

Web accessibility smoke testing for critical issues

Longhorn PHP Conference 2026

October 16 at 13:30



Rachele DiTullio, Senior Accessibility Engineer, Citizens

About me

- Web developer for the first half of my career
- MSIS from the UT Austin School of Information (UXD)
- Learned accessibility and became a Certified Professional in Web Accessibility (CPWA)
- Transitioned from software engineering to accessibility engineering in 2021



What we'll cover today

Web accessibility overview

Smoke test 1: Use a keyboard

Smoke test 2: Check text and UI scaling

Resources

Web accessibility overview

What is web accessibility?

Web accessibility is the extent to which **disabled people** can use a website or application.

Accessibility is a benchmark of product **quality**.



Accessibility is not only a compliance or legal issue

It's a **human** one.

- Disabled people will try to use our websites and web applications.
- Who is being excluded by our design and development decisions?
- How well do we understand the information and assistive technology needs of our disabled customers?

Accessibility testing by developers is impactful

Developers can catch **90%** of **critical blocker** accessibility issues with just a few quick checks.

Accessibility critical blockers

Critical accessibility issues cause web content or functionality to be **completely inaccessible** to some users.

- A video without captions or a transcript available
- A button that can only be clicked with a mouse
- A link that doesn't work with the **enter** key
- A page where nothing has a visible indication of keyboard focus
- Content that is not available on smaller screens or when zoomed
- Etc., this is not an exhaustive list



Cost of accessibility defects

On average, each accessibility defect costs about **\$2600** in **potential ROI** to fix.

(Source: [Karl Groves](#))

Web accessibility standards

- [Web Content Accessibility Guidelines \(WCAG\) version 2.2](#) success criteria are used to evaluate whether components and webpages are accessible.
- ISO standard: ISO/IEC 40500:2025
- WCAG 2.2 has **55** success criteria at Levels A and AA.
- Web smoke testing covers **6 success criteria**.

Smoke test 1: Use a keyboard

If you can do only one thing for accessibility

Make your application
work with a **keyboard**.

If it works with a keyboard, it will likely work with assistive technologies, like screen readers, for free.

Four keyboard checks to smoke test

1. Ensure all actions can be performed with a keyboard
2. Ensure there are no keyboard traps
3. Ensure the keyboard focus order is logical
4. Ensure keyboard focus is visible

Check #1: Ensure all actions can be performed with a keyboard

Goal: Everything can be done with a keyboard except freehand movements.

What to do: Ensure pointer actions have a keyboard equivalent.

Why it's important: Many people rely on the keyboard interface, including blind and some mobility impaired people.

Success criterion: [2.1.1 Keyboard](#)

Basic keyboard navigation commands

Description	Command
Go to the next interactive element	<code>tab</code>
Go to the previous interactive element	<code>shift + tab</code>
Activate a button	<code>enter or space</code>
Activate a link	<code>enter</code>
Select a radio button, tab, list item, etc., scrolling	<code>arrow keys</code> (When in doubt, try these.)
Select a checkbox	<code>space</code>
Close dialogs, expanded content	<code>esc</code>

Mac users: Enable Full Keyboard Access

By default, you cannot navigate webpages on MacOS using standard keyboard navigation commands.

1. Choose *Apple menu > System Settings*; then click **Accessibility** in the sidebar.
2. Choose *Motor > Keyboard* in the Accessibility menu.
3. Toggle **on** the switch for "Full Keyboard Access".
4. Safari: *Settings > Advanced*. Under **Accessibility**, tick the checkbox for "Press Tab to highlight each item on a webpage".



Check #2: Ensure there are no keyboard traps

Goal: Keyboard users don't get stuck.

What to do: Ensure users always know how to navigate away from components.

Why it's important: People who rely on the keyboard often have no other means to navigate.

Success criterion: [2.1.2 No Keyboard Trap](#)



Check #3: Ensure the keyboard focus order is logical

Goal: Keyboard users navigate content in a correct order.

What to do: Elements receive focus in an order that preserves meaning.

Why it's important: Navigating a site or application with only the keyboard will make sense.

Success criterion: [2.4.3 Focus Order](#)



Welcome back to Online Banking!



By logging in you acknowledge that you have read, understand and agree to the Citizens Online Terms & Conditions and Electronic Notice Disclosure & Consent, and that you have provided a valid primary email address. We are pleased to provide you with this enhanced and secure banking experience.

