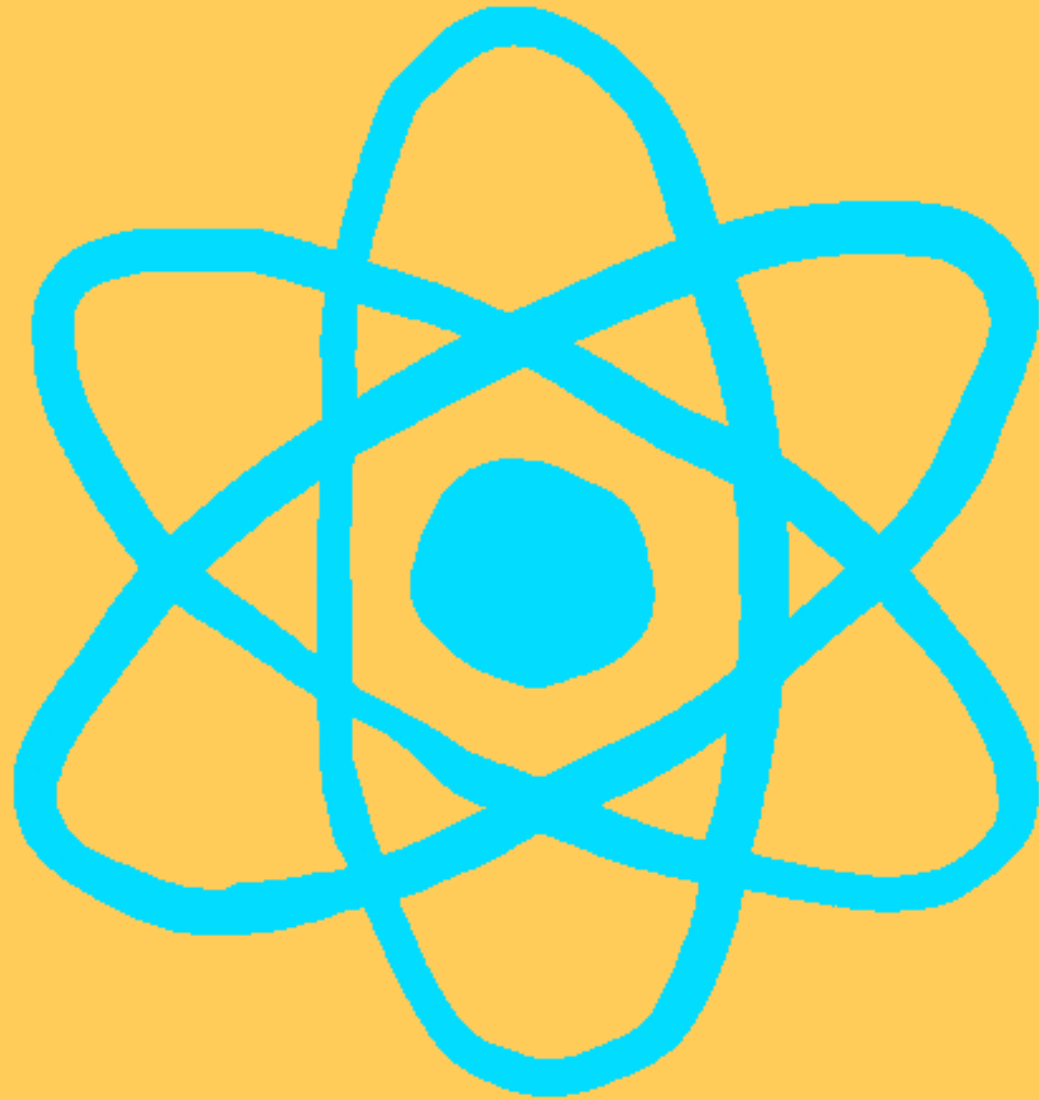


# *FRONTEND. REACT. ЛЕКЦІЯ 3*

## LIFE CYRCLE, EXPORT, IMPORT, CONDITIONAL RENDER, LISTS, KEYS.





# СПИСОК ОЧІКУВАННЯ

- Проблема з відображенням коду

```
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:
*** STOP: 0x0000004e (0x00000099, 0x00900009, 0x00000900, 0x00000900)

Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
```

