



TEST

React est composé de 2 bibliothèques JS à insérer dans la page HTML :

- **React correspond à React lui-même. Il permet de créer des composants d'affichage réutilisable ;**
- **ReactDOM est une extension permettant de visualiser, dans une page HTML, les composants créés avec React. D'autres extensions sont disponibles, par exemple pour afficher ces composants dans une application iPhone ou Android.**
- **On s'aperçoit de la distinction effectuée par React entre la création de composants (indépendante du support) et le média d'affichage.**

on affiche des objets ou éléments à l'aide de la méthode ReactDOM.render en précisant à quel endroit on souhaite l'afficher d'où le `document.getElementById("app")`.

EXEMPLE

Cherche:

Pomme

Banane

Orange

Mangue

Pêche

Papaye

EXEMPLE

```
handleSearch = (event) => {
  const x = event.target.value;
  console.log(x);

  const filtreTexte = (arr, requete) => {
    return arr.filter(el => el.toLowerCase().indexOf(requete.toLowerCase()) !== -1)
  }
  console.log(filtreTexte(this.state.list, x));

  this.setState({fruits: filtreTexte(this.state.list, x)});
}

handleInsert = () => { ...
}

render(){
  return (
    <div className="app">
      <div>
        Recherche: <input name="qr" type="text" onChange={this.handleSearch} />
      </div>
    </div>
  )
}
```

```
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      list: [
        "Pomme",
        "Banane",
        "Orange",
        "Mangue",
        "Pêche",
        "Papaie"
      ],
      fruits : [ ]
    };
    this.state.fruits = this.state.list;
    this.handleSearch = this.handleSearch.bind(this);
    this.handleInsert = this.handleInsert.bind(this);
  }
}
```

EXAMPLE

```
function App() {  
  const list = [  
    "Pomme",  
    "Banane",  
    "Orange",  
    "Mangue",  
    "Pêche",  
    "Papaie"  
  ];  
  const [filterList, setFilterList] = useState(list);  
  const handleSearch = (event) => {  
    if (event.target.value === " ") {  
      setFilterList(list);  
      return;  
    }  
    const filteredValues = list.filter(  
      (item) =>  
        item.toLowerCase().indexOf(event.target.value.toLowerCase()) !== -1  
    );  
    setFilterList(filteredValues);  
  };  
};
```