

LILY BOND: Welcome, everyone, and thank you for joining this webinar entitled "11 and 1/2 Free Tools for Testing Website Accessibility." I'm Lily Bond from 3Play Media, and I'll be moderating today. I'm lucky to be joined by David Berman, who you may recognize from past webinars with us.

David is an internationally acclaimed expert in web accessibility, as well as the top-rated speaker on the subject. Thanks to his leadership and experience in the field, David has been appointed as a high-level adviser of the United Nations and has spoken in over 40 countries in the past few years. And since our previous webinar with David, he is now an invited expert to W3C.

His presentation will be about 45 minutes. And then we'll leave 15 minutes at the end for Q&A. And with that, I will hand it off to David, who has a wonderful presentation prepared for you today.

DAVID BERMAN: Hey, Lily. It's great to be back with another 3Play webinar. Great to be back.

LILY BOND: Great to have you.

DAVID BERMAN: Thank you so much. I know that we've got people who have joined us-- over 100 who were with our last webinar and perhaps about 500 who are new. So well done, and thank you all for taking the time to join us. I know this is an upbeat topic. But there's also some urgency.

I was particularly interested, Lily, in the challenge of what's been going on with the scandal with Volkswagen. And you may be wondering, what in the world does that have to do with website accessibility? But I was thinking, in the same way that VW has committed its crime against the planet in making their product comply with testing tools but not actually work in the real world, before we get into talking about tools for testing website accessibility, it's really important that we know that no matter how many tests we do in terms of automated tools, ultimately the best test is making sure that the site works for people on the road, that is to say people living with disabilities who are actually using the product. So at the very end, we'll talk a little bit about usability testing. But today I'm really excited about sharing my favorite tools for testing website accessibility.

I was really struck by-- and I'm going to show you this right now-- that it's important that we have reliable testing. And it's also important that the declarations we make to the world are

valid as well. So when this whole VW scandal broke open, I was really intrigued with the idea that they were gaming the results of the testing.

But I also went deeper and looked at their sustainability report, because in their sustainability report, they talk about inclusiveness and accessibility. And they also make some sweeping dramatic statements. Just like they claim to be committed to being the most environmentally friendly car-maker in the world, they also make some sweeping arguments about how their whole community is going to be inclusive. And so I thought it would be intriguing today that as we look at these testing tools, we're going to be testing VW's website and looking at their documents and seeing how well they do in terms of delivering products that are authentic to their commitment to the world.

But also, I know a lot of you haven't heard from me before. So I thought I'd just explain the credentials. The reason I know that 3Play likes to have me do presentations is that I'm from Ontario. And here in Ontario, we have the most forward regulations in the world when it comes to accessible websites and accessible documents. We were the first jurisdiction on the planet that insisted that not only government but private companies and NGOs, everyone with a certain minimum number of employees has to have a public-facing website which complies with international standards for accessibility. And so that's resulted in a community of excellence, in a community of innovation here in Ontario.

So it's no surprise that some of my favorite tools I'll be sharing with you today, such as the AChecker tool and the aDesigner low vision simulation tool, are made right here in Ontario. And true to the form of our 11 and 1/2-- Lily, what's the half? I guess we'll find out-- 11 and 1/2 free tools, it's no surprise that some of them are going to be from Ontario.

So I'm just going to step back a moment and also share some great news I have, which is that 3Play and I have also collaborated on a white paper, which is called "Solving Web Accessibility-- Leaving No One Behind." And that white paper is going to be released within the next week. So I realize that I'm going to go right into the accessibility tools here. But I know for a lot of you, you may want to know the background of what's the bigger picture? Why are we doing this? And what are our motivations?

I'd like to point you both to that white paper as well. You can watch our last webinar online as well. And Lily can give us guidance as to where you can go to get that. So you can get an hour of me telling you why we got to do this, what the business benefits are, why we're driving down

costs, increasing reach, and doing the right thing as we go on to it.

But the biggest request we had the last webinar is people want to know, OK, show us your favorite tools. And that's what today is going to be about. So with no further ado, here's my last favorite tool from Ontario. It's the Berman Accessibility Ribbon for Word. We actually created a tool that helps people create more accessible documents.

And the way we do that is we created a ribbon, which is free. You can download it. And it works in Windows 2010, 2013, and 2016. And unfortunately, it doesn't work for Word for Mac. Word for Mac still is not as clever when it comes to accessibility.

But what we did is we took all of the features in Word that help make documents more accessible and put them into one ribbon. And by the way, I should say if you're using Word for Windows 2007 or 2003 or even 2000, there's also a really good ribbon that was developed in Texas. And we've got a link on the same davidberman.com/ribbon. We've also got the link to that one, as well. So no matter what version of Word for Windows you've got, we've got it covered.

Now we need to remind ourselves what we're testing for. And WCAG 2.0 is the international standard for accessibility. No matter where we go in the world, we find that governments, organizations, universities, colleges are pointing towards WCAG 2.0 as the way to measure the minimum level of compliance. And again, in this webinar, we're assuming most of you have some sense of what WCAG 2.0 is about. And if you don't, please consider listening to our previous webinar.

But I wanted to make sure that we're on the same page. So I wanted to remind ourselves that for most of us, we're seeking Level AA of WCAG 2.0 conformance, or perhaps at least Level A. So you may know already, WCAG 2.0 is made up of a whole series of success criteria. 25 of those success criteria are called Level A criteria. And therefore, if you conform to all 25 Level A criteria for your website or your document, you can say that entire website or document is WCAG 2.0 Level A compliant.

