

# Accessibility and Test Driven Development

A match made in heaven

---

Rita Castro | SDC:LX @ VW Digital Solutions | 2023-07-04 require('lx') meetup

# Hello World require('lx')!





*Zero to Hero, Hercules, Disney 1997*

# Accessibility and Test Driven Development

*"Software Requirements are converted into test cases before SW is developed"*



# Accessibility and Test Driven Development

*“easily used or accessed by people with disabilities”*

*“adapted for use by people with disabilities”*



Sidewalk in Amsterdam [📸 by me ❤️]

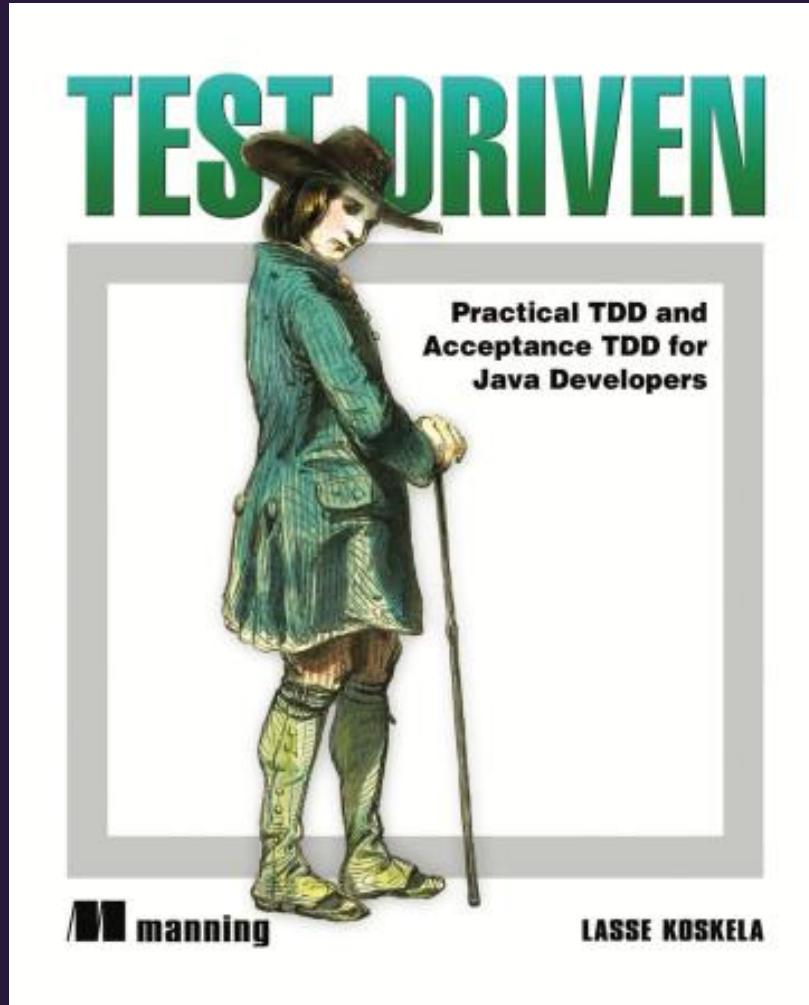
# Accessibility

a11y

*“enabling as many people as possible to use websites  
even when those people's abilities are limited in some way”*



# Test Driven Development



A screenshot of a web browser window titled "A11y and TDD: A match made in heaven". The URL is "localhost:8000". The page displays a "Sign up!" form. At the top right, there's a logo for "require('lx')" which shows a group of people working in an office. Below the logo, it says "Lisbon, Portugal", "2,392 members · Public group", and "Organized by Sérgio Ramos and 1 other". The form itself has a purple header "Personal details". It contains two input fields: "First Name" and "Last Name", both with placeholder text "Enter your name...". Below the form is a purple button labeled "Sign me up!". The footer of the page also has a purple bar with the text "© Rita Castro 2023".

```
describe("Sign up homepage", () => {
  it("takes us through the happy path", () => {
    // We open the page

    // From the logo, we can see that we are signing on to "require('lx')"

    // In the page, we find the section of the Personal Details

    // We click on "Enter your name..." and ...
    // ... we start typing on the input box

    // We press tab to go to the "surname" field ...
    // and we also fill it up!

    // Then we find the only button available ...
    // ... that happens to have the text "Sign up!" on it ...

    // and we click it!

    // Done! See you there! :)
  })
})
```

