ACCESS ABILITY
TRANSCRIPT

It's in the Design: Fundamentals of Web Accessibility

DEREK FEATHERSTONE, Web Accessibility Expert

VIDEO:

"I'm sorry sir but I have to hang up the phone now."

"What do you mean? I need some help with this one thing."

"I'm sorry sir, there's a security and privacy breach. I have to hang up the phone now."

(DIAL TONE)

DEREK FEATHERSTONE: This was a scenario that we had three years ago when we were doing testing with people with disabilities in a financial institution. We were trying to test the online banking and the person that we were testing with was phoning the help line for that institution because he couldn't figure out how to get set up on the online banking. So the user has his access number in a text file on his computer and uses a screen reader to access it. So when the person on the other end of the phone, on the customer service line, says I need to know your access card number so that I can look, he fires up his screen reader to read out the number. His screen reader starts speaking the number and the person on the other end says, "is there someone else in the room with you?" "No, no, that's just my screen reader." "Sir, I'm sorry I have to hang up and end this call because there's somebody else with you. I can't be doing this with you if there is someone else there I have to know

that it's just you. Sorry sir, I have to hang up the phone now. " I was in absolute shock. Stacey was in absolute shock. This had nothing to do with the accessibility of the website. This had everything to do with the accessibility of the overall customer service plan for that financial institution.

Accessibility isn't just a little piece of the website the website or the application of it. It encompasses absolutely everything and that's one of the things we're all facing now with the Ontarians with Disabilities Act. Customer service is a big part of overall accessibility. It's important how we interact with people and it's a big part of why we're here today; to learn more about how accessibility has an impact on what we're doing with our businesses and in our public sector, and wherever it is that you all work. It has a huge impact on us in every possible way imaginable. That's something we all have to consider.

Now that was a very potentially isolated scenario but I would think that probably happens in more places than we'd like to know. There's a lot of good accessibility around us too. I don't know if you've had a chance to walk around the building. This is a sign right near the entrance, room 110, Deafblind Literacy. And I love being here, absolutely love being here. You walk around, you learn a lot about accessibility just by looking at the artifacts and the pieces of equipment that are here and even things like the wall signs that they have. There are some very obvious accessibility features here. If you take a look at the high contrast, the large contrast, white lettering on dark background. You can also see where the Braille is, there's a line of Braille that says the same thing. Room 110, deafblind literacy. If you take a look just above that, you also see raised lettering.

(ADJUSTING MICROPHONE)

Hi. Oh, look at this. Check, check, check. I'm going to feel like Bob Barker.

Where was I? Oh yes. You'll notice that there's not just Braille but raised lettering on each of the signs. And I look at this and I think that's fantastic, it's an alternative means of being able to understand what words are on the sign. Part of this sign that I love the most is the angle. Can you see the angle that the Braille and raised lettering are on? When I first saw it, it didn't really click, it didn't make sense. And then I started asking questions and it actually becomes quite obvious when you think about it. Somebody that needs to use their fingers to read is walking along the hallway with their hand along the wall and they get to the point where they reach the sign and instead of having to really crane their wrist so they can read it flat against the wall, it's on an angle so that it's a much more natural reading angle. It's one of those things where you look at it and it makes complete sense but if I were designing that sign I never would have thought of it. Never, ever, in a million years would I think if I was designing that sign in isolation. The beautiful part of this, there is lots of beautiful parts to this sign, that that little piece there, to me, is what makes this sign. That's the magic of design right there. And that to me is what we want to try and do on the web. Could this sign have been okay with just the Braille and the raised lettering? Probably. It would have been okay. But taking it into the context of use, this sign is now that much more usable by somebody that needs to read with their fingers. And again, I say, it totally makes sense when you look at it. Isn't that one of the beautiful things about design? When we see good design, we know it, we look at it, we say it's intuitive. It makes total sense. That's part of what I would like to bring in, and what I encourage all of you to bring into your designs is not just meeting the guidelines but understanding the context and use and making sure that things work really well for people.

