

3Play Webinars | Redesigning for Cognitive Ease

JACLYN LAZZARI: Thank you for everyone who has just joined us for today's session-- "Redesigning for cognitive ease." My name is Jaclyn. And I'm a 3Play Media's customer marketing specialist. I'll be moderating today's session. And with that, I would like to welcome Alyssa Panetta, a web designer and developer at the University of Albany. Thank you for joining us today, Alyssa.

ALYSSA PANETTA: My pleasure, Jaclyn. Thanks for having me. So hi, everyone. My name is Alyssa Panetta and my pronouns are she and her. And I'm the web designer and front-end developer for the University Libraries at the University at Albany in New York State. I'm excited that 3Play Media invited me to do this webinar today and talk with you about redesigning for cognitive ease. Come on, slides.

So here's our agenda. I'm going to give you a quick glimpse into my new user experience and the challenges that folks with cognitive disabilities might face. Then we're going to talk about cognitive load. And then we're going to talk about making our websites easier to use through design, or designing for cognitive ease.

But first, a little about me-- I started making websites in 1999. I had a freshman internship at a math software company. And the owners asked me if I'd like to learn how to make a web page. I said, sure. And I've been doing that ever since. I freelanced for many years. And I started my current role at the University Libraries just over six years ago.

And since October of 2020, I have cognitive, visual, and motor disabilities thanks to my breakup with Tallulah, my brain tumor. This is her, as she appeared in my very first MRI. This is a scan of my brain oriented parallel to the ground. She is a large black area in the middle that a green arrow is pointing to. She takes up about a third of the space in the frontal lobe in this picture.

But for a better idea of her size, my surgeon told me that when she came out, she filled a coffee cup. He didn't say if it was an espresso cup, or something regular, maybe a Big Gulp. I'm not sure. But I was fortunate to have an amazing surgeon, other amazing doctors, amazing care, and a family that got me through surgery, radiation, and chemotherapy. And as of 2022, I have the good fortune to say that I'm OK. Tallulah is in a tumor bank in New York City. She's unlikely to grow back. And my brain is mostly healed now.

But as a result of all of this, my brain changed. And I developed disabilities, which was challenging for me as a person, but has been mind-blowing for me as a web designer because there is no better teacher than experience. So now, I have a whole new perspective on accessibility. And it's really cool to be in this unique position where I'm a web designer and I have this first-hand user experience.

And I get to share what it's taught with me with you all. Because before working at the libraries, I didn't know anything about accessibility. I had private clients. And they were never concerned with it, so neither was I. There were only two people that my sites needed to look good for. They needed to work for the client and me. And I considered myself a, quote unquote, "normal user."

I was using the most common browsers. I was on the most common operating systems. And I had the most common default settings, literally just turned on a new computer, used it as it was. I didn't think about different kinds of users or compatibility with assistive technologies. I just didn't know anyone who used them. I was young. And I was ignorant.

But working for a public institution changed that because there are laws and standards that I'm beholden to. And as a public institution in New York State, by law, our sites need to have level A compliance with WCAG 2.0. And these are the success criteria for WCAG 2.0-- perceivable, operable, understandable, and robust.

That's super ambiguous. And in 2018, with no experience, all I could go on was internet research. So my basic set of goals for WCAG 2 looked like this. I knew I'd need alternative-- I need alt text on images. I need good color contrast between the text and its background. Videos would need captions. And I'd probably have to add some ARIA tags for screen readers.

So in general, my goal was to make sure people with vision, hearing, and mobility impairments can use the site by making sure the site was compatible with their assistive tech-- so screen readers, keyboards, and input devices other than the mouse. And at that time, I didn't find any resources that had really anything to say about cognitive disabilities. I think all I found were mentions of typefaces that were bad for dyslexia. But that was about it.

And since then, I've learned a lot about disability and accessibility, as we have as a community. We no longer see disability as a condition someone has or something wrong with someone. We now see it as a mismatch between a human and an environment or a context. We also recognize that disability is very diverse and not always a permanent condition.

This is the Persona Spectrum chart from Microsoft. And it shows a matrix of examples of types of disabilities. They have touch, sight, hearing, and speaking with examples of permanent, temporary, and situational versions of that disability. So we're not just talking about people with a permanent disability of having one arm. We're talking about people with temporary disabilities, like broken arm, and a situational disability, like a new parent with a baby in one arm, or even more temporary everyday situations, like you have a sandwich or something in one of your hands, and it's messy, and you don't want to put it down. That happens to me often.