# ВИКОНАНО

- Лекція 1, 2 вже на порталі
- Завдання на самовиконання додано
- Корисні посилання додані
- Рoadмепи додані



# ЗАВДАННЯ НА САМОПІДГОТОВКУ

- Генерація поля для sudoku
- Реалізація інтерфейсу за допомогою UI бібліотек (картинка нижче).  
Допускається покращення інтерфейсу
- Функціонал вперед, назад повинен використовувати редукс.  
Допускається використання рекойлу
- Повинен бути алгоритм підсвідки помилки
- Користувач має змогу почистити одне з полів форми
- Реалізувати можливість підказки
- ЗАВДАННЯ на 100 балів. Додати розв'язання легких sudoku без використання сторонніх бібліотек

# ROADMAP

▶ We now have a YouTube Channel.



NEW [Topic videos being made on YouTube](#) »

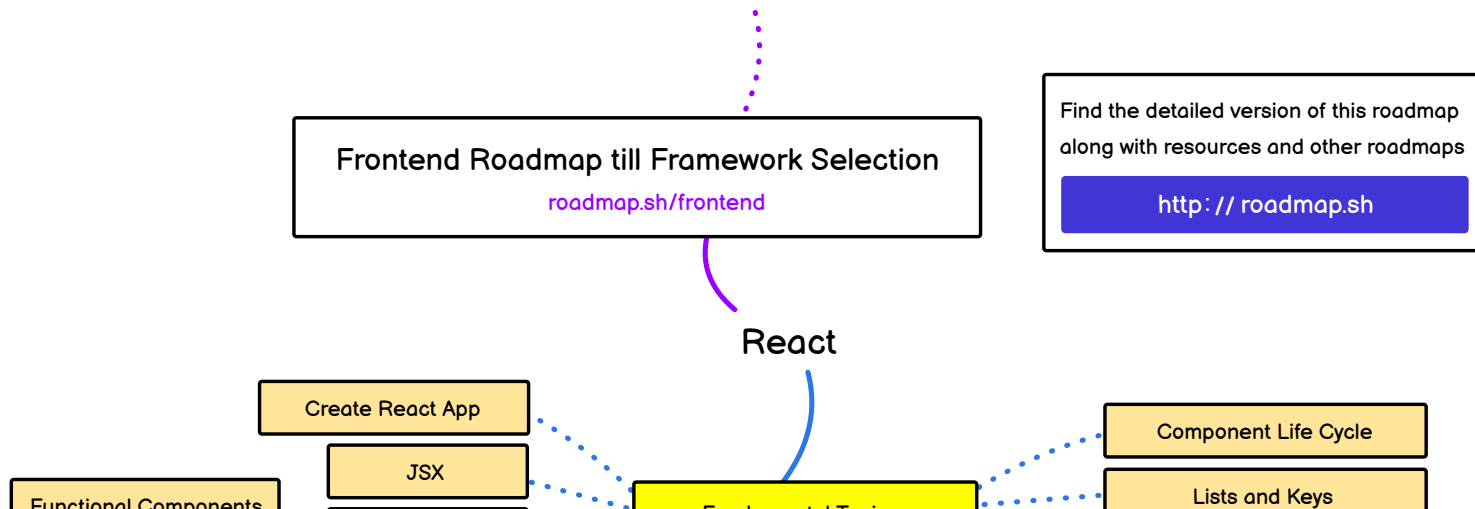
## React Developer

Everything that is there to learn about React and the ecosystem in 2022.

← @ Subscribe

Suggest Changes

NEW Resources are here, try clicking any nodes.



