



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

CTRL+CLICK CAST #115

The State of Web Images with Kitt Hodsden

CTRL+CLICK CAST is proud to provide transcripts for our audience members who prefer text-based content. However, our episodes are designed for an audio experience, which includes emotion and emphasis that don't always translate to our transcripts. Additionally, our transcripts are generated by human transcribers and may contain errors. If you require clarification, please [listen to the audio](#).

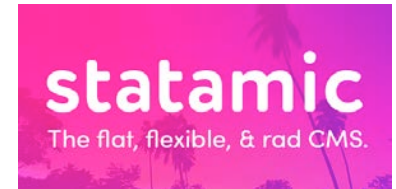
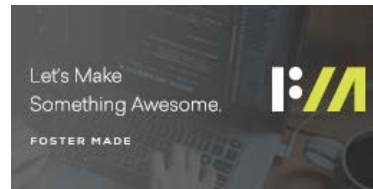
Preview: For the record, perceived performance trumps actual performance every time. Because our goal with web performance is not “Oh, let me load this in a second,” but our goal is, “Is the user happy with our site? Are they going to buy something?” They don’t care that it takes seven seconds to download versus four seconds. They want to know that something is happening.

[Music]

Lea Alcantara: From [Bright Umbrella](#), this is CTRL+CLICK CAST! We inspect the web for you! Today Kitt Hodsden joins the show to discuss the state of web images. I’m your host, Lea Alcantara, and I’m joined by my fab co-host:

Emily Lewis: Emily Lewis!

Lea Alcantara: This episode is sponsored by [Foster Made](#), a versatile web development agency specializing in custom application development, content management systems and user experience design. Through partnerships with designers, agencies and organizations, Foster Made is committed to building better digital experiences. Visit fostermade.co to learn more.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

This episode is also sponsored by [Statamic](#). If you build websites for a living and you haven't checked out Statamic yet, it might be time. Statamic's Flat-File-First approach to content management allows you to version control everything from your content to your config files, share data between sites and create workflows you never knew possible with 40 custom field types and a beautiful control panel. Statamic has something for everyone. Learn more at statamic.com and save 20% off your first license with the code CLICKCLICKCLICK.

Emily Lewis: So I'm super excited about today's episode. I've been wanting to talk about web images for a long time.

Lea Alcantara: [Agrees]

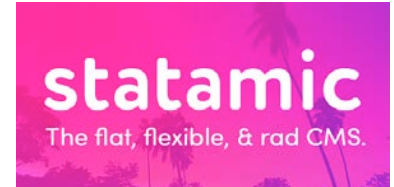
Emily Lewis: I think it's been on our list forever.

Lea Alcantara: Yeah.

Emily Lewis: One of the reasons that I want to talk about it is that every time I start a new front-end project, I check out what techniques and trends are out there for serving mobile-friendly images, and most of the time I had been falling back to methods that I've been using for years, and that's usually because our clients have a lot of older browser support needs. And so like you have to make some compromises. But also because images are like the number one speed killer on our client sites.

Lea Alcantara: [Agrees]

Emily Lewis: And no matter how much we do with the CMS to restrict or how much training we give them or documentation we provide, their images are huge and we're constantly trying to help our clients manage them.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

So today Kitt Hodsdén is joining us to have a “state of the web images” talk. Kitt is a full-stack developer with a penchant for performance and an irrepressible zeal for automation. She is also a co-founder of Hacker Dojo and a dev alum of Twitter, Shopify and Apple. Welcome to the show, Kitt!

Kitt Hodsdén: Thanks! I’m excited to be here!

Emily Lewis: We’re excited to have you! [Laughs]

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

Kitt Hodsdén: I recently moved to Portland, Oregon from the San Francisco Bay Area, and I love playing ultimate Frisbee.

Lea Alcantara: [Laughs]

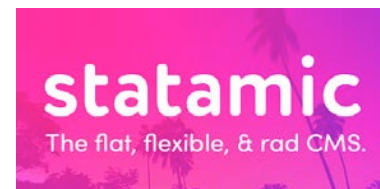
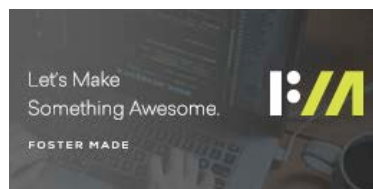
Kitt Hodsdén: That’s a sport that I cannot get enough of, and I’m currently traveling a lot, and so the web has been almost in the background for the last couple of months.

Emily Lewis: Hmm. Well, though it’s in the background, what even got you started in it in the first place?

Kitt Hodsdén: Oh, that’s exciting. So I was working at a post-production visual effects company.

Emily Lewis: Hmm.

Kitt Hodsdén: And a coworker of mine did a website for the company. It was the first one, this was back in – I want to say ’96 or ’97, and it loaded so slowly.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Laughs]

Kitt Hodsden: Oh, my gosh, he had these beautiful images that when viewed from home, it took about ten seconds for each one of them to load.

Emily Lewis: [Laughs]

Lea Alcantara: Oh.

Kitt Hodsden: And I was, “Oh, my goodness, this is horrible,” and so I went ahead and redesigned the website, and that was my first experience in performance.

Emily Lewis: Oh, that’s cool.

Lea Alcantara: Oh, fun.

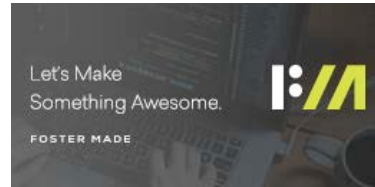
Emily Lewis: And you already knew, like HTML, or you just taught yourself from the get-go?

Kitt Hodsden: Oh, I totally taught myself from the get-go.

Emily Lewis: Yeah.

Kitt Hodsden: I was doing post-production visual effects, and so was doing a lot of programming for movie-sized images and filters and automating dust removal as opposed to web stuff.

Emily Lewis: Oh, it’s still ultimately the same thing, how to get things really looking good, but also small and then making it automated, fast and efficient.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Absolutely.

Emily Lewis: Cool. So I'm curious, especially since you got started with such large images, I think you probably have an interesting historical perspective, but how has the use and implementation of images sort of evolved throughout the time you've been on the web? I mean, first, we had JPG and how do people pronounce it? What is it? JIF? GIF?

Lea Alcantara: GIF versus JIF. [Laughs]

Emily Lewis: Whatever. [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: And then PNG got more popular as IE6 support fell, and now SVGs are a big trend. So I mean, how do you see that kind of that historical movement with images? What prompted all those changes?

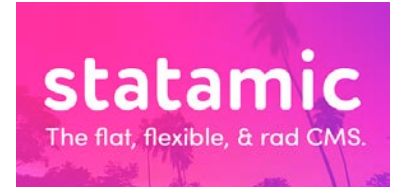
Kitt Hodsdon: Efficiency pretty much prompted most all of the changes, making things faster and making things better as, well, we all want to do.

Emily Lewis: [Agrees]

Kitt Hodsdon: Each image format did something better.

Emily Lewis: [Agrees]

Kitt Hodsdon: So for example, JPEG images were smaller, which meant that they could deliver faster, and then PNGs came about because GIFs were, well, aside from the fact that you had the



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

licensing issues that had come up, PNG was originally the response to the licensing issues, but then it became more efficient because JPEGs didn't, for example, have the alpha channel.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: And now we have SVG because well, vector is just so much better for things that need to scale. So each step along the way has been an improvement.

Emily Lewis: Yeah, and I also feel like each step of the way sort of – I don't know if they are just coincidentally happening at the same time or one is driving the other, but as browser support gets better or like SVGs needing to scale, I mean, that's just completely in line with the move towards mobile.

Kitt Hodsdon: Yeah.

Lea Alcantara: So this might be an obvious question, and you kind of touched on it, but I think it's important to make sure that everyone has a baseline understanding about these file formats.

Kitt Hodsdon: [Agrees]

Lea Alcantara: So which type is best for what?

Kitt Hodsdon: Ah.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: And not necessarily just for performance and all that fun stuff, like what should be better used for decoration versus information, and then afterwards like, okay, so once you decide what that is, visual acuity, like which one is best for serving that particular image alongside file size?

