



DWX

AUTOMATING ACCESSIBILITY TESTING WITH PLAYWRIGHT



rowdy.codes



rowdyrabouw



rowdyrabouw

ROWDY

**Front-End-Focused Senior DevOps Engineer
with a focus on usability and accessibility**



ROWDY

Google Developer Expert in Web Technologies



rowdy.codes



rowdyrabouw



rowdyrabouw

A11Y

Accessibility



Accessibility is a universal right because it is the gateway to other rights, such as voting, education, employment, and healthcare.

If you can't enter a building or access a website, you are effectively denied your citizenship.



1.300.000.000

1 in 6 people worldwide - World Health Organization (WHO)





D
ATASAN

YOL NO	YERİ	WAKIT
K9	Datışe	13:00
K9	Datışe	23:00
K3	Datışe	21:00
K8	Datışe	21:00

ATASAM
HASTANESİ



Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for anyone throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find hidden coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

Let's-a go!

As Mario says in Italy:

“Mamma mia!”

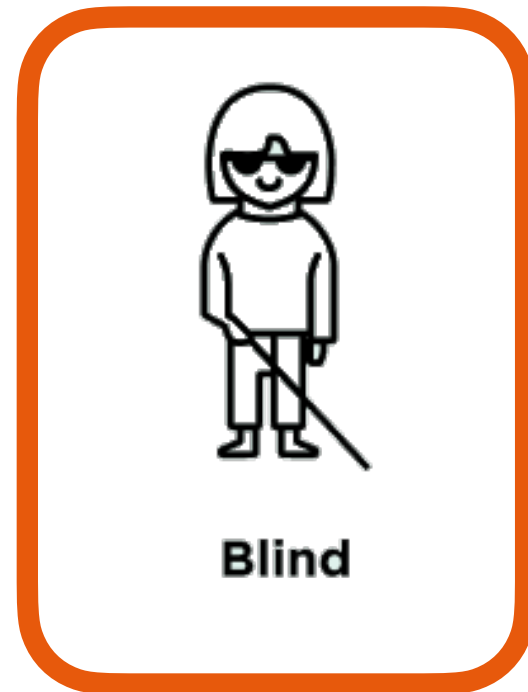


PERMANENT

TOUCH



SEE



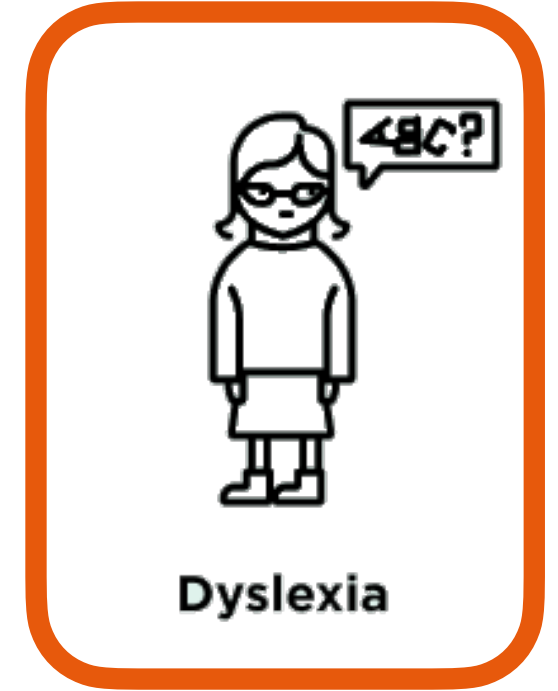
HEAR



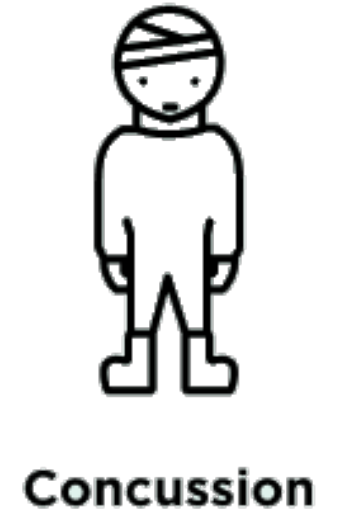
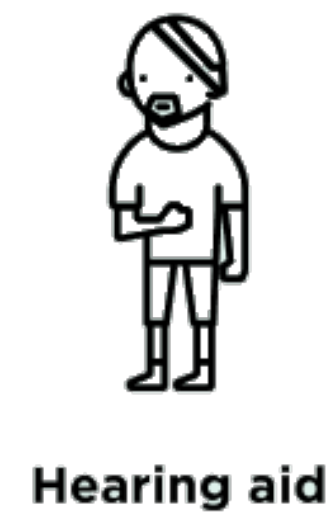
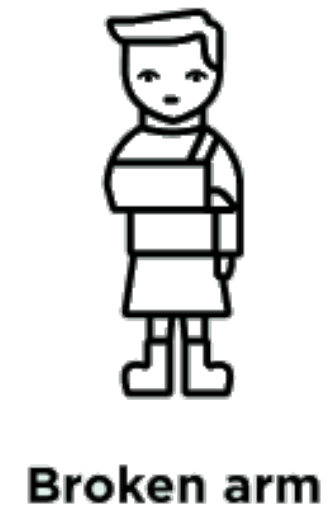
SPEAK



COGNITIVE



TEMPORARY



SITUATIONAL



CURB CUT EFFECT



EEA

European Accessibility Act



B2C

Business to Consumer



B2B

Business to Business





Sh*t to Business



WCAG

Web Content Accessibility Guidelines



POUR

Perceivable, Operable, Understandable, and Robust



87

Success Criteria





WCAG Conformance Levels





WCAG Conformance Levels





WCAG Conformance Levels





Standards/Guidelines

Web Content - WCAG 2

[How to Meet WCAG 2 \(Quick Reference\)](#)

[At a Glance](#)

[The Documents](#)

[Applying to Non-Web ICT](#)

[New in 2.2](#)

[New in 2.1](#)

[Translations](#)

[Commenting](#)

[Conformance Logos](#)

[FAQ](#)

[WCAG 3 Draft](#)

[Authoring Tools - ATAG](#)

WCAG 2 Overview

Summary

This page introduces the Web Content Accessibility Guidelines (WCAG) international standard, including WCAG 2.0, WCAG 2.1, and WCAG 2.2. WCAG documents explain how to make web content more accessible to people with disabilities.

A different page [introduces WCAG 3](#).

WCAG is not an introduction to accessibility. For introductions, see [Accessibility Fundamentals Overview](#).

