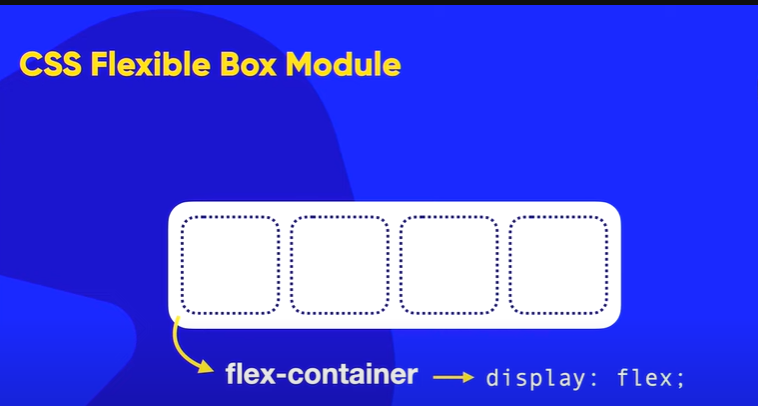
**Aula 3 – Direções e Eixos Flexbox CSS**



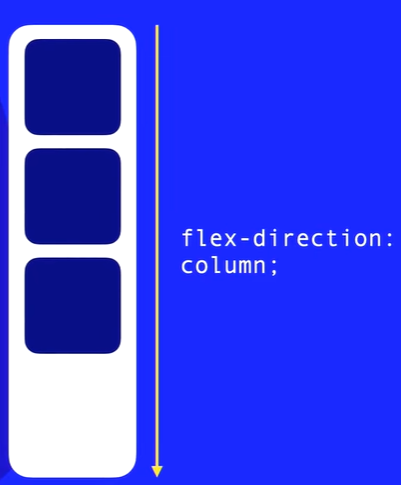
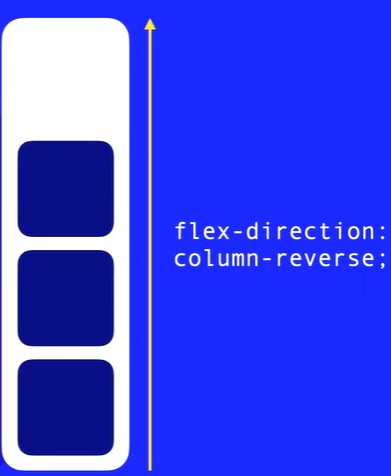
A configuração de display flex será aplicado apenas ao elemento pai.

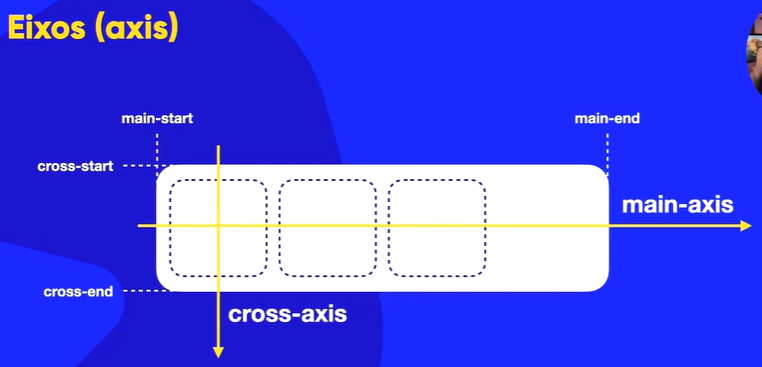


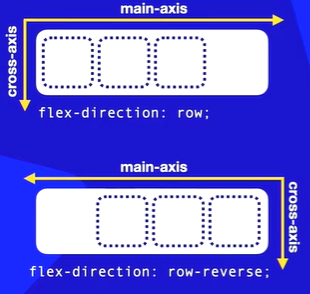
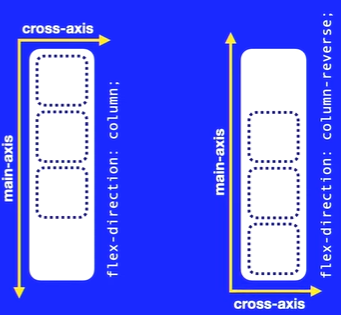
Padrão como row, em uma linha deitada.



Inverso do padrão row, a linha deitada mas o reverso

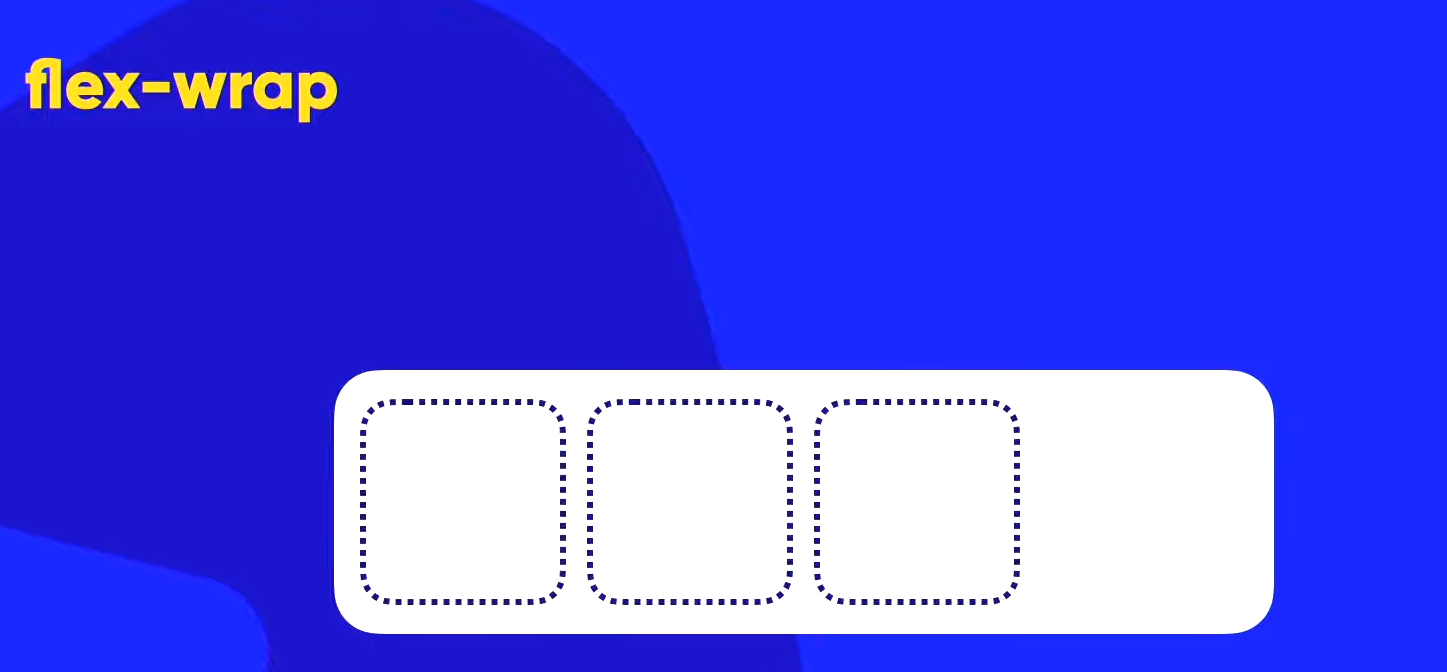
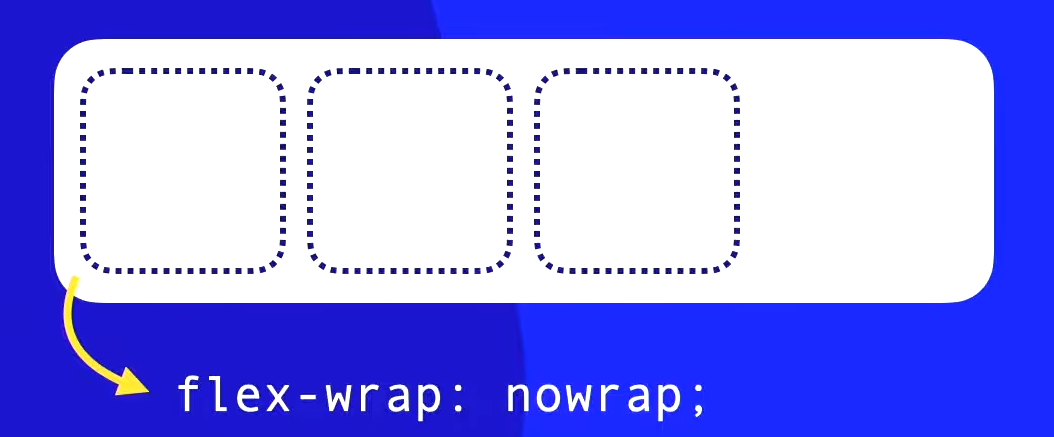
 

