Formation à React

Pierre-Louis Guhur

September 8, 2019

INRIA Paris/Grenoble

Rappels

- Le JSX est compilé en JavaScript
- React utilise des composants
- React développe des applications mono-pages
- Les composants sont affichés dans le DOM de la page HTML

```
1  class App extends React.Component {
2    render() {
3      return (
4  <div className="container h-100 d-flex">
5      <div className="jumbotron my-auto">
6        <h1 className="display-3">Votre mission :</h1>
7        <p className="lead">
8          <ul className="list-group">
9            <li className="list-group-item disabled" aria-disabled="true">
                 ↪ Comprendre le cours</li>
10           <li className="list-group-item">Developper une application en ES5,
                 ↪ ES6, JSX</li>
11           <li className="list-group-item">Faire la documentation de votre projet
                 ↪ sur README.md</li>
12           <li className="list-group-item">Mettre le code sur GitHub (branche
                 ↪ cours1)</li>
13           <li className="list-group-item">Mettre du CSS</li>
14         </ul>
15       </p>
16     </div>
17 </div>
18     )
19   }
20 }
21 var mount = document.querySelector('#app');
22 ReactDOM.render(<App />, mount);
```

Don't Repeat Yourself

Faisons un composant li

```
1  class Li extends React.Component {
2
3      render () {
4        const activity = this.props.activity;
5        const disabled = this.props.disabled;
6
7        if (disabled) {
8          return (
9              <li className="list-group-item disabled"
10                 aria-disabled="true">
11               { activity }
12             </li>
13         );
14       } else {
15           return (
16               <li className="list-group-item">
17                 { activity }
18               </li>
19           );
20       }
21     }
22 }
```

Réutilisons le plusieurs fois

```
1  class App extends React.Component {
2    render() {
3      return (
4  <div className="container h-100 d-flex">
5      <div className="jumbotron my-auto">
6        <h1 className="display-3">Votre mission :</h1>
7        <p className="lead">
8          <ul className="list-group">
9            <Li activity="Comprendre le cours"
10               disabled="true" />
11           <Li activity="Developper une application en ES5, ES6, JSX" />
12           <Li activity="Faire la documentation de votre projet sur README.md"
                 ↪ />
13           <Li activity="Mettre le code sur GitHub (branche cours1)" />
14           <Li activity="Mettre du CSS" />
15         </ul>
16       </p>
17     </div>
18 </div>
19     )
20   }
21 }
```

Avec une boucle, c'est mieux

```
1  class App extends React.Component {
2    render() {
3      const {activities} = this.props;
4      // equivalent : const activities = this.props.activities;
5      return (
6  <div className="container h-100 d-flex">
7      <div className="jumbotron my-auto">
8        <h1 className="display-3">Votre mission :</h1>
9        <p className="lead">
10         <ul className="list-group">
11         { activities.map( (activity) => {
12             return (
13                 <Li activity={activity.activity}
14                 disabled={activity.disabled} />
15             )}
16          )}
17         </ul>
18       </p>
19     </div>
20 </div>
21     )
22   }
23 }
```

Transférer un tableau

```
1   const activities = [{
2           activity: "Comprendre le cours",
3           disabled: true,
4       }, {
5           activity: "Developper une application en ES5, ES6, JSX",
6           disabled: false,
7       }, {
8           activity: "Faire la documentation de votre projet sur README.md",
9           disabled: false,
10      }, {
11          activity: "Mettre le code sur GitHub (branche cours1)",
12          disabled: false,
13      }, {
14          activity: "Mettre du CSS",
15          disabled: false,
16      }
17  ]
18  var mount = document.querySelector('#app');
19  ReactDOM.render(<App activities={ activities } />, mount);
```

