

# 02 - React



# Web Apps

# Wat maakt een web app een web app?

- veel interactie op één 'pagina'
  - ◆ SPA: single page application
- client side templating
- snelle (re)rendering van views
- data uit API's (AJAX calls)
- offline mogelijkheden
- controle over URLs
- data relaties & complexiteit
- authenticatie

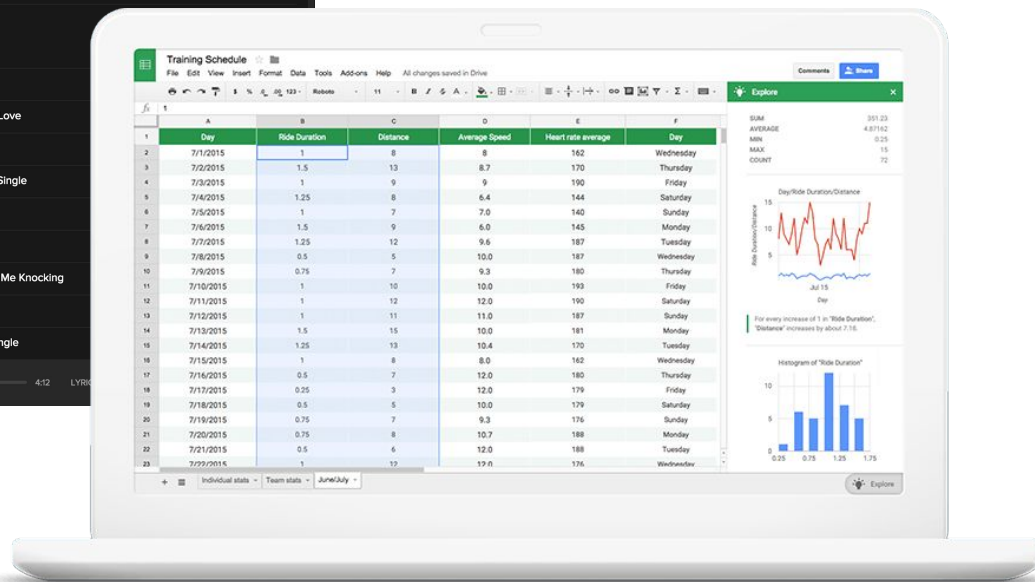
# Web Apps - werking

- 1 keer alle HTML / CSS / JavaScript inladen
- Alle interactie / views afhandelen via JavaScript
- Data via API

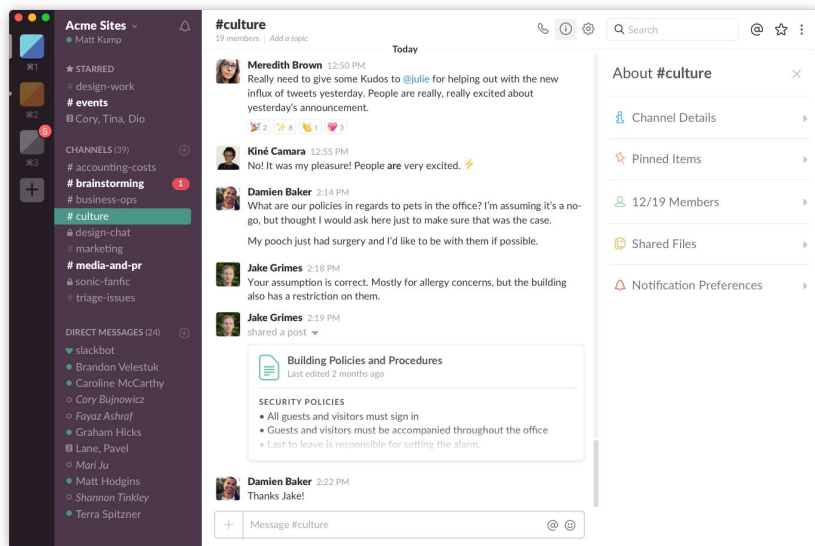
## W

The screenshot shows the Spotify desktop application. At the top is a search bar with the text "Search". Below it are navigation links: MAIN, Browse, Activity, Radio, YOUR MUSIC, Songs, Albums, Artists, Local Files, and PLAYLISTS. The main area features a "Discover Weekly" playlist card with a woman's portrait, the title "Discover Weekly", and the description "Your weekly mix of discoveries and deep cuts, picked just for you. Updated every Monday, so save your favourites!". It also says "Created by: Spotify · 30 songs, 2 hr 04 min". There are buttons for "PLAY" and "FOLLOWING". To the right, there's a "FOLLOWERS" section showing "1". Below the playlist card is a table with columns SONG, ARTIST, and ALBUM. The table lists several songs from the Discover Weekly playlist. On the far right, a portion of another application window titled "Training Schedule" is visible, showing a table with columns Day, Ride Duration, and other metrics.

