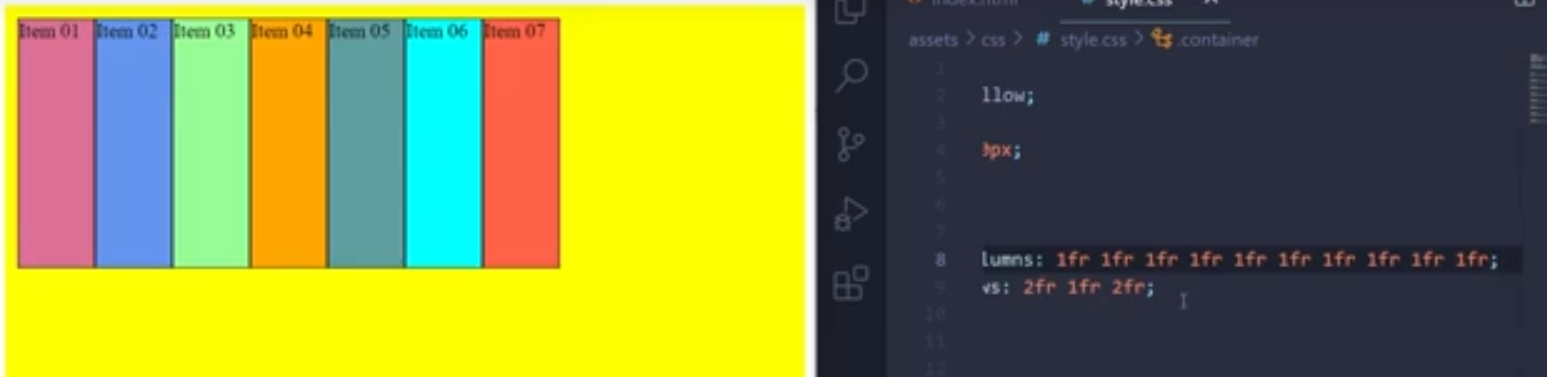


Imagina que temos um grid muito grande e queremos aplicar varias unidades de medidas para cada item dentro dele.



Nesse caso estamos dizendo que cada coluna deve ocupar 1 fr

Imagina fazer isso para 50 colunas...

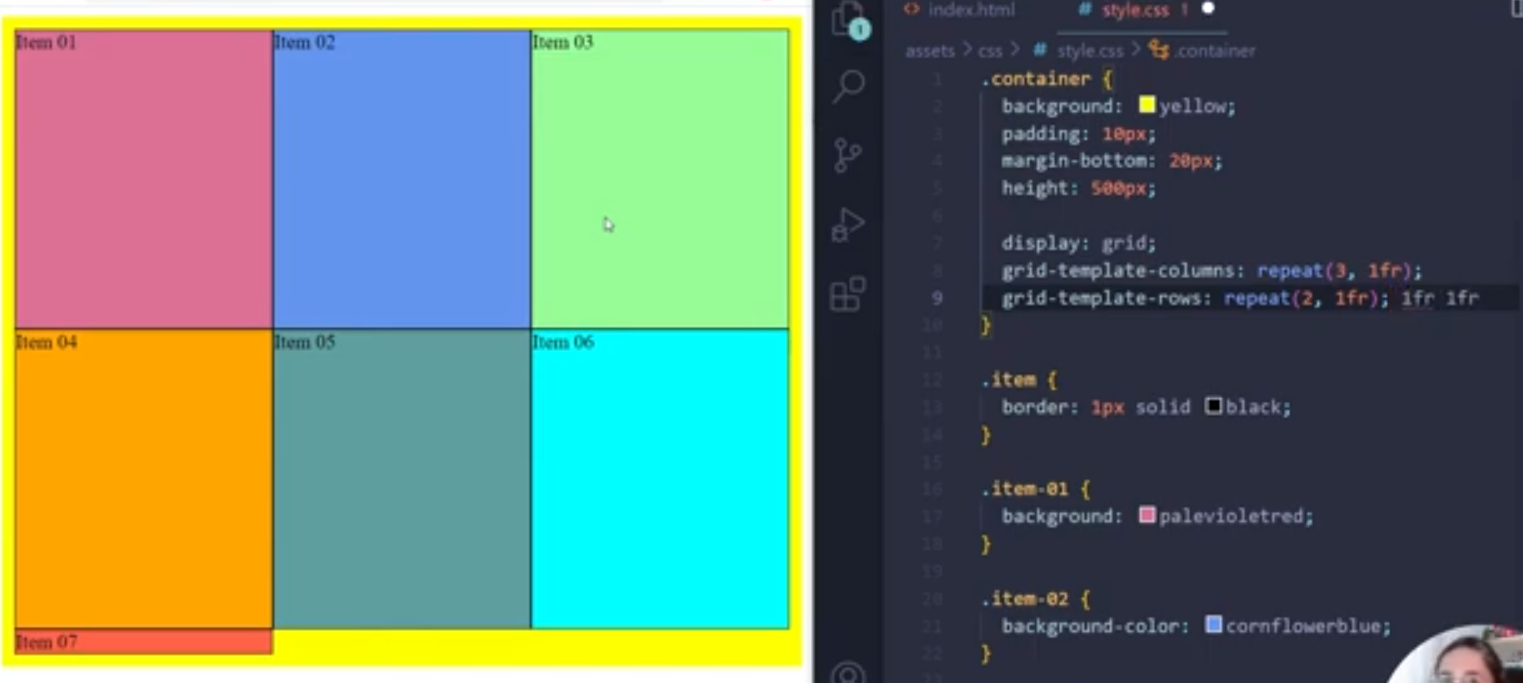
O código ficaria horrível.

grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr;

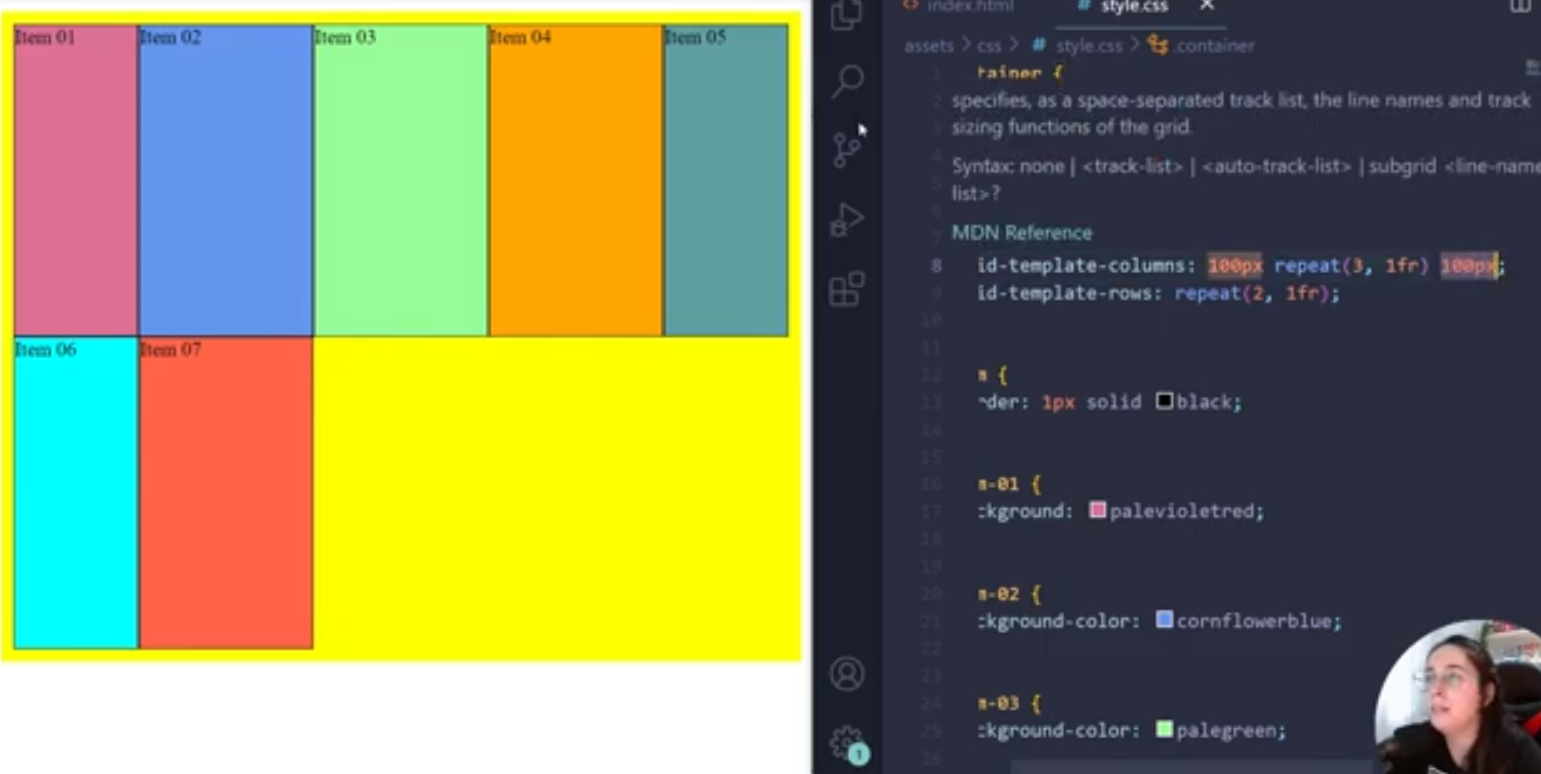
            grid-template-columns: repeat(6, 1fr);

Podemos fazer dessa forma, dizendo que vamos ter 6 itens e que cada um deve ter 1 fr

Outra explicação: parametro1 = quantas vezes quero repetir a coluna por exemplo, qual valor que cada uma vai ter.