If you also comply with the 13 additional Level AA standards, you can say your document or your website is a Level AA-compliant document. And essentially, some people are just testing products just to make them more accessible because they feel-- and they're right-- and it's the good thing to do. But for those of us who are testing and trying to be able to make a declaration that our product conforms with an international standard, we're typically for a

website saying we're trying to get a level of WCAG 2.0 conformance.

And indeed, in the United States-- and I looked at the registration list, and we've got a lot of people from the United States of America today. And so I wanted to clarify that although the US government has a different standard, called Section 508, that that Section 508 is in the midst of a refresh, which within the next year will also point directly at that WCAG 2.0 AA standard. So if you're in the US government or a university or a college who is complying with ADA standards, we know that Section 508 really does point to the same idea.

And therefore, very conveniently for all of us, we can simply say, let's try to get to WCAG 2.0 AA or at least A. And we know we're taking care of Section 508 requirements as well. And therefore by definition, also a lot of you hear about, hmm, I'm supposed to make sure my product is ADA compliant.

Now ADA stands for the Americans with Disabilities Act. And again, this points back to Section 508. So whether someone says, you need Section 508 compliance or you need ADA compliance, as long as you're WCAG 2.0 AA compliant, you're there.

There's one other international standard we have to think about. And it's called PDF/UA. Now PDF/UA is just about PDF files. And some of our documents are PDF files, and many of our websites include many PDF files.

And PDF/UA is another way of defining that same accessibility. But the good news is that the WCAG 2.0 rules, or the WCAG 2.0 success criteria are very much the same ideas as in PDF/UA. And so therefore, we're able to use, for the greater part, often the very same techniques and the very same tools.

And so some of the tools I'm going to show today are just about WCAG 2.0. Some of them are just about documents. But essentially, the same principle's underlying whether we're trying to reach WCAG 2.0 or PDF compliance.

Now also, some of us are working with mobile devices. And so we're thinking, hmm, how am I going to make my iOS application compliant? How would I make the browser-based experience of my website on iOS compliant, or on Android, or on BlackBerry, or on goggles or my smartware or whatever's coming down the pipes?

And the good news is, again, that the WCAG 2 standards are carefully designed to be device

independent. And so we're agnostic to what platform we're on. But when it comes to testing, we can be constrained, because we can discover that this testing tool is a Windows program. And this testing tool, well, how am I going to make that work on Android?

So we thought that through as well. So you'll see some of our tools today only work on Windows. Some only work on Mac. Some are great for mobile. And I'll touch on that as well.

Essentially, when it comes down to, then, coming up with a testing regimen, we have to acknowledge that some things are going to be completely tested by machines. In fact, some things can even be tested and fixed by machines, whereas some of our testing must be done manually. And so any time someone is trying to prove that a website or a document complies or exceeds WCAG 2, they need to use a mixture of things you can automate and things a human has to do.

Now of course, the things you can automate can be very quick and more satisfying. You know, OK, those checks are 100% done. Whereas the manual ones can often be much more subjective. I want to show you testing tools in both camps.

Ultimately, we also have to ask ourselves before we start testing anything, why are we doing this? Because some people have a requirement to prove to their government or their board of directors or their users that they meet or exceed, let's say, WCAG 2 level A or WCAG 2 AA. Some people are simply trying to make the most accessible experience possible. Some people are just trying out stuff.

It's really important that before you start testing anything, that you decide why you're doing it and have clarity amongst your team. Because let's say some people are thinking we're going for level A, and some people think we're going for level A and level AA is the best practice. And some people think we're going for AAA. Well, if we go into testing without having a clear definition of why we're testing, then we may just be wasting our time and using up valuable resources. So it's really important we start out.

Now, I'm going to go from the assumption that we're using these tools to try to create a document or a website that's compliant with WCAG 2 AA or PDFUA. And for that purpose, then, I want you to be aware of something you may not have heard of. I know most of you have heard of WCAG and the W3C. That's the organization that develops and supports the WCAG standard.

But you may not know that in the last year, something's come out called WCAG-EM. WCAG-EM is a definition of what a compliance report should look like. EM stands for Evaluation Methodology because a lot of organizations like ours, at any given time, we have maybe a dozen formal evaluations going on of websites and documents in our organization. We do them for government. We do them for the private sector.

And we make sure that all of our formal reports are WCAG-EM compliant. And that means the report has a certain number of properties which say, this report has all of the things in it that an expert would expect to see. So when we make a declaration, when we give an expert opinion, it's got all the components that that opinion require. So we love that W3C pinned this down in 2014, what should such a report include.

And they even developed-- and this was released just in the spring of this year-- the WCAG-EM report tool. Now, you don't have to use the WCAG-EM report tool, but for those of you who have never done a WCAG-EM compliant report, you may find it pretty handy. The report tool is a very approachable tool that allows you to generate a report that makes sure you have all the pieces.

It is not a testing tool. It will not test your site in any way. What it will do is make it easy for you to decide what your goals are and then allow you to explore the website or the document, decide how many pages are relevant because WCAG allows for a statistically relevant sample of a huge product to be tested because it's often not realistic to test everything on every page. And then, finally, a way to organize your findings in a way that's useful. So consider checking out the WCAG-EM report tool if you're interested in that type of formality.

Now, the W3C, the publishers of WCAG-- and I'm proud, Lily, to say that I'm now an invited expert to the W3C. They asked me to come and help them with educating and outreach on the WCAG and other aspects of accessibility W3C puts out there. So that's thrilling for me.

I've got to say the W3C suggests testing tools. And I've listed them here on the slide. I'm showing a list of them, and I give a URL where they have a more up to date list of tools. But W3C's list, although it's their official list of good tools and they're all solid, doesn't necessarily include all my favorites. We're working on that.

