WeCo

WCAG 2.1 Primer

Monday, May 20, 2019

CAPTIONING PROVIDED BY CAPTION ACCESS LLC
http://www.captionaccess.com/
For assistance email support@captionaccess.com

*** TypeWell transcription provides a meaning-for-meaning representation of spoken conversation to facilitate real-time communication access. This rough-edited copy is provided for reference and is not a verbatim record of the proceedings. ***

I want to welcome everyone. This is our accessibility meet up. We'll be talking about WCAG 2.1. We have the WeCo Access Team with us tonight.

Speaker: This is an option we have available to anyone who is part of the Meet Up who may not be able to make it here. We record everything that's going on. At the end, it will be available at the Meet Up page. You can share this with your boss, colleague, etc.

One of the options we have tonight is we'll be screen sharing the PowerPoint presentation on another attendee. It will be under my name, Andrew Emerson. Select the shared screen with video option. You can go between our camera as well the PowerPoint presentation. We have some options today. We heard that it was tough to read the screen at events. You have the option to watch the PowerPoint presentation as presenters are sharing.

Sue Ann: Hello. Thank you for joining us in person as well everybody joining us on Zoom.

I'll be introducing the presenters for tonight's presentation. However, I wanted to let everybody know that the next Meet Up will be on July 29th. It's going to be kind of expanding upon what we're talking about today, which is WCAG 2.1 guidelines. On July 29th, the Meet Up will be [Inaudible] to 2.1.

Hopefully everybody can join us. Bring your question and your knowledge. We welcome both.

To introduce the presenters, my name is Sue Ann Rodriguez, not Rogers. That's OK, Andy.

I'm the director at Accessibility Services at WeCo.

Co-presenting with me is Dane Denham, an Accessibility Specialist here at WeCo.

Finally, my other co-presenter is Jen Hurst, our brand new Accessibility Specialist.

With that said, I will be officially beginning the presentation. Again, welcome to the WCAG 2.1 Primer.

Next slide.

The people that are on Zoom, toilet you know, I do have a speech impediment. If you think that you are having technical problems, it could just be me getting stuck on a word. I'm blind, so sometimes my screen readers act up, so I get behind on my notes or talk too fast and go ahead of my own notes. Just to let you know.

We're on slide 2, right?

I want to give you a quick overview of how this presentation will be presented. We're going to do some general basics about the WCAG guidelines. Then we're going to introduce the 2.0 guidelines, what that involves, and specifically, go through the brand new 2.1 Success Criterion, after that.

In conclusion, we'll see if there are any general questions. If anybody would like to talk about how they're testing now, if they're starting to implement the 2.1 guidelines yet, and so on.

We'll start off by answering the question, "Who's creating the guidelines?" The WCAG stands for the Web Content Accessibility Guidelines. Who's creating these?

The WCAG guidelines are created by the World Wide Web Consortium, also known as the W3C.

This is a private organization made up of a cross-section of stakeholders from the government, industry, and consumer groups that create these standards.

What exactly are the Web Content Accessibility Guidelines? They are guidelines that specify criteria used to determine whether digital content is used. What is accessible to people living with disabilities?

The next question we'll answer is, "Who's adopting them?" The WCAG 2.1 [Inaudible] has been accepted and adopted by organizations all over the world.

We'll get into a [Inaudible] about this in a minute.

The WCAG guidelines in general are the following. They are organized under 4 principles: Perceivable, operable, understandable, and robust.

For each guideline, there are testable success criteria, which include three levels: A, AA, and AAA.

Next slide.

We're going to talk about the WCAG 2.1 guidelines now specifically. Since its release in 2008, the WCAG 2.0 guidelines have continued to be a viable standard for digital accessibility.

However, known gaps existed in this version that needed to be addressed.

Due to evolving needs of digital accessibility, in June of 2018, the WCAG 2.1 became the official recommendation of the W3C as its first attempt to fill some of the gaps.

The brand new WCAG 2.1 success criteria addresses items related to mobile such as small screens and touch screens that accommodate users with motor and dexterity disabilities, Users with low vision and users with cognitive disabilities, as well, in addition, there are success criteria that benefit users of voice inputs such as speech recognition users, users with vestibular disabilities, and users of screen readers.

