Map list key into jsx using map function

**Mapping and rendering arrays**

Rendering Lists

You will often want to display multiple similar components from a collection of data. You can use the JavaScript array methods to manipulate an array of data. On this page, you’ll use filter() and map() with React to filter and transform your array of data into an array of components.

**Rendering data from arrays**

Say that you have a list of content.

<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>

The only difference among those list items is their contents, their data. You will often need to show several instances of the same component using different data when building interfaces: from lists of comments to galleries of profile images. In these situations, you can store that data in JavaScript objects and arrays and use methods like [map()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map) and [filter()](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array/filter) to render lists of components from them.

Here’s a short example of how to generate a list of items from an array:

* **Move** the data into 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'

];

* **Map** the people members into a new array of JSX nodes, listItems:

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

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

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

example map()

App.js

import { people } from './data.js';

import { getImageUrl } from './utils.js';

export default function List() {

const listItems = people.map(person =>

<li key={person.id}>

<img

src={getImageUrl(person)}

alt={person.name}

/>

<p>

<b>{person.name}</b>

{' ' + person.profession + ' '}

known for {person.accomplishment}

</p>

</li>

);

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

}

Data.js

export const people = [{

id: 0, // Used in JSX as a key

name: 'Creola Katherine Johnson',

profession: 'mathematician',

accomplishment: 'spaceflight calculations',

imageId: 'MK3eW3A'

}, {

id: 1, // Used in JSX as a key

name: 'Mario José Molina-Pasquel Henríquez',

profession: 'chemist',

accomplishment: 'discovery of Arctic ozone hole',

imageId: 'mynHUSa'

}, {

id: 2, // Used in JSX as a key

name: 'Mohammad Abdus Salam',

profession: 'physicist',

accomplishment: 'electromagnetism theory',

imageId: 'bE7W1ji'

}, {

id: 3, // Used in JSX as a key

name: 'Percy Lavon Julian',

profession: 'chemist',

accomplishment: 'pioneering cortisone drugs, steroids and birth control pills',

imageId: 'IOjWm71'

}, {

id: 4, // Used in JSX as a key

name: 'Subrahmanyan Chandrasekhar',

profession: 'astrophysicist',

accomplishment: 'white dwarf star mass calculations',

imageId: 'lrWQx8l'

}];

export function getImageUrl(person) {

return (

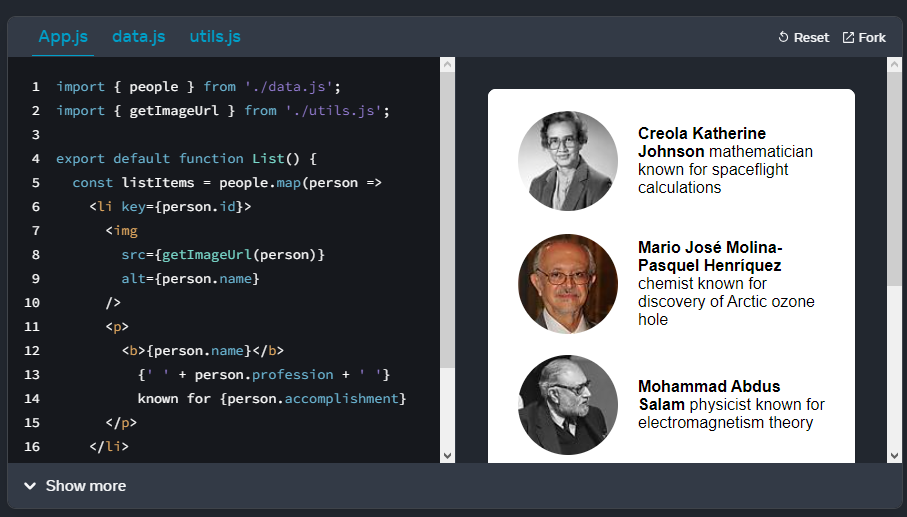
'https://i.imgur.com/' +

person.imageId +

's.jpg'

);

}



**Filtering arrays of items**

This data can be structured even more.

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',

}, {

name: 'Percy Lavon Julian',

profession: 'chemist',

}, {

name: 'Subrahmanyan Chandrasekhar',

profession: 'astrophysicist',

}];

Let’s say you want a way to only show people whose profession is 'chemist'. You can use JavaScript’s filter() method to return just those people. This method takes an array of items, passes them through a “test” (a function that returns true or false), and returns a new array of only those items that passed the test (returned true).

