

# **Testing Web Accessibility**

**How to find accessibility vulnerabilities in your website**

**Adrián Bolonio**



**@bolonio**

# Adrián Bolonio



Madrid (Spain)



Team Lead &  
Frontend Developer



#a11y Advocate



@bolonio



# Accessibility (a11y)

# **Accessibility (a11y)**

**a11y**

---

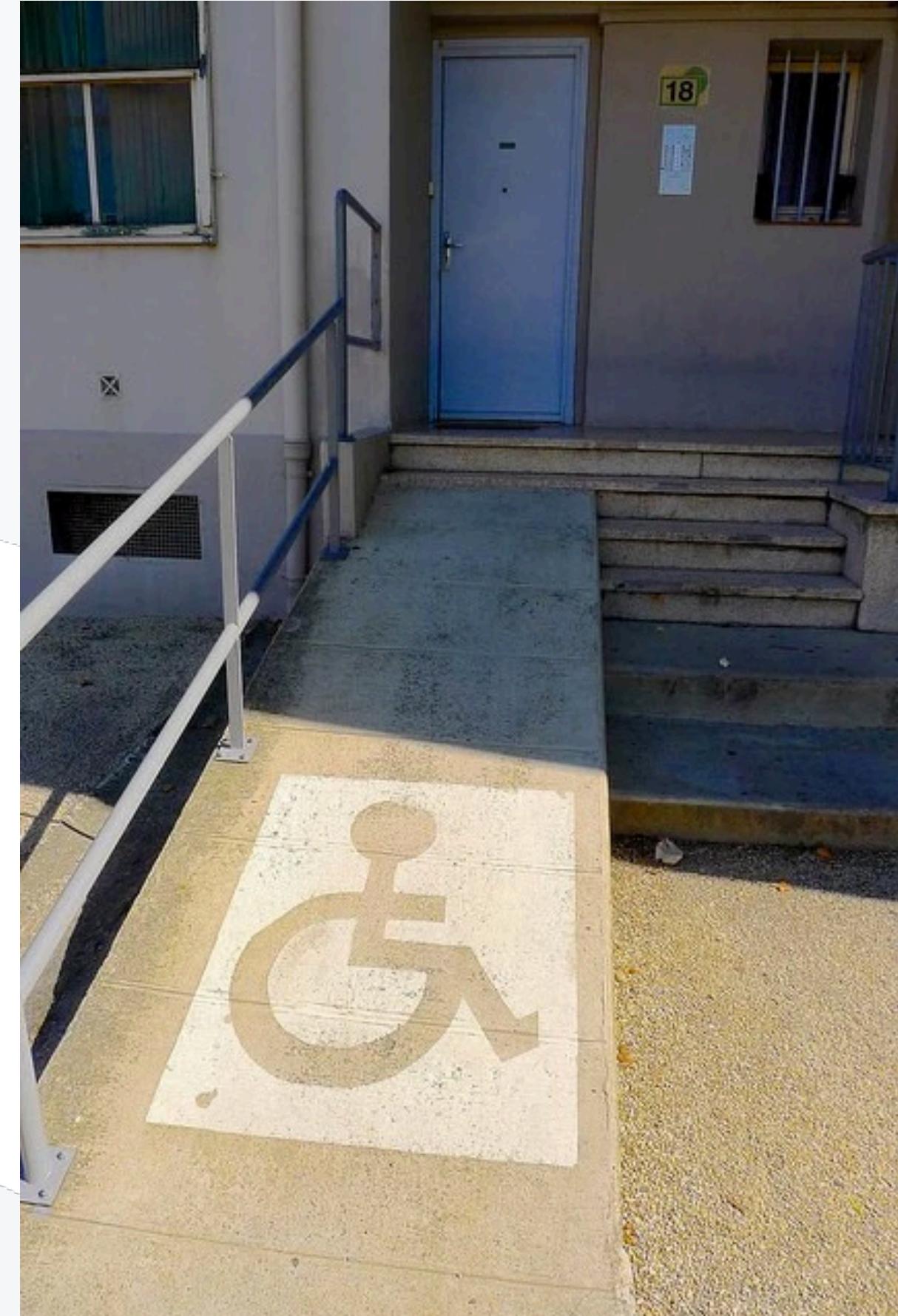
# Accessibility (a11y)

accessibility



1 2 3 4 5 6 7 8 9 10 11

# Accessibility (a11y)



# Web Accessibility (a11y)

# Web Accessibility (a11y)

The Web is fundamentally designed to work for **all people**, whatever their hardware, software, language, location, or ability. When the Web meets this goal, it is **accessible** to people with a diverse range of hearing, movement, sight, and cognitive ability.

Accessibility is essential for developers and organizations that want to create high-quality websites and web tools, and **not exclude** people from using their products and services.

Web accessibility means that websites, tools, and technologies are designed and developed so that **people with disabilities can use them**.

# Web Accessibility (a11y)



You need to click on the button in the **top right** corner.

You need to click on the button with the **engine icon**.

# Web Accessibility (a11y)

2018      7.6 billion

---

1+ billion

people live with some form of vision impairment

# Developing Web Accessibility

# Developing Web Accessibility



<https://www.youtube.com/watch?v=yKYA-7tczuk>

**How does your website  
“sound” like?**

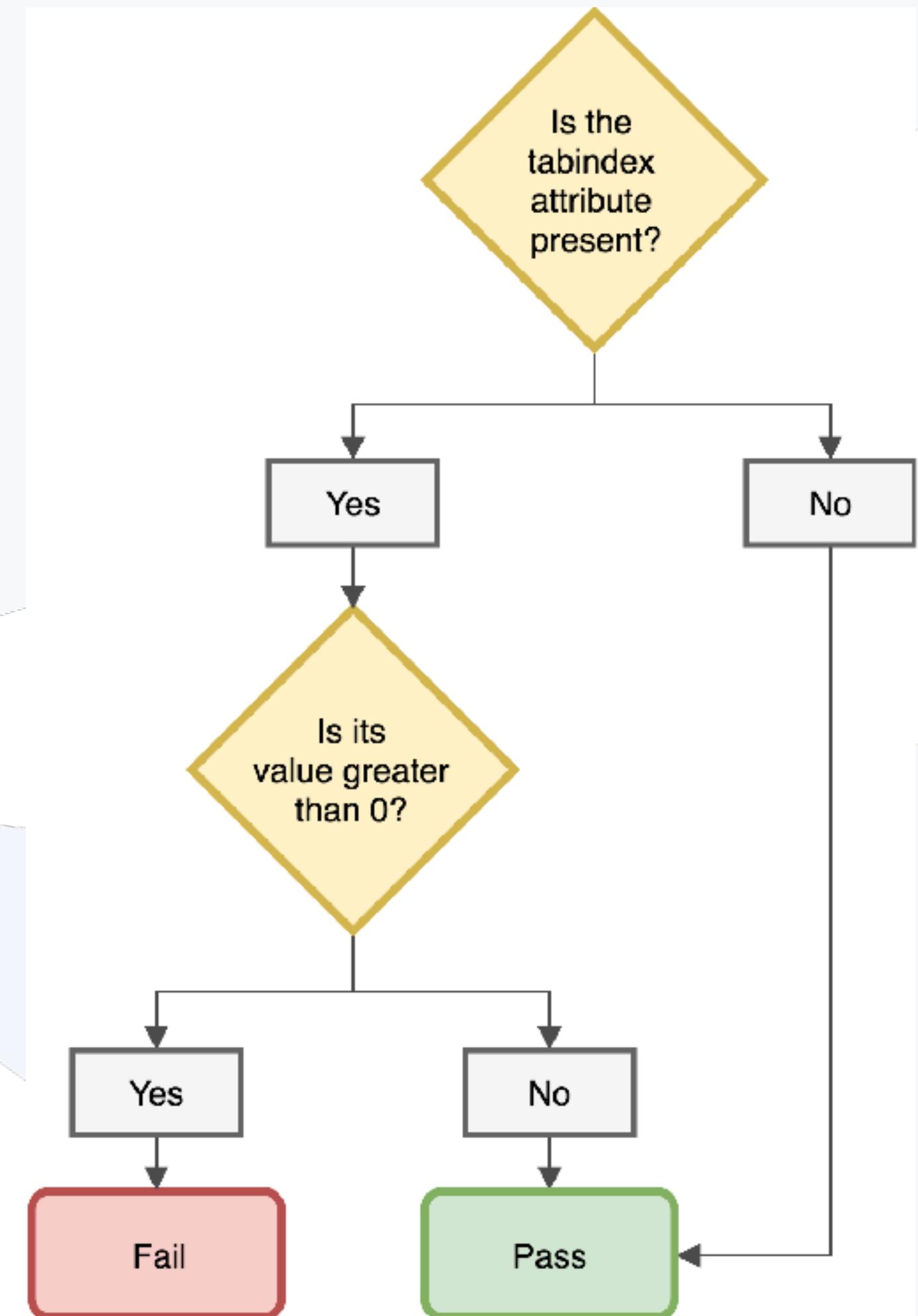
How web accessibility can make impaired people use your website

# Testing Web Accessibility

# Testing Web Accessibility

Automated a11y tests can free up your QA team from manual testing every part of your application...but...they can't automatically make your site accessible.

Use automated a11y tests as one step of a larger testing process. Don't forget that only 20% to 50% of all accessibility issues can automatically be detected



# I've created a simple application with 3 tiny components and a bunch of errors in the HTML

The screenshot shows a code editor with four files open:

- App.js**:

```
function App() {
  return (
    <div className="App">
      <button classes="button" text="Click me NOW" />
      <br />
      <br />
      <button role="button">CLICK ME</button>
      <br />
      <br />
      <div onClick={() => {}} />
      <div tabIndex={0} />
      <br />
      <br />
      <FakeButton classes="button" text="Click me NOW" />
      <br />
      <br />
      <a aria-label="Continue">Click me NOW</a>
      <br />
      <a aria-label="Continue" role="button">
        Click me NOW
      </a>
      <br />
      <a role="myCustomRole">Click me NOW</a>
      <a/>
      <br />
      <br />
      <br />
      <br />
      <br />
      <Image src="https://cms-assets.tutsplus.com/uploads/users/30/posts/29042/previ
      <br />
      <br />
       {
  return <button className={props.classes}>{props.text}</button>;
};
export default Button;
```
- fakeButton.jsx**:

```
import React from "react";
const FakeButton = props => {
  return (
    <a href={props.href} className={props.classes} role="button">
      {props.text}
    </a>
  );
};
export default FakeButton;
```
- image.jsx**:

```
import React from "react";
const Image = props => {
  return <img width="200px" src={props.src} alt={props.alt} />;
};
export default Image;
```

The code in **App.js** contains many errors, particularly in the use of custom attributes like `classes`, `onClick`, `tabIndex`, and `aria-label`. The other three files (**button.jsx**, **fakeButton.jsx**, and **image.jsx**) are valid React components.

# I've created a simple application with 3 tiny components and a bunch of errors in the HTML

The screenshot shows a dark-themed code editor with four tabs open:

- App.js**: This tab contains the majority of the code. It defines a `function App()` that returns a complex structure of HTML-like elements. Many of these elements have errors underlined with red squiggly lines, including `<button classes="button" text="Click me NOW" />`, `<button role="button">CLICK ME</button>`, `<div onClick={() => {}} />`, `<div tabIndex={0} />`, `<FakeButton classes="button" text="Click me NOW" />`, `<a aria-label="Continue">Click me NOW</a>`, `<a aria-label="Continue" role="button">Click me NOW</a>`, `<a role="myCustomRole">Click me NOW</a>`, and `<Image src="https://cms-assets.tutsplus.com/uploads/users/30/posts/29042/preview.jpg" alt="Preview image" />`. The code is heavily annotated with line numbers from 19 to 59.
- button.jsx**: A small component definition:

```
import React from "react";
const Button = props => {
  return <button className={props.classes}>{props.text}</button>;
};
export default Button;
```

- fakeButton.jsx**: A component definition using an `a` tag:

```
import React from "react";
const FakeButton = props => {
  return (
    <a href={props.href} className={props.classes} role="button">
      {props.text}
    </a>
  );
};
export default FakeButton;
```

- image.jsx**: A component definition using an `img` tag:

```
import React from "react";
const Image = props => {
  return <img width="200px" src={props.src} alt={props.alt} />;
};
export default Image;
```

The status bar at the bottom indicates the file is "testing-web-a11y", has 3 lines and 1 character, 2 spaces, LF line endings, and is a JavaScript React file. Prettier is checked.

