# React From The Ground

Lärare: <http://codewithtim.thinkific.com/>

Plattform: Udemy

## Introduktion

### React JS – populärt bibliotek

React JS är ett JavaScripts bibliotek. Inte ett ramverk. React är skapat av utvecklare på Facebook och Instagram.

Här är några orsaker till att React har blivit så populär så snabbt:

Att arbeta med DOM API är svårt. React ger i princip utvecklare möjligheten att arbeta med en virtuell webbläsare som är mer användbar än den riktiga webbläsaren. Reacts virtuella webbläsare fungerar som en agent mellan utvecklaren och den verkliga webbläsaren.

React möjliggör för utvecklare att deklarera sina användargränssnitt och modellera tillståndet för dessa gränssnitt. Detta betyder istället för att komma överens med åtgärder för att beskriva transaktioner på gränssnitt, utvecklare beskriver bara gränssnitten när det gäller ett slutligt tillstånd (som en funktion). När transaktioner händer med det tillståndet, tar React hand om uppdatering av användargränssnitt baserade på det.

React är bara JavaScript, det är ett mycket litet API att lära sig, bara några funktioner och hur man använder dem. Därefter är dina JavaScript-färdigheter det som gör dig till en bättre React-utvecklare. Det finns inga hinder för tillträde. En JavaScript-utvecklare kan bli en produktiv React-utvecklare om några timmar. [https://medium.freecodecamp.org/yes-react-is-taking-over-front-end-development-the-question-is-why-40837af8ab76?fbclid=IwAR3eUIcqYxcMy0lvYNgA99FkiBQmQ7Dn0B5HF-vml6WB7AZyz3ZhsHSUk-Y]

## M **V (View)** C

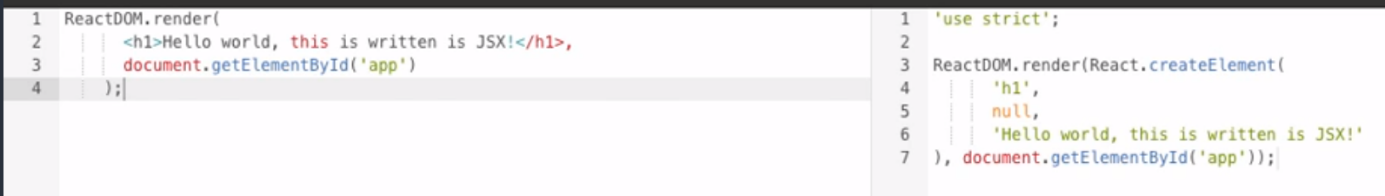
* React, används mest som View i ett MVS ramverk.
* React is used to build the user interface.
* The main goal of React is to build your user interface **simple** and to make your components **reusable**.
  + You can feel it when you start building.
* Problem with large applications where data changes over time.
* Build to look exactly what you want with the data you give to it.

## React === Simple

Reacts ”officiella definition” säger att det är ett JavaScript-bibliotek för att bygga användargränssnitt. Det är viktigt att förstå de två olika delarna av denna definition:

React är ett JavaScript-bibliotek. Det är inte en ram. Det är inte en komplett lösning och vi behöver ofta använda flera bibliotek med React för att skapa en lösning. React antar inte någonting om de andra delarna i någon fullständig lösning. Det fokuserar på bara en sak, och på att göra den där väldigt bra.

Det som React gör riktigt bra är den andra delen av definitionen: Building User Interfaces. Ett användargränssnitt är allt vi lägger framför användarna för att få dem att interagera med en maskin. Användargränssnitt är överallt, från de enkla knapparna på en mikrovågsugn till instrumentbrädan i en rymdfärja. Om enheten som vi försöker gränssnitt kan förstå JavaScript kan vi använda React för att beskriva ett användargränssnitt för det. [https://medium.freecodecamp.org/yes-react-is-taking-over-front-end-development-the-question-is-why-40837af8ab76?fbclid=IwAR3eUIcqYxcMy0lvYNgA99FkiBQmQ7Dn0B5HF-vml6WB7AZyz3ZhsHSUk-Y]

* As a library you be able to express exactly how your app should look.
* Manage all your ui updates when your data is changing.
* React is about Reusable components.
  + Work with reusable components = less, easy to test and more maintainable code.
  + Allows you to build big websites in less time.
* Declarative Programming
* JSX -> Java script extension
  + Allows you to write JavaScript in a cleaner easier way.
  + Java script syntax extension
    - A bit like XML
  + JSX VS JS
* Great performance
  + Uses a virtual DOM witch makes it very fast.
    - The virtual DOM keeps a representation of your current state and next state and only updates what´s need to be updated.
* Maintained and backed by Facebook, unlike Angular 1.
* React Native -> Mobile apps for IOS, Android and Windows.

Websites who uses React

* Facebook
* Uber
* Instagram

## Getting Setup

* Text Editor. Ex **Atom**, Visual Studio
  + Atom
    - https://atom.io/
* Web browser, Using chorme

Getting setup with node.js and webpack and that’s for building and making real projects with React.

