

1) Les avantages de Reactjs sont:

- Il y a l'existence du DOM virtuel qui lui permet d'accroître la performance de l'application.
- Il possède une communauté très active et large, ce qui permet au développeur
- L'existence du composant réutilisable qui permet de faciliter la maintenance et le développement de l'application. Cela permet aussi de gagner du temps.

Les inconvénients de Reactjs sont:

- La mise à jour de React est trop rapide ce qui nécessite un ré-apprentissage pour les développeurs
- Google rencontre un problème d'indexation des sites web créés avec React
- L'apprentissage de la syntaxe JSX et les hooks ne sont pas du tout faciles au début

Les avantages de Vuejs sont:

- Vue.js est souvent considéré comme plus facile à apprendre pour les débutants grâce à une syntaxe plus intuitive et une documentation claire et détaillée.
- Il est facile d'intégrer avec les projets existants.
- Les projets Vue.js sont de petite taille. Ils font environ 18-21 Ko.

Les inconvénients de Vue.js sont:

- Il manque de communauté
- Il manque de plugins, encore moins vastes par rapport à React
- Sa flexibilité est très excessive, ce qui rend les codes moins cohérents d'un projet à l'autre

4) En JavaScript, Promise est un objet qui représente la réussite ou l'échec éventuelle d'une opération asynchrone, permettant de gérer plus facilement les opérations asynchrones en évitant le "callback hell".

Async/await est une syntaxe moderne qui simplifie l'écriture de code asynchrone. `async` marque une fonction pour qu'elle renvoie une Promise, et `await` est utilisé pour attendre la résolution de cette Promise, rendant le code asynchrone plus lisible et ressemblant à du code synchrone.

5)

- `useState`: Ceci permet de déclarer des variables d'état dans les composants fonctionnels.
- `useEffect`: Permet d'exécuter des effets de bord dans un composant fonctionnel, comme les appels API ou la manipulation du DOM. On passe une fonction de rappel qui sera exécutée après le rendu du composant. On peut également spécifier des dépendances pour contrôler quand l'effet doit être réexécuté
- `useContext`: Permet d'accéder au contexte React, ce qui est utile pour partager des données globales (comme les paramètres de thème ou les informations utilisateur) entre composants sans avoir à passer des props à chaque niveau de la hiérarchie.