



CTRL+CLICK CAST #16 Sass Workflows: Setup & Automation with Ben Frain

[Music]

Lea Alcantara: You are listening to CTRL+CLICK CAST. We inspect the web for you! Today we're talking about Sass workflows and automation with [Ben Frain](#). 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 [Hover](#). Unlike other registrars, Hover makes it easy to register a domain without being bombarded with confusing upsells and a messy interface. They include everything you need when registering a domain like Whois privacy for free and no surprise fees. If you want to know what geeks, developers, designers and programmers are raving about, why not give Hover a try for your next domain? Visit hover.com and plug in the promo code SASSY to get 10% off your first purchase.

Emily Lewis: CTRL+CLICK would also like to thank [Pixel & Tonic](#) for being our major sponsor.

[Music ends] Hey Lea, how have you been?

Lea Alcantara: Pretty good. Pretty good. I saw *American Hustle* the other night.

Emily Lewis: Oh, and what do you think?

Lea Alcantara: I thought it was pretty good.

Emily Lewis: [Agrees]

Lea Alcantara: But I was kind of confused that Jennifer Lawrence got nominated for an Oscar.

[Laughs]



Emily Lewis: Yeah. I kind of felt that way about the movie in general. I loved the whole 70's vibe.

Lea Alcantara: Yeah.

Emily Lewis: I freaking love the soundtrack.

Lea Alcantara: [Agrees]

Emily Lewis: I was in heaven from the very first song. It was Steely Dan and Dirty Work, and I was like, "That's awesome." But I thought that it could have been edited better.

Lea Alcantara: [Agrees]

Emily Lewis: I thought the storyline wasn't as compelling as everyone seemed to make it seem, and yeah, I mean, I think everyone just loves Jennifer Lawrence. I mean, I do. [Laughs]

Lea Alcantara: I mean, oh, I love JLaw.

Emily Lewis: [Laughs]

Lea Alcantara: Don't get me wrong, but I'm one of those mobster crime film aficionados.

Emily Lewis:[Agrees]

Lea Alcantara: And I was kind of expecting more of a Scorsese dark dangerous field throughout the entire thing, and it just never got there.

Emily Lewis: I feel like it was a good character story. I don't think it was as good for a plot.

Lea Alcantara: Yeah, yeah, yeah.

Emily Lewis: That was my thought, but it was fun to look at. I do love that sort of flashback.

Lea Alcantara: Yeah.



Emily Lewis: And what a reminder of how badly men dressed in the 70's. [Laughs]

Lea Alcantara: [Laughs] But I mean, I thought it was beautiful though.

Emily Lewis: Yeah.

Lea Alcantara: I thought it was a beautiful film.

Emily Lewis: All right, so let's get to some news. As we told everyone in our last episode, we've been running a few conference ticket giveaways the past weeks on Twitter. Last week, Kevin Lozandier, and my apologies for massacring your name, won a free ticket to the Responsive Web Design Summit going on right now. Congratulations to Kevin, and thanks to everyone who participated.

Lea Alcantara: And for our Peers Conference ticket giveaway, we're announcing the winner right here for the first time, so drum roll please....

[Drum roll sound]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs] Congratulations to Lise Holliker Dykes. Hopefully, I said that properly. I apologize if I didn't. We'll contact you directly to get your registration information and in contact with Jess at Peers. So thanks to everyone who played along.

Emily Lewis: And we've also got a little bit of news in the world of content management systems. First up since the release of EE 2.8, it had a lot of new features, EllisLab's blog is highlighting additions to the user guide that discuss things like the new advanced templates as well as things that developers would be interested in. We'll have a link in the show notes.



<http://ctrlclickcast.com/episodes/sass-workflows-setup-automation>

Lea Alcantara: In Craft CMS news, there's an interesting thread in Google Plus we'll link to about how to convince your boss or client to give Craft a shot.

Emily Lewis: Hmm.

Lea Alcantara: There are some really helpful tips and it's direct from the community, so the link will be in the show notes.

Emily Lewis: Nice. Speaking of community, the Statamic community is also alive and well. Their official blog has a roundup of new add-ons, how-to articles and site launches direct from the community that we will link to in the show notes.

Lea Alcantara: If we overlook news about a CMS you favor or are interested in, let us know and we'll get it on our radar for our future episodes.

Emily Lewis: All right, so today we're talking about Sass workflows and automation with Ben Frain. Ben is a front-end developer specializing in responsive web design and CSS with Sass. He's also a technology writer and author of [Responsive Web Design with HTML and CSS](#) and [Sass and Compass For Designers](#). Welcome to the show, Ben. Thanks for joining us.

Ben Frain: Well, ladies, how are you doing?

Emily Lewis: Great.

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

Ben Frain: Sure. I live in the north of England, in Cheshire, the county of Cheshire. Wife and child and another child on the way, so I...

Lea Alcantara: Oh, congrats.



Ben Frain: Thank you very much, yes. So I'm trying to store up some energy ready for some late nights. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: All the usual stuff, feeding babies and trying to look lively at work at the same time, so that will be fun.