	SONG	ARTIST	ALBUM
+	All My Friends	Dylan Kelly	All My Friends
+	Can't Deny My Love	Brandon Flowers	Can't Deny My Love
+	1000	Ben Khan	1000
+	Fly (FKJ Remix)	June Marleezy	Fly (Remixes) - Single
+	Eventually	Tame Impala	Eventually
+	Abrasive	Retatat	Abrasive
+	Can't You Hear Me Knocking - Alternate Ver...	The Rolling Stones	Can't You Hear Me Knocking
+	OKAGA, CA	Tyler, The Creator	Cherry Bomb
+	That's Love	Oddisee	That's Love - Single



# Web Apps



# Aan de slag, met vanillaJS...

<http://vanilla-js.com>

- veel interactie op één “pagina” (**JavaScript, ES6**)
- client side templating (**ES6 template strings**)
- snelle (re)rendering van views (**euh...?**)
- data uit API's (AJAX calls) (**fetch API**)
- offline mogelijkheden (**Web storage**)
- controle over URLs (**window.history**)
- data relaties & complexiteit (**MVC**)
- authenticatie (**JWT, OAuth**)







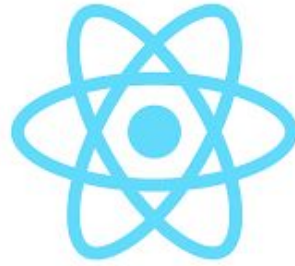
**No worries**



**Let's use a framework**

# JS Frameworks

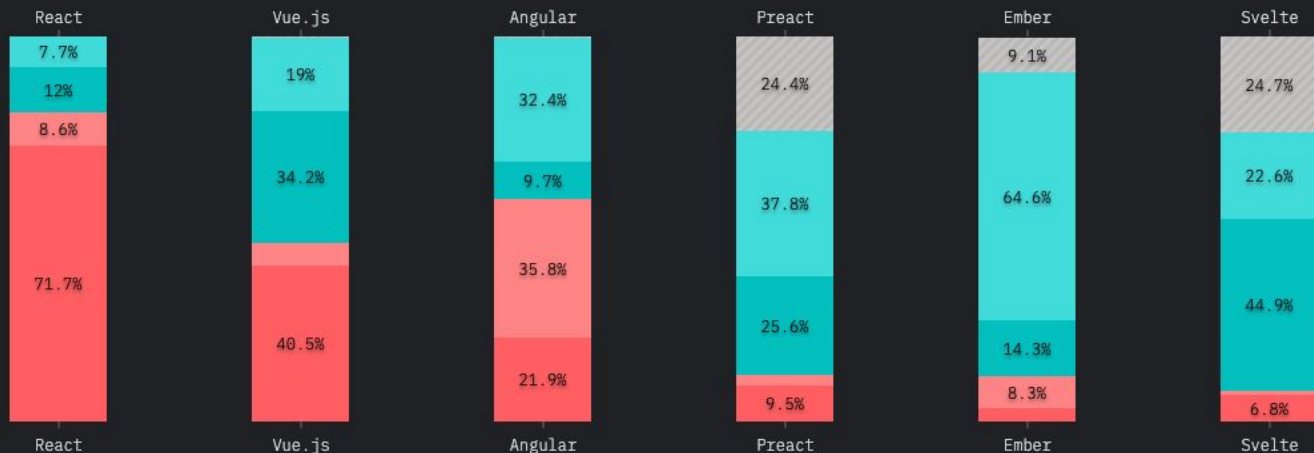
- Angular
- React
- Ember
- Vue
- Svelte
- Flutter
- Backbone
- Meteor



## Category Overview

[Export](#)[Share](#)[Percents](#)[Count](#)

Overview of opinions on the technologies surveyed. Darker segments represent positive opinions, while lighter segments correspond to negative sentiment.