One of the issues we tend to have in accessibility is that we don't know a lot about accessibility. We look for guidance from people. We look for guidelines and we've got guidelines everywhere now.

We've got, one of them, a set of guidelines that's coming in with the Accessibility for Ontarians with Disabilities Act. These things are being presented to us and we don't know much about accessibility. We look at those things and we take a very checklist approach to it. So I'm going to share a quote with you, which I hope resonates with everyone here. And I think I'm very good at demonstrating this quote because graphic design is much more than knowing how to add drop shadows and choosing colours, right? But me, I'll be the first to admit this. I'm not a graphic designer. I'm not a great visual designer but I can use Keynote and I can use Photoshop and I know that I need to put some big quotation marks on there to make it really stand out and I know I want a dark background and a light coloured text and I can choose a font that's not like the standard font. That to me is – so I'm actually quite proud of this slide. But you guys as real graphic designers look at this and say, hmm, it's okay. But you know what I mean with this quote. Graphic design is more than just being able to apply a drop shadow or a gradient, gradients are big, right? Put on the gradients and choose colour combinations. It's much, much more than that. There's so much more to graphic design and it's he same with accessibility. It's not just about following a set of rules or applying a set of formulaic actions. It's really about understanding how people with disabilities use the web or use their computers. And being able to provide a solution for them is much more than being able to just follow those rules.

Part of that really is looking at this idea of having a mindset about accessibility and this is something that we see all the time. I don't know if you're familiar with this chart, but if you take a look at it, it's a two-access chart and you see on the left we have a certain number of design alternatives at the beginning of the project. And as the project continues from the project definition to user research to design and implementation and launch, the number of design alternatives, as we move further along in that process, really goes down to zero. By the time we get to launch, there's really only one alter-

native. Right? Which actually doesn't really make it an alternative. There's only one choice. We get to the end and this is what we have to do because of all the decisions we've made all along. The cost to change a mind when we get to the end increases astronomically. If we made all these decisions all along, we get to the end, it's now too late or if we want to change it's going to double the cost of the project. Because we have to redo work. The biggest problem with this in terms of accessibility is that too much of accessibility happens here. Somewhere between implementation and launch. And that just can't happen. Part of it is that we have a mindset. When we look at things we don't necessarily go on in terms of all the detailed ins and outs of accessibility. We get a set of guidelines and so we go through a project and we get to the end of a project and before we launch a site, did I do this? Did I do this? Did I do this? We follow these guidelines. Really the right time to do it is at the beginning,

Accessibility needs to be part of the process. We've worked on hundreds and hundreds of sites and applications and projects with people and there's one common thread in terms of all the accessibility projects that we see, almost always. At the end, people that are working on the project always say something like this. If I only knew then what I know now, I would do things completely different. If I knew then what I know now, I would have made a different decision. I was in a meeting this morning, literally, where this client, which I shall not name, created a flash version of an e-learning project and they came to us and said okay, well this e-learning project is done, can we add accessibility to it. Kind of, not really at all. You can't. We went through and we tried to retrofit as much accessibility as we could to that flash file but at the end of the day, when we actually tried to use it, even though we had gone through and given everything the appropriate technical level of accessibility, it ultimately didn't work. Because it just wasn't designed with that in mind from the beginning. So it would have actually, you know, we started doing some math on this. In order to go back and re-arche-

type the flash file, it would probably cost them, you know, maybe another $15,000 of work. What the alternative was, let's build an HTML version, can you do that? Absolutely. That's a way it will cost about half that. We're taking the same assets, the same resources, and we're going through, we're going to build an HTML version now. Of course, the question should be asked, why do we need the flash version in the first place? Cause they kind of don't really but it's too late for that so there's nothing I can do about that. How many people do flash? Does anyone in here hate flash? So you guys all need to get together afterwards. It's all okay. Flash can be done quite accessibly. It doesn't mean that it's always the most accessible solution but it is certainly a solution and in certain environments works quite well.

