

React Hooks

Hooks let you use more of React's features without classes

<https://reactjs.org/docs/hooks-intro.html#motivation>

Hooks are JavaScript functions

They impose two additional rules:

- A. Only **call Hooks at the top level**. Don't call Hooks inside loops, conditions, or nested functions.
- B. Only **call Hooks from React function components** or your own **custom Hooks**.

useState

The useState hook is how you **make your function component stateful**

```
const [value, setValue] = useState(null);
```

The hook returns an array with two values:

the state value you are storing and

a function that allows you to update it.

When instantiating the hook,

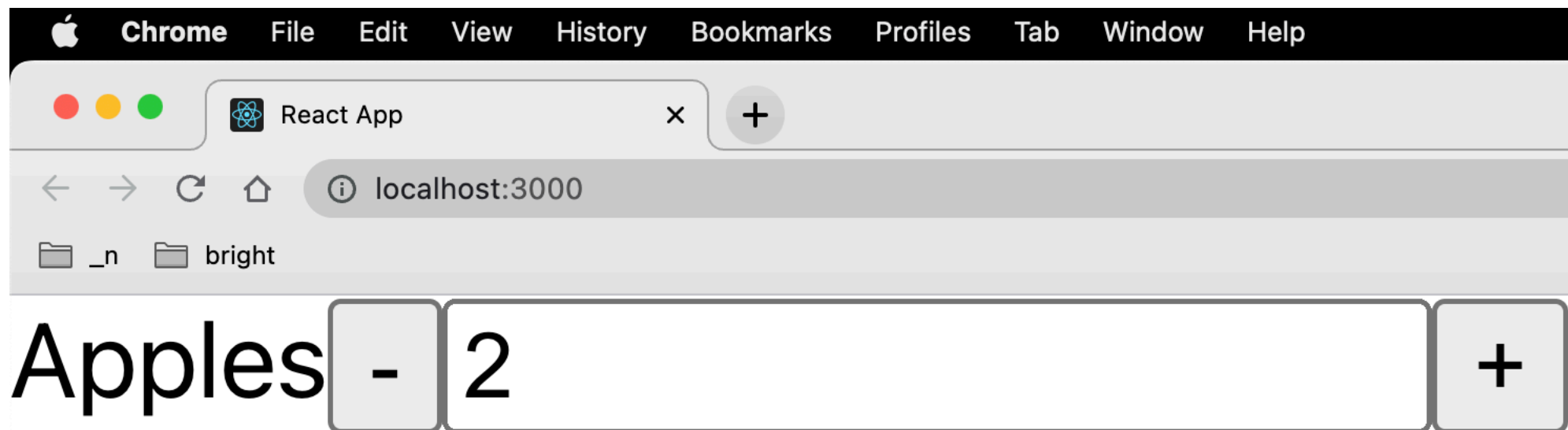
the initial state value is determined by whatever value you pass it.

TODO

- npx create-react-app hooks-playground
- cd hook-playground
- npm start
- refactor App.js

```
function App() {  
  return (  
    <></>  
  );  
}  
  
export default App;
```

TODO



```
function App() {  
  return (  
    <GetItem />  
  );  
}
```

```
const GetItem = () => {  
  return (  
    <>an item</>  
  );  
}
```

```
export default App;
```



```
function App() {  
  return (  
    <CartItem />  
  );  
}
```

```
const CartItem = () => {  
  return (  
    <>  
      <span>{''}</span>  
      <button onClick={''}>-</button>  
      <input value={''} readOnly/>  
      <button onClick={''}>+</button>  
    </>  
  );  
}
```

```
export default App;
```

```
function App() {  
  return (  
    <CartItem />  
  );  
}
```

```
const CartItem = ({name, quantity, remove, add }) => {  
  return (  
    <>  
      <span>`${name}`</span>  
      <button onClick={remove}>-</button>  
      <input value={quantity} readOnly/>  
      <button onClick={add}>+</button>  
    </>  
  );  
}
```

```
export default App;
```

```
import { useState } from "react";

function App() {
  const [quantity, setQuantity] = useState(0);

  return (
    <CartItem
      name={"Apples"}
      quantity={quantity}
      remove={
        () => {
          if (quantity > 0) {
            setQuantity(quantity - 1);
          }
        }
      }
      add={() => setQuantity(quantity + 1)}
    />
  );
}

const CartItem = ({ name, quantity, remove, add }) => {
  return (
    <>
      <span>`${name}`</span>
      <button onClick={remove}>-</button>
      <input value={quantity} readOnly />
      <button onClick={add}>+</button>
    </>
  );
}

export default App;
```