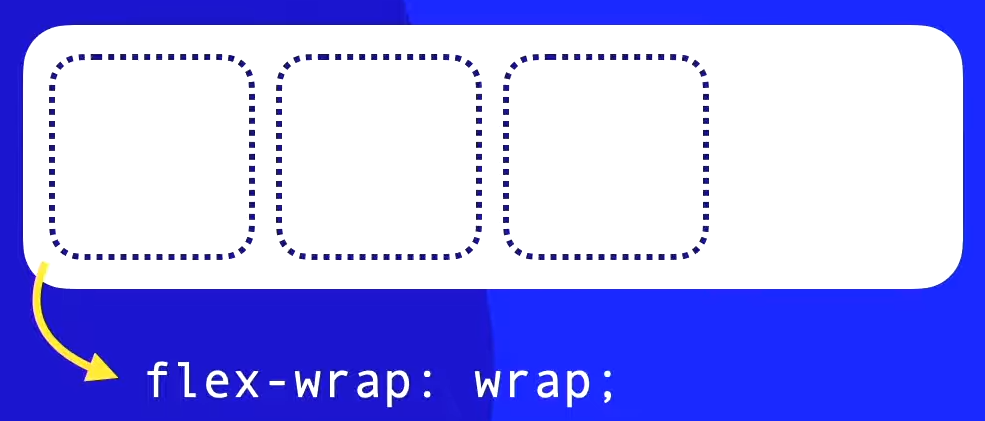
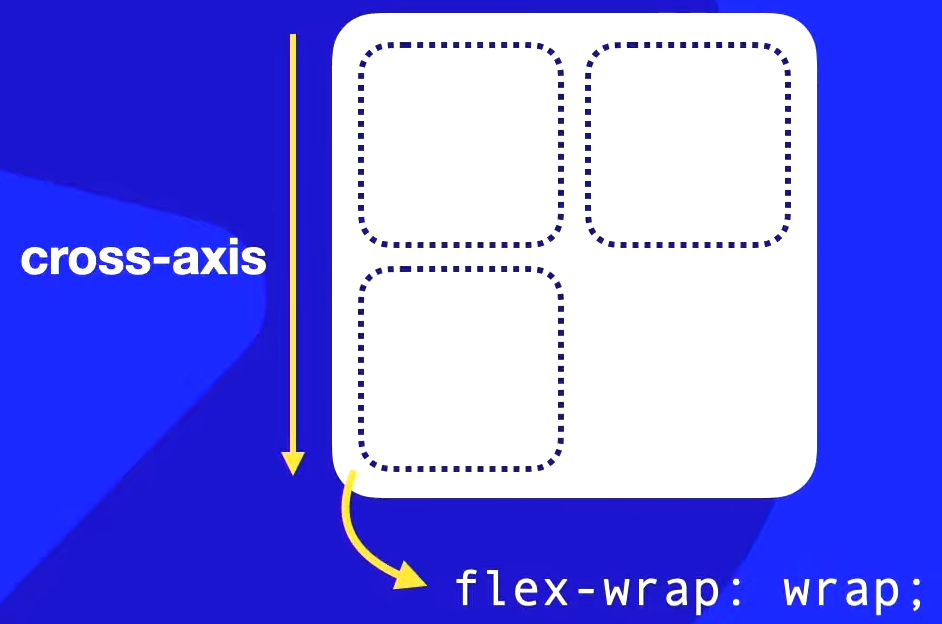
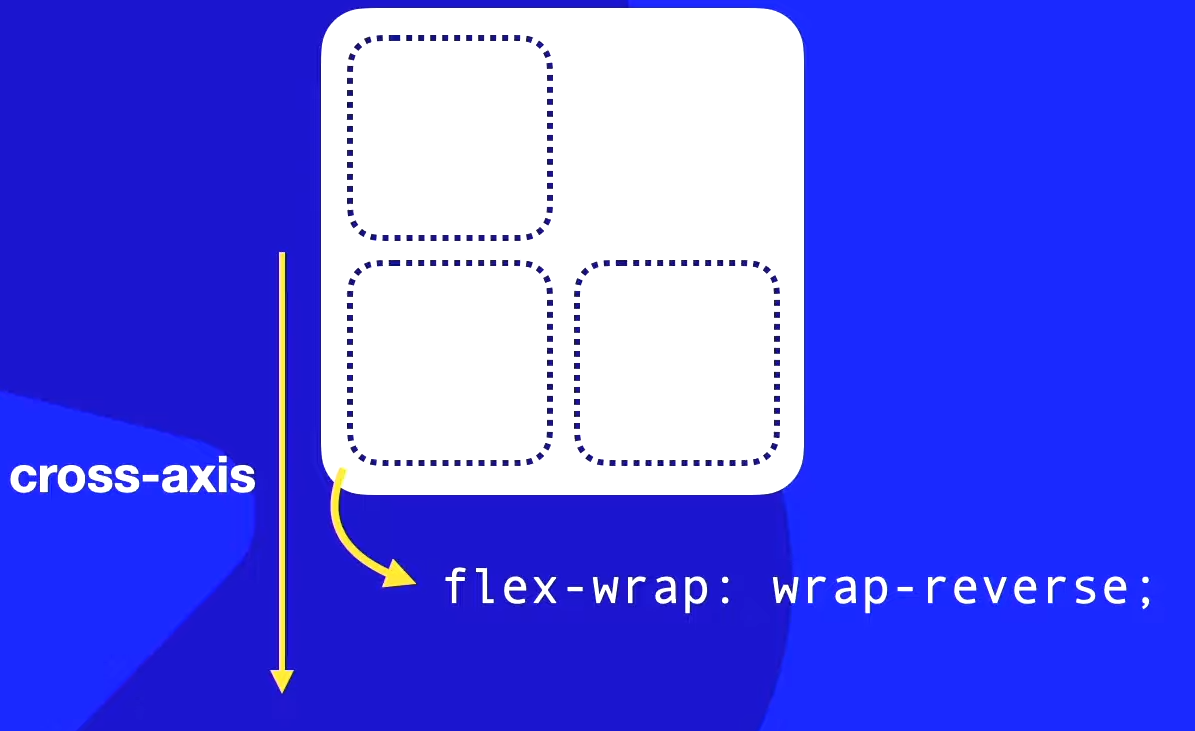


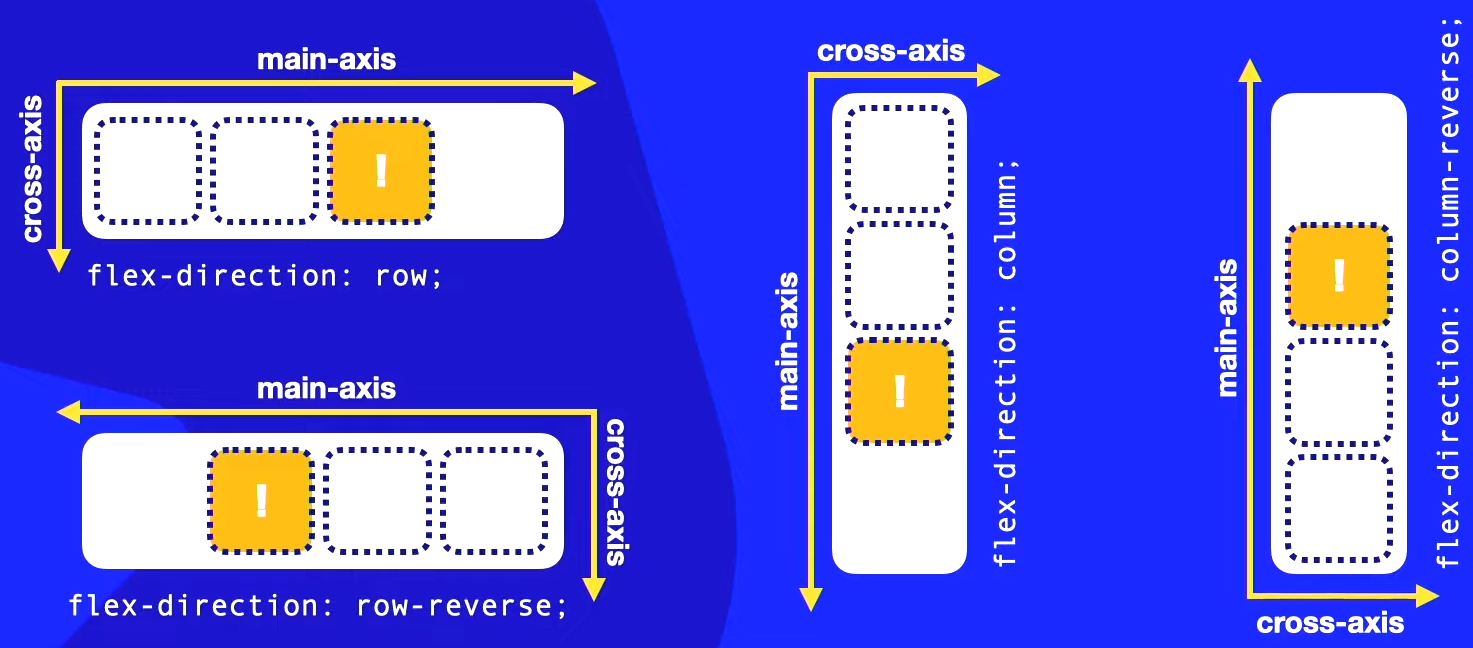
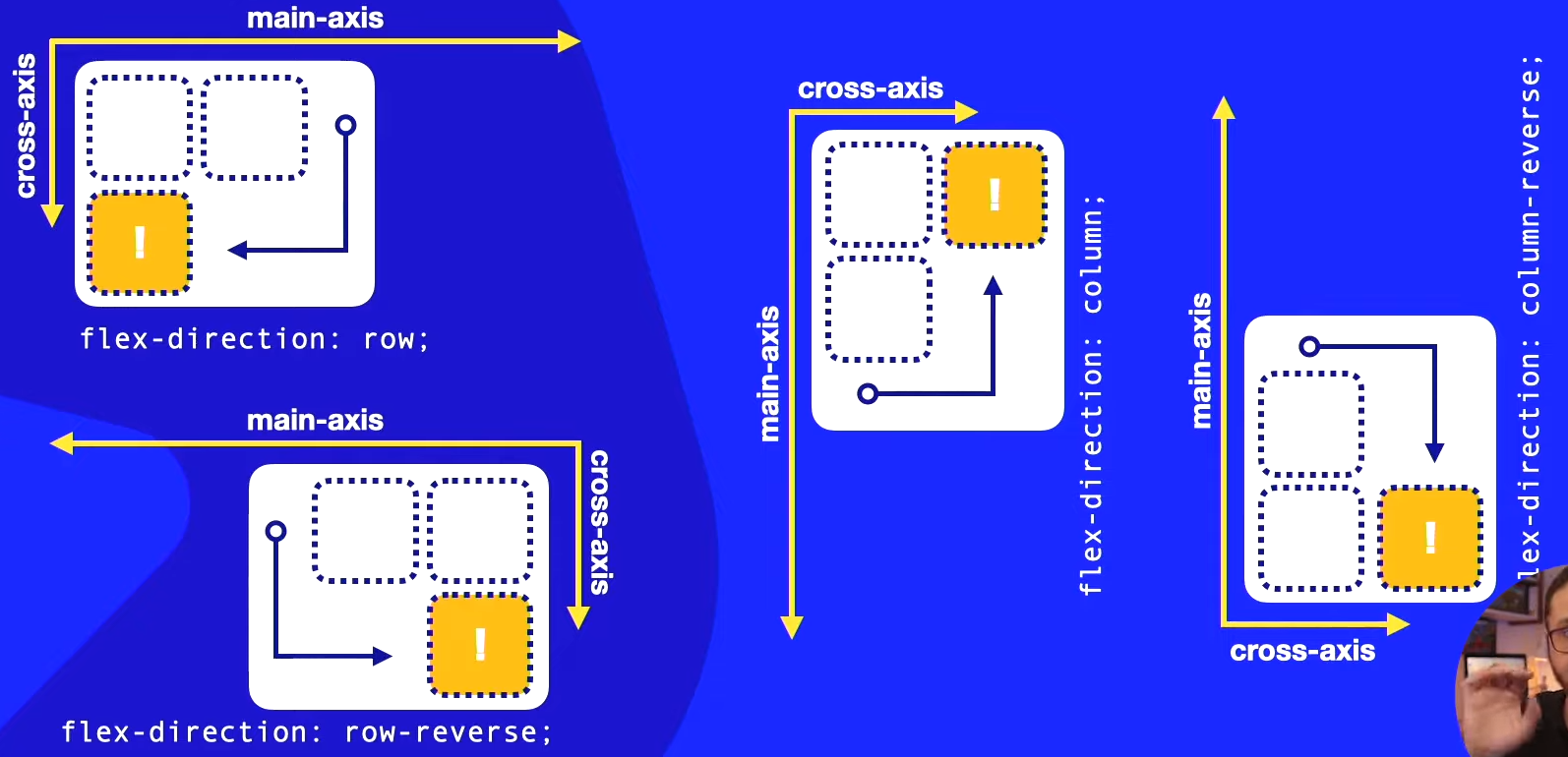
Esses são os eixos **main** e **cross** que ditam a direção em Flex-Direction e seus valores