Next slide.

In the WCAG 2.1 version, there are a total of 17 new success criteria. There are 5 level A criteria. There are 7 level AA criteria. There are 5 level AAA criteria.

The WCAG 2.1 version includes all the success criterion from WCAG 2.0.

Sorry.

Basically, it's backwards compatible.

Sorry. I took a drink of water.

All of the WCAG 2.0 guidelines are now included in the WCAG 2.11.

The European Union directive that comes into effect in the EU members is going to be going to include the WCAG 2.1 guidelines in September of this year.

I'm sorry. It actually occurred in 2018. Sorry about that.

Again, the European Union has [Inaudible] the 2.1 guidelines.

In the United States, Section 508 has just adopted the WCAG 2.1 guidelines. They did say in a webinar that I attended last year that it won't have any intention of adopting WCAG 2.1 at this point in time or the near future.

The Americans with Disabilities Act (ADA) only states currently in title III that communication including web/internet must be made accessible. It doesn't reference any particular standard at this point in time.

However, up until now, the WCAG 2.0 has widely been considered the default standard when it comes to the ADA website compliance.

However, in November of 2018, a county in Florida....

Sorry. I talk a little bit faster than my notes.

A county in California in November of 2018 reached a settlement with the national Federation of the Blind that applied the WCAG 2.1 new guidelines in the settlement. I believe it had something to do with the voting process, if I'm not mistaken.

Nope. Sorry about that.

In Canada, there has been no announcements on whether the AODA will be updated to refer to the WCAG 2.1 new guidelines as of February of this year according to the article that I read about this.

Next slide.

I'm going to talk about the specific new WCAG 2.1 guidelines. We're going to start with level A, then level AA, and finally, we'll do level AAA. I know that those aren't widely required to meet, but we just want to give you a taste if you don't know what they are. I will begin.

The first one is character key shortcuts. When a page has shortcuts that can be activated using a single key such as a letter, number, punctuation, or a symbol key, one of the following must be true, one of the following three conditions must be true.

1. The shortcuts can be turned off.

2. The shortcuts can be changed to also require pressing keyboard keys like control, alt, and CMD.

3. The shortcut for a certain element is only active when that particular element has [Inaudible.]

Following these particular guidelines will improve accessibility for people who use speech input like speech recognition software programs like Dragon Naturally Speaking for browsing and websites. It will also include accessibility for users who have hand tremors and can easily press keyboard keys.

The next one is pointer gestures. Functionality on a page needs to be operated using a single pointer such as single click/tap, click/tap and hold, or a double click/tap.

Following these guidelines will improve accessibility for users with limited motor skills.

The next one is pointer cancellation. At least one of the following must be true for an action such as a click, tap, or a long press.

1. The down event is not used to complete a function.

2. If a function is triggered by an up event, a user can cancel or undo the action afterward.

3. The upward event cancels when triggered on the down event.

Next slide.

Completing the function on the down event is essential. This is my favorite one. [Laughter]

This next one is called label and name.

The accessible name for a control needs to include the text of its visual text label. Text that is shown on interface components like buttons must be able to be the following: Free to users of assistive technologies like screen readers, triggered by voice commands by users who use speech recognition software, and finally, motion actuation. When [Inaudible] user motion such as shaking, tilting, or gestures are picked up by the device's camera and are used for interaction for functionality. An alternative method should be required to perform an equivalent action unless the action is essential.

Next slide.

We're going to pass it over to Dane.

Dane: Can everyone hear me alright?

Sue Ann: Yes.

Dane: Perfect. Thank you.

I'll take it from here.

This is the AA success criterion.

Orientation. Web content should not prevent the user from changing the display orientation to either portrait or landscape.

Constant must be operable in either orientation. This criterion addresses restrictions posed by the content and does not address settings enforced by the platform or device like a screen lock that you put on your smart phone, for example. It wouldn't be covered by this.

