



# BUILDING VUE.JS REACTIVITY FROM SCRATCH

DEMYSTIFYING REACTIVITY IN MODERN FRAMEWORKS

MARC BACKES | 20/05/2020

@\_MARCBA

# MARC BACKES, M.Sc.

🔥 Freelancer

🔥 Based in Luxembourg 🇩🇪

🔥 Working with



🔥 Twitter: @\_marcba



# WHY THIS TALK?



Vuejs Amsterdam



# WHAT IS REACTIVITY?

A close-up shot of Harry Potter, played by Daniel Radcliffe, wearing round glasses and a brown jacket, flying a broomstick over a body of water. He is looking towards the camera with a determined expression. The background shows a large, calm lake or river under a cloudy sky.

**MAGIC!**

A close-up shot of Harry Potter, played by Daniel Radcliffe, wearing his signature round glasses and dark hair. He is looking slightly downwards and to his left with a serious expression. He is wearing a brown corduroy jacket over a white shirt. The background is a blurred landscape of trees and water, suggesting he is flying on a broomstick.

**MAGIC?**

# WHAT IS REACTIVITY?

- 🔥 **Change** one thing
- 🔥 **Update** it wherever it is needed
- 🔥 So we need to **listen** to changes
- 🔥 And **notify** when a change happens

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
  pagesLeft = bookPages - pagesRead
}

function readPage() {
  console.log('Reading. I can\'t believe Dumbledore died')
  pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
    pagesLeft = bookPages - pagesRead
}

function readPage() {
    console.log('Reading. I can\'t believe Dumbledore died')
    pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
    pagesLeft = bookPages - pagesRead
}

function readPage() {
    console.log('Reading. I can\'t believe Dumbledore died')
    pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
    pagesLeft = bookPages - pagesRead
}

function readPage() {
    console.log('Reading. I can\'t believe Dumbledore died')
    pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
  pagesLeft = bookPages - pagesRead
}

function readPage() {
  console.log('Reading. I can\'t believe Dumbledore died')
  pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

You have 12 pages left to read

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
  pagesLeft = bookPages - pagesRead
}

function readPage() {
  console.log('Reading. I can\'t believe Dumbledore died')
  pagesRead++
}

countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

You have 12 pages left to read  
Reading. I can't believe Dumbledore died

# WHY DO WE NEED REACTIVITY?

🔥 JavaScript gets executed in a sequential order

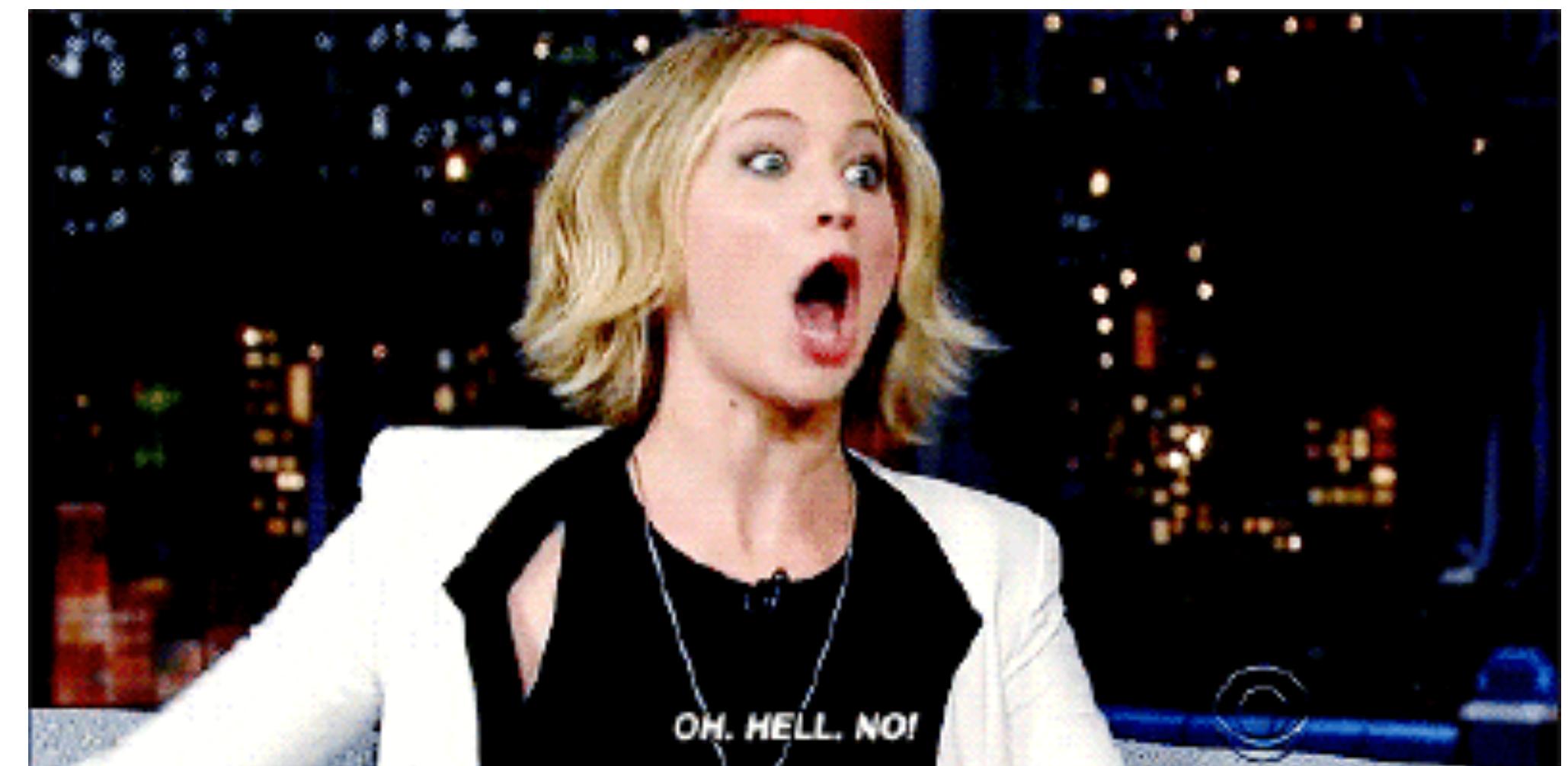
```
let bookPages = 607
let pagesRead = 595
let pagesLeft = 0

function countPagesLeft() {
  pagesLeft = bookPages - pagesRead
}

function readPage() {
  console.log('Reading. I can\'t believe Dumbledore died')
  pagesRead++
}

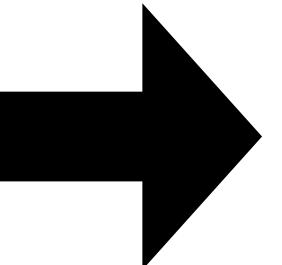
countPagesLeft()
console.log(`You have ${pagesLeft} pages left to read`)
readPage()
console.log(`You have ${pagesLeft} pages left to read`)
```

You have 12 pages left to read  
Reading. I can't believe Dumbledore died  
You have 12 pages left to read



# BUILDING REACTIVITY

🔥 It's **not** rocket science



🔥 And I will prove it to you



# BUILDING REACTIVITY

## Ingredients:

 **Getter / Setter**

 **Observer-Pattern**

 **“Watcher”**

 **Object.defineProperty**

# BUILDING REACTIVITY

PART 1 - DEPENDENCY CLASS

LIVE DEMO

# SUMMARY PART 1

- 🔥 Create a “dependency” class ([Dep](#))
- 🔥 Subscribe to changes in the class when used ([get](#))
- 🔥 Notify when the value changes ([set](#))
- 🔥 Do something when value changes ([watchEffect](#))

# BUILDING REACTIVITY

PART 2 - *REACTIVE FUNCTION*

LIVE DEMO

# SUMMARY PART 2

## Create state (`reactive`)

- 🔥 Iterate over all properties
- 🔥 Create dependency (`new Dep()`)
- 🔥 Define getter and setter for each property  
(`Object.defineProperty`)

# THANK YOU

(FOR NOT FALLING ASLEEP)

# LINKS

🔥 **Dependency example**

[marc.dev/gist/vue-dependency](https://marc.dev/gist/vue-dependency)

🔥 **Final example**

[marc.dev/gist/vue-reactive](https://marc.dev/gist/vue-reactive)