Changer l'état

Un formulaire d'ajout

```
1   class App extends React.Component {
2     constructor(props) {
3           super(props);
4           this.state = {activities: []};
5           this.addLi = this.addLi.bind(this);
6           this.inputRef = React.createRef();
7     }
8
9       addLi() {
10            let activities = this.state.activities;
11            console.log(this.inputRef.current);
12            activities.push({
13                  activity: this.inputRef.current.value,
14                  disabled: false
15            })
16            this.setState({activities: activities})
17      }
```

```
1     render() {
2       const {activities} = this.state;
3       return (
4   <div className="container h-100 d-flex">
5       <div className="jumbotron my-auto">
6         <div className="card">
7           <div className="card-header">
8             TODO liste :
9           </div>
10          <ul className="list-group list-group-flush">
11          <li className="list-group-item">
12              <div className="input-group">
13                <input type="text" ref={this.inputRef} className="form-control"
                      ↪ placeholder={ this.props.placeholder } style={{borderColor: '
                      ↪ white'}} />
14                <button type="button" onClick={this.addLi} className="btn
                      ↪ btn-secondary" data-dismiss="modal">+</button>
15              </div>
16          </li>
17            { activities.map( (activity) => {
18                return (
19                    <Li activity={activity.activity}
20                    className={ activity.disabled ? 'disabled': ''} />
21                )}
22             )}
23          </ul>
```