Quick links to resources:

- [How to Meet WCAG 2 \(Quick Reference\)](#)
- [WCAG 2.2 Standard, What's New in WCAG 2.2](#)
- [WCAG 2.1 Standard](#)

Page Contents

- [Introduction](#)
- [Who WCAG is for](#)
- [What is in WCAG 2](#)
- [WCAG 2.0, 2.1, 2.2](#)
- [Translations](#)
- [ISO/IEC 40500. EAA. EN 301 549](#)



Search articles

Search

- Home
- Principles
 - 1 Perceivable
 - 2 Operable
 - 3 Understandable
 - 4 Robust
- Guidelines
 - 1.1 - Text Alternatives
 - 1.2 - Time-based Media
 - 1.3 - Adaptable
 - 1.4 - Distinguishable
 - 2.1 - Keyboard Accessible
 - 2.2 - Enough Time
 - 2.3 - Seizures and Physical Reactions
 - 2.4 - Navigable
 - 2.5 - Input Modalities
 - 3.1 - Readable
 - 3.2 - Predictable
 - 3.3 - Input Assistance

Perceivable Text Alternatives

WCAG 2.0, 2.1, 2.2 Level A

1.1.1 Non-text Content

All images and other non-text content (like icons, charts, audio, CAPTCHAs, or controls) must have a descriptive text alternative that conveys their meaning. Purely decorative content can be hidden from assistive technologies (e.g. using an empty alt attribute).

Code and Labels, Sensory, Wording Auditory/Hearing, Cognitive, Visual Code, Content

View Details

Perceivable Time-based Media

WCAG 2.0, 2.1, 2.2 Level A

1.2.1 Audio-only and Video-only (Prerecorded)

Prerecorded audio-only content must have a text transcript. Prerecorded video-only content must have a text or audio description.

Sensory, Wording Auditory/Hearing Content

View Details

Perceivable Time-based Media

WCAG 2.0, 2.1, 2.2 Level A





```
> pnpm create playwright
```

```
TypeScript or JavaScript (default: TypeScript)
```

```
Tests folder name (default: tests, or e2e if tests already exists)
```

```
Add a GitHub Actions workflow (recommended for CI)
```

```
Install Playwright browsers (default: yes)
```



> pnpm create playwright

<code>playwright.config.ts</code>	→ Test configuration
<code>tests/example.spec.ts</code>	→ Minimal example test
<code>package.json</code>	→ <code>@playwright/test</code>
<code>pnpm-lock.yaml</code>	→ <code>package-lock.json</code> / <code>yarn.lock</code>



> **npx playwright test**

Running 6 tests using 6 workers

6 passed (3.8s)



> **npx playwright show-report**



Playwright Test Report

Search tests

All 6 ✓ Passed 6 Failed 0 Flaky 0 Skipped 0 ⌚ ⚙️

6/18/2026, 8:28:07 PM Total time: 3.8s

example.spec.ts		
✓ has title	chromium	240ms
example.spec.ts:3		
✓ get started link	chromium	359ms
example.spec.ts:10		
✓ has title	firefox	926ms
example.spec.ts:3		
✓ get started link	firefox	1.1s
example.spec.ts:10		
✓ has title	webkit	724ms
example.spec.ts:3		
✓ get started link	webkit	910ms
example.spec.ts:10		

AXXE

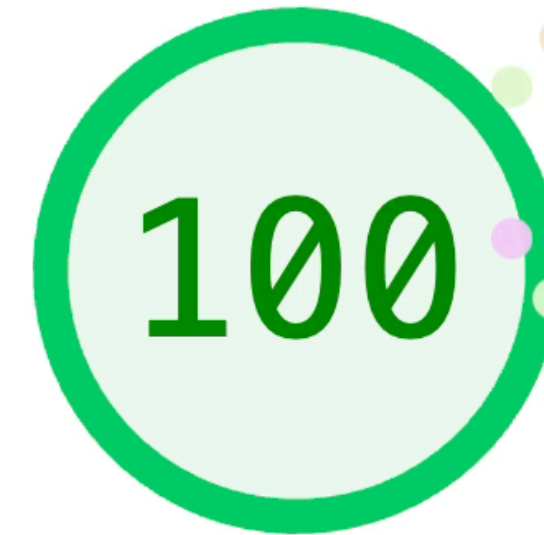
Axe-core by Deque





Performance

Accessibility



Best Practices



SEO



```
> pnpm add -D @axe-core/playwright
```



```
import { test, expect } from '@playwright/test';
```



```
import { test, expect } from "@playwright/test";  
import AxeBuilder from "@axe-core/playwright";
```



```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("page should be accessible", async ({ page }) => {

});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("page should be accessible", async ({ page }) => {

  await page.goto("http://127.0.0.1:8080/");

});
```

[World 1-1](#) | [Power-Ups](#) | [Extra Lives](#) | [Castles](#)

Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for anyone throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find hidden coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

As Mario says in Italy:

“ Mamma mia! ”



```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("page should be accessible", async ({ page }) => {

  await page.goto("http://127.0.0.1:8080/");

  const axeResults = await new AxeBuilder({ page }).analyze();

});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("page should be accessible", async ({ page }) => {

  await page.goto("http://127.0.0.1:8080/");

  const axeResults = await new AxeBuilder({ page }).analyze();

  expect(axeResults.violations).toEqual([]);

});
```

```
> npx playwright test tests/a11y.basic.spec.ts --project=chromium
```





🔍 Search tests

All 1 Passed 0 ✗ Failed 1 Flaky 0 Skipped 0 ⌚ ⚙️

Project: chromium

6/18/2026, 8:30:27 PM Total time: 1.0s

▼ a11y.basic.spec.ts	
✗ page should be accessible chromium	450ms
a11y.basic.spec.ts:4	



Test Steps

- > ✓ Before Hooks 320ms
- > ✓ Navigate to "/" — a11y.basic.spec.ts:5 44ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/... 73ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/di... 1ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/di... 5ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/... 50ms
- > ✓ Create page — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwri... 25ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/... 58ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/di... 0ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/di... 1ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/di... 1ms
- > ✓ Evaluate — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright/... 13ms
- > ✓ Close page — ../node_modules/.pnpm/@axe-core+playwright@4.11.2_playwright-core@1.59.1/node_modules/@axe-core/playwright... 4ms
- > ✗ Expect "toEqual" — a11y.basic.spec.ts:9 4ms
- > ✓ After Hooks 20ms



```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }) => {

  await page.goto("http://127.0.0.1:8080/");

});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page },      ) => {

  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {

    });

  });
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page },      ) => {

  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {

    const axeResults = await new AxeBuilder({ page }).analyze();

  });

});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }, testInfo) => {

  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {

    const axeResults = await new AxeBuilder({ page }).analyze();

    await testInfo.attach("axe-results", {
      body: JSON.stringify(axeResults.violations, null, 2),
      contentType: "application/json",
    });

  });

});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }, testInfo) => {

  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {

    const axeResults = await new AxeBuilder({ page }).analyze();

    await testInfo.attach("axe-results", {
      body: JSON.stringify(axeResults.violations, null, 2),
      contentType: "application/json",
    });

    expect(axeResults.violations.length).toBe(0);


  });

});
```

```
> npx playwright test tests/a11y.testinfo.spec.ts --project=chromium
```



