



Web advanced

Vue.js



**DE HOGESCHOOL
MET HET NETWERK**

Hogeschool PXL - Dep. PXL-IT - Elfde-Liniestraat 26 - B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Waarom Vue?

Opsplitsing tussen logica en presentatie.

Veel libraries, uitbreidingen (Vue-router, vuex state manager)

Virtual DOM

- Wijzigingen in DOM zijn TRAAG

The Virtual DOM

In React, for every **DOM object**, there is a corresponding “virtual DOM object.” A virtual DOM object is a *representation* of a DOM object, like a lightweight copy.

A virtual DOM object has the same properties as a real DOM object, but it lacks the real thing’s power to directly change what’s on the screen.

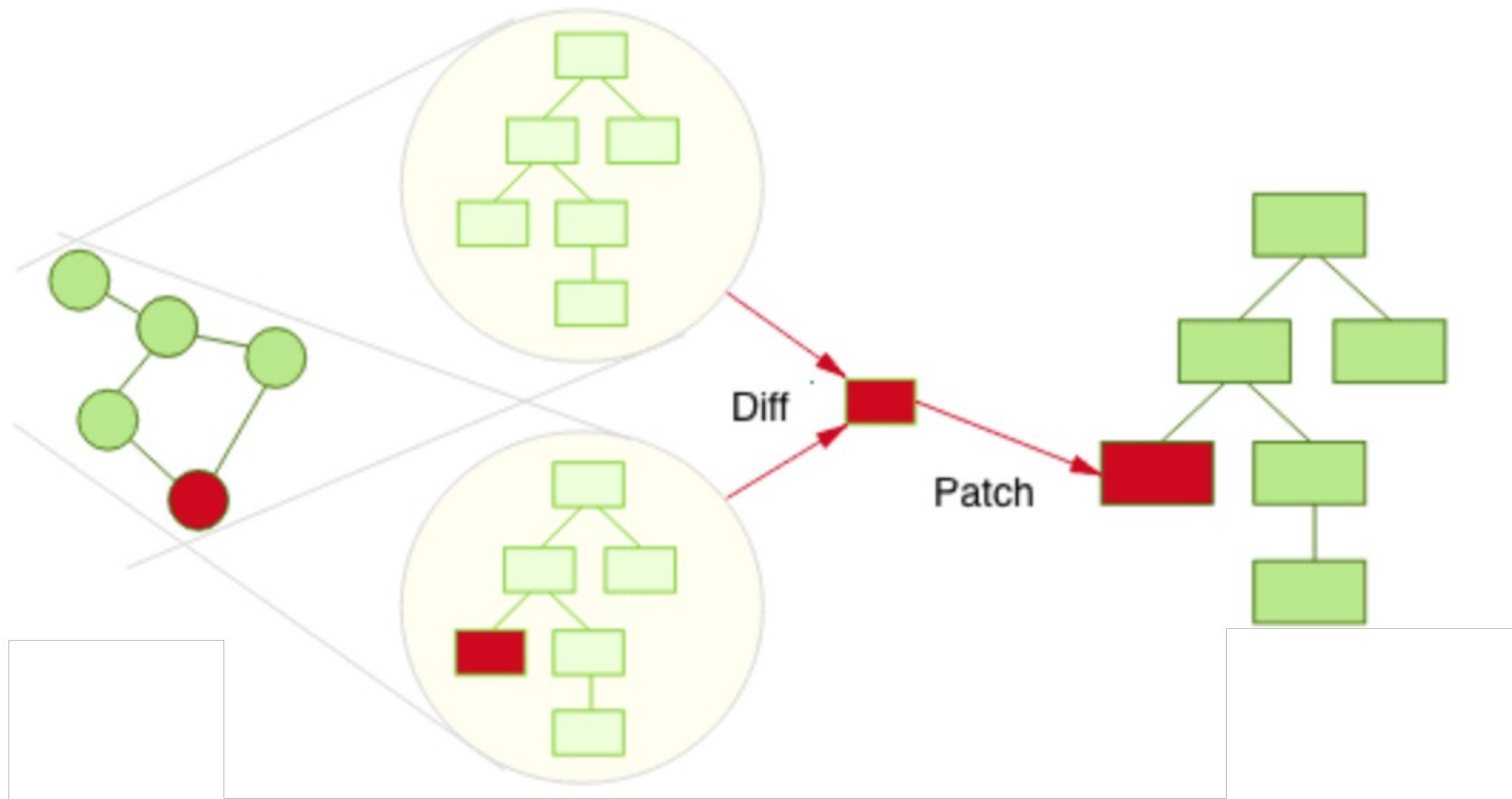
Manipulating the DOM is slow. Manipulating the virtual DOM is much faster, because nothing gets drawn onscreen. Think of manipulating the virtual DOM as editing a blueprint, as opposed to moving rooms in an actual house.



Model

Virtual DOM

DOM



Wijziging in Model?