# Testing the code

- **react-axe**

<https://github.com/dequelabs/react-axe>

- **eslint-plugin-jsx-a11y**

<https://github.com/evcohen/eslint-plugin-jsx-a11y>

- **jest-axe**

<https://github.com/nickcolley/jest-axe>

# react-axe

Install it as a dev dependency



```
npm install --save-dev react-axe
```

Call the exported function passing in the React and ReactDOM objects as well as a timing delay in milliseconds that will be observed between each component change and the time

```
1 var React = require('react');
2 var ReactDOM = require('react-dom');
3
4
5 if (process.env.NODE_ENV !== "production") {
6   var axe = require("react-axe");
7   axe(React, ReactDOM, 1000);
8 }
```

Results will show in the Chrome DevTools console.

A severity level is also assigned for each violation. The possible levels are:

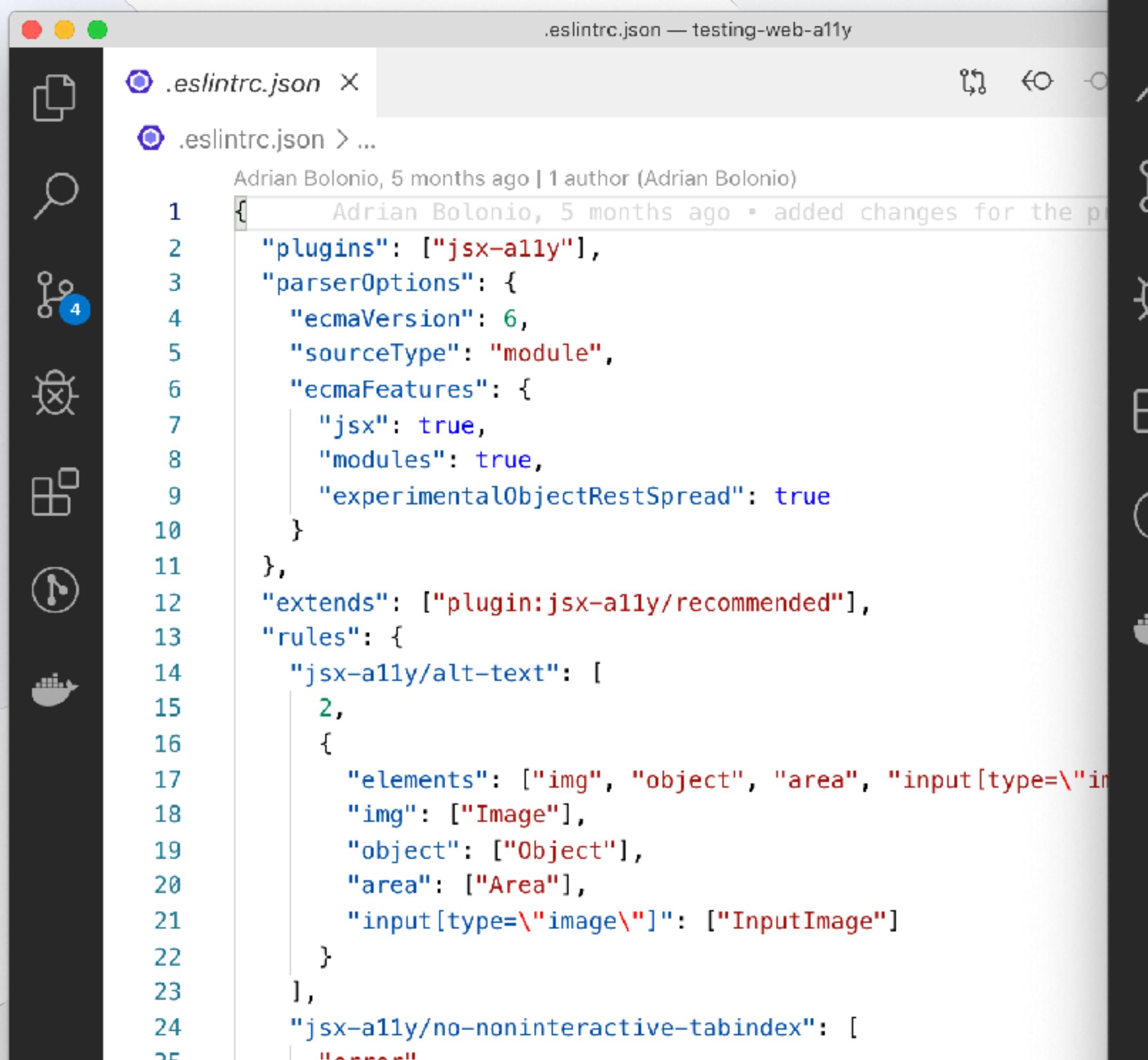
- Minor
- Moderate
- Serious
- Critical

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The URL is set to 'localhost:3000'. The 'axe' tab is active, showing a list of 24 new axe issues. The issues are categorized by severity: critical, serious, moderate, and minor. Each issue includes the element path, the specific rule violated, and a link to the rule's documentation. The 'index.js' file is mentioned multiple times as the source for these violations.

Severity	Description	Source
critical	ARIA roles used must conform to valid values	index.js:154 index.js:180
serious	Elements must have sufficient color contrast	index.js:72 index.js:180
moderate	Images must have alternate text	index.js:72 index.js:180
minor	Form elements must have labels	index.js:72 index.js:180

# eslint-plugin-jsx-a11y

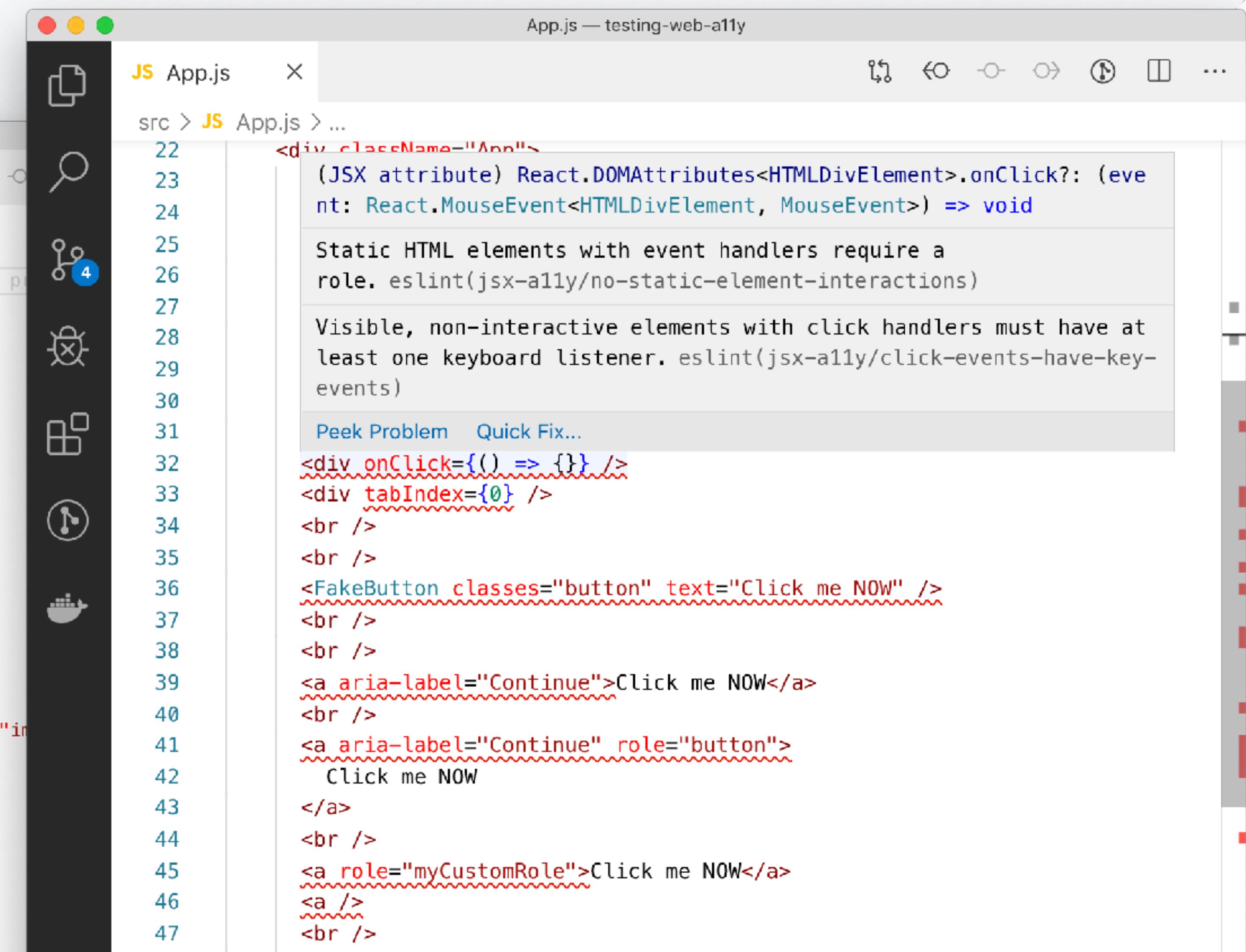
Configure the rules you want to use under the rules section in the `.eslintrc.json` file



A screenshot of a code editor showing the `.eslintrc.json` configuration file. The file contains rules for the `jsx-a11y` plugin. Key sections include `plugins`, `parserOptions`, `ecmaVersion`, `sourceType`, `ecmaFeatures`, `extends`, and `rules`. The `rules` section includes configurations for `jsx-a11y/alt-text` and `jsx-a11y/no-noninteractive-tabindex`.

```
{
  "plugins": ["jsx-a11y"],
  "parserOptions": {
    "ecmaVersion": 6,
    "sourceType": "module",
    "ecmaFeatures": {
      "jsx": true,
      "modules": true,
      "experimentalObjectRestSpread": true
    }
  },
  "extends": ["plugin:jsx-a11y/recommended"],
  "rules": {
    "jsx-a11y/alt-text": [
      2,
      {
        "elements": ["img", "object", "area", "input[type=\"image\"]"],
        "img": ["Image"],
        "object": ["Object"],
        "area": ["Area"],
        "input[type=\"image\"]": ["InputImage"]
      }
    ],
    "jsx-a11y/no-noninteractive-tabindex": [
      "error"
    ]
  }
}
```

