



## CTRL+CLICK CAST #7 Modular CSS with Jonathan Snook

[Music]

**Lea Alcantara:** You are listening to CTRL+CLICK CAST. We inspect the web for you! Today we're talking about modular CSS with Jonathan Snook. 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 [Perch](#). Perch is a content management system for projects where you need a lighter, faster and less expensive CMS solution. It's fully HTML5 and responsive web design friendly and doesn't mess with your front-end code. Perch is a one-off license cost of \$79 per website, and all first-party add-ons are completely free. Right now, there's a 50% discount offer for web designers and developers who would like to try Perch on their own sites. Go to [grabaperch.com/ctrlclick](http://grabaperch.com/ctrlclick) to try Perch and find out more.

**Emily Lewis:** CTRL+CLICK would also like to thank [Pixel & Tonic](#) for being our major sponsor of the year. [Music ends] Hey Lea, how are you doing?

**Lea Alcantara:** Not bad. I'm counting down to the holidays.

**Emily Lewis:** I know! I can't believe how quickly they're coming around.

**Lea Alcantara:** Yeah, I mean, just the end of the year rolling around too, and I'm a little bit nervous because I'm so excited to have all the holiday gorging happening. [Laughs]

**Emily Lewis:** [Laughs]

**Lea Alcantara:** But at the same time it's just like, "Oh, regrets." [Laughs]



**Emily Lewis:** [Laughs]

**Lea Alcantara:** Lots of regrets. [Laughs]

**Emily Lewis:** I know. My sister is coming to visit me for a week, and she's started a Pinterest board of all the recipes of all the food that wants to eat while she's here. [Laughs] It's kind of insane if we're going to make everything that's on it.

**Lea Alcantara:** But that's half the fun, you know?

**Emily Lewis:** Yeah.

**Lea Alcantara:** Just making the stuff together, I think that's fun.

**Emily Lewis:** Yeah, yeah, absolutely, and you're heading back up to Edmonton in December, right?

**Lea Alcantara:** Yeah, yeah, I'm looking forward to that too, although this will probably be the first Christmas I don't cook anything because I don't live there anymore. So everyone will be doing the cooking for me.

**Emily Lewis:** Oh, well, you should cook something special in your new kitchen.

**Lea Alcantara:** Yeah, yeah.

**Emily Lewis:** [Laughs]

**Lea Alcantara:** I mean, I could whip something up. We'll see.

**Emily Lewis:** [Laughs] All right, so let's get to some news. First, starting with news for CTRL+CLICK CAST. I'm just letting everyone know we still have a site sponsorship available. It's a 125 x 125 ad spot on the site. You can visit [ctrlclickcast.com/advertise](http://ctrlclickcast.com/advertise) for more information.



**Lea Alcantara:** Awesome. So we also have some news in the world of content management systems.

**Emily Lewis:** [Agrees]

**Lea Alcantara:** First up, Craft came out with a huge update with 100 new features and improvements and the biggest improvement and feature from that is a new field type called Matrix. Now, this name sounds very familiar to those in the EE community, but it's not like the EE Matrix fieldtype. It's more like a flexible fieldtype with multiple blocks that can contain even more fields, and for me...

**Emily Lewis:** So like a Matrix in a Matrix?

**Lea Alcantara:** Yeah. It sounds more like they should have named this Inception instead of Matrix if they wanted to go toward the movie references.

**Emily Lewis:** [Laughs]

**Lea Alcantara:** And I have to say I love all the movie references they've been making for their marketing push on Pixel & Tonic. It shows their geek colors.

**Emily Lewis:** [Laughs] Also, some news from Perch. As Lea mentioned, they are the sponsor for this episode, and they are offering a 50% discount on a Perch license for agencies and freelancers that want to use Perch for their own portfolio or company sites. You just have to send an email with your name and the URL of your agency or portfolio site, the one you want to use Perch on, and we'll make sure to include a link to those details in our show notes.

**Lea Alcantara:** And in the world of ExpressionEngine, something interesting: Anna Brown recently asked on Twitter whether EllisLab should open up the bug tracker so that Google could index them for easier troubleshooting.



**Emily Lewis:** [Agrees]

**Lea Alcantara:** She actually even linked to a little Polldaddy poll for people to just answer them. Me, personally as a developer, I voted yes because in my mind, I'd like to find the issues as soon as possible.

**Emily Lewis:** [Agrees]

**Lea Alcantara:** And Google is just a stronger search engine.

**Emily Lewis:** Yeah.

**Lea Alcantara:** And often it's the first place I look whenever something goes wonky. But what's interesting is that on the Polldaddy pages, a little bit of back and forth about the conversation. It seems people are a little bit divided about the issue, though right now it's leaning towards more people preferring Google indexing it.

**Emily Lewis:** [Agrees]

**Lea Alcantara:** And what's interesting is Anna does note that before the EllisLab site redesigned, the bug tracker was being indexed already.

**Emily Lewis:** Oh, okay.

**Lea Alcantara:** Yeah.

**Emily Lewis:** Yeah, so I haven't looked at the comments, but the people who don't want to see the Google indexing, what's the reasoning there?

**Lea Alcantara:** I think some people, well, part of it is they believe it could be a potential security risk, although...



**Emily Lewis:** Oh, like for hackers to find some back way in.

**Lea Alcantara:** Yeah.

**Emily Lewis:** Okay.

**Lea Alcantara:** Yeah, but I mean, that's also kind of a nonstarter in that, well, the code is open anyway. I mean, it's not open source, but you can take a look at the code and do whatever you want with it and use that to hack it. But I guess having a bug list is a more obvious way to find that. Some people, on the other hand, say it's not necessarily about the security stuff and more about sales. Some people think that having an easy to find bug list on Google has a perception that there's a lot of problems to ExpressionEngine. Whether that's true or not, well, that's one of the issues that those in the No camp have.

**Emily Lewis:** Well, it will be interesting to see what, if anything, EllisLab does with those results.

**Lea Alcantara:** [Agrees]

**Emily Lewis:** Well, that's about it for news in the CMS world. If we overlooked anything about a CMS you favor or interested in, let us know and we'll get it on the radar for future episodes.

So today, we're talking about modular CSS with one of my favorite Canadians, Jonathan Snook. Jonathan is a designer, developer, author, speaker and all around good guy. He maybe best known for his blog, [snook.ca](http://snook.ca), where he shares some really great web tricks and tips. But he's also a prolific writer who has contributed to just about every major publication in our industry, and because he just can't get enough of sharing what he's learned, he self-published [Scalable and Modular Architecture for CSS](#) (SMACSS), which dives into best practices on CSS architecture. Welcome to the show, Jonathan! Thanks for joining us!

**Jonathan Snook:** Well, thanks for having me. That was a wonderful intro.



<http://ctrlclickcast.com/episodes/modular-css-with-jonathan-snook>

---

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

**Jonathan Snook:** It's kind of hard to ... that was fantastic.

**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** Well, you know...

**Emily Lewis:** You owe me a beer. [Laughs]

**Jonathan Snook:** [Laughs] Absolutely. Well, I'm a designer, developer, and now, a product manager. I work for a lovely little company called Shopify based here in Ottawa, and yeah, that's pretty good addition I think on to what you said.

**Lea Alcantara:** Awesome. So speaking of all the things that we mentioned in the intro, specifically your blog, what was your motivation for starting that and offering so much information to folks, for free no less.

**Jonathan Snook:** It was in a case of wanting to just document the stuff that I was running into. I mean, way back in like 2001, I was finding myself working on issues and felt I wanted to document that and share it, because the web community, even back then (or well, before then) has always had a very open atmosphere to it. It was always really nice to just hop online and see a ton of online resources, and I felt like I wanted to do the same.

**Emily Lewis:** Yeah, I think that was my own motivation for starting a blog, and it was inspired by you and others who had done it for five, ten years even before I even considered putting a blog out there. But yeah, documenting for your own purposes and then the bonus, you get to share it with other people.



**Jonathan Snook:** Yes, very much so.

**Emily Lewis:** So in terms of writing, did your blogging experience lead you into writing for industry publications, or was that something that you sought out and directly contacted publications to write for them?

**Jonathan Snook:** I definitely made a very concerted effort at one point, and I think it was around 2005 or 2006, and I looked to other online magazines, Digital Web was one of them.

**Lea Alcantara:** [Agrees]

**Emily Lewis:** [Agrees]

**Jonathan Snook:** Garrett Dimon actually used to have a site of his own called Your Total Site, and I collaborated with him on that as well, and I used that as an opportunity to, one, get my name out there to an audience that may not have heard me. I mean, I have a blog, it's a little thing on the web. If people don't see it, they don't do a search and happen to come across it, they're just not going to find me. And so I felt like I needed to find other sites to contribute to as a way of also getting traffic back.

**Emily Lewis:** What about the two books you wrote? Was that something where a publisher came to you based on what you'd already put out there, or was it something you also took an intentioned approach to?

**Jonathan Snook:** That was definitely a little bit more chance. I happen to be at SXSW and I was talking to Chris Mills who at that time was an editor for Apress and Friends of Ed, and I was just chatting. I mean, at this point, I really hadn't considered writing a book and then talking about writing for a book. I thought, "You know what, well, that would be pretty cool to do that sometime." He turned and said, "Well, would you like to write a book?"



**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** And I thought, “Okay, I shall write a book.” And we took it from there. It was a lot of fun.

**Lea Alcantara:** So did all that writing experience somehow translate to speaking as well, like why did you pursue speaking about the topics you wrote about?

**Jonathan Snook:** It was another opportunity too. I mean, to be quite frank, actually, it was an opportunity to travel.

**Emily Lewis:** Oh!

**Lea Alcantara:** Sure.

**Jonathan Snook:** To get to go to cool places that I hadn’t really been before, but also using the opportunity to share in a different environment, so it got my name out there. So in the sort of marketing sense, I felt like it was worthwhile. I saw other people in our community doing very well, and then I saw that opportunity to become a part of that community.

**Timestamp:** 00:10:06

**Lea Alcantara:** Yeah, that makes sense. I remember in [our episode with Jenn Lukas](#) and she was talking about the benefits for her for speaking was she got to travel where she never would even consider traveling and getting an opportunity to see the world essentially through speaking opportunities. But speaking of speaking, I saw on Facebook that you had an interesting speaking experience recently.

**Jonathan Snook:** [Laughs]





**Lea Alcantara:** That the power went out when you did your presentation. Why don't you tell us a little bit about that for a second?

**Jonathan Snook:** That was a really fun, interesting day. It was funny because my coworker actually asked me a question. She was attending the conference and said, "You know, how do you prepare for the unexpected?"

**Emily Lewis:** [Laughs]

**Jonathan Snook:** I don't really. I have my slides and I just hope for the best, and I haven't really run into any technical issues to date. I've been pretty lucky in that regard, and with that said, I do try to keep my presentation fairly simple. I don't do audio. I try to make sure that I don't have a lot of animation or anything like that. So the technical issues that I run into tend to be very small.

Now, in this particular case, we were at the conference. We were two sessions in, and I was coming up next. The speaker before me was just finishing up with Q&A, and the power went out. It turns out that the region was doing rolling brownouts where they shut off the power for a while.

**Emily Lewis:** Oh.

**Lea Alcantara:** Oh no.

**Jonathan Snook:** Yeah. [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** And I was in the Green Room and just kind of sitting there and thinking, "Okay, well, I'm coming up next." The organizers came down and they said, "Okay, can you actually do your presentation without power?"

**Lea Alcantara:** [Laughs]



**Emily Lewis:** [Laughs]

**Jonathan Snook:** Normally, the presentations I would normally do have code, they have like stuff you need to look at to understand what it is that I'm talking about. And it just so happened for this particular conference I had written a completely new presentation that didn't really need anybody to see the slides, and so I said, "Okay, you know what, I can actually do this one without power." So I ended up hopping up on stage. They had the emergency lights on so people could still see me and set up my laptop on the ground.

**Lea Alcantara:** Wow!

**Jonathan Snook:** So at least I could still see the slides. I could see what was going on.

**Lea Alcantara:** Just wow! [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** Yeah, and just project it to the crowd. I didn't have a mic since there was no power. Well, it worked out actually. The feedback seemed to be really positive. I was really happy to have been able to still go through with what I needed to do, and by the time my session was done, the power was back on and the rest of the day was okay.

**Emily Lewis:** Oh my God. Oh, I just don't know if I would have been able to pull that off myself.  
[Laughs]

**Jonathan Snook:** [Laughs]

**Lea Alcantara:** Yeah, kudos, kudos.

**Emily Lewis:** How many years have you been speaking? Like how many years does it take to get to that point? [Laughs]



**Jonathan Snook:** Well, my first talk was in 2006 at WebVisions in Portland, and it was a horrible talk. Well, okay, it was kind of interesting in that I, at that time ... it was about JavaScript and I wanted to talk about a whole bunch of different things. It was just topic after topic after topic. I just threw a ton of stuff at the audience, and I had people come up after me saying, “I really enjoyed your talk. I didn’t understand any of it.”

**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** Well, if you didn’t understand any of it, how was that a good talk? [Laughs]

**Lea Alcantara:** Yeah, yeah.

**Jonathan Snook:** So from there, I mean, I didn’t speak again until actually at SXSW the following year doing a panel with a few other lovely folks. That obviously doesn’t need a whole lot of prep. I mean, we did some of the discussions beforehand, but I didn’t have to do the slides or anything like that so that was fairly straightforward, and then after that it was a couple here, a couple there and really trying to build up the confidence, the pacing and getting to a point where you’re comfortable on stage where I can actually have a dialogue with myself while I’m presenting.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** That is huge. When at the beginning, I couldn’t do that.

**Lea Alcantara:** [Agrees]

**Jonathan Snook:** I was so focused on “I have stuff on my slides, I need to get it out.”

**Lea Alcantara:** [Agrees]



**Jonathan Snook:** And now, I'm at that point where I know what my slides are. I know what I want to say and I can, while I'm presenting, look at the time. I can see the audience and I can gauge their reaction and see if I need to change the tempo, change how I'm talking to the crowd and having that ability to see myself and how that self awareness, I think has made a better speaker.

**Emily Lewis:** That's awesome. So I want to change tacks a little bit. I mentioned you work at Shopify. You're a product manager, right?

**Jonathan Snook:** Yes.

**Emily Lewis:** And I remember earlier this year, you blogged about how with this position, at least at that time, you were doing a lot less hands-on development than you had in the past, and perhaps than you'd anticipated. So it's been about nine months since you posted it. How has the transition really been? Are you doing more hands-on development now, or is it still how things started? What kind of skills have you learned? What sort of changed in your career thanks to this job?

**Jonathan Snook:** Before the product management role, I was a design lead. And as a design lead, it was like okay, well, I need to talk to customers and see that our designs; we're building the right thing. And when the job role of product manager popped up, they pretty described it out in that way, "You're going to talk to support. You're going to talk to customers. You're going to get a lot of research to understand what we should be building, and working with the design team, working with the development team." And I'm like, "Okay, well, this is really what I'm doing now, so why not? I'll take this role."

The interesting thing that came out of that was recognizing that I knew next to nothing about product management, so there was a ton of reading, a ton of research, a lot of learning to understand what



actually goes into product management, and yeah, it did mean that I wasn't going to be coding as much, and I've gotten to the point now where I'm actually okay with that.

I can still code on the side. I sometimes put together random projects on the side to keep my skills up. I have been doing a lot of work with SMACSS and still working heavily with the dev and design team to make sure that the code we're writing is still as topnotch as it should be. And I think as a result of being involved at that level, like I haven't removed myself completely. I might not be writing things day to day, but at least I'm still involved with the code, even if I'm not the one writing it.

**Lea Alcantara:** [Agrees]

**Emily Lewis:** [Agrees] Like it's *product* management, but do you also, through that, manage other team members?

**Jonathan Snook:** No, the interesting thing with product management is that you don't actually have a team *per se*. I don't have anybody that reports to me. What I have is a product that I own and that I manage, and I say, "Okay, well, these are the things that we should be building." That I set those priorities out based on all the research that I've done to say, "Okay, these are the most difficult problems or the easiest problems." Whatever it is with the priorities that we set out, these are the things we're going to work on, and then I talk to the design and development team and get them to maybe feel the same sort of urgency. I'm like, "Okay, yeah, that's the kind of stuff that we want to develop with."

So it's a team in that sense, and then we are all working towards the same goal, that we all have priorities that we're working towards, and for the most part, I do have sort of the final say to say, "Yeah, this is the direction we should be going in." It is different in the sense that I don't have people directly reporting to me.



**Lea Alcantara:** With that in mind, does your technical team use the SMACSS' philosophy?

**Jonathan Snook:** Well, when I came on board, a lot of the code for the products had already been written, and it was written in a way that would not necessarily follow inline with the stuff that I've written in the book, and it's been a slow progression. It wasn't a case where we're just going to strip it all the way down and really build there from scratch just to follow a methodology. It was a case of "Okay, you know what, let's do this piece by piece," and so it's not a 100% there, but it is continuing. Even after two years of starting Shopify, we're still evolving that, so there are pieces that we are slowly moving over as we refactor and as we build new components. We try and make sure that those components at least are done so in a way that is more in line with the book.

**Emily Lewis:** Now, before we get into the details of SMACSS in the specific architecture you recommend. I'm curious why you chose to self-publish since you have written two books for established publishers.

**Jonathan Snook:** Yeah, so one of the things that I've never really enjoyed about writing for a publisher is deadlines. [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** They need you to hit certain points and they also need you to have a certain count. They have this expectation that what they're going to have is going to sit on a shelf somewhere, and so they needed to be of a certain size because that needs to be noticed.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** Hey, you can't just have this little pamphlet sitting on the shelf because nobody is going to see it, nobody is going to buy it, and so from that, it was like, "Well, I'm going to write this and I'm going to get as far as I can before I really figure out what I'm going to do with that." Actually, I got



to a point where I was frustrated and I was just like I threw everything out that I had for free on the website. I hadn't put it into an ebook format or anything like. I was just like, "Here it is." And then from there, I started getting a lot more feedback from people saying, "Well, what about this? What about that?" And it allowed me to continue building. I basically used the audience and the readership to be my editors and understand, "Okay, what do I need to evolve here, and once it got to a certain point, it was like, "You know what, I think this is actually ready to be bundled up and sold as an ebook." And thankfully, the world was very receptive of that, and the rest, as they say, is history.

**Emily Lewis:** And generally speaking, aside from deadlines and page counts, have you preferred this self-publishing process more than working with the publisher, or is it not possible to really equate the two?

**Timestamp:** 00:19:57

**Jonathan Snook:** I have thoroughly enjoyed self-publishing.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** It's been, one, just a fantastic learning experience. It was an opportunity to really understand the process better. I had written the book in HTML since, as you know, I was writing it for a website, and having that is a lot different than working in Microsoft Word that a lot of my publishers needed it beforehand.

**Emily Lewis:** Yeah.

**Lea Alcantara:** [Agrees]

**Jonathan Snook:** And HTML is what I felt comfortable in, and so once I had this, then it was a case of, "Well, if I'm going to do this as an ebook, what kind of formats do I want?" A lot of people that self-publish would do PDF only. Having an iPad, I was like, "You know what, no, I want EPUB as well."



So I've got PDF in EPUB. But I got friends with Kindle, and I'm like, "Well, do I want to publish this on Amazon like can people download it through the Kindle program? Well, why don't I try that as well?" I'm really trying to understand all the different file formats, how do I create them.

So I realized in researching things that the EPUB format, for example, is just HTML. Well, I've already got the HTML, how can I convert that HTML into something that can actually work in an EPUB format, and then I learned that the MOBI file is just an HTML format as well. So then the only thing I was left with was the PDF, and how do I go from HTML to PDF, and that was the whole journey to get there. But it was interesting to say that like, "Okay, the HTML that I push to my website is actually the same HTML that goes into the EPUB and MOBI versions," and I think that's pretty cool.

**Emily Lewis:** So now, let's talk about SMACSS itself. I'm curious just generally why you chose CSS architecture as a topic.

**Jonathan Snook:** One, because nobody had really been talking about it. Nobody had written a book on the subject. Everybody was always writing about sort of "Here's how to write CSS. This is what a property is. This is what a value is. This is what a rule is. Here's a selector. These are all the different types of selectors, and then here are the types of layouts that you can create with that CSS."

