React pour la nuit de l'info

Installation

Les fichiers React sont des fichiers ayant pour extensions .jsx

Ces fichiers sont classés par dossiers avec chaque dossier ayant une spécificité :

Assets : données ou images de l'application, les données sont classées par sous-dossier spécifiques aux données : JSON, icones, images, vidéos

Components : composants de l'application (Header, Boutons, etc...), dans ce dossier nous retrouverons uniquement les composants utilisés plusieurs fois.

Pages : pages de l'applications (Accueil, Contact, etc…), les pages seront organisées par dossiers afin de regrouper le code, le style et les tests d'une page sous un même dossier.

Services : code ou fonctions permettant de récupérer des données ou bien d'accéder à des variables spécifiques, le code de l'api sera par exemple ici.

Theme : thème de l'application contenant la typographie, les couleurs et d'autres constantes de l'application relatives au thème.

Utils : Dossier contenant des fonctions ou du code utile à l'application , comme par exemple le formatage des dates, les fonctions gérants les erreurs, ou encore des fonctions de redirection.

Écrire le squelette d'un composant

Les composants marchent sous forme de fonctions par exemple :

```
function exempleComposant() {
   return (
      <div>
         <h1 >Test</h1>
      </div>
   )
}
export default exempleComposant
```

Avec ce code je peux importer le composant n'importe où dans mon application :

Import exempleComposant from './chemin'

Ou bien

Import { exempleComposant } from './chemin' si le composant n'est pas exporté en default.

Je peux ensuite utiliser mon composant dans le code d'une page :

```
function maPage() {
   return (
      <div>
         <exempleComposant/>
      </div>
   )
}
export default maPage
```

Variables spéciales sous React

Une bonne pratique sous React est de nommer toutes les variables en camelCase

uneVariable

Mettre une variable dans le HTML

Pour utiliser une variable dans notre HTML il suffit juste de mettre notre variable entre crochet pour qu'elle soit interprétée par React. Pour l'afficher par exemple :

{variable}

Cela marche

Passer une variable entre les objets React

Pour passer des variables ou des composant entre les différents objets de React, on utilise les props. On passe les variables en écrivant l'objet puis en spécifiant les paramètres ensuite :

<exempleAvecVariable uneVariable={uneVariable} />

Et dans mon objet je récupère ce paramètre comme ceci :

```
function exempleAvecVariable( { unVariable } ){
        /*suite du code */
}
```

On peut récupérer les variables en utilisant le même nom dans le composant parent et le composant parent.

Variables actualisant la page

Sous React nous avons la possibilité d'actualiser un objet (composant ou page) lorsqu'une variable est mise à jour, pour cela nous utilisons useState de React. Pour déclarer ce genre de variable, nous déclarons comme ceci :

Const [variable, setVariable] = useState(0)

Dans cet exemple nous déclarons une variable, avec une fonction associée pour changer son état, avec une variable par défaut, ici 0

Cette variable ainsi que sa fonction peuvent être passé en paramètre à une autre fonction :

<exempleAvecVariable uneVariable={uneVariable} setVariable={setVariable} />

Abonnement à des évènements ou des variables

Sous React, si je veux m'abonner à des variables ou à des événements je peux utiliser useEffect. Cette fonction permet d'exécuter du code lorsque ce que j'observe change.

```
useEffect(() => {
  // Code à exécuter au montage
  console.log("Le composant est monté !");
}, []); // Le tableau vide signifie que cet effet n'a pas de dépendances, donc il ne s'exécutera lors de l'appel de l'objet React.
```

Cette fonction est très utile pour exécuter une fonction lors du montage de la page. S'il faut charger des données avia une api, nous le ferons avec u useEffect. Pareillement s'il faut effectuer une fonction lors du démontage de l'application.

```
useEffect(() => {
  // Code à exécuter à chaque mise à jour de 'count'
  console.log(`Le compteur est maintenant à : ${count}`);
}, [count]); // On met 'count' dans le tableau de dépendances
```

Autres fichiers

Pour utiliser d'autres fichier sous React nous utilisons import. Par exemple pour importer des fichiers de css ou bien des images, cela fonctionne comme pour les objets React :

```
Import ''./mon.css ''
Import monImage from './assets/monImage'
```