So part of the reason that we have this mentality about accessibility is that we've got these guidelines all over the place and we've got the U.S. (they have Section 508), the W3C (The World Wide Web Consortium – they have a set of guidelines we can follow) IBM has their own set of web accessibility guidelines. A lot of groups that have been working with government departments, they have, in addition to their accessibility mandate, they're actually creating their own standards that they're using internally. So they sometimes go beyond what everyone else is doing in terms of accessibility, providing more strict rules of accessibility than others. The problem with this is that quite often with accessibility is that the attention is focused on a checklist. And as we know from my graphic design skills, the checklist isn't always that good.

So really the focus needs to be on people. I think this is an important part of – this is one of the biggest barriers, I think, to providing effective accessible solutions. It's that forget about people and it becomes about the checklist, right? Here's this set of rules we need to follow. The rules are really in place because we want to provide a solution to people in the first place, but we forget about that, we lose sight of that. It's about all these different people right? Any-

one with low vision, people with auditory impairments, people that may just have user mobility or dexterity – everybody. We're talking about everyone here and really it's about people.

So one of the issues that we have is we don't know everything about every different person and the way they use the web. So, we've got this set of guidelines and the W3C has really crystallized those guidelines down into four main principles. These are the things we want to talk about. What are the fundamental errors that people make when we're trying to create accessible solutions and almost all accessibility problems fall into one of these four categories. Its either the content that you're producing is not perceivable or it's not operable or it's not understandable or it's not robust. These are kind of the four corner stones of accessibility and this forms the basis of most accessibility guidelines around the world. Perceivable, operable, understandable and robust. These are just words. What we want to do here is take a look at what those things actually mean. And keep in mind as we do this.

One of the things we really need to understand is that just because something is compliant doesn't mean it's going to be easy to use. We can produce a solution that meets all the guidelines, that meets all these rules – and I'm not going to go through all the rules with you today, I'm going to show you examples of how certain things contravene those rules and what we need to do about it – but just because we follow those rules doesn't mean that we have an accessible solution or that it's going to be easy to use. After we've gone through all these rules we still need to involve and engage users and people with disabilities to use the solutions that we are creating so that we can actually test the validity of what we're doing. Is this site actually usable by people with disabilities? Who cares about the rules - I shouldn't say that - not who cares about the rules, rules matter but at the end of the day it's about whether people can use the solution you're building.

This is the perfect example of that. This is a compliant ramp solution for a transit station in Southern California. It meets all the grade requirements and it meets all the width requirements, but really what person in a wheelchair or not is going to want to use this ramp? Just because it's compliant doesn't mean it's going to be easy to use

Let's take a look at the issue of perceivability. One of the biggest barriers faced in perceivability, and this is sort of the traditional web accessibility starting point, like accessibility 101, we talk about images because people, who can't see, can't see the images. We have to have some other way for people that have no vision to be able to understand and perceive the content that is in those images. We have four basic types of images that we use in web pages and applications. We have images that are content or are functional images, so images that are part of links or are buttons for forms or things like that, those are images that we call functional images. We also have images that replicate and/or support other content on the page. We also have decorative images, things that are just there for mood setting or for tone setting or to put a little curve here and there. Things that don't really contribute a whole lot to content or in terms of meaning. And then we have a category of images that really shouldn't be images at all. Let's take a quick look at some of these and the issues we have with them. So, I'm just going to bring up some volume here.

On this next slide I've got an image that was taken from amazon.com and they're easy to pick on because they're big and we're all small so we're going to pick on them for a minute. It's a screen reader recording. So how many of you have heard of a screen reader before? So by association, the rest of you have not? So, hands up if you have not? So roughly, okay, so what about the rest of you then? Seriously? Ten, twenty people each time and there's got to be at least a hundred people in here. Okay, so this next slide – you don't have to admit anything either way – this next slide is an image from amazon.com about a sale they're having and I've got a clip of a

screen reader reading through it. So we'll do it. This is fun. Don't be scared.