# text-input



```
FAIL  src/ui/molecules/text-input/text-input.test.jsx
      text-input module
        ✕ has the title and the input box (15 ms)

● text-input module › has the title and the input box

  expect(received).toHaveTextContent()

  received value must be a Node.
  Received has value: null

    9 |       render(<TextInput />);
   10 |
  > 11 |       expect(document.body.querySelector("p")).toHaveTextContent("Hi, welcome")
    |
   12 |           expect(document.body.querySelector("input")).toBeInTheDocument()
   13 |     })
   14 |   })

at __EXTERNAL_MATCHER_TRAP__ (node_modules/expect/build/index.js:317:30)
at Object.throwingMatcher [as toHaveTextContent] (node_modules/expect/build/index.js:318:15)
```

```

text-input.jsx ×
src > ui > molecules > text-input > text-input.jsx > ...
You, 37 seconds ago | 1 author (You)
1 import React from "react"
2
3 const TextInput = () => {
4   return (
5     <>
6       <p>Hi, welcome</p>
7       <input />
8     </>
9   )
10 }
11
12 export default TextInput
13

text-input.test.jsx M ×
src > ui > molecules > text-input > text-input.test.jsx > ...
You, 28 seconds ago | 1 author (You)
1 import { render } from "@testing-library/react";
2 import React from "react";
3 import "./text-input";
4 import TextInput from "./text-input";
5
6 describe("text-input molecule", () => {
7
8   it("has a custom title and is controlled input", () => {
9     render(<TextInput title={"Some random title"} />);
10
11     expect(document.body.querySelector("p")).toHaveTextContent("Some random title")
12     expect(document.body.querySelector("input")).toHaveValue("awesome")
13   })
14
15 })

```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL

**FAIL** src/ui/molecules/text-input/text-input.test.jsx

text-input molecule

✗ has a custom title and is controlled input (22 ms)

● text-input molecule > has a custom title and is controlled input

```

expect(element).toHaveTextContent()

Expected element to have text content:
  Some random title
Received:
  Hi, welcome

```

```

9 |       render(<TextInput title={"Some random title"} />);
10 |
11 |     expect(document.body.querySelector("p")).toHaveTextContent("Some random title")

