Rien.js Project Proposal

What is Rien.js and what does it do?

Rien.js is a lightweight JavaScript Framework. It means nothing or naught in French, so when people ask about what technology they used in projects, they may answer “With Rien (With nothing)”, which would be quite interesting. Anyhow, the features of this framework deserve this name, I will show you why.

We all know that there are countless technologies and frameworks for front-end development nowadays, and the most famous and commonly used three would be React, Vue and Angular. These are all great frameworks, but their sizes are quite large — even the smallest, Vue, has 58kB. It will not be a problem for those complicated projects, undoubtedly, but for those relatively small projects, or those which require a great performance, the size of those frameworks in the runtime would be too large. Rien.js does not aim to build extremely complicated projects, nor replace the market share of the existed frameworks. It aims to quickly create relatively small-sized and reusable components with little codes in the runtime, which could collaborate with other frameworks and technologies well.

In order to fulfill features mentioned above, Rien.js would not use Virtual DOM. The diff and re-render algorithms of the Virtual DOM, though prevent inappropriate operations to manipulate native DOM, would be quite slow and resource-costing. Rien.js takes another approach to update DOM without finding the difference between the new and the old DOM tree, that is mapping the state-changing to the operation of the DOM node, and then updating the corresponding DOM node when the data is updated directly.

Who will use it and how will they use it?

Relatively experienced front-end developers may still apply the big three frameworks in their work, but when they build their personal projects, especially some small libraries or simple components, they might not want to use a sledgehammer to crack a nut. It is likely for them to choose Rien.js if they are interested in lightweight and elegant frameworks. With their experiences in other frameworks, it won’t be difficult for them to learn.

The developers with experiences only in Vanilla JS, HTML and CSS but no experiences in modern frameworks, would find Rien.js comes in handy since it has a shallow learning curve and similar syntax with Vanilla JS. They can take advantage of framework features without learning too many concepts.

(Potential) Developers who work in a huge project, which have multiple parts implemented by different frameworks will also find Rien.js useful. They sometimes would require reusable components for all the frameworks, but it is too complicated or not very efficient if implemented with native JS or JQuery, they can use Rien.js to create reusable components quickly.

All the developers, when using Rien.js, will not worry that re-rendering would affect the performance, nor how to import fewer libraries to make the bundle size smaller. All they need to do is to write simple and elegant codes in Vanilla JS style, and the rest would be nothing.

(Note: This project may require some external libraries if necessary, I would try my best to use as less as I can and reach the complexity requirement. I have emailed Mark about the whole idea.)