A Future Date | Put it on my TAB - Glen Walker.mp4

GLEN WALKER: Hey. I'm Glen Walker. I'm the Director of Accessibility at Applause.com. And I have my contact information on the first slide here. I have both my professional and my personal email address. My professional email address is Gwalker@applause.com my personal address is Glen.Walker@gmail.com

Glen is spelled with one N. I also have my camera turned on, but that's just temporarily so you can meet me. I'm going to be turning it off in a few minutes. And that'll help with the screen that's shared over on the right that will get a little bit bigger when I turn the camera off.

And so, I've got my resolution on that display set a little bit medium ish because I'm going to swap back and forth between my slides and showing a live demo in the browser. And I wanted to be able to show the browser in the DevTools, the Code Inspector, side by side. And so, I wanted to make that fit and compromise between a little resolution and hide in a large font versus showing both windows next to each other.

So I'll turn my camera off in a minute. Also, I will be here live when the presentation is ongoing. So I'll be available for a Q&A. If we don't get to your question during that Q&A process, again, my email addresses are listed on the slide. And I'll read those again at the end of the presentation. So let's get into it.

My talk today-- I wanted to talk about this really cool key on the keyboard, the Tab key. I suspect most of you have a Tab key on your computer keyboard. Let me turn off my camera real quick. So if you go back a little ways-- and I'd forgotten about this, not that I was around when this type of typewriter was around.

The Tab key used to be on the right side of the keyboard. I had totally forgot about that. That's like way back. Kind of a little more modern ish typewriter from the 70s. They still have the Tab key over on the right side of the keyboard. If you jump ahead to great late 70s or early 80s, you get the IBM Selectric electric typewriter.

And it finally had a tab key on the left side. And then of course, with our modern keyboards today, the tab key is over on the left side. So I want to talk about what types of accessibility testing you can do with just one key on the keyboard. And it's really fantastic. what you can do with the Tab key.

Now, I do want to start off and say that I'm talking about testing with the Tab key in order to find accessibility issues. I'm not necessarily doing usability testing with the Tab key, that is I'm not simulating how a user would navigate a website using the Tab key. Now, there are some users definitely that have to use the Tab key for navigating.

It maybe a dexterity issue or some other type of issue, but there's other users that use the tab key. Even the power users use tab key all the time. There's other users, for example, a screen reader user. Certainly can use the Tab key to navigate around the page, but a screen reader user also has many other options, such as using the up and down arrow keys to navigate through the Dom.

Well really, the accessibility tree, not the Dom. It's simple if you think of navigating through the Dom with the up and down arrow keys. There's other screen reader shortcut keys, such as H to go to the heading and and B to go to a button and things like that. So there's a lot of different ways to navigate a page, but I'm just going to talk about the Tab key.

And the main point was to use the Tab key to find accessibility issues and not to simulate a user experience. So I just want to start off with that caveat first. Now, why do I talk about the tab key? Well, anybody can test with the Tab key. It doesn't require any special hardware.

There's nothing you need to install. You don't have to spend a lot of money to buy some special hardware. Everyone typically already has a keyboard, whether it's built in your laptop or a separate keyboard for your desktop machine. It's very low budget, even if you had to buy one. You can get a $10 keyboard anywhere and do some testing.

Now, I talk about the Tab key because often when I'm teaching classes about the web content accessibility guidelines, I bring up a slide, such as this that shows all 78 WCAG 2.1 guidelines. And it's pretty overwhelming and that's the purpose of that slide is to show how many different things that we can check for and test for when we're looking for WCAG conformance.

Now, most companies don't check for AAA. And so, those are typically crossed off. So that's 28 that you can cross off the list. That still leaves you with 50 that you need to test for, which is still a pretty big number. It's not just only 50. So when I showed this slide, I say there's a lot of requirements here.

All 50 of these, sometimes that can be a bit overwhelming, so I try to boil that down to maybe a top 10 list. So this top 10 list here, this is based on actually WCAG 2.0 and not 2.1. So if I

wanted to update this and include some 2.1 settings, I might want to include 4.1.3 for status messages and usage of ARIA live.