Emily Lewis: So Ben, do you work for yourself, or do you work for an agency?

Ben Frain: No, I worked freelance for about sort of ten years, and now I work as a senior front-end developer for a company called Bet365, which is one of the largest gambling companies in the world, and they're based sort of not far from where I live. It's about sort of six to seven miles or something like that, so it's jolly convenient in that respect.

Emily Lewis: Do you go into the office regularly?

Ben Frain: Yes, yeah, everyday.

Emily Lewis: Oh okay.

Ben Frain: I mean, it's great because I get a chance to sort of sit. It's funny from doing so long freelance where I never saw anybody and the only work that I did was Skype calls or emails and that sort of stuff.

Lea Alcantara: [Laughs]

Ben Frain: It's quite nice. It's a sea change now because I go and sit and primarily I work with one of the guy, and we do sort of prototype of all the stuff that comes out from the designers. So I am



sitting with all the designers and it's great being able to sort of chat it through. There's that thing when you're building things for people, you think you understand what it is they're trying to achieve.

Emily Lewis: [Agrees]

Ben Frain: And sometimes you think, "Oh, I don't know why they did that, they should have done this." But they can now always tell me why I'm wrong and why it's that way in the first place.

Lea Alcantara: [Laughs]

Ben Frain: So it's really good from that respect being able to do that. So yeah, it's very enjoyable. I mean, I'm enjoying it a lot.

Emily Lewis: So today we're talking about Sass automation and workflows, but I kind of wanted to get started with some of the basics. Can you tell us what is Sass? And I know it's a CSS pre-processor, but what are the benefits of it?

Ben Frain: Yeah, sure. So it started out as the sort of first point of confusion is that the first version of Sass, it was all whitespace and indented, so you didn't have the curly braces that you're used to in CSS, and it was all just indented that's how you sort of got things across. That was the SASS syntax where it gets its name from, but much like a journey, a sort of life cycle if you like, they introduced the SCSS syntax which what I've always used and what most people are familiar with now. So if you like, it's described as a super set of CSS, so it's anything that's normal CSS will work in Sass, but by the same token, you can do things like variables, mixins, loops, that kind of thing which conceive. If you're used to just doing sort of HTML and CSS, it can make you a little bit easier for this. But the most basic things that I think people can get benefits from straightaway are things like being able to use partials or partials may exist that you split. Typically if you're writing CSS, you'll have a block of code which you say, "Right, this handles the header. This will handle the sidebar." Well, it allows you



to split them into individual files and join them back together again. So from a maintenance point of view, it makes it a lot more straightforward.

Emily Lewis: And you mentioned that the earlier version of Sass had a syntax that might not have been as comfortable for some people, but you've been using the SCSS syntax. Is that why you chose Sass as your I guess preferred pre-processor, or did you kind of compare it with others and then decided on using it?

Ben Frain: At the time I sort of got into it, it was like 2011 or 2012, time thereabout, and the only two sort of forums at that point were Less and Sass, and to be honest, there wasn't an awful lot between them at that point. Less had a far nicer website. The Sass website looked a little bit more impenetrable, but I basically just looked to what smart people were using. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: The people I followed and really sort of admired and looked to what they would do, and the majority of them at that point were using Sass, so I thought, "Well, if it's good enough for them, I'm sure I'll be okay with it." So there were sort of considerations as to why I didn't just follow the crowd, but at the same time, there were some things that you couldn't do in Less at that point. So I thought, "Well, even if I never get to that, I'd like the opportunity to grow into that if I'm feeling particularly adventurous." I mean, there's a quite a few that's also the style, those things like post-CSS, which is a thing you can use with node to sort of effectively build your own either pre- or post-processor. So there's a lot to choose these days, and I'm certainly not prescriptive. I wouldn't say, "That one is no good. That one is no good. But here's the one you should do." I say to people finding what works for them, I think that's the most important thing.



Lea Alcantara: [Agrees]

Emily Lewis: Now, is Sass what you used for your daily work? Is it something your entire team works or uses for work?

Ben Frain: Yeah, both. I mean, since switching to it, I very rarely or every time I dip into sort of vanilla CSS is the time I'm looking at somebody else's code. In terms of me and the projects that I work on both personally and professionally, I do everything in Sass now, and then in terms of working with other people as well. Because when I first started at Bet365, the new company, nobody there used Sass, so I just sort of sold them on the idea of it. But once you sort of sat with somebody and you've spent a couple of hours showing them what they can do with Sass and how it can make your life easy, it's a pretty easy sell to be honest.

Emily Lewis: [Agrees]

Ben Frain: But the only sort of complexity that you get is that you need to make sure that you have a system of building out. If you're sharing files between one another a lot, you need to make sure that you've got a system that can handle that. In fairness, it's not a lot different than it used to be with CSS.

