



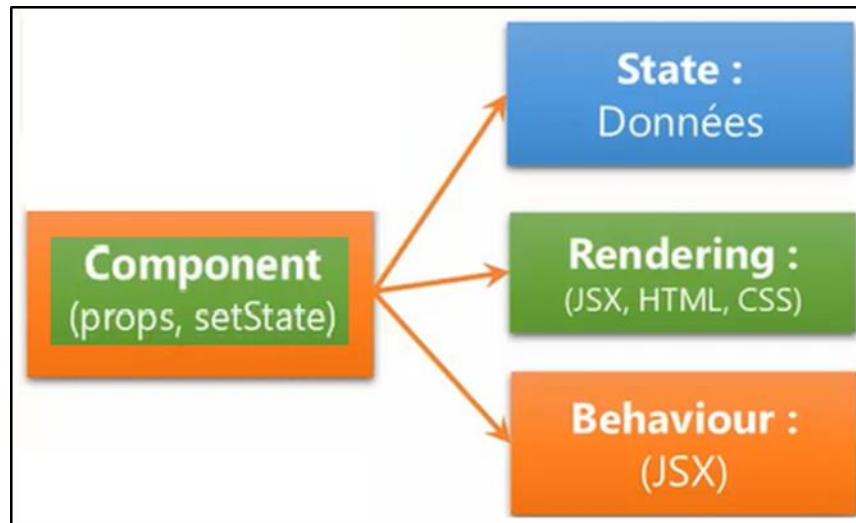
# Les composants React

# C'est quoi un React Component ?

- Les composants servent le même objectif que les fonctions JavaScript, mais fonctionnent de manière isolée et renvoient du HTML via une fonction de rendu.
- Ils peuvent être définis sous forme d'une fonction ou d'une classe.
- Ils sont indépendants et réutilisables.
- Tous les composants, petits ou grands, sont réutilisables, même pour différents projets.
- Un composant React peut être représenté par :
  - ▣ Une **fonction** JavaScript : **Statless Component**
  - ▣ Une **classe** héritant de la classe React.Component : **Statful Component**

# Structure d'un Composant React

- Un composant React est défini par :
  - Son état (State et Props)
  - Son Rendu (JSX, Html, CSS)
  - Son comportement (JSX)



# Stateless vs. Stateful

□ Composant purement fonctionnel (stateless)

```
function App() {  
  return (  
    <h2>Bienvenue à cette formation React</h2>  
  );  
}  
export default App;
```

□ Composant de classe (stateful)

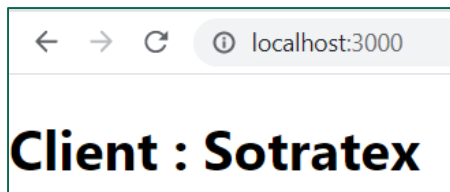
```
class App extends React.Component {  
  render(){  
    return (  
      <h2>Bienvenue à cette formation React</h2>  
    );  
  }  
}  
export default App;
```

# Composant fonctionnel

- Si on écrit un composant React qui ne nécessite pas d'état et qu'on souhaite créer une interface utilisateur réutilisable, on peut l'écrire en tant que composant fonctionnel sans état.

## Exemple :

```
export default function App() {  
  return <h1>Client : Sotratex</h1>;  
}
```



# Composant à base de classe 1 / 2

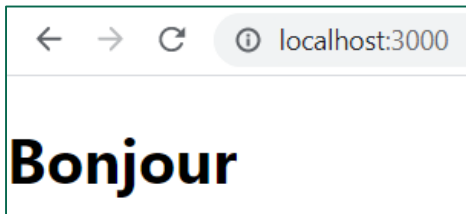
- ❑ On peut utiliser une classe ES6 pour définir un composant.
- ❑ Le nom du composant doit commencer par une lettre majuscule.
- ❑ Le composant doit inclure l'instruction **extend React.Component**
- ❑ Cette instruction crée un héritage à React.Component et donne au composant l'accès aux fonctions de React.Component.
- ❑ Le composant nécessite également une méthode **render()**
- ❑ Cette méthode retourne du HTML.

# Composant à base de classe 2/2

Exemple :

```
import React from 'react';

export default class App extends React.Component {
  render() {
    return (
      <h1>Bonjour</h1>
    );
  }
}
```



# Éléments d'un composant React 1 / 3

- La méthode `render()` ne retourne qu'un seul élément, dans cet exemple, on veut afficher plusieurs composants HTML.

```
class App extends React.Component {  
  render() {  
    return (  
      <h1>Bonjour</h1>  
      <ul>Client1</ul>  
    );  
  }  
}
```



- On voit que ça génère une erreur, cet exemple permet de retourner deux éléments HTML (un élément `h1` et un élément `ul`) ce qui n'est pas possible puisque la fonction `render()` ne retourne qu'un seul élément.