And thanks to the Center for Disease Control's report in 2018, we know that about one in four Americans have some kind of disability. And of US adults, 10.8% report having serious difficulty concentrating, remembering, or making decisions. And that's what they define as cognition. So in this chart, we can see that 10.8 is a greater percentage of US adults than those that have disabilities related to vision or hearing combined. Those only add up to 10.5.

And that statistic really blew my mind. Because here I had been focused on helping users with vision, hearing, and mobility disabilities, but I was missing this huge disability category entirely. There are other things that we know about these numbers. They're from 2018. So they're certainly higher now, especially post-COVID.

Number 2, these are underreported because these were self-reported statistics. And my favorite, number 3, cognition really involves things that happen as we age. So anybody on this call, do you have issues with your memory? You can raise a hand. I can't see it, but I'm sure you do.

Is that something that ever gets better? Has anyone's memory ever improved? No, right? So not only are we going to age, you never know when you might have a brain tumor removed and need some help. Another thing I learned is health can be changed at any time. So that's what it took for me to get this new perspective.

Until I went through this lived experience, I had never used-- I never considered users with cognitive disabilities. I had never asked myself this question, how accessible is your site for users with cognitive disabilities? How can you tell? What does that mean? And where can you start?

Now, has anybody here ever thought about this? Do you know anyone in that user category? Do you know what their life and experience online is like? You may have said yes. But if you said no, now you do because you know me. And I'm going to give you a little tour of my experience.

So within five days after brain surgery, I took one look at the screen. And oh my God, I had no idea how many issues I'd have. In the cognitive category, I faced a seizure disorder, poor memory. I had this weird choice dyslexia, where I'd want to say one thing, and then I'd say the opposite. Like I really mean to say yes, but only no we'll come out of my mouth. That was very hard to explain to people, too.

But worse than that, while my body was recovering from surgery, and then it's fighting cancer, I found myself in a constant state of fatigue. I was exhausted all the time. My vision suffered. I couldn't see as well as I used to. My eyes became super sensitive to light. I still have to wear a baseball cap and polarized sunglasses when I go outside.

And I also am very sensitive to motion. The normal speed at which my eyes used to dart around the screen, those are just way too much for my eyes now. And then because of where the tumor was on my left side, I experienced motor impairments on my right side, which is my dominant side. And the general weakness, I lost some muscle control.

So here, we have eight issues. And that's eight intersecting challenges. And have any of you ever had one of those challenges, let alone all of them at once? Well, if not, how can you know what it's like to be me and use the internet with this combination of issues? Can you test for me? How can you know how I have my system set up to use the web, and even more importantly, to do my job making it. So would testing for a, quote unquote, "normal user" that I thought I was before cover me? No.

So let's talk about my new issues. Like I said, I was in a constant state of fatigue, which increased my chance of seizure. Literally, if I thought too fast, I could have a seizure. And my doctor said that the only way to deal with that was to slow down. I had to take breaks and listen to my body. And that was really hard to do. It still is.

And at the time, I used to take a quick break by just scrolling through Instagram. But that didn't cut it anymore. That made it worse, because I'd also lost all impulse control. And I couldn't stop myself from scrolling. And it was hard before, but impossible now. I'd have to take the kind of breaks where I'd lay down for 15 minutes. And I'd put a silk light-blocking sleep mask over my eyes. And I'd have to wait about 15 minutes for the flickering under my eyelids to stop.

I also knew that there was assistive tech that could help me. But I could never find the right combination. I tried all the screen readers. I tried dark mode, high contrast mode. I tried any setting I could find, any browser extension. And none of it played nice. None of it helped.

But the worst problem-- I had to distill down what do I actually need? And the worst problem of all was how small all the text size was. It was everywhere. It wasn't just online. In print, I had to buy large print Sudoku because I just couldn't see them anymore.

So since it was a global need, I went straight to my operating system. But Mac OS makes you trade larger text for screen real estate. And that just didn't work for me, because here's a browser window with a DevTools panel open. And I have the window at full screen. But I can't even see the page I'm working on at full size.

Luckily, most of the work I do is in a web browser. So that means I can change the size of the text on any web page by relying on my browser's built-in zoom function. This is not a setting or a special mode. I just need to hit Command-Plus and it changes instantly.

So here's a page from the World Wide Web Consortium's website. It has a table of contents in the left column for navigation. There's some notes on the version. And then there's the main document. Now, I can't really read that. But when I zoom to 200, though I lose the table of contents, the text is a lot easier to read.

But when I really want to read something long, I switch to another great browser feature I found called Reader Mode. Here's the same page in reader mode in Firefox. Now, in this mode, the main content is front and center. And because this page is designed for readability, Reader Mode, all the extra metadata is gone. You don't need any of that.

