CTRL+CLICK CAST #61 Flexbox with Zoe Gillenwater

[Music]

Lea Alcantara: From Bright Umbrella, this is CTRL+CLICK CAST! We inspect the web for you! Today we are talking about flexbox with special guest, Zoe Gillenwater. I'm your host, Lea Alcantara, and I'm joined by my fab co-host:

Emily Lewis: Emily Lewis!

Lea Alcantara: This episode is brought to you by Craft Commerce, a brand new ecommerce platform for Craft CMS. If you're a web shop that likes to create custom-tailored websites for your clients, you're going to love Craft Commerce. It's extremely flexible, leaving all the product modeling and front-end development up to you, and it's got a simple and intuitive back end for content managers. To learn more and download a free trial, head over to craftcommerce.com.

[Music ends]

Emily Lewis: If you are a front-end dev, you know the pains of building a flexible layout, especially if you've been working with CSS for a long time and can remember the old days. [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: Today we're going to talk about how CSS layouts have evolved to give us the flexible layout model, a.k.a., flexbox. Helping us navigate this brave new world is Zoe Gillenwater. Zoe is a senior designer for booking.com in Amsterdam. She's the author of two books on CSS and visual web design, and last year, wrote the chapter on flexbox for Smashing Magazine's new *Real-Life Responsive Web Design* book. Welcome to the show, Zoe.

Zoe Gillenwater: Hey, thanks for having me.

Lea Alcantara: So Zoe, can you tell our listeners a bit more about yourself?

Zoe Gillenwater: Sure, so like Emily mentioned, I live in Amsterdam. I am not Dutch. I'm American, but I've lived here about two years just for a little European adventure. I have two kids and a husband that are on the adventure with me so I feel like my kids are way smarter than me, they're bilingual now. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: And I only speak English.

Lea Alcantara: Cool.

Zoe Gillenwater: And yeah, so that's me.

Emily Lewis: Amsterdam is one of my favorite cities I absolutely love. I've been there a number of times. I think it's beautiful and you don't have to have a car if you're on vacation. You can really walk around the whole city and it's got great food and great museums. Was it for the job that you went to Amsterdam or was it the adventure and then you got the job?

Zoe Gillenwater: It was for the job. I originally saw the job on LinkedIn and at first I mentioned it to my husband jokingly like, "Oh, wouldn't it be fun to live in Europe. Ha-ha-ha." [Laughs]

Emily Lewis: [Laughs]

Zoe Gillenwater: Like nobody really does that.

Lea Alcantara: Right.

Zoe Gillenwater: But then looking into it more, looking into the company and everything, it seemed like actually this is a really good sounding job and it would be kind of a cool opportunity for us and for our kids to experience new things, and so yeah, so I applied for the job and they are the ones who brought me out here. But yeah, we were looking for kind of a change anyway, so this fit really well.

Emily Lewis: It's exciting.

Lea Alcantara: Very cool. That's a similar story with my husband and I because I'm from Canada and we were ready for our own adventure, but a smaller jump from Canada to the United States. [Laughs]

Emily Lewis: [Laughs]

Zoe Gillenwater: Yes, yeah, yeah.

Emily Lewis: All right, so before we get into flexbox, Zoe, I wanted to first chat a bit about the history of CSS layouts. Years ago, you wrote *Flexible Web Design* and discussed techniques for flexible layouts. Can you describe what flexible layouts were then, like what that meant to us back in the day? [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Yeah. Basically back then, there were kind of three terms for flexible layouts. There was liquid, elastic and hybrid, and liquid basically just meant that your unit of measurement was percentages, so your layout would adjust with the browser viewport. Elastic was when you used *ems* as your unit of measurement.

Lea Alcantara: [Agrees]

Zoe Gillenwater: So this was before we had *rems* and viewport units, but you had *ems* that could allow the layout to adjust to the content size.

Lea Alcantara: [Agrees]

Zoe Gillenwater: And then hybrid could be a mixture either of those two types or a fixed layout with a liquid layout, so you might have a fixed width side bar in pixels and a percentage main content area.

Emily Lewis: [Agrees]

Zoe Gillenwater: Which is kind of the trickiest type of layout of all to achieve, and still is pretty tricky actually today.

Emily Lewis: So how have things evolved to get us to where we are today? Like how did we move from those liquid, elastic, hybrid models to the flexbox? Well, I guess maybe we should describe flexbox and then describe the evolution. So what is flexbox? So a high level summary of it.

Zoe Gillenwater: It's basically a layout system that gives you a lot more control over alignment and order and proportional sizes of boxes so that the boxes can be more flexible to the space available as well as to the content within them. So unlike a lot of the other layout methods that we've used in CSS in the past, this one was actually designed to be an actual layout system in CSS directly.

Emily Lewis: So the techniques we used in the past were just leveraging what CSS could do at the time, but it wasn't necessarily for layout, but it was just how we were applying different techniques.

Zoe Gillenwater: Right. The CSS too, other than positioning, which turned out to be too rigid to use for your full-page layout, other than positioning, there wasn't really anything in CSS too that was designed specifically for layout, either kind of component layout or overall page layout. So now, with flexbox and some other layout models, but with flexbox, it's an actual system designed to handle the sort of layouts that have become common on the web and that web designers want to be able to accomplish with CSS.

Lea Alcantara: Do you think the mobile design/mobile web movement pushed something like flexbox faster into the consciousness, the front-end developer consciousness?

Zoe Gillenwater: Yeah, I definitely think that's the case. I feel like when first wrote *Flexible Web Design* in 2008, there was kind of a small group of people who were interested in liquid layouts and flexible layouts at that time, but a lot of people didn't really understand why it even mattered.

Lea Alcantara: Right.

Zoe Gillenwater: Everybody was just kind of designing for 1024 x 768.

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Zoe Gillenwater: And they were satisfied with that, and so having all of these different mobile devices come in with all of these different screen sizes finally convinced people, "Oh yeah, I do have to design for different sizes and I have to make my websites flexible in some way, and the tools that I have now are not very good at that.

Emily Lewis: Right.

Zoe Gillenwater: So I think that did probably drive the development of flexbox sooner than it might otherwise have come about.

Emily Lewis: Was flexbox something that was kind of an experiment and then became part of the specification, or it just became part of the specification and people started experimenting with it?

