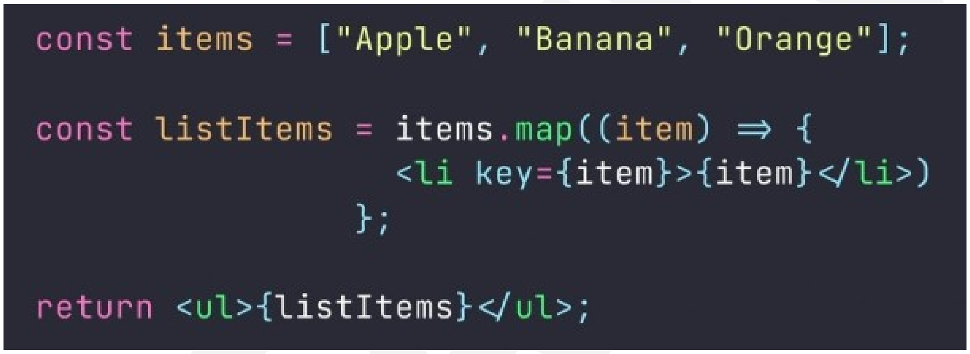
React JS List & Keys

- In React, when rendering a list of elements (such as an **array of items**), it's important to assign a **unique "key"** prop to each element.
- The key prop is a special attribute that helps React efficiently update and reorder the elements in the list.



```
const items = ["Apple", "Banana", "Orange"];

const listItems = items.map((item) => {
                    <li key={item}>{item}</li>)
                  };

return <ul>{listItems}</ul>;
```

- In the example above, the **key={item}** assigns the value of each item in the items array as the **key** for the corresponding list element.

- **What is a list in React?**
  - A list in React refers to an array or collection of items that you want to render as components. For example, a list of to-do items, a list of products, or a list of blog posts.

- **Why do we need keys?**
  - React uses keys to identify individual elements in a list. Keys help React understand the relationship between the previous and current in the list. They enable React to update the list efficiently by identifying added, removed, or moved items accurately.

- **What makes a key unique?**
  - Each key in a list should be unique within the scope of that list. Typically, you use an identifier or a unique property from your data source as the key.

- **How to assign keys to list elements?**
  - When rendering a list of elements in React, you need to assign a key prop to each element. You can do this by using the key attribute within the map function.

- **Why are unique keys important?**
  - Unique keys help React **efficiently update the list** when changes occur. React uses the keys to determine which elements are **added, removed, or reordered**.
  - **Without unique keys**, React may have difficulty distinguishing between similar elements, leading to **unexpected behavior** or degraded performance.