

React: Events & State & Forms

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

Was wir lernen werden (hoffentlich)

- React Basics
 - Komponenten, Events, Templates, Props & Forms
- React Router
 - Routen, Route Parameters, Redirects
- Redux
 - Stores, Actions, Reducers



React: DOM Events

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

Wie funktionieren React DOM Events

- Events in React werden nicht in Kleinbuchstaben, sondern in camelCase benannt: statt onclick also onClick
- IN JSX übergeben Sie eine Funktion als Eventhandler und nicht als String, also statt "activateLasers()" dann {activateLasers}
- Diese Events heißen synthetische Events und funktionieren Browserübergreifend

Zum Beispiel dieses HTML:

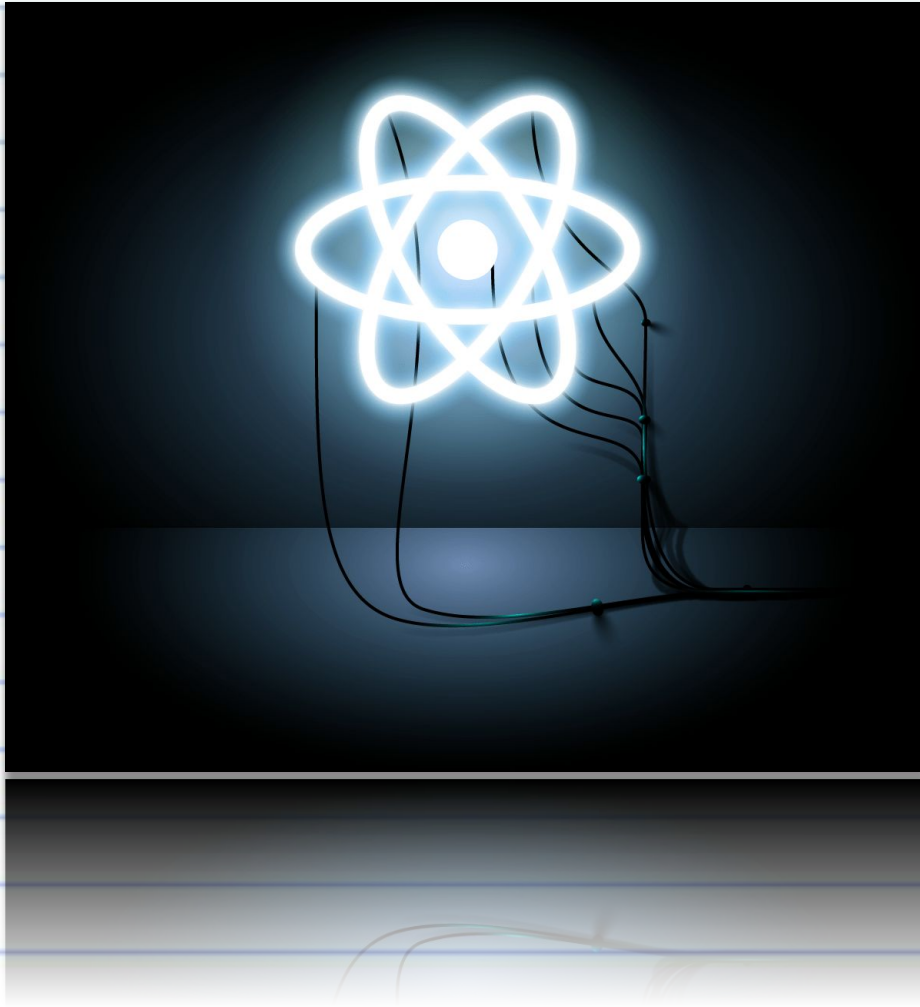
```
<button onclick="activateLasers()">  
  Aktiviere Laser  
</button>
```

ist in React ein wenig anders:

```
<button onClick={activateLasers}>  
  Aktiviere Laser  
</button>
```

```
<\button>  
  AKTIVIERE LASER  
<button onclick={activateLasers}>
```

React DOM Events Beispiele



```
<button onClick={this.handleClick}>Click  
Me</button>
```

```
handleClick(e) {  
    console.log(e.target);  
}
```

```
<button  
onMouseOver={this.handleMouseOver}>Hover  
Me</button>
```

```
handleMouseOver(e) {  
    console.log(e.target, e.pageX);  
}
```

