Regra de Estilo @function

O que faz a Regra de Estilo @function?

A regra de estilo @function do Sass é usada para trabalharmos com funções, ou seja, trechos de código que utilizam valores como: number, string, colors, lista de valores e etc, para gerar operações que resultem na saída de um determinado valor que desejamos utilizar no nosso código Sass.

As funções são muito úteis para gerar abstrações complexas de serem implementadas, fazendo isso de uma maneira simples e tornando o código muito mais legível.

Como utilizar @function?

Para criar uma função devemos usar a regra @function, seguida pelo nome da função e os parênteses que receberão seus parâmetros, á partir daí damos início a função por através dos colchetes, onde colocamos as operações que irão acontecer sequencialmente. E ao final temos que usar a regra @return dentro da função para que ela retorne um resultado ao final da sua execução. Depois, para utilizar a função, basta chamarmos por ela normalmente, assim como chamamos as funções built-ins do CSS, desde que o arquivo módulo da função já tenha sido importado no documento atual que estivermos mexendo.

Fazemos isso da seguinte forma:

Primeiro criamos uma função dentro de um módulo:

Arquivo: _calc.scss

```
@function sum($numbers...) {
        $sum: 0;
        @each $number in %numbers {
             $sum: %sum + %number;
  }
  @return $sum;
}
```

Acima temos uma função que soma o total de valores que forem passados a ela separados por vírgula.
Perceba que, após a regra @function demos um nome á função, o nome “sum”, após o nome temos os parênteses, que resguardam os parâmetros que a função deverá receber, esses parâmetros serão alocados na variável $numbers que é um array que vai receber uma sequência de valores. E dentro da função nós declaramos uma variável chamada $sum, essa variável será usada futuramente na regra @return para apresentar o valor da somatória ao usuário.

A seguir, temos uma regra dentro da função, que é a regra @each – ou “cada” – essa regra faz com que, para cada valor índice dentro do array em $numbers, ela calcule esse valor somado ao total que estiver em $sum, ou seja, está fazendo a soma de todos os valores passados.

Ao final da função, temos uma regra @return, essa regra é obrigatória para que possamos entregar um resultado ao usuário final.

Importando o módulo para um arquivo scss:

Arquivo style.scss:

```
@use “calc”;

.width-total {
        Width: calc.sum(50px, 30px, 100px);
}
```

Veja que importamos o módulo “calc.scss” dentro do nosso Sass gerador do nosso css final. Lá dentro do arquivo Sass nós chamamos a função “sum” dentro do seletor “width-total” para calcular a largura total de um elemento segundo 3 parâmetros passados;

Resultado no arquivo final:

Arquivo Final:

```
.width-total {
        width: 180px;
}
```

Perceba que no arquivo final, o que aparecerá para ele é o valor já calculado e passado por através da regra @return dentro da própria função “sum”.

Cuidados ao nomear funções!!!

Quando damos nomes as nossa funções devemos tomar o cuidado de não escolher nomes que comecem com números, caracteres especiais ou que tenham o mesmo nome que uma função built-in do Sass, pois, podemos acabar usando um seletor universal para renomear nossos módulos e acabar designando uma função com o mesmo nome de uma função built-in, o que poderia gerar um conflito.

Além disso devemos tomar cuidado com o uso de hífen (-) e underline (_), afinal ambas são entendidas da mesma forma pelo Sass, então se tivermos uma função cujo nome é: “scale-color” e outra “scale_color”, para o Sass elas serão a mesma função.

Recomendação importante!!!

Embora alguns se sintam tentados a usar funções para definir formatação sobre um trecho de código, essa prática é fortemente desencorajada, pois, o papel principal das funções é gerar cálculos e apresentar resultados calculados, enquanto a definição de formatações é um trabalho dos mixins.

Por isso, é bom deixarmos cada coisa no seu lugar, @function deve ser usado somente para fazer cálculos e @mixin para definição de efeitos sobre os nossos códigos.

A Regra @return:

A regra @return indica o valor a ser usado como resultado da chamada de uma função. Só é permitido dentro de um corpo de @function, e cada @function deve terminar com um @return.

Quando um @return é encontrado, ele termina imediatamente a função e retorna o resultado para o código solicitante.

Podemos até utilizar mais de @return em casos especiais, como em casos de desvio condicional com @else por exemplo, porém o resultado sempre vai sair de um único @return.