

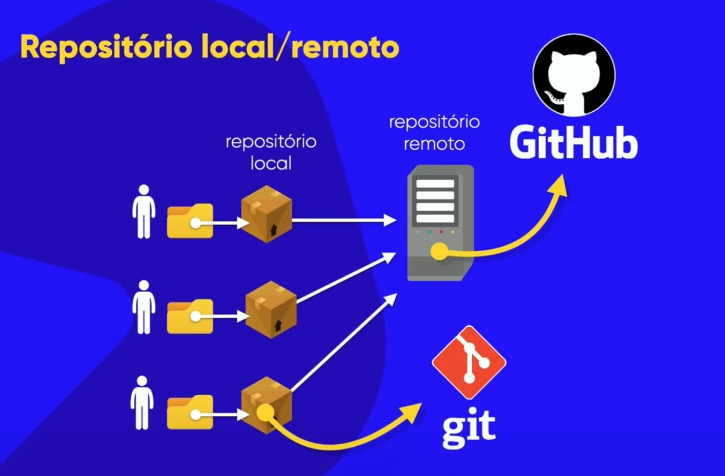
Capitulo 18 a Capitulo 21 dos PDFs!

**Repositório local/remoto**

**Gerenciador de versões**: (guarda todas as versões que você informou que é uma versão).

Software que faz isso: *Git (repositório local)*

Repositório online: *Github (repositório remoto)*



Hospedar site: **GitHub pages**.

Vou na configuração e “pages” em branch seleciona main

(gabrieldjesus012.github.io.)

E pronto o site vai ta disponível!

Ele não tem compatibilidade com varias coisas (somente HTML, CSS e JS)!

Para fazer **ajuste no site** que você hospedou:

Abra o projeto no git e faça as modificações! Depois commit eles!

**Clonar um projeto github:**

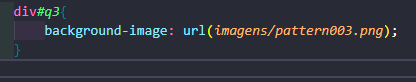
Vai no reprositorio e em <code> aperta open no github desktop! E clone!

**Imagens de fundo**

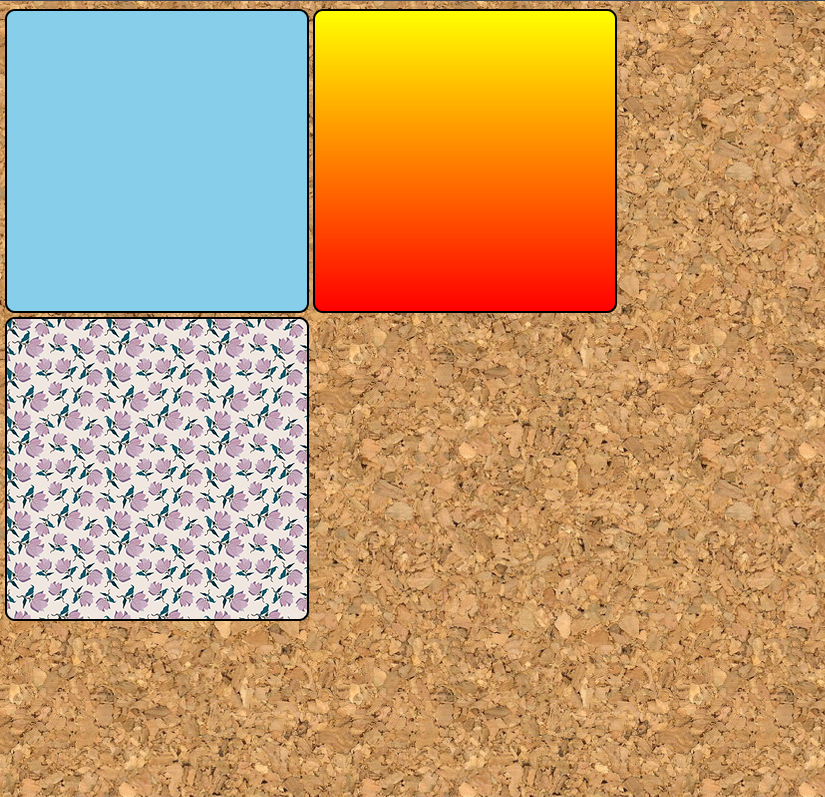
Eu utilizo a função “background-image: url(‘caminho da imagem/nomedoarquivo) – CTRL+ESPAÇO da o caminho;

* Quando coloca imagem muito pequeno ele repete a imagem varias vezes! E quando coloca imagem muito grande ela não coloca tudo.

Quando coloca assim ele não da o sentido semântico para a image!



Também da para colocar no body e pega no site toda! (pode ser um url mesmo)

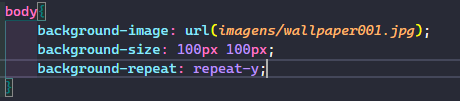


* Mexe no tamanho da imagem: Background-size 100px 100px

É importante destacar que, caso configuremos o tamanho da nossa imagem com esse tipo de valor (largura x altura), a dimensão especificada para a imagem será fixa na tela independentemente do tamanho da viewport (navegador), resultando em páginas não responsivas, em quebras de repetição ou bordas no background.

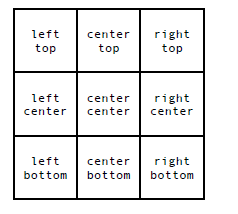
É possível "contornar" isso inserindo valores em % (ex: 100% 100%)... Mas dessa forma, apesar de resolver o problema das quebras de repetição ou bordas no fundo, a nossa imagem ficará DISTORCIDA. Por isso não é muito recomendado usar esse tipo de valor (largura x altura) aplicado para o background, nem mesmo em %.

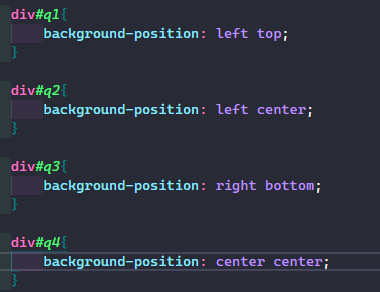
* Para falar quanto repete a imagem: background-repeat: no qual pode ser>
  + no-repeat; -> sem repetição (apenas 1 imagem)
  + repeat; -> repetir normalmente
  + repeat-x; -> repetição na horizontal
  + repeat-y-> repetição na vertical



Ponto de partida é sempre no inicio da pagina! Mas tem como alterar.

* Mudar de posição da imagem: background-position no qual pode ser:



Ex: 

primeiro indicamos o valor para a coluna e depois para a linha.



Na questão de mexer na altura tem que ligar para que:

**vh** ->altura da view port (height) (100vh para pegar a tela toda)

**vw** ->largura da view port (width)

Usando a Propriedade de Estilo apresentada acima e os seus valores, nós conseguimos mudar o Ponto de Ancoragem das imagens de fundo.

Importante: No nosso background, só vamos ver a imagem de fundo se mover por completa de um canto para o outro, caso a dimensão da caixa seja maior que o tamanho da imagem. Caso a imagem seja maior que a dimensão da nossa caixa, veremos apenas uma diferença no Ponto de Ancoragem dessa imagem dentro da caixa (outra parte de imagem como background).

* Background-size (auto, contain e cover)

**auto** - Esse é o valor padrão, se não determinarmos nada, é esse valor que vai prevalecer. Ele faz justamente a imagem de fundo ser aplicada na dimensão original. Dessa forma, normalmente a imagem fica com quebras de repetição ou bordas no background, ocasionando em um resultado não responsivo.

**contain** - Muda o tamanho da imagem para que ela sempre seja totalmente exibida na tela, sem nenhum corte. Dessa forma, geralmente a imagem também fica com quebras de repetição ou bordas no background, ocasionando em um resultado não responsivo.

**cover** - Redimensiona a imagem para que ela cubra todo o contêiner, mesmo que para isso ocorram alguns eventuais cortes ( não tem distorção ). Ele vai cobrir a tela inteira, independentemente do tamanho dessa tela, mesmo que pra isso tenha que cortar um pouco a imagem. Esse valor é o mais recomendado para se obter resultados responsivos.