# *ROADMAP*

[https://miro.com/app/board/o9J\\_IVQm6fI=](https://miro.com/app/board/o9J_IVQm6fI=)





# ВЕБ ПЕРФОМЕНС

## Балачки про все на світі під пивце

Роман Савіцький

в гостях у Віталія Рубана та Сергія Бабіча

говорим про веб-перформанс

30 вересня 19:00

YouTube-канал Віталія Рубана



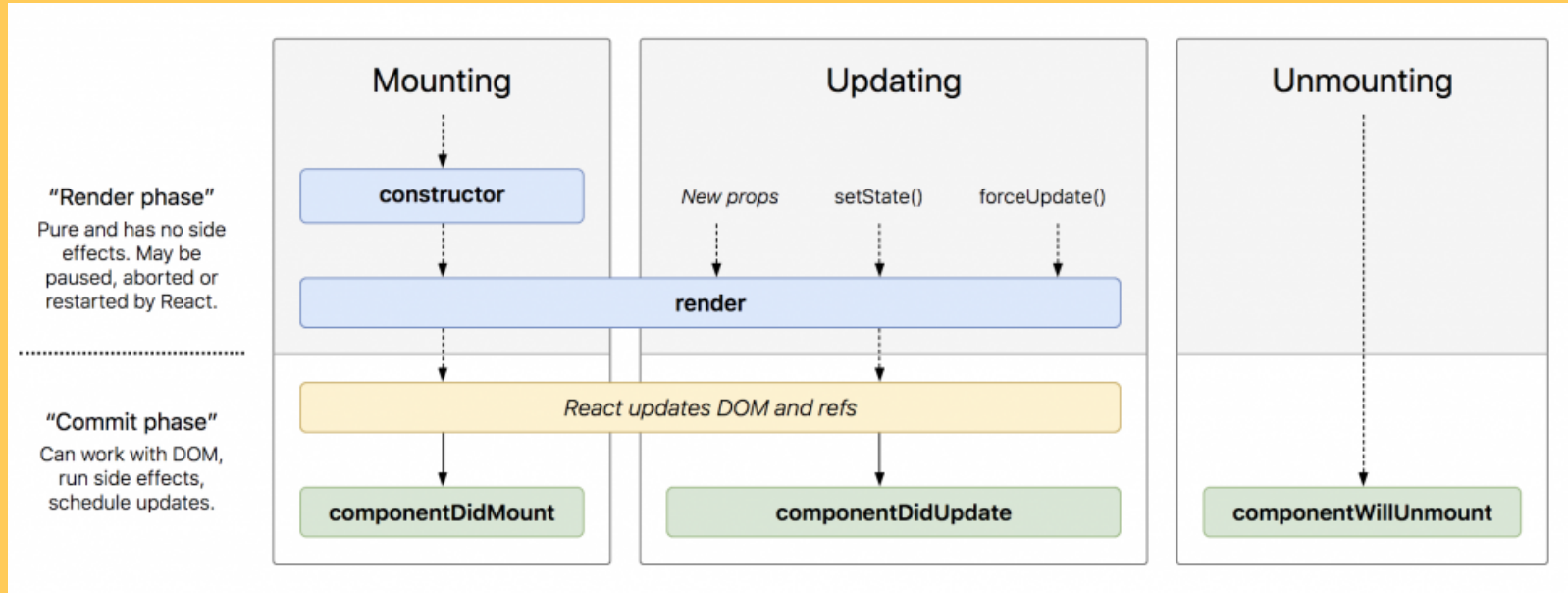
# DEVCHALLENGE

<https://www.devchallenge.it/>

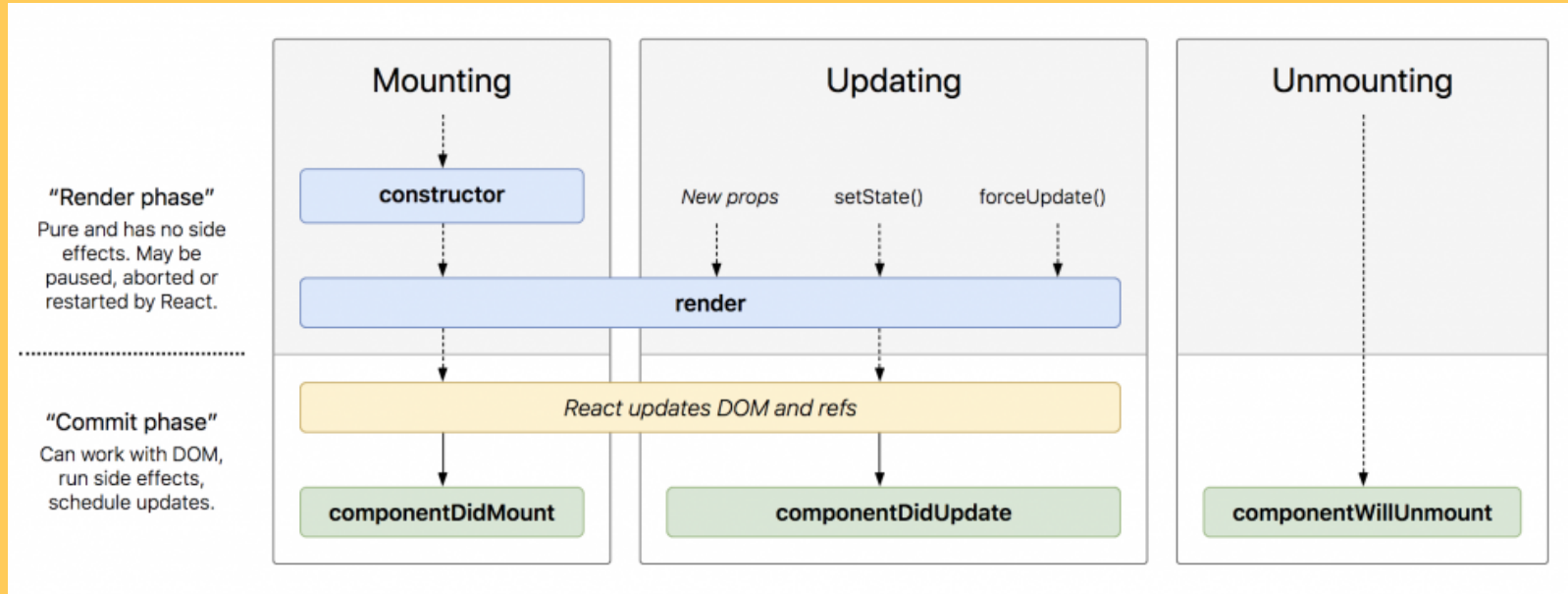
EN



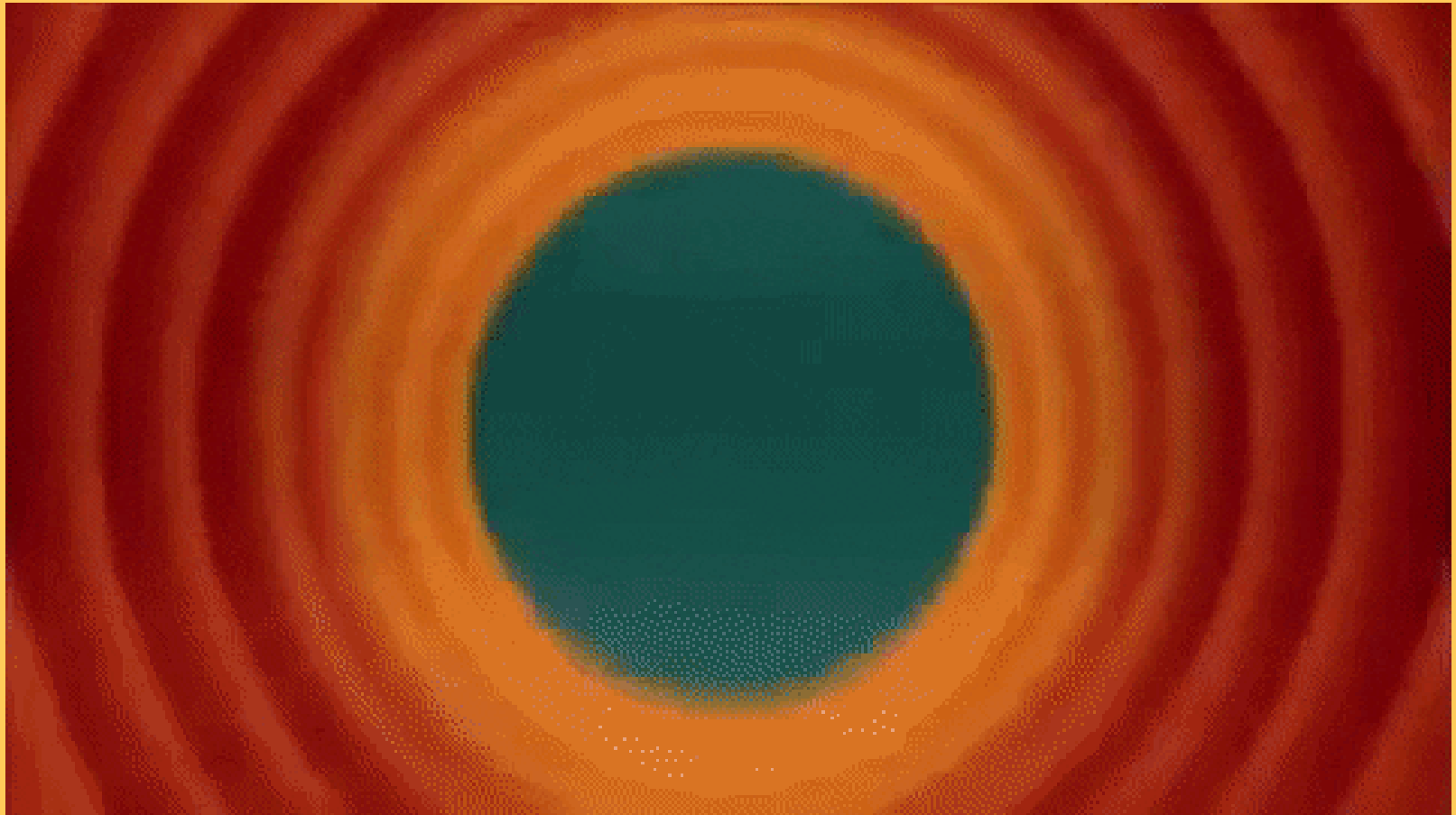
# МЕТОДИ ЖИТТЕВОГО ЦИКЛУ



# МЕТОДИ ЖИТТЕВОГО ЦИКЛУ



- useEffect



# УМОВНИЙ РЕНДЕРИНГ



# УМОВНИЙ РЕНДЕРИНГ

- Зберігання в змінну

```
export function idNumber(props) {
  const { id } = props;
  let message;

  if (id) {
    message = <title tag="p">{id}</title>;
  } else {
    message = <title tag="p">The ID is not exist</title>;
  }

  return (
    < >
      {message}
    < />
  );
}
```