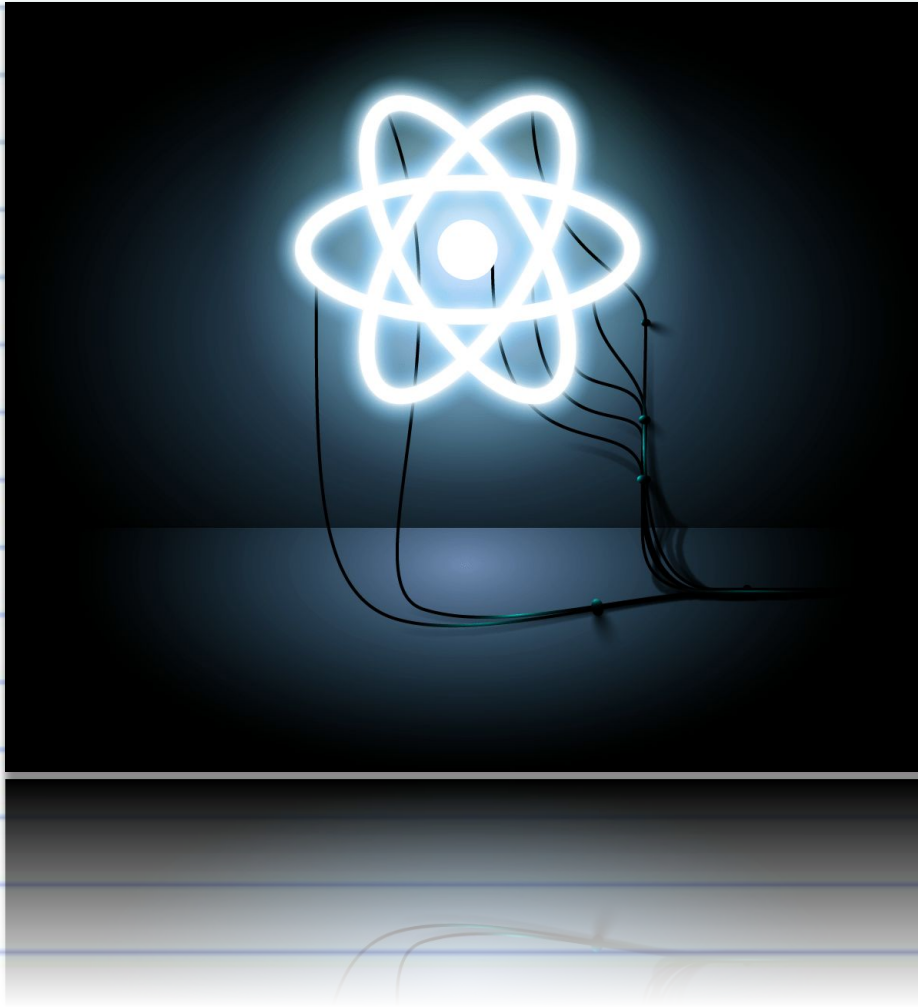
```
<p onCopy={this.handleCopy}>What we think,  
we become</p>
```

```
handleCopy(e) {  
    console.log('Try being original for  
once!');  
}
```

Browser & VS Code Demo



Wie funktionieren React DOM Events



Hier ist es ein synthetisches Event. React definiert diese synthetischen Events gemäß der W3C spec, also browserübergreifend.

Mit React sollten Sie im Normalfall `addEventListener` nicht aufrufen müssen, um Events an DOM Elemente zu binden, nachdem sie erstellt worden. Stattdessen stellen Sie einfach einen Listener zur Verfügung, wenn das Element initial gerendert wurde.

React DOM Events zum Nachlesen

- Handhabung von Events in React-Elementen ist ähnlich wie bei DOM-Elementen
- <https://de.reactjs.org/docs/handling-events.html>
- Die ganze Liste unterstützter Events finden Sie hier
- <https://de.reactjs.org/docs/events.html>



React: Changing State

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

Browser & VS Code Demo



React - State verändern (+ "this")

Angenommen wir wollen auf den State im Eventhandler zugreifen:



```
handleClick(e) {  
  console.log(this.state);  
}
```

Wir bekommen eine Fehlermeldung "this is undefined".

Warum geht das nicht?

=> anderes this als im render() !!!

Zwei verschiedene "this":

1. JavaScript

- Das "this" (Schlüsselwort) im JavaScript referenziert auf das Objekt selbst
- Wird also normalerweise beim Aufruf automatisch "verbunden"

```
const test = {  
  prop: 42,  
  func: function() {  
    return this.prop;  
  },  
};  
  
console.log(test.func());  
// expected output: 42
```