And it has a really good balance of text and whitespace. And the text column itself has a limited width. So it's easy for your eyes to go back and forth, like a book. So it's very readable. And this isn't static. You can adjust the text size in the little menu off to the left. You can turn on your screen reader and save it to your pocket, which Firefox calls your something-- I don't remember.

So how many of you use a Zoom feature? I can't see your hand. So I'm just going to assume a lot of you do. It's very helpful, super convenient, and very accessible. And this is really why WCAG 2.1 introduced a success criteria for reflow, which means that I should be able to zoom in to 400% and have all the same content within my window's width. I shouldn't have to scroll horizontally.

So here's a site that I use all the time. This is called LibApps. And it has several different apps that libraries need. And this is their admin dashboard at 100% zoom. So there is a menu bar across the top. They call it a dash bar, I think. And on the left, there's a dropdown menu, where you can get to all the different apps that we have. And there's about 10 of those. And then there's breadcrumbs under the menu bar and some columns in the middle with content.

So here's a zoom to 150%. And all the menu links are still there, but they're hiding my breadcrumbs now. You can't see those. And when I Zoom in all the way to 400%, this has gone mobile. The menu bar is under the three-line menu trigger we call a hamburger menu. And if you want to switch apps, I can use the same pulldown. But I can only see about five out of the 10 apps. And I can't scroll to them. When I scroll, just the page underneath scrolls. So I can't get to the second half of those apps.

So this site would fail that reflow criteria. But luckily, I rarely zoom that much. And I just set my default to 120%. And I found that to be the safest default for me. Because sometimes when I zoom more than that, weird things happen. This is an email from my bank. I use Outlook online exclusively because I can use the zoom feature.

In this, my default 120%, Outlook is giving me half of the window for my message list in my inbox. And then the other half is for the message I have selected. But if I zoom more-- here it is at 170%-- Outlook is now giving the message the full window width. But the message inside is only taking up about a third of it. And the message text has actually gotten smaller than it was before.

So why is this happening? And this is because of CSS media queries. Media queries were invented to help us handle different screen sizes when the iPhone came out around 2007 and we needed to adapt our landscape mode designs to portrait mode. And media queries allowed us to deliver a completely different set of CSS rules. And they were great until the iPhone and Android came out in like a million different sizes.

And now, they can really muck things up like this. Because when you use your browser zoom like this on a site, the browser thinks you reduce the width of your viewport. And that triggers those media queries. So when I zoom too much, those media queries kicked in. And in this case, Outlook thinks I'm on a desktop. But the bank thinks I'm a mobile user, which is not the worst thing. But being somewhere between a mobile and desktop, you can get really confusing, especially for more complicated sites.

So now, I'm going to show you our library catalog that I work on. This is a third-party product that we license. Let me just pause that for a second. Sorry. This is our library catalog at 100%. And today, say I'm coming to find articles related to CSS. I don't want any books. My professor said I should only use recent articles.

So this layout is pretty familiar. It looks a lot like amazon.com. You've got a big search field across the top. Our results are in the center. And in the left sidebar, there is a list of options. And it says, tweak your results. And it says, availability, date, resource type. So this is pretty familiar to me. My brain likes this. And I know the left sidebar under Resource Type is where I'll be able to drill down to just articles.

So I can't really see if that's what it says. Let me play this video now at 100%. So I want to zoom in. I want to zoom in and make sure that under this tweak you know, your results that I can see that that one says article. So we'll start zooming in. And when we get to 200%-- I'm at 120% now. And when I get to 200%, I don't have a sidebar anymore. And I don't know where it went. This is not familiar to me. And my brain is not happy.

So now, I have to find where it is, because my professor said, just use articles. So now, I'm just doing this thing where I scan the page from left to right and try to find something that looks like that left sidebar. And now, I find this little icon. It looks like a funnel. And so let me try clicking on it. And then they showed up, all the way on the left. Ah, there they are.

But I didn't expect them there because there's other icons on that side of the page. And my bar was on the left. So it's out of context to me. And when I scroll, it gets caught under these other icons that I have on the right. So I really never expected that.

Now, for most people, this is a seemingly small change. All they did was they moved it under an icon. But the first time I did this, I had no idea where it was. It really did take me a long time. I'm a web designer. So I know the web term for what I was looking for was filters. And I've seen hundreds of icons that look like funnels used next to search filters. And if I didn't know that, I don't think I ever would have found it.