# УМОВНИЙ РЕНДЕРИНГ

- JS like. If-else or ternary

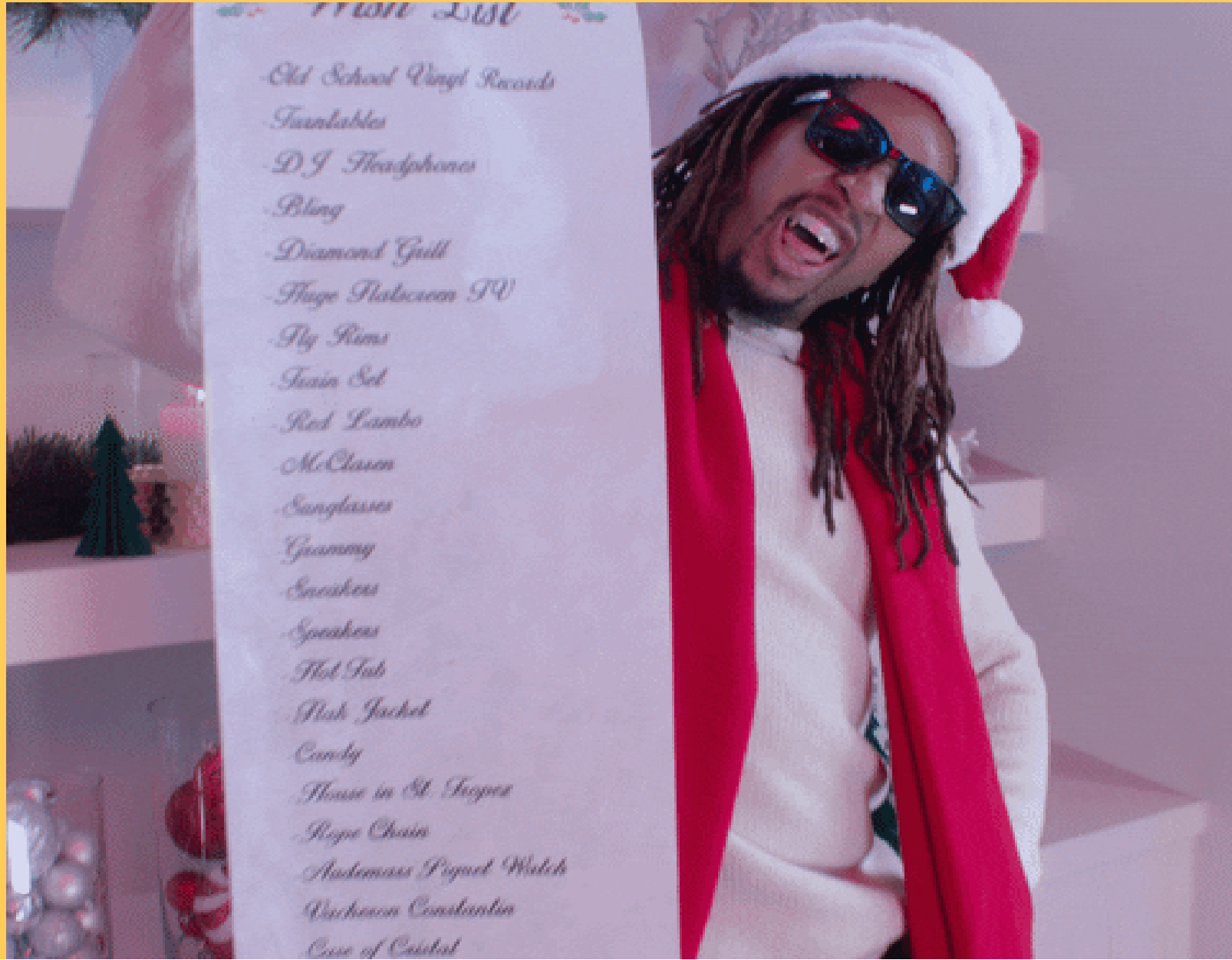
```
export function Title(props) {  
  const { tag } = props;  
  
  return (  
    < >  
      {(tag === 'h1')  
        ? <h1>{props.children}</h1>  
        : <p>{props.children}</p>  
      }  
    < />  
  );  
}
```

# УМОВНИЙ РЕНДЕРИНГ

- return null

```
export function Title(props) {  
  const { tag } = props;  
  
  return (  
    < >  
      {(tag === 'h1')  
        ? <h1>{props.children}</h1>  
        : {null}  
      }  
    < />  
  );  
}
```

# СПИСКИ ТА КЛЮЧІ



*Peace in Death*  
*Old School Vinyl Records*  
*Franklin*

# СПИСКИ

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

# СПИСКИ

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

# СПИСКИ

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



# СПИСКИ

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

# СПИСКИ

# СПИСКИ

- На вхід може бути масив елементів

# СПИСКИ

- На вхід може бути масив елементів
- Ключ є обов'язковим

# КЛЮЧИ

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

# КЛЮЧИ

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

# КЛЮЧИ

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

# КЛЮЧИ

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



КЛЮЧИ

# КЛЮЧІ

- Стабільна ідентичність

# КЛЮЧІ

- Стабільна ідентичність
- Ключ унікальний в межах списку

# КЛЮЧІ

- Стабільна ідентичність
- Ключ унікальний в межах списку
- Ключі є рядковими

# КЛЮЧІ

- Стабільна ідентичність
- Ключ унікальний в межах списку
- Ключі є рядковими
- Ключ застосовується для елементів масиву, а не до тегів

ЧИ ВИКОРИСТОВУВАТИ ІНДЕКС ЯК КЛЮЧ Є ГАРНОЮ  
ІДЕЄЮ?