■ I've USED it before, and WOULD use it again

■ I've USED it before, and would NOT use it again


















































■ I've HEARD of it, and WOULD like to learn it

■ I've HEARD of it, and am NOT interested


















































■ I've never heard of it

[https://2019.stateofjs.com/front-end-frameworks/#front\\_end\\_frameworks\\_section\\_overview](https://2019.stateofjs.com/front-end-frameworks/#front_end_frameworks_section_overview)

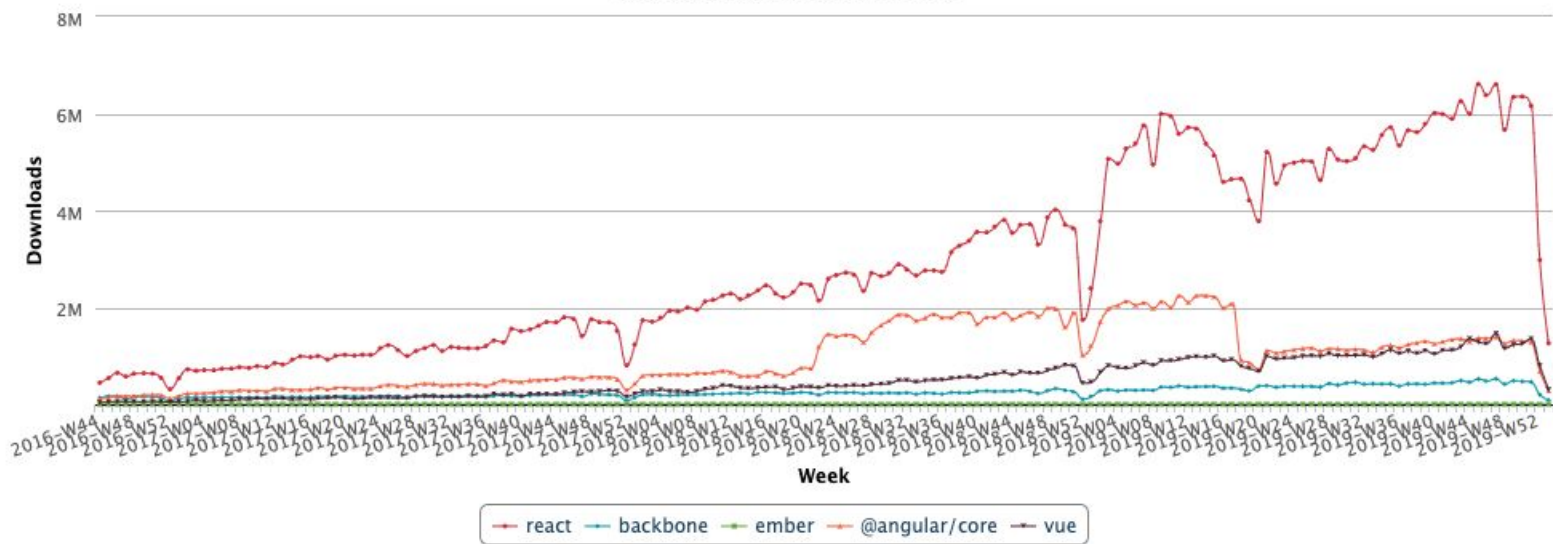


 <a href="#">vuejs / vue</a>	 Sponsor	 Used by ▾ 1.2m	 Watch ▾ 6k	 Star 155k	 Fork 23.3k
 <a href="#">facebook / react</a>		 Used by ▾ 2.9m	 Watch ▾ 6.7k	 Star 142k	 Fork 27.2k
 <a href="#">flutter / flutter</a>			 Watch ▾ 2.7k	 Star 84.1k	 Fork 10.8k
 <a href="#">angular / angular</a>			 Watch ▾ 3.2k	 Star 56.1k	 Fork 15.4k
 <a href="#">meteor / meteor</a>			 Watch ▾ 1.8k	 Star 41.5k	 Fork 5.1k
 <a href="#">sveltejs / svelte</a>		 Used by ▾ 9.3k	 Watch ▾ 562	 Star 29.1k	 Fork 1.3k
 <a href="#">jashkenas / backbone</a>		 Used by ▾ 36.3k	 Watch ▾ 1.4k	 Star 27.6k	 Fork 5.7k
 <a href="#">preactjs / preact</a>	 Sponsor	 Used by ▾ 18.2k	 Watch ▾ 442	 Star 25.1k	 Fork 1.3k
 <a href="#">Polymer / polymer</a>		 Used by ▾ 6.3k	 Watch ▾ 1k	 Star 21.3k	 Fork 2k
 <a href="#">emberjs / ember.js</a>		 Used by ▾ 19.7k	 Watch ▾ 1k	 Star 21.3k	 Fork 4.2k



