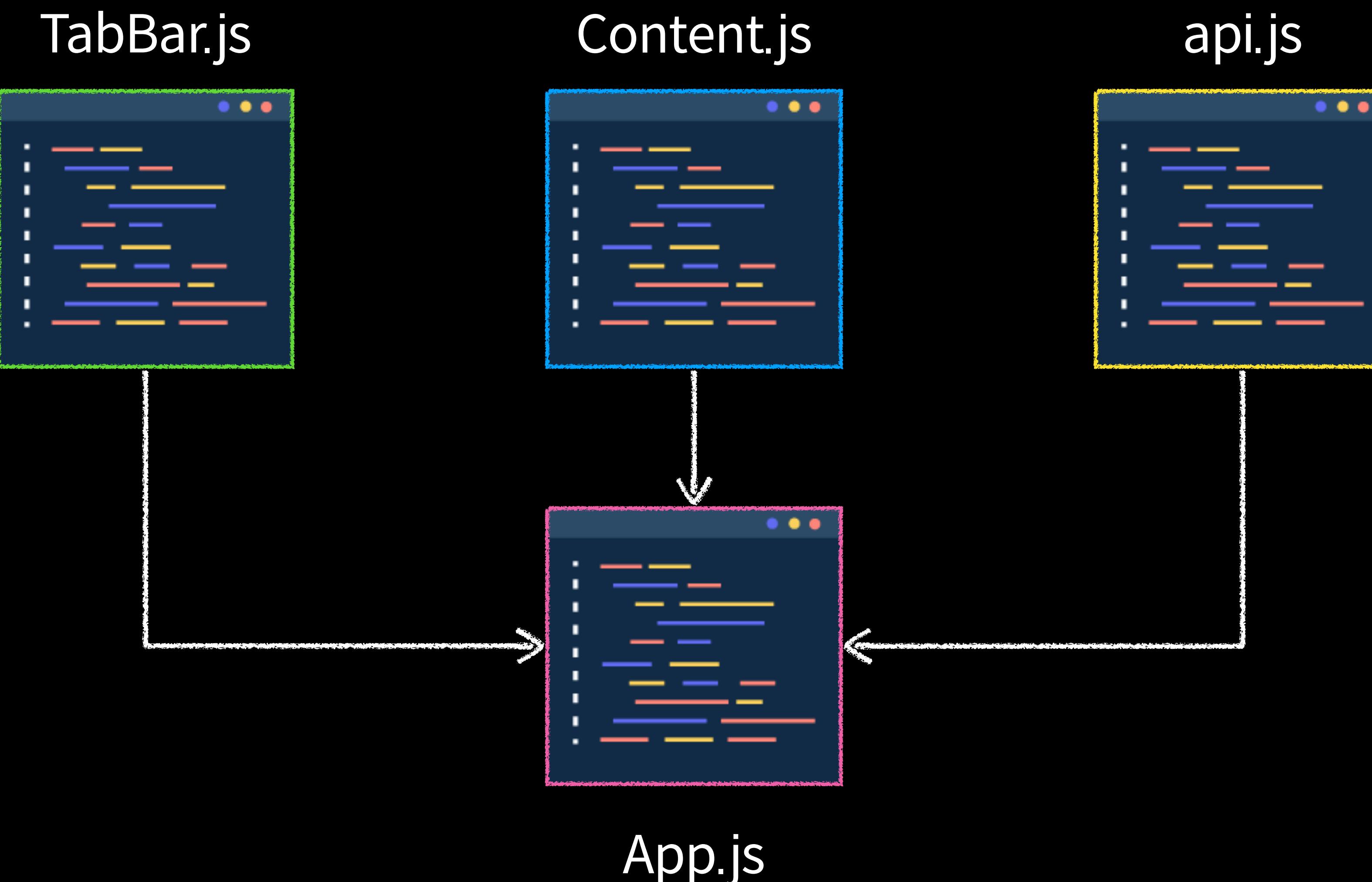


07. 상태 관리와 SPA

상태 관리

파일 구조



상태(State)

상태(State)란? 값이 변하는 데이터



Hyobin
FrontEnd Developer,
Educator

[GitHub](#) [LinkedIn](#) [Twitter](#)

© 2024

Dark Mode

No Results

자바스크립트 엔진은 자바스크립트 코드를 어떻게 실행시킬까?