The first tool I want to share with you is the type of tool that looks at an entire website. Now, there's a company out there called HiSoftware. And they put out a product that some of you may have heard of called Compliance Sheriff. Now, Compliance Sheriff is perhaps the most in

depth, enterprise-wide web testing tool there is for accessibility, but it's also very expensive. We're talking like five digits a year.

But they also kindly get out for absolutely free something called Cynthiasays.com. And Cynthia Says is a website. And we're going to go there right now, and we're going to see what happens with the VW site if I run it through Cynthia Says. So I'm just going to explain this as I do it.

I go to Cynthia Says. And now I've wandered into a browser, Cynthia Says. And you can see down here at the bottom, that I've got a place where I put in my URL. So I'm going to put in the URL vw.com. Now, we then get some choices. So we can choose to test for Section 508. I'm going to test for WCAG 2 AA.

Then I agree to the terms. Then I click Test Your Site. And away it works-- chunk, chunk, chunk, chunk. Takes a few minutes. While it's thinking about that, we're going to go look at another tool. Put this in the oven. We'll take this out of the oven shortly, kind of like a cooking show. So while we wait for Cynthia Says to take probably about two or three minutes to chump through that one page, we're going to look at another product.

Now, I mentioned how Compliance Sheriff and Cynthia Says have this relationship where there's a paid tool and a free tool, well, the other big player in that world is called Deque. It's pronounced like Dairy Queen, DQ, even though it's spelled D-E-Q-U-E. And they also have a huge, powerful, amazing tool that costs a lot of money. But they also have a single page analyzer. It's called the Worldspace Single Page Analysis. And it's very similar, like the Cynthia Says tool. So you can try them out.

The reason, for those of you who are more nerdy, the Deque product is more powerful in that there's a plug-in for Firefox that allows you to actually track your issues and work as a team and mark down what issues have been fixed and what haven't. So you go into Firebug and Firefox and you manage that. Having said that, the setup is kind of tricky. But if you're into it, it's a very powerful approach.

I'm going to go back to Cynthia Says now and see how it's done. Hm, would have thought it would be done by now. That's OK. We'll come back and check the oven again soon.

So there's a number of these types of tools will give you an analysis of a page one page a time. And we're not going to look at this, but I am going to tell you about it. This is a nice one.

It's from a company called Tenon. And Karl at Tenon has, again, made a powerful tool. You put in one URL and it gives you very human readable results. I think that's the Tenon tester's superpower, if you're trying to choose. The Tenon gives very clear answers in a really nice interface. Good graphic design.

Now, the next type of tool we're going to look at are toolbars. And there's a number of different toolbars that you can plug into a browser. Our favorite is called WAVE. WAVE toolbar, again a completely free product, is a toolbar that you can put into Firefox or Chrome. They even also have a plug-in for Dreamweaver if you're using Dreamweaver. I'm going to show it to you in Firefox. So I'm just going to pull up Firefox right now and show you how this works.

So here we're back in Firefox and Cynthia Says is completed. So let's come back to Cynthia Says. Here's the type of results we got looking at the vw.com home page. And just so you know what we're dealing with, this is what the vw.com home page looks like. So it's a page that looks kind of like that. And now Cynthia Says has lots to say.

So here's Cynthia Says's report. And it's a little hard to read. I mean, it's hard to read even if I zoom in on this, because I know the type is quite small. I'm just going to zoom in a little bit. But it's well organized.

What they do is they give you a tree structure, and they've got all the level A issues and all the level AA issues. So I've just collapsed the whole tree. Now if I open up the level A issues it says, success criteria 111. That's a WCAG thing. All the WCAG success criteria have numbers. So it's saying, hm, I found a bunch of things having to do with criteria 111.

And then it names the rule that it believes has been broken. And it talks about the techniques because WCAG 2 has technique issues. So this is pretty handy. It's saying, you should use Alt attributes on image elements. If I were to follow this link, H37, it would take me to WCAG 2.0 Technique H37. The H stands for HTML. The 37 means it's one of over 100 techniques for HTML that are in WCAG 2.0. And if I took it there, it would explain to me how to fix it.

And now it goes into every instance it found, line by line. So it's showing you the line of code. So if I were to look at the source code of the vw.com site, and I looked at line 409, I would find that indeed there's an image. And it has no alt attribute. And well, that's a Level A fail because those of us who know WCAG 2.0 know well that every image must have an alt attribute. Whether it's empty or not, it must have an alt attribute.

So just for these reasons, the vw.com site would fail WCAG compliance, even though this is easily fixed. And indeed, I could look at the other parts of the report now. Some of the failures are marked with an X to say, Cynthia Says says, I absolutely know this is a failure.

Some are marked with this symbol of an eye. And that's saying, I need a human to take a look at this because I'm not sure if it's a failure or not. But would a human please take a look? And then you go into this instance and look at the issue and decide for yourself.

And then the check mark shows cases where everything is hunky-dory. It's saying, I checked for this thing. And it was cool. And I'm confident a machine test can tell me that there's nothing else to check for.

So as you go back down the list, there's a whole bunch of criteria. And it goes through them all. So it's gone through every one of the criteria at our Level A in WCAG 2.0.

And then it goes through every one of the criteria in AA. And for each one of them it says, I'm sure it's not right; or I'm kind of not sure, please, human, check it out; or I'm absolutely sure we're good. Now it can't be absolutely sure we're good very often because most of them need some human check. But a great system-- absolutely free.

So now I'm going to go back to the VW homepage. And I'm going to show you the WAVE toolbar, another type of testing tool. So in the WAVE toolbar, now here's how the WAVE toolbar works. You see this right here in my Firefox, I've got this extra toolbar. Normally, it would look like no toolbar. Now I'm adding the toolbar.

