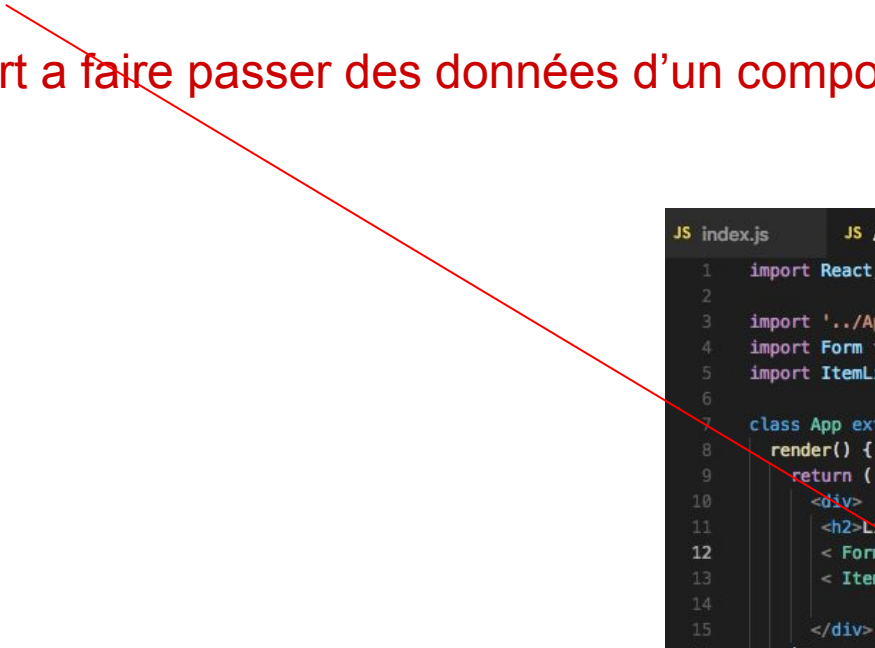


Les props simple

Les props sont les paramètres que l'on donne à nos composant 'class' quand ils sont instanciés dans un composant parent.

Les props sert a faire passer des données d'un composant parent a un composant enfant.



```
JS index.js JS App.js x JS Form.js JS ItemL
1 import React, { Component } from 'react';
2
3 import '../App.css';
4 import Form from './Form';
5 import ItemList from './ItemList';
6
7 class App extends Component {
8   render() {
9     return (
10      <div>
11        <h2>Liste de course en construction</h2>
12        < Form formTitle="le good titre" />
13        < ItemList />
14      </div>
15    );
16  }
17 }
18
19 export default App;
20
21
```

On peut définir un 'props' dans le composant pour ensuite l'utiliser comme paramètre dans le composant parent

