react h5 sight instructions

react h5 sight instructions provide essential guidance for developers and users working with the React H5 Sight component, a popular tool for integrating HTML5 video and multimedia capabilities within React applications. This article explores detailed instructions on how to effectively implement and optimize React H5 Sight in various projects. It covers fundamental setup steps, configuration options, advanced usage tips, and troubleshooting techniques to ensure smooth video playback and user interaction. Additionally, the article addresses compatibility considerations and performance optimizations relevant to different devices and browsers. Understanding these instructions is crucial for developers aiming to enhance user experience and maximize the potential of React H5 Sight in their web applications. The following sections will offer a structured and comprehensive overview to facilitate mastery of this technology.

Understanding React H5 Sight

React H5 Sight is a React-based component designed to simplify the integration of HTML5 video and multimedia features into web applications. It leverages modern React hooks and context to provide a seamless interface for controlling video playback, managing events, and customizing the user interface. This component supports a wide range of video formats and streaming protocols, making it versatile for different use cases such as educational platforms, entertainment sites, and interactive media applications. By utilizing React H5 Sight, developers can avoid dealing with raw HTML5 video APIs directly and instead benefit from a declarative, component-driven approach aligned with React’s architecture.

Core Features of React H5 Sight

The core features of React H5 Sight include:

- Responsive video player with adaptive sizing
- Support for multiple video sources and formats
- Customizable controls including play, pause, seek, and volume

- Event handling for playback state changes
- Integration with React state and lifecycle methods
- Accessibility compliance with keyboard navigation and ARIA attributes

Use Cases and Benefits

React H5 Sight serves various use cases such as embedding instructional videos, creating interactive tutorials, and delivering streaming content. Its benefits include improved developer productivity through reusable components, enhanced user experience via optimized playback controls, and compatibility across modern browsers and devices. These advantages make it a preferred choice for React developers looking to implement robust multimedia solutions.

Installation and Setup

Setting up React H5 Sight involves installing the component package, configuring the project environment, and incorporating the component into the React application. This section outlines the step-by-step process to ensure a successful installation and initial setup.

Prerequisites

Before installing React H5 Sight, ensure the development environment meets the following prerequisites:

- Node.js and npm installed (recommended latest stable versions)
- React version 16.8 or higher to support hooks
- Basic understanding of React component structure

Installation Steps

Follow these steps to install React H5 Sight:

1. Open the terminal in the project directory.
2. Run the command *npm install react-h5-sight* to add the package.
3. Import React H5 Sight into the desired component file using *import ReactH5Sight from 'react-h5-sight';*
4. Integrate the component within the JSX markup, supplying necessary props such as video source URLs.

Basic Usage Example

After installation, a basic implementation of React H5 Sight might look like this:

```
<ReactH5Sight
src="https://example.com/video.mp4"
controls
width="640"
height="360"
/>
```

This example demonstrates a simple video player with controls enabled and fixed dimensions.

Configuration Options

React H5 Sight offers a variety of configuration options that allow developers to tailor the video player’s behavior and appearance according to specific needs. Understanding these options is crucial for optimizing user experience.

Common Props and Their Usage

The key props available in React H5 Sight include:

- **src**: Specifies the video source URL; supports multiple formats for fallback.
- **controls**: Enables the default video control interface.
- **autoplay**: Allows the video to start playing automatically when loaded.
- **loop**: Enables continuous playback looping.
- **muted**: Starts the video muted, often used in autoplay scenarios.
- **width** and **height**: Define the player’s display dimensions.
- **poster**: Displays an image placeholder before the video plays.

Customizing Controls and UI

Beyond default controls, React H5 Sight supports custom control implementation through render props or component overrides. This flexibility allows developers to design unique user interfaces that align with branding or functionality requirements. Examples include custom play/pause buttons, progress bars, and volume sliders.

Advanced Usage Techniques

For complex applications, advanced techniques can enhance React H5 Sight's capabilities. These include event management, integration with state management libraries, and dynamic source switching.

Handling Playback Events

React H5 Sight exposes event callbacks such as *onPlay*, *onPause*, *onEnded*, and *onTimeUpdate*. Managing these events enables synchronization with other components, analytics tracking, and user interaction feedback.