useEffect

The useEffect hook is used to **handle side-effects** caused by a component's actions, e.g.:

- **fetching data** from an API,
- **attaching event listeners** to the document,
- **manipulating the DOM**

```
useEffect(() => {  
    // Runs every time the component renders  
})
```

```
useEffect(() => {  
    // Runs only on initial render  
}, []) // Optional second argument: the dependency array
```

```
useEffect(() => {  
    // Runs only when 'apiData' changes  
}, [apiData])
```

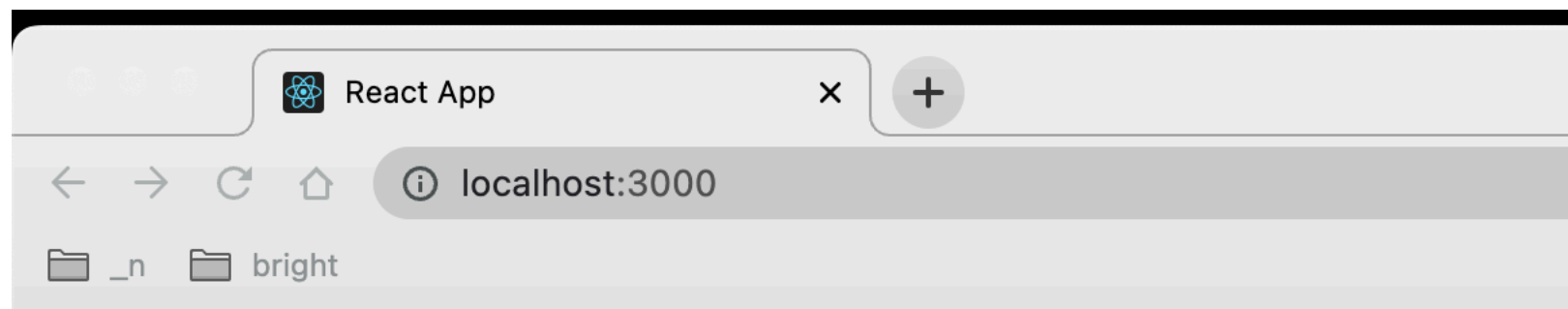
The **useEffect** takes two values:

- a function defining the logic to be executed and
- an optional array called the "dependency array".

<https://reactjs.org/docs/hooks-effect.html#tip-optimizing-performance-by-skipping-effects>

TODO

Fetch api <https://swapi.dev/api/films/1> and display a movie title



Name: Attack of the Clones

```
function App() {  
  return (  
    <></>  
  );  
}
```

```
export default App;
```

```
function App() {  
  return (  
    <GetItem />  
  );  
}
```

```
const GetItem = () => {  
  return (  
    <>  
    <p>Name: </p>  
    </>  
  );  
}
```

```
export default App;
```



```
import {useState} from "react";
```

```
function App() {  
  return (  
    <GetItem />  
  );  
}
```

```
const GetItem = () => {
```

```
  const [data, setData] = useState({});
```

```
  return (  
    <>  
      { /* Use Truthy object to avoid errors */}  
      <p>Name: {data && data.title}</p>  
    </>  
  );  
}
```

```
export default App;
```

```
import {useEffect, useState} from "react";
```

```
function App() {  
  return (  
    <GetItem id="5" />  
  );  
}
```

```
const GetItem = (props) => {
```

```
  const [data, setData] = useState({});
```

```
  useEffect(  
    /*
```

```
    Two arguments: an arrow function and an array (empty or with some properties)
```

```
    https://reactjs.org/docs/hooks-effect.html#tip-optimizing-performance-by-skipping-effects
```

```
    */
```

```
    ()=>{
```

```
      fetch(`https://swapi.dev/api/films/${props.id}`)
```

```
      .then(response => response.json())
```

```
      .then(result => setData(result))
```

```
    }, []
```

```
  )
```

```
  return (  
    <>
```

```
    <p>Name: {data.title}</p>
```

```
  </>
```

```
  );
```

```
}
```

```
export default App;
```