Jan 13, 2024

자바스크립트 엔진은 자바스크립트 코드를 어떻게 실행시킬까? 자바스크립트의 실행 컨텍스트라는 개념을 알고 있다면, 이를 이해할 수 있다. 실행 컨텍스트에 대해 알아보자.

자바스크립트 실행컨텍스트

상태(State)

상태(State)란?
값이 변하는 데이터

The screenshot shows a dark-themed web application interface. At the top right is a search bar with a magnifying glass icon and the placeholder "No Results". On the left, there's a circular profile picture of a person with long dark hair, labeled "Hyobin" below it, and the text "FrontEnd Developer, Educator". Below this is a row of social media icons for GitHub, LinkedIn, and Twitter. At the bottom left is a copyright notice "© 2024" and a "Dark Mode" toggle switch. The main content area features a yellow-bordered box containing an article titled "자바스크립트 엔진은 자바스크립트 코드를 어떻게 실행시킬까?" with a date "Jan 13, 2024", two tags ("자바스크립트" and "실행컨텍스트"), and a brief description.

상태(State)란?
값이 변하는 데이터

Hyobin

FrontEnd Developer, Educator

자바스크립트 엔진은 자바스크립트 코드를 어떻게 실행시킬까?

Jan 13, 2024

자바스크립트 실행컨텍스트

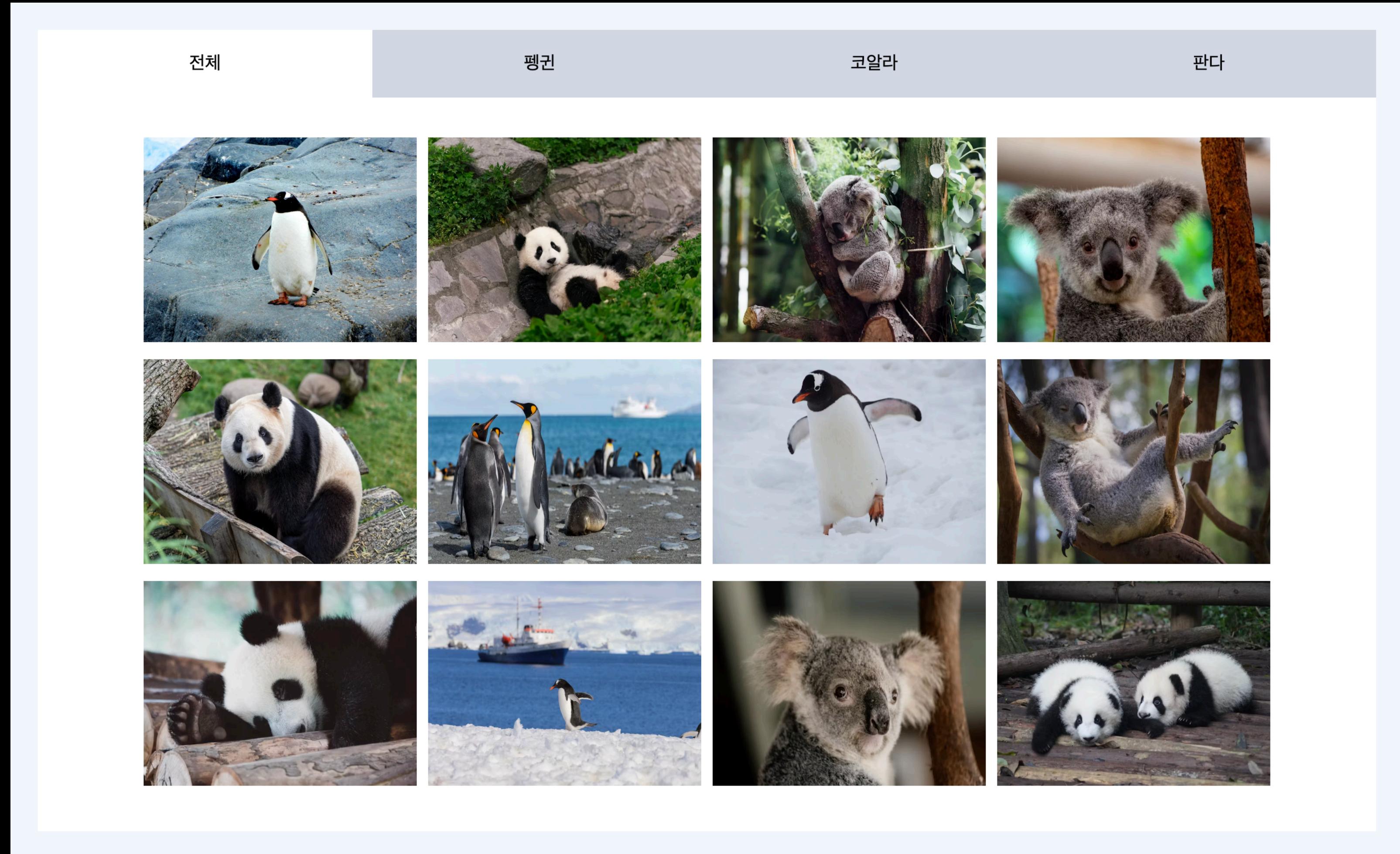
자바스크립트 엔진은 자바스크립트 코드를 어떻게 실행시킬까? 자바스크립트의 실행 컨텍스트라는 개념을 알고 있다면, 이를 이해할 수 있다. 실행 컨텍스트에 대해 알아보자.

