Atelier 1 : Mise en place de l'environnement du travail – Initiation JSX

Dans cet atelier, on se propose de créer une simple application React et de la faire exécuter sur le navigateur. Egalement, on va se familiariser avec la syntaxe de JSX. En effet, React dispose d'un moyen d'alléger l'écriture du code à travers l'utilisation de JSX (JavaScript eXtension).

Objectifs :

- Exécuter une simple application React via CRA (Create-React-Application).
- Comprendre la syntaxe de JSX.
- Maîtriser nouvelle forme d'écriture des éléments React.
- Savoir écrire un Component React.

Exercices à faire :

Exercice 1 :

1. Installer d'abord Node JS : https://nodejs.org/en/download/
2. Créer un nouveau dossier.
3. Créer une application react
4. Sur la ligne de commande et afin de créer l'application, taper :

npx create-react-app my-react-app

5. Pour pouvoir lancer l'application, taper les deux instructions suivantes :

cd my-react-app

npm start

6. Vérifier le lancement de l'application par défaut sur : http://localhost:3000
7. Ouvrir le projet qui vient d'être créé à partir de l'éditeur.
8. Ouvrir le fichier src/App.js puis supprimer son contenu.
9. Ecrire le code correspondant dans App.js qui génère le résultat suivant dans le navigateur :

Solution :

```
export default function App() {

      return(
        <div>
        <h1>Le Header</h1>
        <h2>Le contenu</h2>
        <p>Le contenu de la page</p>
        </div>

)

}
```

Remarque :

  principale peut être remplacée par <></>

Exercice 2 :

Nous pouvons utiliser nos propres valeurs personnalisées qu'on pourrait afficher en plus du code HTML. Modifier le contenu de l'exercice 1 pour obtenir le résultat suivant, avec 2021 une valeur préalablement initialisée dans une constante appelée value.

Solution :

```
export default function App() {
  const value = "2021";
      return(
        <div>
        <h1>Le Header</h1>
        <h2>Le contenu</h2>
        <p >Le contenu de la page avec valeur = {value}</p>
     </div>

)
}
```

Exercice 3

Nous ne pouvons pas utiliser les instructions if else dans JSX, à la place nous pouvons utiliser des expressions conditionnelles (ternaires). Dans le code initialiser une variable i à une valeur égale à 1. Dans ce cas le navigateur rendra true et on affichera le header. Si nous la modifions en une autre valeur, elle rendra false et le header n'est plus affiché.

Solution :

```
export default function App() {
  let i=1;
      return(
        <div>
            <h1>{i == 1 ? 'Le Header' : ''}</h1>
            <h2>Le contenu</h2>
            <p >Le contenu de la page </p>
         </div>
)

}
```

Exercice 4

React recommande d'utiliser des styles en ligne. Lorsque nous voulons définir des styles en ligne, nous devons utiliser la syntaxe camelCase (exemple font**S**ize à la place font-size). React ajoutera également automatiquement px après la valeur numérique sur des éléments spécifiques. Ajouter une variable myStyle qu'on attribuera à l'élément h1.

Solution :

```
export default function App() {
  let myStyle = {
    fontSize: 100,
    color: '#FF0000'
 }

      return(
        <div>
        <h1 style = {myStyle}>Header</h1>
       </div>
```

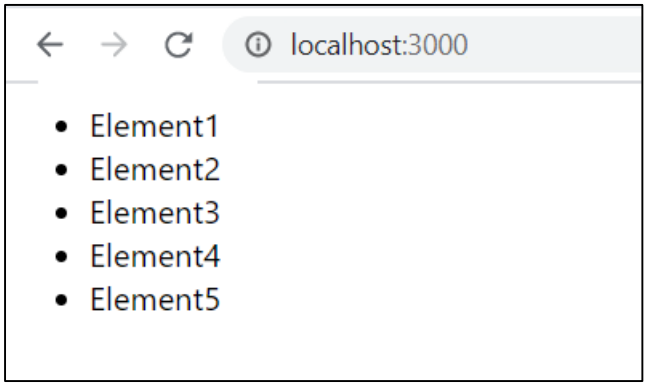
```
)

}
```

Exercice 5

Dans cet exercice, on va créer un tableau elems qui qui contient les éléments["Element1","Element2","Element3","Element4","Element5"]

Ecrire un code qui affiche le contenu de ce tableau en utilisant la fonction map.



Solution :

```
export default function App() {
  var elems = ["Element1","Element2","Element3","Element4","Element5"];
      return(
        <ul>
        {
            elems.map((elem,index)=>{
                return <li key={index}>{elem}</li>;
            })
        }
        </ul>

)

}
```

Exercice 6

Dans cet exercice, on va procéder à faire un affichage défini dans une fonction qu'on va faire appel dans return de App.js.

On voudrait afficher la date système obtenue à partir de la méthode toLocaleDateString() appliquée sur Date(). Ainsi que l'heure système à travers `toLocaleTimeString()`.

Solution :

```
export default function App() {

    const Affiche=()=>{
         return( <div>
        <h1>Bonjour !</h1>
        <h2>Aujourdh'ui : {new Date().toLocaleDateString()}</h2>
        <h2>Il est {new Date().toLocaleTimeString()}</h2>
        </div>
         );
         }

      return(
        <>
        {Affiche()}
        </>

)

}
```