The linting issues will be displayed directly in the code editor

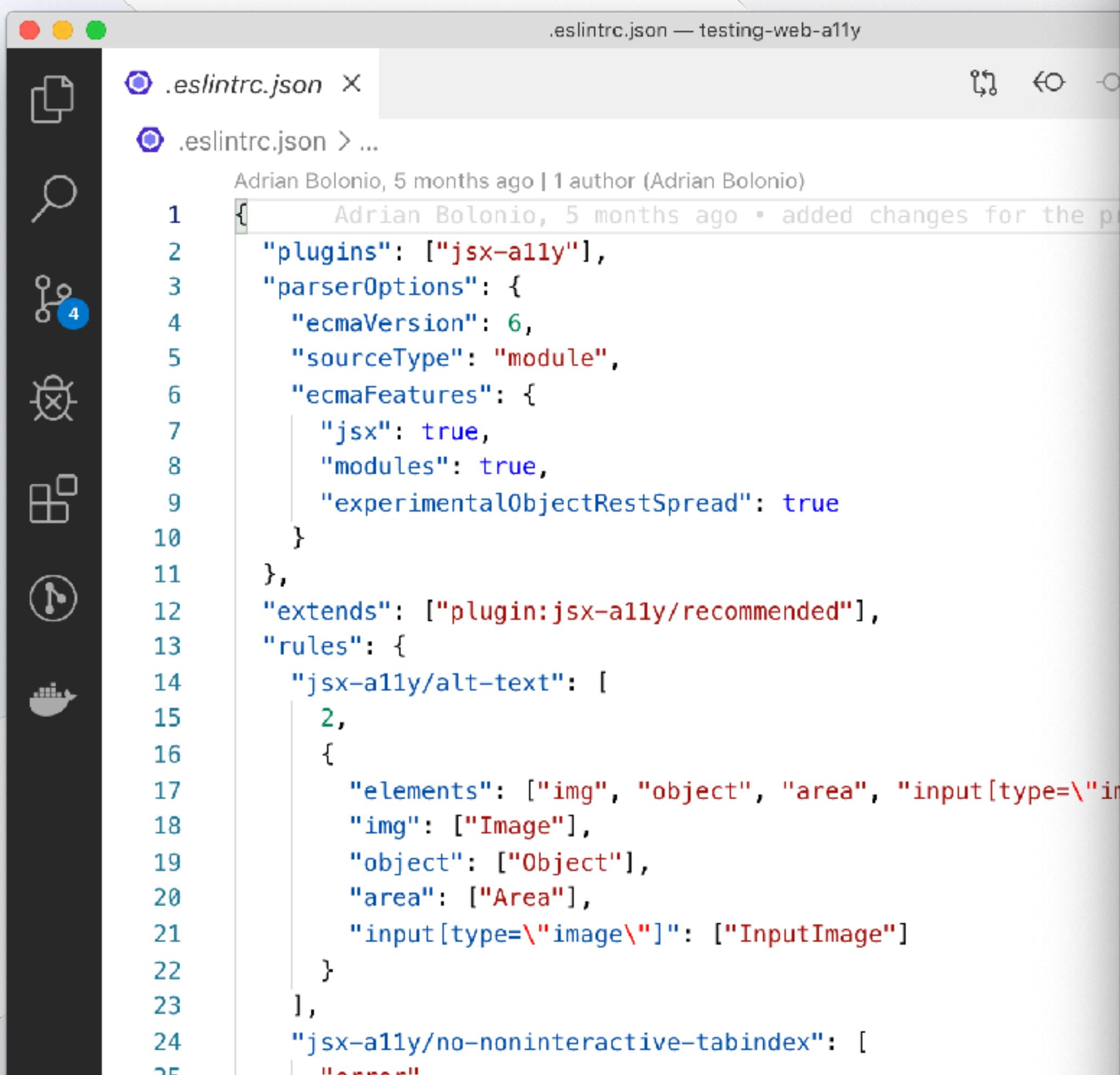


A screenshot of a code editor showing the `App.js` file with several linting issues highlighted by red squiggly lines. The issues are related to static HTML elements with event handlers and visible, non-interactive elements with click handlers. The `onClick` attribute on the first `div` is flagged as problematic.

```
<div className="App">
  (JSX attribute) React.DOMAttributes<HTMLDivElement>.onClick?: (event: React.MouseEvent<HTMLDivElement, MouseEvent>) => void
  Static HTML elements with event handlers require a role. eslint/jsx-a11y/no-static-element-interactions
  Visible, non-interactive elements with click handlers must have at least one keyboard listener. eslint/jsx-a11y/click-events-have-key-events
<div onClick={() => {}}>
<div tabIndex={0}>
<br />
<br />
<FakeButton classes="button" text="Click me NOW" />
<br />
<br />
<a aria-label="Continue">Click me NOW</a>
<br />
<a aria-label="Continue" role="button">
  Click me NOW
</a>
<br />
<a role="myCustomRole">Click me NOW</a>
<a />
<br />
```

# eslint-plugin-jsx-a11y

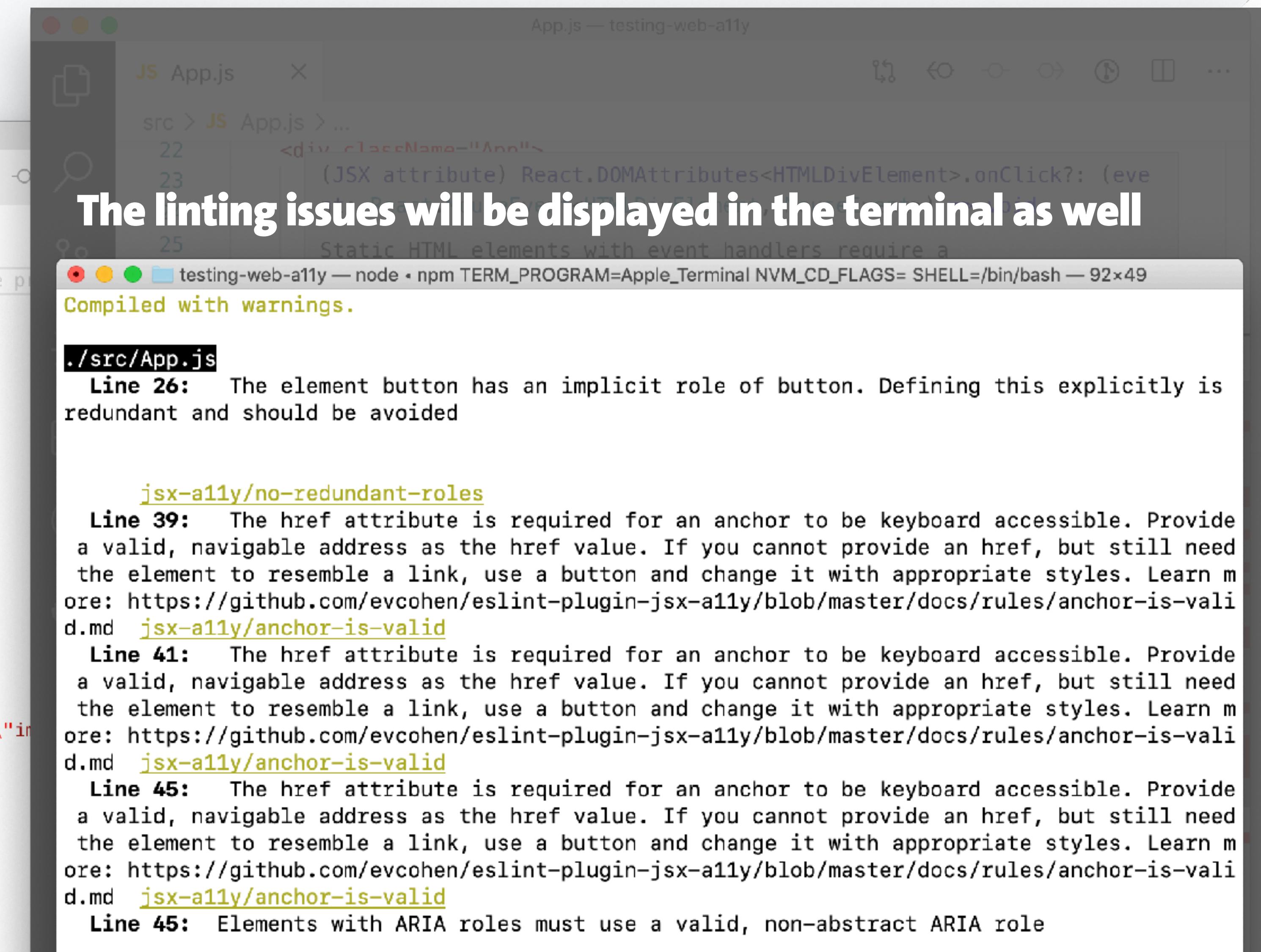
Configure the rules you want to use under the rules section in the `.eslintrc.json` file



A screenshot of a code editor showing the `.eslintrc.json` configuration file. The file contains rules for the `jsx-a11y` plugin, including `plugins`, `parserOptions`, `extends`, and `rules` sections.

```
1 {  
2   "plugins": ["jsx-a11y"],  
3   "parserOptions": {  
4     "ecmaVersion": 6,  
5     "sourceType": "module",  
6     "ecmaFeatures": {  
7       "jsx": true,  
8       "modules": true,  
9       "experimentalObjectRestSpread": true  
10    },  
11  },  
12  "extends": ["plugin:jsx-a11y/recommended"],  
13  "rules": {  
14    "jsx-a11y/alt-text": [  
15      2,  
16      {  
17        "elements": ["img", "object", "area", "input[type=\"image\"]"],  
18        "img": ["Image"],  
19        "object": ["Object"],  
20        "area": ["Area"],  
21        "input[type=\"image\"]": ["InputImage"]  
22      }  
23    ],  
24    "jsx-a11y/no-noninteractive-tabindex": [  
25      "error"  
26    ]  
27  }  
28}
```

The linting issues will be displayed directly in the code editor



A screenshot showing the code editor and terminal output for the `App.js` file. The code editor shows the `App.js` file with ESLint errors highlighted. The terminal shows the command `node` being run, followed by a warning message and a list of ESLint errors.

App.js — testing-web-a11y

```
JS App.js X  
src > JS App.js > ...  
22 <div className="App">  
23   (JSX attribute) React.DOMAttributes<HTMLDivElement>.onClick?: (eve  
25 Static HTML elements with event handlers require a
```

The linting issues will be displayed in the terminal as well

```
testing-web-a11y — node • npm TERM_PROGRAM=Apple_Terminal NVM_CD_FLAGS= SHELL=/bin/bash — 92×49  
Compiled with warnings.  
  
./src/App.js  
Line 26:  The element button has an implicit role of button. Defining this explicitly is redundant and should be avoided  
  
  jsx-a11y/no-redundant-roles  
Line 39:  The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/evcohen/eslint-plugin-jsx-a11y/blob/master/docs/rules/anchor-is-valid.md  jsx-a11y/anchor-is-valid  
Line 41:  The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/evcohen/eslint-plugin-jsx-a11y/blob/master/docs/rules/anchor-is-valid.md  jsx-a11y/anchor-is-valid  
Line 45:  The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/evcohen/eslint-plugin-jsx-a11y/blob/master/docs/rules/anchor-is-valid.md  jsx-a11y/anchor-is-valid  
Line 45:  Elements with ARIA roles must use a valid, non-abstract ARIA role
```

# jest-axe

Install it as a dev dependency



```
npm install --save-dev jest-axe
```

Write an accessibility test using **ReactDOMServer** to render the html of your application and check for violations

```
App.test.js — testing-web-a11y
App.test.js X
src > App.test.js > ...
Adrian Bolonio, 5 months ago | 1 author (Adrian Bolonio)
1 import { axe, toHaveNoViolations } from "jest-axe";
2 import React from "react";
3 import ReactDOMServer from "react-dom/server";
4 import App from "./App";
5 expect.extend(toHaveNoViolations);
6
7
8 it("should demonstrate this matcher's usage with react", async () => {
9   const html = ReactDOMServer.renderToString(<App />);
10  const results = await axe(html);
11  expect(results).toHaveNoViolations();
12 });
13
```

The results of the test will be displayed in the terminal

```
FAIL src/App.test.js
✖ should demonstrate this matcher's usage with react (605ms)

• should demonstrate this matcher's usage with react

expect(received).toHaveNoViolations(expected)

Expected the HTML found at $('a[role="myCustomRole"]') to have no violations:

<a role="myCustomRole">Click me NOW</a>

Received:

"ARIA roles used must conform to valid values (aria-roles)"

Try fixing it with this help: https://dequeuniversity.com/rules/axe/3.1/aria-roles?application=axeAPI

_____

Expected the HTML found at $('img:nth-child(28)') to have no violations:

 input') to have no violations:

<input type="text">

Expected the HTML found at $('label:nth-child(40) > input') to have no violations:

<input type="text">

Received:

"Form elements must have labels (label)"
```