**Lea Alcantara:** [Agrees]

**Jonathan Snook:** And with the stuff that I was doing at Yahoo!, it really meant rethinking how I wrote code for a large scale project, and once I had that concept and really it took a lot from the industry as well with people like Nicole Sullivan, Jina Bolton and Jeremy Keith. A lot of people were already writing about architecture, but in really small ways and not really on a larger scale, which is something that I wanted to do. I wanted to have something that was a little bit more holistic and that was very





generalized since that's the kind of person I am as well. I'm not the kind of person that will take libraries off the shelf *per se* and integrate them.

**Lea Alcantara:** [Agrees]

**Jonathan Snook:** I wasn't using a CSS reset. I wasn't using Bootstrap. Well, at that time, Bootstrap wasn't out yet, but even to this day, Bootstrap wouldn't be something I would necessarily pull in. I like hand-coding my projects, and so I wanted something that reflected that, that still have that flexibility for people to take the libraries that they might use and pull it under an umbrella of how they should be writing code.

**Emily Lewis:** So how would you describe scalable, modular architecture for CSS?

**Jonathan Snook:** I would describe it as scalable and modular.

**Emily Lewis:** [Laughs]

**Lea Alcantara:** [Laughs]

**Emily Lewis:** Well, I mean, that you mentioned you started looking at the work you were doing at Yahoo! as something that was, I guess, easier to maintain and develop on a large site. Is it literally just looking at content elements on your site and making modules from them or what's the fundamental goal here. Are they reusable modules?

**Jonathan Snook:** Yeah, the need at Yahoo! was to be able to chunk things in a way that we could only pull them in when we needed them, and as a result of that, we evolved a process such that, "Okay, what does it take to load in your inbox?" So I was working on Yahoo! Mail. What is the inbox and what does it entail? Well, you've got the header. You've got buttons. You've got the sidebar, and then you've got the customized list.



But then you go to Compose, what do you need to load for the Compose? Well, you've got the Compose screen and so on and so forth, and Yahoo! Mail wasn't the only product that we are working on. We're also working on [Yahoo!] Messenger. We're working on [Yahoo!] Calendar, Address Books or the whole communication suite, and we needed to be able to share components easily between them.

I even had this sort of grand, long-term vision where the components we were building could actually be used all through Yahoo! That never happened, at least, not while I was there, but that was sort of the long-term vision that I was working towards. But we have this library of code that we could reuse on a bunch of different places because it meant that we could solve problems once, and reuse these components on a lot of places, so I saw a lot of benefits from that.

**Emily Lewis:** So when you have this library of code, was this stored somewhere online where everyone could reference it?

**Jonathan Snook:** Yeah, we had our central repository, and we had built it around prototyping engine. We had extracted all the HTML and CSS away from everything else so that we could then deploy our code into each individual project. So the Calendar team would pull from our repository, the Mail team would pull from our repository, and it allowed us to develop outside of their build process so we didn't have to install a whole bunch of systems.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** We could actually build stuff out using just HTML and CSS and view it in the browser and we would have a list of components and we could share those components. We had an internal web server that we could send, "Here is this particular page or even this particular



component.” As a result of this templating that we had done, this modularity that we had created, we could then take this modal dialogue and show people what this modal dialogue would look like completely separated from everything else.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** And in order to do that meant you were separating the styles for that modal dialogue from everything else so that you knew that there wasn't these dependencies, where everything had to be loaded on the page in order to test that everything worked. Because that's the way we used to do it. We would have a page, we would build out the entire page, and then we view it. Does it look like everything is working? Yes, good. Now, we'll move on to the next page.

Well, when you have like a dozen or two dozen or three dozen or 50 or 60 or 70 or 80 pages, like you get to a size where that is just untenable. You can't make a CSS change and then go through 80 pages to see that everything looks right.

**Lea Alcantara:** [Agrees]

**Emily Lewis:** [Agrees]

**Jonathan Snook:** You need to make sure that what you're building will work in a small way and you'll know that when you take that small thing and put it into a larger system, that it will just work.

**Emily Lewis:** So it sounds like it's not the styles, it's the actual, I guess, naming conventions and how, I guess, specific your selectors are?

**Jonathan Snook:** It is. It's recognizing that the code you're writing may have impacted a lot more elements than you care to, that the way we tended to style things was: here's all this HTML on my page, let me take this string of <div>, <div>, <div> where you've got your outer container, your content, the inner container, and you have all these things and you're saying, “I want to style that one



particular element based on this huge structure.” And I took it from another angle and said, “Well, how can I break this down so that the only thing that I’m doing is really applying one class to one element and that I’m trying to minimize the impact that that has on anything below it and anything above it.”

**Emily Lewis:** [Agrees]

**Lea Alcantara:** [Agrees]

**Emily Lewis:** And your book is entitled or is focused on CSS, but how much is HTML involved, or is it truly just going to be based on class names? Is it dependent upon the DOM structure?

**Jonathan Snook:** It is trying to minimize how much dependency there is on the DOM. I talked a little bit about decoupling HTML from CSS, which sounds weird because CSS is designed to style HTML, but in trying to minimize the impact that that has, it gives you a lot more flexibility. Because if you can just take the class, a single class, and apply it to something, then there’s a lot of mobility with that, to be able to take that component and move it somewhere else because you don’t have to worry that that HTML element, whether it’s a <div> or <section> or <article> or <li> (list item), whatever it is, that the moment you move it somewhere else, it’s suddenly going to break because it had this dependency where, well, on the sidebar because I had the sidebar selector and the content area selector, it just totally breaks everything. So I’m trying to fix that by throwing a bunch of more selectors at it. That’s the kind of thing I want to avoid.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** So how can I make these little components as self contained as possible?



**Lea Alcantara:** So when you are talking about this type of philosophy to me, right now we've been talking a lot about products, Yahoo! Mail or Shopify or those types of things. Is SMACSS applicable to non-product websites like, let's say, a regular brochureware site even or a large information site?

**Jonathan Snook:** Yeah, absolutely. It was an interesting thing to go and look at, the process that I had put out and thought, "Okay, well, how can this actually get applied to something different?" I'm realizing that my own website could actually be redone, my blog, which admittedly hasn't been touched in like five years and the code is reflective of that. I could have rewritten it much simpler because we do have a lot of shared components even in content-driven sites. That we have these shared UI paradigms, and we have seen that in a lot of other examples.

The team at Paravel, for example, worked on the Microsoft website. Dave Rupert has a blog post that's called *Responsive Deliverables*, and it's a really good blog post. He shows how they were building a content-based website for Microsoft and how they had taken a number of these components for what is essentially brochureware. So I think that the same concepts apply very well in those types of situations.

**Emily Lewis:** Now, when it comes to the naming conventions that you follow for these classes, is it something similar to the Block Element Modifier methodology, or do you have your own approach that you recommend?

**Timestamp:** 00:30:07

**Jonathan Snook:** It is. There's definitely a lot of similarities to the Block Element Modifier or BEM, however you want to call it. The idea that you have these modules, these components. I break things down a little bit differently in the sense that I'm trying to layer things in, so I talked about base styles, which is something that I don't think BEM specifically covers. And then I talked about layout where



you have these containers that you want to throw stuff in. And then we have the actual content, and those are the modules that end up getting placed within the layout.

So understanding that delineation and then there's a couple more where if you have anything theme-based which doesn't necessarily mean, "Hey, I have that color on my site, I need to create theme style." It just means that if you have anything specifically related to user customization where they can change parts of the UI, those are the types of things that I kind of categorize as theming.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** And then the last category is states, so that you have layout and modules that might be reflected in different states, like you have active states or disabled states and those types of things that need to be reflected in the UI.

**Emily Lewis:** Now, you did mention in response to Lea that this could easily be applied equally to a product versus a content-driven site, but is it also something that could benefit a really small site or even a small team?

**Jonathan Snook:** Well, obviously, the methodology came from a large product. It came from a large team, and I worked on a team of 5 prototypers, we worked with 30 designers, and we worked with 200 engineers. That's a large team.

At Shopify, when I started there, the team was definitely a lot smaller. There was myself and three other designers, and we had a team of five developers. So, definitely a much smaller team. But as I was saying with my website, I could just have applied the same technique where it's just me working on a blog where I've got two templates, the techniques would apply very much the same.

**Emily Lewis:** [Agrees]



**Jonathan Snook:** It's interesting. I talk a little bit about how as both as a freelancer and somebody who just kind of works on his websites from time to time, a lot of the stuff that I talk about in the book talks about maintenance. It's the idea that things change, and we're trying to minimize the impact of that change across time, right?

**Emily Lewis:** [Agrees]

**Jonathan Snook:** So over time a large project is going to change a lot and it's going to have a lot of people getting their grubby hands on it. So that's something that you want to use the techniques as a way of minimizing the impact that all those changes have. Now, a small site where you might be the only person working on it, isn't going to be changing the site that often, but that doesn't mean that the techniques can't be just as applicable.

**Emily Lewis:** [Agrees]

**Jonathan Snook:** There are just making your selectors small is one step of just making it simple. You're just going to use less code obviously by having simpler selectors, or you're trying to impact only the things that you care about.

**Emily Lewis:** So since you've sort of mentioned that you could start with having simpler selectors, is a modular approaching that you can just sort of ease into, or is it only effective when everything is modular?

**Jonathan Snook:** I think that you can definitely take pieces and do it bit by bit, and that's something that we've had to do at Shopify to evolve our platform in that way where, "Okay, we're going to take one thing and then we're going to factor that out." So it doesn't necessarily mean that the entire site needs to follow the methodology for it to work. You can definitely start small and work your way up.



**Lea Alcantara:** Has there been improvements in those little pieces that you've started using this approach at Shopify?

**Jonathan Snook:** Yeah, absolutely. We've gone through a number of refactorings for each of our components, and we've realized that we had a lot of duplicate code, or we had design elements that had a lot of similarities, but weren't exactly the same, and it gave us an opportunity to refactor those, so that they were the same. And as a result of that, it means we ended up using less code.

We're actually going through a refactor of a component right now. A coworker has been working on it, and she was able to take three components that were very similar and actually tied them all together so they were exactly the same, and as a result of that, we're using a ton less code, which is fantastic. Less to maintain, less to send over the pipe, it makes for a much easier implementation.

**Emily Lewis:** You mentioned Nicole Sullivan a little bit earlier. I'm just curious, is SMACSS the same as Object Oriented CSS, or is it just kind of abstracted from it?

**Jonathan Snook:** There are some differences, but also a lot of similarities. It is a modular approach. We want to try to make sure that class names are simpler. With Object Oriented CSS, I think there's a little bit more granularity to it, and a little bit more freedom to mix and match the classes that you need in order to build an object. Another is you're taking a bunch of little Lego pieces from different buckets and pulling them together to build this wonderful thing.

**Lea Alcantara:** I'm curious about all this build process, and there's working with teams, and you've got all these tiny components. I'm wondering about the documentation of every specific module. Do you have a style guide that you follow, or a style guide site for Shopify designers and developers to refer back to?





**Jonathan Snook:** Yeah, at Shopify, we're only now really getting to the point where we've got a style guide that everybody can refer to. Yahoo! is a little bit better in that we had the prototype engine which really acted as our style guide, that you could build through template by template, and see what each of those components would look like, so it's a little bit easier to take the things that you needed to build the stuff that you want to see. You can say like, "Here's my modal dialogue. Well, I've got my modal framework and then what are my headings going to look like? What are my buttons going to look like?"

There is a lot of sort of premade buttons like OK and Cancel. Well, we would make a component out of that, and I can just say, "Well, does my modal dialogue need OK and Cancel, two buttons. They're going to be on the bottom right. Yes, I'm going to take those components, and I'm just going to take that template and drop it in." So it's a lot easier to mix and match the stuff that we needed. We haven't quite gotten there yet at Shopify. I think there's a little bit more of our infrastructure that we need to revise so that we can pull out the components that we need, and to where we kind of automate our style guide a little better.

**Emily Lewis:** Now, I'm curious. Well, actually, I know the answer to this question, but for the benefit of our listeners, is there any relationship between this modular approach to CSS and CSS preprocessing?

**Jonathan Snook:** No. [Laughs]

**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]



**Jonathan Snook:** When I had written this at Yahoo!, we didn't use a preprocessor. We had our own build process that allowed us to sort of compile and cut and nail all the stuff that we needed. The ability to keep everything separate and just because of our infrastructure at that time having something like Sass or LESS integrated wasn't something that we could easily fit in well with the fact that we were working with a variety of teams. And so as a result, it was very much everything that I wrote about does not have any dependency on a CSS preprocessor, whereas at Shopify, we use Sass. We love Sass, and we have been able to apply that very easily to the modular approach as well. So each of the modules are separated out into their own files so they're easy to find, easy to reference.

**Emily Lewis:** Oh nice.

**Jonathan Snook:** And then we just used Sass `@import` to compile everything into one style sheet.

**Emily Lewis:** Yeah, that makes sense. It seems like it's a perfect fit. It's nice that it's not dependent, but it's a nice fit to maintaining those files.

**Jonathan Snook:** Yeah, absolutely.

**Emily Lewis:** Now, you mentioned you guys write Sass at Shopify?

**Jonathan Snook:** We do.

**Emily Lewis:** Is that your preferred or do you ever worked with LESS?

**Jonathan Snook:** Yeah, I mean, I like Sass. I know Chris Eppstein so it's kind of handy that way too.

**Emily Lewis:** [Laughs]



**Jonathan Snook:** But I like the features and I haven't really needed to branch out from that, and because they're liking them at Shopify, that's what they are using.

**Emily Lewis:** Nice.

**Jonathan Snook:** But I was cool at sticking with it.

**Emily Lewis:** Now, I'm just curious, obviously, the time that takes to write a book probably makes it a different way to measure the investment of time and learning a new approach to writing CSS, but on average, what would you say is like the learning curve for going in a more modular direction, maybe even just based on what you've seen with some of your colleagues at Shopify?

**Jonathan Snook:** I don't think the hurdle is necessarily that steep. I think that it requires a bit of a shift in thinking that when you're looking at a design, that you're not thinking too much about the context and thinking more about here is a design pattern, "Here's something that I'm reusing in a bunch of different places." Just being able to recognize that, and I think that if somebody can recognize that upfront, they're going to have a much easier time implementing it versus someone who just can't really separate the two. Somebody can look at a design and go, "Okay, well, I am understanding the design system." And that is what it is.

Really, it's about establishing the design system, understanding the hierarchy, how things should be working in, what are all the different components in the UI and when do you use one UI component over another. Once you've established that, that modularity comes really easily.

**Emily Lewis:** [Agrees]

**Lea Alcantara:** I mean, it sounds like this type of approach is great for all types of projects, but do you believe there's any type of project or site where modular approach doesn't really make any sense?



**Timestamp:** 00:39:54

**Jonathan Snook:** Yeah, I think there are going to be certain site designs, specifically those marketing sites where there's a lot of unique elements. So for those unique elements, well, that modularity is not really going to be of a whole lot of benefit, because you're not going to be able to reuse those components. This is like, "Here's a one-off." The layout on one page might be four columns. On another page, it might be three columns. You might have components that need to sit outside of the regular flow because you're trying to create this eye-catching layout where things don't have a rhythm.

When you have a lot of different components and there's not a lot of similarity between them, in that kind of situations, SMACSS just doesn't really seem all that applicable.

**Emily Lewis:** Although I would think that if your naming conventions are good, that even if you don't have shared modules, the quality naming convention with less specificity is probably going to be easier for you to pick up and look at later on down the road.

**Jonathan Snook:** Yes, absolutely. Actually, it's one thing that over time I have recognized. It wasn't something I had set out from the beginning. I mean, when I first wrote the book, I felt like the categorization was really the most important part. You've got your base styles, your layout, your modules, your themeing, all that stuff is really what it's about. And over time I've gotten to appreciate that naming convention is really the tough part. That is the part that is going to keep you up at night, "How do I name things?"

**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]



**Jonathan Snook:** Because what we learned from the get-go, for anybody hopefully learning that a class name like "red-background" probably isn't the best naming convention that we should use because what if it could change and it's no longer red? Well, obviously, that class needs to get thrown out.

But at the other end of the scale, you don't want something so completely generic that it doesn't make any sense what it is when you're using a bunch of different contexts.

And I found myself in that kind of situation where we had these boxes, and like CSS has a box. Well, we had a bunch of boxes, and they had a specific style where they get like rounded corners and whatnot, but then in another part of the website, we had these different boxes. They had a completely different look and feel and they interact differently with each other, and using the "box" class for us didn't actually make sense. We needed to come up with a different name for these boxes.

The content, really we didn't know what the content was. It wasn't like we can say they're movies or they're emails or whatever it is. We have a bunch of boxes and the content is going to be different there, what we're going to throw in them, and so we have to come up with a different name, and the name we came up with was "pod."

**Emily Lewis:** Oh.

**Jonathan Snook:** Like pods and boxes.

**Lea Alcantara:** [Agrees]

**Jonathan Snook:** We had to come up with another name as opposed to like "box-2" or "box-the-sequel".

**Emily Lewis:** [Laughs]



**Lea Alcantara:** [Laughs] box the sequel ...

**Emily Lewis:** Part deux ...

**Lea Alcantara:** [Laughs]

**Emily Lewis:** [Laughs]

**Jonathan Snook:** Yeah.

**Lea Alcantara:** That'd be Canadian friendly.

**Emily Lewis:** [Laughs]

**Jonathan Snook:** Yes. [Laughs]

**Emily Lewis:** Before we wrap up today, Jonathan, what's on the horizon for you and for SMACSS?

**Jonathan Snook:** With SMACSS, it is at this point, just kind of going off on its own. I've been really lucky in that. It's already been translated into French and Japanese, which is pretty cool.

**Lea Alcantara:** Cool.

**Jonathan Snook:** I had a couple of people that were kind of working in spare time. I mean, it was a spare time project for me, so that might at some point get available on a couple of other languages, which will be awesome. But at this point, I think about maybe updating it just to maybe touch a little bit more on some topics like responsive web design.

A lot of people have had questions about how does it apply in those types of situations, and I haven't really gone into a lot of depth in that regards. I do a little bit more in the workshops, but I haven't done that. So I might get around to actually writing it. Me personally, I'm having a lot of fun at Shopify, so a lot of my time these days is really focused on that.



**Emily Lewis:** Are you going to be doing more SMACSS workshops in 2014?

**Jonathan Snook:** I am. Right now I've only got one scheduled. Hopefully, maybe one or two more. I'm not doing as many as I had. I mean, I've managed to do 13 in the last two years.

**Emily Lewis:** Holy...

**Lea Alcantara:** Wow!

**Jonathan Snook:** Yeah, it's been a busy couple of years.

**Emily Lewis:** That's on top of your other speaking engagements, right?

**Jonathan Snook:** I know. I've done like 20 events on top of that over the last few years.

**Lea Alcantara:** Holy moly.

**Emily Lewis:** [Laughs]

**Jonathan Snook:** It's been kind of busy. So I'm not doing as much. I'm definitely slowing down, so I have only one workshop planned. I've got another that I might have booked soon, and that's pretty much it. Maybe I'll get one or two more in. So it's definitely got to be a much quieter year.

**Lea Alcantara:** That sounds like fun though.

**Jonathan Snook:** It is. It's nice to ... yeah, it's nice. I'm going to enjoy the pace.

**Lea Alcantara:** Yeah. Well, thank you, Jonathan. That's all the time we have for today. Thanks for joining us!

**Jonathan Snook:** Thanks for having me! It's been a lot of fun.

**Emily Lewis:** Oh, same for us. In case our listeners want to follow up with you, where can they find you online?



<http://ctrlclickcast.com/episodes/modular-css-with-jonathan-snook>

---

**Jonathan Snook:** Of course, they can reach me at [snook.ca](http://snook.ca). They can also find me on Twitter @snookca. Nice and easy.

**Emily Lewis:** [Laughs]

**Jonathan Snook:** And yeah, I'm on LinkedIn and I'm on all those other sites. I'm around.

**Lea Alcantara:** [Music starts] Sounds good! Now, we'd like to thank our sponsors for this podcast, [Perch](#) 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](http://ctrlclickcast.com).

**Emily Lewis:** Don't forget to tune in to our next episode when Jack McDade is joining the show to talk about his Statamic content management system. Be sure to check out our schedule on our site, [ctrlclickcast.com/schedule](http://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:** 00:45:41