So the tech that we use in libraries, it's not simple or straightforward. This is a very complicated site to use. So this seemingly small change has really a giant effect for users with cognitive disabilities. It leads to an increased cognitive load, meaning it takes more effort, brainpower, and energy to use the site.

Now, since my brain is not at 100%, this results for me in an overwhelming cognitive load, which increases my fatigue, which saps my energy, which makes using the page harder. And here I am trapped in this cycle. And my poor brain just can't handle it. So now, I need to go take one of my 15-minute eye mask breaks before I can go back and try to figure out even which articles I want.

So that's a good example of my everyday experience. I don't do research every day, thank goodness. But everything I do takes me longer than it used to. And it's harder than it was before. I just cannot bear the cognitive load that I used to.

So what is cognitive load? Well, we have two kinds of memory. We have a long-term memory and a working memory. We use the long-term memory for storage. And working is what we use to do stuff.

Cognitive load only deals with our working memory, exactly how much of it we are using. And there are three forms of cognitive load. And this is according to John Sweller's cognitive load theory. There's the intrinsic cognitive load, which describes how complex the given task is. And this is something that cannot change.

For example, my online banking platform is going to have a much lower intrinsic cognitive load than a stockbroker's trading dashboard. That catalog application, that has a very high intrinsic cognitive load. Then there are extraneous cognitive loads. And these are things that add friction to your user experience-- using unnecessary distractions or design complexity, like using unfamiliar patterns.

And then there's the germane. And this connects your current challenge or information with existing mental models that are stored in your long-term memory. So we can't do anything about intrinsic. So our goal for reducing cognitive load is to minimize the extraneous and maximize the germane.

And we do that by capitalizing on affordances. We use familiar patterns that people will have seen and interacted with before. And we're not going to get too creative with our designs. Because when we can do this, we can deliver a site with a reduced cognitive load. And this is what I mean when I say designing for cognitive ease. So how do we do that? How can we design to reduce cognitive load? Where do we start?

So the World Wide Web Consortia, whose site I showed you before, has a task force dedicated to cognitive accessibility. And they have published a document called "Making Content Usable for People with Cognitive and Learning Disabilities." And these provide some goals. I have eight of them here in this slide.

And they range from helping users find and understand what they need, helping them avoid mistakes and knowing how to correct them, to using clear, understandable content and supporting personalization, all of which, to me, essentially add up to make it easier. Make your site easier to use. So there are a lot of ways to make that happen through design. But today, I just want to show you three big ways that I think are going to have the most impact.

First, we're going to add familiarity. And we're going to use semantic HTML to make it easier to find and understand what the user needs on our site. We're going to add flexibility with modern CSS to make our sites adaptable and support personalization. And then we're going to improve clarity by saying important things twice. And that's in different ways to ensure that our content is clear and understandable.

So to maximize the germane and minimize the extraneous, we're going to want to stick with what's familiar to our users, what's stored in their long-term memory already. So we'll rely on some classic layouts. And we'll make sure that they are conveying structural meaning. And we'll capitalize on a common habit that humans use to read by using clear and semantic headings.

So step 1 to helping users with cognitive and learning disabilities is really to adopt best practices for accessibility in general. And one thing we know about accessibility is that the earlier it's included in your design in process, the easier it is to do successfully. CQ, an accessibility company, did a study. And they found that 2/3 of all accessibility errors originate in the design process. So that's the best place to start. This is the idea of shift left that you may have heard about, which just means build it in as early into your process as you can, which is a big deal, right?

You may be freaking out if you hear, oh, I have to go-- it needs accessibility help. You have a site. It needs accessibility help. And you just got scared because you can't imagine going all the way back to the very beginning of the design phase to fix things. A lot of people in that situation get tempted to throw an accessibility overlay on their site that promises to fix everything for you.

But those are no good. There's lots of lawsuits against them because that's treating accessibility as an afterthought. And when we do that, we get something like this. This is a real photo I took on my campus. This is of a rickety-looking aluminum ramp about the width of a wheelchair. It's laid across a set of four or five really shallow concrete steps that lead from a concrete path down to a grassy area that's surrounded by trees.

The ramp has handles on either side. But the ramp doesn't meet the stairs in all the right places. So they've had to shim some rotten chunks of wood in there to make it meet. So I'll ask you-- you're the architect of this campus. And you know, when you're drawing your plans, that people are 100% going to need wheelchair access. Is this what you would design and build? And if you use a wheelchair and you want to get to that grassy area, does this ramp look like something you want to use? Does this make you feel welcome?