Kitt Hodsdon: When in doubt, the smallest file size always wins.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

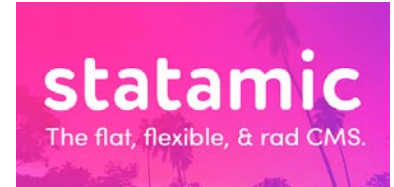
Kitt Hodsdon: Like I emphasize this with every client, that you want things to get over the wire as quickly as possible. Good guidelines are if it's a photo, if it's a complicated picture, then JPEG is going to be the default, and WebP if you have normal statistics, and what I mean by normal statistics is Chrome browsers take up about 60% of the market share for the top 100 websites, and Chrome is the only browser that supports WebP, and WebP is more efficient. So if you can, you want to do WebP over JPEG. If you can do any kind of image vectorized to you, SVG is the way to go. The files are smaller, they scale beautifully and they just are wonderful. Again, though, smallest file size wins and PNG is likely to be smaller for a lot of images that are not photographic images.

Emily Lewis: [Agrees]

Kitt Hodsdon: Animation is also important for a lot of the people.

Lea Alcantara: [Agrees]

Kitt Hodsdon: And animated GIFs are the old technology. MP4s are the way to go for that, unless you want to be experimental. PNGs have an animated format, APNG, which is supported by everything but Microsoft products, i.e., in Edge.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Lea Alcantara: Hmm.

Kitt Hodsdon: So that's an option if you're interested. There are also a large number of other formats that are vendor specific. For example, JPEG 2000 is Safari only. JPEG XR for Microsoft is Edge and IE only. Both of them work great for the specific browser that they are intended for and have like a greater color depth, they have alpha channels, and great for the file size, but not universal. So if you're looking for a simplified process, JPEG for photos, SVG for vectors, and PNG for pretty much everything else.

Emily Lewis: Wow, I hadn't even heard of WebP.

Lea Alcantara: Yeah.

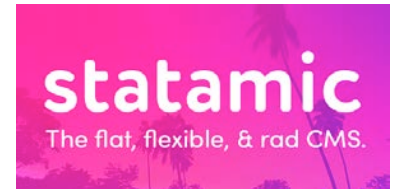
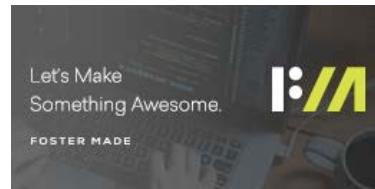
Emily Lewis: Can you describe that a little bit more?

Kitt Hodsdon: I sure can. WebP is Google's, excuse me, Chrome's more efficient file format.

Emily Lewis: [Agrees]

Kitt Hodsdon: It isn't universally supported. It's pretty much only Chrome, and oh, I don't know how many people wish that it were also supported in Safari, that would be so amazing.

Emily Lewis: [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdn: There are converters out there that will convert from JPEG to WebP, and if your server, your web server, for example, detects... so here's a trick of web browsers, they're incredibly resilient, and if you send the wrong file, it will still try to render it.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdn: So one of the tricks that I do is if I have configuration access on the server, I'll generate a WebP image, which is smaller typically than a JPEG image and serve it. Even though I say that the file is a JPEG, the browsers will say, "Oh, this is a WebP," and it will stick it in that JPEG anyway.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

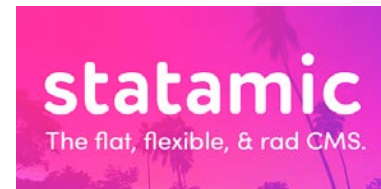
Kitt Hodsdn: But it's worthwhile generating the WebP and serving them, if you can. There's also a WebM for animated ones.

Emily Lewis: [Agrees]

Kitt Hodsdn: They are...

Emily Lewis: And that's also Chrome?

Kitt Hodsdn: It's also Chrome and significantly less supported toolwise. So tools are also important with images, not just the end user, but the developer having this also.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: [Agrees]

Kitt Hodsdon: And that's a lot harder.

Emily Lewis: So in terms of – because we have an order of questions we want to get to, but I feel like I want to bounce into something that we have a little bit later in reference to what you just said.

Kitt Hodsdon: Sure.

Timestamp: 00:09:48

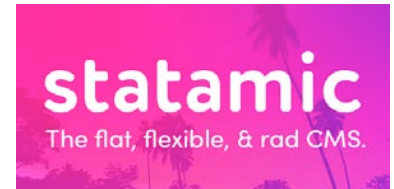
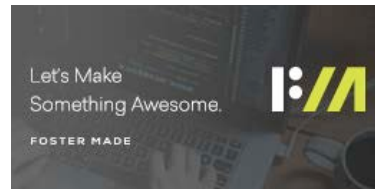
Emily Lewis: So when you have that situation where you could be doing these much more efficient smaller images for the Chrome browser, but you're the developer, you've built it, you've created this thing and you've optimized the best you could, but at some point, you have to hand it over to a client who at least in our experience, if I said, "Yeah, make some WebP images," they'd be like, "What are you talking about?"

Kitt Hodsdon: [Agrees]

Emily Lewis: And now I've got to have more training.

Kitt Hodsdon: [Agrees]

Emily Lewis: So are those realistic solutions when you're talking about the client end user? Are they really for them? It's almost exclusive to development shops or people who have like this high-level in-house expertise?



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: It depends on who you're developing for and which part of the website you're developing. So for example, I was working on a news website and every piece of content that went up had a picture with it, and that picture needed to be processed in some way, and so part of the delivering of the project that I had with that one was when the file was uploaded, it was resized, it was optimized, and all this was done server-side versus the case where you're designing and the developer is developing these images.

Emily Lewis: [Agrees]

Kitt Hodsdon: And these are often decorative or content, but content that isn't changing rapidly, and so the developer has an opportunity to build up the page, an About page, for example, that has graphics on that isn't going to change very frequently, in which case, those can be done one-off.

Emily Lewis: [Agrees]

Kitt Hodsdon: But in terms of projects that are delivered, I'm a big fan of automating things, and I'm a big fan of delivering things that don't need the client the client to come back to talk to me.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: If they have to come back and say, "How about this?" I have done my job poorly.

Emily Lewis: Yeah, that makes sense. I mean, I do feel like there is this. I think with this topic in particular, there is just this constant balance of what it's used for, who's going to be managing it, the bigger picture.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsden: Yeah.

Emily Lewis: It's not just getting it implemented and up on the screen.

Kitt Hodsden: Right.

Emily Lewis: So let's talk a little bit about optimizing the images in the first place before we even get them into a web page. What are the techniques that you follow to optimize images. Does it vary based on the type of image file format?

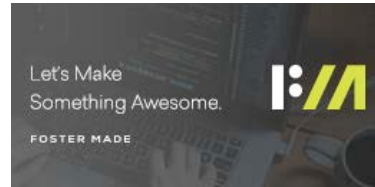
Kitt Hodsden: Typically, it does vary, but the tools and the processes are similar. For example, with JPEG images, there are baseline images, which when you look at the website or when they load on the website, they will start at the top and it will just render the rest of it, from the top to the bottom, versus progressive JPEG images, which will render blurry to sharp. It gives a low res image and then it will continue to download data until it gets a sharp image. So with progressive versus JPEG, I want to strip out a lot of the extra data. I want to reduce the quality to lossy image, but I don't want to necessarily lose the fidelity.

Emily Lewis: [Agrees]

Kitt Hodsden: I can convert those to WebP when I resize them, for example. So what I mean by quality images, JPEG, because it's a lossy image, it means that we can make smaller files, but we end up losing some of the sharpness and the fidelity and sometimes even the color when we reduce the quality.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: WebP has a similar quality. We can also encode those JPEG with a different encoder, so we can specify the quality, but then we do it with an encoder, for example, [MozJPEG](#) is an encoder that reduces file sizes by about 10% and [Guetzli](#) is another file encoder that came from Google that reduces about 30%.

Emily Lewis: [Agrees]

Kitt Hodsdon: And this is great in part of this process for optimizing images and not something that I ever want to do by hand.

Emily Lewis: [Agrees]

Kitt Hodsdon: So for images that are part of the website that are not necessarily decoration, but static and aren't changing very much, I'll do those by hand. I will bring that into [ImageOptim](#) or I will run a note script that I have that will render these and generate the resized images.