# Éléments d'un composant React 2/3

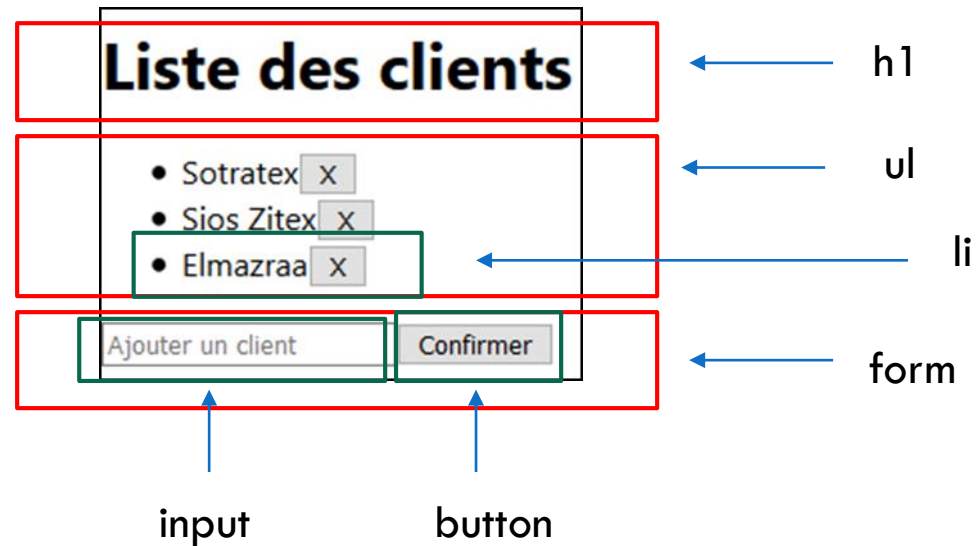
**Solution :**

**Placer les éléments dans une seule balise (<> </>) :**

```
class App extends React.Component {  
  render() {  
    return (  
      <div className="App">  
        <h1>Liste des clients</h1>  
        <ul> <li>Sotratex<button>X</button></li>  
          <li>Sios Zitex<button>X</button></li>  
          <li>Elmazraa<button>X</button></li> </ul>  
        <form> <input type="text" placeholder="Ajouter un client"/>  
          <button>Confirmer</button>  
        </form>  
      </div>  
    );  
  }  
}  
export default App
```

# Éléments d'un composant React 3/3

- Un composant App est un ensemble d'éléments React (h1, ul, li, etc.).



# Appel de composants 1 / 4

- React consiste à réutiliser du code, et il peut être judicieux d'insérer certains des composants dans des fichiers séparés.
- Les composants peuvent donc faire référence à d'autres composants dans leur sortie.
- Ça nous permet d'utiliser la même abstraction de composants pour n'importe quel niveau de détail.
- Pour ce faire, on va créer deux nouveaux fichiers avec une extension de fichier .js appelés respectivement First et Second.
- Notez que les fichiers doivent avoir l'instruction **export default**.

# Appel de composants 2/4

## □ Composant App.js fonctionnel

```
import First from './First';
import Second from './Second';
export default function App() {
  return (
    <>
      <First />
      <Second />
    </>
  )
}
```

## □ Composant App.js de type classe :

```
import React from 'react';
import First from './First';
import Second from './Second';

class App extends React.Component {
  render() {
    return ( <div>
              <First />
              <Second />
            </div>
    );
  }
}

export default App;
```

# Appel de composants 3/4

## □ Composant *First.js* fonctionnel

```
const First = () => {  
  return (  
    <>  
    <h1>Contenu du composant First</h1>  
  
    </>  
  );  
}  
export default First;
```

## □ Composant *First.js* de type classe

```
import React from 'react';  
class First extends React.Component {  
  render() {  
    return (  
      <>  
      <h1>Contenu du composant First</h1>  
      </>  
    );  
  }  
}  
export default First;
```

# Appel de composants 4/4

## □ Composant Second.js fonctionnel

```
const Second = () =>{  
  return (  
    <>  
    <h1>Contenu du composant Second</h1>  
    </>  
  );  
}  
export default Second;
```

## □ Composant Second.js de type classe

```
import React from 'react';  
class Second extends React.Component {  
  render() {  
    return (  
      <>  
      <h1>Contenu du composant Second</h1>  
      </>  
    );  
  }  
}  
export default Second;
```