# Testing the DOM

- **axe-cli**

<https://github.com/dequelabs/axe-cli>

- **pa11y**

<http://pa11y.org/>

- **Lighthouse**

<https://developers.google.com/web/tools/lighthouse/>

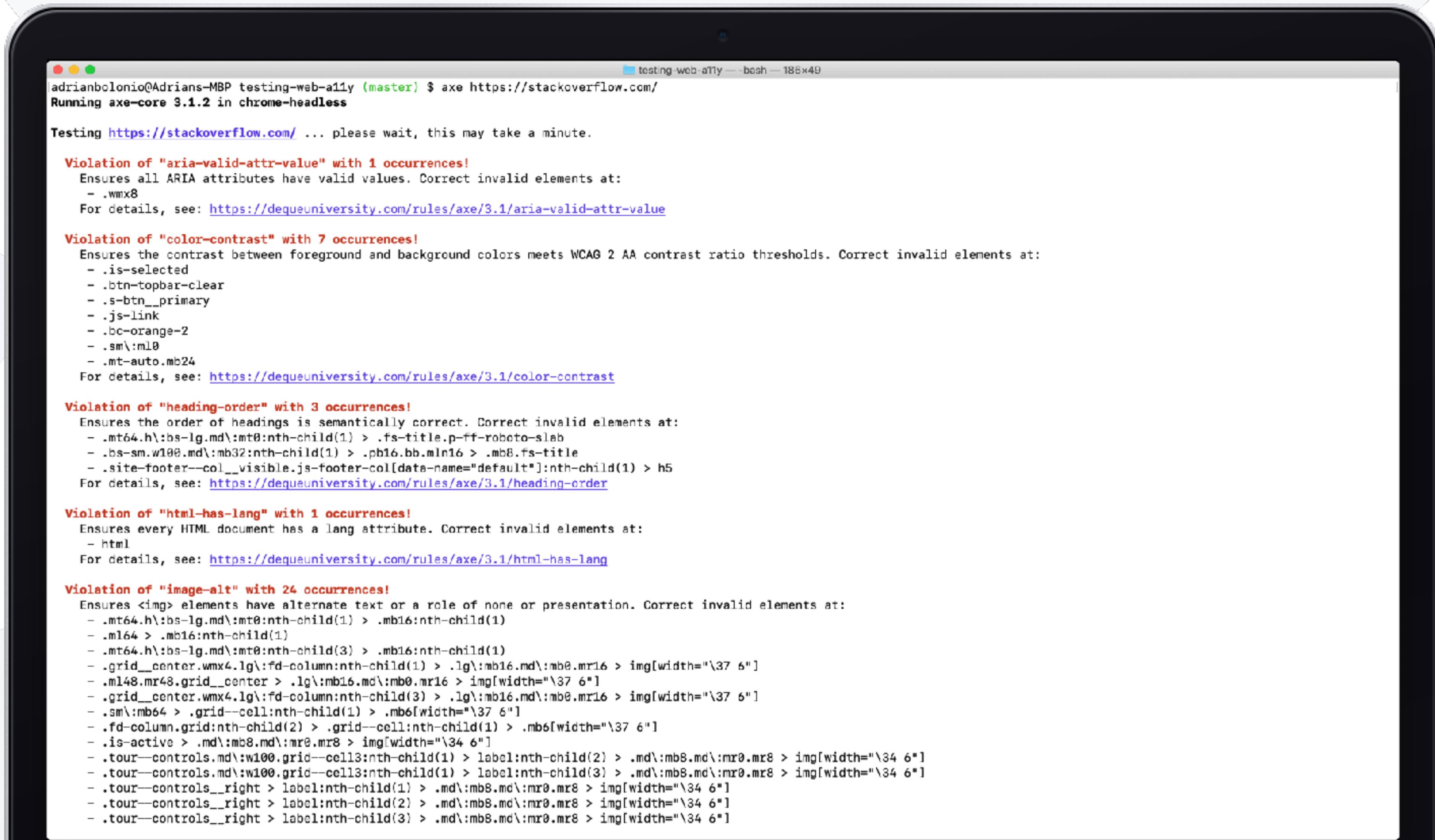
# axe-cli

Install axe-cli globally



```
npm install axe-cli -g
```

You can now run the **axe** command in your CLI, followed by the URL of the page you wish to test



```
adrianbolonio@Adrians-MBP testing-web-ally (master) $ axe https://stackoverflow.com/
Running axe-core 3.1.2 in chrome-headless

Testing https://stackoverflow.com/ ... please wait, this may take a minute.

Violation of "aria-valid-attr-value" with 1 occurrences!
Ensures all ARIA attributes have valid values. Correct invalid elements at:
- .wmx8
For details, see: https://dequeuniversity.com/rules/axe/3.1/aria-valid-attr-value

Violation of "color-contrast" with 7 occurrences!
Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds. Correct invalid elements at:
- .is-selected
- .btn-topbar-clear
- .s-btn__primary
- .js-link
- .bc-orange-2
- .sm\::ml0
- .mt-auto.mb24
For details, see: https://dequeuniversity.com/rules/axe/3.1/color-contrast

Violation of "heading-order" with 3 occurrences!
Ensures the order of headings is semantically correct. Correct invalid elements at:
- .mt64.h\::bs-lg.md\::mt0:nth-child(1) > .fs-title.p-ff-roboto-slab
- .bs-sm.w100.md\::mb32:nth-child(1) > .pb16.bb.mln16 > .mb8.fs-title
- .site-footer--col__visible.js-footer-col[data-name="default"]:nth-child(1) > h5
For details, see: https://dequeuniversity.com/rules/axe/3.1/heading-order

Violation of "html-has-lang" with 1 occurrences!
Ensures every HTML document has a lang attribute. Correct invalid elements at:
- html
For details, see: https://dequeuniversity.com/rules/axe/3.1/html-has-lang

Violation of "image-alt" with 24 occurrences!
Ensures <img> elements have alternate text or a role of none or presentation. Correct invalid elements at:
- .mt64.h\::bs-lg.md\::mt0:nth-child(1) > .mb16:nth-child(1)
- .ml64 > .mb16:nth-child(1)
- .mt64.h\::bs-lg.md\::mt0:nth-child(3) > .mb16:nth-child(1)
- .grid_center.wmx4.lg\::fd-column:nth-child(1) > .lg\::mb16.md\::mb0.mr16 > img[width="\37 6"]
- .ml48.mr48.grid_center > .lg\::mb16.md\::mb0.mr16 > img[width="\37 6"]
- .grid_center.wmx4.lg\::fd-column:nth-child(3) > .lg\::mb16.md\::mb0.mr16 > img[width="\37 6"]
- .sm\::mb64 > .grid--cell:nth-child(1) > .mb6[width="\37 6"]
- .fd-column.grid--child(2) > .grid--cell:nth-child(1) > .mb6[width="\37 6"]
- .is-active > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
- .tour--controls.md\::w100.grid--cell3:nth-child(1) > label:nth-child(2) > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
- .tour--controls.md\::w100.grid--cell3:nth-child(1) > label:nth-child(3) > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
- .tour--controls__right > label:nth-child(1) > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
- .tour--controls__right > label:nth-child(2) > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
- .tour--controls__right > label:nth-child(3) > .md\::mb8.md\::mr0.mr8 > img[width="\34 6"]
```

# pa11y

## Install pa11y globally

```
npm install -g pa11y
```

You can now run the pa11y command in your CLI, followed by the URL of the page you wish to test

```
adrianbolonio@Adrians-MBP testing-web-a11y (master) $ pa11y https://stackoverflow.com/
(node:50633) ExperimentalWarning: The fs.promises API is experimental

Welcome to Pa11y
> Running Pa11y on URL https://stackoverflow.com/

Results for URL: https://stackoverflow.com/

• Error: The html element should have a lang or xml:lang attribute which describes the language of the document.
  └── WCAG2AA.Principle3.Guideline3_1.3_1.H57.2
    ├── html
    └── <html class="html__responsive html__unpinned-leftnav"><head> <title>Stack Ov...</html>

• Error: Anchor element found with a valid href attribute, but no link content has been supplied.
  └── WCAG2AA.Principle4.Guideline4_1.4_1.H91.A.NoContent
    ├── html > body > header > div > div:nth-child(1) > a:nth-child(1)
    └── <a href="#" class="left-sidebar-toggle p0 ai-center jc-center js-left-sidebar-toggle" role="menuitem" aria-haspopup="true" aria-controls="left-sidebar" aria-expanded="false"><span class="ps-relative"></span>
```

• Error: This form does not contain a submit button, which creates issues for those who cannot submit the form using the keyboard. Submit buttons are INPUT elements with type attribute "submit" or "image", or BUTTON elements with type "submit" or omitted/invalid.
 └── WCAG2AA.Principle3.Guideline3\_2.3\_2.H32.2
 ├── #search
 └── <form id="search" role="search" action="/search" method="get" class="grid--cell fl-grow1 searchbar px12 js-searchbar" autocomplete="off"> <div class...</form>

• Error: Anchor element found with no link content and no name and/or ID attribute.
 └── WCAG2AA.Principle4.Guideline4\_1.4\_1.H91.A.EmptyNoId
 ├── #js-gdpr-consent-banner > div > a
 └── <a class="grid--cell fc-white js-notice-close" aria-label="notice-dismiss"> <svg aria-hidd...</a>

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 4.16:1. Recommendation: change background to #353a3f.
 └── WCAG2AA.Principle1.Guideline1\_4.1\_3.G18.Fail
 ├── #announcement-banner > div > div > a
 └── <a target="\_blank" class="s-link js-link js-gps-track fw-bold fc-blue-400" href="https://stackoverflow.blog/2019/11/13/were-rewarding-the-question-askers/" data-ga="["rep-re calc-2019","Announcement Banner","https://stackover..."]">

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.64:1. Recommendation: change text colour to #c35600.
 └── WCAG2AA.Principle1.Guideline1\_4.1\_3.G18.Fail
 ├── #content > div:nth-child(1) > div > a:nth-child(1)
 └── <a class="grid--cell bc-orange-2 ba fc-orange-400 bg-white py12 px24 bar-sm js-scroll-link js-gps-track" data-gps-track="product\_homepage.hero.click({ cta: 'For developers' })" href="#for-developers" data-ga="["home page","for develop..."]">

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.64:1. Recommendation: change background to #c35600.
 └── WCAG2AA.Principle1.Guideline1\_4.1\_3.G18.Fail
 ├── #content > div:nth-child(1) > div > a:nth-child(2)

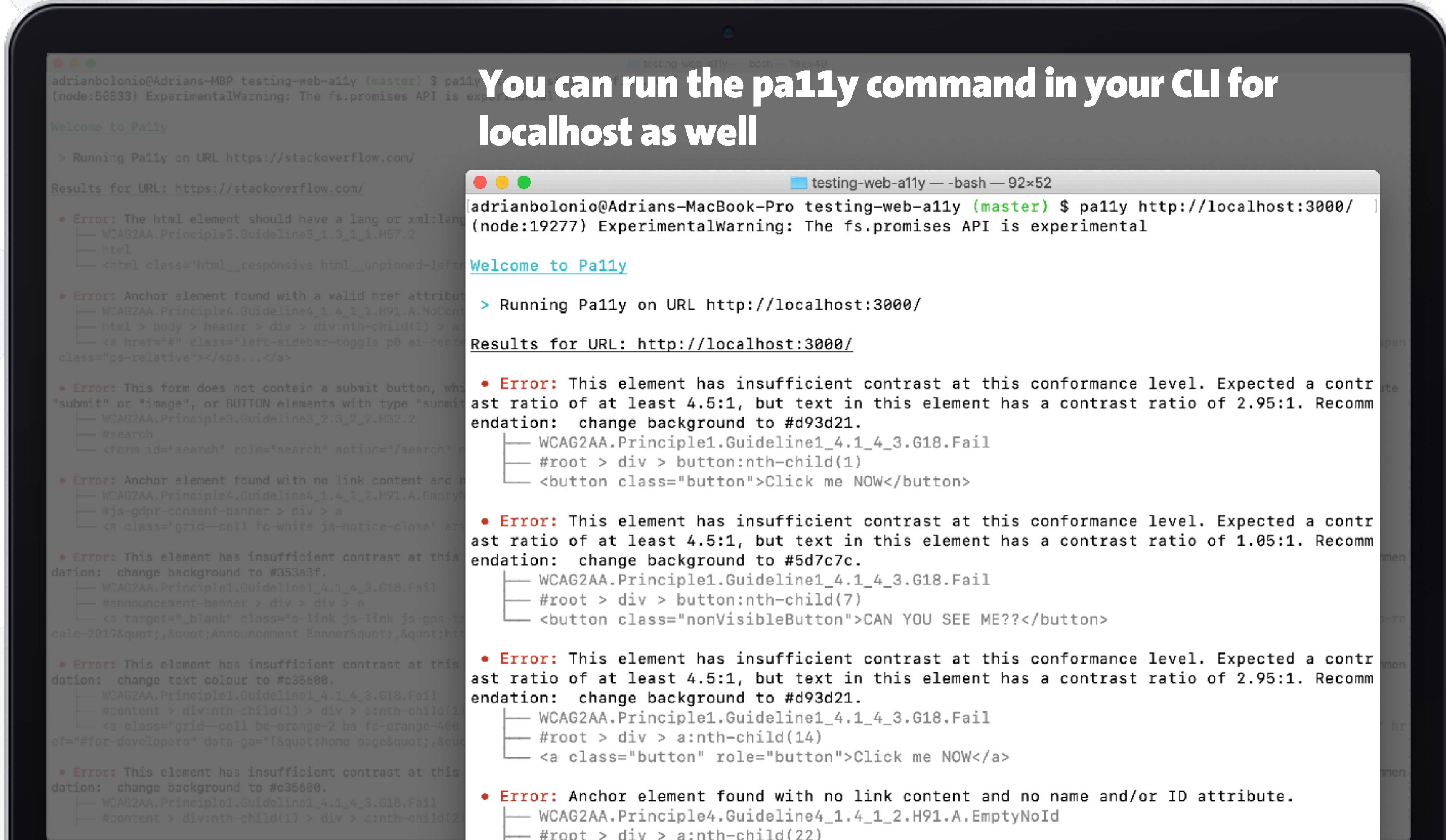
# pa11y

## Install pa11y globally



```
npm install -g pa11y
```

You can now run the pa11y command in your CLI, followed by the URL of the page you wish to test