Lea Alcantara: [Agrees]

Kitt Hodsdon: And ImageOptim will run a number of things. So for example, it will reduce the PNG files by compressing the way that it's written in the file, and JPEG, it will also reduce them. There are programs out there like [Pngcrush](#) and [pngquant](#). If you want to change PNG, which is for the most part, a lossless file format, what you see is what you get, and it's almost like GIF in the sense that it shows all the pixels, which can also make the image smaller by making it lossy and changing the algorithm by which those pixels are encoded.

So for those static images that aren't changing very much, I'll run it through ImageOptim. That's the simplest way to do it, but for the ones that are uploaded in content that I don't have access to, I will make that a process. So if the project requires a CMS somewhere in that upload process where an



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

editor will upload an image, it will get resized, it will have all the various parts of the file squished and removed and adjusted, and my current favorite tool for that is [Gulp](#) in terms of using [imagemin](#) and the associated modules with that.

For example, I'll also include Guetzli in that, and sometimes it works and sometimes it doesn't, but sometimes it works about 99% of the time and every once in a while you get that one picture that you have to hand tweak by.

Emily Lewis: [Agrees]

Lea Alcantara: So I'm just so curious about all this entire process and how it can be as automated as possible so it's as invisible as possible to the client.

Kitt Hodsdon: Yeah, yeah.

Lea Alcantara: So are you saying that if you got a CMS project, that there's a script that's triggered on every upload?

Kitt Hodsdon: That's what I do, yeah.

Emily Lewis: [Agrees]

Lea Alcantara: Okay.

Kitt Hodsdon: For example, WordPress, if the people are using WordPress, it's already in core to do responsive images, and those responsive images will display and [srcset](#) in their resized images, which is fantastic because from the end user, they just upload a picture and WordPress about three or four years ago also reduced the JPEG resize quality down to, I think, it was 75, it was 98 and then



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

75 and boom, like 50% of those image file sizes in all of these websites just went away because suddenly these images are – most people can't tell the difference in these images. I personally have to look really, really closely and run all the statistic packages to see the differences between these images because they look great.

Emily Lewis: Yeah.

Kitt Hodsdon: But render a lot faster because WordPress did this one change, which was just wonderful.

Emily Lewis: [Agrees]

Lea Alcantara: Well, I will say though, generally speaking, I agree with you where most people really can't tell the difference between a 98% compressed versus 75 versus even 60% compressed type of file, right?

Kitt Hodsdon: Oh, aggressive.

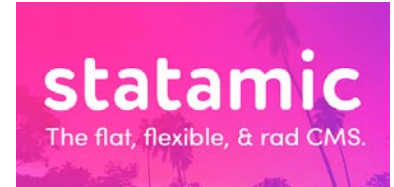
Lea Alcantara: But our Retina phones are high-res items.

Kitt Hodsdon: Yes.

Lea Alcantara: Depending on the site, of course and depending on how compressed certain things are, oh, you notice.

Kitt Hodsdon: Yeah, yeah.

Lea Alcantara: You start to notice.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Yes, and this is tailoring the solution to the client.

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Kitt Hodsdon: Sometimes it's working on Apple projects. Oh, you do those good, good pictures. There is no sense of compromise on that, and that's part of their products. That's who they are, and that's totally fine. And sometimes you kind of slip something in there. It's like, "Oh, did you really notice the difference between 98 and 85?"

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Kitt Hodsdon: And sometimes you can get away with it, and so there's always the, yes, tailored for the client. There's also an opportunity to show through data and through numbers why this is important.

Emily Lewis: Right.

Lea Alcantara: [Agrees]

Kitt Hodsdon: So you want these retina images and we can absolutely do with srcset and the media queries and we can tailor our HTML for that, and then we send them out into the field where you know what, you have a 3G connection. Go ahead and try and download that image and it takes 30 seconds to download that wonderful, beautiful image and this real world example gives us an



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

opportunity to say, “You know, maybe not,” and so it’s always an education process both internally with myself and externally with the client.

Emily Lewis: You mentioned a few tools when you were describing how you optimize images. One of our listeners, Sheelah Brennan, she said to ask if there were any that you thought were ideal for non-technical content creators.

Kitt Hodsdon: [Agrees]

Emily Lewis: ImageOptim comes to my mind.

Kitt Hodsdon: Oh, yeah.

Emily Lewis: That’s the one we usually recommend to our clients to use. Is that yours?

Kitt Hodsdon: That’s also my recommendation.

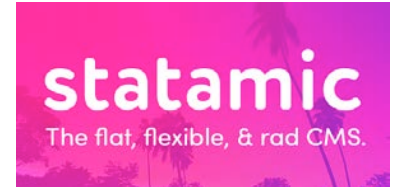
Emily Lewis: [Agrees]

Kitt Hodsdon: I strongly recommend it. I will often buy clients support for that.

Emily Lewis: [Agrees]

Kitt Hodsdon: It’s available online, but if you want support on that, you can find it and educate them on that.

Emily Lewis: [Agrees]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Yeah, I'm a big fan of ImageOptim. Some clients aren't Mac based.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: And so some of them, they have Photoshop and I attempt very strongly to steer them away because Photoshop doesn't have really great out-of-the-box compression and optimization. There's also support for that. There are filters that will optimize that a little bit better.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: But I tend to steer them away from handcrafting images. Even if they have those scripts in Photoshop, I try to steer them away.

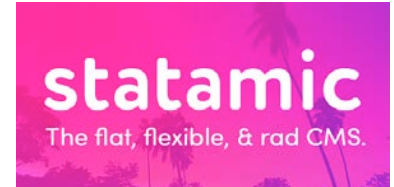
Emily Lewis: Yeah, aside from the technology or the final output, I would also be inclined to steer them away from it because it becomes this tedious chore and then they end up not doing it at all in the first place.

Lea Alcantara: [Agrees]

Kitt Hodsdon: Yes.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: And so anything that we can automate for them, and to a degree, ImageOptim will do that with batch processing, but yeah, I agree with you aside from the technical concerns, just knowing clients and knowing their work loads.

Kitt Hodsdon: Yes.

Lea Alcantara: So I'm curious because we've been talking about software that you can download, like for example, like ImageOptim, is there, say, in-browser apps or something that clients or even developers can potentially use? The first thing I thought of is for me with like SVG, I use [SVGOMG](#) to cut down sizes.

Kitt Hodsdon: I actually haven't used any of them. Usually, clients are like they want the easiest, cheapest thing.

Lea Alcantara: Sure.

Emily Lewis: [Agrees]

Kitt Hodsdon: So I have no input on that.

Emily Lewis: Okay. [Laughs]

Lea Alcantara: Fair enough.

Timestamp: 00:19:58

Emily Lewis: Well, Lea, what do you think about SVGOMG, would you recommend that to one of our clients?



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: Yeah, absolutely. I think part of the reason is because it's in-browser, so generally speaking, you could just point them to that particular thing, and also it's a drag and drop, so it's just you take a file from your folder and then you just drag into the interface and then you can tweak some settings after it's been compressed, but usually like whatever default already kind of generally works.

Emily Lewis: [Agrees]

Lea Alcantara: So if they have a ton of SVGs that they need to optimize, you can't do like a batch like select ten and then drop it in. It still is like one by one.

Emily Lewis: [Agrees]

Lea Alcantara: But it's simple enough that it can be easily taught.

Emily Lewis: This reminded me of our episode with [Brenda Storer on SVGs](#) so I'll make sure to link that in the show notes because I think she also had a lot of recommendations about how to really optimize an SVG beyond that sort of batch processing tools.

Lea Alcantara: Yeah.

Kitt Hodsdon: I'm excited on new tools.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: So, another listener, and he's actually been a guest on our show, Ben Callahan, he wanted suggestions on how we can help our clients deal with images via CMS. We sort of touched



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

on this a little bit, but I wanted to get into it a little bit more. It's something that Lea and I are constantly trying to find the right solution for our CMS. I'd love it, Kitt, if you could first sort of described what you've done with regard to what you were describing where upon an upload, a script gets triggered. Is that on the server side?

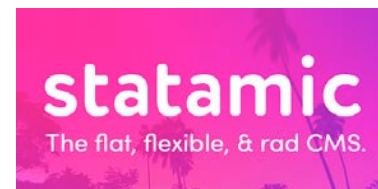
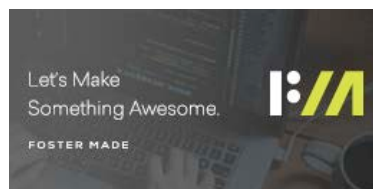
Kitt Hodsdon: It is on the server side, yes, depending on the CMS or how custom the project is. Sometimes for very common CMSs, there are processes that are already in place. For example, WordPress has responsive images, and I think Drupal has something similar, and Django has something similar where when an image is uploaded, it's resized. The difficulty with this is, for example, you have an image-heavy website, if you're building, for example, an Instagram clone.