Zoe Gillenwater: It's been in development for several years, and it's gone through a couple of revisions in the specification. So it was something that people have been working on for a long time, but I think there were also a lot of experiments that real designers and developers were doing with it that led to a lot of the changes in it.

Emily Lewis: [Agrees]

Zoe Gillenwater: So the initial version of it wasn't as robust as it needed to be to handle the real-world sort of layouts that people wanted to create, and so they made some changes in the spec that solved some of those issues, and so it's a lot more fully featured now to really fit with how people want to use it.

Emily Lewis: You know we talked a little bit about just layout in general and mobile layout and the necessity for that to be flexible, but another use case in a way that occurs to me, I saw Eric Meyer give us a brief flexbox kind of primer a couple of years back, and you can like make equal height columns.

Zoe Gillenwater: Right.

Emily Lewis: Is that correct?

Zoe Gillenwater: Yeah, absolutely.

Emily Lewis: Which is kind of mind boggling if you've never, you know. You can't do that, I guess, unless you're using flexbox or some sort of forced height value or JavaScript to determine height.

Zoe Gillenwater: Right. It seems like such a simple thing visually.

Emily Lewis: [Laughs]

Lea Alcantara: Right.

Zoe Gillenwater: But it's been so hard to accomplish with CSS. So yeah, you could use flexbox on fixed-width layouts still, but you can use it to make equal height columns or you can use it for vertical centering, or all those sort of, yeah, good things.

Emily Lewis: So let's talk a little bit more about sort of the use cases or more generally, the benefits of flexbox.

Zoe Gillenwater: Okay.

Emily Lewis: So you mentioned having the ability to like center vertically, and if you're not basically in a table, you kind of can't center vertically. It's very challenging to do so.

Zoe Gillenwater: Right. It's super hard to do if you don't know the height of the content that you're dealing with.

Emily Lewis: And so flexbox gives us a way. I mean, if it's not too difficult, can you share like what the syntax sort of looks like, but verbally.

Zoe Gillenwater: Yeah. You know, actually I can because it is super simple with flexbox. So basically you set *align-item: center;* on a container.

Lea Alcantara: Wow! [Laughs]

Zoe Gillenwater: Yeah. [Laughs]

Emily Lewis: [Laughs]

Zoe Gillenwater: That's it. So if you just have container, you say, "I want this to be displayed flex and I want it to have *align-item: center;* and then the children inside will all be vertically centered with each other. Done." I mean, it's logical, like that's how it always should have been. It should just have been a property, like set vertical centering on or off," you know?

***Timestamp**: 00:10:01*

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Zoe Gillenwater: So that's how it is now with flexbox, which is awesome.

Emily Lewis: And applied to like a mobile experience, what kind of benefit does it bring as opposed to a more traditional approach to establishing the layout of the page?

Zoe Gillenwater: Well, it's really good for responsive web design because the boxes can change size automatically without you even needing media query sometimes, so it just is based on when there's enough room for them to switch, they can change layout or resize and you can even rearrange boxes to some extent with flexbox, which can also help when you're making a mobile version of the layout, and you might want to have the content in a different order than it might otherwise appear in its default stacking order in the HTML.

Emily Lewis: Oh, cool. So it's kind of a way to subvert the source order a bit?

Zoe Gillenwater: Yeah, you can't put anything anywhere, like you can with the grid layout CSS.

Emily Lewis: [Agrees]

Zoe Gillenwater: That gives you kind of complete carte blanche to move stuff wherever you want. With flexbox, you don't have quite that level of flexibility because you can only move things within the same container. You can only move siblings relative to each other, but you do have some ability to reorder things. You also have to be careful with accessibility though when you do that.

Lea Alcantara: Right.

Zoe Gillenwater: You're only going to be reordering things visually, so the screen reading order and the tabbing order is not going to change. So it's good for doing reordering where you might be tempted to put something in an illogical HTML order just because you need it to make floats work or something like that.

Emily Lewis: [Agrees]

Zoe Gillenwater: So instead, you can put things in the logical HTML order and then use flexbox to move things around a little bit.

Emily Lewis: Nice.

Lea Alcantara: You know what, it actually occurs to me, Emily, would this technique have helped with our particular client that had fixed-width ads, but the rest of the layout had to be responsive?

Emily Lewis: I know it's possible. With me having never touched flexbox, I couldn't say.

Zoe Gillenwater: Right.

Emily Lewis: But Zoe…

Lea Alcantara: But conceptually?

Emily Lewis: Yeah, I mean, Zoe, have you ever worked in a situation like that using flexbox where there were elements that had to be fixed widths and heights, but the rest of it had to be responsive or flexible?

Zoe Gillenwater: Yeah, that's one of the best use cases for flexbox, I would say.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Zoe Gillenwater: It's allowing having some fixed-width pieces of content or some pieces of content or the box size is driven by the content width, and for instance, like a form label.

Emily Lewis: [Agrees]

Lea Alcantara: Yeah.

Zoe Gillenwater: So that the form label you might want to be just as wide as the text inside it needs it to be, and then you might want to input that beside it to just take up all the space left, and it could be the same with the layout with the fixed-width ad so you have the ad that you say is 200-pixels wide and then you want the content beside to just take whatever space is left. That's a really good use of flexbox. So it's kind of like the hybrid layouts that I mentioned before, that flexbox is making that sort of layout much, much easier.

Emily Lewis: Yeah. It's one of those things where I'm like, "Okay, I've got to get into the code and write to actually see it working. [Laughs]

Lea Alcantara: Right. [Laughs]

Emily Lewis: But yeah, because when you're even just trying to get two things to align, most of the time, both things need some sort of value for their width in order to get them to fit within a given container.

Lea Alcantara: Yeah.

Emily Lewis: So it would be nice for it to just be flexible based on one thing, a specified width.

Zoe Gillenwater: Right, yeah.

Emily Lewis: So it sounds really cool, but what are we looking at in terms of support?

Zoe Gillenwater: Basically, you have everything except for IE 9 and earlier versions.

Emily Lewis: Oh.

Lea Alcantara: [Agrees]

Zoe Gillenwater: So IE 10 supports a prefixed older version of the spec, the properties basically work the same, but they have different names with some of them.

