

Practical, Browser-based A11Y testing

Jeff Coburn

@Coburnicus

jeff.coburn@umb.edu

Web Services Manager

Institute for Community Inclusion
School for Global Inclusion and Social Development
University of Massachusetts Boston

<https://github.com/coburnicus/browser-testing>





**Why should we develop web surveys
and pages with accessibility in mind?**

Because we are not monsters.



NOTE: This is the bare minimum of acceptable testing

User testing with people with disabilities is combined with unit testing is preferable.



Testing page/survey

|-----Launching without testing, FAIL

|

|

| <-----**We are here**

|

|-----Unit Testing at code level

|

|

|-----Testing with actual users



Testing Electrical Socket

|-----Touching wires with screwdriver, dying.

|

|

| <-----**We are here**

|

|-----Using Multi-meter|

|

|

|-----Calling an electrician



What is accessible?


It depends on who you ask?

If you ask the Federal Government, the Section 508 is the standard.



Section 508 Refresh

Currently close to 10 years late, but should be coming out soon...



Live View of Section 508 Refresh Process





Use WCAG 2.0

<https://www.w3.org/WAI/WCAG20/quickref/>
<http://webaim.org/standards/wcag/checklist>

It says says you web pages and apps should be:

- ❖ **Perceivable** - You can detect what is there, and what it's purpose is, using your senses.
- ❖ **Operable** – Its usable. Things behave in an expected way.
- ❖ **Understandable** - The content/instructions are comprehensible and useful. Are things labeled correctly.
- ❖ **Robust** - The page or app works on appropriate devices and allows for assistive technology.



Tools you can use

No command line, no installing apps, no coding.



Accessibility auditors:

- ◇ - [Wave](#)
- ◇ - [Axe](#)
- ◇ - [Chrome Accessibility tools](#)
- ◇ - [Tenon](#)

This looks at your page and outputs all the potentials errors and issues.
Think of it like the last bit of your TurboTax experience.



Other tools

- ❖ [Web Developer Toolbar](#) (by Chris Pederick)
- ❖ Specialized testing Tools ([color blindness simulators](#))
- ❖ [Screen Reader](#) (simulators)
- ❖ -Browser emulators (<https://www.browserstack.com/>)



Let's test

These tools are available in Chrome and or Firefox

Start by running a general audit.



Perceivable

Image Alternatives and Color Issues

test: Disable images (do the images have alt tags)

Use: Web developer toolbar (chrome and firefox)

Test: The RGB Values of background vs foreground

Use: Auditor, or color blindness simulator



Operable

Keyboard-only Accessible

Test: Is form keyboard usable and logical.

Use: Keyboard-only to navigate form. Is it usable is it logical?



Understandable

Test: Does the form implement labels.*

Use: Web Developer Toolbar
(FORMS --> outline form fields without labels)

*Labels are super important



Robust Screen-readers and Devices

Test: Does it work with a screen-reader

Use: ChromeVox to simulate a screen-reader

Test: Does it work on IE 8 or Android 4.8.56.38.58?

Use: Emulators like browserstack.com

Thank You

<https://github.com/coburnicus/browser-testing>

Jeff Coburn

Jeff.coburn@umb.edu

@coburnicus (twitter, github)

