



Introdução a HTML e CSS

Sumário

[IHC00 - Hello, World](#)

[IHC01 - Conhecendo as tags](#)

[IHC02 - As estruturas de um site](#)

[IHC03 - Tipos de títulos](#)

[IHC04 - Parágrafos, negrito e quebra de linhas](#)

[IHC05 - Imagens](#)

[IHC06 - Lista sequencial](#)

[IHC07 - Links clicáveis](#)

[IHC08 - Vídeos](#)

[IHC09 - Curriculum bagaceira](#)

[IHC10 - Criando um formulário](#)

[IHC11 - Inputs](#)

[IHC12 - Labels](#)

[IHC13 - Textare](#)

[IHC14 - Checkboxes e Radio Buttons](#)

[IHC15 - Select](#)

[IHC16 - Botões](#)

[IHC17 - Testando o formulário](#)

[IHC18 - Google Fontes](#)

[IHC19 - Fontawesome](#)

[IHC20 - Primeiros passos no CSS](#)

[IHC21 - Conhecendo a Tag Div](#)

[IHC22 - Dando estilo a DIV](#)

[IHC23 - Atributo Class](#)

[IHC24 - Font-size e Inline-style](#)

[IHC25 - Salvando CSS separadamente](#)

[IHC26 - Criando o Index](#)

[IHC27 - Ajustes iniciais do cabeçalho](#)

[IHC28 - Aperfeiçoando o cabeçalho](#)

[IHC29 - Fechando o cabeçalho](#)

[IHC30 - Área de contato](#)

[IHC31 - Demais áreas](#)

[IHC32 - Ajuste no layout](#)

[IHC33 - Ajustes finos](#)

[IHC34 - Desafio do dia-a-dia](#)

[IHC35 - Ajustes iniciais](#)

[IHC36 - Lista horizontal](#)

[IHC37 - Título, botão e imagem](#)

[IHC38 - Área dos clientes](#)

[IHC39 - Meus serviços](#)

[IHC40 - Área do formulário](#)

[IHC41 - Portfólio](#)

[IHC42 - Ajustes finais](#)

[IHC43 - Hospedando um Site](#)



Exercícios:

- HTML https://www.w3schools.com/html/html_exercises.asp
- CSS https://www.w3schools.com/css/css_exercises.asp

IHC00 - Hello, World

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/060ffb26-da2c-4459-b959-01ea8b9a5fa1/primeiro_site.html

Antes de tudo, eu queria mostrar para vocês como os sites da internet são construídos. Então vamos abrir aqui o site da DNC, clicar com o botão direito e depois clicar em Exibir código fonte da página. Vocês podem repetir estes passos aí em qualquer site no navegador de vocês, não precisa instalar nem configurar nada.

O que vocês vão ver aqui é o código fonte do site que é composto por HTML, CSS e Javascript. Todas as páginas da internet são compostas por pelo menos uma dessas tecnologias, que é o HTML. O HTML é uma linguagem de marcação (e cuidado para não confundir com linguagem de programação) que é responsável por definir os elementos, ou seja, botões, formulários, textos, títulos, tabelas e etc, que vão fazer parte da sua página pra internet. Já o CSS e o Javascript são opcionais para potencializar o visual e a interatividade dessas páginas, mas de qualquer forma, o HTML é a base de todos os sites que vocês conhecem...

Sites como o Facebook, o Google, o da DNC ou qualquer outro site que vocês acessarem vai ter como base código fonte em HTML.

Por isso, como Product Manager - que é uma profissão que participa da criação de produtos para internet - você precisa ter uma noção básica das bases que compõem esses produtos e por isso a gente vai falar disso durante este módulo.

Vou começar então mostrando para vocês como se cria uma página em HTML. Não se preocupe por enquanto em entender ou decorar o que eu vou escrever, o intuito

aqui é só te mostrar como funciona, mas fique a vontade para tentar repetir os passos aí no seu computador e já criar o seu primeiro site.

Então vamos abrir aqui o bloco de notas e escrever o código do nosso site.

Vou abrir uma tag `<html>` e dentro dela duas tags, a tag `<head>` e a tag `<body>`. Dentro da tag `<head>`, eu vou criar uma outra tag chamada `<title>` e escrever o título do meu site, que será: Meu Primeiro Site.

Agora, dentro do `<body>` eu vou adicionar uma tag `<h1>` e em baixo, uma tag `<p>`. Entre as tags `<h1>` eu vou escrever Hello World e entre as tags `<p>` eu vou escrever Bem vindo ao meu primeiro site. Bom, esta pronta nossa primeira pagina só que, para que ela funcione e a gente consiga acessá-la em nosso navegador, a gente precisa salvar ela no formato HTML.

Para isso, você vai clicar em Arquivo, Salvar como, vai definir o tipo para Todos os arquivos e vai salvar este arquivo com o nome o nome que você quiser, mas vai botar no final do nome a extensão de arquivo .html. Agora a gente salva e clica para abrir o arquivo, e pronto!

A gente está vendo aqui o nosso primeiro site... É obvio que ele é muito simples, mas é a partir deste tipo de código - em html - que toda a internet é construída. Esse é o primeiro passo pra que vocês possam criar páginas de venda, landing pages, sites institucionais e até manipular ferramentas de construção de sites como o Wordpress.

Nesse exemplo aqui a gente usou o nosso próprio editor de textos que já vem instalado no computador, mas essa é a pior forma de escrever código. Para isso, existem ferramentas, que são os editores de textos, que são especialmente criadas para nos ajudar na tarefa de escrever código.

O editor de textos mais famoso e mais usado pelos programadores é o VS Code. Que é gratuito e você pode baixa-lo acessando o link code.visualstudio.com e clicando em Download. Quem quiser baixar e se aprofundar nisso, pode baixar, instalar e usar durante o curso, mas a gente tem uma outra opção que é mais rápida e prática que seria usar a versão online do VSCode. Pra isso, é só acessar vscode.dev e essa tela aqui vai aparecer.

Nas próximas aulas, a gente vai começar a se aprofundar nas tags do HTML, entendendo a estrutura, conhecendo algumas tags e mais importante, colocando a mão na massa. Então salve aí o link do VSCode Online ou baixe e instale, pra que você esteja preparado para a parte prática. Para fixar esse tipo de conteúdo, é extremamente importante que você pratique MESMO e por isso a gente vai focar bastante nisso.

IHC01 - Conhecendo as Tags



Referência de tags do HTML: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b8957246-6789-4afc-97ff-7ed149608f16/primeirosite.html>

Nessa aula, vocês vão começar a entender melhor sobre as estruturas do HTML e por a mão na massa criando seu próprio código.

O objetivo dessa aula é que vocês consigam criar uma página de currículum, como a seguir:

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/06c14eed-fd7e-417d-a2bf-c7e7b97524c8/cursoppm2-curriculum2.html.pdf>

Bom, o HTML, como já falamos na aula anterior, é uma linguagem de marcação. E novamente, Cuidado para não confundir com linguagem de programação, pois são coisas diferentes. Como linguagem de marcação, nós vamos usar marcadores para dizer como o conteúdo do nosso site deve ser exibido pelo navegador. Esses marcadores que a gente utiliza são chamados de TAGS.

A estrutura de uma tag é a seguinte: Símbolo menor que - Nome da Tag - Símbolo maior que

E aí a gente usa as tags pra dizer o que a gente quer que seja mostrado na página. Por exemplo:

- Tag <table> para criar tabelas

<table></table>

```
HTML <table>
1 <table border="1" cellpadding="10">
2   <tr>
3     <td>Produto</td>
4     <td>Preço</td>
5   </tr>
6   <tr>
7     <td>Caneta BIC</td>
8     <td>R$ 14,40</td>
9   </tr>
10 </table>
```



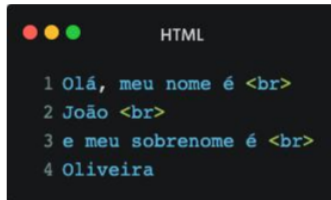
Produto	Preço
Caneta BIC	R\$ 14,40

- Tag para criar imagens


```
HTML
1 
```



- Tag
 para pular linha



```
HTML
1 Olá, meu nome é <br>
2 João <br>
3 e meu sobrenome é <br>
4 Oliveira
```



Olá, meu nome é
João
e meu sobrenome é
Oliveira

E por aí vai. Algumas tags são acompanhada de uma tag de fechamento, que é basicamente a repetição da primeira tag com a adição de uma barra depois do símbolo de menor que, da seguinte maneira:

```
<table></table>
<p></p>
<h1></h1>
```

Enquanto outras não precisam:

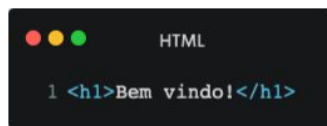
```
<br>
<img>
```

Em geral, pra facilitar a sua memorização, tenta pensar que as tags que demandam algum conteúdo dentro delas, vão precisar de fechamento, afinal, você precisa dizer onde começa e acaba aquela marcação. Já as tags que não demandam conteúdo entre elas, não vão precisar, como por exemplo a tag de quebrar linha. Ela simplesmente diz: quebre a linha aqui, e não necessita de nenhum conteúdo dentro.

Em HTML, absolutamente tudo que você quer criar deve ser feito com o uso de TAGs. Vamos supor que eu queira criar um título. A tag para criar títulos é a tag H1. Logo, o código seria:

```
<h1>Meu título</h1>
```

```
<h1></h1>
```



Bem vindo!

E dessa forma, a gente vai estruturando nosso código. Você pode até usar tags dentro de outras tags, como a gente fez na primeira aula, onde criamos uma tag `<html>` e dentro dela adicionamos a tag de cabeça `<head>` e a tag de corpo `<body>`.

Existem dezenas de tags no HTML. Até eu que trabalho com isso há anos as vezes esqueço de algumas e as vezes preciso consultar, então não se preocupa em decorar todas elas porque eventualmente você vai acabar memorizando as mais importantes sozinho enquanto você for usando aí no seu dia a dia.

E você pode a qualquer momento consulta-las no Google ou diretamente no link de referência que nós incluiremos aqui na plataforma.

IHC02 - As estruturas de um site



Referência de tags do HTML: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b8957246-6789-4afc-97ff-7ed149608f16/primeirosite.html>

Nosso objetivo hoje, é criar uma página de currículo. E pra fixar esse tipo de conteúdo, é indispensável que você pratique um pouquinho, então eu quero te convidar a me acompanhar, criando a sua página de currículo aí no seu computador.

Bom, na aula anterior, nós falamos sobre o VS Code, o editor de texto que nós usaremos durante o curso. Então a primeira coisa a fazer, é você criar uma pasta no seu computador com o nome que você quiser. No meu caso, vou chamar essa página de ppm. Criei a pasta no computador, vou acessar agora o link do nosso VS Code (que é [vscode.dev](https://code.visualstudio.com/)). Vou clicar em open folder e selecionar a minha pasta. Depois, vou criar um novo arquivo clicando aqui em New File. Clique em **select a language** e vamos digitar e selecionar HTML.

Nós vamos começar aqui criando uma tag chamada `<html>`. A tag `<html>` representa a raiz do nosso código em HTML, ela é a tag pai (ou mãe) de todas as outras tags que estão por vir. Essa hierarquia é importante no HTML, porque algumas tags (como a tag de listas que vocês verão mais pra frente) só funcionam se estiverem corretamente posicionadas dentro umas das outras.

Mas enfim, voltando ao assunto, criamos a nossa tag `html` que é a raiz do documento, e dentro dela vamos criar, lembrando sempre de abrir e fechar, mais duas tags: que são as tags `<head>` e a tag `<body>`, traduzindo, a cabeça e o corpo do nosso documento. E aqui, mais uma vez, é importante prestar a atenção na hierarquia porque existem algumas tags que a gente só vai usar dentro da tag `<head>` e outras que a gente só vai usar dentro da tag `<body>`.

Outra coisa importante é a organização. Se você sai digitando o código sem nenhuma formatação, vai ficar muito difícil entender e fazer alterações no futuro. Então é regra sempre dar um espaço usando a tecla TAB sempre que você for descer um nível na hierarquia. Por exemplo, a tag `HTML` é a primeira da hierarquia, então ela não tem nenhum espaço. Já as tags `head` e `body`, são irmãs porque estão no mesmo nível da hierarquia, e são filhas da tag `html`, logo, elas são as primeiras filhas e por isso só tem 1 espaço.

Agora, dentro da tag `<head>` vou criar uma tag chamada `<title>`, ou título em Português. Essa tag serve pra definir o título que aparece aqui na aba do navegador quando a gente acessa algum site. E aí mais uma vez, a tag `title` é filha da tag `head`,

logo, eu vou adicionar um espaço usando a tecla TAB antes dela. E dentro da tag `<title>` a gente define o título da nossa página, que vamos chamar de “Meu curriculum”

Vale ressaltar mais uma vez a questão da hierarquia, reparem que temos a tag `<html>` como pai, temos as tags `<head>` e `<body>` como filhos dela. E temos ainda a tag `<title>` como filho da tag `<head>`, beleza? Bastante atenção a isso.

Certo, vamos salvar essa página aqui no nosso computador clicando aqui no cantinho, em File, depois Save As. Vou escolher o nome curriculum.html. É importante terminar com .html para que a gente consiga abrir essa página no nosso navegador.

Depois que eu salvar, eu vou abrir a pasta onde eu salvei e clicar na página que criamos. Reparem que temos aqui uma página em branco, sem nenhum conteúdo, mas nosso título já está lá em cima! Agora, vamos começar a adicionar conteúdo aqui dentro dessa parte branca. Vou voltar aqui para o meu VS Code e começar adicionando um título.

IHC03 - Tipos de títulos



Referência de tags do HTML: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b8957246-6789-4afc-97ff-7ed149608f16/primeirosite.html>

No HTML, nós temos varias tags de título que são sempre a letra H acompanhada de um número que indica a importância desse titulo (e consequentemente, seu tamanho). Por exemplo, vamos começar adicionando uma tag `<h1>` e o título: Curriculum vitae.