Emily Lewis: [Agrees]

Kitt Hodsdon: Having the resizing and the optimization happening at the same time as the user is waiting for the images, it's really not good user experience, and so in those particular cases, instead of having the image as its immediately uploaded and resized, having the image uploaded, having a single resize to display to the user, then the rest of the process is done asynchronously, and that's often with either a cron job or a separate web service, depending on how complicated things were.

For example, at Shopify, we were dealing with images all the time with the products that were uploaded, and so there was this separate service that we had that handled all the images uploaded. You uploaded them, they resize them, they generate all these, and put them out to the CDN that we had. Something similar to that, depending on the size of the project, can be built for CMS for a smaller project or with fewer editors having upload, and then during the cron process, optimizing those images, creating the other ones, pushing them out to the CDN is a process.

Lea Alcantara: [Agrees]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: But it has to be automated. It has to be something that, as you have said, invisible to the user.

Emily Lewis: Yeah. So, Lea, am I correct that what Kitt was describing with the asynchronous generating of the different image resizing, is that what's part of Craft core?

Lea Alcantara: No. I do believe that there are plugins though. I think there's the [Imager](#) plugin, and then I know that there are a few others that don't come to mind immediately right now that have those types of things as part of the image optimization process.

Emily Lewis: [Agrees]

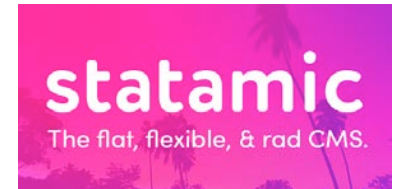
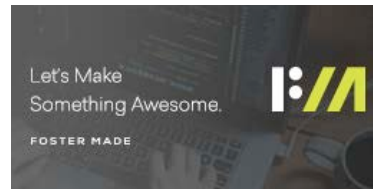
Lea Alcantara: However, it's up to us as the developer to set up the cron job.

Emily Lewis: [Agrees]

Lea Alcantara: So that it happens, say, like at one in the morning when no one is really on the site in case there are any server traffic issues while that's being generated.

Emily Lewis: Yeah, and some older EE sites that we've worked with, ExpressionEngine has some image resizing capabilities built in, but in my opinion, they're nowhere near as much as the kind of control that you really want to have, and I think what Kitt is describing with that automation would probably be the better way to go, but I just kind of want to comment, like we're still getting old legacy EE sites that our people are getting ready to move from or still need to maintain, and for a while there, we were using or we would inherit things where people used – what was it, Channel Images, Lea?

Lea Alcantara: Right, right, yeah.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: And oh, my god, it killed performance.

Lea Alcantara: Right.

Emily Lewis: Like it's supposedly helping you by generating all of these different images, but oh, it was just a really process heavy add-on on the page, and the project that we cleared it out on, we just had to wipe it out of the system because it was just such a pain, it just slowed the pages down so much. And that was one of the things we did, first and foremost, on a major performance project. It was just try and get a hold of how these add-ons were working because some of it was just so outdated and not really doing what current browsers, what current networks, what current users really need, and so if you inherit something like that and you're struggling, try and find a replacement for Channel Images. [Laughs]

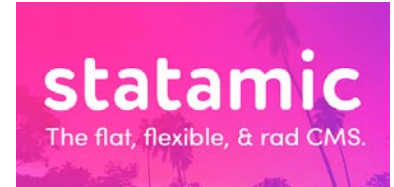
Lea Alcantara: Well, I mean, I will say that like the entire statement that this old technology, like all the things that Kitt is mentioning right now is after years...

Emily Lewis: It's the new.

Lea Alcantara: Yeah. It's after years of understanding how the web works and trying to make everything better, and I think one of the issues, too, that like a CMS developers were learning, like I remember that certain resizing wasn't even – okay, there are two trains of thought, like some resizing happened during the upload, so during the actual management of the images.

Emily Lewis: [Agrees]

Lea Alcantara: And other times, we upload the full file size and then the resize happened on page load.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Ugh.

Emily Lewis: That's the worst.

Lea Alcantara: Right? [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: I heard everyone was like, "Oh, my god," but you all know a ton of systems that used to do this.

Kitt Hodsdon: Oh, yeah.

Lea Alcantara: Because this was commonplace.

Emily Lewis: [Agrees]

Lea Alcantara: This was what people used to think was okay because people weren't optimizing for the type of traffic and popularity that people are doing now and understanding that it really affects conversions. It was just like, "Hey, it does the job, here it is," and then now there is like a spike in popularity of a post or something, and then the site crashes.

Emily Lewis: [Agrees]

Lea Alcantara: And a lot of the time you're like, "It's the images." [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: That's the truth. All right, so let's talk about actual implementing web images on websites. I feel like the answer to this is no, but I could be wrong because I'm not on the cutting edge of this, but is there a single solution that used across the board, or does it vary?

Kitt Hodsdon: For front-end?

Emily Lewis: [Agrees]

Kitt Hodsdon: I always use srcset.

Emily Lewis: Always, really?

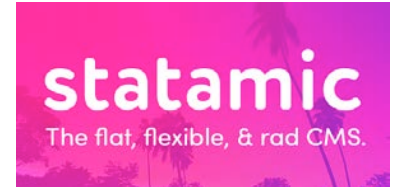
Kitt Hodsdon: I always use srcset. I have for the last couple of years use srcset as a way to send different image sizes.

Emily Lewis: [Agrees]

Kitt Hodsdon: I can't imagine not using it. [Laughs]

Emily Lewis: And that's something that because you have the tools for automating, it's not as big of an image management challenge?

Kitt Hodsdon: Yeah, exactly.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: Okay.

Kitt Hodsdén: What's important is when the images are resized, that they're resized in a way that is well-defined naming scheme.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdén: For example, my personal website resizes images and then file names are like 1237-D and 1432. One is a smaller image of the other and that does not work. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

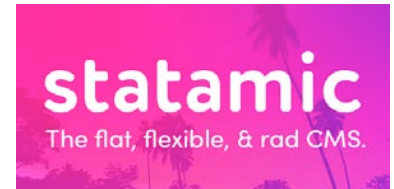
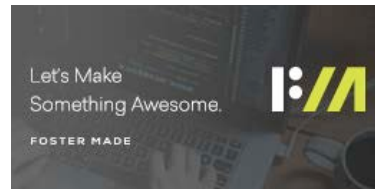
Kitt Hodsdén: Not at all.

Emily Lewis: How far do you go down the rabbit hole of image sizes, because you could get really discrete?

Kitt Hodsdén: You can. You can absolutely can. Image sizes are best defined by the design.

Emily Lewis: [Agrees]

Kitt Hodsdén: And the design is you make it as big as you can, and when it doesn't look good, that's when you change, and I think that that's been a common theme in web design for a while now.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

However, there is a bookmarklet that you can install in your browser and click through. It's Responsive Image Linter.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: It's on GitHub, and I can send you the link. It's [ResplImageLint](#) and you run this bookmarklet and it will say, "Hey, you should resize this here and you should resize this there."

Emily Lewis: [Agrees]

Kitt Hodsdon: And it does a fantastic job of...

Lea Alcantara: Oh, useful.

Kitt Hodsdon: Yes, it's wonderful. It does a fantastic job of giving you, "Here are the common sizes for screen resolutions. Why don't you try and edit this?" And oh, it is wonderful. It will not work if you have your content security policies set up properly on your website, and that's the headers that you send to the browser to say, "Here are the hosts and the domains that you can accept JavaScript from."

Emily Lewis: [Agrees]

Kitt Hodsdon: And it's important to have that on so you can reduce cross-browser JavaScript hacking sort of things and it will cause this bookmarklet not to work.

Emily Lewis: [Agrees]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: So you can test it on a local development or you can test it on a test staging room without this header sent by.

Emily Lewis: [Agrees]

Kitt Hodsdon: So you can't run it on Twitter, for example, if you try that.

Emily Lewis: Right.

