

# React (Part 3)

Conditional Rendering and Lists

## Learning Outcomes

- Implements conditional rendering
- Implements lists of elements

...in React with JSX

## Starter Project

- Download and unpack
- Open in VSCode
- Run **npm install**
- Run **npm start**
- Look around code



## Conditional Rendering

Try to find a way:

- The star icon is only displayed if the movie's **favorite** attribute is **true**.

Hint: <https://reactjs.org/docs/conditional-rendering.html>

Conditional Rendering

Many paths to success...

## Element Variables

- React Elements (JSX Expression) can be assigned to Variables

```
let myElement;  
if(isCool) {  
  myElement = <span className="cool">cool</span>;  
}  
else {  
  myElement = <span className="not_cool">Not cool</span>;  
}
```

- Insert them into your returned JSX Expression using curly brackets

```
return <div>  
  <h1>Conditional Rendering</h1>  
  { myElement }  
</div>;
```

# Logical Operators

- React does not display the values true and false
- We can use Logical Operators to combine conditions and elements

```
return <div>
  <h1>Conditional Rendering</h1>
  { isCool && <span className="cool">cool</span> }
  { isCool || <span className="not_cool">Not cool</span>}
</div>;
```

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical\\_AND](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND)
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical\\_OR](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_OR)

## Conditional (Ternary) Operator

Use the conditional operator to provide alternative values for true and false condition

```
return <div>
  <h1>Conditional Rendering</h1>
  { isCool ? <span className="cool">cool</span> : <span className="not_cool">Not cool</span> }
</div>;
```

[https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Operators/Conditional\\_Operator](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Operators/Conditional_Operator)



## Create small component with multiple returns

```
function CoolnessIndicator(props) {  
  if (props.isCool) {  
    return <span className="cool">cool</span>;  
  }  
  return <span className="not_cool">Not cool</span>;  
}
```

Use it:

```
return <div>  
  <h1>Conditional Rendering</h1>  
  <CoolnessIndicator isCool={isCool} />  
</div>;
```



## Conditional Rendering

Use a conditional (ternary) operator:

Show the "thumbs up" icon only if the movie's rating is at least 8.7, otherwise show the "thumbs down" icon.



## Conditional Rendering

Choose one of the discussed paths:

Show the "old time classic" analysis element only if the movie is from before the year 2000 and has a rating of at least 8.7

## Recap: Conditional Rendering

- Element Variables
  - flexible
- Logical Operator
  - Useful for "render element only if" scenarios (no alternatives)
- Conditional (ternary) operator
  - Useful for alternating elements
- Create own component
  - flexible, reusable

## Lists

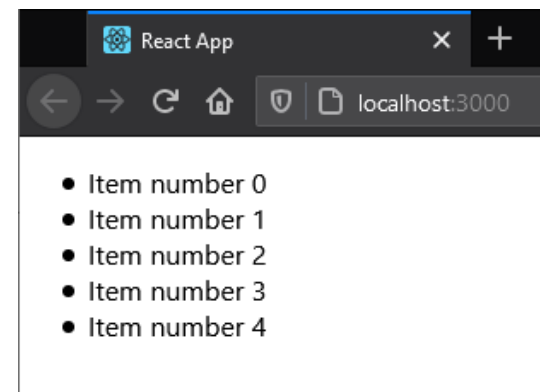
Look at App.js

- Same line repeated six times
- What if there are actually seven movies? What if there were only five?

# Lists

React handles arrays of elements just fine:

```
const elements = [];  
for(let i = 0; i < 5; i++) {  
  elements.push(<li>Item number {i}</li>)  
}  
  
return <ul>{elements}</ul>;
```





## Lists

Try to find a way:

- From the array **movies** create an array of **MovieInfo** component elements and insert it into the returned JSX in **App** component

Should work no matter what size the **movies** array is.

Hint: <https://reactjs.org/docs/lists-and-keys.html>

## Lists using loop

Easy to understand path:

- Create new empty array
- Loop over original array, adding entries to new array

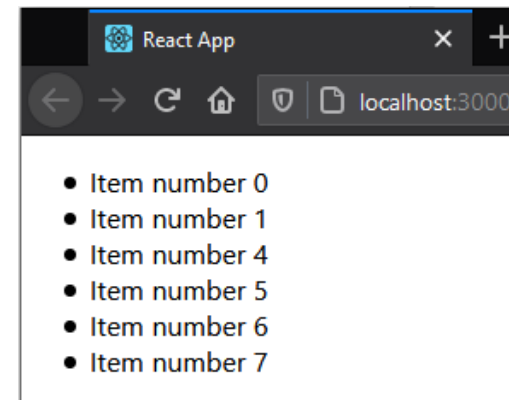


## Lists using map

Alternate path if you're comfortable with Array map method:

Insert map expression in JSX

```
const numbers = [0, 1, 4, 5, 6, 7];  
return <ul>  
  {numbers.map(i => <li>Item number {i}</li>)}  
</ul>;
```



# Keys

```
! Warning: Each child in a list should have a unique "key" prop.  
  
    Check the render method of `BookTable`. See https://fb.me/react-warning-keys for more information.  
      in tr (at BookTable.tsx:133)  
      in BookTable (at App.tsx:13)  
      in main (at App.tsx:11)  
      in App (at src/index.tsx:8)  
      in StrictMode (at src/index.tsx:7)
```

React asks for your help to efficiently update the list of elements

<https://fb.me/react-warning-keys>

Our array of movies never changes so we can ignore this warning for now.

You can fix it by assigning a unique value to the element's key prop (e.g. the movie title)

## Lists: Recap

From an array of Objects (movies), create an array of React Elements

- Using a for loop
- Using the array map method



## Things to try

- Create an array with some names (your friends or favorite musicians) and have them rendered into an `<ol>` List
- Add an array of actor names to all movie objects. Show the list of actors in the **MovieInfo** component
- Show a list of all actors (from all movies)
- Show a list of only the movie names
- Show **MovieInfo** components only for the favorite movies
- Implement a few different results for the "Analysis"
  - Underrated favorite
  - New bad movie

## Learning Outcomes Review

- Implements conditional rendering
- Implements lists of elements

...in React with JSX