# **Rendering Lists in React**

## Transforming lists in JavaScript

In JavaScript, when you deal with lists of items of any kind, you need to use the array type. JavaScript offers different methods that you can use with arrays to perform various operations. In order to perform a transformation operation, you must use the map method.

Text

Description automatically generated

*Which of the following is true about the map() method in JavaScript.*

*Select all that apply.*

* *The map() method is a transformation operation.*
* *The map() method is useful for handling third party data.*
* *The map() method returns a new array.*

## Render a simple list component

With React, you can transform any list of items into a collection of React components.

Text

Description automatically generated

*When you are working with lists in JSX, you can return a React component. What is the purpose of curly braces in this process?*

* *To access the content of the variable that represents your list item.*

## What are Keys in React?

One important advantage of using React is its ability to automatically optimize updates in your user interfaces or UIs.

Keys are identifier's that help React to determine which items have changed or added or are removed. They also instruct how to treat a specific element when an update occurs and whether its internal state should be preserved or not.

Text

Description automatically generated

To illustrate, adding a key to the last example can make the tree conversion efficient. That's because react now knows that the element with the key cider is the new one and the elements with the keys, beer and wine have just moved. The general rule of thumb with keys is to use a stable identifier that is unique among its siblings. This allows React to reuse as many elements from the list as possible, avoiding unnecessary recreations, especially when their content is exactly the same and the only thing that has changed is their position in the list. The key used most often is a unique ID that comes from your data. Those IDs typically mirror a database ID, which has an ID given to an item in a database that by nature is guaranteed to be unique.

*Which of the following statements about keys in React are true? Select all that apply.*

* *Keys instruct React how to treat a specific element when an update occurs.*
* *Keys instruct React about whether a specific element’s internal state should be preserved or not.*
* *Keys help React determine which items have changed, are added or are removed.*

## Using Keys Within List Components

True or false: It is recommended to use item indexes as keys when selecting keys for rendered list items.

* False ( Although item indexes can be used as keys, using indexes as keys can create problems if the order of your list of items is prone to change and can negatively affect performance. Using unique and stable identifiers, such as item IDs, is recommended instead. )

# **Forms in React**

## What are controlled components?

Graphical user interface, text, application

Description automatically generated

When it comes to React applications, HTML forms work differently to other DOM elements. You may recall that the DOM is a logical tree-like structure representing the HTML document, and it uses nodes to describe the various parts of the document. Traditional HTML forms keep some internal state inside the DOM and have some default behaviour when submitting them. That's normally done via the action attribute, which points to the endpoint that will handle the request.

That's where controlled components come in. Controlled components are a set of components that offer a declarative application programming interface or API to enable full control of the state of form elements at any point in time using React state. Rather than relying on the native state from DOM elements, the React state is made the single source of truth, controlling the displayed value of your form elements at all times. The way you perform this state delegation is via the value prop. Value is a special property, the React added to most of the form elements to determine the input content at any point in time during the render life cycle. In order to create a controlled component, you need to use a combination of local state and the value prop. Initially, you will assign the local stage to the value property.

Graphical user interface, text, application

Description automatically generated

*You are using controlled components in React in order to have more control over the forms in an application you are creating.*

*Which of the following props is used to perform state delegation?*

* *Value (State delegation is performed via the value prop. A combination of local state and the value prop is needed to create a controlled component.)*

## Controlled vs Uncontrolled components

Table

Description automatically generated

## Creating a Form component in React

Text

Description automatically generated

Let's work through the necessary steps to transform this form into a controlled version. First, I need to create some local state for the text input, which I'm going to call name. Secondly, I need to hook up that stage to my text input via two props, the value prop to turn the input into a controlled one and onChange to receive all the changes per keystroke and thus update the state of my input. Last, to control the submission of the form, I have to use the onSubmit prop in the form tag.

Text

Description automatically generated

*Imagine you’d like to better control the submission of a form you’re creating for an app.*

*Which prop is required to prevent the default behaviour of the form submit event in React?*

* *preventDefault*

## Create a Controlled Form Component

Text

Description automatically generated

*What are the two props you need to add when creating a controlled range component?*

* *value and onChange*

# **React Context**

## What you know about Props and State

Props and state are both plain Javascript or Js objects.

While both props and state influence the render output, they are different in one important way props get passed to the component like parameters in a function where state is managed within the component like variables declared within a function.

Another similarity is that props and state changes. Both trigger a render update now.

*Which of the following statements are true about state in React? Select all that apply.*

* *State is a plain JavaScript object that React uses to hold information.*
* *Attributes a component needs to change at some point in time should be part of its state.*
* *State is managed within the component.*

## What is Context, and why is it used?

Context provides an alternative way to pass data through the component tree without having to pass props down manually at every level. It is the right tool when you need to share data that can be considered global for a tree of React components.

Text

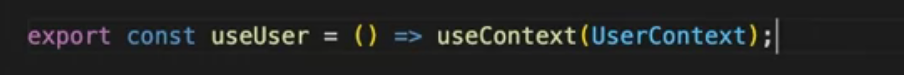
Description automatically generated

The UserContext.Provider component is what allows consuming components to subscribe to context changes. This component accepts a value prop, which is what will be passed to consuming components that are descendants of this provider.

Text

Description automatically generated

Now that the state is defined, I will hook it to the value prop.



Next, it's necessary to provide a way for components to subscribe to the context. For that, I am going to create a custom hook that wraps the use context hook, which is the way to consume a context value.

This external function is created just for convenience, so there's no need to export the user context to external components.

Text

Description automatically generated with medium confidence

The user context is now defined, but the app is still not aware of it. For that, the provider component is needed, so I'll go ahead and wrap the whole app with it. The last step is to consume the user context in the places where the username has to be displayed, which are in the logged in user component and the page component. I'll go ahead and use the custom hook defined earlier.

NOTE: So far, you have learned that when a component consumes some context value and the value of this context changes, that component re-renders.

*Which of the following is true about the Context API? Select all that apply.*

* *Context was introduced by React as a way to resolve the props drilling problem.*
* *Context should be used when you need to share global data.*