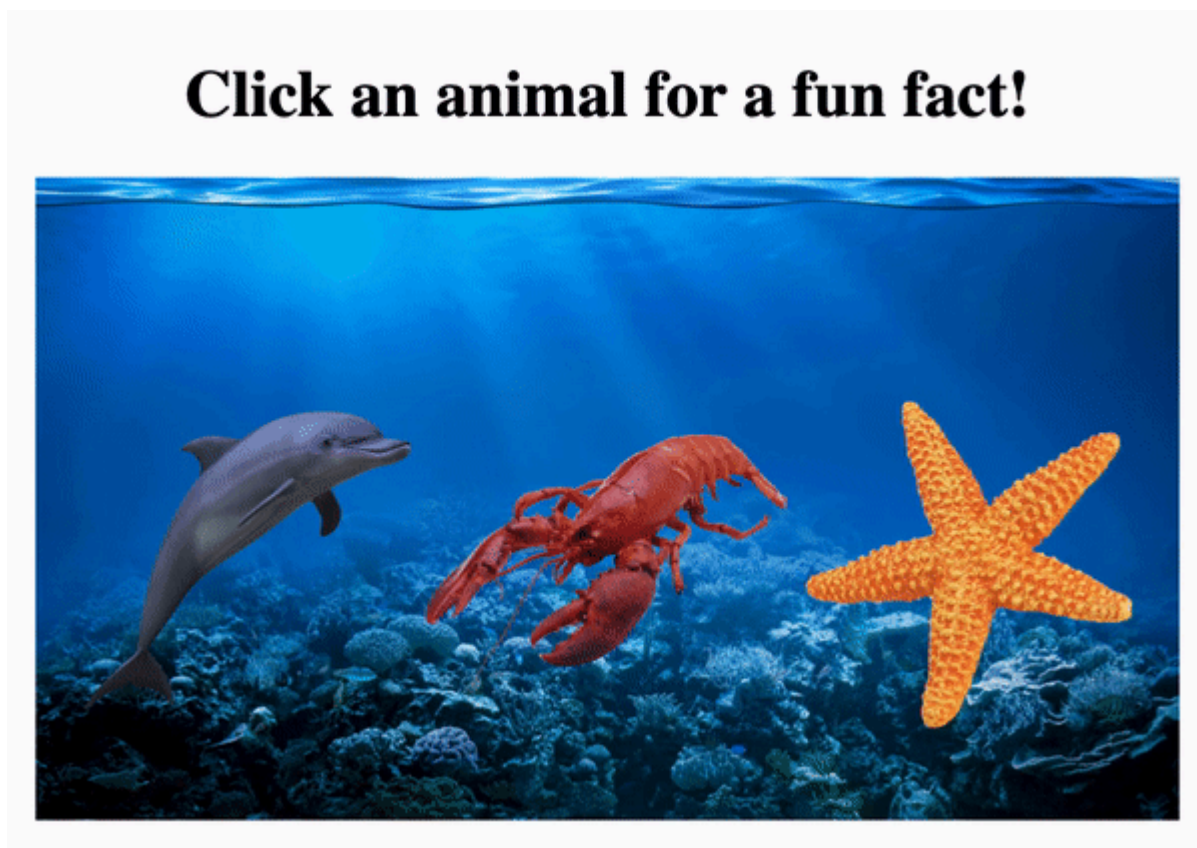


- [Faits amusants sur les animaux](#)
 - [App.js](#)
 - [AnimalFacts.js](#)
 - [Bonus](#)

Faits amusants sur les animaux

Dans ce projet, nous allons créer un programme qui permet aux utilisateurs de cliquer sur un animal sur l'écran afin de faire apparaître une information amusante.



Notre programme affichera une sélection d'animaux à l'écran. Nous serons autorisés à décider si nous voulons inclure un arrière-plan ou non. En cliquant sur un animal, un fait sera sélectionné au hasard parmi une liste d'options potentielles. Le fait sélectionné apparaîtra à l'écran. En continuant à cliquer, nous pourrions voir différents faits.

`App.js`

1.