SCREEN READER EXAMPLE: Hundreds of DVDs up to fifty per cent off at the big DVD sale. Link graphic. P selectE 0000YMHD6.001 underlineZE underlineSTZZZ underline…

DEREK FEATHERSTONE: Okay, I can't take anymore of that. Did you hear all of those zero, zero, underline, zero, blah, blah, blah? Those are images that are on the page. So you've got here, hundreds of DVDs up to fifty per cent off at the big DVD sale. The next thing you should hear, out of the text, is save on hundreds of DVDs or maybe something to do with those images. I'm going to play it again just to see how easy it is for you to follow along.

SCREEN READER EXAMPLE: Hundreds of DVDs up to fifty per cent off at the big DVD sale. Link graphic. P selectE 0000YMHD6.001 underlineZE underlineSTZZZ underline…

DEREK: Okay, so as I said, I can't take much more of that. So we went from that heading, we went straight to those images. Did you hear the screen reader say Link graphic? So it said link, it told the person it was a link, and then it said graphic. Hey, cool. And then it gave absolutely nothing of any use at all. What we need to do is make sure that when we have graphics on a webpage that are functional like this, that you can click on, we have to have an alternative for people that can't see the image or for cases when that image is on display. What a screen reader does when it can't find an alternative that you specify is it looks for something. It's part of a link, it has to say something, it can't just say link graphic and nothing. It has to say something. So what it does, the only thing that it can rely on (it can't even rely on it) but the only thing it can find is it uses the source attribute of the image. So when you play an image in the page and you say the source for this image is, whatever it is. In this case, Amazon is powered by a huge content management system and if you've used content management systems before you'll know

that quite often you have an asset uploaded. So you click on this, you upload all of your images and the names get renamed into something so it's kind a unique identifier. 000underscoreP00

underscorePL00 – whatever and that's what we end up with here. So that's absolutely of no use to anybody at all. So, what we need to make sure we do is put, an alt attribute on there, an alt text it says, that should say – I can't even read it - it just says accessibility issue. So the first one is a movie, for The Haunting. There should be information that says, link graphic The Haunting, fifty per cent off or link graphic, The Haunting DVD, fifty per cent off. Something that's meaningful that actually replicates the function of that image in the page because that's a functional image. Makes sense? It's obvious right. It's obvious yet, so many times it doesn't happen for whatever reason and these are the things we need to look out for. The other one should be Season Five of The West Wing, fifty per cent, whatever the appropriate text is.

This was a particularly frustrating image for us. We want to make sure that everybody understands. We're not just talking about people that have visual impairment, we're looking at people with all different types of disabilities and accessibility needs. This is a case where this is a navigational image and again, a functional image on a page, and the image says, contact us. So we were doing some work testing with a person that was using voice recognition software and one of the tasks was to be able (when we were testing) was to be able to submit a contact form or find out the contact information for a certain person. So, Kyle is sitting there in his wheelchair using voice recognition software and he's trying to complete this task and he says, "contact us" and nothing happens. "Contact us…contact us…contact us…contact us," trying to tweak his voice so that the voice recognition software would recognize him. And we'd been doing testing for about half hour, so his mouth was getting dry, so he was starting to think well maybe it's me. We all, I always, blame myself when I can't make things work, like 'oh, I must

have done something wrong.' Right? And it's not necessarily him. So he turns to his father and says, 'hey, can you get me a drink of water? I need a little bit of a break here.' So he takes a sip of water, gets himself ready, comes back ready with his voice booming and says, "Contact Us." Nothing. So I finally said we have to stop and look at what's under the covers. The problem was that the alt text on that image said, comments. It's very likely that that image, at one point, was a comments image, but what happens when we change a design or somebody from up above says, we're not going to call it comments anymore, we're going to call it contact us, because that's way better. So they switched the image but they didn't switch the alt text.

Alt text is not just an important concept for people who can't see. It's an important concept for everyone. People who use voice recognition software rely on this as well. But another example here of an image used on Amazon site again. I want you to listen to how many times you hear the word 'help'. So the screen reader's going to go through it. It's going to read the title bar of the pop-up window and then it's going to go through the content itself. So let's listen and hear how many times you hear the word 'help'.

