

# Oefening 1 – Vue essentials

Op het einde van deze oefening kan je

- Een vue app aanmaken met build tools
- Declarative rendering gebruiken via [text interpolation](#)
- 'v-bind' of ':' [directives](#) gebruiken
- User input en events opvangen met 'v-on' of '@' directives
- '[methods](#)' gebruiken om [events af te handelen](#)
- Het verschil aangeven tussen '[data properties](#)' en '[computed properties](#)'
- 'v-if' gebruiken voor [conditional rendering](#)
- 'v-for' gebruiken voor [list rendering](#)
- [Class binding](#) gebruiken
- Een HTML element aanspreken via [template refs](#)
- Een [watcher](#) gebruiken om te reageren op data changes

Extra

- [Vue transitions](#) gebruiken

## Doel

We bouwen een 'slotmachine'.

Klikken op 'PLAY!' zorgt dat de drie slots met een vast interval een andere emoji tonen. Elk slot 'draait' voor een willekeurige tijd.

Zolang de slots 'draaien' kan je de 'PLAY!' knop niet bedienen en verschijnt de tekst 'Rien ne va plus'.

Wanneer alle slots terug in rust zijn wordt automatisch gecontroleerd of je gewonnen hebt.

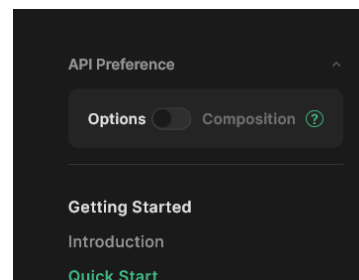
Winst of verlies wordt aangegeven m.b.v. css class binding.

## Opgelet!

We maken voor dit labo gebruik van de **options API**.

Zorg dat de vue documentatiesite goed is ingesteld!

Verwijder telkens het 'setup' attribute uit de script tags.



## Gegeven

```
<template>
  <main class="container">
    <div class="machine">
      <h1>Hello slotmachine</h1>
      <div class="slots">
        <div class="slot">
          <span>?</span>
        </div>
        <div class="slot">
          <span>?</span>
        </div>
        <div class="slot">
          <span>?</span>
        </div>
      </div>
      <button>PLAY!</button>
      <div lang="fr">Rien ne va plus</div>
    </div>
  </main>
</template>

<script>
const emoji = ["🎰", "🎲", "🎯", "🎳"];
export default {
  data() {
    return {};
  },
  computed: {},
  watch: {},
  methods: {},
};
</script>

<style lang="scss">
@import "../assets/base.css";
</style>
```

## Opdracht

### 1. Maak een vue app aan

- npm init vue@latest
  - geef een naam op 'vue-essentials'
    - dit wordt je project directory!
  - Kies 'no' voor alles **behalve prettier en ESLint**
  - Navigeer naar ./vue-essentials
  - npm i
- Het startproject bevat véél te veel!
  - Wis alle components
  - Maak de template, script en style tags leeg in App.vue

- Haal alles uit base.css wat je niet verstaat, laat die import desgewenst staan tussen de style tags.
- Verwijder het 'setup' attribute uit de script tags. Dat is voor de composition API, wij gebruiken de options API
- We gaan altijd gebruik maken van SCSS in plaats van CSS.
  - npm install --save-dev sass
  - voeg een attribute toe aan je style tag: lang="scss"
  - vanaf nu kan je selectors nesten etc. als je je daartoe geroepen voelt
- Maak een nieuwe gitlab repo: **2122WMFS-Vue-essentials**
- In je project:
  - git init
  - git add -A
  - git commit -m 'Created Vue App'
  - git remote add origin [jouw git ur]
  - git push -u -f origin main
- read the error → gitlab project settings
- Configureer je editor zodat die overweg kan met ESLint en Prettier

## 2. Voeg basis styling toe



## 3. Voorbereiding

Voeg een 'title' property (string) toe aan het data object en gebruik dat in de h1 tag.

```
<h1>{{ title }}</h1>
```

Voeg nu ook een 'slots' property (array) toe aan het data object. Elk 'slot' object in de slots array bestaat uit een emoji (string) en finished (boolean):

```
{
  emoji: '🎲',
  finished: true
}
```

Vervang de drie slots in de template door [één v-for loop](#). Hou je ESLint errors en warnings in de gaten!

Maak een method 'play' en gebruik een directive om het [click event op de knop te koppelen aan die method](#).

```
methods: {
  play () {
    alert('dit werkt')
  }
}
```

## 4. Play

Drukken op 'play' zorgt dat elk slot gaat 'draaien'.  
Maak nog een method 'rotate'.

```
rotate (slot) {
  // 1. zet finished op false
  slot.finished = false

  // 2. draaien
  let i = 0
  const int = setInterval(() => {
    slot.emoji = emoji[i]
    // verhoog i,
    // wat als i groter wordt dan emoji.length?
  }, 120)

  // 3. elk slot 'draait' voor een willekeurige tijd
  setTimeout(() => {
    clearInterval(int)
    slot.finished = true
  }, 2000 + Math.random() * 3000)
},
play () {
  this.slots.forEach(this.rotate)
}
```

## 5. Finished?

Maak een computed property 'finished',  
die returnt true als de finished property van alle slots true is.

Tip: [array.prototype.every\(\)](#)

Gebruik die computed property om te verhinderen dat op de knop geklikt kan worden wanneer minstens één slot nog 'draait' en om de tekst 'Rien ne va plus' te tonen wanneer minstens één slot draait.

Gebruik een disabled directive enerzijds en conditional rendering anderzijds.

## 6. Gewonnen?

Maak nog een computed property: 'hasWon'.

Deze returnt true als finished true is EN alle emoji zijn gelijk EN ze zijn niet gelijk aan '🔔'

Gebruik 'hasWon' om [via class binding](#) iets leuks te doen met de CSS bij winst. Wijzig bijvoorbeeld de achtergrondkleur van de slots.

## 7. Voeg geluidseffecten toe

Download

- <https://freesound.org/people/kubawolanin/sounds/185611/>
- <https://freesound.org/people/InspectorJ/sounds/411088/>

Voeg twee audio tags toe aan je HTML en geef ze een 'ref' attribuut.

```
<audio preload="auto"
  volume="0.1"
  ref="tick"
  src="185611__kubawolanin__metal-tick-1.wav"
  loop
  hidden>
</audio>
<audio preload="auto"
  ref="ding"
  src="411088__inspectorj__bell-candle-damper-a-
h4n.wav"
  hidden>
</audio>
```

Voeg een [watcher](#) toe die kijkt naar onze finished property.

Als finished true wordt dan pauzeren we de 'tick' audio, als finished false wordt dan starten we de 'tick' audio.

Pas de rotate method aan.

Wanneer een slot stopt met draaien zet je de '[currentTime](#)' van het 'ding' audio element op 0 en [speel je het terug af](#).

Gebruik this.\$refs om de audio elementen te bedienen.

## 8. Extra: vue transitions

Zorg voor 'rolling' animatie.

Tips:

1. From translateY(-50%) to translateY(50%)
2. Using Keys to Track Elements

## Evaluatie

Geen minimale uitwerking = 0/20

Minimale uitwerking is minstens alle onderstaande:

- Git repo
- Vue App die gestart kan worden
- Tekst interpolation,
- List rendering

### Quotering

/2 template  
/2 computed properties  
/2 methods  
/4 watcher  
/4 refs  
/4 CSS  
/2 code quality

ESLint errors = -5

Extra inzet = +2