React Hello World HTML Blog

Minimale Voraussetzungen

Minimal kann React in einer einfachen HMTL Web Site eingebunden und verwendet werden. In der Regel beginnt man mit einer leeren HTML Site:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello World</title>
 </head>
 <body>
 </body>
</html>Alsann referenziert man die react.development.js und react-dom.development.js Scripts:<script src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>Sie können die React Scripts auch downloaden und lokal referenzieren.Für die Entwicklung referenziert man die React Development Versionen, solche sind nicht minimiert und bieten mehr Infos bei der Fehlersuche.

React basierend auf Javascript Vanilla

React Scripts arbeiten in der Regel mit dem HTML DOM Tree. Wir definieren im body-Element ein div-Element mit der id "root". In diesem div-Elmement erstellen wir mit React ein h1-Element mit dem Text "Hello, world" dynamisch unter Einsatz von Javascript Vanilla Befehlen:<div id="root"></div>
<script>
 ReactDOM.render(React.createElement("h1", null, "Hello, world!"), document.getElementById('root'));
</script>React ist der Einstiegspunkt in die React-Bibliothek. Da wir React von einem <script>-Tag laden, sind diese APIs der obersten Ebene im React global verfügbar.Mit React.createElement erstellt man ein neues React-Element des angegebenen Typs und geben solches zurück. Das Typargument kann entweder eine Tag-Name (z. B. 'div' oder 'span'), ein React-Komponententyp (eine Klasse oder eine Funktion) oder ein React-Fragmenttyp sein. In unserem Fall erstellen wir also ein h1-Tag Element mit einem Character Child.ReactDOM gehört zum react-dom Package, solches bietet DOM-spezifische Methoden, die auf der obersten Ebene der Anwendung verwendet werden können.ReactDOM.render für ein Element in den zugeordneten Container ein. Falls der Container nicht leer ist, wird solcher aktualisiert.

Lösung React Develop Javascript Vanilla

Das HTML Script mit React Develop Javascript:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello World</title>
 <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
 <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
 </head>
 <body>
 <div id="root"></div>
 <script>
 ReactDOM.render(React.createElement("h1", null, "Hello, world React Develop Javascript!"), document.getElementById('root'));
 </script>
 </body>
</html>Die laufende Lösung zu diesem Blog und React Develop finden Sie hier

React JSX Babel

JSX ist eine Javascript Syntaxerweiterung welche z.B. folgendes Statement erlaubt:const element = <h1>Hello, world</ h1>Es handelt sich hier um keinen String und auch kein HTML Code. Es ist ein JSX AusdruckFür die Entwicklung von React Scripts wird JSX empfohlen. JSX erzeugt React "Elemente" mit einer einfachen Syntax. JSX ist eigentlich nur ein Compiler, welcher Anweisungen nach Javascript übersetzt. Es handelt sich hier um den Babel Compiler.Unter dem Link https://babeljs.io/en/repl kann man den Compiler online testen und

damit dessen Arbeitsweise anwenden.Damit wir Babel benutzen können müssen wir das folgende Script referenzieren:<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>Unsere Aufgabe lässt sich nun wie folgt etwas einfacher lösen:ReactDOM.render(
 <h1>Hello, world!</h1>,
 document.getElementById('root')
);

Lösung React JSX Develop Babel

Das HTML Script Hello World mit React JSX Develop Babel:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello World</title>
 <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
 <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
 </head>
 <body>
 <div id="root"></div>
 <script type="text/babel">
 ReactDOM.render(
 <h1>Hello, world React Develop JSX Babel!</h1>,
 document.getElementById('root')
 );
 </script>
 </body>
</html>Die laufende Lösung zu diesem Blog und React Develop finden Sie hier

React JSX (Babel) Prod

In einer produktiven Umgebung arbeiten wir mit dem Minified React Production Script:<script src="https://unpkg.com/react@16.7.0/umd/react.production.min.js"></script>
<script src="https://unpkg.com/react-dom@16.7.0/umd/react-dom.production.min.js"></script>Vom Programm her ändert sich nichts.

Lösung (React JSX Prod)

Das HTML Script React JSX Prod:<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Hello World</title>
 <script src="https://unpkg.com/react@16.7.0/umd/react.production.min.js"></script>
 <script src="https://unpkg.com/react-dom@16.7.0/umd/react-dom.production.min.js"></script>
 <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
 </head>
 <body>
 <div id="root"></div>
 <script type="text/babel">
 ReactDOM.render(
 <h1>Hello, world React Prod JSX Babel!</h1>,
 document.getElementById('root')
 );
 </script>
 </body>
</html>Die laufende Lösung zu diesem Blog und React Prod finden Sie hier

Feedback

War dieser Blog für Sie wertvoll. Wir danken für jede Anregung und Feedback

Kontakt

Simtech AG

Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

https://www.simtech-ag.ch/typs