

MÓDULO 3 – HTML5 E CSS3

O que é git e github?

Antes de tudo, vamos entender o que é repositórios locais e remotos. Ele basicamente é um local onde você guarda coisas.

O exemplo usado é eu criar um site e ir aprimorando ele com o tempo. Porém, se caso eu terminar uma versão e não gostar do resultado, não tem como eu voltar na versão anterior que estava melhor que a versão que eu digitei agora.

Para isso, se você não fez um backup para salvar as versões anteriores, você perde ela, so que também não é adequado que eu fique salvando e cuidando para salvar toda a hora, se caso esquecer uma hora, você perdeu uma parte do projeto. Para isso, baixamos um gerenciador de versões, assim ele volta versões e salva todas as versões. **O nome dele é Git. Ou seja, ele é um software que vai manter versões do código salvas somente dentro do seu computador.**

Agora, como foi dito, ele salva somente dentro do seu computador, e se caso acontecer alguma eventualidade com o mesmo, você perde igualmente todo o conteúdo armazenado dentro do git. Para salvar ele em um local onde quem sabe o seu cliente queira acessar da máquina dele por exemplo, **vamos mandar ele para um servidor remoto, que pode ser acessado de outros lugares, chamado então de GitHub.**

Ou seja, **o repositório local é o Git, e o repositório remoto é o GitHub.**

Segunda Aula – Instalação do git e do github no pc.

Esta foi a aula em que instalamos o GitHub dentro do nosso PC.

Terceira Aula – Criando a conta para acessar o github.

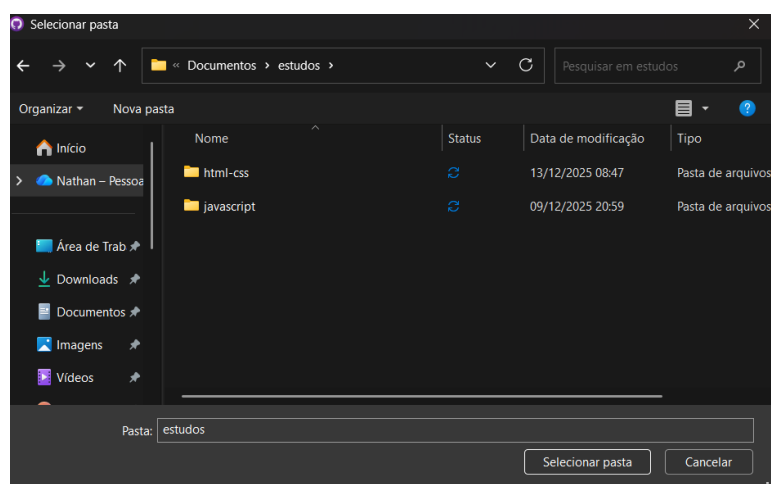
Nesta etapa, foi criada uma conta dentro do GitHub.

Quarta aula – Criando o primeiro repositório.

Para criarmos o nosso repositório local (somente dentro da nossa máquina), devemos entrar no GitHub Desktop e clicarmos em **“Create a New Repository in you local drive”**.

Após isto, devemos nomear o nosso repositório com o nome EXATO da pasta em que está localizado o nosso repositório, ou seja, temos em nosso computador a nossa pasta “estudos”, e dentro dela temos as duas pastas “html-css” e “javascript”. Então, se quisermos criar um repositório para estas pastas, devemos nomear o nosso repositório de “html-css” ou “javascript” (o que foi feito nesta aula).

Depois de nomearmos o nosso repositório em **“name”**, escrevemos um comentário acerca do conteúdo localizado dentro do mesmo. Em **“Local path”**, indicamos a máquina onde localizar a pasta que compõe este repositório. Esta pasta será a nomeada de “estudos”, pois ali dentro se encontra os nossos cursos. O nome da pasta desejada não pode ser encontrado dentro do direcionamento, ou seja, paramos em estudos:



Marcamos após isso a caixa escrita com **“Initialize this repository with a README”**. Em **“Git Ignore”** nos selecionamos a linguagem em que nós vamos programar, para nos podermos ignorar certos arquivos, o que não é o caso. E para finalizar, em **“license”** nós podemos escolher qualquer licença. O indicado pelo Professor Gustavo Guanabara é marcar **MIT**. Após isso, basta clicar em **“Create repository”** e então aguardar alguns instantes para encontrarmos todos os nossos arquivos.

Com este processo finalizado, nos terminamos de criar o nosso Repositório LOCAL, ou seja, na nossa máquina. Tanto que se entrarmos em nossa pasta (neste caso “html-css”), veremos diversos itens novos, e importante ressaltar a pasta oculta **“.git”**, responsável por manter cópias dos meus projetos.

Agora, depois de criarmos o nosso repositório local dentro do GitHub Desktop, temos a sugestão, dada pelo próprio GitHub Desktop, para publicarmos o nosso repositório local para o nosso GitHub. Para isso, basta clicar em **“Publish Repository”**. Se mantivermos o **“Keep this code private”**, ninguém poderá ver o seu repositório, a não ser você. E também, com isso marcado, não poderemos publicar os nossos sites. Então, para este caso, iremos desmarcar essa box. E depois, somente publicar.

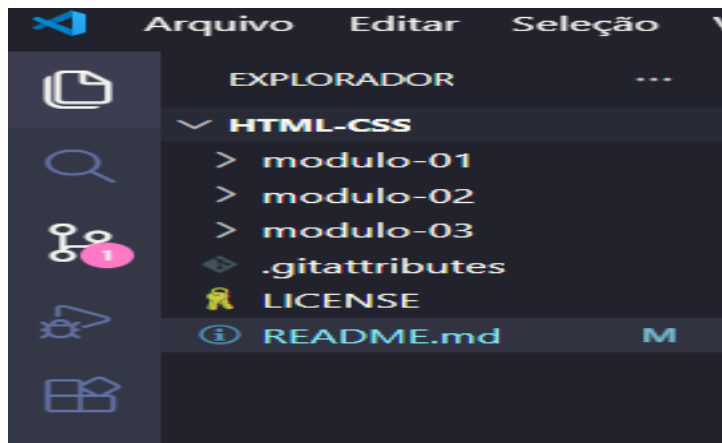
Nos lançamos agora somente OS CÓDIGOS dos nossos projetos, e não as páginas em si, já que elas serão lançadas mais tarde através do GitHub Pages. Agora então nos já temos um repositório local (GitHub Desktop) e também um repositório remoto (GitHub).