Timestamp: 00:09:53

Emily Lewis: [Agrees]

Ben Frain: It was always a problem. So if you're working with another party, you're doing a better work in CSS and you upload it to somewhere and there you bring it down, do a bit of change and you upload it. They forgot to tell you that they've made a change. Do you know what I mean? There you quickly get into a sort of hot water in that respect.

Emily Lewis: [Agrees]



Ben Frain: And so I don't think there's anything exclusive to Sass there. If you got to a point where you're consistently working with another person, then I think version control is the only sort of sensible way to go.

Emily Lewis: Right, right.

Lea Alcantara: So why don't we talk a bit about the tools that you use to code and compile Sass in order to start this type of workflow?

Ben Frain: Yeah. I mean, you see, that's another thing originally when I started using it, the only way I've gotten into it is with the command line, which instantly 80% of the populations go, "Oh, forget it."

Lea Alcantara: [Laughs]

Ben Frain: But I was a bit stupid and thought I'd give it a go. So I started using that on the command line, and then it was quite facetious.

Emily Lewis: Oh my God, really?

Ben Frain: Yeah, yeah.

Emily Lewis: [Laughs]

Ben Frain: And then lots of other tools started to come about that just made it a heck of a lot easier. But well, now, I think most people sort of nearest and dearest to most people's hearts on the Mac community is CodeKit, and I think that thing was the first one that really did it well and elegantly. There were ones before that. There were a couple of tools that would do it for you, but they never felt sort of rock solid, whereas I think CodeKit was the first one that you fired it up, you tweak the settings and then you're pretty confident it was doing what it should do, and since then, I mean, I'm basically



and personally, I've been on a sort of trajectory that's more and more geeky, so I'm at a point now where I use Grunt for the most part as a tool for building and doing all that sort of front end stuff basically. I use Grunt in the meantime. I'm keeping an eye on Gulp and Broccoli and things like that. But again, to go back to that thing at the beginning, it really doesn't matter what you use. The whole point is it should make your life easier, so it's not a competition. Just find whatever tools work for you and go with that.

Emily Lewis: [Agrees]

Ben Frain: So there's certainly no shame if you want to use a nice graphical user interface and never work at command line, I can completely understand that. That's absolutely fine.

Emily Lewis: Well, that's a nice segueway for workflow, so let's talk about workflow generally. What would be a typical front-end workflow even outside of Sass?

Ben Frain: Well, I think for me, I mean, I work as a developer primarily so I tend to be given a composite of some sort to start with. Even if not just somebody's drawing something on a piece of paper, I have some idea for what I need to build. I can't even remember the last time I did something that wasn't responsive, so it always begins with a sort of a mobile first kind of approach. So as soon as I possibly can, I'll get that thing into code, get the layout put together and code and sort of build it out from there. In the past, I mean, this was another big sell for me. In the past, because I was doing responsive stuff right from off of that coming out, it was always a pain writing all the CSS and then having to write another file or a big chunk at the bottom that handled all the media queries, and sort of going up and grabbing chunks of code and then saying, "All right, but when it's in this situation, it needs to do this thing." One of the best features for me about Sass is the ability to do in-line media queries, so that means you can do, like you can say `.header open your braces and say which 100%` whatever background call, and then I can just type `please text expandable`. There are plenty of other



of those sort of things about it. I just type them and key it, and it pops a little box up and I can just say, m+, which would be like my breakpoints, and I'd type in with 40%, and that's all I need, and then when that's compiled into CSS, that generates the media query right for that block without me having to do anything. So in terms of me altering the files and looking at the code later, I could look at something .header and every single thing that happens to .header regardless of viewports or other sorts of media queries, it's all contained in that one code block so it's really handy for me, especially the bigger the project that you're building, it's then easy to just go about to find out one bit of code, and you know that anything that happens with the headers in that one single block.

Emily Lewis: Yeah, that's another thing I like about Sass as well. I remember the moment I adjusted my workflow from what you described where I was sort of grouping all my media queries at the end of the CSS document, and just going back and forth between the original selectors.

Ben Frain: Yeah, it's hard work. I mean, it was error prone as well. That was the thing if I can't find it, and it makes you feel like an idiot, and I don't want feeling like an idiot.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Ben Frain: And so my sort of thing is if I can stick tools in place that, again to automate things that I, would be just you, the human factor, get wrong, I'm always happy to do that.

Emily Lewis: Now, I feel like that example of how you can put the in-line media queries in your Sass selectors is a great example of how Sass improves a front-end workflow. You also mentioned partials in the beginning. Does that also changed your workflow and make it more automated?

Ben Frain: I wouldn't say it makes it more automated, but it makes it far more maintainable for me.

Emily Lewis: [Agrees]



Ben Frain: I love now, I mean, like partials for me are probably one of the top 3 features for me in terms of Sass. It's a very sort of on rock star feature to love, but the fact that every single little feature, I could split into its own file. I mean, you'll know if you've written CSS running all the time, in the good old days, you can end up with a CSS file that was 2,000 line long and skipping back and forth through that to find the chunk that you want. It's a bit rubbish. [Laughs] So being able to have a tiny little partial that contains just the piece of code that you need just helps you keep everything really clean, and it also means that sort of six months down the line when you're still working on that project, if that feature is no longer needed, it's easy to just delete that partial and know that it's not going to affect anything else.