Dynamic Source Management

Applications that require switching video sources dynamically can leverage React's state and props to update the *src* attribute of the React H5 Sight component. This approach supports playlists, adaptive streaming, or user-driven content selection.

Integration with Redux or Context API

For stateful control over playback and user preferences, React H5 Sight can be integrated with Redux or React's Context API. This allows centralized management of video states such as play status, volume level, and playback position across multiple components.

Troubleshooting Common Issues

Despite its robustness, users may encounter common problems when working with React H5 Sight. Identifying and resolving these issues ensures smooth functionality.

Video Playback Failures

Playback failures often stem from unsupported formats, incorrect source URLs, or browser restrictions. Verifying video compatibility and ensuring correct MIME types can mitigate these issues.

Autoplay Restrictions

Many browsers restrict autoplay of videos with sound. To comply, use the *muted* prop or require user interaction before playback starts.

Responsive Design Problems

Improper sizing or scaling can cause display issues. Employ responsive CSS techniques and configure *width* and *height* props effectively to maintain layout consistency.

Performance Optimization

Optimizing React H5 Sight ensures efficient resource usage and a smooth user experience, especially on mobile devices or slow networks.

Lazy Loading Videos

Implement lazy loading to defer video loading until the player is in view. This reduces initial page load time and bandwidth consumption.

Using Adaptive Streaming

React H5 Sight supports adaptive streaming protocols like HLS or DASH, which adjust video quality based on network conditions for uninterrupted playback.

Minimizing Re-renders

Optimize React component updates by memoizing the React H5 Sight component or managing props carefully to prevent unnecessary re-renders that can degrade performance.

Frequently Asked Questions

What is React H5 Sight and how is it used?

React H5 Sight is a React-based library or tool designed for creating interactive HTML5 (H5) sight or visual instructions, often used in e-learning or guided tutorials. It allows developers to build step-by-step visual instructions or walkthroughs within React applications.

How do I integrate React H5 Sight into my React project?

To integrate React H5 Sight, you typically install it via npm or yarn, import the necessary components into your React project, and then configure the instruction steps as per the documentation. This involves defining the visual elements, steps, and interactions for your sight instructions.

Can React H5 Sight handle multimedia elements like images and videos in instructions?

Yes, React H5 Sight supports multimedia elements such as images, videos, and animations within the instruction steps, enabling richer and more engaging guidance for users.

Is React H5 Sight mobile-friendly and responsive?

React H5 Sight is designed to be responsive and mobile-friendly, leveraging React and HTML5 capabilities to ensure that sight instructions render well

across different devices and screen sizes.

How do I customize the style and appearance of React H5 Sight instructions?

You can customize the styles of React H5 Sight instructions by overriding default CSS classes or using styled-components within React. The library often provides props or configuration options to adjust colors, fonts, sizes, and layout to match your application's design.

Are there any alternatives to React H5 Sight for creating interactive HTML5 instructions?

Yes, alternatives include libraries like React Joyride, Shepherd.js (with React wrappers), and Intro.js. These tools also provide ways to create interactive guides and visual instructions in React applications, each with unique features and customization options.

Additional Resources

1. *Mastering React H5 Sight Instructions: A Comprehensive Guide*
This book offers an in-depth exploration of React H5 sight instructions, focusing on how to efficiently implement and optimize sight-based user interfaces. It covers best practices, common pitfalls, and performance tuning to help developers create intuitive and responsive applications. With practical examples and case studies, readers gain hands-on experience in building accessible and user-friendly React components.

2. *React H5 Sight Instructions for Beginners*
Designed for newcomers, this book breaks down the fundamentals of React H5 sight instructions into easy-to-understand concepts. Readers will learn how to integrate sight instruction elements into their React projects step-by-step. The book includes simple projects and exercises to reinforce learning and build confidence in using React's H5 capabilities.

3. *Advanced Techniques in React H5 Sight Instructions*
Targeted at experienced React developers, this book dives into advanced strategies for managing and customizing sight instructions in React H5 environments. It explores dynamic rendering, state management, and accessibility enhancements to create powerful interactive applications. Readers will find tips on debugging and optimizing sight instruction workflows for better performance.

