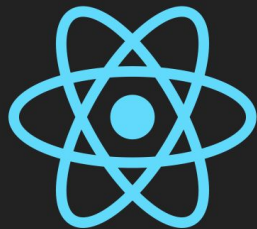


Lecture 3.1

Keys & Maps



Topics

- Handling Data
 - Axios API
 - Axios Config Options

Lists & Map

ES5 recap: map

- The `array.proto.map()` method creates a new array with the results of calling a provided function on every element in the calling array.

Array.map()

```
var array1 = [1, 4, 9, 16];

// pass a function to map
const map1 = array1.map(x => x * 2);

console.log(map1);
// expected output: Array [2, 8, 18, 32]
```

Array.map() vs for loop

```
let arr = [1, 2, 3]

let duplicatedArr = arr.map(function(el) {
  return el * 2
}) // [2, 4, 6]
```

```
let duplicatedArr = []
for (let i=0; i< arr.length; i++) {
  duplicatedArr.push(arr[i] * 2)
}
```

Rendering Multiple Components

You can build collections of elements and include them in JSX using curly braces {}.

Below, we loop through the numbers array using the JavaScript `map()` function. We return a `` element for each item. Finally, we assign the resulting array of elements to `listItems`:

```
const numbers = [1, 2, 3, 4, 5];  
const listItems = numbers.map((number) =>  
  <li>{number}</li>  
);
```

We include the entire `listItems` array inside a `` element, and render it to the DOM:

```
ReactDOM.render(  
  <ul>{listItems}</ul>,  
  document.getElementById('root')  
);
```

Basic List Components

Usually you would render lists inside of a component. To refactor the previous example..

```
function NumberList(props) {  
  const numbers = props.numbers;  
  const listItems = numbers.map((number) =>  
    <li>{number}</li>  
  );  
  return (  
    <ul>{listItems}</ul>  
  );  
}
```

```
const numbers = [1, 2, 3, 4, 5];  
ReactDOM.render(  
  <NumberList numbers={numbers} />,  
  document.getElementById('root')  
);
```

Basic List Components cont..

When you run this code, you'll be given a warning that a key should be provided for list items..

```
function NumberList(props) {  
  const numbers = props.numbers;  
  const listItems = numbers.map((number) =>  
    <li key={number.toString()}>  
      {number}  
    </li>  
  );  
  return (  
    <ul>{listItems}</ul>  
  );  
}  
  
const numbers = [1, 2, 3, 4, 5];  
ReactDOM.render(  
  <NumberList numbers={numbers} />,  
  document.getElementById('root')  
);
```

Keys

Keys

Keys help React identify which items have changed, are added, or are removed. Keys should be given to the elements inside the array to give the elements a stable identity:

```
const numbers = [1, 2, 3, 4, 5];
const listItems = numbers.map((number) =>
  <li key={number.toString()}>
    {number}
  </li>
);
```

The best way to pick a **key** is to use a string that uniquely identifies a list item among its siblings. Most often you would use **IDs** from your data as keys:

```
const todoItems = todos.map((todo) =>
  <li key={todo.id}>
    {todo.text}
  </li>
);
```

Incorrect Key Usage

Keys only make sense in the context of the surrounding array.

```
function ListItem(props) {  
  const value = props.value;  
  return (  
    // Wrong! There is no need to specify the key here:  
    <li key={value.toString()}>  
      {value}  
    </li>  
  );  
}
```

```
function NumberList(props) {  
  const numbers = props.numbers;  
  const listItems = numbers.map((number) =>  
    // Wrong! The key should have been specified here:  
    <ListItem value={number} />  
  );  
  return (  
    <ul>  
      {listItems}  
    </ul>  
  );  
}
```

```
function ListItem(props) {  
  // Correct! There is no need to specify the key here:  
  return <li>{props.value}</li>;  
}  
  
function NumberList(props) {  
  const numbers = props.numbers;  
  const listItems = numbers.map((number) =>  
    // Correct! Key should be specified inside the array.  
    <ListItem key={number.toString()} value={number} />  
  );  
  return (  
    <ul>  
      {listItems}  
    </ul>  
  );  
}
```

Correct Key Usage

If you extract a `ListItem` component, you should keep the key on the `<ListItem />` elements in the `array` rather than on the `` element in the `ListItem` itself.

Video - Nesting Components

<https://youtu.be/VLrGSRcxpHw>