TODO

fetch the API and display a list of movies



Featured Movies

- A New Hope, Director: George Lucas, Release: 1977-05-25
- The Empire Strikes Back, Director: Irvin Kershner, Release: 1980-05-17
- Return of the Jedi, Director: Richard Marquand, Release: 1983-05-25
- The Phantom Menace, Director: George Lucas, Release: 1999-05-19
- Attack of the Clones, Director: George Lucas, Release: 2002-05-16
- Revenge of the Sith, Director: George Lucas, Release: 2005-05-19

```
import { useEffect, useState } from "react";
```

```
function App() {  
  return (  
    <MyMovieList />  
  );  
}
```

```
const MyMovieList = () => {
```

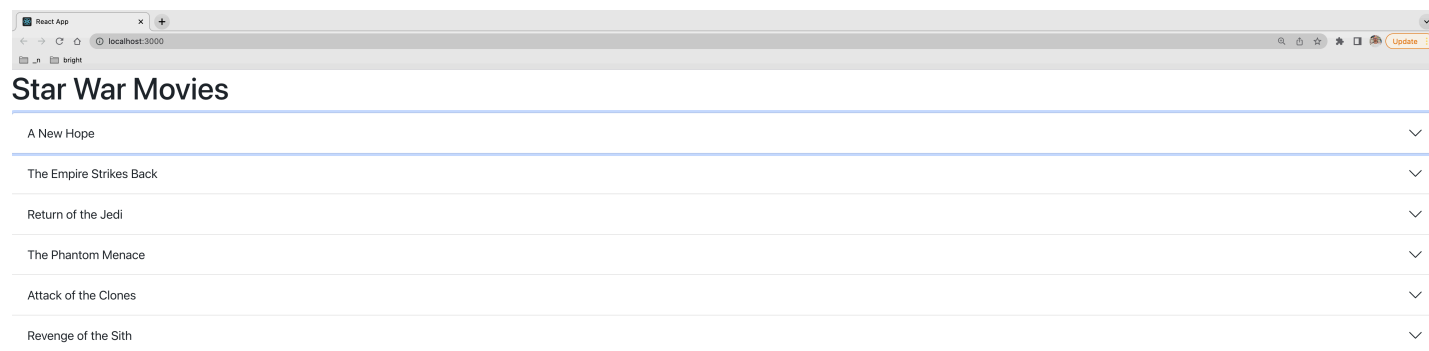
```
  const [movies, setMovies] = useState([]);
```

```
  useEffect(  
    () => {  
      fetch(`https://swapi.dev/api/films/`)  
        .then(response => response.json())  
        .then(result => setMovies(result.results))  
    }, []  
  )
```

```
  return (  
    <>  
      <h2>Featured Movies</h2>  
      {movies.map((movie) => (  
        <li key={movie.id}>  
          `${movie.title}, Director: ${movie.director}, Release: ${movie.release_date}`  
        </li>  
      ))}  
    </>  
  );  
}
```

```
export default App;
```