I probably would include that if I update this list, but then I have to think about which one I want to cross off, and I'm not sure. I almost have to have a top 11 list. Also, your top 10 may depend on the type of site you have or what type of industry you're in. For example, this list here does not have any multimedia requirements.

But if you're a site like ESPN, or BBC, or maybe a TV network site, you're probably going to have a lot of videos. And so, I would probably want to include some 1.2 multimedia thing. I mean, whether it's closed captioning, or transcripts, or audio description, or something like that. One of those probably would be in a top 10 list if I was a media heavy site.

So this top 10 list is just a quick pass that WCAG 2.0 used on a non multimedia site. As I mentioned, it would adjust. But the point was going from that previous slide with 50 requirements for single and double A, down to 10-- and this is to help companies prioritize and focus on things first.

Now, the other 40 are certainly important and need to be fixed, if there's issues. But you can maybe focus on these top 10, as far as a lot of these top 10 can cause blockers for users. They may not be able to access your site or perform a certain action on your site if one of these WCAG requirements for success criteria does not work.

So if 10 is still too many, I often boil that down to maybe like a top 2 list. And this is where my Tab key comes in. So if you're focusing on 2.1.1, keyboard, 4.1.2 name, role, value, you can actually implement a lot of accessibility by focusing on these two success criteria first.

And then with keyboard, there's a lot of different interactions you can have with a web page with a keyboard, not just the Tab key. You have Enter, and Space, and arrow keys. And again, if you're using a screen reader, you can use some other shortcut keys. So I just wanted to focus on just these two requirements, which is the impetus for talking about the Tab key.

Now, when testing for these two requirements, obviously tabbing through the interface-- but I also do two other things in combination with the Tab key. The first is that I use a custom style sheet associated with my browser and I have a focus indicator in my style sheet. And I'll go into that in a minute.

And then, I also run with a screen reader. In this particular case, where we can run with a

screen reader running in the background, not really using any of the features of the screen reader and just combining that with the customer focus indicator and the Tab key. And you can find a lot of accessibility issues just by focusing on that.

So let me talk real quick about custom focus indicator and then I'll just briefly touch on screen readers. So all browsers have some way to support custom style sheets. And it's in this custom style sheet that I specify my focus indicator for all elements. And I do that.

So I'm going to slide here. The first line on the slide after the title is star colon focus. So that means every element on the page that can receive focus, I set the outline property. And in this case here, I'm setting it to x f0f, which is purple. And it's a solid four pixel just to make it a little bit thicker.

And I put bang important on there so that I can override whatever the page might be doing with its focus indicator. Now, I'll touch on that in a second. When you're testing the actual focus indicator for a page, you want to turn off your custom one. So you don't want to override what the page is doing.

This is just for keyboard testing that I have this turned on. So both Safari and Internet Explorer have options in their settings window that let you point directly to CSS. And it will load that CSS when it runs within browser. With Firefox, which is what this slide is showing, there's actually two different things you can do with Firefox.

One is you can go into your profile folder. You create a chrome sub folder. And the word chrome here does not mean chrome browser. It's talking about-- it's an older school word. Chrome, meaning the decoration of a window. So you go into your profile folder.

You create a chrome sub folder. And then, you put a CSS file in they're called userContent.css. And in fact, I can just show you that real quick. I have Firefox up and running. So if I go to about support, I will see my profile information. And down the eighth row in the table, they actually implemented this as a table.

And they have table row headers specifies. This is fantastic. The Firefox profile page is using semantic HTML, which is great. But down in the eighth row, there's a profile folder and there's an open folder button. So if I open that, it's my profile. And I already have a chrome sub folder because I already created it for my browser.

But typically, you won't have that. So if you create that folder and you put a file in there called userContent.css, the mind says x user content because I don't want this file to be used right now. I'm using a plugin instead of this feature here. So I renamed it so wouldn't be used.

But let me just bring it up real quick. And so again, as what was shown on my slide, I just more or less have one line in the CSS that says star colon focus. So every element that has focus that can receive focus, set the outline to purple solid 4 pixel and make it important.

All right. So that's one way to deal with Firefox. Now another way you can do it with Firefox-- and in fact, the only way you can do it with Chrome is using a plugin. And the plugin that I use is called Stylus. And one of the features of using the Stylus plugin is that you can turn on and off your custom focus indicator in the browser.

