

La propriété `this` et la méthode `bind`

La propriété `this` est un mot-clé qui fait référence à l'objet courant. En JavaScript, la valeur de `this` dépend de la manière dont une fonction est appelée et du contexte dans lequel elle est appelée.

La vidéo suivante explique en détail le fonctionnement de la propriété `this` en JavaScript :

<https://www.youtube.com/watch?v=fVXp7ZWjIO4>

Contexte global

Lorsqu'une fonction est appelée dans le contexte global, la valeur de `this` est l'objet global (`window` en navigateur).

```
function afficherThis() {  
  console.log(this);  
}  
  
afficherThis(); // Affiche `window` en navigateur, `global` en Node.js
```

Contexte d'un objet

Lorsqu'une fonction est appelée en tant que méthode d'un objet, la valeur de `this` est l'objet sur lequel la méthode est appelée.

```
const personne = {  
  prenom: "Maxime",  
  nom: "Lacasse",  
  afficherNom() {  
    console.log(this.prenom, this.nom);  
  },  
};  
  
personne.afficherNom(); // Affiche "Maxime Lacasse"
```

Contexte d'une classe

Lorsqu'une méthode d'une classe est appelée, la valeur de `this` est l'instance de la classe sur laquelle la méthode est appelée.

```
class Personne {  
  constructor(prenom, nom) {
```

```

        this.prenom = prenom;
        this.nom = nom;
    }

    afficherNom() {
        console.log(this.prenom, this.nom);
    }
}

const personne1 = new Personne("Maxime", "Lacasse"); // this = personne1
personne1.afficherNom(); // Affiche "Maxime Lacasse"

```

Contexte d'un écouteur d'événement

Lorsqu'une fonction est utilisée comme écouteur d'événement, la valeur de `this` est l'élément HTML sur lequel l'événement est déclenché.

```

const bouton = document.querySelector("button");
bouton.addEventListener("click", function () {
    console.log(this); // Affiche l'élément HTML du bouton
    const declencheur = this;
    const declencheur2 = event.currentTarget;
    // this et event.currentTarget sont équivalents
});

```

Contexte d'une fonction fléchée

Lorsque vous devez utiliser `this`, il est préférable de ne pas utiliser de fonction fléchée, car elle ne crée pas son propre contexte. La valeur de `this` dans une fonction fléchée est celle du contexte dans lequel elle a été définie. Pour éviter les problèmes de contexte, il est recommandé d'utiliser une fonction normale.

Bind

La méthode `bind` permet de lier un contexte personnalisé à une fonction. Elle prend en premier argument le contexte de la fonction, c'est à dire l'objet sur lequel la fonction sera appelée, et les paramètres suivants sont les paramètres de la fonction.

Cela sera particulièrement utile dans avec les classes qui auront des méthodes qui seront appelées par des écouteurs d'événements.

```

// L'objet `event` est toujours le dernier paramètre
function clonerElement(evenement) {
    const declencheur = evenement.currentTarget;
    //this ici représente l'élément HTML qui est mis dans le bind
}

```

```
}
```

```
const body = document.querySelector("body");
```

```
body.addEventListener("click", clonerElement.bind(body));
```