

Partie 1 : Intérêt de Next.js pour un site éco-responsable

1. Rendu côté serveur (Server-Side Rendering - SSR) : Next.js permet le rendu côté serveur, ce qui réduit la charge de traitement sur les appareils des utilisateurs, diminuant ainsi la consommation d'énergie .
2. Génération de sites statiques (Static Site Generation - SSG) : Cette fonctionnalité permet de pré-rendre les pages au moment de la construction, réduisant les requêtes serveur et la consommation d'énergie lors de l'accès au site .
3. Optimisation automatique des images : Next.js optimise automatiquement les images, réduisant leur taille et améliorant les performances, ce qui se traduit par une consommation de bande passante et d'énergie moindre .
4. Code splitting automatique : Next.js divise automatiquement le code JavaScript, ne chargeant que le nécessaire pour chaque page, ce qui réduit le temps de chargement et la consommation de ressources .
5. Optimisation des polices : Next.js optimise le chargement des polices, réduisant le temps de chargement et la consommation de bande passante.

Partie 2 : Bonnes pratiques mises en œuvre dans le cadre de votre site

1. Utilisation de composants réutilisables : J'ai créé des composants modulaires et réutilisables, réduisant la duplication de code et améliorant la maintenabilité du site.
2. Optimisation des images : J'ai utilisé le composant Image de Next.js pour optimiser automatiquement les images, réduisant leur taille et améliorant les performances de chargement.
3. Utilisation de la génération statique (SSG) : J'ai utilisé la génération statique pour les pages qui ne nécessitent pas de mises à jour fréquentes, réduisant ainsi la charge serveur.
4. Mise en place d'un système de navigation efficace : J'ai utilisé le routage intégré de Next.js pour créer une navigation fluide et rapide entre les pages.