And with the previous slide with the chrome sub folder with Firefox, if I want to change my style sheet, I have to restart my browser in order for that style sheet to be reloaded. But with the plugin, I can just turn it on and off on that page or whatever page I happen to be looking at.

So it's a nice feature to be able to turn on and off. So that's what I'll be using in Firefox and I use it in Chrome, as well. And you can just go to your browser plug-ins or extension and search for Stylus and install that. And you basically have to edit a style sheet there.

And again, I only have one line in my style sheet. These other two slides are for Internet Explorer and Safari. I just included them here to be complete so that you can go and look at them later to see how you set an Internet Explorer, if anyone is still testing with that, and with Safari. I believe with Edge, with it based on chromium, that I think you can set your Chrome settings so you can use Stylus.

Should be able to use Stylus with Edge, but I have not tested that yet. Now real quickly for screen readers, if you have a Mac, then you have a screen reader built in called VoiceOver. And you turn that on and off using command F5. Now, I found when I teach screen reader classes, I found that learning voiceover on the Mac if that's your first screen reader you've ever used, there's a little bit steeper learning curve in learning the Mac gestures and things like that.

It's not just a simple tab through the website. In fact, when you're testing on Safari, as many of you know, you have to go in and turn on some options so that the Tab key will go to every interactive element on the page and not just form elements. So that's another other option you

have to turn on.

Now, the PC, there's actually quite a few options. The two most popular are JAWS and NVDA. JAWS is from Freedom Scientific and it requires a license to run. You can download a demo version and it runs for 40 minutes. And then, you have to reboot your computer, but you can try it out to see how that works.

Or you can use NVDA, and it's a free screen reader. They accept donations, if you want to do that. But you can go to nvaccess.org to download that. And I'll be using NVDA today. And if I have time towards the end of the show, a little bit of screen reader and tab support when I'm going through it.

So using a screen reader, don't be afraid of a screen reader, if you haven't used one before. You can actually just run it. Have it running in the background while you're doing your tab testing. The nice thing about NVDA is it has a live speech viewer. So you can open up a dialogue to see exactly what this screen reader is saying.

And then, you can copy and paste from that window and do a bug report, if you needed to. But it's nice to listen and to watch it, what it's saying. All right. So going back to testing with the Tab key, 2.1.1 keyboard, 4.1.2 name, role, value. Running with the Tab key with a custom focused indicator and with a screen reader.

So a little bit more than just the Tab key, although you can test with just the Tab key. But adding these two other items to it makes the Tab key much more powerful. So one of the awesome things about using the Tab key is that you can actually check many different success criteria. So in addition to 2.1.1 for keyboard, you can also check 2.1.2 looking for keyboard traps.

You can look for bypass blocks. You can check the focus order, which is fabulous. And one note about focus order, which we'll see in a minute for that success criteria, is that if that focus order typically in left to right languages, such as English and other romantic languages, the focus order tabs from left to right and then from top to bottom for left to right languages.

If the focus order moves from right to left, that's not necessarily a 2.4.3 focus order problem. If you look at the success criteria carefully, it says if the focus order affects the operation or the meaning of the page, then, it's an issue. But if it just tabs from right to left, but doesn't really affect the operation, then it's OK.

We'll see an example of that in a minute. You can also check for a link purpose probably not just for the Tab key by itself. If you're running with a screen reader and you're tabbing, and you can listen to how that link sounds and see if it has enough context by itself or there's some conditions with 2.4.4, for example.

If the link is contained in the same paragraph as its context or if it's in the same table cell. If the context is in the same table cell as a link or if there's a table column header, and some other cases like that. You can check for focus visible, but again, I have this asterisk that when you're testing for the actual 2.4.7 success criteria, you want to turn off your custom style sheet because you want to see what the page's focus indicator really looks like.

And then, it's visible. But if your focus indicator-- if your custom style sheet is turned on and your focus disappears, then that is a 2.4.7 issue, even though you have your custom style sheet turned on. And it's also an indicator there's an element on the page that got pushed off the page, maybe through the x-coordinate or could be passively set to zero.

