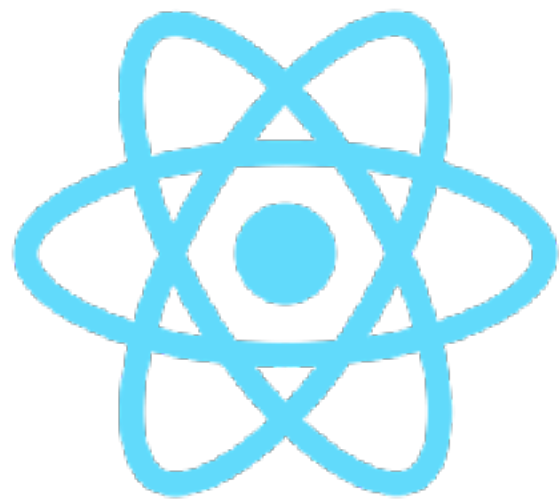


Javascript / React

Composants / Découpage



Correction

TicTacToe

État initial

```
class App extends Component {  
  state = {  
    grid:[  
      [null, null, null],  
      [null, null, null],  
      [null, null, null],  
    ],  
    currentPlayer: 'X',  
    winner: null,  
  };  
  // ...  
}
```

Affichage de la grille

```
<table>
  <tbody>
    {this.state.grid.map((row, i) => (
      <tr>
        {row.map((cell, j) => (
          <td onClick={() => this.onCellClick(i, j)}>
            {cell}
          </td>
        ))}
      </tr>
    ))}
  </tbody>
</table>
```

Modification de l'état

```
onCellClick(row, column) {  
  let newGrid = this.state.grid;  
  newGrid[row][column] = this.state.currentPlayer;  
  
  this.setState({  
    grid: newGrid,  
    currentPlayer: this.state.currentPlayer === 'X' ? 'O' : 'X',  
  });  
}
```

Types de composants

Composant fonctionnel

```
const SubmitButton = ({onClick}) => (  
  <button type="submit" onClick={onClick}>  
    <Icon name="submit" />  
    Valider  
  </button>  
);
```

Composant fonctionnel

- « Simple » fonction
- Retourne un « single-root element »
- Récupère ses props en 1er argument
- N'a pas d'état (state)

Composant objet

```
class SubmitButton extends Component {  
  render() {  
    return (  
      <button type="submit"  
        onClick={this.props.onClick}>  
        <Icon name="submit" />  
        Valider  
      </button>  
    );  
  }  
}
```

Composant objet

- Classe
- Peut posséder des méthodes
- A ses props comme propriétés
- A un état (state)

Ancienne notation

```
const SubmitButton = React.createClass({  
  render: () => {  
    // ...  
  }  
});
```

Cycle de vie

Cycle de vie : Montage

```
constructor() {}  
componentWillMount() {}  
render() {}  
componentDidMount() {}
```

Cycle de vie :

Mise à jour

```
componentWillReceiveProps() {}  
shouldComponentUpdate() {}  
componentWillUpdate() {}  
render() {}  
componentDidUpdate() {}
```

HTTP

Requête http

```
async componentWillMount() {  
  const response = await fetch('https://api.reddit.com/r/reactjs');  
  const json      = await response.json();  
  
  this.setState({threads: json.data.children});  
}
```