You can run the pa11y command in your CLI for localhost as well

```
adrianbolonio@Adrians-MBP testing-web-a11y (master) $ pa11y https://stackoverflow.com/
(node:50833) ExperimentalWarning: The fs.promises API is experimental

Welcome to Pa11y

> Running Pa11y on URL https://stackoverflow.com/

Results for URL: https://stackoverflow.com/
• Error: The html element should have a lang or xml:lang attribute.
  └── WCAG2AA.Principle3.Guideline3_1_3_1.H57.2
      └── html
          └── <html class="html__responsive html__unpinned-left">

• Error: Anchor element found with a valid href attribute.
  └── WCAG2AA.Principle4.Guideline4_1_4_1_2.H91.A.NoContent
      └── html > body > header > div > div:nth-child(1) > a
          └── <a href="#" class="left-sidebar-toggle p0 ai-center ps-relative"></a>

• Error: This form does not contain a submit button, which is required for forms to be accessible. Forms must contain either a "submit" or "image", or BUTTON elements with type "submit".
  └── WCAG2AA.Principle3.Guideline3_2_3_2.H32.2
      └── #search
          └── <form id="search" role="search" action="/search">

• Error: Anchor element found with no link content and no name and/or ID attribute.
  └── WCAG2AA.Principle4.Guideline4_1_4_1_2.H91.A.EmptyNoId
      └── #js-gdpr-consent-banner > div > a
          └── <a class="grid-cell fc-white js-notice-close" href="#">Close</a>

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.95:1. Recommendation: change background to #d93d21.
  └── WCAG2AA.Principle1.Guideline1_4_1_4_3.G18.Fail
      └── #root > div > button:nth-child(1)
          └── <button class="button">Click me NOW</button>

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 1.05:1. Recommendation: change background to #5d7c7c.
  └── WCAG2AA.Principle1.Guideline1_4_1_4_3.G18.Fail
      └── #announcement-banner > div > div > a
          └── <a target="_blank" class="s-link js-link js-gps-track" href="https://www.stackoverflow.com/help/announcements?utm_source=gdpr&utm_medium=consent&utm_campaign=gdpr&utm_content=consent&utm_term=.Announcement Banner">CAN YOU SEE ME??</a>

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.95:1. Recommendation: change background to #d93d21.
  └── WCAG2AA.Principle1.Guideline1_4_1_4_3.G18.Fail
      └── #root > div > a:nth-child(14)
          └── <a class="button" role="button">Click me NOW</a>

• Error: Anchor element found with no link content and no name and/or ID attribute.
  └── WCAG2AA.Principle4.Guideline4_1_4_1_2.H91.A.EmptyNoId
      └── #root > div > a:nth-child(22)
```

# pa11y-ci

Install pa11y-ci globally

```
npm install -g pa11y-ci
```

```
{}
{} .pa11yci.json X
{} .pa11yci.json > ...
You, a few seconds ago | 2 authors (You and others)
1 {
2   "urls": [
3     "https://stackoverflow.com/",
4     {
5       "url": "https://stackoverflow.com/",
6       "timeout": 5000,
7       "screenCapture": "captures-stackoverflow.png"
8     },
9     {
10    "url": "https://stackoverflow.com/",
11    "timeout": 5000,
12    "actions": [
13      "screen capture captures-stackoverflow-home.png",
14      "navigate to https://stackoverflow.com/questions",
15      "screen capture captures-stackoverflow-questions.png"
16    ]
17  }
18]
19}
20
```

By default, Pa11y CI looks for a config file (.json) in the current working directory. Each URL in your config file can be an object and specify pa11y configurations which override the defaults too.

```
adrianbolonio@Adrians-MBP testing-web-a11y (master) $ pa11y-ci
(node:50843) ExperimentalWarning: The fs.promises API is experimental
Running Pa11y on 3 URLs:
> https://stackoverflow.com/ - 82 errors
> https://stackoverflow.com/ - 82 errors
> https://stackoverflow.com/ - 392 errors

Errors in https://stackoverflow.com/:
● The html element should have a lang or xml:lang attribute which describes the language of the document.
(html)

<html class="html__responsive"><head> <title>Newest Q...</title>
● Anchor element found with a valid href attribute, but no link content has been supplied.
(html > body > header > div > div:nth-child(1) > a:nth-child(1))

<a href="#" class="left-sidebar-toggle p0 ai-center jc-center js-left-sidebar-toggle" role="menuitem" aria-haspopup="true" aria-controls="left-sidebar">
  class="ps-relative"></spa...</a>
● This form does not contain a submit button, which creates issues for those who cannot submit the form using the keyboard. Submit buttons are INPUT elements with type "submit" or "image", or BUTTON elements with type "submit" or omitted/invalid.
(#search)

<form id="search" role="search" action="/search" method="get" class="grid--cell fl-grow1 searchbar px12 js-searchbar" autocomplete="off"> <div class="grid--cell fc-white js-searchbar">
  <input type="text" class="grid--cell fc-white js-searchbar">
  <button type="submit" class="grid--cell fc-white js-searchbar">Search</button>
</div>
● Anchor element found with no link content and no name and/or ID attribute.
(#js-gdpr-consent-banner > div > a)

<a class="grid--cell fc-white js-notice-close" aria-label="notice-dismiss"> <svg aria-hidden="true" class="grid--cell fc-white js-notice-close"></svg>
● This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 1.0:1. Recommendation: change background to #353a3f.
(#announcement-banner > div > div > a)

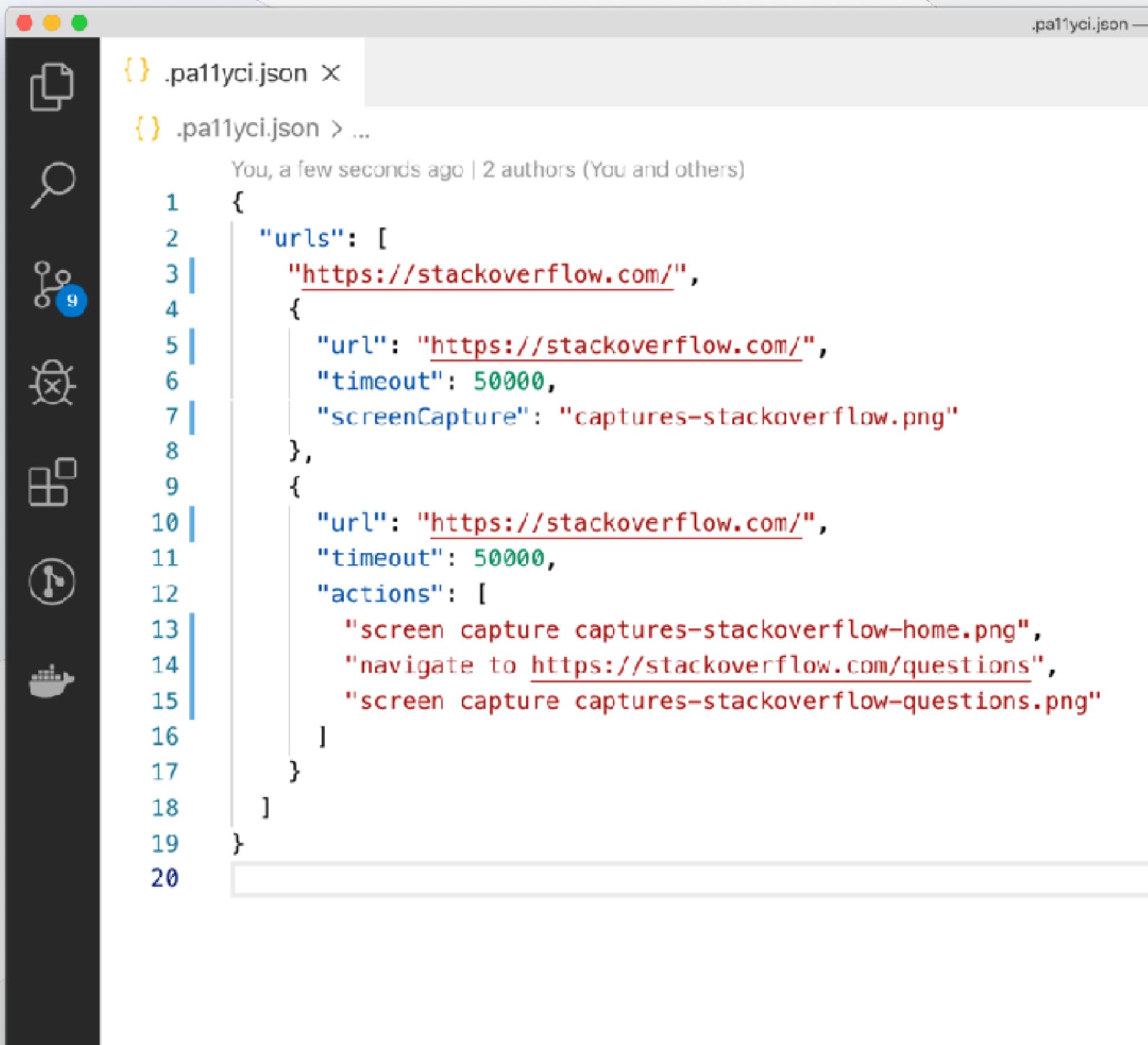
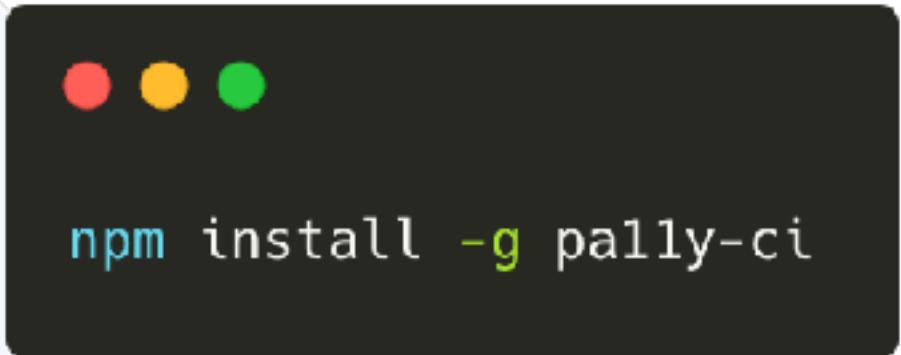
<a target="_blank" class="s-link js-link js-gps-track fw-bold fc-blue-400" href="https://stackoverflow.blog/2019/11/13/were-rewarding-the-question-answering-community-with-new-rewards/&gt; rep-recalc-2019Announcement Banner&quot; href="https://stackover...
● This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 1.0:1. Recommendation: change text colour to #fcfeff.
(#nav-questions > span)

<span class="-link--channel-name">Stack Overflow</span>
```