That it's fakey-- anyway, we'll talk about that in a second. You can check for the unfocused success criteria, 3.2.1. When an element receives focus, does it cause a change of context? And also, your identification 3.3.1. Typically, testing this with the Tab key and the screen reader so that if you try to submit a form with errors and the errors are displayed in text, which is fantastic.

But now, if you tab off of that form element and then tab back to it, do you hear that error message? Is that error message associated with that form entry when you tab off and tab back on? Typically, that's done through ARIA described by. But that's another thing you can test for using the keyboard or using the Tab key, and the screen reader.

All right. Let's talk about a few examples here. So when you're tabbing through the interface, pretty much anything that you can click on with a mouse should be both focusable and actionable with a keyboard. So I should be able to put my focus on that element. I should be able to tab to that element.

My focus should be on it and I should be able to see that my focus is [INAUDIBLE]. There needs to be some visible focus there. And then, once my focus is on that element, I should be able to select it, or choose it, or whatever the element might be. It could be the Space key, could be the Enter key, the arrow key, whatever the interaction is.

Now, that's more than just the Tab key, but while you're there with the tab key, hit space or Enter to see if you can select it, or arrow key to expand it, or whatever that case might be depending on the widget. And you also want to look for the focus indicator and this is with your custom one.

As I mentioned before, look for any elements that don't display a focus indicator. And that would be with your style sheet turned off. Or notice when the focus indicator just disappears from the screen. And as mentioned, that often happens when elements are fake hidden because they're either put underneath another element, or they're pushed off the top or the side of the page, or their opacity is set to zero so that you can't see it.

But a keyboard user can still tab those elements. A screen reader user can still hear those elements when they navigate to them. They're not really hidden. And so, when you run with this focus indicator with a custom style sheet and the Tab key, you can find those issues. And then if you are checking for the real focus indicator, you probably want to test on the various browsers, especially if the page doesn't have its own custom focus indicator.

If the developers or designers didn't design one, they're just using the default browser one. Then the different browsers have different focus indicators, so you're going to want to check those different browsers. And as mentioned, when an element-- when the focus disappears, look for the left or the x properties. Look for opacity, whatever the case may be.

If you're running with a screen reader, it's fantastic. Because if your focus disappears, you can't tell where on the screen it is. But with a screen reader running, you can hear what that element is saying. It might say read more, or hamburger menu, or something like that.

You can't see it, but you heard what it is. Now, you may have an indication and go, oh, OK. Yeah, I know exactly what that element is. We go look at it and then, inspect the code. Now, if your focus does disappear when it's either one of these fake hidden elements, you can use the Code Inspector to see which element is on.

But if you're on a PC, F12 normally opens the Code Inspector. Don't use the F12 key and don't try to right click with your mouse because you don't know where the focus is. It disappeared. You can't right click on it. If you try to use F12 to open up the Code Inspector, it puts your focus at the very top of the Dom in the Code Inspector.

What you want to do is use the context menu. Again, this is on the PC. Use your context

menu, which is often to the right of the Space key at the bottom right of your keyboard. When you press the context menu, you'll get a context menu will appear. And the image on this screen here is the one from Firefox.

And there's an Inspect Element about halfway down the menu. It also has a letter Q next to it, which is a shortcut key. So I can hit my context menu and the Q button, and very quickly bring up the Code Inspector and inspect that element. With Chrome, if you do that context menu, I believe at the very bottom of the menu, it is the inspect menu item.

So I can hit my context menu key, hit the up arrow, which will wrap me to the bottom of the menu and hit Enter. So that's also fairly quick, quick way to inspect that. And I'll show you an example of that when we get to it. And then, if we can get to where the demo here, I'll show you.

You can tab through the interface with a screen reader running to inspect all the elements and listen for things that are displayed. Let's jump over to the browser. And I'm going to lowes.com. So I've been doing a little bit of home renovations. And with all the quarantining in effect around most of the globe, I've been doing a lot of shopping online.

So that's the only reason I'm here on the Lowe's website. I'm not picking on them for any reason. It's just that it's a personal page I've been using and, probably like most of the accessibility nerds out there, when you're on a page for personal use, you test it with a keyboard. You test it with a screen reader just to see how it does and just keeps your skills in check and up to speed.

