HIDDE DE VRIES: This is "Amplifying Your Accessibility with Better Authoring Tools." And my name is Hidde de Vries, and I am an accessibility specialist at the Web Accessibility Initiative, or short, WAI. Now the Web Accessibility Initiative is a part of the W3C, and it has as its aim to make sure that the web is accessible for everyone, including people with disabilities. And we're trying to make that happen with a number of different web standards as well as a lot of information about how to make the web more accessible.

We have a number of different web standards. WCAG, the best known one I would say, is adopted by governments worldwide, and it is for making web content more accessible. And then there is ATAG, which is about authoring tools, UAAG, which is about user agents like browsers, for example, and other tools that you can use to access the web.

And then there is ARIA, which might be very familiar to any front end developers that are watching. It's basically a superset of HTML that allows you to set accessibility metadata on your content. And besides these different standards, which are kind of the meat of what WAI is doing, there is also the WAI website, w3.org/wai. And I want to mention it in particular-- because I haven't been with WAI for that long-- I've been around for about half a year now, and I continue to be surprised by how much amazing information is on this website.

So this might sound a bit like an advertisement-- don't really mean it as such. I just want to make sure that people who aren't aware of it like I wasn't before, that there is a lot of information not just for developers and designers about how to make accessible websites, but also for people who plan accessibility, people who do advocacy, training, all these sorts of different things. So YEAH, I want to make sure people know about this website at w3.org/wai.

So now I want to take you back a bit to 2012, because this was when the Olympics were taking place in the United Kingdom, and the opening ceremony was taking place in London. Now they had this huge stadium and invited a number of different people that were important to the history of the UK, and one of them was Sir Tim Berners-Lee, who invented the world wide web. What he did there is he made these four words appear-- "this is for everyone." And these words also appeared as a tweet on Twitter, and they made a lot of people discuss, what does this mean?

And I think there's three different meanings that you could assign to this and that you could apply this to. So one is that everyone can get onto the web. I mean, given you have a cable subscription or you have some kind of data connectivity, you're able to get on the web.

So you don't need to apply for a passport in order to get onto the web. You don't need to get permission from anyone. You can actually just access web pages.

The second bit is that everyone can have a website. So there's no kind of vetting committee that will check what kind of content you want to put online. Anyone with server space and maybe a domain name-- but even that you don't really need these days anymore-- can create a website and be online and, at that point, talk about anything they find important, whether it is their profession, or they want to sell certain services, or food, or whatever, or they maybe want to talk about some of their hobbies or their favorite animals.

All of this you can just do on the web. So it is also for everyone in that sense. Everyone can put stuff on the web. Everyone can get onto it. And in the last part-- I think probably the most important part, especially for this talk, is that the web is also a medium that really, really tries to work for people with disabilities. So it includes people with disabilities as the audience.

Now the web has changed quite a bit since the early days. It's always going to be for everyone, but it has changed quite a bit. So in the very early days, it was very much a platform for scientists to communicate with each other over a really long distance. And scientists like to do that with documents-- long documents, maybe nuanced documents, lots of text. And maybe those texts didn't look so attractive. They could just be like long blocks of text like I'm showing here.

But a lot has changed since then. The world has changed in the sense that we can do a whole lot more with our content on the web. Layout has changed quite a bit, so there are lots of different new possibilities in CSS like flexbox, and grid layout, and other things. So the web has gotten a bit more interesting in that sense, or we have the tools, at least, to make the web more interesting, because we can do all sorts of layouts now, and it's built into browsers.

There are also new ways to format content. And with that, I'm not just talking about CSS, but also about different systems where you can just drag and drop, and can create web pages without any knowledge of HTML or CSS. Components are everywhere, so they've really changed the way we think about the web as well.

Where previously, especially in the time of scientists, when you think about different documents that were published to the web, now people, more and more, start to think about their work in terms of components. This is something that was already common in terms of maybe backend code, but it's really found its way also in how content gets developed and designed. So components are really big now, and I think that's also something that really changes how we create content for the web.

If we talk about creating content for the web, we really ought to talk about authoring tools. The most common authoring tool is a content management system. So you can think of content management systems, CMSes, as authoring tools, but there is a whole bunch more. So let's look at some of those. WYSIWYG editors are a prime example of authoring tools-- provide an interface to create HTML, to create content for the web.