SCREEN READER EXAMPLE: For Amazon.com: help at amazon.com; help(help help)link)...

DEREK: Amazon.com:help and we get into the page itself and then it said help bracket, help help, link bracket, close window. Why did it say help three times? We have an image there, a little question mark image, that's a help image. That help graphic is really a visual reinforcement of the word help right beside it. If we have images that support content that's already on the page, we don't necessarily want that image to have any alt text because then it's going to sound repetitive like that. So if we got a scenario like that, when you've got a graphic that is supporting or replicating other information that's already on the page, we want to leave that out. We

don't want that read, generally speaking, by a screen reader. So 'alt=quotequote' in that case, that would have been read as 'help'. It's easier to understand.

AUDIENCE MEMBER: Wouldn't it read out the name of the image?

DEREK: If you say 'alt=quotequote,' the screen readers know to skip right over it. So that's a technique we use so that it won't get read on at all. So that's actually a useful technique. 'Alt=quotequote' : screen readers will ignore it so it will just read 'help' once, thankfully.

This is another one of my favourite ones. This is an image in a web page. An image of text. An image of a schedule. Here's the alt text for that image. Alt="Debate Schedule," which is kind of accurate because that's what it is, but it doesn't give us any information that we're unable to perceive. One of the problems that we have in accessibility is that stuff like this, you can't test with a tool. A tool can test this and tell you whether or not there's alt text there or not but it can't tell you whether that's appropriate. So one of the things we need to do as designers and developers is take a look at things and say, what's the purpose of this image? What do I need to do for the alt text for that? In this case, a better solution – that's a lot of information to be putting in alt text. This should just be text inserted in the page in the first place. We tend to have, and I've worked with design teams and even myself before I knew all the things I know now, it's very logical. I want a specific font so I'm going to make that table an image. It's a very logical thing or it seems like a logical thing to do so we can present it the way we want it to. It doesn't always mean it's the right thing to do so lots of things to consider when we're talking about that idea of perceivability.

I'm going to share some more things on operability. So, we want to make sure that everything in a web page is operable. We want to make sure you can use and you do what you need to do to it. So let's take a look at an example of an operability failure.

This is a Google map and I love Google Maps. Love Google maps. I think it's a game-changing web technology because it really drove forward the use of ajax, which can be really helpful for a lot of people. And I'd go so far as to say it's helpful for people with disabilities when it's used right. This Google map has a route on it that is plotted and you can see, in the upper left hand corner of the map, they have the panning controls and the zoom in and zoom out, pan up, pan down, pan left, pan right and a re-centering button. So we started looking at this, this is a native Google map and we wanted to embed Google Maps into our blog. My wife and I both race tria-8thalons so we wanted to chart our routes with a GPS and embed them in our blog. In order to do that, I wanted to make sure, as an accessibility person, that there's actually a reasonable amount of accessibility with this solution. So we started to say, what is required for compliance for this? Well in order for this to be compliant there are a lot of things we need to do, in particular with the controls, they have to be operable with a keyboard. We have to be able to do things with a keyboard. We can't rely on mouse click for everything. So we started looking at this. Again, I pulled up some voice recognition software (Dragon NaturallySpeaking) and you can see at the top of the screen, you'll see a little Dragon bar. That's the Dragon bar, which is always running when you've got NaturallySpeaking running and if you look in the very left-hand corner you should see a little yellow box that says, 'link'. That tells me that the last command that it understood, that I said, was link. And you can see on the page there, there's an overlay of a whole bunch of numbered arrows. And the reason that we use this is that, if I wanted to go to link 23, which says Birthday Bike Ride, instead of me having to say 'Birthday Bike Ride' to Dragon NaturallySpeaking and having it understand what I said and acclimate that link, I can just say 'link 23'. Link: it shows an array of all those numbers. I say 23 and it acclimates that one. It's actually quite useful technology. So the zooming in and the zooming out and the panning controls are not links. So the only other thing we can usually manipulate with the key-