```

You, 1 minute ago | 1 author (You)

```
1 import { render } from "@testing-library/react";
2 import React from "react";
3 import "./text-input";
4 import TextInput from "./text-input";

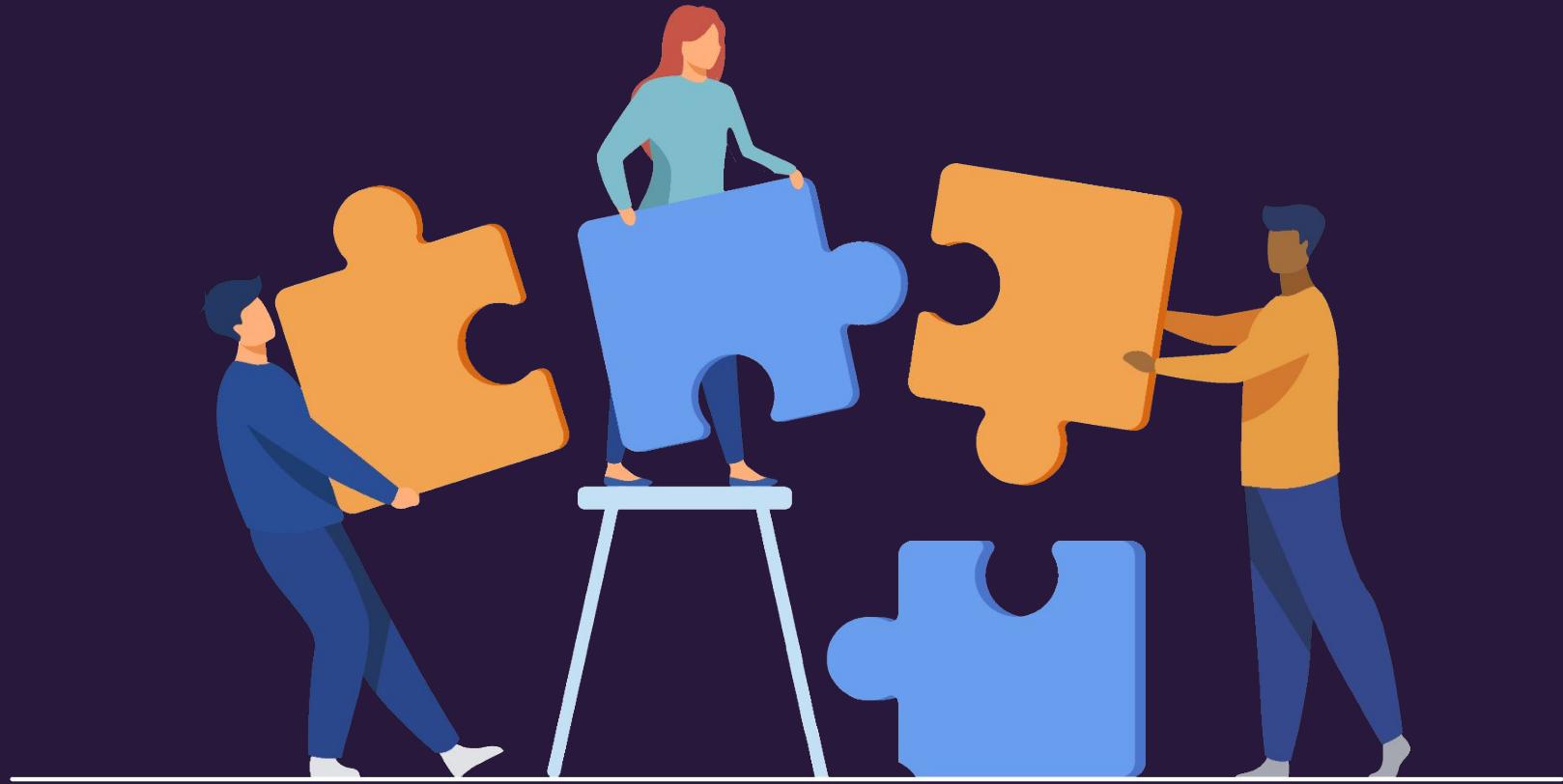
5
6 describe("text-input molecule", () => {
7
8   it("has a custom title and is controlled input", () => {
9     render(<TextInput title={"Some random title"} />);
10
11    expect(document.body.querySelector("p")).toHaveTextContent("Some random title")
12    expect(document.body.querySelector("input")).toHaveValue("awesome")
13  })
14})
15
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL

```
PASS  src/ui/molecules/text-input/text-input.test.jsx
      text-input molecule
        ✓ has a custom title and is controlled input (41 ms)
```

```
Test Suites: 1 passed, 1 total  
Tests:       1 passed, 1 total  
S snapshots: 0 total  
Time:        0.945 s, estimated 1 s  
Ran all test suites related to changed files.
```

**Watch Usage:** Press w to show more.



*Teamwork people vector created by pch.vector [Adapted]*