Course creators are a prime example. So anyone who works in an education or has enjoyed education, recently anyway, they will have come across learning management systems, so systems online that allow students to do their things online-- e-learning software. This is obviously happening now a lot that people move online with courses and stuff. But it's been happening for the last many, many years.

I already mentioned content management systems, or CMSes. They're also a prime example of tools that create web content. Wikipedia is another one. So Wikipedia is a platform where you can create new pages, but you can also update existing ones. So it's a perfect example of an authoring tool.

And that also includes, by the way, Wikis that people maybe have internally at large companies, like documentation systems and that kind of stuff. Save as HTML is a feature many people don't use as often these days, but in many word processors, you can actually save a document and save it as HTML. Again, you're creating web content at that point.

Now I wonder if anyone who's watching this has ever tweeted-- you probably have. And at that point, you've also created web content. You've created new content when you've created a new tweet.

Social media platforms, like this popular microblogging platform-- they're also authoring tools. Form generators are big ones, so that's something you find a lot at governments, for example, or other organizations that need to create a lot of forms, or have people sign up, or have people input a lot of information. So form generators are a big authoring tool as well

And then there are site builders which have become fairly popular, especially when there's less WYSIWYG or less people who want to maybe learn HTML themselves. There's these tools where you can drag and drop, and create web pages that way. So in summary, authoring tools are tools that create web content.

That's the summary of it. And the web content is basically HTML if you think about it from a technical point of view. So anything that can generate HTML is, in a sense, web content. And with that, it has an opportunity to improve a lot of accessibility at once,

Because that's something we focus a lot on. We try to make web content more accessible-- individual bits of web content. But what if we can focus on the tools that create that web content? At that point, we could improve a lot of accessibility at once, and that's kind of the main thesis of this talk. We can do a lot of accessibility by improving the tools that make the web content.

Now this makes sense when we look at the different accessibility standards and see what the role of each of them is in trying to realize this "this is for everyone" kind of thinking. So WCAG is a standard for web content, as mentioned before. Of course, when you want an accessible web that works for everyone, you need to make sure that the content on it is accessible, so it's a good point to start.

But what's also important is that the tools that are used to view the content-- browsers, user agents-- are also accessible. And that's what UAAG exist for, the User Agent Accessibility Guidelines. So that's generally browsers, but it's also other tools like Siri that lets you interact with the web-- that those tools have accessibility in mind, because if the content is accessible, but people with disabilities cannot get to the content because the browsers aren't, then the thing doesn't work. So for the web to be accessible, both of these things need to be accessible.

And then the third component is ATAG. It stands for the Authoring Tools Accessibility Guidelines. And again, if we want the full picture, the whole web to be accessible, it isn't just about web content, it isn't just about browsers, but it is also about those tools that create web content, so authoring tools.

So these three things, they're all needed for an accessible web, for a web that truly works for everyone. Authoring tools are important. That is the main thesis of this talk, because if they work for everyone, if they have accessibility in mind, they can let us include more content creators, so more people will be able to create web content.

And secondly, they can deal with content problems before a website goes live. Because they are the place where content is created, they can also be a place where content is vetted. So many content management systems and other authoring tools might have staging environments and that kind of thing.

So if employee 1 creates a page, then employee 2 can have a look at it before it goes live. But instead of employee 2, or maybe in addition to, maybe some kind of accessibility mechanism can also have a look. So authoring tools provide an opportunity to improve accessibility, and vet and check for accessibility before content actually makes it to the web. And we'll get into a bit more detail about that later on.

So again, the standard for authoring tools that we created is called ATAG 2.0. That's the latest iteration of it, so Authoring Tool Accessibility Guidelines. It consists of two parts. Now I'll show you briefly what these parts are about, and after that, we'll get into the details of what could be improved in these parts and according to these recommendations.

Now Part A is about the editing experience. And the editing experience is basically where the content gets created, so the CMS admin interface, or the place where someone creates a course in a course creator, or the new tweet screen on a social media platform. All of these places are editing experiences, and they are all covered in Part A of ATAG.

Now Part B of ATAG is about the output, so basically about the HTML that comes out of these systems. And it's about making the HTML there better.

So we have the editing experience and then the output. So these are two different parts, and we'll talk about these in turns. So at first, let's talk about the accessibility of the editing experience. What I want to do is talk about a number of different people that all could use better authoring tools, basically.

So let's start with Bob. He is a doctor, and he is colorblind. Now he's just seen a patient, and he wants to set up a recurring prescription for that patient. And the practice where Bob works, they've just set up this new web-based system that allows him to do the prescriptions online. It's really useful for him, but also for his patients, because they can then access this web thing, and they can see their prescription and take it to the pharmacy. And it's perfect, as you can imagine.