The intent of the criterion is to give user flexibility on how they use the content. This [Inaudible] a device mounted in a fixed orientation like the arm of a wheelchair or some other fixed mount.

It also benefits users with low vision because they can rotate the device to change the text size and size of objects on the screen.

Developers can change this by changing CSS content to allow porter and landscape orientations.

Identify input purpose. As indicated in WCAG 2.1, it needs to be communicated pragmatically such as via auto complete and [Inaudible] attributes in HTML. The purpose can be communicated by tools in different ways like icons or symbols.

The intent of this is to help people better recognize the intent and [Inaudible.] It also allows users to auto fill information.

Text should reflow without requiring scrolling in two dimensions when the horizontal width is [Inaudible] pixels and the vertical height is 250 CSS pixels.

Reflow must not cause loss of content or functionality. Although, content and functionality may be presented in different ways like a pop up menu rather than a navigation bar.

This guideline is good for the user experience, specifically it helps users with visual disabilities zoom into content by requiring a 320 pixel fit. It's also ensuring that websites are mobile, friendly, and responsible.

Next slide.

Non-text contrast. Visual details needed to identify graphics, user interface controls, and their states need to have a contrast ratio of at least 3:1 with adjacent colors. This includes, but is not limited to buttons, form fields, focus indicators, selected state indicators.

This guideline allows users with low vision to see content more clearly.

Text spacing. Ensure that when users override text spacing, there's no loss of content or functionality. Text must not be cut off or missing. Only specific settings are required to meet this standard.

Languages or technologies that don't support a particular setting only have to meet the settings that apply to that circumstance. This guideline helps to avoid overlapping text or buttons being moved to places where the user can't interact with them.

The purpose of this guideline is to help users with low vision customize their reading experience and still being able to interact with content.

Content on hover or focus. If a user triggers content in a mobile window tool tip or similar component, the user must be able to dismiss the content without using the mouse or moving the keyboard focus, move their mouse over the content without making it disappear, and dismiss the content when they want to.

Status messages. When new status messages are added to the screen without changing the users' context, users should be made aware of the important changes in content without moving focus to content. Status should be communicated in ways that don't unnecessarily interrupt their work.

Excuse me.

The messages to the user should be pragmatically determinable through a role or properties.

For example, if the user presses the add to shopping cart button, the screen reader must announce, "item added" or number of items added to cart without disrupting what the user is doing.

For example, if the user enters the postal code incorrectly, the screen reader must announce "invalid entry." This purpose is to assist blind and low vision users who rely on screen readers.

Sighted users can tell how many items are in a shopping cart or if items are added by looking at the screen. Often, this doesn't interrupt their shopping. Adding a status message allows low vision users to be aware of the same information in a more equivalent manner.

That's it. I'll pass it off to Jen.

Jen: Thank you. Next slide.

For the level AAA criteria, first item is identify purpose. This criterion builds on the 1.3.5 criterion and includes communicating the purpose of icons, regions, links, buttons, and other user interface elements to support personalization for people with cognitive and learning disabilities.

Next, we have time outs that advise users of durations of activities that will cause the time out and result in lost data.

Next is animation from interactions. Users can turn them off unless animations are essential to the functionality or information conveyed. Examples include moving, growing, or shrinking content or parallax scrolling triggered by user interaction.

Target size. A clickable element must be at least 44x44 CSS pixels.

Concurrent input mechanisms. Users should be able to add, remove, or switch between different input mechanisms like a mouse, keyboard, stylus, touch or voice input.

Next slide.

We'll go over primary requirements to conformance to WCAG 2.1.

For A conformance, the page must satisfy all level A success criteria or an alternative version is provided.

For AA, it provides all A or AA success criteria or a level AA alternative version is provided.

For AAA, it satisfies all A through AAA success criteria or another AAA alternative version is provided.

It's not recommended that AAA be required as a general policy for entire sites because it's impossible to satisfy all AAA success criteria for some content.

Conformance and conformance level are for full web pages only. It can't be achieved if a page is excluded. [audio cut out.]