© 2024

Dark Mode

상태(State)

상태(State)란?



상태(State)

상태(State)란?

전체

펭귄

코알라

판다

The collage illustrates the concept of 'state' by showing different animals in various situations or 'states'. The images include a penguin standing on rocks, two pandas resting, a koala climbing a tree, a giant panda sitting on a log, a group of penguins on a beach, a penguin walking on ice, a koala hanging from a branch, a giant panda eating, a penguin on a snowy slope, a close-up of a koala's face, and two giant pandas lying down.

상태 관리

상태 관리란?

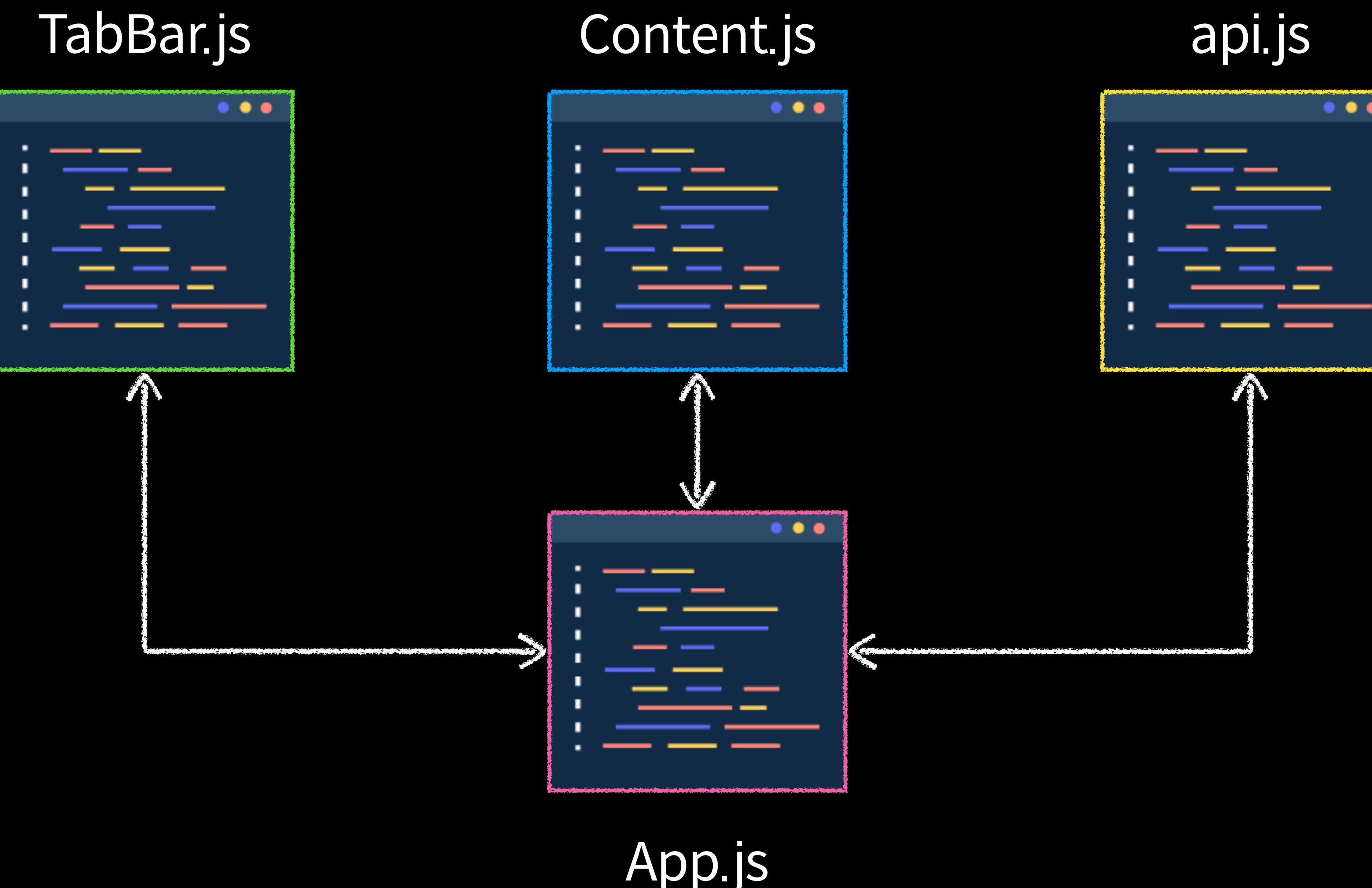
데이터의 흐름을 편리하게 관리

상태를 효율적으로 변경

불필요한 리렌더링 발생 X

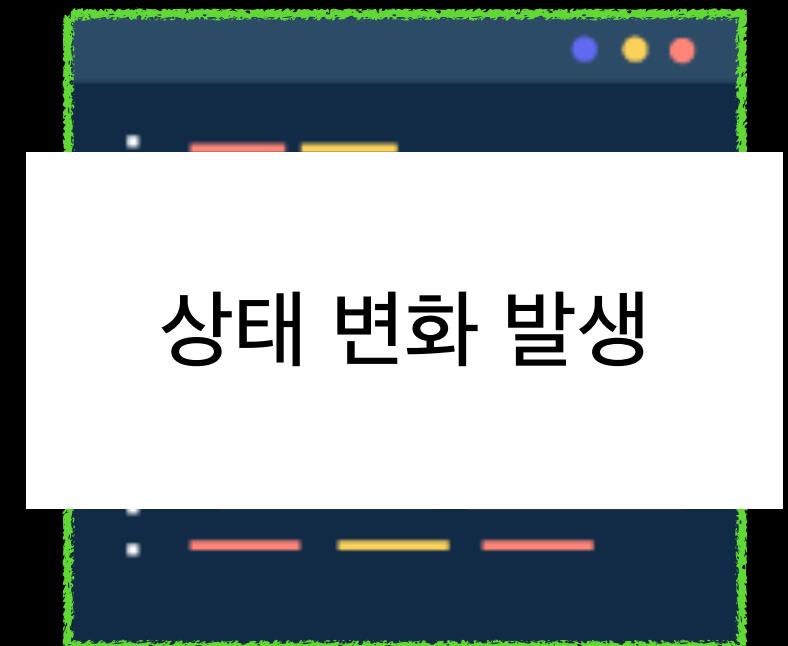
사용자에게 좋은 경험 제공

파일 구조

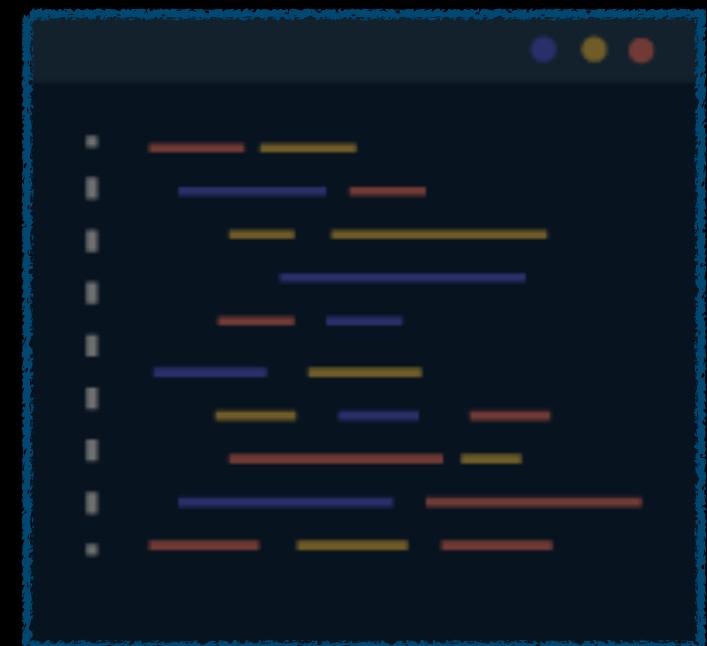


파일 구조

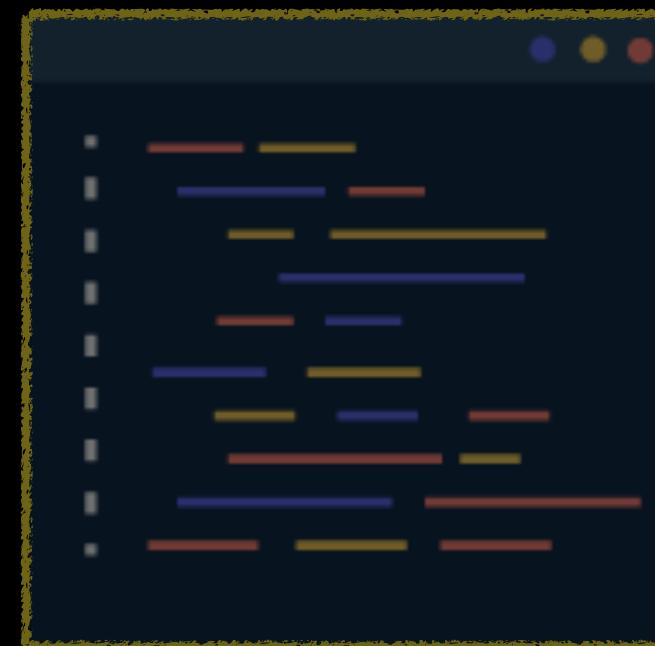
TabBar.js



Content.js



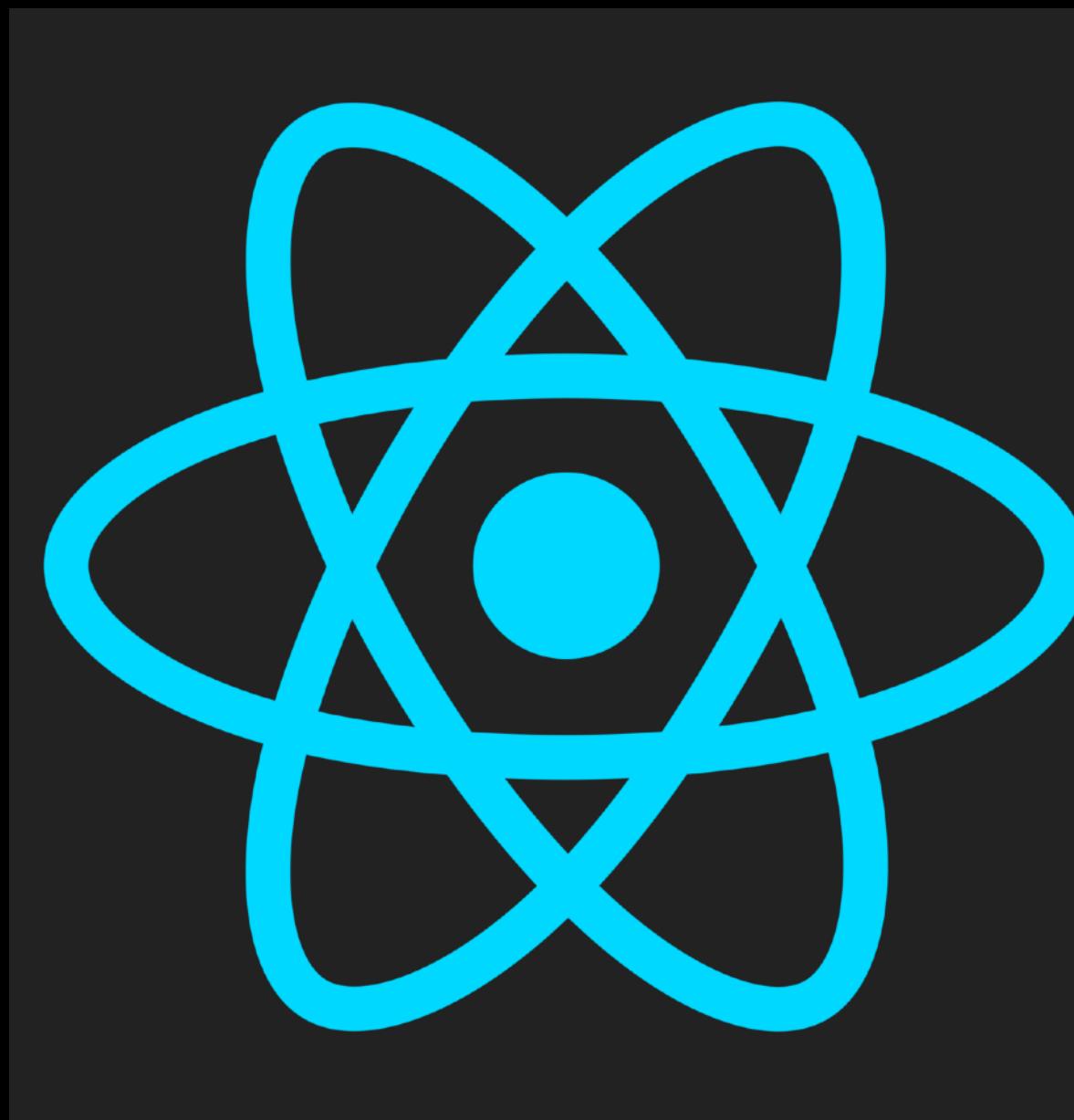
api.js



App.js

상태 관리를 사용하는 기술

React.js - redux, hooks



The screenshot shows the React Redux website and a React documentation page side-by-side.

React Redux Website:

- Header: React Redux, Getting Started, Tutorial, Using React Redux, API, GitHub, Need help?, Search.
- Main Content: React Redux logo, "Official React bindings for Redux", and a "Get Started" button.

React Documentation Page:

- Header: React, 문서 (Documentation), 자습서 (Tutorials), 블로그 (Blog), 커뮤니티 (Community), 검색 (Search).
- Text: "온 특징들을 사용할 수 있습니다." (You can use all these features).
- Text: "Hook 소개에서 아래 예시를 통해 Hook과 친해졌습니다." (You can get familiar with Hooks through the example below).
- Code Example (in a dark box):

```
import React, { useState } from 'react';

function Example() {
  // 새로운 state 변수를 선언하고, count라 부르겠습니다.
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```
- Text at the bottom: "아래의 클래스 예시와 비교하며 Hook의 특징에 대해 배울 예정입니다." (We will learn about the characteristics of Hooks by comparing them with class examples.)

상태 관리 적용

기본적인 틀

```
this.state = {  
    // 초기 상태 값들  
};  
  
this.setState = (nextState) => {  
    this.state = nextState;  
    // 상태 업데이트 후 렌더링  
    this.render();  
};  
  
this.render() {  
    // UI 렌더링 로직  
}
```