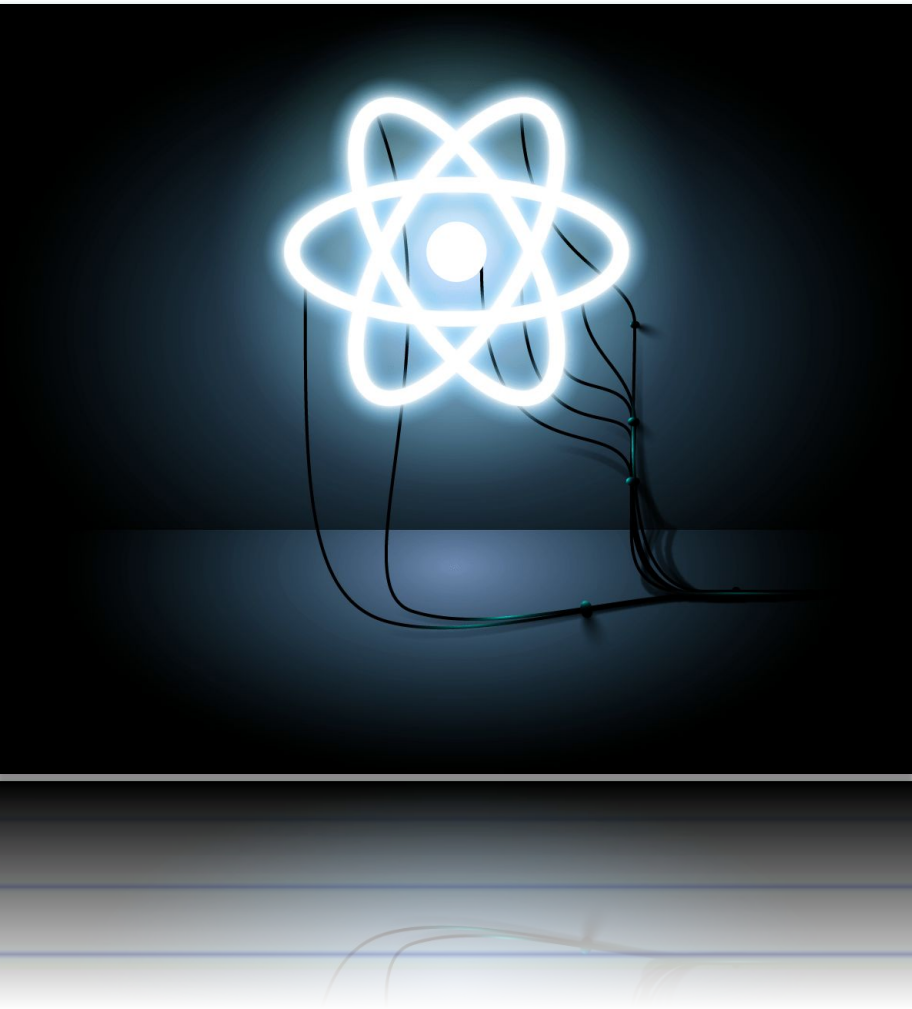
Mehr Infos unter

<https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Operators/this>



Zwei verschiedene "this":

2. React



- Um auf `this.state` oder Ähnliches zugreifen zu können, müssen wir es explizit "binden", z.B.:
- `this.clickFunction.bind(this)`
- Diese geschieht im Konstruktor der Klasse
- Danach können wir auf alle eigenen `this` Properties zugreifen
- Mehr Infos unter <https://medium.com/byte-sized-react/what-is-this-in-react-25c62c31480>

Even better:

Quick Fix React (mit ES6 JS)

Statt es händisch zu binden mit `bind()`

Können wir auch einen kleinen JS-Trick anwenden:

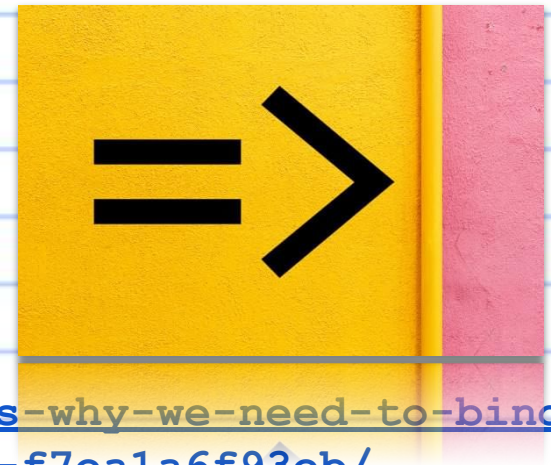
Statt einer normalen Funktion können wir auch eine moderne JS Arrow-Funktion benutzen!

Damit ändert sich auch automatisch der "this" Wert und wird automatisch gebunden, ohne das wir es explizit zu binden müssen.

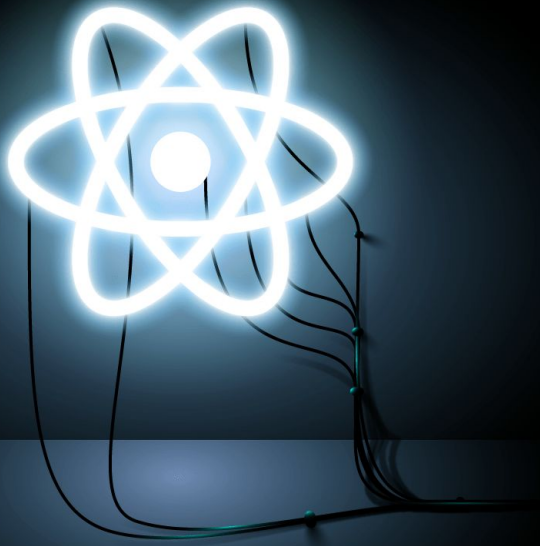
```
handleClick = (e) => {  
  console.log(this.state);  
}
```

Mehr darüber

<https://www.freecodecamp.org/news/this-is-why-we-need-to-bind-event-handlers-in-class-components-in-react-f7ea1a6f93eb/>



Finally State Change!!!



```
handleClick = (e) => {  
  this.setState({  
    name: 'Yoshi',  
    age: '25'  
  });  
  console.log(this.state);  
}
```

Damit wird beim Klick der State verändert.

ACHTUNG! Bitte NICHT Direkt ändern, sondern IMMER über `this.setState({...})`.

Browser & VS Code Demo



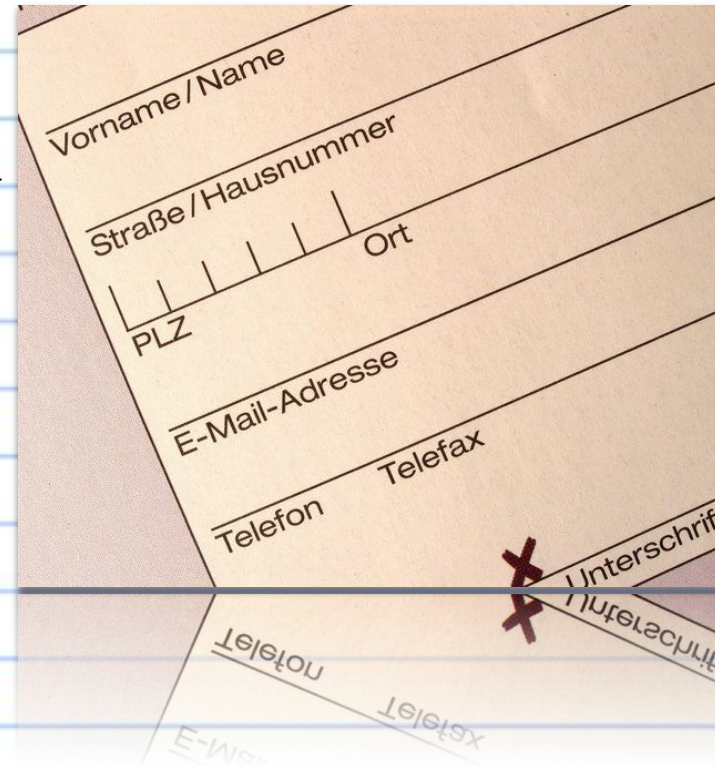
React:

Intro to Forms

Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin

Zielsetzung des Formulars

- Titel der Website "My name is Ryu"
- Wir haben ein einziges Input Feld, wo der Nutzer seinen Namen eintippen kann
- Wir haben einen Button, mit dem er das Formular abschicken kann
- Sobald er seinen Namen abgeschickt hat, soll sich der Titel dem Usernamen anpassen
- Und dieser Name soll auch in der Konsole geloggt werden!



A photograph of a physical form, likely a registration or contact form, with the following fields and labels:

- Vorname / Name
- Straße / Hausnummer
- PLZ
- Ort
- E-Mail-Adresse
- Telefon
- Telefax
- Unterschrift

The form is partially filled out, and there is a red 'X' mark over the signature field.

Input Form Code Beispiel

Vorname/Name

Straße/Hausnummer

PLZ

Ort

E-Mail-Adresse

Telefon

Telefax

Unterschrift (Bitte)

```
<form>  
  <input type="text"  
    onChange={this.handleChange}  
  >  
  </input>  
  <button>Submit</button>  
>  
</form>  
  
handleChange = (e) => {  
  this.setState({  
    name:  
    e.target.value  
  });  
}
```

Browser & VS Code Demo



Formular Funktion onSubmit

```
<form onSubmit={this.handleSubmit}>  
  <input type="text" onChange={this.handleChange}>  
  </input>  
  <button>Submit</button>  
</form>
```

```
handleSubmit = (e) => {  
  e.preventDefault();  
  console.log('form submitted', this.state.name);  
}
```



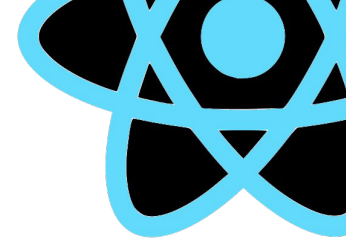
Browser & VS Code Demo



React: Create React App

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

How To Setup React ?



Wir haben zwei Möglichkeiten, React für unser Projekt zu benutzen

1. "create-react-app"

Ein Webpack Setup Paket, das hilft ein komplettes React-Projekt zu kreieren und den Flow zu verwalten

2. Erste zwei CDN*-Links von folgender Website in unseren HTML HEAD (unter dem CSS) einbinden, wenn wir nur Teile von React ausprobieren wollen (schneller & besser zum Lernen):

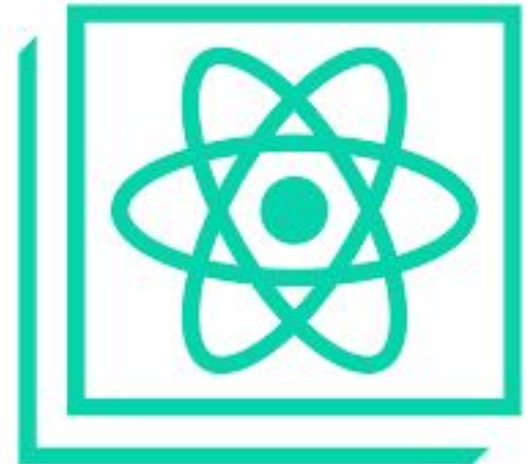
<https://reactjs.org/docs/cdn-links.html>

*Content-Delivery-Network

Was ist Create React App ?

Ein Kommandozeilentool um React Apps zu generieren (> 1 Komponente ;)

- Npm Entwicklungsserver
- Nutzt moderne ES6 Features, die normalerweise nicht unterstützt werden
- Unser Code bleibt modular
- Benutzt Build Tools wie Babel und Webpack um unseren Code zu optimieren



How To Use: create-react-app

Keine Installation nötig!

Außer aktuelles Node (\geq npm 5.2+)

<https://nodejs.org/en/download/>

```
> npx create-react-app my-app (--use-npm)
```

```
> cd my-app
```

```
> npm start
```



Bei Problemen hilft Node Version Manager (nvm)