Emily Lewis: [Agrees]

Zoe Gillenwater: But yeah, all current browsers support it without prefixes. Even if you go several versions back on most of those, you still get support with prefixed versions, so browser support is actually really good now.

Emily Lewis: Wow, you know it just proves how fast things move because just a couple of years ago it wasn't as supported. So when we're talking about using flexbox, it's pretty viable then right now for production level sites.

Zoe Gillenwater: I think so. We use it in little bits and pieces on booking.com. I've used it here and there, yeah, on production-level code, and I'm always using it though as progressive enhancement because we do support back to IE 7 at booking.com.

Emily Lewis: [Agrees]

Zoe Gillenwater: So you can add flexbox as progressive enhancement and still have an acceptable experience for those older browsers that don't support flexbox.

Emily Lewis: So how are you doing? Is it simply a matter of the order of your selectors in your CSS?

Zoe Gillenwater: For the most part. So flexbox will override most other layout properties automatically.

Emily Lewis: [Agrees]

Zoe Gillenwater: So if you add first a float and then you add flexbox, the flexbox will take precedence and it will just be ignored if the browser doesn't understand it.

Emily Lewis: Okay.

Zoe Gillenwater: So that makes it really easy to just kind of lay your flexbox on top.

Emily Lewis: Right.

Zoe Gillenwater: There are some things that if you have certain margins that you need to make floating work, but you don't those margins to apply in the case of the flexbox version, that's where you need to start selectively feeding styles to only one set of browsers or do feature detection or whatever is your chosen way of separating out styles for old browsers, but in a lot of cases, you can just add the flexbox right on top. It just overrides the other layout with CSS that you have in place and it gives that new improved layout to the browsers that understand it, which like I said is almost all of them.

Emily Lewis: For browser detection, what do you use?

Zoe Gillenwater: Well, here at Booking, we have our own custom thing. I have used Modernizr on some personal projects, and I like it a lot. It also was updated. The Version 3 I believe of Modernizr that's out now has some new flexbox checks so it can check not just if flexbox is supported or not, but it can check if the flex-wrap property is supported or not as that was something that kind of lagged behind some of the other properties in certain browsers. So yeah, I like Modernizr a lot. You can use the @supports rule in CSS as well.

Emily Lewis: [Agrees]

Zoe Gillenwater: But the issue with that is that it's not supported in IE 10 and possibly 11, I'm not certain about that, so that it wouldn't allow you to feed flexbox rules to IE 10, so that's why I'm kind of avoiding that. For right now I'm sticking with Modernizr.

Emily Lewis: You mentioned in the middle of that that flexbox wrap.

Zoe Gillenwater: Yeah.

Emily Lewis: Did I hear you right?

Zoe Gillenwater: Yeah, the flex-wrap property.

Emily Lewis: What does that property do?

Zoe Gillenwater: So that property says whether or not your boxes are allowed to wrap and in what direction they wrap. So they could wrap from bottom to top, for instance, but the default is for all items to stay on the same line, which means row or column, no matter what. So it kind of takes care of that issue of float wrapping that it sometimes made. Float layout is really difficult when you have a float or something after the float dropping down and you didn't understand why it was dropping down.

Emily Lewis: [Agrees]

Zoe Gillenwater: And it was because of one extra pixel somewhere. So with flexbox, you can say, "No, you always have to stay on the same line or yes, you are allowed to wrap."

Emily Lewis: Oh, I love that.

Lea Alcantara: It's interesting because Zoe sent us a few links to just read up flexbox and resources, and one of the awesome ones I think is this flexbox cheat sheet because it like visually shows you what the CSS would look like and then like in little icons and illustrations how that will translate in the layout, and that flex-wrap concept, I'm just looking at that and I'm like my brain is exploding because there's a wrap-reverse concept. [Laughs]

Zoe Gillenwater: Yeah.

Lea Alcantara: And I'm like, "What?" [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: I've only ever seen for years and years and years, it's always like, "Okay, the last element goes to the next line as always forever and ever."

Zoe Gillenwater: Yes, yeah.

Lea Alcantara: And now I'm like, "What, there is a different way if you want that?" [Laughs]

Zoe Gillenwater: Yeah, that's the thing. It was designed to work like with any sort of language, any writing direction and flow content in any direction, which I think that's one of the reasons why a lot of people are kind of overwhelmed by it at first.

Lea Alcantara: Right.

Zoe Gillenwater: It's because the names of the properties, it's not like float left. That's really clear, like something is going to go to the left, you know?

Lea Alcantara: Right.

Emily Lewis: [Agrees]

Zoe Gillenwater: But when it's like saying, "*align: flex-start;*" you're like, "What is flex-start? What side is that?"

Lea Alcantara: Right.

Zoe Gillenwater: But it uses those terms on purpose so that it can be worked with any sort of language and any writing direction.

Lea Alcantara: Right.

Zoe Gillenwater: So seeing it visually like you were talking about it in those diagrams or just playing with it yourself like in your developer tools in your browser, changing the value and seeing immediately how it updates, that's really helpful to make sense.

Lea Alcantara: I like that because I feel like that's a lot more friendly for just general discussion, like do you know the concept when we're writing classes, you're not supposed to be so prescriptive so like it's not going to be blue or whatever, right?

Zoe Gillenwater: Right, yeah.

Lea Alcantara: Like you talked about the description. So here it's like flex-start, it doesn't specify like a specific position per se.

Zoe Gillenwater: Right.

Lea Alcantara: Like left or right so it's a lot more flexible.

Zoe Gillenwater: Yeah.

Lea Alcantara: What also seems to occur to me is that these types of options allow for better international sites.

Emily Lewis: [Agrees]

Lea Alcantara: Because I feel like in the Western world, we're always going left to right up and down.

Zoe Gillenwater: Yeah.

Lea Alcantara: While a lot of Asian, Chinese or Japanese characters, it's up and down right to left.

Zoe Gillenwater: Yeah. Arabic and Hebrew are right to left.

Lea Alcantara: Right.

***Timestamp**: 00:19:56*

Zoe Gillenwater: Like yeah, there are a lot of different variations and the flexbox, because it has those kinds of neutral terms, it just automatically switches things for you if the website has a different direction like on the body tag.

Lea Alcantara: Right.

Zoe Gillenwater: So you don't have to write CSS and changing all of your float left to float right. It just automatically switching things.

