# Tips and tricks, or ‘what’s great in my web accessibility crate’.

Perth Web Accessibility day, 20/2/2021, Chris Leighton

Feel free to take notes or sit back and relax, I’ll make the script from today available on the groups website.

It’s a Word document, the accessibility checker says ‘no issues’!

Hello friends,

Please bear with me as I take us on a short journey of tips and tricks, or ‘what’s great in my web accessibility crate’.

A few words to explain my perspective

I’ve been rubbing up against web accessibility for a decade. I started working in an environment supporting and training people using What You See Is What You Get (WYSIWYG) interfaces and building new websites in a Content Management System that was pretty good, fairly well developed in achieving accessibility and locked up so that going off-piste wasn’t easily possible.

Since then I’ve become and advocate, adviser and for alliterative purposes an auditor for web accessibility.

I hope to have something for everyone including advice for any coders and developers we have with us today. At the conferences I’ve attended I often feel that we steer away from coding. Maybe because where talking about bits and bytes may been thought to be too alienating for those of us in policy, audit, content or advocacy. However, without great bits and bytes we really won’t get too close to web accessibility because those ones an 0s do matter.

Here are a few things I’ve learned and some pointed, personal advice. None of this new but maybe it is new for you?

## Coding

HTML5

For the developers, coders and anyone who logs in to build interfaces and create content.

Use all of the HTML5 elements every time and before you do anything else. Creating from scratch? Sketch it out adding the elements.

Do you have all of the elements under your belt, I recommend that these all become second nature to you. Look to always use them appropriately. When looking at other peoples work look for where they have used HTML5, or haven’t used them. It may tell you a lot. Looking for HTML5 may not be as much fun as the comments but, you know…

Why use HTML5 you say? Yes I know you may be able to do much with CSS and javascript and yes Flash is now gone. Because the browsers and assistive technology know, understand and respond in a consistent way to most if not all elements.

Why most? The combinations of your stack from the server to CMS to the end users browser and assistive technology may make a few things not work in that terribly complex combination. There are many moving parts there. HTML5 has most occasions covered and many pieces of the puzzle recognise the elements and respond to them as intended. But not all and not always.

Try ‘Can I Use’ or ‘HTML5 Test’ for compatibility lists.

There is also tension among big-tech in leading in design, maybe leading in accessibility too. One makes the intended-to-be-standard item work ‘this-way’, the other big-tech company takes the same item and makes it work ‘this-way’. Errgh. Things however do mostly work reliably, consistently. Mostly, especially if your code is reliable and consistent. Maybe using HTML5?

The greatest chance you will give both your customers, colleagues and end users to get close to an accessible experience is in using HTML5 a lot. You ‘can’ however achieve accessibility in less standard ways.

Buttons

I have seen things…

There are many ways to make buttons and hyperlinks.

There have been international discussions over what buttons do compared to what hyperlinks do.

I have seen things coded as hyperlinks next to things coded as buttons, the hyperlink submitted data and the button sent the browser to a new address. And I have seen these things where the visual presentations were the same for both different things and where the hyperlinks had a contrasting coloured surround (that most buttons in the world have?) and I have seen buttons presented as underlined like most links in the world are.

I have tried to steer developers away from these varieties and into consistency, consistency is a WCAG criteria

Why did I try and change these things?
Why - one

That international discussion and the stack we talked about earlier, and being consistent.

I fall on the side of a button submits data, as far as the user is concerned. A hyperlink sends you somewhere, again, as far as the user is concerned.

They should be coded as the good HTML lords intended and look like the world expects they may. In the Western World there really is a consistent design palette for these already. If you move away from existing, easily understood design do give weight to people who want ‘your thing’ to work seamlessly compared to being impressed by new and different web design. That weight may be enough to keep your designs close to what we know already works.

Try reading Don Norman or Jakob Nielsen on design.

#### Why - two

One of the screen-readers that I use has a really quick way to explain a page, it takes code and content and splits it into types of content found on the page. By example: this type of element in this column, each item coming with their programmatic labels. Headings, Landmarks, Navigation and others. One of the types is ‘links’, another is ‘buttons’.