Outra forma de fazer é assim também, usando unidade de medida e repeat tudo junto



Como ficaria?

Primeiro item 100px

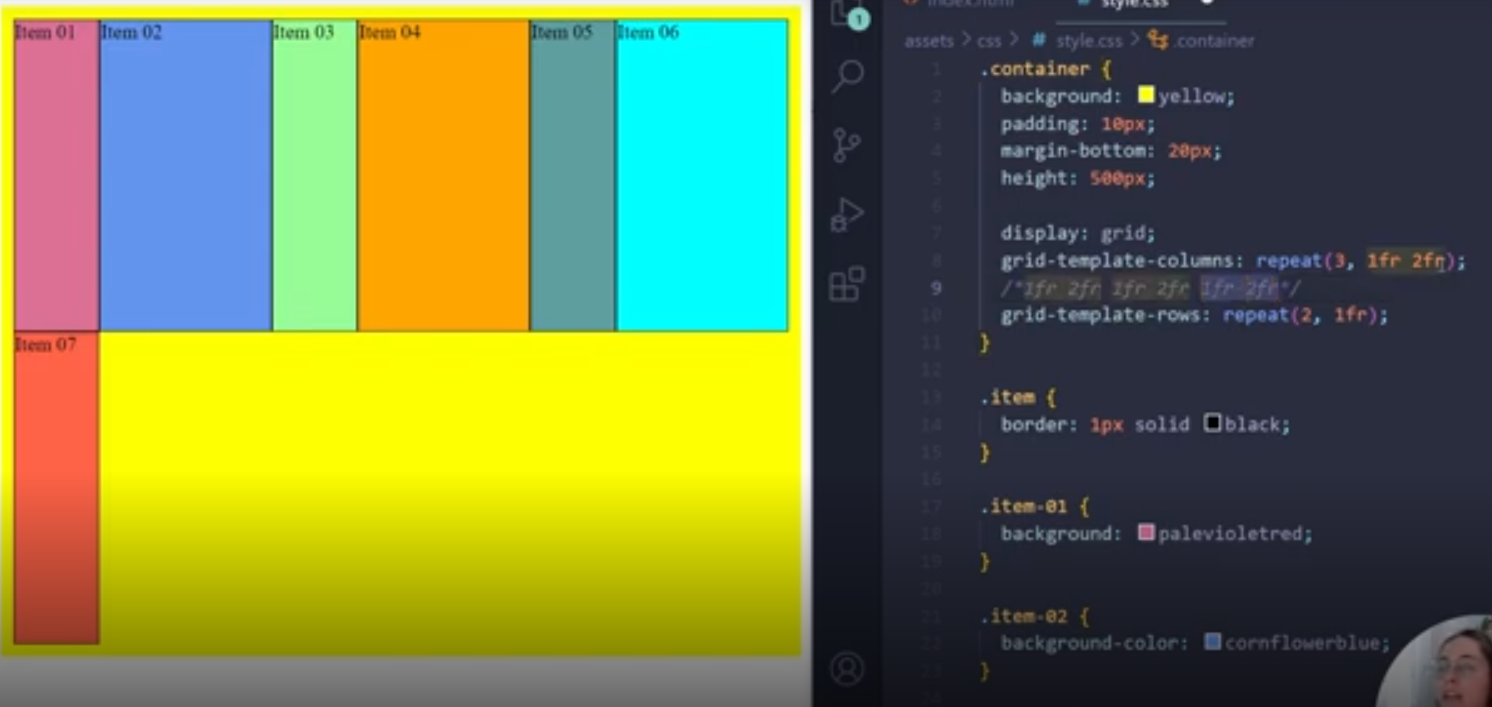
Item 2, 3 e 4 1fr

Item 5 100px

Ao total vamos ter 5 colunas.

Assim vamos combinar diferente valores para cada coluna

Outra forma de definir o tamanho também é assim



Aqui e como se a:

Item 1 tivesse 1fr

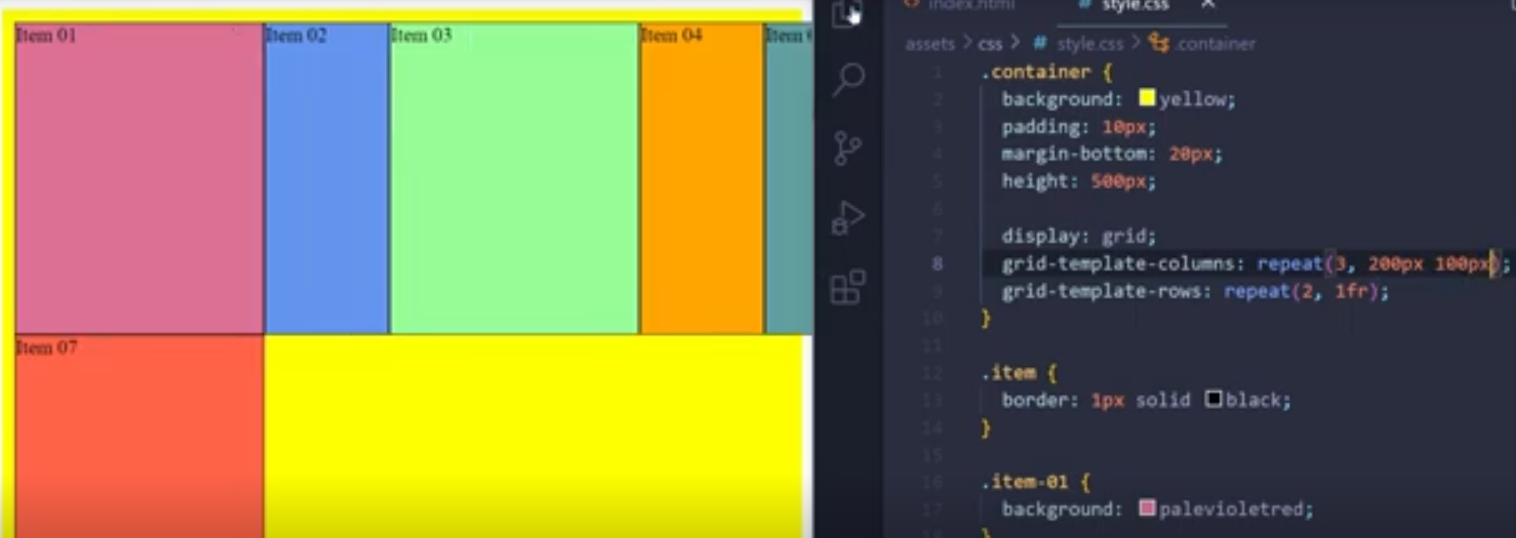
Item 2 tivesse 2fr

Item 3 1fr

Item 4 2fr

E assim por diante.