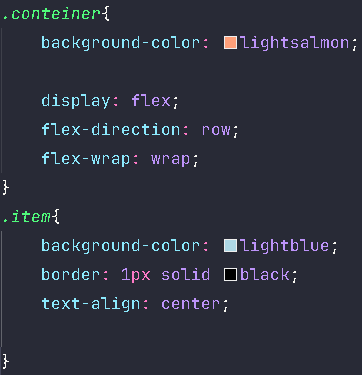
  

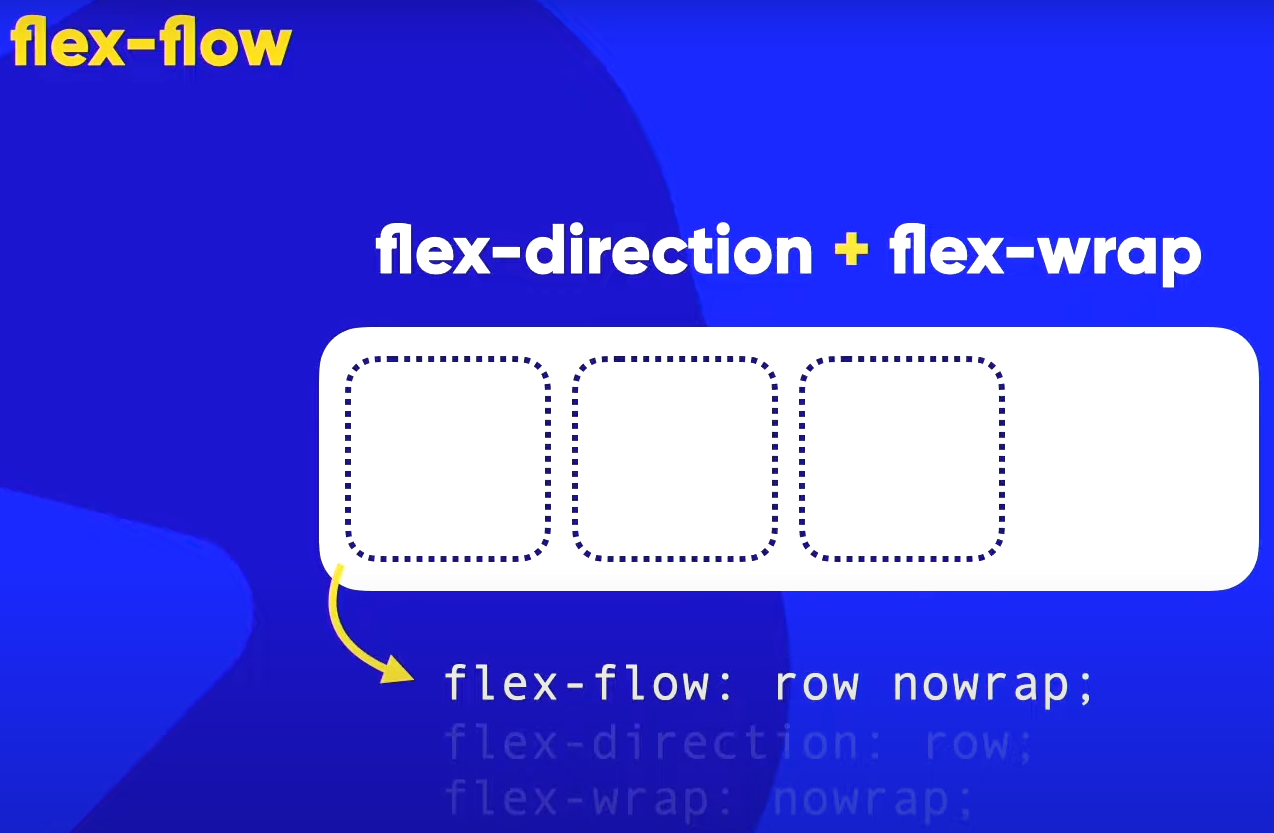
Aqui temos pequenas diferenças entre o uso de wrap ou nowrap. Wrap faz com que um conjunto de objetos quebre ordenadamente ao atingir o limite da tela e nowrap não deixa isso acontecer, até mesmo “redimensionando” o objeto em questão. Agora wrap-reverse faz a quebra, mas de modo “ao contrário”.

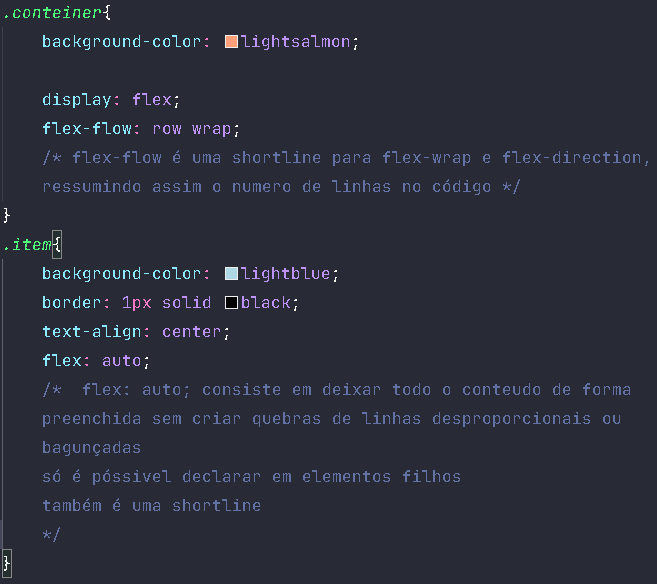
Aqui podemos observar os mais diferentes casos usando Flex-Direction + Flex-Wrap, e como se comportam os objetos.

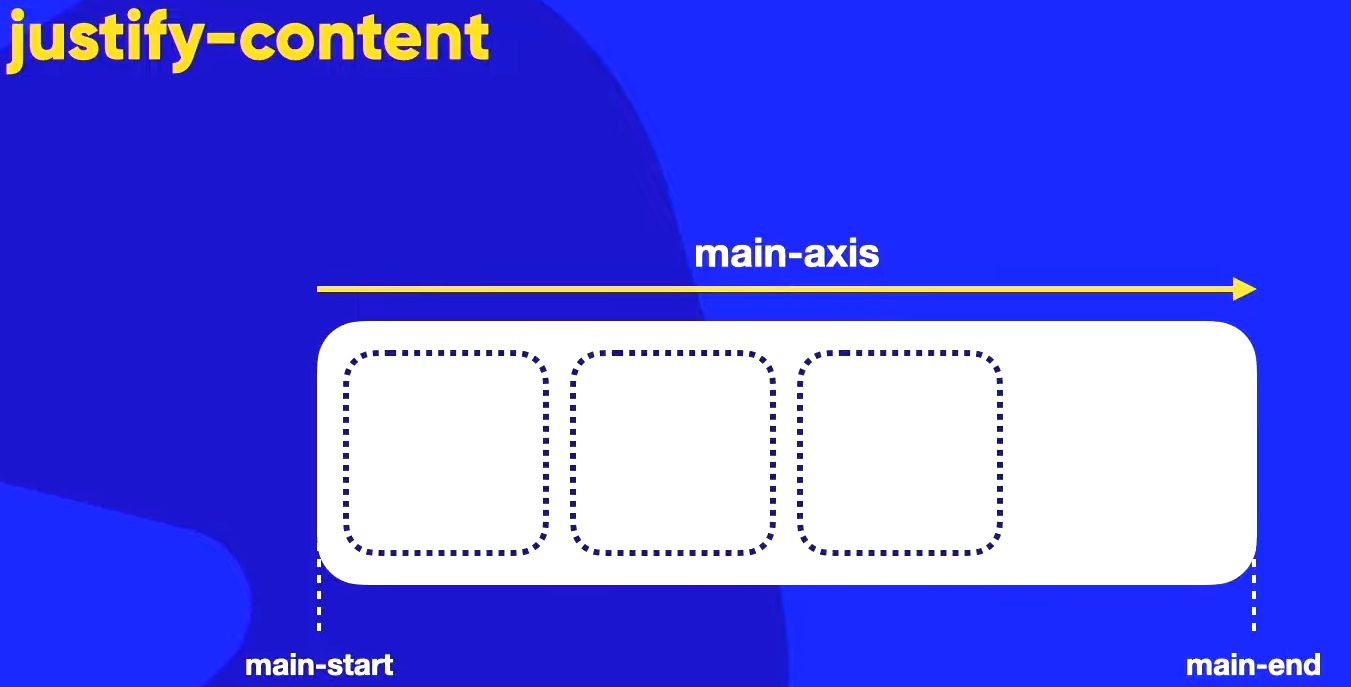
 

Um exemplo um pouco mais prático de como seria em código, e um exemplo descrevendo o comportamento usando outros valores.

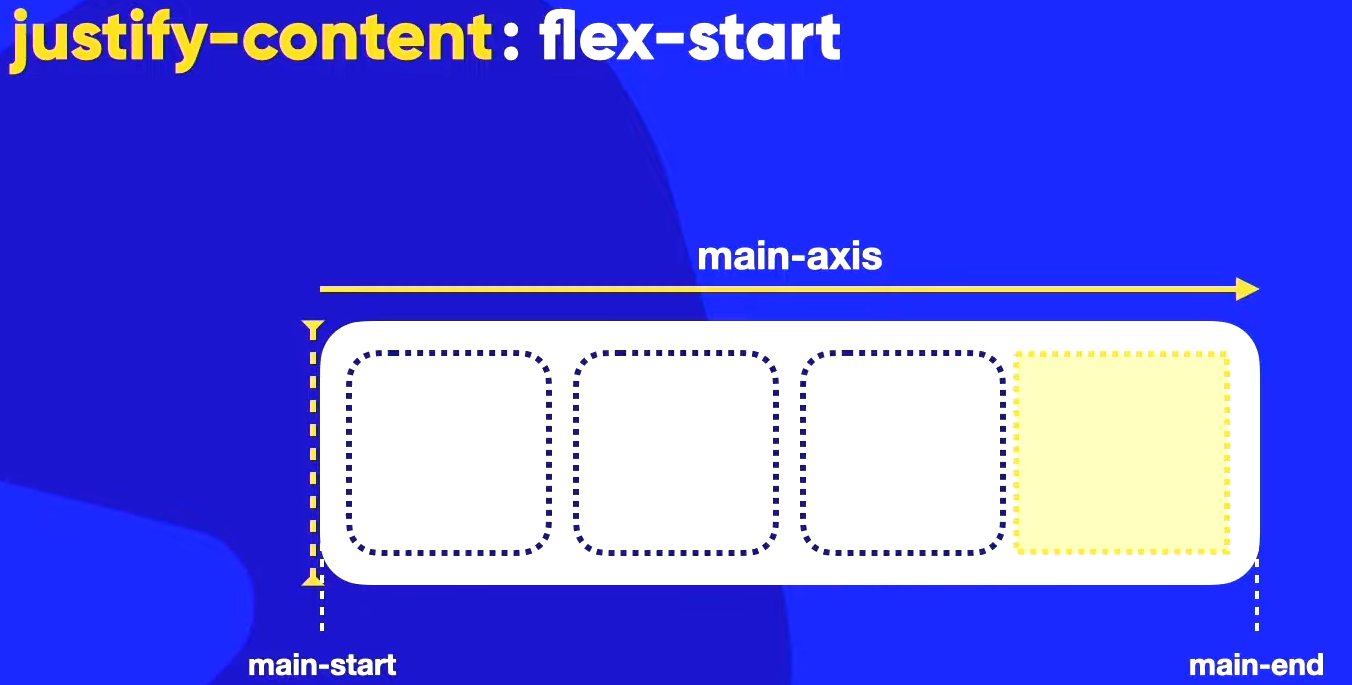
 

Para agilizar a digitação do código temos um **shorthand(shortline)** que une **flex-direction** e **flex-wrap**, que se chama **flex-flow** que recebe os valores **direction** e **wrap** respectivamente.

