

Définir ou lier this

L'utilisation de `this` avec les fonctions fléchées et le mode strict est simple, et vous aurez rarement besoin d'utiliser ce qui va suivre.

Cependant, dans certains cas d'utilisation avancée, vous souhaitez modifier le comportement par défaut de `this`.

Notre recommandation est de bien réfléchir avant d'utiliser les méthodes suivantes car il y a probablement un autre moyen plus simple.

Définir explicitement la valeur de this

Il est possible de définir la valeur de `this` lorsque l'on invoque une fonction.

Il suffit d'utiliser `call()` ou `apply()` et de passer en premier argument la valeur de `this`.

```
"use strict"

const personne = { name: 'Jean Dupont' };

function formuleSalutation(formule) {
  return this.name + formule;
}

formuleSalutation(' vous saluez'); // TypeError: Cannot read property 'name' of undefined
formuleSalutation.call(personne, ' vous saluez'); // => 'Jean Dupont vous saluez'
```

Normalement, comme vous le savez, le `this` dans un appel de fonction simple vaut `undefined` en mode strict.

Parfois, nous voulons lui donner une autre valeur, comme dans notre exemple, nous voulons lui donner la référence d'un objet.

La seule différence avec `apply()` est que les autres arguments doivent être passés dans un tableau. Aussi, vous utiliserez `call()` la plupart du temps.

Lier la valeur de this

La méthode `bind()` permet de créer un clone d'une fonction en liant définitivement la valeur de `this` à l'argument passé en premier paramètre.

```
function multiplier(nombre) {
```



```
    return this * nombre;
}

const doubler = multiplier.bind(2); // définir this comme étant la valeur 2
console.log(doubler(4)); // 8
```