Também podemos usar px como unidade, para fazer essas 3 repetições



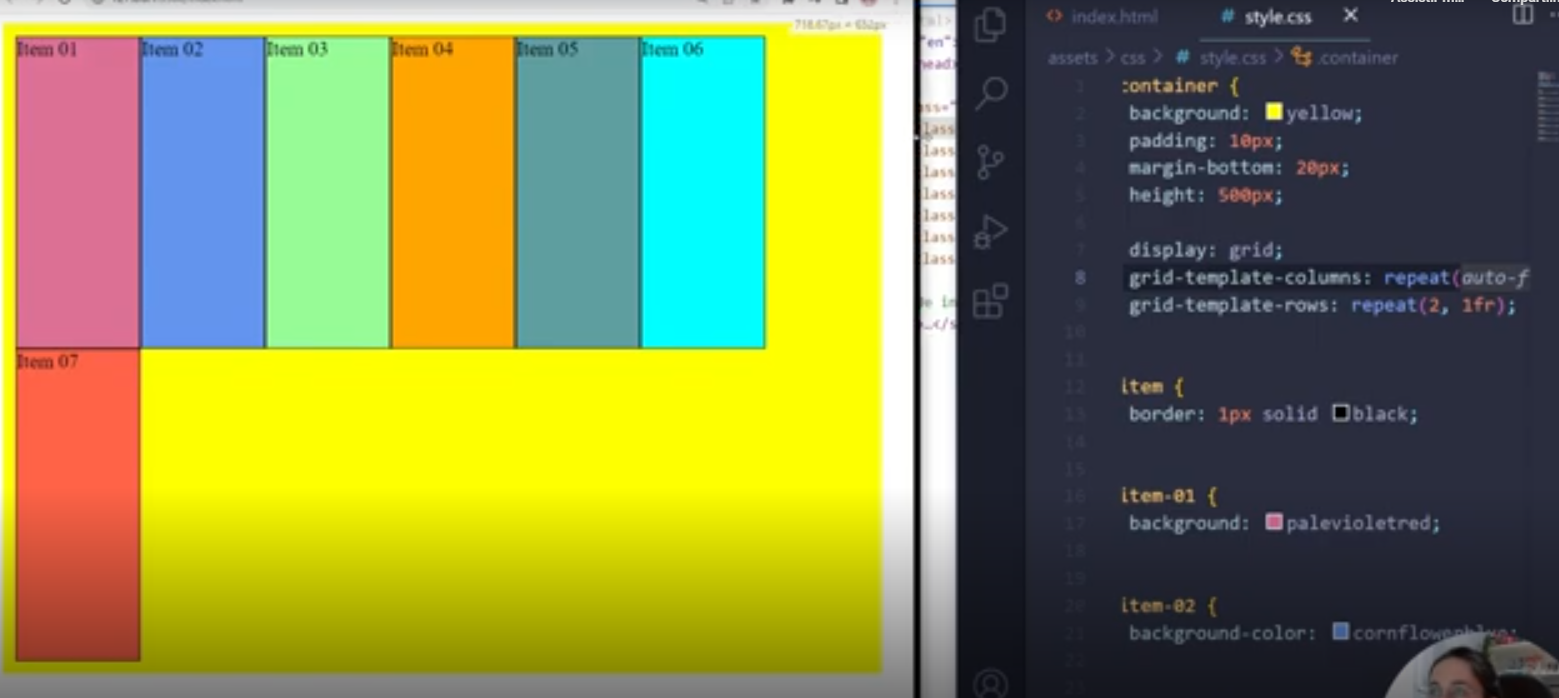
# Palavras chaves para função Repeat

## Auto-fill

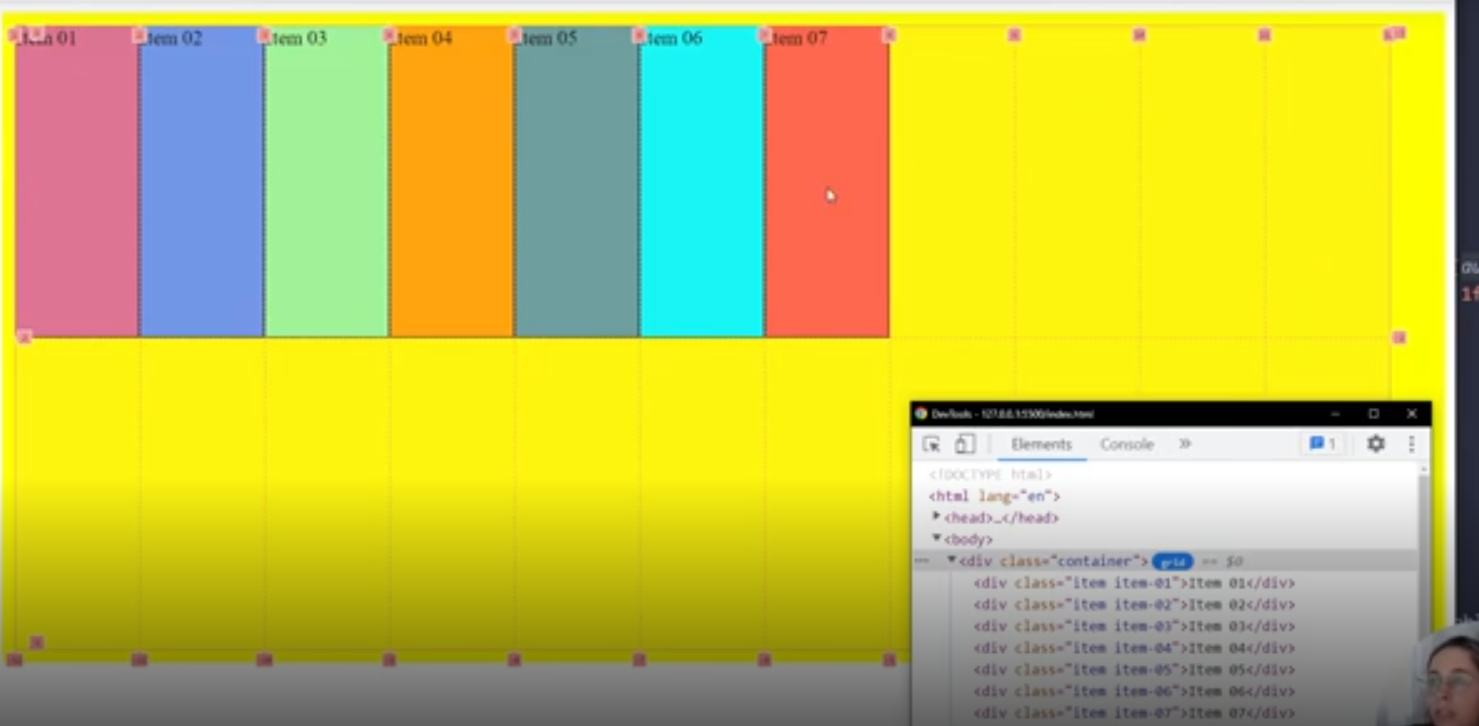


Auto fill ele vai ajustar para colocar o máximo de coluna que tiver.