Emily Lewis: Right. Do you have a general organization approach you take for your partials? For example, I've seen some people advocate putting all over their typography selectors and type partial, and then maybe lay out in a separate one. Do you have an approach like that?

Ben Frain: I do, and again, I wouldn't be prescriptive about it. I think each sort of developer or team has got to find what works best for them. What I tend to do, I'll have a base partial. I'll have a place where I'll just partial a variable partial and lay out, and then I tend to make a file I call modules and inside the modules I'll go crazy with as many partials as I'll need. It can either be individual things visually, or they might be like a feature branch. If I want to create a new feature but not sure if it's going to stay in there or not, I can make a partial just for that, and if it isn't needed in the end I can just rip it back out.

Emily Lewis: [Agrees]

Ben Frain: So it keeps the generated CSS really lean.



Emily Lewis: Now, before Sass entered the picture, were there parts of your workflow that you were able to automate, or did Sass really make the automation possible?

Ben Frain: Yeah, I think it kind of forced me into it in a way. I mean, a lot of these things were sort of came about the same period of time, so it's hard now to look back and think, "Will this have still happened if it wasn't for that?" So I think in some ways because there was a move towards pre-processors. We all started to then look at, "Well, we need better tooling to make this easier and not more difficult." Because obviously we're trying to solve problems, not introduce problems. So I think there were a few things that sort of happened. They all sort of came out at the same time like Live Reload kind of thing. As you know, the first time you fired that up on your machine, it was like, "Oh my God, this is incredible." [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: And you can't imagine going back to the days of making a save, going back to the browser and clicking refresh, and that was my life for like six years, you know? [Laughs]

Emily Lewis: [Laughs]

Ben Frain: That's all I did all day long, it was that, and once Sass actually moved, it feels incredible. So I think just a lot of these things started to sort of converge at the same point. I'm not sure, but maybe somebody can, but I can't point to one thing in particular.

Lea Alcantara: So one of our listeners, Kevin Lozandier, asked us to ask you, "What did you wish you knew about Sass sooner to make a more efficient process than the time it took you to discover such things because now you know better now?"



Ben Frain: Well, I think probably I remember the distinct moment when I looked at my code and couldn't understand why I had selectors that were about 80 selectors long.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: And there were two sort of real school boy errors that I made when I first started using it. The first one was nesting.

Emily Lewis: [Agrees]

Ben Frain: Obviously, yeah.

Lea Alcantara: [Agrees]

Ben Frain: Every time you nest a selector and another selector, the generated code just obviously it becomes an enormous one selector then. At the first one I was using Sass, I was just giddy with the excitement being able to place that within inside itself that I basically didn't realize I was making a selector sort of eight levels deep. So that was the first thing, and then the other sort of faux pas I made was with placeholders where I would extend, if I'm not pleased, I would do the extend, and I would do it on a class. So let's say I did `.thing1`, `.thing2`, `.thing3`, and then I would extend `.thing3`, I thought I would just extend what was in that one `.thing3`, but obviously, it extends everything and it extends every instance of `.thing3`. So if you used – terrible to have said `.thing3` now.

Emily Lewis: [Laughs]

Ben Frain: But `.thing3` in like six different partials, it's going to extend every single one of those even though you perhaps want to extend the one, so another thing now, and there's a good tool for that, which I'll mention in a minute, actually, but it's just always, if you're going to use extend, try to only



extend placeholders which are the ones that start with percentage sign, because that way, it will always be set in the root. You'll only be extending something that sits in the root rather than three or four levels deep, and then if this tool would have been out, I think it would have caused this a lot easier. But there's a great now SCSS linting tool which if you use Sublime or I think it's in Atom now as well or I think it's just starting working at Atom now.

Emily Lewis: Oh.

Ben Frain: It's a bit of a faster set tool because the new Sublime Linter in Sublime 3 is a bit tricky to set up, but it's worth it for me because I love the linting tool for SCSS, and it allows you to configure it so you can say, "I do care about this, and I don't care about any of these." So it's not continually flagging on things that you like, "Yes, I know that. I don't want to. Grrr."

Timestamp: 00:20:07

Emily Lewis: [Laughs]

Ben Frain: So I would definitely recommend that because that can point out things to you like, do you realize you're extending something that is in the placeholder, and all that sort of stuff.

Emily Lewis: [Agrees]

Ben Frain: So yeah. I think now the two sort of big things which would have got further on too, and in fact, all these things, you'll learn as you go, don't you?

Emily Lewis: Yeah, absolutely. In fact, as you were describing extend, I was thinking on the last bit of Sass I was writing, and I'm like, "You know what, I think I need to go back and check that."

[Laughs]

Ben Frain: [Laughs]



Lea Alcantara: [Laughs]

Emily Lewis: Because it's one of those things where for a while I was using include for a lot of things.