# ЧИ ВИКОРИСТОВУВАТИ ІНДЕКС ЯК КЛЮЧ Є ГАРНОЮ ІДЕЄЮ?


<https://robinpokorny.medium.com/index-as-a-key-is-an-anti-pattern-e0349aeece318>





# ВІРТУАЛІЗОВАНИЙ СПИСОК

<https://bvaughn.github.io/react-virtualized/#/components/ArrowKeyStepper>



 **react virtualized** COMPONENTS WIZARD SOURCE DOCUMENTATION ISSUES

COLLECTION GRID LIST MASONRY TABLE

ARROWKEYSTEPPER AUTOSIZER CELLMEASURER COLUMNSIZER INFINITELoader MULTIGRID SCROLLSYNC

WINDOWSCROLLER

## ArrowKeyStepper [Source](#) | [Docs](#)

This high-order component decorates a `List`, `Table`, or `Grid` and responds to arrow-key events by scrolling one row or column at a time. Focus in the `Grid` below and use the left, right, up, or down arrow keys to move around within the grid.

Note that unlike the other HOCs in react-virtualized, the `ArrowKeyStepper` adds a `<div>` element around its children in order to attach a key-down event handler.

**mode:**  cells  edges (default)

Enable click selection? (resets selection)

Most-recently-stepped column: 0, row: 0

r:0, c:0	r:0, c:1	r:0, c:2	r:0, c:3	r:0, c:4	r:0, c:5	r:0, c:6
r:1, c:0	r:1, c:1	r:1, c:2	r:1, c:3	r:1, c:4	r:1, c:5	r:1, c:6
r:2, c:0	r:2, c:1	r:2, c:2	r:2, c:3	r:2, c:4	r:2, c:5	r:2, c:6

# ВІРТУАЛІЗОВАНИЙ СПИСОК

# ВІРТУАЛІЗОВАНИЙ СПИСОК

- MutationObserver

# ВІРТУАЛІЗОВАНИЙ СПИСОК

- MutationObserver
- Активна робота з VirtualDOM

# ВІРТУАЛІЗОВАНИЙ СПИСОК

- MutationObserver
- Активна робота з VirtualDOM
- Велике покращення швидкодії продукту

