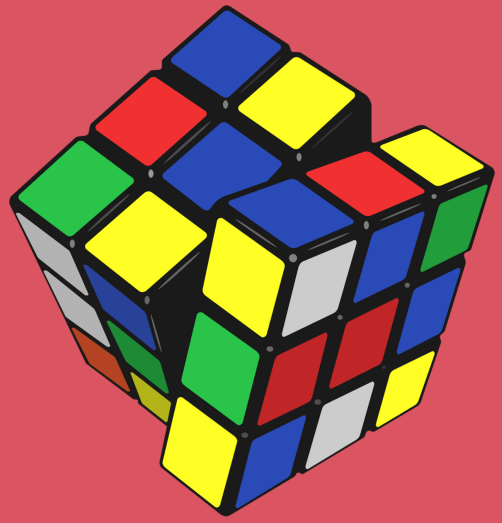
An Introduction to Test-Driven Development with Vue.js

Testing is one of the pillars of writing robust software.

Tests should give you confidence that you aren't shipping broken software.

But... testing can be tough.

Especially with UI components.



```
s="color-picker">
ss="swatches">

y="index"
or="(swatch, index) in swatches"
yle="{ background: `#${swatch}` }"
ss="swatch"
ass="[
 active: index === activeSwatch },
 light: isLight(swatch) }

ick="activeSwatch = index"

eck-icon />


ass="color">
on
y="index"
or="(mode, index) in colorModes"
ss="color-mode"
ass="[{ active: index === activeMode }, `color-mode-${mode}`]"
ick="activeMode = index"

mode }}
ton>

ass="color-code">{{ activeCode }}</div>
```

```
<script>
import { rgb, hex, hsl, isLight } from '@/utils/color'
import CheckIcon from '@/assets/check.svg'
const modes = { rgb, hex, hsl }

export default {
  components: { CheckIcon },
  props: {
    swatches: {
      type: Array,
      default() {
        return []
      }
    }
  },
  data() {
    return {
      activeSwatch: 0,
      activeMode: 0,
      colorModes: ['hex', 'rgb', 'hsl'],
      isLight
    }
  },
  computed: {
    activeCode() {
      const activeColor = this.swatches[this.activeSwatch]
      const activeMode = this.colorModes[this.activeMode]
      return modes[activeMode](activeColor)
    }
  }
}
</script>
```

HTML?

CSS classes?

View logic?

Event handlers?

Methods?

Computed properties?

Lifecycle steps?

100% coverage?

Unit vs. integration vs. e2e?

Artwork by Simon Ålander

Artwork by Simon Ålander

Test-Driven Development (TDD)

Popularized by Kent Beck

1

Red

Write a test that describes an expected behavior, then run it, ensuring it fails.

2

Green

Write the dumbest, most straightforward code you can to make the test pass.

3

Refactor

Refactor the code to make it right.

TDD ♥ 

What does TDD look like with Vue?

2 of 5

src/components/Rating.vue

```
1 <template>
2 </template>
3
4 <script>
5 </script>
```

tests/unit/Rating.spec.js

```
1
2  import { shallowMount } from '@vue/test-utils'
3  import Rating from '@/components/Rating'
4
5  let wrapper = null
6
7  beforeEach(() => {
8    wrapper = shallowMount(Rating)
9  })
10
11 afterEach(() => {
12   wrapper.destroy()
13 })
```

Thinking time!

