**CURSO EM VÍDEO – HTML / CSS3 – MÓDULO 3:**

**Git e GitHub:** São plataformas para **repositório** e **versionamento** de código.

**Repositório local (git):** Por vários motivos precisamos ter backups de versões de nossos projetos. Para isso usamos o gerenciador de versões (nós indicamos quando queremos salvar determinada versão), tornando possível voltar a usar alguma versão a qualquer momento. Git (mesmo criador do Linux) é um software 100% gratuito que guarda versões de um código **dentro do seu computador. Git 🡪 Mantém versões de software em seu computador**

**Repositório remoto / online (GitHub):** Comprado pela Microsoft. Faz um backup de versionamentos de forma gratuita na nuvem (internet). Ele permite que outra pessoa ou cliente consiga ver o projeto de um website, apenas com um link. **GitHub 🡪 Mantém versões de software na nuvem.** A grande sacada do GitHub é trabalhar em equipe, onde mais pessoas podem trabalhar na mesma ou outra versão de um projeto com você.

* Quando mandamos algo do git (repositório local) para o github (repositório remoto) damos o nome de **PUSH.**

Curso completo / gratuito de GitHub.

**Criando repositório:** A primeira etapa é criar um repositório local (disco rígido) de sua máquina. Depois ele permitira publicar (fazer pushing) para o github (remoto).

\*Dica: na hora de criar repositórios prestar atenção em nomes de pastas, subpastas e arquivos.

**Gerenciamento de repositórios:** Menu do canto superior esquerdo você acessa seus repositórios, ao fazer alterações nos arquivos é preciso fazer um **commit to main** é uma manifestação de modificação. E depois clicar em Push origin no github desktop. **COMMITS** basicamente é criar versões para o código.

**Repositório local < > repositório remoto:** Apagar o local (repositor > remove...), não garante apagar o remoto (settings > delete this repositor) e vise versa.

**Como hospedar um site (gratuitamente) usando o GitHub:** O GitHub consegue hospedar páginas que usam, somente, HTML, CSS e JS. Caso precise usar PHP, node.js entre outros é necessário um serviço de hospedagem.

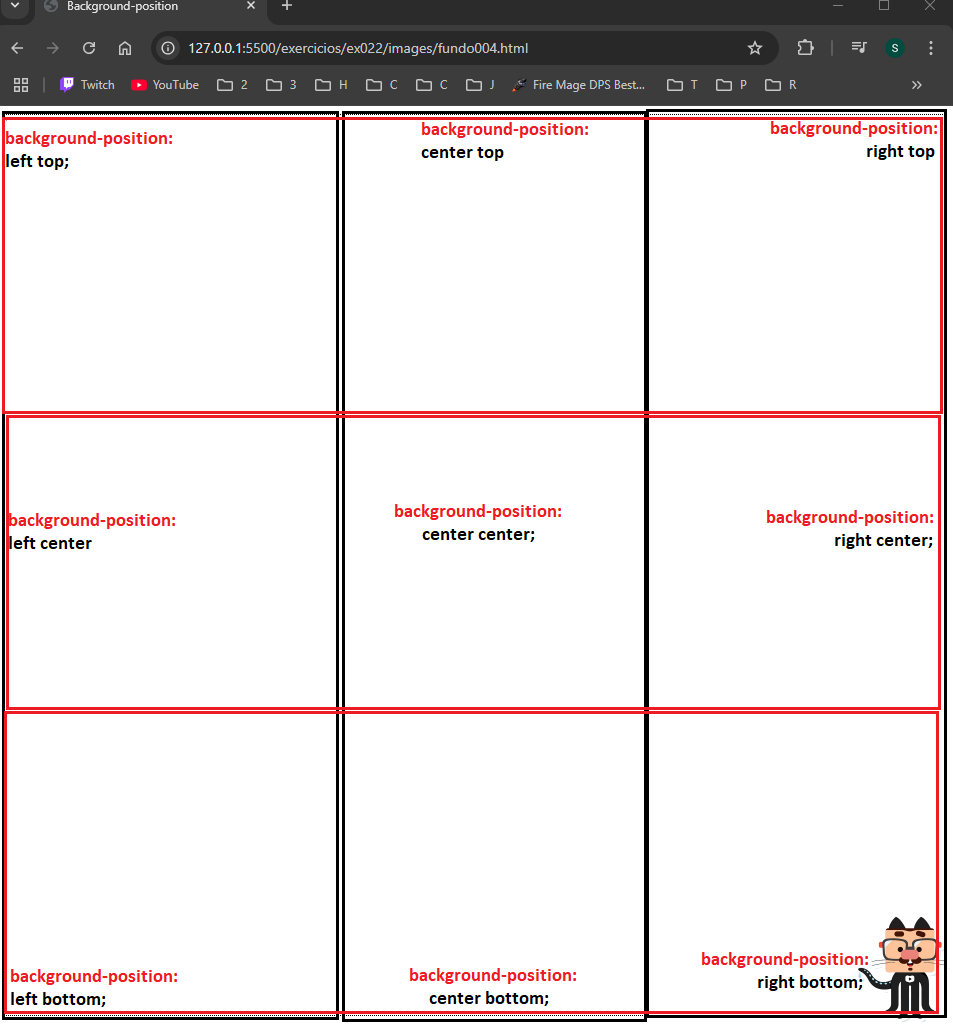
**CLONANDO PROJETOS DO GITHUB:** Basta abrir um repositório e clicar em <> Code > Open with GitHub Desktop e em Choose escolher uma pasta para salvar.

Ideal para trabalhar em uma máquina diferente e “não segura”. Basta clonar o projeto pelo Github Desktop, depois file > options > git > save. Abrir com VScode e fazer todas alterações necessárias. Antes de sair dar push para o repositório online. Fechar tudo > GitHubDesktop > remove repositories + reclicle Bin. File > Options >Sign Out + Git remover dados. Esvaziar lixeira. Desse modo o projeto será atualizado online e limpo da máquina local “não segura” e sem deixar rastro.

**Background-image: url(‘endereço.png’);** É possível usar uma imagem na propriedade background-image no css, podendo inclusive passar valores de arquivos do tipo .jpg, .png, .gif entre outras e também endereços da web (que ficam refém dessa imagem continuar existindo na web). **Por padrão** uma imagem começa a ser esparramada dentro de uma <div> da mesma maneira que lemos (encostada no topo e na esquerda).

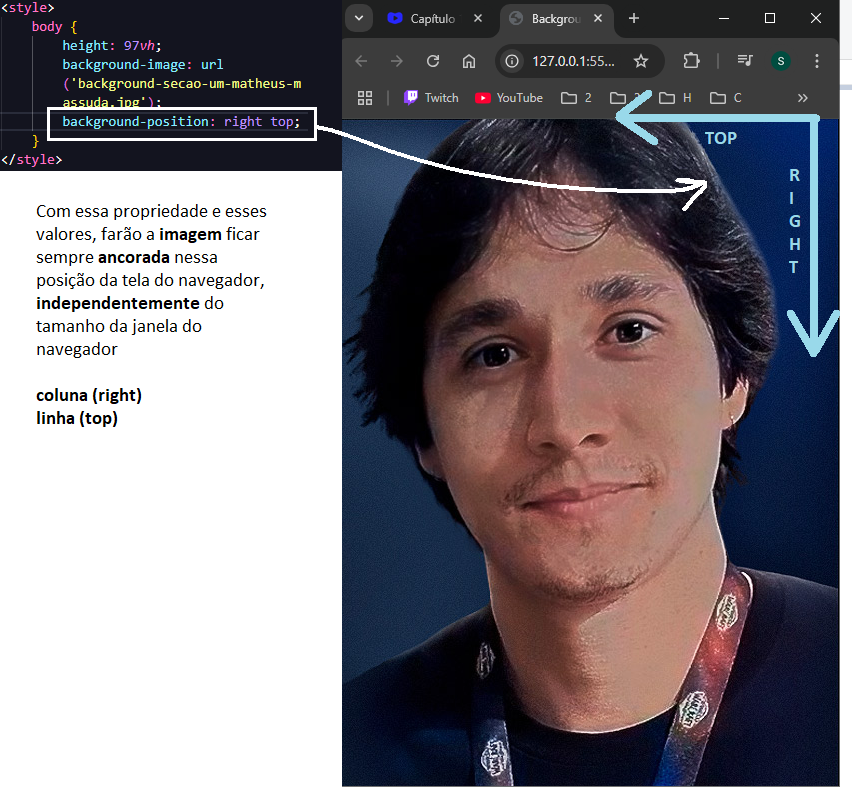
**Background-repeat: repeat;** **REPETIÇÃO:** Essa propriedade sempre causa repetição da imagem para ela completar a tela por causa que por padrão ela passa também (ocultamente) a propriedade **background-repeat: repeat;**

* **No-repeat:** não repete (deixa apenas uma no ponto de partida “canto superior esquerdo”).
* **Repeat:** repete até completar a tela inteira.
* **Repeat-x:** repete no primeiro eixo x (horizontal).
* **Repeat-y:** repete no primeiro eixo y (vertical).

 **Background-position: COLUNA LINHA;** Posiciona a imagem dentro de uma div ou dentro de um espaço imaginário como body. Além disso podemos passar quaisquer valores dentro desse parâmetro para posicionar a imagem da maneira que desejamos como, por exemplo: left top; left center; left bottom; center top; right bottom; Teoricamente podemos dividir a imagem em **9 posições diferentes** com essas combinações.

**Background-size: 100px;** Altera o “tamanho” visual da imagem. Se apenas um valor for passado ele irá repetir esse tamanho para largura e altura.

\***IMPORTANTE:** Se colocarmos o body { height: 100vh;} e colocarmos o background-position: center center; ele irá se ajustar para se posicionar literalmente no centro da tela do navegador. Quando passamos height: 98vh; estamos determinando que a tela de visualização terá 98% da tela, e se passarmos os valores center center em background-position toda vez que mudarmos o tamanho da view port a imagem irá se ajustar para center center. A imagem irá **ancorar** no posicionamento que pedirmos, independentemente do tamanho da janela. Uma outra dica importante é também pré estabelecer uma cor de fundo para o site, pois quando um site é carregado, primeiramente se carrega o conteúdo do site (texto e outros códigos leves e simples). Portanto a experiência do usuário pode ser melhor se tiver uma cor pré estabelecida caso a imagem ainda não tenha sido carregada



**Background-size: auto;** Existe uma maneira de deixar a página “mais ou menos” responsiva para a imagem se ajustar ao tamanho da tela independente do dispositivo que ela for aberta. Por padrão o parâmetro background-size vem com o valor: auto;

1. **CONTAIN:** A imagem será 100% exibida, sem distorção (com proporções originais) e sem corte de pedaços. Com esse valor ele força a imagem acontecer 100% na tela, no entanto ele irá sobrar espaço na tela do navegador a depender do tamanho da janela do navegador.
2. **COVER:** Ele cobre a tela inteira, mesmo que para isso tenha que dar uma cortadinha. É o valor mais utilizado, geralmente. O que pode ser feito é usar {background-size: cover;} e também usar {background-position: coluna linha;} para focalizar a imagem no ponto mais importante e que deve ser exibido para diferentes tamanhos de tela. Nesse caso ele ocupará 100% da View Port, mesmo que para isso não use 100% da imagem.