Vorige toestand wordt vergeleken met nieuwe toestand (Diff)

Beperkt deel van de DOM wordt aangepast (Patch)

Templates & data

data: beschikbare gegevens

template: te tonen op het scherm

Binnen template:

- **directive**: logica die bepaalt hoe data getoond w.
(v-if, v-else, v-for)
- **interpolation**: print gegevens in een element
Mustache syntax: {{ name }}
- **binding**: print gegevens in een attribuut
(v-bind)

```

<!doctype html>
<html lang="en">
<head>
  <title>VUE</title>
</head>
<body>
<div id="app">
  <p v-if="isMorning">Good morning</p>
  <p v-else>Hello</p>
</div>
<script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></scr
ipt>
<script>
  let timeInHours = new Date().getHours();
  let app = new Vue({
    el: '#app',
    data: {
      isMorning: timeInHours < 12
    }
  });
</script>
</body>

```

data

constructor van Vue
arg = object; fields: el, data



Good morning

```
<!doctype html>
<html lang="en">
<head>
  <title>VUE</title>
</head>
<body>
```

template

```
<div id="app">
  <p v-if="isMorning">Good morning</p>
  <p v-else>Hello</p>
</div>
```

directives

```
<script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></scr
ipt>
```

```
<script>
  let timeInHours = new Date().getHours();
  let app = new Vue({
    el: '#app',
    data: {
      isMorning: timeInHours < 12
    }
  });
```

```
</script>
</body>
```



Good morning

The time is 9 : 31

<body>

template

<div id="app">

<p v-if="timeInHours < 12">Good morning</p>

<p v-else>Hello</p>

<p> The time is {{ timeInHours }}

{{ timeInMinutes }}</p>

interpolation

</div>

<script

src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

<script>

let app = new Vue({

el: '#app',

data: {

timeInHours: new Date().getHours(),

timeInMinutes: new Date().getMinutes()

}

});

</script>

</body>



- jan
- tim
- sofie

```
<body>
<div id="app">
  <ul>
    <li v-for="name in names">{{ name }}</li>
  </ul>
</div>
<script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script
>
<script>
  let app = new Vue({
    el: '#app',
    data: {
      names: ['jan', 'tim', 'sofie']
    }
  });
</script>
</body>
```


binding: plaats gegevens in een attribuut

vb4.html

Hello Tim



```
<div id="app">
  <p v-bind:style="myStyle">Hello {{ name }}</p>
  
</div>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
  let app = new Vue({
    el: '#app',
    data: {
      name: 'Tim',
      myStyle: 'color:red',
      myImage:
'https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Vue.js_Logo_2.svg/220px-
Vue.js_Logo_2.svg.png'
    }
  });
</script>
```

2 way-binding

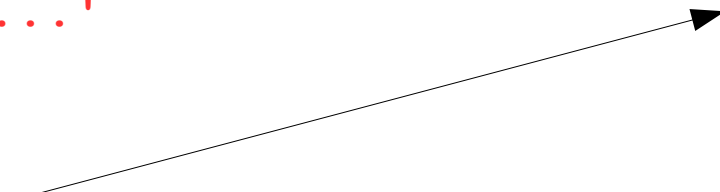
v-bind: one-way binding

wijziging in data wordt gemerkt in template

wijziging vanuit template wordt niet gemerkt in data

v-model: two-way binding

2 way-binding

```
...  
<h1>v-model</h1>  
<div id="app1">  
  <h1>message heeft waarde: {{ message }}</h1>  
  <input v-model:value="message" type="value"/>  
</div>  
...  
<script>  
  let app1 = new Vue({  
    el: '#app1',  
    data: {  
      message: '...'   
    }  
  });  
  app1.message='ok';  
  //wijziging van data wordt getoond in template  
</script>
```

vb5.html

v-model

message heeft waarde: ok

ok

v-bind

message heeft waarde: ok

ok

2 way-binding

```
...  
<h1>v-model</h1>  
<div id="app1">  
  <h1>message heeft waarde: {{ message }}</h1>  
  <input v-model:value="message" type="value"/>  
</div>  
...  
<script>  
  let app1 = new Vue({  
    el: '#app1',  
    data: {  
      message: '...'  
    }  
  });  
  app1.message='ok';  
  //wijziging van data wordt getoond in template  
</script>
```

vb5.html

v-model

message heeft waarde: nieuw!!!

nieuw!!!

v-bind

message heeft waarde: ok

nieuw???