5 stars to display

2 of 5

Red

```
describe('Rating', () => {
  test('renders a list of stars', () => {
    const stars = wrapper.findAll('.star')
    expect(stars.length).toBe(5)
  })
})
```

Terminal

```
npm run test:unit --watchAll
```

```
FAIL  tests/unit/Rating.spec.js
 Rating
   ✕ renders a list of stars (7ms)

 ● Rating › renders a list of stars

   expect(received).toBe(expected) // Object.is equality

   Expected: 5
   Received: 0

     16 |   test('renders a list of stars', () => {
     17 |     const stars = wrapper.findAll('.star')
   > 18 |     expect(stars.length).toBe(5)
        |                          ^
     19 |   })
```

Green

```
<template>
  <ul>
    <li class="star"></li>
    <li class="star"></li>
    <li class="star"></li>
    <li class="star"></li>
    <li class="star"></li>
  </ul>
</template>
```

```
 PASS  tests/unit/Rating.spec.js
  Rating
    ✓ renders a list of stars (21ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
```

Refactor

```
1
2 beforeEach(() => {
3   wrapper = shallowMount(Rating, {
4     propsData: {
5       maxStars: 5
6     }
7   })
8 })
```

```vue
<template>
  <ul>
    <li
      :key="star"
      v-for="star in maxStars"
      class="star"></li>
  </ul>
</template>

<script>
export default {
  props: {
    maxStars: {
      type: Number,
      default: 5
    }
  }
}
</script>
```

```
 PASS  tests/unit/Rating.spec.js
  Rating
    ✓ renders a list of stars (6ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
```


WHY

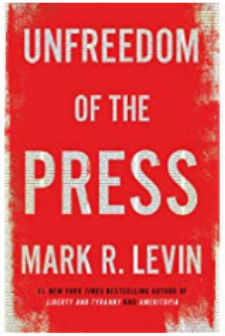
The whole idea of TDD is not to write code to make things work, but **to make tests pass**.

Thinking time!

2 active stars to display

2 of 5

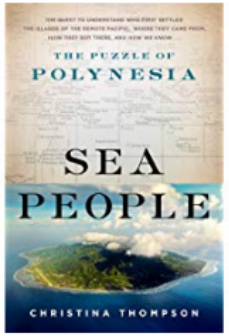
Best sellers



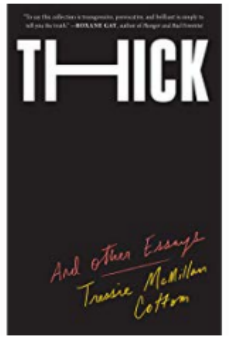
Unfreedom of the Press
Mark R. Levin
Hardcover
4.8 out of 5 stars 898
$16.80 prime

Bad Therapist (Exposure collection)
Evan Wright
Kindle Edition
2.5 out of 5 stars 7
$1.99

The Mueller Report
The Washington Post
Paperback
4.0 out of 5 stars 282
$9.00 prime

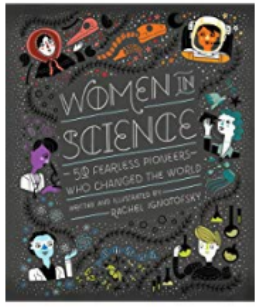


Sea People: The Puzzle of Polynesia
Christina Thompson
Kindle Edition
4.5 out of 5 stars 22
$14.99



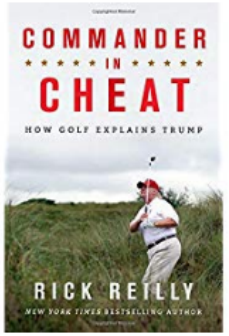
Thick: And Other Essays
Tressie McMillan Cottom
Kindle Edition
4.8 out of 5 stars 62
$13.99

Most gifted



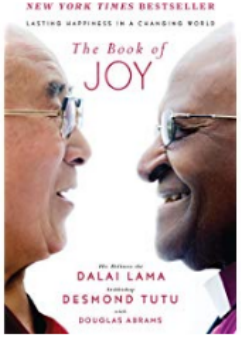
Women in Science: 50 Fearless Pioneers Who...
Rachel Ignotofsky
Hardcover
4.5 out of 5 stars 607
$12.73 prime

The Life-Changing Magic of Tidying Up: The...
Marie Kondō
Hardcover
4.5 out of 5 stars 14,345
$9.69 prime