# pa11y-ci

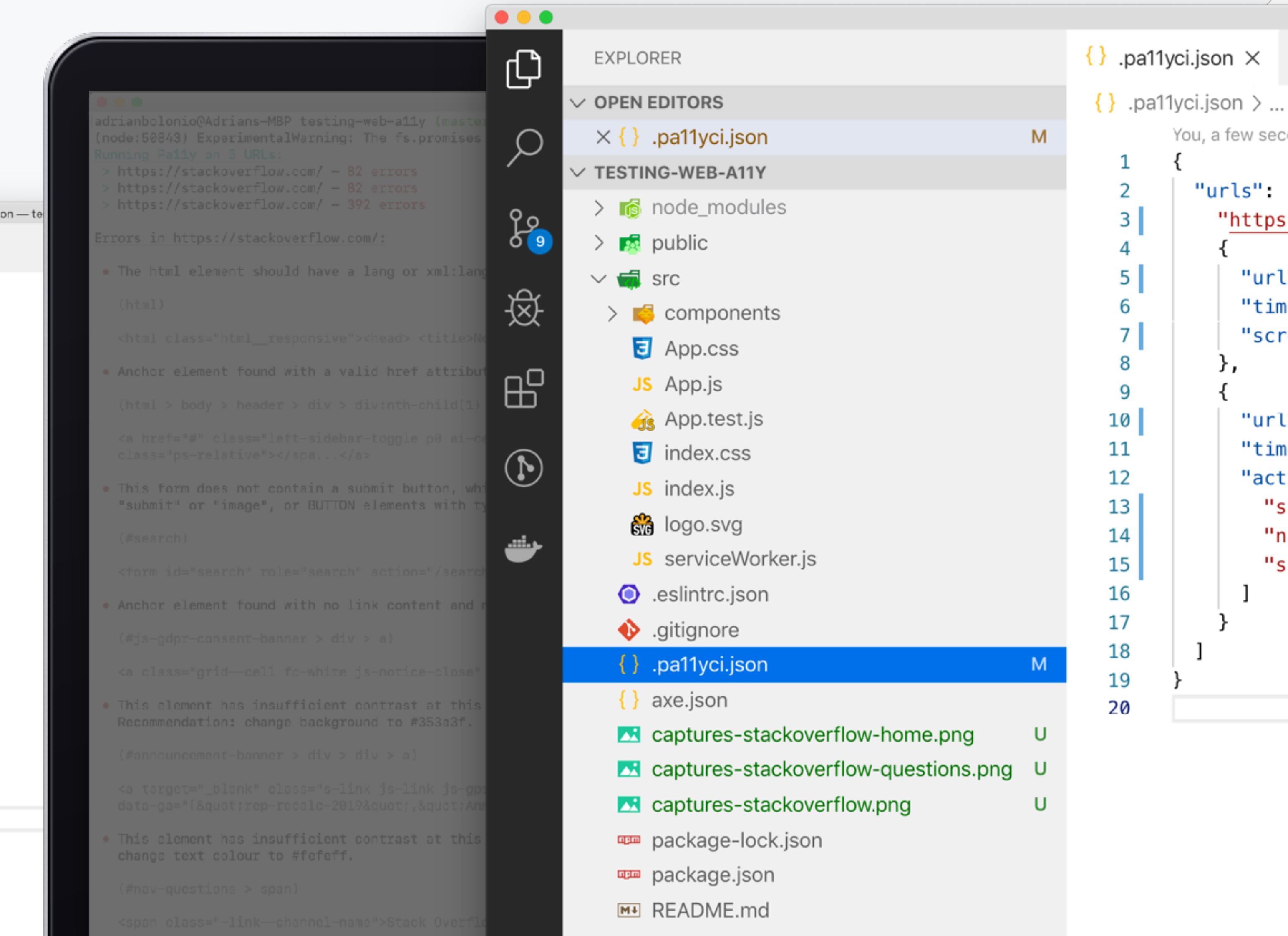
## Install pa11y-ci globally

```
npm install -g pa11y-ci
```



```
{ "urls": [ { "url": "https://stackoverflow.com/", "timeout": 50000, "screenCapture": "captures-stackoverflow.png" }, { "url": "https://stackoverflow.com/questions", "timeout": 50000, "actions": [ "screen capture captures-stackoverflow-home.png", "navigate to https://stackoverflow.com/questions", "screen capture captures-stackoverflow-questions.png" ] } ] }
```

**By default, Pa11y CI looks for a config file (.json) in the current working directory. Each URL in your config file can be an object and specify pa11y configurations which override the defaults too.**



```
{ "urls": [ { "url": "https://www.stackoverflow.com", "timeout": 50000, "screenCapture": "captures-stackoverflow.png" }, { "url": "https://www.stackoverflow.com/questions", "timeout": 50000, "actions": [ "screen capture captures-stackoverflow-home.png", "navigate to https://www.stackoverflow.com/questions", "screen capture captures-stackoverflow-questions.png" ] } ] }
```

# Lighthouse

Install lighthouse globally

```
npm install -g lighthouse
```

You can now run the lighthouse command in the terminal ( with --view) and it will immediately open the generated html report.

The screenshot shows a Mac OS X desktop environment. On the left, a terminal window has the command `npm install -g lighthouse` entered. To its right, a web browser window titled "Lighthouse Report" displays the results of a performance audit for the URL `https://stackoverflow.com/`. The report card at the top shows scores: Performance (75), Accessibility (63), Best Practices (79), SEO (87), and a "PWA" button. Below the card, the "Accessibility" section is expanded, showing a list of opportunities to improve. The "Failing Elements" section lists one failing element with the code snippet `<div class="wnx8 mx-auto grid jc-center" role="alertdialog" aria-describedby="notice-message">`. The "Contrast" section lists two failing items related to contrast ratios. The "Internationalization and localization" section lists one failing item about the `<html>` element missing a `[lang]` attribute. The "Names and labels" section lists one failing item about semantic controls. On the far right of the browser window, the terminal output of the Lighthouse command is visible, showing the audit results and command history.

```
status Evaluating: Interactive controls are keyboard focusable +1ms
status Evaluating: Headings don't skip levels +0ms
status Evaluating: Interactive elements indicate their purpose and state +0ms
status Evaluating: The page has a logical tab order +0ms
status Evaluating: The user's focus is directed to new content added to the page +0ms
status Evaluating: Offscreen content is hidden from assistive technology +0ms
status Evaluating: HTML5 landmark elements are used to improve navigation +0ms
status Evaluating: Visual order on the page follows DOM order +0ms
status Evaluating: Uses efficient cache policy on static assets +1ms
status Evaluating: Avoids enormous network payloads +6ms
status Evaluating: Defer offscreen images +3ms
status Evaluating: Eliminate render-blocking resources +45ms
status Evaluating: Minify CSS +24ms
status Evaluating: Minify JavaScript +108ms
status Evaluating: Remove unused CSS +86ms
status Evaluating: Serve images in next-gen formats +20ms
status Evaluating: Efficiently encode images +14ms
status Evaluating: Enable text compression +9ms
status Evaluating: Properly size images +16ms
status Evaluating: Use video formats for animated content +13ms
status Evaluating: Avoids Application Cache +11ms
status Evaluating: Page has the HTML doctype +1ms
status Evaluating: Avoids an excessive DOM size +3ms
status Evaluating: Links to cross-origin destinations are safe +5ms
status Evaluating: Avoids requesting the geolocation permission on page load +2ms
status Evaluating: Avoids 'document.write()' +1ms
status Evaluating: Avoids front-end JavaScript libraries with known security vulnerabilities +0ms
status Evaluating: Detected JavaScript libraries +9ms
status Evaluating: Avoids requesting the notification permission on page load +1ms
status Evaluating: Allows users to paste into password fields +1ms
status Evaluating: Uses HTTP/2 for its own resources +1ms
status Evaluating: Uses passive listeners to improve scrolling performance +2ms
status Evaluating: Document has a meta description +0ms
status Evaluating: Page has successful HTTP status code +1ms
status Evaluating: Document uses legible font sizes +0ms
status Evaluating: Links have descriptive text +3ms
status Evaluating: Page isn't blocked from indexing +9ms
status Evaluating: robots.txt is valid +17ms
status Evaluating: Tap targets are sized appropriately +2ms
status Evaluating: Document has a valid 'hreflang' +10ms
status Evaluating: Document avoids plugins +2ms
status Evaluating: Document has a valid 'rel=canonical' +2ms
status Evaluating: Structured data is valid +1ms
status Generating results... +0ms
Printer html output written to /Users/adrianbolonio/Documents/workspace/testing-web-a11y/stackoverflow.com_2019-11-14_16-41-59.report.html +145ms
Chromelauncher Killing Chrome instance 50686 +11ms
adrianbolonio@Adrians-MBP testing-web-a11y (master) $
```

# Testing the DOM

## Main feature

---

- **axe-cli**

**Direct link from terminal to documentation at dequeuniversity**

---

- **pa11y**

**pa11y-ci allows to include several urls with different config options that allows you to perform actions in the website**

---

- **Lighthouse**

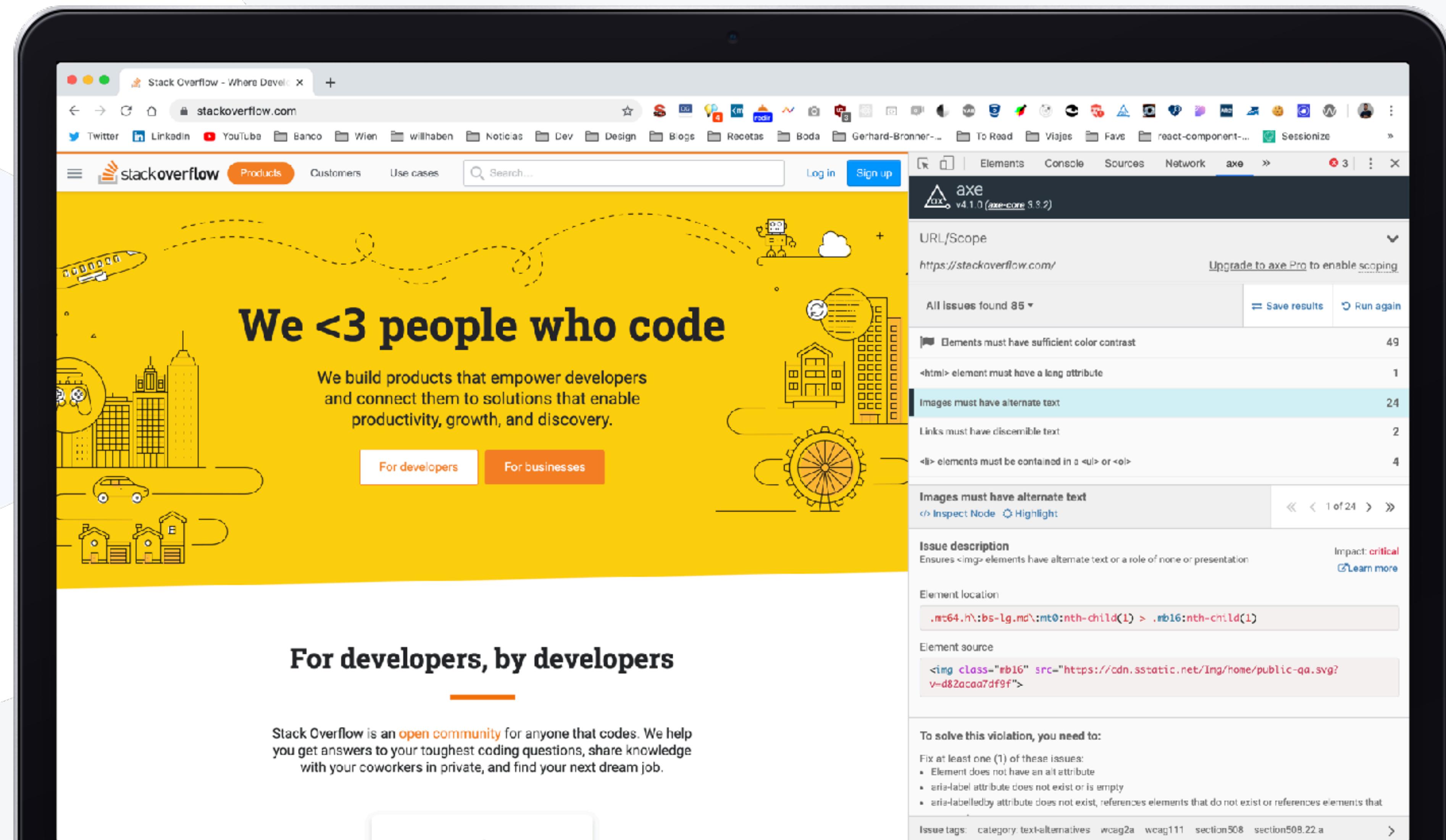
**At the end of the analysis, the tools will generate a final report, either HTML or JSON/CSV the can be use for reporting purposes**

# Testing in the browser

- **axe chrome extension** <https://www.deque.com/axe/>
- **ARC Toolkit** <https://www.paciellogroup.com/toolkit/>
- **Accessibility Insights** <https://accessibilityinsights.io/>
- **WAVE** <https://wave.webaim.org/>
- **NoCoffee** <https://accessgarage.wordpress.com/>

# axe chrome extension

<https://www.deque.com/axe/>



# ARC Toolkit

<https://www.paciellogroup.com/toolkit/>

The screenshot shows the ARC Toolkit extension running in a browser window. The main content area displays the Stack Overflow homepage with a yellow background and the text "We <3 people who code". Below this, there's a section for developers and businesses, and a "For developers, by developers" tagline. The right side of the screen is occupied by the ARC Toolkit interface, which includes a navigation bar with tabs like Elements, Console, Sources, Network, ARC Toolkit, and a search bar. The main panel is titled "ARC Toolkit Version: 3.2.0.0" and contains several sections: "Test Groups", "Visible Assertions", "Test Results", and a "Keyboard Shortcuts" sidebar.