4. *React and H5 Sight: Building Accessible Interfaces*
This book emphasizes creating accessible applications using React and H5 sight instructions, focusing on compliance with accessibility standards. It guides developers through designing inclusive user experiences that cater to users with visual impairments. Practical advice on ARIA roles, keyboard navigation, and screen reader compatibility is provided to ensure usability for all.

5. *Hands-On React H5 Sight Instruction Projects*
Filled with real-world projects, this book encourages learning through doing by guiding readers through building various applications utilizing React H5 sight instructions. Each project highlights different features and challenges, fostering problem-solving skills. Readers will develop a solid

portfolio of sight-based React applications by the end of the book.

6. *Optimizing Performance in React H5 Sight Applications*
Focusing on performance optimization, this book teaches developers how to streamline React H5 sight instruction implementations for faster load times and smoother interactions. Topics include code splitting, lazy loading, and efficient state updates to minimize rendering overhead. The book also covers profiling tools and techniques to identify and fix bottlenecks in sight instruction components.

7. *React H5 Sight Instruction Patterns and Practices*
This book presents a collection of design patterns and best practices for managing sight instructions within React H5 applications. It covers component composition, reusable hooks, and context management to build scalable and maintainable codebases. Readers will learn how to create consistent and predictable sight instruction behaviors across their projects.

8. *Integrating React H5 Sight Instructions with Backend Services*
Exploring the integration of React H5 sight instructions with various backend APIs and services, this book shows how to build dynamic and data-driven sight interfaces. It includes examples using RESTful APIs, WebSockets, and GraphQL to fetch and update sight instruction data in real time. The book also addresses authentication, error handling, and state synchronization challenges.

9. *The Future of React H5 Sight Instruction Development*
This forward-looking book discusses emerging trends and technologies shaping the future of React H5 sight instruction development. Topics include AI-powered sight instruction enhancements, augmented reality integration, and next-generation accessibility tools. Readers will gain insights into how to prepare their React applications for evolving user needs and technological advancements.

React H5 Sight Instructions

react h5 sight instructions: *Web Development with Django* Ben Shaw, Saurabh Badhwar, Chris Guest, Bharath Chandra K S, 2023-05-26 Create your own websites easily, securely, and quickly with Django by tackling practical activities based on realistic case studies Key Features Understand Django functionality and the Model-View-Template (MVT) paradigm Create and iteratively build a book review website, adding features as you build your knowledge Explore advanced concepts such as REST API implementation and third-party module integration Book DescriptionDo you want to develop reliable and secure applications that stand out from the crowd without spending hours on boilerplate code? You've made the right choice trusting the Django framework, and this book will tell you why. Often referred to as a "batteries included" web development framework, Django comes with all the core features needed to build a standalone application. Web Development with Django will take you through all the essential concepts and help

you explore its power to build real-world applications using Python. Throughout the book, you'll get the grips with the major features of Django by building a website called Bookr – a repository for book reviews. This end-to-end case study is split into a series of bitesize projects presented as exercises and activities, allowing you to challenge yourself in an enjoyable and attainable way. As you advance, you'll acquire various practical skills, including how to serve static files to add CSS, JavaScript, and images to your application, how to implement forms to accept user input, and how to manage sessions to ensure a reliable user experience. You'll cover everyday tasks that are part of the development cycle of a real-world web application. By the end of this Django book, you'll have the skills and confidence to creatively develop and deploy your own projects.What you will learn Create a new application and add models to describe your data Use views and templates to control behavior and appearance Implement access control through authentication and permissions Develop practical web forms to add features such as file uploads Build a RESTful API and JavaScript code that communicates with it Connect to a database such as PostgreSQL Who this book is for This book is for programmers looking to enhance their web development skills using the Django framework. To fully understand the concepts explained in this book, basic knowledge of Python programming as well as familiarity with JavaScript, HTML, and CSS is assumed.

react h5 sight instructions: Specifications and Drawings of Patents Issued from the United States Patent Office United States. Patent Office, 1897

react h5 sight instructions: Animals Alive! Walter Dennis Holley, 1997 A teacher's guide and resource book for designing and conducting live animal activities that are non-invasive and observation-oriented.

react h5 sight instructions: Asian Discourses of Rule of Law Randall P. Peerenboom, 2004 Rule of law, one of the pillars of the modern world, has emerged in Western liberal democracies. This book considers how rule of law is viewed and implemented in the different cultural, economic and political context of Asia.