Ben Frain: Yeah.

Emily Lewis: And then I found extend might be a better option, but I think it forces you. Not only can it improve your workflow and it kind of presents an efficiency. It's not only for authoring and organization, but it's forcing me to look at my actual CSS a lot closer, having less specific selectors. It's kind of what you've described with the extend, which you can end up in your final compiled CSS quite messy compared to what you're looking at in your Sass code. It forces you to sort of embrace the efficiencies that you can get, but you also still have to pay attention to that end result.

Ben Frain: Yeah, and I think the funny thing is, I sort of got quite happy. First, I was really sort of look at the generated output to make sure I wasn't organizing absolutely stupid, and then I got kind of complacent, a bit cocky, "Oh, I don't need look. I know what I'm doing."

Emily Lewis: [Laughs]

Ben Frain: And I'm right back the other way now, so every sort of couple of days, I'll look at the generated code, because it's generally a big project that I'm working on, and I'll go and open the CSS files, switch on the linting tool and really go through it, and it always find something that I think, "Oops, didn't mean to do that." So it's definitely worth taking the time to do that thing.

Emily Lewis: So where else in your workflow are you able to automate?

Ben Frain: Oh, everywhere now if possible. I mean, I think this is one of the great things about Grunt, which is another of these things at first you are thinking, "Oh God, another of these funny names that I need to understand."



Emily Lewis: [Laughs]

Ben Frain: But now, I have completely got my head around it. I use it for sort of almost anything, and what I particularly like is I can find anything that can be automated. Somebody far clever than me has usually written a book in that will do what I need to do.

Emily Lewis: [Agrees]

Ben Frain: So as an example, I do a lot of stuff with SVGs at the moment, and Compass, I don't know if it has now, but it certainly hasn't for the longest time. It didn't have the ability to do image bright with SVGs, but there are now a couple of Grunt plugins that will do just that, so you can make a folder, stick your 20 or 30 SVG separate graphics in there, run the Grunt plugin, and it will generate you the image right for that set of SVGs...

Emily Lewis: Wow.

Ben Frain: And make a single SVG on the back of PNG.

Lea Alcantara: Oh wow!

Ben Frain: And the CSS that you need to put background positions in. So yeah. Basically anything that I think I'm sure a computer could do is better than me, usually it can, and so I just sort of set it up. Also, there are loads of things like that. I mean, going back to the media query thing, you shouldn't really care about whether you've got lots of different media queries lettered, and I'm done a couple of posts that sort of demonstrated that it doesn't matter, but at the same time, it will take the media queries and squish the same media query into one block, and so there are loads of little efficiencies like that you could never be bothered to do yourself, but it will do it for you. Similarly another good one is if you're writing with a team and you want to enforce a sort of code style, there's a tool called CSS Comb, so again, you can specify like the order you want properties taken, whether you want a



space between the property and the value. You can be a spit-on as you like, and once you've set it up, you press a couple of buttons, and all of a sudden, everybody's code looks the same.

Emily Lewis: [Agrees]

Ben Frain: So it's all things like that which you know you could do it yourself, but you're never going to do it yourself because you've got better things to do, but you can now let something do that for you automatically.

Emily Lewis: Now, you've mentioned Compass briefly when you were describing some of those automation tools. Can you tell our listeners what Compass is?

Ben Frain: Yeah, I mean, in the book, I described Compass as like, "Pimp my ride for Sass."

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: So if you can imagine when an Xzibit takes a car in. If the car was Sass, the first thing they would do is add Compass to it, and it's basically, I feel like it's less important now than when I wrote the book simply because there are other tools now. One of the primary things that Compass used to do was auto-prefix all the experimental CSS properties, and at the time, that was incredible. To be able to just write one syntax and know that all the different vendor prefixes were going to be taking care of. Now, I think there are better tools like there's a thing again which you can use with Grunt called Autoprefixer, and what's great about that is, I mean, Compass is using the same sort of set of information now, but you write normal W3C experimental code and your Sass or your CSS, it doesn't matter which you're using, and it's a post-processor, so once your code is saved, Autoprefixer fix it up. It looks at what browser versions you've specified. So let's say, for example, I've set it at Android 2.3 and Explorer 9, it knows which properties are needed, which prefix it is, and it



automatically parses them through after them. Now, what's good about this is you don't need to think about it. You get to write just W3C code. You don't need to write a foreign syntax like you have to with Compass, and generally speaking, what comes out the other end is a lot like live too because you're not adding prefixes to things. Because I used to be in the habit of just going around and just putting every prefix on this.

Emily Lewis: [Agrees]

Ben Frain: I know in MS. XML, XHTML and just bind them all in there. I just wanted to make sure everything is okay. But a lot of the times, you don't need them. You think you do, but you don't, and obviously, a computer is far smarter and knows what to put on them, so what comes out at either end is just a lot leaner, and it's so incredibly quick. You don't even notice it hardly, Like I was quite so skeptical at first to think, "You know, if it's going to parse this, go and find out." But it's split second. We're talking fractions of a second, even on a huge CSS file, so that does that better than Compass now, in my opinion. But the other part that Compass is great, it gives you a lot of frameworks, like there is a lot of fantastic grid systems like Suzy which builds on top of Compass. So I wouldn't say don't consider Compass, I think it's perhaps less important than it once was because we used it so much for prefixing.