```

src > ui > organisms > sign-up-form > sign-up-form.jsx > SignUpForm
1  const SignUpForm = () => null
2  ^
3  export default SignUpForm
4

src > ui > organisms > sign-up-form > sign-up-form.test.jsx > ...
1  import { render } from '@testing-library/react'
2  import React from "react"
3  import SignUpForm from "./sign-up-form"
4
5  describe("Form to sign up for an event", () => {
6    it("asks for first name and last name", () => {
7      render(<SignUpForm />)
8
9      expect(document.body.querySelector("p")).toHaveTextContent("First Name")
10   expect(document.body.querySelector("p")).toHaveTextContent("Last Name")
11
12   expect(document.body.querySelector("button")).toHaveTextContent("Sign me up!")
13 })
14
15

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    GITLENS    TERMINAL

**FAIL** src/ui/organisms/sign-up-form/**sign-up-form.test.jsx**

Form to sign up for an event

- ✗ asks for first name and last name (14 ms)

● Form to sign up for an event > asks for first name and last name

```

expect(received).toHaveTextContent()

received value must be a Node.
Received has value: null

```

```

7 |       render(<SignUpForm />)
8 |
> 9 |       expect(document.body.querySelector("p")).toHaveTextContent("First Name")

```

**FAIL** src/ui/organisms/sign-up-form/sign-up-form.test.jsx  
Form to sign up for an event

\* asks for first name and last name (58 ms)

- Form to sign up for an event > asks for first name and last name

```
expect(element).toHaveTextContent()
```

Expected element to have text content:

Last Name

Received:

**First Name**

```
8 |
9 |     expect(document.body.querySelector("p")).toHaveTextContent("First Name")
> 10 |    expect(document.body.querySelector("p")).toHaveTextContent("Last Name")
   |    ^
11 |
12 |    expect(document.body.querySelector("button")).toHaveTextContent("Sign me up!")
13 |}
```

PROBLEMS OUTPUT DEBUG CONSOLE GIT LENS TERMINAL

Form to sign up for an event  
✓ asks for first name and last name (68 ms)

```
Test Suites: 1 passed, 1 total  
Tests:       1 passed, 1 total  
Snapshots:   0 total  
Time:        0.927 s, estimated 1 s  
Ran all test suites related to changed files
```

**Watch Usage:** Press w to show more.

text-input.jsx sign-up-form.jsx M X

```
src > ui > organisms > sign-up-form > sign-up-form.jsx > SignUpForm
  You, 1 second ago | 1 author (You)
1 import React from "react"
2 import TextInput from "ui/molecules/text-input/text-input"
3
4 const SignUpForm = () => {
5   return (
6     <>
7       <TextInput title={"Last Name"} />
8       <TextInput title={"First Name"} />
9     </>
10    <button>Sign me up!</button>
11  )
12}
13
14
15 export default SignUpForm
16
```

text-input.test.jsx sign-up-form.test.jsx X

```
sign-up-form.test.jsx > describe("Form to sign up for an event") callback > it("asks for first name and last name") callback
  You, 26 seconds ago | 1 author (You)
1 import { render } from "@testing-library/react"
2 import React from "react"
3 import SignUpForm from "./sign-up-form"
4
5 describe("Form to sign up for an event", () => {
6   it("asks for first name and last name", () => {
7     render(<SignUpForm />)
8
9     expect(document.body.querySelectorAll("p")[0]).toHaveTextContent("First Name")
10    expect(document.body.querySelectorAll("p")[1]).toHaveTextContent("Last Name")
11
12    expect(document.body.querySelector("button")).toHaveTextContent("Sign me up!")
13  })
14})
15
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL

```
7 |       render(<SignUpForm />)
8 |
9 > 9 |       expect(document.body.querySelectorAll("p")[0]).toHaveTextContent("First Name")
10 |       expect(document.body.querySelectorAll("p")[1]).toHaveTextContent("Last Name")
11 |
12 |       expect(document.body.querySelector("button")).toHaveTextContent("Sign me up!")