react h5 sight instructions: Films and Other Materials for Projection Library of Congress, 1974

react h5 sight instructions: A dictionary of chemistry Henry Watts, 1863

react h5 sight instructions: Audiovisual Materials , 1980

react h5 sight instructions: *A Dictionary of Chemistry and the Allied Branches of Other Sciences* Henry Watts, 1866

react h5 sight instructions: *Audiovisual Materials* Library of Congress, 1980

react h5 sight instructions: A Dictionary of Chemistry and Allied Branches of Other Sciences Henry Watts (F.C.S.), 1863

react h5 sight instructions: Economic Papers , 1997

react h5 sight instructions: Library of Congress Catalogs Library of Congress, 1973

react h5 sight instructions: Focus on physical science Charles H. Heimler, 1989

react h5 sight instructions: The New Yorker , 1979

react h5 sight instructions: JMR, Journal of Marketing Research , 1973

react h5 sight instructions: UCLA Pacific Basin Law Journal , 2002

react h5 sight instructions: Conference Papers Index , 1982 Monthly. Papers presented at recent meeting held all over the world by scientific, technical, engineering and medical groups. Sources are meeting programs and abstract publications, as well as questionnaires. Arranged under 17 subject sections, 7 of direct interest to the life scientist. Full programs of meetings listed under sections. Entry gives citation number, paper title, name, mailing address, and any ordering number assigned. Quarterly and annual indexes to subjects, authors, and programs (not available in monthly issues).

react h5 sight instructions: New Statesman , 1963

Related to react h5 sight instructions

写半个月**react**了，感觉还是**vue**好用。有没有大佬能说一下**react**好 至于水平么，怎么讲呢～这么说吧，在需求明确的前提下，无论是 React 还是 Vue 我都可以流畅开发，不存在什么不适应或者不能解决的问题（求助他人也是自身能力嘛～）。先说结论吧——

开发跨平台 **App** 推荐 **React Native** 还是 **Flutter**？ - 知乎 不论是 React Native 还是 Flutter 都可以用于开发跨平台应用，选哪个看你自身需求，但首先得了解清楚两者都是什么。 React Native 能够跨平台是因为它允许你使用 React/JavaScript/JSX

国内 **vue** 这么火，为什么大厂都是用 **react** 居多**? -** 知乎 我是看不明白，为啥有人拿“vue需要记很多api，学习成本高，react的api少学习成本低”来作为react更好用的依据？是的，react官方对自己的定位就是个view层的库，使用的时候记

为什么尤雨溪说**react**的性能不如**vue**？ - 知乎 先讲结论，React Forget救不了React，React再不转向以subscription（订阅式响应）为主的reactivity响应，还固守comparison（比较式响应），迟早会被Solidjs、Svelte甚至是Vue干掉

react有什么用？优点和缺点有哪些？ - 知乎 喜欢React的人很多，但是喜欢它的原因都不太一样 比较现实的优点：能够实现服务器端的渲染，便于搜索引擎优化。这一点要比Backbone, Angular 1.x和Ember早期强 能够很好的和现有

react脚手架的对比选择**? -** 知乎 react生态中有着众多的react脚手架，如服务端渲染比较有名的next.js,remix。静态渲染的Gatsby。客户端渲

有人说若依框架代码质量不行，那有其他类似框架的推荐吗？ - 知乎 若依只是一个简单的脚手架项目，其中的技术目前已经相对比较陈旧。学习框架设计和新的设计思想可以参考今年3月份刚开源的新一代低代码平台：Nop平台。 Nop平台与其

有哪些开源 **React** 前端项目推荐？ - 知乎 react-bilibili 和 pilipala 这两个开源项目，可以说高度还原了 BiliBili 客户端，实现了大部分 B 站客户端的功能。如果对其中的源码感兴趣，快去下载安装体验吧～ 大家的点赞、收藏和评论都是对

为什么很多大厂优先考虑使用 **React** 而不采用 **Vue**？ - 知乎 老弟，哥今年搞前端第9个年头，马上“前端九年义务教育”就要毕业，负责任的告诉你，react那可是前端赚大钱的标配。今天我就给你念叨念叨，为什么前端不能放弃React？ 1、从市场流行度