The thing is Bobby is colorblind. So he has this disability, so the tool really needs to also work for him, because otherwise, he might not be able to create the prescription. So a number of improvements need to be made about color in order for Bob to use this software effectively.

Or what to think of Alice, who is a human resources specialist. And she recently went on a skiing holiday and broke her arm. But she just went back to work, because why not? She got bored at home, and she learned to use the Tab key on her keyboard so she can do most stuff with one arm.

Anyway, that's Alice, and she wants to publish a story about her colleague's promotion, because that's what they do at this company. They really like to celebrate success, and the company internet is where they do that, so she goes on to enter that story. Again, it needs to work for Alice, so we need to make sure that she can actually use the interface with just a keyboard, because she only has one hand available, and that's what she can use to use the interface.

Or what to think of Angela, who is 32 and a teacher, and she has reduced dexterity. What she wants to do is to upload a take-home exam for her students, because it's almost end of term, and they need to do their exams in order to finish the school year. So she's going to upload that, but, again, we need our software to work for her. Otherwise, she cannot do this basic task.

And then we end with Anne who's 21, a typical millennial, she works as an influencer, so she does things like producing vlogs and stuff. And she's motor impaired, and she just wants to publish her latest vlog on social media. So again, the social media tool needs to work for her in order to do this basic task in this part of her daily life.

All of these different people are trying to perform tasks that are part of their daily lives, an integral part of their daily lives. They're normal people that just want to get normal things done. And that's a very common theme in authoring tools, then. The kind of those that are used, they need to work for everyone.

So let's look at a couple of examples. One example is to test with text zoom. It's something you would do on a regular website as well, but it makes sense to do it in an authoring tool, too. You want to make sure that if you zoom in the browser, if it's a browser-based tool, that that just works.

So if someone goes up to 200%, 300%, maybe 400%, the interface still makes sense to them. And this is something that gets a little bit more common these days, because a lot of CMSes go mobile, and they find they have to reject kind of their interface and make it work for smaller screens. The same effort really works for people that zoom in their interface quite often, because you're making sure of the same kind of affordances. You need to make sure buttons and stuff still make sense with regards to each other.

Or what to think of text alternatives for icons? A lot of content management systems, they come with a lot of icons these days without any text labels. That's become a popular interface design thing.

This is a screenshot of a CMS. I don't remember exactly which one, and I also don't know what the icons are for. So there is a star, and there is a thunder thingy, a question mark. That's probably for help, somehow. There's an admin thingy, and I think when I click it, it's going to expand some expandable menu.

Now one thing that's important to get right when you're building this kind of authoring tool or you're involved with this kind of authoring tool is to make sure that the icons have alternatives so that people who cannot see them still have buttons that make sense to them. Or maybe just have visual alternatives so that everyone can make sense of them, including me, who has just forgotten what these icons are actually for.

Make it work with keyboard is another thing that I've seen in a lot of authoring tools that I looked at for my projects. And a lot of these tools don't actually work with keyboards. In general, one thing it means is that you want to have visible focus. So when someone uses a Tab key to browse through the interface, you want to make sure they can see where they are.

Previews are a big thing in authoring tools, so you want to make sure that previews are displayed accessibly. And this is a bit different for tools that exist on the web, because they are usually in a browser. So the preview will usually be a web page, so the accessibility of that will probably be the same as the accessibility of the actual product that you're shipping. But yeah, you want to make sure the preview is accessible so that someone who has a disability is able to use the preview just as well as someone without a disability.

Help with spelling is important, too. Maybe for users who have dyslexia, it makes it easier for them to create effective web content, because you can make sure maybe their spelling mistakes are caught before the content goes live. These are a couple of things that you can do to make the accessibility better of the editing experience, so for people who are trying to edit web content.

That's the A part of ATAG, and I want to now go to the B part of ATAG. And the B part of ATAG is not about the editing experience, but about the output, so the stuff that comes out of this authoring tool, the HTML that is produced by it. And there is a number of ways that we could improve things there, but let's first, like with the A part, look at a number of different people that would benefit from good outputs.

Maybe that's John, a student who has low vision, and he just wants to access the results for his mathematics test. Or let's look at Vivian, who is 26, a privacy lawyer. She's blind, and she has to organize a business lunch with some of her associates, and she wants to figure out if the restaurant that they're going to go to serves vegan food. So again, the restaurant needs to make sure that the system that creates their menu online creates accessible menus.