Search tests

All 1 Passed 0 ✗ Failed 1 Flaky 0 Skipped 0  

Page should be accessible

a11y.testinfo.spec.ts:4

480ms

chromium

✗ Run

Errors

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

Copy prompt

Expected: 0
Received: 8



```
13 |     });  
14 |  
> 15 |     expect(axeResults.violations.length).toBe(0);  
    |                                     ^  
16 |   });  
17 | });  
18 |
```

at /Users/rowdy/Cloud/double-R webdevelopment/Presentations/playwright-a11y/tests/a11y.testinfo.spec.ts:15
at /Users/rowdy/Cloud/double-R webdevelopment/Presentations/playwright-a11y/tests/a11y.testinfo.spec.ts:7:

Test Steps

Filter steps

Q Search tests

All 1 Passed 0 ✗ Failed 1 Flaky 0 Skipped 0  

Page should be accessible

a11y.testinfo.spec.ts:4

480ms

chromium

✗ Run

> Errors

∨ Test Steps

Q Filter steps

- > ✓ Before Hooks 388ms
- > ✓ Navigate to "/" — a11y.testinfo.spec.ts:5 50ms
- > ✗ Run accessibility scan — a11y.testinfo.spec.ts:7 229ms
- > ✓ After Hooks 18ms
- > ✓ Worker Cleanup 18ms

∨ Attachments

- >  axe-results

Attachments

axe-results

```
[
  {
    "id": "heading-order",
    "impact": "moderate",
    "tags": [
      "cat.semantics",
      "best-practice"
    ],
    "description": "Ensure the order of headings is semantically correct",
    "help": "Heading levels should only increase by one",
    "helpUrl": "https://dequeuniversity.com/rules/axe/4.11/heading-order?application=playwright",
    "nodes": [
      {
        "any": [
          {
            "id": "heading-order",
            "data": null,
            "relatedNodes": [],
            "impact": "moderate",
            "message": "Heading order invalid"
          }
        ],
        "all": [],
        "none": [],
        "impact": "moderate",
        "html": "<h3>Get the Star Power!</h3>",
        "target": [
          "h3:nth-child(3)"
        ],
        "failureSummary": "Fix any of the following:\n  Heading order invalid"
      }
    ]
  },
  {
    "id": "html-has-lang",
    "impact": "serious",
    "tags": [

```

```
> pnpm add -D axe-html-reporter
```



```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    await testInfo.attach("axe-results", {
      body: JSON.stringify(axeResults.violations, null, 2),
      contentType: "application/json",
    });

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    await testInfo.attach("axe-results", {
      body: JSON.stringify(axeResults.violations, null, 2),
      contentType: "application/json",
    });

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";
import { createHtmlReport } from "axe-html-reporter";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";
import { createHtmlReport } from "axe-html-reporter";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    const reportHTML = createHtmlReport({
      results: axeResults,
      options: {
        projectKey: "Playwright A11Y Test",
      },
    });

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
import { test, expect } from "@playwright/test";
import AxeBuilder from "@axe-core/playwright";
import { createHtmlReport } from "axe-html-reporter";

test("Page should be accessible", async ({ page }, testInfo) => {
  await page.goto("http://127.0.0.1:8080/");

  await test.step("Run accessibility scan", async () => {
    const axeResults = await new AxeBuilder({ page }).analyze();

    const reportHTML = createHtmlReport({
      results: axeResults,
      options: {
        projectKey: "Playwright A11Y Test",
      },
    });

    await testInfo.attach("axe-results-html", {
      body: reportHTML,
      contentType: "text/html",
    });

    expect(axeResults.violations.length).toBe(0);
  });
});
```

```
> npx playwright test tests/a11y.html.spec.ts --project=chromium
```





🔍 Search tests

All 1 Passed 0 ✖ Failed 1 Flaky 0 Skipped 0 ⌚ ⚙️

Project: chromium

6/20/2026, 8:31:25 PM Total time: 980ms

▼ a11y.html.spec.ts
✖ Page should be accessible chromium 444ms
a11y.html.spec.ts:5



Axe-core® Accessibility Results for Playwright A11Y Test project

Page URL: <http://127.0.0.1:8080/>

axe-core found **8** violations

#	Description	Axe rule ID	WCAG	Impact	Count
1	Heading levels should only increase by one	heading-order	Best practice	moderate	1
2	<html> element must have a lang attribute	html-has-lang	WCAG 2 Level A, WCAG 3.1.1	serious	1
3	Images must have alternative text	image-alt	WCAG 2 Level A, WCAG 1.1.1	critical	1
4	Form elements must have labels	label	WCAG 2 Level A, WCAG 4.1.2	critical	1
5	Links must be distinguishable without relying on color	link-in-text-block	WCAG 2 Level A, WCAG 1.4.1	serious	1
6	 and must only directly contain , <script> or <template> elements	list	WCAG 2 Level A, WCAG 1.3.1	serious	1
7	Zooming and scaling must not be disabled	meta-viewport	WCAG 2 Level AA, WCAG 1.4.4	moderate	1
8	lang attribute must have a valid value	valid-lang	WCAG 2 Level AA, WCAG 3.1.2	serious	1

Failed

1. Heading levels should only increase by one

heading-order

Ensure the order of headings is semantically correct

Issue Tags: `cat.semantics` `best-practice`

[Learn more](#)

Best practice

moderate

#	Issue Description	To solve this violation, you need to...
1	Element location	Fix any of the following:



.with Tags

Only run selected rules



Default

WCAG 2.0 Level A + AA (2008)

WCAG 2.1 Level A + AA (2018)

WCAG 2.2 Level A + AA (2023)