Lea Alcantara: Right.

Emily Lewis: I'm just curious, Zoe, does this save you time in terms of your front-end development? I'm just looking at the examples you shared with us, and I was like, "Wow, this probably would be a lot faster than me writing four dozen media queries to get things to work differently." [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: Is there a huge learning curve, but then it becomes like faster development, or is it about the same?

Zoe Gillenwater: Let's say there definitely is a learning curve if you're used to making layouts with like floats or inline-block or *display: table-cell;* But then after that, I think it is faster. It usually results in a fewer lines of CSS that you have to write, and like I said, there are a lot of properties that take care of common alignment and spacing things that would have taken like all of these, like extra <div>s and like crazy CSS to accomplish and instead you can just set like one property and one value and be done with it. So it helps a lot with those sorts of things.

Emily Lewis: Yeah, I like how it seems to allow content blocks to take up the space that they're in as opposed to like I call it the squishy middle, like I have different breakpoints that I just generally aim around and then as I'm developing, there's like squishy bits on this page where this needs to fill this much space, and on a different page, it's a little different and so I have all these media queries for these discreet issues as opposed to the overall site layout.

Zoe Gillenwater: Yeah.

Emily Lewis: And I feel like this appears that it would resolve that sort of squishy middle aspect the sort of things the media queries that I have to set once I'm looking at a page and kind of resizing it and seeing how it flows because when, let's say, a content box, there's not enough space in a given viewport for it, it drops to the next level, but then it becomes the full width of it.

Zoe Gillenwater: Right.

Emily Lewis: But without me saying, "Oh, well, in this situation, it's 50% wide and then when it goes up to 960 pixels, then it's a 100% wide."

Zoe Gillenwater: Right.

Emily Lewis: It just sort of fills the space.

Zoe Gillenwater: Yeah, absolutely. It's good for making the content kind of wrap more elegantly between your breakpoints, and so you have those breakpoints to kind of optimize like, "Here is what I want the layout to look like at this size because it looks horrible if I don't add a breakpoint here." Right?

Emily Lewis: Right.

Zoe Gillenwater: But then often, getting close to those breakpoints, it might start looking a little bit awkward so flexbox can sometimes help, yeah, to kind of clean up some of those little awkward areas by making the content. The box is more responsive to their content.

Emily Lewis: So this wasn't a question that we sent you in advance. But I'm just curious, is flexbox like your preferred method now, or do you still use other methods? Do you ever do layouts with the Grid Layout?

Zoe Gillenwater: Yeah, I still use other methods because I do have to support back to IE 7 at Booking.com. If I can accomplish something with display: inline-block, then I'll just do it that way. I will add flexbox if there's a benefit to having flexbox.

Emily Lewis: [Agrees]

Zoe Gillenwater: But I'm not going to just use it just because it's there. So there is still a lot of the validity in using *display: inline-block* or *table-cell;* but flexbox is something that I probably use the majority of the time. So not on every layout thing I write nowadays, but probably more often than not, I do use flexbox.

Emily Lewis: And you mentioned earlier an accessibility concern regarding like tabbing order or screen reader order. Are there any other accessibility issues that flexbox introduces that we should be concerned about?

Zoe Gillenwater: No, pretty much the only issue is how it affects the non-visual order. But like I said, you can use that for good or for evil. [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: So you could use it as an advantage or you could kind of abuse it and make things really crazy in HTML and then try to fix it with flexbox. So as long as you're using the flexbox order property just to do purely decorative reordering, so not changing the meaning of the content by reordering it, then you should be fine in terms of accessibility.

Emily Lewis: And I feel like things like HTML source order, that's just fundamental, like you should be making sure that it is good to start with.

Zoe Gillenwater: Absolutely.

Lea Alcantara: So you kind of mentioned when you were saying part of the benefits of flexbox is you end up writing less CSS and that leads to me thinking like, "Oh, are there performance benefits to flexbox or are there performance concerns with flexbox? Like what are pros and cons in terms of like things rendering quickly and working?

Zoe Gillenwater: I think the performance aspects of flexbox are something that I would love to see tested a whole lot more, and I think a lot of it is just because it is kind of new in being used in real, live production code, so there hasn't been a lot of research done about this. I do think it can have benefits and disadvantages like you mentioned. So it doesn't have any speed problems in terms of the actual page load time compared to like *display: table-cell;*

Paul Irish did a test a couple of years ago and found that the current spec, the current version of flexbox didn't have any problems there compared to display: table-cell; and it was much faster than the old version of flexbox, so that was good.

Lea Alcantara: [Agrees]

Zoe Gillenwater: It can cause issues in that since you don't have to put explicit dimensions on things and you can shift boxes around with it like I was just talking about, that means that when the browser is loading a page with a bunch of flexbox boxes on it, it's possible that the users will see content resizing and shifting as new content loads in because the browser doesn't know yet how big to make it.

Emily Lewis: Right.

Lea Alcantara: [Agrees]

Zoe Gillenwater: So that's the perceived performance sort of thing. So yeah, there are kind of pros and cons. Basically, if you wanted to avoid that perceived performance problem, you could just avoid using flexbox for your overall page layout and then use it mainly to layout individual components. I find that works better anyway. It wasn't really designed to be an overall page layout tool. The Grid Layout Module is going to be a lot stronger for that use case, but flexbox is really good laying out kind of widgets and components and modules inside those major page sections.

Emily Lewis: So in terms of like you were describing, I mean, is that kind of like a "Flash of Unstyled Content (FOUC)" as it's trying to like figure out what it looks like as the page is rendering?

Zoe Gillenwater: It's kind of similar to that, but it would be that the content…

Emily Lewis: Flash of unlaid out content? [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Yeah, if you have a whole bunch of flexbox boxes that none of them have explicit dimensions, then as they come in, then they'll have to resize and move around to make room, and so that might be kind of an awkward user experience.

Emily Lewis: Would progressive enhancement address that where you had specified widths and then the layout was triggered in terms of the order you wrote it in your CSS?

Zoe Gillenwater: It's possible. I think, yeah, like I said, this would be something that is ripe for experimentation. I'd love to do more there and see what other people find using it again in the real world, seeing how it performs for people. I think we need more information on this and more examples.

Emily Lewis: Well, speaking of, what's your perception of its use right now? Do you sense that a lot of people are starting to use it, or is it still a ways off from being widely adopted?