Commander in Cheat: How Golf Explains Trump
Rick Reilly
Hardcover
4.5 out of 5 stars 212
$18.30 prime

Expecting Better: Why the Conventional...
Emily Oster
Paperback
4.5 out of 5 stars 882
$12.03 prime



The Book of Joy: Lasting Happiness in a...
Dalai Lama, Desmond Tutu...
Hardcover
4.8 out of 5 stars 1,366
$15.97 prime

```
1 test('renders active stars with class `active`', () => {
2   const activeStars = wrapper.findAll('.active')
3   expect(activeStars.length).toBe(2)
4 })
```

```
 FAIL  tests/unit/Rating.spec.js
  Rating
    ✓ renders a list of stars (5ms)
    ✕ renders active stars with class `active` (9ms)

  ● Rating › renders active stars with class `active`

    expect(received).toBe(expected) // Object.is equality

    Expected: 2
    Received: 0

      24 |   test('renders active stars with class
`active`', () => {
      25 |     const activeStars =
wrapper.findAll('.active')
    > 26 |     expect(activeStars.length).toBe(2)
         |                                ^
      27 |   })
      28 | })
```

```
<li
  :key="star"
  v-for="star in maxStars"
  :class="{ 'active': star <= 2 }"
  class="star"></li>
```

```
 PASS  tests/unit/Rating.spec.js
  Rating
    ✓ renders a list of stars (6ms)
    ✓ renders active stars with class `active` (4ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
```

```
beforeEach(() => {
  wrapper = shallowMount(Rating, {
    propsData: {
      maxStars: 5,
      initialGrade: 2
    }
  })
})
```

```
<template>
<!-- ... -->
<li
  :key="star"
  v-for="star in maxStars"
  :class="{ 'active': star <= initialGrade }"
  class="star"></li>
<!-- ... -->
</template>

<script>
export default {
  props: {
    // ...
    initialGrade: {
      type: Number,
      default: 0
    }
  }
}
</script>
```

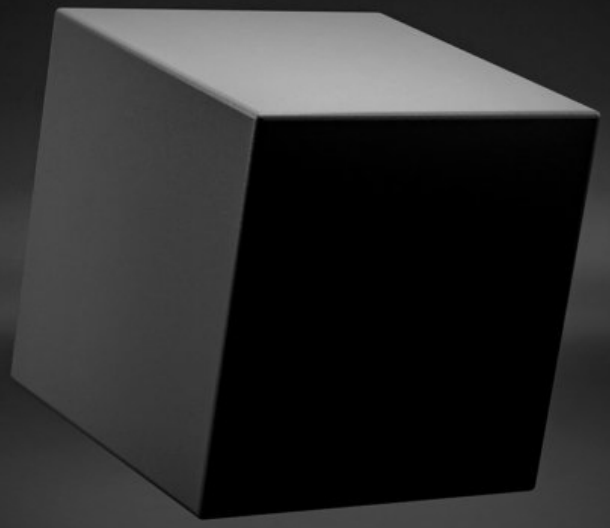
```
 PASS  tests/unit/Rating.spec.js
  Rating
    ✓ renders a list of stars (4ms)
    ✓ renders active stars with class `active` (2ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
```

What exactly are we testing?

Tests should give you confidence that you aren't shipping broken software.

Black box testing



Assert only the public interface.

Who are your users?

Final user

Developer

submit event

click event

props

With UI components,

your public interface is bigger than you think.