[Electronic Notice Disclosure and Cons](#)



[Terms and Conditions](#)

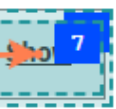


Icon should not be in the focus order.

User ID



Password



Need help?

[Find quick answers in our f](#)



[Contact](#)



☐ Remember user ID

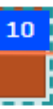
[Trouble logging in/Password char](#)



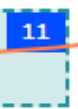
New to Online Banking? [Enroll ne](#)



 Log In



Cancel



Checkbox should be in the focus order.

Non-deposit products: Are not FDIC Insured; Are not deposits; May lose value.

Check #4: Ensure keyboard focus is visible

Goal: Users know which element has keyboard focus.

What to do: Ensure each item receiving focus has a visible indicator.

Why it's important: Without a focus indicator, sighted keyboard users cannot operate the page.

Success criterion: [2.4.7 Focus Visible](#)



FDIC FDIC-Insured - Backed by the full faith and credit of the U.S. Government

Welcome back!

To log in, enter your user ID and password below.



Welcome back to Online Banking!



By logging in you acknowledge that you have read, understand and agree to the Citizens Online Terms & Conditions and Electronic Notice Disclosure & Consent, and that you have provided a valid primary email address. We are pleased to provide you with this enhanced and secure banking experience.

[Electronic Notice Disclosure and Consent](#)

[Terms and Conditions](#)

User ID

Password

[Show](#)

☐

Remember user ID

[Trouble logging in/Password change](#)

Need help?

[Find quick answers in our FAQs](#)

[Contact Us](#)



HTML elements that receive keyboard focus

Commonly used elements

1. Link
2. Button
3. Text input (email, password, tel, url)
4. Select and option
5. Radio button
6. Checkbox
7. Text area
8. File input
9. Reset button
10. Submit button
11. Summary
12. iframe

Custom components

Add keyboard focus with the `tabindex="0"` attribute.

Less used elements

1. Search input
2. Range input (slider)
3. Date input
4. Time input
5. Number input
6. Etc.



[CodePen example](#)

Smoke test 2: Check text and UI scaling

All webpages must resize text and reflow content

Adaptive content helps people viewing webpages on smaller mobile device screens and people who have scaled the browser or OS settings to view text and content at larger sizes.

MacOS browser zoom

- Press `command +` to increase zoom level.
- Press `command -` to decrease zoom level.
- Press `command 0` to reset zoom level to 100%.

Windows browser zoom

- Press `ctrl +` to increase zoom level.
- Press `ctrl -` to decrease zoom level.
- Press `ctrl 0` to reset zoom level to 100%.

Check #5: Ensure text resizes 200% in a 1024x768 viewport

Goal: Text can be enlarged.

What to do: Ensure text can be doubled in size.

Why it's important: Some people can only read text when it is bigger.

Success criterion: [1.4.4 Resize Text](#)



Check #6: Ensure content reflows in a 320x256 viewport

Goal: Content can be enlarged without increasing line length.

What to do: Make lines of text reflow within the viewport.

Why it's important: People who need bigger text find it difficult if they must scroll to read long lines.

Success criterion: [1.4.10 Reflow](#)

Check the webpage for resized text and reflowed content

1. Is there a meaningful change in the text displayed onscreen?

All text should scale up or down linearly.

2. Does any content overlap or run off the screen?

No content should overlap, and text should either wrap or (in some limited cases) be truncated. In cases where text wraps, containers should be scrollable if there is a height constraint; text and webpages should not scroll horizontally. Text should not run off the screen altogether.

3. Does the screen still make sense?

There should still be some visual hierarchy in the relationships between headings, paragraphs of text, graphics, controls and other elements on the screen.

100% zoom in a 1024x768 viewport


FDIC FDIC-Insured - Backed by the full faith and credit of the U.S. Government

Welcome back!

To log in, enter your user ID and password below.