And how do you install that toolbar? You google WAVE toolbar. You'll find it. Click Install. Two minutes later, you've got this toolbar in your Firefox.

So the WAVE toolbar has basically four areas of superpowers. The first one is called Errors, Features, and Alerts. And I'm going to follow this link, Errors, Features, and Alerts. And it does a quick analysis. And it's saying it found, uh-oh-- WAVE has detected 47 accessibility errors. Hmm.

Now that may sound like an awful lot of errors. But often it's the same thing again and again. And if we scroll down the page, you'll see WAVE has now inserted visual cues to say, hmm, there's a problem here. I've got incorrectly ordered headings. It's thinking, huh, why am I seeing H2 before I see an H1? That's what it's kind of saying.

As we scroll down, we see all these different markers. Red is for an absolute fail. Yellow's for a maybe. And then there's greens to show you, hey, this is done perfectly well.

And we find again and again in all these testing tools there's always these three categories. There's always things that are absolutely definitely wrong, things that may be wrong and need a human to inspect, and things that are cool. And so as I hover my mouse-- whoa. Here I'm on another page. As I hover the mouse, it explains to me why there's an issue. This picture of a car, it's missing alternative text. So this marker is telling me that's an absolute fail.

Now the second superpower of WAVE is it has a structural order analyzer. And what this does is it shows me the order of the code of the object. So you could tab through a page or have a screen reader read it all out aloud to you.

But for those who can see, this is a pretty handy way of being able to quickly see the order of objects on the screen. And they're numbered. So typically they may go top left to bottom right. But often, it's defensible that things go in different orders than the visual order.

The third of the four areas is the text-only view. We use this a lot. It's pretty handy. What's this it's showing me is if you linearize all the content on the page, that is, if you drag it out in one long line, this is the order the screen reader would typically announce things.

So first of all, it shows me amazing things, like I don't hear that this is the Volkswagen site for quite from time until I'm into it, which isn't so good. The first thing it's telling you is about live chat. That's probably not the best. And as I go through, I can see the order of everything, good and bad. I get to experience everything that's going to be announced by a screen reader, as well as what order it's in.

And then finally the fourth shows me the heading structure. It's called the Outline View. And it shows me the heading structure of the page, which in this case is pretty good. They have almost one and only one H1. And then they have a selection of H2s.

But if we go to their homepage, then we'll find that there's more heading troubles. So we go to their heading homepage. I discover, oh no, look at this-- red markings for headings that are empty and oddly, a whole bunch of H1 headings that really shouldn't be here. So there's troubles there. So that gives you a sense of what WAVE can do.

Now if you're forced to use IE, we have a backup for you. It's the Accessibility Toolbar 2011 for

IE. But most of us don't tend to work in IE when we don't have to.

Now here's another tool, Karl Groves' Diagnostic.css. And this is a cool thing. Karl created this tool for you to very quickly find errors on a page. And I'm going to show you this ever so quickly.

And there's a page I'm going to feature. We're going to go back to my Firefox. Just going to have to pop out of full PowerPoint window here to be able to bring that up. Hey, since Canada's Toronto Blue Jays are definitely in the playoffs for good, I'm going to go to the MLB page.

So I'm going to turn off the WAVE toolbar. Thank you, Boston, for helping defeat the Yankees. [LAUGHS] OK. Do you see right up here I've got this button called Diagnostic.css? This is Karl Groves' Diagnostic.css.

And it's so simple. You put this button on your toolbar. And if you Google this, you find out how in five minutes you can have it installed.

I click this button. And what it does is it simply adds commentary directly to the page and lists over 20 basic errors. So for instance, you see this red rectangle here that announces in this table-- this table that declares how Toronto has the best record in the American League-- it says, error, replace empty TH element with TD.

So that's a classic table coding failure. And so you're basically saying, this table is really well done. Way to go, Major League Baseball, except for this one cell. So it needs to be fixed.

And that's how it does this. So it basically puts these red markers in wherever there's a challenge. So Karl Groves, awesome guy.

Our next group of tools, what they do is they allow you to take your code and submit them for assessment. So whereas Cynthia Says you pointed a URL at something, well, this tool, made in Ontario-- AChecker, I mentioned it before-- has two superpowers because it can also do a Cynthia Says type of analysis. But as well, sometimes your code isn't live yet. And so you can't use a web-based tool.

So with AChecker, you can actually take a chunk of code and just paste it into the site. And it'll give you an analysis of the HTML. And another thing that's cool about it is, with AChecker you can create an account.

So let's say I'm analyzing vw.com. I create an account and log in. And I say, hey AChecker, do your thing. And AChecker says, I've got my reds, I've got my yellows, and I've got my greens. That is, I've got my things I'm certain are wrong and things I need a human to inspect.

The things you're certain are wrong, well, you've got to fix them. But the things that require human inspection, AChecker allows you to go through them and mark the ones that you say, no, that one's fine, no, that one's fine. And it remembers them. So later when you come back to test the same property again, it's not going to bug you about those things again. And that's a cool thing, so yay AChecker.

A very similar tool like that-- another one of my favorites-- HTML_CodeSniffer, of course, also free, otherwise I wouldn't be talking about it. And HTML_CodeSniffer, another tool that allows you to take chunks of code and paste it in. Why we love HTML_CodeSniffer is the gorgeous interface it gives you, as well as it leads you right back to the W3C's WCAG site where not only does it say, here's the problem, but it takes you right to the very technique that can be used to solve the problem so a very pleasing tool.