Emily Lewis: [Agrees]

Ben Frain: But it certainly is one of the great tools. Does that make sense?

Emily Lewis: Yeah, absolutely.

Ben Frain: Yeah.

Emily Lewis: Now, you've mentioned Grunt a couple of times. What was the learning curve on that for you?



Ben Frain: Yeah, it was tough. [Laughs] Simply because I'm not a JavaScript guy so all of the configuration in Grunt is basically handled by what's called JSON object, which again it's all these acronyms that make your head blow out.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: Once you understand the basic sort of idea of a JSON object, it's basically like a curl. It's not like CSS. It's a curly brace, and then you have like, imagine, like a property and then a colon and then the setting. So that's essentially it. Once you understand that, what Grunt does, it basically just go down the file and runs through a task and when it's finished that task, it runs the next task, just down and down and down, and I'd say it's quite declarative which is good if you use this sort of CSS because you can say, "Take this file here, do this to it and then export it to this file over here." It gets a bit of stick from people to say that it's verbose, and I'm perfectly happy with that. But I think once you've got it set up, I mean like that's all I use Grunt with it now, so it compiles my Sass. It compiles all my different JavaScript files. It optimizes all the images and it makes my SVGs bright. All these things which I wish I was aware before. It just tumbled so it is a learning curve. I mean, Chris Coyier has done a very good post. I think it was on *24 Ways If You've Never Used Grunt Before*. It's a nice lengthy post, but I mean, he always explain things well anyway. But particularly, if you've never used it for, go and read that first, and then just thankfully, there are lots of posts about that now just explaining how to get Grunt to work for you, and I think it's well worth it. If you set and write the JavaScript or CSS for the most part of your day, I think it's well worthwhile.

Lea Alcantara: So I wanted to take a little bit of a step back. Earlier where you were talking about how there are some things that we used to do like putting every single prefix and all these things because you thought just in case, right?



Ben Frain: Yeah.

Lea Alcantara: But our listener, Kevin, actually had a great question that kind of goes towards that, which is, “What are the common attempts of optimizing that you now find superfluous or wasteful to do too early in the project beside putting every prefix under the sun, and why?”

Ben Frain: I suppose it’s anything that I used to do manually.

Emily Lewis: Yeah.

Lea Alcantara: [Agrees]

Ben Frain: So things like optimizing images, things like if you do want to squish like media queries together, even things like concatenated JavaScript sometimes because there’s now tools and build chains that you can sort of make. I don’t consider myself a complete code head, and yet, I can still gets it out at the end which is exactly what I need, because there are tools that let you do that now. I basically don’t worry about any of that stuff until I can see something in front of me that looks, behaves, and acts exactly as I would like to, then I’ll worry about all the optimizations at the end. Because often you find that you iterate so much when you do a project, particularly with responsive ones.

Emily Lewis: [Agrees]

Ben Frain: Let’s say you’ve only been given a composite of the mobile site, or the tablet sort of view, and if you’ve only got one viewport composite, and there’s going to be so much iteration as you walk through all the different sort of sizes it could be seen at, but until you’ve looked at it on loads of different devices and you’re actually certain you’re happy with it, that’s the time then I think to start worrying about optimizations. Because at the end of the day, a lot of the optimizations we make that you’ll start, they don’t make the product better for the end user, they make it faster, which in turn I



know makes it better, but you can still test whether that's the right product or not before you start doing the optimizations, I think.

Timestamp: 00:30:02

Emily Lewis: I'm curious, you do a lot of responsive work. Have you noticed that the things you were doing to automate your workflows are freeing you up so that you have more time for, like you described, the additional iterations that are involved in a responsive project?

Ben Frain: I think so. I mean, people tense up on this because if you can do more faster, people just expect you to do more. [Laughs]

Emily Lewis: Right. [Laughs]

Ben Frain: So I mean, I'm looking at it. Where I'm working at the moment, we really get a chance to sort of refine stuff, and even if we don't like it, we knock it down and start again, and certainly, in previous years, if I was working for an agency, it's not that it was a sausage factory exactly, but you get very imposed, rigid timeframes of when things have got to be done. So I think what sort of happened is we've had tools to automate a lot of the stuff, but by its nature, because of responsive work, everything has got a lot more complicated in terms of you're not just giving it in 960 pixel like a composite, you're now doing something that's going to look across multiple platforms, multiple sizes, and I mean, lots of sort of these. So I think what actually happened is if you imagine a seesaw as the two ends got better, what we've needed to build has got greater as well so its sort of state is an equilibrium.

Emily Lewis: Now, you mentioned that when you first joined the company you're with now, that they weren't using Sass and you sold it on them. How did you define a workflow for the people who are working with you on your CSS, your Sass, your JavaScript?