Note: On peut nommer le props comme on le souhaite.

```
class Form extends React.Component{

  state = {
    name: ' ',
    quantity: 0
  };

  _handleSubmit = (oEvt) =>{
    oEvt.preventDefault();
    console.log(this.state);
  }

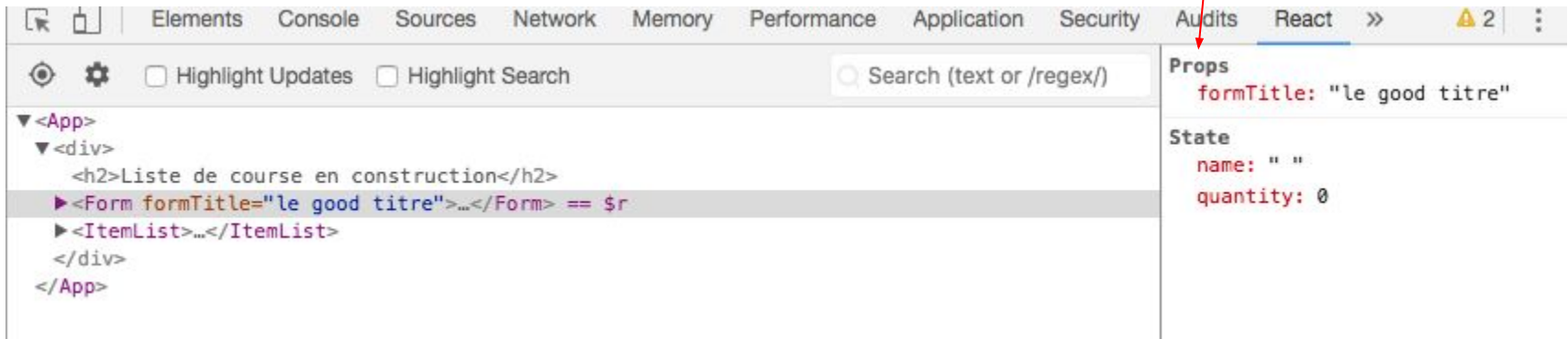
  render(){
    return(
      <div>
        <h3>{this.props.formTitle}</h3>
        <form onSubmit={this._handleSubmit}>
          <input type="number" placeholder="quantité"
            value={this.state.quantity} onChange={this._handleQuantityChange}/>
          <input type="text" placeholder="article"
            value={this.state.name} onChange={this._handleNameChange}/>
          <button type="submit">ajouter</button>
        </form>
      </div>
    )
  }
}

export default Form;
```

```
JS index.js  JS App.js  x  JS Form.js  JS ItemL

1  import React, { Component } from 'react';
2
3  import '../App.css';
4  import Form from './Form';
5  import ItemList from './ItemList';
6
7  class App extends Component {
8    render() {
9      return (
10       <div>
11         <h2>Liste de course en construction</h2>
12         <Form formTitle="le good titre" />
13         <ItemList />
14       </div>
15     );
16   }
17 }
18
19
20 export default App;
21
```

Maintenant notre composant <Form> a des props



The screenshot shows the React DevTools component inspector. The left pane displays the component tree with the following structure:

```
<App>  
  <div>  
    <h2>Liste de course en construction</h2>  
    <Form formTitle="le good titre">...</Form> == $r  
    <ItemList>...</ItemList>  
  </div>  
</App>
```

The `<Form formTitle="le good titre">...</Form>` component is selected and highlighted. The right pane shows the props and state for this component:

Props

- `formTitle`: "le good titre"

State

- `name`: ""
- `quantity`: 0

A red arrow points from the text "Maintenant notre composant <Form> a des props" to the `formTitle` prop in the right pane.

Les 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 composant 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 class App extends Component {
8
9
10  _addArticle = (article) => {
11    console.log('article vue par le parent App', article);
12  }
13
14  render() {
15    return (
16      <div>
17        <h2>Liste de course en construction</h2>
18        < Form formTitle="Ajouter un article" _addArticle={this._addArticle} />
19        < ItemList />
20      </div>
21    );
22  }
23 }
24
25 export default App;
```

```
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={this._handleQuantityChange} />
23          <input type="text" placeholder="art"
24            value={this.state.name} onChange={this._handleNameChange} />
25          <button type="submit">ajouter</button>
26        </form>
27      </div>
28    );
29  }
30 }
31
32 export default Form;
```

ASTUCE AJOUT DE PROPRIETE A UN OBJET A LA VOLER:

On peut ajouter une propriété à un objet que l'on reçoit en argument dans une fonction ou méthode.

On reçoit en argument l'objet {state} de la class Form en argument de la méthode `_addArticle()`.

Dans la méthode on lui ajoute une propriété `id` basé sur la longueur du tableau.

```
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 }
```

```
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 }
```

La propriété `id` a bien été ajoutée

```
Line 24: NO duplicate props allowed
▼ {name: "pain", quantity: 0} ⓘ
  id: 1
  name: "pain"
  quantity: 0
  ► __proto__: Object
```