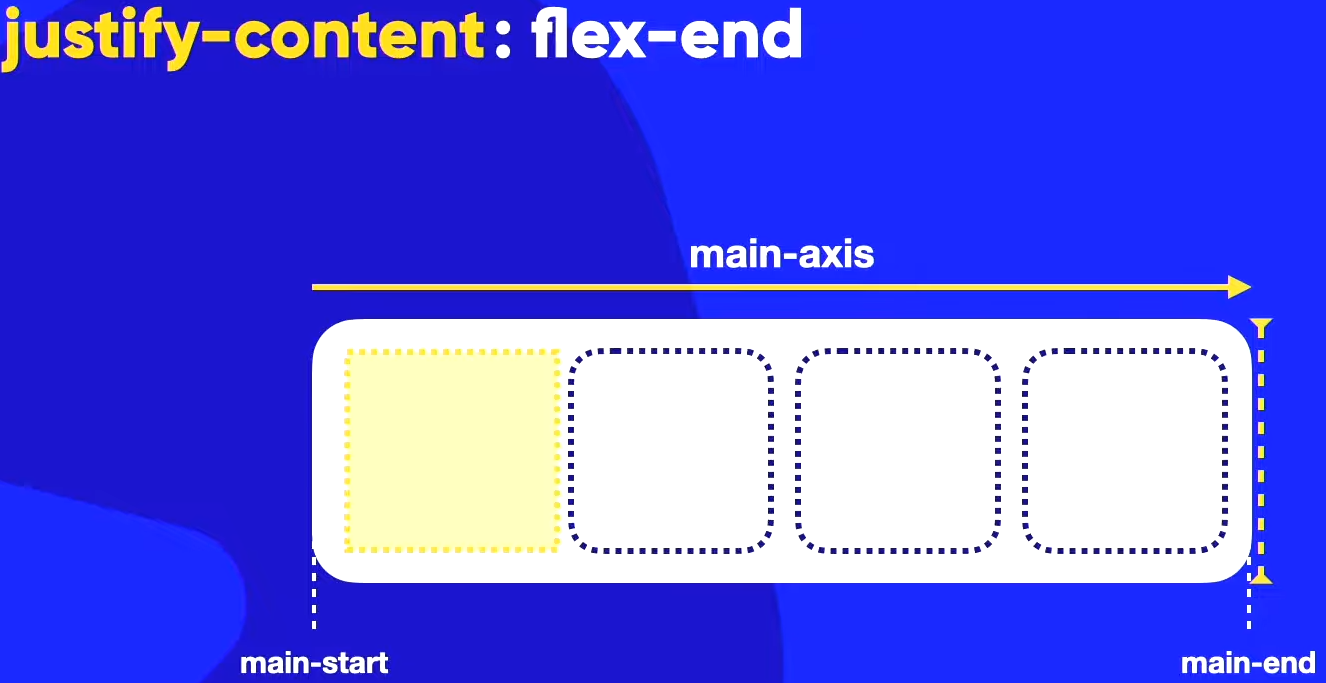




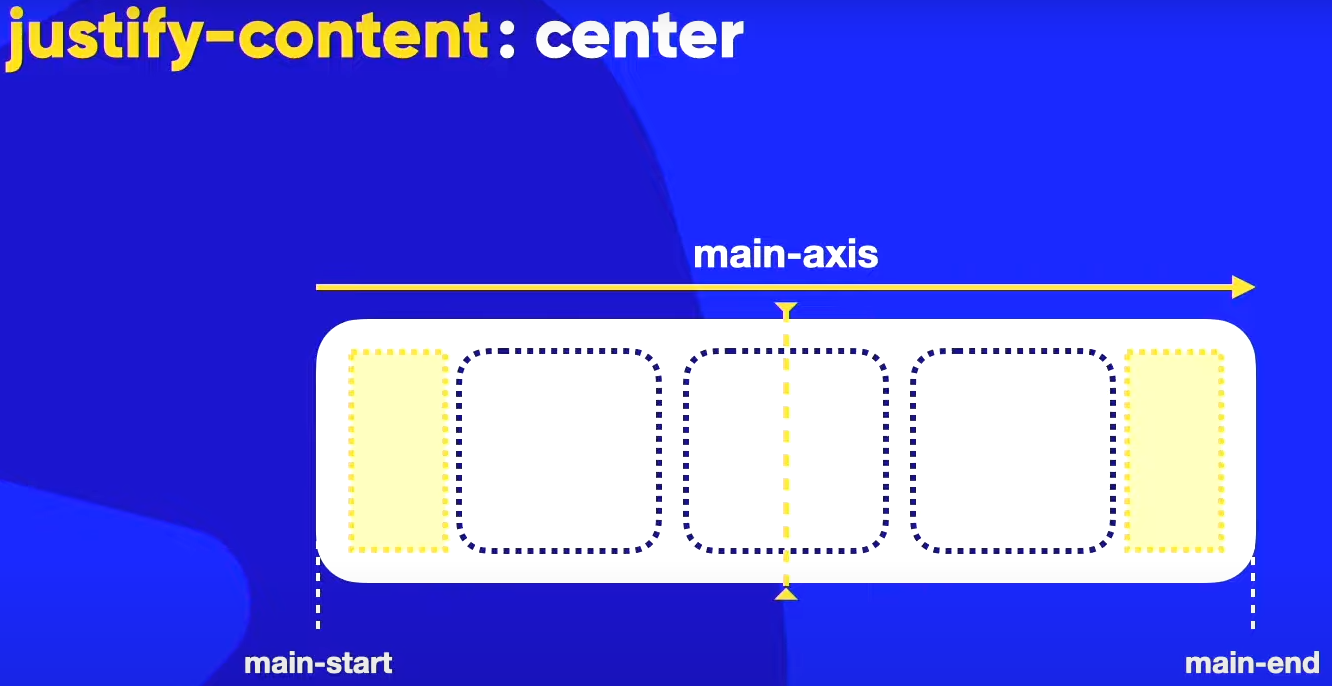
O ***justify-content*** faz um alinhamento juntamente ao eixo ***main-exis***, não confunda com “alinhamento *deitado*” porque nem sempre estará assim, porque nesse exemplo estamos usando ***flex-flow: row nowrap;*** Vale lembrar que main-start e main-end fazem referencia ao inicio e ao fim do ***main-exis***.



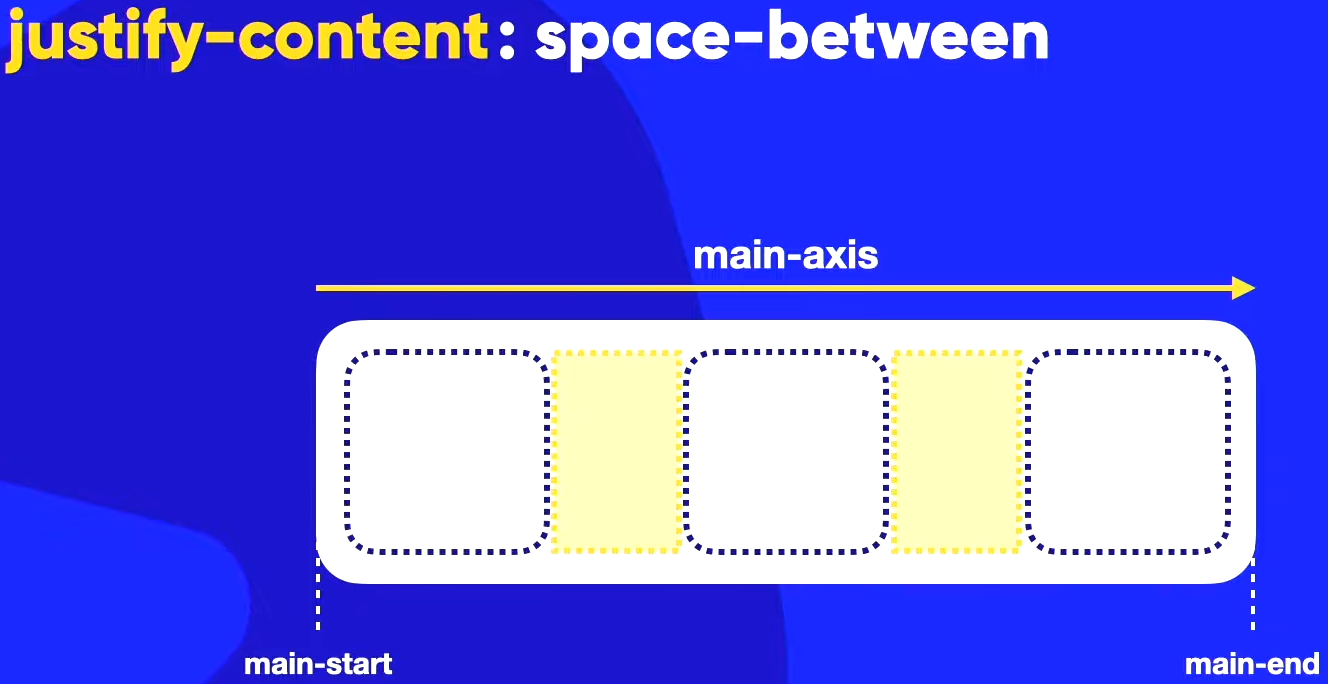
O ***flex-start*** faz com que os objetos fiquem “colados” no ***main-start***, deixando todo o final como uma “área livre” independentemente do tamanho dessa área.



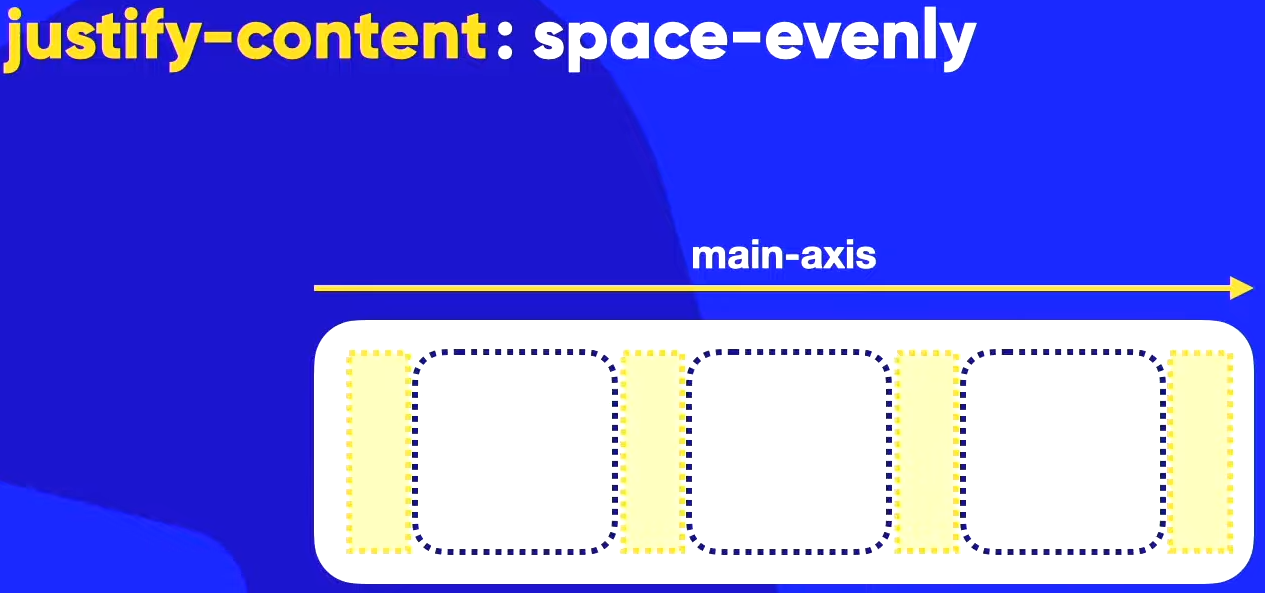
O mesmo que ***flex-start***, porém ***flex-end*** fixa os objetos em relação ao ***main-end***. Posso fazer uma menção ao ***flow***, se usarmos um ***row*** reverse vamos ter o mesmo comportamento mais a ordem dos objetos vão ser bem diferentes se usarmos somente ***row***.