If I’m using that method to follow a page and I listen to the links in the page I may miss the link that is coded as a button and sits in the other unheard column. And that is the crux of all my blather on buttons.

If it is a hyperlink don’t make it coded and looking like a button. The same goes for every other HTML5 element with an expected use.

There is an unending arrangement of examples where well meaning but incorrectly applied coding can create these ‘does not work’ situations. In many cases being transparent and obvious in the use of HTML5 can eliminate many of these.

So, in a nutshell for the developers, you know there are complex stacks. I say being consistent with the obvious will help.

And that is HTML5 finished. Now, back to HTML5.

There is a site called;

[html](https://html.com/html5)

, it’s good, but I like this older, not under current development site;

[html5 doctor](http://html5doctor.com).

I like HTML5 doctor because it’s advice is reliable, clean and clear and has Steve Faulkners fingerprints all over it. Steve is an Australian living in the UK in recent years. I like Steves work. If you develop I recommend that you follow his work, maybe in Twitter. He has changed employers and his pet-project focus over the years but I believe he works for the U.S. of A based The Paciello Group at the moment.

In Steves tweets you will find insightful and reliable recommendations. Like the one where title= tags are great, except in the many cases where they simply don’t work. I was shattered to read that title= advice, because it disavowed lots that I leant on. However it was right! Like, how title doesn’t work in touch because the important accessibility pop-up tip tends to be under a finger and cannot be read!

Chris Coyier is another recommended accessible code leader, he has been big at CSS-Tricks and Code-Pen. Adrian Roselli is another. Check them out. They may also lead you to others and more great advice. Bookmark the things that resonate, you may go back to them at anytime, there may be an update waiting for you.

Be aware that not all advice is best practice, you may have to be discerning.

In listening to these guys and leveraging HTML5 I was able to get a contractor to include <article in a blog template design. That element may help shared news pieces spread across the world and be more accessible to all. And that’s great.

HTML5 let me liaise with Microsoft and get them to include figure and figcaption in a recent Sharepoint template. Can you imagine taking on a giant like them without having the back-up of this set of reliable and widely leveraged elements!?

### ARIA

And, like lots of people I can’t leave coding without a swipe at ARIA.

Here are the issues; it over-rides other things that may offer the same or better utility; it’s complex, you may be great at it but the next person who forks or edits yours may not be and in essence it offers little to those not using text-to-audio and similar.

ARIA offers much in amending designs that were not sufficiently generous in the first place. It gives much better semantics to screen-readers.

It’s my opinion that going back to the original source that created a need for ARIA and being more generous in programmatic and visible labelling will more likely help more people.

But if you ‘do ARIA’ chase up Bryan Garaventas work at Whatsock, he’s amazing!

## Auditing

So, somethings been put in front of you to audit. My tip, use an automatic checker.

‘But they have a huge hole in them where they may only show 40% of all possible errors’ you say!

‘Yes’, I respond, ‘but you have to start somewhere’!

And it is true that the checkers may throw false-positives, where they report, ‘this is broken’, but, if you dive in and check the checker, the thing it has flagged is indeed OK. And that, I say, is a good thing. Having the opportunity to verify that an element is OK leaves you with a check-item ticked saying ‘this is good’.

The other thing about checkers is that the 40% of things it does recognise gets those items highlighted quickly. And time is money and time isn’t elastic in this universe. The automatic checkers also, in my experience, get you quickly into the swing of understanding a document or experience. Is a layout table seen early in the reportage? If yes you may expect to see more and know to look for them.

The style of reporting you do may also require all errors of a similar type to be listed or recorded for fixes in the same space or list item. 50 contrast failings or similar. An auto checker may be able to help you with that.

Is your auto-checker in sync with the latest standards? Sad face if not but if you have a checker as a service like I do and it is a WCAG version behind that’s OK because all the bits and bobs from the previous version are in the current version. You may have to catch-up on the missing bits elsewhere.

‘Which checker is best’ you ask. Well, I haven’t really come across one that’s ‘worse’ than any other and one way or the other they are all different from each other.

