom de méthode.

Les objets permettent de stocker des collections de données et des fonctionnalités dans un ordre qui n'est pas nécessairement linéaire.

Ce sont un peu l'équivalent des tableaux associatifs en PHP.

Création d'un objet

Pour créer un objet, vous pouvez utiliser la syntaxe suivante :

```
let personne = {
  prenom: "John",
  nom: "Doe",
  age: 25,
  adresse: {
    rue: "Rue de la Paix",
    ville: "Paris",
    codePostal: "75000",
  },
  direBonjour: function () {
    console.log("Bonjour!");
  },
};
```

Contexte d'utilisation

Vous allez bientôt utiliser des bases de données pour stocker des données. Les objets sont un excellent moyen de stocker des données dans une base de données, car ils permettent de stocker des données de manière structurée et de les manipuler facilement.

De plus, lorsque vous récupérez des données d'une base de données, elles sont souvent sous forme d'objets.

Créer un tableau d'objets

Vous pouvez créer un tableau d'objets en utilisant la syntaxe suivante :

```
let personnes = [  
  {  
    prenom: "John",  
    nom: "Doe",  
    age: 25,  
  },  
  {  
    prenom: "Jane",  
    nom: "Smith",  
    age: 30,  
  },  
  {  
    prenom: "James",  
    nom: "Brown",  
    age: 35,  
  },  
];
```

Accéder à une propriété

Pour accéder à une propriété d'un objet, vous pouvez utiliser la notation pointée ou la notation crochet.

```
let prenom = personne.prenom;  
let nom = personne["nom"];  
  
console.log(prenom); // John  
console.log(nom); // Doe
```

Si la propriété a été définie avec des espaces, des caractères spéciaux ou des mots-clés réservés, vous devez utiliser la notation crochet.

```
let personne = {
  prénom: "John",
  "nom de famille": "Doe",
  âge: 25,
  adresse: {
    rue: "Rue de la Paix",
    ville: "Paris",
    "code postal": "75000",
  },
  "dire bonjour": function () {
    console.log("Bonjour!");
  },
};

let prenom = personne["nom de famille"];
```

Meilleures pratiques

Il est préférable d'utiliser la notation pointée pour accéder à une propriété d'un objet, sauf si vous avez une bonne raison d'utiliser la notation crochet. Donc nommez vos propriétés avec des noms simples et évitez les espaces, les caractères spéciaux et les mots-clés réservés.

Ajouter ou modifier une propriété existante

Pour modifier la valeur d'une propriété d'un objet, vous pouvez utiliser la notation pointée ou la notation crochet. Si la propriété n'existe pas, elle sera ajoutée à l'objet.

```
personne.prenom = "Jane";
personne["nom de famille"] = "Smith";
personne.telephone = "0123456789"; // Ajoute une nouvelle propriété
```

Supprimer une propriété

Pour supprimer une propriété d'un objet, vous pouvez utiliser l'opérateur `delete`.

```
delete personne.telephone;
```

Parcourir un objet

Comme vu dans le cours sur les boucles, pour parcourir les propriétés d'un objet, vous pouvez utiliser une boucle `for...in`.

```
for (let propriete in personne) {  
    console.log(propriete + ": " + personne[propriete]);  
}
```

méthode `Object.keys`

La méthode `Object.keys` renvoie un tableau contenant les noms des propriétés propres à un objet. Cela vous permet de parcourir les propriétés d'un objet en utilisant une boucle `for`.

```
let proprietes = Object.keys(personne);  
  
for (let i = 0; i < proprietes.length; i++) {  
    console.log(proprietes[i] + ": " + personne[proprietes[i]]);  
}
```

Méthode `Object.values`

La méthode `Object.values` renvoie un tableau contenant les valeurs des propriétés propres à un objet.

```
let valeurs = Object.values(personne);  
  
for (let i = 0; i < valeurs.length; i++) {  
    console.log(valeurs[i]);  
}
```

Méthode `Object.entries`

La méthode `Object.entries` renvoie un tableau contenant des paires de clés-valeurs pour chaque propriété propre à un objet. Chaque paire est un tableau (Array) de deux éléments.

```
let entrees = Object.entries(personne);  
  
for (let i = 0; i < entrees.length; i++) {  
    console.log(entrees[i][0] + ": " + entrees[i][1]);  
}
```