

Fusionner des objets

Fusionner des objets avec la méthode assign()

La méthode `Object.assign()` permet de copier toutes les propriétés d'un ou plusieurs objets sur un autre objet cible.

La syntaxe est :

```
Object.assign(objetCible, objetACopier1, objetACopier2...)
```



Par exemple, vous souhaitez copier toutes les propriétés de deux objets dans un nouvel objet littéral :

```
const objet1 = {a: 1, b: 2};  
const objet2 = {c: 3};  
const résultat = Object.assign({}, objet1, objet2);  
console.log(résultat); // {a: 1, b: 2, c: 3}
```



Autre exemple, vous souhaitez ajouter les propriétés d'un objet sur un objet existant :

```
const objet1 = {a: 1, b: 2};  
const objet2 = {c: 3};  
Object.assign(objet1, objet2);  
  
console.log(objet1); // {a: 1, b: 2, c: 3}
```



A noter que si l'objet ciblé et / ou les objets sources ont des noms de propriété en commun, alors ce sera la dernière valeur qui sera utilisée :

```
const objet1 = {a: 1};  
const objet2 = {a: 2};  
const objet3 = {a: 3};  
const résultat= Object.assign({}, objet1, objet2, objet3);  
console.log(résultat); // {a: 3}
```



Fusionner des objets avec l'opérateur Spread ...

L'opérateur `Spread` a la même syntaxe que l'opérateur `Rest` : `...`.

La différence entre les deux est le contexte d'utilisation, ils n'ont pas le même effet.

L'opérateur spread , permet d'étendre un objet, c'est-à-dire de lister ses propriétés.

Cela peut paraître obscure au premier abord, mais c'est très simple en réalité. Vous pouvez l'utiliser pour copier un objet dans un nouvel objet :

```
const monObj = {a: 1};  
const copie = {...monObj};  
  
copie.a = 2;  
console.log(monObj); // {a: 1}  
console.log(copie); // {a: 2}
```



Vous pouvez aussi l'utiliser pour fusionner deux ou plusieurs objets :

```
const monObj = {a: 1};  
const monObj2 = {b: 2};  
const copie = {...monObj, ...monObj2};  
  
copie.a = 2;  
console.log(monObj); // {a: 1}  
console.log(copie); // {a: 2, b: 2}
```

