

Les fonctions parent/enfant props avancée

La classe parent app **transmet sa fonction** `_addArticle()` au composant enfant `<form>` via le props `'_addArticle'`, on précise `'this'` pour préciser qu'on parle de la fonction du parent qui s'exécute, pour récupérer les information de l'enfant on précise `'this'` dans le context enfant.

```
JS App.js x JS Form.js
1 import React, { Component } from 'react';
2
3 import '../App.css';
4 import Form from './Form';
5 import ItemList from './ItemList';
6
7
8 class App extends Component {
9
10   state = { articles : [] }
11
12   _addArticle = (article) => {
13     this.setState({ articles : [...this.state.articles, article]});
14     article.id = this.state.articles.length + 1;
15     console.log(this.state.articles);
16   }
17
18   render() {
19     return (
20       <div>
21         <h2>Liste de course en construction</h2>
22         < Form formTitle="Ajouter un article" _addArticle={this._addArticle} />
23         < ItemList />
24       </div>
25     );
26   }
27 }
```

Résultat

```
Line 24: No duplicate props allowed
{
  name: "pain", quantity: 0
}
id: 1
name: "pain"
quantity: 0
__proto__: Object
```

```
JS App.js JS Form.js x
1 import React from 'react';
2
3 class Form extends React.Component{
4
5   state = {
6     name: ' ',
7     quantity: 0
8   };
9
10   _handleSubmit = (oEvt) => {
11     oEvt.preventDefault();
12     console.log(this.state);
13     this.props._addArticle(this.state);
14   }
15
16   render(){
17     return(
18       <div>
19         <h3>{this.props.formTitle}</h3>
20         <form onSubmit={this._handleSubmit} >
21           <input type="number" placeholder="q"
22             value={this.state.quantity} onChange=
23             <input type="text" placeholder="art
24             value={this.state.name} onChange=({
25             <button type="submit">ajouter</butt
26         </form>
27       </div>
28     )
29   }
30 }
31
32 export default Form;
33
```