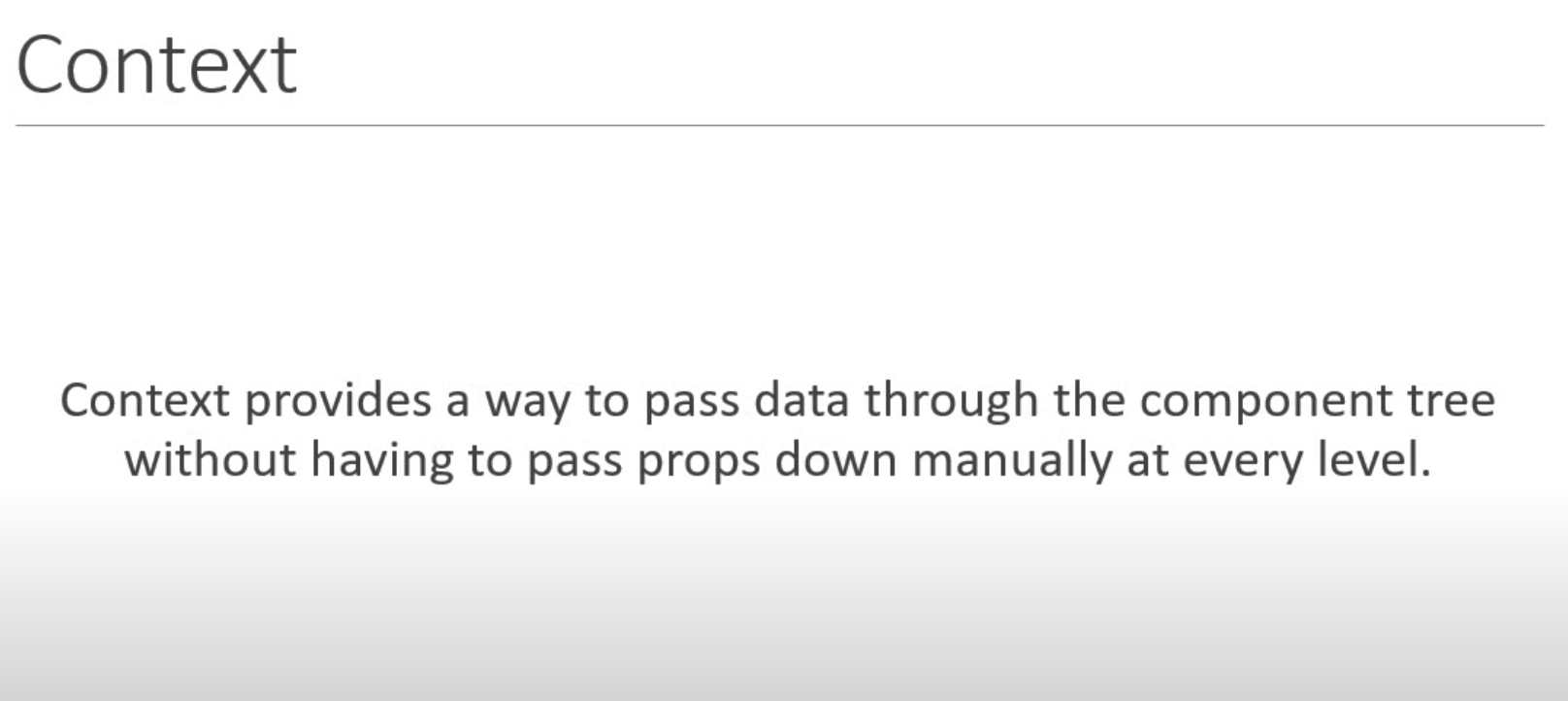
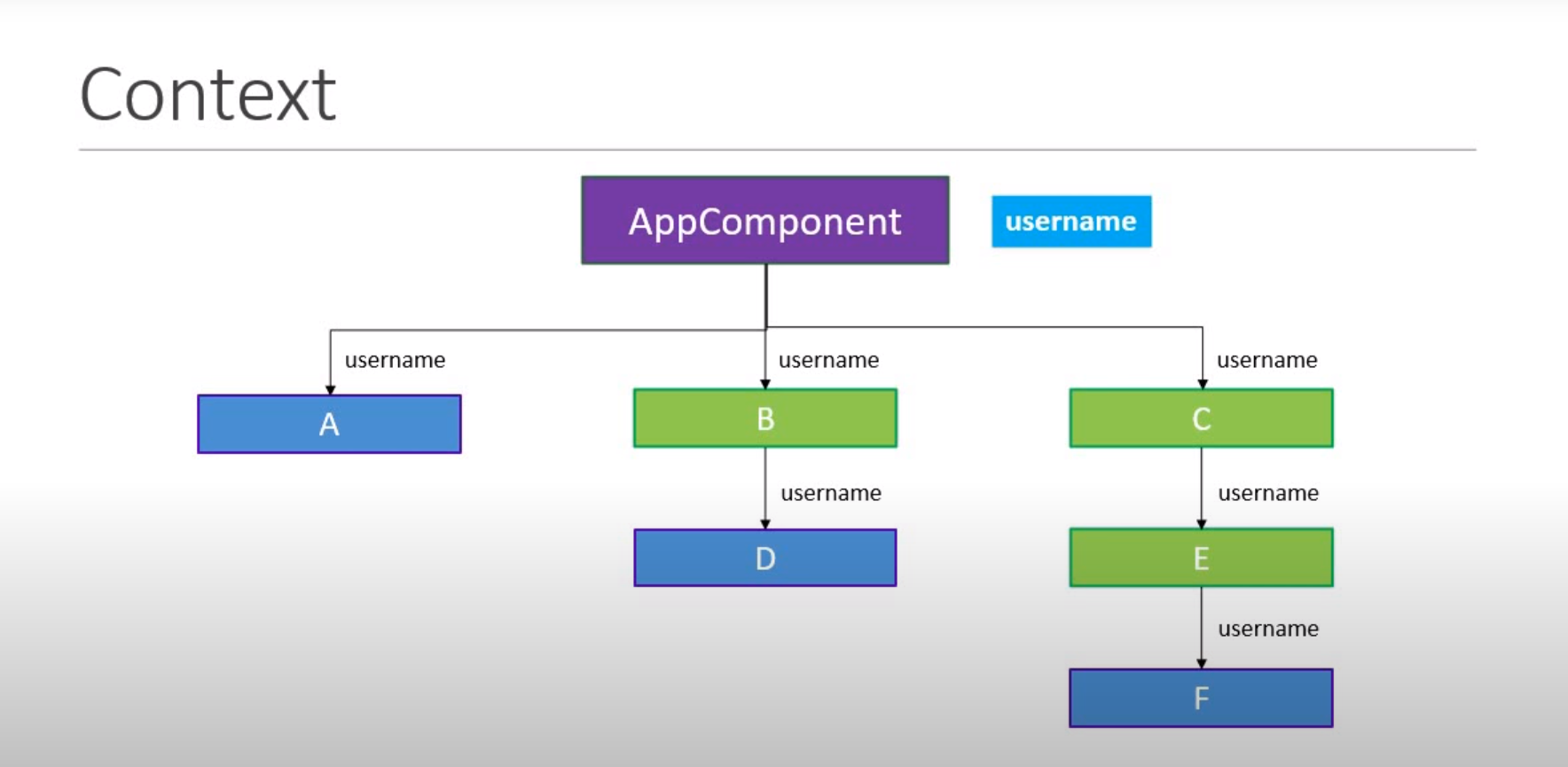
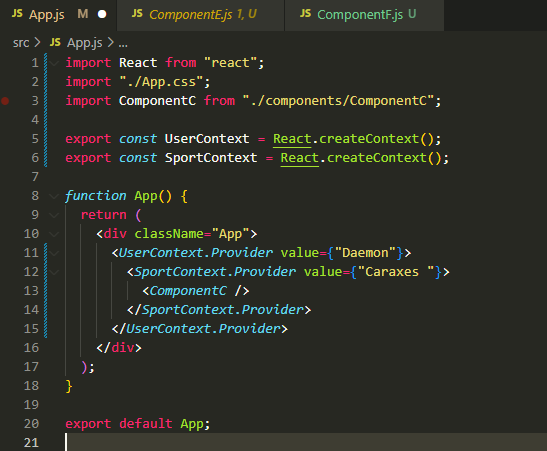
**useContext Hook**



