

Faça como eu fiz: Página “Sobre Nós”

Vamos utilizar as Regras de estilo e Sintaxe para criar uma página Sobre Nós no nosso site. A primeira coisa que vamos fazer é criar o arquivo para a página e linkar o nosso menu a este arquivo. No nosso VS Code vamos criar um arquivo chamado “about.html” e dentro dele vamos copiar a estrutura base do html de todo nosso site, todo o nosso index.html, e colar no arquivo about, e antes de salvar vamos retirar toda a tag `<main>`. Depois, vamos para o index.html e alterar o href do link Sobre nós, adicionando o about.html, e então fazer a mesma coisa no nosso novo arquivo. Ao salvarmos e irmos para o browser, podemos clicar no link Sobre nós, e ele vai redirecionar para nossa nova página ainda vazia, apenas com o header e o footer. Vamos criar o conteúdo dela. Voltando para o vscode, embaixo da tag header vamos criar uma tag article com a classe “about”. Dentro do article teremos a div “about__title” com um h1 Sobre Nós e mais uma div com o breadcrumb, que nada mais é que uma lista de links que redirecionam o usuário para as páginas navegadas anteriormente.

Primeiro temos um `<a>` que vai redirecionar de volta para a home, depois uma imagem que será o ícone de uma seta e mais um `<a>` que é a própria página Sobre Nós. Abaixo dessa div do título, teremos a div do conteúdo, primeiro com uma imagem principal e um texto de introdução. E depois mais uma div, chamada `about__split`, onde vamos dividir o conteúdo ao meio, tendo uma imagem menor do lado esquerdo e um texto do lado direito com 3 parágrafos. Ao irmos para o navegador e recarregarmos a página, todos os componentes estarão lá, porém, nada organizados. Por isso precisamos do nosso css, ou no nosso caso, Sass. Vamos no nosso arquivo `style.scss` e adicionar o seletor da classe `about`, dentro dele teremos um `width` de `80%`, o `font family` `Arial` e `margin: 0 auto 40px`. Para o título, vamos colocar um `background` cinza claro, alinhar o texto no centro e incluir um `padding` de `20px`. Para o H1 teremos um `font size` de `22px`, a cor será

o cinza escuro, padrão dos nossos títulos, o espaçamento entre as letras será de 1px, margin top de 5px, e margin bottom de 10px.

Nosso próximo passo é estilizar o breadcrumb, para isso vamos chamar a classe, definir um display flex e alinhar ao centro. Para o link, colocaremos um font size de 12px, tiraremos o estilo padrão do link, e a cor também será o cinza escuro. Vamos adicionar um hover roxo para o usuário saber que é um link ao passar o mouse em cima dos textos, e definir um tamanho fixo para o ícone da setinha, de 10px, e margin de 8px para o espaçamento entre os textos e a imagem. Fora do breadcrumb, a imagem principal da página terá width de 100%, um margin bottom de 20px e o texto de introdução terá fonte de 16px, cor cinza escuro, peso da fonte de 100, largura de 70%, sendo menor que a largura da imagem, alinhamento do texto no centro e margin top e bottom de 30px.

Vamos salvar e atualizar o navegador com tudo que fizemos até agora. Para definirmos a divisão da nossa div que possui imagem e texto, vamos usar o display flex, flex wrap e alinhamento ao centro na div principal. Antes de dividirmos a imagem e o texto, vamos setar o estilo dos parágrafos com padding 10px e 20px e tamanho da fonte de 14px. Agora sim, para a divisão da nossa div nós vamos usar a interpolação, mas antes de tudo vamos ver o que ela é. Interpolação é usada para injetar valores em expressões como variáveis e funções nos nossos seletores. Isso pode ser muito útil ao se passar parâmetros que podem ser alterados.

No caso da nossa divisão nós vamos usar um cálculo do css para a largura da div do nosso texto, vamos fazer sem a interpolação primeiro, definindo a largura da imagem 45%. A largura do texto será o resultado da largura total menos a largura da imagem, ficando “width: calc(100% - 45%);”, porém, como já vimos, é muito mais interessante usarmos variáveis em valores que podem ser alterados caso haja alguma necessidade, então vamos transformar os 45% em uma variável “\$img-width: 45%;”, substituir no estilo da imagem e usar a interpolação para

chamá-la na div do texto, ficando: “width: calc(100% - #{\$img-width});”. Ao recarregarmos o navegador podemos ver que o cálculo funcionou e conseguimos dividir a imagem de um lado e o texto do outro.