

Cours ReactJS

React.js est une bibliothèque JavaScript open-source développée par Facebook. Elle est principalement utilisée pour construire des interfaces utilisateur interactives et réactives pour les applications web. Voici un cours de base sur React.js :

Introduction à React.js :

React.js est basé sur un paradigme de programmation appelé "programmation déclarative", où vous décrivez comment votre interface utilisateur devrait se comporter et React se charge du "comment". React utilise un DOM virtuel pour améliorer les performances en réduisant le nombre de manipulations du DOM réel.

Installation de React.js :

Pour créer un projet React, vous devez d'abord installer Node.js et npm (Node Package Manager). Ensuite, vous pouvez créer un nouveau projet React avec la commande suivante :

```
bash
```

```
npx create-react-app nom-de-votre-projet
```

Structure d'un projet React :

- **src** : Contient les fichiers source de l'application React.
- **public** : Contient les fichiers statiques tels que l'index HTML.
- **node_modules** : Contient les dépendances du projet.
- **package.json** : Fichier de configuration du projet.

Composants React :

Les composants sont la pierre angulaire de React. Un composant est une partie réutilisable de votre interface utilisateur. Il peut être une classe ou une fonction.

Composant de classe :

```
jsx
```

```
import React, { Component } from 'react';

class MonComposant extends Component {
  render() {
    return (
      <div>
        <h1>Mon Composant de Classe</h1>
        <p>Ceci est un exemple de composant de classe.</p>
      </div>
    );
  }
}

export default MonComposant;
```

Composant fonctionnel :

```
jsx

import React from 'react';

const MonComposant = () => {
  return (
    <div>
      <h1>Mon Composant Fonctionnel</h1>
      <p>Ceci est un exemple de composant fonctionnel.</p>
    </div>
  );
}

export default MonComposant;
```

Props et State :

Les **props** (propriétés) sont des données passées à un composant, tandis que le **state** représente l'état interne d'un composant.

```
jsx

import React, { useState } from 'react';

const ExemplePropsEtState = (props) => {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Prop passée : {props.maProp}</p>
      <p>État local : {count}</p>
      <button onClick={() => setCount(count + 1)}>Incrémenter</button>
    </div>
  );
}

export default ExemplePropsEtState;
```

Manipulation du DOM virtuel avec JSX :

React utilise une syntaxe appelée JSX (JavaScript XML) pour décrire la structure de l'interface utilisateur.

```
jsx

import React from 'react';

const MonComposantJSX = () => {
  return (
    <div>
      <h1>Utilisation de JSX</h1>
      <ul>
        <li>Élément 1</li>
        <li>Élément 2</li>
        <li>Élément 3</li>
      </ul>
    </div>
  );
}
```

```
export default MonComposantJSX;
```

Gestion des événements :

jsx

```
import React, { useState } from 'react';

const GestionDesEvenements = () => {
  const [texte, setTexte] = useState('');

  const handleChange = (e) => {
    setTexte(e.target.value);
  }

  return (
    <div>
      <input type="text" value={texte} onChange={handleChange} />
      <p>Texte saisi : {texte}</p>
    </div>
  );
}

export default GestionDesEvenements;
```

Utilisation de composants enfants et composition :

jsx

```
import React from 'react';

const Parent = (props) => {
  return (
    <div>
      <h1>Parent</h1>
      {props.children}
    </div>
  );
}

const Enfant = () => {
  return (
    <p>Ceci est un composant enfant</p>
  );
}

const CompositionDeComposants = () => {
  return (
    <Parent>
      <Enfant />
    </Parent>
  );
}

export default CompositionDeComposants;
```

Utilisation de routes avec React Router :

React Router permet de gérer la navigation dans une application React.

```
bash
```

```
npm install react-router-dom
```

```
jsx
```

```
import React from 'react';
import { BrowserRouter as Router, Route, Link } from 'react-router-dom';

const PageAccueil = () => <h2>Page d'accueil</h2>;
const PageAPropos = () => <h2>À propos</h2>;

const App = () => {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li><Link to="/">Accueil</Link></li>
            <li><Link to="/a-propos">À propos</Link></li>
          </ul>
        </nav>

        <Route path="/" exact component={PageAccueil} />
        <Route path="/a-propos" component={PageAPropos} />
      </div>
    </Router>
  );
}

export default App;
```

Conclusion :

Ceci est une introduction de base à React.js. React offre de nombreuses fonctionnalités avancées telles que les hooks, le contexte, et la gestion d'état plus avancée. Pour en savoir plus, consultez la documentation officielle de React et continuez à pratiquer en construisant des applications.