

Mapping in React

Topics covered:

- What are array methods?
 - map()
- How to use a map()?
 - Array
 - Array of object
- Mapping an array without using the map()

1. Array methods:

- There are numerous array methods in JavaScript.
- The .map() array method in React is one of the most helpful.
- By applying a function to each element of the array using the .map() method, a new array is created as a consequence.
- Map() in React can be used to create lists.

Syntax: Map()

```
Array.map((item)=><p>{item}</p>
```

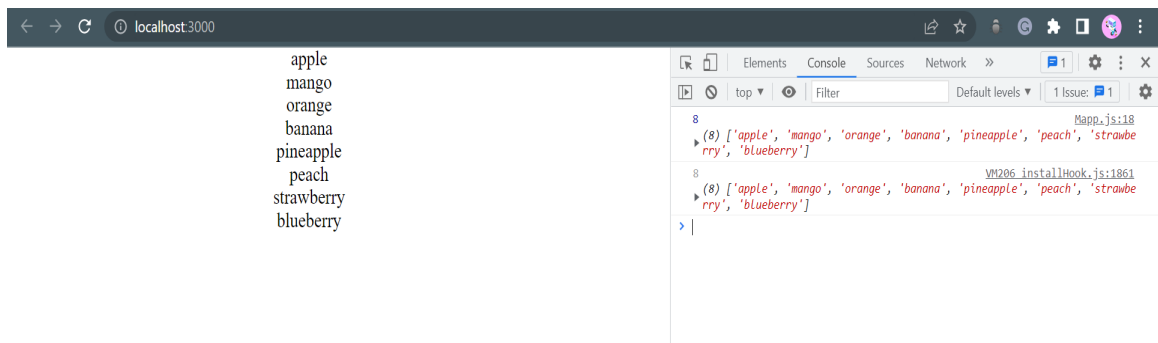
2. How to use map():

- **Array**

Let's see how to use a map() method with the help of the example:

```
export default function Mapp() {
  var fruits = [
    "apple",
    "mango",
    "orange",
    "banana",
    "pineapple",
    "peach",
    "strawberry",
    "blueberry",
  ];

  return (
    <div>
      {console.log(fruits.length, fruits)}
      {fruits.map((fruit, key) => <p key={key}>{fruit}</p>)}
    </div>
  );
}
```



Here's how it works:

- We take the fruits array sent down as fruits and place it within a div.
- Create a new `<p>` for every fruit in the array by using our `map()` to traverse over each one. Keep in mind that the `map()` function accepts a function to apply to each element of the array. In this instance, the function only returns a `<p>` tag.
- **An array of objects:**
 - A map is a sort of data collection that stores data in the form of key-value pairs. The value recorded in the map must be mapped to the key. The map function in JavaScript can be used on any array. In a single line of code, we use the map function to map every element of the array to the custom components. This eliminates the need to repeatedly refer to components and their properties as array items.
 - You may quickly present similarly grouped data to your user using `map()`. The array will update in your component if it changes elsewhere else. You may conveniently show information using mapping rather than manually adding a new HTML element for each entry.

Employees.js: An array containing the employee's name and employee Id as key-value pairs.

```
var employees = [
  {
    name: "Deepak",
    empId: "123",
  },
  {
    name: "Yash",
    empId: "124",
  },
  {
    name: "Raj",
    empId: "125",
  },
]
```

```

    },
    {
      name: "Rohan",
      empId: "126",
    },
    {
      name: "Puneet",
      empId: "127",
    },
    {
      name: "Vivek",
      empId: "128",
    },
    {
      name: "Aman",
      empId: "129",
    },
  ],
  export default employees;

```

Info.js: Username and employee Id as props in the Info component.

```

import React from 'react';

function Info (props) {
  return (
    <div key={key} style={{ margin: 20, textAlign: "left" }}>
      <p>Name: {props.name}</p>
      <p>Employee Id:{props.empId}</p>
      <hr />
    </div>
  )
}
export default Info;

```

Example: Using the map function to map data from the *Employees.js* file to a custom Info component.