Kitt Hodsdon: But it is great, it is wonderful, and it's highly recommended for anyone who is implementing responsive images.

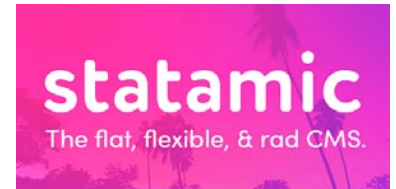
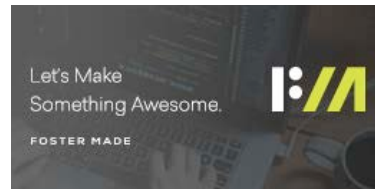
Emily Lewis: That's great, because I'll be honest, I hadn't used srcset yet until a recent project, and I was really struggling, also cognizant of the budget and how much time I could spend on this to identify the "ideal" images that I should create for the situation.

Kitt Hodsdon: Yeah, yeah.

Timestamp: 00:29:50

Emily Lewis: And that would have been a really useful tool at the time. [Laughs] Yeah, and I think it might be useful for our listeners. Can you describe how srcset works? And also the listener, Sheelah Brennan, that I mentioned earlier, she asked, "When does <picture> as an option come into play, and is that a versus srcset or is it not even an option in your repertoire?"

Kitt Hodsdon: Let me answer the <picture> element question first. <picture> elements are fantastic when we have time.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Laughs]

Lea Alcantara: Hmm.

Kitt Hodsdon: And what I mean by that, srcset is what you want to use when you just have different sizes of the same image, and the <picture> element is what you want to use when you have art-directed sites.

Lea Alcantara: Oh.

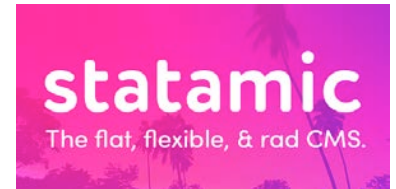
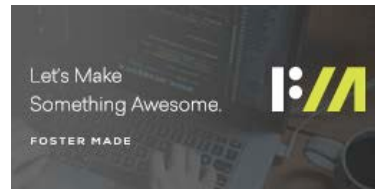
Kitt Hodsdon: So if you're having this giant picture of the White House and you just want a close-up picture of the dog, that's the classic example of the <picture> element. So it gives you an opportunity to crop the picture and do something that's important. The difficulty you have using the <picture> element with art-directed is you need someone to do the art direction.

Lea Alcantara: [Agrees]

Kitt Hodsdon: A news website does working on its content, content, content, get it up, get it up, get it up, and there's no one that has any time to say, "Oh, let's crop this image to show just the face instead of all this background." When you have a client that's uploading 10-megabyte images and they aren't optimized, and you want throttle in frustration because you really want to educate the entire organization on images, and you just don't have the time.

Emily Lewis: Right.

Kitt Hodsdon: And so resizing, getting something out, srcset is the way to go, but when there is time or when you have a client that is focused and has the resources for art direction, <picture> is always the way to go because you'll end up with focused pictures instead of this generic content picture.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Kitt Hodsdon: That's the when do I use <picture>. For srcset, srcset is the HTML that you use in order to say, "Here are my images and here is when you should display this image, and an attribute that you include on that are the sizes. So here's the image. I want you to display it this big and here's when I want you to use it." So for example, this srcset, if you have a large and medium and a small image, you can say, "Hey, when the screen is this big, you can use media queries, which is great, then use this large one when we have this tiny screen, I want you to use this image." And you can get incredibly complicated HTML.

Emily Lewis: [Agrees]

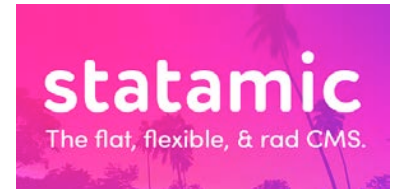
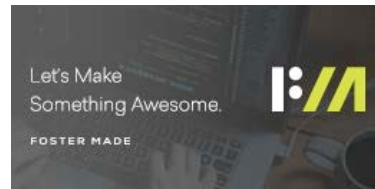
Kitt Hodsdon: And it takes a very short time for the browser to download that HTML compared to the length of time it can take to download the wrong-sized imaged. So srcset gives you the browser clues as to here's a scaled-down image that you should download instead of this giant image.

Emily Lewis: I was going to ask, are there any performance concerns? Like when you are listing the various size in the srcset attribute, is the browser downloading all of those?

Kitt Hodsdon: It is not, and that's the wonderful thing.

Emily Lewis: [Agrees]

Kitt Hodsdon: So if the browser just not understand srcset, it's going to download the image element, and source and the image element that's the companion with this srcset, but when you have the srcset, the browser is going to say, "Oh, this space is 600 pixels wide, so I want to download the image that is going to best fit in that 600 pixels wide."



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Kitt Hodsdon: And it's going to ignore the one that's 1,000 pixels. It's going to ignore the one that's 300 pixels. It's going to pick the one that fits best, and the great thing is that the browsers have lots and lots and lots of data for optimizing this stuff. They know what they're doing. I, as a developer who have worked on a couple of hundred websites, am not necessarily going to pick the right one, but I can give the browser hints, and that's what the srcset and sizes does, it gives hints to say, "Do this," which is fantastic.

Emily Lewis: And I feel like it's a nice technique in the sense that if you have an older browser, it just ignores srcset and uses the source.

Kitt Hodsdon: It does, yes.

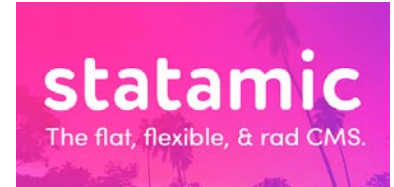
Emily Lewis: So that may not be optimized, but it's an older browser, "how far can you go" kind of thing.

Kitt Hodsdon: Yeah. I think that was the best feature with HTML. If you don't know what it is, ignore it.

Emily Lewis: [Agrees]

Kitt Hodsdon: And so you can put all these new things in, and if it doesn't know, it just ignores it, and that's like the best design decision ever.

Emily Lewis: Sheelah was also asking, and this is a question that's relevant to me, too, because I use these techniques, but do you ever do like an inline image technique with maybe like a CSS max-



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

width: 100% or skip inline images and use something like CSS multiple background images with media queries?

Kitt Hodsdon: I always do CSS max-width: 100%.

Emily Lewis: [Agrees]

Kitt Hodsdon: And I think that that's a little controversial sometimes with people, it's like, "Well, then the browser has to scale." And yes, the browser has to scale, but the browser doesn't have to download the image again, which is fantastic.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Kitt Hodsdon: Again, with the Responsive Image Linter, I can get a good idea of what the best sizes are going to be for common screen resolutions, and so if I get 80% of the cases, I'm doing a fantastic job.

Emily Lewis: [Agrees]

Kitt Hodsdon: With respect to CSS background images, I almost never do this. I've been lucky to have designs that don't require different background images, but if I needed to do that, if I needed to, the first performance enhancement I would try would be to include the CSS in my critical CSS to select the image in the first 14K of the page that I send over.

Lea Alcantara: [Agrees]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: So the browsers are going to download the HTML and then they get the CSS and they download that and then they parse that, and if I have the CSS background image buried in there, I've got two things that I need to download before the browser ever knows about that CSS or that image.

Emily Lewis: [Agrees]

Kitt Hodsdon: So I would absolutely try to put that embedded likely in the page itself without any kind of and totally inline that and have that in there instead of putting it so that the browser knows when it gets the HTML and process the HTML, it knows to start grabbing that background image.

Emily Lewis: [Agrees]

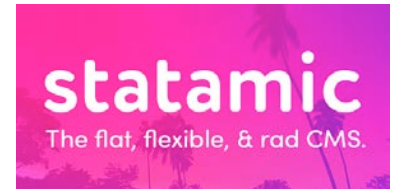
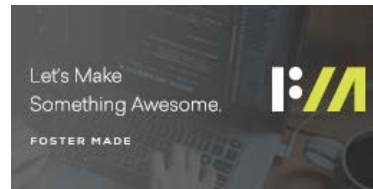
Kitt Hodsdon: But again, I haven't done that yet. I haven't had to require different background images.

Emily Lewis: Yeah, I have an interesting project, the design was such that my solution, I hope it was the right solution.