So it's a fun thing to do. All right. So I've got Lowes.com up here. And it just loaded. Let me tab. Now, I'm in the page, but I don't see my purple outline. And you probably can't see this, but in the bottom left corner of the browser, I can see the URL that, apparently, my focus is on a link. And it's pointing to Main Content.

Tab again. There is still don't see my focus indicator, and it's pointing to main navigation. Those sound like skip links to me. Could be that they implemented a skip link just for screen reader users and not for sighted keyboard users, which would be a shame. But let me tab again.

All right. Now, my focus is on the COVID-19 banner at the very top. And tab again. And now, my focus is on the Lowe's logo in the upper left to the page. Now, you can see the purple

outline. Well, let me go back. I'm going to do Shift Tab. So I'm back on the banner.

Shift Tab, Shift Tab. So now, I'm on these hidden elements. So I can't tell where my focus is. I would like to look at the Code Inspector to see where the problem is. So I'm going to use the context menu on my keyboard. So the menu shows up in the upper left corner of the browser.

And I'm just going to use the up arrow to get to Inspect Element. I just used the up arrow because I know it's only five up arrows. Whereas if we use down arrow from the top, there's 8 or 10 down arrows. That saves me five keystrokes. Whatever. Anyway, context menu. Up arrow to Inspect Element.

There's the Code Inspector. And indeed, the actual test associated with this link is skip to Main Content. And then afterwards, there's another link after it in the Dom called skip to main navigation. So these are skip links, but I believe-- I'm guessing that the Lowe's developers may have added this banner, this COVID banner fairly quickly.

Let me inspect that in the inspector, Code Inspector. There is a global message bar. So here's a banner here. Let me go over in the style sheet. Let me hide this banner. I'm going to the message bar class because that's what I see on this Div here. I'm just going to add display none to that element.

All right. So now, that banner is gone. And let me-- Yep. So that's all it was. So now, they actually have a skip to Main Content, skip to main navigation, and then the Lowe's logo. So I think just in their rush, maybe, perhaps to put up the current virus banner on the top, they covered up their skip links.

So that would be a problem. You should still report it, if you happen to be testing it now. If somebody was testing for Lowe's, I don't know if they are. But I would report it, that the focus indicator disappears, that these elements are underneath the banner. I believe-- let me go to Amazon real quick.

Amazon also has-- well, pretty much any website you go to is going to have a banner at the top about virus updates. Now if I tab, here at the first tab, I get to skip to Main Content. Fantastic. They have a bypass block. Plus, that link is on top of this virus banner. So Amazon did a good job there.

Now, they missed a few other things, which I'll show you in a second. Let me jump back to Lowe's. All right. So that was the first issue was-- go back to my slide. So we saw 2.4.7 focus

visible. It was not visible because it was underneath. Fantastic. All right. Caught one issue.

Let me go ahead and start tabbing through. So now, if you notice, there's two tab stops on this magnifying glass. Now, that's not necessarily a problem. You probably don't want to have-- you want to reduce the number of tabs stops you have, if you can do that.

But I'm a little curious why there are two tab stops here. Let me do the context menu. Again, Inspect Element. They have a button. Fantastic. They're using semantic HTML. Inside that button is an eye element. I usually is-- well, it's the italic tag or used for emphasis. It's changed or morphed its meaning over time.

Now, I is often used to be an icon pneumatically, but it doesn't really matter what you call it. So they have an I element. They have role equals button, tab index equals 0, which means I can tab to it. But they also have ARIA hidden equals true. So for a screen reader user, I can tab to this icon, which is the magnifying glass.

But I'm not going to hearing anything because ARIA hidden is set to true. There really isn't any reason to have this I that's focusable. So they should remove that from there. Anybody who's listening that works for Lowes, here's some free advice there. Anyway, so there's two tab stops there.

So let me keep tabbing. Now, I'm in the search field. Tab again. I'm back on the icon. Tab. Look at this. I keep tabbing and I'm stuck. That would be 2.1.2, keyboard trap. That's probably-- so if I inspect that code-- So I'll do a context menu.

I'm not going to arrow up through the menu anymore. I'm going to use the Q key because it's really quick. They have an input field that has an event handler with a lot of different keyboard event handlers. So I am guessing-- and I think the way the search field works on Lowe's is that as you start typing a search, they're listening for the keyboard because they want to do some auto completion and give you suggestions.

