Sass only really benefits people who havenÂ¤t mastered writing CSS properly

Friday 22 January 2010 08:27

Harry Roberts is right. Sass is nothing more than a work–around for a work–around that CSS authors that know how to write proper CSS in the first place don't need. But we can't just blame the HTML monkeys.

HTML and CSS thrive in an environment where everything revolves around semantics. In a workflow where the content is created first and more or less semantic markup is added later and finally that markup is used for styling, it all comes together naturally as intended. But with today's focus on cutting costs and thus reusable frameworks the process is often completely reversed. In that situation a CSS author has to create a style sheet before anything is known about the content and the semantics that will be forced in it. The only thing the author then can come up with is a very elaborate general purpose collection of rules and selectors that should be able to handle about any content in an orthogonal and diagonal way at the same time. Of course that will result in an incredible mess of CSS for which you'll need all the help you can get to stay sane handling it. It's like designing a datamodel in which you aren't allowed to do any normalization because it has to be so generic you can't tell what data is of the same type.

If you know what content you're handling you can in a datamodel for example easily tell suppliers, employers and clients all have something that resembles an address, so you only need one generic address type in your model. CSS created in the reverse workflow can't tell yet if it boils down to the same type and has to stick with three seperate definitions. Of course the nesting feature of Sass comes in handy to define that mess.

Variables are something that is indeed missing in CSS. But not in the way Sass aims to fix it. The practical use and nature of CSS often lead to having to define the same color value to varying properties. If it's just about the same property it's possible to gather all the selectors and define the property in a single rule, but you can't link for example a border–color and a background–color so changing a single value changes both. What Sass adds beyond that is a work–around for when the HTML monkey isn't allowed to put additional class names in the markup. Once you have defined the name of the variable in a meaningful way the same effect can be achieved by using that variable as an extra classname and add it to the selectors. The same is valid for mixins: Once you know why a single mixin is used for several selectors, the proper solution is to fix it by adding the semantics to the project.

But because in the reverse workflow it's not possible to give the variables and mixins semantic names, adding classnames breaks out of the illusion of seperation of content and presentation. In the normal workflow you'll notice the semantic relation between elements first, and have no problem adding those semantics to the markup in the form of a classname and happily use that as a hook for style information. The reverse workflow prevents that logic.

Variables and mixins in CSS are often a sign of latent semantics. If the same design is used for multiple elements, a CSS author should inquire if there is a semantic reasoning behind it or the elements have different semantics and have just the same design. In the reverse workflow this is futile because the designer just designs some generic boxes and the CSS author just styles some DIVs without caring about semantics. And if CSS isn't based on semantics it's all a bit pointless anyway. Then you get maintenance issues like a client wants to have public holidays a different color and the HTML monkey wasn't even aware there were dates attached to his perfectly styled generic boxes. Often the only way out is then to blame someone and tell the client it's impossible and screw experience design. You know clients who can't afford a custom solution can't afford experience design tests anyway.

What Sass is good at is making it easier to do by hand what tools like Dreamweaver are even better at. There is no point in writing CSS by hand if that means you've thrown all the added value of what that meant away and try to combine the brand value of hand coded CSS with the ease of generic tools. Nobody is interested in a hand built Volkswagen.

by Roland van Ipenburg
http://www.xs4all.nl/~ipenburg/blog/posts/work/2010/01/22/sass-only-really-benefits-people-who-havent-mastered-writing/