Another tool, if you're testing in Chrome, that's a no-brainer for us is that you add in the free tool that Google has made available called Accessibility Developer Tools. Now Accessibility Developer Tools, it's kind of like the Firebug for if you're a Firefox user and you have Firebug. Same idea-- you add this to Chrome, and you get a window where you can inspect elements. And it also gives you an accessibility checker that runs alongside, so sweet. Why not? Install it today.

Now all of these tools I've been showing you examine one page. But of course, sometimes you want to examine a whole bunch of pages at once. Now you could spend \$50,000 a year on Compliance Sheriff, or you could use Total Validator Pro.

Now I've got to say, Total Validator is a beautiful product in that it does a one-page analysis for free. But if you want to do a whole bunch of pages at once, you've got to pay a little bit of money. And this is where the half is, because I said 11 and 1/2 free tools. So Total Validator is a great tool, but it's really worth about \$40 you pay to the guy. He's from the UK, invented Total Validator. You pay about \$40 and you get Total Validator Pro.

And the reason you want to do that is it allows you to spider a whole site, even an offline site. Or you can say, hey, Total Validator Pro, I have these 28 pages we've decided are going to be

our representative sample of this site. Here's a list of those pages. Go at them. Or you can say, hey, Total Validator Pro, go down and spider three levels deep. Follow every link and go down and try to find troubles. Now, the feedback you get from Total Validator Pro is pretty nerdy, but it's very good too. So we love this tool as well. I encourage you to give it a try.

The tools from here out, I'm going to show you some tools that actually look at specific issues. That is to say, up to now, I'm showing you stuff that looks at the whole page and tries to find trouble. But for some issues, we need to test one thing. And so a great example of that is testing color contrast.

As those of you who are familiar with WCAG 2 know, there's two different rules to make a site AA compliant that involve contrast ratio. Either we're comparing the ratio of a foreground color to its background. Or sometimes, we're also measuring the color of letters to compare them with other words in that same paragraph.

Either way, there's two tools I'm going to show you. I'm going to show you the Color Contrast Analyzer. And while we love it, it isn't just useful for web pages. It's useful for any document.

So let's say you're about to release a white paper and you say, wait a moment. The blue headline in that white paper looks pretty light. Is that going to be OK? Maybe we'd better check that before we share our PDF with the whole world. So let's do that.

So I'm going to show you how this works. I'm going to call up the WCAG Color Contrast Analyzer. I'm just going to make that appear. Here we go. So I just had to park the screen. So here's the Color Contrast Analyzer. And you're seeing it on Windows, but it works equally well on Mac. The interface just looks slightly different.

And now you can see how I can take any screen, for instance, a PowerPoint screen, or it could be a PDF file or a Word document. And as long as I've got the color I'm interested in analyzing on my screen-- and here I'm just getting that picture of our draft of the white paper up there. OK, there we go. I'm just going to zoom in to make it easier for me to target it.

So I take the Color Contrast Analyzer and I take this eyedropper. And I point it at the blue, and I take a measurement. And then I take a measurement of the blue. I'm sorry. I'm having a technical problem with the way that our classroom software works. It doesn't want to let me take the color.

LILY BOND:

David, do you just want the hex code for that blue?

DAVID BERMAN: That would be sweet. Oh, I got it. OK. So I take the foreground color. I take the background color, which is clearly white, so I'm just going to hit some white.

LILY BOND: The foreground color is actually not the right blue there.

DAVID BERMAN: OK. So what's the foreground color? I'm going to type it in.

LILY BOND: 0095D6.

DAVID BERMAN: D6?

LILY BOND: Yep.

DAVID BERMAN: OK. There we go. Thank you so much. So you see, we can either use the color or we can type in a hex code, which is also helpful if we are color blind like me. So here's the result we get it. It's telling us there's a contrast ratio of 3.31, which fails WCAG AA. It also fails WCAG AAA.

However, because it's a headline, it passes because WCAG 2 has rules for headlines versus body type. So what this tells us is that 0095D6 is thumbs up for WCAG 2 compliance for headlines. But if we use that type in our body, it will be a problem. But we didn't, so we're good. So good on you, Lily. There you go. So that's how the Color Contrast Analyzer works.

Now, the next tool I'm going to use, then, is my other favorite color tester. It's called the WCAG Contrast Checker. Now, the downside is it can only work on websites. But if you're in a website, it's awesome. So this is a plug-in for Firefox. I'm going to show you how it works by again convincing this machinery to bring up the Firefox window as well, again. And we're going to go back to the VW site. I'm just going to reset their home page.

Now, I've installed this tool already. And therefore, when I right-click on my page, you see I have a choice of WCAG Contrast Checker. Again, it's a free tool. Google "WCAG Contrast Checker" and you'll have it plugged into Firefox in two minutes.

So I choose the WCAG Contrast Checker, and what it does is it adds this great panel on the side of the screen. And in a moment, it analyzes the whole page and shows me every color pair there is and gives me the luminosity number. That's the ratio we're looking for. It decides if it's small type or large type, i.e. headline or body type.

And so what it's saying here is, hm, we've got a potential fail here, a fail, a fail, a fail, a fail, a

fail, a fail, a fail, and then a whole bunch of passes. Because what it does is it sorts them from least contrast to the most. And in the case of here, there's 71 instances of this. So if I wanted to, I can open this up and it can just break them down further.

But also sweet is as I focus on each of these cases, it will then show me in the main screen where the problem is and even allow me to play with the colors to see if I can break through and show me how they fare in AA and A. So this is a really powerful tool. We use it a lot. It just only works on web pages. Sweet, huh?

It's actually looking at the code and analyzing it. You don't really need to see the page for this to work. But of course, it's handy to be able to check it out. So two great tools for checking color contrast.

