Day 9: Lists & Keys – Rendering Arrays with map()

🔹 What are Lists in React?

In React, a **list** is simply an array of items that we want to display in the UI. Instead of manually writing every `<li>` or `<div>` for each item, we can **render them dynamically** using JavaScript's `map()` method.

For example:

```
const fruits = ["Apple", "Banana", "Mango"];
```

Instead of:

```
<li>Apple</li>
<li>Banana</li>
<li>Mango</li>
```

We use `map()` to generate them dynamically.

🔹 Why do we use Lists?

- Makes rendering **dynamic data** easier (e.g., product lists, user profiles, blog posts).
- Avoids repetitive code.
- Works seamlessly with APIs where data usually comes in arrays.

🔹 Rendering Arrays with `map()`

The `map()` function takes each item in an array and returns a new element for React to render.

Example:

```
function FruitsList() {
  const fruits = ["Apple", "Banana", "Mango", "Orange"];

  return (
    <div>
      <h2>Fruit List</h2>
      <ul>
        {fruits.map((fruit, index) => (
          <li key={index}>{fruit}</li>
        ))}
      </ul>
    </div>
  );
}

export default FruitsList;
```

Here: - `map()` loops through the array. - Each `fruit` is turned into a `<li>` element. - We added a `key` to help React track elements.

🔹 What are Keys in React?

A **key** is a unique identifier assigned to each element in a list. React uses keys to know **which item changed, added, or removed**, which helps with efficient re-rendering.

✅Good Practice for Keys:

- Use a unique property (like `id` from database).
- Avoid using array `index` unless you have no unique values.

Example:

```
function StudentsList() {
  const students = [
    { id: 1, name: "Ali" },
    { id: 2, name: "Sara" },
    { id: 3, name: "Ahmed" }
  ];

  return (
    <div>
      <h2>Students</h2>
      <ul>
        {students.map(student => (
          <li key={student.id}>{student.name}</li>
        ))}
      </ul>
    </div>
  );
}
```

🔹 Why Keys are Important?

- Improves **performance** of rendering.
- Prevents UI bugs when items are reordered.
- Helps React **efficiently update** only the changed elements instead of re-rendering the whole list.

Without keys, React may re-render incorrectly.

🔹 15-20 Minute Practice Exercise

👉**Exercise:** Build a small React component that displays a list of programming languages using `map()`.

Steps:

1. Create a new component called `LanguagesList`.
2. Inside it, define an array: `const languages = ["JavaScript", "Python", "C++", "Java", "Go"];`
3. Use `map()` to render each language inside an `<li>`.
4. Use a **unique key** (either index or create a fake `id`).
5. Style the list with a border and some spacing.

✅Expected Output Example:

```
Languages
- JavaScript
- Python
- C++
- Java
- Go
```

If you finish early, try modifying the component to: - Show both the `id` and the `name`. - Render them in a table format instead of a list.

👎🏿 By completing this exercise, you will: - Understand how to render arrays with `map()`. - Learn why `keys` are important. - Be ready to build **dynamic UI components**.