Pour récapituler

- props
- state
- setState force render
- binding
- Gestion des événements

Cycle de vie

Les cycles couramment utilisés

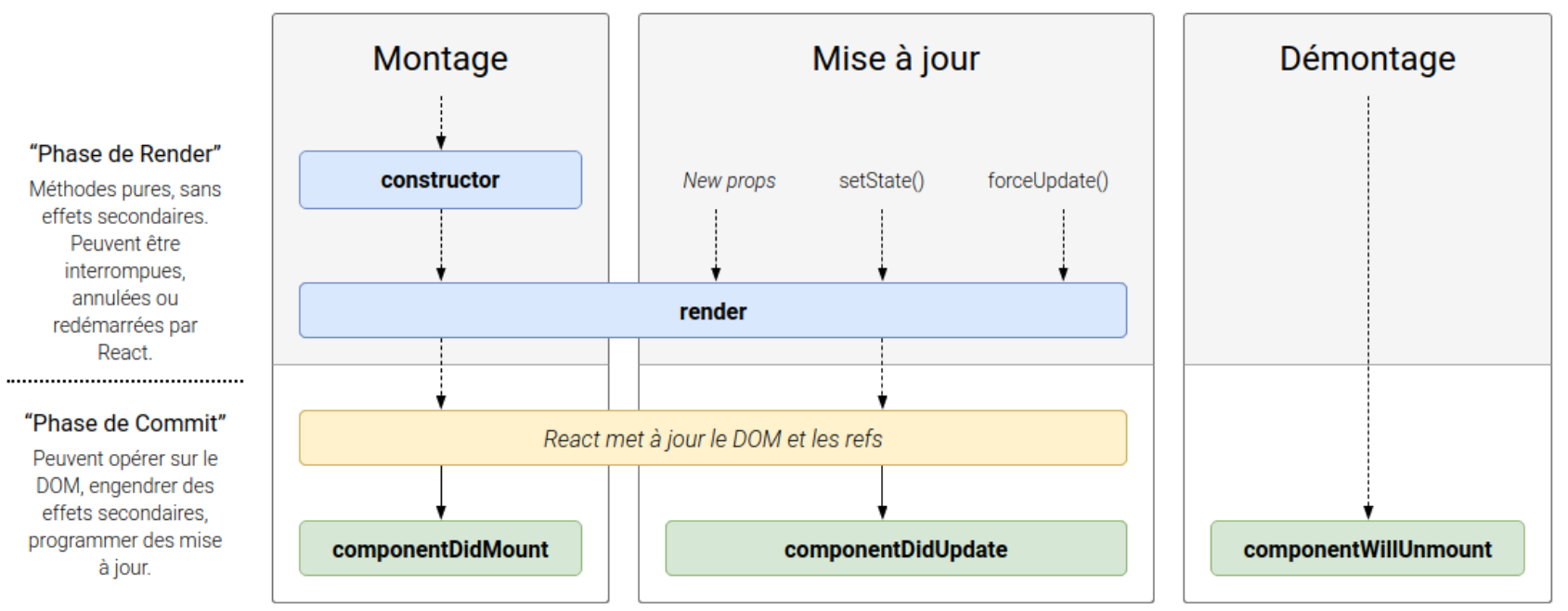
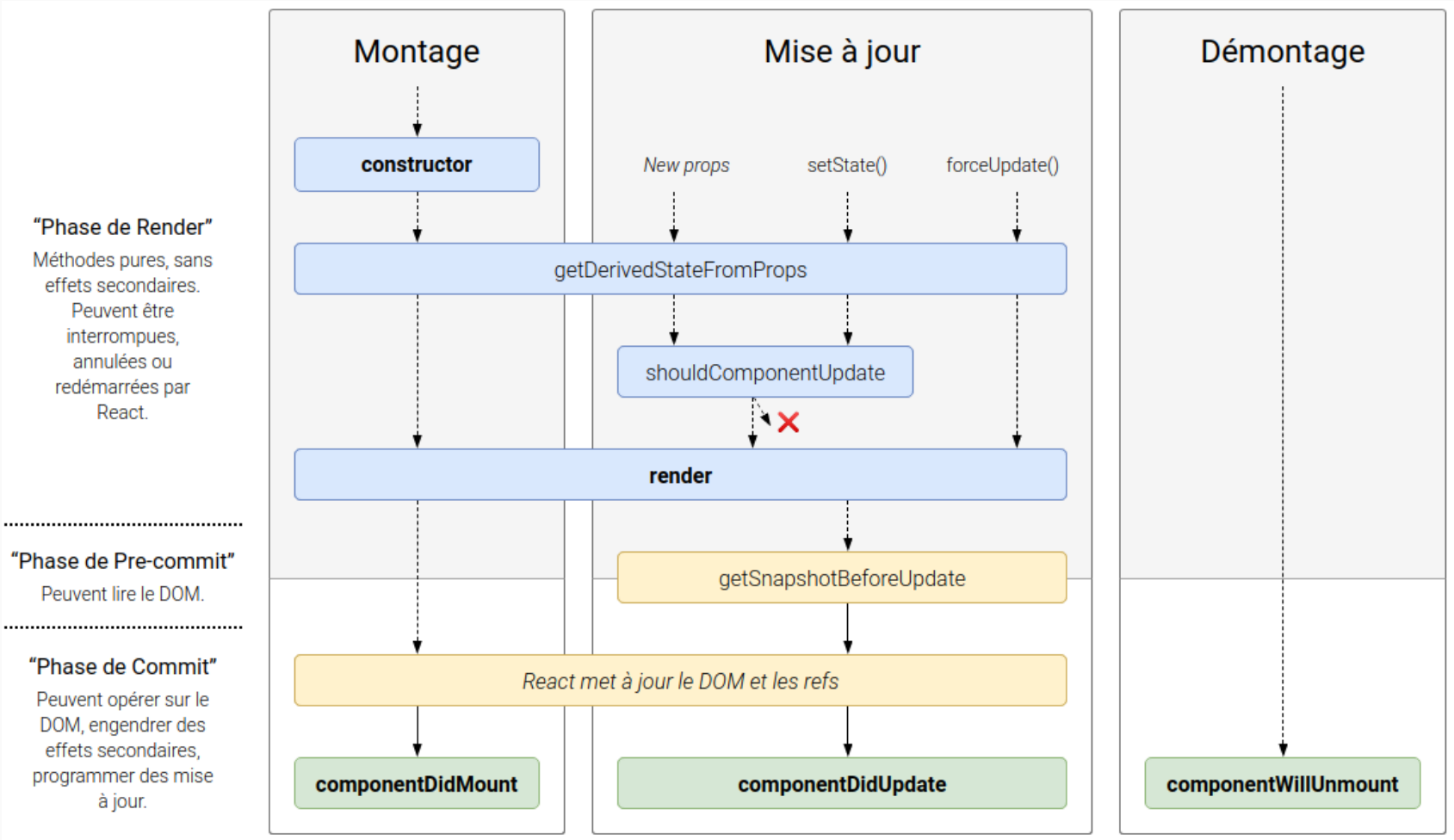


Figure 1: Image issue de projects.wojtekmaj.pl/react-lifecycle-methods-diagram/

Application : le compte à rebours

```
1  class Clock extends React.Component {
2      constructor(props) {
3          super(props);
4          this.state = {date: new Date()};
5      }
6      componentDidMount() {
7          this.timerID = setInterval(() => this.tick(), 1000);
8      }
9      componentWillUnmount() {
10         clearInterval(this.timerID);
11     }
12     tick() {
13         this.setState({date: new Date()});
14     }
15     render() {
16         return (
17             <button className="btn btn-primary">
18               {this.state.date.toLocaleTimeString()}
19             </button>
20         );
21     }
22 }
```

Pour être exhaustif



Pratique

- Utilisation de state et de props
- Imbriquer au moins 3 components
- Utilisation de componentDidMount