App.js: In the App.js file, import the Employees array and the Info component. Map each element of the Employees arrays to the Info component using the map function.

```

import React from 'react';
import './App.css';
import Users from './list';
import Info from './Info';

function App() {

return (
  <div style={{margin:'20px'}} >
    {Users.map((e)=>{

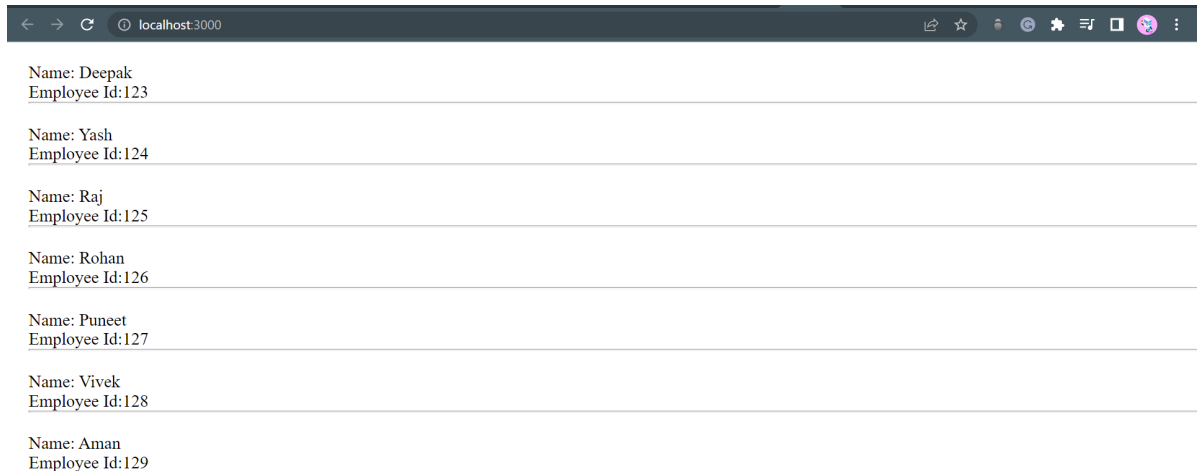
```

```

    return (
      <Info name={e.name} empId={e.rollNo}/>
    );
  }
</div>
);
}
export default App;

```

Output:



3. Mapping an array without using the map():

Example: Using the map function to map data from the Employees.js file to the Info component.

```

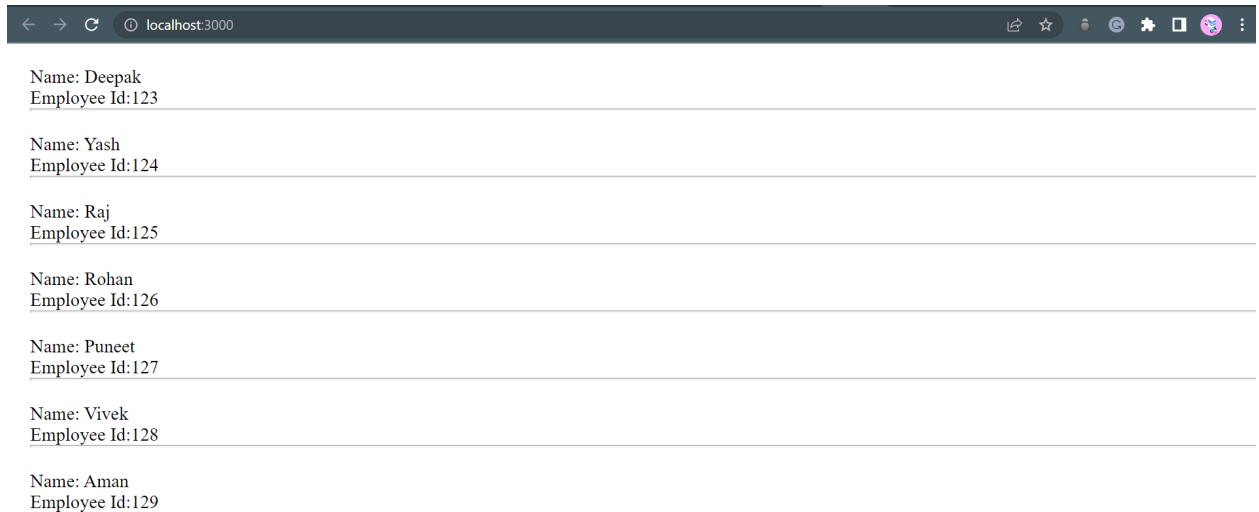
import React from "react";
import "../App.css";
import Users from "../list";
import Info from "../Info";

function App() {
  return (
    <div style={{ margin: "20px" }}>
      <Info name={Users[0].name} empId={Users[0].empId} />
      <Info name={Users[1].name} empId={Users[1].empId} />
      <Info name={Users[2].name} empId={Users[2].empId} />
      <Info name={Users[3].name} empId={Users[3].empId} />
      <Info name={Users[4].name} empId={Users[4].empId} />
      <Info name={Users[5].name} empId={Users[5].empId} />
      <Info name={Users[6].name} empId={Users[6].empId} />
    </div>
  );
}

```

```
export default App;
```

Output:



Both examples produce the same result, but utilizing the map method in the App.js file makes mapping array data to components easier and requires less code.