

React Native



Ein Einstieg

aus Sicht eines Web Developers

React Core Concepts

- + JSX
- + Components & Props
- + State
- + Rendering Elements

JSX

```
const element = <h1>Hello, world! </h1>;
```

JSX

```
const element = (  
  <h1 className="greeting">  
    Hello, world!  
  </h1>  
);
```

JSX

```
const element = React.createElement(  
  'h1',  
  {className: 'greeting'},  
  'Hello, world!'  
);
```

Components & Props

- + Elements sind kleinste Einheit

```
const element = <h1>Hello, world!</h1>;
```

- + Components (UI + Logik) akzeptieren props als Inputparameter und returnen React Elements

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

ES6 Class Components

```
class Welcome extends React.Component {  
  nameToUpperCase (name) {  
    return name.toUpperCase()  
  }  
  
  render() {  
    return <h1>Hello, {this.nameToUpperCase("thomas")}</h1>;  
  }  
}
```

State

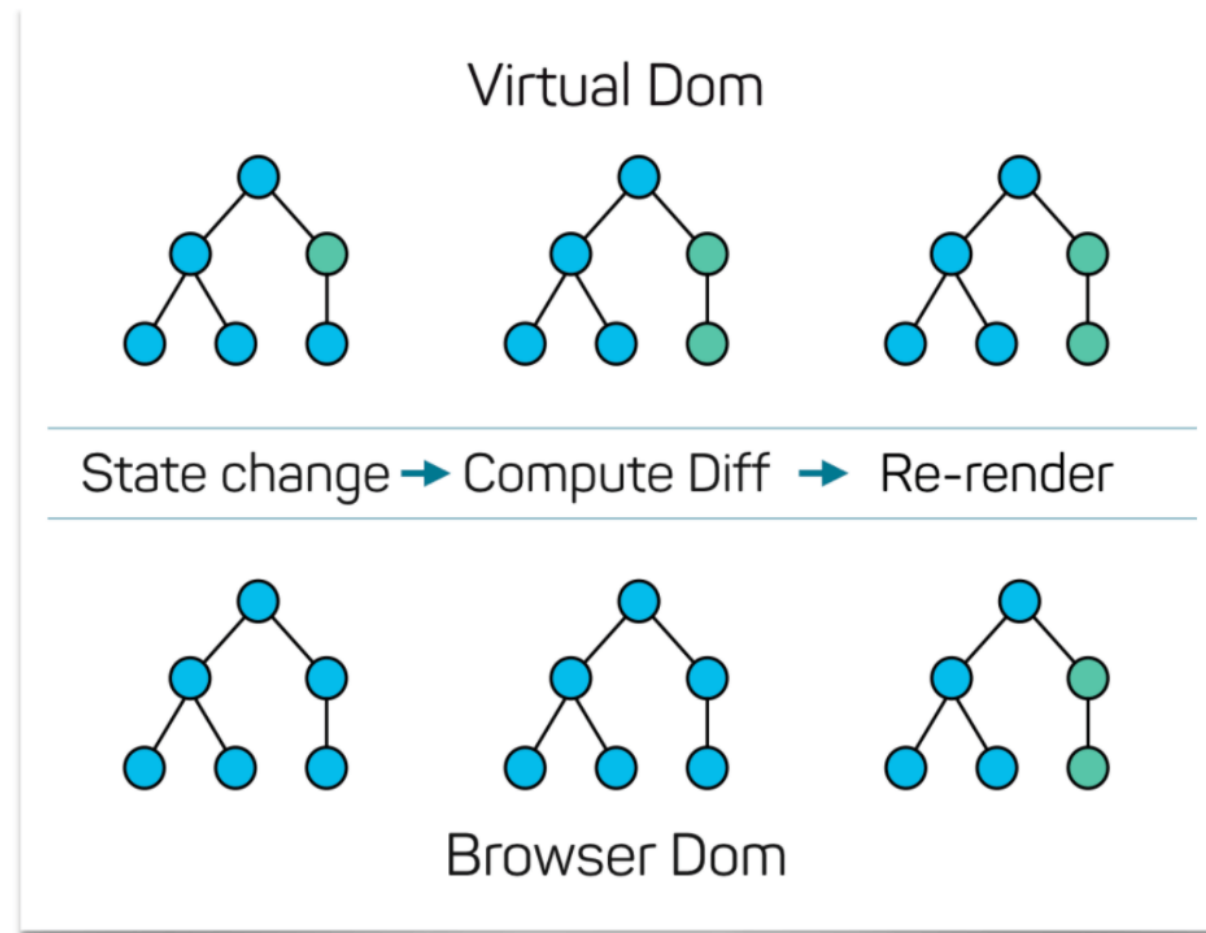
- + Props read-only
- + State innerhalb einer Component
- + Redux für Application State Management

State

```
class Name extends Component {
  constructor(props) {
    super(props);
    this.state = {name: 'Thomas'};
  }
  render() {
    const {name} = this.state
    return (
      <div>
        <h1>{name}</h1>
        <button onClick={() => this.setState({name: 'Klaus'})}>
          Change Name
        </button>
      </div>
    )
  }
}
```

Rendering Elements

Rendering Elements



(Quelle <https://mfrachet.github.io/create-frontent-framework/vdom/intro.html>)

React vs. React Native

React

React Native

Web Components

Native Components

Web APIs

Native APIs

CSS

Style in JavaScript

-

Native Modules

Native Bridge



(Quelle <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>)



SHOPGATE

- + Shopping App bereits Hybrid-Lösung mit React + Native Wrapper
- + Shopgate CONNECT Platform – React Native App als Frontend
- + Neues Produkt vollständig React Native + Shopgate CONNECT

Getting Started

- + Expo.io oder React Native CLI
- + Neue App initialisieren
- + Erste Component schreiben
- + Camera API ansprechen



Quellen

JSX <https://reactjs.org/docs/introducing-jsx.html>

Elements <https://reactjs.org/docs/glossary.html#elements>

Components <https://reactjs.org/docs/components-and-props.html>

State <https://reactjs.org/docs/state-and-lifecycle.html>

Redux <https://redux.js.org/>

Rendering <https://reactjs.org/docs/rendering-elements.html>

Virtual DOM <https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/>

JavaScript Bridge <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>

Getting Started

Dependencies installieren:

- Android Studio, ADB und Android SDK und Build Tools
- React Native CLI

```
react-native init NAME  
react-native run-android
```

Live Reloading aktivieren durch schütteln des Geräts und auswählen der Option

Camera Example

Module installieren `npm i react-native-camera`

Module linken `react-native link react-native-camera`

`run-android` neu ausführen

Berechtigungen nicht vergessen

Example code

github.com/thomasrehm/reactnativeapp-example

