

Front Avancé

TP-1.3



A M I E N S

Groupe: Romain

TP1.3 Premier composant Réact

Etape 1 Lire la vidéo :

[React02] comment coder son premier composant React.js

- 1) Crée l'app react

```
● debian@debian11:~/Front-Avancer/tp1.3$ npx create-react-app my-server

Creating a new React app in /home/debian/Front-Avancer/tp1.3/my-server.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 10m
```

```
debian@debian11:~/Front-Avancer/tp1.3$ npx create-react-app my-server
Starts the test runner.

npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

cd my-server
npm start

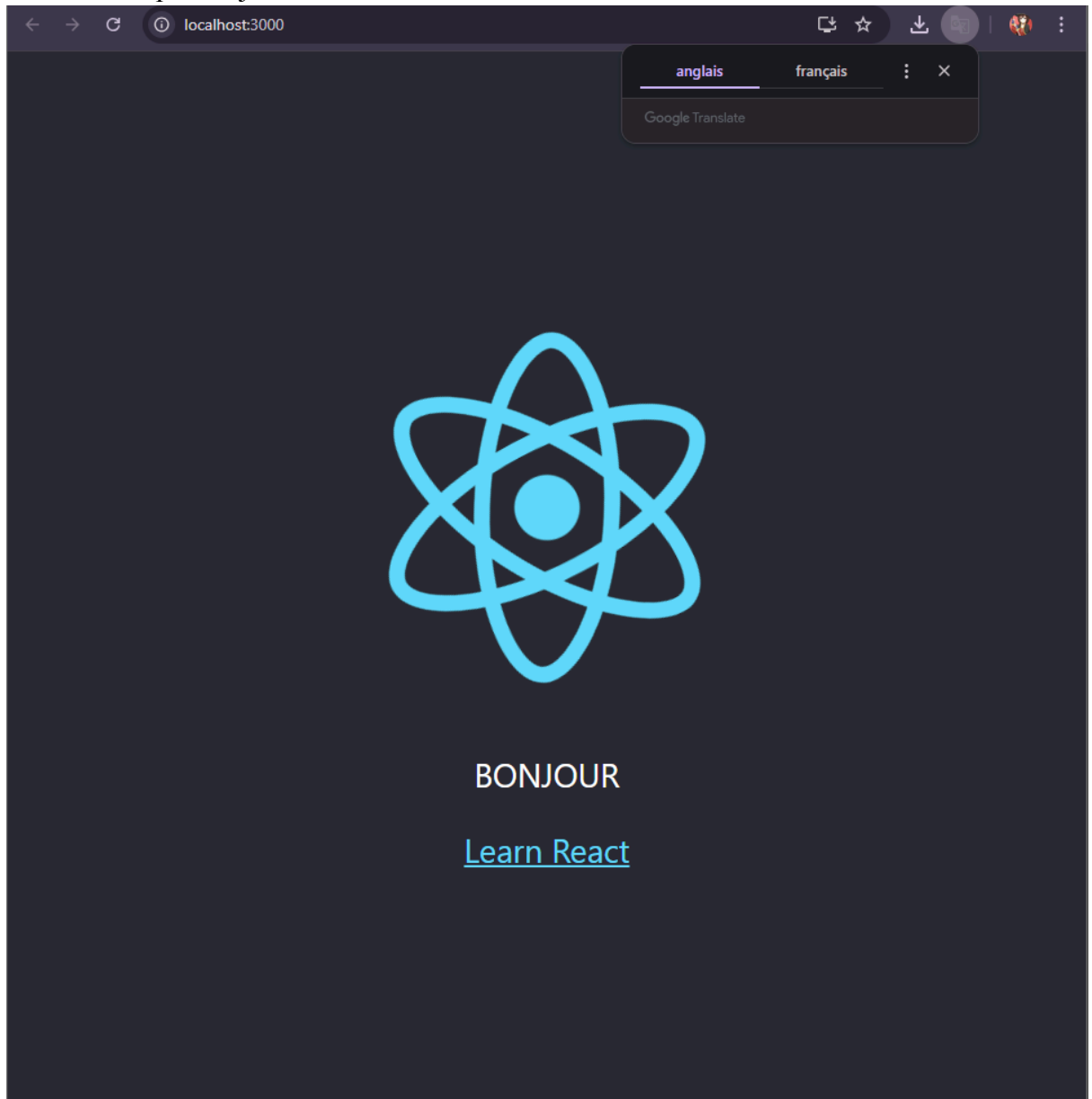
Happy hacking!
```