at Object.toHaveTextContent (src/ui/organisms/sign-up-form/sign-up-form.test.jsx:9:56)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:  0 total
Time:        0.918 s, estimated 1 s
Ran all test suites related to changed files.
```

Watch Usage: Press w to show more. [ ]



*The more your tests resemble the way your  
software is used, the more confidence they  
can give you.*



[Kent C. Dodds, <https://twitter.com/kentcdodds/status/977018512689455106?lang=en>]

The screenshot shows a dark-themed VS Code interface with two code editors and a terminal at the bottom.

**Code Editors:**

- Left Editor:** File: `text-input.jsx`. Content:

```
src > ui > organisms > sign-up-form > sign-up-form.jsx > SignUpForm
You, 6 minutes ago | 1 author (You)
1 import React from "react"
2 import TextInput from "ui/molecules/text-input/text-input"
3
4 const SignUpForm = () => {
5   return (
6     <>
7       <TextInput title={"Last Name"} />
8       <TextInput title={"First Name"} />
9
10    <button>Sign me up!</button>
11  </>
12)
13
14
15 export default SignUpForm
16
```
- Right Editor:** File: `sign-up-form.test.jsx`. Content:

```
gn-up-form.test.jsx > describe("Form to sign up for an event") callback > it("asks for first name and last name") callback
You, 1 second ago | 1 author (You)
1 import { render, screen } from "@testing-library/react"
2 import React from "react"
3 import SignUpForm from "./sign-up-form"
4
5 describe("Form to sign up for an event", () => {
6   it("asks for first name and last name", () => {
7     render(<SignUpForm />)
8
9     expect(screen.getByText("First Name")).toBeInTheDocument()
10    expect(screen.getByText("Last Name")).toBeInTheDocument()
11
12    expect(screen.getByRole("button", { name: "Sign me up!" })).toBeInTheDocument()
13  })
14
15
```

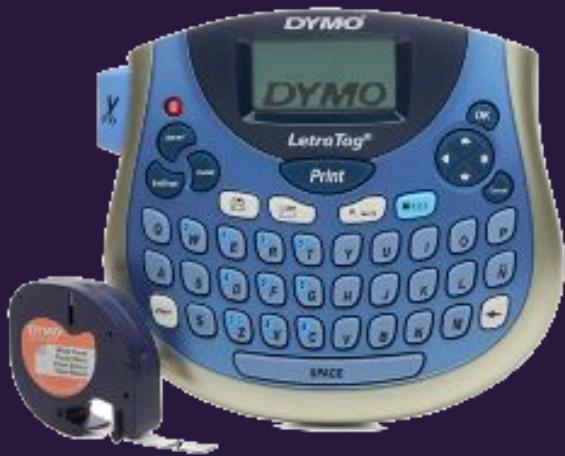
**Terminal:**

```
PASS  src/ui/organisms/sign-up-form/sign-up-form.test.jsx
  Form to sign up for an event
    ✓ asks for first name and last name (67 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Schemas:    0 total
Time:        0.603 s, estimated 1 s
Ran all test suites related to changed files.