# БАНДЛІНГ

# БАНДЛІНГ

- Webpack

# БАНДЛІНГ

- Webpack
- Rollup



# БАНДЛІНГ

- Webpack
- Rollup
- Parcel

# БАНДЛІНГ

- Webpack
- Rollup
- Parcel
- Vite

# WEBPACK

# WEBPACK

- `npm run eject`

# WEBPACK

- `npm run eject`
- Створення єдиної точки входу

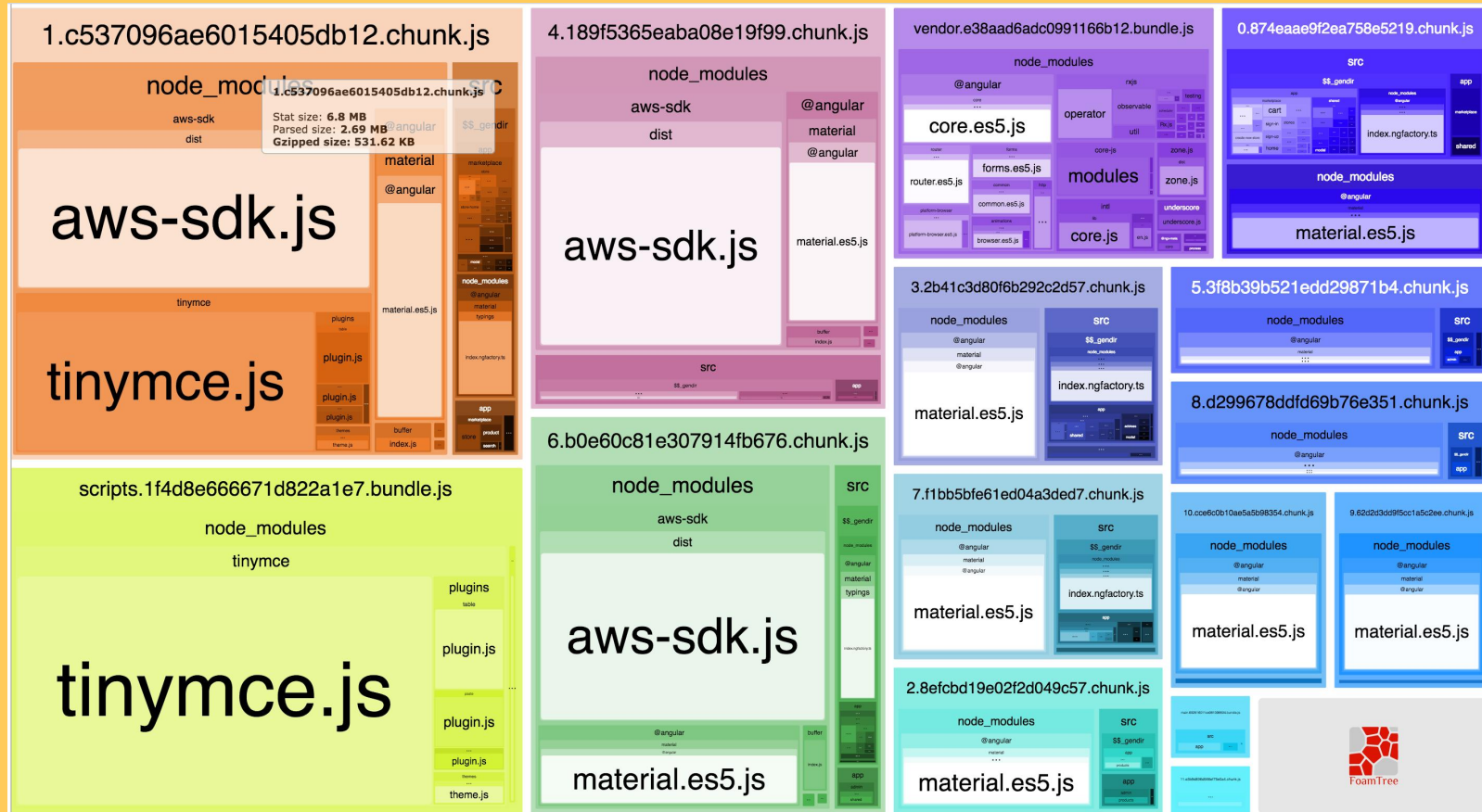
# WEBPACK

- npm run eject
- Створення єдиної точки входу
- Процес мініфікації, агліфікації та об'єднання

# WEBPACK

- `npm run eject`
- Створення єдиної точки входу
- Процес мініфікації, агліфікації та об'єднання
- `Webpack-bundle-analyzer`

# WEBPACK-BUNDLE-ANALYZER





*IMPORT*

# IMPORT

- Статичний імпорт

```
import { add } from './math';  
console.log(add(16, 26));
```

# IMPORT

- Динамічний імпорт

```
import("./math").then(math => {  
    console.log(math.add(16, 26));  
});
```

# IMPORT

- Динамічний імпорт babel

```
npm install babel-plugin-syntax-dynamic-import
```

*EXPORT*

# EXPORT

- Іменований експорт

```
export { myFunction };  
export const foo = Math.sqrt(2);
```

# EXPORT

- Дефолтный экспорт

```
export default function() {}
```

# EXPORT

- Дефолтний експорт єдиний на скрипт



# TREE SHAKING



# REACT LAZY



# WHO? WHO?