User ID

Password

[Show](#)

☐ Remember user ID

[Trouble logging in/Password change](#)

New to Online Banking? [Enroll now.](#)

 **Log In**

[Cancel](#)

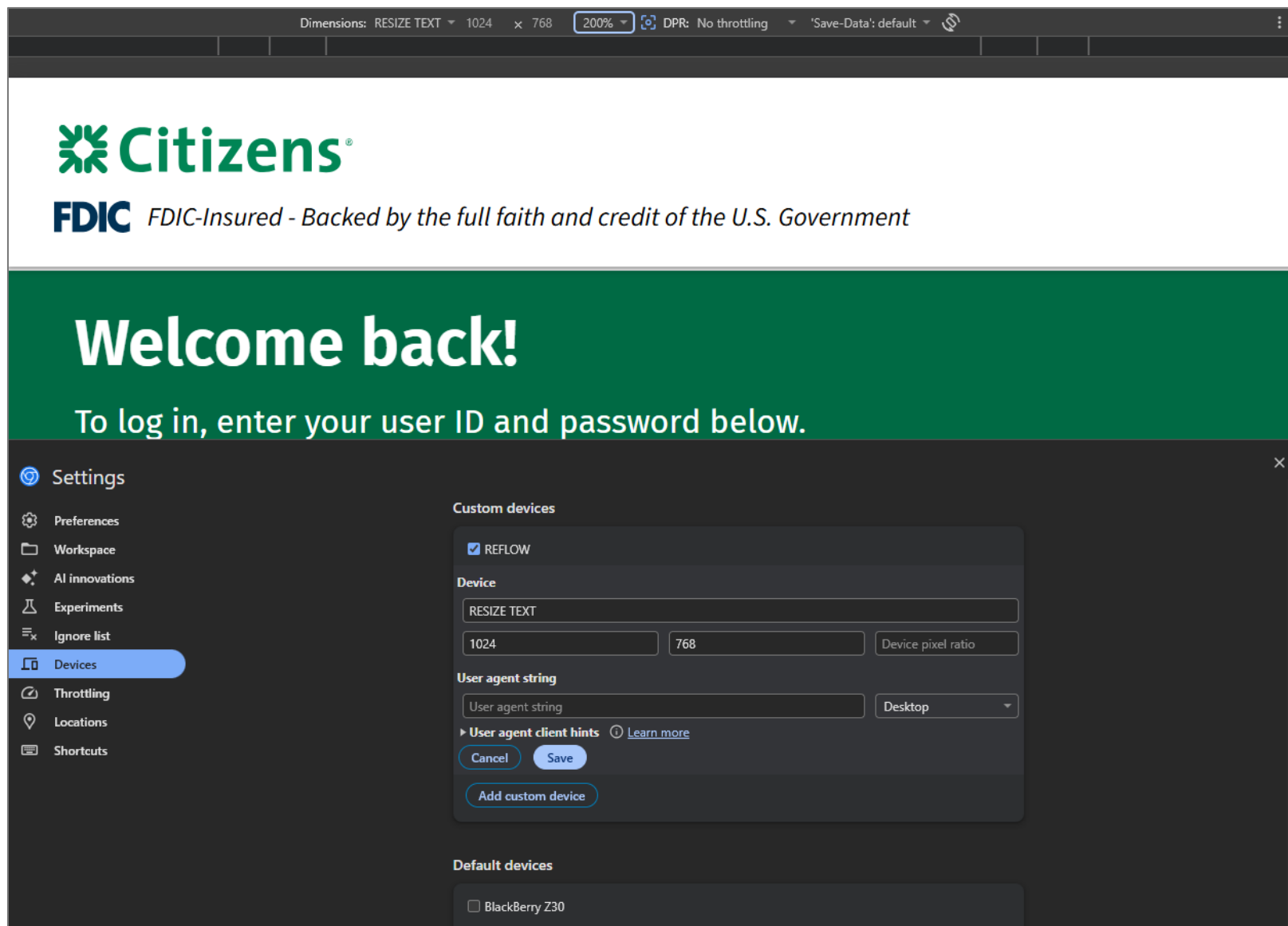
Need help?

[Find quick answers in our FAQs](#)

[Contact Us](#)