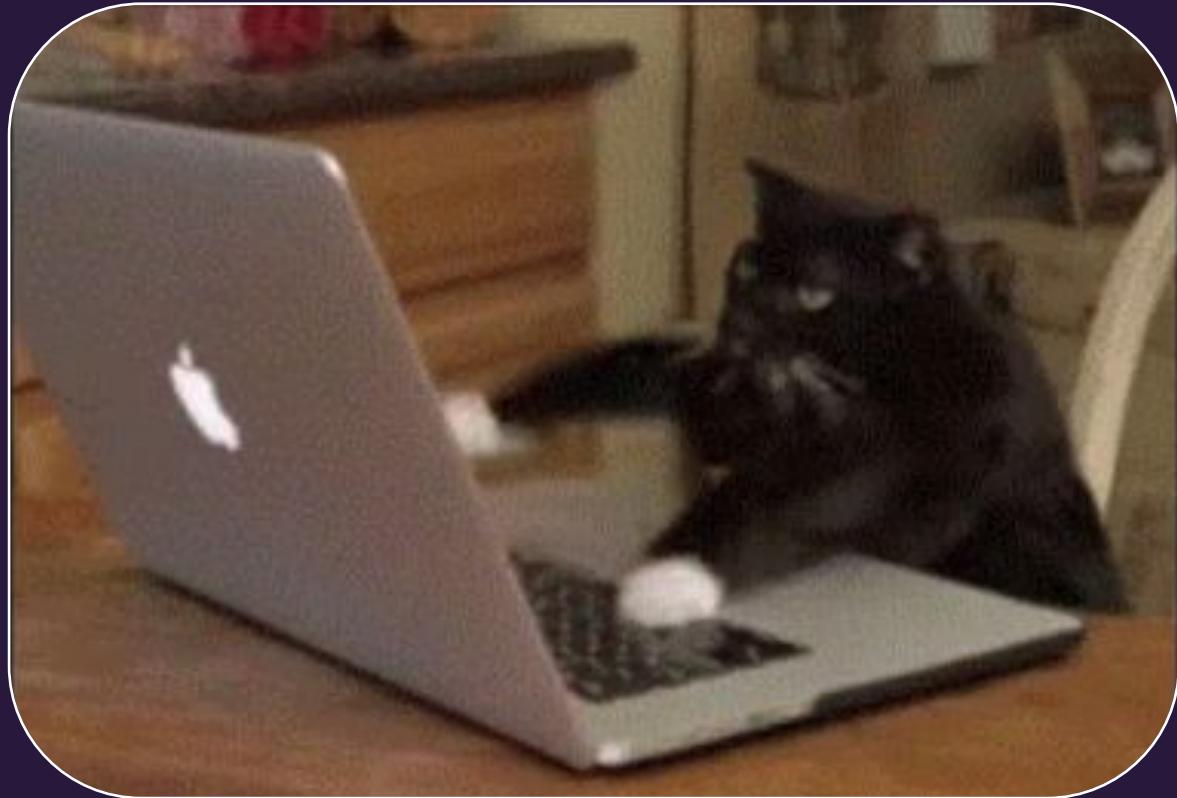
Watch Usage: Press w to show more.[]
```

# About labels...



*“a small piece of paper (...) attached to an object and giving information about it”*

The **<label>** HTML element represents a caption for an item in a user interface.



text-input.jsx X sign-up-form.jsx

```
src > ui > molecules > text-input > text-input.jsx > ...
You, 28 minutes ago | 1 author (You)
1 import React from "react"
2
3 const TextInput = ({ title }) => {
4   return (
5     <>
6       <p>{title}</p>
7       <input value="awesome" />
8     </>
9   )
10
11 You, 31 minutes ago + 1 ...
12 export default TextInput
13
```

text-input.test.jsx M X sign-up-form.test.jsx

```
text-input.test.jsx > describe("text-input molecule") callback > it("has a custom title and is controlled input") callback
4   import TextInput from "./text-input";
5
6   describe("text-input molecule", () => {
7
8     it("has a custom title and is controlled input", () => {
9       render(<TextInput label={"Some random label"} fieldName={"hello"} />);
10
11       expect(screen.getByLabelText("Some random label")).toBeInTheDocument()
12
13       const input = screen.getByRole("textbox", { name: "Some random label" })
14       expect(input).toHaveValue("Awesome")
15     })
16
17
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL

```
/>
</div>
</body>

9 |   render(<TextInput label={"Some random label"} fieldName={"hello"} />);
10 |
11 |   expect(screen.getByLabelText("Some random label")).toBeInTheDocument()
12 |
13 |   const input = screen.getByRole("textbox", { name: "Some random label" })
14 |   expect(input).toHaveValue("Awesome")

at Object.getElementError (node_modules/@testing-library/dom/dist/config.js:37:19)
at getAllByLabelText (node_modules/@testing-library/dom/dist/queries/label-text.js:111:38)
at node_modules/@testing-library/dom/dist/query-helpers.js:52:17
at node_modules/@testing-library/dom/dist/query-helpers.js:95:19
at Object.getByLabelText (src/ui/molecules/text-input/text-input.test.jsx:11:23)

Test Suites: 1 failed, 1 total
Tests:      1 failed, 1 total
Snapshots:  0 total
Time:      0.6 s, estimated 1 s
Ran all test suites matching /src/ui/molecules/text-input/text-input.test.jsx/i.

Watch Usage: Press w to show more.[]
```

```

src > ui > molecules > text-input > text-input.jsx > ...
You, 1 second ago | author (You)
1 import React from "react"
2
3 const TextInput = ({ label, fieldName }) => {
4   return (
5     <label for={fieldName}>{label}</label>
6     <input id={fieldName} value="Awesome" name={fieldName} />
7   )
8 }
9
10 export default TextInput
11
12
13

