***Apostila de Sass ( Syntactically Awesome Stylesheets )***

***Introdução ao Sass:***

**Sass ou ( Syntactically Awesome Stylesheets )**,em português Folhas de Estilo Sinteticamente Incríveis, é um pré-processador CSS, ou mais conhecido como Super CSS.

Você criará o código Sass, e após criado, haverá uma compilação para o CSS padrão. Podendo nesse arquivo criar variáveis, funções, mixins, loops, condicionais, aninhamento de seletores e herança entre estilos.

O Sass tratá diversas vantagens para seu código, incluindo uma melhor organização, redução de repetição de código, e modularidade, podendo gerar maior produtivade, facilidade de manutenção, e melhor adaptação para projetos com estruturas mais robustas e complexas.

***Começando a usar o Sass:***

Para começarmos criaremos um arquivo com a extensão “**.scss”**, e neste arquivo faremos toda a nosso código Sass.

E para começarmos definimos na primeira linha do arquivo o conjunto padrão de caractéres, e o arquivo de cores que pode ser usado futuramente nas funções nativas do Sass, explicaremos futuramente…

**@charset “UTF-8”;**

**@use “sass:color”;**

Acrentamos os blocos…

**table {**

**margin: 0 auto;**

**border-collapse: collapse;**

**}**

**td, th {**

**border: 1px solid black;**

**padding: 5px 20px;**

**text-align: center;**

**}**

Agora é só compilar. E para isso utilizaremos o seguinte comando…

**sass** caminho\_do\_arquivo\_sass **:** caminho\_onde\_o\_arquivo\_css\_será\_gerado

Para uma melhor experiência é possível deixar o sass assistindo as mudanças, caso o desevolvimento esteja sendo feito em tempo real. Para isso usamos o seguinte comando...

**sass --watch** caminho\_do\_arquivo\_sass **:** caminho\_onde\_o\_arquivo\_css\_será\_gerado

Agora as mudanças serão feitas automaticamente enquanto o arquivo **scss** é modificado.

***Hover:***

No Sass podemos acrecentar o hover dos elementos diretamente no bloco, como nesse exemplo…

**td, th {**

**border: 1px solid black;**

**padding: 5px 20px;**

**text-align: center;**

**&:hover {**

**cursor: pointer;**

**background-color: lightgreen;**

**}**

**}**

***Herança:***

No Sass também podemos herdar as propriedades de elementos em outros elementos usando a diretiva **@extend**.

**.botao {**

**padding: 12px;**

**border-radius: 5px;**

**border: 1px solid rgba(0, 0, 0, 0.364);**

**margin: 10px;**

**color: rgb(23, 23, 23);**

**font: normal 1em cursive;**

**transition:**

**0.5s background-color ease-in-out,**

**transform 0.3s ease-in-out;**

**}**

**.botao\_laranja {**

**@extend .botao;**

**background-color: orange;**

**&:hover {**

**cursor: pointer;**

**background-color: rgb(249, 165, 86);**

**transform: scale(1.1);**

**}**

**}**

**.botao\_azul {**

**@extend .botao;**

**background-color: rgb(49, 49, 246);**

**&:hover {**

**cursor: pointer;**

**background-color: lightblue;**

**transform: scale(1.1);**

**}**

**}**

Lembrando que podemos criar isso em outro arquivo e importar, tornando o código mais limpo e organizado. E para isso usamos a seguinte diretiva…

**@use** nome\_do\_arquivo **as \*;**

***Variáveis:***

No Sass também podemos criar variáveis, declarando elas dessa maneira…

**$azul: rgb(49, 49, 246);**

**$laranja: orange;**

**$azul\_hover: lightblue;**

**$laranja\_hover: rgb(249, 165, 86);**

E fazer a chamada dessas da seguinte maneira…

**.botao\_laranja {**

**@extend .botao;**

**background-color: $laranja;**

**&:hover {**

**cursor: pointer;**

**background-color: $laranja\_hover;**

**transform: scale(1.1);**

**}**

**}**

**.botao\_azul {**

**@extend .botao;**

**background-color: $azul;**

**&:hover {**

**cursor: pointer;**

**background-color: $azul\_hover;**

**transform: scale(1.1);**

**}**

**}**

***Mixins:***

Uma melhor maneira de fazer a extensão de dados apenas em elementos, ao invés de precisar importar um arquivo por completo, podemos criar mixins, que são como funções que podem receber em si um argumentos como parâmetros. Podemos utilizamos por exemplo, da seguinte maneira…

**@mixin titulo($color) {**

**display: inline-block;**

**border: 1px solid rgba(0, 0, 0, 0.417);**

**border-radius: 5px;**

**margin: 10px;**

**padding: 10px;**

**color: $color;**

**}**

**.titulo\_preto {**

**@include titulo(black);**

**}**

**.titulo\_amarelo {**

**@include titulo(rgb(255, 255, 0));**

**}**

**.titulo\_verde {**

**@include titulo(green);**

**}**

**.titulo\_rosa {**

**@include titulo(pink);**

**}**

***Funções:***

O Sass em também tem funções, que funcionam basicamente da mesma maneira que as funções nas Linguagens de Progamação.

Por si só ele trás funções de cores, strings, números, listas e outros. Falaremos um pouco de cada uma delas abaixo…

***Cores:***

**lighten** **(** cor **,** porcentagem **)** **→** Clareia uma cor.

**darken (** cor **,** porcentagem **)** **→** Escurece uma cor.

**mix (** cor1 **,** cor2 **,** peso **)** **→** Mescla duas cores.

**rgba** **(** cor , alpha **)** **→** Aplica transparência à cor.

**adjust-hue (** cor **,** graus **)** **→** Altera o tom da cor.

**saturate** **(** cor **,** porcentagem **)** **→** Aumenta a saturação da cor.

**desaturate (** cor **,** porcentagem **)** **→** Reduz a saturação da cor.

**invert** **(** cor **)** **→** Inverte a cor.

***Números:***

**percentage** **(** valor **)** **→** Converte um número para porcentagem.

**round (** valor **)** **→** Arredonda o valor para o inteiro mais próximo.

**ceil** **(** valor **)** **→** Arredonda o valor para cima.

**floor (** valor **)** **→** Arredonda o valor para baixo.

**abs** **(** valor **)** **→** Retorna o valor absoluto (ex: abs(-5) → 5).

**min (** valor1 **,** valor2 **, … )** **→** Retorna o menor valor.

**max** **(** valor1 **,** valor2 , **…** **)** **→** Retorna o maior valor.

***Listas:***

**length** **(** lista **)** **→** Retorna o tamanho da lista.

**nth (** lista **,** $n **)** **→** Retorna o n-ésimo item da lista ( a contagem começa em 1 ).

**join** **(** lista **,** lista2 **,** separador **)** **→** Junta duas listas, usando o separador informado ( pode ser space ou comma ).

**append (** lista **,** valor **)** **→**  Adiciona um item ao final da lista.

***Outros:***

**if** **(** condição **,** verdadeiro **,** falso **)** **→** Condicional inline.

**unit** **(** 10px **)** **→** Retorna a unidade (ex: "px").

**unitless (** 10px **)** **→**  Verifica se não tem unidade ( retorna false nesse exemplo ).

**comparable** **(** 10px **,** 2em **)** → Verifica se as unidades podem ser comparadas ( retorna false nesse exemplo ).

**$azul: blue;**

**$azul\_hover: darken($azul, 20%);**

***Minificações (Minify):***

Percebemos que ao compilar o nosso arquivo Sass, é gerado um arquivo CSS padrão, onde cada bloco é gerado linha a linha… Para uma melhor experiência, é possível compilar e permitir que todos os blocos sejam declarados em uma linha só, deixando o arquivo extramemente mais leve. Para isso utilizaremos os seguintes parâmetros no comando de compilação…

**sass** caminho\_do\_arquivo\_sass **:** caminho\_do\_arquivo\_css –**style=compressed**