So they're correctly listening for keyboard events, but the problem I think here is that they're also listening for the Tab key or maybe they're accidentally listening for the Tab key. And they're not letting the Tab bubble up and let the browser handle the Tab key. So they really should be letting it bubble up and then, handling all of the other keys themselves.

In fact, I think if I can't Tab out of here-- it's interesting. I think if I hit Enter, it expands it. Still

can't tab out. But if I start typing-- so they do some auto suggesting. So having a keyboard handler is OK. They just need to add a special case for the tab key and let it filter through. All right.

So let's close that. Let me keep tabbing. So now I'm on the main menu of Lowe's. And the main menu says shop, idea, savings, and services. Hit Tab. I'm going from left to right. Fantastic. Now if I keep tabbing, I get to the menu over on the right side, but it went to the far right. It went to weekly ad.

And I'm tabbing forward. So if I tab forward, now it goes to Lowe's, Lowe's credit cards. And then, tabbing again goes to order status. So it's tabbing from right to left. So as I mentioned before, for our focus order 2.4.3, it's not necessarily a problem. If we look at the criteria for 2.4.3, it says that navigation-- if this navigation sequence is affecting the meaning or the operation of the page, then it's a problem.

In this case here, the fact that I tabbed to weekly ad, Lowe's credit cards, and then order status, so what. It's OK. It's not a big deal. They're probably using float right. So in a Dom, it has weekly ad, and then credit cards, and then order status. So it's correct in the Dom.

And then, they use float right with that element over to the right. So it tabs from right to left. All right. Let's keep going. OK. Now, I'm down in the recommended search area about halfway down the page or so. And it's on pruning shears. So I actually was doing some pruning on my tree.

So I bought some pruning shears. So now, they have all these other recommendations on garden gloves, and rakes. And I guess they figure I want to go out and do some more work outside. So my focus is on that very first element of recommendations. Now if I tab, now the focus is on the far right on the right arrow for this. I guess it could be considered a carousel.

So it jumped over those other items in the list. So by having this focus indicator and tabbing, I can see very quickly where the problems are on this page. Well, that's good for Lowe's. Let's jump over to Amazon. And I mentioned they have their skip to main on top of the banner. And so, that's good.

Let's tab through here. I'm on the left side of the page on the hamburger menu. Amazon logo. Tab through the search field. Fine. All right. Well, I went from the search field to the sign in page. In between there, there's a language page. It jumped over that. Now, that's not

necessarily, again, a 2.4.3 issue that it jumped over it.

Hopefully, I can get to it at some point, but it's a little interesting that it jumped over it. Let me just keep going. I got to cart. Now, I'm on a menu of items. Tab through that. All right. So now, on the last one. I just tabbed. My focus disappeared. Tab again.

Oh, now it's on the English setting. Well, there's something between that language setting and that shopper tool kit as the last menu item. Or something where it's-- something that's hidden that I can't see. So again, let me bring up the inspector. Well, that's interesting. So they have a link.

Let me put my focus up above it so that some of-- you can see the colors here. It makes it a little bit easier to read. So my focus was on the link that has a class of Nav hidden ARIA. So they've hidden it. The text for that link is disability customer support. And it has an ARIA label of click to call our disability customer support line or reach us directly at 1 888 whatever.

All right. Well, that's an interesting link. They have a disability link that perhaps, maybe I have lower vision and augmenting my experience using a screen reader. But I have enough sight where I can see, but I'm using the keyboard. But it could be a really important link for me to get to.

And so, they've hidden it somehow. They probably think they pushed-- Yeah. It looks like-- so their left position is at minus 10,000. I'm looking at the style sheet over the far right. The width and height are set to 1 pixel, overflow is hidden. So let me just-- turn off. I'm just going to change the class name to put an x in front of it so that it's not using that class.

And let me tab. All right. You may not be able to see that. So on the far right of the main banner, my focus is on the disability link. It's off the screen a little bit. Plus, it's blue text on a black background so it's really hard to see. We could change that. We could change it to white text.

But the general point is that there's a link that's hidden that could be useful for a lot of other users. I think there's one more on here, as well. So let me tab again. So I'm on the language page. Tab again. The focus disappeared again. All right. Let's inspect the code.