选择**vue3+TS**还是**react**？ - 知乎 React 本身不内置状态管理方案，依赖生态中的第三方库，复杂场景可选择 Redux, Zustand（轻量灵活）、Recoil（专注原子化状态）等。 Vue 在这一块官方提供了完

写半个月**react**了，感觉还是**vue**好用。有没有大佬能说一下**react**好 至于水平么，怎么讲呢～这么说吧，在需求明确的前提下，无论是 React 还是 Vue 我都可以流畅开发，不存在什么不适应或者不能解决的问题（求助他人也是自身能力嘛～）。先说结论吧——

开发跨平台 **App** 推荐 **React Native** 还是 **Flutter**？ - 知乎 不论是 React Native 还是 Flutter 都可以用于开发跨平台应用，选哪个看你自身需求，但首先得了解清楚两者都是什么。 React Native 能够跨平台是因为它允许你使用 React/JavaScript/JSX

国内 **vue** 这么火，为什么大厂都是用 **react** 居多**? -** 知乎 我是看不明白，为啥有人拿“vue需要记很多api，学习成本高，react的api少学习成本低”来作为react更好用的依据？是的，react官方对自己的定位就是个view层的库，使用的时候记

为什么尤雨溪说**react**的性能不如**vue**？ - 知乎 先讲结论，React Forget救不了React，React再不转向以subscription（订阅式响应）为主的reactivity响应，还固守comparison（比较式响应），迟早会被Solidjs、Svelte甚至是Vue干掉

react有什么用？优点和缺点有哪些？ - 知乎 喜欢React的人很多，但是喜欢它的原因都不太一样 比较现实的优点：能够实现服务器端的渲染，便于搜索引擎优化。这一点要比Backbone, Angular 1.x和Ember早期强 能够很好的和现有

react脚手架的对比选择**? -** 知乎 react生态中有着众多的react脚手架，如服务端渲染比较有名的next.js,remix。静态渲染的Gatsby。客户端渲

有人说若依框架代码质量不行，那有其他类似框架的推荐吗？ - 知乎 若依只是一个简单的脚手架项目，其中的技术目前已经相对比较陈旧。学习框架设计和新的设计思想可以参考今年3月份刚开源的新一代低代码平台：Nop平台。 Nop平台与其

有哪些开源 **React** 前端项目推荐？ - 知乎 react-bilibili 和 pilipala 这两个开源项目，可以说高度还原了 BiliBili 客户端，实现了大部分 B 站客户端的功能。如果对其中的源码感兴趣，快去下载安装体验吧～ 大家的点赞、收藏和评论都是对

为什么很多大厂优先考虑使用 **React** 而不采用 **Vue**？ - 知乎 老弟，哥今年搞前端第9个年头，马上“前端九年义务教育”就要毕业，负责任的告诉你，react那可是前端赚大钱的标配。今天我就给你念叨念叨，为什么前端不能放弃React？ 1、从市场流行度

选择**vue3+TS**还是**react**？ - 知乎 React 本身不内置状态管理方案，依赖生态中的第三方库，复杂场景可选择 Redux, Zustand（轻量灵活）、Recoil（专注原子化状态）等。 Vue 在这一块官方提供了完

写半个月**react**了，感觉还是**vue**好用。有没有大佬能说一下**react**好 至于水平么，怎么讲呢～这么说吧，在需求明确的前提下，无论是 React 还是 Vue 我都可以流畅开发，不存在什么不适应或者不能解决的问题（求助他人也是自身能力嘛～）。先说结论吧——

开发跨平台 **App** 推荐 **React Native** 还是 **Flutter**？ - 知乎 不论是 React Native 还是 Flutter 都可以用于开发跨平台应用，选哪个看你自身需求，但首先得了解清楚两者都是什么。 React Native 能够跨平台是因为它允许你使用 React/JavaScript/JSX

国内 **vue** 这么火，为什么大厂都是用 **react** 居多**? -** 知乎 我是看不明白，为啥有人拿“vue需要记很多api，学习成本高，react的api少学习成本低”来作为react更好用的依据？是的，react官方对自己的定位就是个view层的库，使用的时候记