Depois, em baixo da tag H1, vamos criar uma tag `<h2>` com o titulo Informações Pessoais. Depois, em baixo da tag `<h2>` eu vou criar uma tag `<h3>` com o título: Contatos. E outra com o título: Endereço.

Pra finalizar essa parte dos títulos, vamos criar mais uma tag <h2> com o título "Experiência Profissional" debaixo de todos os outros títulos. Agora vamos salvar e visualizar nosso documento, lembrando sempre de adicionar a extensão .html. Veja como os títulos aparecem.

Meu Curriculum

Informações Pessoais

Contatos

Endereço

Experiência Profissional

O H1, como o maior de todos. O <h2> reduz um pouco e o <h3> reduz mais ainda. Dessa forma, a gente consegue organizar nosso conteúdo através dos títulos, como se fosse um livro ou um artigo mesmo, sendo que o <h1> é o título principal e mais importante, e conforme o número vai aumentando o título vai se reduzindo, Sendo que o número 1 é o maior e o 6 é o menor e último que vocês podem usar.

Depois vocês vão ver que usando CSS, a gente consegue alterar isso, mudar estes tamanhos e as cores, mas isso é assunto pras próximas aulas.

IHC04 - Parágrafos, negrito e quebra de linhas




Referência de tags do HTML: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b8957246-6789-4afc-97ff-7ed149608f16/primeirosite.html>

Beleza, já aprendemos um pouco sobre os títulos e voltando agora pro editor, vamos falar sobre os parágrafos, que são representados pela tag `<p>`.

`<p></p>`



```
HTML
1 <p>Primeiro parágrafo</p>
2 <p>Segundo parágrafo</p>
```

Primeiro parágrafo

Segundo parágrafo

Então aqui, depois do título `<h1>` eu vou criar um parágrafo escrito: "Meu curriculum vitae em HTML."

Outra tag bastante usada é a tag que faz o texto entre ela ficar em negrito. Então eu vou vir aqui no parágrafo que a gente criou e adicionar a tag `` em volta do texto "curriculum vitae".

Ou seja, teremos a tag `` e entre ela o texto "curriculum vitae". Logo, o texto "tipos de animais" vai ficar em negrito. Vou salvar e abrir e vocês poderão ver isso aqui.

Outra tag bastante útil é a tag de quebrar linha. Podemos criar aqui um novo parágrafo com o texto: LEIA COM ATENÇÃO.

Vamos supor que eu quero que cada palavra desta frase fique em uma nova linha. Eu posso criar vários parágrafos, mas por padrão, o parágrafo tem um espaçamento maior entre as linhas. Pra criar quebras de linhas sem o espaçamento que existe entre os parágrafos, a gente usa a tag `
`. Essa tag tem uma característica diferente do que vimos até agora: ela não precisa de tag de fechamento. Então é só

abrir a tag
 onde você quer que a linha seja quebrada e Pronto! Vamos salvar e testar para conferir.

Por último, vamos falar sobre os comentários. Os comentários são elementos de texto que não podem ser vistos pelo usuário que abrir o seu site no navegador. Eles funcionam como espécies de anotações que os desenvolvedores fazem pra poderem se organizar melhor, por exemplo, se você quer deixar anotado que esse texto será temporário e deverá ser removido em breve. Ou se você quer se lembrar o porque de ter adicionado uma determinada TAG HTML. Ou até simplesmente se você quiser apenas deixar o seu documento HTML.

Pra usar comentários, a gente não usa TAGS, a gente escreve da seguinte maneira:

Símbolo menor que, exclamação, traço traço. Ai você digita seu comentário e fecha a tag com traço traço maior que.

Vou digitar aqui um comentário dizendo:

<!-- Esta é a minha primeira página em HTML -->

Esses caracteres que a gente digitou antes e depois do comentário, servem pro HTML identificar que aquilo é apenas um comentário.

Se a gente salvar e visualizar a nossa página no navegador, vai ver que os comentários não aparecem. Logo, a gente pode usar ele pra separar nosso código de uma forma mais organizada, já que apenas as pessoas que forem olhar o código fonte da sua página vão conseguir visualizar os comentários. Se eu clicar aqui com o botão direito e clicar em Visualizar código fonte, vocês verão que o comentário está aqui no código, o navegador só não os exibe.

IHC05 - Imagens

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8beb27d1-8eb0-4af0-9c95-db16070bcdac/primeirosite.html>

Na aula de hoje, nós vamos incrementar um pouco o curriculum que a gente começou. Antes de tudo, eu peguei uma imagem de perfil qualquer ([Face Generator](#)) e salvei na pasta do nosso projeto com o nome profile.jpeg.

Então aqui no vscode, vocês já conseguem ver que existe um novo arquivo `profile.jpeg` e a primeira coisa que faremos é adicionar essa foto no topo do nosso site. Então eu vou aqui em cima da tag `H1` e vou adicionar uma tag `img`.

E aí vem mais um conceito muito importante sobre as tags em HTML. As tags podem possuir propriedades, que são características que a gente define para nossas tags. Para definir uma propriedade da tag, a gente escreve o nome da propriedade, depois o sinal de `=`, abre aspas (simples ou dupla) e escreve o valor daquela propriedade.

No caso da imagem, por exemplo, eu preciso dizer de onde vem essa imagem, onde ela está localizada. Pra isso, a gente usa a propriedade `src` da tag ``. No meu caso, a imagem está na própria pasta, com o nome `profile.jpeg`. Então logo, o `src` vai ser `=` a abre e fecha aspas, `profile.jpeg`. Vale ressaltar que você pode usar um link aqui também, por exemplo, então você pega o link da imagem e coloca aqui no `src`.

Se eu abrir minha pagina, a imagem vai aparecer aqui. Mas eu quero reduzir um pouco essa imagem, eu quero que ela tenha uma largura de 100 pixels. Pra isso, eu adiciono mais uma propriedade chamada `width` (que significa largura em inglês), `= "100"`. Vou salvar aqui e vocês verão que a imagem vai ser reduzida.

IHC06 - Lista sequencial

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c20ab27f-1d94-4d52-9d8d-aeac2ef730f6/primeirosite.html>

Bom, agora que já aprendemos a criar imagens, vamos adicionar também uma lista, que é um tipo de elemento diferente do que vimos até agora porque ela depende de 2 tags (uma dentro da outra).

Começaremos adicionando aqui depois do título de “Experiências Profissionais” a tag `ul`, sempre lembrando de abrir e fechar ela. Depois, dentro da tag `ul` vou adicionar algumas tags `li`, e cada uma dessas tags vai representar um item da lista. Lembrando também de manter a formatação aqui dos espaços na hierarquia.

Vou criar aqui um item de experiencia profissional, com o título “Estagiário de marketing no DNC Group” e outro item com o título “Assessor de marketing no DNC

Group”. Vou salvar e agora vocês podem ver o resultado. No seu currículo aí, você adiciona também as suas experiências profissionais.

Aqui em cima, nas informações pessoais, vou adicionar mais uma lista e vou adicionar as informações pessoais, da seguinte maneira:

Nome: José da Silva

Sendo que o título da informação, que é aqui o Nome:, eu vou colocar entre uma tag de negrito, que é a tag **b** como a gente já falou.

Agora, eu vou copiar os itens da lista e adicionar mais informações, como o e-mail exemplo@email.com e o telefone, que eu vou inventar aqui.

IHC07 - Links clicáveis

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4af795d1-2d32-41fb-a140-d1c1b1eb6f3e/primeirosite.html>

Agora, a gente vai aprender também como criar links. Os links, como vocês já sabem, são tags pra redirecionar o usuário para outro endereço na internet. E eu vou começar aqui criando uma tag H2, com o título Links pessoais.

Depois, vou criar mais uma lista usando a tag `ul` com 2 itens dentro dela através das tags `li`. No primeiro item eu vou escrever LinkedIn. No segundo item, eu vou escrever Facebook.

Agora vem a tag de link. Tudo que a gente escreve dentro desta tag vai mudar de cor e passará a ser um link clicável. O nome dessa tag, é a letra `A`. Simplesmente uma letra mesmo, a letra `a`. Vou abrir e fechar ela aqui colocando o texto LinkedIn dentro dela. Agora, eu preciso dizer pra onde esse link vai ser direcionado. No caso, vou colocar o link do LinkedIn usando a propriedade `href`, que diz pra onde o usuário vai ser redirecionado. Vou digitar aqui “<https://linkedin.com>”.

Agora eu vou salvar a página e abrir essa página no navegador. Vou clicar e por padrão, os links vem programado para serem abertos na mesma página onde o usuário está. Reparem que eu cliquei no link e ele foi aberto na mesma página. No nosso caso, eu acho que seria legal esse link abrir numa nova aba, pra que a pessoa que estiver lendo o meu curriculum não saia dele.

Para isso, vamos utilizar uma propriedade dos links que se chama target (que em inglês significa "alvo"). E pra dizer que o target é uma nova pagina, a gente vai digitar _blank. Ou seja, abra este link numa nova pagina em branco. Vou salvar e vou mostrar para vocês agora como vai ficar.

Pronto, agora pra finalizar vamos só criar um link para o facebook também, que eu vou deixar SEM a propriedade target, só pra vocês visualizarem a diferença entre eles de maneira mais clara. Vou salvar e vamos no navegador atualizar a pagina pra ver as alterações.

IHC08 - Vídeos

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/07ec41bc-1f68-4c06-aa1f-eaf370748065/primeirosite.html>

A gente vai adicionar também um vídeo de apresentação no nosso curriculum. Esse vídeo vai ser um vídeo hospedado no Youtube, mas várias outras plataformas de conteúdo oferecem a opção de você incorporar conteúdo delas na sua página HTML. Isso é feito geralmente pela tag embed (que significa incorporar) ou pela tag iframe. Nesse caso aqui, em geral a gente não escreve o código, a gente vai até o site ou plataforma como eu vou fazer aqui agora e procurar o link pra gerar o código de incorporação.

No Youtube, basta você vir aqui no botão compartilhar, copiar esse código gerado e colar na página do seu curriculum. Vou colocar também em cima do vídeo aqui um título "Apresentação em vídeo".

Agora vamos salvar e ver o resultado.

IHC09 - Curriculum bagaceira

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9be9efdb-db4a-47b3-83cf-162e9ec4c31d/primeirosite.html>

Para fixar os detalhes até então:

1. É importante se atentar para a ordem dos itens no seu site;
2. A organização das listas é necessária, precisa se atentar na ordem que cada um dos itens é colocado;
3. A ordem dos atributos não importa;
4. Um erro bastante comum é a falha no fechamento das Tags.

IHC10 - Criando um formulário

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ddad22a1-8286-4c53-ac6c-fc90b42e7e55/Exemplo_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0eb6ff1b-343b-42aa-8e87-c40cee416c9a/Resultado_Forms.html

Até agora, a gente trabalhou apenas com tags estáticas, ou seja, nosso site não tinha nenhuma interatividade além dos links. Nós não trabalhamos com formulários, botões e outras tags mais interativas.

Na aula de hoje, a gente vai focar nisso. O objetivo é criar um formulário de contato pros nossos usuários enviarem uma mensagem para nós. Isso é comum nos sites das empresas, blogs e tudo mais.

Pra fazer isso, a gente vai trabalhar com algumas tags que são usadas para criar formulários. A primeira dela, é a tag `<form>`.

Os formulários são a forma principal de interação entre os usuários e os sites de vocês. Eles permitem que você colete dados preenchidos pelos usuários. Um

formulário é basicamente um conjunto de Campos, ou inputs. Os inputs, como o nome já diz, são as entradas de dados dos nossos formulários, ou seja, são aqueles campos onde a gente costuma digitar as informações nos sites que a gente visita.

Essa tag Form não muda nada visualmente na página, ela só diz pro navegador que tudo que está dentro dela vai ser preenchido e enviado.

Os formulários são definidos pela tag FORM e necessitam de dois atributos básicos. O primeiro é o atributo action, que representa o link para onde os dados do seu formulário serão enviados.

Já o segundo, é o atributo Method, que representa qual é o método pelo qual os dados serão enviados e ele pode ter 2 valores, GET OU POST. Basicamente, o método GET vai passar esses dados aqui através da URL, ou seja, esses dados vão poder ser visualizados através da URL. Enquanto no nosso caso, que usaremos o método POST, a gente vai estar dizendo que esses dados vão ser enviados dentro de um pacote HTTP, onde os dados não vão ser visíveis pela URL.

Vai ficar mais claro já quando eu mostrar para vocês, mas vale ressaltar que a gente não vai entrar muito em detalhes dessa parte porque isso envolve conceitos mais avançados de redes de computadores e de programação, pra receber esses dados, por exemplo, a gente precisaria programar um servidor rodando alguma linguagem de programação. Para enviar esses dados por e-mail, também. Por isso, no nosso caso, a gente vai usar um serviço que faz isso pra nós de forma automática. Na internet tudo funciona assim, sempre existem alternativas que não necessitam de conhecimento técnico avançado para funcionar.

O próprio código HTML por exemplo, pode ser gerado automaticamente usando alguma ferramenta das chamadas de NO CODE, que como o nome já diz, o intuito é que você não precisa saber programar para manipular. Então assim, hoje em dia, não saber programar não te limita em nada, sempre há uma solução alternativa, como essa que vamos usar hoje.

Mas voltando pro nosso formulário, a gente vai usar um serviço que vai ser responsável em receber os dados do nosso formulário e encaminha-los para algum endereço de e-mail. O nome desse serviço é formspree.io e ele é gratuito até 50 envios do formulário por mês. Vamos acessar aqui o formspree.io, Clicar em get started, preencher nossos dados e criar nossa conta. Você vai precisar acessar seu e-mail e clicar no link de validação que eles vão te enviar pra poder criar formulários, então faça isso antes de seguir.

Depois, você vai logar e clicar aqui em New Form. Depois que criar o form, você vai ver aqui que eles disponibilizam um link. E se a gente ler aqui, vocês vão ver que

eles mencionam as duas propriedades do nosso form, veja: "coloque essa URL no atributo action e use o método POST para enviar os dados" e é isso que vamos fazer aqui.

IHC11 - Inputs

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b5c4f2df-58ab-4b6b-8eb2-bf342b054a33/Exemplo_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/caea4de4-088a-4f9d-a232-7d238484dca3/Resultado_Forms.html

Voltando ao nosso código do VS Code, vou adicionar aqui um título h1 e escrever "Meu Formulário" dentro dela.

Depois, teremos aqui a nossa tag form e no action, vou colocar o link lá do formspree e no method vou colocar "POST". Pronto, agora a gente já vai poder começar a criar nossos inputs, que são as entradas de informação dos formulários. Existem vários tipos de campos, como inputs, textareas, checkboxes, radio groups, entre outros.

Aqui nessa página vocês podem ver alguns tipos de inputs e como a gente pode interagir com eles. Esses são os mais comuns, mas no link disponível aqui na aula você pode conhecer vários outros tipos. A gente teria que fazer um curso inteiro pra falar sobre cada um deles, então a gente vai abordar aqui os mais importantes e mais comuns.

Agora, vamos voltar aqui pra prática na nossa página de formulário. Vou começar criando um input aqui, que é feito pela tag input. Os inputs, assim como qualquer outra tag HTML, podem possuir atributos. Aqui a gente vai falar dos atributos mais importantes, que são esses:

- placeholder
- type

- required
- value
- name

O primeiro é o atributo placeholder. Como o próprio nome diz, este atributo define um texto que vai ficar guardando o lugar dentro do campo. Por exemplo, se eu colocar o placeholder Digite seu E-mail, o campo vai aparecer com este texto, porém, ao clicar com o mouse, o texto some. Sempre que o campo estiver em branco o texto deste atributo será mostrado.

O segundo é o atributo type. Este atributo define o tipo de dados que nosso campo vai receber. Existem vários tipos de dados que a gente pode colocar aqui, como cores, datas, números e etc. Em geral, você pode sempre usar o tipo text porque ele vai abranger qualquer entrada de dados.

O terceiro atributo é o atributo required. O único valor possível para esse atributo é required, ou seja, required= "required". Quando este atributo existe, o formulário não poderá ser enviado até que este campo seja preenchido. Sendo que uma mensagem de erro será mostrada, da seguinte maneira.

Já o atributo value, é o valor daquele campo. Ou seja, se você quiser que esse campo já venha preenchido com algo para facilitar a vida do usuário, você pode usá-lo. Por exemplo, se você tem um site destinado a pessoas que moram em São Paulo, mas que eventualmente pode receber dados de pessoas fora de lá. Nesse caso, você pode definir o value dele como São Paulo para facilitar a vida da maioria e, quando a pessoa for de outro local, ela pode editar esse valor. A diferença dele pro placeholder, é que o valor preenchido no placeholder nunca vai ser enviado como dado. O valor do value sim, é de fato o valor do campo e será enviado junto com ele.

No caso do atributo name, a gente define para que quando o formulário é enviado a gente saber a qual input cada dado está relacionado. Você pode colocar o nome que quiser, mas por exemplo, se o meu objetivo é captar o nome de alguém, nada mais objetivo do que chamar esse input de nome.

IHC12 - Labels

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/05c2da8d-9743-4e4f-add9-ada158b36079/Exemplo_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ac7047c8-cc50-4463-9afe-4c728539ea6e/Resultado_Forms.html

Beleza, já aprendemos a criar os inputs. Agora, a gente vai falar sobre as labels.

As labels funcionam como etiquetas, ou títulos para os nossos campos. Aqui, temos um formulário de login de exemplo com as labels e-mail e senha. A label costuma ser posicionada em cima do campo, mas você pode tratar dela da maneira que desejar, usando css.

A sintaxe da label é como a de qualquer outra tag, da seguinte maneira. Abre a tag label, escreve o texto e fecha a tag label.

IHC13 - Textarea

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e8ac4d07-6194-4238-b896-ef104681df07/Resultado_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2e0d0eea-0b71-4749-93f8-59acee07d18b/Exemplo_Forms.html

Além dos inputs e labels, temos também as textareas, que são como os inputs, porém, são maiores e comportam uma quantidade maior de texto, além de possuírem algumas qualidades especiais, como a possibilidade de se usar quebras de linhas no texto digitado.

Um detalhe: Diferente dos campos, ou inputs, as tags da textarea precisam ser fechadas. As textareas possuem basicamente os mesmos atributos que nós já

vimos pros inputs, como o placeholder, o name e o required. Porém, você pode usar dois atributos especiais nas textareas, que são os atributos rows e cols. Eles especificam a quantidade de linhas e colunas, respectivamente, que sua textarea vai comportar.

Veja nesse exemplo:

```
<textarea required="required" placeholder="Digite seu endereço" rows="5" cols="10">
</textarea>
```

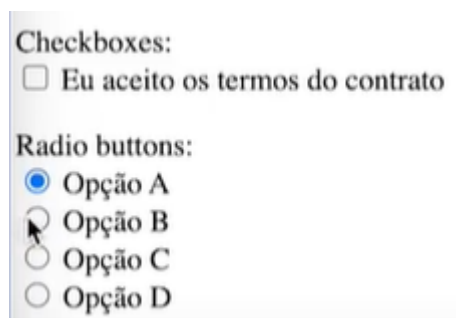
Outro fator importante a respeito das textareas é que elas não possuem o atributo value como os outros campos. Pra especificar um valor para uma textarea, você precisa inseri-lo entre as tags textarea.

IHC14 - Checkboxes e Radio Buttons

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/12e084f2-2444-438c-a88d-ecfa25102731/Resultado_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/162ab6a4-924d-4e98-b4f1-f9b1120af953/Exemplo_Forms.html

Já temos um input e uma textarea, agora vamos falar sobre as checkboxes e os radio buttons.



Checkboxes:

☐ Eu aceito os termos do contrato

Radio buttons:

☒ Opção A

☐ Opção B

☐ Opção C

☐ Opção D

Os checkboxes são esses botões, que podem ter os status marcado ou desmarcado, onde a gente clica pra selecionar uma ou várias opções. Reparem que os checkboxes são quadradinhos.

Já os radio buttons são usados quando existem várias opções, mas apenas uma delas deve ser selecionada, sendo que ao clicar em uma opção quando outro já esta marcada, esta outra se desmarca sozinha.

Mas vamos praticar um pouco. Primeiro vamos criar um checkbox. Pra isso, usamos a tag INPUT, porém, alteramos o type dela para checkbox.

Os atributos essenciais são:

value: que representa o valor que aquela checkbox vai passar quando estiver selecionada. Por exemplo, digamos que vai haver um formulário pra que os esportes que você pratica sejam selecionados.

Vou criar aqui dois checkboxes com a label "Esportes que você pratica", vou adicionar o br para quebrar uma linha e vou adicionar as opções Judô e futebol. Sendo que os values dessas opções serão judo e futebol.

Ao selecionar Judô, o valor da checkbox vai ser judô, e quando selecionar Futebol, o valor vai ser futebol.

Também temos o atributo name, que representa o nome daquele checkbox. Vou definir o nome dos dois como esportes praticados. Você pode colocar um nome diferente pra cada também, vai depender da sua necessidade aí.

Por último, existe o atributo checked que vai definir se um checkbox já deve ser marcado ao abrir a página. O valor possível pra esse atributo é checked, deste jeito.

Notem, que como estamos falando de uma checkbox, múltiplas opções podem ser selecionadas.

Agora, vamos falar dos radiobuttons. Eles são usados quando apenas uma opção deve ser selecionada. Por exemplo, no caso dos esportes praticados, o usuário pode praticar mais de um esporte. Se eu tivesse uma lista que dissesse "Selecione sua cor favorita". Nesse caso, só uma opção poderia ser selecionada e teríamos múltiplas opções de escolha para que o usuário defina uma única resposta.

Neste exemplo, vou criar um formulário para seleção que vai dizer: escolha sua cor favorita. Os mesmos atributos que usamos pro radio button continuam valendo, então desse modo, só precisamos clonar nosso primeiro formulário e alterar o atributo type para RADIO e o name para cor. Vou adicionar aqui também as opções de cores: vermelho, azul, preto e amarelo.

Vou salvar e testar e agora notem que apenas uma opção pode ser selecionada por vez.

Escolha os esportes que você pratica:

- ☒ Judô
- ☐ Futebol
- ☐ Tennis

Escolha sua cor favorita:

- ☒ Vermelho
- ☐ Azul
- ☐ Verde

IHC15 - Select

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bdce8371-fa31-480e-89f9-3091194a0680/Resultado_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7a5e91fb-dbf3-449c-83d0-7b4168a1778e/Exemplo_Forms.html

Outro input importante é o select, ou Dropdown lists, que são campos de seleção de opções. Pra mostrar para vocês, vou criar um aqui usando a tag select e adicionar o name dela como "Cidade". Também vou adicionar um label com o texto “escolha sua cidade”.

O select assim como o textarea também precisa da tag de fechamento, e além disso, possui tags que vocês insere dentro da tag principal, que são as opções que o usuário vai poder selecionar. Pra cada opção, a gente usa a tag option. Vou criar aqui então 3 tags de option e adicionar as cidades São Paulo, Rio de Janeiro e Brasília.

Porém, nos precisamos agora criar as opções da nossa lista. Pra isso, inserimos dentro da tag select, pra cada opção, uma tag chamada option. Entre essa tag,

inserimos o valor que desejamos que seja mostrado. Esta tag possui o atributo value, que indica o valor que será transmitido quando aquela opção for selecionada.

IHC16 - Botões

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0095619c-e924-4bea-ba15-edc7fd0926bb/Resultado_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/34d979ee-0451-4106-adc2-18c1fe3a5dc5/Exemplo_Forms.html

Pra fechar o assunto dos formulários, nós vamos falar agora dos botões. Os botões são responsáveis por enviar os formulários ou executar alguma ação na página, como vocês verão no módulo que fala de JavaScript. Nesse caso, vou criar um botão com a tag button e usar o atributo type pra dizer que aquele botão vai ser responsável por enviar o formulário.

Existem botões com ações diferentes de enviar, por exemplo, um botão de limpar todos os campos do formulário, então por isso que a gente precisa especificar que o type é submit.

O name não é necessário, porque a gente não vai precisar receber esse dado do botão, ele simplesmente vai disparar a ação de enviar, mas poderia ser um atributo o name também.

IHC17 - Testando o formulário

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/41829702-0f80-4b9f-bfe8-6f8f111071af/Resultado_Forms.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/832502c8-463b-4474-9b67-ef0ee40e0a44/Exemplo_Forms.html

Agora a gente consegue testar, bastando preencher o formulário e clicando em Enviar, como farei aqui.

No final, vamos receber os dados do nosso formulário por e-mail, graças ao formspree.

Você também pode ver os valores enviados pelo nosso form através do próprio formspree.

IHC18 - Google Fontes

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/827f05fd-5e93-4863-9e6f-1c6b1a8abcf1/Fontes_e_cones.html

O objetivo desta aula é a gente revisar um pouco dos conceitos que vimos até agora e trazer alguns novos elementos que vão dar mais vida para nossos sites.

Até agora, a gente trabalhou bastante com tags. A gente aprendeu que existem tags que precisam de fechamento, como a tag H1, e tags que não precisam de fechamento, como a tag BR. Também aprendemos que algumas tags podem ter filhos, como as tags UL pra criar listas, ou como as tags body e head. A gente também viu que as tags podem ou não ter propriedades que indicam como essas tags vão se comportar e conhecemos algumas propriedades de algumas tags e vale ressaltar que existem muuuitas outras propriedades, mas a gente precisaria de um novo curso inteiro só pra abordar uma a uma. Em geral, o importante é que você vá aprendendo novas tags conforme for precisando delas e se aprofundando no tema.

Talvez você tenha achado muita informação até agora, mas não se desespere em não ter decorado tudo. A intenção aqui é que você aprenda sobre como o HTML é estruturado e como as tags funcionam. Os detalhes em relação a propriedades de tags e até os próprios nomes, são vastamente documentados na internet e é mais importante que você saiba como encontrar isso na internet do que você decore. E

pra te mostrar como é fácil, eu vou pegar um exemplo aqui e vamos supor que eu precise criar uma tag textarea, mas eu me esqueci dos atributos ou do funcionamento.

Se eu venho no Google e digito html textarea, o primeiro link já é a documentação da mozilla, que é uma das mais confiáveis e completas da internet.

Clicando [aqui](#), você encontra toda a descrição da tag, exemplos de utilização, suas propriedades e etc. Esse site é a melhor referência pro caso de você ficar com alguma dúvida, então eu vou disponibilizar o link aqui pro caso de você querer se aprofundar.

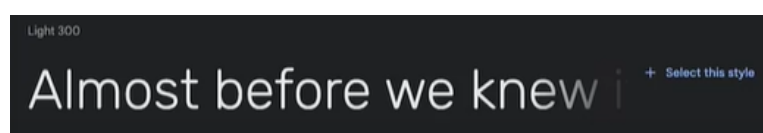
A primeira delas é o Google Fonts, que fornece pra nós fontes gratuitas pra a gente poder usar nos nossos projetos e deixar eles mais bonitos. Eu vou começar com um exemplo, voltando aqui pro nosso vscode e criando um novo documento html em nossa pastinha com o nome fontes.html.

Dentro desse arquivo, vou criar a tag HTML, depois a tag HEAD e a irmã dela, a tag BODY. Dentro do body, vou criar uma tag H1 com o conteúdo "Bem vindo ao meu site".

Vou salvar e vou deixar aberto aqui no navegador pra a gente poder ir acompanhando as alterações. Por padrão, o HTML vai usar a fonte padrão configurada no seu navegador, no meu caso é a Times New Roman. Mas a gente pode querer alterar essa fonte, e pra isso existe o Google Fonts.

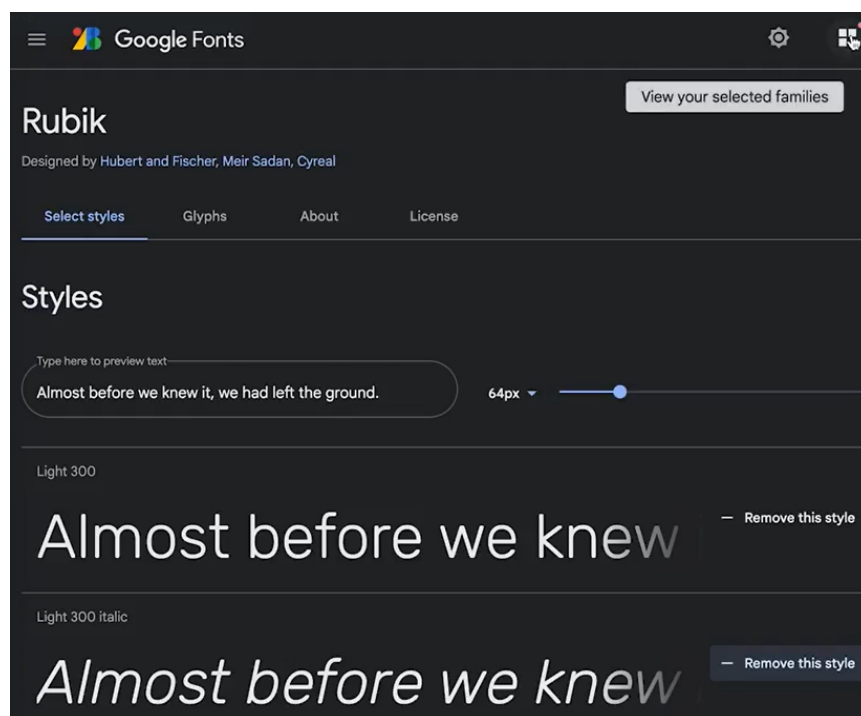
Vamos abrir aqui o site do Google Fonts, que é fonts.google.com, e vocês vão ver aqui um monte de fontes que a gente vai poder usar de graça. No meu caso, eu quero usar essa fonte Rubik aqui, então eu vou clicar nela e a página dela vai abrir.

Agora, a gente seleciona aqui do lado qual estilo da fonte a gente quer usar, vocês vão ver aqui que ele disponibiliza pra nós várias grossuras diferentes para esta fonte. No meu caso, eu vou pegar essa Bold 700. Pra isso, é só clicar em select this style aqui do lado dela.



Se você quiser usar mais de uma fonte ou mais de uma grossura da mesma fonte, você pode clicar em select this style várias vezes e o Google Fonts vai gerando pra nós o código que usaremos, como vocês verão.

Pra isso, é só clicar aqui no canto superior direito, nesse ícone. E teremos aqui o código html da nossa fonte.



Esse primeiro código, a gente vai copiar e colar no head da nossa página. OK, mas porque a gente vai colar isso no head e não no body? Bom, o head é a parte do documento que é carregada primeiro. Ou seja, antes de poder usar a minha fonte no body, eu preciso que ela já esteja 100% carregada pelo navegador, então sempre que acontecer essa situação, a gente vai adicionar esses recursos na head do nosso documento. Então vou pegar aqui, colar no head e depois formatar o código direitinho lembrando sempre daquela questão de manter o espaçamento e a hierarquia certinhos. Esse código que a gente adicionou na cabeça do documento, vai usar a tag script como vocês podem ver aqui. A função dessa tag é carregar algum script que pode estar em um link fora do seu site, que pode ser um arquivo do seu site ou voce pode adicionar o seu script dentro da tag mesmo.

No nosso caso, ele vai carregar o script que traz a fonte pra nós diretamente do site do google fonts. Vamos voltar a usar essa tag é como injetar um script externo no nosso site e vocês vão notar que vários outros recursos que a gente encontra na internet usam esse método para modificar o nosso site.

Mas beleza, se vocês salvarem e abrirem isso, a fonte ainda não aparece na página. Continua tudo igual. Então pra resolver isso, a gente precisa agora dizer pra nossa tag h1 usar essa fonte. Pra isso, vamos usar uma nova propriedade das tags que ainda não abordamos, que é a propriedade style, ou estilo em português.

Voltando na pagina do google fonts, a gente tem esse segundo código aqui que eu vou copiar. Vou voltar ao vscode, criar a propriedade style no meu h1 e colar esse conteúdo dentro dela. Vou salvar o arquivo e atualizar a página no navegador, e vocês verão que o estilo do meu título vai mudar pra fonte que eu desejo, que é a fonte rubik. E assim você já poderá combinar diversas fontes no seu site.

Nas próximas aulas, a gente vai aprender a mudar a cor da fonte, o tamanho e várias outras características, mas continuando aqui, eu queria mostrar para vocês mais uma dessas ferramentas disponíveis na internet.

IHC19 - Fontawesome

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8a6cdf3b-7faf-4981-84c0-40d7a1349231/Fontes_e_cones.html

Um outro recurso bastante utilizado são os ícones que a gente pode colocar dentro de links, dentro de botões, no meio do texto, nas listas, enfim, tem várias formas de utilizar. E eu queria mostrar pra vocês uma das ferramentas mais utilizadas para a criação de ícones que é a biblioteca fontawesome.

A primeira coisa a fazer, é acessar o fontawesome.com e clicar no botão start for free.



Aí você digita seu e-mail aqui e eles vão te enviar um link por e-mail. Aí você clica no link do e-mail, cria uma senha e pronto, você vai estar logado no painel. Aí voce vem aqui e clica em New Kit, e pronto!

Eles já vão te dar acesso ao código que você vai inserir dentro da tag head do seu site.

Mais uma vez, lembrem que a gente só insere o código na head quando a gente precisa que aquilo seja carregado primeiro pra ser usado depois no body.

Em geral, os plugins e ferramentas que você for utilizar vão te avisar isso, assim como aqui no fontawesome eles estão pedindo pra que a gente adicione o conteúdo na tag head, veja. Então a gente copia esse código e traz ele aqui pro nosso código e cola ele aqui na head. Agora, a gente volta no fontawesome pra poder escolher os ícones que a gente vai utilizar, usando essa caixa de pesquisa aqui. A gente precisa pesquisar em ingles, então eu vou buscar o ícone house, que significa casa.

Depois, eu vou selecionar aqui Free, do lado esquerdo, pra filtrar só os ícones que são gratuitos. Aí é só clicar no ícone que voce deseja usar e o código html pra usar esse ícone já vai aparecer aqui. Vou clicar no código para copia-lo e vou voltar aqui pro vscode. Vou criar um link com o texto "Voltar para o início" e dizer que o link dele é a página inicio.html, lembrando que a gente define o link usando o atributo href.

Dentro deste link, eu vou colar o código do ícone que eu peguei lá no fontawesome. Vou salvar e atualizar a página e vocês vão ver o ícone aparecendo aqui.

E aí você pode navegar no fontawesome, pegar os ícones que você quer e ir adicionando conforme a sua vontade no seu site. E você pode combinar o ícone colocando ele dentro de outras tags, como eu fiz aqui dentro da tag link, mas poderia ser dentro de uma tag h1, dentro de uma lista, dentro de um botão e etc.

Nas próximas aulas, a gente vai começar a trabalhar com os estilos, que são adicionados através do CSS. Em resumo, o CSS é o que faz o seu site ficar bonito. Até agora, tudo que a gente fez foi sem CSS, então vocês devem ter notado que não fica tão bonito quanto os sites que vocês estão acostumados a ver aí pela internet. Com CSS, tudo isso vai mudar e é sobre isso que vamos conversar nas próximas aulas.

IHC20 - Primeiros passos no CSS

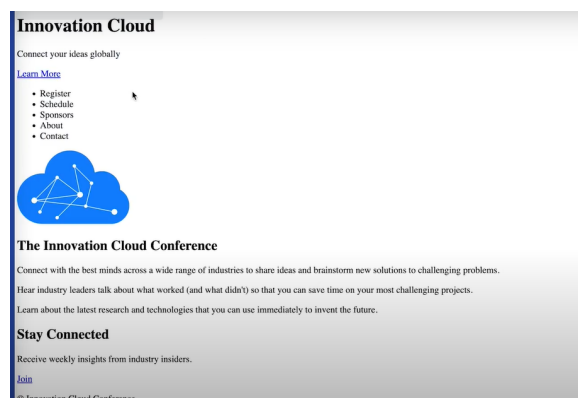
<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/35ab2003-5927-4a6a-a6e7-dc656342ca2c/div.html>

A gente vai começar a falar sobre CSS e aprender como deixar nosso site mais bonito. Até agora, a gente viu que HTML é a estrutura, ou o esqueleto, do nosso site. Vocês também repararam que o HTML sozinho é bastante feio, então fica faltando uma "embelezada" nele pra ficar mais bacana.

O HTML nunca foi criado com a intenção de permitir a formatação visual de uma página, mas sim para descrever o conteúdo de uma página e por isso que é chamado de linguagem de marcação. O CSS, que significa cascading style sheets (ou em português, folha de estilos em cascata) veio pra resolver este problema, lá no começo da internet a gente só tinha essas páginas mais feias. Aí criaram o CSS e usando ele, a gente consegue alterar fontes, cores, cor de fundo, largura, altura, posicionamento e várias outras características dos elementos HTML, ou seja, resumidamente, a função do CSS é descrever como os elementos HTML vão ser exibidos na tela.

Pra exemplificar o poder do CSS, eu queria mostrar para vocês um exemplo bem bacana. Eu tenho aqui 2 arquivos que eu vou carregar dentro do meu VSCode, clicando aqui em File, Open Folder e selecionando a pasta aqui onde meus arquivos estão. Bom, vou abrir esse arquivo aqui no navegador e a gente vai analisar um pouquinho. Esse site é um site que usa apenas HTML, como vocês podem ver, até aqui são apenas coisas que vocês já conhecem... Titulos, listas, links, etc.

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fe7e590d-6b47-4125-a1d1-38af872efbea/sitesemcss.html>



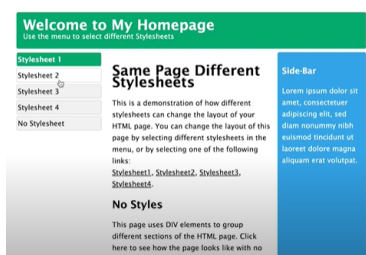
Agora, eu vou voltar aqui no vscode e eu vou carregar um arquivo com o CSS desta página. Esse CSS estava comentado aqui e como eu falei pra vocês anteriormente, o conteúdo comentado não é renderizado pelo navegador, é como se ele não existisse. Então agora eu vou 'descomentar' esse conteúdo aqui pra a gente ver o resultado.



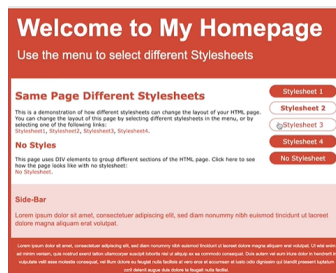
E olha só a diferença, olha como fica muito mais profissional com CSS. E reparem também que tudo que vocês viram até agora de conteúdo, é a base, é o esqueleto. Com CSS, a gente vai aprender a dar vida pra esse esqueleto, deixando ele bem mais estiloso.

Outra coisa que eu queria mostrar é que só o CSS te dá a flexibilidade total que você precisa para construir qualquer tipo de site ou página para a internet. Então eu tenho aqui um outro exemplo pra mostrar pra vocês, que é o seguinte.

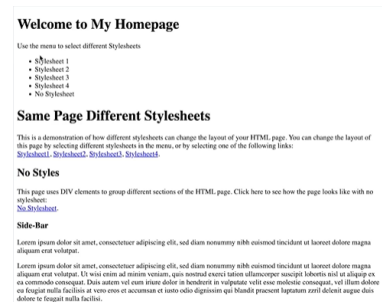
Temos aqui este site que ele tem uma estrutura em HTML definida. Aqui no menu da esquerda, a gente consegue alterar os estilos em CSS desse site, e vocês vão notar que o mesmo site com a mesma estrutura HTML pode virar coisas completamente diferentes dependendo da construção do CSS, olha só:



Estilo 1



Estilo 2



Sem estilo

IHC21 - Conhecendo a Tag Div

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9df9143e-894e-4e72-9b71-0fe03859592b/div.html>

Então agora que vocês já viram o poder do CSS, eu quero apresentar pra vocês primeiro uma nova tag, que é a tag div.

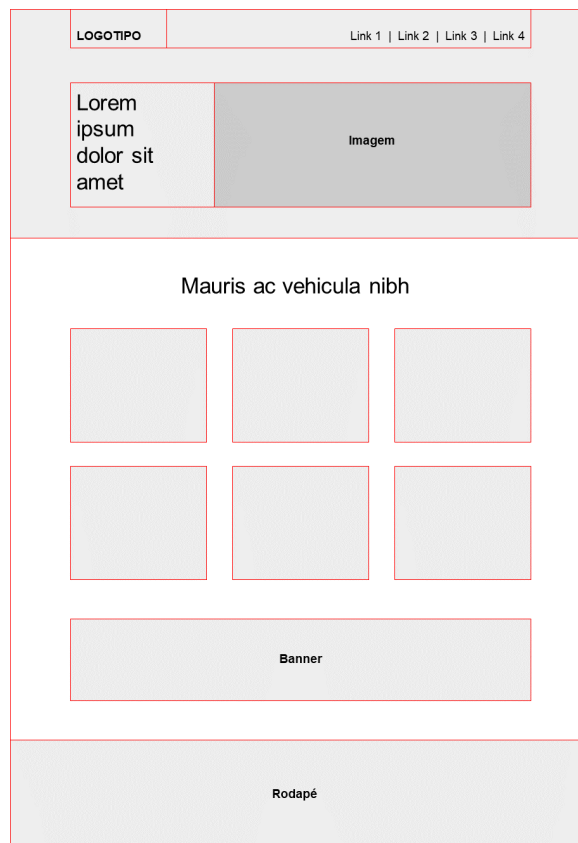
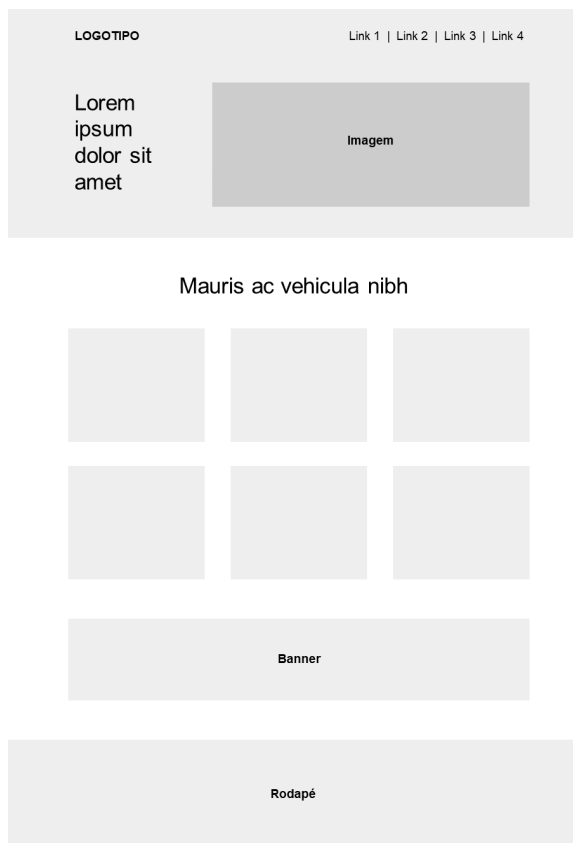
A div é uma das tags mais usadas no HTML hoje em dia. A div é como uma espécie de retângulo, ou container, que pode ser totalmente manipulado com CSS para se transformar. Elas servem pra delimitar áreas ou seções usando HTML e qualquer tipo de conteúdo pode ser inserido dentro de uma div.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a459b8e6-777a-44d1-bf72-77db7e988a3c/Pgina_Curriculum_-_Divs_selecionadas.pdf

Pra ficar mais claro pra vocês, eu montei este exemplo de estrutura do layout de um site fictício.

Onde a gente tem aqui um header, que é uma espécie de cabeçalho do site, com logotipo, menu, imagem e texto. Logo em baixo, a gente tem uma lista que poderia ser uma lista de produtos ou serviços, por exemplo. Um banner e um rodapé.

Reparem que é uma estrutura bem comum e simples de site. Agora, eu queria mostrar pra vocês onde a gente poderia usar divs pra estruturar este site aqui.



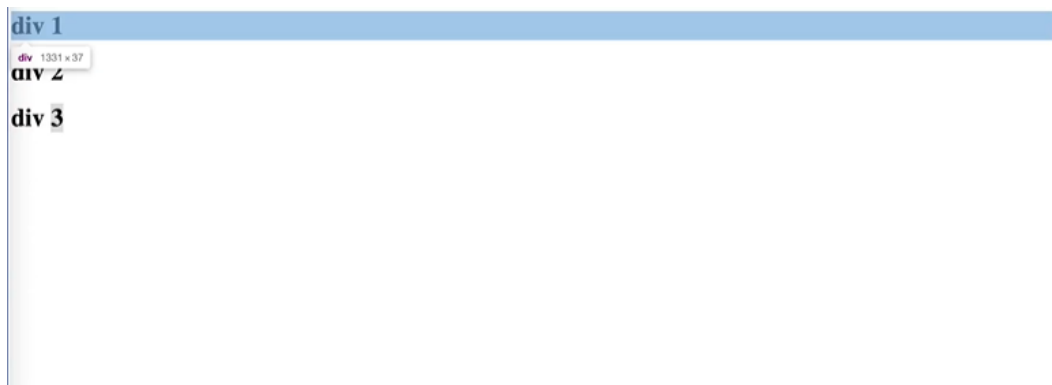
Pronto, reparem que as divs demarcam as seções do site. A estrutura do site é, basicamente, um conjunto de retângulos e a gente usa essas divs pra poder facilitar a organização e a estruturação visual do nosso site e como vocês podem ver, um layout é nada mais do que um conjunto de divs, que são estilizadas e formatadas pra formar um design.

Pra criar uma div, é só usar a tag `div` e inserir dentro dela o conteúdo que você deseja que seja mostrado. Você pode inserir links, imagens ou qualquer outro tipo de tag dentro das divs. Por padrão, as divs são invisíveis e retângulos que ocupam 100% da tela, mas você pode modificá-las de várias formas pra fazer o que quiser, por exemplo adicionar sombras, torná-las redondas, quadradas, triangulares e etc.

Pra exemplificar pra vocês, vou abrir aqui o VSCode, criar uma tag HTML com `head`, `title` e `body` e depois criar três divs, abrindo e fechando a tag `div` três vezes. Vou adicionar dentro de cada div uma tag `H1`, com os títulos `div 1`, `div 2` e `div 3`.

Vou salvar e abrir aqui no navegador pra mostrar pra vocês que é como se essas divs não existissem, porque até agora nada mudou. Porém, se eu clicar com o

botão direito na página e navegar até Inspecionar, fica nítido que as divs existem, elas estão aqui.



Beleza, isso acontece porque como eu falei pra vocês, as divs por padrão são invisíveis e ocupam 100% da tela.

Antes de começarmos a adicionar estilos nas nossas divs, eu quero mostrar pra vocês um novo atributo das tags HTML que a gente não viu até agora, que é o atributo id. O atributo ID é como se fosse o CPF daquele elemento HTML, ou seja, cada id deve pertencer a apenas um elemento da página, você não deve ter mais de um elemento com a mesma ID, assim como seu CPF, seu CPF só você tem e ninguém mais no Brasil tem.

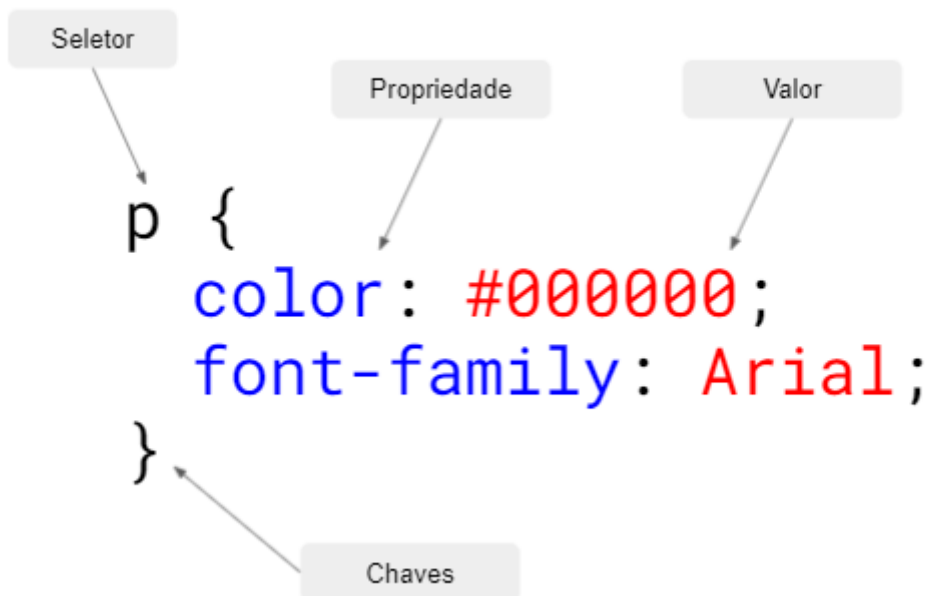
E pra que isso serve? Isso vai ser usado quando a gente começar a fazer a parte de CSS. No CSS a gente vai precisar definir o estilo do elemento e pra especificar QUAL elemento a gente quer estilizar, a gente vai usar esse ID dele. Como é um identificador único, só esse elemento tem, a gente vai poder dizer que só esse elemento tem um determinado estilo que nós vamos definir.

Então pra isso, nós vamos definir o ID dessas três divs que nós criamos. A primeira nós vamos chamar de div1, a segunda de div2 e a terceira de div3.

Reparem que a gente nunca usa espaços, porque vai impossibilitar a gente trabalhar com o CSS depois.

Agora, eu quero mostrar pra vocês como é a sintaxe do CSS, ou seja, a forma correta de se escrever e se especificar os estilos em CSS. Por exemplo, em uma tag HTML a sintaxe é símbolo de menor que, nome da tag, símbolo de menor que, certo?

Agora deem uma olhada nesse exemplo:



Essa é a sintaxe do CSS. Basicamente, temos o seletor, que é o elemento onde nós vamos aplicar este estilo. Nesse exemplo aqui, nós estamos usando um seletor genérico, dizendo que TODAS as tags p do site terão o estilo que esta entre as chaves.

Mas você também pode selecionar um único elemento através da ID, e estilizar este único elemento, ou você também pode selecionar um grupo de elementos através do atributo class, a gente vai ver exemplos práticos disso em breve, o importante aqui é entender a sintaxe, que é a forma correta de se escrever código em CSS.

Então temos o seletor, abrimos chaves, e aí declaramos as propriedades e os valores. Nesse exemplo, estamos dizendo que o texto de TODOS os parágrafos vai ter a cor preta (que é representada pelo código #-6-zeros e também vai usar a fonte Arial nos textos escritos dentro dele. Reparem que é bem simples, propriedade, dois pontos, valor da propriedade, ponto e virgula. Funciona assim para todas as propriedades. E claro, assim como tags, existem dezenas de propriedades que vocês podem usar e, você também não precisa decorar todas, você pode sempre recorrer ao google ou outros sites como esse que deixarei de referência aqui pra te ajudar durante o processo:



Referência Mozilla: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

IHC22 - Dando estilo a DIV

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8ae63acb-72d7-49d8-9e13-5317645d8268/div.html>

Então beleza, agora que você conhece a sintaxe do CSS, vamos começar a por a mão na massa definindo os estilos dessas divs que a gente criou. Pra definir os estilos, a gente pode fazer no head ou no body do documento HTML.

Em geral, a gente costuma fazer isso no head do documento porque como eu já expliquei pra vocês, o que está no head é carregado primeiro do que o resto, então quando o seu conteúdo HTML estiver carregado, o CSS também já vai estar. Depois eu vou ensinar pra vocês como fazer isso de algumas outras formas, mas por enquanto vamos nos concentrar em praticar a sintaxe do CSS

Pra definir os estilos, a gente vai abrir e fechar aqui na head uma tag chamada style. Dentro dessa tag style, a gente vai começar a definir os estilos dos nossos elementos.

Dentro das tags style, vou digitar primeiro o meu seletor. No meu caso, eu quero aplicar meus estilos CSS a um único elemento, que é a div 1, que possui o ID div1.

Pra dizer isso, no CSS, a gente digita hashtag antes do id que a gente quer selecionar. No meu caso, #div1.

Vale ressaltar que o hashtag só serve pra selecionar elementos através da ID, e apenas o elemento que possuir a id definida será selecionado. Agora eu vou abrir e fechar chaves e em seguida, adicionar as declarações das propriedades que eu quero que sejam alteradas.

Primeira propriedade, é o background-color que, como o próprio nome já diz, define a cor de fundo da div. Vou digitar aqui background-color dois pontos, e agora precisamos adicionar o código da cor que a gente quer usar.

Esse código se chama código hexadecimal e não, você não precisa decorar todos de cabeça. Pra achar o código correto, você pode usar alguma ferramenta como o color picker, que esta disponível no google.

Basta você pesquisar por html color picker no google, escolher a cor, copiar o código hexadecimal e colar lá como valor da propriedade. Repare também, que no vscode vai aparecer um quadradinho representando a cor que você especificou aqui

no código hexadecimal. Agora você pode clicar nesse quadradinho, selecionar a cor que deseja, e o código dela vai ser automaticamente preenchido aqui na página.

Antes de a gente ver o resultado, vou setar também a propriedade width para 400 pixels, ou seja, essa div agora vai ter 400 pixels de largura. Também vou setar a propriedade height, que significa altura, para 400 pixels e aí teremos um quadrado. Vou salvar e a gente e a gente vai abrir a página aqui pra testar e vocês verão que as alterações foram feitas na nossa primeira div, porque a gente só selecionou ela.

Vou voltar agora pro meu código e alterar as cores e os tamanhos das outras divs. Vou selecionar a div2 e alterar a cor, a largura e a altura dela. Dessa vez, vou usar porcentagens para definir a largura dela.

Por último, vou selecionar a div3, alterar a cor, a largura e a altura dela e também a fonte, usando a propriedade Font Family e usando a fonte verdana. Reparem que a gente também pode definir a cor pelo nome, vale apenas para algumas cores principais. Se a gente salvar e visualizar, as alterações estarão aqui também.

IHC23 - Atributo Class

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/68b362f1-6c7f-4802-ac06-3567770f903e/div.html>

Agora, vou apresentar pra vocês também uma nova propriedade que vocês ainda não conhecem, que é a propriedade class. Vou definir que o atributo class de todas as divs tem a classe allDivs. Lembram do atributo ID que eu expliquei pra vocês que deve ser único pra cada div? A class é ao contrário, a gente define classes pra poder reutilizar várias vezes o mesmo estilo.

Você também pode misturar mais de um estilo, eu poderia por exemplo adicionar aqui classe1, classe2 classe3 sempre separando com espaço.

Beleza, mas vamos agora aprender como definir o estilo de uma classe.

Pra selecionar uma classe no CSS, ao invés do hashtag, a gente digita o ponto e o nome da classe, que é allDivs. E aí é o mesmo padrão, abre chaves e digita as propriedades que a gente quer. Nesse caso, eu vou definir que todas as divs vão ter

uma borda de 2 pixels da cor preta, através da propriedade border, onde a gente precisa especificar 3 detalhes.

