

Comparativo entre PrÉ-processadores CSS

**Gislaine Ferreira Gonçalves**

Tecnologia em Análise e Desenvolvimento de Sistemas

Thiago Henrique Poiani

Tecnologia em Análise e Desenvolvimento de Sistemas

Thiago Silva Barros

Tecnologia em Análise e Desenvolvimento de Sistemas

Resumo: O CSS é, sem dúvida, um facilitador na inserção de estilos em uma página web, como as fontes, cores, entre outras funcionalidades. Com a evolução do CSS, algumas funções foram tornando-se complexas e a partir da ideia de simplificar folhas de estilo, surgiram alguns pré-processadores. Entre eles, o trabalho elaborado, destaca dois: LESS e SASS. Suas características e vantagens são apresentadas no decorrer do artigo, juntamente com as conclusões a cerca da comparação entre os pré-processadores e relatos da experiência e conhecimento adquirido para desenvolvimento desse trabalho. Alguns componentes desses pré-processadores se destacam e são detalhados no texto com a comparação entre eles e seus respectivos processamentos para geração de folhas de estilo.

Palavras-chave: Pré-processador CSS. Front End. LESS. SASS.

Abstract: CSS is undoubtedly a facilitator in the insertion of styles into a web page, such as fonts, colors, and other features. With the evolution of CSS, some functions were becoming complex and from the idea of ​​simplifying stylesheets, there were some preprocessors. Among them, the work done, highlights two: LESS and SASS. Its features and advantages are presented throughout the article, together with the conclusions about the comparison between the preprocessor and reports the experience and knowledge gained in the development of this work. Some components of these preprocessors stand out and are detailed in the text with the comparison between them and their processing to generate stylesheets.

Keywords: CSS Preprocessor. Front End. LESS. SASS.

1 INTRODUÇÃO

CSS é um acrônimo para Cascading Style Sheets que determina a aparência de uma estrutura criada em HTML. Com CSS pode-se controlar a estilização para diferentes dispositivos (Media Types), como desktop (screen), smartphones (handheld), impressão (print). Atualmente, com CSS3 e por meio das Media Queries - utilização de Media Types envolvendo características de formatações - podemos controlar a estilização dependendo da media e da resolução para criação dos layouts responsivos.

Com a ideia de facilitar a implementação de códigos CSS, os pré-processadores são ferramentas nas quais pode-se inserir o código com uma sintaxe diferente - que varia em alguns aspectos de acordo com cada pré-processador -, permitindo a mudança para a sintaxe do CSS, por meio do pré-processamento, para compreensão dos browsers do código CSS.

Existem vários pré-processadores CSS disponíveis, sendo que os mais populares atualmente são Sass e LESS e no decorrer desse trabalho poderá acompanhar as vantagens da utilização dos pré-processadores e conhecer os conceitos de alguns componentes utilizados por eles.

2 Desenvolvimento

2.1 Materiais e Métodos

Para o desenvolvimento do trabalho foi necessário tanto estudos teóricos quanto práticos a respeito do CSS e de alguns pré-processadores. Os materiais analisados foram as documentações de cada pré-processador utilizado, as avaliações de desenvolvedores front end sobre o uso desses recursos e artigos de comparações entre eles.

2.2 Discussão

Para iniciar a discussão sobre os temas é preciso conhecer alguns conceitos básicos a respeito do tema. O termo CSS vem do inglês Cascading Style Sheets e permite ao programador web ou ao Webdesigner, por exemplo, definir os estilos e como serão exibidas as páginas ao usuário, ou seja, define a aparência dos elementos implementados e visualizados. A princípio, o CSS era uma forma simples para adicionar os estilos, contudo alguns prefixos surgiram, as folhas de estilo ficaram cada vez maiores e mais repetitivas causando transtornos quando havia necessidade de edição do estilo.

Com a dificuldade encontrada com o aperfeiçoamento do CSS surgiram os pré-processadores, com o intuito de facilitar e agilizar o trabalho dos programadores. Uma das funções que os pré-processadores possuem é compreender o desenvolvedor, ou seja, entender a forma adotada pelo programador de passar o comando para o estilo desejado, o que o torna mais amigável.

O primeiro pré-processadores que será abordado e apresentado é o LESS, que amplia, por meio de comportamentos dinâmicos como *variáveis*, *mixins*, *operações* e *funções*, as possibilidades originais do CSS. O LESS é uma linguagem baseada no CSS e que necessita ser compilada para gerar o código CSS que será usado, isso quer dizer que o arquivo ‘. less’ é gerado e a partir dele obtêm-se a folha de estilo em formato ‘.css’.

Dentre os comportamentos dinâmicos do LESS estão as variáveis que permitem valores específicos que podem ser reutilizados em toda a folha de estilo, o que permite que mudanças globais sejam feitas. Outro comportamento utilizado por esse pré-processadores é o *mixins* que viabiliza a inserção e utilização das propriedades em uma determinada classe em qualquer outra classe da mesma folha de estilo. Para isso é necessário apenas incluir o nome da classe dentre as propriedades na classe que deseja reutilizá-la.

Outras duas facilidades do LESS são as operações e funções, na qual as operações possibilitam somar, subtrair, multiplicar e dividir, conforme as quatro operações matemáticas, dentro dos valores de cores, por exemplo. As funções, por outro lado, permitem a mesma manipulação de valores, mas da forma como for necessária. O uso de operações dentro de funções pode facilitar o desenvolvimento das folhas de estilo, que sempre serão compiladas gerando o CSS.

Continuando a apresentação das características dos pré-processadores, temos o SASS, que assim como o LESS deriva do CSS, e é considerado um dos pré-processadores mais estáveis, além de estar entre os mais utilizados para geração de folhas de estilo de forma mais dinâmica, reaproveitando os recursos disponibilizados em suas versões. As variáveis, *mixins* e operações são exemplos de componentes desse pré-processador e serão apresentados na sequência junto às funções de aninhamento (*nesting*), importação (*import*) e modularização (*partials*) do CSS através do SASS.

O procedimento de criação de variáveis inicia-se com o símbolo $ e, assim como no pré-processador LESS, as variáveis podem ser reutilizadas em toda a folha de estilo. Os *mixins* e as operações não são diferentes do que foi apresentado anteriormente, ou seja, independente de qual dos dois pré-processadores seja utilizado esses três comportamentos dinâmicos serão encontrados, visando facilitar o trabalho do desenvolvedor.

A disponibilização do *nesting*, ou aninhamento, possibilita a mesma sequência visual disponível no HTML dentro do código gerado pelo SASS, ou seja, os seletores que forem usados na folha de estilo serão aninhados de forma organizada, o que facilitará na localização para edições e/ou acréscimos de informação ao estilo. O *import*, já existente nas folhas CSS, no SASS permite a divisão de um CSS em partes menores, para que no momento necessário seja solicitado pelo arquivo principal onde consta a chamada da importação de seu conteúdo, evitando assim o processo de requisição utilizado pelo CSS puro, no qual a cada novo *import* realiza-se uma nova solicitação HTTP. Para concluir os componentes citados, anteriormente, apresenta-se o *partials* que nada mais é que a modularização do código. A ideia dessa funcionalidade é criar um arquivo que disponibilize trechos de CSS para incluí-los em outros arquivos '.scss'.

3 Conclusão

Após análise de documentações dos pré-processadores LESS e Sass, observa-se as diferentes vantagens que ambos proporcionam através de suas funcionalidades, mas antes de tudo a escolha pelo uso e quando utilizar é muito pessoal.

Pelas leituras realizadas o Sass possui funcionalidade que agregam as empresas economia de tempo, o que aumenta a produtividade e assim o lucro, por consequência sem perda de qualidade. O seu crescimento e associação a alguns frameworks vem crescendo gradativamente. O LESS por outro lado já está em frameworks de grande utilização, um exemplo é o Bootstrap, que possui uma versão disponível para esse pré-processador.

Existem também algumas ferramentas que trabalham em conjunto com os pré processadores e os tornam ainda mais vantajosos, como é o caso do Compass que associado ao Sass amplia sua eficiência e usabilidade, por meio de funcionalidades como a geração de *sprites*, por exemplo, que é um mapa de imagem, na qual uma única imagem possui todas as outras. O crescimento dos pré-processadores tem estimulado a portabilidade dos frameworks tanto para o LESS quanto para o Sass, permitindo ao desenvolvedor decidir qual deseja utilizar em seus projetos e o qual a necessidade para esse uso.

Além das funcionalidades disponíveis e apresentadas esses dois pré-processadores possuem vários outros componentes, mas um dos fatores que se destaca no momento da análise para escolha é a sintaxe de cada um deles. Enquanto o Sass baseia-se na indentação o LESS permanece de forma simples como o CSS e isso permite a migração de arquivos já existentes de forma gradual enquanto no Sass é preciso converter completamente os arquivos.

Apesar do LESS ainda ser considerado o mais utilizado entre os pré-processadores, o crescimento do Sass é considerável como mostra a figura abaixo retirada do artigo de COYIER (2013) sobre as mudanças e implementações realizadas pelos desenvolvedores nesses pré-processadores.

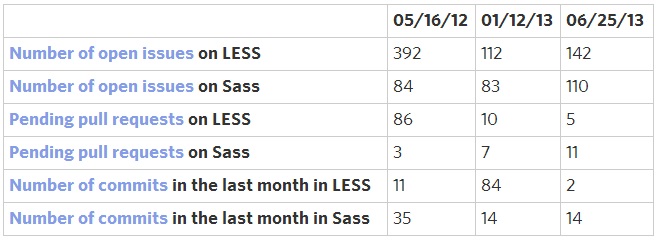


Figura 1 – Comparação no número de usuários sobre o LESS e o Sass

A partir disso, recomenda-se leitura e experimentação de cada um dos pré-processadores para definir qual deles se adequa a necessidade e ao projeto que esteja desenvolvendo.

rEFERENCIAS

COYIER, Chris. Sass vs. LESS. Disponível em: <http://css-tricks.com/sass-vs-less/>. Acesso em: out. 2013.

LESS. LESS: The Dynamic Stylesheet Language. Disponível em: <http://lesscss.org/>. Acesso em: out. 2013.

SASS. SASS: CSS With Superpowers. Disponível em: <http://sass-lang.com/>. Acesso em: out. 2013.

USABLI.CA. CSS Front-End Frameworks with Comparison. Disponível em: <http://usablica.github.io/front-end-frameworks/compare.html>. Acesso em: out. 2013.