You only want the items where profession is 'chemist'. The “test” function for this looks like (person) => person.profession === 'chemist'. Here’s how to put it together:

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'

);

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>

);

Lastly, return the listItems from your component:

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

**example filter()**

App.js

import { people } from './data.js';

import { getImageUrl } from './utils.js';

export default function List() {

const chemists = people.filter(person =>

person.profession === 'chemist'

);

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>

);

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

}

export const people = [{

id: 0,

name: 'Creola Katherine Johnson',

profession: 'mathematician',

accomplishment: 'spaceflight calculations',

imageId: 'MK3eW3A'

}, {

id: 1,

name: 'Mario José Molina-Pasquel Henríquez',

profession: 'chemist',

accomplishment: 'discovery of Arctic ozone hole',

imageId: 'mynHUSa'

}, {

id: 2,

name: 'Mohammad Abdus Salam',

profession: 'physicist',

accomplishment: 'electromagnetism theory',

imageId: 'bE7W1ji'

}, {

id: 3,

name: 'Percy Lavon Julian',

profession: 'chemist',

accomplishment: 'pioneering cortisone drugs, steroids and birth control pills',

imageId: 'IOjWm71'

}, {

id: 4,

name: 'Subrahmanyan Chandrasekhar',

profession: 'astrophysicist',

accomplishment: 'white dwarf star mass calculations',

imageId: 'lrWQx8l'

}];

export function getImageUrl(person) {

return (

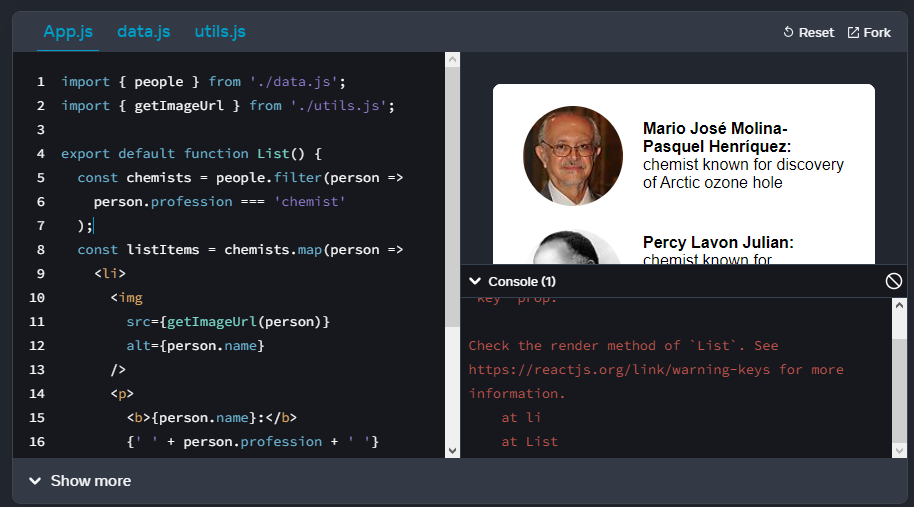
'https://i.imgur.com/' +

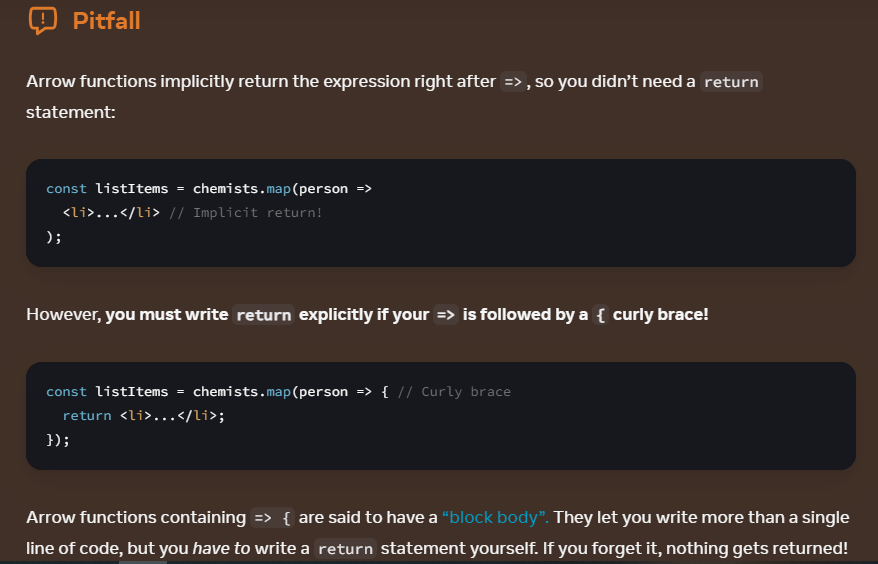
person.imageId +

's.jpg'

);

}





**Keeping list items in order with key**

Notice that all the sandboxes above show an error in the console:

Console

Warning: Each child in a list should have a unique “key” prop.

You need to give each array item a key — a string or a number that uniquely identifies it among other items in that array:

<li key={person.id}>...</li>

**Note**

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.

Rather than generating keys on the fly, you should include them in your data: