INSTITUTO FEDERAL
Santa Catarina

Programação para Internet I

INSTITUTO FEDERAL
Santa Catarina

Tópicos desta aula

SASS - Syntactically Awesome StyleSheets

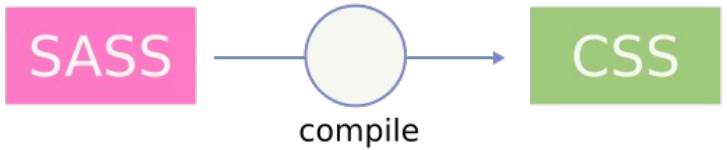
1) Introdução
2) Utilização
3) Variáveis, Funções e Listas
4) Programação no CSS
5) Mixins
6) Extends

INSTITUTO FEDERAL
Santa Catarina

Introdução

Introdução ao SASS

- SASS é um dos pré-processadores para CSS mais utilizados. A ideia é adicionar recursos especiais ao CSS como variáveis, operações e outras opções variadas.
- É considerado por seus autores uma extensão do CSS3.
- Existem vários outros como Stylus, Less e PostCSS.
- O SASS é utilizado por grandes frameworks como o Bootstrap.



Introdução ao SASS

- Existem duas sintaxes, uma mais antiga que utilizava a extensão “.sass” e outra atual com a extensão “.scss”.
- A versão SCSS é mais parecida com o CSS.
- Embora ainda seja possível desenvolver na sintaxe antiga, manteremos o foco na sintaxe SCSS.

.SCSS - nova

```scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;
body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

.SASS - antiga

```sass
$font-stack: Helvetica, sans-serif
$primary-color: #333
body
  font: 100% $font-stack
  color: $primary-color
```

INSTITUTO FEDERAL
Santa Catarina

Introdução ao SASS

- Para que nosso código Sass seja transformado em CSS, precisamos configurar nosso ambiente de desenvolvimento.
- Uma forma simples é utilizar um programa, como o Koala disponível para Windows, Mac e Linux
- Outra opção é o wrapper do próprio SASS feito em Dart. A vantagem de utilizar a lib do próprio SASS é a possibilidade de integrar a automatizadores de tarefas já existentes no seu projeto como o GULP para compilação automática do CSS durante o desenvolvimento.
- Também existem compiladores SASS online.
- Por fim, é possível instalar extensões no VSCode como o “Live Sass Compiler” que também automatiza o processo.

INSTITUTO FEDERAL
Santa Catarina

Utilização

Introdução ao SASS

- É normal precisarmos fazer formatações específicas de elementos filhos ou pseudo-classes de um determinado container.
- Neste exemplo, podemos perceber a “repetição” de seletores em um arquivo CSS.

```css
.section-1 {
  background-color: black;
}

.section-1 h1 {
  color: white;
}

.section-1 p {
  color: #CCC;
}

.section-1:hover {
  background-color: darkblue;
}
```

Introdução ao SASS

```
.section-1{
    background-color: black;
    h1{
        color:white
    }
    p{
        color: #CCC;
    }
    &:hover{
        background-color: darkblue;
    }
}
```

- No SASS, podemos seguir a seguinte sintaxe de aninhamento, deixando o código mais elegante.
- O símbolo “&” indica que o seletor não é descendente.
- Este arquivo é criado exatamente como o CSS, porém com extensão “**.scss**”.

Introdução ao SASS

```
.section-1{
    background-color: black;
    h1{
        color:white
    }
    p{
        color: #CCC;
    }
    &:hover{
        background-color: darkblue;
    }
}
```