```

```

text-input.test.jsx M × sign-up-form.jsx ...
4 import TextInput from "./text-input";
5
6 describe("text-input molecule", () => {
7
8   it("has a custom title and is controlled input", () => {
9     render(<TextInput label="Some random label"(fieldName="hello") />);
10    expect(screen.getByLabelText("Some random label")).toBeInTheDocument();
11
12    const input = screen.getByRole("textbox", { name: "Some random label" });
13    expect(input).toHaveValue("Awesome");
14  });
15})
16
17

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    GITLENS    TERMINAL

**PASS** src/ui/molecules/text-input/**text-input.test.jsx**  
 text-input molecule  
 ✓ has a custom title and is controlled input (81 ms)

**Test Suites:** 1 passed, 1 total  
**Tests:** 1 passed, 1 total  
**Snapshots:** 0 total  
**Time:** 0.902 s, estimated 1 s  
 Ran all test suites matching /src/ui/molecules/text-input/text-input.test.jsx/i.

**Watch Usage:** Press w to show more. [ ]

```

src > ui > organisms > sign-up-form > sign-up-form.jsx > ...
You, 13 seconds ago | 1 author (You)
1 import React from "react"
2 import TextInput from "ui/molecules/text-input/text-input"
3
4 const SignUpForm = () => {
5   return (
6     <div>
7       <TextInput label={"First Name"} fieldName={"firstName"} />
8       <TextInput label={"Last Name"} fieldName={"lastName"} />
9
10      <button>Sign me up!</button>
11    </div>
12  )
13}
You, 22 minutes ago * 3 ...
14
15 export default SignUpForm
16

```

```

src > ui > organisms > sign-up-form > sign-up-form.test.jsx > ...
describe("Form to sign up for an event", () => {
  it("asks for first name and last name", () => {
    render(<SignUpForm />)
    expect(screen.getByText("First Name")).toBeInTheDocument()
    expect(screen.getByText("Last Name")).toBeInTheDocument()
    expect(screen.getByRole("button", { name: "Sign me up!" })).toBeInTheDocument()
  })
  it("shows an image to identify what you're signing up for", () => {
    render(<SignUpForm />)
    expect(screen.getByAltText("Meetup Logo")).toBeInTheDocument()
  })
})

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    GITLENS    TERMINAL    zsh    +    ⌂    ⌂    ⌂    ⌂    ⌂    ⌂    ⌂    ⌂

**FAIL** src/ui/organisms/sign-up-form/**sign-up-form.test.jsx**

Form to sign up for an event

- ✓ asks for first name and last name (125 ms)
- ✗ shows an image to identify what you're signing up for (18 ms)

● Form to sign up for an event > shows an image to identify what you're signing up for

TestingLibraryElementError: Unable to find an element with the alt text: Meetup Logo

Ignored nodes: comments, script, style

```

<body>
  <div>
    <label
      for="firstName"
    >
      First Name
    </label>
    <input
      id="firstName"
      name="firstName"
      value="Awesome"
    </input>
  </div>

