JavaScript und JavaScript Frameworks

React – Erste Schritte

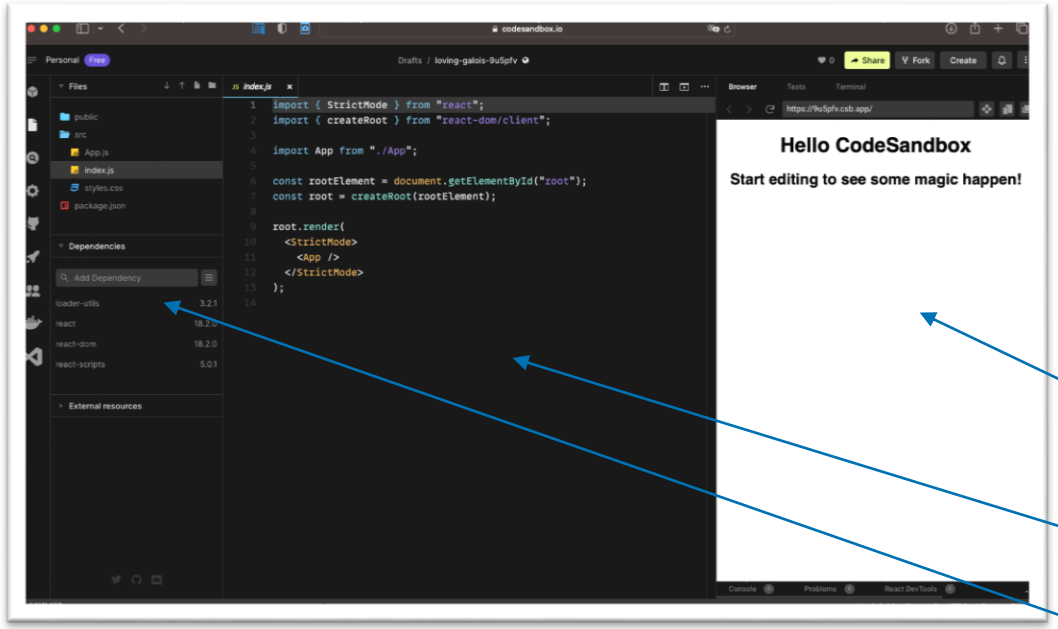
Übersicht und Zielstellung

- In diesem Abschnitt erfahren Sie:
  - wie ein React-Projekt organisiert ist,
  - wie die einfache Dateistruktur aufgebaut ist,
  - wie die React-Anwendung „tickt“,
  - wie man Komponenten erstellt,
  - wie man mit Komponenten arbeitet,

Start in das Thema

CodeSandbox.io zur Entwicklung nutzen

Das erste Mal!



Öffnen Sie die Entwicklungs-Umgebung im Browser.

Legen Sie ein neues Projekt an oder laden sie ein bestehendes.

Betrachten Sie den Bildschirm.

Folgende Inhalte sind zu sehen:

- Vorschau
- Code-Editor
- Verzeichnisstruktur und Dateiarbeit

CodeSandbox.io zur Entwicklung nutzen

Die vorliegenden Dateien

index.js

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";

import App from "./App";

const rootElement = document.getElementById("root");
const root = createRoot(rootElement);

root.render(
   <StrictMode>
      <App />
   </StrictMode>
);
```

Die erste Zeile importiert die React-StrictMode-Komponente aus der React-Bibliothek. StrictMode ist ein Tool, das Entwicklern hilft, potenzielle Probleme in ihren React-Code zu finden und zu beheben, indem es zusätzliche Warnungen und Fehlermeldungen im Console-Output generiert.

Die zweite Zeile importiert die "createRoot" Funktion aus der "react-dom/client" Bibliothek. "createRoot" ist eine neue Methode, die seit React 18 verfügbar ist und die Möglichkeit bietet, eine App im Concurrent Mode zu rendern.

Die dritte Zeile importiert die "App" Komponente aus einer separaten Datei namens "App.js".

In der vierten Zeile wird das DOM-Element "root" aus dem HTML-Dokument ausgewählt und der Konstante "rootElement" zugewiesen.

Die fünfte Zeile ruft die "createRoot" Funktion auf und gibt das DOM-Element "rootElement" als Argument an. Das Ergebnis wird in der Konstante "root" gespeichert.

In der sechsten Zeile wird die "render" Methode auf "root" aufgerufen, um die "App" Komponente in das "root" DOM-Element zu rendern. Die "App" Komponente wird in ein <StrictMode> Element eingewickelt, um Entwickler zu warnen, falls sie veraltete Funktionen oder Technologien verwenden.

CodeSandbox.io zur Entwicklung nutzen

Die vorliegenden Dateien

App.js