And if you don't use a wheelchair and you're just hanging out in this grassy area, is this ramp something you want to be looking at? No, right? It's kind of ruining the space for everyone. But at least this ramp is still usable. Someone in a wheelchair can go down it or up it.

But in the digital space, one of those afterthought overlays, they're not only useless to the browser and assistive tech like screen readers, what we call user agents, they actually get in their way. Because screen readers start with the HTML. They specifically start with semantic elements.

Semantic means relating to meaning. And a lot of HTML tags have meaning and tell the browsers, these user agents like screen readers, what they're for. There's header. There's a nav. There's a main. There's footer. These provide landmarks for your page. There are other elements like divs, spans, paragraphs. Those are just containers. They don't tell a screen reader or assistive tech anything about what is inside them.

So here I have a diagram. And we see those semantic tags inside of what's called the Holy Grail layout. You've got a header across the top of the whole page. You've got the branding on the left side and a navigation element all the way to the right. Then you have, in the next row, you have a main content in the middle broken into sections. And then there are sidebars on either side. And then there's a footer across the full bottom width.

And this is such a familiar layout. You can rely on it almost every time. So this is really good at maximizing that germane cognitive load.

Meaning also applies to your heading. The structure of headings is one of the most important semantic details. Headings make your text content more navigable, both for sighted readers and screen readers.

One of the most common habits we share is how we read things online. Most of us use what's called an F-shaped reading pattern, which means we scan the first line of a paragraph or block of text, and then scan down the left side for the next. And you just read the first line. So if you use headings properly, when your visitor reads in that F pattern, they'll get the info that they need.

So here are some rules about headings. They should be easy to scan and meaningful. You can only have one H1 on a page. And headings are not for making non-heading text bigger. I used to do that. And it's not good. So this slide is not an ideal format for a web page, is it?

So let me show you a better example. This is much easier to scan, right? Your eyes are probably just hitting those headings. And in the last paragraph, you can see why we don't use headings for non-heading text. It's confusing, isn't it? Yeah, I used to do this. I didn't know better. But now, if you can't read it, it says, it's confusing if you start yelling in the middle of this paragraph, isn't it? But now, I know that if we want our text to be bigger, we use CSS.

Specifically, I like to use modern CSS, or what is sometimes called intrinsic design. And this is where we give the browser enough information to decide itself when the layout will shift instead of hard-coding things ourselves. Old CSS, premodern CSS, was about making things pixel-perfect. It was about making it look exactly the same for every user. But that has changed. And it's for the best.

Modern CSS is all about flexibility. And it's exciting to work with because all of this is now done with really minimal code. This new approach reminds me of how skyscrapers and bridges are designed to withstand high winds. So architects know they can't beat the wind. And if their really tall structure is too brittle, it's just going to snap and break. So they design them to move with the wind instead and be flexible. They build them out of flexible steel instead of stone and concrete that can't bend.

So this is my new approach to designing my websites. And I like to think of them as web pages or as bridges or skyscrapers. And I don't know how windy it's going to be for each user. I can't know how they set up their system, just like you can't know mine. But I know that my design, if it's flexible, it will adapt to their system, no matter how windy it is.

So I try to embrace this uncertainty. And I get comfortable-- it's really hard, by the way, because I've been doing this for a long time. And I try to get comfortable with the fact that the user controls the visual experience of my site as much as I do, if not more. So when it comes to bang for your buck, CSS is growing huge. And there's all kinds of new features almost every day it feels like.

There are some modern CSS features that I think are going to have the most impact for you. So these are using relative units for text and whitespace, which are going to greatly improve readability, using CSS Grid, Flexbox, and Math for your layouts. These are real layout tools. They're not hacks like we used to use-- designing web pages using tables, floats. These are real layout tools. They're awesome.

And then, instead of just relying on media queries, which are still useful to some extent, we're going to use new preference queries. We're going to use new queries for preferences and containers that are drastically going to help us. So I'm going to just try to give you a brief overview of these. But I am not a CSS expert. But I do have resources in my talk notes that have resources, references for all of this stuff. You can dig in real deep.

So talking about readability, again, which has giant effect on cognitive load and staying flexible for users, you don't have to work with pixels. We now have relative units in CSS. These are rem, em. You can use a percent. You can use a view width, a view height, a view minimum, view maximum. And there's several others that are cool. Again, you can look at the-- there's a link on-- the link at the bottom of this slide goes to my GitHub page, where I have all of these resources. And that goes specifically to the part about using these, using type and spacing to flexibility.