**Test Groups**

	✓	✗	?	!	?	
Totals	741	82	55	360	7	135
Page Info	4	0	0	0	0	0
Media	46	25	19	11	6	6
Images	2	0	0	2	0	0
Rich Media	0	0	0	0	0	0
Headings	24	0	3	8	0	0
Landmarks	5	1	0	1	0	0
Lists	10	0	0	23	0	0
Paragraphs	26	0	1	3	0	0
Pseudo Content	0	0	0	0	0	0
Tables	0	0	0	0	0	0
Forms	0	6	1	5	0	6
Frames	0	0	0	0	0	0
Titles	0	0	0	110	0	4
Text Formatting	3	0	3	5	0	5
Language	1	1	0	0	0	0
Links	70	2	3	132	1	107
Internal Links	2	0	2	0	0	0
Buttons	1	0	0	2	0	0
Access Keys	0	0	0	0	0	0
TabIndex	0	0	0	0	0	0
Tab Order	69	0	0	0	0	0
ARIA UI	6	0	0	5	0	0
ARIA Live	0	0	0	0	0	0
ARIA Usage	29	0	0	19	0	0

**Visible Assertions**

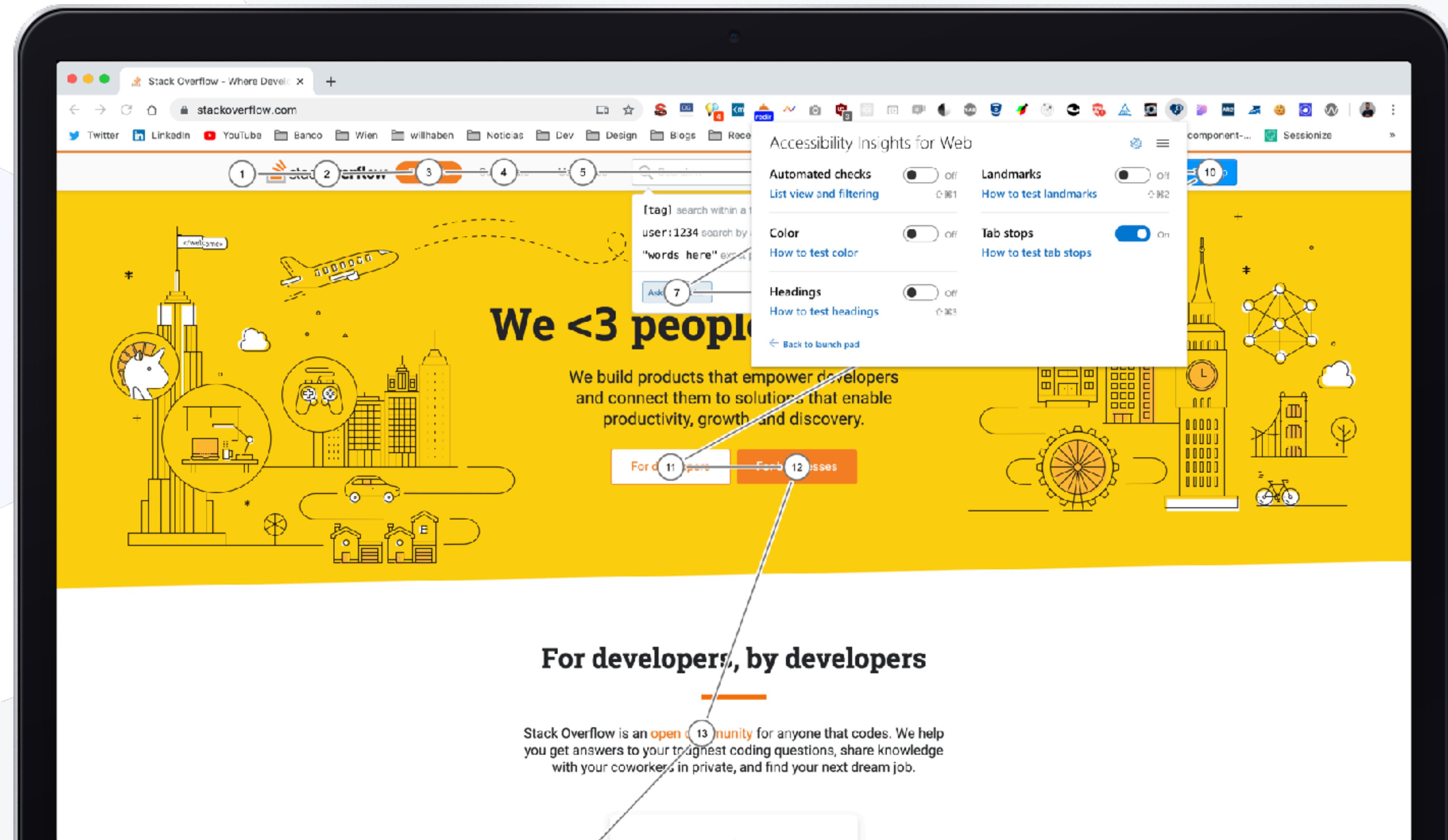
Test	Errors	Warnings	Gr
✓ noAlt	24	0	img
✓ activeSvgMissingAriaLabel	1	0	img
✓ SVGFocusable	0	19	img

**Test Results**

1. **WARN** <svg aria-hidden="true" class="svg-icon s-input-icon s-input-icon\_search iconSearch" width="18" height="18" viewBox="0 0 18 18"></svg> alt: NIL
  - SVGFocusable - Warning: SVG is focusable - Recommend: Set focusable='false'
2. **ERR** 
  - noAlt - Error: no alt - Recommend: add an alt attribute with an appropriate text description
3. **ERR** 
  - noAlt - Error: no alt - Recommend: add an alt attribute with an appropriate text description