Do you need an exported report in a stand-alone document? Some checkers do, some don’t. Is there a user interface that will suit your needs better? OK, what interface do you want, there’s one to suit every day of the week, and price points to match. Do you want one that records all items found **except** ones already listed and with false positives from known-code omitted; do you want those items filed in project management software and assigned to people to fix them with all the tips for how to fix the item included? Yes! There’s a few of them.

Me, I have many, all the free one’s, because why not and as many of the paid ones as I can get. I try all the trial-offers if they look to offer something achievable and different, again, because why not.

Want to rely on something close to the WCAG source? Try:

[Nu validator](https://validator.w3.org/nu)

it is in-sync with most of the latest standards and has fall throughs to old-standard checking too. It has a disclaimer as most checkers do. Reminding us that they assist and that those false-positives may be a thing.

### So, that’s 40% done. What about the next 60%.

I’ll reverse the pyramid now and say that user testing for accessibility by a large sample of all representative disabled audience types is the go. Now take a breath, that won’t be easy to do for everything, but you can and it should be remembered that it is the gold standard.

Flipping the pyramid back onto its base, it is **more likely** it will be automated tests, tests by an auditor and then user testing by volume and order of action. Each finding different areas to improve on.

Why we must remember the gold standard is because ‘reality should beat out emulation’ and having great experiences for all people is the aim. Real people come across real problems, and can give best insight into the affect of the problems. An accessibility organisation led by disabled people is known to say ‘nothing about us, without us’, that resonates with me.

Me, I use NVDA screen-reader most often to listen to the web, it provides text and programmatic advice into audio. It lacks the heuristics of interpretation that JAWS adds, where JAWS sometimes can give the answer to a complex series very succinctly. NVDA tends to work through things one by one.

In my auditing I want to know the one by one as NVDA is being used not just to test for NVDA conformance in my stack but also as a reliable and industry accepted agent to point me to things of interest. Maybe something that looks like one thing but in audio is something very different, at the most basic poor alt text. At the most complex, efforts to circumvent accessibility testing. If I find something that doesn’t work in NVDA I may test it in Chrome Vox, the browser extension screen-reader. Does it work in Vox, yes, hmm, test it in JAWS. A mix n match may expose something significant or it may be a browser or assistive tech. specific bug. Bugs, they’re worth finding too.

As much as discovering the failing of something in a complex stack is a job well done some perspective may be needed. Bugs get listed with the particular providers and we move onto things where we can provide for a positive effect.

Auditing is a team sport. I hope you win.

## Staying on top of things

How to find out what you need to know

Me, I do all the free, seemingly good quality accessibility courses I can, the more the merrier. It’s often the case that the same ground may be covered, however no course was a waste of time, every course has added something. Here’s a short list that are worth following up:

* University of South Australia has a Certificate course
* W3C has an audit (no-charge) or signed-in (paid) course
* Check the G. Raymond Chang school at Ryerson University in Ontario for their free or paid courses. It’s good to take a look at accessibility from a national, provincial perspective that isn’t the same as your own.
* Lots of the businesses set-up for accessibility do webinars, some paid, some free. Be prepared for the business spiels. There’s an association for this too, they do training and accreditation.
* There is the mysterious US Homeland Security Trusted Tester program. If I can hack-in I’ll do their course too.

But the one I really like is the W3C accessibility discussion board. You get all sorts on it but to be honest not a lot of noise. You get to hear from some passionate experts who really know their chops. And if you follow their links or advice or aren’t sure you’re in step with what they say you can check-it. I did recently and got a finer understanding of what I thought I knew. And these things sometimes make a difference.

One of my favourites sometimes on the board is Leonie Watson, now operating TetraLogical. Do you know the Document Object Model, the DOM? Leonie was the first to mention, that I know of, the Accessibility Object Model, AOM. That’s something for the future maybe. Look it up. And Leonie takes the heat out of impassioned discussions with really sensible commentary.

Oh, and read all the W3C pages that you can starting at Web Accessibility Initiative (WAI). When you get to the last of all pages call me, I will be suitably impressed, there is a lot there.

So there, some tips and tricks. HTML5 for the developers, auditing and study.
I have much more. Are there any questions?