Zoe Gillenwater: I think a lot of people are starting to use it, so I've been speaking about it at conferences for a few years really, and in the last – I don't know – six months when I've spoken about it, lots of people have said that they are already using it. I mean, obviously, that's kind of skewed towards the type of people who go to conferences. [Laughs]

Emily Lewis: Right. [Laughs]

Lea Alcantara: Sure. [Laughs]

Zoe Gillenwater: So I don't think it's in wide use among just kind of ordinary plugging away at your job sort of web developer person.

Emily Lewis: [Agrees]

Zoe Gillenwater: But among a certain subset of the web world, it definitely is getting use now.

Emily Lewis: So we talked a little bit about accessibility, a little bit about performance and even progressive enhancements, are there any things more along those challenge aspects of flexbox that we should consider?

Zoe Gillenwater: Well, like anything, there are bugs with certain browsers.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Zoe Gillenwater: Unfortunately, IE tends to be the main offender here still.

Lea Alcantara: [Laughs]

Zoe Gillenwater: So IE 11 has some flexbox bugs. There's work around for many of them. Phil Walton has like a repo on GitHub called flexbugs where he documents a bunch of these. It's not just IE 11, but there are bugs with some other browsers as well, but at least since there's starting to be documented now. You can learn maybe some workarounds for them, and it's a new part of the layout algorithm that browsers are adding so it make sense that there are going to be some bugs, but so far I haven't run into too many problems there.

Emily Lewis: That's always good to hear. [Laughs]

Lea Alcantara: Yeah, yeah, for sure. [Laughs]

Emily Lewis: So is flexbox particularly suited to a particular type of site or even a code base?

***Timestamp**: 00:29:47*

Zoe Gillenwater: Nothing kind of immediately comes to mind there. I mean, the main thing that comes to mind is, just like I said before, it's really good for laying out kind of components, flexible components. So it's good for kind of web appy-like sites, if I could call it that and good at those sorts of widgets and that sort of thing. [Laughs]

Emily Lewis: Right.

Lea Alcantara: Right.

Zoe Gillenwater: It wouldn't be probably as much use for kind of just a small business website where you have just your two columns and you have your big hero image and that sort of thing.

Emily Lewis: [Agrees]

Zoe Gillenwater: And not that you couldn't use it there, but it just doesn't offer as many benefits in those sorts of situations.

Emily Lewis: You had mentioned that flexbox wasn't necessarily addresses the layout of an entire page. Can you expand on that a little bit, why it's more suitable for kind of the internal components?

Zoe Gillenwater: Sure. So, one thing that makes it not great at the overall page layout is that it is not an actual grid system. So you see people trying to use flexbox to make grids and you can do that to some extent, but it wasn't designed to be a grid system and you'll find that flexbox will make boxes be proportional to one another within the same line, so row or column.

Emily Lewis: [Agrees]

Zoe Gillenwater: But as soon as you get onto the next row or column, it doesn't care what happened in the row or column before, so it's not going align things up with the other rows or columns.

Emily Lewis: [Agrees]

Zoe Gillenwater: If you're trying to get a very precise grid with flexbox, you will probably run into trouble there, and the way to work around that with flexbox is to go back to giving everything explicit percentage widths, but then you're losing a lot of the point of flexbox if you're still going to have the same sort of grid CSS that you had before where everything has like 66.3784%, you know, and that's good with things that you do with those sort of grid systems.

Lea Alcantara: [Laughs]

Zoe Gillenwater: So that's a big reason why it's not that well suited to overall page layout, unless it's a pretty simple sort of page layout. So if you have kind of the stacked bar sort of layout where you have your big hero image and then you have your bar with your three little features of your product and then you have your bar with your testimonials and you want them to all be vertically centered with each other, like you could use flexbox for that sort of pretty simple page layout, but it's not going to create like perfect grids for you. Does that make sense?

Emily Lewis: Yeah.

Lea Alcantara: Yeah.

Zoe Gillenwater: I know it's kind of hard to talk about without seeing it.

Emily Lewis: Yeah, I mean, every time you talk about code, it's hard because a lot of times you have to visualize it.

Zoe Gillenwater: Yeah.

Emily Lewis: But we make a lot of those kind of sites so I know exactly what you're talking about. [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Yeah. [Laughs]

Lea Alcantara: But I think the summary is really like if there are multiple components, flexbox can probably work better with that type of page or site, and if it's just literally a simple here's one piece of content and then like one sidebar and one hero, do you really necessarily need to have like everything flow in a mystical interesting way? [Laughs]

Zoe Gillenwater: Yeah, right. Yeah, I mean, the layout tools we already were using before handled that sort of simple layout already pretty well.

Emily Lewis: And if I'm hearing you correctly, like I don't think flexbox is going to replace what we've done in the past. It's just another option that we can build into our layouts.

Zoe Gillenwater: That's what I think. I think that's a good way to look at it, and I think that when we finally do have the CSS Grid Layout Module supported in browsers that flexbox and grid layout are going to go really well together. So I think someday we're going to be laying out the overall page layout with CSS Grid Layout and then within those major sections that the grid creates, then we will use flexbox to create the components and to lay out individual pieces of content with each other in a proportional flexible way.

Lea Alcantara: It's very cool. So when you're actually sitting down to write all this code, do you use something like Sass or any other preprocessor to build it out?

Zoe Gillenwater: Yeah, Sass is really useful for flexbox because, as I mentioned it, it went through a bunch of changes and so there are the different browser prefixed versions as well as different names for some of the properties for older browsers, so there are a lot of Sass mixins that help manage all of that generating the different browser variants for you, so that makes things a lot easier. You can use Autoprefixer as well to add all of that stuff.

Emily Lewis: [Agrees]

Zoe Gillenwater: The problem with using Autoprefixer with flexbox I think is that you cannot tell it to not put in the 2009 syntax of flexbox.

Lea Alcantara: Oh, okay.

Zoe Gillenwater: So that's used by old Safari and old Android, and they supported an old version of flexbox that's pretty slow to render, and it's also kind of buggy. It doesn't have flex-wrap support. So I don't really like using that version, but Autoprefixer is going to add that version no matter what, unless you exclude those browsers from all of your Autoprefixer CSS, that's perfectly valid. But if you want to support those browsers, but just not with flexbox, you don't have that option. So I like the Sass way of doing it, having the Sass mixins which kind of control those variants, and Sass also can come in handy when you're trying to add Modernizr versions of the CSS so you can keep your flexbox and non-flexbox styles like indented with each other and so it's easier to see how they're related to each other and have your overrides together in the same place.

Emily Lewis: So is there such a thing as basic and then advanced flexbox or is it just flexbox?

Zoe Gillenwater: [Laughs] You can definitely use flexbox for really simple stuff, so you could use it to just do equal height columns or vertical centering.

Emily Lewis: [Agrees]

Zoe Gillenwater: So you could still have floating or whatever to create your columns, but then you add flexbox just so that they are equal in height, the background color stretches all the way to the bottom, like that's it, and that could be just adding two lines of flexbox CSS on top of your other CSS. So you can definitely use it in a really simple way like that without having to lay out like entire components with it. That would be kind of the more advanced use of flexbox.

Emily Lewis: I think that's probably a nice way to get comfortable with it, just using it in small, discreet sections of a given site, like I feel like equal height columns or the vertical centering alone comes up so often in even your basic "marketing" or brochure-style site.

Zoe Gillenwater: Yeahl.

Emily Lewis: That that would be a really good way to sort of experiment, see how it fits in, and get yourself comfortable with it.

Zoe Gillenwater: Yeah, and I mean, another use for flexbox that I love that's really simple is just putting two things on opposite ends of the same line.

Emily Lewis: [Agrees]

Lea Alcantara: Yes.

Zoe Gillenwater: Which we have so many ways to do that with CSS and they work okay. [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: But the problem that usually comes in when those two items can no longer fit on the same line, and one of them has to wrap.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Zoe Gillenwater: That's usually where it gets ugly, and so flexbox can make that wrapping a lot less ugly. It can make it a lot more smart as to what point that wrapping takes place at and make it wrap to, like instead of having one wrap down to the right edge, it can wrap to the left edge underneath the other piece of content so it looks more natural of how things stack and wrap.

Emily Lewis: [Agrees]

Zoe Gillenwater: And so that's another just really simple use of flexbox is just putting one thing on the left and putting one thing on the right.

Lea Alcantara: And a specific use case of that example is footers because we all have the on the left hand side is the copyright and maybe a copy of the navigation links, but on the right side, we've got our headline plus our social media icons, you know?

Zoe Gillenwater: Yes.

Lea Alcantara: And currently in the "old-fashioned" way, we just have to like be okay with it wrapping and right aligned and float it to the right and then the other one is floated to the left. But if you want to have your social media icons start where your headline "Share this post" starts, flexbox makes more sense.

Zoe Gillenwater: Yeah.

Lea Alcantara: And it makes it easier.

Zoe Gillenwater: Yeah, absolutely.

Lea Alcantara: And naturally.

Zoe Gillenwater: Yeah, again, in terms of progressive enhancement, you can still have that old CSS in place there and then you can just add flexbox on top so that almost everybody, like 95% of your site visitors will see the more elegant flexbox version.

Emily Lewis: [Agrees]

Zoe Gillenwater: But it will still be on the left and right for other browsers most of the time.

Emily Lewis: What about a more complicated implementation? What's the coolest thing you've seen or built yourself using flexbox? [Laughs]

Zoe Gillenwater: Well, one really complicated but cool implementation of flexbox is the Guardian's website, the newspaper in the UK.

Emily Lewis: [Agrees]

Lea Alcantara: Oh, okay.

Zoe Gillenwater: So you should definitely go check out their website and look at their CSS. So they are using flexbox to lay out their story blocks in a responsive way to kind of rearrange the layout from the mobile view to the wider view. It's really modular and really cool to see how they can kind of shift content around and make again equal height columns and things like that using flexbox. So that is a really good example of how it's being used, like in the real world in a really comprehensive way.

Lea Alcantara: Oh, very cool. I'm looking at it right now and sometimes we just take those kind of like vertical, everything is so perfectly aligned vertically for granted.

Zoe Gillenwater: Yeah.

Lea Alcantara: And I'm looking at them and I'm like, "Oh my God, it's so pretty." [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: [Laughs]

Zoe Gillenwater: Well, that's the thing with a lot of flexbox demos, it's like they don't look that fancy because it's just like this is how it should be, right?

Emily Lewis: Right.

Zoe Gillenwater: Like everything should be lined up.

Lea Alcantara: It's a demo.

Emily Lewis: Right, right.

Zoe Gillenwater: Like that's one of the main benefits of flexbox. It's not like it's doing anything crazy outrageous, but it's just like making stuff lined up, like we always wanted it to be in an easy way with just a little bit of CSS.

Emily Lewis: Right.

Zoe Gillenwater: But there are some really cool demos with it on like CodePen. There's like a tournament brackets layout that uses flexbox.

***Timestamp**: 00:40:01*

Emily Lewis: Is this one of those CodePens you sent us?

Zoe Gillenwater: Yes, yeah.

Emily Lewis: We'll be sure to post them on our show notes.

Zoe Gillenwater: Yeah, and then I have like a responsive menu, like the off-canvas menu pattern using flexbox, and then it shifts automatically to a top nav far when the window is wider.

Lea Alcantara: Cool.

Zoe Gillenwater: So those sorts of things are really cool to look at as well.

Emily Lewis: Oh, that's nice. That's one of the things that is so cool about CodePen. You can really see these examples nicely. So in not our last episode, but our first episode of this year, one of our main points of conversation was making smart decisions about what trends to invest our time and energy in.

Zoe Gillenwater: Yeah.

Emily Lewis: So where do you think flexbox should rank for someone like us, we've been working on the web for a really long time?

Zoe Gillenwater: I definitely think it should rank highly because it has been around for quite a while now and it's really stable in terms of the syntax now and it's really well supported, and so I think it's going to be used pretty extensively in component layout from now on, and it does make things a lot easier than the old layout hacks, so it will probably be a welcome relief to seasoned web developers and designers. [Laughs]

Emily Lewis: Well, yeah, just looking at some these examples, I'm like, "Oh gosh, this will save me time on some of the things that I've tried to do before, yeah."

Zoe Gillenwater: Yeah.