=> <https://github.com/nvm-sh/nvm#install--update-script>

Kommt auch ursprünglich von Facebook

Wird von der Community weiterentwickelt

Mehr auf <https://github.com/facebook/create-react-app>

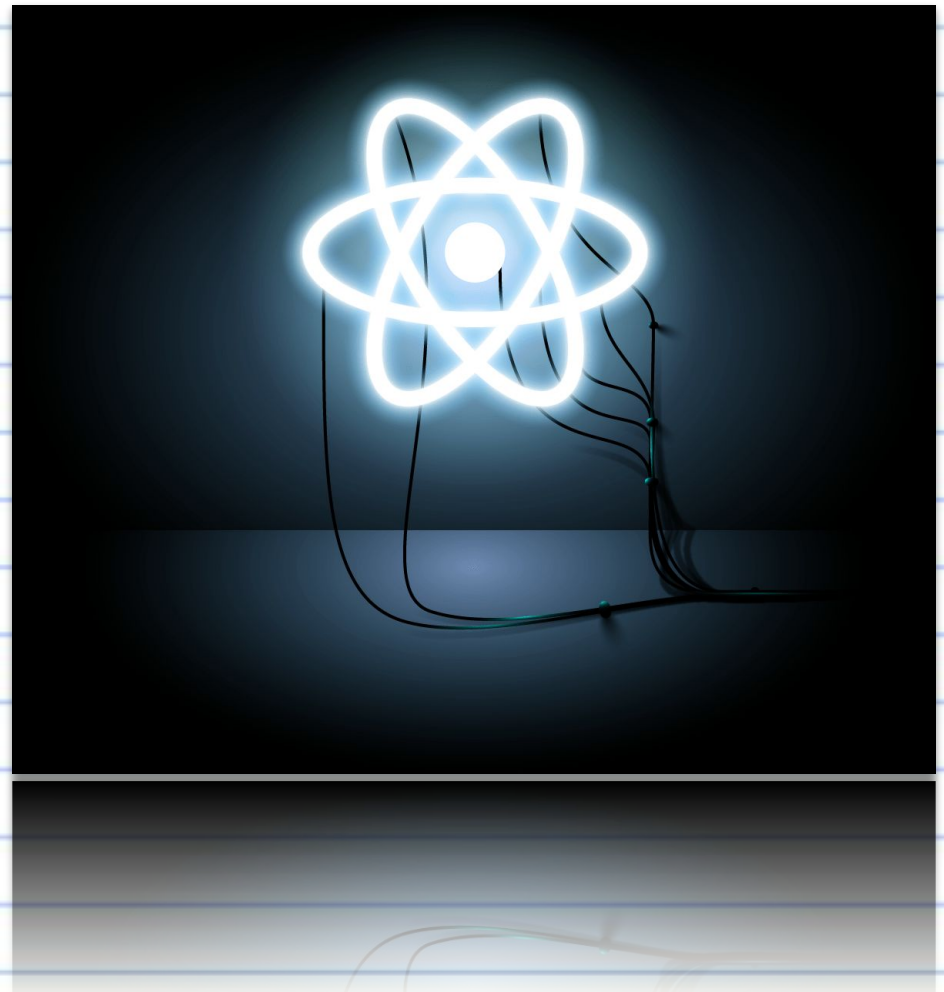
Konsole & VS Code Demo



Ordnerstruktur create-react-app

my-app

- |— README.md
- |— node_modules
- |— package.json
- |— .gitignore
- |— public
 - |— favicon.ico
 - |— index.html
 - |— manifest.json
- |— src
 - |— App.css
 - |— App.js
 - |— App.test.js
 - |— index.css
 - |— index.js
 - |— logo.svg
 - |— serviceWorker.js
 - |— setupTests.js



Möglichkeiten create-react-app

Success! Created my-app at /home/htw/my-app

Inside that directory, you can run several commands:

```
npm start
```

Starts the development server.

```
npm run build
```

Bundles the app into static files for production.

```
npm test
```

Starts the test runner.

```
npm run eject
```

Removes this tool and copies build dependencies, configuration files

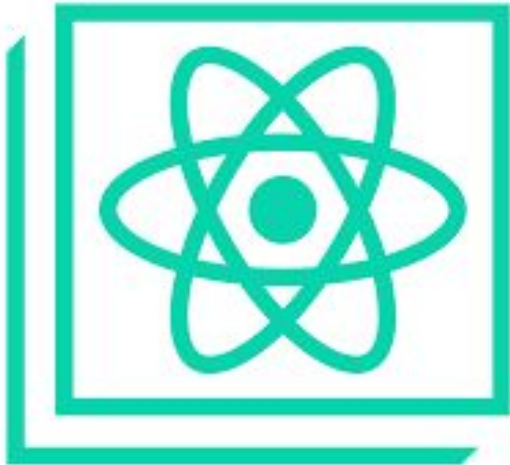
and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

```
cd my-app
```

```
npm start
```

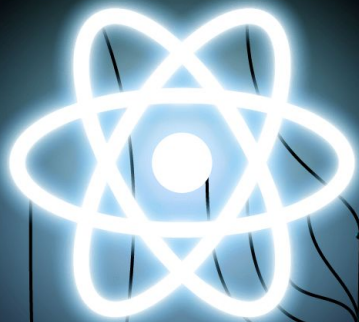
Happy hacking!