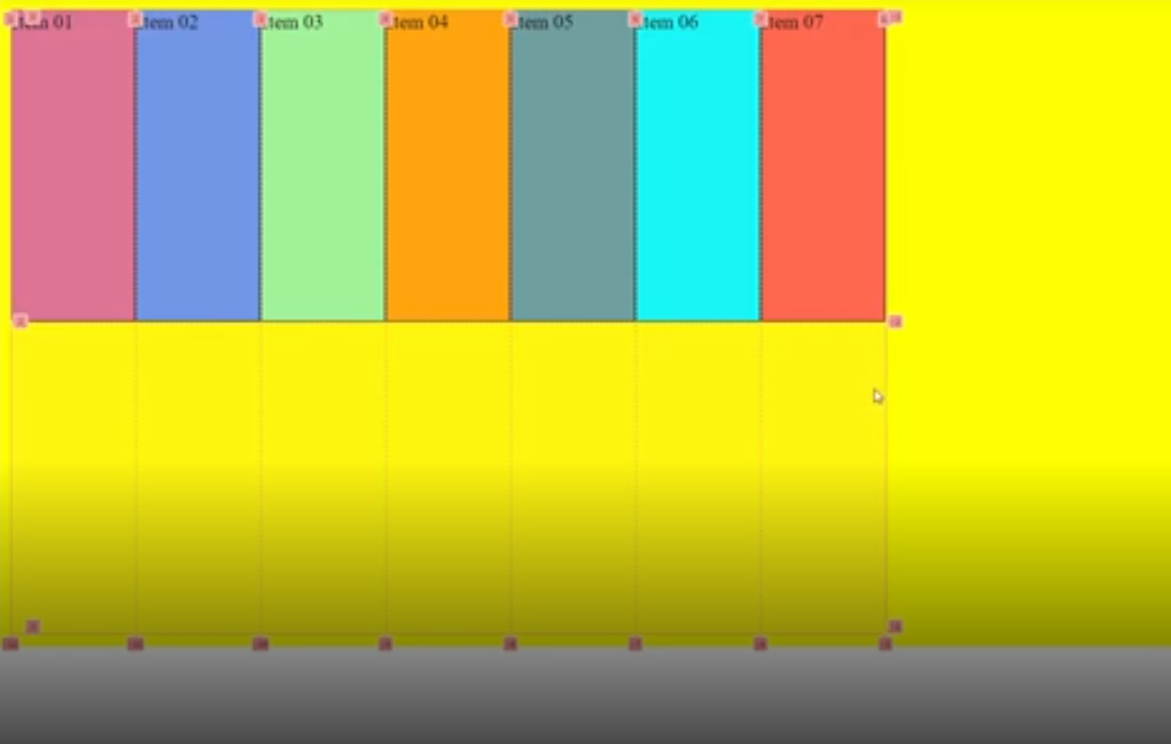
Então conforme vai diminuindo a tela e ele percebe que não cabe ele joga para a linha de baixo.



Repara que mesmo não tendo item o suficiente ele deixa o espaço disponível.

## Auto-fit

Já aqui, vai ser ao contrario, se não tem item o suficiente ele vai automaticamente recolher essas faixas.



Repara que aqui não tem item suficiente, então ele removeu as faixas.

# Min-content

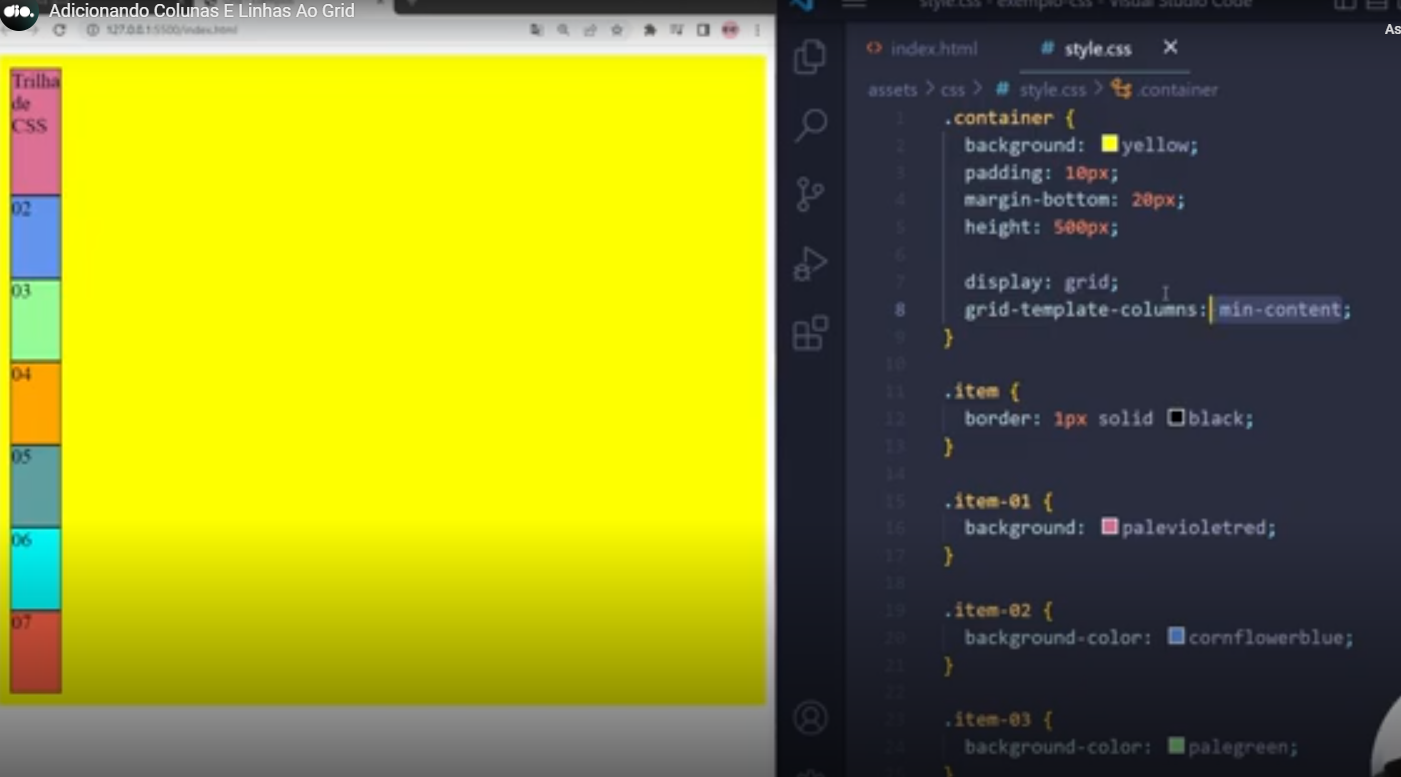
Vamos criar uma coluna com o espaço mínimo que temos do nosso conteúdo.

Ela vai quebrar tudo que puder para caber dentro

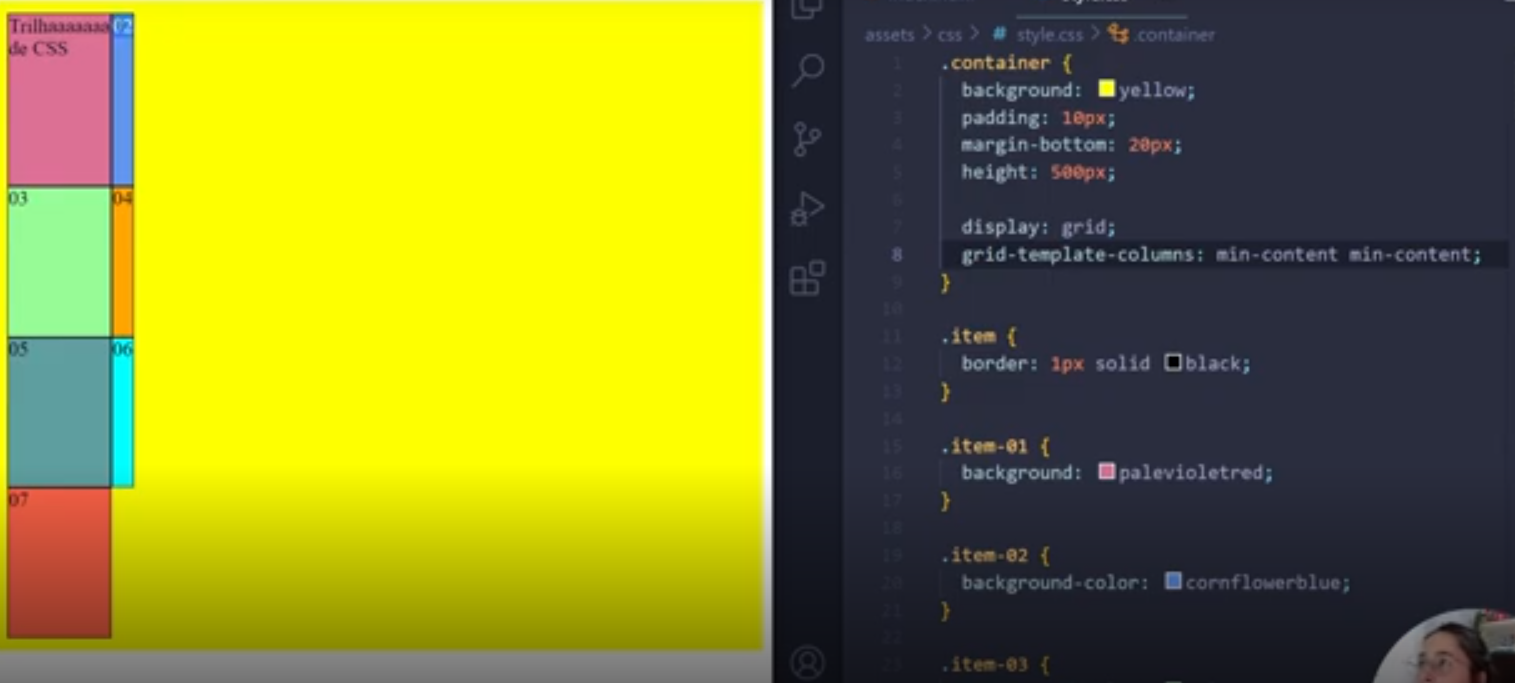
🐜 min-content → “o menor que dá pra ser”

* A prateleira vai **encolher o máximo possível**, mas **sem cortar o conteúdo**.
* Exemplo: se você tem um texto “banana amassada”, ele vai quebrar em várias linhas pra caber.
* É tipo: “quero que isso ocupe o mínimo de espaço, mas ainda dê pra ler”.