(problema! Chega uma hora que a imagem quebra mesmo com o cover)

Para não quebrar imagem isso se chama attachment!

Background-attachment: scroll (**rola junto com o conteúdo**) e fixed (**imagem fixa**)

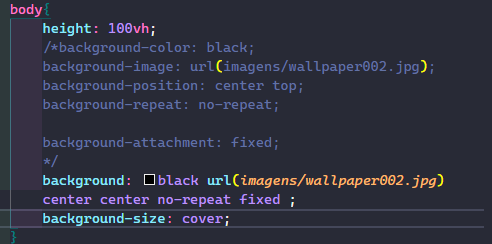
Para efeito Parallax = fixed



Tem como fazer tudo junto de uma vez, seguindo a ordem: background:

**background-color**>**background-image**>**background-position**>**background-repeat**> background-size >**background-attachment**

se usar o size a imagem vai sumir! Então tira ele



**Centralização vertical**

Se for centralizar algo verticalmente precisar ter 2 div (ou seção)

Que seja “**container**” e “**conteúdo**”.



Geralmente utiliza margin auto para aliamento centralizado horizontalmente!

Sobre pósiton: [Entendendo sobre position no CSS (youtube.com)](https://www.youtube.com/watch?v=Y7NeqpwLM2g&t=328s&ab_channel=GiovannaMoeller)

O **container** tem que ter uma position: relative; (pode ser div id=”fora”)

O **conteúdo** tem que ter uma uma position: absolute; (pode ser div id=”dentro”)

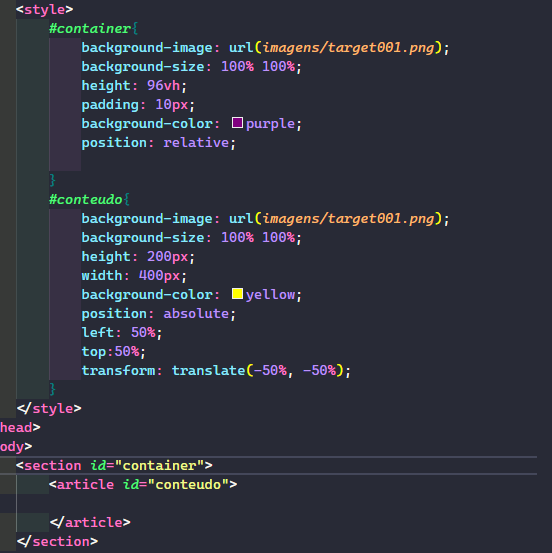
Quando coloca posicionamento absoluto eu posso colocar 4 propriedades: (vamos usar 2)

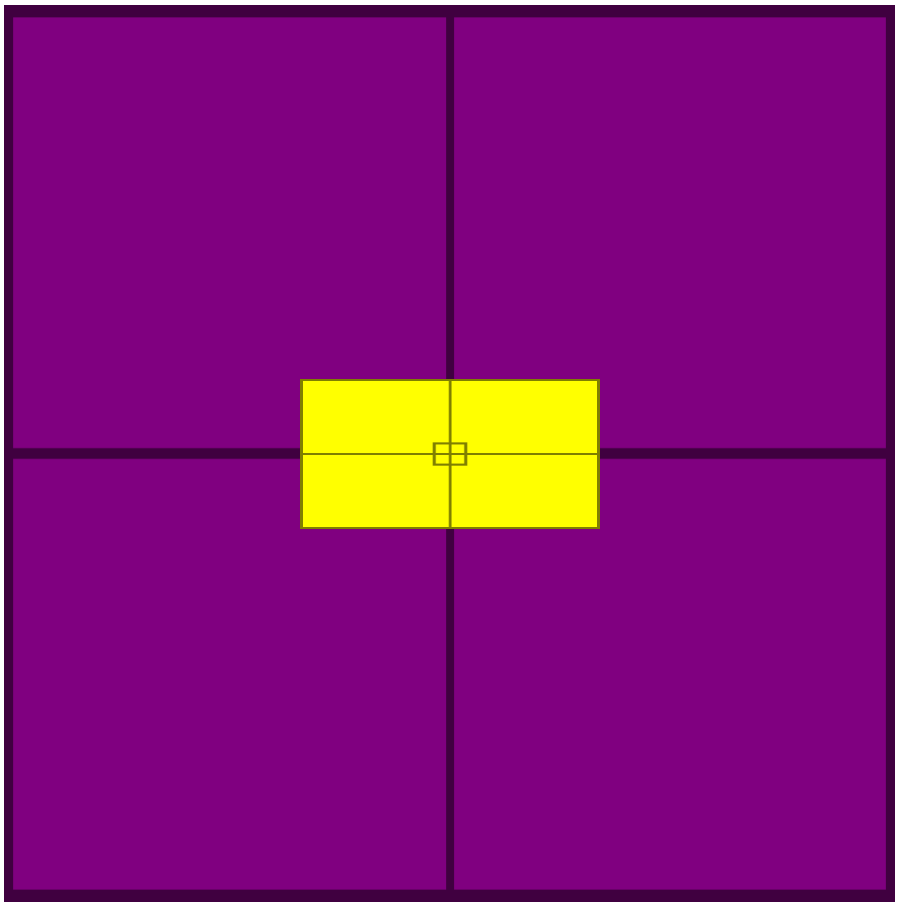
“Left” = 50%

“Top” = 50%

Só que isso só vai centralizar o conteúdo no centro!

Tenho que usar transform: (-50%, -50%)





A função transform também faz outras coisas como: rotate, scale, skew e translate.

**translate** - A função de translate serve para mover o elemento para os lados ou para cima/baixo. Mudar o seu "ponto de alinhamento"... A sintaxe pode ser especifica para um dos eixos x/y ( translateX ou translateY ), como pode ser apenas translate, abrangendo os dois eixos ( x, y ). Obs.: Para ir para o lado esquerdo ou para cima, use valores negativos.

**rotate** - Como o nome já sugere, é utilizada para rotacionar o elemento, de 0 a 360 graus. Utilizando valores positivos, a rotação será realizada no sentido horário e, utilizando valores negativos, no sentido Anti-horário.

**scale** - Utilizado para redimensionar os elementos, tendo como escala padrão o valor "1". Valores maiores que 1, aumentarão o tamanho do elemento, e valores menores que 1 diminuirão sua escala.

**skew** - É utilizado para "distorcer" os elementos, alterando os ângulos.

**Nota:** *A propriedade de estilo "transform" e suas funções é muito utilizada juntamente com a pseudo-classe "hover" e com a propriedade de estilo "transition" para criar efeitos visuais.*

Existe um jeito muito mais fácil de alinhar items, com apenas três linhas

#container {

display: flex; /\*Possui diversas funções, uma delas é habilitar o manejamento do conteúdo\*/

justify-content: center; /\*Alinhamento horizontal, ou seja, dispensa o "margin: auto"\*/

align-items: center; /\*Alinhamento vertical\*/

}

**Essas propriedades sempre são colocadas no elemento-pai (container).**

**Desafio 012**

Colocar **tudo em** **letra maiúscula**: text-transform: uppercase;

Ou colocar font-variant: small-caps; 1º letra maior

Às vezes o TEXTO do nosso site fica adequado para telas maiores (desktop) mas quando abrimos a página em uma tela menor (celular), começa a ficar grande demais para ler...

No desafio "Cordel Moderno" usamos um recurso que faz com que o TEXTO do site aumente ou diminua de acordo com o tamanho (tamanho da largura) da janela do navegador.

Para fazer isso, a dica é usar as unidades de medidas relativas das CSS, como o vh e vw. Basicamente, vh significa " viewport height " e vw significa " viewport width ".

Sendo assim, se você quer formatar o seu texto dessa forma, ao invés de configurá-lo com unidades de medida absolutas (como px ou em), configure o tamanho do texto (font-size) com a unidade de medida "vw".

Exemplo:

Para o título principal: font-size: 10vw;

Paro o corpo da página: font-size: 3.5vw;

Na primeira opção o tamanho da fonte vai ser referente a 10% da largura da viewport. Já na segunda opção, será referente a 3.5% da largura dessa viewport.