Now, the next tool I'm going to speak to is called PEAT, which stands for the Photosensitive Epilepsy Accessibility Test. And this is for the one rule in WCAG, and it's a level A rule that makes sure that we don't accidentally invoke a seizure in our audience because there are people in our audiences that are at risk for this. And if you understand the science, you can avoid it by just looking at the video.

But if you don't understand the science, PEAT is a free tool developed in the United States of America that you can download and install and it allows you to take any video and run it through. And what happens is we take the video, we run through, and we get an analysis which tells us if there's any parts of that video that require further inspection or removal because they potentially have flashes that are in such a speed and in certain color frequencies that could potentially cause a seizure.

Now, the last type of tool I'm going to show you is a screen reader because the assistive technology we often dwell on the most is screen readers that read things out loud for people who either can't read because they can't see the words or can't read because they don't know how to read or don't have the skill to read. Either way, screen readers read stuff out loud, which is awesome.

However, the Cadillac of screen readers, JAWS, is really expensive. It's about \$1,400. And also, it's really hard to learn. So even if you had the money, we recommend using a free screen reader. And the screen reader that's free that we recommend most on Windows is NVDA. On the Mac, we recommend the voiceover screen reader that's baked into Mac OS or iOS. And on Android, we recommend the screen reader baked into Android.

But NVDA is the best tool if you're going to use one screen reader. NVDA is free. Well, it's donation ware, actually. So maybe you should just give them a bit of money. But NVDA downloads for free. And I'm not going to demonstrate NVDA today, but I am going to point out that not only will it read out loud. But it includes a speech viewer which is really handy. Because listening to it is interesting, but it's often much handier to have a record of what was read out loud.

And so you turn on NVDA's Speech Viewer feature, and it shows you every word it would have read out loud. And then you can look at that and say, hm, is that what I was expecting? Is that what I wanted? David, I'm emailing you this. Why is it doing this?

The other superpower we love at NVDA is that there's a free plug-in for NVDA, because NVDA is a platform. So a lot of people add plug-ins to it. It's open source. And so someone developed something called the Focus Highlight. And what it does is, as you move through NVDA, it shows you focus wherever you are.

So this is something useful for sighted developers for products that don't have visible focus. Because, of course, visible focus is not just something we want to see in our products, but when visible focus isn't present, when you can't see where the cursor is, it makes it hard to test. So often, we just want to turn on visible focus temporarily just to make it easier for us to see where we are. And that's it. So there you have it.

Now there's one other free screen reader I'm going to tell you about. It's called Window-Eyes. Now, Window-Eyes costs money, but if you have a full version of Microsoft Office, it now includes a free license to Window-Eyes.

And the reason I like Window-Eyes is there's only two screen readers on the planet that do a reasonable job of testing Flash applications. One of them is Jaws, which costs a bundle, but Window-Eyes also does it. And so whether you're just using the 60-day free trial of Window-Eyes or you've got a license that came with your Microsoft Office, Window-Eyes is an excellent way of testing for Flash.

Now the last screen reader we're going to play with something called Fangs, and it's not even really a screen reader at all. What it is is a development tool which shows you what a screen reader would have shown. And so just as screen readers announce words and also show links lists and also show the headings, Fangs emulates screen readers for sighted developers.

Fangs is a plug-in for Firefox.

You install Fangs in two minutes, and if you right click on any page, it will show you these three windows for Fangs. And this is the last tool I'm going to demonstrate. So we're going to go back to the VW site in Firefox. I'm going to turn off my contrast checker, go full screen, right click, choose View Fangs. When I say View Fangs, Fangs is showing me three views.

The first view is called the screen reader output. These are the words that a screen reader would have said, and it's searchable, and it's cut and pastable and all that, so this is great. So if we have a screen reader, we'd hear, page has two frames, 43 headings, and 170 links, vw.com, vertical bar, all official home of Volkswagen cars and SUVs dash.

Now, it could be different depending on which browser you're using or which screen reader, but it gives you a pretty good sense of how well that site's going to do. It also gives you a headings list, which all screen readers includes a headings list, so it gives you an overview of the headings, including the heading numbers. And it gives you a links list, which are all the links on the page. And you can see how they're worded. So you can tell the difference between links that are useful like Live Chat, or links that are completely pointless like 11 or 14 or 12. Hey, VW, you've got to fix those. So that's Fangs, my favorite screen reader for people who can see.

And those are all the tools I'm showing you today, but you know, as I said at the get start, we can test our products all we want in the lab, but it's not until we put the car on the road that we discover if it's actually accessible. And that's why, no matter how much testing we do, real world testing is crucial. We've got to make sure that we have people living with particular deficits try our products, and whether that's in one-on-one or in group, that's the best way of making sure. So we do work really hard to make sure our products meet or exceed the standards.

But then we also take people with visual challenges, people with hearing difficulties, people with substantial mobility challenges, whether they have them all the time or temporarily, and we see, how does this work for them? How does it work for them using assistive technologies? How does it work for us using assistive technologies? Because if you've heard me preach before, you know that I can prove to you the majority of us live with a disability, and therefore, the majority of us are affected and the majority of us are candidates for testing.

So that's what I had to share with you today, and I'm eager to hear questions about how to

use these tools or what tools are your favorites, which tool you think you'd use the most. I'm also eager to let you know that there's ways of getting in touch with us. If you want to follow up, I love hearing from you. And as well, we have products that go deep, and we have training courses, and we have whole manuals on this stuff, so if you found this intriguing, imagine how fun I am to spend the day with. Lily, maybe it's time for me to hand it back to you.

LILY BOND: Sure, thank you, David. That was a wonderful presentation full of really valuable resources.