Lea Alcantara: Yeah. [Laughs]

Emily Lewis: But essentially, the way it was, I needed an inline image that then not became a CSS background image, but the media query triggered then larger views of CSS background image because of how the design worked, and it felt clunky and gross.

Kitt Hodsdon: Yeah.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: Like I couldn't come up with anything else to accommodate this particular design, and I feel like that's one of the things that that prompted me to start a document for Lea that was like, "If you see this in a design..."

Lea Alcantara: Don't do it. [Laughs]

Emily Lewis: "Don't do this in mobile to desktop because it's going to make things complicated and take us out of budget." You know?

Kitt Hodsdon: Yeah.

Emily Lewis: Because we're always dealing in budgets, that's the challenge.

Kitt Hodsdon: It is.

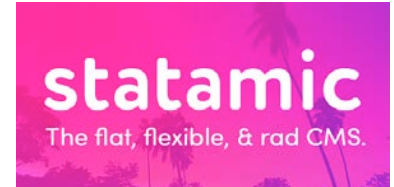
Emily Lewis: If we have an unlimited amount of time, these things wouldn't be problems.

Kitt Hodsdon: Yes.

Emily Lewis: But to create like a resource was like when you're designing mobile and then you take it to desktop, if you make this change between these images, it's actually a little more complicated than just swapping the images.

Kitt Hodsdon: Yeah. I love that you have that documentation, though. That's like notes to future self and notes to coworkers I feel are so important.

Emily Lewis: Yeah.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: So I love that you do that.

Emily Lewis: Oh, right now, as we're talking, I have all an images best practices document that I have opened at all times whenever I'm developing. At the beginning of the project, I revisit it to make sure if there's anything new that I need to check out, and at the end of every project, I update it with everything I learned along the way.

Kitt Hodsdon: I love it.

Emily Lewis: Yeah. [Laughs] I think documentation is probably one of the ways that helps both Lea and I manage all of these details.

Lea Alcantara: [Agrees]

Emily Lewis: I think it gets hard when new techniques come out.

Kitt Hodsdon: Yeah.

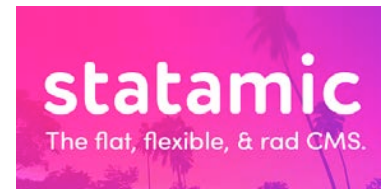
Emily Lewis: Clients have different needs.

Lea Alcantara: [Agrees]

Emily Lewis: And there are just too many things to try and just keep in your head and trust that you're going to remember it all.

Kitt Hodsdon: Yeah.

Emily Lewis: Do you ever rely on lazy loading for your images?



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: As much as I can, oh, my gosh, the fastest picture loading is the one that you don't load, right?

Emily Lewis: [Laughs]

Lea Alcantara: [Agrees]

Emily Lewis: Right.

Kitt Hodsdon: Because if you're never loading it, if the user never sees it, don't load it. The other one that frustrates me is, "Oh, just do a display none." No, don't just do display none because if you do display none, the picture is still downloaded, and that totally bars up on performance.

Emily Lewis: [Agrees]

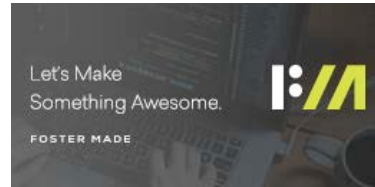
Kitt Hodsdon: But yeah, I use lazy loading as much as I can, and I tend to do gray images instead of a small one resized up in the browser.

Emily Lewis: [Agrees]

Kitt Hodsdon: That's also dependent upon the client desires, "Oh, that doesn't show what it says. Okay, well, all right, let's do this other one instead."

Emily Lewis: [Agrees]

Kitt Hodsdon: And it is a favorite technique of mine, because when someone doesn't scroll down to see that big image, don't download that big image.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Kitt Hodsdon: I love that.

Emily Lewis: Is there any reason to be concerned about using an additional script to manage that? Does that affect performance in a measurable way, or is the tradeoff just worth it?

Kitt Hodsdon: The tradeoff is worth it. You have the first page load when you first get that JavaScript into the cache. It might be somewhat problematic if the user has never been on the site, but with subsequent ones, it totally makes sense. There are libraries out there that are really tiny, too, that make it so that it's not difficult, and browsers are also embracing this technique. For example, there's Intersection Observer, which triggers on resizes and rotation. So if you want to be very, very, very particular and keep things really, really small, you can watch for that in browsers that support it, but in reality, the lazy loading libraries are small and well worth the effort to implement them in terms of saving bandwidth and perceived performance.

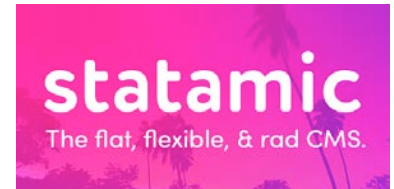
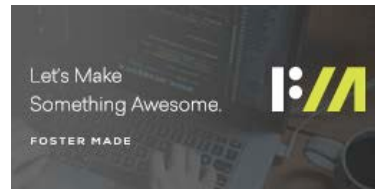
Emily Lewis: [Agrees]

Kitt Hodsdon: I highly recommend it.

Emily Lewis: Yeah. I've been implementing that in our projects for quite – I guess maybe two years now.

Lea Alcantara: [Agrees]

Kitt Hodsdon: Yeah.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: And I had a client who I guess I did some front-end templates for them about five years ago, and then I built a whole new set of templates for a different site we just wrapped up last month.

Kitt Hodsdon: [Agrees]

Emily Lewis: And the client is somewhat familiar with front-end, but like it's not her main job. She knows enough to take the templates and implement them in their system.

Kitt Hodsdon: [Agrees]

Timestamp: 00:39:59

Emily Lewis: And I was explaining to her this lazy loading, and it like blew her mind.

Lea Alcantara: [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: It was just she really, really appreciated it and she really, really loved the concept of it, and yeah, I think it's one of those things that not only does it create a great user experience, but clients can actually see the value of that perceived performance actually does make a difference.

Kitt Hodsdon: Yeah, it does.

Lea Alcantara: Can we dive a little bit more about that, like the actual performance versus perceived performance, like is there, for example, with actual performance, you've got tools to be like the page loaded this fast, right? But how do we measure perceived performance other than, "I saw something, and it seemed fast."



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Yeah, yeah. For the record, perceived performance trumps actual performance every time, because our goal...

Emily Lewis: [Agrees]

Lea Alcantara: Good point.

Kitt Hodsdon: Yeah. Our goal with performance is not, “Oh, let me load this in a second,” but our goal is, “Is the user happy with our site? Are they going to buy something?”

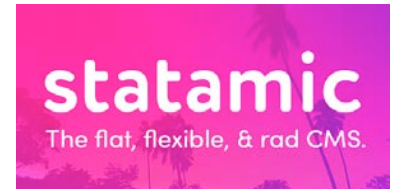
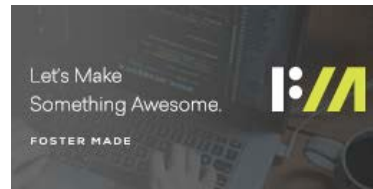
Emily Lewis: Right.

Lea Alcantara: [Agrees]

Kitt Hodsdon: They don’t care that it takes seven seconds to download versus four seconds. They want to know that something is happening. So with perceived performance, there are tools that can help. For example, webpagetest.org, which is my favorite and my go-to has “I’m loading this page. Here is when the user is going to see something. Here’s when the user can actually start interacting with the page.” And then recently, I was asked about RUM testing, which is Real User Monitoring.

Lea Alcantara: [Agrees]

Kitt Hodsdon: And those tests are also relevant. The two are not exclusive, don’t do one and not the other, if you have the time and the resources, but the real user monitoring says, “Here’s what they are seeing, and here’s what your users are clicking on, and here’s what the users are experiencing,” and that is incredibly important. Bottom line is, are you getting the click-throughs and the interest in e-commerce site? Are you getting the purchases that you need?”



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Kitt Hodsdon: But barring that, RUM testing is important as it perceives performance, and the tools are definitely out there to see when people see things.

Emily Lewis: Yeah. I'm glad we're having this conversation because I feel like it's going to come in really useful when we do our next project and kind of take a few different approaches to how we're dealing with this perceived performance thing and also the automation.

Kitt Hodsdon: Yeah.

