React Lecture - 6

Rendering a List and Keys

Rendering Lists

Rendering Data from an Array

Suppose you have this list on your webpage. Here you have to repeat li again and again for each new list item. In such cases we can store the data in an array and render it on the page

```
<ul>
  <li>Creola Katherine Johnson: mathematician</li>
  <li>Mario José Molina-Pasquel Henríquez: chemist</li>
  <li>Mohammad Abdus Salam: physicist</li>
  <li>Percy Lavon Julian: chemist</li>
  <li>Subrahmanyan Chandrasekhar: astrophysicist</li>
</ul>
```

Rendering Data from an Array

1. Move the list data to an array

```
const people = [
  'Creola Katherine Johnson: mathematician',
  'Mario José Molina-Pasquel Henríquez: chemist',
  'Mohammad Abdus Salam: physicist',
  'Percy Lavon Julian: chemist',
  'Subrahmanyan Chandrasekhar: astrophysicist'
];
```

Rendering Data from an Array

2. Use array map method to create and array of elements or components with the array data.

```
const listItems = people.map(person => <li>{person}</li>);
```

Rendering Data from an Array

3. Return listItems from your component wrapped in a <ul>:

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

This method will render a list using an array

Did you notice a warning in the console ? We will discuss how to fix it later in this lecture.

Rendering Data from an Array

Now suppose we have this array of objects and we only want to render the person whose profession is chemist.

How can we do it ?

```
const people = [{
  id: 0,
  name: 'Creola Katherine Johnson',
  profession: 'mathematician',
}, {
  id: 1,
  name: 'Mario José Molina-Pasquel Henríquez',
  profession: 'chemist',
}, {
  id: 2,
  name: 'Mohammad Abdus Salam',
  profession: 'physicist',
}, {
  id: 3,
  name: 'Percy Lavon Julian',
  profession: 'chemist',
}, {
  id: 4,
  name: 'Subrahmanyan Chandrasekhar',
  profession: 'astrophysicist',
}];
```

Rendering Data from an Array

We can use filter method to create an array with people whose profession is chemist and then use the map method on the new array to create elements and render it

```
const people = [{
  id: 0,
  name: 'Creola Katherine Johnson',
  profession: 'mathematician',
}, {
  id: 1,
  name: 'Mario José Molina-Pasquel Henríquez',
  profession: 'chemist',
}, {
  id: 2,
  name: 'Mohammad Abdus Salam',
  profession: 'physicist',
}, {
  id: 3,
  name: 'Percy Lavon Julian',
  profession: 'chemist',
}, {
  id: 4,
  name: 'Subrahmanyan Chandrasekhar',
  profession: 'astrophysicist',
}];
```

Rendering Data from an Array

1. Create a new array of just “chemist” people, chemists, by calling filter() on the people filtering by person.profession === 'chemist':

```
const chemists = people.filter(person =>
  person.profession === 'chemist'
);
```

Rendering Data from an Array

2. Now map over chemists

```
const listItems = chemists.map(person =>
  <li>
    <img
      src={getImageUrl(person)}
      alt={person.name}
    />
    <p>
      <b>{person.name}:</b>
      {' ' + person.profession + ' '}
      known for {person.accomplishment}
    </p>
  </li>
);
```

Rendering Data from an Array

3. Lastly, return the listItems from your component:

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

Keys in React

JSX elements directly inside a map() call always need keys!

Keys tell React which array item each component corresponds to, so that it can match them up later. This becomes important if your array items can move (e.g. due to sorting), get inserted, or get deleted. A well-chosen key helps React infer what exactly has happened, and make the correct updates to the DOM tree.

Rules of Keys

- Keys must be unique among siblings. However, it’s okay to use the same keys for JSX nodes in different arrays.

- Keys must not change or that defeats their purpose! Don’t generate them while rendering.

Why does react need keys ?

Imagine that files on your desktop didn't have names. Instead, you'd refer to them by their order — the first file, the second file, and so on. You could get used to it, but once you delete a file, it would get confusing. The second file would become the first file, the third file would be the second file, and so on.

File names in a folder and JSX keys in an array serve a similar purpose

They let us uniquely identify an item between its siblings. A well-chosen key provides more information than the position within the array. Even if the position changes due to reordering, the key lets React identify the item throughout its lifetime.