Rick, who is 67, he's an investor in real estate, and he has severe hearing loss. Now he wants to watch an interview with the prime minister on some news website, because the prime minister's talking about that year's budget. And one of the important talking points is the real estate market, so it's very important for him to figure this out.

But he can only do so if there are captions, because he has that severe hearing loss. So we want to make sure that there are captions, and that means the news organization that publishes this interview needs to use a system that is actually able to include captions for all of this to work.

Erica, who is 31 and a photographer with repetitive stress injury, wants to file a tax return online. Now this seems like a basic task. It's not really in most countries. But they do improve it, but it's one of the harder things to do-- one of the harder interactions to do in a government, especially if you have RSI, because there might be a lot of buttons and a lot of forms to go through. So again, the system that the tax office uses to produce these forms-- maybe it's a form generator-- that also needs to take accessibility into account so that Erica, the photographer, is able to use this thing.

The common theme you might find from this is that good authoring tools can actually help content editors to comply with accessibility requirements. So I quoted a couple of different people and explained what they are trying to do, and all of them had accessibility requirements. Now a good authoring tool is able to help the content editors, or the people who create the content that these people need to use, help them comply with the requirements and make it easier for them.

Now let's look at a couple of things that a content management system or an authoring tool in general could do to make the world better in terms of accessibility. Now one thing they can do is allow for adding alternative text to images. And if you're an accessibility expert, or you know anything about accessibility, you probably know this-- an image needs to have an alt attribute-- if it has any meaning, it needs to have the content that is in the image.

So in this case, there is an image that says, "35% off," because it's using a font that is really hard to use on the web, and it really needs to be an image for some reason. So the contents of that image needs to go into the alt attribute. Now that's common knowledge. I probably don't need to tell any of you about that.

But what isn't so common is that authoring tools actually allow for entering this information. So in CMSes, it's quite common that there is some way to add alternative text to images. In some WYSIWYG editors, it could be a little harder to get it on. In some social media, it used to be impossible until very recently.

And you might think it's not really required on social media, because we only talk about nonsense there, but that's not really true. There's politicians who lay out their campaigns there as they post images of text or campaign images or those kind of things. So yeah, alternative text is important also on social media. And it wasn't as common there. It is becoming a lot more common now.

So that is cool. It might sound like a trivial thing, but especially until recently, it wasn't for many platforms. And there are still authoring tools that don't offer this capability. But beyond the basic functionality of adding alternative text, there's more stuff that an authoring tool could make a difference in.

What if maybe [INAUDIBLE] would complain if you haven't added an alternative text, or you'd have to explicitly say, no, no, this is just a photo of balloons. It's decoration. You don't need to put any. Maybe it should force users to make that conscious decision there.

Or maybe if you're creating a link, it could be proactive and say, you've created the link. The only content of that link is an image. The image doesn't have alternative text, so you need to fill that in. Otherwise, we will not save the page, or we'll throw a very serious error. We'll tell your manager, whatever.

There would be lots of ways where an authoring tool could really help the user to get this right, because for those who don't know, what technically happens when you put an image inside a link, and that's the only content, then the alternative text of that image will become the link text. So someone who uses assistive technology-- maybe a screenreader, for example-- and gets to that link will hear the alternative text announced as the name of that link.

Now if that's empty, there is no way to know what the link is for. So someone can hear it as a link, but they won't know what the link is going to send them to. So it's very unhelpful.

Now a content management system could really help with this. It would prevent the page from even being saved, or it could be [? a little kinder ?] or whatever. There would be lots of different ways where a content management system could actually recognize this and tell the user what to do. But I believe most content editors wouldn't have any harm in mind when they create this accessibility problem of no alternative text in a link. But if they know, they are in a position to fix, so that's something where we can really help.

Generating accessible markup is another thing where content management systems or other authoring tools can really make a big difference. So if you have a tool that can generate lists-- maybe if someone uses an asterisk, you can create a list [? space ?] out of GIFs with bullets and everything. Or you could create the semantic tags-- unordered list tags and list items tags in order to make this work.

That's a choice other content management systems or another authoring tool can make, and it can make a big difference in, because when it creates accessible markup by default, it can prevent a lot of inaccessibility.