 facebook / react		 Used by ▾	2.9m	 Watch ▾	6.7k	 Star	142k	 Fork	27.2k
 vuejs / vue	 Sponsor	 Used by ▾	1.2m	 Watch ▾	6k	 Star	155k	 Fork	23.3k
 jashkenas / backbone		 Used by ▾	36.3k	 Watch ▾	1.4k	 Star	27.6k	 Fork	5.7k
 emberjs / ember.js		 Used by ▾	19.7k	 Watch ▾	1k	 Star	21.3k	 Fork	4.2k
 preactjs / preact	 Sponsor	 Used by ▾	18.2k	 Watch ▾	442	 Star	25.1k	 Fork	1.3k
 sveltejs / svelte		 Used by ▾	9.3k	 Watch ▾	562	 Star	29.1k	 Fork	1.3k
 Polymer / polymer		 Used by ▾	6.3k	 Watch ▾	1k	 Star	21.3k	 Fork	2k
 flutter / flutter				 Watch ▾	2.7k	 Star	84.1k	 Fork	10.8k
 angular / angular				 Watch ▾	3.2k	 Star	56.1k	 Fork	15.4k
 meteor / meteor				 Watch ▾	1.8k	 Star	41.5k	 Fork	5.1k

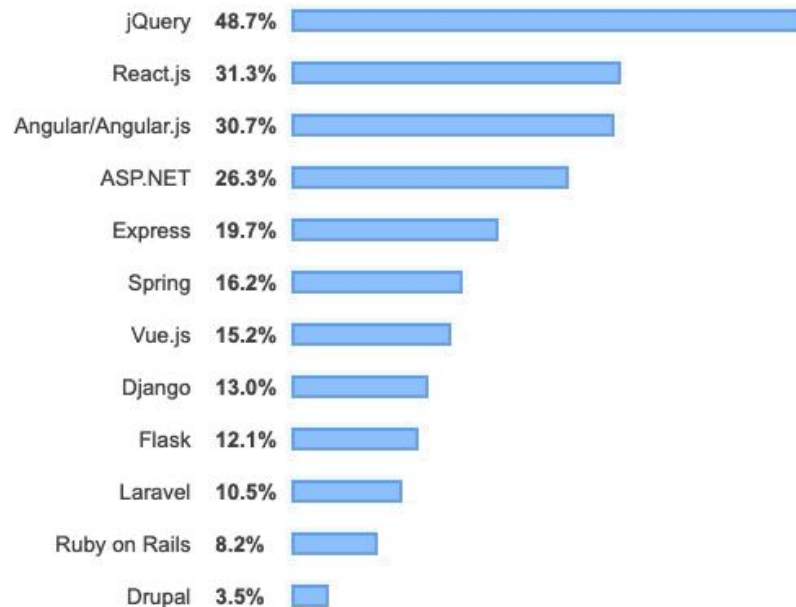
# npm downloads per week



## Web Frameworks

All Respondents

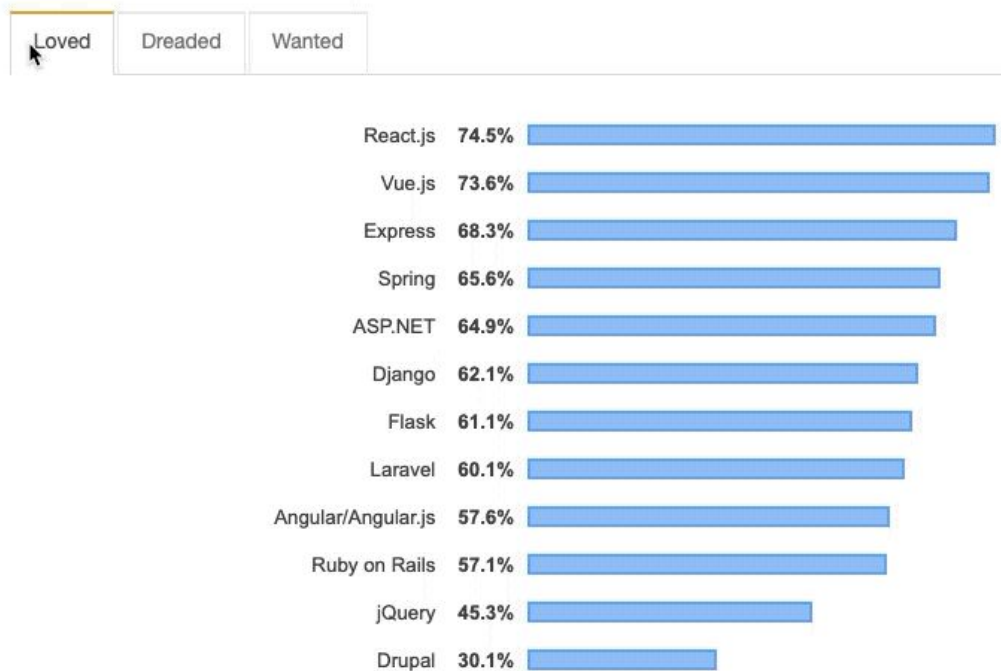
Professional Developers



