

Le State

Le **'state'** est une variable que l'on retrouve **uniquement** dans les classes react.

On doit **toujours** déclarer et initialiser le 'state' dans le **constructeur** de la classe.

Le **'state'** ne peut être modifier uniquement que dans une méthode avec la fonction **this.setState = ({la propriété du state qui change})**

A chaque fois que la fonction setState() est appelé la fonction render est appelé également pour mettre à jour les éléments qui ont eu des modifications de leur objet state{}

Note importante: Il faut bien penser à veiller à conserver le this soit lors d'appel de méthode avec la méthode bind,call ou apply, sinon il faire une méthode flecher pour fixer le this a l'objet courant (c'est a dire a la classe).

On a fixé le this à la l'instance de la classe en déclarant la méthode onPressButton() avec une fonction fléchée.

Avec la fonction setState() on peut redéfinir le state le l'objet courant (qui est l'instance de la classe Header).

```
JS index.js JS App.js JS style.js JS color.js
1 import React from 'react';
2 import { View, Text } from 'react-native';
3 import { Button } from 'react-native-elements';
4 import Icon from 'react-native-vector-icons/FontAwesome';
5
6
7 import Header from './components/header/'
8
9 export default class App extends React.Component {
10
11   constructor(props){
12     super(props);
13     this.state = {myText:'Un texte par default'};
14   }
15
16   onPressButton={() =>{
17     this.setState({myText:'Changement du state'});
18     alert("Click !");
19   }}
20
21   render() {
22     return (
23       <View>
24         <Header content="Liste de tache" />
25         <Button
26           icon={
27             <Icon
28               name='arrow-right'
29               size={3}
30               color='white'
31             />
32           }
33           title='BUTTON '
34           onPress = {this.onPressButton}
35         />
36         <Text>{this.state.myText}</Text>
37       </View>
38     );
39   }
40 }
41
42 }
```