fetch -> react-bootstrap Accordion



* react-bootstrap needs to be installed before using: `npm install react-bootstrap bootstrap`

```
import { useEffect, useState } from "react";
import { Accordion } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
function App() {
  return (
    <MyMovieList />
  );
}
```

```
const MyMovieList = () => {
```

```
  const [movies, setMovies] = useState([]);
```

```
  useEffect(
```

```
    () => {
      fetch(`https://swapi.dev/api/films/`)
        .then(response => response.json())
        .then(result => setMovies(result.results))
    }, []
  )
```

```
  return (
    <>
      <h1>Star War Movies</h1>
      <Accordion>
        {movies.map((movie) => (
          <Accordion.Item eventKey="0">
            <Accordion.Header>{movie.title}</Accordion.Header>
            <Accordion.Body>
              {movie.opening_crawl}
            </Accordion.Body>
          </Accordion.Item>
        ))}
      </Accordion>
    </>
  );
}
```

```
export default App;
```

fetch -> react-bootstrap Table

Star War Movies

#	Title	Release Date
4	A New Hope	1977-05-25
5	The Empire Strikes Back	1980-05-17
6	Return of the Jedi	1983-05-25
1	The Phantom Menace	1999-05-19
2	Attack of the Clones	2002-05-16
3	Revenge of the Sith	2005-05-19

* react-bootstrap needs to be installed before using: `npm install react-bootstrap bootstrap`

```
import { Table } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
```

```
function App() {
  return (
    <MyMovieList />
  );
}
```

```
const MyMovieList = () => {
```

```
  const [movies, setMovies] = useState([]);
```

```
  useEffect(
```

```
    () => {
      fetch(`https://swapi.dev/api/films/`)
        .then(response => response.json())
        .then(result => setMovies(result.results))
    }, []
  )
```

```
  return (
    <>
      <h1>Star War Movies</h1>
      <Table striped bordered hover variant="dark">
        <thead>
          <tr>
            <th>#</th>
            <th>Title</th>
            <th>Release Date</th>
          </tr>
        </thead>
        <tbody>
          {movies.map((movie) => (
            <tr>
              <td>{movie.episode_id}</td>
              <td>{movie.title}</td>
              <td>{movie.release_date}</td>
            </tr>
          ))}
        </tbody>
      </Table>
    </>
  );
}
```

```
export default App;
```