为什么尤雨溪说**react**的性能不如**vue**？ - 知乎 先讲结论，React Forget救不了React，React再不转向以subscription（订阅式响应）为主的reactivity响应，还固守comparison（比较式响应），迟早会被Solidjs、Svelte甚至是Vue干掉

react有什么用？优点和缺点有哪些？ - 知乎 喜欢React的人很多，但是喜欢它的原因都不太一样 比较现实的优点：能够实现服务器端的渲染，便于搜索引擎优化。这一点要比Backbone, Angular 1.x和Ember早期强 能够很好的和现有的

react脚手架的对比选择**? -** 知乎 react生态中有着众多的react脚手架，如服务端渲染比较有名的next.js,remix。静态渲染的Gatsby。客户端渲

□□ **-** □□ □□□3□□ □□□□□□□□□□□□□□□□□□Nop□□□ Nop□□□□

□□□□□ **React** □□□□□□□□ **-** □□ react-bilibili □ pilipala □□□□□□□□□□□□□□□□□□□□□□ BiliBili □□□□□□□□□□□□ B □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□~ □□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ **React** □□□□ **Vue**□ **-** □□ □□□□□□□□□□□□□9□□□□□□“□□□□□□□□□□”□□□□□□□□□□□□□□□□□react□□□□□□□□□ □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□React□ 1□ □□□□□□

□□**vue3+TS**□□**react**□ **-** □□ React □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Redux□Zustand□□□□□□□□Recoil □□□□□□□□□□□□□□□ Vue □□□□□□□□□□□□□

□□□□**react**□□□□□□**vue**□□□□□□□□□□□□□□**react**□ □□□□□□□□□□□□~□□□□□□□□□□□□□□□□□□□□□□□□ React □□ Vue □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□~□□ □□□□□——

□□□□□ **App** □□ **React Native** □□ **Flutter**□ **-** □□ □□□ React Native □□ Flutter □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ React Native □□□□□□□□□□□□□□□□□□□ React/JavaScript/JSX

□□ **vue** □□□□□□□□□□□□□□□□ **react** □□**?** **-** □□ □□□□□□□□□□□□□□□□□“vue□□□□□api□□□□□□□□react□api□□□□□□”□□□react□ □□□□□□ □□□react□□□□□□□□□□□□□□view□□□□□□□□□□□□

□□□□□□□**react**□□□□□**vue**□ **-** □□ □□□□□React Forget□□□React□React□□□□□subscription□□□□□□□□□□□ □reactivity□□□□□□comparison□□□□□□□□□□□□□□□Solidjs□Svelte□□□Vue□□

react□□□□□□□□□□□□□□□□□□ **-** □□ □□React□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □Backbone, Angular 1.x□Ember□□□ □□□□□□□□□

react□□□□□□□□□**?** **-** □□ react□□□□□□□□□react□□□□□□□□□□□□□□□□□□□next.js,remix□□□□□□□Gatsby□□□□□

□□ **-** □□ □□□3□□ □□□□□□□□□□□□□□□□□□Nop□□□ Nop□□□□

□□□□□ **React** □□□□□□□□ **-** □□ react-bilibili □ pilipala □□□□□□□□□□□□□□□□□□□□□□ BiliBili □□□□□□□□□□□□ B □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□~ □□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ **React** □□□□ **Vue**□ **-** □□ □□□□□□□□□□□□□9□□□□□□“□□□□□□□□□□”□□□□□□□□□□□□□□□□□react□□□□□□□□□ □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□React□ 1□ □□□□□□

□□**vue3+TS**□□**react**□ **-** □□ React □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Redux□Zustand□□□□□□□□Recoil □□□□□□□□□□□□□□□ Vue □□□□□□□□□□□□□

□□□□**react**□□□□□□**vue**□□□□□□□□□□□□□□**react**□ □□□□□□□□□□□□~□□□□□□□□□□□□□□□□□□□□□□□□ React □□ Vue □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□~□□ □□□□□——

□□□□□ **App** □□ **React Native** □□ **Flutter**□ **-** □□ □□□ React Native □□ Flutter □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ React Native □□□□□□□□□□□□□□□□□□□ React/JavaScript/JSX

□□ **vue** □□□□□□□□□□□□□□□□ **react** □□**?** **-** □□ □□□□□□□□□□□□□□□□□“vue□□□□□api□□□□□□□□react□api□□□□□□”□□□react□ □□□□□□ □□□react□□□□□□□□□□□□□□view□□□□□□□□□□□□

