JSX en state variabelen

In REACT maken we gebruik van de 'taal' JSX. In deze les leren we wat JSX is.

Verder leren we wat in React een state variabele is en hoe we daar mee kunnen werken.

Met dit alles maken we een kleine applicatie waarmee de CSS-stijl van een kan worden aangepast door code.

Wat moet je weten?

Om deze les te kunnen volgen heb je kennis van:
Javascript (ES6), HTML, CSS en de REST API maken we in Yii, dus om deze API te maken is ook ervaring met Yii nodig.

ES6 versus ES5

React gebruikt JavaScript ES6 (ECMAScript 6). ES6 heeft op bepaalde punten een ander syntax dan ES5.

Kijk op

> https://medium.com/recraftrelic/es5-vs-es6-with-example-code-9901fa0136fc

waarin het verschil tussen ES5 en ES6 met voorbeelden wordt uitgelegd.

JSX

JSX staat voor JavaScript XML en is een mengeling van JavaScript en XML. XML is niets meer dan een formaat waarbij je telkens `<begin>..</begin>` gebruik. HTML-tags zijn eigenlijk XML tags. De XML tags in JSX lijken heel veel op HTML maar zijn hier en daar net even iets anders.

Een voorbeeld van JSX staat hieronder. Plaats deze code in de app.js die je tijdens de installatie hebt gemaakt.

```
function App() {
  return (
    <div>
      <h1 className="greeting">Hello!</h1>
      <h2>Good to see you here.</h2>
```

```
      <h2>It is {new Date().toLocaleTimeString()}.</h2>
    </div>
  );
}

export default App;
```

Dit is bijna allemaal 'gewone' HTML. Alleen *class* is geen *class,* maar heet in JSX *className*.

Verder zien we dat tussen de HTML JavaScript staat. Dit is zo omdat deze JavaScript in de HTML moet worden gezien als tekst staat deze tussen { }.

Kijk naar de volgende code die in app.js staat en probeer te voorspellen wat de code doet, voer het daarna uit en kijk of je begrijpt wat er gebeurt en waarom.

```
function App() {

  var myString = "";
  for (var i=0; i < 10; i++) {
    myString = myString + String(i) + ", ";
  }

  return (
    <div className="App">
      {myString}
    </div>
  );

}

export default App;
```

In deze functie staat een verwijzing naar een CSS class. Laten we deze toevoegen.

Maak een nieuwe file App.css in dezelfde folder als de App.js staat en zet daar het volgende in:

```
.App {
  text-align: center;
  color: red;
}
```

Voeg nu de regel: `import './App.css';` toe boven de functie App() in App.js

State

In React kun je gegevens bewaren door deze op te slaan in een speciale variabele, de state variabele. Om deze te gebruiken is het makkelijker om onze functie om te zetten in een class. Een class heeft in React twee belangrijke methods (functies); constructor() en render().

De constructor wordt eenmalig aan geroepen en hierin kun je state variabelen initialiseren.

In de render() method bepaald de output gemaakt. Je kunt dit vergelijken met de view (van MVC).

We gaan de bovenstaande functie omzetten naar het object model van React.

```
import './App.css';

import React from 'react';

class App extends React.Component {

  constructor(props) {
    super(props);
  }

  getNumbers() {
    for (var i=0; i < 10; i++) {
      myString = myString + String(i) + ", ";
    }
  }

  render() {
    return (
      <div className="App">
        {myString}
      </div>
    );
  }

}

export default App;
```

Als je deze code runt dan werkt die niet :(Dat komt omdat myString geen state variabele is. myString is alleen bekend in getNumers() en niet daarbuiten!

In de volgende stap hebben we van myString een state variabele gemaakt.

In de constructor method (deze wordt één keer aangeroepen!) wordt de state variabele gedeclareerd en daarna word de method getNumbers() aangeroepen. Met hetNumbers wordt de myString gevuld.

```
import './App.css';

import React from 'react';

class App extends React.Component {

  constructor(props) {
    super(props);
    this.state = {
      myString: ''
    }
    this.getNumbers();
  }

  getNumbers() {
    for (var i=0; i < 10; i++) {
      this.state.myString = this.state.myString + i + ", ";
    }
  }

  render() {
    return (
      <div className="App">
        {this.state.myString}
      </div>
    );
  }

}

export default App;
```

Let op, je krijgt een waarschuwing/foutmelding. Voor nu is dat even nog niet belangrijk.

Voor de gevorderen: Je kunt de *setState* functie gebruiken (zoals het hoort), maar dan moet je de int i casten naar een string. Op de manier zoals in het voorbeeld hoeft dat niet.

Opdracht

Als je het voorbeeld goed hebt uitgewerkt dan worden de nummer 0 tot en met 9 afgedrukt.

Verander de code nu zodat je de nummer 1 to en met 10 afdrukt en dat de cijfers worden gescheiden door een spatie en druk jouw naam af op de regel onder de nummers.

De output ziet er dus bijvoorbeeld als volgt uit:

image-1661519211351.png

Inleveren

Een schermafdruk van jouw output.

--

Revision #26
Created 2021-08-07 18:31:47 UTC by Max
Updated 2022-08-27 11:35:23 UTC by Max