Best practices

Deque's recommended rules that aren't required by WCAG but improve the overall user experience



wcag2a	(WCAG 2.0 Level A)
wcag2aa	(WCAG 2.0 Level AA)
wcag2aaa	(WCAG 2.0 Level AAA)
wcag21a	(WCAG 2.1 Level A)
wcag21aa	(WCAG 2.1 Level AA)
wcag21aaa	(WCAG 2.1 Level AAA)
wcag22a	(WCAG 2.2 Level A)
wcag22aa	(WCAG 2.2 Level AA)
wcag22aaa	(WCAG 2.2 Level AAA)

best-practice

experimental



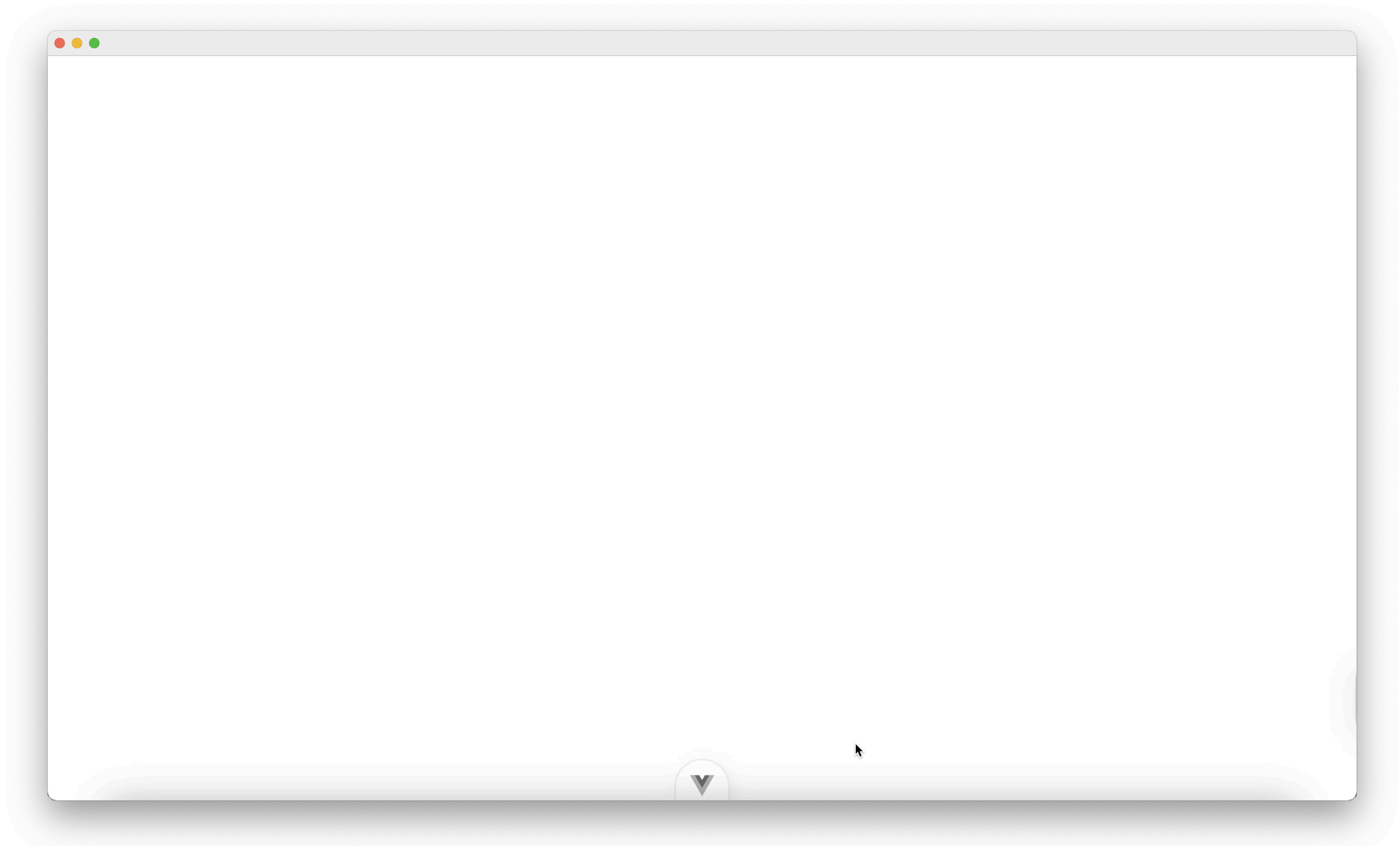
```
test("Page should be accessible", async ({ page }) => {  
  await page.goto("http://127.0.0.1:8080/");  
  await test.step("Run accessibility scan", async () => {  
    const axeResults = await new AxeBuilder({ page })  
      .withTags(["wcag2a", "wcag2aa", "wcag21a", "wcag21aa"])  
      .analyze();  
    expect(axeResults.violations.length).toBe(0);  
  });  
});
```



`.exclude`

Prevent all rules from running against the specified elements and all of their descendants





rowdy.codes



rowdyrabouw



rowdyrabouw



```
test("Page should be accessible", async ({ page }) => {  
  await page.goto("http://127.0.0.1:8080/");  
  await test.step("Run accessibility scan", async () => {  
    const axeResults = await new AxeBuilder({ page })  
      .exclude(".vue-devtools__panel")  
      .analyze();  
    expect(axeResults.violations.length).toBe(0);  
  });  
});
```



```
test("Page should be accessible", async ({ page }) => {  
  await page.goto("http://127.0.0.1:8080/");  
  await test.step("Run accessibility scan", async () => {  
    const axeResults = await new AxeBuilder({ page })  
      .exclude("#element-with-known-a11y-issue")  
      .analyze();  
    expect(axeResults.violations.length).toBe(0);  
  });  
});
```



Axe-core® Accessibility Results for Playwright A11Y Test project

Page URL: <http://127.0.0.1:8080/>

axe-core found **8** violations

#	Description	Axe rule ID	WCAG	Impact	Count
1	Heading levels should only increase by one	heading-order	Best practice	moderate	1
2	<html> element must have a lang attribute	html-has-lang	WCAG 2 Level A, WCAG 3.1.1	serious	1
3	Images must have alternative text	image-alt	WCAG 2 Level A, WCAG 1.1.1	critical	1
4	Form elements must have labels	label	WCAG 2 Level A, WCAG 4.1.2	critical	1
5	Links must be distinguishable without relying on color	link-in-text-block	WCAG 2 Level A, WCAG 1.4.1	serious	1
6	 and must only directly contain , <script> or <template> elements	list	WCAG 2 Level A, WCAG 1.3.1	serious	1
7	Zooming and scaling must not be disabled	meta-viewport	WCAG 2 Level AA, WCAG 1.4.4	moderate	1
8	lang attribute must have a valid value	valid-lang	WCAG 2 Level AA, WCAG 3.1.2	serious	1