Next up, there's CSS Grid for layouts. Grid is the best layout tool CSS has ever seen. It has special superpowers. You can define columns and rows. You can use relative and flexible units, including a special fr unit, which just stands for fractional width. So you can say, repeat to-- repeat the columns and do 2 at 1 fractional unit. So it'll just keep making columns until-- it'll just do two. And it'll do them-- yeah, sorry, not thinking about math right now.

So you can make them repeat. You can use a gap. You don't have to do margin and padding hacks anymore. You can define templates for columns and areas. And like I said, my favorite thing is that it can repeat things. I am still a math nerd. So some of these math functions are some of my favorite recent CSS features.

Specifically, there's calc, which performs basic math operations. But what's really cool is you can mix units. So I use this whenever I have a bootstrap framework site, which is all pixel-based. And I want it to go relative. So I have to work with both. And generally, what I can do is I can like calculate the view width. And I can subtract a pixel.

Clamp is another really cool math superpower. Clamp lets you define a range. And then you set a minimum, you set a maximum, and an ideal value. You can pass all kinds of math in there too, which there are some examples. I've been like, I don't even know what any of that does.

But luckily, if you're not into math, you don't need to worry about it, because some people who are. And we have computers that do this kind of stuff for us. So there are some really smart CSS folks that have made tools for us. And these can calculate type scales and other things, clamp units.

And the one I like best is called Utopia. It's utopia.fyi. And they say this provides fluid responsive design without using breakpoints. And breakpoints are those media queries. And these go beyond type. You can calculate space, grid specs. And they just released a clamp calculator, which is really cool. Basically, you tell it what width, your minimum width and your maximum width. And then it gives you a string of CSS variables to implement. It's cool. Check it out.

Preference queries, these are the new queries in CSS. Preference queries are what they sound like. When your user sets a preference for something like reduced motion or dark mode, windows in high contrast or forced colors, you can change the CSS based on their preferences they have, like you would for a viewport size.

There's a link at the bottom of this slide, which I highly recommend checking out, from Carie Fisher, who is an accessibility pro that I follow a lot. And this is called "CSS and Accessibility Inclusion Through User Choice." And it's all about these preference queries. So that's a great article.

And then we have container queries, which are like media queries that you define within your page. So now, you can have elements that change their layout depending on where they're placed on your page. Like if they're in the main body, which has an 800-pixel width versus in the sidebar, which may max out at like 200 pixels, you can change the layout based on that for one component-- very cool.

So here is an example of changing the font size based on a container width. Remember when I said modern CSS does all of this with minimal code and how cool clamp can be? Here, another accessibility CSS genius was-- he was exploring container query units. And he was using clamp. He was able to reduce four different container query declarations down to one. He got 18 lines of CSS, 21 including the comments, down to just three lines, which is really cool, very efficient, and much easier to read.

Moving on to clarity-- another technique I think is really helpful is saying it twice. It can be a really quick, effective upgrade. By providing your users with multiple modes of understanding, you increase the chances your users will understand what you mean. Your user may not speak your language. So adding an icon can help them, can provide them some additional context about that work. It's simple, but it's a very effective upgrade when you do it in your menus and your buttons.

You can go beyond color and add different styling effects. Beyond color blindness, there are cultural differences, like red doesn't mean stop everywhere. So you can't rely on color alone for meaning. You can also add motion, which helps grab the user's attention, like letting them know when they're hovering over a link.

So here's an example of adding icons to text. This is GitHub's Account Settings page. And next to my username, there is an option for me to use more than one account. So next to that link, they've got a pair of arrows that point in opposite directions. And over in the left navigation menu, they've got an icon next to each link. There's a person next to the public profile link, a gear for the account. There's a bell next to notifications. So that's an easy upgrade to do.

Going beyond color, here's a form for our interlibrary loan request that I recently upgraded. As suggested in the "Making Content Usable" document, if there are errors, we're going to make sure that the user knows what they are and how to correct them. So this is a form that gets submitted all the time. And we wanted to make sure that the error message is at the top and that it stands out. So we did that.

And then each of the fields that has an error was also in red. And it's highlighted on the left-hand side with a thicker, darker border because here's that form in Windows High Contrast mode. Everything that was the background that was white is now black. And what was red is now bright yellow. It's still easy to spot the errors.

Another way to say it twice is to add motion, especially on hover, because motion brings things to our attention. It pulls something in our peripheral view into focus. So if you want your users to know that your page heard your mouse movement and are hovering over the right link, you can help by using motion.

You can use CSS to do some basics. There are transforms like scale, rotate, translate, skew, and transitions to change things like the background color and make it slide from left to right, like on this link. If you can see it, it's pretty clear you've hovered over it, right?