* In the previous code with only included React and Babel libraries as script tags.
  + Witch is great if you wanna get up and running fast with React. But its not great when you wanna build projects.
  + We will not either write the JS it a html file. It just good for testing it.
* Node and Web Pack will be used too help us include scripts.
* Node Js allows us to run javascript on the computer or on the server. So rather than being in the browser you can run it as a backend language as well.
  + - <https://www.google.com/chrome/>
* Q/A
  + [www.udemy.com/draft](http://www.udemy.com/draft)
    - if you need help man
    - <https://twitter.com/codewithtim>
    - http://codewithtim.thinkific.com/
* Bable -> <https://cdnjs.com/libraries/babel-core>
  + For start example JSX Programming.
  + React need to know that you are using JSX

### Required scripts

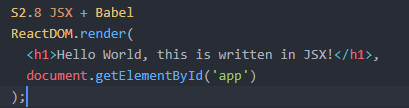
* CDN
  + <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.7/react.js"></script>
  + <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.7/react-dom.js"></script>

## Hello World

Första test med React för att se hur man skapar ett element, med olika parametrar som man via ”ReactDOM.render” renderar fram på webbsidan.

## JSX

2.7/8 🡪 JavaScript Syntax extension

* Its recommended to wright the JavaScript in the from of JSX.
  + Don’t have to use it.
  + Its more concise and easier to use when you are dealing with large tree structures.
  + Easier to read and understand.
* Looks similar to XML
* Written by Facebook
  + To use it in the example you need to include babel, where its helpful to transform JSX in the browser.
    - Babel-core 5.8.2 is used in the tutorial
      * https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.20/browser.min.js
    - <https://cdnjs.cloudflare.com/ajax/libs/babel-core/6.1.19/browser.min.js>
* On babeljs.io you can look and see the transformation going on in babel.

## Skapa en klass med React

På bilden skapas en klass som en variabel i med Reacts bibliotek. Detta blir en slags är en React komponent som gör en klass och innehåller ett specifikt objekt. Objektet har en renderings metod som är en slags funktion som returnerar JSX. För att få in komponenten i React använder man ReactDOM. ReactDOM tar två argument vår React komponent vi skapat och sen vart vi vill att den ska.

## Variabler

Let is used to define variables in ES6. Its similar to var but has a few difference. Let removes the problems that comes with var, when it comes to scope.

Scope is the area of a program in witch you can access a variable or where its legal to use.

The keyword **let** gives us block scope on our variables. Giltigt I sitt block endast. Block = {}

Let gör det inte tillgängligt längre upp i kedjan men gör sig tillgängligt till allt under den i scopet.

## Properties

It is the second argument that’s passed in the React.createElement component. 🡪

* Using Props we can pass values or data into our components.
  + Simple to do
  + Using {} allows us to render things in JSX
    - Witch is useful
* Använd this.props
  + this.props.(ditt värde)
    - ex this.props.name
  + This representerar här själva komponenten man är inuti.
* Man kan skicka med hur många properties man vill.

Vi skapar här en React komponent som en variabel vid namn ”SecondComponent” med React.createClass som är en funktion som tar ett specifikt objekt. Den behöver en rendering metod som returnerar något, tex innehåll. Som man sen renderar ut via ReactDOM.

Man kan skapa default properties. Så om man det värdet saknas så ställ det in till ett default.

* Det gör man med getDefaultProps
  + Som är en funktion som returnerar ett objekt.
    - Detta objekt ska innehålla nyckeln till det objekt som man vill specificera.

## Get Started With React!

1. Install Node.JS
   1. node -v 🡪 To check version in CMD.
2. Initialize the project
   * 1. Project.json 🡪 Defines the project. Name and dependencies etc. Kod för att kunna köra projektet.
     2. It’s says what we need to run the project.
3. Initialize out node repository. Our project directory.
   1. npm 🡪 node package manager + init
      1. npm init 🡪 I project mappen för att skapa.
4. Install React to the project
   1. npm install react --save
      1. --save adds dependencies to our project.json
   2. npm install react-dom --save
   3. npm install webpack –save
      1. Makes it easier with script tags. Only need one with WebPack who bundels them together.
   4. npm install webpack-dev-server --save
      1. This will spin up a server for us so that we can view our code in the browser.
   5. npm install babel-core --save
   6. npm install babel-loader --save
   7. npm install babel-preset-es2015 --save
   8. npm install babel-preset-react –save
   9. npm install babel-preset-stage-1 –save

* With a cloned project you just write
  + npm install to install all the modules in the project.json.