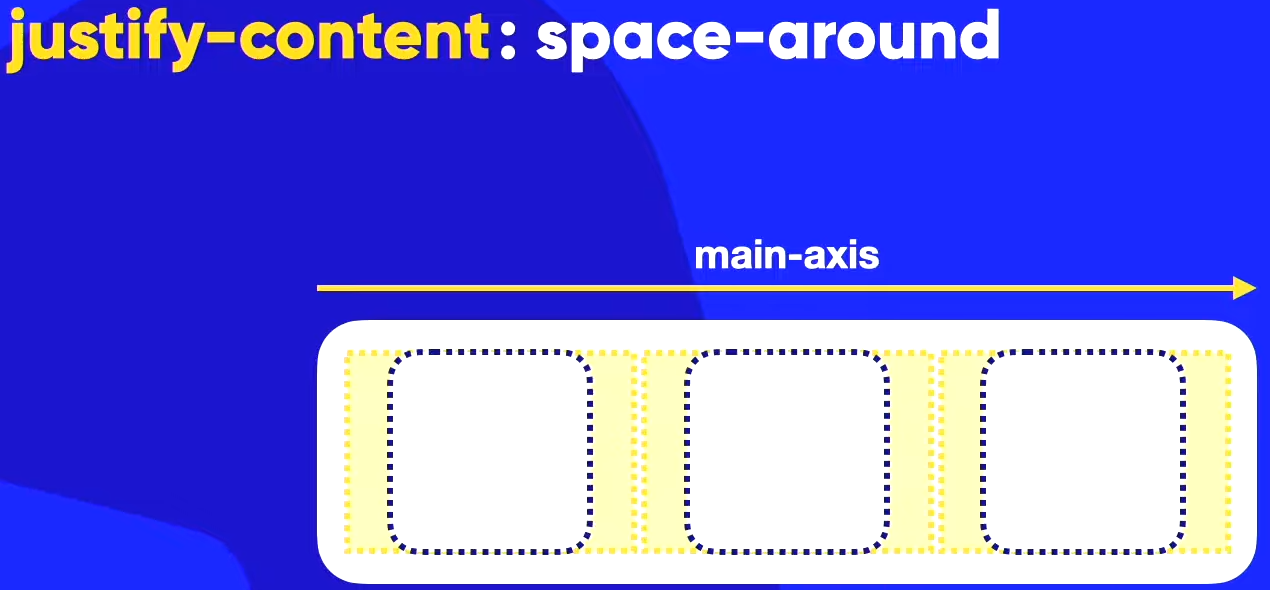
Aqui o *center* centraliza os objetos ao centro do ***main-axis***, e calcula de forma automática esse centro. Deixando os espaços em branco exatamente do mesmo tamanho.



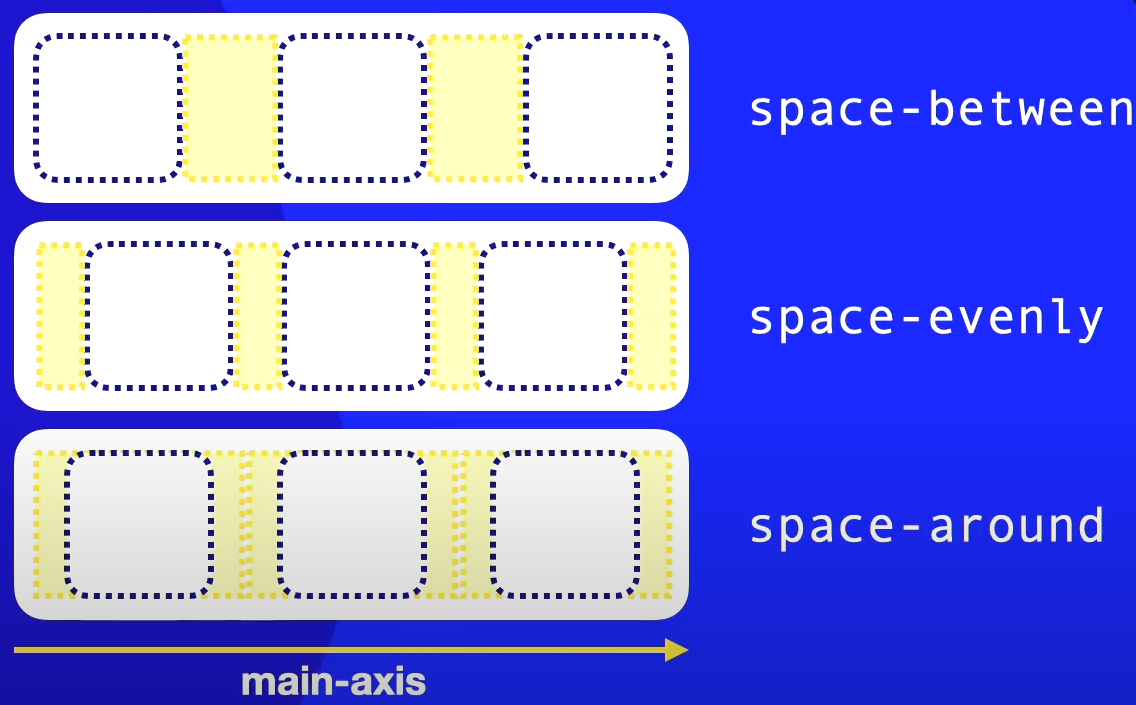
O ***space-between*** força os objetos ficarem em ***main-start*** e ***main-end*** e os espaços entres os demais objetos vão ser distribuídos por igual.



Em ***space-evely*** todos os espaços entre, antes e depois dos objetos terão o mesmo tamanho.

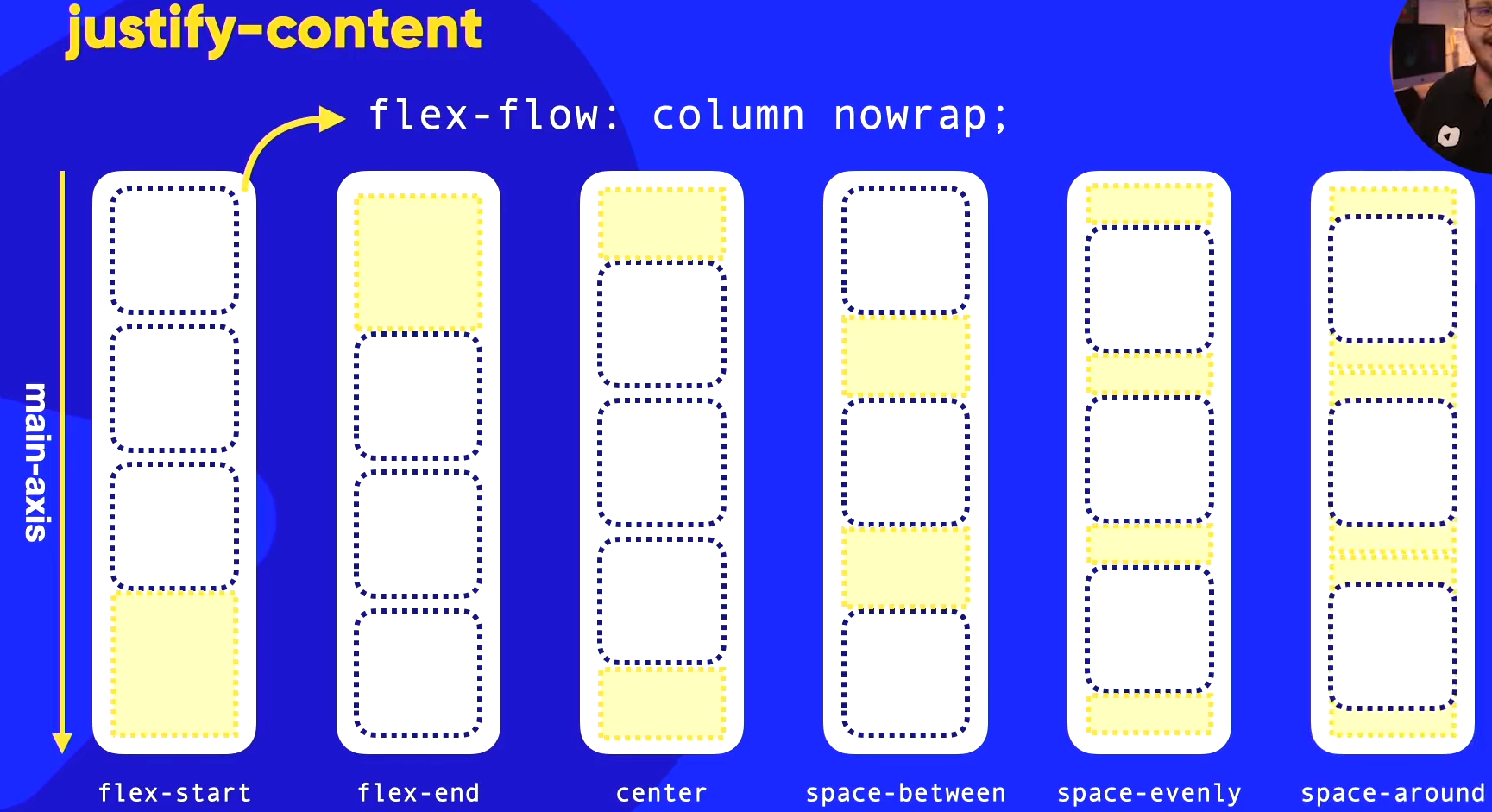


De todos os “***space***” esse é o mais diferente. Aqui é calculado o tamanho total da tela e é dividida em quantidade dos objetos. No exemplo temos 3 objetos, então a tela é dividida em 3 partes e assim cada objeto é centralizado na sua respectiva parte da tela. Assim ***space-around*** define o último dos “***space***”



