



Regras de Estilo @each

O que faz a Regra de Estilo @each?

A regra de estilo @each gera um laço de repetição para “cada” valor de uma list ou map. Para isso temos que gerar dentro da regra @each uma variável comparadora que vai executar um determinado comando a cada valor índice dentro da lista ou map. E dentro do bloco de execução da regra nós colocamos o que cada valor da list ou map deve fazer a cada laço de repetição.

Essa regra é muito útil para a criação de formatações que sigam o mesmo estilo, porém com valores diferentes, como por exemplo tamanhos de imagem, efeitos gradiente e coisas desse tipo.

OBS: O que são lists e maps? List é quando possuímos uma variável no Sass que possui uma porção de valores divididos por vírgula, é como um array, por exemplo: `$var: 10px, 30px, 50px;` E maps é quando temos uma lista, porém referenciada pelo conjunto chave e valor, onde um valor deverá ser chamado por através da sua chave, dessa forma: `$var: (“xs”:10px, “sm”:30px, “xl”:50px);` Perceba que no map temos um nome para cada valor, esse nome corresponde a sua “chave” para referenciá-lo, e em seguida temos o valor dele. Detalhe importante!!! Nos maps os conjuntos chave-valor devem ficar dentro dos parênteses.

Como utilizar a Regra @each?

```
$sizes: 40px, 50px, 80px;

@each $size in $sizes {
  .icon-#{ $size } {
    font-size: $size;
    height: $size;
    width: $size;
  }
}
```

```
.icon-40px {
  font-size: 40px;
  height: 40px;
  width: 40px;
}

.icon-50px {
  font-size: 50px;
  height: 50px;
  width: 50px;
}

.icon-80px {
  font-size: 80px;
  height: 80px;
  width: 80px;
}
```

Perceba que criamos várias formatações de estilo para tamanhos de fonte diferentes a serem usadas no nosso documento. Fizemos isso automaticamente apenas passando os valores das fontes dentro de uma lista na variável “\$sizes”, e a regra @each fez todo o trabalho grosso para cada uma delas.

Para isso a regra @each usou uma variável comparativa a variável \$size, que a cada laço seria comparada com os valores dentro da lista em \$sizes, gerando os nomes dos ícones – com a ajuda de uma interpolação (template string) – e além disso atribuindo valores as propriedades dentro da regra.

OBS: Caso estivéssemos usando um map neste exemplo, a nossa variável comparativa deveria possuir dois valores separados por vírgula, o 1º valor seria para a chave e o 2º valor seria para o valor da chave. Geralmente precisamos de usar template string para a geração dos nomes dos seletores.