```

text-input.jsx sign-up-form.jsx M X

```
src > ui > organisms > sign-up-form > sign-up-form.jsx > SignUpForm
  You, 1 second ago | 1 author (You)
1 import React from "react"
2 import TextInput from "ui/molecules/text-input/text-input"
3
4 const SignUpForm = () => {
5   return (
6     <div>
7       <TextInput label={"First Name"} fieldName={"firstName"} />
8       <TextInput label={"Last Name"} fieldName={"lastName"} />
9
10      <img alt="Meetup Logo" /> You, 1 second ago * Uncommitted changes
11
12      <button>Sign me up!</button>
13    </div>
14  )
15
16  export default SignUpForm
17
18
```

text-input.test.jsx sign-up-form.test.jsx M X

```
src > ui > organisms > sign-up-form > sign-up-form.test.jsx > describe("Form to sign up for an event") callback
  4
  5   describe("Form to sign up for an event", () => {
  6     it("asks for first name and last name", () => {
  7       render(<SignUpForm />)
  8
  9       expect(screen.getByText("First Name")).toBeInTheDocument()
10       expect(screen.getByText("Last Name")).toBeInTheDocument()
11
12       expect(screen.getByRole("button", { name: "Sign me up!" })).toBeInTheDocument()
13     })
14
15     it("shows an image to identify what you're signing up for", () => {
16       render(<SignUpForm />)
17
18       expect(screen.getByAltText("Meetup Logo")).toBeInTheDocument()
19     })
20   })
21
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL

**PASS** src/ui/organisms/sign-up-form/sign-up-form.test.jsx

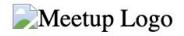
Form to sign up for an event

- ✓ asks for first name and last name (100 ms)
- ✓ shows an image to identify what you're signing up for (13 ms)

Test Suites: 1 passed, 1 total  
 Tests: 2 passed, 2 total  
 Snapshots: 0 total  
 Time: 1.187 s, estimated 2 s  
 Ran all test suites related to changed files.

Watch Usage: Press w to show more. [ ]

## Sign up!



Personal details

F

L

S



**it's something**



Graph	Description
○	○ ⚙ main   origin ⚙ origin/HEAD 🎉 Make the ui very pretty
●	✓ Add the other organisms for the form
●	✍ Add the happy path for filling out the form
●	✨ Add end-to-end testing infrastructure with Cypress
●	⚡ Create gitconfig for pairing
●	✨ Use jsnation form in the page
●	♻️ Use personal-details on form
●	✨ Create personal-details organism
●	✓ Text input now has a type!!
●	✍ Add option for type in text-input
●	✓ Make usage of FormData api to gather the form data
●	✍ Add a user driven test to populate the form
●	✓ Make correct usage of text-input in form
●	✓ Update text-input molecule
●	✍ Revisit the text-input molecule text
●	✍ Add semantics to the form
●	♻️ Make the form.test use the TL
●	+ Add the Testing Library
●	♻️ Refactor the test code to make it correct!
●	✍ Create the form test and an attempt of its implementation
●	✓ Make it configurable!
●	✍ Make the text-input molecule a configurable and controlable input
●	✓ Make the text-input molecule test pass
●	✍ Add the failing test for text-input and tests setup
●	🎉 Create basic skeleton
●	Initial commit



[ritamcastro/a11y-and-tdd-react](https://github.com/ritamcastro/a11y-and-tdd-react)



*Zero to Hero, Hercules, Disney 1997*



```
describe("Sign up homepage", () => {
  it("takes us through the happy path", () => {
    // We open the page

    // From the logo, we can see that we are signing on to "require('lx')"

    // In the page, we find the section of the Personal Details

    // We click on "Enter your name ..." and ...
    // ... we start typing on the input box

    // We press tab to go to the "surname" field ...
    // and we also fill it up!

    // Then we find the only button available ...
    // ... that happens to have the text "Sign up!" on it ...

    // and we click it!
    // Done! See you there! :)

  })
})
```

# Thank you!

 [rita.castro@vwds.pt](mailto:rita.castro@vwds.pt)

 <https://github.com/ritamcastro>

 [ritamcastro@gmail.com](mailto:ritamcastro@gmail.com)

 <https://www.linkedin.com/in/ritamcastro>

