ToiletPaper #117

Was ist eigentlich JSX/TSX

Autor: Santo Pfingsten / Software Engineer / Standort Leipzig

✕ Problem

Jeder, der schon einmal mit React gearbeitet hat, wird die Dateiendung JSX (TSX bei TypeScript) und die HTML-ähnliche Syntax kennen. Doch wer weiß eigentlich, was genau sich dahinter verbirgt?

```
const profile = <div>
  <img src={u.avatar} className="profile" alt={`Pic of ${u.firstName}`} />
  <h3>{[u.firstName, u.lastName].join(' ')}</h3>
</div>;
```

✓ Lösung

JSX/TSX wird von einem Transpiler (z. B. `babel` oder `tsc`) zu reinem JavaScript-Code gewandelt. Das obige Beispiel würde zu folgendem JS kompiliert werden:

```
const profile = React.createElement("div", null,
  React.createElement("img", { src: u.avatar, className: "profile", alt: `Pic
of ${u.firstName}` }),
  React.createElement("h3", null, [u.firstName, u.lastName].join(" "))
);
```

- Die Funktion `React.createElement` erhält als ersten Parameter entweder einen Tagnamen oder eine Komponente (Funktions- oder Klassenkomponente).
  - Fängt der Tagname im JSX klein an, z. B. <`button`>, ist der Parameter ein String, andernfalls (z. B. <`Button`>) eine Funktion oder eine Klasse (welche so benannt im Scope vorhanden sein muss).
- Der zweite Parameter ist ein Objekt mit allen Properties, die dem Tag zugewiesen wurden. Falls keine Properties zugewiesen wurden, ist dieser Parameter null.
- Die restlichen Parameter sind die Kindelemente, welche im Tag eingefügt wurden.

➔ Beispiel

Man kann sich also ganz einfach einen Ersatz für React schreiben, welcher, anstelle eines Virtual DOM, echte HTML Elemente mit `document.createElement` erzeugt:

```
function h(tag, props, ...children) {
    if (typeof tag === "function") return tag({ ...props, children });
    const element = document.createElement(tag);
    if (props) setAllAttributes(element, props);
    applyChildren(element, children);
    return element;
}
```

Diese Funktion könnte dann anstelle von `React.createElement` aufgerufen werden. Dieses sehr simple Beispiel unterstützt sogar Funktionskomponenten.

+ Weiterführende Aspekte

- Babel: https://babeljs.io/docs/en/6.26.3/babel-plugin-transform-react-jsx
- TypeScript: https://www.typescriptlang.org/docs/handbook/jsx.html
- Das letzte Beispiel in TypeScript und etwas ausführlicher: https://github.com/Lusito/tsx-dom
- React ohne JSX: https://reactjs.org/docs/react-without-jsx.html
- Warum JSX in React: https://reactjs.org/docs/introducing-jsx.html