React: Single Page Apps

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

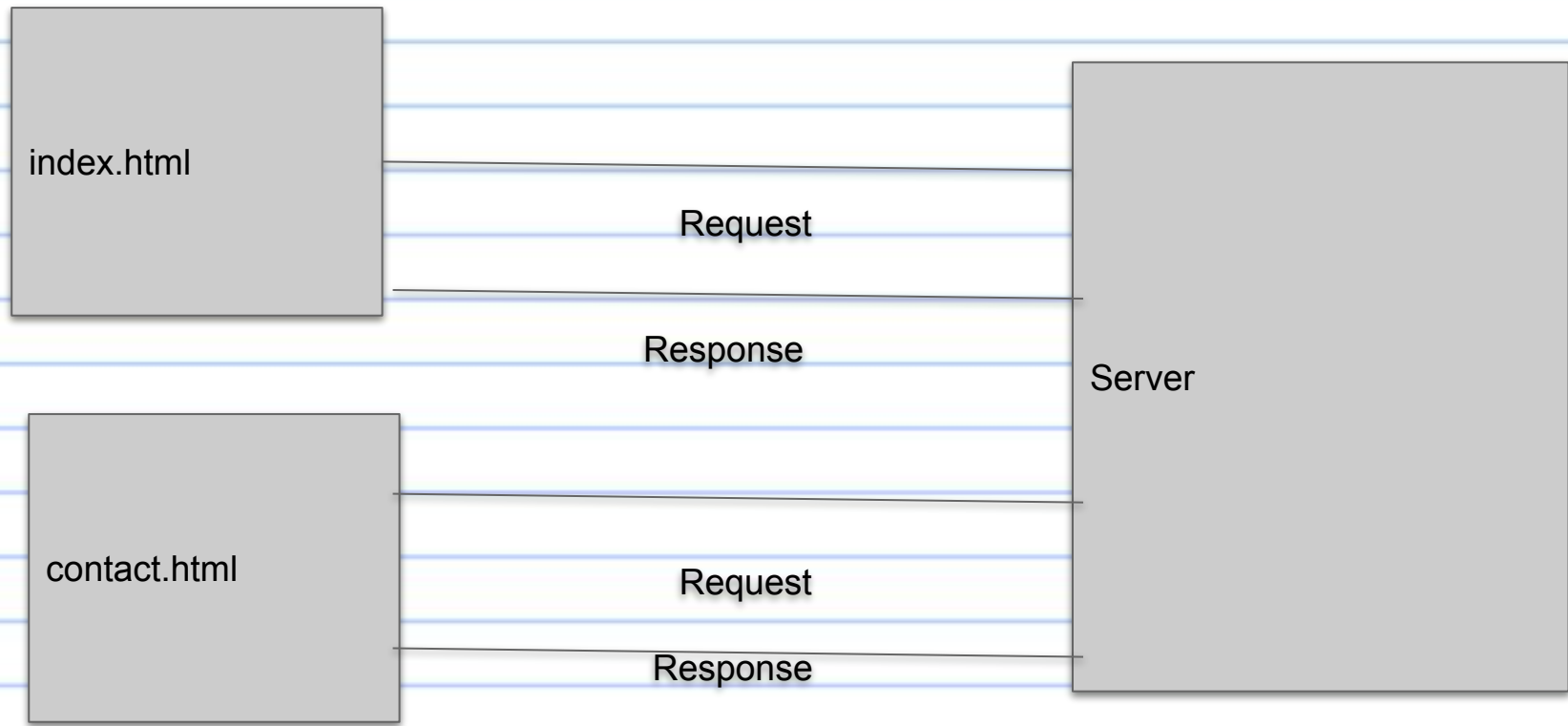
React: Single Page Apps



- React Apps sind typischerweise SPA's
- Es wird immer nur EINE HTML Seite dem Browser vom Server gesendet
- React kontrolliert was der Nutzer auf der Seite sieht

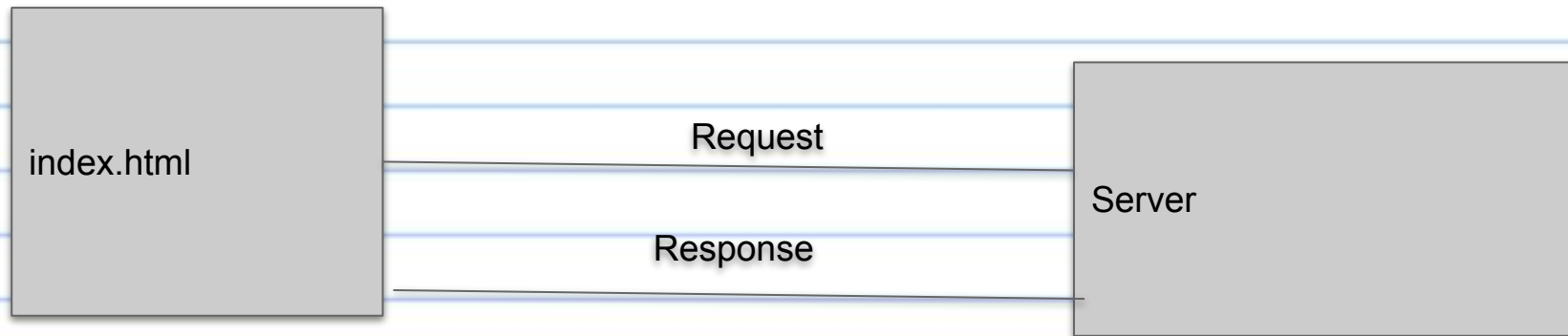
Kein React: Multi Page Apps

`/index`

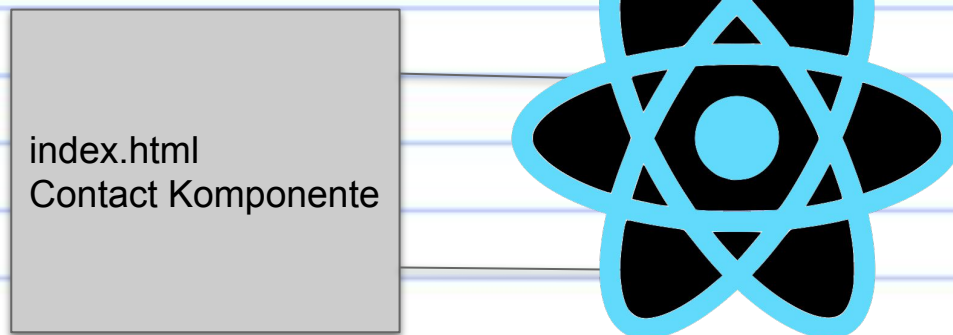


React: Single Page Apps

`/index`



`/contact`



VS Code & Browser Demo



Den npm Entwicklungsserver starten

```
> npm start
```

```
Compiled successfully!
```

```
You can now view my-app in the browser.
```

Local: <http://localhost:3000>

On Your Network: <http://192.168.1.22:3000>

Note that the development build is not optimized.

To create a production build, use `npm run build`.