All right. It's on a link. Expand that link. There's a lot of text there. It says welcome to Amazon. If you prefer a simplified shopping experience, try the mobile web version of Amazon at amazon.com/access instead of using the mobile version, similar to the mobile app or stay on

amazon.com for access to all the features. Well, that's another cool feature that would be really nice to see that other users might like to see.

Again here, let me change the class name so that it shows up. Now, this link is on the page. Now, Amazon probably doesn't want this big fat link with all this text showing all the time. But they could treat it like a skip link. So if a tab through, it appears when I tab to it. Otherwise, I don't see it.

I mean, that's a really fantastic feature. Now, what they don't tell you, though, is if you follow this link and go to Amazon.com/access, they save a cookie on your site. Now, every time you go to amazon.com, it goes to this limited site. And they don't have an easy way for you to go back to the full featured site.

So you have to go into your browser, cookie settings, and delete the Amazon cookies in order to get back to the main page here. That was interesting there. Oh, one of the things I didn't point out was that-- so on Amazon, they really like their tab index. They have tab index on everything in their header, in the banner.

All the menus have tab index, which you can get away with if you specify tab index for everything, but it's pretty dangerous. I would not recommend doing that. But if they got it to work OK, then fine. All right. Let me jump back to Lowe's real quick. I want to show something with a screen reader and the Tab key.

Have a few minutes left here.

COMPUTER: Lowe's home improvement. Lowe's home improvement model of Firefox.

GLEN WALKER: I think you can hear that. I'm going to start back up at the address bar.

COMPUTER: Navigation tool bar. Toolbar.

GLEN WALKER: I'm just going to navigate into the page.

COMPUTER: Lowe's home improvement document. Skip to main content link. Skip to main navigation link.

GLEN WALKER: Now, my next tab is going to go to the Lowe's logo link, which takes you to the home page no matter where you are. But listen how this announced. Let me turn on--

[INAUDIBLE]

GLEN WALKER: Speech viewer.

COMPUTER: Speech viewer dialogue. Show speech viewer on startup check box. [INAUDIBLE] Lowe's home impro--

GLEN WALKER: All right. So I have the speech viewer up. You may not be able to see. That's a little bit smaller font, but it lets you see what the screen reader is announcing. So listen for what's going to happen. And if you can see the speech viewer, listen to what's announced for the home logo.

COMPUTER: Clickable banner landmark. Lowe's home improvement logo. Graphic link to Lowe's home improvement home page. Linked to Lowe's home improvement home page. Visited link.

GLEN WALKER: Wow. That's a lot of stuff, isn't it? Let me turn off the--

COMPUTER: MBD.

GLEN WALKER: Speech viewer.

COMPUTER: Lowe's home improvement model of Firefox.

GLEN WALKER: And let me turn it off. All right. Thank you. Turn off the screen reader. Let's look at that link because it was reading so much information there. So it is a link. Fantastic. It has-- so the link itself does not have any visible text, so there's nothing for this screen reader to read. But it's going to look at the child elements of the link of the anchor tag.

And there's two child elements, an image and a span. Now, the image has alt text. Great, because this is an image that has meaning. It has information. So their Alt text is Lowe's home improvement logo. Well, OK. That's great. I'm glad to have that. The word logo probably isn't necessary.

You just say Lowe's home improvement. That's fine. Now, they also have ARIA described by and it's pointing to the span element. So when it reads the image, it's going to read the Alt text as the accessible name and it's going to read the ARIA described by as the accessible description.

So it's going to read Lowe's home improvement logo and then, it's going to read linked to Lowe's home improvement home page. That's all just for the image. And now, it's also going to read the span because that's also a child element in the anchor tag. So it's going to

basically repeat the linked to Lowe's home improvement page.

Now, the span itself doesn't need the word link in it because it's in an anchor tag so it's already going to state link. So now, we're getting three pieces of information that are read just for this one link when really, all it needs to say is Lowe's home improvement. But I just thought that was interesting.

Again, my name is Glen Walker. My contact info is gwalker@applause.com or glen.walker@gmail.com. And Glen is spelled with one N. Hopefully, we've answered all your questions during this presentation. Thank you.