

## PROJECT

### Animal Fun Facts

In this project, we'll build a program that allows users to click an animal on the screen in order to have a fun fact pop up.

**Click an animal for a fun fact!**



Our program will display a selection of animals on the screen. We'll be allowed to decide if we want to include a background or not. Clicking an animal will cause a fact to be randomly selected from a list of potential options. The selected fact will pop up on the screen. As we keep clicking, we'll be able to see different facts.

Let's get started!

If you get stuck during this project or would like to see an experienced developer work through it, click **"Get Unstuck"** to see a **project walkthrough video**.

#### Tasks

12/12 Complete

[Mark the tasks as complete by checking them off](#)

Add a Title

1.

On line 1 in `app.js` you should see an import statement. This is importing the `animals` object from `animals.js`. Feel free to take a look at it now, but we'll be coming back to it in later tasks.

For now, let's import the React and ReactDOM libraries starting on line 2.

Hint

In JavaScript, we refer to the libraries by the names `react` and `react-dom`.

2.

Add a `title` constant that will hold the value of the title. For now, set its value to an empty string.

In addition, create an `animalFacts` constant to hold the JSX expression that we'll want to be compiled. Set its value to a `<h1>` element that contains our title.

We still shouldn't see anything in the browser yet! We'll have to wait until we write our `ReactDOM.render()` method before anything shows up.

Hint

When nesting JavaScript inside of a JSX expression, the JavaScript must be wrapped in curly braces.

3.

We could fill in the empty string assigned to `title` if we wanted, but we could also leave it blank and let the JSX use a default value instead.

Using the ternary operator, let the `<h1>` heading use 'Click an animal for a fun fact' as the default if `title` is an empty string.

Hint

The following line of code will evaluate to `John Doe`.

```
name === '' ? 'John Doe' : name
```

4.

It's time to write our `ReactDOM.render()` method.

Let's pass in `animalFacts` as the JSX expression that we want to be compiled and rendered. Click on `index.html` to find out what we should pass to `document.getElementById()`.

When finished, click Save. If all goes well, we should see the text 'Click an animal for a fun fact!' appear on the screen!  
Hint

Remember, the first argument of `ReactDOM.render()` is the JSX expression you want to be compiled. The second is the HTML element you want to append it to.

For the argument passed to `document.getElementById()`, look at the `<div>` element between the `<body></body>` tags in `index.html`.

Add a Background

5.

Let's add a background!

Somewhere above where `animalFacts` is defined, create a constant named `background`. Set its value to a `<img />` element.

Now let's give it some attributes!

- Give it a class of 'background'
- Let's use 'ocean' for `alt`.
- Finally, use `'/images/ocean.jpg'` as the value of `src`.

Hint

When adding attributes to an element, they go inside of the opening tag (and they are written in camelCase in JSX). Here's an example:

```
const example = (  
  <div  
    className='banner'  
    tabIndex='2'  
  </div>  
>);
```

6.

Let's reformat the JSX expression stored in `animalFacts` to include the `background` variable.

Wrap the current `<h1>` element and our new `background` variable inside of `<div></div>` tags. Since the expression is going to be multiple lines, wrap it in parentheses.

Click Save. If everything is working as it should, we should see our background image showing up underneath the title!

Hint

Here's how the syntax for including an expression could look like:

```
const example = (  
  <div>  
    <p></p>  
    { /*some JavaScript variable or expression*/ }  
  </div>  
)  
);
```

Add an Array of Images

7.

Use a `for...in` loop to iterate over the `animals` object that we're importing on line 1. Before the `animalFacts` definition, define an `images` array. For each animal, add a new `<img />` to that array.

Assuming `animal` is the iterator variable, each image should have the following attributes:

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

Hint

Here's an example `for...in` loop that adds JSX elements to an array:

```
const todo = ['Make bed', 'Brush teeth', 'Eat breakfast'];  
  
const list = [];  
for (const item in todo) {  
  list.push(<li>{item}</li>)  
};
```

8.

Now that we have our array of images, let's inject it into the JSX expression.

Within the `animalFacts` JSX, underneath `{background}`, create a `<div>`. Give it a `className` attribute and set it equal to `'animals'`. Nest the array of images inside of this element.

Finally, click Save. We should see our animals!

Hint

An array of JSX elements is nested inside of a JSX expression just like any other JavaScript: wrapped in curly braces.

Add an Event Listener

9.

Now that we have our animals displaying on the screen, we're ready to add an event listener! But first, let's write a function to handle this event.