Failed

1. Heading levels should only increase by one

heading-order

Ensure the order of headings is semantically correct

Issue Tags: `cat.semantics` `best-practice`

[Learn more](#)

Best practice
moderate

#	Issue Description	To solve this violation, you need to...
1	Element location	Fix any of the following:



Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for anyone throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find hidden coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

As Mario says in Italy:

“ Mamma mia! ”



1. Heading levels should only increase by one



```
<h1>Welcome to Mushroom Kingdom Adventures</h1>  
<p>Whether you are a heroic plumber ...</p>  
<h3>Get the Star Power!</h3>
```



```
<h1>Welcome to Mushroom Kingdom Adventures</h1>  
<p>Whether you are a heroic plumber ...</p>  
<h2>Get the Star Power!</h2>
```

2. <html> element must have a lang attribute



```
<!doctype html>  
<html>  
  <head>
```



```
<!doctype html>  
<html lang="en">  
  <head>
```



3. Images must have alternative text



```

```



```

```



4. Form elements must have labels



Enter your Player ID (It's a-me!)
`<input type="text" />`



`<label for="player">Enter your Player ID (It's a-me!):</label>`
`<input type="text" id="player" />`



5. Links must be distinguishable without relying on color



```
a {  
  text-decoration: none;  
}
```



```
nav a {  
  text-decoration: none;  
}
```

6. and must only directly contain elements



```
ul div::before {  
  content: "▪ ";  
}
```

```
<ul>  
  <div>Essential for throwing fireballs at Piranha Plants.</div>  
  <div>Great for smashing through brick blocks to find coins.</div>  
</ul>
```



```
<ul>  
  <li>Essential for throwing fireballs at Piranha Plants.</li>  
  <li>Great for smashing through brick blocks to find coins.</li>  
</ul>
```



7. lang attribute must have a valid value



```
<p class="quote">As Mario says in Italy:</p>  
<span lang="italian">  
  <blockquote>Mamma mia! </blockquote>  
</span>
```



```
<p class="quote">As Mario says in Italy:</p>  
<span lang="it">  
  <blockquote>Mamma mia! </blockquote>  
</span>
```



8. Zooming and scaling must not be disabled



```
<meta name="viewport" content="width=device-width, initial-scale=1.0,  
maximum-scale=1.0, user-scalable=no"  
>
```



```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```



Axe-core® Accessibility Results for Playwright A11Y Test project

Page URL: <http://127.0.0.1:8080/index.fixed.html>

axe-core found **0** violations

axe returned 29 passed axe checks. [Expand details on click](#)

axe returned 0 incomplete checks. [Expand details on click](#)

axe returned 60 inapplicable checks. [Expand details on click](#)

axe was running with 0 rules. [Expand details on click](#)



Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for anyone throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find hidden coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

As Mario says in Italy:

“ Mamma mia! ”



Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

As Mario says in Italy:

« Mamma mia! »



Welcome to Mushroom Kingdom Adventures

Whether you are a heroic plumber or a helpful Toad, the Mushroom Kingdom provides endless adventures and exciting challenges in the 8-bit world. Grab a Super Mushroom and get ready for the specific challenges of stomping Goombas, dodging fireballs, and rescuing Princess Peach from Bowser's castle.

Get the Star Power!

Your ultimate defense against falling into lava, getting hit by spiked shells, or running out of time before reaching the flag pole.

- Essential for throwing fireballs at Piranha Plants.
- Great for smashing through brick blocks to find coins.

Secret Items

When exploring the levels, always remember to look for invisible blocks containing "1-Up Mushrooms" which give you an extra chance if your jump down a pipe mistakenly lands you in a bottomless pit.

To learn more about finding special power-ups, click [here](#).

Level Select

Enter your Player ID (It's a-me!):

As Mario says in Italy:

« Mamma mia! »



CI/CD

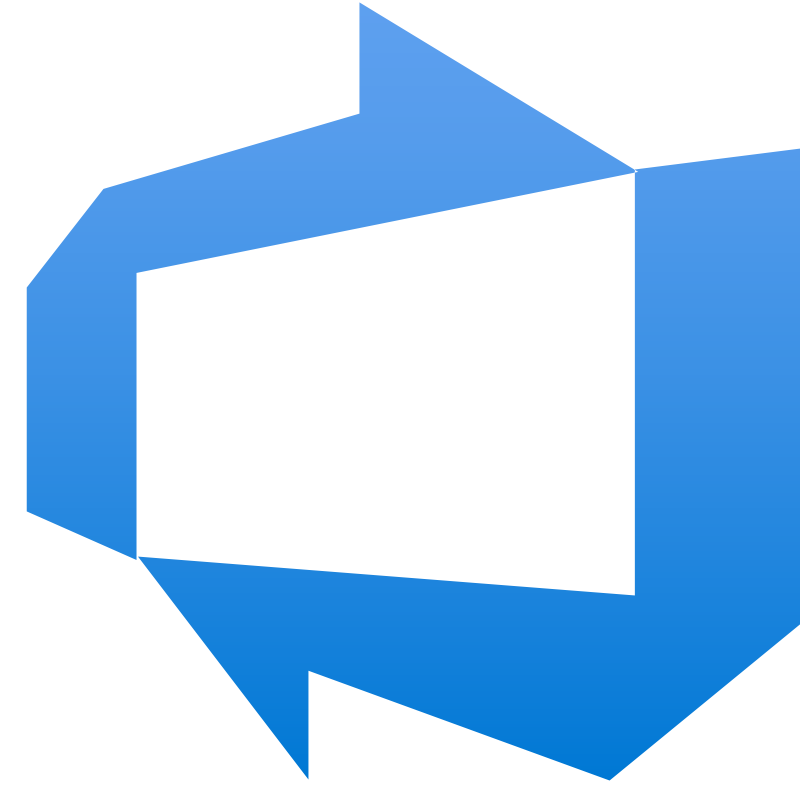
Automated Accessibility Audit



CI/CD

~ 30% Automated Accessibility Audit





Feature branches

- Run on Chromium only*
- Failures should not block deployments to TST
- Failures should block merging to main

** Running accessibility tests on Chromium is sufficient for initial checks, as accessibility violations are primarily DOM-level issues rather than engine-specific rendering differences.*



MORE

We can do more ...



Playwright Test Report

Search tests

All 6 ✓ Passed 6 Failed 0 Flaky 0 Skipped 0 ⌚ ⚙️

6/18/2026, 8:28:07 PM Total time: 3.8s

example.spec.ts		
✓ has title	chromium	240ms
example.spec.ts:3		
✓ get started link	chromium	359ms
example.spec.ts:10		
✓ has title	firefox	926ms
example.spec.ts:3		
✓ get started link	firefox	1.1s
example.spec.ts:10		
✓ has title	webkit	724ms
example.spec.ts:3		
✓ get started link	webkit	910ms
example.spec.ts:10		