DAVID BERMAN: I'm glad you feel that way.

LILY BOND: Absolutely. As David said, I think we're ready for Q&A. There are a lot of great questions coming in. While we're compiling them, please feel free to continue to ask questions. I just wanted to let people know about some of our upcoming webinars. Later this month, we have webinars on tips for creating accessible online courses and closed captioning standards and best practices, and you can register for our webinars on our website at 3playmedia.com/webinars.

So David, I'm going to start out with a question for you. If you could use only one tool, which would it be?

DAVID BERMAN: Are you saying if was allowed to only use one testing tool on the whole planet to test websites?

LILY BOND: Yeah, what's your favorite tool and why?

DAVID BERMAN: Well, it depends. And when we started out, I said, you have to start with a strategic question. You have to ask yourself, why are you doing this?

And so it would depend whether we're testing a website or a document, whether we're testing for Level A compliance or Double A, whether we're doing regulatory testing, or whether our goal is simply to help a certain audience. Because when we develop a website, we start off with strategy, and we identify who our audiences are. And by knowing who our audiences are, we know if it's going to be conspicuous audiences that would trend towards having specific deficits.

So for example, let's say we had a website that was all about issues, particularly of interest to elderly people. Then we'd be even more concerned than we usually are about making sure the type is large enough to read, or other issues that occur more often in an elderly population.

So we have to start out with what our goal is before we decide which tool we would use if none else. The answer's going to be different in every situation, and in fact, if our goal is full compliance, I'd say if you only allow me one tool, I'm not even going to try. Because if our goal is to prove that a product meets a certain level of compliance, then unless we're planning on testing for all of those success criteria and making sure we meet or exceed each one, then there's little point in just doing part of the work. We have to commit to doing it all, and that's going to require a basket of tools.

LILY BOND: That makes sense. Thank you.

DAVID BERMAN: Just a second. I weened out of the question. I love the WAVE toolbar. Yeah, I think the WAVE toolbar. I mean, if I'm allowed to view source code, I'm allowed to play with it, I'd probably take the WAVE toolbar as my one tool.

LILY BOND: That looked really cool. So another question here-- would it be possible to address testing within learning management systems, for example, Blackboard? Our experience is that web page accessibility testing tools cannot also correctly check content within the LMS.

DAVID BERMAN: Yeah, so this is a big topic. In fact, we have a course on nothing but accessibility for distance learning and for meetings like this one. And part of the challenge is that we don't just have to worry about the experience of the audience member being accessible. We also have to make sure that presenters living with disabilities can also present. And then we also have to concern ourselves with people who are dealing with the LMS who are administering the whole system also being able to interact with the system.

So it's a big question. And in fact, we did a study of all of the major systems, and we couldn't find one of them that would be compliant with level Double A. The Canadian government, who's quite committed to accessibility, perhaps as much as any government in the world, had us go deep with this. And we found the only way to create a completely accessible experience in LMS was to actually put a basket of elements together. There wasn't one platform which you could do it all on at once yet.

So the person asking this question, I'd love to give a fuller answer, because it's a 20-minute answer. I'd like to give it maybe-- contact me offline, and we can talk about this, because it's a fascinating topic, and we'd love to help you figure out the best way forward.

LILY BOND: Thanks, David. Someone else is asking, is it possible to test for closed captioning on web videos?

DAVID BERMAN: Yes, certainly. One of the simplest ways is just to identify whether a caption file is present, because ultimately on the server, let's say your captioning is all done in SRT files, which is the most common file type used to contain the markup language, which contains the captions. One test we could do is just say, hm, are the SRT files present in the same folders as the video? Because if they're not there, we can be pretty sure there's no captioning.

Now it's possible people have burned the captioning into the video, like movies from years ago, and therefore, captioning is present. But in most cases, we do machine captioning where the captions are independent of the video file, and therefore, simply checking for the presence of the captioning file tells us where the captions are missing.

So a part of the Compliance Sheriff can be programmed to scan your whole site and look for those missing pairs, and then send you an email saying, hey, there's a video posted in sector 17 that has no SRT file. Now even if the captions are there, though, it's another matter to make sure the captions are all high quality, and that requires a human eye-- or you smartly order all your captioning from 3Play. Shameless plug for 3Play. They're probably the best captioning house in the continent. We used to do our own captioning, but we found we got better captioning for far less money if we just used 3Play's process. So we know when we get captioning from 3Play, they're always excellent.

LILY BOND: Thank you, David.

DAVID BERMAN: You're welcome.

LILY BOND: Another question here is asking, should a site pass 100% of the criteria, or what is an accessible measure of success?

DAVID BERMAN: Well, there's no software on the planet that's bug free. And so our ideal is to have perfect accessibility on every platform at every bandwidth with every operating system on every platform, every device, at all times. And we'll never reach that. And no one expects you to reach that. And I don't want you to be intimidated by that.

But to declare a site is being compliant with WCAG 2, the WCAG Evaluation Methodology, the WCAG-EM, allows us to take a statistically relevant sample of pages. Now, for a very small site, let's say you have a six page site. Well, you're going to test every page and test it

thoroughly. But if you add a site that had 3,000 pages, or a document that has 7,000 pages, it's not realistic to test every one.

So instead, the methodology identifies a way to identify different page types, as well as a randomization technique to also get a percentage of random pages. And then we test each one of those pages for all of the properties. And only then do we declare that it's compliant. And in that compliance report, we declare our methodology.

LILY BOND: Great. Thank you, David.

DAVID BERMAN: And also, and it's important to know this, because I know I'm also looking at the questions that have been posted. I've just got an eye on that. It's perfectly OK to have accessible alternatives. So for instance, let's say you had a PDF file that isn't accessible enough to comply.