□□□□□□□**react**□□□□□**vue**□ **-** □□ □□□□□React Forget□□□React□React□□□□□subscription□□□□□□□□□□□ □reactivity□□□□□□comparison□□□□□□□□□□□□□□□Solidjs□Svelte□□□Vue□□

react□□□□□□□□□□□□□□□□□□ **-** □□ □□React□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □Backbone, Angular 1.x□Ember□□□ □□□□□□□□□

react□□□□□□□□□**?** **-** □□ react□□□□□□□□□react□□□□□□□□□□□□□□□□□□□next.js,remix□□□□□□□Gatsby□□□□□

□□ **-** □□ □□□3□□ □□□□□□□□□□□□□□□□□□Nop□□□ Nop□□□□

□□□□□ **React** □□□□□□□□ **-** □□ react-bilibili □ pilipala □□□□□□□□□□□□□□□□□□□□□□ BiliBili □□□□□□□□□□□□ B □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□~ □□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ **React** □□□□ **Vue**□ **-** □□ □□□□□□□□□□□□□9□□□□□□“□□□□□□□□□□”□□□□□□□□□□□□□□□□□react□□□□□□□□□ □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□React□ 1□ □□□□□□

□□**vue3+TS**□□**react**□ **-** □□ React □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Redux□Zustand□□□□□□□□Recoil □□□□□□□□□□□□□□□ Vue □□□□□□□□□□□□□

□□□□**react**□□□□□□**vue**□□□□□□□□□□□□□□**react**□ □□□□□□□□□□□□~□□□□□□□□□□□□□□□□□□□□□□□□ React □□ Vue □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□~□□ □□□□□——

□□□□□ **App** □□ **React Native** □□ **Flutter**□ **-** □□ □□□ React Native □□ Flutter □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ React Native □□□□□□□□□□□□□□□□□□□ React/JavaScript/JSX

□□ **vue** □□□□□□□□□□□□□□□□ **react** □□**?** **-** □□ □□□□□□□□□□□□□□□□□“vue□□□□□api□□□□□□□□react□api□□□□□□”□□□react□

好用的依据？是的，react官方对自己的定位就是个view层的库，使用的时候记

为什么尤雨溪说**react**的性能不如**vue**？ - 知乎 先讲结论，React Forget救不了React，React再不转向以subscription（订阅式响应）为主的reactivity响应，还固守comparison（比较式响应），迟早会被Solidjs、Svelte甚至是Vue干掉

react有什么用？优点和缺点有哪些？ - 知乎 喜欢React的人很多，但是喜欢它的原因都不太一样 比较现实的优点：能够实现服务器端的渲染，便于搜索引擎优化。这一点要比Backbone, Angular 1.x和Ember早期强 能够很好的和现有

react脚手架的对比选择**？** - 知乎 react生态中有着众多的react脚手架，如服务端渲染比较有名的next.js,remix。静态渲染的Gatsby。客户端渲

有人说若依框架代码质量不行，那有其他类似框架的推荐吗？ - 知乎 若依只是一个简单的脚手架项目，其中的技术目前已经相对比较陈旧。学习框架设计和新的设计思想可以参考今年3月份刚开源的新一代低代码平台：Nop平台。 Nop平台与其

有哪些开源 **React** 前端项目推荐？ - 知乎 react-bilibili 和 pilipala 这两个开源项目，可以说高度还原了 BiliBili 客户端，实现了大部分 B 站客户端的功能。如果对其中的源码感兴趣，快去下载安装体验吧～ 大家的点赞、收藏和评论都是对

为什么很多大厂优先考虑使用 **React** 而不采用 **Vue**？ - 知乎 老弟，哥今年搞前端第9个年头，马上“前端九年义务教育”就要毕业，负责任的告诉你，react那可是前端赚大钱的标配。今天我就给你念叨念叨，为什么前端不能放弃React？ 1、从市场流行度

选择**vue3+TS**还是**react**？ - 知乎 React 本身不内置状态管理方案，依赖生态中的第三方库，复杂场景可选择 Redux，Zustand（轻量灵活）、Recoil（专注原子化状态）等。 Vue 在这一块官方提供了完

Back to Home: https://dev.littleadventures.com