When a web page is in a series of web page such as steps needed to accomplish an activity, all web pages conform at the specified level. For example, an online store has a series of pages to select and purchase process, all pages from start to check out must conform for the process to conform.

Conformance isn't possible at a particular level if any page doesn't perform at that level or better.

For a complete list of requirements, you can visit https://www.w3.org/TR/WCAG21/

Sue Ann: Thank you, Jen and Dane.

That officially concludes our presentation. However, before we take questions, there are just a couple of points that I'd like to point out.

As you heard, that was a lot of information to be presented in a short amount of time. I just wanted to give you a highlight of what these new success criteria are and which users they would help when they are met.

As you probably could tell, they focus on multiple devices and they adjust for cognitive impairments, low vision users, and in particular, speech recognition software. I know that was really part of the big 2.1 version.

With the 2.1 new version or guidelines that are included, note that only a few of them can be officially verified for accessibility using an automated tool. For example, [Inaudible.] Just letting you know that.

At WeCo, we haven't officially made the 2.1 guidelines part of our test product quite yet, but when a particular client requests it -- again, it's about to be a year old. There isn't a lot of standards/guidelines that are requiring them. We are probably going to incorporate them into our test products in 2020.

Also, it is a good idea, even though it's not required yet, to try to start to incorporate those 2.1 AA and/or level A so you can be ahead of the game.

With that said, we'll take questions. We're not experts on these 2.1 guidelines by any means, but if we can't answer your question, we will put it out there for other attendees to possibly answer that question. We could also potentially do some research for you to get that information, as well.

With that said, I'll take questions from Zoom people first. Andy, are there any question via Zoom Yet?

Andy: There are some folks on. Feel free to share your questions through the microphone or chat feature. I can read it now. Right now, we don't have anything yet.

Sue Ann: OK. Why don't we start with the people here in person?? If you don't have questions, I have a question or two for everybody to respond to.

Any questions?

Attendee: The auto cancellation portion of the requirements, is that intended to require that any error mouse click is cancelable or is it for things like [Inaudible.] You can cancel that?

Sue Ann: Dane, do you know the answer to that question? I'm going to assume it's only related to the down events, or is it related to all mouse events?

Dane: The pointer cancellation, the one where you can drag the pointer out to cancel.

Attendee: [Inaudible]

Dane: Yeah. If I look at the notes I just read -- forgive me. We aren't experts on these yet because they're so new. The pointer cancellation -- let's see. That's content on hover.

The pointer cancellation would be similar to making sure that you if you do something on a down event, you can cancel it.

For example, it says don't rely on the down event. Wait for the up event. If the user holds down the mouse button while their pointer is on a link on a web page, for instance, don't start navigating the user to the page until the up event has been received, meaning the person has released the mouse button within that same clickable area. If the user holds it down -- and I've done this before as a screen reader user -- I am partially sighted. I use a reader to read content under my mouse. I'll sometimes accidentally hold down the mouse button. I don't want to click there. I'll drag the mouse out as I drag the hyperlink. I'll let go of it outside the area of the content.

When it activities on me, that's what that's talking about.

Attendee: Thank you.

Sue Ann: Any other questions from attendees here in person?

OK. Any questions from anybody via Zoom?

Andy: We haven't received anything yet.

Sue Ann: OK. I have a question for everybody, Zoom and in-person attendees.

Has anyone started testing for WCAG 2.1 version or has anybody started to actually develop for WCAG 2.1 version yet?

Attendee: [Inaudible]

Sue Ann: [Laughter]

Anybody on Zoom care to share in response to either of those questions?

OK. With that said, are there any other 2.1 version questions?

Attendee: I kind of have a question, but it's maybe not exactly about 2.1. If I can implement some of these things right away, would what the most important aspect be? Which one should be focused on first?

Sue Ann: That's a really good question. In my professional opinion, I would recommend starting with the level A ones. They are what I'd call the easiest compared to the AA ones. Anything that you can do is going to help the affected users. That would be my opinion. It's better to start somewhere than not at all.

I would think the level A ones will be the best way to go.

Dane, any thoughts on that one?

Dane: As far as which ones to do first? Not really, no.