As long as that same content is available somewhere else that is, like let's say you had the same content in HTML on a web page, it's perfectly OK to say in that PDF file, we're sorry this PDF file isn't accessible. But this very same content is available in a completely accessible way on this web page way over here. So you don't have to make the content accessible in every place it's presented, though, of course, that would be nice. You just have to make sure you point people to at least one place it is.

It's even reasonable in some cases to be able to-- maybe the experience of a certain website on an iOS device isn't fully accessible, but that same website is fully accessible on a Windows desktop. Then you could have a message that warns iOS users, we're sorry, but for a fully accessible experience of this content, please go to a desktop and view it there. Not the best solution, but definitely a compliant solution.

LILY BOND: That makes sense. I think we have time for one or two more questions.

DAVID BERMAN: I can stick around as long as you'd like, too.

LILY BOND: OK, great. Someone else is asking, can these tools test pages that are behind a password?

DAVID BERMAN: That's a great question. So some of them can and some of them can't. So for instance, Total Validator Pro, just to compare. We used Cynthia Says, and I used that on the VW site. But if it was password protected, there's no way of telling Cynthia Says, go beyond the login screen.

But Total Validator Pro does have a feature where you actually load it up with the credentials so it can get to the next layer, so it can get past the login screen, past one layer of validation. We even have some tricks where sometimes we can get through two layers of validation. However, otherwise, we sometimes have to go to the more sophisticated tools, the paid tools, in order to programmatically move through an entire site and get behind various security walls.

LILY BOND: Great. Someone else is asking, at which phases of a website development cycle would you include accessibility testing?

DAVID BERMAN: We believe the best practice is to be thinking about accessibility at every step in the process. So even when we're wire framing, were already thinking about how we're going to build accessibility in every step of the way, or while we're choosing a CMS, or saying, hm, accessibility of that CMS gets a vote.

However, once we're in the development process, the best way is every step of the way. Some things are editorial issues. So we want the writers writing alternative text. Even as they designate a photo in a Word document that's eventually going to be web content, we're saying, write that alternative text now. And then when this document goes to translation, the Spanish translation will include the Spanish translations of the alternative text. So we want to build that in.

Or if we're making sure that no instructions include instructions that assume a person can perceive color, that's something our writers need to learn to do. We don't want to catch it at the 11th hour and have to go back. But then there's other aspects that the graphic designers need to inject, making sure the color contrasts work. And there's other things that only developers can do.

So every step of the way, different people have different jobs to do. So the worst thing to do is to wait till your product's almost done and then test for accessibility because you can't come up with a more expensive, more painful, more time consuming and least effective way of developing a product than that. Instead, we want to inject into our entire process testing for accessibility, or mediating designing for all. When we do that, we actually get the cost savings, and we get the audience reach, and we get the better search engine optimization, and we get better sites, and we get everyone included. Why wouldn't we want that? So that's the best way.

However, the tools I'm showing, whether you're using them unfortunately at the very end for

the first time or whether you're using them in a continuous process where everyone's made a part of the workflow, these are good tools. And I didn't get into document testing, for instance, like the Accessibility Checker for Word. Or I didn't get into the best tools for PDF like the Pack, which is the tester for PDFUA.

But you can't show me a document type, whether it's PowerPoint, or whether it's Captivate or InDesign, Excel. We've got a methodology for every document container that you come up with. And for every one, our answer's the same. We got to build in accessibility every step of the way. That's the best way to do it.

LILY BOND: Thank you. Another question here is, what do you recommend for total novices? For example, a faculty member who has made their own WordPress site but otherwise doesn't know much about HTML?

DAVID BERMAN: Fair enough. So what I'd say is let's deploy to that faculty member a CMS template. Let's say it's WordPress. Let's create our theme, which is already vetted. It's AA. It's all there. And then that faculty member, out of the box, gets a product that's hard to break. And then we give them guidance so they know that even when they're writing words or paragraphs or they're making decisions on how to give priority and insert headings, it vastly increases the chance that the product will remain accessible.

Whereas if we just say, faculty member, here's WordPress. Knock yourself out. Go crazy. Choose the prettiest theme you want. We're undoubtedly going to get a site that's not accessible, especially if the faculty member knows squat about accessible programming. But if we've created the perfect theme for University of Post Office Box 2000 has a theme that all faculty members use, then we get them in the right place. And then we just have to give them guidance that as they populate that theme, these are the things they need to keep in mind to reduce the chance that they're going to break the accessibility of the product. And then we can just do a spot check and we're good.

LILY BOND: Great. Thank you, David. I think there are a few other questions, but they might make more sense to reach out separately for. And I think we're just over 3 o'clock. So I think it's a great time to stop. But David, thank you so much for your incredibly valuable resources. This was just a great presentation and people really enjoyed it.

DAVID BERMAN: Well, Lily, thank you very much. And I want to thank everyone for showing up early. And I want to thank everyone on the 3Play team and my team here, Tamara, who produces, and Ben and

Steven and David and everyone up here in Ottawa, all these people you never meet who are behind the scenes making sure that all this content is true and that it's presented well.

But especially, thank you all for embracing the importance. This is the decade of inclusion. And it's going to be thousands of little things we do rather than just a few big things we're going to do to get to the place where we have a society that truly knows how to not leave anyone behind.

LILY BOND: That's a great way to put it, David. Well, thank you again for joining us. And thank you to everyone who attended. I hope you enjoyed the presentation. We will send out an email tomorrow with a link to view the recording. And I hope that everyone has a great rest of the day.

DAVID BERMAN: Thank you, Lily.