

ToiletPaper #161

Preact – A light-weight alternative to React

Author: Philipp Miller / Software Engineer / Automotive

✗ Problem

React is great, but it's also big. It tries to abstract the DOM as much as possible to save developers from browser-specific compatibility issues – even when those have already been straightened out by improved standards. For example, the events generated by the browser are not simply passed through in React but replaced by custom objects. This also affects performance and bundle size.

✓ Alternative

Preact [1] replicates React's API to a large extent – at first glance it is barely distinguishable from React. The usual hooks are available (plus a very convenient `useErrorBoundary` hook). Many 3rd-party libraries for React work out of the box with Preact (e.g. Material-UI [4]).

Which differences [2] are most important depends on the use case. Here are a few:

- Preact is written entirely in TypeScript [3] – the types themselves are often different from those in React. For example, instead of `ReactDOM` and `FunctionComponent<P>`, there are other types like `ComponentChildren` and `VNode<P>`.
- Event listeners receive browser-native event objects as parameters. This also means that they do not bubble through portals.
- Imports are structured slightly differently (see example).

➔ Example

```
import { Fragment, h, render } from "preact"
import { useState } from "preact/hooks"

const App = () => {
  const [count, setCount] = useState(0)
  const increment = () => setCount(n => n + 1)
  return (
    <>
      <h1>Clicks: {count}</h1>
      <button onClick={increment}>Add one</button>
    </>
  )
}

render(<App />, document.getElementById("app"))
```

+ Advantages

- Good documentation that helps with the migration
- Small. I built a minimal Hello World app to show that:

| React | Preact |
|-------------------------------|-----------------------------|
| 129.23 KiB gzip: 41.68 KiB | 11.08 KiB gzip: 4.54 KiB |

- Less DOM abstraction (only an issue if you need to support particularly old browsers)
- Implemented in modular TypeScript

Disadvantages

- The API is not completely identical (especially for the types). This may require a bit of relearning. Some features needed for more React compatibility (e.g. in 3rd-party libraries) are available via the `preact/compat` package.
- Possible compatibility issues with more complex libraries that access React internals. However, especially for the larger libraries, tutorials or adapters are often already available (e.g. `mobx-preact`).

[1] <https://preactjs.com>

[2] <https://preactjs.com/guide/v10/differences-to-react>

[3] <https://preactjs.com/guide/v10/typescript>

[4] <https://twitter.com/preactjs/status/1152267975078154240>