

React JS

Jordan Diaz

Tech talk 02/03/21

Définition

React est une bibliothèque JavaScript pour la construction d'interfaces utilisateur (UI). Très utilisé pour le Frontend développement.

React was developed by Facebook to meet the needs of its products, and successfully presented and open-sourced to the tech audience in 2013

```
npx create-react-app my-app  
cd my-app  
npm start
```

npx
npm
npm

Définition

React est une bibliothèque JavaScript, pas un framework

```
const element = <h1>Bonjour, monde !</h1>;
```

Cette drôle de syntaxe n'est ni une chaîne de caractères ni du HTML.

Ça s'appelle du JSX, et c'est une extension syntaxique de JavaScript.

Pourquoi JSX?



Au lieu de séparer artificiellement les technologies en mettant le balisage et la logique dans des fichiers séparés, React sépare les préoccupations via des unités faiblement couplées appelées « composants », qui contiennent les deux.

Avec JSX: CreateNewElement, innerText, innerHTML, appendChild sont finies. HTML + JS = JSX.

Example:

```
const name = 'Clarisse Agbegenou';  
const element = <h1>Bonjour, {name}</h1>;  
  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

Vous pouvez utiliser n'importe quelle expression JavaScript valide dans des accolades en JSX. Par exemple, `2 + 2`, `user.firstName`, ou `formatName(user)` sont toutes des expressions JavaScript valides.

Example:

```
function formatName(user) {  
  return user.firstName + ' ' + user.lastName;  
}  
  
const user = {  
  firstName: 'Kylia',  
  lastName: 'Mbappé'  
};  
  
const element = (  
  <h1>  
    Bonjour, {formatName(user)} !  
  </h1>  
)  
);  
  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

On découple le JSX en plusieurs lignes pour une meilleure lisibilité. Par la même occasion, nous recommandons également de le mettre entre parenthèses afin d'éviter les pièges d'insertion de point-virgule automatique, même si cette pratique n'est pas obligatoire.

Example de Dynamique:

```
function tick() {  
  const element = (  
    <div>  
      <h1>Bonjour, monde !</h1>  
      <h2>Il est {new Date().toLocaleTimeString()}.</h2>  
    </div>  
  );  
  ReactDOM.render(element, document.getElementById('root'));  
}  
  
setInterval(tick, 1000);
```

Les éléments React sont immuables. Une fois votre élément créé, vous ne pouvez plus modifier ses enfants ou ses attributs. Un élément est comme une image d'un film à un instant T : il représente l'interface utilisateur à un point précis dans le temps.

Avec nos connaissances actuelles, la seule façon de mettre à jour l'interface utilisateur est de créer un nouvel élément et de le passer à ReactDOM.render().

<https://codepen.io/pen?&editors=0010>

Hello, world!

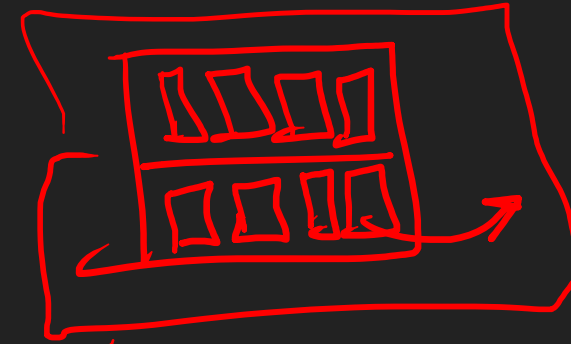
It is 12:26:46 PM.

Console Sources Network Timeline

```
<div id="root">  
  <div data-reactroot=>  
    <h1>Hello, world!</h1>  
    <h2>  
      <!-- react-text: 4 -->  
      "It is "  
      <!-- /react-text -->  
      <!-- react-text: 5 -->  
      "12:26:46 PM"  
      <!-- /react-text -->  
      <!-- react-text: 6 -->  
      ", "  
      <!-- /react-text -->  
    </h2>  
  </div>  
</div>
```



Composants et props



Les composants vous permettent de découper l'interface utilisateur en éléments indépendants et réutilisables, vous permettant ainsi de considérer chaque élément de manière isolée

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Bonjour, {this.props.name}</h1>;  
  }  
}
```

Class component

2013

```
function Welcome(props) {  
  return <h1>Bonjour. {props.name}</h1>;  
}
```

Function component + props

2017

Comment utiliser le composant?

```
<Welcome name="Sara" />;
```

<Welcome>

- Lorsque React rencontre un élément représentant un composant défini par l'utilisateur, il transmet les attributs JSX et les enfants à ce composant sous la forme d'un objet unique. Nous appelons cet objet « props ».

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Bonjour, {this.props.name}</h1>;  
  }  
}
```

```
function Welcome(props) {  
  return <h1>Bonjour, {props.name}</h1>;  
}
```

Composition de composants

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

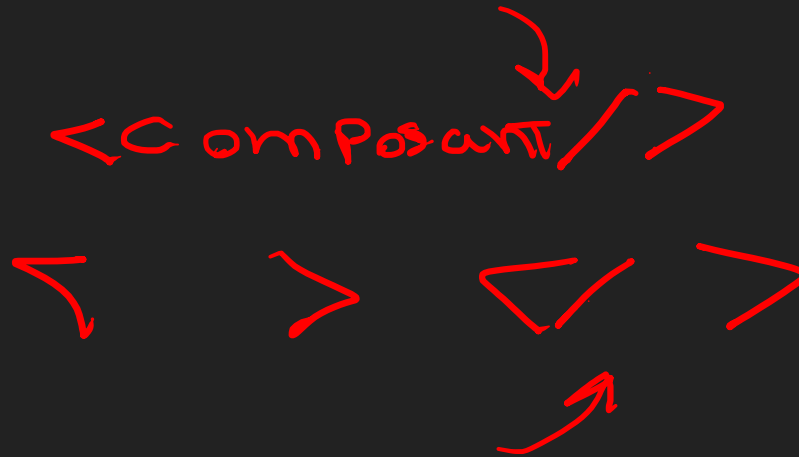
```
function App() {  
  return (  
    <div>  
      <Welcome name="Jordan" />  
      <Welcome name="David" />  
      <Welcome name="Adlane" />  
    </div>  
  );  
}
```

```
ReactDOM.render(<App />,  
  document.getElementById('root'));
```

Hello, Jordan

Hello, David

Hello, Adlane





Exemple des composants dynamiques

- Générer des composants et voir ses propriétés (props)
- Architecture basique d'une app en React (en test)
- React développer Tools
- Comment React change le DOM en temps réel
- Un aperçu de React Hook.

<http://rapelletoi.herokuapp.com/>

<https://github.com/Jordandiazfr/keeper-note-app>

<https://github.com/facebook/create-react-app>



There are 10 kinds of people, those who understand binary and those who don't.

Merci