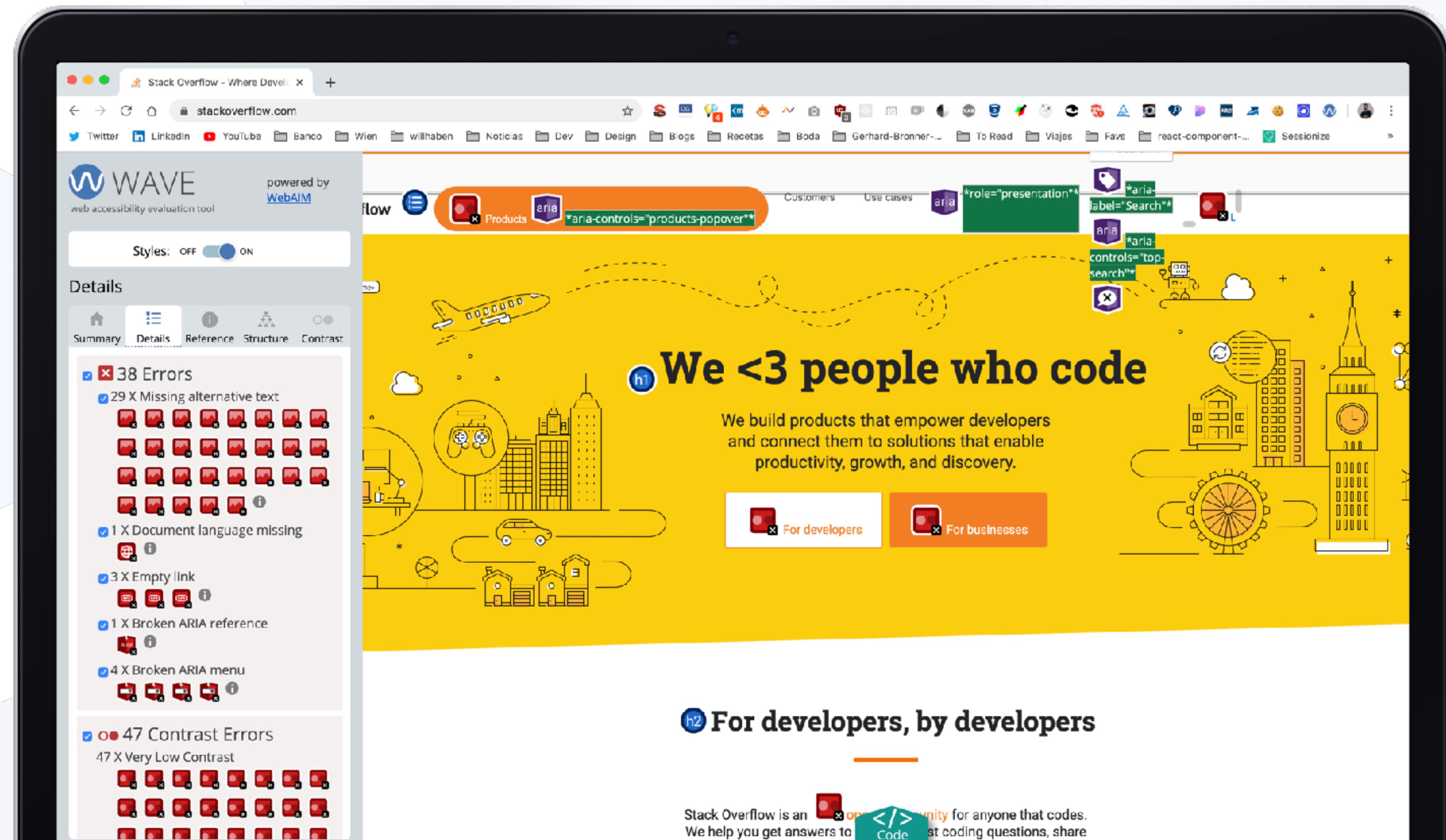
# Accessibility Insights

<https://accessibilityinsights.io/>



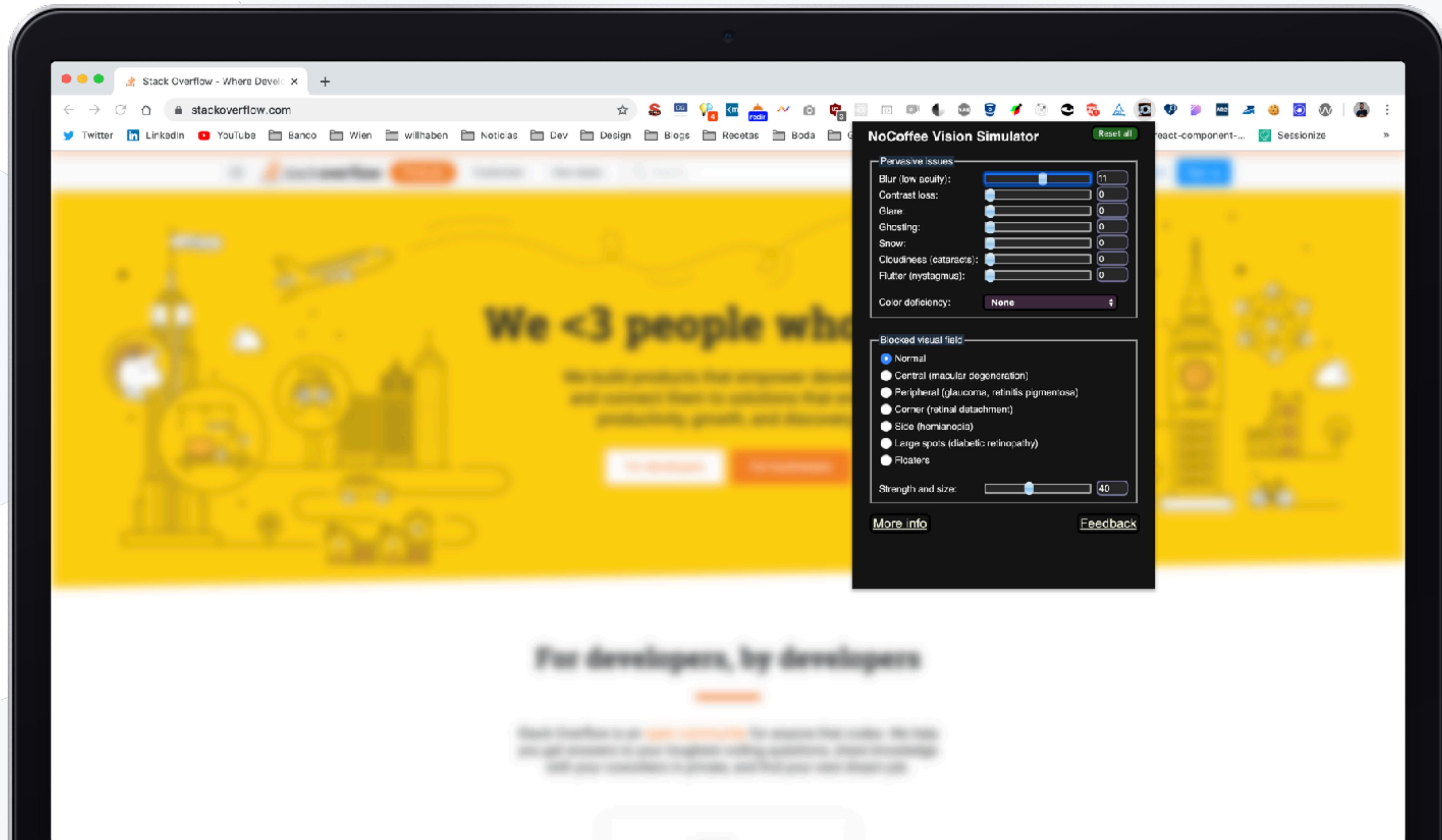
# WAVE (web accessibility evaluation tool)

<https://wave.webaim.org/>



# NoCoffee - Vision Simulator

<https://accessgarage.wordpress.com/>



# Testing the code

---

- react-axe
  - eslint-plugin-jsx-a11y
  - jest-axe
- 

# Testing the DOM

---

- axe-cli
  - pa11y
  - Lighthouse
- 

# Testing in the browser

---

- axe
- ARC Toolkit
- Accessibility Insights
- NoCoffee
- WAVE

# <https://github.com/bolonio/testing-web-a11y>

The screenshot shows a code editor interface with four files open:

- App.js**:  
A function component that contains several accessibility-related elements:
  - A button with the text "Click me NOW".
  - A button with the role="button" attribute and the text "CLICK ME".
  - A div with an onClick handler and a tabIndex of 0.
  - A FakeButton component with the text "Click me NOW".
  - An a element with an aria-label of "Continue" and a href attribute.
  - An a element with an aria-label of "Continue" and the role="button" attribute, containing the text "Click me NOW".
  - An a element with a role="myCustomRole" attribute and the text "Click me NOW".
  - An Image component with a src attribute pointing to a URL.
- button.jsx**:  
A functional component named Button that takes props.className, props.text, and props.classes.

```
1 import React from "react";
2
3 const Button = props => {
4   return <button className={props.classes}>{props.text}</button>;
5 };
6
7 export default Button;
```
- fakeButton.jsx**:  
A functional component named FakeButton that takes props.href, props.className, and props.classes.

```
1 import React from "react";
2
3 const FakeButton = props => {
4   return (
5     <a href={props.href} className={props.classes} role="button">
6       {props.text}
7     </a>
8   );
9 }
10
11 export default FakeButton;
```
- image.jsx**:  
A functional component named Image that takes props.src and props.alt.

```
1 import React from "react";
2
3 const Image = props => {
4   return <img width="200px" src={props.src} alt={props.alt} />;
5 };
6
7 export default Image;
```

Annotations in the code editor include:

- Red underlines and highlights are present on the following lines:
  - Line 20: <Button classes="button" text="Click me NOW" />
  - Line 26: <button role="button">CLICK ME</button>
  - Line 29: <div onClick={() => {}}>
  - Line 30: <div tabIndex={0}>
  - Line 33: <FakeButton classes="button" text="Click me NOW" />
  - Line 36: <a aria-label="Continue">Click me NOW</a>
  - Line 38: <a aria-label="Continue" role="button">
  - Line 42: <a role="myCustomRole">Click me NOW</a>
  - Line 49: <Image src="https://cms-assets.tutsplus.com/uploads/users/30/posts/29042/previe...>
  - Line 53: <img width="200px" src="https://cms-assets.tutsplus.com/uploads/users/30/posts/29042/previe.../>
- A blue vertical bar on the left margin indicates a selection or a specific line of code.

<https://speakerdeck.com/bolonio>

**How does your  
“sound” like?**

How web accessibility can make impaired people use your website

Adrián Bolonio

# Accessibility in Web Development

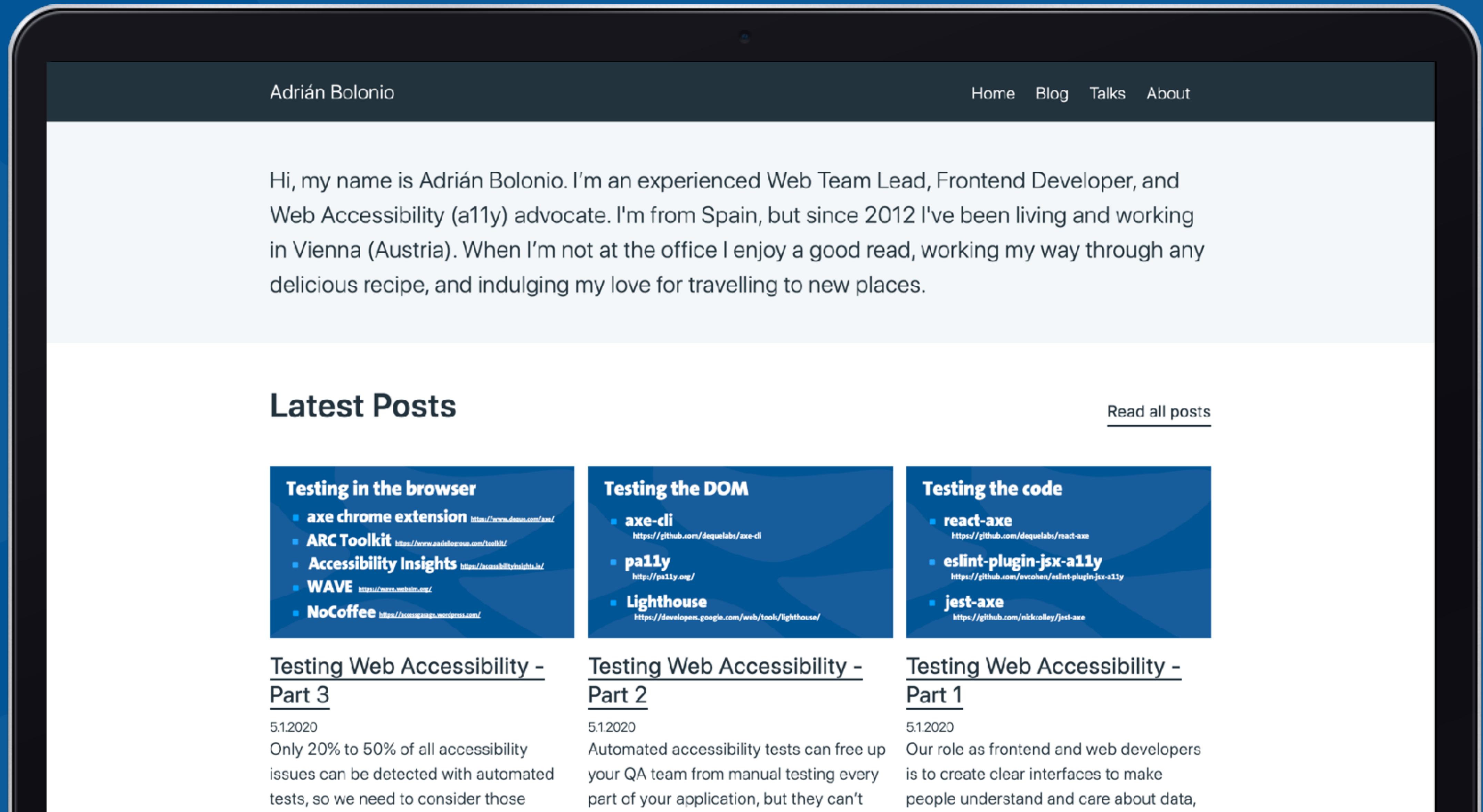
How web accessibility can make impaired people use your website

Adrián Bolonio

 @bolonio

 @bolonio

# <https://www.adrianbolonio.com/>



The screenshot shows a dark-themed website for Adrián Bolonio. At the top, there's a navigation bar with 'Adrián Bolonio' on the left and 'Home' (in white), 'Blog', 'Talks', and 'About' on the right. Below the navigation, a bio paragraph introduces Adrián as a Web Team Lead, Frontend Developer, and Web Accessibility advocate from Spain, currently living and working in Vienna, Austria. A 'Read all posts' link is located below the bio. The main content area features three cards under the heading 'Latest Posts': 'Testing in the browser' (with links to axe chrome extension, ARC Toolkit, Accessibility Insights, WAVE, and NoCoffee), 'Testing the DOM' (with links to axe-cli, pa11y, and Lighthouse), and 'Testing the code' (with links to react-axe, eslint-plugin-jsx-a11y, and jest-axe). Below these cards are three blog post summaries: 'Testing Web Accessibility - Part 3' (5.1.2020, about automated accessibility tests), 'Testing Web Accessibility - Part 2' (5.1.2020, about automated tests freeing up QA teams), and 'Testing Web Accessibility - Part 1' (5.1.2020, about creating clear interfaces).

Adrián Bolonio

Home Blog Talks About

Hi, my name is Adrián Bolonio. I'm an experienced Web Team Lead, Frontend Developer, and Web Accessibility (a11y) advocate. I'm from Spain, but since 2012 I've been living and working in Vienna (Austria). When I'm not at the office I enjoy a good read, working my way through any delicious recipe, and indulging my love for travelling to new places.

[Read all posts](#)

**Testing in the browser**

- [axe chrome extension](#) https://www.deque.com/axe/
- [ARC Toolkit](#) https://www.adrianbolonio.com/toolkit/
- [Accessibility Insights](#) https://accessibilityinsights.io/
- [WAVE](#) https://wave.webaim.org/
- [NoCoffee](#) https://accessgarage.wordpress.com/

**Testing the DOM**

- [axe-cli](#) https://github.com/dequelabs/axe-cli
- [pa11y](#) http://pa11y.org/
- [Lighthouse](#) https://developers.google.com/web/tools/lighthouse/

**Testing the code**

- [react-axe](#) https://github.com/dequelabs/react-axe
- [eslint-plugin-jsx-a11y](#) https://github.com/evcohen/eslint-plugin-jsx-a11y
- [jest-axe](#) https://github.com/nickcolley/jest-axe

**Testing Web Accessibility - Part 3**  
5.1.2020  
Only 20% to 50% of all accessibility issues can be detected with automated tests, so we need to consider those

**Testing Web Accessibility - Part 2**  
5.1.2020  
Automated accessibility tests can free up your QA team from manual testing every part of your application, but they can't

**Testing Web Accessibility - Part 1**  
5.1.2020  
Our role as frontend and web developers is to create clear interfaces to make people understand and care about data,

# Introduction to Web Accessibility

<https://www.edx.org/course/web-accessibility-introduction>



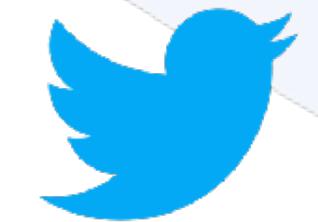
**“It’s not just about disabled users being able to access your website, it’s about everyone being able to access your website.”**

Trenton Moss, Owner of Webcredible Consultancy Firm, UK

# **Accessibility is NOT a feature**

# Thanks

**Adrián Bolonio**



**@bolonio**