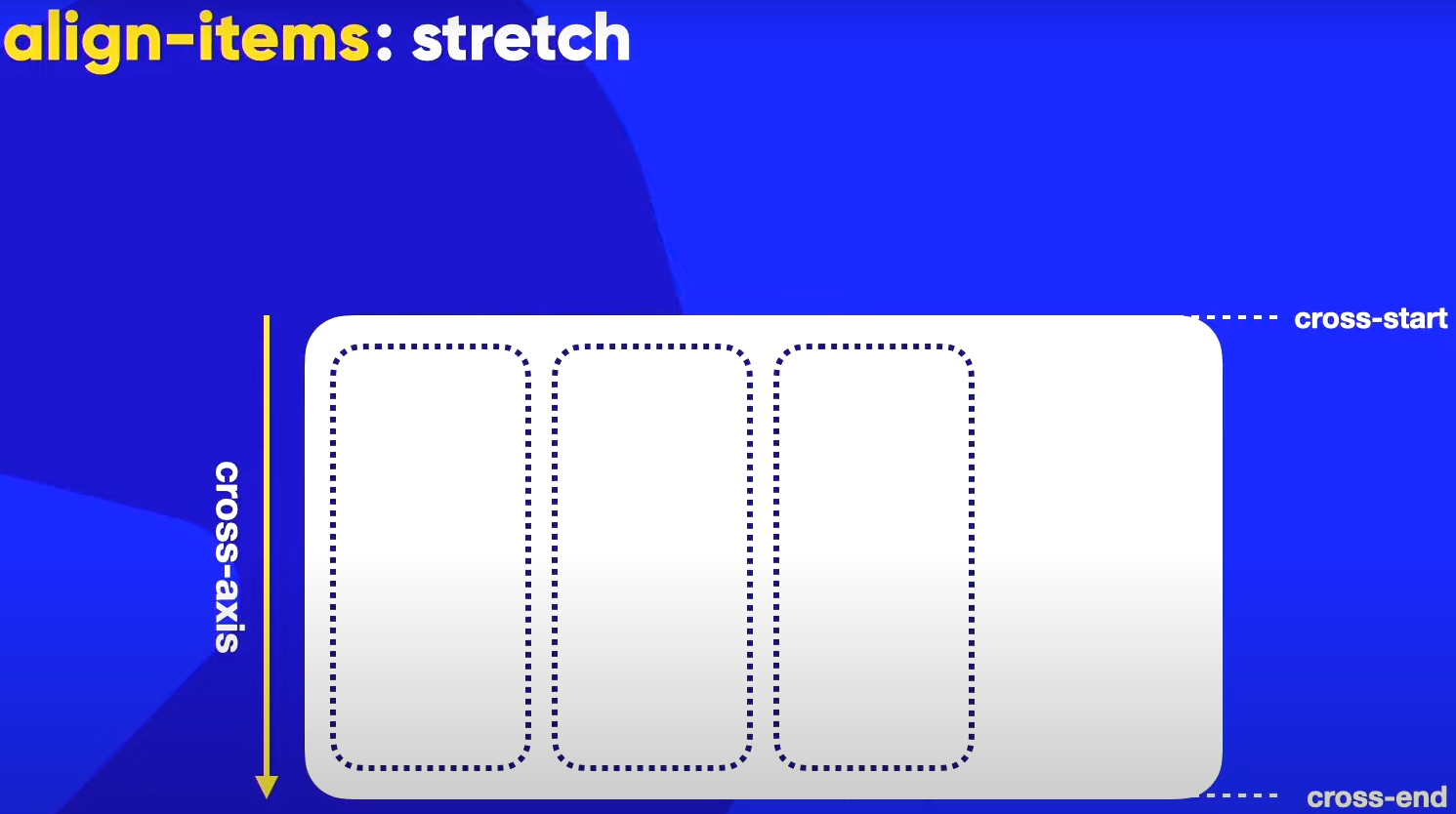
Se houver algum questionamento, sobre os “space” serem parecidos, temos aqui as suas diferenças sendo mostradas lado a lado.

Até agora mostramos os ***justify-content*** em ***flex-flow: row nowrap;*** mas como que ficariam em ***column***, e aonde que o eixo ***main-axis*** ficaria?

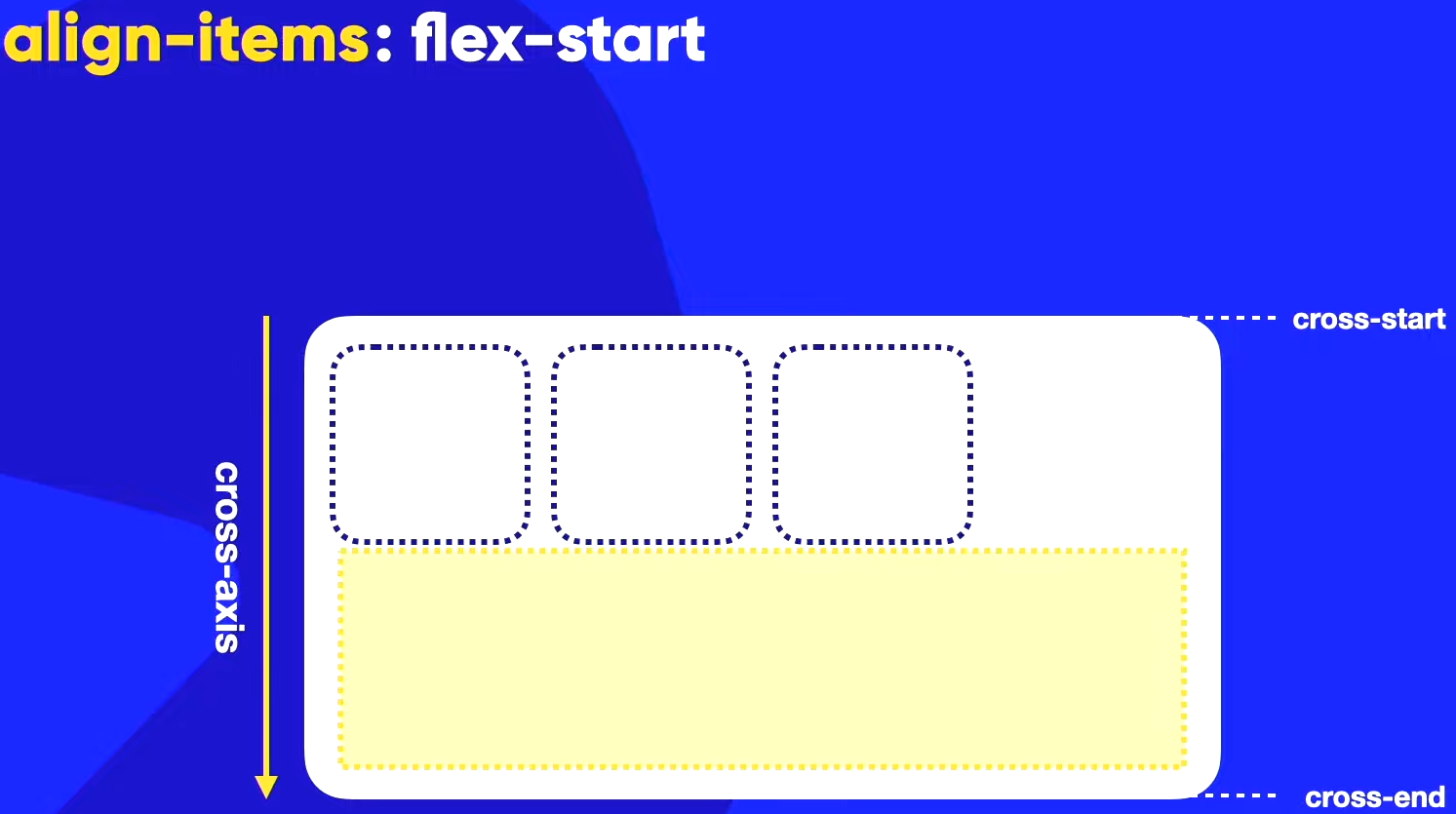




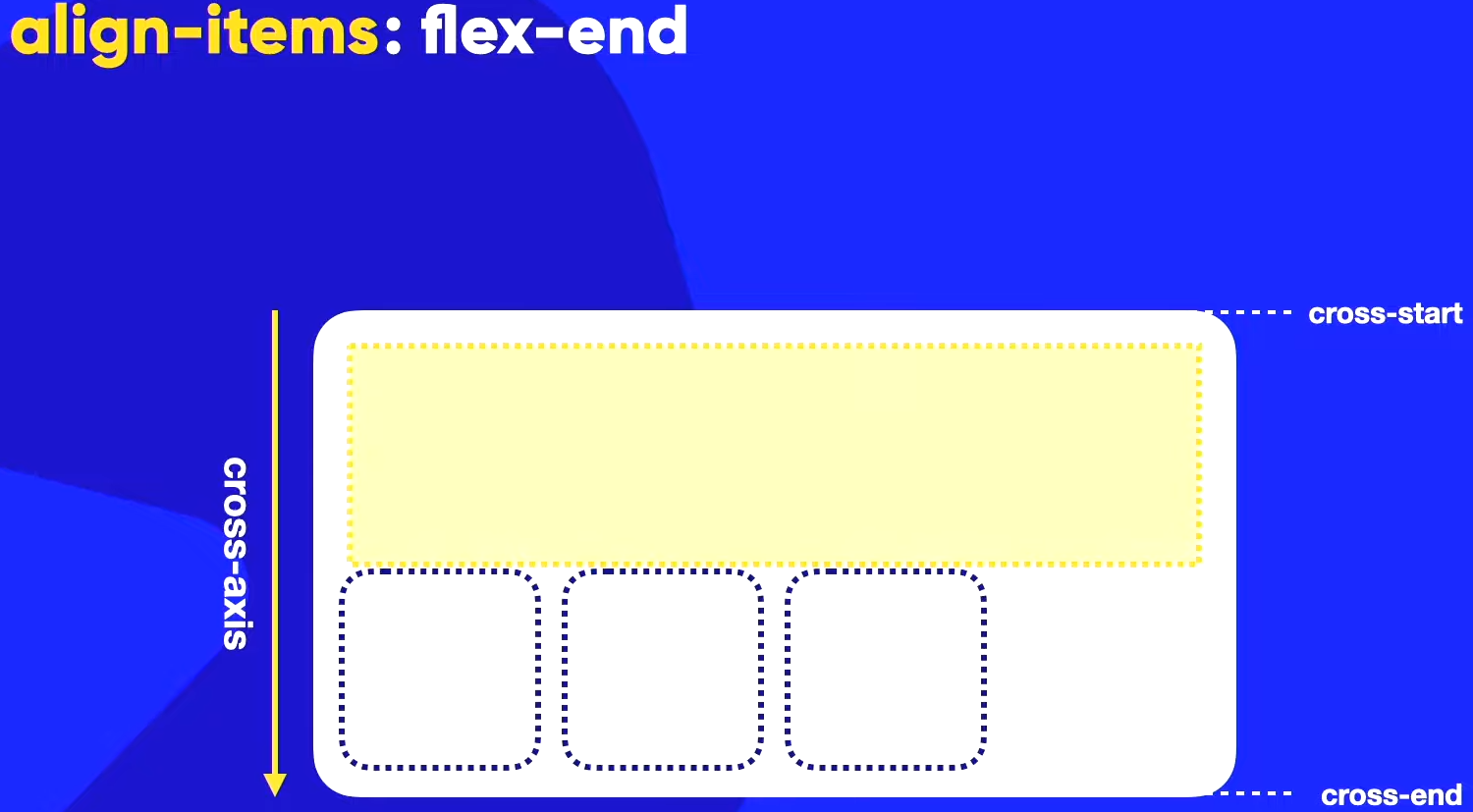
Em ***aling-items*** temos o alinhamento em ***cross-axis***, e como no último exemplo tempos o ***cross-start*** e ***cross-end***.