Chrome emulator: 1024x768 viewport at 200% zoom





Resize text: 200% zoom in a 1024x768 viewport

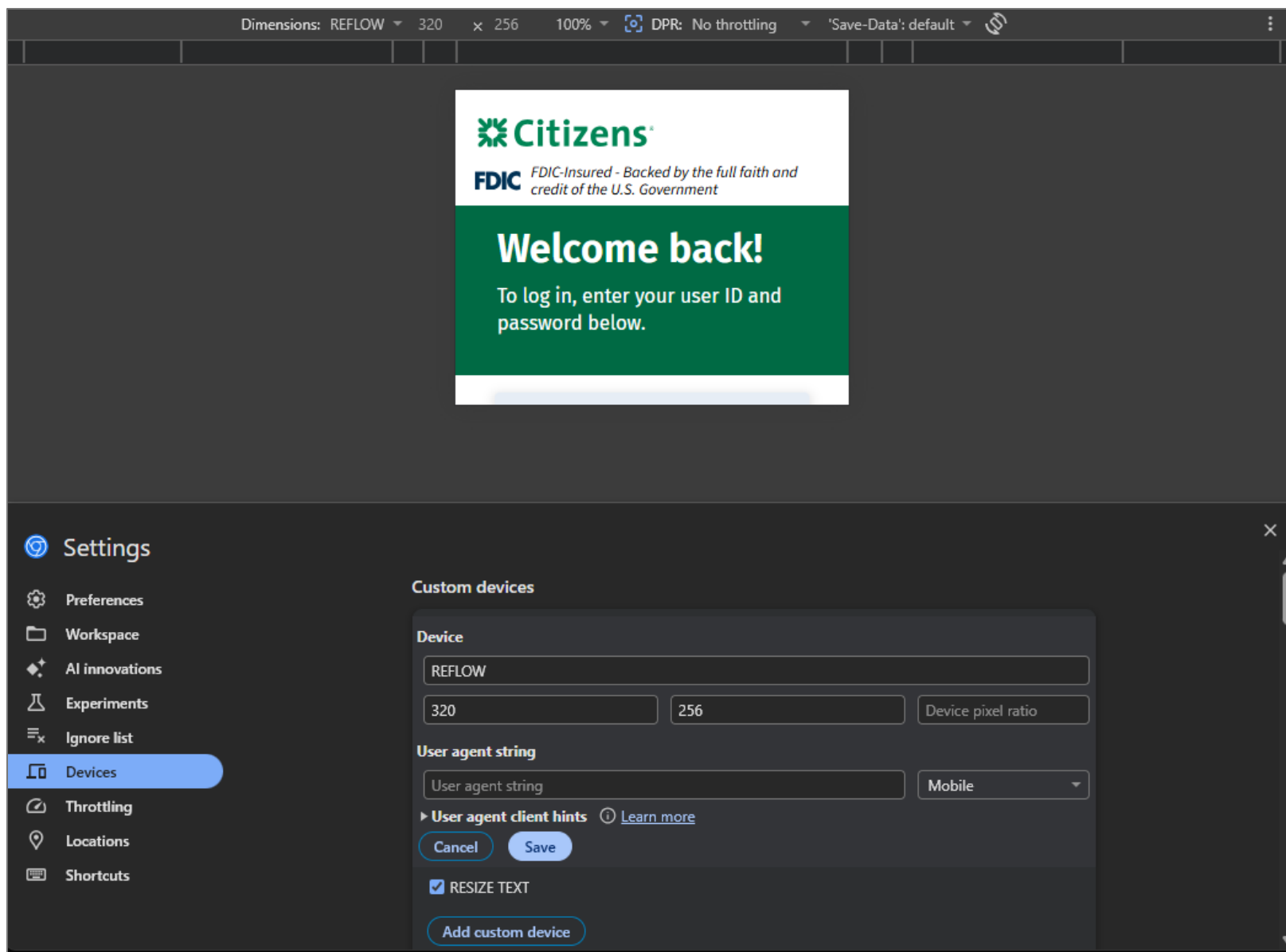

FDIC *FDIC-Insured - Backed by the full faith and credit of the U.S. Government*

Welcome back!

To log in, enter your user ID and password below.

User ID

Chrome emulator: 320x256 viewport at 100% zoom





Reflow: 320x256 viewport = 400% zoom in a 1280x1024 viewport



FDIC *FDIC-Insured - Backed by the full faith and credit of the U.S. Government*

Welcome back!

To log in, enter your user ID and password below.



Smoke testing summary: 6 quick checks

1. Can all actions can be performed with a keyboard?
2. Are there are any keyboard traps?
3. Is the keyboard focus order logical?
4. Is keyboard focus visible for all interactive elements?
5. Can text resize 200% in a 1024x768 viewport?
6. Can content reflow in a 320x256 viewport?

Thank you

- Download slides
- Blog article of this talk
- Contact me
- <https://RacheleDiTullio.com/>