board is our buttons. If you see on the top left here, I've said 'button' and it doesn't recognize any of those as buttons. So we started to think what are these things and what are my alternatives now if I'm using voice recognition software? One of the fallback mechanisms the voice recognition user has is something called mouse grid. So what you would do is say, 'mouse grid' and the same software overlays a grid on the page with nine blocks in it. You can see the numbers in the centre of each block. So in order to click on that little, tiny zoom in button, what I would need to do is say, 'mouse grid 4...6...1...6'. And now I've finally got a small square that's overlaid that top control and I can say, 'click that'. And when I click it, it virtually moves the mouse cursor there, engages a click and you can see the map is actually zoomed in. These controls that are part of a default Google map are not a button, they're not a link, they're not natively keyboard focused, you can't do anything. So we found out what they were, we actually looked, and it's essentially, for those of you who are more code oriented, they've got a dib on the page and that set of controls is the background image in CSS. The style sheets are specifying that there's a background image here and then the dib has some JavaScript running on it so when you click, it figures out what should have been pressed and what action to take, completely inaccessible to the keyboard. To Google's credit, they have keyboard alternatives. You can use the plus key and the minus key to zoom in and zoom out. You can use the arrow keys to move around. The problem is they only really work when the mouse curser is already overtop of the map. Do you see the problem with this? If you're a keyboard user, how do you get the mouse curser over the map? There are issues there and the direct issue is that if you have an onscreen control, you have to be able to manipulate that directly. You need to be able, no matter what alternative you can provide, if there's an onscreen control it needs to be something that you can manipulate directly.

This is what we're talking about when we're talking about operabil-

ity. When we implemented these in our blog, all we did was create a custom controls. You can see here as I've said 'button', in the top left hand corner, each one of those controls is actually just an HTML button and it has all the behavior attached to it so that they function and are tapable with the keyboard. So you can tap to them, you can use the enter key or the space key to activate those buttons.

Have you guys heard all about the stuff going on with the TTC and the trip planner? I don't know if you've heard of that but anyways, there's lots of stuff going on with the TTC. We're about to launch a trip planner and I can't say when because I'll never see anybody again. I'm not allowed to talk about it. But look for these when the TTC trip planner launches, look for these types of controls. They are custom controlled but they are keyboard usable, you can use them with a keyboard. I think I've probably already said too much actually.

Understandability. We're going to wrap this up here. Understandability is a really interesting topic to me. I was at a hotel in London, in the U.K., and I got into the shower and I faced this. A red hot water tap on the left and a red hot water tap on the right and I had no idea what was going on. I'm in a foreign country and I don't get it and I got out. I got out of the shower and I stood there and I looked in the mirror and I over my shoulder I see this notice: the left tap operates the shower and the right tap operates the bath. So now I at least know if I'm going to get burnt on my head or my feet. At least I know, right? The problem with this was, the tap is down here and the instructions are five feet away, up on the wall. I couldn't see them until I was actually looking in the mirror and saw it over my shoulder. Those instructions need to be right by the tap so that I can look at the tap and see the instructions in my line of sight. This is a principle in design that's called proximity. Things that are related to one another should be close to one another. They have to be close to one another.

Let's look at this, in an interface. This is a recording of a screen reader interacting with this, 'what is this?' link on the Chapters website. So they've done a really great job on the Chapters website. They're using really robust technology. They've done a great job. The 'what's this?' link works. Whether you have JavaScript on or off, it doesn't matter. With JavaScript on, what it does is go off to the page where the 'what's this?' content is and brings it back to the page in a little dialogue box that looks all float and nice and Ajax in it and I'm sure all of the managers were excited about it. Thinking about the principle of proximity, let's listen to what happens to a screen reader and we go through.

SCREEN READER EXAMPLE: ...Tag. What is this link. Enter.

DEREK: So you saw what happened. We clicked enter on the 'what is this?' link and that dialogue box came up. I want you to watch what happens next as that has come into the visual field of view, I'm going to continue to try and interact with the keyboard.

