

React

dinsdag 8 oktober 2019 13:17

29

Return `React.createElement('div', null, 'h1', 'Hi, I\'m a React App')`

Configuration for the div = second element

Children in div = third element h1 zal al tekst tonen voor een tag H1 moet je een nieuwe element creëren

Voor css kan je het in null => `{className: 'App'}`

This is the reason that we use JSX (does like html)

30

In html is het class maar bij jsx is het className

Het moet 1 root element hebben in return dus `<div></div>` `<h1></h1>` zal niet werken

```
src > JS App.js > ...
1  import React, { Component } from 'react';
2  import './App.css';
3  import Person from './Person/Person.js';
4
5
6  class App extends Component {
7    render() {
8      return (
9        <div className="App">
10          <h1>Hi, I'm a React App</h1>
11          <button>Verander naam</button>
12          <Person name="Cansu" age="21">Test children</Person>
13        </div>
14      );
15    }
16  }
17
18  export default App;
```

```
import React from 'react';

//function
const person = (props) => {
  //return <p>I'm a person and I am {Math.floor(Math.random()*30)} years old!</p>
  return (
    <div>
      <p>I'm {props.name} and I am {props.age} years old!</p>
      <p>{props.children}</p>
    </div>
  )
};

export default person;
```

2)

```

src > JS App.js > App > render
1  import React, { Component } from 'react';
2  import './App.css';
3  import Person from './Person/Person.js';
4
5  class App extends Component {
6    state = {
7      persons : [
8        { name:'Cansu', age:21 },
9        { name : 'Test', age:23 },
10     ],
11   }
12
13   switchName = () => {
14     //console.log('test')
15     this.setState({
16       persons : [
17         { name:'Cansu', age:21 },
18         { name : 'Test', age:30 },
19       ],
20     })
21   }
22   render() {
23     return (
24       <div className="App">
25         <h1>Hi, I'am a React App</h1>
26
27         /* only pas the reference this.switchName not this.switchName() */
28         <button onClick={this.switchName}>Verander naam</button>
29         <Person name={this.state.persons[0].name} age={this.state.persons[0].age}>Test children</Person>
30         <Person name={this.state.persons[1].name} age={this.state.persons[1].age}>Test children</Person>
31       </div>
32     );
33   }
34 }
35
36 export default App;

```

```

src > Person > JS Person.js > ...
1  import React from 'react';
2
3
4  //function
5  const person = (props) => {
6    //return <p>I'm a person and I am {Math.floor(Math.random()*30)} years old!</p>
7    return (
8      <div>
9        <p>I'm {props.name} and I am {props.age} years old!</p>
10       <p>{props.children}</p>
11     </div>
12   )
13
14 };
15
16 export default person;
17

```

3) 43 dia is een moeilijker sectie 3

44: Statefull component is app (smart component or container comp)
More functional

Stateless, presentational, dom component is person because he has not interenal management

45: Bind oef

46: Tekstvakken inputvelden doorgeven

```
const app = (props) => {
  const [personsState, setPersonsState] = useState( {
    persons : [
      { name: 'Cansu', age: 21 },
      { name : 'Test', age: 23 },
    ],
  });

  const [otherState, setOtherState] = useState({otherState: 'OtherState'})

  console.log(personsState, otherState);

  const changeName = (event) => {
    setPersonsState({
      persons : [
        { name: 'Cansu', age: 21 },
        { name : event.target.value, age: 30 },
      ],
    })
  }

  return (
    <div className="App">
      <h1>Hi, I'm a React App</h1>
      <Person
        name={personsState.persons[0].name}
        age={personsState.persons[0].age}>Test childeren</Person>
      <Person
        name={personsState.persons[1].name}
        age={personsState.persons[1].age}
        changed={changeName}>Test childeren</Person>
      </div>
    );
  }
}
export default app;
```

```
person > JS Person.js > ...
import React from 'react';

//function
const person = (props) => {
  //return <p>I'm a person and I am {Math.floor(Math.random()*30)} years old!</p>
  return (
    <div>
      <p onClick={props.click}>I'm {props.name} and I am {props.age} years old!</p>
      <p>{props.children}</p>
      <input type="text" onChange={props.changed} value={props.name}/>
    </div>
  )
};

export default person;
```

babel src/app.js --out-file=public/scripts/app.js --presets=env,react --watch

Live-server public

Sectie6: laatste stap E6 class prop

```

class OldSyntax{
  constructor(){
    this.name = "Mike";
    //DIT IS DE OPLOSSING:
    this.getGreeting = this.getGreeting.bind(this);
  }

  getGreeting(){
    return `Hi my name is ${this.name}`
  }
}

const oldSyntax = new OldSyntax();
const getGreeting = oldSyntax.getGreeting;
console.log(getGreeting()); // GEEFT EEN FOUT WE LOSSEN DIT OP MET EEN BIND in constructor

class NewSyntax{
  name='Jen';
  getGreeting = () =>{
    return `Hi my name is ${this.name}`
  }
}

const newSyntax = new NewSyntax();
const newGetGreeting = newSyntax.getGreeting;
console.log(newGetGreeting());