Another example of where authoring tools can help is color contrast issues. So imagine I've created this web page with a header, and it has a photo of my cat, and it's a very dark photo. But on top of it, there is some text, and it's a very light text. So on the white text on a very dark background, there is no color contrast issue here. It's good. But what inevitably will happen, especially if I put this on some kind of marketing website, someone will want to change that header image.

Maybe there is some seasonal thing going on, or maybe they just want to have fresh imagery, or it's a new page with the same header components on it, whatever. Maybe someone replaces this image with a photo of Korean food. And the photo is very light-- the text is also still very light.

Now this has created an inaccessibility, because there is now a color contrast issue. I now have color contrast problems. What my tool could do is tell me about that, because I might be doing this by accident. Maybe the place where I edit this text isn't where the photo is.

So maybe in the tool, it's in two different places, so I don't notice until I go to look at the actual website. There's lots of reasons why this could go wrong. A tool could help. It would say, well, this is not enough contrast. You've just reduced the contrast and made it illegible.

And it could also suggest solutions like, maybe put a background color underneath the text, or make the text larger, or use a bolder font, or all of the above. It could even offer these as options and say, let's do this automatically. Imagine the possibilities.

Something else authoring tools could help with is spell checking. I mentioned it before on the side of editors, but it can also be very helpful for the end user. What if I was an editor and I created a Submit button, but I had accidentally used a W instead of a U in the word "submit." so now it says S-W-B-mit.

Now a user is trying to use this with speech recognition, and they say, click Submit button. And the tool goes, there is no Submit button. This will be a confusing experience, which would have been solved if there was a spell checker, which would have told me, hey, that doesn't say "submit." It says S-W-mit. So again, it would help. If there was a spell checker, maybe I wouldn't have shipped this button to the website.

Readability levels is a thing where authoring tools can make quite a big difference, because readability is actually something that can, at least for a part, be verified automatically. You can verify automatically if sentences are too long, if there's too many adjectives, if there is lots of commas, all those sorts of things. Even words that are very unusual, jargon-- it can all be detected automatically, and you can decide how far you want to go with this.

But if a content management system or another authoring tool would give warnings or some kind of readability score, it could improve the accessibility or the readability of this page before it even goes live. And I think that's very valuable. This is actually something that exists in some content management systems or in some plugins for the purposes of search engine optimization. So some tools have this built in just to make sure you can score better on your favorite search engine. So it does exist in some places, but it will also have accessibility benefits.

Examples are another thing that ATAG quotes as a place where you can really make things better for the end user, because what happens a lot is if developers implement a content management system that they are using examples from the content management vendor, and they might copy-paste those. Now if your examples really make a point of being super accessible, and they get copy-pasted, then that accessibility also gets copy-pasted.

Not always. It's still tricky, because people might not understand what exactly the accessibility properties are and kind of remove them or get it wrong. But still, you have a better chance if you have some semantic markup in your examples there and if you have a whole bunch of divs, and especially if you also help explain what everything is for. So documentation is really helpful as well in trying to create products that put accessibility first, because then the people who work at the authoring tool vendor can spend some time getting the accessibility right, and really pass on that knowledge to people who are just reading the documentation. So documentation is a good place to recommend more accessibility and encourage more accessibility.

That's what the right authoring tools can do-- they can encourage more accessible content in all sorts of different ways. And I think this underlies this whole talk. I think it's very important that we look at it in terms of encouraging more accessible content, because that departs from the point of view that nobody wants to make things inaccessible on purpose.

But it's on the contrary. They would love to make things more accessible. They don't know how, and you're helping them at a really useful point of the development flow, namely, the part where the content gets created and the part where there is still opportunity to make improvements to the content. So if you get your authoring tools right, if you configure them right, if they are built right-- and that's something only the vendors can do-- but also if they get implemented right, [? it's ?] the side of the client. There's lots of parts where you can make improvements that help improve these tools.

The mission of WAI, as I said before, is to try and make the web more accessible with lots of different standards and guidance and all sorts of things. Authoring tools have always been a part of that. We've always found that an important part of making the web accessible. As I showed before, it's not just about the content. It's also about the tools that look at the content, and also about the tools that create the content, the authoring tools.

Now one thing that we did to help with that is the document called "Implementing ATAG 2.0." So this is a document that was created at the time of ATAG 2.0 at the standards, and it explains the standard. So it's a guide to understanding and implementing ATAG.

Inside of it, you will find a rationale and intent. So some information on why certain criteria exists, what the goal is with them, and why we think they help. And also practical examples, so how you could actually use them in an authoring tool.