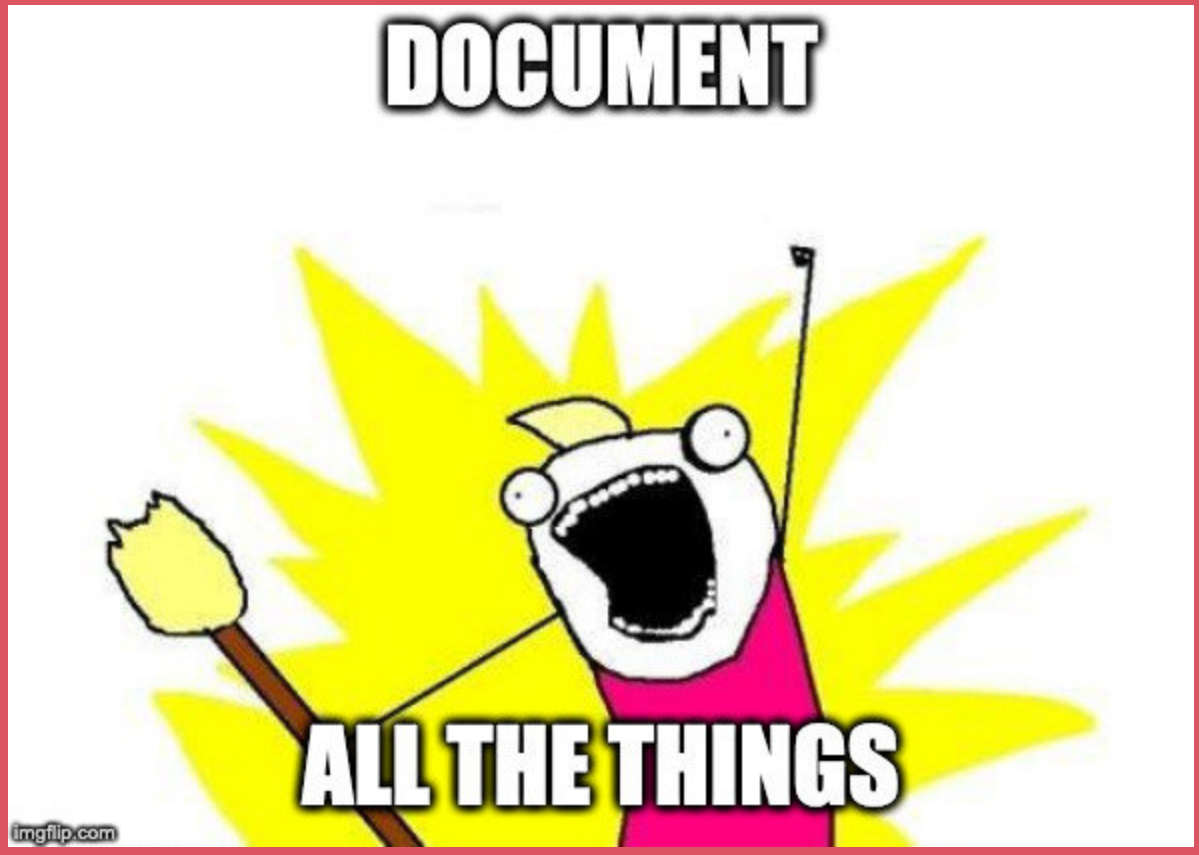
HTML

classes

scroll event

hover event

Do I care about this if it changes?


DOCUMENT
ALL THE THINGS
imgflip.com

1

TDD is a **fantastic way to write robust tests**, not too many and not too few.

2

TDD encourages refactors,
which leads to
better software design.

Sarah Dayan
@frontstuff_io

Follow

One of the things I like with TDD is that it forces you to focus on the right things. Your software's contract is infinitely more important than the way you implement it. Implementation will change over time. It's not that important. But a strong contract is.

11:26 AM - 5 May 2019

3

TDD is **much easier to follow with specs.**

But...

TDD takes

a lot of time.

So, is it worth it?

1

With practice, **you will get faster at TDD.**

2

Fixing bugs is **far more costly than preventing them**.


JUST
DO IT!

Thank you!
Questions?

@frontstuff_io

github.com/sarahdayan

f; frontstuff.io/an-introduction-to-tdd-with-vuejs