```
.section-1 {
  background-color: black;
}

.section-1 h1 {
  color: white;
}

.section-1 p {
  color: #CCC;
}

.section-1:hover {
  background-color: darkblue;
}
```

Introdução ao SASS

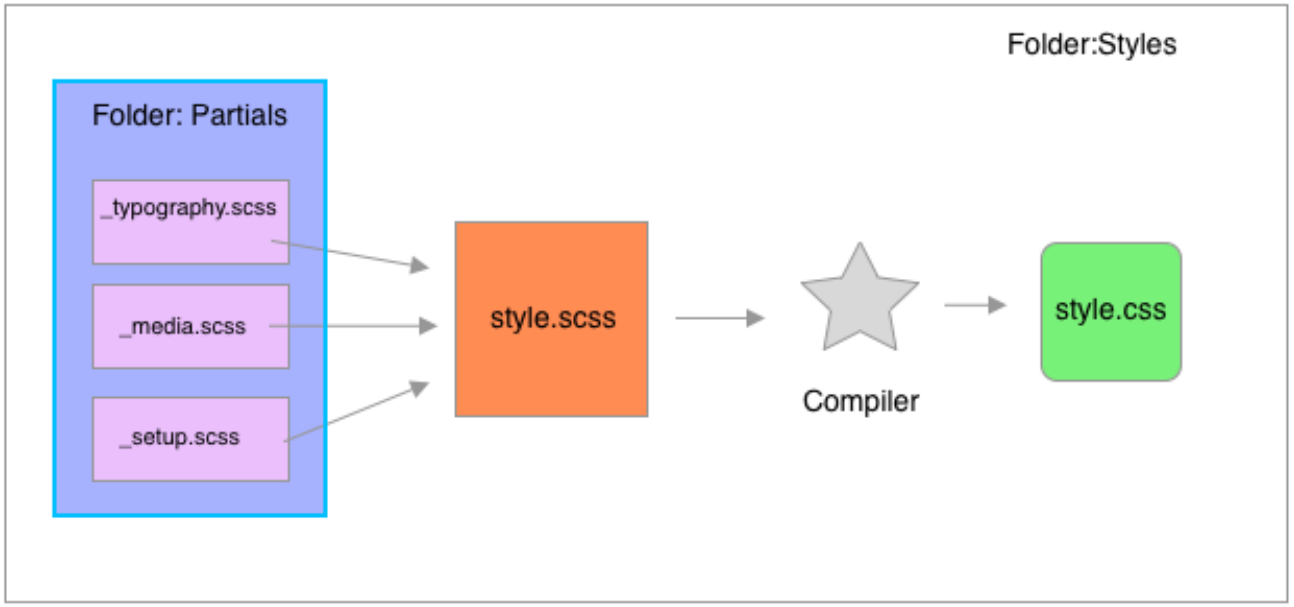
```
.section-1{
    background-color: black;
    // propriedade border
    border:{
        top:5px solid blue;
        bottom:5px solid red;
    }
}
```

- Também é possível fazer este agrupamento para propriedades CSS.
- Neste caso, é importante utilizar um “:” do lado do nome da propriedade.
- No Sass podemos fazer comentários de linha única.

Introdução ao SASS

```
.section-1{
    background-color: black;
   // propriedade border
    border:{
        top:5px solid blue;
        bottom:5px solid red;
    }
}
```

```
.section-1 {
  background-color: black;
  border-top: 5px solid blue;
  border-bottom: 5px solid red;
}
```



- Você pode decompor seu código Sass em diversos arquivos, para fins de organização, e no final gerar apenas um CSS.

Introdução ao SASS

```
@import “formularios”

.section-1{
    background-color: black;
    // propriedade border
    border:{
        top:5px solid blue;
        bottom:5px solid red;
    }
}
```

- Você pode criar diversos arquivos Sass, para fins de organização.
- Considere que você tenha feito outro arquivo “formularios.scss”
- Ao importar este arquivo (sem extensão), você gerará um arquivo CSS único, ganhando performance.

Introdução ao SASS

```
@import "_formularios"

.section-1{
    background-color: black;
    // propriedade border
    border:{
        top:5px solid blue;
        bottom:5px solid red;
    }
}
```

- Caso adicione um **underline** antes do nome do arquivo, o Sass saberá que seu arquivo é um "*partial*" e não criará o correspondente e desnecessário "formularios.css"

INSTITUTO FEDERAL
Santa Catarina

Variáveis, Funções e Listas

Trabalhando com variáveis

```
$fundo-escuro: black;
.section-1{
    background-color: $fundo-escuro;
    $texto-claro: #CCC;
    p{
        color: $texto-claro;
    }
}
.section-3{
    background-color: $fundo-escuro;
}
```

- No SASS, podemos criar variáveis globais e locais utilizando o prefixo $
- $fundo-escuro é uma variável global
- $texto-claro é uma variável que só funciona no escopo de .section-1

Trabalhando com variáveis

```scss
$fundo-escuro: black;
.section-1{
    background-color: $fundo-escuro;
    $texto-claro: #CCC;
    p{
        color: $texto-claro;
    }
}
.section-3{
    background-color: $fundo-escuro;
}
```

```css
.section-1 {
  background-color: black;
}

.section-1 p {
  color: #CCC;
}

.section-3 {
  background-color: black;
}
```

INSTITUTO FEDERAL
Santa Catarina

Funções nativas Scss

- Assim como vimos no CSS nativo, o Scss possui diversas funções implementadas.
- Podemos consultar a referência completa em https://sass-lang.com/documentation/modules

Funções Scss - – análise de exemplo

```scss
// misturando cores
p{
    color:mix(red, blue);
}

// 70% de preto e 30% de verde
h2{
    color:mix(black, green, 70%);
}

// escurecer o amarelo
h3{
    color:darken(yellow, 20%);
}
```

```css
p {
  color: purple;
}

h2 {
  color: #002600;
}

h3 {
  color: #999900;
}
```

Lists e Maps – análise do exemplo

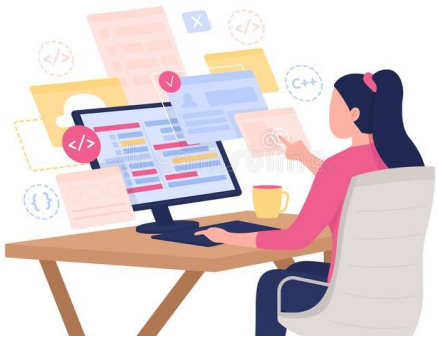
-webkit para o Chrome e o Safari, -moz para o Firefox, -o para o Opera, -ms para o Internet Explorer)

```
$tamanhos: 300px 650px 90%;
$cores:(
    titulo:white,
    paragrafo:yellow,
    links:orange
);

.container{
    // o primeiro elemento é o 1
    width:nth($tamanhos,1);
}
a:any-link{
    color:map-get($cores, links);
}
```

```
.container {
  width: 300px;
}

a:-webkit-any-link {
  color: orange;
}

a:-moz-any-link {
  color: orange;
}

a:any-link {
  color: orange;
}
```

INSTITUTO FEDERAL
Santa Catarina

Desafio de código 1



Desafio de código

```
.nav {
  background-color: black;
  color: white;
}

.nav a:link, .nav a:visited {
  text-decoration: none;
  color: yellow;
}

.nav a:hover {
  color: lime;
}
```

- Faça um código Sass que formate um menu de navegação <nav> e seus respectivos links em seus estados.
- As cores devem estar em variáveis ou listas Sass.

INSTITUTO FEDERAL
Santa Catarina

Programação

Interpolação

```
$a:50;

.class-#{$a}{
    color:red;
}
```


```
.class-50 {
  color: red;
}
```

- A interpolação pode ser usada em quase qualquer lugar em uma folha de estilo Sass para incorporar o resultado de uma expressão SassScript em um pedaço de CSS. Basta envolver uma expressão em #{}
- Veremos como a interpolação pode ser útil um pouco mais adiante.

Controle de fluxo

```
$test: 12;

p {
  @if $test < 3 {
      text-color: red;
    } @else if $test == 3 {
      text-color: blue;
    } @else {
      text-color: white;
    }
}
```

- No SASS, podemos controles de fluxo com @if @else exatamente como linguagens de programação.

Laços de repetição

```
$sections:(1:red, 2:blue, 3:green);

@for $i from 1 to 4 {
    .section-#{$i}{
        background-color:map-get($sections , $i );
    }
}

//ou

@for $i from 1 through  3 {
    .section-#{$i}{
        background-color:map-get($sections , $i );
    }
}
```

- Também podemos trabalhar com laços de repetição @for, @while e @ each.
- Neste exemplo estamos combinando mapas, interpolação e laço de repetição.

Laços de repetição - @for

```
$sections:(1:red, 2:blue, 3:green);

@for $i from 1 to 4 {
    .section-#{$i}{
        background-color:map-get($sections , $i );
    }
}
```

```
.section-1 {
  background-color: red;
}

.section-2 {
  background-color: blue;
}

.section-3 {
  background-color: green;
}
```

Laços de repetição - @each

```
$cores:red, blue, green;

@each $cor in $cores {
    .btn-#{$cor}{
        background-color:$cor
    }
}
```

```
.btn-red {
  background-color: red;
}

.btn-blue {
  background-color: blue;
}

.btn-green {
  background-color: green;
}
```

Laços de repetição - @while

```
$cores:red, blue, green;
// cuidado com o loop infinito
$i:1;
@while $i <= 3 {
    .class-#{$i}{
        background-color:nth($cores, $i);
    }
    $i:$i+1;
}
```

```
.class-1 {
  background-color: red;
}

.class-2 {
  background-color: blue;
}

.class-3 {
  background-color: green;
}
```

INSTITUTO FEDERAL
Santa Catarina

Criando funções

```
@function pow($base, $exponent) {
  $result: 1;
  @for $i from 1 through $exponent {
    $result: $result * $base;
  }
  @return $result;
}

.sidebar {
  float: left;
  margin-left: pow(4, 3) * 1px;
}
```

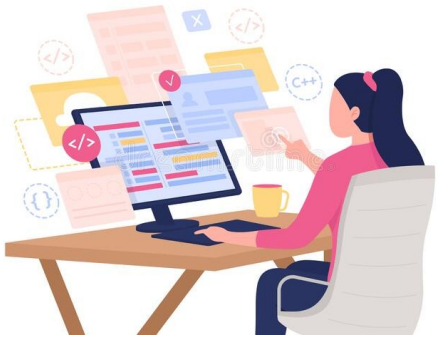
Sass

```
.sidebar {
  float: left;
  margin-left: 64px;
}
```

CSS

INSTITUTO FEDERAL
Santa Catarina

Desafio de código 2



Desafio de código

```
.titulo-1 {
  font-size: 29px;
}

.titulo-2 {
  font-size: 28px;
}

.titulo-3 {
  font-size: 27px;
}

.titulo-4 {
  font-size: 26px;
}
```

```
.titulo-5 {
  font-size: 25px;
}

.titulo-6 {
  font-size: 14px;
}

.titulo-7 {
  font-size: 13px;
}

.titulo-8 {
  font-size: 12px;
}
```

- Faça um código Sass que gere a saída CSS ao lado utilizando laço de repetição.

INSTITUTO FEDERAL
Santa Catarina

Mixins

INSTITUTO FEDERAL
Santa Catarina

O que são Mixins?

- Mixins permitem que você defina estilos que podem ser reutilizados em toda a sua folha de estilo.
- Mixins são considerados um dos elementos mais poderosos do Sass.
- São como funções, onde podemos passar ou não parâmetros, e eles nos retornam uma ou mais propriedades.

Mixin

```
@mixin estilo-tabela {
    border-spacing: 0;
    border-collapse: collapse;
    border:2px solid black;
    td{
        border:1px solid #CCC;
        padding: 10px;
    }
}
```

- Neste exemplo criamos um @mixin para definir um estilo para tabelas e suas respectivas células.
- Ao salvar, nada mudará no CSS pois @mixins pertencem somente ao Sass.

Mixin

```
@mixin estilo-tabela {
    border-spacing: 0;
    border-collapse: collapse;
    border:2px solid black;
    td{
        border:1px solid #CCC;
        padding: 10px;
    }
}
```

```
.tabela-media{
    width:75%;
    @include estilo-tabela();
}

.tabela-grande{
    width:100%;
    @include estilo-tabela();
}
```

- Para utilizar o @mixin, basta incluir no seu seletor com o comando @include.
- É uma boa prática fazer @mixin em arquivos separados.

Mixin

```
@mixin estilo-tabela($cor1:black, $cor2:#CCC) {
    border-spacing: 0;
    border-collapse: collapse;
    border:2px solid $cor1;
    td{
        border:1px solid $cor2;
        padding: 10px;
    }
}
```

- Um @mixin também pode receber parâmetros
- Estes parâmetros podem ter valores previamente definidos, caso seja omitido.

Mixin

```
@mixin estilo-tabela($cor1:black, $cor2:#CCC) {
    border-spacing: 0;
    border-collapse: collapse;
    border:2px solid $cor1;
    td{
        border:1px solid $cor2;
        padding: 10px;
    }
}
```

```
.tabela-media{
    width:75%;
    @include estilo-tabela(purple);
}

.tabela-grande{
    width:100%;
    @include estilo-tabela(black, red);
}
```

- Quais cores de borda a tabela média e a tabela grande terão?

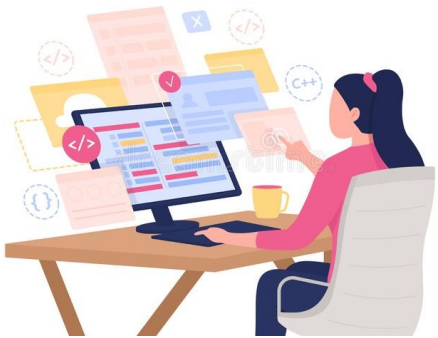
Mixin

```
@mixin links{
    text-decoration: none;
    color: black;
    @content;
}

a{
    @include links{
        &:hover{
            text-decoration: underline;
        }
    }
}
```

- Também é possível enviar blocos inteiros para o @mixin
- Para isto, adicione o @content no @mixin

INSTITUTO FEDERAL
Santa Catarina

Desafio de código 3



Desafio de código

```
div{
  width: 100px;
  height: 100px;
  background-color: blue;
}
```

- Faça um código Sass contendo um @mixin que recebe dois parâmetros: uma medida em px e uma cor.
- O objetivo é fazer que uma se torne um quadrado da medida e cor informadas.

INSTITUTO FEDERAL
Santa Catarina

Extends

Extend

```
.btn{
    min-width: 200px;
    padding: 20px;
}

.btn-confirm{
    border: 1px solid green;
    @extend .btn;
}
```

- Também é possível trabalhar @extend para herdar propriedades de outras classes.
- O @extend deixa o código CSS resultante menor que @mixin pois evita repetições de propriedades

Extend

```
.btn{
    min-width: 200px;
    padding: 20px;
}

.btn-confirm{
    border: 1px solid green;
    @extend .btn;
}
```


```
.btn, .btn-confirm {
  min-width: 200px;
  padding: 20px;
}

.btn-confirm {
  border: 1px solid green;
}
```

INSTITUTO FEDERAL
Santa Catarina

Projeto

Projeto com Sass

- Utilizando Sass, formate seu projeto “Formulários – Envio de currículo”