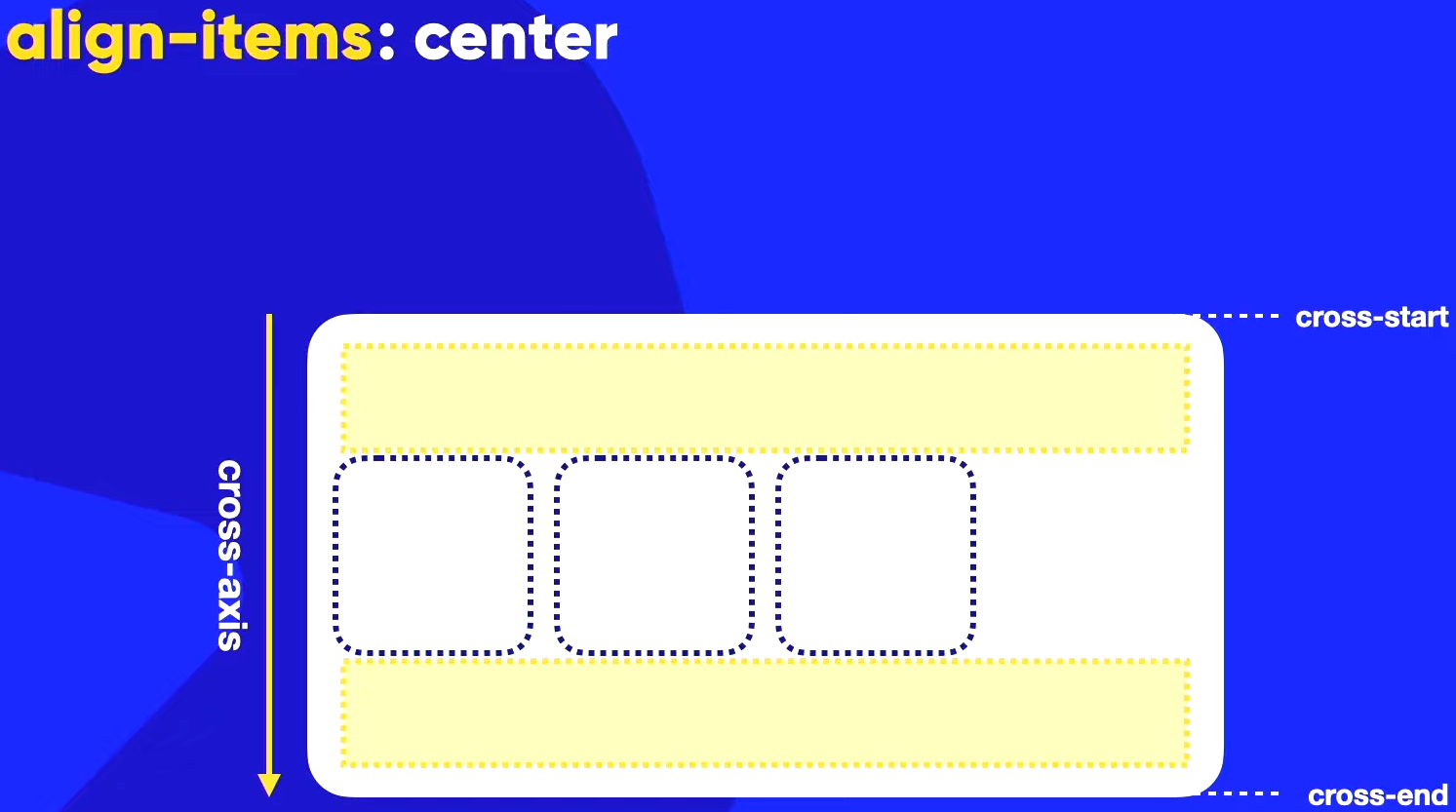
Por padrão o ***align-items*** não tem por padrão ***flex-start***, o padrão aqui seria ***stretch***. Que significar esticar, por isso para bem demonstrar aumentamos o tamanho da tela, como visto os elementos se esticam junto com o tamanho da tela.



Assim como no anterior temos ***flex-star***, que irá colocar nossos objetos no topo da tela.

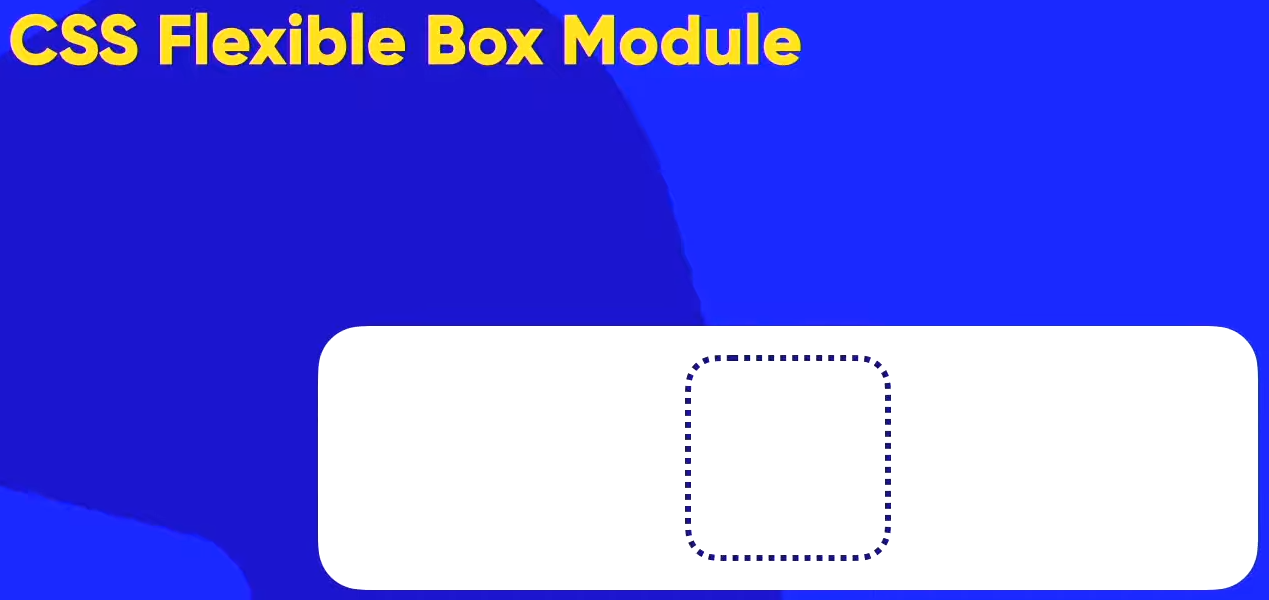
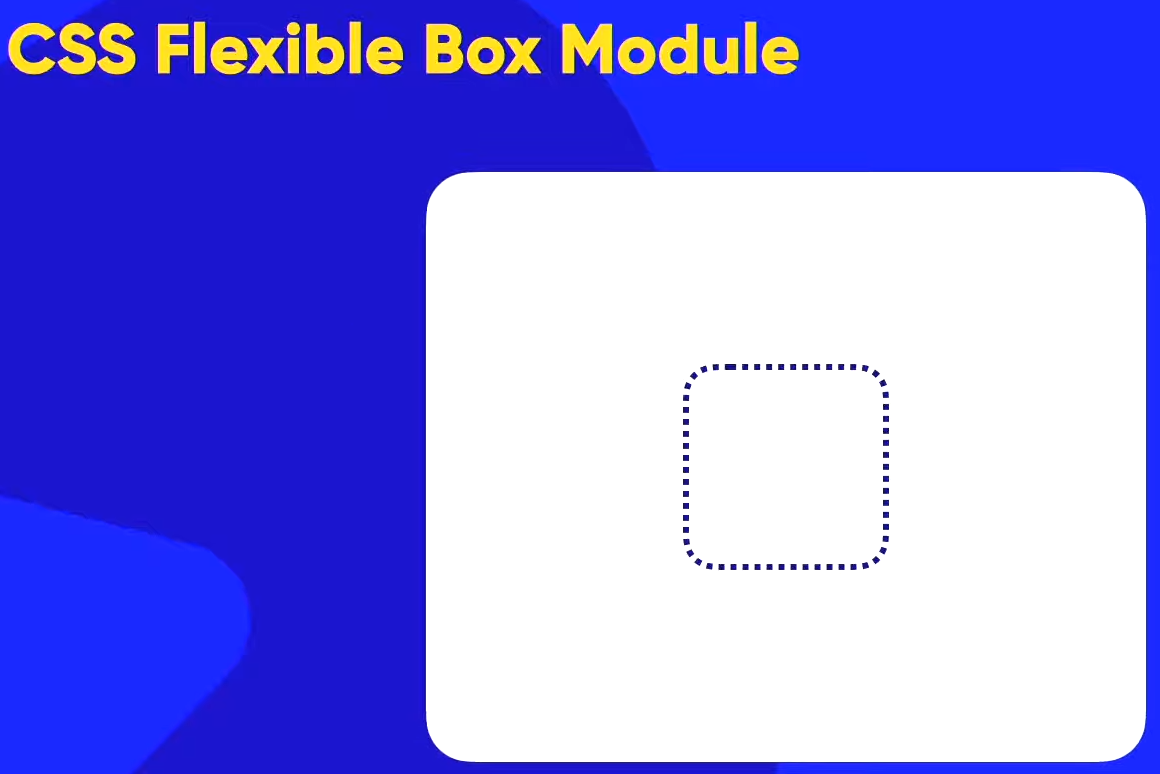


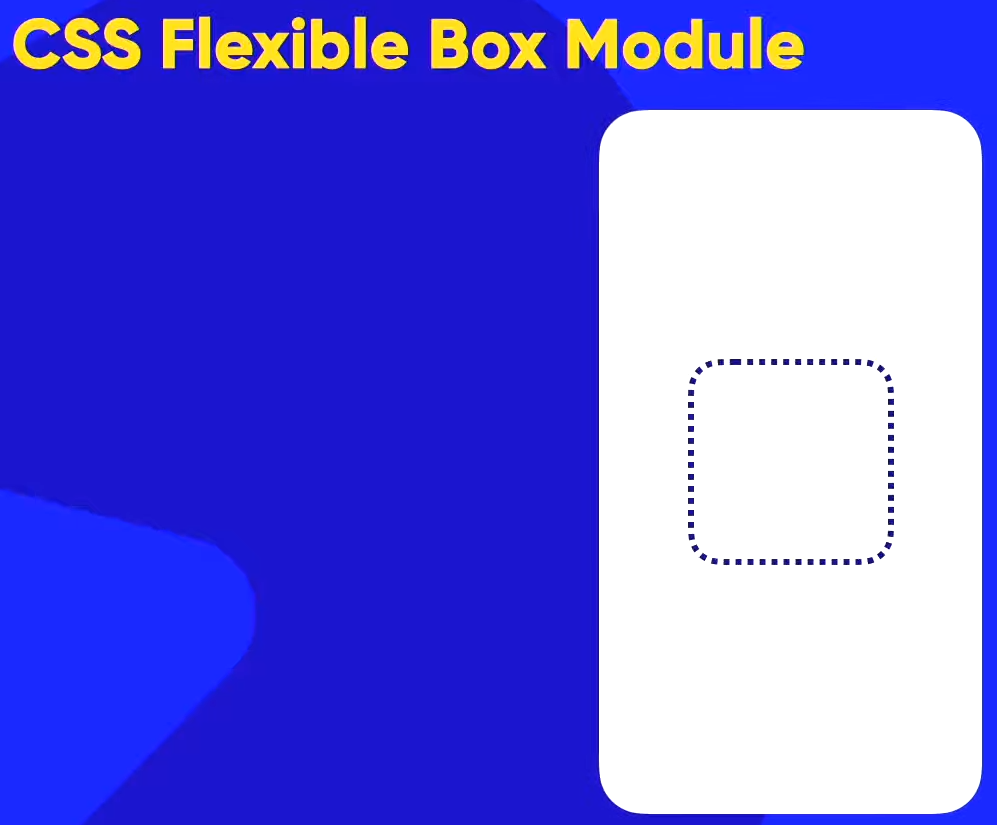
E também vamos ter o ***flex-end*** que trará nossos objetos para o final da tela

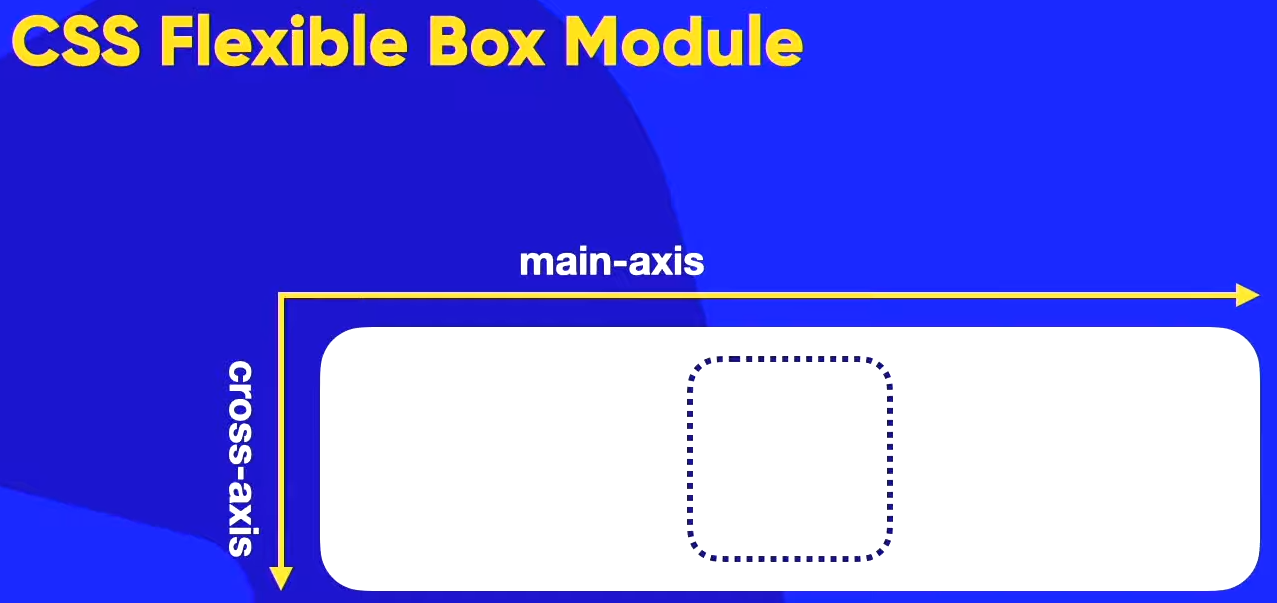


E por fim, o ***center*** que ira centralizar nosso objeto em razão do eixo ***cross-axis***.

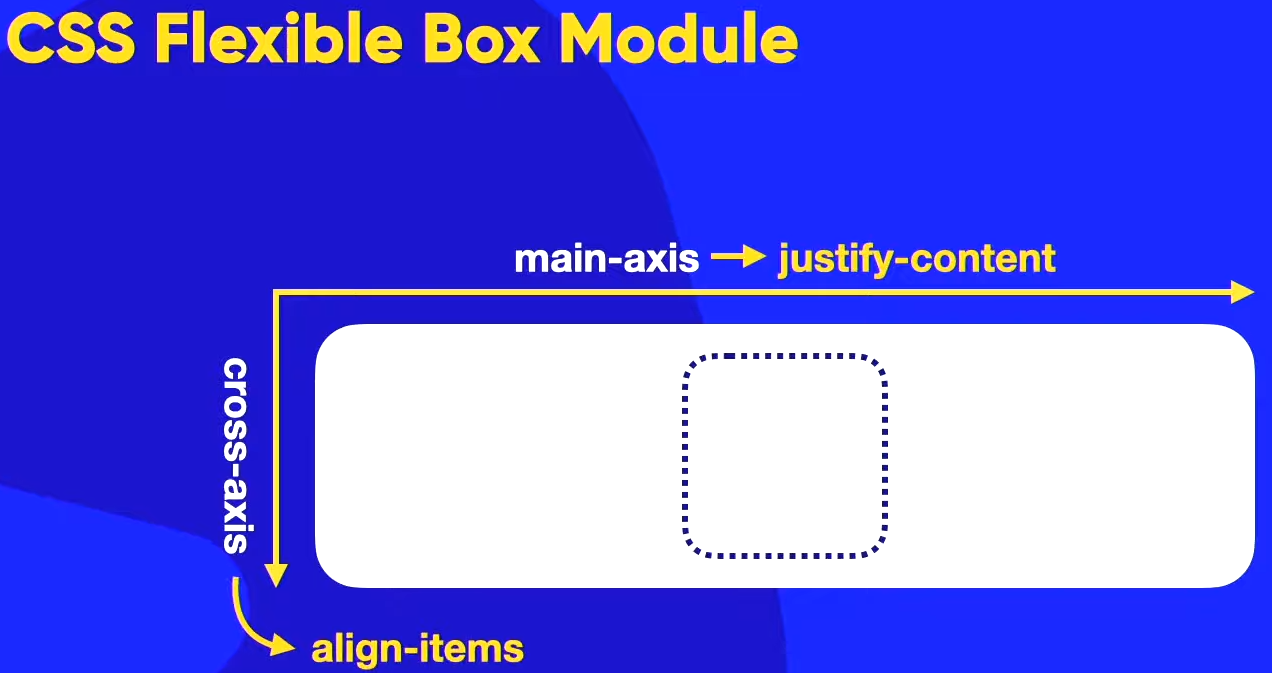
Agora oque vamos ver é uma “centralização absoluta”, onde não importa o tamanho da tela, o objeto sempre irá se posicionar no centro do elemento pai. Para ter um exemplo, realizamos o desafio da tela de login, e realizamos muitas “gambiarras” no ***css*** para estilizar e centralizar nossos objetos, e aqui basta 2 linhas e o objeto irá se comportar de modo muito mais assertivo.

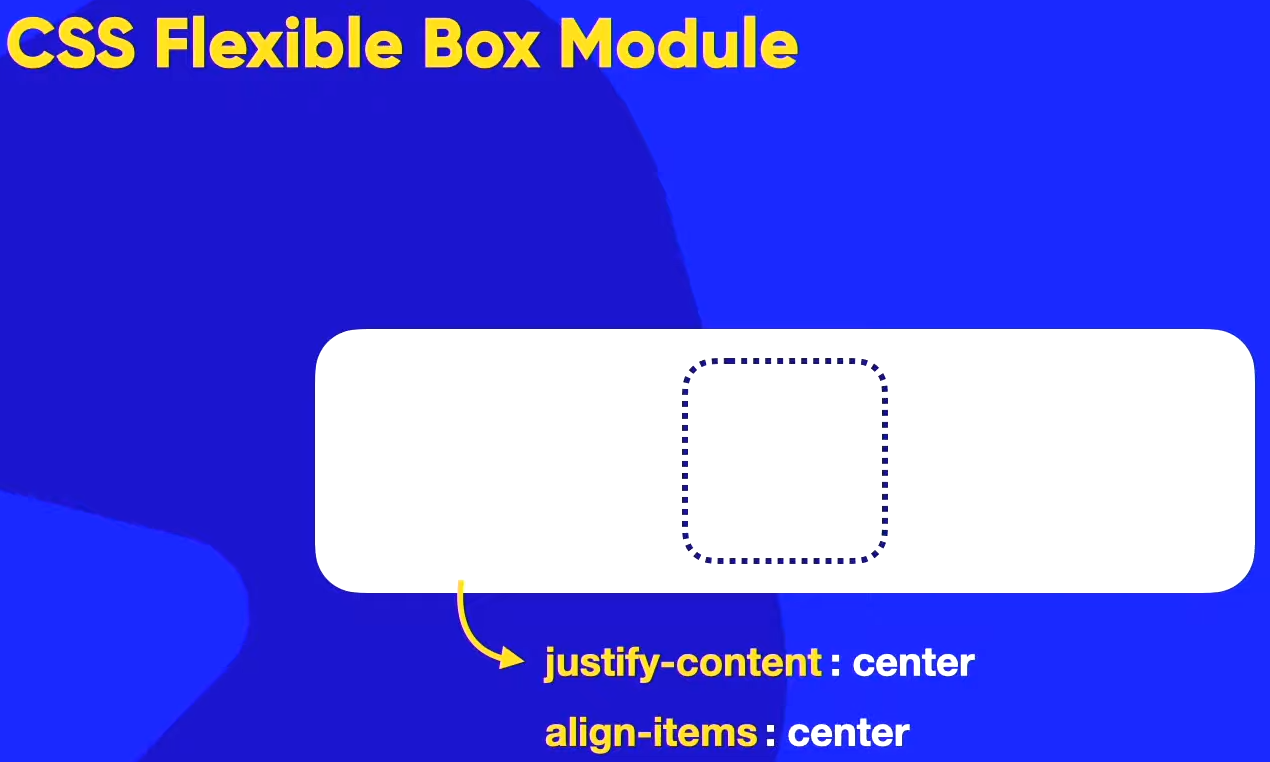




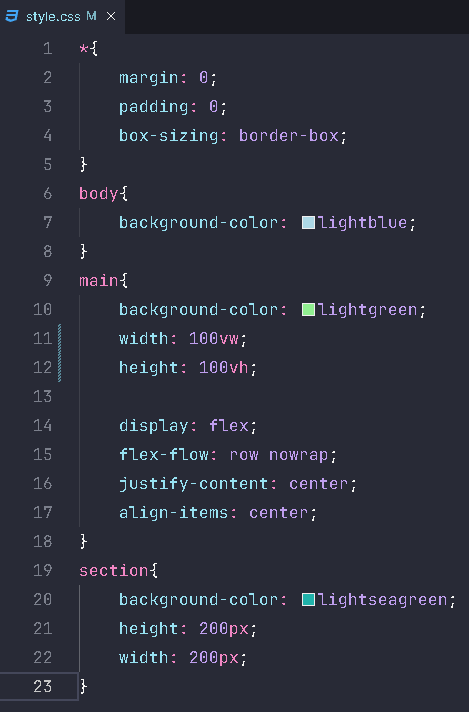
Primeiro vamos relembrar quais são os eixos do nosso elemento pai, ***main-axis*** e ***cross-axis***.



E novamente para recapitular, o eixo ***main-axis*** ajustamos com ***justify-content*** e o eixo ***cross-axis*** modificamos com ***align-items***.



E nos dois vamos passar o valor ***center***, e como “num passe de mágica” nosso objeto centraliza de forma absoluta. Vale lembrar que essas configurações ***css*** são aplicadas no elemento pai, para que surtam o efeito desejado.



Nosso código final, se comportaria dessa maneira.