fetch <- json server

TODO

- create a folder, e.g. **data** in the project's root directory
- inside **data**, create a file **db.json**
- copy paste data from the next slide
- run the JSON server (new terminal):
 - **npx json-server --watch data/db.json --port 8080**
- test the running server in a browser:
 - **http://localhost:8080/movies**
- **Refactor the previous example to call the JSON server & display the data**

```
{
  "movies": [
    {
      "title": "A New Hope",
      "episode_id": 4,
      "opening_crawl": "It is a period of civil war.\r\nRebel spaceships, striking\r\nfrom a hidden base, have won\r\ntheir first victory against\r\nthe evil Galactic Empire.\r\n\r\nDuring the battle, Rebel\r\nspies managed to steal secret\r\nplans to the Empire's\r\nultimate weapon, the DEATH\r\nSTAR, an armored space\r\nstation with enough power\r\nto destroy an entire planet.\r\n\r\nPursued by the Empire's\r\nsinister agents, Princess\r\nLeia races home aboard her\r\nstarship, custodian of the\r\nstolen plans that can save her\r\npeople and restore\r\nfreedom to the galaxy....",
      "director": "George Lucas",
      "producer": "Gary Kurtz, Rick McCallum",
      "release_date": "1977-05-25"
    },
    {
      "title": "Return of the Jedi",
      "episode_id": 6,
      "opening_crawl": "Luke Skywalker has returned to\r\nhis home planet of Tatooine in\r\nan attempt to rescue his\r\nfriend Han Solo from the\r\nclutches of the vile gangster\r\nJabba the Hutt.\r\n\r\nLittle does Luke know that the\r\nGALACTIC EMPIRE has secretly\r\nbegun construction on a new\r\narmored space station even\r\nmore powerful than the first\r\ndreaded Death Star.\r\n\r\nWhen completed, this ultimate\r\nweapon will spell certain doom\r\nfor the small band of rebels\r\nstruggling to restore freedom\r\nto the galaxy...",
      "director": "Richard Marquand",
      "producer": "Howard G. Kazanjian, George Lucas, Rick McCallum",
      "release_date": "1983-05-25"
    },
    {
      "title": "The Phantom Menace",
      "episode_id": 1,
      "opening_crawl": "Turmoil has engulfed the\r\nGalactic Republic. The taxation\r\nof trade routes to outlying star\r\nsystems is in dispute.\r\n\r\nHoping to resolve the matter\r\nwith a blockade of deadly\r\nbattleships, the greedy Trade\r\nFederation has stopped all\r\nshipping to the small planet\r\nof Naboo.\r\n\r\nWhile the Congress of the\r\nRepublic endlessly debates\r\nthis alarming chain of events,\r\nthe Supreme Chancellor has\r\nsecretly dispatched two Jedi\r\nKnights, the guardians of\r\npeace and justice in the\r\ngalaxy, to settle the conflict....",
      "director": "George Lucas",
      "producer": "Rick McCallum",
      "release_date": "1999-05-19"
    },
    {
      "title": "Attack of the Clones",
      "episode_id": 2,
      "opening_crawl": "There is unrest in the Galactic\r\nSenate. Several thousand solar\r\nsystems have declared their\r\nintentions to leave the Republic.\r\n\r\nThis separatist movement,\r\nunder the leadership of the\r\nmysterious Count Dooku, has\r\nmade it difficult for the limited\r\nnumber of Jedi Knights to maintain\r\npeace and order in the galaxy.\r\n\r\nSenator Amidala, the former\r\nQueen of Naboo, is returning\r\nto the Galactic Senate to vote\r\non the critical issue of creating\r\nan ARMY OF THE REPUBLIC\r\nto assist the overwhelmed\r\nJedi....",
      "director": "George Lucas",
      "producer": "Rick McCallum",
      "release_date": "2002-05-16"
    },
    {
      "title": "Revenge of the Sith",
      "episode_id": 3,
      "opening_crawl": "War! The Republic is crumbling\r\nunder attacks by the ruthless\r\nSith Lord, Count Dooku.\r\n\r\nThere are heroes on both sides.\r\n\r\nEvil is everywhere.\r\n\r\nIn a stunning move, the\r\nfiendish droid leader, General\r\nGrievous, has swept into the\r\nRepublic capital and kidnapped\r\nChancellor Palpatine, leader of\r\nthe Galactic Senate.\r\n\r\nAs the Separatist Droid Army\r\nattempts to flee the besieged\r\ncapital with their valuable\r\nhostage, two Jedi Knights lead a\r\ndesperate mission to rescue the\r\ncaptive Chancellor....",
      "director": "George Lucas",
      "producer": "Rick McCallum",
      "release_date": "2005-05-19"
    }
  ]
}
```

Work on your own as long as you can :).

The solution is in the next slide.

```

import { useEffect, useState } from "react";
import { Table } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";

function App() {
  return (
    <MyMovieList />
  );
}

const MyMovieList = () => {

  const [movies, setMovies] = useState([]);

  useEffect(
    () => {
      fetch(`http://localhost:8080/movies`)
        .then(response => response.json())
        .then(result => setMovies(result))
    }, []
  );

  return (
    <>
      <h1>Star War Movies</h1>
      <Table striped bordered hover variant="dark">
        <thead>
          <tr>
            <th>#</th>
            <th>Title</th>
            <th>Release Date</th>
          </tr>
        </thead>
        <tbody>
          {movies.map((movie) => (
            <tr>
              <td>{movie.episode_id}</td>
              <td>{movie.title}</td>
              <td>{movie.release_date}</td>
            </tr>
          ))}
        </tbody>
      </Table>
    </>
  );
}

export default App;

```

React functional component pass props () or ({})