A lot of these are pretty simple. One that I would say would be a good one to try to implement first would be the status messages one. That's easy to do with Aria Live [sp?] regions. HTML and JavaScript would be one.

Like Sue Ann was saying, just the level A ones for being pretty simplistic.

Sue Ann: Dane makes a good point. We've recently seen a lot of clients -- when you zoom something -- forgive me. I can't think of a specific example right now.

When you select something, there's an alert that pops up on the page. It seems like clients are going that route, the ones that we've seen here at WeCo. It's so frustrating because Dane and I work together when we do these, our testing for one of our products. I'm like, "What do you mean it says that? I don't see that anywhere." It's really helpful to incorporate those alerts because it announces that a change has occurred on the page even though the page itself hasn't refreshed or I haven't went to a particular different page.

Dane: One more here if I could jump in for a quick moment. Since you're finished with that one, I'll jump in and say that on pointer, on hover, or focus one would be another good one. I've seen a lot of websites that have custom tool tips that show up on form fields. As a user with low vision, I sometimes magnify the screen, but not often. My vision is so low that that's sometimes useful. Often it isn't but when it is and my reader helps me read it and it keeps disappearing, that's problematic, too. That's another relatively easy one to implement, I would assume.

Sue Ann: We had a question?

Attendee: On the [Inaudible] messages, ARIA, a single page app has kind of fallen into three categories [Inaudible.] The other is that the wheels fell off. Then there's the inline, "don't do that. You're typing that wrong. That's not validating."

From the sound of it, the [Inaudible] thing should never interrupt user flow. They could say [Inaudible.] We don't need to tell them to change the focus, [Inaudible] to read that. In other cases, it needs to be read that something went wrong.

[Inaudible]

Sue Ann: To answer your question, the first item was called what? What's the message?

Attendee: It's a success message. It could say, "congratulations, it worked."

Sue Ann: That's important to convey to the user just to confirm to them that you've filled this out correctly or thanking them for filling out the form.

Attendee: [Inaudible.] You don't need to display that? It's disruptive? [Inaudible.]

Sue Ann: Dane, what's your opinion about it? I guess I can go both ways on that.

Dane: Can someone repeat the question into the microphone?

Attendee: The question is regarding the status messages. It sounds like there were two possible status to be conveyed. One's that something went wrong. The other is that everything worked perfectly. With the perfect message, it sounded like it's best not to interrupt the user's flow because they know they did it. Don't [Inaudible] to the cart because you're disrupting the shopping flow. The other is the error message that we want to highlight to tell them something went wrong. Are we on track with that?

Dane: Not quite. Maybe I didn't say it clearly enough when I was going off of the notes for the slide deck.

Really, Sue Ann was saying any status message is important to convey to the user because if you've conveyed it -- if you code it correctly and it's conveyed properly, it doesn't interrupt their flow to have their screen reader quickly announce it.

What would interrupt their flow would be to move their focus from one text field to another part of the page entirely. That would definitely interrupt the user's flow.

Coded correctly and implemented correctly by both the browser and screen reader, the status message conveyance itself doesn't actually interrupt the user's flow. That's why they recommend having them conveyed even if they're success messages. That would be a form of status message and would fall under that success criterion if you're doing 2.1.

Sue Ann: To go along with that, I don't know if you know, but [Inaudible] came to our last Meet Up. There are different types of other messages, like the light version . . .

Dane: The other one is assertive.

Sue Ann: You do have options. I think the one that is polite says that you can get the alert message when the person isn't doing any movement or performing an action.

You have different options on when to have those alert messages announced.

They don't have to be 2.1 questions. As you can see, we veered off a bit. Any other accessibility-related questions? We can answer those for you. We're here until officially 7:00. We can go longer if necessary.

Attendee: I have another question. [Inaudible.] Say you have a form. You push "submit." Often, it redirects you to another part of the site. I would think trijet would be a problem because then you don't know where your focus is? How do you tell the person that they've been redirected?

Sue Ann: Typically, screen readers announce to users of screen readers is that they say, "redirecting to page." They specifically say the page. We're aware as screen reader users that we're going to a different page. Dane, is that correct?

