# **Getting started with hooks**

## Revising useState hook

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

*Which of the following is true about the useState hook?*

* *The useState hook’s return value in React is the array data structure.*
* *The useState hook allows you to work with state in components.*
* *The useState hook invocation returns a two-member array.*
* *When using the useState hook, you must use the state-updating function to update state.*

## Using the useState hook

*You can use an object with multiple properties as the initial value of the state variable.*

* *True*

## What are side effects?

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated with medium confidence

So, what is a side effect? A side effect is something that makes a function in pure. Did you know that functions can be classified as pure and impure simply put pure functions don't have side effects.

*True or False:*

*A pure function will perform a side effect.*

* *False*

## What is the useEffect hook

Graphical user interface, text, application, email

Description automatically generated

## Using the useEffect hook

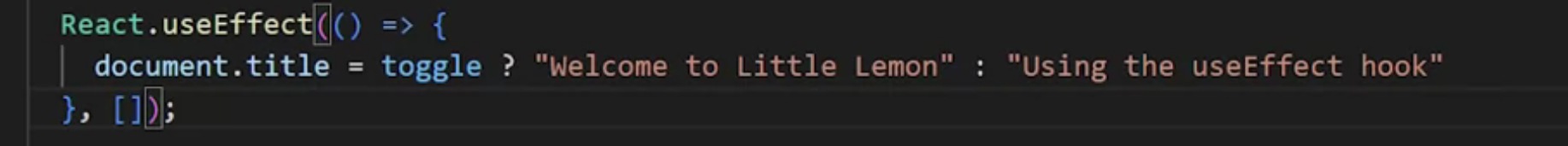
Text

Description automatically generated

Text

Description automatically generated

The above doesn’t have a dependency array so it runs every time there is a side effect.



For now, I'll update my code with an empty dependency array, meaning I'm not tracking the state of any state variables. In other words, regardless of what is happening in my app, I don't want the useEffect hook to be invoked. This means that it'll be invoked only once. After that, no matter what happens in my app, the useEffect hook will no longer be run.

The dependency array is there to watch for changes to a specific variable. Based on that, execute the function that's passed in as the first argument of the useEffect function call. This means that if I want to run the useEffect hook whenever there's an update to the value stored in the toggle variable, I need to add the toggle variable to the dependencies array. After this change, back in the browser, the useEffect hook will be run every time the Toggle message button is clicked.

*How should you add a side effect functionality In React?*

* *Using the useEffect hook*

# **Rules of hooks and fetching data with hooks**

## What are the rules of hooks?

Graphical user interface, text, application, email

Description automatically generated

Only call hooks from a react component function. (You should not call hooks from regular JavaScript functions.), State setting calls (setState) can be called from anywhere, however.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Only call hooks at the top level of a react component.

You are allowed to call multiple state or effect hooks.

Always make multiple hook calls in the same sequence.

*Graphical user interface, text, application

Description automatically generated*

*Is this code using a valid invocation of a hook?*

*if (data !== '') {*

*useEffect( () => {*

*setData('test data')*

*})*

*}*

* *No*

## What you need to know before fetching data

Fetch is used to make a server requests to retrieve some JSON data from it. Fetch API is a set of functionalities that we have at our disposal to use in JavaScript to make such a server request.

A screenshot of a computer

Description automatically generated with medium confidence

*Which of the below statements is an accurate description of JavaScript utilizing the fetch function?*

* *When JavaScript uses the fetch function it is delegating duties to an external API so that it can continue its process. This is known as asynchronous JavaScript (The fetch function utilizes external APIs to perform tasks that would be cumbersome for JavaScript to fulfil alone)*

## Data fetching using hooks

Text

Description automatically generated

There is only one more ingredient that you need to keep in mind when working with React, namely, that fetching data from a third-party API is considered a side-effect.

Being a side-effect, you need to use the useEffect hook to deal with using the Fetch API calls in React.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

## Fetching data – Putting it all together

Text

Description automatically generated

*Which of the statements below best describes how the fetchData function is working?*

* *The fetchData function is initially fetching data from the randomuser.me API, next it retrieves a response from the API in JSON format, and finally updates the state variable with the returned data.*

# **Advanced Hooks**

## What is useReducer and how it differs from useState

You can think of the useReducer as a superpower useState. They useState hook starts with an initial state, but they useReducer hook gets a reducer function in addition to the initial state. This is beneficial because the reducer functions second argument is the action object. This object has multiple type values, and based on each of these types values, you can invoke the dispatch function to perform a specific operation.

Text

Description automatically generated

Text

Description automatically generated

The reducer function takes in the previous state and an action and returns the new state. The action type determines the specific action of the reducer. Actions can have any form. By convention, it's common to pass objects with a type property identifying the action. You should include the minimal necessary information that the reducer needs to compute the next state.

*Which of the following statements is true about the useReducer hook?*

* *The useReducer hook starts with an initial state and reducer function.*

## When to choose useReducer vs useState

Graphical user interface, text, application

Description automatically generated

## useRef to access underlying DOM

Text

Description automatically generated

*What is the data type of the returned value from the useRef hook invocation?*

* *An object*

## Custom hooks

Graphical user interface, text, application, email

Description automatically generated

SGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated