SASS POUR LES WEB DESIGNERS pdf

1: CSS Preprocessors Compared: Sass vs. LESS - Hongkiat

Sass Pour Les Web Designers N0 1 PDF - www.enganchecubano.com - Sass Workflow - Using Sass - Sass and Media Queries. - Dan Cederholm is a designer, author, and.

That allows me to import those same variables in other stylesheetsâ€”for example, in other pages or sections of the same project, where the style guide appliesâ€”without also importing the rest of the layout. So, you have the benefits of a single file download, with the flexibility of multiple files that contain reusable code. The Compass framework With mixins, variable files, and import, you can build your own mighty CSS frameworks to save an immense amount of time when starting new projects. Use import to merge chunks of your SCSS into one file. Taking that organizational advantage to the next level is Compass http: Compass offers many pre-written CSS patterns, which will be updated as the properties evolve and vendor prefixes can be stripped away. Compass also makes image sprites and typographic systems easier to handle Fig 3. But as a learning experience, other frameworks can be incredibly beneficialâ€”they let you see how others set up projects and increase their efficiency. And Compass is no exception. A little Googling will unearth many helpful Sassers sharing their mixins on GitHub or their own sites. CSS3 stack that looks ugly and unmaintainable already has a mixin written for it, so take advantage of the community! You have an alert message style with two options at the top of the page. One style handles standard alerts Fig 3. These styles are almost identical, save for the background color. Typically, we might create a base class for normal alerts and then a second class that overrides the background color only. With the styles for each alert set up like so: Two alert styles on the Sasquatch Records site. Additionally, we can then add extra overriding rules to make a new unique style without duplicating the shared styles. So if we wanted to use extend to handle the two types of alert messages, we could simplify the markup to just one class: This is a positive alert! Using extend also allows us to be terser in our semantics, defining class names based on meaning rather than appearance. Multiple extends You can also extend multiple classes within a declaration, which chains together all the styles from each class: You can reference placeholders using extend. If you find yourself using a mixin over and over throughout the stylesheet, keep in mind how that will compile, and consider whether it makes sense to use extend or turn those repeated styles into a class that gets reused in the markup instead. Extending the same class repeatedly throughout the stylesheet can result in a monster declaration. You now have the power! Sass is capable of even more, if you want to dive deeper. But in this final chapter, I do want to talk about some advanced techniques with Sass and media queries that have greatly simplified some otherwise complex CSS in my daily work. Sass is as powerful as you want it to be. Using it for variables and a few mixins will make your life easier. But it can go beyond that if you want it to. S a ss a nd M edi a Q ueries 69 For instance, you may want to adjust the width of a containing element should the browser be less than pixels wide, using a media query: Using variables to define breakpoints Media-query bubbling is a wonderful convenience that Sass brings to responsive design, but there is still quite a bit of repetition. Sass variables to the rescue! The above SCSS will compile to: But we can even simplify things further with content blocks, also introduced in Sass 3. The content placeholder allows us to further pass blocks of styles to the mixin that get inserted inside the media query. With this single mixin set up, we can now call it from any declaration using a compact pattern that reflects the way we think about things: Sass feeds any styles to the appropriate media query and reconstructs the declaration with everything in its right place. Using content blocks for writing contextually-placed media queries makes responsive design a heck of a lot simplerâ€” with less repetition. The entire progression is spelled out in one spot: Ideally, Sass would let us nest the queries to keep the contextual connection of rules collected in one space, but then group shared media query rules when compiled. Normal-resolution logo on the left, HiDPI version on the right. That means beautiful clarity and saying goodbye to fuzzy pixels! But only if you take the time to create graphics that reflect this super-sharp new world. For elements on the page, this typically means creating images twice as large and compressing them to half their size by using the width attribute in the markup. Social network links in the sidebar of the Sasquatch Records site. For background images, we simply use CSS media queries in modern browsers that support them to determine whether the

display is HiDPI and render the appropriately sized image. On the Sasquatch Records site, for instance, we have a little list of social network links in the sidebar. In the Dribbble link, the CSS to align a normal-resolution icon to the left of the text might look like this: To detect whether the display is HiDPI, we use a media query and the min-device-pixel-ratio property in CSS3 which varies depending on the browser vendor. The difference, when viewed on a HiDPI display, is stunningly sharp. All those fuzzy edges disappear Fig 4. We can create a Sass mixin that handles all the heavy lifting, even forming two different file names with a bit of crafty concatenation. But I think -2x works just fine. Mixins can include other mixins. A mixinception, if you will. In fact, we can go a step further in DRY-ing up this code, extracting the vendorprefixed min-device-pixel-ratio rules into a variable and the background-size property stack into its own mixin. Then these extracted parts could be reused in other sections of the stylesheet or additional mixins. As mentioned earlier in the chapter, a variable that appears within a selector needs special interpolation characters around it: Anytime we want to use background-size on a selector, we can now call this mixin: But Sass can be as powerful as you want it to be. Now go simplify your stylesheets with reusable blocks of Sassy CSS, save yourself a boatload of time, and, most important, build awesome things! Chris Coyier has covered Sass pretty extensively, and we share very similar views on how Sass can help the CSS hand-coder http: In particular, be sure to check out his Sass Style Guide http: Lots of great insight on where Sass is headed http: An extensive framework for Sass created by Chris Eppstein. A nice collection of Sass mixins from Jake Bresnehan http: A plugin for Sass that makes writing media queries even simpler http: A helper for Compass and Sass for creating responsive grid systems http: A kick-start framework for creating responsive web design projects using Compass and Sass. Also has a great name http: A handy Firefox add-on that will display the original Sass filename and line number of Sass-compiled stylesheets, for debugging http: A tutorial on how to best use Chrome while developing with Sass. Some of it is experimental, but you can bet more of this kind of thing will emerge as Sass continues to gain steam http: LESS also has a client-side option for compiling, which serves. However, client-side compiling is not recommended for production sites. From my perspective, Sass has a more active development cycle and community behind it, and is a bit more robust in its functionality. Both will help immensely in easing the creation of smart stylesheets. The important thing is to take advantage of one of these tools to make your life a little easier http: Chapter 1 1 http: Rich cured my reluctance in using Sass by way of relentless prodding and evangelism while we worked on our little design community. It only took a whole year. Thanks for being persistent, Rich! They are a fantastic team, and quite frankly make it difficult to think of writing anywhere else. Such a pleasure working with you. To Erin Kissane, thank you for making me sound like a better writer than I actually am. And to Jina Bolton for being a wonderful technical editor and ambassador for Sass. A gigantic thanks to Hampton Catlin for inventing Sass, and Nathan Weizenbaum and Chris Eppstein for developing Sass and making it the indispensable tool that it is today. And finally, thanks to you, for reading.

SASS POUR LES WEB DESIGNERS pdf

2: sass_for_web_designers

Sass for Web Designers proves once again, A Book Apart's series of self help books, get you to do more. I think that is the goal of self help books so I generally go for all their books. This book had me shaking my fist in the air, not at anything thing the author implies(not really) but at the time I have wasted not already picking up the SASS.

I think we have to agree Sass and Compass is a great duo and the Sprite image feature is really kickass, so one point for Sass here. They still do the same thing: Below, we will examine some of the most commonly used languages both in Sass and LESS based on my experience. So, I think it is clear which one is doing better in this case. Here is an example: In Sass, this method is taken further with Selector Inheritance. The concept is identical, but instead of copying the whole properties, Sass will extend or group selectors that have the same properties and values using the extend directive. Take a look at this example below: Sass is one step ahead by distinct Mixins and Selectors Inheritance. See how they perform this random calculation: Imagine thousands of lines of code and a tiny bit of error somewhere in the chaos. A clear error notification will be the best way to figure out the problem quickly. Sass will generate an error notification whenever there is invalidity in the code. In this case we will remove one semicolon on line 6, and this should turn to an error. Take a look at the screenshot below. When I first saw this notification, I could hardly understand it. Also, it seems Sass is slightly off with where the error is. It said that the error is on line 7, instead of 6. With the same error scenario, LESS notification is more well-presented and it also appears to be more accurate. Take a look at this screenshot: LESS delivers better experience on this matter, and wins hands down. Sass 4 â€" LESS 2 Documentation Documentation is very crucial part for every product; even seasoned developers would find it difficult to do things without Documentation. If we take a look the documentation over at the official site, I, personally, feel like I am in the middle of a library, the documentation is very comprehensive. Yet, the look and feel, if that matters to you, is not motivational for reading, plus the background is plain white. The presentation is much more like W3 documentation or WikiPedia. On the other hand, LESS documentation is clearer without too many text explanations, and it dives straight into the examples. It is also has good typography and a better color scheme. I think this was why LESS attracted my attention in the first place and I can learn it faster because of the layout and presentation of the documentation. The LESS documentation presentation is better, although Sass has more comprehensive documentation, so I think we can call this one a tie. Sass 5 â€" LESS 3. Be it Sass or LESS, as long as they are comfortable and more productive, then that is the winner in their list. Lastly, if you have something in mind about this subject, feel free to share it in the comment box below.

SASS POUR LES WEB DESIGNERS pdf

3: Sass for Web Designers by Dan Cederholm

WorldCat is the world's largest library catalog, helping you find library materials www.enganchecubano.com more â€ºâ€º.

Clearly organized and full of excellent, practical examples. The book is concise, easy to read and understand. I like the passion in the author, the enthusiasm is definitely palatable throughout the book. Jul 12, Jesse rated it really liked it Solid, tight, and entertaining intro to Sass. Oct 01, Anton Serdyuchenko rated it it was amazing Excellent book helps from setting up the environment to work with the basic functions of the language. For me, this book has become a great revelation that the process of writing styles can be optimized. Also a very useful description of the environment settings. I often have difficulty setting up the tool, so the author has helped me a lot. I was happy to apply some of the techniques from this book on my website. Maybe at the moment there have been no major changes in the process of my developm Excellent book helps from setting up the environment to work with the basic functions of the language. Maybe at the moment there have been no major changes in the process of my development, but I am sure that after a while, learning sass will help me a lot. I also plan to return to this book in order to use this or that technique. Many thanks to the author for an excellent book. I recommend reading this. Easy to jump in. Dan writes with his typical easy style. They are nice books, but very expensive. The reason this is a great book for those thinking of playing with Sass is that Dan weaves reference in with narrative. He eases you in, instead of just providing a huge cheat sheet. Plus the art direction, like the other books, is delicious.

SASS POUR LES WEB DESIGNERS pdf

4: TÃ©lÃ©charger Livre Sass pour les web designers, nÂ° 10 PDF FranÃ§ais | Telechargeryib

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

If you want to just browse here, go ahead, but we recommend you go install Sass first. Go here if you want to learn how to get everything setup. Preprocessing CSS on its own can be fun, but stylesheets are getting larger, more complex, and harder to maintain. This is where a preprocessor can help. Once you start tinkering with Sass, it will take your preprocessed Sass file and save it as a normal CSS file that you can use in your website. The most direct way to make this happen is in your terminal. For example, running sass input. You can also watch individual files or directories with the --watch flag. The watch flag tells Sass to watch your source files for changes, and re-compile CSS each time you save your Sass. If you wanted to watch instead of manually build your input. Variables Think of variables as a way to store information that you want to reuse throughout your stylesheet. Be aware that overly nested rules will result in over-qualified CSS that could prove hard to maintain and is generally considered bad practice. This is a great way to organize your CSS and make it more readable. Partials You can create partial Sass files that contain little snippets of CSS that you can include in other Sass files. This is a great way to modularize your CSS and help keep things easier to maintain. A partial is simply a Sass file named with a leading underscore. The underscore lets Sass know that the file is only a partial file and that it should not be generated into a CSS file. Sass partials are used with the import directive. Sass is smart and will figure it out for you. A mixin lets you make groups of CSS declarations that you want to reuse throughout your site. You can even pass in values to make your mixin more flexible. A good use of a mixin is for vendor prefixes. After you create your mixin, you can then use it as a CSS declaration starting with include followed by the name of the mixin. Using extend lets you share a set of CSS properties from one selector to another. It helps keep your Sass very DRY. A placeholder class is a special type of class that only prints when it is extended, and can help keep your compiled CSS neat and clean. This helps you avoid having to write multiple class names on HTML elements. Operators Doing math in your CSS is very helpful. Operations in Sass let us do something like take pixel values and convert them to percentages without much hassle.

5: Sass: Sass Basics

Dan Cederholm,Chris Coyier,Charles Robert TÃ©lÃ©charger Sass pour les web designers, nÂ° 10 Livre PDF FranÃ§ais www.enganchecubano.comt The Drive CHWK FM, Chilliwack Please describe the issue you experienced.

CSS is a declarative, not a programming language. This simply means that the style properties and values that we declare within the rules of CSS are exactly what the browser uses to paint the screen. A programming language on the other hand provides some means of defining logic. Crudely put, a logical statement might be in the form of: A programming language also facilitates variables. These can be thought of as placeholders for something reusable for example, one might have a variable for a specific color value. Then there are functions, a means to manipulate values with operations for example, make this color 20 percent lighter. Sass and Compass provide these capabilities and more. If some of the terminology in that last paragraph sounded alien, fear not. We will be dealing with all those concepts in due course. After all, we use CSS everyday. We can hopefully use it to fix all the usual layout problems that get thrown at us, build responsive websites that are displayed beautifully on all devices, and generally make ourselves feel like, for the most part, we know what we are doing. However, Sass enables us to write CSS faster and more easily while also keeping the style sheets far more maintainable. The chances are, if you are reading this, you have already done a little research and decided to look into Sass as opposed to LESS or Stylus. Why you should use Sass and Compass As mentioned previously, there is a growing number of organizations such as the BBC, eBay, and LinkedIn that have already embraced Sass and Compass and use it to write and maintain their style sheets. It stands to reason that when large organizations are switching from writing CSS directly to using the Sass preprocessor there must be some enormous economies to make it worth their while. Usually as a hex hexadecimal value like bfbfbf. However many it ends up being, I often in fact usually struggle to remember hex values, especially with colors in a site. With Sass, we can just define the colors as variables. A variable is merely a mechanism for referencing a value. Take at look at these three variables: Then comes the name of the variable with no space after the dollar sign. Then a colon indicates the end of the variable name, meaning that the value will come after the colon but before a closing semicolon. With those colors defined as variables, we can use them anywhere in the Sass style sheet like this: For our purposes, all you need to know is that compile just means go from Sass in either. Writing and remembering variable names is far easier than remembering unnatural hex values, no? Furthermore, when those color values need to change, they only need changing at the variable and everywhere else takes care of itself. To exemplify that technique, to enable a color with some alpha transparency, at present we might do this: Before Sass, I had a handy little application just for the job. Now, I am pleased to say that thanks to Sass, that application has gone the way of the Dodo. With Sass, it is possible to simply do this: When compiled, it produces the following CSS code: The alpha channel is at 90 percent. This means that we can see 10 percent of whatever is behind the color in browsers that understand RGBA. It lets us ditch images and do more things with pure CSS than ever before. However, implementing these new features background gradients, box-shadows, transformations, and a few more , that are often still experimental features, often requires the use of vendor prefixes and occasionally different syntaxes. You know the drill. As an example, this is what historically has been needed for rounded corners: Instead of remembering prefixes and associated syntaxes, you can just write the following code: Nesting rules Sass allows rules to be nested within one another. So for example, if you wanted to make a set of links with a nav element and provide alternative pseudo states for hover and active states, you could write this in Sass: This probably looks more complicated than it actually is. For instance, here is an example of the kind of CSS rule that can make life difficult: From a maintainability point of view it would be far easier to simply have this to achieve the same effect: In CSS terms, this typically involves writing lots of media queries for the different breakpoints in a design. Personally, I think this is quite verbose and a lot to remember. With Sass, instead, it is possible to do this: The value of the variable is then used inside a mixin called MQ. Using that MQ mixin, media queries can be placed easily wherever they are needed, arguably making it simpler to understand where they are being applied in the code. How do you compress CSS before using it on a live site? Compressing the CSS makes it a

fraction of its original size, resulting in faster code for every device that requests it. Compressing CSS can easily reduce the file size by half its original uncompressed size. It just does it. Sass can be configured to compile the CSS in a number of formats, one of which is compressed. So as soon as the Sass file is saved, it gets automatically compiled into compressed CSS; production ready and in the smallest possible file size. The website of Sass http: Sass both provides a simpler, more elegant syntax for CSS and implements various features that are useful for creating manageable style sheets. Since then, it has been fostered, loved, and cared for by a number of others. However, most notable in the development of Sass is Nathan Weizembaum the primary designer and developer who developed Sass with Hampton Catlin until Version 2 and Christopher Eppstein who joined Weizembaum on the Sass core team in and developed the project with him from Version 2. Eppstein is also the creator of the Compass framework. There are also a number of other contributors to the Sass project. As Sass started life in the Ruby community Ruby is itself a programming language , much of the documentation associated with Sass has always been programmer friendly. Historically, this has made Sass difficult for non-programmers to get their heads around. This is a great shame, as designers who also write their own frontend code arguably stand to benefit as much as anyone else from its power and features. Sass also supports two syntaxes. The original syntax known as Sass, with files ending in a. You can find more documentation on the indented syntax at http: The syntax we will be using throughout this book is the SCSS based syntax, with Sass files ending in the. This syntax is more verbose than the original indent-based syntax but similar to existing CSS. Take a look at http: Sass is free to use, requiring no license. The Compass website is at http: It describes itself as follows: In fact, Compass was the first Sass-based framework. What this boils down to is that by installing Compass alongside Sass we get access to lots and lots of reusable patterns and tools for easily creating CSS. Have you seen the TV show Pimp my ride? Put another way, Compass lets you write CSS3 goodies like box-shadow, gradients, columns, and transforms with a single syntax, and it magically creates cross-browser compatible CSS of everything that just works without headaches. It also paves the way for additional plugins to enable incredible, lightweight grid systems that we will be looking at in due course. This means that while you can use it as much as you like, you are encouraged to make a donation to help the UMDF find a cure for mitochondrial disease. Install Sass and Compass In days gone by, to use Sass and Compass, it was necessary to use the command line to install them. You can always come back here later. Compass creator Chris Eppstein has created a graphical installer package. Just head to https: However, understanding just a little about the command line will be beneficial, so you may opt to flex your command line muscles just a little and install from there. Getting around on the command line There are only a few commands that are essential to work with Sass and Compass on the command line. List the items in the current folder: If you are on OS X, you already have it. Installing Ruby on Windows If you are on Windows, head over to http: Linux users should be able to install Ruby direct from their package manager. Compass actually requires Sass, so by installing Compass, Ruby will automatically install Sass too. Think of a gem as a tiny application or plugin; it simply extends the functionality of something that uses Ruby. For example, there are gems for grid systems like Susy, gems for button styles like Sassy Buttons, and lots, lots more. Now just type this and then press Enter: Just type it in and press Enter. When entering a password, the Terminal provides no feedback. With that done, Compass and Sass will install. How about that, you just installed a Ruby gem. Do you feel like a nerd now? Windows command prompt install For Windows users using Windows 7 or Vista, just click on the Start button and type ruby, and then click on the highlighted Start Command Prompt with Ruby option.

6: Sass for Web Designers - Avivia | www.enganchecubano.com

Chris Coyier is the author of Practical SVG (avg rating, 96 ratings, 9 reviews), Digging Into WordPress (avg rating, ratings, 9 reviews, pu.

7: PDF Sass and Compass for Designers - Packt | www.enganchecubano.com

SASS POUR LES WEB DESIGNERS pdf

Here is Download Sass pour les web designers, nÂ° 10 or Read online Sass pour les web designers, nÂ° 10 Download Now Read Online. price: EUR 12,00 Buy Now. Sass pour les web designers, nÂ° 10 Download Link.

8: Chris Coyier (Preface of SASS pour les web designers)

Why we are the best website for downloading this sass pour les web designers n0 1 Of course, you could choose the book in various report types and also media. Try to.

9: Sass (stylesheet language) - Wikipedia

Sass stands for Syntactically Awesome Stylesheets.. Sass and is basically just an extension to CSS that help us write more flexible styles. It helps us make larger and complicated stylesheets clearer to understand and easier to maintain.

SASS POUR LES WEB DESIGNERS pdf

Drug Therapy in Nursing, Second Edition and Lippincotts Nursing Drug Guide 2007, Canadian Version Grandmothers secrets Seven days to remember Gps and society research paper Bradley County (TN (Images of America) Mickey mouse magazine General index to the evidence [etc. given in vol I-III. 1875 (C. 1363 Third[-eight report[s 1873-75: 3d (Evolution of Freemasonry Colored Theatre Parts University of Illinois Pork Industry Conference Constraint Propagation in Flexible Manufacturing Grade 11 exam papers and memos 2014 A tale of a hero and the song of her sword Travellers Iceland, 2nd Ambari archaeological site Disney fake book 3 What is osi reference model Tales from College Footballs Sidelines The family office book investing capital for the ultra-affluent The Night Angel Trilogy Boxed Set On trial without knowing it A guide to Confederate monuments in South Carolina Financial management issues Hidden Job Market 2000 Anti-Government Opposition during Khrushchev-Brezhnev Years CRC handbook of methods for oxygen radical research Social networks of older people Square peg round hole good engineering mishra Response by the government to the fourth report from the social services committee session 1988-89 on res Our Solar System (Jump Space) Criminality in the Philippine Islands, 1903-1908. Macrame fashions and furnishings The stage as a moral institution (1784 ; On the use of chorus in tragedy (1803 Friedrich Schiller 150 Cards for All Seasons La campanella liszt piano Fundamentals of oil and gas accounting 5th edition Excerpts from Born After Midnight Responsibility of the southern white man to the Negro, by A. H. Stone. Ukrainian pictures Feeding the Flame