63,585 responses; select all that apply



## Most Loved, Dreaded, and Wanted Web Frameworks



*% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it*

React.js and Vue.js are both the most loved and most wanted web frameworks by developers, while Drupal and jQuery are most dreaded.

## Top JavaScript frameworks on desktop

Web Almanac 2019: JavaScript

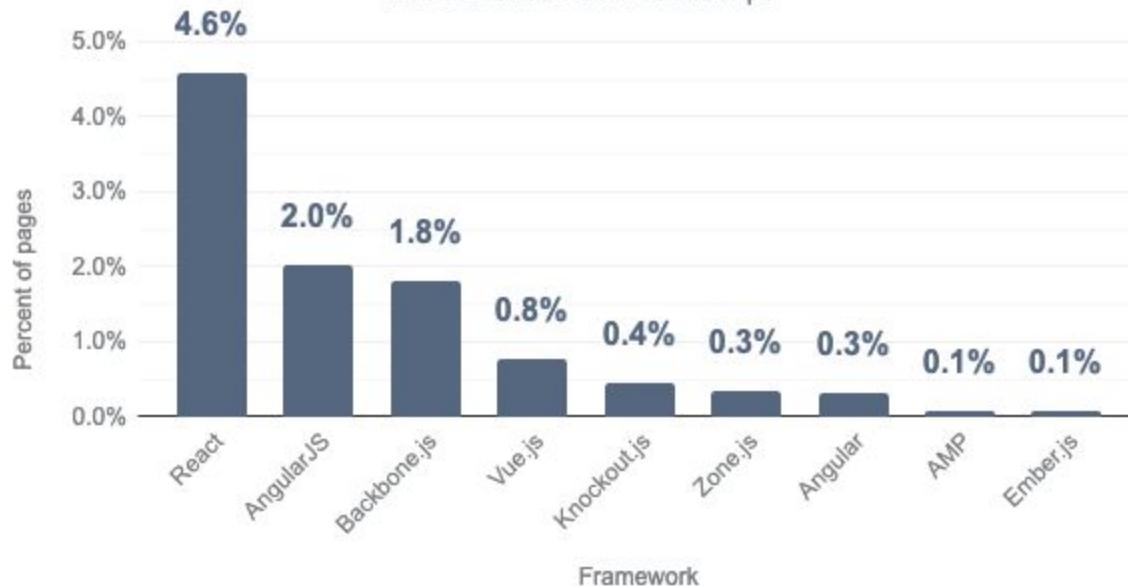
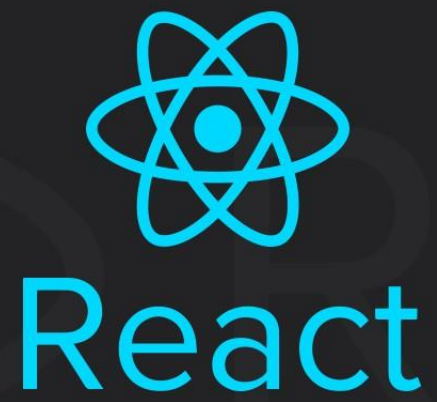


Figure 12. Most frequently used frameworks on desktop.







**You Might Not Need React!**



**Niet alles moet in React**



**React**

# React

## → Ontstaan bij Facebook (2011)

◆ Open sourced in 2013

### Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

### Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.


Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

### Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using [React Native](#).