Quinta aula – Gerenciando repositórios dentro do GitHub

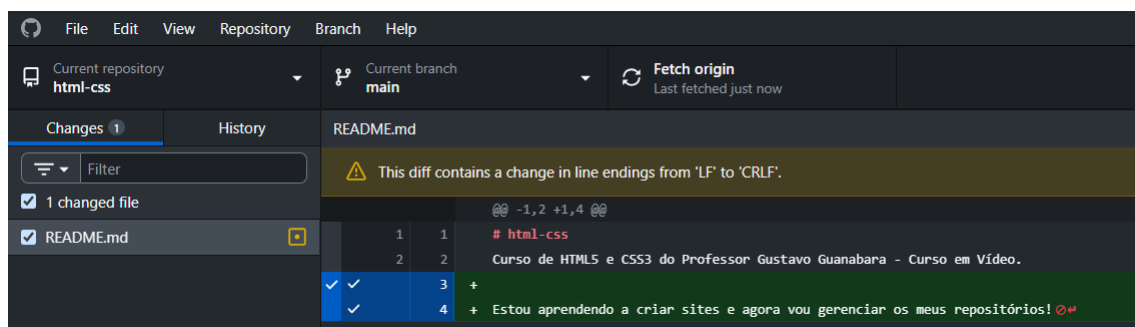
Para vermos quais repositórios estão atualmente ativos em nosso servidor local, basta verificarmos a aba **“Current repository”**.

Se clicarmos em **“Open with Visual Studio Code”** abriremos o nosso repositório inteiro dentro do mesmo, podendo ver todos os códigos e itens contidos dentro de nossos repositórios.

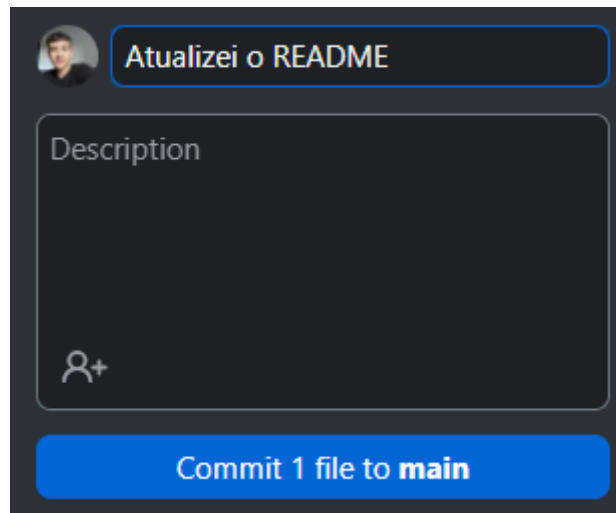
Quando abrimos o nosso repositório dentro do Vscode, e fizermos alguma alteração, como neste caso, incluindo mais coisas dentro do README, veremos que aparecerá um ícone indicando que temos alterações dentro do repositório:



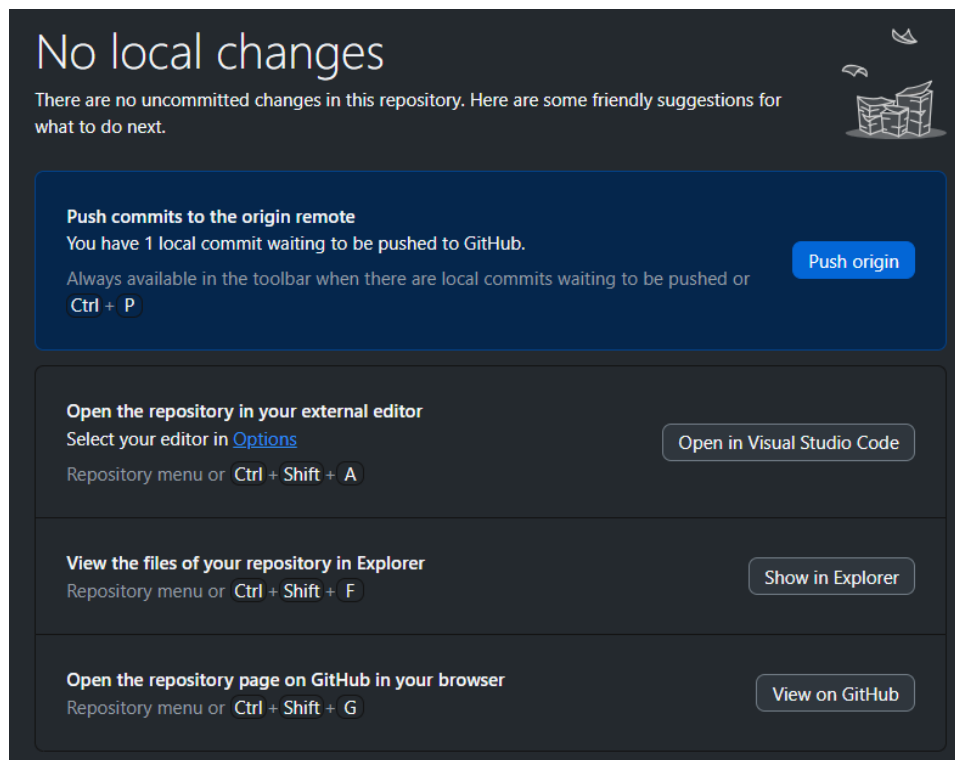
Quando retornarmos para o nosso GitHub Desktop, ele já identificará as alterações pendentes que foram feitas dentro do Vscode. E para vermos tudo o que foi alterado, basta ir no canto da página do GitHub Desktop e verificar os itens pendentes. Para ver a real alteração feita (Antes x Depois) basta clicar em cima de uma delas e verificar:



Para que atualizemos o nosso repositório local com estas novas alterações que já estão pendentes, devemos realizar um processo nomeado de **“commit”**, colocando um título, com neste caso “Atualizei o README”, e, se quiser, uma breve descrição. Depois, basta clicar em **“Commit to main”**, e assim, atualizamos o nosso repositório LOCAL:



Agora que estas alterações foram atualizadas dentro do meu computador (Repositório Local), devemos também atualizar o nosso GitHub (Repositório Remoto). Para isso, basta clicar em “**Push Origin**”:



E assim, atualizamos com os novos conteúdos tanto o repositório local, quanto o repositório remoto.

CRIAR NOVO REPOSITÓRIO: Para criarmos um novo repositório, temos que clicar **CTRL + N** ou **“FILE” + “NEW REPOSITORY”**. E após isto, basta seguir o mesmo processo citado acima para a criação de novos repositórios.

CRIAR REPOSITÓRIO INEXISTENTE: Vamos supor que eu quero começar a aprender MySQL, e para isso, preciso criar uma nova pasta dentro de estudos nomeada de “MySQL”, e ela futuramente irá ser um repositório. Eu não preciso criar esta pasta lá dentro de estudos para criar este repositório, basta eu voltar no GitHub Desktop, criar um novo repositório, vou nomear ele de “mysql”, dar a descrição, jogar na pasta de estudos, iniciar com README e criar. Depois de criar ele, basta publicar. E assim criamos o nosso novo repositório.

ALTERAR ESTADOS DE REPOSITÓRIOS: Ao invés deletarmos um repositório, podemos simplesmente deixar ele privado, assim, impedindo que outras pessoas possam ver o meu repositório, e somente eu tenha acesso. Para isso, entramos dentro do nosso GitHub, entramos no repositório que queremos deixar privado ou até mesmo público, entramos em **“Settings”**, rolamos até **“Danger Zone”** e para finalizar **“Change Repository Visibility”**.

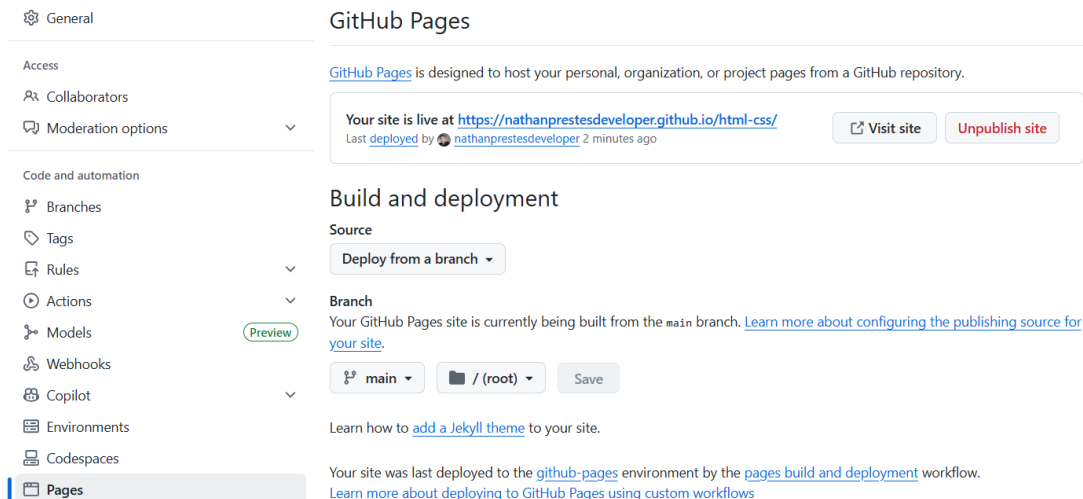
EXCLUIR REPOSITÓRIOS: Para excluirmos um repositório, devemos primeiramente deletar ele do nosso Repositório Local. Para isso, entramos no repositório que queremos apagar, e clicamos em **“Repository”** e depois em **“Remove”**. Agora que apagamos ele localmente, devemos apagar o Repositório Remoto. Para isso, vamos dentro do nosso GitHub e entrar no nosso repositório que apagamos. Vamos em **“Settings”**, vamos até a **“Danger Zone”** e clicar em **“Delete this Repository”**. Agora excluimos totalmente este repositório de todos os lugares possíveis.

Sexta aula – Hospedando páginas gratuitamente dentro do GitHub Pages

Agora nós iremos aprender a colocar os nossos sites no ar. Nesta aula, iremos rodar todos os nossos códigos.

Em primeiro lugar, vamos abrir o repositório de html-css que é o que queremos colocar no ar. Vamos em **“Settings”** até a parte **“Pages”** que por

padrão, não está habilitada. Na seção **“Branch”** onde está escrito **“None”** nós mudamos para **“Main”** e deixamos o **“Root”**. Após isso, nos habilitamos o GitHub Pages SOMENTE PARA ESTE REPOSITÓRIO. Agora, nesta seção, podemos localizar o nosso link de usuário para o GitHub Pages:



The screenshot shows the GitHub repository settings page for GitHub Pages. On the left is a sidebar with navigation options: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages (which is highlighted). The main content area is titled 'GitHub Pages' and includes a description, a live site link, deployment status, and build/deployment settings.

General

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://nathanprestesdeveloper.github.io/html-css/>
Last deployed by [nathanprestesdeveloper](#) 2 minutes ago

[Visit site](#) [Unpublish site](#)

Build and deployment

Source
Deploy from a branch

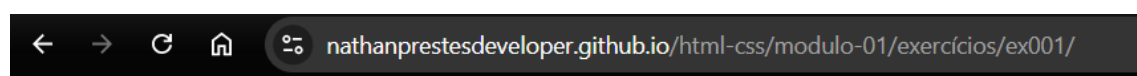
Branch
Your GitHub Pages site is currently being built from the `main` branch. [Learn more about configuring the publishing source for your site.](#)

`main` `/ (root)` [Save](#)

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the `github-pages` environment by the `pages build and deployment` workflow. [Learn more about deploying to GitHub Pages using custom workflows](#)

Por padrão, ao clicarmos em **“Visit Site”**, abriremos o nosso arquivo README. Para abrirmos um exercício em específico, precisamos dar o endereçamento EXATO do arquivo com o link do GitHub Pages dentro do repositório, como por exemplo, o do exercício 01:



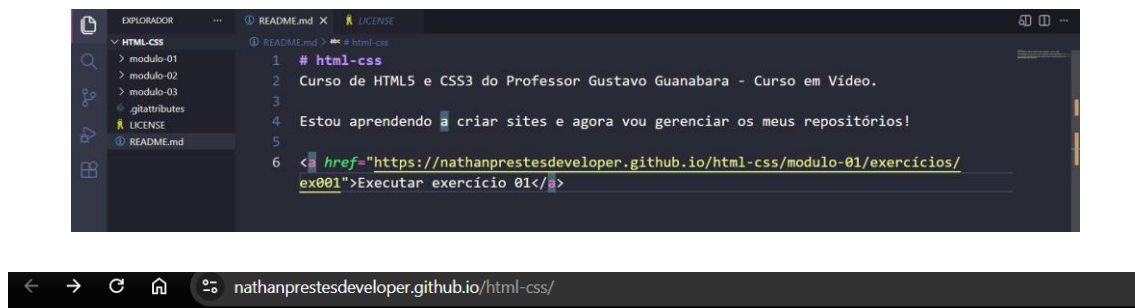
Olá, Mundo!