```
import { test, expect } from "@playwright/test";

test("has title", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});

test("get started link", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Click the get started link.
  await page.getByRole("link", { name: "Get started" }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole("heading", { name: "Installation" })).toBeVisible();
});
```

Fast and reliable end-to-end te...r modern web apps | Playwright

Playwright Docs MCP CLI API Node.js

Search

Playwright enables reliable web automation for testing, scripting, and AI agents.

One API to drive Chromium, Firefox, and WebKit — in your tests, your scripts, and your agent workflows. Available for [TypeScript](#), [Python](#), [.NET](#), and [Java](#).

Get started

Star 86k+

Playwright Test

Full-featured test runner with auto-waiting, assertions, tracing, and parallelism across Chromium, Firefox, and WebKit.

```
npm init playwright@latest
```

[Testing documentation](#)

Playwright CLI

Token-efficient browser automation for coding agents like Claude Code and GitHub Copilot. Skill-based workflows without large context overhead.

```
npm i -g @playwright/cli@latest
```

[CLI documentation](#)

Playwright MCP

Model Context Protocol server that gives AI agents full browser control through structured accessibility snapshots.

```
npx @playwright/mcp@latest
```

[MCP documentation](#)

```
import { test, expect } from "@playwright/test";

test("has title", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});

test("get started link", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Click the get started link.
  await page.getByRole("link", { name: "Get started" }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole("heading", { name: "Installation" })).toBeVisible();
});
```

getByRole

How assistive technology perceives the page



DevTools - playwright.dev/

Elements Console Sources Network Performance Memory Application Security Lighthouse >>

```
<nav aria-label="Main" class="theme-layout-navbar navbar navbar--fixed-to
p"> flex
  <div class="navbar__inner"> flex
    <div class="theme-layout-navbar-left navbar__items"> ... </div> flex
    <div class="theme-layout-navbar-right navbar__items navbar__items--righ
t"> ... </div> flex
  </div>
  <div role="presentation" class="navbar-sidebar__backdrop"></div>
</nav>
<div id="__docusaurus_skipToContent_fallback" class="theme-layout-main main
-wrapper mainWrapper_z2l0"> flex
  <header class="hero hero--primary heroBanner_UJJx"> flex
    ::before
    <div class="container">
      <h1 class="hero__title heroTitle_ohkl"> ... </h1>
      <p class="heroSubtitle_GKHc"> ... </p>
      <div class="buttons_pzb0"> flex
        <a class="getStarted_Sjon" href="/docs/intro">Get started</a> == $0
        <span class="github-btn github-stargazers github-btn-large"> ...
        </span>
      </div>
    </div>
  </header>
  <main> ... </main>
</div>
<footer class="theme-layout-footer footer footer--dark"> ... </footer>
</div>
</body>
```

Styles Computed Layout Accessibility >>

Show accessibility tree

ARIA Attributes

No ARIA attributes

Computed Properties

Name: "Get started"

aria-labelledby: Not specified
aria-label: Not specified
Contents: "Get started"
title: Not specified

Role: link

Focusable: true
url: "https://playwright.dev/docs/intro"

Source Order Viewer

No source order information available

hero--primary.heroBanner_UJJx div.container div.buttons_pzb0 a.getStarted_Sjon



Installation | Playwright

Playwright Docs MCP CLI API Node.js

Getting Started

- Installation
- Writing tests
- Generating tests
- Running and debugging tests
- Trace viewer
- Setting up CI
- VS Code
- Release notes
- Canary releases
- Playwright Test
 - Agents
 - Annotations
 - Command line
 - Configuration
 - Configuration (use)
 - Emulation
 - Fixtures
 - Global setup and teardown
 - Parallelism
 - Parameterize tests
 - Projects

Getting Started > Installation

Installation

Introduction

Playwright Test is an end-to-end test framework for modern web apps. It bundles test runner, assertions, isolation, parallelization and rich tooling. Playwright supports Chromium, WebKit and Firefox on Windows, Linux and macOS, locally or in CI, headless or headed, with native mobile emulation for Chrome (Android) and Mobile Safari.

You will learn

- How to install Playwright
- What's installed
- How to run the example test
- How to open the HTML test report

Installing Playwright

Get started by installing Playwright using one of the following methods.

Using npm, yarn or pnpm

The command below either initializes a new project or adds Playwright to an existing one.

`npm` `yarn` `pnpm`

Introduction

- Installing Playwright
 - Using npm, yarn or pnpm
 - Using the VS Code Extension
- What's Installed
- Running the Example Test
- HTML Test Reports
- Running the Example Test in UI Mode
- Updating Playwright
- System requirements
- What's next



DevTools - playwright.dev/

Elements Console Sources Network Performance Memory Application Security Lighthouse >>

```

<div class="container padding-top--md padding-bottom--lg">
  <div class="row"> flex
    <div class="col docItemCol_V0Vn">
      <div class="docItemContainer_Djhp">
        <article>
          <nav class="theme-doc-breadcrumbs breadcrumbsContainer_Z_b l" aria-label="Breadcrumbs"> ... </nav>
          <div class="tocCollapsible_ETCw theme-doc-toc-mobile tocMobile_ITEo"> ... </div>
          <div class="theme-doc-markdown markdown">
            ::before
            <header>
              <h1>Installation</h1> == $0
            </header>
            <h2 class="anchor anchorTargetStickyNavbar_Vzrq" id="introduction"> ... </h2>
            <p> ... </p>
            <p> ... </p>
            <ul> ... </ul>
            <h2 class="anchor anchorTargetStickyNavbar_Vzrq" id="installing-playwright"> ... </h2>
            <p>Get started by installing Playwright using one of the following methods.</p>
            <h3 class="anchor anchorTargetStickyNavbar_Vzrq" id="using-npm-yarn-or-pnpm"> ... </h3>
            <p> ... </p>
            <div class="theme-tabs-container tabs-container tabList_CuJ"> ... </div>
          </div>
        </article>
      </div>
    </div>
  </div>
</div>

```

Styles Computed Layout Accessibility >>

Show accessibility tree

ARIA Attributes

No ARIA attributes

Computed Properties

Name: "Installation"

aria-labelledby: Not specified
aria-label: Not specified
Contents: "Installation"
title: Not specified

Role: heading
Level: 1

Source Order Viewer

No source order information available

div.docItemContainer_Djhp article div.theme-doc-markdown.markdown header h1



**alert alertdialog application article banner
blockquote button caption cell checkbox
code columnheader combobox complementary
contentinfo definition deletion dialog directory
document emphasis feed figure form
generic grid gridcell group heading img
insertion link list listbox listitem log main
marquee math meter menu menubar
menuitem menuitemcheckbox menuitemradio
navigation none note option paragraph
presentation progressbar radio radiogroup
region row rowgroup rowheader scrollbar
search searchbox separator slider spinbutton
status strong subscript superscript switch
tab table tablist tabpanel term textbox time
timer toolbar tooltip tree treegrid treeitem**



