

8. Array loops, react core

Sunday, 28 December 2025 6:17 PM

To iterate over array of elements in react:

- Answers - is the array
- Inside div ::
 - <div className = "answers">
 - {quizState.answers.map((answer, index) => (
 - </div>
- Also we need key in js for each element for it to identify so add index or some unique key for each element

...state

- Copy everything from the existing state into a new object as React does not track normal variables. So we need to create new object everytime
 - { name: "John", age: 25 }
 - return {
 ...state,
 name: "Alice"
};
 - State now copies all values and store to new object and replace only the name
 - Its like setter in java

useContext

- way to share state across components without prop drilling, similar to ApplicationContext in java
 - App
 - └ Parent
 - └ Child
 - └ GrandChild
 - If App has data, You must pass it through **every level** — even if they don't need it.
 - const UserContext = createContext();
 <UserContext.Provider value={user}>
 <App />
 </UserContext.Provider>
 - Consume anywhere like this..
 - const user = useContext(UserContext);

Decode inside javascript

- To decode a string inside a java script we can use function "decodeURIComponent"
 - E.g str = "what%20is%20js"
 - decodeURIComponent(str) => what is js