React Mapping over Data

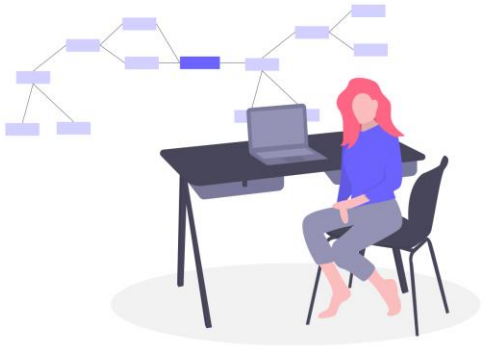
Learning Objectives

- Recap on iterators
- How to map over Data.
- Why each item needs a key.
- Look at the Virtual DOM.

Array Methods/Iterators Recap

.sort()

.map()



What does this have to do with react?

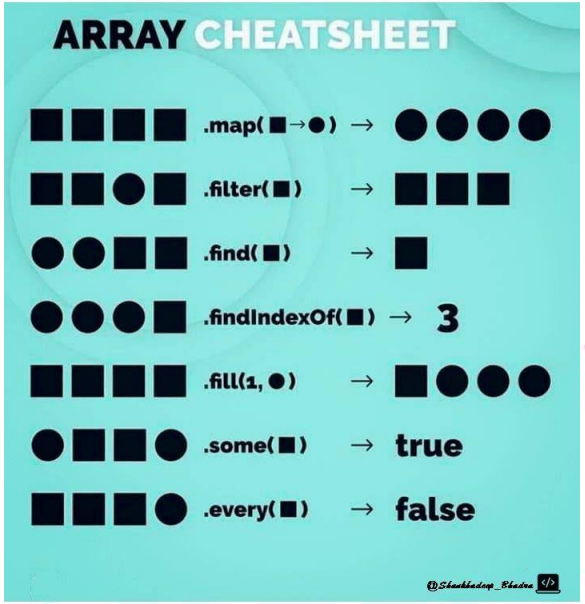
.filter()

.forEach()

.reduce()

Array iterators

Iterator	Returns
For Each	Undefined
Filter	New array
Map	New array
Reduce	Something new
Every	Boolean



Code along - Sandbox

1. Recap on map syntax.
2. Use map to create an array of objects.
3. Create a new component that will use the values from the array of objects.
4. Use map to create an array of these new components.

Challenge

1. There is a data file in the data folder
2. Create a reusable component to display each item from that file.
3. Map over the data file and create a component for each object
4. Have fun and display it in the app

Why do we add Keys?

Reconciliation

Keys help React identify which items have changed (added/removed/re-ordered).

To give a unique identity to every element inside the array, a key is required.

It is always best to have a unique id for each key.

Indexes are not great but better than nothing.

Virtual DOM

Real DOM

- User Interface of application.
- Will re-render to update to represent any change of state in the UI.
- Frequent renders will make a large applications slow.

Virtual DOM

- A virtual representation of the DOM
- The virtual DOM gets updated with any change of state.
- This is then compared with the DOM. Only the items that need to REACT will be re rendered.
- The virtual DOM makes applications faster.

Code along - Mealworm

Map over data to produce both the list component and CardFront components

Steps

- In App.jsx map over the recipes to produce one CardFront for each object of data
- Explain the unique key error in the console
- Create the List component and have it map over an array of ingredients passed in as props

Additional Reading

- Virtual DOM
- MDN docs .map()
- Simplify your code with abstraction