🧠 **Use quando** quiser que o conteúdo se ajuste ao menor tamanho possível sem sumir

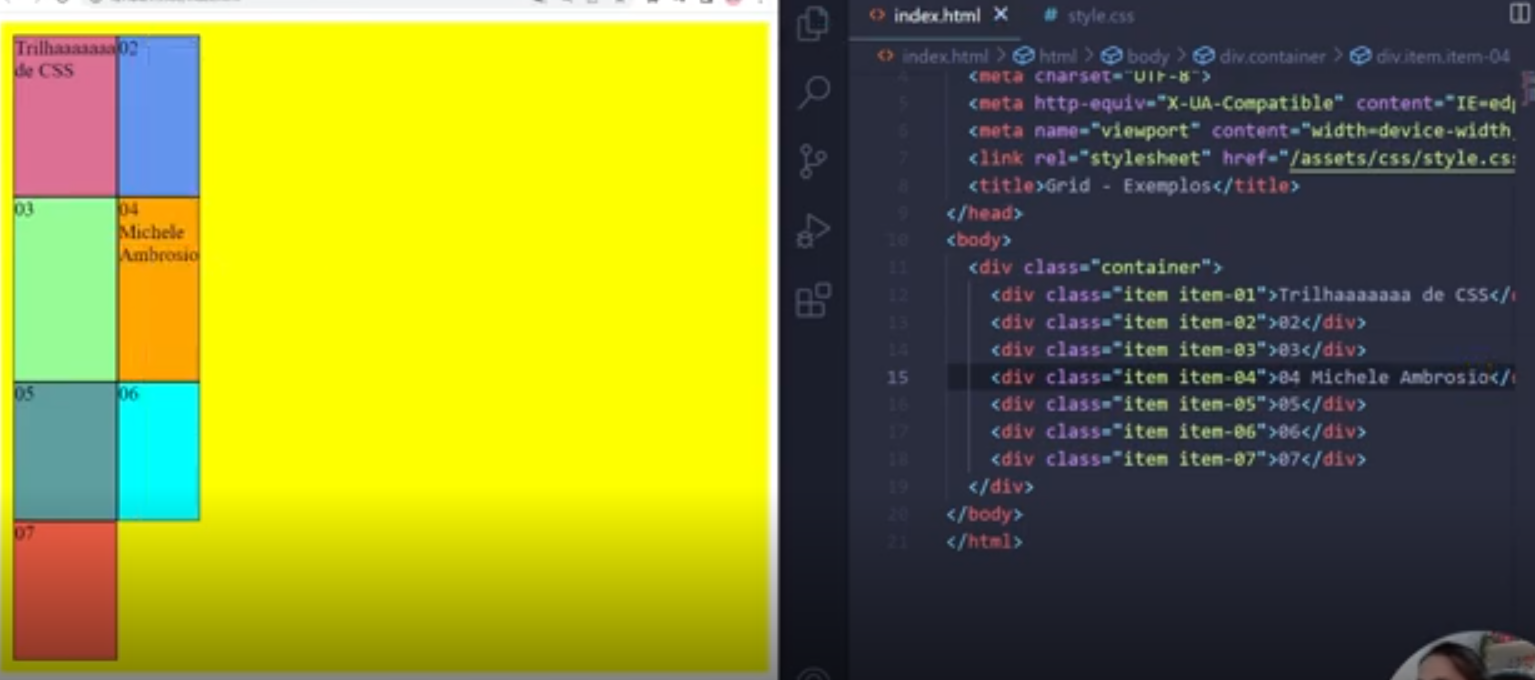


Atenção ele vai sempre respeitar o tamanho do conteúdo para não quebrar.



Repara que agora criamos 2 colunas, e a segunda ele criou bem pequena, porque ele ta usando o tamanho mínimo para alocar o elemento.

Agora colocando um texto maior dentro do conteúdo.



# Max-content

🐘 max-content → “o maior que dá pra ser”

* A prateleira vai **crescer até o conteúdo acabar**, sem quebrar linha.
* Exemplo: “banana amassada” vai ficar tudo numa linha só, mesmo que fique gigante.
* É tipo: “deixa isso esticado até o fim, sem cortar nada”.

🧠 **Use quando** quiser que o conteúdo fique todo visível numa linha só, sem quebra



# Auto

🧩 auto no CSS Grid é tipo:

“Deixa o tamanho ser o que **precisar ser**, dependendo do conteúdo e do espaço disponível.”

🛠️ Como funciona o auto:

* Ele **cresce ou encolhe** conforme o conteúdo **e** o espaço disponível no grid.
* Se o conteúdo for pequeno, a célula fica pequena.
* Se o conteúdo for grande, ela **tenta** crescer… mas **não ultrapassa** os limites do grid.