```
test("get started link", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Click the get started link.
  await page.getByRole("link", { name: "Get started" }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole("heading", { name: "Installation" })).toBeVisible();
});
```



```
test("get started link", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Click the get started link.
  await page.getByRole("link", { name: "Get started", exact: true }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole("heading", { name: "Installation" })).toBeVisible();
});
```



```
test("get started link", async ({ page }) => {
  await page.goto("https://playwright.dev/");

  // Click the get started link.
  await page.getByRole("link", { name: "Get started", exact: true }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole("heading", { name: "Installation",
    level: 1
  })).toBeVisible();
});
```



getByLabel

```
<label for="username">Username</label>  
<input type="text" id="username" />
```

```
await page.getByLabel("Username").fill("Mario");
```



getByText

```

```

```
const logo = page.getByAltText("Nintendo logo");  
await expect(logo).toBeVisible();
```



getByText

```
<p class="quote">As Mario says in Italy:</p>  
<span lang="it">  
  <blockquote>Mamma mia!</blockquote>  
</span>
```

```
await expect(page.getByText("Mamma mia!")).toBeVisible();
```



getByTitle

```
<span title="Life count" class="sr-only">3</span>
```

```
await expect(page.getByTitle("Life count")).toHaveText("3");
```



getByTestId

```
<select data-testid="character-select">  
  ...  
  <option value="toad">Toad</option>  
  ...  
</select>
```

```
const dropdown = page.getByTestId("character-select");  
await dropdown.selectOption("toad");
```



getBy

older



```
<label for="email">  
<input type="email"
```

```
.com" />
```

```
await page.getByPlace
```

```
("mario@mushroom.com");
```



Locator

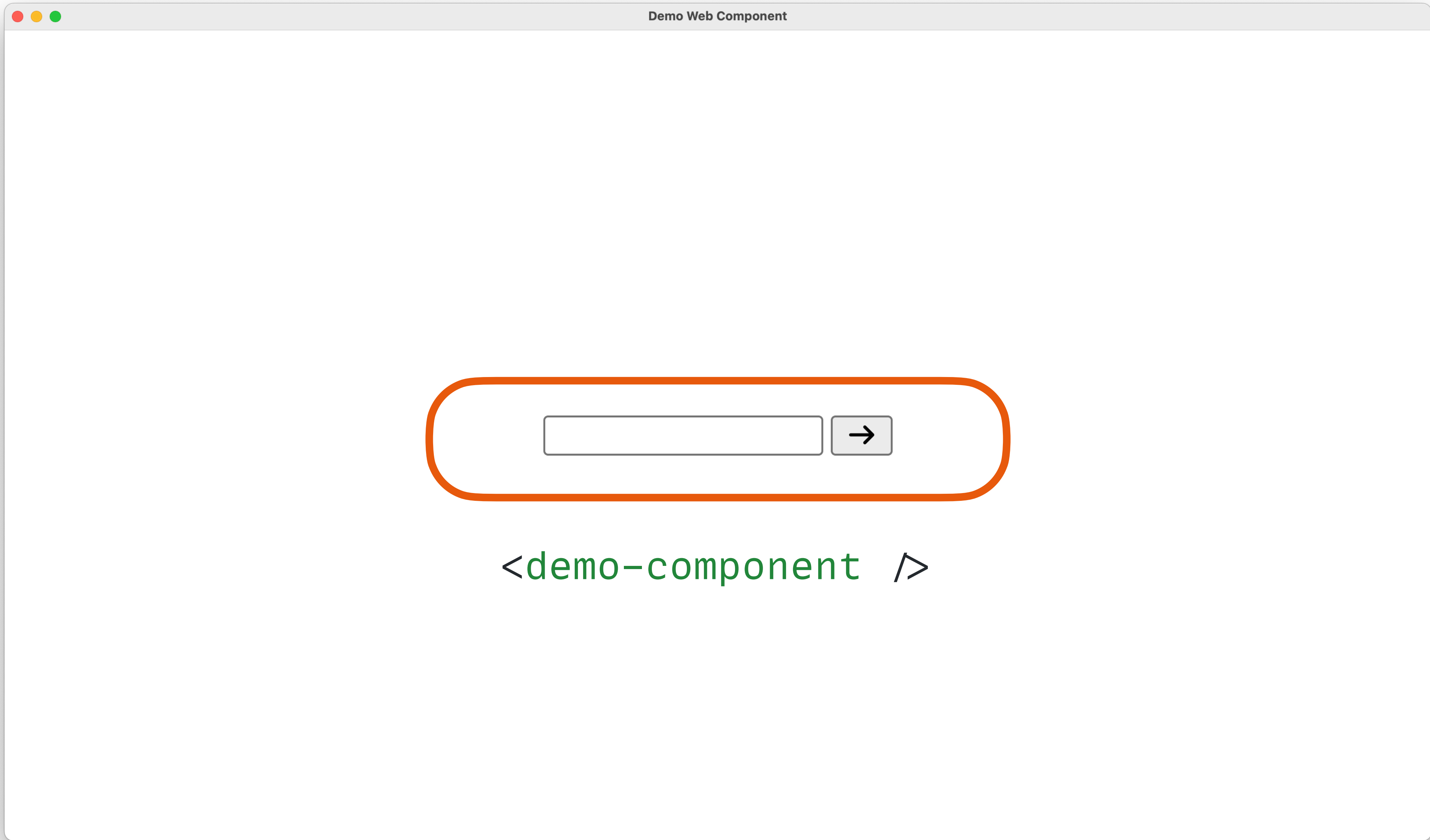