Here's a bigger motion example. I have two calls to action at the end of all of our blog posts on the library website. And they are next to each other. So to highlight which one you're hovering over, I use a scale transform in addition to the color change. Just a little motion, but it's enough. Here's what that looks like in Windows High Contrast. No problem.

So where do you start? Well, you've got a lot of options. These are eight possible upgrades you could start with. Again, if you have the presentation downloaded, this slide will link you directly to the resources on my GitHub page. It has links for best practices, how to do it. And for some, I've linked to tools and utilities I like to use for things, like Utopia. There's some guidance there.

But for right now, so these are what we talked about before-- convert type and spacing to flexible units, convert your page layout to CSS Grid and Math, add preference queries, migrate from media queries to container queries, clean up your headings-- that's a good one-- add text to icons, icons to text, add those transforms where only color changed before.

But I want to just show you this last one, how to improve a micro-interaction, like a navigation switch. So we looked at one of these before. You probably have one on your site because they are everywhere. This is the three horizontal line icon that we call a hamburger menu. And this one is in a button. And the button has a border that's the same color as the icon.

So I was telling you before, if you want to say it twice, just add a text to that icon. So my first instinct to tell you to fix this was, oh, you just add the word menu. But then I found a conversion rate report where someone compared four versions of the hamburger menus mixing icons and the word menu within a button. And my recommended version only performs second best.

The best version was just the word menu in a button outlined with the same color as the text. But what if that outline isn't the same color as the text? Well, that actually makes it the worst performer. That's the last, just the word menu in a button with a very lightly colored outline where you almost can't see the outline. That performed worse than just the hamburger icon on its own.

So what's important here is that the color of the outline matches the color of the word. That is what we recognize in our long-term memory as a button. And we know the button means click me. This is our affordance. It's for buttons. It's not the icon at all. So a button means click me. So go ahead and ditch the icon here and just replace it with the word.

So back to my interlibrary loan application that I just upgraded, this is what we had on mobile. It's just a hamburger icon. And it's in a button. But it has that very low contrast outline, so the worst performer. So I made that upgrade, replaced it with the word menu. And I work with a user experience librarian. And she knows what librarians need and students need.

So we pulled out the most important actions and gave them their own menu links. Because we know if our patients are concerned with their interlibrary loans while they're on the go, they're going to want to get to the good stuff fast. So we reduced that cognitive load and gave them one less tap or click.

So I have a CodePen put together with the code that we used here. And there's a link here and a link in the slides and the GitHub page to get to it as well. So feel free to steal my code and make it work on your site. So all right, let's recap.

I gave you a quick tour of my experience using the web with a cognitive disability. We talked about cognitive load and about making websites easier to use through design. We looked at these three major ways to do that. We're going to add familiarity through semantic HTML. We're going to make our sites flexible with modern CSS. And we're going to add clarity by saying important things twice in different ways.

So when you leave and you think about how accessible your site is for users with cognitive disabilities, I hope you have a better sense of their possible challenges. And if you have a hamburger menu and the power to do something about it, I hope you take that opportunity and give it that little upgrade. I've heard from some people who have viewed this talk that they've done it. And they were very happy about it. So I hope that makes your site easier for your users, those with and without cognitive or any other type of disability.

So that's it for me. I thank you all so much for being here today. There's a link to that special GitHub page, where I have all the links and valuable resources for performing most of the recommended upgrades that I didn't get a chance to show you. Feel free to reach out to me any time with questions, comments, recommendations. Always happy to talk about cognitive design accessibility. Special thanks to all of you for being here, everybody at the University Libraries that I work with, the team here at 3Play Media, all my health care pros, my family, a lot of thank yous, and of course, Tallulah. So thanks.

JACLYN LAZZARI: Thank you, Alyssa. We're getting a lot-- you're getting a lot of claps, hearts, and emojis. So I think it's safe to say the audience enjoyed your presentation, as well as me. We do have some questions and a little bit of time left, if you're ready to dive into those.

ALYSSA PANETTA: Sure. Should I stop sharing my slides?

JACLYN LAZZARI: Yeah, you're welcome to. Or you can leave it up, whatever. Yeah, that's perfect.

ALYSSA PANETTA: I'll go back if anybody needs anything.

JACLYN LAZZARI: Absolutely. Since it's kind of fresh in our minds, as you were just talking about it, someone asked, in referencing the hamburger conversion metrics, where did you find these conversions? Or how did you find them? Their team would love to show the data this way.

ALYSSA PANETTA: That one, I have a link to in that GitHub page, which I can put in the chat real quick, once I just dig it up. GitHub. Give me two seconds. Do you have another question?