Emily Lewis: I feel like those are two things where I feel like we can be doing even better and I think the technology is out there where it's not like this massive endeavor.

Kitt Hodsdon: Yeah.

Lea Alcantara: And I just want to clarify, like real user monitoring, for example, that's stuff like [New Relic](#), isn't it?

Kitt Hodsdon: Yeah.

Lea Alcantara: Yeah. So we do have an episode talking about [a tale of two EE sites](#), and we chat a little bit about real user monitoring using New Relic, so we'll link to that in the show notes.

Kitt Hodsdon: Awesome.

Emily Lewis: Our listener Sheelah had a bunch of questions for us, so we've got one more from her. [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: [Laughs]

Kitt Hodsdon: Okay.

Emily Lewis: She asked about services like [Cloudinary](#) or [Imgix](#).

Kitt Hodsdon: [Agrees]

Emily Lewis: Have you used those? When are those a good option?

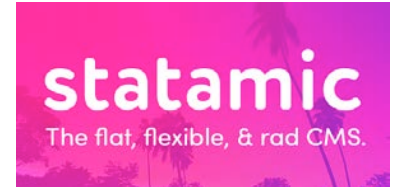
Kitt Hodsdon: I have used them and I strongly recommend those for large clients or large projects that are going to need a CDN.

Emily Lewis: [Agrees]

Kitt Hodsdon: So for example, if you have many, many users and you have many, many images and you're going to be on a CDN anyway because your user base is so incredibly diverse that having a single location is just not going to cut it, having a CDN like Cloudinary and Imgix is fantastic because they do all the stuff that you want to do. They do resizing and they do CDN part so that you have faster loading. They do, for example, Cloudinary will also, if you're requested to do the Guetzli encoding, which means the files are going to be 30% faster.

Emily Lewis: [Agrees]

Kitt Hodsdon: And it makes it easier for the end user if they use the various plugins to manage all the images. If it's a smaller project and the support for it is sufficient in the CMS or the client just isn't going to be uploading a lot of images, then doing it locally is totally fine.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Agrees]

Kitt Hodsdon: But I recommend a CDN if possible if you don't have the resources to tune those asset servers or a DevOps to manage a lot of the processes.

Emily Lewis: You know, you mentioned small, big, what is sort of the line for you? Is it really about the amount of media they're serving from their site? If they're serving videos too, does that sort of push over to say yes, this kind of service makes a lot of sense because you're serving not only a lot of images, but videos, too?

Kitt Hodsdon: Yeah, that's a great question, where that border is, and I tragically have to say it depends.

Emily Lewis: [Agrees]

Kitt Hodsdon: And it often depends not on me, but it depends on the client's resources.

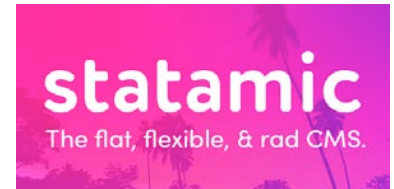
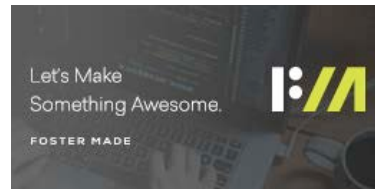
Lea Alcantara: [Agrees]

Kitt Hodsdon: Do they have someone that cares about this or do they just want something to get up and gone? Are they concerned about third parties? Do they want everything in-house? Do they have the support for that in-house?

Lea Alcantara: Right.

Kitt Hodsdon: Or just get it done sort of thing.

Emily Lewis: [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: A lot of the smaller volunteer work that I do, I don't use the CDN at all because it's typically localized. The users are sharing a small geographic area and a single server with a fallback is sufficient.

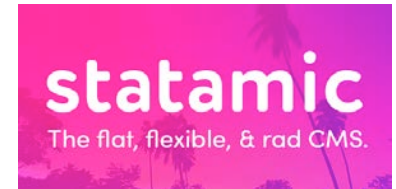
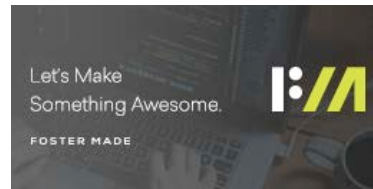
Emily Lewis: Yeah, this just stands out to me, Lea, our client that has the services in multiple metropolitan areas that this may be something that would be really useful for them as they evolve to their new website.

Lea Alcantara: Yeah, absolutely. I also like the point that she pointed out as in like do they want, especially because privacy issues are kind of top of mind in our industry right now with GDPR and all that fun stuff. If they want their images in house for whatever reason, for control or privacy versus in the cloud, what does the cloud mean? I mean, having to explain all of that and where does it get served? It gets complicated really quickly, but as Kitt mentioned, it depends on the client's needs and then the eventual needs of whatever the organization is.

Emily Lewis: [Agrees]

Emily Lewis: You know, one thing that I find really surprising is that no matter how much we advise our clients, like we can have automation in place, we can help them with resizing and stuff, but they need to start with the good image. I always feel like sometimes they upload these massive, massive images and not only can that just kind of clog up the system, but it's just not necessary, and do you have any suggestions on how to get our clients or maybe to our employers to embrace better image optimization before they're even putting it into the system?

Kitt Hodsdon: I think awareness really, really helps, and that's it's a hard thing to do organizational education because there's institutional knowledge and then there's turnover and it's really difficult to do.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: [Agrees]

Kitt Hodsdon: Demonstrating how performance helps a website before, during and after and automating that process really, really helps projects to embrace performance, and image optimization, while it's a smaller part in the whole optimization process, is still really important and something that you can see immediately. If possible, setting up something like – I don't know if you're familiar with sitespeed.io.

Emily Lewis: [Agrees]

Kitt Hodsdon: It's this beautiful interface to performance checks that are done and you set it up so that it runs frequently and displays these graphs that are at a glance, you can see, "Oops, I shouldn't have done that sort of thing for performance," and I find that that to be really helpful, but if a project hasn't started, it's about education and demonstrating assumptions before you even start. Yes, you are going to optimize this. Yes, you are going to do this, and you kind of learn by example.

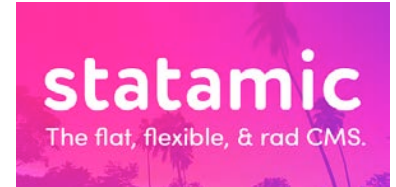
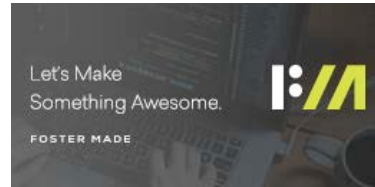
Emily Lewis: Speaking about assumptions, are there any assumptions or misconceptions that even developers have about optimizing and serving images for the web?

Kitt Hodsdon: I think the biggest misconception is that you don't have to do it. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Kitt Hodsdon: Well, of course, you have to do it. And the second one is that you have to spend days and weeks and months on it, and you don't. You really don't. The process does not have to be



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

exact, and you can tell a browser that you're sending a JPEG image, and send a WebP image instead, and it will be fine.

Emily Lewis: [Agrees]

Kitt Hodsdon: Those kind of things, for example, just getting srcsets and resized images is huge for performance, and it's just something is better than nothing. I think that's the common misconception, "Oh, this could be this horribly hard thing." It's like, "No, it's really not. You can do this."

Emily Lewis: Yeah, I feel like once upon a time, it was a hard thing and it's just gotten better, especially as automation tools have evolved.

Kitt Hodsdon: Yeah.

Emily Lewis: We'll be sure to link in our show notes to our episode that we had – oh gosh, Lea, what was his name, Dan Tello. Is that it?

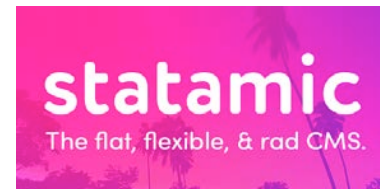
Lea Alcantara: I don't remember.

Emily Lewis: Anyways, on [Gulp automations?](#)

Lea Alcantara: Oh yes.

Emily Lewis: But we will make sure to link to that.

Kitt Hodsdon: Okay, awesome.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Lea Alcantara: So as we are wrapping up, part of the reason we wanted to have you on the show is that there are just so many disparate options about how to deal with images.

Kitt Hodsdon: [Agrees]

Lea Alcantara: What do you refer to for your favorite resources when finding out about speed optimized images, what are your favorite resources?

