

<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

CTRL+CLICK #36 Twig Templating for Craft with Marion Newlevant

[Music]

Lea Alcantara: You are listening to CTRL+CLICK CAST. We inspect the web for you! Today we are diving into the topic of Twig templating for Craft with our special guest, Marion Newlevant. I'm your host, Lea Alcantara, and I'm joined by my fab co-host:

Emily Lewis: Emily Lewis!

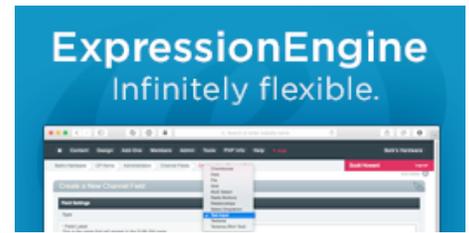
Lea Alcantara: This episode is sponsored by [EllisLab](http://ellislab.com). ExpressionEngine is an infinitely flexible content management system that powers some of the web's best. ExpressionEngine 2.9.2 comes jam-packed with even more power and flexibility. Smarter, faster conditionals, reordering template routes, Markdown Extra which support for tables, fenced code blocks, footnotes and more and a massive array of security and stability fixes, to just name a few. ExpressionEngine can power your next project no matter how big or small. Visit ellislab.com/expressionengine to learn more download a free copy of ExpressionEngine core today.

Emily Lewis: Today's guest, Marion Newlevant is no stranger to CMSes. She's as versed in the back end as she is to the front, with expertise in PHP and ExpressionEngine and CodeIgniter, and if you take a peek at her recent Twitter stream, you'd know she's a big fan of Craft and very active in that community. Welcome to the show, Marion.

Marion Newlevant: Thank you.

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

Marion Newlevant: Well, I don't really recognize myself from all that expertise you just gave me, but...



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: [Laughs] I just got that from your website. I was so impressed when I was checking out your portfolio. [Laughs]

Marion Newlevant: There you go, it's a good marketing website, and I need to update it as well. Let's see, I'm here in Portland, Oregon. It's a gray-out, which is we say as lovely weather.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Marion Newlevant: I am, in fact, a software developer since kind of the dawn of time. I go back to the age of punch cards, and I like to dabble in this, that and the other thing, so this, that and the other thing technical-wise, and then I'm a creative crafty person. Let's see, if you would ask me what my hobbies are, I would start talking about how I've been to ceramics and throwing pots and I like to spin, making yarn and knit some and saw some.

Emily Lewis: Oh, cool.

Marion Newlevant: I don't know. I live in an old house in Portland. I have a lot of cane-seat chairs and I re-cane their seats every so often.

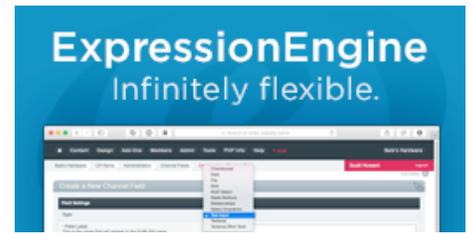
Lea Alcantara: Oh, interesting.

Marion Newlevant: So lots of minority crafts.

Emily Lewis: And so you have a lot of interest in technical stuff, but you also seemed to do a lot of stuff with your hands too.

Marion Newlevant: Yeah, they relate for me. It's kind of creating things.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Lea Alcantara: [Agrees]

Emily Lewis: So you've had some experience with ExpressionEngine. I'm curious what made you look into Craft?

Marion Newlevant: I was hearing good things about it, and it seemed like it was a new thing to jump on and I could get in on the beginning. I sort of came to ExpressionEngine actually fairly late in the game, so I thought, "Oh, I'll check out this new thing."

Emily Lewis: [Agrees]

Marion Newlevant: So because everybody was learning together, it was less of the learning curve for me.

Emily Lewis: Did you get in on the beta like the early versions or was it only after it kind of was out for public release?

Marion Newlevant: After it was out for public release.

Emily Lewis: [Agrees]

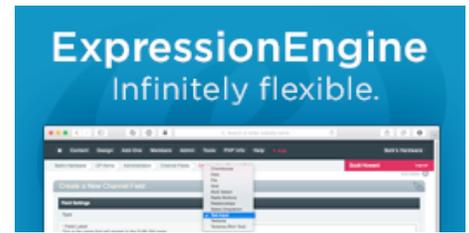
Marion Newlevant: I'd heard about it before then when it was called Blocks and it was sort of like list of things I wanted to check out, but I didn't actually get around to it until it was released.

Emily Lewis: And Twig is the underlying templating language for Craft. Had you already had any experience with Twig at that point?

Marion Newlevant: I've worked with Django.

Emily Lewis: Right.

Lea Alcantara: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: And Django has a very, very similar templating language. It compiles into Python and not PHP, but the underlying language is very similar, so that's my previous experience, not exactly with Twig but very close.

Emily Lewis: So in terms of that, did that have any effect on you being more open to Craft? Like how that underlying language for the templating for a given CMS, does that affect whether you pick it up or choose it to use for a particular project?

Marion Newlevant: It affects a lot how I feel about it, I think.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Marion Newlevant: Twig is a nice, clean and simple language, and I go, "Oh, this is lovely, and I could really get into this."

Emily Lewis: Yeah, I agree. I feel like it's really, you know. It sounds like you have more of a programming mind than maybe I do, but I feel like it's just really easy to read, you know?

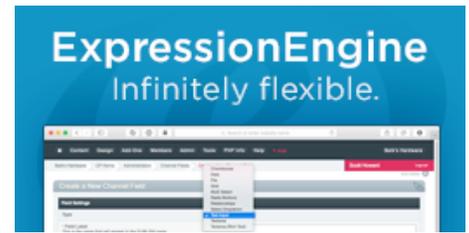
Lea Alcantara: Right.

Emily Lewis: You don't have to know a lot of logic or programming to read it right away and say, "Oh, that's what that does." I understand it immediately.

Marion Newlevant: It's pretty to read and it's consistent as a language.

Lea Alcantara: Yeah, because what I was about to ask is, what defines a clean, easy-to-understand language to you?

Marion Newlevant: I guess it's more easier for me to understand the converse, what is an unclear, difficult-to-understand language? But a clean one, there are a few pieces of syntax that Twig has, the



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

curly bracket percent sign sort of tags and the double curly bracket tags, and then it has the Twig comments, and that's it. There are those three things and anything else is not Twig, and then within the tags, there is a consistent way of making assignments and testing things and so on. So it's all fairly compact as a language. It doesn't have a lot of extra special cases or two kinds of conditionals or any of that.

Lea Alcantara: So when you started playing around with Craft, obviously, you were playing around with EE a little bit before. Did that affect the learning curve for you in terms of templating? Was it okay because like you said, you already were playing with a similar type of language within Django?

Marion Newlevant: Well, it's another programming language, and from my perspective, it's yet another programming language, and that's not the intimidating part.

Lea Alcantara: So what was the intimidating part?

Marion Newlevant: Of coming to Craft, I think Craft was really not very intimidating at all.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

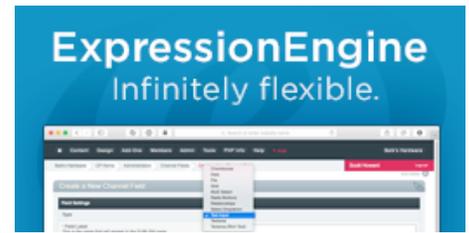
Marion Newlevant: That's what I like about it.

Emily Lewis: You know I was looking at a resource that Lea had shared with me in advance of this conversation, and it's Brandon Kelly's templating in EE versus Craft.

Marion Newlevant: [Agrees]

Emily Lewis: And who knows how long this has been out there, I'm always like the last person to find something like this. [Laughs]

Lea Alcantara: [Laughs]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: [Laughs]

Emily Lewis: But it's a really great document. I think I would love to see this if we could create this for every kind of CMS that we work and say, "Well, this is how you do comments in ExpressionEngine, in Craft, in Statamic, in whatever." It makes it contextual, so for me, having been familiar with ExpressionEngine, it gives me an instant, "Oh, this is how I did it in ExpressionEngine. This is how I did it in Craft."

Marion Newlevant: Right.

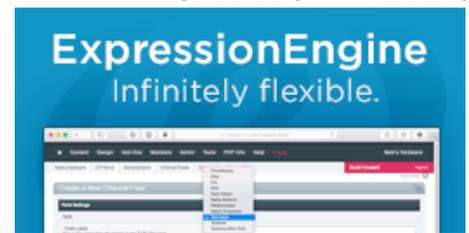
Emily Lewis: I would love to see this extended for other things. I mean, obviously not from Brandon, but for even our own internal references, I just think this is such a really simple, straight way to get someone quickly familiar with some of the concepts, drawing a parallel with something they're already familiar with.

Marion Newlevant: Absolutely.

Lea Alcantara: Yeah, for me, because I'm not a programmer by trade, so for me like the system that I'm very familiar with obviously is ExpressionEngine, so I started to understand how things work using that particular templating language. For me, when I jumped into Craft and into Twig, it was a bit more difficult, not necessarily because I thought Twig was a difficult language per se, it's just that there was some translation issues.

Emily Lewis: [Agrees]

Lea Alcantara: Even in terms of terminology like whatever an embed means or whatever the extends mean or whatever the import means, they don't necessarily mean the same thing in ExpressionEngine as it is in Craft or even pagination, like the concept of how pages go from left to right or next or previous is what it is, but that actual concept of the content is slightly different so in



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

order to do maybe the same thing in one versus the other is you have to figure out what the translation is first before you can dive into it.

Emily Lewis: I think that's a great word translation. I feel like this document sort of attempts to do that translation. It's like, "Here's the word in English, here's the word in Spanish, and here's how you do it in EE, and here's how you do it in Craft."

Lea Alcantara: Yeah. [Laughs]

Marion Newlevant: Right. I think that's really useful.

Emily Lewis: So Marion, you said that you didn't really feel like Craft was particularly intimidating to pick up, so I'm curious, what do you think was so comfortable for you about picking it up and using it? What was so easy about it for you?

Marion Newlevant: Again, I think I said it was sort of a small pond. It's possible to read through all of the Craft documentation that exist or was at that point.

Emily Lewis: Right.

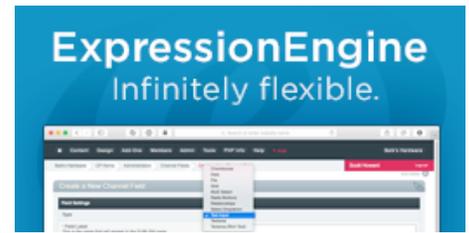
Marion Newlevant: And you say, "All right, there aren't any real hidden corners here."

Emily Lewis: That's a good point about a comparatively new community and/or tool.

Marion Newlevant: Right.

Emily Lewis: Everything is right there and it's not distributed across ages and ages of, you know. Sometimes I do find it a little frustrating when I go into forums and finding solutions, but they're for such old versions and I have no idea.

Lea Alcantara: Right.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: Because it's just such a deep wealth of information related to ExpressionEngine.

Marion Newlevant: Right. Yes, and so if you look at WordPress, for instance, which is kind of the other extreme, you know?

Emily Lewis: Right.

Lea Alcantara: [Agrees]

Marion Newlevant: You're going to spend an awful long time trying to get yourself oriented in the sea of information about just even what are the useful resources.

Emily Lewis: [Agrees]

Marion Newlevant: And you say Craft, well, you want to go to the Craft website.

Lea Alcantara: Yeah, right.

Emily Lewis: [Laughs]

Marion Newlevant: And you read the documentation there and that's where it is.

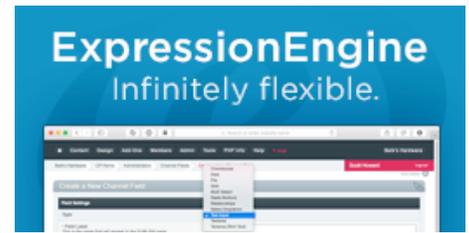
Lea Alcantara: Yeah.

Marion Newlevant: It's spread all over everywhere.

Lea Alcantara: Although like, I guess it is a double-edged sword too because I started playing around with Craft when it was still the Version 1, the public version after the beta.

Marion Newlevant: [Agrees]

Lea Alcantara: And I remember I had to ask Brandon about that part. I know I'm going on about the pagination thing.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: [Laughs]

Lea Alcantara: Because I was so confused about it, and the tutorial you see on the Craft documentation was after I asked Brandon to write that tutorial. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: Because I'm like, "Okay, I don't get what I'm doing."

Timestamp: 00:09:44

Marion Newlevant: Yeah, but he probably answered you right away.

Lea Alcantara: Yeah, exactly, but the fact is if you're coming from my perspective where I'm a front-end developer, I'm not a programmer and the logic doesn't make immediate sense to me, so early documentation wasn't as useful to me as it is now, now that people have been battle testing it for a little bit.

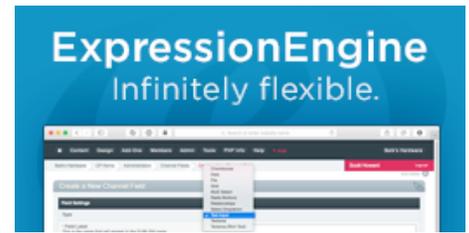
Marion Newlevant: Yeah, but at least you know that if you're not finding the documentation, it's not because you just don't know where to look, it's because it isn't there.

Emily Lewis: [Laughs] That's true too. It just needs to be created. You've mentioned Django and that that language is similar. How is working in Django and templating in Django for you? What is your experience? Is it pretty simple and straightforward? I have personally have never worked with it.

Marion Newlevant: It's very like Craft, you know?

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: It kind of depends on whether you set yourself up and to do things in a simple way or whether you set yourself up to do things in a way that has a lot of this one is like that one except different in this other way and let's encapsulate all of that with a bunch of inheritance and structure and so on. So you can kind of do it either way.

Emily Lewis: Well, let's talk about that just a second. So you say setting yourself up to create a simple template. What does that mean to you? Is that like we've talked about it before on the podcast like DRY (Don't Repeat Yourself) using like a modular approach to your templating, breaking things up and into...

Marion Newlevant: Well, if you'd focus on being DRY and being modular and breaking things up and not repeating yourself, you might come up with something that works really well, but it will not be the simple, most superficially easy-to-understand way.

Emily Lewis: Right.

Lea Alcantara: Right.

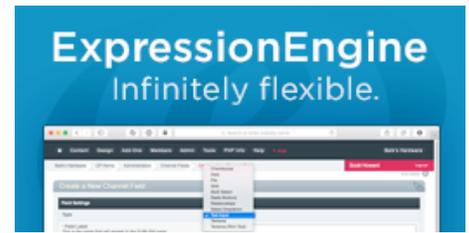
Emily Lewis: Is that the approach you follow though? I mean, so when you're looking to build something simple, what does that mean to you?

Marion Newlevant: I'd actually end up going down the road of making things very DRY. I just don't think that that's the same as making things superficially easy to understand.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Marion Newlevant: If you'd just write it all out and if it happens twice, it happens twice, that's very simple. It may not be what you want going down the road.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: Right, right.

Marion Newlevant: But yeah, so there's a trade off between how much you want to invest in understanding the structure you've created versus just keeping it simple.

Lea Alcantara: Regarding that, how do you find Craft handles making things DRY? Is it easier to make things DRY in Craft than it is, say, other systems you have played around with?

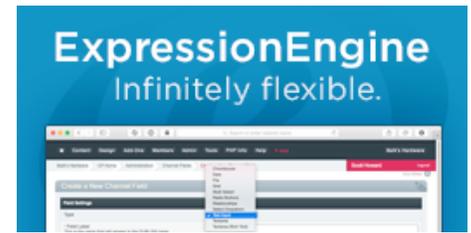
Marion Newlevant: I think it's got quite good support for it, and I like to... I don't know, I have a complex programming mind, so I tend to go down that path of "Let me set up all the structures so I'd never have to repeat myself." So I kind of enjoy structuring things that way. I use a lot of macros. I'll use a lot of includes.

Emily Lewis: Can you explain what you mean by macros? I'm not familiar with that.

Marion Newlevant: Oh, Twig macros. Twig macros are wonderful. Twig macro is like a little function call, except that what it returns is always a bit of string that you output rather than being able to return other sorts of structures. But you have a macro in Twig and you pass it some parameters and it does its thing and it outputs whatever it is that you want to output. So it's just like little function call, and I use them anywhere that I want to encapsulate some things.

So if I'm just reading down my template and I say, "Okay, and there's the header. Poof, let's call a macro that does the header. Let's call a macro that does this navigation." Then I can come along later and say, "Okay, now what does the navigation actually look like?" Well, I'll put more details in that macro and it's going to call these other macros, and so it ends up for me being a very programming approach to things.

Emily Lewis: I'm just looking at the SensioLabs Twig example of the macros. So correct me if I'm wrong here, but is it a kind of way to spit out your markup and the attributes for that markup, like a



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

programmatic way. I mean, because I'm looking at their example and it's got macro for an input form input, so it just generates the input element in a form or wherever you want it to go.

Marion Newlevant: In the Twig documentation there, that's a macro which is sort of like a function and the name of the function is input, so you say, "I want this input thing and I'm going to pass it to these things." Yeah, and it assembles it for you.

Emily Lewis: And you would use that for, like you said, a header in your template, so it can also include the Craft tags for the actual header value?

Marion Newlevant: It can include more or less anything. I use a macro whenever I have some piece of typically HTML that the same sort of thing is going to happen more than once.

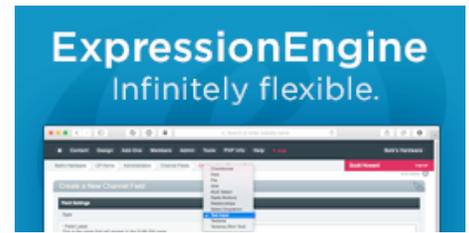
Emily Lewis: Oh.

Marion Newlevant: Like if I'm displaying a bit of Rich Text, and then it turns out that I want to always have that Rich Text sitting inside a div with a class so that I can format it as Rich Text, and so instead of just always remembering to do that every time, I call my Rich Text macro.

Emily Lewis: Oh, I see.

Lea Alcantara: I see.

Marion Newlevant: And I pass it my Rich Text and I pass it an array of class names that I want it to have and it generates it for me, so that's one example of a macro. I use macros often in place of the Craft nav tag. The Craft nav tag is really nice. It does this thing with the children and so on, and if you push it so far and then it starts to break down because you want to do something different on this level and you want it to be able to know this thing which is your active element and it gets complicated. So at a certain point, I say, "Okay, well, it's not working for me so I'm just going to make myself a macro and I'm going to pass it in my top level of the tree, and you can write as it turns out



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

recursive macros. So I just call myself, that macro calls itself when it gets to the next level of the structure, and you can have more control than you do with the nav tag. You can count how many levels deep you are and hard coat it in that it's not willing to go past four.

Lea Alcantara: Right.

Marion Newlevant: That it's going to do something different on Level 3 or those sorts of things.

Emily Lewis: Yes, so I could see how that's a way to reduce the amount of markup you're creating for those repeatable elements.

Marion Newlevant: [Agrees]

Emily Lewis: It's very interesting. What do you think about Craft's temple inheritance and how that can support a more DRY approach, not having to repeat, like something as simple as like your common header and footer.

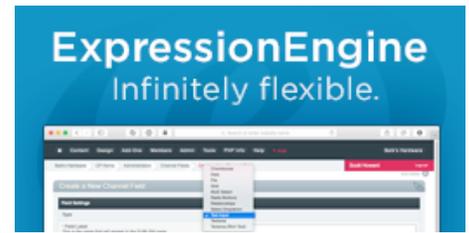
Marion Newlevant: Twig template inheritance?

Emily Lewis: Or is it, I guess, it's Twig's template inheritance. It's not specific to Craft, just part way.

Marion Newlevant: It's not specific to Craft.

Lea Alcantara: Aha.

Marion Newlevant: Yeah. So you could ask me about the includes and extends and embeds, which are kind of old ways of keeping things DRY for sure, and I'll talk about all of them. They're kind of related and also in my mind, they relate to macros because they're all about keeping things DRY. So include is the simplest of those three. They're all related, but include is the simplest, and it's just like right here as I've been along generating stuff, now I'm going to include the contents of this other



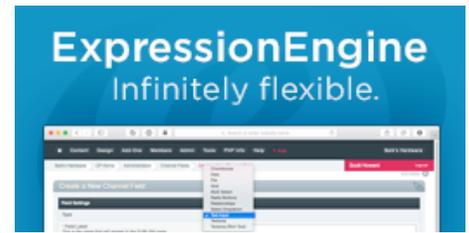
<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

template, and you can include the contents of this other template more than once and you can include the contents of this template here and that template there. It's not restricted in that way.

I think of include is building things out inward and extends is kind of going upward, which is a little bit different. So you have extends, you have your template and you say, "This template here extends that template over there." So the template that you're extending, typically it will be like a layout template, will have in it a number of one or more blocks, and then the template which is doing the extending, it's job is to stuff content into those blocks. So in the examples, they typically just have one block and it's called Content, but you could have a block that was called Content and you could have a block that was called Header and you could have a block that was called More JavaScript and whatever.

When you're extending, you get to extend one template and that's all you get to do really basically that the template that is doing the extending is allowed to create these blocks, but is not allowed to output anything else, and you can nest those. You can have this template extends that template extends the other template which is one way of you know. So you have your basic layout with your header and your footer and your stuff in the middle and then your stuff in the middle, sometimes it's one column and sometimes it's two columns, and then you could have both the one-column template and the two-column template both extending from the basic template so you can build up repeating structures that way. But you only get to extend one template per template file. Now, embeds is kind of you take both of those and you combine them together, and when you combine them together, you have a template and it can embed.

An embed is like an include plus you get to fill in blocks in the thing that you're embedding. So you can embed more than one. You can embed something in more than one place. You can embed more than one something, but you get to pass stuff down in this using this block structure into the



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

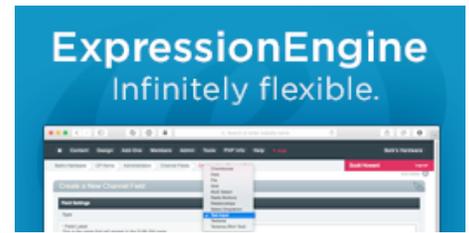
things you're embedding that you don't get to do with an includes. I think it's a very cool concept. I don't think I have yet come up with a project where I say, "Yes, with this I need to use embeds for." I'm more likely to use includes and I'm more likely to use macros personally so far, but on the other hand, I may come up with something where I just go, "Oh yes, now I know what this is really good for because I've come to that example."

Lea Alcantara: So I'm curious, because I haven't had a chance to use macros yet. I've been using the includes and extends, and I haven't even touched embeds because it confuse me. [Laughs] Because when I was reading all this stuff about embeds, I'm like, "What's the point? Like I can just use includes and extends and just have a bunch of templates of includes and then make my own pages, et cetera." But with the macros, do you have like one master macro page or do you have macros per template, like I'm just trying to figure out how it would work in terms of the workflow, like breaking up the files and stuff?

Timestamp: 00:19:58

Marion Newlevant: Right. What I do in terms of the workflow is I create folder in my templates folder that's called `_macros` and then I stick all of my macros in there, and I'll have related macros will go in one file, and then I have a sort of sometimes a catch-all sort of oodles for things that get used a lot. Then inside the individual macro files, so typically I'll have, for example, I have a macro file called `_navigation`, and then in there I will have a macro called `Output`, so that in my template when I'm calling that macro, first, you have to import the macro and then you call it, you say, "`Navigation.Output`," and pass it the parameters so that to me make sense to call it output.

So I'll have the macro called `Output` and then that `Output` macro will quite possibly call lots of other macros that are sort of internal to the macro file, and I name them with an `_underbar` just so that `_underbar This` and `_underbar That`, just so I know that those are local to that file in terms of how



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

they're used and you could call them from outside, but since I don't want to, I name them with an `_` underbar, another convention. Does that help?

Lea Alcantara: Yeah, I think so.

Emily Lewis: I think it's interesting. I think that obviously it's so hard to talk about code without...

Lea Alcantara: You seeing it, yeah.

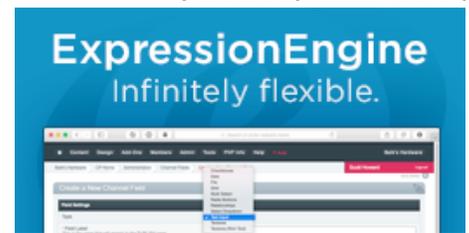
Emily Lewis: Yeah, exactly. So I think this would be one of those things that I bet Lea is going to want to try it the next time we build a Craft site. [Laughs]

Lea Alcantara: Yeah, I know for sure.

Marion Newlevant: Oh, you should totally try it, and one of the things that you can do which is really nice with Craft is you can have these templates that you just navigate though they don't have an entry associated with them, so you can just go play around and create a bunch of templates that includes other templates and extend and embed and have defined macros and just write yourself some little test code, "What happens if I do this, and what happens if I do that, and will the other work?"

Emily Lewis: So I have a question, and I hope it's not an irrelevant question to Craft in particular, but with the ExpressionEngine, there are issues with parse order sometimes and performance, when something gets parsed, whether it's early or later in the parse order. Is that any of a concern when you're dealing with Craft templating, especially like if you have macros inside macros and you're using includes, like is there anything that you've noticed that causes an error or causes a major performance hit, or are all of these operating, as far as you can tell, just like seamlessly?

Marion Newlevant: They operate pretty well. What happens is that the Twig templates get compiled into PHP, and so all of that parsing of Twig templates and then including and macro calling and so on, those are just turned into PHP function calls at the worst. You can go and look at the compiled



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

templates. They're in *craft/storage/runtime/compiled_templates*, and there they all are. If you'd clear your cache and then just look at one thing, then you'll have maybe less than to look at, but that's just PHP code. So calling a macro, I think just turns into a PHP function call and in the scheme of things, a function call is not something to worry about, but you were asking me about tricky thing I find that reminds me of the parse order issues and so on in Craft with Twig, and I think there just maybe no way around it is figuring out how to use the cache tag so that things don't break and so that you get benefit out of it.

Emily Lewis: Oh.

Lea Alcantara: [Agrees]

Marion Newlevant: So that's my challenging thing.

Emily Lewis: So that's something you're still trying to find the sweet spot for?

Marion Newlevant: That's something I'm still trying to find the sweet spot for.

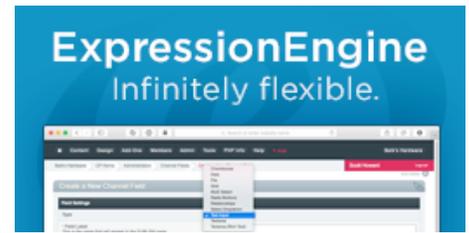
Lea Alcantara: Speaking of that caching, I feel like it's one of those steps where you're trying to be really efficient with writing Twig and writing your templates. What do you think are the biggest ways you can write Twig more efficiently?

Marion Newlevant: Well, it depends if you mean write Twig more efficiently or write more efficient Twig. If you mean write Twig more efficiently, I suspect there are some clever things that you can do with Sublime text and...

Emily Lewis: Oh right.

Lea Alcantara: Right, right, right.

Marion Newlevant: And I would recommend that, but I haven't actually gotten into it myself.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: Oh, you don't do that yourself yet?

Marion Newlevant: I use Sublime text and I use sort of the stuff that it comes out with out of the box, but I haven't spent the time to really figure out how to make it jump through hoops for me.

Emily Lewis: Yeah.

Marion Newlevant: But I hear good things about it.

Emily Lewis: Yeah, same here. I'm in the same boat. I've been wanting it to write my code for me, but I have to take the time to tell it how to. [Laughs]

Lea Alcantara: [Laughs]

Marion Newlevant: Yeah, yeah. [Laughs] Maybe you can get somebody else to come on and tell us how to do that, that would be great.

Emily Lewis: Well, actually, we had a podcast with Sean Smith last year where we talked about this a bit, and he didn't do it necessarily with Twig or Craft, but for some of his common ExpressionEngine code snippets, he created a text expander for Sublime text where he could just pop or just drop in and how he writes those different areas, how he does pagination, how he might do a form.

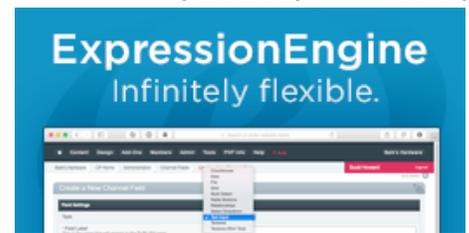
Marion Newlevant: Right, right, and closing those tags, I'd like to...

Emily Lewis: [Laughs]

Lea Alcantara: Right, right.

Marion Newlevant: You know, `{% if %}` and have the `{% endif %}` just there for me.

Lea Alcantara: Right.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: So I have not done that, but I suspect there are advantages there, and the other thing as I'd try and write for writing more efficiently, I'd try and keep stuff well organized and tidy so I don't spend so much time floundering.

Emily Lewis: Yeah, and it also sounds when you were describing your macro organization, that you have a convention that you follow for how you name things and save directories, and I think for whatever you're writing, whether it's front-end work or you're doing stuff for Craft or Statamic, having a convention is really important, whether it's something you've established on your own or you're borrowing from someone else.

Marion Newlevant: Yes.

Emily Lewis: So what about writing more efficient Twig?

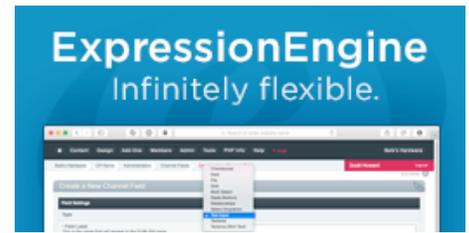
Marion Newlevant: Writing more efficient Twig, yeah, well, it gets compiled into PHP, which is really helpful. So you really need to worry mostly about the efficiency of the PHP and not so much the efficiency of the Twig parsing.

Lea Alcantara: [Agrees]

Marion Newlevant: So there, you want to watch out for inefficient algorithms, if you're looping through things and looping and looping in n^2 algorithms and so on. That's going to be bad, and sometimes just because you can do something in Twig doesn't mean you should do something in Twig.

Emily Lewis: Right.

Lea Alcantara: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: Once you get into complicated Twig programming, you might as well be writing plugins. It's really no harder and then that gives you the power of PHP and something that was being really inefficient in Twig might just be a snap in PHP.

Emily Lewis: Oh, that's a good point. That actually wouldn't have occurred to me coming from my perspective in developing templates, that yeah, I could see myself struggling trying to get some Twig to something when, yeah, I could just ultimately decide to do it in PHP.

Marion Newlevant: Yeah.

Lea Alcantara: Although, I do believe there is a bit of a parallel in ExpressionEngine world because I know one of the tips for making sure your templates are speedy is don't render your template as PHP because some people might shove in like a PHP code in a template and the first line of defense is can you make a plugin in PHP before you shove this PHP code inside this template to make whatever you want to work, and more often than not, the plugin is the better solution.

Marion Newlevant: Well, you can't just shove PHP into your Twig templates.

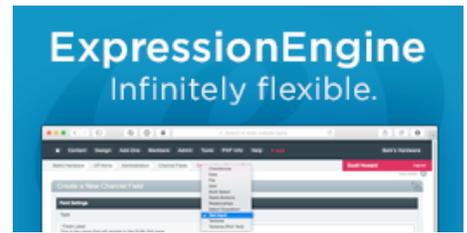
Lea Alcantara: Yes.

Marion Newlevant: You don't...

Lea Alcantara: Okay, that's a positive.

Marion Newlevant: You're not able to do that, but you certainly can decide to search for something by writing loops within loops within loops and that might be much better done in PHP. You have a lot more power there.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Lea Alcantara: So I want to dive in to something very specific about Twig. Just like with any sort of programming language, there's always more than one way to do the same thing, and I think a very common one that people who create Twig templates and Craft templates will find is the element criteria model.

Marion Newlevant: [Agrees]

Lea Alcantara: And I remember because the way I learned Twig was I went through the Mijingo tutorials and Ryan wrote his templates one way, and then I started looking at the Craft documentation and Brandon wrote it slightly different.

Marion Newlevant: [Agrees]

Lea Alcantara: And literally, it's the same thing, but simply two different ways. That's functions or objects, right?

Marion Newlevant: Right. The function chaining or the objects.

Lea Alcantara: Yeah, I'm still not a 100% clear why you do one versus the other.

Marion Newlevant: I think a lot of it personal preference.

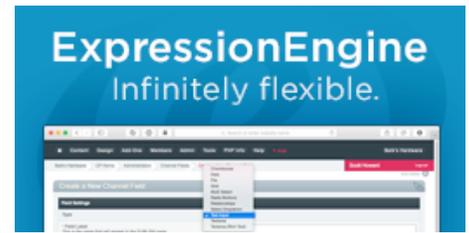
Lea Alcantara: [Agrees]

Marion Newlevant: And so speaking of my personal preferences, if things are not complicated, then I find the function chaining to be maybe just a little bit easier to read and more friendly.

Lea Alcantara: Yeah, and that's what I've been doing.

Marion Newlevant: Right.

Lea Alcantara: Yeah.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: But if things start getting complicated and I get to the point where I'm not quite sure what's going on, then I find that using an object feels to me more explicit and like I more exactly know what is going on and I have a little better control over things. So I will fall back to that when things start...

Lea Alcantara: Not working?

Marion Newlevant: Not working or not feeling clear and simple to me, and also if you're going to be doing or repeating the same function chaining again and again, then you might as well create an object and reuse it so you can get more reuse out of the object thing.

Lea Alcantara: Sure.

Emily Lewis: With something like that, there are things that are personal preference, but are there things that you've seen particularly in kind of a new, very open community common Twig mistakes that people are doing that they should avoid doing?

Marion Newlevant: Yeah, I am thinking about things that I have done in StackExchange where people were asking questions. Probably the first one, and people don't make this mistake and keep going with it because it brings things through a crashing halt, but if you ever find yourself putting those double curly bracket Twig tags inside of the curly bracket percent sign Twig tags, that's wrong.

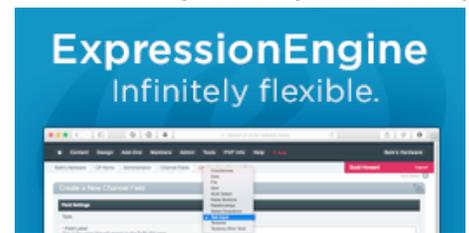
Lea Alcantara: Yeah, okay.

Emily Lewis: [Laughs]

Marion Newlevant: That's just not Twig.

Lea Alcantara: It's very common. Okay, yeah.

Emily Lewis: So no double curly brackets inside curly bracket percentage.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Timestamp: 00:29:46

Marion Newlevant: Right. Both curly bracket thing means it's like a PHP eco. It means print thing out, and what's inside there is whatever, something that evaluates to a string, and if you want something that evaluates to a string because you want to test it or something inside the curly bracket percents, then that's fine. It's a string, but you're not printing it out at that point. You're just evaluating it otherwise. So that's one thing that I see people are doing wrong. One Twig mistake I often make so therefore it feels common to me.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Marion Newlevant: If you have a template that's extending another template, then the template that's doing the extending can't directly output anything. All it can do is fill in those blocks.

Emily Lewis: [Agrees]

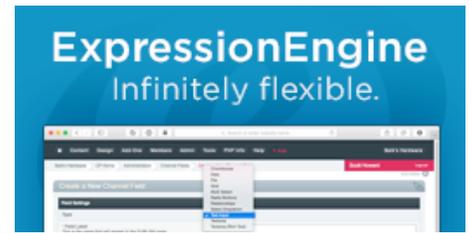
Lea Alcantara: Yes.

Marion Newlevant: And if you try and directly output something, then bang, you get what becomes a really familiar Twig error and you go, "Oh yeah. That again."

Emily Lewis: [Laughs]

Marion Newlevant: Things that people get hung up on and maybe stay hung up on a little more is if you want to test. Often, you will want to test, does this asset exist, so you'll have entry.myassetfield and you say, "I have got one of those." And if you say, "If entry.myassetfield blah, blah, blah," you'd think, "Okay, that should work." But it doesn't because that's always going to evaluate to true.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: The question that you want to be asking is not does that field exist, but is there anything in it.

Emily Lewis: Oh.

Lea Alcantara: Right, right. Is it empty?

Marion Newlevant: Yes, you say you want to get the length of that field, you use the Twig length filter.

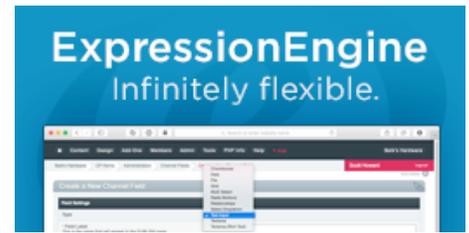
Lea Alcantara: Right.

Marion Newlevant: So `entry.myassetfield | length`, and then that will return you an integer which will be zero or one or twenty or whatever.

Lea Alcantara: Right.

Marion Newlevant: And if you just say, "If thing of the length," and if it's zero, that's false, and if it's not zero, that's true, and so that will be a good test for you, so that works, and the other thing that I'd see people doing this is a pet peeve of mine is in the control panel, you can say, "Here's my channel. Here's my field layout, and this asset, this is required. It is required and there's the limit of one." So I know that there's going to be one of those, and so you'll code into your template, just print the URL of the first one.

Okay, you've gotten past the point of just trying to print the URL of the thing without having gotten the first one of the thing. But the problem with that is that it may be a required field, but that doesn't actually mean that it's necessarily always there. If somebody has deleted that asset out from under you, then maybe you required but it's not there anyway, so you should always, always test that something an asset for an entry or a user, they are all the same things underneath actually exists before you try and reference fields on it. So before you try and get the URL of the first asset, check



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

that you have a first asset, and if not, do something other than nothing at all. Do something other than give the poor end user a Twig template error.

Emily Lewis: So I'm curious, especially because you do seem to have a good familiarity of Twig itself, do you think there's something that Craft should do better or "exploit to better harness the power of Twig?"

Marion Newlevant: My Twig experience is really all in the context of Craft, so if there is something I don't come at it from saying, "Well, in this other system, they do it so much better." Because I don't know of any other systems that use Twig. I'm not familiar with any of them. From what I know of Twig, I'm not seeing anything that I'm saying, "Well, boy, those Craft people should surface this more in their interface." But you know there are some things that they might call out in their documentations and say, "Hey, you guys, you might do it this way."

Lea Alcantara: [Agrees]

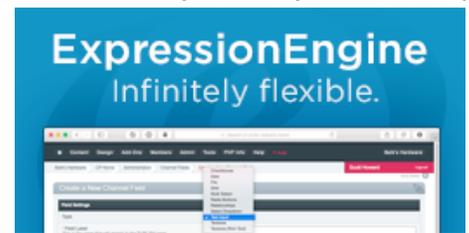
Marion Newlevant: But I don't think that there are really missed opportunities in Craft.

Emily Lewis: So with that documentation in mind, and I have to admit, I haven't gone deep in the documentation, but this macro thing was kind of new to me. Are there samples or examples that you think Craft should be adding to the documentation that just aren't there yet specifics?

Marion Newlevant: Oh.

Emily Lewis: Well, like the thing you mentioned with pagination and how once you've reached the limit with what Craft provides, you then build something yourself with macros. Is that something that's like a case study that's been documented with things like that?

Marion Newlevant: I don't think that's in the formal Craft documentation. I know it's out there on StackExchange.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: [Agrees]

Marion Newlevant: So that's a useful thing. I think that I'd like to see some best practices documentation for like, "Here are some more examples of how we think you should do things," perhaps all in one place.

Emily Lewis: [Agrees]

Marion Newlevant: That would be nice.

Emily Lewis: I found that Craft's documentation is really approachable. It's better than some that I've seen from not CMSs, but just technology in general documentation.

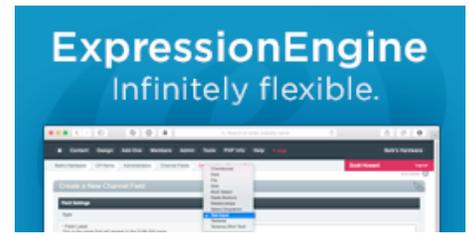
Marion Newlevant: Yeah, I think that Craft documentation really is good. It's not super extensive, but it's also not full of dead ends and obsolete things.

Emily Lewis: [Agrees]

Marion Newlevant: If I had one wish list, I might say it would be nice to see more examples of how to do things with the cache tag.

Emily Lewis: So I do want to mention, because we're getting towards the end of our conversation, that you had sent us some really good links and I just want to make sure our listeners know we will be including these in our show notes. But there is a particularly great presentation from Anthony, I think it's Colangelo, on the template organization in Craft and I think it gives a really nice summary of how you can approach what we talked about earlier with you, Marion, using the extend for like a layout, and then includes and also generally how he's recommending organizing the file structure.

Marion Newlevant: [Agrees]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: So that's a great resource to check out. Before we wrap up though, I'm really curious, you've mentioned StackExchange a couple of times, but even before we started this conversation, we had a number of our listeners on Twitter really excited that you were on the show because literally this is what @narration_sd said, he said, "Marion is like a hidden forest on Craft's StackExchange," and so I'm wondering if you could talk briefly about your experience with StackExchange and why do you decided to be so active on it.

Marion Newlevant: I don't know. It's like a video game for nerds.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Marion Newlevant: So it's slow moving. You get to level up. You get points. So it's kind of fun that way. I think that the web community has been so friendly and welcoming to me that it's awe inspiring, and if I can give back a little bit, then that's great.

Emily Lewis: That's how I've always felt about our industry. I feel very privileged that I'm part of a community where sharing information is just so commonplace helping people out.

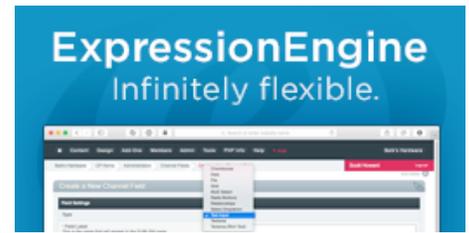
Marion Newlevant: It's not true of the whole software field. Not to knock any other parts of it, but this little corner is so amazingly friendly that it's pretty warm and nurturing in that sense. [Laughs]

Emily Lewis: Yeah.

Lea Alcantara: Yeah.

Marion Newlevant: But not true everywhere.

Lea Alcantara: So it sounds like positive feedback loop. You get to play a game, you get to help out people, and then they appreciate it, so you keep coming back.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: Right.

Emily Lewis: And it's creating additional documentation for everyone else who comes down and encounters the same kind of question or problem.

Marion Newlevant: Yeah.

Lea Alcantara: All right, so before we wrap up, what's your final best practice tips when coding with Twig?

Marion Newlevant: Comment your code.

Emily Lewis: [Agrees]

Lea Alcantara: Oh, excellent.

Emily Lewis: That's just a good coding tip, period.

Lea Alcantara: Yeah, I know, for sure.

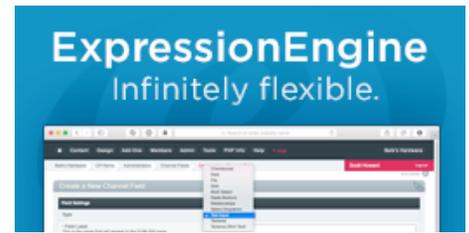
Marion Newlevant: Yes, yes, the most likely person who's going to want to read these comments and understand them is you six months on and so be nice to yourself. Explain what the hell you were doing.

Lea Alcantara: Excellent. So before we finish up, we've got our rapid-fire ten questions, so our listeners can get to know you a bit better. Are you ready, Marion?

Marion Newlevant: Sure.

Lea Alcantara: All right, first question, Android or iOS?

Marion Newlevant: It's iOS.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: If you're stranded on a desert island and can only bring three things, what would you bring?

Marion Newlevant: I want to bring a Swiss army knife. I'm going to bring a rescue flare and lots of drinking water.

Emily Lewis: Those are the smartest ones.

Lea Alcantara: That was so practical.

Emily Lewis: You're a very practical woman.

Lea Alcantara: What's your favorite TV show?

Marion Newlevant: I think the black screen of television off.

Emily Lewis: [Laughs]

Marion Newlevant: I haven't actually lived with a television since 1977.

Lea Alcantara: Oh, wow!

Marion Newlevant: And so I'm really, really unable to ignore them when they're on.

Emily Lewis: All right, what's your favorite dessert?

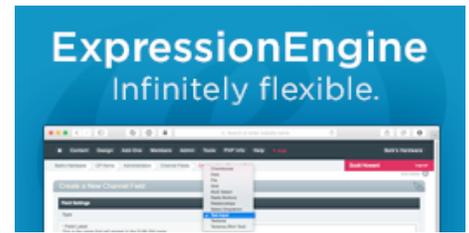
Marion Newlevant: Things with chocolate, I guess.

Lea Alcantara: What profession other than yours would you like to attempt?

Marion Newlevant: Oh, I'm a real dabbler, so font designer.

Emily Lewis: Oh.

Lea Alcantara: Oh!



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: What profession would you not like to try?

Marion Newlevant: Anything that involves making cold calls.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: What's the latest article or blog post you've read?

Marion Newlevant: Twig documentation. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

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

Marion Newlevant: I have a super power, it's failing to recognize people. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: Like if you saw someone you know in the supermarket, you wouldn't recognize them?

Marion Newlevant: Completely.

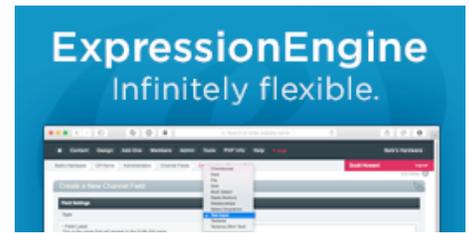
Lea Alcantara: That could be useful actually. [Laughs]

Marion Newlevant: [Laughs] Well, you can have it. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Marion Newlevant: I'll trade.



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

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

Marion Newlevant: Silent. When I worked in a cube and I had people around me, then I used to listen to vocal music and languages other than English, but now that I work at home, I do better without anything also coming into my ears.

Emily Lewis: No distractions.

Marion Newlevant: [Agrees]

Emily Lewis: All right, last question, cats or dogs?

Marion Newlevant: Cats and also snakes.

Emily Lewis: Oh, that's interesting.

Lea Alcantara: Oh, interesting.

Emily Lewis: Not together. [Laughs]

Marion Newlevant: No.

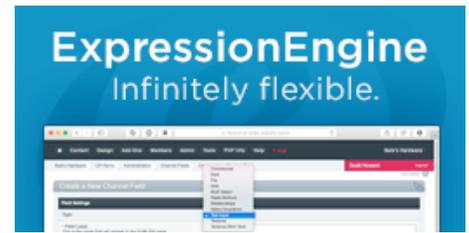
Emily Lewis: Do you have a pet snake?

Marion Newlevant: Not anymore.

Emily Lewis: But you did, yeah?

Marion Newlevant: Yeah, I could be a snake sitter. That's what I want for my profession, you know?

Emily Lewis: [Laughs]



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Marion Newlevant: Other people could leave their snakes with me when they go on vacation and I could make sure their boas get lots of handling until then and I could give them back when they come home from vacation and before they have to be fed.

Emily Lewis: Actually, it seems like a real niched area you might want to consider like putting it out in a local newsletter. I bet there are people who need help.

Marion Newlevant: Do contact me on Twitter if you're local to Portland and want your boas pet sit.
[Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Lea Alcantara: So that's all the time we have for today. Thanks for joining us, Marion.

Marion Newlevant: It's a pleasure.

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

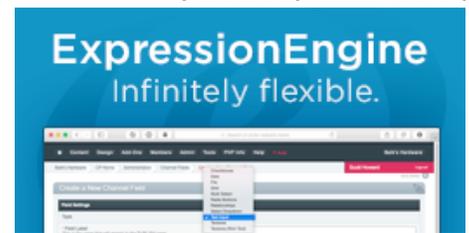
Marion Newlevant: I'm marion.newlevant@gmail.com, and I'm also [@marionnewlevant](https://twitter.com/marionnewlevant) on Twitter. There are three Newlevants in the universe so you'll get me or my husband or my daughter and it's not that hard to tell us apart.

Emily Lewis: Oh wow! It was so great talking with you, Marion. I learned a lot. It was also really fun. Thank you again.

Marion Newlevant: Oh, you're welcome. It was a pleasure.

[Music starts]

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



<http://ctrlclickcast.com/episodes/twig-templating-for-craft>

Emily Lewis: And thanks to 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 our friend, Marcus Neto returns to the show this time to talk about pricing for profit. 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: *00:41:05*