Ben Frain: Yeah, I mean, at first, because having to get them buying it first just that Sass was worth doing, how I said that, that was fairly easy. But then because I was always working with at least one of the persons on the code, the old sort of cowboy method of copying the CSS, generate CSS up to a folder somewhere and then bringing it down, it very quickly became untenable. So pretty quickly, even I had to use Git myself at that point, I've never used it or sat with somebody using to version a project we were making together. It's just been for my own paranoia of backing up projects. So we moved to that, and again, there's certainly been a learning curve there as well getting my head around Git, which is another of these acronyms, oh god.

Emily Lewis: [Laughs]

Ben Frain: But in terms of what that has done, it lets you share everything, and again, because everything is like a flat file like Grunt, it just uses two files as a Grunt file and a packaged up JSON file so you can include those with the repository that gets pushed and pulled around everywhere. So they have little problems in terms of things going wrong basically. So again, it's difficult. I mean, I think it's more difficult because sometimes, particularly if you're in an agency and things like that, it's difficult to sort of stop the world and say, "Right, we're going to do this for the next three weeks, so we understand it."

Emily Lewis: [Laughs]

Lea Alcantara: [Agrees]

Ben Frain: Because obviously, at the end of the day, you're trying to get pixels on the screen and get paid for doing that, but if you possibly can, the situation I've been in one was freelancing as I used to do. If there were things that I want to try, I'd make a sort of list of and I'm trying to do just one on each new project. So if I've never used Sass before, I'd go right on this project. I'm going to start



with Sass, and that's the only thing that I'll change, and then on the next project, I'd go, "Okay, well, I'm going to stick with Sass, and this time I'm going to do Git version control on it as I go." And that's how I kind of built it up myself, and in that way, you kind of you don't feel completely overwhelmed.

Lea Alcantara: So I'm curious, with all these changes in your workflow as well as trying to communicate with the team, do you have any particular documentation tips? What do you guys do to communicate this type of workflow, and is it written down anywhere?

Ben Frain: Yeah, I mean, I'm quite, as you probably know, if anybody ever goes to my site, I turn anything out, I learned I think it could be useful. I do try and document it because, A, my memory is terrible, and also it means that I've got a sort of record of how and why I solved that problem the way that I solved it, but I do the same thing in the code, I try and document it as heavily as possible because my sort of feeling is that if somebody doesn't want to read the comments, then they don't have to, they can delete them and it gets stripped out when it goes to CSS anyway. But the plus side of that is, if I'm not there or there's another team that picks this code down the line, they can at least have a really sort of thorough read of what I was trying to do even if they decided to throw it out. So documenting the code themselves and then sometimes there are philosophical decisions you make when you are designing something, and then I will often just put a folder called Docs in the root of the project, and if anybody wants to make a particular bit of documentation, something that would usually be a blog post, that kind of thing, you just stick a markdown file or a text file or whatever you want in there, put, let's say, your name and the date you made it and write as much as you want, and then again it's a record there that follows the whole project around, and obviously, it never goes on the site. But anybody else comes to that project down the line, they can go into the Docs and see what's there. So it's low tech, but I found it's very useful.



Emily Lewis: So what would be your top tip that you would tell someone who is looking to improve their workflow?

Ben Frain: If you are a HTML and CSS sort of person and you've never done command line stuff, and that gets you a little bit worried, go and buy yourself a copy of CodeKit or I think there's one called PrePos which is a free sort of similar version, and just drop an existing project, get the CSS, drop it in, and ask for the CSS so you've got a Sass file that will work perfectly well, and then just try out a few of the features so you get used to using Sass a little bit. Don't worry about all the fantasy things you've heard about mixins and functions and all that sort of stuff. All you need to worry about are some simple things like maybe splitting your code open to separate files and variables. Do that on your first project and see how you feel. I always sort of try to say to people, "Just walk before you try and run."

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Ben Frain: Because if you try and do everything on the first time out with a new tool or technique, it can really put you off it and make you feel like you never want to see it again for the rest of your life.

Lea Alcantara: [Laughs]

Ben Frain: And it's all about you're always trying to make your life better, your professional life easier, more consistent, so just be kind with yourself. Don't set yourself to higher challenge, and then all the other stuff can come down the line if it's for you, but even something like CodeKit, if you just used the features that are in there, it will get you so far along in terms of automation because it will do things like squishing all your files together, optimizing your JavaScript, linting your JavaScript, and so if you're like me, I consider myself a beginner with JavaScript, and it will point out all the schoolboy



errors that I make and tell me, “You’re missing a semicolon here. This function is outside scope.” And things like that. But without that, I just get the screen basically because I will never be clear why it wasn’t working. So yeah, I think that will be it.

Emily Lewis: I also just think it’s really good to echo what you said earlier about not trying to do everything at once. Certainly, I don’t even know what the timeframe is, but I’d guess I’ve been writing Sass for about eight months or nine months now, and it’s sort of introducing new tools into my own workflow and I’ve done exactly what you described as I just say, “I’m going to try this one discreet thing for this project and see how it goes, see what I learn from it.” When the next project comes up, if what I learned seemed really good, I do that and then I add the next thing to it. Because otherwise, it’s completely overwhelming. [Laughs]

Ben Frain: Yeah, absolutely.

Lea Alcantara: Speaking of like geeky things to learn and new things to learn, we have a final question from our listener about Sass 3.3. Are you familiar with it?

Ben Frain: I am. I hope so. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: We’ll find out in a minute, won’t we? [Laughs]

Lea Alcantara: All right. So well, his question was, “What meta programming features of Sass 3.3 do you enjoy using towards accelerating your workflow of testing CSS?”

Ben Frain: Yeah, it’s quite funny because I remember the feature of Sass 3.3 coming out, and there was sort of a list of how to look on a chap called Hugo Giraudel, I think that’s how you pronounce his



surname. He did a post on David Leuliette's blog. Have a look at it. I was thinking, "I don't care about that. I don't care about that at that point because I'm never going to use that." One of the features was being able to make what's called maps which will let you do sort of property value pairs, which you can then use a function to pluck out values through. I thought, "I'm never going to use that one. Where do I need that for?" And about four days later, I was using it because somebody had sent me this massive file.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: If you can imagine like header colors with a different top and bottom border color, and there was no sort of specific logic of the borders always 10% more, the bottom border is always 10% less. So I've been sent it. Basically, I was able to use the maps features to just stick it to an object and then auto-generate about a 120 selectors with the right values, and it took me like an hour to set up, but if I had done it by hand, it would have been most of the day, so that was fantastic, and I have to sort of be humbled by it.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: Because I've been moaning myself about it that I thought it was useless, and then it's probably the most useful thing I've used so far.

Lea Alcantara: Well, very cool. Very cool. Well, we got through a lot of really interesting Sass and workflow items. But before we finish up, Emily and I usually have this thing called rapid fire ten questions. It's based on Inside the Actors Studio, but really geeky.

Ben Frain: Yeah.



Lea Alcantara: So our listeners can get to you better. So we basically ask you a question and the first thing that comes to mind you give your answer.

Ben Frain: This is scary, I think so.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: [Laughs]

Lea Alcantara: So are you ready?

Ben Frain: Yeah.

Lea Alcantara: All right. First question, Mac OS or Windows?

Ben Frain: Mac OS.

Emily Lewis: What is your favorite mobile app?

Ben Frain: Tweetbot, I suppose.

Lea Alcantara: What is your least favorite thing about social media?

Ben Frain: Facebook.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

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

Ben Frain: Filmmaking.

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



Ben Frain: Bin man.

Emily Lewis: What's that?

Ben Frain: A bin man.

Lea Alcantara: Trash guy.

Ben Frain: A garbage man.

Emily Lewis: Oh, okay.

Lea Alcantara: Yeah.

Ben Frain: Yeah.

Emily Lewis: Sorry, I'm American. I don't know.

Ben Frain: Yes. My apology. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: A bin man.

Emily Lewis: Who is the web professional you admire the most?

Ben Frain: Oh God, there are lots. Probably Paul Irish.

Lea Alcantara: What music do you like to code to?

Ben Frain: Terrible pop.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]



Emily Lewis: What's your secrete talent?

Ben Frain: I make a good cup of tea.

Lea Alcantara: So what's the most recent book you've read?

Ben Frain: It was probably *The Book Thief*.

Emily Lewis: And lastly, *Star Wars* or *Star Trek*?

Ben Frain: *Star Wars*. *Star Wars*.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: Oh, only the last three, not the first three. Not the abominations they made afterwards.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Ben Frain: I want to make that clear. Make sure that stays in!

Lea Alcantara: Good clarification.

Ben Frain: Yeah. [Laughs]

Lea Alcantara: So that's all the time we have today, Ben. Thank you for joining us. But before we let you go, I believe you have a special deal for our listeners?

Ben Frain: Oh yes. Basically, anybody that would like to purchase one of my fine tomes from the PacktPub website, you can use offer code, CTRLCLICK, and you will get 35% off the e-books.

Lea Alcantara: Fantastic.



<http://ctrlclickcast.com/episodes/sass-workflows-setup-automation>

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

Ben Frain: I'm basically Ben Frain everywhere, so benfrain.com and Twitter has the same handle, GitHub has the same handle, so it's [@benfrain](#).

Emily Lewis: Thanks again, Ben, it was a pleasure talking to you.

Ben Frain: No problem. Thank you very much for having me.

Lea Alcantara: [Music starts] We'd now like to thank our sponsors for this podcast, [Hover](#) and [Pixel & Tonic](#).

Emily Lewis: We also want to thank our partners, [Arcustech](#), [Devot:ee](#) and [EE Insider](#).

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](#).

Emily Lewis: Don't forget to tune in to our next episode when we will be talking about refactoring web interfaces with Jina Bolton. 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: 00:42:05