Dane: Yes. In fact, the whole typical form submittal -- you get a response back from the website that takes you to another page. That actually falls outside of the purview of the status messages success criterion. That's a normal web browser thing that's been common for screen readers to convey ever since the dawn of the web basically and of screen readers. So yeah, that wouldn't be any issue at all, just going to another web page, as long as it's a full new page load. Then yes.

Attendee: What about single page apps where you don't have refresh? [Inaudible.]

Dane: You mean like a single page app where the content is changed dynamically like Ajax or something like that?

Attendee: You submit the form. The URL changes. You're on a different "page," but the browser isn't refreshed.

Dane: It just pushes new content into the same page? That, you would need a sort of status message. If it doesn't cause the browser to request a new URL and load a new page, it's not necessarily going to be conveyed automatically. That would fall under the purview of this. A status message would need to be provided.

Attendee: Would you recommend that status message be customized in some way for screen readers beyond what the normal user would see?

Dane: No. Not necessarily.

If the status message is conveyed visually like a banner that shows up at the top that says, "update applied" or "password changed successfully" or "one item added to cart," whatever the case may be, you can have that coded as an alert with ARIA. Usually you'd have an empty element in the page that's marked up that way to push the content to so the screen reader knows to monitor it when the user first launches your app. You'd have to convey that as a status message. You wouldn't need to do it any differently.

Attendee: [Inaudible] in my brain the screen reader [Inaudible.] The screen reader will know when the alerts have content added to them.

Dane: Yeah. The ARIA live property in the spec tells a screen reader, if it's applied to the element when the initial page of the app loads, it knows to monitor that and announce the new content. It doesn't actually require it to receive focus, but generally, for any other element, just to answer your question, yes. You would need to put focus on it for the screen reader to get to it but not for elements marked up with the ARIA live.

Attendee: Actually, when a page loads?

Dane: Because it's not going to know that, to monitor it, you can't add that attribute later on.

Attendee: A single page app basically [Inaudible.]

Dane: Right. The whole dom is already there. It's just changing it with script. If you put an empty container that is maybe not even visually shown yet, you'd push the alert into it. That would allow the reader to announce it when needed. Does that make sense?

Attendee: Sorry if it's a rabbit hole. Is it possible to pragmatically tell screen readers that this element now exists? In most apps, you don't have a dom loaded. You have a script. You have the dialogue, including the ARIA element dynamically. On page load, there's no content. JavaScript puts content on the page. At that point, you have an aura alert container. Is there a way to offer the initial dom on the page to focus a screen reader on it to let the user know that it's there?

Dane: That's a good question. We'd have to research that. Do you have anything, Sue Ann?

Sue Ann: Can't you add the code -- do the dom part first? Then further down, have that container for the ARIA live?

Attendee: Maybe. [Inaudible.]

Dane: Here's the thing. I know it's been done before, obviously. I've seen it happen. I just don't know exactly from a scripting/JavaScript side -- I do know HTML, but not good with JavaScript. We'd have to research that and figure out how that works. I assume that there might be a way. It's just that I've heard that there needs to be some sort of empty container present. As soon as that element first exists. That's how I understand it. It's kind of hard to explain actually.

Sue Ann: Any other general questions? Any 2.1 questions before we officially conclude this Meet Up?

If there's anybody either in person or on Zoom who has any ideas of topics that you'd like for us to talk about in upcoming Meet Ups, we're always looking for ideas and can always use some ideas about different topics that you'd like to hear about and know more about.

If there's anybody on Zoom who'd like to present some ideas, please type them or say them.   Anybody in person can stop by and talk to me on your way out.

I just want to give a final opportunity for the those on Zoom and in person if there are any questions.

Again, thank you so much for joining us.   I hope that you learned something from this presentation today.   Please remember that we're going to be having another Meet Up on July 29th.   The topic will be moving mobile to 2.1.   If you have any experience, knowledge, or questions on this topic, please attend in person or in zoom.

Thank you, everybody on Zoom.

[Applause.]

Thank you, everybody.

[End of meeting]