JACLYN LAZZARI: Yes. Someone else asked, does WCAG require a certain font size minimum beyond being able to zoom distinguishably? Or do you recommend a minimum size?

ALYSSA PANETTA: I try not to go under 14 pixel size. And that's for like the smallest text. I find 16 pixels, which is usually one em or one rem, to be like my base font size. But I never go under 14. People do. But if you need to, if you really need to, it's cool because people can usually zoom. But yeah, would I wouldn't go under 16 as my default.

JACLYN LAZZARI: Great. And another question-- do you have advice for balancing conflicting accessibility needs, since cognitive disability is a wide spectrum? For example, some people may need very high color contrast to be able to read the text, but others may be-- or they may also be bothered by the contrast that is too high.

ALYSSA PANETTA: Again, this is where those preference queries come in really handy. Because the user should be able to set what they want for theirs, for their setup. And whatever they're comfortable with, that should be-- you should give them the power to change those things themselves. If you want to, you could give them-- I've seen people give toggles between dark mode and light mode. You could do things like that. But yeah, the preference queries is really my number one suggestion on that. Does that answer the question enough?

JACLYN LAZZARI: I think so, yes. That's great. Do you have advice for users with motion sensitivity and navigating websites that aren't designed with cognitive accessibility in mind? Things moving on the screen make me dizzy. Are there other things I can do on the user side to reduce that motion?

ALYSSA PANETTA: If you haven't already set your preference to reduce motion, I think it's usually an operating system level preference. Set that. A lot of sites, I think some browsers will respect that and stop animations. But really, there's no way to stop certain things. And I have that issue myself. So it can be very frustrating when sites don't adhere to that. Yeah.

JACLYN LAZZARI: Yeah, absolutely.

ALYSSA PANETTA: I feel for you. Let me pop in-- oh.

JACLYN LAZZARI: Yeah.

ALYSSA PANETTA: You popped in that. Yeah, that's the link to-- yeah, to the-- yeah. Yeah, turning off JavaScript can help. But now, there's a new thing coming out, scroll animations in CSS that people are working on. It's going to be a thing where-- I'm pretty sure the browsers are implementing it with a reduced motion query in it. But people are going to start going nuts with it because they're excited with these scroll animations.

JACLYN LAZZARI: Yeah, definitely.

ALYSSA PANETTA: Yeah, with or without JavaScript, you'll still see them.

JACLYN LAZZARI: And another question, for those of us new to HTML and coding, are there templates you have used to get started towards the goal of creating a more accessible site that we could also refer to?

ALYSSA No. Very simply, no. The way I learned my HTML-- and this was 20 years ago-- was looking at sites I liked and

PANETTA: was comfortable using, and then just looking at the source and copying and editing. There are frameworks now. I started doing this before there were frameworks.

Every framework has its issues, though. So it's hard to say what's a good one to use. It also depends on a lot of things. A lot of HTML sites are not HTML first. They're built in JavaScript frameworks that then make a mess. But yeah, I would start-- I would refer to the MDN Docs. Those are great places to start with just basic HTML. They have lots of-- lots of tests and little code bits for you to use.

And then I know there's different layout ones. Layout Land is a good one. Actually, there are some in my page here. There's a whole section on HTML. So yeah, there's some. Here, I'll put the link here into chat again. So this is straight to the part on HTML. So there's a link to a good course on just learning HTML and a couple of them, the W3Schools.

Jen Kramer is a great teacher of HTML, the 15 days of HTML. And she has a paid course. Oh, Frontend Masters is a really good one. I should add that back in here. Yeah, Frontend Masters is also a great HTML-specific starting point.

JACLYN Great, wonderful, lots of great resources. A couple more questions, and I think this one came up while you were

LAZZARI: talking about the clarity piece. So someone asked, would providing a video option of, say, transportation options be a worthwhile investment on a website?

ALYSSA I think any way you can disseminate information in different modes is helpful. The only problem with that is you

PANETTA: have to keep all of those modes up to date, right? So if you change something in the text, then you have to go edit your video and change that, too. So as long as you are aware and willing to do that, yeah, video is great. Yeah.

JACLYN Great. I think that's all the time we have for questions. We got through quite a bit. So again, Alyssa, just want to

LAZZARI: thank you so much for your time and your wonderful presentation today. And thank you to everyone for joining us, and asking great questions, and engaging with Alyssa today. And thanks again to everyone. And I hope you all have a great rest of the day. Thank you again, Alyssa.

ALYSSA My pleasure. Thank you, everybody.

PANETTA: