



Regra de Estilo @import

O que faz a Regra de Estilo @import?

A regra de estilo @import do Sass é uma extensão da regra @import do CSS padrão. Como havemos de lembrar a regra @import do CSS padrão têm a missão de importar elementos de outras folhas de estilo para o nosso arquivo CSS. Mas o @import do Sass é mais evoluído que o @import do CSS, pois, enquanto o @import do CSS faz várias solicitações HTTP á medida que a nossa página vai sendo renderizada, o @import do Sass faz toda a importação das folhas de estilo durante a compilação, gera um único arquivo que combina todas as formatações advindas do @import e só depois renderiza a página.

Além disso, assim como o @import no CSS, o @import do Sass utiliza a sintaxe recuada permitindo que possamos escrever URLs sem o uso das aspas e somente o nome do arquivo. E diferente do @import do CSS, não precisamos de declarar um @import para cada URL, em vez disso podemos utilizar um único @import e importar os URL uma atrás da outra separadas pela vírgula.

IMPORTANTE!

A equipe Sass desencoraja o uso contínuo da regra @import, pois o Sass irá gradualmente eliminá-lo ao longo dos anos e eventualmente removê-lo inteiramente da linguagem. Em vez de usar o @import, prefira a regra @use que tudo o que o @import faz e ainda melhor.

Problemas que podem ser gerados ao usar @import!!

A regra @import torna todos os elementos importados acessíveis apenas globalmente, não podemos redefiní-las localmente como fazemos na regra @use, ficando difícil de saber á qual biblioteca um elemento pertence.

Outro problema é o inchaço no código final, já que o `@import` executa cada folha e estilo, aumentando o tempo de compilação.

Como utilizar `@import`?

Veja como utilizá-la da seguinte forma:

Primeiro temos que ter um módulo para ser importado:

Arquivo: `_list.scss`

```
@mixin list-reset {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
}
```

Criamos um mixin dentro de um módulo para resetar as margens, padding e estilo de uma lista;

Importante o módulo para um arquivo bootstrap com todos os imports necessários:

Arquivo `bootstrap.scss`:

```
@import "foundation/list";  
  
li {  
    @include list-reset;  
    line-height: 0;  
}
```

Perceba que no arquivo bootstrap foi carregado só o módulo `"_list.scss"`, o arquivo bootstrap serve só para armazenar os módulos que serão usados. No exemplo acima temos apenas um módulo, mas geralmente utilizamos "n" módulos dentro de um arquivo bootstrap;

Importante o arquivo bootstrap no arquivo Sass com `@import`:

Arquivo `style.scss`:

```
@import "foundation/bootstrap";
```

Perceba que, simplesmente importamos o arquivo, não é possível fazer qualquer mudança local no arquivo;

Resultado no arquivo final:

Arquivo Final:

```
.li {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    line-height: 0;  
}
```

Migrando @import para o @use!!!

O novo sistema de módulo e a regra @use tratam de todos os problemas que o @import têm e é possível migrar a formatação de @import para @use usando uma ferramenta de migração do Sass, que é o **sass-migrator**. Basta apontá-lo para os nossos arquivos de entrada e deixa-lo funcionar.

Antes devemos instalar o **sass-migrator**, podemos instalá-lo usando o comando:

```
npm install -g sass-migrator
```

Por padrão o sass-migrator só mudará os arquivos que você passar explicitamente na linha de comando, por através do comando:

```
sass-migrator nomeDoModulo --verbose NomeDoArquivo.scss
```

Se você quiser fazer um teste para ver que mudanças serão feitas sem realmente salvá-las, você pode usar:

```
sass-migrator nomeDoModulo --dry-run --verbose (ou -nv para abreviar)
```

Podemos também mudar os demais arquivos carregados junto com o arquivo que desejamos fazer a migração mesmo que os demais arquivos anexados a ele estejam usando @use, @forward ou @import, para isso usamos o comando:

```
sass-migrator nomeDoModulo --verbose --migrate-deps NomeDoArquivo.scss
```