Dans `app.js`, vous devriez voir des instructions `import`. Il s'agit d'importer le fichier `animalFacts` depuis `animalFacts.js` et de la bibliothèque `React` depuis `react`

2.

Importer l'objet `animals` depuis le fichier (module) `animals`

Dans `AnimalFacts.js`, éditons un JSX pour notre composant `animalFacts` afin d'inclure la variable `background`.

Enveloppez l'élément `<h1>` actuel et notre nouvelle variable `background` à l'intérieur des balises `<div> </div>`. Puisque l'expression va comporter plusieurs lignes, placez-la entre parenthèses.

3.

Utilisez une boucle `for...in` pour parcourir l'objet `animals` que nous importons à la ligne 1. Avant la définition (déclaration) de `animalFacts`, définissez (déclarez) un tableau `images`. Pour chaque animal, ajoutez-en un nouveau tag `` à ce tableau.

En supposant que `animal` soit le nom de la variable d'espace réservé dans votre boucle `for...in`, chaque image doit avoir les attributs suivants :

- `key:{animal}`
- `className:'animal'`
- `alt:{animal}`
- `src:{animals[animal].image}`
- `aria-label:{animal}`
- `role:'button'`

4.

Maintenant que nous avons notre tableau d'images, injectons-le dans l'expression JSX.

Dans le JSX de `animalFacts`, en dessous de `{background}`, créez un élément `<div>`.

Donnez-lui un attribut `className` et définissez-le égal à `'animals'`. Imbibez (inserez) le tableau d'images à l'intérieur de cet élément.

Enfin, Enregistrez, puis lancez le serveur. Nous devrions voir nos animaux !

5.

Maintenant que nos animaux s'affichent à l'écran, nous sommes prêts à ajouter un écouteur d'événement ! Mais d'abord, écrivons une fonction pour gérer cet événement.

1. Créez une fonction `displayFact()` qui prend un paramètre , l'événement `e`. Nous voulons que cette fonction sélectionne un fait amusant aléatoire en fonction de l'animal sélectionné.
2. À l'intérieur de la fonction, utilisez `e.target.alt` pour obtenir le nom de l'animal sur lequel vous cliquez.
3. Générez un index aléatoire et utilisez-le pour accéder à un élément du tableau `.facts` de l'animal.
4. Enregistrez le fait amusant dans une variable.

6.

Nous avons besoin d'un endroit pour afficher nos faits. Créez un élément `<p>` vide dans `animalFacts` et donnez-lui un attribut `id` égal à `'fact'`

7.

Nous devons inclure l'écouteur d'événement avec chacun `` et modifier l'écouteur d'événement afin qu'il affiche le fait dans notre nouvel élément `<p>`.

1. Dans la boucle `for...in`, à l'intérieur de chaque ``, ajoutez un écouteur d'événement `onClick` qui appelle la fonction `displayFact`.
2. À l'intérieur `displayFact()`, utilisez `document.getElementById('fact')` pour saisir l'élément `<p>` où nous ajouterons notre fait. Remplacez `.innerHTML` de l'élément `<p>` par notre fait sélectionné au hasard.

Enregistrez maintenant le code et cliquez sur un animal. Nous devrions voir un fait apparaître à l'écran !

Bonus

8.

Ajoutons une dernière fonctionnalité à notre superbe application !

Créez une constante `showBackground`. Vous pouvez définir sa valeur sur `true` ou `false`.

Si `showBackground` est `true`, `background` devrait apparaître. Si c'est `false`, cela ne devrait pas être le cas. Utilisez l'opérateur `&&` dans `animalFacts` pour implémenter cette fonctionnalité.

Basculez la valeur de `showBackground` entre `true` et `false` et enregistrez le code pour voir si cela fonctionne !

Tâche facultative : En plus de l'opérateur `&&` (AND), nous pouvons utiliser l'opérateur `||` (OR). Étant donné une liste de variables ou d'expressions, `||` renverra la valeur de la première dont le booléen est évalué à `true`.

Considérant que le booléen d'une chaîne vide est `false`, pouvez-vous penser à un moyen `||` de remplacer l'opérateur ternaire dans le titre ?