- 2) lancé l'app react :

```
● debian@debian11:~/Front-Avancer/tp1.3$ cd my-server
○ debian@debian11:~/Front-Avancer/tp1.3/my-server$ npm start

> my-server@0.1.0 start
> react-scripts start
```

3) Modifier le <p> en ajoutant BONJOUR dedans :



```
JS App.js U X
tp1.3 > my-server > src > JS App.js > App
1  port logo from './logo.svg';
2  port './App.css';
3
4  nction App() {
5  return [
6    <div className="App">
7      <header className="App-header">
8        <img src={logo} className="App-logo" alt="logo" />
9        <p>
10         BONJOUR
11       </p>
12       <a
13         className="App-link"
14         href="https://reactjs.org"
15         target="_blank"
16         rel="noopener noreferrer"
17       >
18         Learn React
19       </a>
20     </header>
21   </div>
22 ];
23
24
25 port default App;
26
```

- 4) Recopier ce qui a été présenté au début de la vidéo :

```
JS index.js U JS composant.js U X
tp1.3 > my-server > src > composant > JS composant.js > [⌘] default
1  import React, {useState} from 'react';
2
3  const FunctionalComponent = ({couleur}) => {
4      const [color, setColor] = useState(couleur);
5
6      const changeColor = () => {
7          const newColor = color === 'red' ? 'blue' : 'red';
8          setColor(newColor);
9      };
10
11     return(
12         <div>
13             <div style={{color}}>
14                 Je suis un composant
15             </div>
16             <button onClick={changeColor}>Changer la couleur
17         </div>
18     );
19 };
20
21 export default FunctionalComponent;
```

- 5) Ajouter notre composant dans [index.js](#) et Donner une couleur a notre composant :

```
JS index.js U JS composant.js U
tp1.3 > my-server > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import reportWebVitals from './reportWebVitals';
5  import Composant from './composant/composant.js';
6
7
8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  root.render(
10     <React.StrictMode>
11         {}
12         <Composant couleur={'red'} />
13     </React.StrictMode>
14 );
15
16
17 reportWebVitals();
```

6) Résultat :

Je suis un composant
Changer la couleur

Je suis un composant
Changer la couleur

Etape 2 Faire l'exercice N°28 demandé en fin de vidéo

- 1) Quand on survole le texte :
Le code:

```

JS index.js U • JS composant.js U •
tp1.3 > my-server > src > composant > JS composant.js > [⌘] ComposantSurvole > [⌘] colors
1  import React, { useState } from 'react';
2
3  const ComposantSurvole = ({ couleur = 'red' }) => {
4    const [color, setColor] = useState(couleur);
5
6    const colors = ['red', 'blue', 'green', 'orange', 'purple']
7
8    const changeColor = () => {
9      const currentIndex = colors.indexOf(color);
10     const nextIndex = (currentIndex + 1) % colors.length;
11     setColor(colors[nextIndex]);
12   };
13
14   return (
15     <div>
16       <div
17         style={{ color, cursor: 'pointer' }}
18         onMouseEnter={changeColor}
19       >
20         Je suis un composant
21       </div>
22     </div>
23   );
24 };
25
26 export default ComposantSurvole;
  
```

- 2) Résultat :

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant

3) faire en sorte que la couleur change aléatoirement :

```

JS index.js U • JS composant.js U •
tp1.3 > my-server > src > composant > JS composant.js > [?] ComposantSurvole
3   const ComposantSurvole = ({ couleur = '#000000' }) => {
4
5
6   // Fonction qui génère une couleur aléatoire
7   const getRandomColor = () => {
8     const letters = '0123456789ABCDEF';
9     let newColor = '#';
10    for (let i = 0; i < 6; i++) {
11      newColor += letters[Math.floor(Math.random() * 16)];
12    }
13    return newColor;
14  };
15
16  const changeColor = () => {
17    setColor(getRandomColor());
18  };
19
20  return (
21    <div>
22      <div
23        style={{ color, cursor: 'pointer' }}
24        onMouseEnter={changeColor}
25      >
26        Je suis un composant qui change de couleur aléatoire
27      </div>
28    </div>
29  );
30 };
31
32 export default ComposantSurvole;

```

Code :

Résultat :

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant

Je suis un composant