Kitt Hodsdon: This is going to sound kind of funny, but there are a couple of people I follow on Twitter. [Tim Kadlec](#) is one. [Addy Osmani](#) is another. [Ilya \(Grigorik\)](#) is another one. And listening to them speak and looking at what they publish is really, really helpful.

Google does an amazing job of [hosting documentation](#) and producing tools to help developers optimize, and reading a lot of their documentation really helps. They have one in particular. It's [Optimizing Content Efficiency](#) with respect to automating image optimization. I think it's by Addy Osmani, and that one is my favorite go-to to hand people. It's a little bit technical.

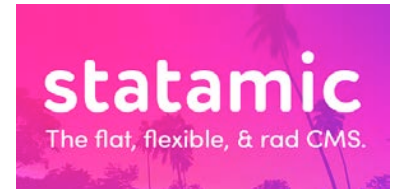
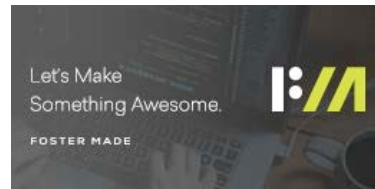
Emily Lewis: [Agrees]

Timestamp: 00:49:59

Kitt Hodsdon: So I will often fill in the gaps for them. Essentially, you want to resize your images, optimize your images, and use a CDN, and if you can't use a CDN, then optimize the server a little bit.

Emily Lewis: [Agrees]

Kitt Hodsdon: And other than that, it's all gravy.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: How about the checklist that you referenced in your Performance Matters' talk.

Kitt Hodsden: Yes.

Emily Lewis: Is that available for download?

Kitt Hodsden: It is. It's ki.tt/imgchk and that's a transcript of my talk and if you scroll all the way to the bottom, it has the four steps, which were resize your images, optimize your images, use resource hints in the CDN.

Emily Lewis: We'll make sure to link that in our show notes, too.

Kitt Hodsden: Awesome.

Emily Lewis: So any final tips for developers to work better with images on the web?

Kitt Hodsden: Resize images and automate the process.

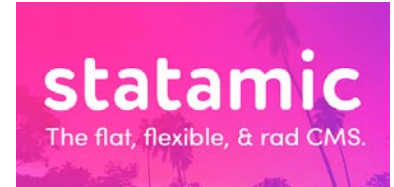
Emily Lewis: [Laughs]

Lea Alcantara: Nice.

Kitt Hodsden: Everything else is gravy.

Emily Lewis: Yeah.

Lea Alcantara: Awesome. Thank you. So that's all the time we have for today.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Kitt Hodsdon: Thank you for having me.

Lea Alcantara: But before we finish up, we've got our rapid fire ten questions so our listeners can get to know you a bit better.

Kitt Hodsdon: Oh-oh.

Emily Lewis: [Laughs]

Lea Alcantara: Are you ready?

Kitt Hodsdon: I am.

Lea Alcantara: Okay. First question, what's your go-to karaoke song?

Kitt Hodsdon: Oh, my goodness, I don't have one.

Emily Lewis: That's okay. Is there one that you like hearing others sing?

Kitt Hodsdon: Oh, my goodness. I've never been to a karaoke.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Kitt Hodsdon: I don't have an answer to this.

Lea Alcantara: Fair.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: You know, I have to say that we're midway through the year and I thought a lot more people in the web industry would be into karaoke than this question has revealed so far. [Laughs]

Lea Alcantara: Yeah.

Emily Lewis: I think maybe that's just because I work with Lea. [Laughs]

Lea Alcantara: Yeah, I know. It's true.

Kitt Hodsdon: [Laughs]

Lea Alcantara: It's true.

Kitt Hodsdon: So much for rapid fire. Oh, my goodness. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

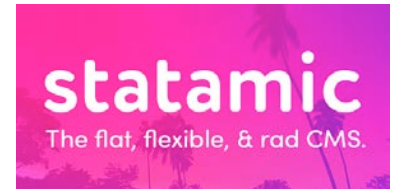
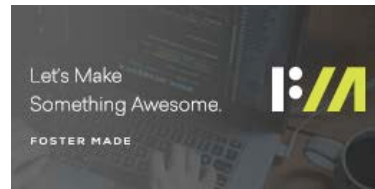
Emily Lewis: All right, what advice would you give your younger self?

Kitt Hodsdon: Do the work.

Emily Lewis: [Agrees]

Lea Alcantara: What's your favorite PG-rated curse word?

Kitt Hodsdon: Phooey. [Laughs]



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: Who is your favorite superhero?

Kitt Hodsdon: Scarlet Witch.

Lea Alcantara: Cool.

Emily Lewis: [Agrees]

Lea Alcantara: What is your favorite time of the year?

Kitt Hodsdon: Fall.

Emily Lewis: If you could change one thing about the web, what would it be?

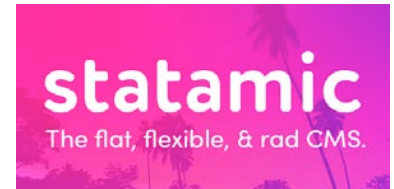
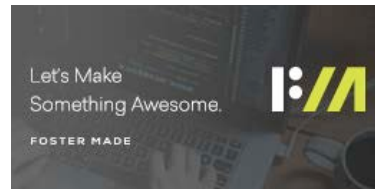
Kitt Hodsdon: I would remove mansplaining.

Lea Alcantara: [Agrees]

Emily Lewis: [Laughs]

Lea Alcantara: What are three words that describe you?

Kitt Hodsdon: Bouncy, intelligent and enthusiastic.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: How about three words that describe your work?

Kitt Hodsdon: Complicated, entertaining and interesting.

Lea Alcantara: What's your favorite meal of the day?

Kitt Hodsdon: Hmm, wow, so much with fast rapid questions, I'd have to say breakfast, but I typically don't eat until 2 in the afternoon.

Emily Lewis: We had a guest that said that recently, too. [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: Last question, coffee or tea?

Kitt Hodsdon: Tea absolutely. Coffee is disgusting.

Lea Alcantara: [Laughs]

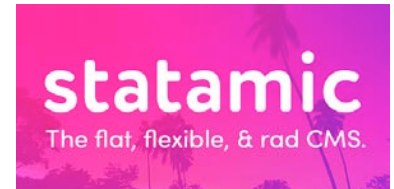
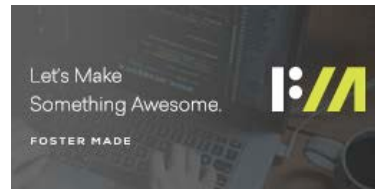
Emily Lewis: [Laughs]

Lea Alcantara: Okay. So that's all the time we have for today. Thanks for joining the show.

Kitt Hodsdon: Thank you very much for having me.

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

Kitt Hodsdon: At ki.tt.



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: Awesome. Kitt, this is really, really great. I feel like I've learned some stuff that I kind of want to start building today. [Laughs]

Lea Alcantara: [Laughs]

Kitt Hodsdon: [Laughs]

Emily Lewis: Thank you for sharing.

Kitt Hodsdon: Thank you much again for having me.

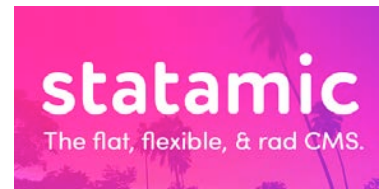
Lea Alcantara: CTRL+CLICK is produced by Bright Umbrella, a web services agency invested in education and social good. Today's podcast would not be possible without the support of this episode's sponsors! Many thanks to [Foster Made](#) and [Statamic](#)!

Emily Lewis: We'd also like to thank our hosting partner: [Arcustech](#).

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, consider donating to the show — then give us a review on [Stitcher](#) or [Apple Podcasts](#) or both! Links are in our show notes and on our site!

Emily Lewis: Don't forget to tune in to our next episode when Fabian Elliot joins the show to discuss how to improve access and democratize tech. Be sure to check out [ctrlclickcast.com/schedule](#) for more upcoming topics.

Lea Alcantara: This is Lea Alcantara ...



<https://ctrlclickcast.com/episodes/the-state-of-web-images>

Emily Lewis: And Emily Lewis ...

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

Emily Lewis: Cheers!

[Music stops]

Timestamp: *00:54:07*