Wijziging in template (nieuw!
als value textfield) wordt
doorgevoerd naar data

2 way-binding

```
<h1>v-bind</h1>
<div id="app2">
  <h1>message heeft waarde: {{ message }}</h1>
  <input v-bind:value="message" type="text"/>
</div>
<script>
  ...
  let app2 = new Vue({
    el: '#app2',
    data: {
      message: '...'
    }
  });
  app2.message='ok';
</script>
</body>
</html>
```

vb5.html

v-model

message heeft waarde: nieuw!!!

v-bind

message heeft waarde: ok

Wijziging in template wordt
niet doorgevoerd naar data

Methods

- J is for JAN
- T is for TIM
- S is for SOFIE

```
<div id="app">
  <ul>
    <li v-for="name in getUppercaseNames ()">
      {{ firstSymbolOnly(name) }} is for {{ name }}
    </li>
  </ul>
</div>
<script>
  let app = new Vue({
    el: '#app',
    data: {
      names: ['jan', 'tim', 'sofie']
    },
    methods: {
      getUppercaseNames () {
        return this.names.map( x => x.toUpperCase() );
      },
      firstSymbolOnly(text) {
        return text[0];
      }
    }
  });
```

Events

```

<div id="app">
  <ol>
    <li v-for="todoItem in todos">
      {{ todoItem }}
    </li>
  </ol>
  new todo: <input v-model="newTodo"/>
  <button v-on:click="addTodo">ADD</button>
</div>
<script>
  let app = new Vue({
    el: '#app',
    data: {
      todos: [],
      newTodo: ''
    },
    methods: {
      addTodo: function () {
        this.todos.push(this.newTodo);
      }
    }
  });

```

v-on methods

1. eat
2. sleep
3. ...

new todo:

Templates & props

vb8.html

Hello Tim

Child en parent-components

prop (ipv data)

gegevens doorgeven van parent naar child

vb8:

- parent-component
aangemaakt via constructor
- child-component
aangemaakt via Vue.component, 'intern' template

props: name, backgroundColor

opmerking:

backgroundColor (js) → background-color (template)




```
<div id="app">
  <greeter name="Tim" background-color="red">
  </greeter>
</div>

<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
  Vue.component('greeter', {
    props: ['backgroundColor', 'name'],
    computed: {
      computedStyle() {
        return `background-color: ${this.backgroundColor}`;
      }
    },
    template:
      '<div v-bind:style="computedStyle">Hello {{ name }}</div>'
  });

  let app = new Vue({
    el: '#app'
  });
</script>
```

parent component



```
<div id="app">
  <greeter name="Tim" background-color="red">
  </greeter>
</div>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
  Vue.component('greeter', {
    props: ['backgroundColor', 'name'],
    computed: {
      computedStyle() {
        return `background-color: ${this.backgroundColor}`;
      }
    },
    template:
      '<div v-bind:style="computedStyle">Hello {{ name }}</div>'
  });

  let app = new Vue({
    el: '#app'
  });
</script>
```

Binnen parent component wordt child-component greeter gebruikt, waarden voor de props name en background-color

```
<div id="app">
  <greeter name="Tim" background-color="red">
  </greeter>
</div>

<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
  Vue.component('greeter', {
    props: ['backgroundColor', 'name'],
    computed: {
      computedStyle() {
        return `background-color: ${this.backgroundColor}`;
      }
    },
    template:
      '<div v-bind:style="computedStyle">Hello {{ name }}</div>'
  });

  let app = new Vue({
    el: '#app'
  });
</script>
```

Via computed computedStyle wordt de string
background-color: red gegenereerd
<https://vuejs.org/v2/guide/computed.html>

Opsplitsen van componenten in .vue files

zie map vb8_herwerkt

Voer npm install uit om alle dependencies te installeren

npm run serve (compileert en start development server)



```
1 <template>
2   ...<div v-bind:style="computedStyle">Hello {{ name }}</div>
3 </template>
4 <script>
5 export default {
6   ...name: 'Greeter',
7   ...props: ['backgroundColor', 'name'],
8   ...computed: {
9     ...computedStyle() {
10      ...return `background-color: ${this.backgroundColor}`;
11     ...}
12   ...}
13 }
14 </script>
```

Fetch voorbeeld

```
<div id="app">
  <ul>
    <li v-for="person in persons">
      {{ person.name }}
    </li>
  </ul>
  <button v-on:click="fetchPersons">Get all persons</button>
  <span>{{ error }}</span>
</div>
```

- jan
- tim
- tim

Get all persons

```
<script>
```

```
  let app = new Vue({
    el: '#app',
    data: {
      persons: [],
      error: ''
    },
    methods: {
      fetchPersons: function () {
        let url = 'http://localhost:3000/persons/';
        this.persons=[];
        fetch(url)
          .then((response) => {
            if (response.status == 200) {
              return response.json();
            } else {
              throw `error with status ${response.status}`;
            }
          })
          .then((persons) => {
            this.persons = persons;
          })
          .catch((error) => {
            this.error = error;
          });
      }
    }
  });
```

vb9.html

- jan
- tim
- tim

Get all persons

Fetch voorbeeld

Get all persons

GET

5

tim

change

6

jan

change

7

sofie

change

PUT

Extra's

vuex

Data beschikbaar maken in meerdere componenten.
'Single source of truth'

Vue-router

Laat path (url) overeenkomen met een component

Vue cli