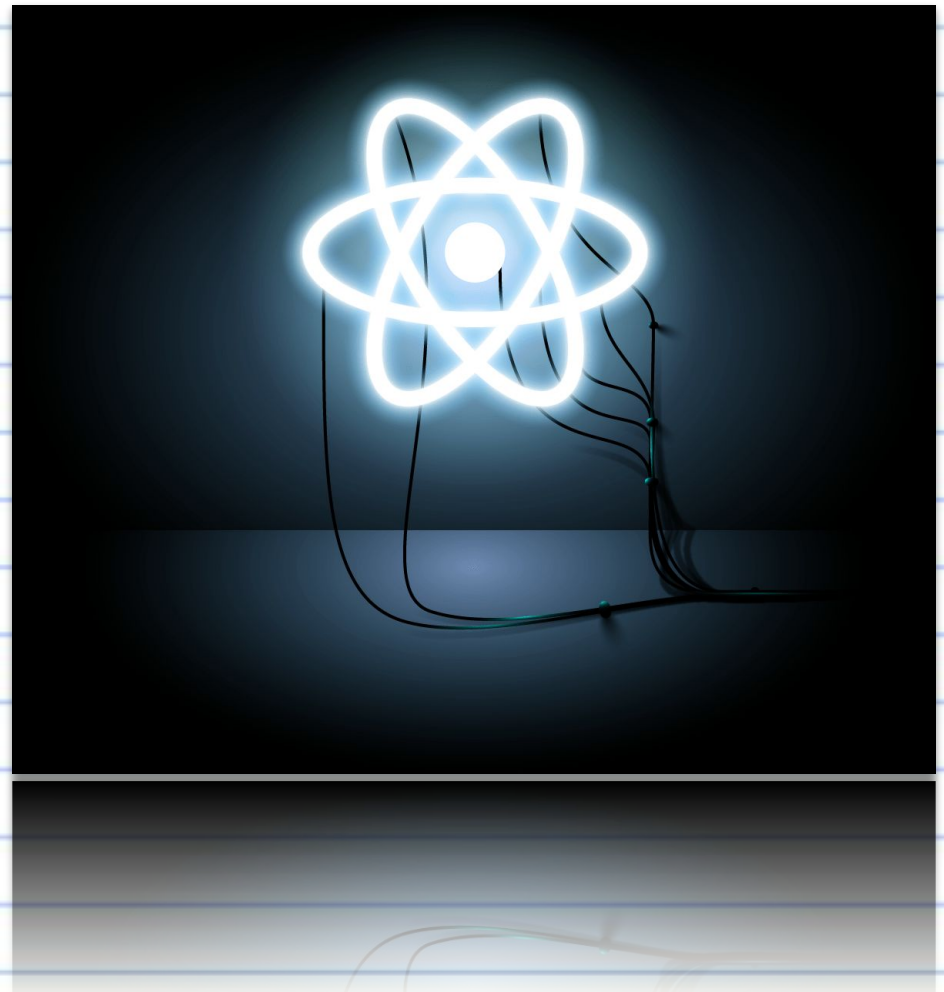
```
> Strg + C zum Stoppen des Servers
```



Automatisch generierte Dateien aufräumen

my-app

- ├── README.md
- ├── node_modules
- ├── package.json
- ├── .gitignore
- ├── public
 - ├── favicon.ico
 - └── index.html
- ├── manifest.json
- └── src
 - ├── App.css
 - ├── App.js
 - ├── App.test.js
 - ├── index.css
 - ├── index.js
 - ├── logo.svg
 - ├── serviceWorker.js
 - └── setupTests.js



VS Code & Browser Demo



React: Nesting Components

*Martina Freundorfer
Hochschule für Technik &
Wirtschaft (HTW) Berlin*

<App>: Die Mutter aller Components

```
import React, { Component } from 'react';
```

```
class App extends Component {
```

```
  render() {
```

```
    return (
```

```
      <div className="App">
```

```
        <h1>My first React app</h1>
```

```
        <p>Welcome :)</p>
```

```
      </div>
```

```
    );
```

