

TIN Responsive Web Design

Prof. Alexandre Carlos profalexandre.jesus@fiap.com.br



CSS with superpowers





SASS = Syntactically Awesome StyleSheets

Sass é um préprocessador.

Mas o que é um **pré-processador CSS**?

Um **pré-processador CSS** é um programa que permite você gerar **CSS** a partir de uma sintaxe (en-US) única desse **pré-processador**.

Essas funcionalidades fazem a estrutura do **CSS** mais legível e fácil de manter, além de que permite que você use variáveis, regras aninhadas, mixins, importações e mais!

 \cdot \cdot \cdot + \cdot \cdot

https://sass-lang.com/



@IMPORT/PARTIALS

 Nós já vimos os a parte de organizar os arquivos em pastas, agora vamos ver como importar arquivos dentro do sass e também poder criar partials, uma espécie de arquivo que só é compilado se for importado.

main.scss X Ex: css > 8 main.scss > \$back-color: □#e20775; C FI D ✓ SASS body { Utilize aspas parao importe deste arquivo e background-color: \$back-color; não colo a extensão e nem a @import "base"; < base.scss sublinha(underscore), apenas o nome. Nunca main.css se esqueca do pont e virgula (;) main.css.map main.scss css > 8 _base.scss > 4 .container .container { index html \$text-color: □#00ff00; Quando criamos arquivos partials a idéia principal é organizer color: \$text-color; todo o trabalho em componentes. Ou seja, criar CSS para partes do projeto. Estes arquivos devem possuir a color: □ #0000ff; extensão scss e iniciar o nome com uma (), sublinha ou underscore. Assim eles serão ignorados pelo compilador e poderão ser importados tranquilamente. https://sass-lang.com/



@IMPORT/PARTIALS

- A utilização dos import e partials é muito importante nos dias de hoje, pois com isso conseguimos tornar nosso código muito mais organizado.
- Ex:

```
main.css X
css > 3 main.css > 4 body
   1 \square body {
         background-color: □#e20775;
      v body .container
         color: □#00ff00;
      v body .container p {
         color: □#0000ff;
  10
  11
       /*# sourceMappingURL=main.css.map */
```



@MIXINS / @INCLUDE

- Mixins são agrupamento de declarações que podem ser reusadas.
- Imagine que você criou um código CSS que se repete e se repete, porém tem algo que mude nesse código. Você pode utilizar o mixins com include para solucionar este problema.
- Criei um mixin de um box-shadow passando uma cor como parâmetro, vou utilizar demtro do arquivo importado base.

- Ex:



@MIXINS / @INCLUDE

https://sass-lang.com/

Utilizando o include posso chamar o mixin que foi criado no arquivo principal e

passar o parâmetro solicitado.

```
Ex:
       base.scss > ...
           .container
              @include box-shadow(\square#0000ff);
               $text-color: □#00ff00;
              color: $text-color;
              width: 200px;
              height: 200px;
              background-color: #fff;
                   background-color: ☐#ff0000;
                  height: 100px;
                   width: 100px;
                   color: □ #0000ff;
                   @include box-shadow(\square#0000ff);
           Podemos utilizar quantas vezes quisermos.
```

```
INICIO COM SASS

EXEMPLO ESCOPO
Somenthing
```

```
index.html>_
index.html>_
thml lang="pt-br">
chtml lang="pt-b
```



@IF/@ELSE IF/@ELSE

INICIO COM SASS

 Como toda boa linguagem de programação que o sass não é ele também possui estruturas de decisão if/else if e else. Onde podemos realizar as mais varidas tarefas.

- Ex:

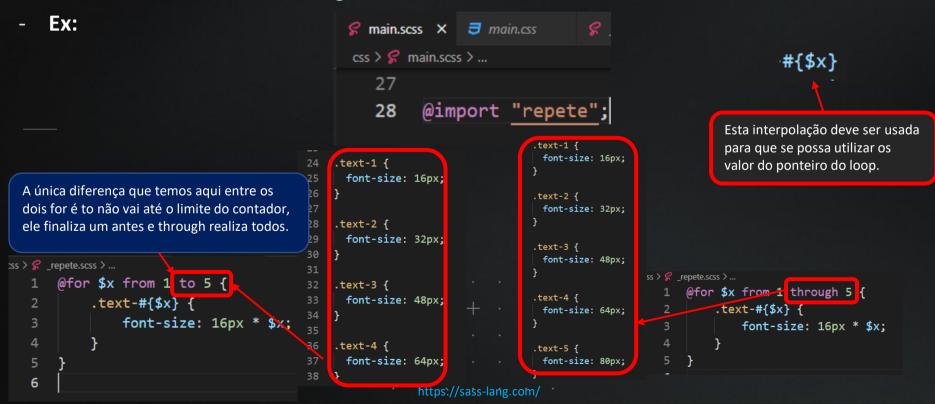
```
base.scss > ...
    $back-color: □#e20775:
                                                 .container {
                                                    @include box-shadow( #0000ff);
                                                    $text-color: □ #00ff00
    @mixin box-shadow($color) {
                                                    color: $text-color:
        box-shadow: 5px 5px 5px 5px $color;
                                                    width: 200px;
                                                    height: 200px:
                                                    background-color: #fff:
    @mixin make-bold($bool) {
        @if $bool == true {
            font-weight: bold:
                                                        background-color: 
    #1c1c1c;

                                                        height: 100px;
                                                         width: 100px:
                                                        color: □ #0000ff;
                                                        @include box-shadow( #0000ff):
    @mixin text-effect($val) {
                                                                                                1 !Doctype html>
        @if $val == danger {
                                                        @include text-effect(danger);
                                                                                                  <html lang="pt-br">
            color: #ff0000:
        } @else if $val == alert {
                                                              Realizamos um mixin recebendo uma
                                                                                                                         content="width=device-width, initial-scale=1.0">
            color: #fff000:
                                                                                                                        " href="./css/main.css">
                                                              variavel com um texto para ser avaliado.
        } @else {
            color: \( \precent{\pi} #000000; \)
                                                              Obs: Quando trabalhamos com texto no sass.
                                                              não utilizamos ASPAS.
                                                                                                                        SASS</h1>
                                                              O bloco padrão de validação por
    body {
                                                                                                                        ass="container">
                                                              If / else if /else
        background-color: $back-color;
                                                                                                                        EXEMPLO ESCOPO
                                                                                                                         Somenthing
        @import "base";
25
                                                                  https://sass-lang.com/
```



@FOR / [TO ,THROUGH]

 E não poderíamos deixar de ver os laços de repetição onde podemos aplicar dinamicidade em nossos códigos.





@FOREACH/@KEY

 Podemos criar listas aqui no sass o que pode nos auxiliar na hora de criar uma palleta de cores por exemplo.

- **Ex:**

Aqui estamos utilizando chave para cria a lista

```
$colors: ( blue, red, red, green, yellow);

@each $color in $colors {
    .#{$color}-text {
        color: $color;
    }
}
```

https://sass-lang.com/



CHEKPOINT 2 - UTILIZAÇÃO DE SASS

 Para amplificar nossos conhecimentos dentro dessa tecnologia vamos aplicar os conhecimentos obtidos aqui no nosso projeto piloto [O SITE].

Aplicar:

- @import/partials.
- @mixins
- @if/@else if /@else

Repetições e listas se houver a necessidade, será um plus no projeto.

Possuimos no projeto O SITE 5 seções distintas que podem ser estilizadas unicamente através das técnicas aprendidas.

https://sass-lang.com/



CHEKPOINT 2 - UTILIZAÇÃO DE SASS

ORIENTAÇÕES

- 1 Número de participantes: Grupo do Challenger
- 2 Componentizar em estilos 5 seções e ou áreas distintas do projeto [O SITE] no minímo.
- 3 Entrega:

- Deve ser gerada uma entrega deste projeto da seguinte forma:

- a) Entregar um PDF e um arquivo compactado no TEAMs contendo os nomes e RM dos integrantes mais o link do respositório e o projeto respectivamente.
- b) O projeto deve ser versionado com o Git e sincronizado com o Github em um repositório privado e ter commits relativos as alterações realizadas, onde todos os participantes devem ter commits. Evitem commits muito grandes. No mínimo 20 commits.
- c) Este repositório deve ter acesso ao usuário alecarlosjesus.
- d A entrega deve ser feita até o dia 07/10/2021 as 23:55 no TEAMs e no Github.

5 - Critérios:

- a) Aplicação dos conceitos do SASS. (2 Pontos)
- b) Aplicação de componentização da estilização por área ou seção do projeto (4 Pontos)
- c) Versionamento (Utilização do Git/GitHub) (2 Pontos)
- d) Semântica (1 Ponto)
- e) Boas Práticas (1 Ponto)

Copyright © 2015 - 2021 Prof. Alexandre Carlos de Jesus

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proíbido sem o consentimento formal, por escrito, do Professor (autor).