```

Sectie7

```

const Layout = (props) =>{
  return(
    <div>
      <p>Header</p>
      {props.content}
      <p>Footer</p>
    </div>
  );
};

const template = (
  <div>
    <p>Page Title</p>
    <p>This is my page</p>
  </div>
);
ReactDOM.render(<Layout content={template}/>,document.getElementById('app'));

```

```

const Layout = (props) =>{
  return(
    <div>
      <p>Header</p>
      {props.children}
      <p>Footer</p>
    </div>
  );
};

const template = (
  <div>
    <p>Page Title</p>
    <p>This is my page</p>
  </div>
);

ReactDOM.render((
  <Layout>
    <p>Test inline</p>
  </Layout>
),document.getElementById('app'));

```

Sectie9

```

const NotFoundPage = () =>(
  <div>
    404! <Link to="/">Go home</Link>
  </div>
)

const Header = () => (
  <header>
    <h1>Expensify</h1>
    <NavLink to="/" activeClassName="is-active" exact>Home</NavLink>
    <NavLink to="/create" activeClassName="is-active">Create</NavLink>
    <NavLink to="/edit" activeClassName="is-active">Edit</NavLink>
    <NavLink to="/help" activeClassName="is-active">Help</NavLink>
  </header>
)

const routes = (
  <BrowserRouter>
    <div>
      <Header/>
      <Switch>
        <Route path="/" component={ExpenseDashboardPage} exact />
        <Route path="/create" component={Create} />
        <Route path="/edit" component={Edit} />
        <Route path="/help" component={Help} />
        <Route component={NotFoundPage}/>
      </Switch>
    </div>
  </BrowserRouter>
);
ReactDOM.render(routes ,document.getElementById('app'));

```

Setcie 10 redux

```
import { createStore } from 'redux';

const store = createStore((state = { count: 0 }, action) => {
  if (action.type === "INCREMENT") {
    return {
      count: state.count + 1
    };
  } else {
    return state;
  }
});

console.log(store.getState());

//dispatch zal store twee keer uitvoeren
store.dispatch({
  type: 'INCREMENT'
});

store.dispatch({
  type: 'INCREMENT'
});

console.log(store.getState());
```

Object { count: 0 }

Object { count: 2 }

```
import { createStore } from 'redux';

const store = createStore((state = { count: 0 }, action) => {
  switch (action.type) {
    case 'INCREMENT':
      return {
        count: state.count + 1
      };
    case 'DECREMENT':
      return {
        count: state.count - 1
      };
    default:
      return state;
  }
});

console.log(store.getState());

//dispatch zal store twee keer uitvoeren
store.dispatch({
  type: 'INCREMENT'
});

store.dispatch({
  type: 'DECREMENT'
});

store.dispatch({
  type: 'DECREMENT'
});

console.log(store.getState());
```

Object { count: 0 }

[redux-101.js:18](#)

Object { count: -1 }

```

//wordt altijd geroepen als het state veranderd
const unsubscribe = store.subscribe(() => {
  console.log(store.getState());
});

//ACTIONS
//dispatch zal store twee keer uitvoeren
store.dispatch({
  type: 'INCREMENT'
});

unsubscribe();

store.dispatch({
  type: 'DECREMENT'
});

store.dispatch({
  type: 'RESET'
});

store.dispatch({
  type: 'DECREMENT'
});

```

Zal enkel eerste tonen

Descruterig

```

const person = {
  name: 'Cansu',
  age: 21,
  location: {
    city: 'Gent',
    temp: 2
  }
};

const { name: firstname = 'Anonymous', age } = person; //destructured statement
console.log(`${firstname} is ${age}`);

const { city, temp: temperature } = person.location
if(city && temperature){
  console.log(`${city} is ${temperature} graden`);
}

const book = {
  title: 'Boek1',
  author: 'Een auteur',
  publisher: {
    name: 'Maan'
  }
}

const {title, author, publisher} = book;
const {name: publisherName='Zon'} = book.publisher;

if(publisherName){
  console.log(`Title: ${title} Author: ${author} Publisher: ${publisherName}`);
}

```

```
//ARRAY DESTRUCTOR//  
const address = ['desmetstraat','Gent','9000'];  
const [street, , zip= '0000'] = address;  
console.log(`Odisee: ${street} ${zip}`);
```