1. Create a function `displayFact()` that takes one argument `e`, the event. We want this function to pick a random fun fact based on the selected animal.
2. Inside of the function, use `e.target.alt` to get the name of the animal being clicked.
3. Generate a random index and use it to access an element in the animal's `.facts` array.
4. Save the fun fact in a variable.

Hint

To generate a random index to help select an element from an array, you can use with the following expression:

```
const optionIndex = Math.floor(Math.random() * selectedAnimal.facts.length);
```

10.

We need a place to display our fact. Create an empty `<p>` element in `animalFacts` and give it an `id` attribute equal to `'fact'`.

Hint

Add a `<p>` element within the `animalFacts` expression, after the `<div>` containing the animal images.

11.

We'll need to include the event listener with each `<img>` and edit the event listener so that it displays the fact in our new `<p>` element.

1. In the `for...in` loop, inside each `<img>`, add an `onClick` event listener that calls `displayFact`.
2. Inside `displayFact()` use `document.getElementById('fact')` to grab the `<p>` element where we'll add our fact. Change the `.innerHTML` of the `<p>` element to our randomly selected fact.

Now save the code and click on an animal. We should see a fact pop up on the screen!

Hint

In JSX elements, event listeners are specified as attributes. Event names are written in camelCase, such as `onClick` for an `onclick` event, or `onMouseOver` for an `onmouseover` event.

Extra Credit

12.

Let's add one last feature to our awesome app!

Create a `showBackground` constant. You can set its value to either `true` or `false`.

If `showBackground` is `true`, `background` should show up. If it's `false`, it should not. Use the `&&` operator in `animalFacts` to implement this feature.

Toggle the value of `showBackground` between `true` and `false` and save the code to see if you got it working!

**BONUS:** In addition to the AND `&&` operator, we can use the OR `||` operator. Given a list of variables or expressions, `||` will return the value of the first one whose boolean evaluates to `true`.

Considering the boolean of an empty string is `false` can you think of a way to use `||` to replace the ternary operator in the header?

Hint

In this example, only 'Fruit' and 'Bread' will show up.

```
const fruit = true;
const bread = true;
```

```

const                eggs                = false;

const                groceryList          = (
  <ul>
    {fruit            &&                <li>Fruit</li>}
    {bread            &&                <li>Bread</li>}
    {eggs             &&                <li>Eggs</li>}
  </ul>
);

```

app.js

```

import { animals } from './animals';
import React from 'react';
import ReactDOM from 'react-dom';

const title = "";

const background = (
  <img
    className='background'
    alt='ocean'
    src='/images/ocean.jpg'
  />
)

const images = [];

function displayFact(e) {
  const selectedAnimal = e.target.alt;
  const animalInfo = animals[selectedAnimal];
  const optionIndex = Math.floor(Math.random() * selectedAnimal.facts.length);

  const funFact = animalInfo.facts[optionIndex];
  document.getElementById('fact').innerHTML = funFact;
}

for (const animal in animals) {
  images.push(
    <img

```

```

        key={animal}
        className='animal'
        alt={animal}
        src={animals[animal].image}
        ariaLabel={animal}
        role='button'
        onClick={displayFact}
      />)
    }

const animalFacts = (
  <div>
    <h1>{title === "" ? "Click an animal for a fun fact" : title}</h1>
    {background}
    <div className='animals'>
      {images}
    </div>
    <p id='fact'></p>
  </div>
)

ReactDOM.render(animalFacts, document.getElementById("root"))
);

```

## animal.js

```

export const animals = {
  dolphin: {
    image: '/images/dolphin.jpg',
    facts: ['Dolphins have been shown to give distinct names to each other!', 'Dolphins are known to display their own culture!', 'Dolphins have two stomachs!']
  },
  lobster: {
    image: '/images/lobster.jpg',
    facts: ['Lobsters taste with their legs!', 'Lobsters chew with their stomachs!', 'Lobsters can live as long as 100 years.']
  }
}

```



```
    },  
    starfish: {  
      image: '/images/starfish.jpg',  
      facts: ['Starfish can have up to 40 arms!', 'Starfish have no brain and no blood!', 'Starfish can regenerate their own arms!']  
    }  
  };
```

## index.html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8" />  
    <link rel="stylesheet" href="./styles.css" />  
  </head>  
  <body>  
    <div id="root"></div>  
    <script src="https://content.codecademy.com/courses/React/react-course-bundle.min.js"></script>  
    <script src="/app.compiled.js"></script>  
  </body>  
</html>
```