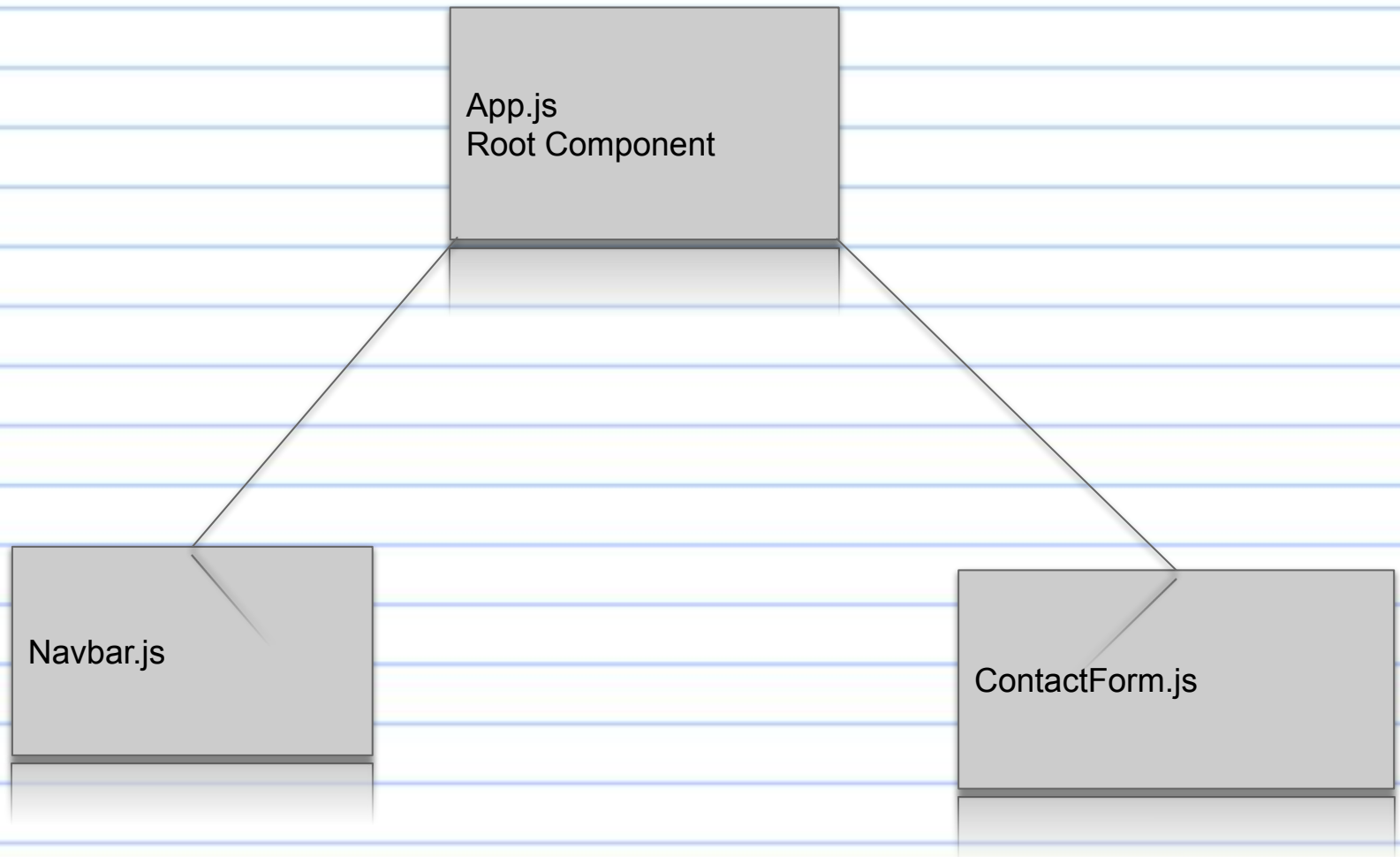
```
  }
```

```
}
```

```
export default App;
```



React: Nesting Components



VS Code & Browser Demo



Neue Komponente: <Ninjas>.js

```
import React, { Component } from 'react'
```

```
class Ninjas extends Component{
```

```
  render() {
```

```
    return (
```

```
      <div className="ninja">
```

```
        <div>Name: Ryu</div>
```

```
        <div>Age: 30</div>
```

```
        <div>Belt: Black</div>
```

```
      </div>
```

```
    )
```

```
  }
```

```
}
```

```
export default Ninjas
```



Importieren in <App>.js

```
import React, { Component } from 'react';  
import Ninjas from './Ninjas'
```

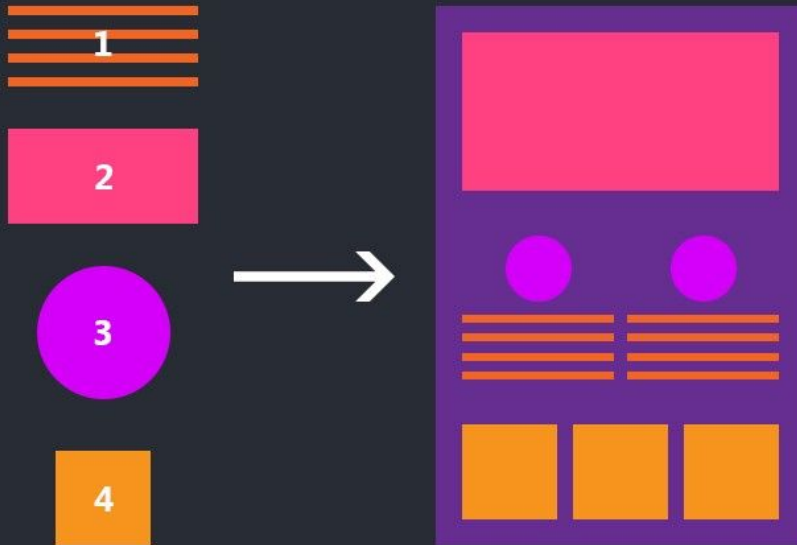
```
class App extends Component {  
  render() {  
    return (  
      <div className="App">  
        <h1>My first React app</h1>  
        <Ninjas />  
      </div>  
    );  
  }  
}
```

```
export default App;
```



Warum nochmal verschiedene Komponenten?

React Components



TD TECH DIAGONAL

JD TECH DIVCONVT

- Wiederbenutzbarkeit
- Kapselung
- Dry Principle: Don't repeat yourself!
- Eigene States, eigene Methoden
- Dadurch möglich, es auch innerhalb anderer Komponenten hinzuzufügen

Credits & More Lesestoff



Die Kursdateien für das React & Redux Tutorial

<https://github.com/htw-web/react-redux-complete-playlist>

Shaun "The Net Ninja" Pelling (Github: iamshaunjp)

<https://www.youtube.com/channel/UCW5YeuERMmlnqo4oq8vwUpq>

<https://github.com/iamshaunjp>

Getting Started with React - An Overview and Walkthrough
Tutorial (including create-react-app)

<https://www.taniarascia.com/getting-started-with-react/>

Vielen Dank für Ihre Aufmerksamkeit!