**Declara-wadde?**

# Declaratief?

## Declaratief

→ **Wat** er moet gebeuren

Ga naar Howest, Campus Kortrijk Weide,  
Gebouw B

## Imperatief

→ **Hoe** het moet er gebeuren

### Budafabriek

Dam 2, 8500 Kortrijk

↑ Loop naar het zuidwesten op de Dam, richting de  
Budastraat

84 m

↪ Sla rechtsaf naar de Overleiestraat

50 m

↪ Sla linksaf naar de Fabriekskaaï

160 m

↑ Weg vervolgen naar de Nijverheidskaai

240 m

↪ Sla rechtsaf richting de Noordstraat

60 m

↪ Sla linksaf naar de Noordstraat

38 m

↑ Weg vervolgen naar de Beheerstraat

48 m

↪ Sla rechtsaf naar de Havenkaai

700 m

↪ Sla linksaf naar de Graaf Karel de Goedelaan

34 m

↪ Sla rechtsaf

99 m

### Howest, Campus Kortrijk Weide - Gebouw B

Graaf Karel de Goedelaan 5, 8500 Kortrijk

# Declaratief?

Declaratief

→ **Wat** er moet gebeuren

```
function double (arr) {  
  return arr.map((item) => item * 2)  
}
```

Imperatief

→ **Hoe** het moet gebeuren

```
function double (arr) {  
  let results = []  
  for (let i = 0; i < arr.length; i++){  
    results.push(arr[i] * 2)  
  }  
  return results  
}
```

<https://tylermcginnis.com/imperative-vs-declarative-programming/>



**Data driven**

# Data driven

- Data aangepast? => View opnieuw renderen
  - ◆ Héél belangrijk concept

~~→ Document.querySelector()~~



**JSX**

```
<h1>Hello World!</h1>
```

```
const hello = <h1>Hello world!</h1>;
```



# JSX

- Geen String of HTML maar JavaScript
- React element
- Babel maakt van JSX een JS object
- Attributen in camelCase
  - ◆ readonly => readOnly
- class => reserved keyword
  - ◆ className gebruiken

<https://reactjs.org/docs/jsx-in-depth.html>

```
const hello = (  
  <h1>  
    Hello World!  
  </h1>  
);
```

```
const hello = React.createElement(  
  'h1',  
  'Hello World!',  
);
```

```
const hello = {  
  type: 'h1',  
  props: {  
    children: 'Hello World!'  
  }  
};
```

# The power of the { }

→ Je kan de { } gebruiken om expressies uit te voeren

◆ lees: javaScript code uit te voeren

```
<p>Ik ben Jantje</p>
```

```
<p>Ik ben {naam}</p>
```

```
<p>Ik ben {anoniem ? "anoniem" : naam}</p>
```

```
<p>Ik ben {getName()}</p>
```

<https://codesandbox.io/s/jsx-expressions-sd1ly>

# Expressies in JSX

→ Items in een array weergeven

```
{lijst.map(item => <li>item</li>)}
```

⚠ Warning: Each child in a list should have a unique "key" prop.

→ Key attribuut (prop) toevoegen

```
{lijst.map(item => <li key={item}>item</li>)}
```

<https://codesandbox.io/s/jsx-expressions-sd1ly>

# Expressies in JSX

→ If() en for loops kunnen niet

◆ ~~{if(condition){...}}~~



# Elements

# Element

- Kleinste deeltje van een React App
- Klein, eenvoudig JavaScript Object
- $\neq$  HTML element

```
const element = <h1>Hello World!</h1>;
```

# Hoe krijgen we een element op de pagina te zien?

- index.html
- script.js met daarin:

```
<h1>Hello World!</h1>
```

# Hoe krijgen we een element op de pagina te zien?

- Een element nodig op de pagina waarin het terecht moet komen
  - ◆ root DOM node

```
<h1>Hello World!</h1>  
document.querySelector(`#root`)
```



# Hoe krijgen we een element op de pagina te zien?

→ React nodig

```
import React from 'react';
```

```
<h1>Hello World!</h1>  
document.querySelector(`#root`)
```

# Hoe kunnen we een element renderen?

→ React element renderen in de DOM dmv ReactDOM

```
import React from 'react';
import ReactDOM from 'react-dom';

ReactDOM.render(
  <h1>Hello World!</h1>,
  document.querySelector(`#root`)
);
```

# Hoe kunnen we een element renderen?

→ Proper!

```
import React from 'react';  
import {render} from 'react-dom';  
  
render(  
  <h1>Hello World!</h1>,  
  document.querySelector(`#root`)  
);
```

<https://codesandbox.io/s/0q7284xwyl>

# Fragment

→ Altijd 1 root-element retourneren

- ◆ Maar wat als je bv meerdere siblings wil retourneren?
- ◆ Fragment tag `<></>` gebruiken



```
return (  
  <p>Meer info volgt</p>  
  <button>Klik hier om in te schrijven</button>  
);
```



```
return (  
  <>  
    <p>Meer info volgt</p>  
    <button>Klik hier om in te schrijven</button>  
  </>  
);
```

Adjacent JSX elements must be wrapped in an enclosing tag. Did you want a JSX fragment `<>...</>`?