In this example, we want to pass *username* from *AppComponent* to Component *F* without needing to pass it through Component *C* & Component *E.*



*App.js:*



1) We create a context using the *createcontext()* method.

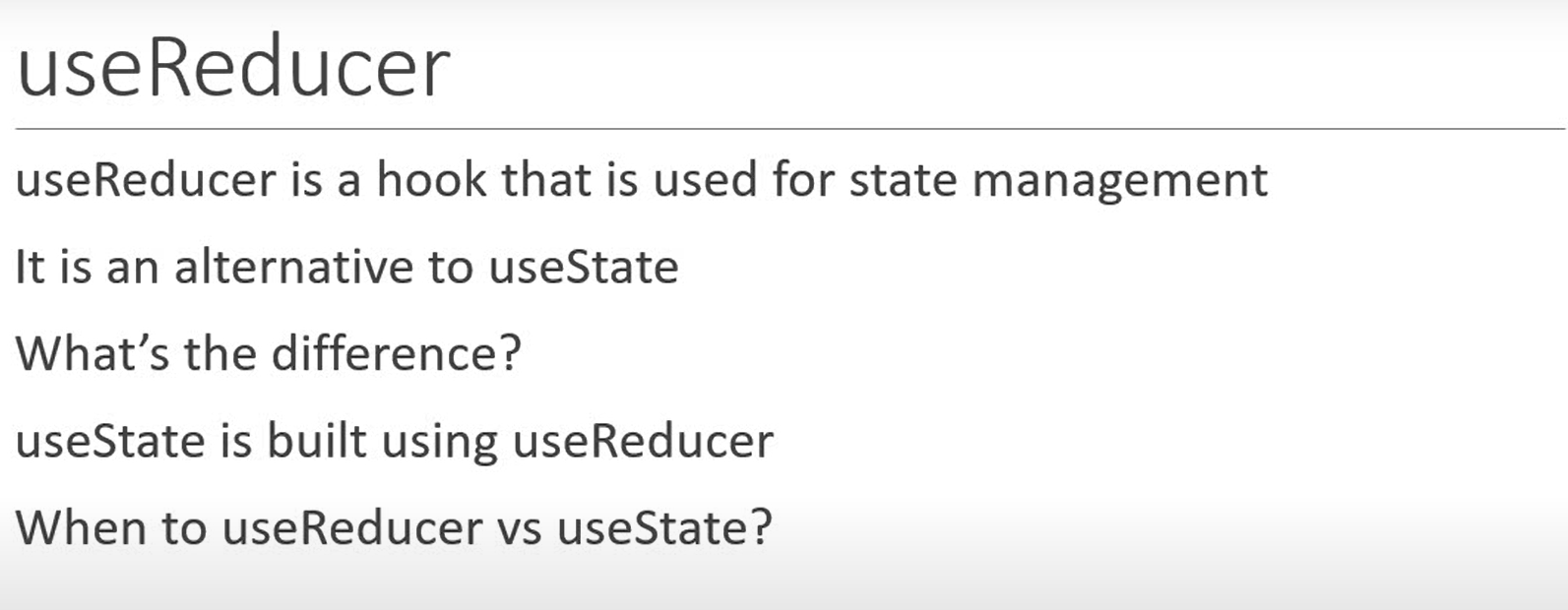
2) We provide the context value at a high level (*<ComponentC/>*) in the component tree.

3) We consume the context value using the render props pattern.

the *useContext()* hook only makes the consumption of the context value simpler.

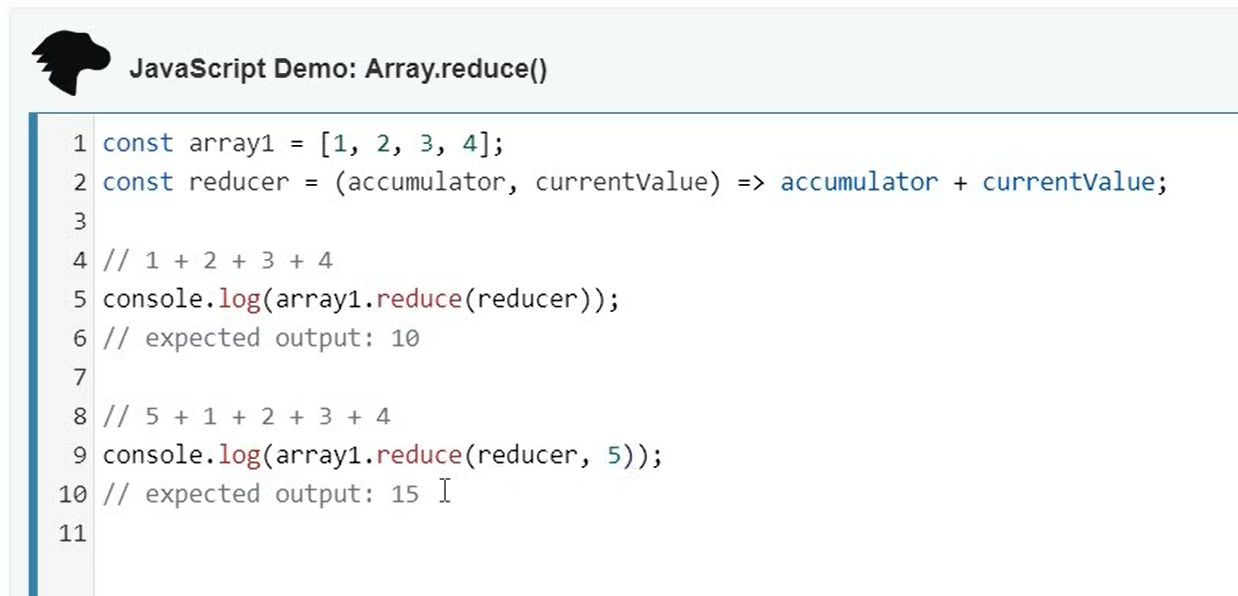


**useReducer Hook**

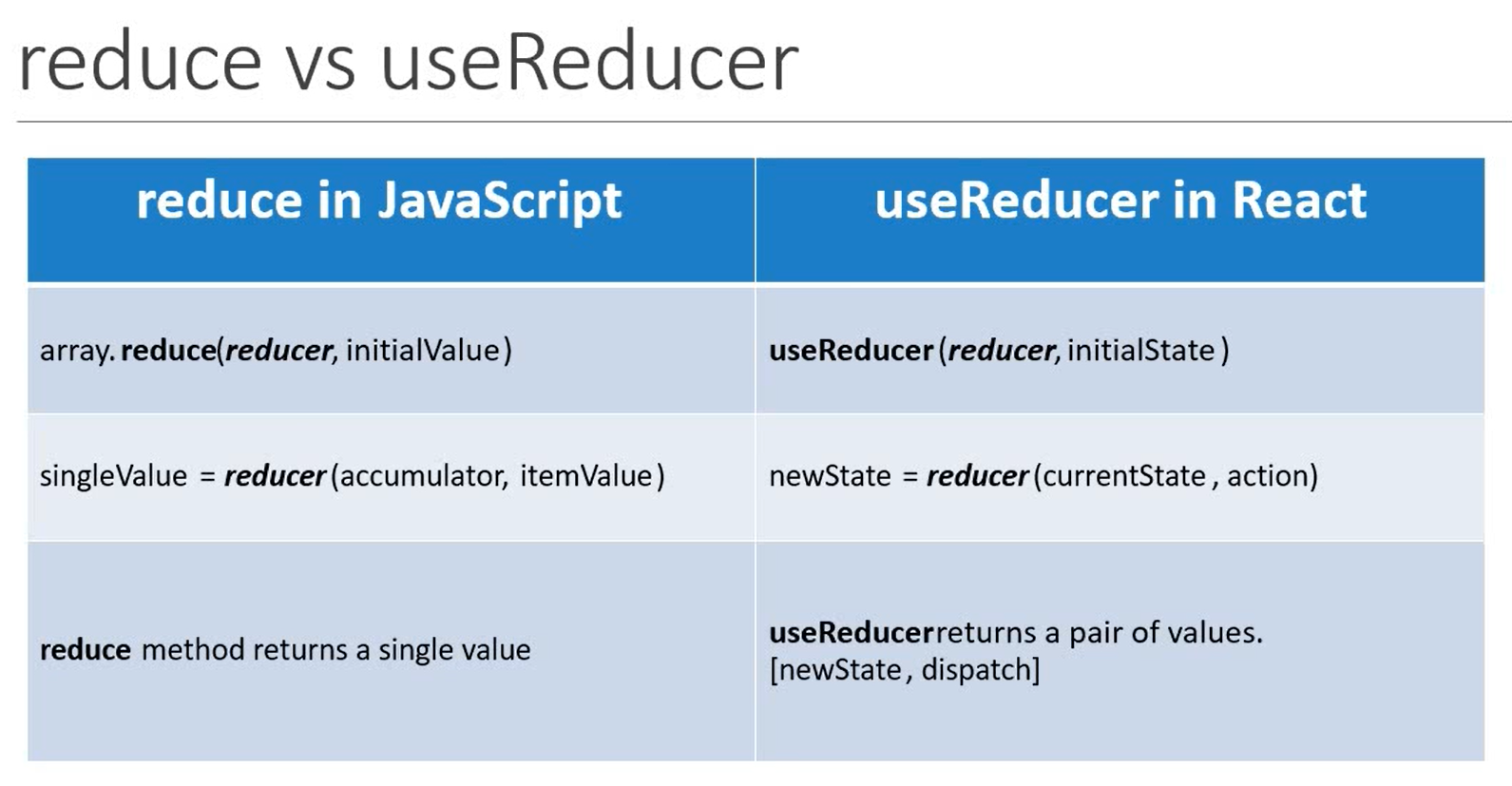


*useReducer* hook is similar to the *reduce()* method in JavaScript.

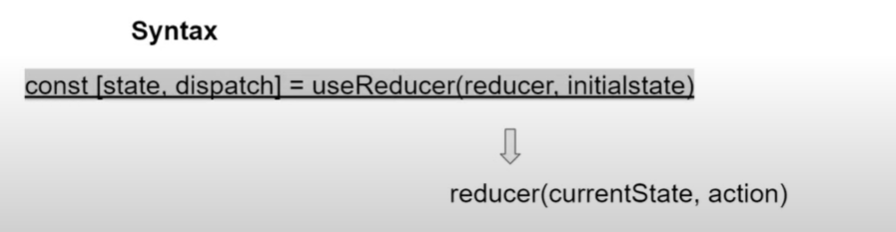
The *reduce()* method has two forms.



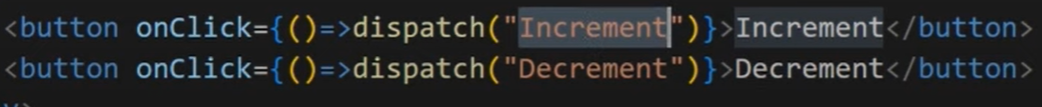
*reduce()* vs *useReducer* hook:



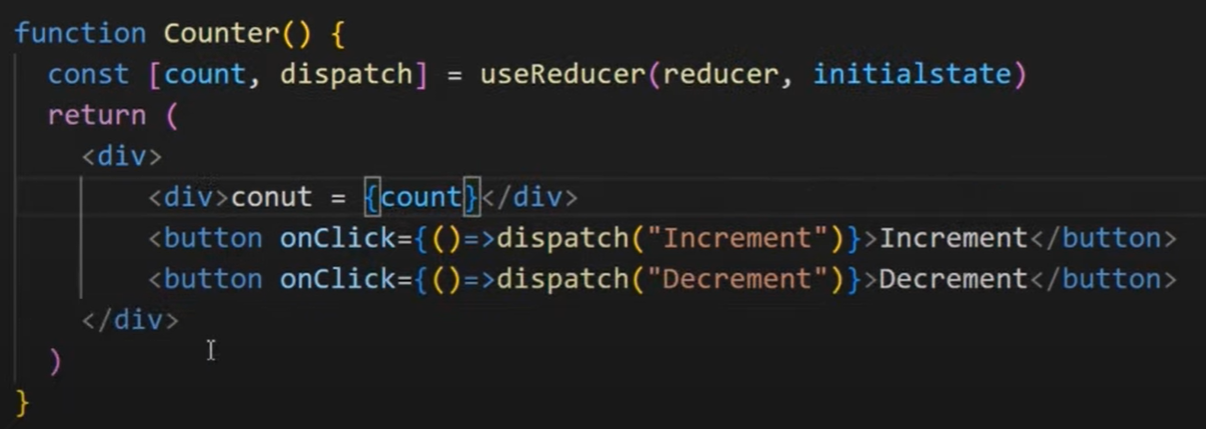
The syntax for *useReducer* is:

**

In *Counter.js* define two buttons:



The *Counter* function will look something like:



We need to declare the initial state variable (*initialstate*) & the *reducer* function.