Emily Lewis: What about for like a newbie, someone who's just getting into the web, should flexbox be on their radar?

Zoe Gillenwater: I think so as well, but like I said, it's still perfectly valid to use other layout methods. So I would want newer devs to have a good understanding in just like the box model and how that affects layout. I feel like a lot of people don't really understand that. They don't understand how it really works. They just know, "Okay, if I google this and I copy and paste this code and I paste it in there, it makes things go to the left and the right." So I think having a real understanding of how the box model works and how like display: block; vs inline; vs inline-block, how those things work is more important first. But once you understand those things, I think flexbox is a good thing to add on because it does allow you to do all sorts of different alignment, things that you can't do before, and it is so direction neutral, so you can run content any which way, so I think that's really fun to play with when you are somewhat new to making web pages.

Lea Alcantara: So if you are a newbie, and like if I was a newbie and then I went up to you and you said, "Yes, learn the box model," where would you take them to? What resource would you send them so they learn the foundation of the box model before jumping willy nilly into the flexbox?

Zoe Gillenwater: That's a really good question. I like a lot of the stuff on MDN (Mozilla Developer Network).

Emily Lewis: Network.

Zoe Gillenwater: Though I don't know if they, in particular, have like a great like box model or display tutorial, but I feel like they explain things pretty clearly and give you a good understanding of like what technically is going on without being as dense as the actual spec at the W3 site.

Lea Alcantara: Right.

Emily Lewis: [Laughs]

Lea Alcantara: Right. [Laughs]

Zoe Gillenwater: So that I think is a good resource for people who are learning.

Emily Lewis: And what about for those of us who want to get more familiar with flexbox, what would be your top resources for that?

Zoe Gillenwater: Well, one thing would be the flexbugs' site on GitHub that I mentioned before just because, yeah, I mean, those are things that you might run into and if you are more experienced, it's good to have that knowledge ahead of time so that you can just avoid some of those issues to begin with, that you don't have to even ever run up against them.

Lea Alcantara: Right.

Emily Lewis: I love that suggestion. I think far too often, getting familiar with the bugs is the last thing we end up doing.

Lea Alcantara: Right.

Emily Lewis: But it actually makes sense to know what you might be getting into before you get into it.

Zoe Gillenwater: Yeah.

Lea Alcantara: Yeah.

Zoe Gillenwater: And it's not like there are like fifty bugs on there. I mean, there are literally like twelve bugs or something like that, you know?

Emily Lewis: [Laughs]

Zoe Gillenwater: So you can read it and click on some of the demo links or whatever in an hour and have a good knowledge of what you're heading into, and then there's a couple of really cool visual, interactive tools to play with flexbox, and you can change the values and see how they move things around. One is this [Flexbox] Froggy game where you're like moving frogs around onto lily pads and stuff using flexbox properties. So I think that's a good tool for even experienced developers. If you haven't used flexbox yet, it's a good way to get a visual sense of what the weird property names and values translate into in actual layouts.

Emily Lewis: Yeah. I also like this flexbox in 5 that you sent us.

Lea Alcantara: Yes, totally.

Zoe Gillenwater: Yeah.

Emily Lewis: It's like a tutorial/demo/interactive kind of thing, but I also think that everyone learns in different way so you got to find the thing that connects most to you, but this one, I particularly like because it's informative but it's also making me be interactive with it so I can see how the different values change and then I also really love that it's a code that I can see what it would look like, the syntax and such.

Zoe Gillenwater: Yeah, absolutely. I really like that one too.

Lea Alcantara: So besides the Guardian website where you're like, "This is really, really cool and complex use of flexbox," do you have any other cool sites for our visitors to take a look at?

Zoe Gillenwater: Well, CodePen is always a great resource for seeing demos of things that people are doing with flexbox and other tools. Since flexbox is pretty well suited to little component layout, I think it's also pretty well suited to CodePen, so you can see how you would use it in small bits and pieces. There's not really like, at least that I know of, there's no like flexbox-dedicated inspiration site. I would like to gather more examples of real sites that are using it. I know like Shopify uses it a little bit. I know Skyscanner I think uses it a little bit. So there are some other sites out there other than the Guardian who are using it, but I think it would be great to kind of have them gathered together somewhere. I should really work on that. [Laughs]

Emily Lewis: Like a gallery of something.

Lea Alcantara: Yeah. [Laughs]

Emily Lewis: Because I imagine it's one of those things you may not just look at a site and say, "Oh, it's using flexbox." You'd have to get behind the scenes and see what's happening.

Lea Alcantara: Right.

Zoe Gillenwater: Right, exactly, yeah, because a lot of times it just looks like existing layouts. It's not until you've come to a point where it would have like broken down and looked horrible that like flexbox like swoops in and saves the day.

Emily Lewis: So before we wrap up, you mentioned earlier that you wouldn't necessarily recommend flexbox for an entire page layout for the external things, especially grid-based. One of our listeners wrote in and specifically said, "What should we not use it for? Is there anything that flexbox is just not a good fit for?" Like if you're building a site for IE 6, definitely don't use flexbox. [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: But is there anything else?

Zoe Gillenwater: Yeah, that's a good one. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Yeah, I mean, basically, any of just kind of rigid grid sort of use, like you can use flexbox to create like kind of the grid gallery sort of thing, but also there's no way to control like the last item on the line. Like there's no Pseudo element for controlling that stuff with flexbox right now.

Emily Lewis: [Agrees]

Zoe Gillenwater: So that also makes grid sort of layouts with flexbox really tricky. So yeah, it's not really great for that sort of thing. I'm trying to think anything else that I would absolutely not use it for. No, I mean, that's the main thing that comes to mind. Yeah, if you have to have a picture perfect layout in IE 6 and 7, then flexbox is not going to offer any advantages there, but yeah, that's it.

Emily Lewis: I feel like that's really refreshing because like I said earlier, just a couple of years ago, the browser support wasn't where it is today, and so I don't think that would have been your same answer, and it's one of the reasons why at the time I didn't look into flexbox. I was just like, "Well, my clients need this kind of support so I'm not going to be investing in that right now."

Lea Alcantara: Right.

Emily Lewis: But from what I'm hearing now, it's to a point where, yeah, it's very viable. The browser support is as broad as it is, and obviously, progressive enhancement for those. Hopefully for our listeners, fewer of the sites you have to account for older versions of IE. [Laughs]

Zoe Gillenwater: Oh yeah, I agree. I feel like it just really snowballed in the last year.