O primeiro é a grossura da linha da borda, que vou colocar como 2 pixels. A segunda, é o estilo da linha, se é uma linha tracejada ou sólida por exemplo. Vou usar uma linha sólida, então através do valor solid. E a terceira é a cor da borda, que no nosso caso vou definir como black (ou preta).

Também vou criar uma outra propriedade chamada border-radius, que é a curvatura dos cantos da borda, que vou definir como 10 pixels. Vamos salvar e testar aqui e vocês vão ver o resultado.

Pra fechar, vamos usar um último seletor, que é o seletor que seleciona todas as tags. Vou definir que toda tag h1 vai ter uma fonte com o tamanho de 40 pixels, através da propriedade font-size. Vamos salvar de novo e conferir o resultado.

IHC24 - Font-size e Inline-style

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9680266e-f25b-4c86-8ef5-818033ee7a8f/div.html>

Pra encerrar, gostaria de mostrar pra vocês outras formas de podermos declarar estilos do CSS.

Você pode estilizar todas as tags equivalentes de uma única vez. Selecionando o h1 e abrindo as chaves, é possível fazer as adaptações nela.

Através dos inline styles, ou seja, os estilos são definidos diretamente dentro das tags. Aqui no exemplo, vou pegar alguma tag H1 e fazer isso usando o atributo style nela, que funciona como qualquer outro tipo de atributo das tags.

Aqui dentro, é só você digitar as propriedades e os valores do CSS, separadas por ponto e vírgula, exatamente como a gente aprendeu no começo. Vou adicionar aqui uma font family verdana e uma color green, pra mudar a fonte e a cor dela.

Vamos salvar e vocês vão ver aqui o resultado. Vale ressaltar que fazer dessa forma seu código HTML vai ficar meio bagunçado, então é bom a gente evitar declarar o estilo inline, a não ser que seja uma única coisa pequena que você queira estilizar.

IHC25 - Salvando CSS separadamente

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c2d101df-3c59-4831-ba77-f5d24fa89df8/div.html>

Vocês podem salvar o código CSS num arquivo separado, pra ficar mais organizado ainda. Pra fazer isso, a gente vai copiar todo o conteúdo que esta aqui dentro da tag style, criar um novo arquivo chamado styles.css, lembrando que é importante não esquecer da extensão ponto CSS no final. Dentro desse arquivo, vou colar aqui o conteúdo do meu CSS.

Beleza, só que a gente ainda não disse lá no arquivo principal que precisaremos carregar esse estilo de um arquivo externo. Pra isso, vamos apagar a tag style e adicionar uma nova tag no head, que é a tag link. Essa tag link, faz referência a algum arquivo externo, e pra definir esse arquivo externo, a gente usa o atributo href e digita aqui dentro o nome do arquivo que a gente quer carregar. Pra fechar, a gente adiciona um novo atributo pra dizer que esse arquivo é uma folha de estilos, através do atributo rel e o valor dele será stylesheet, aí posso fechar a tag link.

Vou salvar a pagina, atualizar no navegador, e vocês vão reparar que a pagina continua exatamente igual, porque os estilos estão sendo carregados do arquivo com sucesso.

É isso, hoje aprendemos sobre divs, sobre novos atributos e sobre diferentes formas de carregar CSS. Por hoje é só e até a próxima aula.

IHC26 - Criando o Index

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9833d3de-3de4-4c06-b1cb-3c6e5aa2e580/index.html>

Vamos nos aprofundar um pouco mais no CSS com bastante prática. Nos primeiros vídeos, a gente desenvolveu

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9833d3de-3de4-4c06-b1cb-3c6e5aa2e580/index.html>

uma página de curriculum bem rustica usando só HTML.

[com/secure.notion-static.com/731d7d43-51bb-43bd-8fc0-40ebf674c91b/cursoppm2-curriculum2-html.pdf](https://secure.notion-static.com/731d7d43-51bb-43bd-8fc0-40ebf674c91b/cursoppm2-curriculum2-html.pdf)

O objetivo de hoje é que a gente desenvolva uma nova pagina online de curriculum como essa daqui, dessa vez mais caprichada e usando o que a gente aprendeu sobre CSS e alguns conceitos novos.

Começamos criando uma nova pasta no meu computador com o nome curriculum, pode ser em qualquer lugar que você quiser. Depois vou no meu VScode, no menu lateral vou clicar em file, open folder e abrir essa pasta que eu criei.

Agora deve criar um arquivo chamando index.html. Index.html é o padrão de nome que a gente usa para ser a página principal do nosso site.

Em geral, todos sites usam esse padrão e até os servidores onde os sites ficam já são programadores para ler esse arquivo como principal.

Então depois de criado o arquivo, vou ensinar um novo truque pra vocês que funciona no VScode. Pra acelerar o processo de criação da tag html, da head e do body, você pode digitar aqui html:5 e apertar a tecla TAB. Você vera que o código vai ser gerado de maneira automática, mas é importante que você também saiba criar isso por conta própria.

Além das tags HTML, HEAD, TITLE e BODY que vocês já conhecem, ele criou também 3 tags que até agora a gente não tinha interagido. São as tags META que são responsáveis por definir os metadados da nossa página, que são configurações que a gente coloca pra definir a linguagem, as resoluções que o site vai atender e várias outras características.

A mais importante aqui que vocês precisam conhecer e que nós vamos começar a utilizar agora é a tag meta com o atributo charset=utf-8, que em geral é necessária para dizer pro navegador qual é o tipo de caracteres e acentuações que vamos utilizar. Se esse site fosse escrito em árabe por exemplo, os caracteres seriam diferentes, então a gente define isso pra garantir que todo navegador abra os caracteres corretos no mundo todo.

IHC27 - Ajustes iniciais do cabeçalho

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c63c9088-5472-409a-aa9b-f916a359455b/index.html>

Então vamos começar aqui criando essa parte de cima que contém o nome e o cargo atual ou a sua profissão, que a gente chama de header, ou cabeçalho da página. Pra isso, a gente vai criar duas divs. Uma dentro da outra.



O motivo é o seguinte: reparem que aqui, temos dois retângulos, um dentro do outro. O retângulo de trás é o colorido enquanto o da frente é o retângulo em branco. A gente também poderia criar 3 retângulos aqui, um colorido, um branco e outro colorido, mas eu pessoalmente acho mais simples fazer com apenas dois.

Então vamos criar aqui as duas divs, uma dentro da outra. Dentro da div mais de dentro, eu vou adicionar um título usando a tag H1 que vocês já conhecem e a tag H2 que é o subtítulo, ou o cargo no nosso curriculum. Vou escrever aqui na tag H1 o nome e na H2 vou escrever a formação ou profissão, vocês façam o mesmo aí usando seu nome e sua formação.

Lembram do atributo ID?

Então, agora nós vamos definir o ID dessas duas divs que nós criamos. A primeira que é pai da outra, nós vamos chamar de header. A segunda que é filha dela, nós vamos chamar de nameAndTitle, reparem que a gente nunca usa espaços, porque vai impossibilitar a gente trabalhar com o CSS depois. Eu também costumo nomear em inglês e usar um padrão de nomeação que se chama camel case, onde a primeira palavra começa com letra minúscula e todas as outras seguintes tem a primeira letra maiúscula.

Agora, vamos definir os estilos dessas divs que a gente criou e agora que a mágica do CSS começa. Pra definir os estilos, a gente pode fazer no head ou no body. Em geral, a gente costuma fazer isso no head do documento porque como eu já expliquei pra vocês, o que está no head é carregado primeiro do que o resto, então quando o seu conteúdo HTML estiver carregado, o CSS também já vai estar.


Pra definir os estilos, a gente vai abrir e fechar aqui na head uma tag chamada style. Dentro dessa tag style, a gente vai começar a definir os estilos dos nossos elementos. Das aulas anteriores, você já conhece a sintaxe do CSS, então vamos começar a por a mão na massa.

O objetivo aqui é a gente botar essa corzinha aqui no retângulo de trás, alterar a fonte do título e também centralizar o retângulo no meio. Antes, vamos só dar uma olhadinha no que temos até agora. Vou salvar e abrir no navegador e vocês verão que, até agora, a gente só criou as divs e o título, então o que temos até agora é isso daqui.

Abrindo novamente aqui o projeto no VS CODE, vamos continuar estilizando nossas divs.

Dentro das tags style, vou digitar primeiro o meu seletor. No meu caso, eu quero aplicar meus estilos CSS a um único elemento que possui a ID header. Pra dizer isso, no CSS, a gente digita hashtag antes do id que a gente quer selecionar. No meu caso, hashtag header. Isso quer dizer que apenas o elemento que possuir a id header será selecionado. Depois, vou abrir e fechar chaves e em seguida, adicionar as declarações das propriedades que eu quero que sejam alteradas.

Primeira propriedade, é o background-color que, como o próprio nome já diz, define a cor de fundo da div. Vou digitar aqui background-color dois pontos, e agora precisamos adicionar o código da cor. Esse código como já falamos se chama código hexadecimal eu vou definir aqui a cor branca, só pro VS Code nos mostrar o quadradinho de alteração das cores. Repare que no vscode vai aparecer um quadradinho representando a cor que você especificou aqui no código hexadecimal. Agora voce pode clicar nesse quadradinho, selecionar a cor que deseja, e o código dela vai ser automaticamente preenchido aqui na página. Vou salvar e vamos visualizar o resultado:



Fulano da Silva
Analista de sistemas

IHC28 - Aperfeiçoando o cabeçalho

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a519679d-9540-4b2e-80b7-6161abeb8e11/index.html>

Agora precisamos criar uma margem aqui no topo e outra em baixo, um espaço em branco. Pra criar margens, usaremos as propriedades `margin-top` e `margin-bottom`, que significam, margem do topo e margem inferior. Então vou adicionar uma margem no topo de 50px e em baixo de 50px também.

Agora eu vou selecionar a fonte do topo, usando o Google Fonts que eu já expliquei pra vocês. Vou acessar aqui o site do google fonts e selecionar a fonte que eu quero, que é essa Hubballi. Vou aproveitar também pra selecionar uma outra fonte que usaremos só depois, que é a fonte Montserrat.

Agora que já selecionei as fontes, eu tenho aqui o código HTML pra carregar as fontes. Vou copiar esse código e colar na cabeça do nosso código HTML. Agora a gente já pode usar as fontes. Vou copiar aqui no google fonts o nome da fonte, voltar aqui pro meu estilo da div header e colar aqui a fonte hubballi.

Pronto, já temos a fonte. Agora eu vou definir outra propriedade pra fazer todos os títulos ficarem em caixa alta, ou caps lock, que é a propriedade `text-transform` e o valor dela será `uppercase`.

Pronto, para o header acabamos. Vamos ver como esta agora e vocês vão notar que ainda não está do jeito que a gente quer, porque ficou faltando estilizar a div `nameandtitle`.

O estilo dela será:

`background-color: white`; Ou seja, cor do fundo será branco, e reparem que eu digitei o nome da cor, algumas cores podem ser definidas assim, como `white`, `blue`, `red`, e algumas outras.

O proximo ajuste será o de tamanho. Por padrão, como eu já falei pra vocês, as divs ocupam a tela inteira. Nesse caso aqui, eu quero que ela ocupe um espaço menor, que vou definir como 400 pixels usando a propriedade `width`.

Também vou declarar aqui que o texto vai ser alinhado ao centro, usando a propriedade `text-align` e o valor `center`.

Agora, pra alinhar essa div ao centro, a gente vai usar a propriedade `margin`. Eu vou definir que o valor da margem será 0 e auto e isso significa que a margem do topo será zero enquanto todas as outras serão calculadas automaticamente e

distribuídas igualmente. Anteriormente, a gente declarou a `margin top` e a `margin bottom`, onde a gente só adiciona um valor fixo. Usando a propriedade `margin` apenas, sem definir se é `top`, `bottom`, `left` ou `right`, a gente pode definir todas as margens de uma vez, ou seja, eu poderia botar aqui as 4 margens seguidas em ordem, primeiro a do topo, depois da direita, depois a inferior e por ultimo a da esquerda.

E se eu declaro apenas duas margens por exemplo, ele infere que todas as margens não declaradas terão o mesmo valor da ultima margem declarada. No caso, se eu declarei 0 e depois `auto`, o resultado será que a margem do topo será zero e todas as outras serão automaticamente definidas, por isso usamos o `margin auto`.

Reparem que nessa `div` a gente não declarou a fonte que será usada. Isso é porque a gente declarou na `div header`, que é o pai dessa e todas as `divs` filhas acabam herdando essa característica da pai, mas você poderia por exemplo declarar a fonte aqui também sem problemas.

Com isso, a gente já consegue ver o resultado do que fizemos até agora, vamos salvar aqui e abrir nossa página no navegador e teremos aqui nosso header.

Reparem que ficou um espaço muito grande aqui. Isso acontece porque por padrão os elementos em HTML já possuem certos estilos que o navegador coloca. Clicando aqui na página com o botão direito e depois em inspecionar, vocês verão que existem duas barras laranjas nesses elementos. Essas barras são as margens padrão deles e eu gostaria de tira-las só pra esses dois títulos aqui de cima, não para o resto. Pra isso, nós precisamos seleciona-los e vocês já conhecem vários caminhos para isso.

O que a gente não pode fazer, é adicionar aqui, por exemplo, a tag `H1` e estilizar, porque isso vai passar a valer para todos os `H1` e a gente só quer alterar este primeiro. Já uma opção válida seria adicionar uma `ID` em cada um e estilizar. Outra opção seria criar uma classe única e adicionar essa classe aos dois elementos. As duas funcionam e estão corretas, mas eu queria apresentar uma nova pra vocês hoje, que funciona da seguinte maneira:

Bom, meu objetivo é selecionar os títulos dentro da `div` com o `id header`, certo? Lembrando que os títulos são filhos da `id header`, eu posso declarar o seguinte.

IHC29 - Fechando o cabeçalho

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/080964ef-0e37-45fa-9639-419dc1f617eb/index.html>

Declaramos a família da fonte na div Header, que funciona como div 'avô' das demais, mas ainda assim o efeito foi aplicado nas outras. Isso acontece pois os elementos internos herdam as características dos elementos externos.

Poderíamos fazer as alterações através da criação de uma Class, também poderia funcionar sem problemas.

Podemos salvar nossa tela em PDF, basta apenas clicar com o botão direito do mouse, selecionar 'imprimir' e salvar o PDF.

IHC30 - Área de contato

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/aa8bc054-0168-4c62-93e6-e947cd9f5cb5/index.html>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9cec35c7-39d8-40c8-a818-1c478c125605/style.css>

Na aula de hoje, nós vamos falar sobre layouts em CSS e vamos aprender 2 funcionalidades muito bacanas que nos ajudam a estruturar o layout do nosso site de maneira mais simples.

Então vou voltar pro VS Code, pro nosso projeto de curriculum. A gente só criou o header até agora, então o próximo passo é criar o corpo do nosso projeto. Vamos começar criando uma div chamada container, que vai ser pai de 2 outras divs. Uma chamada sidebar e a outra chamada content.

A sidebar vai ser essa parte lateral aqui e a div content vai ser esse corpo aqui de informações. Reparem que dentro da sidebar e dentro do content, a gente tem blocos de conteúdo, que a gente vai chamar de cards. Então agora eu vou criar

cada um desses cards que serão divs com a classe card inclusa, já que eu quero que todas essas divs tenham a mesma formatação de estilo.

Dentro da sidebar, eu quero criar 4 cards, enquanto na div content, eu vou criar 2 cards. Agora, eu vou começar a adicionar o conteúdo desses cards. O primeiro card, é o de contato, então eu vou criar um título h3 com o texto “Contatos” aqui, e eu estarei usando os títulos H3 porque lá em cima eu defini que o nome e a formação seriam H1 e H2, logo, seguindo a ordem de importâncias, esses títulos abaixo seriam menos importantes que os de cima, por isso eu usarei o H3. Não é regra, você poderia usar h2 ou h1, mas por questão de hierarquia, pra mim faz sentido fazer dessa forma.

Depois do título, eu vou criar uma lista, usando primeiro a tag UL e depois as tags LI pra cada item da lista. Como serão 3 itens, eu vou adicionar 3 tags LI. Agora, a gente precisa trazer os ícones que eu já mostrei pra vocês como fazer usando o fontawesome. Vou acessar então o site do fontawesome.com, fazer o login para acessar meus kits, vou selecionar o primeiro e copiar o código que a gente vai adicionar na head.

Agora, já podemos usar os ícones. Eu vou pegar aqui um ícone de telefone, um de e-mail e um de localização, lembrando sempre de marcar a opção free do lado esquerdo pra mostrar pra nós só os ícones gratuitos.

Depois, do lado dos ícones, vou escrever o e-mail, telefone e localização. Vou salvar agora e já podemos ver o resultado. Funcionou, mas ainda ficaram faltando alguns ajustes. Faltou acertar a cor, o alinhamento do texto e remover essas bolinhas. Então vou começar removendo as bolinhas, apenas nessa lista, nas outras eu quero que apareça, então eu vou atribuir um ID pra ela, como contactList. Depois vou subir aqui pro CSS e selecionar ela, usando hashtag contactList. Vou abrir chaves e adicionar a propriedade: `list-style-type: none;`

Com isso, a gente diz que não queremos mais as bolinhas lá no começo de cada item. Agora, vamos adicionar cor nos ícones. Reparem aqui que cada ícone é uma tag i e pra alterar o estilo dessa tag, eu posso dizer no CSS que todas as tags i que estão dentro da div contactList devem ser estilizadas com a mesma cor que eu já defini aqui pro header anteriormente através da propriedade color.

Beleza, pra fechar, precisamos alinhar os textos. Eles estão desalinhados porque cada ícone tem uma largura diferente, como vocês podem ver aqui. Pra corrigir isso, a gente pode simplesmente definir uma largura fixa para todos os ícones. Então vou escrever aqui que o width vai ser sempre 30 pixels.

Eu escolhi esse valor porque o ícone nunca vai passar desse tamanho, então sempre teremos uma folguinha do lado que deixa as coisas mais organizadas. Vou salvar e vamos testar aqui.

IHC31 - Demais áreas

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c7795d2e-6282-4417-a994-497b09134431/index.html>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6a61cf08-d14a-4df3-a43d-006ff83769ff/style.css>

O segundo card é o de skills. O terceiro card com formação acadêmica, estes criamos adaptando o que foi feito acima. Por último temos o card com links e uma lista com os links pro portfolio e pro linkedin, que eu vou setar o target como `_blank` para que eles sejam abertos em uma nova aba, como eu já expliquei pra vocês anteriormente.

Certo, com isso, nós terminamos a sidebar. Não se preocupe por enquanto com isso não estar alinhado na latera. Antes disso, a gente precisa criar o corpo do curriculum.

Dentro da div content, eu vou criar 2 cards, um com o sobre mim e outro com o experiências profissionais.

No card sobre mim, eu vou colar um texto genérico aqui só pra ocupar o espaço, mas você escreve aí o seu sobre mim. No card de experiências profissionais, eu vou criar 2 parágrafos, um pra cada experiência profissional.

A empresa e o período serão uma linha em negrito, e pra fazer negrito, a gente poderia usar uma tag `b` que serve pra negrito, mas já que estamos aprendendo CSS, eu vou criar uma tag chamada `SPAN`, que é usada quando eu quero estilizar um pequeno trecho dentro de outra coisa. Por exemplo, se eu tenho uma frase onde eu quero que apenas uma parte dessa frase tenha um tamanho de fonte ou uma cor diferentes, eu posso criar a div da frase e colocar a parte que eu quero destacar

dentro do span. Isso é útil porque diferente das divs, o span não ocupa a linha inteira, ele ocupa apenas o espaço do conteúdo que está dentro dele.

Então vou adicionar a tag span aqui e vou colocar nela uma class de professionalExperience. Lá em cima no CSS, eu vou definir que essa class professionalExperience vai ter a font-weight bold, que equivale a negrito.

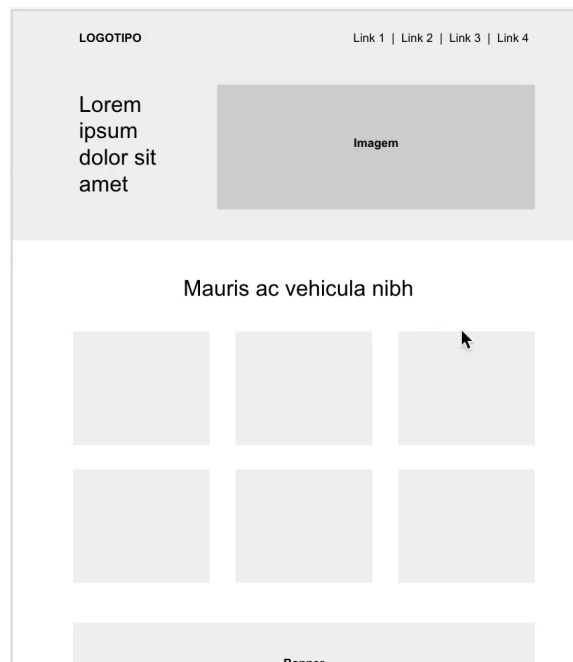
Ao invés de p aqui como container disso, eu poderia também usar divs e essa é a parte bacana do HTML. Não tem jeito absolutamente certo nem errado, é claro que tem o jeito mais eficiente ou mais indicado, mas em geral, é como um jogo de lego. Você pega os bloquinhos que são as tags e vai encaixando da forma que você quiser pra chegar num determinado resultado. Vou escrever aqui a primeira experiencia profissional e a data. Vou copiar esse paragrafo aqui e colar em baixo, e alterar as experiencias aqui só pra que a gente tenha mais uma experiencia profissional.

IHC32 - Ajuste no layout

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ab6f227a-4f3c-47d4-8e33-71654c23692c/index.html>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/32284ac1-939c-477f-844d-f560fbef8113/style.css>

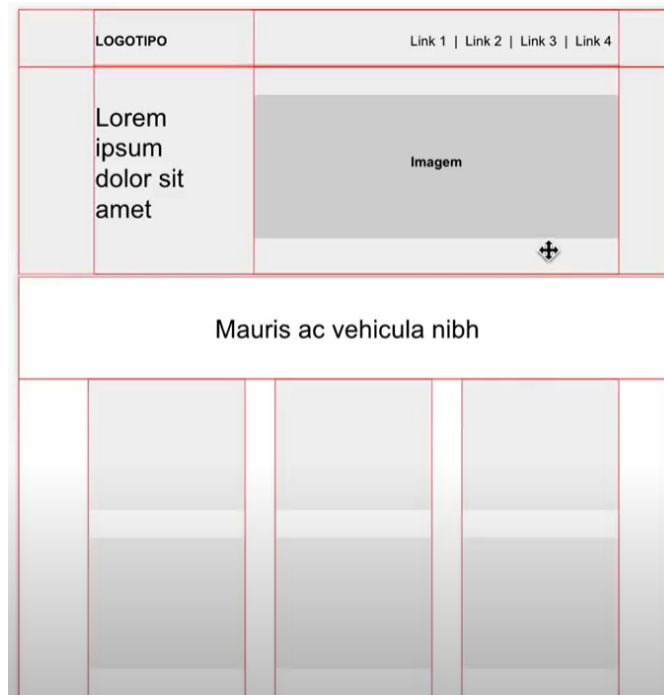
Bom, vocês que já navegam na internet há um bom tempo, devem ter reparado que os sites costumam ser estruturados mais ou menos assim, através de blocos ou seções onde o conteúdo é organizado e exibido para o usuário.



Todo site é organizado de certa forma através de blocos e esses blocos são organizados em linhas e colunas e essa organização se chama layout. Reparem no exemplo de como grande parte dos sites é estruturada.

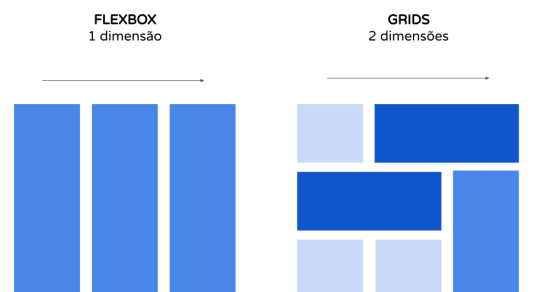
Já sabem que a gente pode dividir isso em divs e isso continua valendo, mas o que a gente ainda não falou, é que isso daqui pode ser dividido em linhas e colunas que vão ser os containers dessas divs que existem aqui.

Reparem que agora eu dividi o layout em linhas e colunas explicar as linhas e colunas do layout



E isso não invalida a questão das divs. Essas linhas e colunas, nada mais são do que divs que a gente estrutura para criar o esqueleto do nosso site de forma organizada.

Pra fazer isso, a gente vai usar as funcionalidades de grid e flexbox, que facilitam muito o nosso trabalho na hora de organizar isso tudo e o objetivo agora é a gente pegar esse conhecimento e implementar lá na nossa página de curriculum.

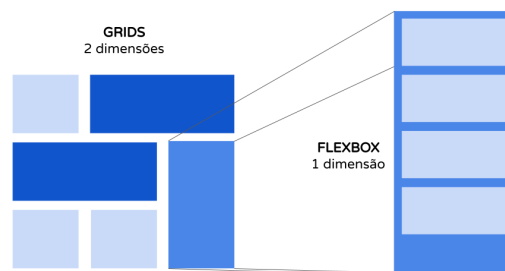


O conceito é o seguinte, reparem nestes dois exemplos aqui. No primeiro, temos o flexbox, que trabalha com apenas uma dimensão, ou seja, ou na vertical OU na horizontal. Ele é bastante útil pra que a gente consiga, por exemplo, definir que dentro de uma coluna vertical vão existir vários blocos que serão organizados um a cada linha.

E aí a gente junta isso com o grid, que permite que a gente organize tudo tanto na horizontal quanto na vertical. E aí, dessa forma, eu uso o grid pra montar a estrutura geral e, quando eu precisar de uma repetição em um sentido, eu aplico o flexbox. Uma analogia bacana pra tentar entender a relação entre os dois é a seguinte, se

Se você tem uma casa, o grid seria a organização dos cômodos da casa, como onde fica a sala, onde fica a cozinha, onde fica o banheiro.

Já o flexbox, seria a disposição dos móveis dentro de cada um destes cômodos. Então juntando os dois, a gente tem o seguinte:



Eu usei o grid pra criar as estruturas do layout e aqui dentro de uma div que faz parte do layout descrito pelo grid eu usei o flexbox para posicionar os elementos em uma única direção, que é a vertical.

Bom, isso vai ficar mais claro agora que a gente vai começar a por a mão na massa mesmo. A gente tem aqui nossa div container e dentro dela, nós vamos criar uma estrutura de grid.

Pra isso, eu vou aqui no CSS do container e definir a propriedade display com o valor grid. Depois, eu vou dizer como as colunas desse grid serão organizadas, através da propriedade grid-template-columns. Os valores dessas propriedades serão as medidas das nossas colunas, que podem ser uma porcentagem, um valor fixo ou mesmo o valor auto pra automaticamente definido. No nosso caso, serão duas colunas, a primeira com 30% de largura e a segunda com tamanho automático. Poderia botar 70% aqui também, daria na mesma.

Se a gente salvar agora, já vai ser possível ver que os elementos se organizaram aqui. Beleza, agora, a gente precisa organizar a parte que fica dentro de cada uma das colunas, tanto da sidebar quanto da content. Pra isso, a gente vai agora usar o flexbox.

Lembram da analogia do grid ser os cômodos da casa e do flexbox ser os móveis? Agora é a hora de arrumar os móveis.

Então eu vou selecionar aqui tanto a sidebar quanto a div content, usando o primeiro seletor hashtag sidebar virgula o segundo seletor que é o hashtag container. O display deles será flex. Como eu falei, o flex é unidirecional, então a gente precisa dizer a direção, usando a propriedade flex-direction e o valor column.

Por ultimo, eu quero especificar que entre cada elemento da coluna haja um espacinho de 20 pixels pra que eles não fiquem colados um no outro, como cada elemento desse é uma linha, um row em ingles, pra isso vamos usar a propriedade row-gap com o valor de 20 pixels, porque apesar da direção ser column cada elemento será uma linha dessa coluna.

IHC33 - Ajustes finos

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1b6a94d0-f801-4a6f-b694-477cae37d39e/index.html>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5d5205c4-3dd1-4e64-9623-c6028114a278/style.css>

O que falta agora são os ajustes finos. Já finalizamos a parte de layout, agora a gente só precisa dar uma última ajeitada no visual através das fontes e das bordas pra ficar igual o nosso objetivo. Como voces podem ver aqui, as fontes dos titulos são a mesma dos titulos lá de cima, já a fonte dos conteúdos, é a MontSerrat que a gente já pegou lá do google fonts. Pra definir corretamente as fontes, eu vou fazer o seguinte.

Primeiro, no estilo da classe card, vou dizer que a fonte será montserrat, ou seja, todos os textos vão ter a mesma fonte montserrat.

Depois do estilo do card, eu vou criar um novo estilo dizendo que todos os h3 dentro da classe card, vão usar a fonte Hubali através da propriedade font-family e também vão ser convertidos em uppercase, ou caps lock, usando a propriedade text-transform uppercase.

Por ultimo, quero que cada elemento com a classe card também tenha uma bordinha cinza, usando a propriedade border e setando os valores dela para 1 pixel solid e com a cor #eaeaea que é o cinza. Também vou dar uma arredondada nos cantos da borda, usando a propriedade border-radius com o valor de 10 pixels. Também quero que haja um espacinho entre a borda do card e o conteúdo, usando a propriedade padding com o valor de 20pixels.

Vamos salvar e ver agora como está, porque só falta agora uma coisinha. A gente não definiu o espaçamento entre as colunas do Grid, por isso ficou meio colado aqui. Então eu vou voltar aqui no estilo do meu container e definir a propriedade `grid-column-gap` com o valor 20 pixels.

Agora sim, podemos salvar e visualizar aqui o resultado. E pronto!

Criamos a nossa primeira página mais organizada e estilizada usando CSS. Seria bacana se você conseguisse agora modificar essa página com o que você aprendeu até agora, mudando planos de fundo, fontes, bordas e etc, pra você poder praticar e o conteúdo fixar melhor na sua cabeça.

IHC34 - Desafio do dia-a-dia

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/301733e4-47e0-4db2-86a4-5a8dc8c4e0e9/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6d505e45-020a-487e-9acb-59d5ce12e02a/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/aab7156f-40a3-411c-94ea-dc5df3df7b5a/LP_Portfolio.pdf

No processo comum do dia-a-dia existe uma série de trabalhadores que atuam na criação de um site, normalmente, um projeto chega aos programadores para que ele possa ser realizado e o site, com a arte proposta seja lançada.

O figma é muito utilizado para a criação de interfaces, é comum que sejam criados sites a partir de protótipos deles, como o que segue:



IHC35 - Ajustes iniciais

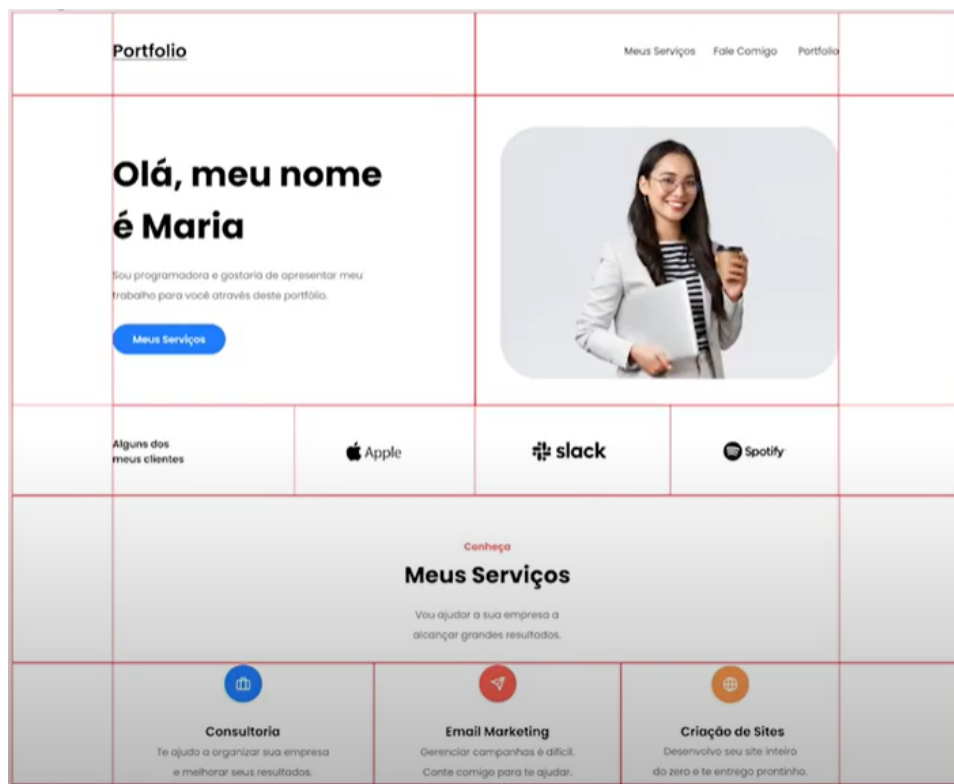
<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1d7a6c89-0e23-419d-b735-299c173240cf/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b364c020-ee45-4a49-8eb5-44794f11df07/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4cca5cdb-1f33-4a63-828d-779acb5d59b0/LP_Portfolio.pdf

Para iniciar, precisamos criar uma pasta, nesse caso, chamaremos de portfolio. Por padrão, chamaremos a página inicial de index. É necessário que criemos um arquivo .html.

A estrutura base pode ser feita com html:5 no vscode. Podemos ver que, no fundo, nosso layout pode ser dividido em linhas e colunas:



Vamos tratar as intersecções como as diferentes divs do site. Iniciamos abrindo uma tag style.

Começamos criando o espaço lateral. Então, criamos um body para fazer os ajustes de margem. Vamos criar uma classe chamada container, sendo essa com um display grid e dividindo em colunas.

Criamos as divs identificando a classe caso elas sejam utilizadas mais de uma vez no mesmo estilo, se ela tiver características únicas, o id se torna necessário para essa identificação.

O figma poupa trabalho do programador, pois a estilização já pode ser visualizada de maneira ágil, utilizando a inspeção na própria ferramenta. Mas é importante tomar cuidado com os estilos já pré-setados, eles não são dinâmicos.

IHC36 - Lista horizontal

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/915b952c-1133-42d0-a623-7036003ce8ec/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fe3fd7e9-e23d-4f8e-a01b-7d9f07c9147a/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/047f5928-3f06-4291-aae2-66acf7aad3ad/LP_Portfolio.pdf

Podemos criar listas horizontais! Iniciamos criando a lista propriamente dita, mas posso estilizar a lista específica definindo o list-style-type sendo none, e sem margin e padding. Então, podemos colocar o inline-block no display.

Por final, podemos colocar o alinhamento do texto de forma centralizada com os outros elementos, fazendo os ajustes de margem na lista.

Podemos inserir então os links dentro da tag, nos diferentes elementos da lista.

IHC37 - Título, botão e imagem

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/133d4d65-7264-4457-84e1-f494c81c9ce8/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/09a811bf-0d7a-401c-bea9-cd68f662e95b/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/42971899-0aa-a-4421-833e-1a1e64e49060/LP_Portfolio.pdf

Agora precisamos criar os elementos de título, botões e as imagens presentes na página. Em uma nova div podemos criar os títulos, e inserir um parágrafo nessa mesma div. Por fim, para criar o botão, podemos inserir um novo link, colocar um id padrão para que possamos estilizá-lo.

Novamente, a utilização do figma é muito bem-vinda para poupar tempo de quem cria as interfaces, trazendo as características do botão de forma ágil, mas alguns ajustes precisam ser feitos.

Para retirar o underline, basta estilizar os links com o text-decoration sendo none.

Para colocar as imagens, basta salvarmos a imagem presente no próprio figma na pasta do código como SVG. Para fazer isso basta criar um arquivo .svg dentro da pasta, clicar com o botão direito do mouse na imagem no figma, copiá-la como SVG e depois colar o código dentro do arquivo na pasta.

Após isso feito, podemos colocar ela através da tag img, selecionando a página pelo src. Assim é necessário ajustar a margem da div da imagem e deixar o tamanho ajustado com 100% da div.

IHC38 - Área dos clientes

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/68fee1e5-26b3-4c23-a68a-72bfe8fa65b2/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ffdbf7d7-d805-4f79-b82a-fd1915c51516/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e25a8c08-f7aa-40b3-b567-b991b9ec496d/LP_Portfolio.pdf

Para criar a área dos clientes fazemos o mesmo procedimento de cópia da imagem através do SVG. Criamos um arquivo .svg dentro da pasta, clicar com o botão direito do mouse na imagem no figma, copiá-la como SVG e depois colar o código dentro do arquivo na pasta.

Agora criamos um terceiro container, sendo ele composto por 4 divs, com duas diferentes formatações. A primeira é composta por um texto e as outras pelas logos dos clientes. Basta inserir as imagens e o texto e alinhá-los no centro da div.

Para alinhar o texto, basta setar a margem no topo e abaixo e então, estilizar o texto como o padrão fornecido pelo figma.

IHC39 - Meus serviços

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d7257d2b-0ad0-4615-9e33-1c58ac9c36f9/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6dc5086e-e9b7-4321-b91b-39817c14e752/LP_Portfolio.pdf

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b75692ca-aacf-4cb5-8dec-dd5748dc5ea4/img.zip>

Para criar a parte de 'Meus serviços' e 'Ganhe mais dinheiro' podemos criar uma mesma classe para ambos, visto que seguem o mesmo padrão. Podemos fazer uma seleção 'condicional', colocando que toda div específica que atenda por uma classe possa ser alterada. Por exemplo, h2.red, seleciona todo h2 da classe red.

É importante lembrar de colocar os elementos dentro da mesma div, assim o padrão do layout definido não fará com que se dividam em outros elementos. Além disso, pode-se fazer os ajustes de margem para que o texto fique como o esperado novamente.

Seguimos o padrão de criação de imagens SVG para criar os ícones necessários na div, seguindo os ajustes de margem e estilização do figma.

IHC40 - Área do formulário

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/479341af-9bc4-4c67-9b4b-9a9a59946cfc/LP_Portfolio.pdf

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1814b3ce-9a9a-4f08-a498-7b290ad14a3f/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0b80dc41-96b9-4928-8b16-01c4dd7e7b16/index.html>

A parte final do projeto consiste em duas divs, uma com uma imagem e outra com texto e formulário. Para início, criamos mais uma vez uma class container, identificando um id, então criamos as divs.

A primeira será a imagem, que seguirá o mesmo padrão das demais de SVG. Precisamos dizer que a imagem não terá o tamanho original, mas sim o tamanho da div especificada. Os textos já possuem o estilo criado, basta que sejam classificados nos estilos respectivos e depois preenchamos os textos.

O último ponto é a criação de um formulário. Novamente retornamos ao formspree, utilizando o method post, selecionamos os atributos do input e criamos um botão com o tipo submit.

Para fazer os ajustes podemos criar uma classe específica para os inputs e botões, fazendo as alterações de borda, fonte, texto e cor. Além disso, precisa-se ajustar o padding e a margin para deixar alinhado com o que se espera.

Assim também precisa ser feito com o botão, copiando o estilo setado no figma, fazendo os ajustes finos.

IHC41 - Portfólio

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c1bd399d-c1d1-4cd4-9b8e-73623d562e81/index.html>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0d60c9a1-7f20-4b35-98ae-1b28dc145ce2/LP_Portfolio.pdf

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/197e0a77-48d2-4853-b18e-85f857f0381f/img.zip>

Por final, temos a seção de portfólio, mais uma vez criamos uma nova div container para que possamos inserir os textos. Eles seguem padrões já setados antes. É necessário ajustar as margens para que o site siga o que é esperado do protótipo.

Além disso, essa parte possui mais três divs com imagens, novamente, basta criarmos novos arquivos SVG e chamá-los no site.

Além disso, é necessário que criemos as bordas entre as imagens e bordas.

IHC42 - Ajustes finais

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c8d0ecdc-6a3a-4520-b1bf-081f4792cf0e/LP_Portfolio.pdf

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/eeeb4b3b-16b8-4a78-bd87-b54993a5cfb5/img.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7f2ce94f-f8df-442f-9ffa-2b7271c5bd54/index.html>

O tempo de carregamento da página deve ser uma preocupação. Existem ferramentas que podemos utilizar para medir a velocidade das páginas, também deve ser uma preocupação de produto se preocupar com o carregamento dos conteúdos no site. É importante para a experiência do usuário que possa acessar o site o mais rápido possível.

O [PageSpeed Insight](#) é uma ótima ferramenta para analisar isso.

Para utilizar um link ancora (que direciona para a mesma página), é necessário fazer um chamado de href para o id daquela div específica, colocando uma hashtag na frente, #services, por exemplo.

IHC43 - Hospedando um site

Para que outras pessoas possam acessar os sites, é necessário hospedar ele através de uma plataforma de hospedagem. o [netlify.com](#) é um site simples para fazer isso.

Para conseguir hospedar, basta criar uma conta no site e realizar seu login. Assim que a conta for criada, pode-se simplesmente inserir seus arquivos de maneira ágil, inserindo o arquivo.

É possível personalizar o domínio do seu site por essa plataforma, se assim desejar.