1. Config WebPack = Create a js file, named webpack.config.js
   1. In this file we tell Webpack what it is going to to.
   2. Web Pack will use Babel to transform our code again to say which file we want to load. This is server side javascript.
   3. Define an entry in code.
   4. Define an output
      1. Specifies a file to put everything in.
2. Create a src (source) folder with a jsx file in called index.
   1. type nul > src/index.jsx
   2. This is going to be where we writhe our REACT code.
3. Create .babelrc file including the packages we are going to use

{

"presets": ["react", "es2015", "stage-1"]

}

1. Create Index.html file
   1. Tip: With emmet package you just type ! + [tab] for fast template of html!! :D
2. Define our start point for node
   1. In our package.json in the scripts section.
   2. When we run its gonna find the node modules.

* Create React Apps with NO configuration
  + Get so much faster and is easier to start up a new React Project.
    - Usuallt it takes quite a lot to get from nothing to a Hello World app, that is not “React”.
      * Ex Web Pack, Node etc.

<https://reactjs.org/docs/create-a-new-react-app.html>

<https://facebook.github.io/create-react-app/>

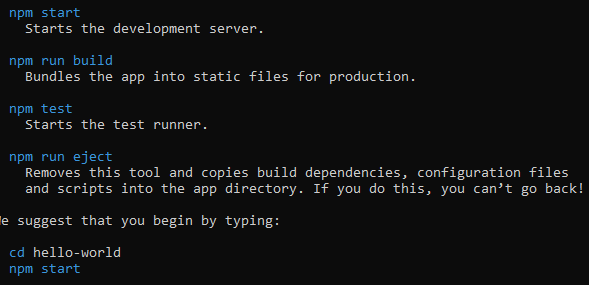
* Npm package to quickly create a React app.
  + It gets the Web Pack config for you and gets all you need.
  + RUN for global installation first time:
    - npm install -g create-react-app

To create a React app using this tool:

* create-react-app hello-world
  + create project in the folder your at.

Has everything you need for the React Application.

Contains more than we used in previous projects.



npm start 🡪 Start the app

npm run build 🡪 Compiles the project nice. Manly used if you are posting it on a web server.

npm run eject 🡪 Updates everything and imports dependencies.

## Källförteckning

[1] – Udemy Course

[2] – <https://github.com/trending/javascript>

[3] - React - <https://reactjs.org/>

[4] - React: Making faster, smoother UIs for data-driven Web apps, Paul Krill, 2014 -https://www.infoworld.com/article/2608181/javascript/react--making-faster--smoother-uis-for-data-driven-web-apps.html

[5] – Redux - <https://redux.js.org/>

[6] – ES6, <https://es6.io/>

[7] – Node JS, <https://nodejs.org/en/>

[8] – Atom, <https://atom.io/>

[9] – React create App

[10] - <https://react-bootstrap.github.io/>

[11] - <https://react-bootstrap.github.io/getting-started/introduction>

## React Bootstrap

9.32 v2

React Bootstrap 🡪 <https://react-bootstrap.github.io/>

* + Site give you documentation about how to use components.
* React Bootstrap 🡪 taking the CSS library and converting it to React components.
* Use basic bootstrap components in React.
  + Requires install in project
    - npm install react-bootstrap –save
* Need bootstrap cdn in index-html header.

<!-- Latest compiled and minified CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

React Bootstrap Pagination Component

* To help us with pagination
* https://react-bootstrap.github.io/components/pagination/

Efter vårt möte hos er har jag tänkt igenom vad vi diskuterade.

## State

* Witch is a collection of dynamic data that is managed by our components self.
  + To do this we start by defining a component. With var component with React.createClass.
  + Se kod
    - I denna kod så ändrar vi till olika states beroende om vi skriver i inputen.
      * Detta görs genom att använda sig utav events.
        + Ex onChange, onClick.
        + Sen skickar vi in en funktion till detta event.
  + State är en kollektion av data som sköts av komponenten själv.
    - Man bestämmer det initiala statet och sedan så uppdaterar vi statet.
      * Vi uppdaterar statet genom events.
        + Vi lägger till event där vi vill ha de i vår renderings funktion.
* We can say exactly how we want the state to look at any point.

4.13 State Challenge. Se kod 4.13

getInitialState == Att sätta eller hämta ett värde när sidan laddas. För att sen vidare kunna manipulera värdet med funktioner. Returnerar värden.

4.15 Användbart när man vill undvika att hårdkoda in värden.

5. React is about reusable components.

Kolla om videon och skriv han säger bra saker.

## Component Life Cycle

6. Component Life Cycle of react component

Whenever a component gets render to the dom. Some things happen before its get rendering and after its rendering. And these are known as the component’s life cycle and its methods. Ehan a component is added to the DOM. It´s called mounting and when its removes from the DOM it’s called unmounting.

* When it rendered to the DOM its mounted to the DOM and when its taken out its unmounted.

Kod s6

Koll om filmerna och gå igenom/skriv ner från filmer. Funktioner och tänk.

S7 ES6

Nytt sätt att skriva JavaScript.

* Behöver WebPack eller Babel för att kunna skriva det. Och för att webbläsaren ska kunna tolka det.
* Man behöver inte använda detta. Men många utvecklare tycker om att använda detta.

## Create React App

9.30 Create React App