**Repeat count auto fit**

No vídeo anterior, aplicamos o auto-fit! Agora eu e você vamos entender mais a fundo do que se trata essa palavra reservada.

A função repeat() recebe como parâmetro dois argumentos, o primeiro é o **repeat count**, que se refere a quantas vezes o valor irá se repetir e o segundo é o **track** que é o valor em si que será repetido.

Voltando a atenção para o *repeat count*!

O Repeat count pode ser um valor inteiro de 1 ou mais, como por exemplo no código: repeat(3, auto), 3 é o valor do repeat count. Mas também o valor pode ser palavras-chaves, como auto-fill ou auto-fit.

Mas o que isso significa?

**auto-fill**

Preenche a linha do Grid Container com novas colunas sempre que há espaço para caber mais uma, o objetivo é preencher a linha com o máximo de colunas possível. Caso o viewport seja maior do que a largura do Grid container e número de colunas existentes, esse será o resultado:

Dessa forma, haverá um espaço sobrando do lado direito, as colunas não expandirão o tamanho para preencher esse vazio.

**auto-fit**

Ajusta o tamanho das colunas para ocupar o tamanho total do espaço disponível do container, expandindo-as para que não fique nenhum espaço sobrando. Independente do tamanho da tela, esse será o resultado:

Nos exemplos acima, o grid container possui uma largura de aproximadamente 600px, note que nesta mesma largura de tela, não há diferença visual entre os valores auto-fill e auto-fit. Veja:

Mas a partir do momento que aumenta o tamanho da viewport, começa a aparecer a diferença dos repeat count.

Você pode visualizar melhor esses exemplos [nesse codepen](https://codepen.io/beatrizmouradev/pen/XWEZYEz)!

Ok! Mas porque eu pedi para você não se preocupar naquele momento do vídeo com o valor auto-fit?

Se você trocar o auto-fit por auto-fill, verá que em qualquer tamanho de viewport não fará diferença, mas porque isso?

No caso do AluraCast, estamos utilizando os valores 100%/2, 100%/3 e 100%/6 e isso faz com que obrigatoriamente gere um contexto de auto-fit independentemente do repeat count que é aplicado. Isso ocorre porque a largura do Grid Container é dividida por um valor x e isso é distribuído para cada coluna e em qualquer tamanho de tela, a adaptação é automática e o resultado sempre será o tamanho das colunas sendo ajustado para ocupar totalmente o espaço disponível de largura do container.

Então, nesta situação o repeat count não interfere, mas experimente aplicar os seguintes código para o elemento .secao-vertical .secao\_\_cartoes e visualizar a partir da tela de 1600px:

* Exemplo 1: auto-fit

grid-template-columns: repeat(auto-fit, minmax(156px, 1fr));

COPIAR CÓDIGO

* Exemplo 2: auto-fill

grid-template-columns: repeat(auto-fill, minmax(156px, 1fr));

COPIAR CÓDIGO

Note que agora, que está sendo utilizado um valor fixo é possível visualizar a diferença entre os dois tipos de repeat, pois agora o resultado não é a distribuição automática de 100% da largura do container para cada coluna.