Emily Lewis: [Agrees]

Zoe Gillenwater: Like it was kind of going, and there wasn't a whole lot of momentum on new browsers taking it up for a while and then like all of a sudden, it seemed like all of them were supporting it. So that's been really great that we can really use it now.

Emily Lewis: Definitely. This is going on in the list. Even following our discussion earlier this year talking about being picky about what I'm going to invest my time in this, I think this will save me time in the long run so it's worth me looking into now.

Zoe Gillenwater: Yeah, and one thing that I'll mention about that too is if you're going to use it for progressive enhancement, at first it might seem like, "Well, then how is it saving me any time if I have to make the layout first for IE 7 through 9 and then I have to like make the layout again with flexbox, like how can that help?"

Emily Lewis: Right.

Zoe Gillenwater: But basically how I'm using it for progressive enhancement is I'm not really making like a perfect layout first IE 7 through 9, but I'm just giving them like, yeah, display: inline; or something.

***Timestamp**: 00:49:50*

Emily Lewis: [Agrees]

Zoe Gillenwater: So everything is on one line together, fine. It doesn't stretch full width or the last item doesn't move to the right side like I wanted to, but it's fine, they're all laying out on the same line together and now I'm going to add flexbox on top to make the middle section stretch to full width and the right thing actually go over to the right edge and all of those sort of layout improvements, and in that sense, it really does save me time not having to create some complicated HTML structure and

CSS to accomplish that sort of layout. So it doesn't have to look exactly the same in the older browsers, you just give them something very basic lay of the foundation and then just add flexbox as an enhancement.

Emily Lewis: Cool.

Lea Alcantara: Very cool. Very cool. I feel like Emily is already starting to code off to the side right now. [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: But before we finish up, we do have our Rapid Fire Ten Questions so our listeners can get to know you a bit better.

Zoe Gillenwater: All right.

Lea Alcantara: Are you ready?

Zoe Gillenwater: I'm ready.

Lea Alcantara: All right, first question, morning person or night owl?

Zoe Gillenwater: Morning person.

Emily Lewis: What's one of your guilty pleasures?

Zoe Gillenwater: I really like organizing things. It's so nerdy and horrible.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: But I don’t care, I love it.

Lea Alcantara: Oh, you’re speaking to the right people. [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: What software could you not live without?

Zoe Gillenwater: Software, I guess Sublime. I have to use it all day every day.

Emily Lewis: What profession other than your own would you like to try?

Zoe Gillenwater: Should I say professional organizer, I guess. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: No.

Emily Lewis: That would be the best job. [Laughs]

Zoe Gillenwater: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Professional organizer would be pretty good, or I really like gardening, so I don’t know what. I don’t know what sort of gardening job you could have. I guess I could work on the tulip fields here in Amsterdam or something. That might be nice.

Emily Lewis: Cool.

Zoe Gillenwater: [Laughs]

Lea Alcantara: What profession would you not like to try?

Zoe Gillenwater: Dentist.

Emily Lewis: If you could take us to one restaurant in your town, where would we go?

Zoe Gillenwater: [Laughs] Well, there's a Dutch restaurant that I think has really good food and it's all cute and quaint and Dutch and everything, and so I usually like to take visitors there, but it's called – you see, but if any Dutch people are listening, they're going to laugh at my horrible pronunciation. Okay, it's called something like Haesje Claes or something, but I don't know, it's got a lot of vowels. That's all I could say. [Laughs]

Zoe Gillenwater: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: All right, if you could meet someone famous, living or dead, who would it be?

Zoe Gillenwater: Jane Austen.

Emily Lewis: If you could have a super power, what would it be?

Zoe Gillenwater: Teleportation.

Lea Alcantara: What is your favorite band or musician?

Zoe Gillenwater: Stone Temple Pilots. No. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Zoe Gillenwater: Red Hot Chili Peppers.

Emily Lewis: Last question, pancakes or waffles?

Zoe Gillenwater: Waffles.

Emily Lewis: Oh, you're our first waffle answer. [Laughs]

Zoe Gillenwater: Really? Very cool. Yeah, I'm in the land of pancakes and waffles so I've…

Emily Lewis: Yes, stroopwafel too.

Lea Alcantara: Yeah, that's true.

Zoe Gillenwater: Yeah. It's good stuff.

Lea Alcantara: So that's all the time we have for today. Thanks for being on the show, Zoe.

Zoe Gillenwater: Thank you for having me. This was a lot of fun.

Emily Lewis: In case our listeners want to follow up with you, where can they find you online?

Zoe Gillenwater: So my website is zomigi.com. My name is way too long so I just shortened it to Zomigi and that's also my Twitter handle (@zomigi) if you want to tweet at me.

Emily Lewis: Awesome. Thanks again, Zoe. This was exactly what I needed to get off my butt and start working with flexbox. [Laughs]

Zoe Gillenwater: Good, good. Let me know how it goes.

Emily Lewis: I will. [Laughs]

[Music starts]

Lea Alcantara: CTRL+CLICK is produced by Bright Umbrella, a web services agency obsessed with happy clients. Today's podcast would not be possible without the support of this episode's sponsor! Thank you, Craft Commerce.

Emily Lewis: We'd also like to thank our partners: Arcustech and Devot:ee.

Lea Alcantara: And thanks to our listeners for tuning in! If you want to know more about CTRL+CLICK, make sure you follow us on Twitter @ctrlclickcast or visit our website, ctrlclickcast.com. And if you liked this episode, please give us a review on iTunes, Stitcher or both! And if you really like this episode, consider donating to the show. Links are on our site and in our show notes.

Emily Lewis: Don't forget to tune in to our next episode when we are revisiting an episode from our EE Podcast days. Travis Smith is returning to the show to discuss his updated statistics on large ExpressionEngine sites. Be sure to check out our schedule on our site, ctrlclickcast.com/schedule for more upcoming topics.

Lea Alcantara: This is Lea Alcantara …

Emily Lewis: And Emily Lewis …

Lea Alcantara: Signing off for CTRL+CLICK CAST. See you next time!

Emily Lewis: Cheers!

[Music stops]

***Timestamp**: 00:54:13*