Esse é o meu primeiro documento HTML! Estou muito feliz!

Este é um momento único! Estou criando um site!

Agora nós podemos mostrar para as pessoas os nossos sites. O Professor Gustavo Guanabara deu a dica de colocarmos links direcionando os nossos

exercícios dentro do README. Para isso, entramos no nosso arquivo e digitamos a nossa tag link endereçando o nosso exercício (tem que digitar manualmente porque ele não tem autofill) e salvar no Repositório Local e depois mandar para o Repositório Remoto:



Entretanto, temos uma outra maneira mais simples de fazermos isso. Além de ser mais simples, também é um pouco mais profissional. Vamos usar como exemplo para esta explicação o Projeto Android que fizemos no módulo 2. Basicamente, iremos criar um repositório específico para este site. Nós vamos copiar os arquivos que se referem ao site do Android da pasta de estudos, vamos criar um novo repositório no nosso GitHub Desktop e nomear ele de “**projeto-android**”. Após isso, vamos entrar na pasta que ele criou dentro dos documentos e colar os nossos arquivos copiados referentes ao site do Android. Vamos renomear o arquivo principal para “**index**” (sem o html no final) e publicar ele no nosso Repositório Remoto. Após isso, fazer o processo de publicação das páginas e temos o nosso Projeto Android no ar com um ar mais profissional de maneira gratuita, possuindo uma URL própria, e assim, facilitando o compartilhamento do site publicado no GitHub Pages.

Professor dá a dica de gerar um QR code para o seu site. Vale ressaltar que o GitHub Pages somente suporta as seguintes linguagens: html, css e javascript (na página). Somente isso.

Sétima aula - Manutenção em sites hospedados no GitHub

Pages

Nesta aula ele fez a correção de um bug que aparentemente eu não necessito corrigir, já que ele já está corrigido dentro da minha máquina.

Oitava aula - Recursos Sociais do GitHub

Podemos seguir pessoas dentro do GitHub, e também, repositórios. Basta ir na caixa de “**Search or jump to**” depois você escreve o usuário e busca em todo o GitHub. Vai em user e clica em seguir.

Ele ensina várias formas de seguir pessoas através do github.com/explorer, mas não acredito ser necessário agora.

Temos também as chamadas “**issues**”, que são comentários a respeito de problemas que pessoas podem te ajudar ou até mesmo você ajudar. Ela também pode ser resolvida por outras pessoas, e você verificar se não é o mesmo que você.

Nona aula – Clonando repositórios no GitHub

Podemos clonar repositórios em duas situações: clonar os próprios projetos para trabalhar em outro computador, seja da faculdade, escola, de um amigo ou clonar o repositório PÚBLICO de qualquer pessoa.

Vamos usar de exemplo para esta aula a suposição que queremos clonar o nosso Repositório de html-css. Basta entrar nele e acessando ele, clicar em “<> **Code**” e você tem duas opções: baixar ele em um arquivo zip ou abrir no GitHub Desktop. Abrindo no GitHub Desktop, ele clona todo o projeto que você selecionou e você pode trabalhar nele e fazer o que bem entender. Baixar ele zip é mais complicado. Mais fácil é abrir ele com o GitHub Desktop. Ao clicar em “**Open with GitHub Desktop**” ele vai perguntar onde você quer salvar na tua

máquina, e depois clicar em **“Clone”** e ele será baixado para o computador, e também já pode ser aberto com o Vscode.

Décima aula – GitHub em vários PC's

Vamos supor que eu vou trabalhar em um computador que não é o meu. Eu quero usar, baixar, trabalhar e sair sem deixar nenhum rastro meu dentro desta máquina. É isso que vamos aprender durante esta aula.

Os aplicativos que eu preciso ter nesta máquina nova são: Um navegador web (Google Chrome), o Visual Studio Code, o Git e o GitHub Desktop. Vamos abrir o GitHub Desktop, clicar em **“File”**, **“Options”** e entrar na área (entra automaticamente) **“Accounts”**. Feito isto, clica no primeiro link para o *GitHub.com*. Quando você acessar a sua conta, automaticamente ele vai te sugerir os teus repositórios locais. Após entrar no GitHub Desktop, vamos clicar em **“File”** de novo, **“Options”** e clicar em **“Git”**. Logo após isto, clica nas duas caixas de texto (so clica em cima das duas, do usuário e do email), e depois, só clicar em **“Save”**. Agora vamos selecionar qual repositórios queremos trabalhar, depois vamos clicar em cima dele e fazer um clone dele. Você cria uma pasta nova em um lugar que você quiser.

Você faz o trabalho que quiser fazer agora, altera itens, faz tudo que precisa. Retorna para o GitHub Desktop, faz o seu *commit* (Atualiza o Repositório Local) e lança para o repositório remoto.

Está na hora de ir embora. Primeiro fechamos tudo, e deixamos somente aberto o GitHub Desktop. Dá um *remove* no repositório que você trabalhou, e joga para a lixeira. Mas ainda não acabou. Clica em **“File”**, **“Options”**, **“Sign Out”**. Nesta mesma aba, vai em **“Git”**, apaga os dois itens e salva. E para finalizar, apaga a lixeira.

Décima primeira aula – Download das imagens do capítulo

As imagens que serão baixadas não são as imagens que compõem o conteúdo, e sim fazem parte do visual do site.

Foram baixadas e adicionadas dentro da pasta referente ao módulo três do curso.

Décima segunda aula – Colocando uma imagem de fundo no seu site

A tag principal para colocarmos uma imagem no fundo de algo, seja uma div, como feito no exercício, ou até mesmo no fundo do site, é: *background-image: url (local da foto desejada);*

Décima terceira aula – Imagens que se repetem no fundo do site

Podemos colocar literalmente uma URL de uma imagem como fundo. Para isso, basta copiar uma URL qualquer e colar dentro dos parênteses do código.

Background-size: A imagem SEMPRE, por padrão, irá ser carregada no seu tamanho padrão, a não ser que definamos com esta tag o seu real tamanho. Esta tag define por padrão o tamanho da imagem no fundo

Background-repeat: Por padrão, o background-image repete automaticamente a imagem a depender do seu tamanho, se mantendo fixa no fundo do site. Quando colocamos esta propriedade no nosso site, nos damos uma ordem para não repetir esta imagem. Temos também o repeat-x (Horizontal) e o repeat-y (Vertical) tendo sempre como ponto de partida o canto superior esquerdo (para todo o tipo de repeat).

Décima quarta aula – Configurando a posição da imagem de fundo do site

Por padrão, as imagens SEMPRE irão começar no canto superior direito. Para dizer onde você quer que a imagem comece, basta usar a tag: *background-position: coluna linha;* (ex: *background-position: center right*)

RECAPTULANDO TUDO:

Background-size: A imagem SEMPRE, por padrão, irá ser carregada no seu tamanho padrão, a não ser que definamos com esta tag o seu real tamanho. Esta tag define por padrão o tamanho da imagem no fundo

Background-repeat: Por padrão, o background-image repete automaticamente a imagem a depender do seu tamanho, se mantendo fixa no fundo do site. Quando colocamos esta propriedade no nosso site, nos damos uma ordem para não repetir esta imagem. Temos também o repeat-x (Horizontal) e o repeat-y (Vertical) tendo sempre como ponto de partida o canto superior esquerdo (para todo o tipo de repeat).

Background-position: Define onde a imagem irá iniciar e permanecer, independentemente do tamanho ou tipo de tela.

Décima quinta aula – Mudando o tamanho da imagem de fundo do site

Para definirmos que uma imagem ocupe toda a dimensão alinhada ao seu tamanho no fundo do nosso site, usamos uma tag específica: *background-size*; agora vem o pulo do gato, iremos definir o tamanho da imagem, para que ela fique bonita no site selecionando uma dessas duas opções: "contain" que faz com que ela fique 100% exibida, porém fica com espaço preto de fundo, então neste caso não resolve. E temos o "cover", que tem a função de cobrir tudo, independentemente do tamanho da tela, fazendo o necessário, como cortar uma parte dela. Também se faz necessário ajustar a View Port da tela para cem por cento.

Entretanto, este modelo infelizmente não consegue se ajustar a todo o site se caso o conteúdo for muito longo. Por exemplo, no exercício, que é curto, ela consegue se adaptar bem com dois parágrafos de lorem, entretanto, se

colocarmos mais conteúdo, a imagem simplesmente acaba, fazendo com que o resto do site fique com o fundo preto. Isso nós iremos resolver na próxima aula.

Décima sexta aula – Background attachment e shorthand

Como dito anteriormente, se usarmos somente o *background-size* não teremos o nosso resultado esperado (o cliente rodar a página com uma foto padrão de fundo, porque quando acabar o real tamanho da imagem e o conteúdo do site for maior que ela, ficara bordas pretas). Para isso, nós usamos outra tag: *background-attachment: fixed;* ou seja, a imagem não vai scrolar com o conteúdo e chegar uma parte que ela termina, ela vai ficar parada no fundo. Outra coisa interessante que não foi citada: definir *max-width* ao invés de *width*, porque assim, a gente define um tamanho máximo para o conteúdo, e quando for para celular, ele se ajusta, já que não foi delimitado um valor ESPECÍFICO para a largura do conteúdo, e sim, um valor máximo.

Existe uma shorthand para juntar todos esses backgrounds em uma só declaração. Usar nesta ordem: (verifique o código)

- Background-color
 - Background-image
 - Background-position
 - Background-repeat
 - Background-size (não está aplicado, não usa, tira ele e coloca separado)
- Background-attachment

Décima sétima aula – centralização vertical de caixas

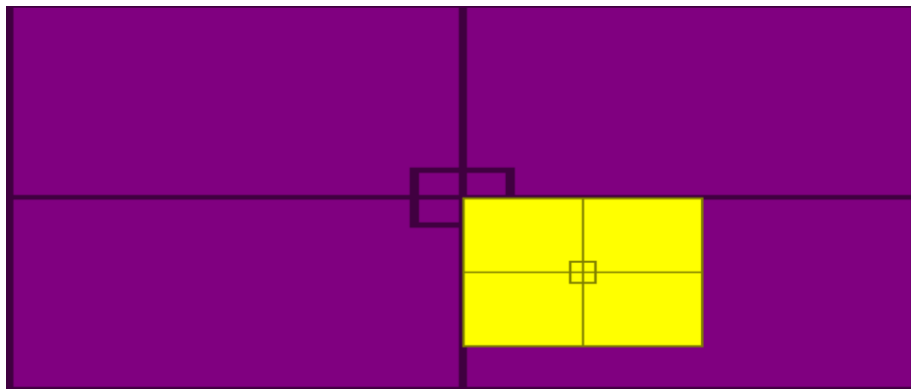
Para usarmos de exemplo, temos uma *section* que dentro dela tem um *article*. Para alinharmos horizontalmente este item no centro, basta darmos um *margin: auto;* entretanto, para alinharmos o mesmo conteúdo verticalmente, a coisa muda.

Primeiramente, no nosso container (Section), vamos colocar uma *position: relative;* (todo o elemento absoluto dentro de mim irá me usar como referência),

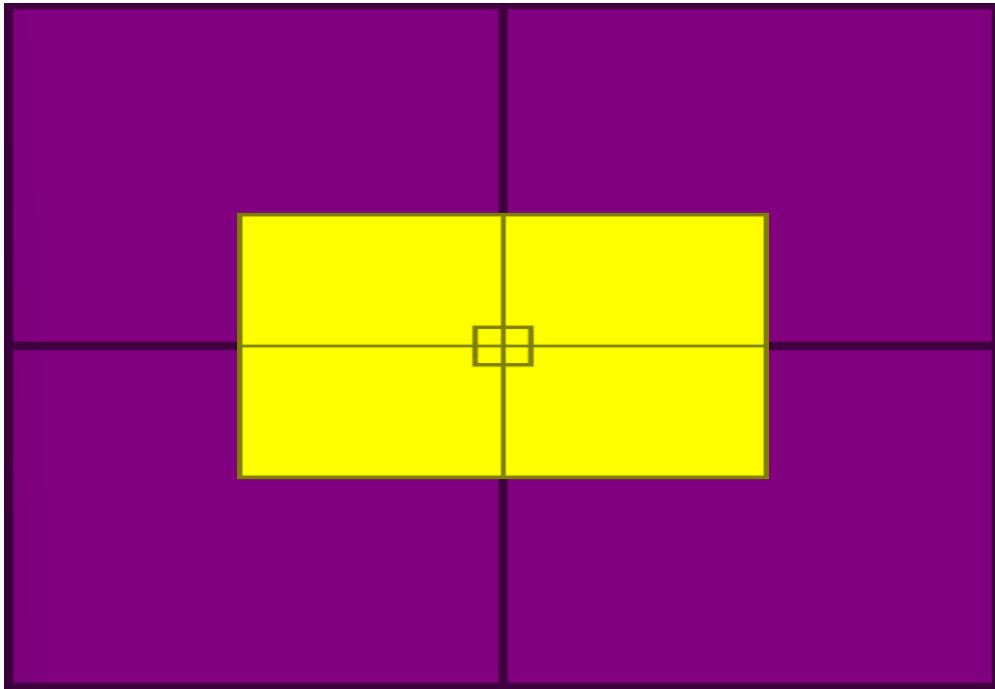
mas estamos sendo mais explícitos. No nosso conteúdo (article), vamos mudar isso, colocando uma *position: absolute*; (Não uso mais como referência a página, e sim o Section)

A regra é, fora é relativo e dentro é absoluto. Mas só com isso, não muda nada.

Quando colocamos um elemento na posição absoluta (no caso do conteúdo), podemos modificar duas propriedades dele: left e top. Eles também podem ser definidos com porcentagens. Quando colocarmos 50% e 50%, estamos dizendo para o código: Coloque o canto superior direito do article no meio do section.



Ele ocupou os 50% colocados, porém levando como o centro do container a referência. Basta então, para colocarmos eles alinhados, usar uma tag de transformação com translação para ajustarmos exatamente no meio os dois. As tags são: *transform: translate (V1, V2)*. Se o V1 for positivo, mexe para direita. Se o V1, for negativo, ele mexe para esquerda. Logo, se o V2 for negativo, ele mexe para cima, se for positivo, mexe para baixo. Porém, nós não vamos usar valores em pixels, e sim, em porcentagem, em valores negativos, já que o v1 negativo mexe para a esquerda e o v2 negativo mexe para cima, com os mesmos valores de 50% para os dois, os mesmos que foram usados no *left* e no *top*. Aqui basicamente estamos dizendo para o código: mexa 50% da largura do article para a esquerda, e mexa 50% da altura do article para cima, assim, compensando o que faltou na nossa última tag.



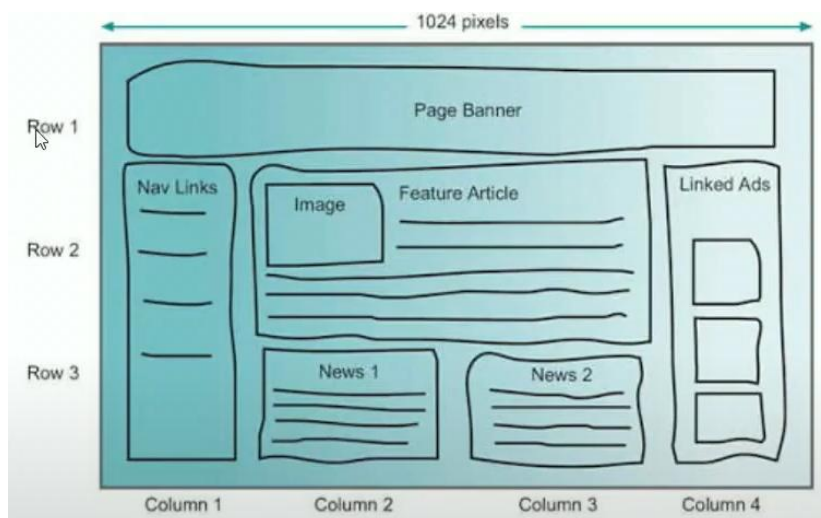
Em resumo, o navegador não entende como centralizar um item, então usamos um truque lógico com ele.

FRASE PARA GRUDAR NA CABEÇA: Primeiros 50% posicionam o canto, e o translate com os dois 50% negativos corrigem o corpo.

Décima oitava aula – Ainda podemos usar tabelas em HTML?

O objetivo desta aula é mostrar os usos de uma tabela dentro de um site e suas funções. Ensinar como usar corretamente as tabelas e saber quais itens realmente colocar dentro. Professor diz: “Você não pode usar tabela para criar a estrutura do seu site. ”

Professor pesquisa “população por estado” para mostrar uma tabela. Professor diz que antigamente, no html4, se criavam muitos sites tendo como base das estruturas tabelas.



Antigamente se usava esta base para fazer sites. Usar tabelas funciona PARA MOSTRAR DADOS, mas não para estrutura de site!

Décima nona aula – Sua primeira tabela em HTML

Vamos usar para base da nossa primeira tabela, uma desenvolvida no Excel bem simples:

A1	B1	C1
A2	B2	C2
A3	B3	C3
A4	B4	C4

Por padrão, tabelas não tem linhas!! Todas as tabelas simples em html tem a seguinte hierarquia: TABLE > TABLE ROW > TABLE HEADER > TABLE DATA. (Tabela > Linha de tabela > Cabeçalho de Tabela > Dados de tabela). No caso da tabela simples acima, não possui cabeçalho de tabela, só dados.

Fazendo então uma verificação da tabela, temos quatro linhas. Dentro da primeira linha, temos três dados (A1, B1, C1). Então a ordem é abrir um Table, dentro de table abrir um Table Row (TR) e um Table Data (TD) para colocar a primeira linha da tabela, e assim, sucessivamente


```

<table>
  <tr> <!--Primeira linha (Tr) + conteúdos da primeira linha (Td)-->
    <td>A1</td>
    <td>B1</td>
    <td>C1</td>
  </tr>
  <tr> <!--Segunda linha (Tr) + conteúdos da segunda linha (Td)-->
    <td>A2</td>
    <td>B2</td>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>

```

Com esse código, obtemos este resultado:

Minha primeira tabela

```

A1 B1 C1
A2 B2 C2
A3 B3 C3
A4 B4 C4

```

Para essa tabela se parecer mais com uma tabela, nós marcamos as suas linhas, suas bordas, dentro de *style*.

```

<style>

    table {
        width: 400px;
    }

    td {
        border: 1px solid black;
    }

</style>

```

Com isso, obtemos isso:

Minha primeira tabela

A1	B1	C1
A2	B2	C2
A3	B3	C3
A4	B4	C4

Dentro do código foi tirado este espaçamento das linhas e umas brincadeiras com a formatação da tabela. O W3C, no html5, coloca como opcional o fechamento dessas tags, mas é melhor fechar. (`</td>`, `</tr>`). Se os dados não forem colocados na sua hierarquia os itens podem ficar desalinhado.

Depois, no style, é modificada a distância das linhas com as tags: *border-collapse: collapse*.

Vigésima aula – Alinhando o conteúdo em tabelas

Para alinhar os conteúdos da tabela horizontalmente, basta abrir um seletor para “**td**” e usar o “**text-align**”. Agora para alinharmos verticalmente, usamos outra tag: “**vertical-align**”, e aí, é só fazer os seus testes.

Vigésima primeira aula – Aprendendo a trabalhar com tabelas grandes

Para trabalhar com tabelas com mais dados, sendo assim, maiores, elas precisam ser organizadas semanticamente assim como o corpo de uma estrutura de html: cabeçalho (definido por *thead* (pode conter *th* (título), *tr* (linha) ou *td* (dado)) para o título da tabela, *tbody* (pode conter *th* (título), *tr* (linha) ou *td* (dado)) e por último o *tfoot* (pode conter *th* (título), *tr* (linha) ou *td* (dado)).

Mantendo uma semântica de organização, podemos trabalhar com tabelas muito maiores sem se perdermos dentro do nosso código.

Vigésima segunda aula – Caption e Escopo de títulos em tabelas

Na última aula, ficou faltando apenas um conteúdo para tabelas antes de começar a falar de Caption e Escopo, será visto agora (Caption).

Caption é a legenda da tabela, geralmente acima do *thead*, e nele, você coloca somente um texto, para a legenda da tabela. SEMÂNTICAMENTE, o ideal é colocar os itens nos seus devidos lugares.

O assunto principal dessa aula são os Escopos dos *th*. Escopo é reconhecido pela tag *scope* é responsável por uma parte de SEO (mecanismos de busca do google) para o site ser mais sugerido. Dentre as opções de *scope*, é sugerido quatro opções (porém, não será falado dos *groups*, somente mais para frente): *col*, *colgroup*, *row* e *rowgroup*. No exemplo da tabela, onde temos estados e população, sabemos que são uma coluna referente aos itens abaixo delas, então, usaremos a tag *col*. No rodapé, que comporta o total de habitantes,

como ele é referente ao conteúdo localizado na mesma linha, usamos o `row`.
Veja na imagem:

```
<th scope="col">Estado</th>  
<th scope="col">População</th>
```

```
<th scope="row">Total da população</th>
```

Populações por estado no Brasil (2025)	
Estado	População
São Paulo	46 081 801
Minas Gerais	21 393 441
Rio de Janeiro	17 223 547
Bahia	14 870 90
Paraná	11 890 517
Rio Grande do Sul	11 233 263
Pernambuco	9 562 007
Ceará	9 268 836
Pará	8 711 196
Santa Catarina	8 187 029
Goiás	7 423 629
Maranhão	7 018 211

Roraima	738 772
Total da população	213 421 037

Vigésima terceira aula – Efeito Zebrado na tabela com HTML e CSS

O efeito zebrado em uma tabela tem a finalidade de auxiliar a leitura da tabela, para que os olhos corram e não se percam no decorrer do curso.

Para aplicar isto a uma tabela, usaremos:

```
tbody > tr:nth-child(2n) {
    background-color: lightgray;
}
```

O que foi feito é abrir um seletor específico para as linhas que são filhas diretas do *tbody* e especificar que de duas em duas, seja colocado o fundo *lightgray*. Pode-se também, ao invés de colocar *2n*, colocar *odd* (ímpar em inglês), que irá pintar as linhas ímpares da tabela, ou também *even* (par em inglês), que pintará as linhas pares, par começar pintando a primeira linha colocar *2n-1* e ir adaptando conforme a necessidade. Pode-se abrir também dois seletores, um para números ímpares (*odd*) com uma determinada cor de fundo, e outro seletor para os números pares (*even*), com outra cor de fundo. Ou também dois seletores, um para *2n* e outro *2n-1*. Enfim, fazer as adaptações necessárias.

Vigésima quarta aula – Cabeçalho fixo em tabelas grandes

Este recurso não está funcionando perfeitamente em todos os navegadores, porém o professor não quer deixar passar.

Basicamente consiste em colar o cabeçalho quando a tabela for grande e exigir scroll do usuário. As colunas que denominam os itens abaixo, fica como se fosse um efeito Parallax na tabela, enquanto o usuário scrolla.

Para fazer isto, é necessário primeiramente, no seletor *table* colocar uma posição relativa. Depois, abrir um seletor exato do caminho para se chegar. Definir uma posição *sticky* (colado), e depois colocar *top: 0;* (Ou seja, estou dizendo que vai colar na parte zero. Assim, se for notado, ao rolar a tela as palavras irão ficar atrás da tabela enquanto rola, pois foi colado AS PALAVRAS da coluna. Isso porque, não definimos o background-color que estava antes. Para isso, colar o background-color igual ao de antes. Para fazer um leve ajuste do View Height, vamos colocar em -1px no top para que não sobreponha o www do site.

Definido isso, quando a tabela rolar, as nomeações da coluna irão seguir o conteúdo da tabela, e quando ela acabar, ele acaba também.

```
table {
  border-collapse: collapse;
  width: 400px;
  border: 1.5px solid black;
  position: relative;
}

thead > tr > th {
  position: sticky;
  top: -1px;
  background-color: rgba(36, 29, 29, 0.889);
}
```

Piauí	Estado	População
	Alagoas	3 220 848
	Distrito Federal	2 996 899
	Mato Grosso do Sul	2 924 631
	Sergipe	2 299 425
	Rondônia	1 751 950
	Tocantins	1 586 859
	Acre	884 372
	Amapá	806 517
	Roraima	738 772
	Total da população	213 421 037

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Veniam delectu
consectetur dolorum, accusantium ad ipsum nulla sit omnis natus dignis
consequatur voluptatem ea. Lorem ipsum dolor sit amet consectetur ad
praesentium quos fugit odio velit! Natus, nihil! Autem, natus quasi sint ic
nobis deleniti eum qui distinctio. Lorem ipsum dolor sit amet consectetu
voluptatem repellat repellendus quaerat enim autem aliquam libero solu
corporis perferendis. Amet culpa ea expedita incidunt similique! Lorem i
adipiscing elit. Distinctio, perspiciatis repellat quibusdam, maiores reru
velit obcaecati asperiores eaque dicta? Cumque, quidem voluptas nulla
quasi!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Veniam delectu
consectetur dolorum, accusantium ad ipsum nulla sit omnis natus dignis
consequatur voluptatem ea. Lorem ipsum dolor sit amet consectetur ad
praesentium quos fugit odio velit! Natus, nihil! Autem, natus quasi sint ic
nobis deleniti eum qui distinctio. Lorem ipsum dolor sit amet consectetu

Vigésima quinta aula – Mesclagem de células

Se tem este caso aqui: O objetivo é que B ocupe duas COLUNAS, pois ele vai se mexer horizontalmente. Para isso, formata-se no html onde está B isto: *colspan='2'*; (Expansão de coluna, 2).

A	B	
D	E	F
G	H	I
J	K	L

Ficando então:

A	B	
D	E	F
G	H	I
J	K	L

Supondo agora, que tiremos a letra G e J, ao serem retiradas, fique os espaços em branco, certo? ERRADO! Toda a tabela se desregulará, veja:

A	B	
D	E	F
H	I	
K	L	

Ou seja, como está por padrão as os itens html terem como nascimento a direita, as células se mexeram para ocupar os espaços perdidos. Agora, supondo que queira que a célula F ocupe então AS DUAS LINHAS (movimento vertical), faremos outra alteração no elemento F em html: *rowspan="3"*; obtendo este resultado.

A	B	
D	E	F
H	I	
K	L	

Entretanto, no início, o objetivo era retirar os elementos G e J e fazer o elemento D OCUPAR DUAS LINHAS (movimento vertical). Para isso, se remove os elementos G e J e aplicamos *rowspan="3"*; para o elemento D, obtendo este resultado:

A	B	
D	E	F
	H	I
	K	L

Apague o elemento que será substituído, senão:

A	B		
D	E	F	
		H	I
	K	L	

Vigésima sexta aula – Desafios (parte 1)

Professor solicita criação de tabela exatamente igual a dele para treinar conhecimento. Encontra-se dentro do desafio 013. Desafio:

	A	B	C	D
1	1	2	3	4
2	5	6		7
3	8			9
4	10	11		12
5		13	14	15

RESULTADO:

	A	B	C	D
01	1	2	3	4
02	5	6		7
03	8			9
04	10	11		12
05		13	14	15

SEGUNDO DESAFIO:

A	B	C	D	E
A				B
C	D	E	F	
	G	H	I	
	J	L	M	
	N			

RESULTADO:

A	B	C	D	E
A				B
C	D	E	F	
	G	H	I	
	J	L	M	
	N			

Vigésima sétima aula – Exemplo de tabela completa

Professor sugere praticar mais uma vez a construção de tabelas, porém com dados e não letras e números.

Vigésima oitava aula – Escopos de grupo

Agora será ensinado como usar as tags `scope` `colgroup` e `rowgroup`. Basicamente se refere a dizer o seguinte: se caso tiver uma tabela onde uma coluna se refere a mais de um dado, usamos o `colgroup`, e a mesma coisa para linha.

Grupos	Nomes	Filmes		
Mulheres	Ana Maria Santos	Alien	Rambo	Vingadores
	Beatriz Souza	Hulk	Inception	Batman
	Cláudia Melo	Oblivion	Matrix	Big Hero
Homens	Bruno Mendonça	Intocáveis	Amnésia	Gladiador
	Daniel Lourenço	Wall-E	Oldboy	Dangal
	Fabiano Mota	Star Wars V	Taxi Driver	Toy Story
Total	1000			

Em filmes, se colocarmos somente *col*, ele irá considerar somente a coluna que contém: alien, hulk, oblivion, intocáveis, wall-e. star-wars V. Colo tem mais de um conteúdo para esta coluna, usamos o *colgroup*. Mesma coisa para a linha, exemplo: Mulheres, Homens.

Vigésima nona aula – Desafios (parte 2)

Professor tem mais um desafio.

Área	Disciplinas	Notas		Média
		Nota 1	Nota 2	
Exatas	Matemática	0.0	0.0	0.0
	Física	0.0	0.0	0.0
	Química	0.0	0.0	0.0
	Biologia	0.0	0.0	0.0
Média de Exatas				0.0
Humanas	História	0.0	0.0	0.0
	Geografia	0.0	0.0	0.0
Média de Humanas				0.0

RESULTADO:

Quadro de notas escolares

Área	Disciplinas	Notas		Média
		Nota 1	Nota 2	
Exatas	Matemática	00	00	00
	Física	00	00	00
	Química	00	00	00
	Biologia	00	00	00
Média de Exatas				00
Humanas	História	00	00	00
	Geografia	00	00	00
Média de Humanas				00

Trigésima aula – Agrupando Colunas com colgroup

Basicamente, é possível abrir um seletor logo abaixo de *table* para agrupar colunas inteiras e facilitar sua edição em css:

```
<table>
  <colgroup>
    <col class="nome">
    <col class="sexo">
    <col class="idade">
    <col class="profissao">
  </colgroup>
  <caption>Cadastro</caption>
  <thead>
    <tr>
```

Com a tag *colgroup* é aberto a seleção para agrupar colunas de tabelas. Depois é aberto *col* com a quantidade de colunas necessárias. Só identificar cada uma e editar no CSS.

Trigésima primeira aula – Tabelas responsivas

Quando uma tabela é muito grande na tela do celular, cria rolagem lateral. Para isso, envelopamos ela inteira em uma div e chamaremos de container. No css, abrimos a tag *overflow-x* e *auto*. Assim ele cria rolagem lateral somente para a tabela, e não para a página.