```
import "./styles.css";

export default function App() {
   return (
      <div className="App">
         <h1>Hello CodeSandbox</h1>
         <h2>Start editing to see some magic happen!</h2>
      </div> );
}
```

Dies ist ein einfaches React-Komponenten-Codebeispiel, der eine HTML-DOM-Struktur (h1 und h2-Elemente) zurückgibt. Die CSS-Styles für diese HTML-Elemente werden in einer separaten Datei namens "styles.css" definiert und in die Komponente importiert..

Die Zeile `import "./styles.css";` importiert die Styles aus der "./styles.css" Datei und fügt sie in die Komponente ein. Dadurch können die definierten Styles auf die HTML-Elemente angewendet werden.

Die Funktion "App" ist die Hauptkomponente und sie gibt die HTML-Struktur zurück. Die "className" -Eigenschaft wird verwendet, um der Wurzel-Div einen Klassennamen "App" zuzuweisen, auf den in der CSS-Datei verwiesen werden kann.

CodeSandbox.io zur Entwicklung nutzen

Das Vorgehen – Der Einstieg mit einfachen Komponenten

Das Grundprinzip

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";

import App from "./App";

const rootElement = document.getElementById("root");
const root = createRoot(rootElement);

root.render(
   <StrictMode>
      <App />
   </StrictMode>
);
```

Im HTML-Dokument wird ein Element gesucht – die ID root.

Für das React-Rendern wird das Element übergeben.

Dann wird der Inhalt für das Element gerändert.

Der eigentliche Inhalt ist in der vorher importierten Komponente App zusammengesetzt

CodeSandbox.io zur Entwicklung nutzen

Das Vorgehen – Der Einstieg mit einfachen Komponenten

Nun geht es darum, weitere Komponenten zu erstellen und einzubinden

```
src/
├── App.js
├── index.js
├── components/
│   ├── Header.js
│   ├── Footer.js
│   ├── Home.js
│   └── ...
└── ...
```

Alle Quell-Dateien liegen im Ordner src

Es bietet sich an, mit Unterordnern eine gewisse Ordner-Struktur zu definieren

Im Ordner components sind nun ein paar Komponenten definiert

Die Komponente App sollte im src-Ordner liegen

In der Komponente App werden die Komponenten zusammengefügt.

CodeSandbox.io zur Entwicklung nutzen

Das Vorgehen – Der Einstieg mit einfachen Komponenten

Schauen wir uns den Aufbau einer einfachen Komponente an:

```
// components/Header.js

import React from 'react';

export function Header() {
  return (
    <header>
      <h1>Header</h1>
    </header>
  );
}
```

Als erster wird der React Bestandteil geladen

Dann wird eine export Funktion für die Ausgabe definiert
Der Name sollte dem Componenten-Namen entsprechen.

Für die Rückgabe wird nun der Code definiert (hier einfacher HTML Code)

So können individuelle Komponenten definiert werden.

CodeSandbox.io zur Entwicklung nutzen

Das Vorgehen – Der Einstieg mit einfachen Komponenten

Nun muss die Komponenten noch in die App eingebunden werden:

```
// App.js

import React from 'react';
import { Header } from './components/Header';

function App() {
  return (
    <div>
      <Header />
    </div>
  );
}

export default App;
```

Als erster wird der React Bestandteil geladen

Dann wird die Komponenten importiert.

Die Komponente wird in die App eingebunden

Durch export wird die Gesamtkomponente ausgeliefert und gerändert.

Aufgabe: Erstellen Sie eine Komponente für dem Inhalt und eine Komponente für den Fuß.

CodeSandbox.io zur Entwicklung nutzen

Komponenten mit Properties

Eine Komponenten kann auch für einen mehrfachen Einsatz vorbereitet werden.

Durch die Übergabe von Parameter(props) können Komponenten mit Platzhaltern ausgestattet werden.

```
// components/Visitenkarte.js

import React from 'react';

export function Visitenkarte(props) {
  return (
    <div>
      <h1>Angaben zur Person</h1>
      <p>{props.vorname} {props.famname}</p>
    </div>
  );
}
```

Die „Platzhalter“ werden im Code mit angegeben.

Sie werden in {} eingebunden.

Die Bezeichnung setzt sich zusammen aus dem Parameternamen in der Funktion und dem Namen, der in der App verwenden wird.

CodeSandbox.io zur Entwicklung nutzen

Das Vorgehen – Der Einstieg mit einfachen Komponenten

Nun muss die Komponenten noch in die App eingebunden werden:

```
// App.js

import React from 'react';
import { Header } from './components/Header';
import { Visitenkarte } from './components/Visitenkarte';


function App() {
  return (
    <div>
      <Header />
      <Visitenkarte vorname="Gerald" famname="Reinhardt" />
    </div>
  );
}

export default App;
```

Beim Einbinden der Komponente können die Werte definiert werden.

Achten Sie darauf, dass die „Attributnamen“ den entsprechen, die in der Komponente verwendet werden.

VIELEN DANK!