# Components

# En wat is een component dan?

→ Voer voor volgende week

→ Wat moet je voor deze week weten: (om het zo eenvoudig mogelijk te houden)

- ◆ Een functie die JSX element(en) terug geeft

```
const App = () => <p>Ik ben eigenlijk al een component</p>
```

- ◆ *Roep je aan als een tag*

```
ReactDOM.render(<App/>, document.getElementById("root"));
```

A large, cyan-colored parallelogram is positioned diagonally across the center of the slide. The word "Events" is written in bold black text across the middle of this shape.

# Events

# Events afhandelen

→ Een beetje zoals in DEV2

## Luisteren naar events

~~`<div class="me" onclick="handleClickMe()">Click Me</div>`~~

~~`document.querySelector('.me').onclick = handleClickMe;`~~

`document.querySelector('.me').addEventListener('click', handleClickMe);`



# Events afhandelen

→ Maar niet helemaal...

Het is op deze manier

## Luisteren naar events

~~`<div class="me" onclick="handleClickMe()">Click Me</div>`~~

~~`document.querySelector('.me').onclick = handleClickMe;`~~

`document.querySelector('.me').addEventListener('click', handleClickMe);`

# Events afhandelen

- camelCase
- {} gebruiken om de functie door te geven (geen " ")

```
const handleClickButton = e => {  
  console.log(e);  
}
```

```
return <button onClick={handleClickButton}>Hello</button>;
```



**MobX**

# MobX + React

- MobX ontwikkeld ifv React
  - Extra package voor nodig
    - ◆ `yarn add mobx-react-lite`
  - bye bye `autorun(() =>...)`
  - welcome `useObserver(() =>...)`
- ```
return useObserver(() => <p>Ik ben observed</p>)
```



# Controlled Components

# Controlled components

- *In HTML, form elements such as `<input>`, `<textarea>`, and `<select>` typically maintain their own state and update it based on user input. (\*)*
- React/MobX moet de 'single source of truth' zijn
  - ◆ State overnemen van form elements
- `value` + `onChange` attribuut gebruiken
  - ◆ `value`: observable
  - ◆ `onChange`: action oproepen
- Foutmelding indien je enkel 'value' gebruikt:
  - ◆ `Warning: Failed prop type: You provided a `value` prop to a form field without an `onChange` handler. This will render a read-only field. If the field should be mutable use `defaultValue`. Otherwise, set either `onChange` or `readOnly`.`

# CSS

→ Css file importeren

- ◆ `import "../styles.css";`

→ Css class aanspreken via className

- ◆ `<p className="user__name">Jantje</p>`

- ◆ `<p className={expression}>Jantje</p>`

→ Niks fancy eigenlijk (nu nog)

→ Inline styles kunnen ook

- ◆ `<p style={{ backgroundColor: "red" }}>Jantje</p>`



# Virtual DOM



# Virtual dom

- Virtuele versie van de DOM van je element
- (re)render -> zo effectief mogelijke update
- enkel updaten wat er moet geüpdatet worden
- optimalisatie / best mogelijke moment uitzoeken
- batched updates / RAF

**Hello, world!**

**It is 12:26:46 PM.**

```
Console Sources Network Timeline
▼ <div id="root">
  ▼ <div data-reactroot>
    <h1>Hello, world!</h1>
    ▼ <h2>
      <!-- react-text: 4 -->
      "It is "
      <!-- /react-text -->
      <!-- react-text: 5 -->
      "12:26:46 PM"
      <!-- /react-text -->
      <!-- react-text: 6 -->
      "."
      <!-- /react-text -->
    </h2>
  </div>
</div>
```



# Opdracht

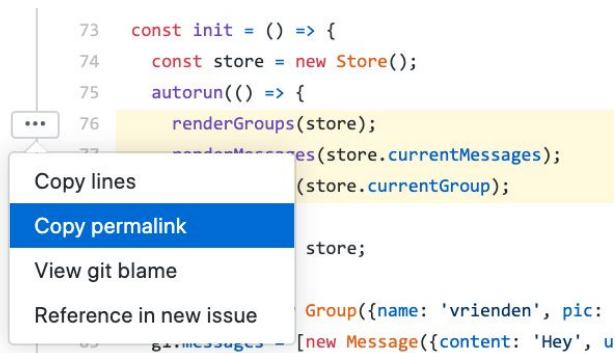
# Opdracht Week02

- Combineer jouw Model + Store met een React frontend
- Uiteraard met MobX
- Mag nog allemaal in 1 component
- Aanmaken/updaten...



# Hulp nodig? Tips voor issues

- Een issue heeft een titel + een description...
- Altijd handig als je kan tonen waar het mis gaat
  - ◆ Gebruik permalinks



A screenshot of a code editor with a context menu open over a code block. The code block contains the following JavaScript code:

```
73 const init = () => {  
74   const store = new Store();  
75   autorun(() => {  
76     renderGroups(store);  
77     renderMessages(store.currentMessages);  
78     (store.currentGroup);  
79   });  
80 }  
81  
82 Group({name: 'vrienden', pic:  
83 [new Message({content: 'Hey', u
```

The context menu has the following options:

- Copy lines
- Copy permalink
- View git blame
- Reference in new issue



A screenshot of a code preview window. The window has two tabs: "Write" and "Preview". The "Preview" tab is selected. The preview shows the following code block:

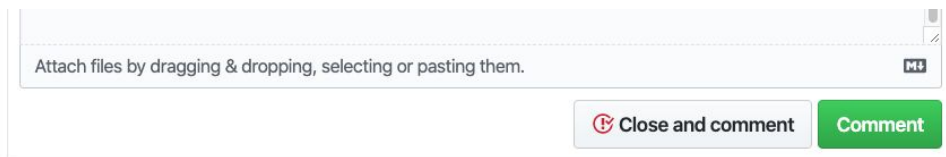
```
DEV4-2020/02 - Mobx/docent/04 - ThatsApp groups/src/index.js  
Lines 76 to 78 in 79b4aa1  
  
76   renderGroups(store);  
77   renderMessages(store.currentMessages);  
78   renderHeader(store.currentGroup);
```

# Hulp nodig? Tips voor issues



→ Labelen met 'help wanted'

→ Als wij antwoorden, dan sluiten we de issue ook.

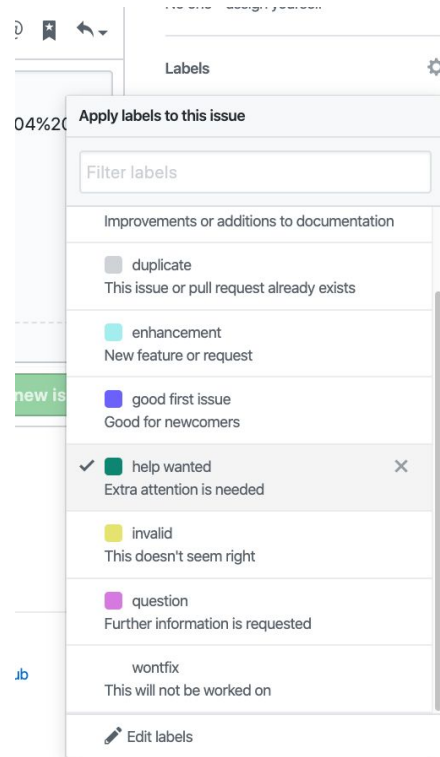
- ◆ Zo behouden wij overzicht
- ◆ Nog niet duidelijk?
  - Terug op replyen, dan is de issue weer open



Attach files by dragging & dropping, selecting or pasting them.

 Close and comment 

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).




Labels

Apply labels to this issue

Filter labels

- Improvements or additions to documentation
- ☐ duplicate  
This issue or pull request already exists
- ☐ enhancement  
New feature or request
- ☐ good first issue  
Good for newcomers
- ☒ help wanted  
Extra attention is needed
- ☐ invalid  
This doesn't seem right
- ☐ question  
Further information is requested
- ☐ wontfix  
This will not be worked on

 Edit labels



**Development** 

# Keep up

## → [Dev.to](#)

- ◆ Tip: mailinglist met top-post (instellen via profile -> notifications)

*DEV is a community of software developers getting together to help one another out. The software industry relies on collaboration and networked learning. We provide a place for that to happen.*



## → [Wes Bos](#)

- ◆ [🔥 tips op Twitter](#)
- ◆ [Syntax.fm](#)
- ◆ [BBQ Bonus](#)