```
<button class  
<button class  
<button class  
  
const marioElem  
await marioElem  
  
await page.locator("id=jump", 3)".click();
```

VITEST

Component Testing





```
export class DemoComponent extends HTMLElement {
  constructor() {
    super();
    this.attachShadow({ mode: "open" });
  }

  connectedCallback() {
    this.shadowRoot.innerHTML = `
      <link rel="stylesheet" href="https://kit.fontawesome.com/5c9c09d32c.css" />
      <form>
        <input type="text" />
        <button type="submit"><i class="fa-solid fa-arrow-right"></i></button>
      </form>
    `;
  }
}

customElements.define("demo-component", DemoComponent);
```

```
import { describe, it, expect, beforeEach, afterEach } from "vitest";
import { configureAxe } from "vitest-axe";
import "../demo-component.js";

describe("DemoComponent", () => {
  let element;

  // create the element and append it to the DOM before the test
  beforeEach(() => {
    element = document.createElement("demo-component");
    document.body.appendChild(element);
  });

  // clean up the DOM after the test
  afterEach(() => {
    document.body.removeChild(element);
  });
});
```

```
it("should have no accessibility violations (raw)", async () => {  
  
  const form = element.shadowRoot.querySelector("form");  
  
  const axe = configureAxe({  
    rules: {  
      // disable landmark rules when testing isolated components  
      region: { enabled: false },  
    },  
  });  
  
  const results = await axe(form);  
  
  expect(results).toHaveNoViolations();  
});  
});
```

Expected the HTML found at \$('input') to have no violations:

```
<input type="text">
```

Received:

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

Fix any of the following:

- Element does not have an implicit (wrapped) <label>

- Element does not have an explicit <label>

- aria-label attribute does not exist or is empty

- aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty

- Element has no title attribute

- Element has no placeholder attribute

- Element's default semantics were not overridden with role="none" or role="presentation"

You can find more information on this issue here:

<https://dequeuniversity.com/rules/axe/4.12/label?application=axeAPI>



Expected the HTML found at \$('button') to have no violations:

```
<button type="submit"><i class="fa-solid fa-arrow-right"></i></button>
```

Received:

```
"Buttons must have discernible text (button-name)"
```

Fix any of the following:

- Element does not have inner text that is visible to screen readers
- aria-label attribute does not exist or is empty
- aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
- Element has no title attribute
- Element does not have an implicit (wrapped) <label>
- Element does not have an explicit <label>
- Element's default semantics were not overridden with role="none" or role="presentation"

You can find more information on this issue here:

<https://dequeuniversity.com/rules/axe/4.12/button-name?application=axeAPI>



**YOU CAN'T STIR CHOCOLATE
CHIPS INTO BAKED COOKIES.**



CORE

HTML

CSS

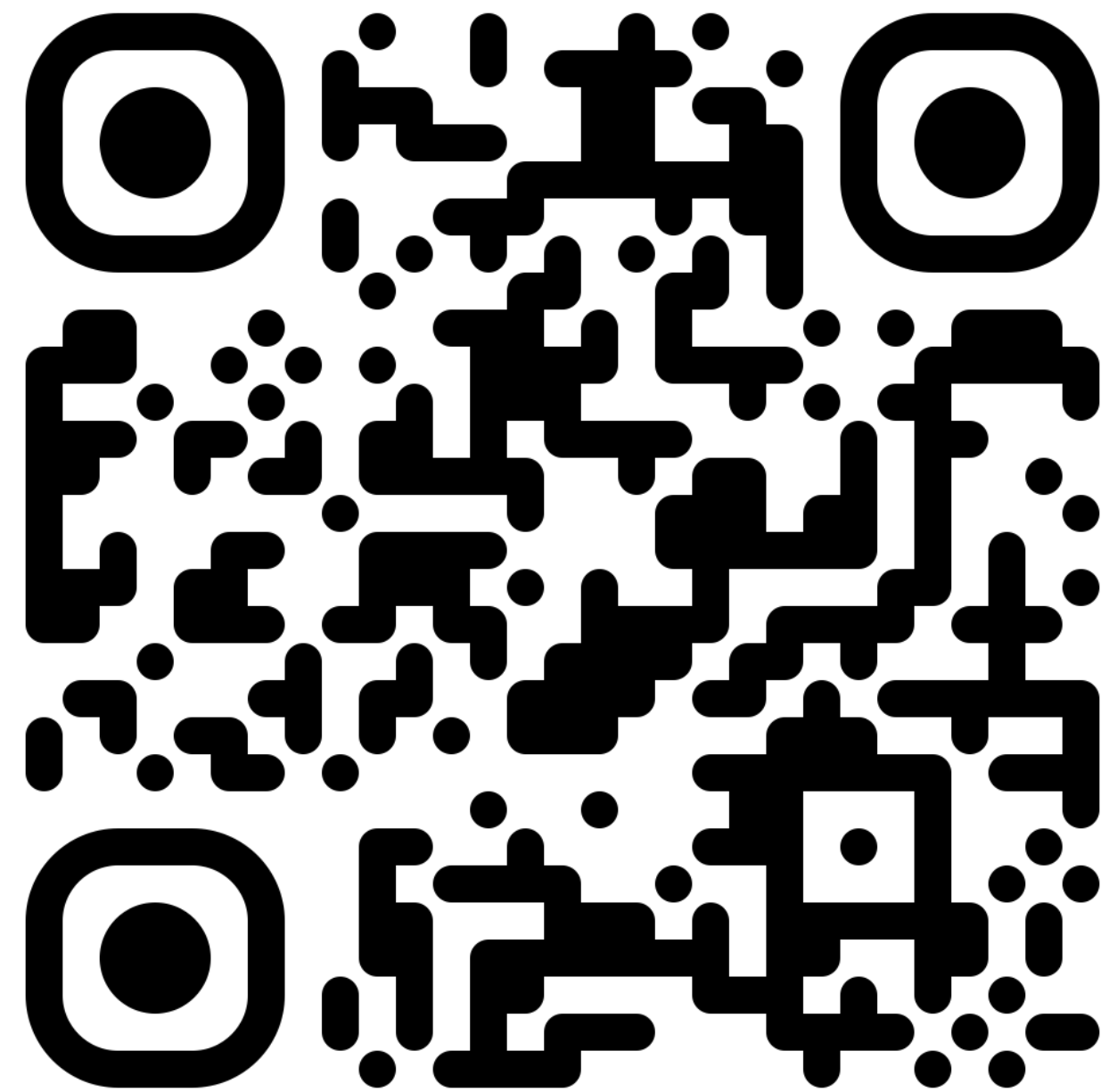
Accessibility

JavaScript



THANKS

rowdy.codes/dwx-a11y



rowdy.codes



rowdyrabouw



rowdyrabouw