Now in addition to that, at W3C WAI, we've started doing more work around authoring tools. And I want to briefly talk to you about that and explain the work we are doing there. Now all of this work is part of WAI-Guide, which is a project sponsored by the European commission's Project Horizon 2020, where they basically sponsor projects that are useful for European citizens. In this case, it's useful for people even worldwide.

It's a three-year project, and it is embedded, as the proposal says, in W3C's vendor neutral environment, which means that the way we approach things in this project is vendor neutral. So we don't try and improve altering tools of a specific vendor. We don't try to make a specific learning management system better. We try to make the whole landscape better, and give people tools to understand better how they can do accessibility in their own systems. This project, WAI-Guide, has a lot of other things within it, but the part that I'm mostly involved with is called Accelerate Support for Accessible Authoring.

Now one thing that I am working on and that I started working on as one of the first things in this project is a list of authoring tools. That's something that we hear a lot. So there's lots of organizations out there that want to prioritize accessibility, and they find it very important.

But usually, when they're convinced of that, their next question will be, OK, which CMS should I use, or which e-learning software is the best one? And we're vendor neutral, as I just explained, so we cannot say, oh, use this one or use that one.

But what we can do is try and create a list that has objective comparisons between all these different tools, and that's kind of what we're trying to do here. So it's a list of authoring tools that you can filter by criteria that exist inside the ATAG standard. You can find tools that support accessibility, and you can submit your own tool because that's kind of how we want to manage things in this.

It's not a list of tools that we've checked ourselves, but it's a tool that lists authoring tools that have proclaimed themselves how accessible they are. So they've created a report of their own accessibility, and we just list it. And we hope that that will be useful as a way for people to compare different authoring tools, and get a better idea of what the tools are that put accessibility first.

And there's lots of ways to try and compare that, but the way that we've chosen is the ATAG standard, because that has to be made with a lot of different stakeholders taken into account. So we think that that is probably the best way to compare things. Oh, I should say that this list of authoring tools has only dummy data at the moment, so it is very much in development. We've kind of created the framework for it, but we're now working on getting data for it, so getting actual authoring tools in there and trying to get that data model to work with this tool.

Now the second thing we work on is a page where we explain how to pick authoring tools, so what kind of things to look at when you're choosing a new learning management system, or a new CMS, or anything like that. And this page will kind of have some details about ATAG as well. And then the last thing that we're working on and that are currently very busy with is the ATAG Report Tool, which will take people through the ATAG standard, which has a number of different success criteria, just like WCAG does.

And then for each of them, you can say whether your tool meets them or not. [INAUDIBLE] a number of observations as well. So maybe for one thing, you will say, yes, it works with keyboard. However, the side bar doesn't, or that kind of thing. You can really create a reform that gives some detailed information about how accessible your tool is. And you can then export it into JSON, which can then be imported again, that sort of thing.

So this is really meant to do self-reporting on accessibility, specifically of authoring tools, because as I've shown in most of this presentation, authoring tool accessibility is really important. So we want to give people a way to report on their authoring tool accessibility.

So coming to the conclusion, I want to give you a brief overview of what I think are the main takeaways of this presentation. And one is that content creators can really benefit. So we can really have more people make stuff for the web. If we want to say the web is for everyone, that means everyone should be able to make stuff for it. And that means we need to make sure that our content tools work for people with disabilities.

Also, end users benefit, obviously. So if the tools have good output, and if the output of these tools is both good by default, but also, if there are a lot of encouragements for better accessibility, like things that shout if there is not enough color contrast, then end users also benefit, because the things that are created by these tools will be more accessible themselves.

And lastly, authoring tool accessibility is important across different sectors that really intersect with people's daily life, like education, and health care, and social media. It's maybe less-- it's less about what maybe traditionally we think about, if we think about the web. It is a whole bunch of different systems that are used by a whole bunch of different people as part of their work, as part of their life, as maybe a consumer, or a tax payer, or whatever. And in all of these situations, it really matters a lot that authoring tools are good.

Now that's all I have to say about the subject today. What I want to leave you with-- two things. One, the slides are online, so you can see them there. Everything that I've linked to in these slides, you can find there. For the Q&A, I want to hear from you how you would improve accessibility with authoring tools.

So what kind of things would you build in, and, what kind of things can be made better inside authoring tools, So in the editing experience. So let's talk about that. Thanks very much for watching.