SCREEN READER EXAMPLE: Link. Level 4. Read to order. Link. Link what is this?

DEREK: So now I'm going forward and bag to see what's happening.

SCREEN READER EXAMPLE: Tag. What is this link? Tag. Add to shopping bag. Ship tag. What is this link? Tag. Tag. Tag. What is this? Tag.....Tag. Customer Profile Selection link. Tag...What is this? It's a tag.

DEREK: How sad is that? It took forty keystrokes just to get to that thing. Why? Because when they brought that new content into the page they put it in a convenient place for them, which is at the bottom of the body at the HTML page. They brought that new content and put it in as the last thing in the body. What they didn't do was think, not just visually, but think from a keyboard perspective. They did a fantastic job when we clicked on it, it came into the visual

field of view. It's kind of like, hey, I'm here, this is it. Interact with me. They need to make it so that the keyboard user gets the same advantage. Here I am, interact with me. We can't just let things pop up and show them in the middle or the end of the page or in a random spot. This is the tap instruction for the tap being too far apart. By the time we got here to this dialogue box, I forgot that I even clicked. It's not relevant anymore. So if you're bringing new content in through a page, we can't just think about the visual, we have to think about the actions of actually tapping through that page and how we would interact with that. So we would want to put the focus in there and we would want to make sure that when that gets injected is right after the link that spawned it. Make sense? We can't just think visually.

Last bit here. Really, the last piece of POUR (Perceivable, operable, understandable, robust) is that we need to use technology in a robust way. For documents on the web, we generally do this, we create our content in HTML, we create a layer of presentation with CSS and we create behavior with JavaScript. So hands up if you've heard this before. Okay so about twenty people in the room. For those of you who haven't heard of this, this is a methodology and philosophy of building websites and applications so that if we do this it provides for flexibility and adaptability. At the end of the day, if all you have is really beautiful design and no content, do you have anything? I would – well maybe you do, I don't know. But I think isn't that what we call art? The art is the content right? But in a lot of cases – I didn't mean that as a diss or anything, I'm serious. Art is the content, right? But when we take that content away, if all we're left with is design, we kind of have nothing. If we have HTML and we have that content and we take that design away, we still have something. It may not be as pretty, it may not be as wonderful or as engaging or as compelling but at least we still have something, right? The foundation of all accessibility, from my perspective, is that HTML comes first, well-structured HTML, and then we create our

layer of presentation making it look absolutely fantastic and then we enhance the behavior with JavaScript. That works for content, that works for applications where we have content and behavior, things like clicking links and submitting forms. Those are things that are core to HTML. We still create presentation with our CSS in an app and we still have advanced behaviors in JavaScript. Using this methodology is what being robust is all about. And you may have heard of this before but really this is philosophy called progressive enhancement. If you haven't heard of that, when I first discovered it, it kind of changed the way I approached building anything on the web because the foundation of it is HTML and we layer everything overtop of it, but at the very least we always have that content. So definitely look up more on that.

This really lets us provide a flexible and adaptable solution and I think what this does is, it comes back to people. We cannot ever predict the variety, the range of needs of accessibility needs that our audience will have. Two people that are blind, they interact with things in a completely different way. Two people with low vision may have different degrees of sight. They may need different degrees of magnification, they may need different colours. Everyone needs potentially something different. And don't think that this may not happen to you. This may end up being one of us or all of us, who knows. We need to provide flexible, adaptable solutions. If you're looking for more info on he progressive enhancement approach or the web standards approach you can check out the Web Standards Project. The Web Standards Project was founded by Jeffry Zeldman who was one of the first people to basically say, browsers are part of our problem right now. This was way back in the nineties when the browser wars were going on and there was inconsistent implementation between browsers so we started the Web Standards Project. I'm heavily involved and were basically a group, I feel like I'm all preachy right now or something, but basically if you look for searches for web standards approach and web design, it will help you in your quest to deliver more accessible solutions.

With that I think we're out of time so I'll thank you very much and hope you have a good conference.