**FLEXIBLE BOX (Caixa Flexível)**

Me permite alinhar elementos filhos dentro de um elemento pai de forma dinâmica.

**PROPRIEDADES**

**Flex-direction:** define o eixo principal dos elementos (horizontal ou vertical). Por padrão é row/linha.

**Valores**

* Row;
* Row-reverse;
* Column;
* Column-reverse.

**Justify-content:** permite definir o alinhamento do eixo principal.

**Valores**

* Flex-start;
* Flex-end;
* Center;
* Space-between;
* Space-around

**Align-items:** permite alinhar os elementos do eixo secundário/perpendicular.

**Valores**

* Flex-start;
* Flex-end;
* Center;
* Stretch (obs. Os elementos filhos não devem ter um height definido);
* baseline

**Flex-wrap:** por padrão os elementos tentam sempre caber em única linha – nowrap.

**Obs.:** há uma propriedade chamada **flex-flow** que me permite definir valores de Direction e wrap. Ex: **flex-flow: row wrap**.

**Valores**

* nowrap;
* wrap;
* wrap-reverse

**Align-content:** me permite alinhar elementos que ficam em mais de uma linha. Utilizamos sempre com wrap.

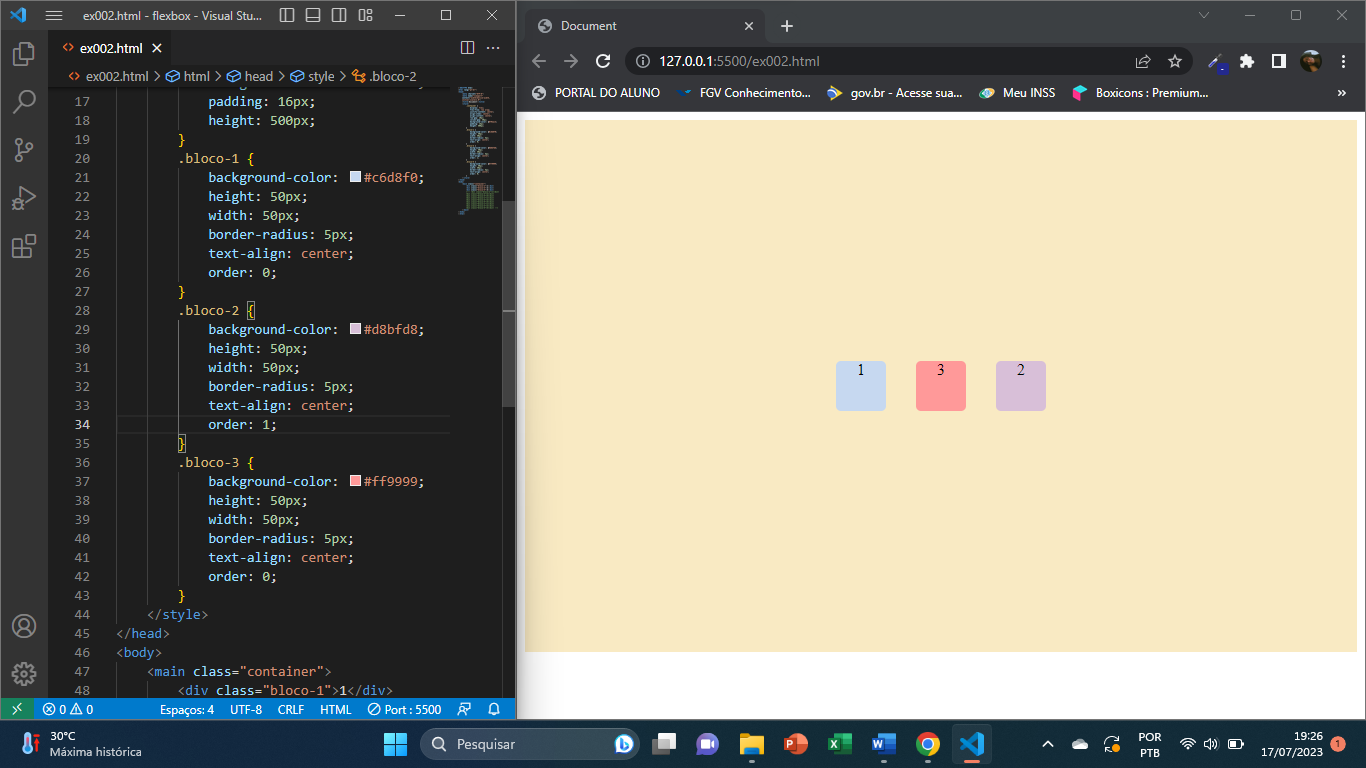
**Valores**

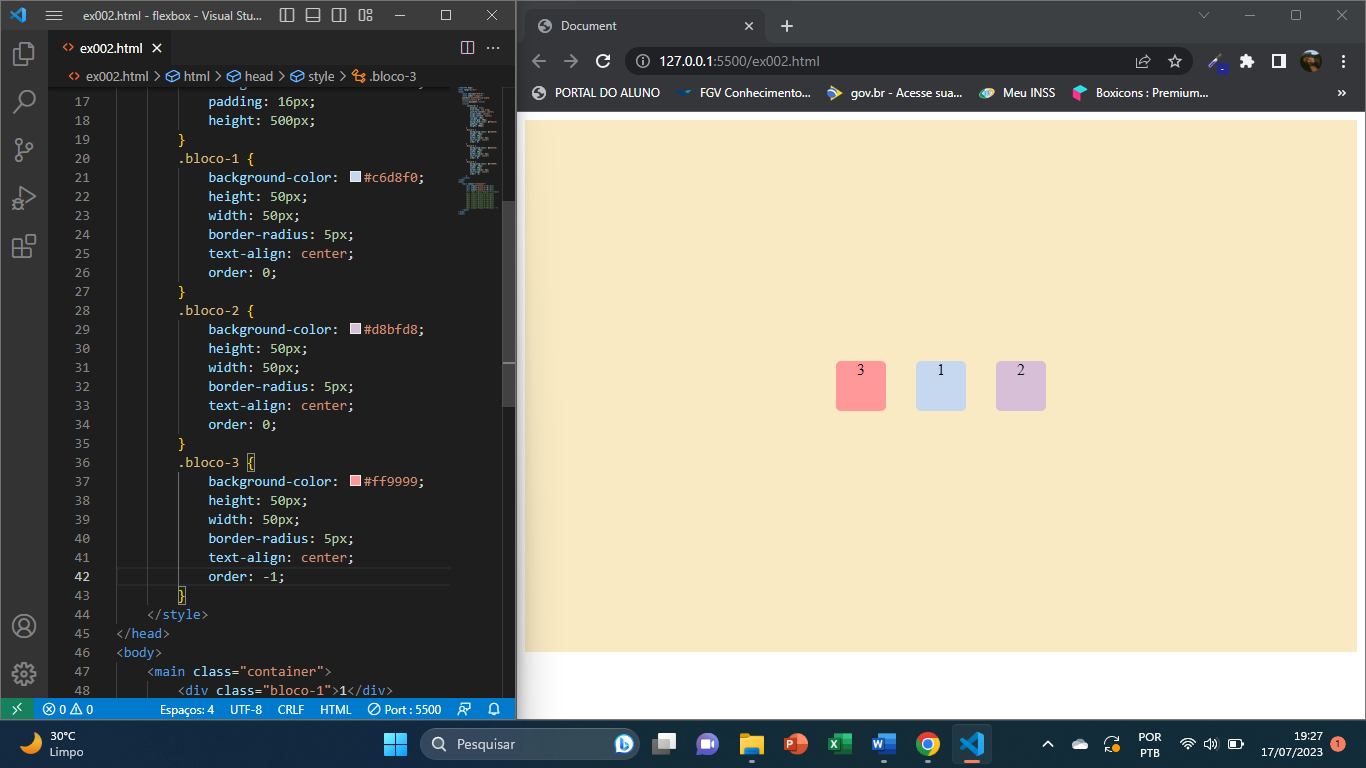
* flex-start;
* flex-end;
* center;
* space-between;
* space-around

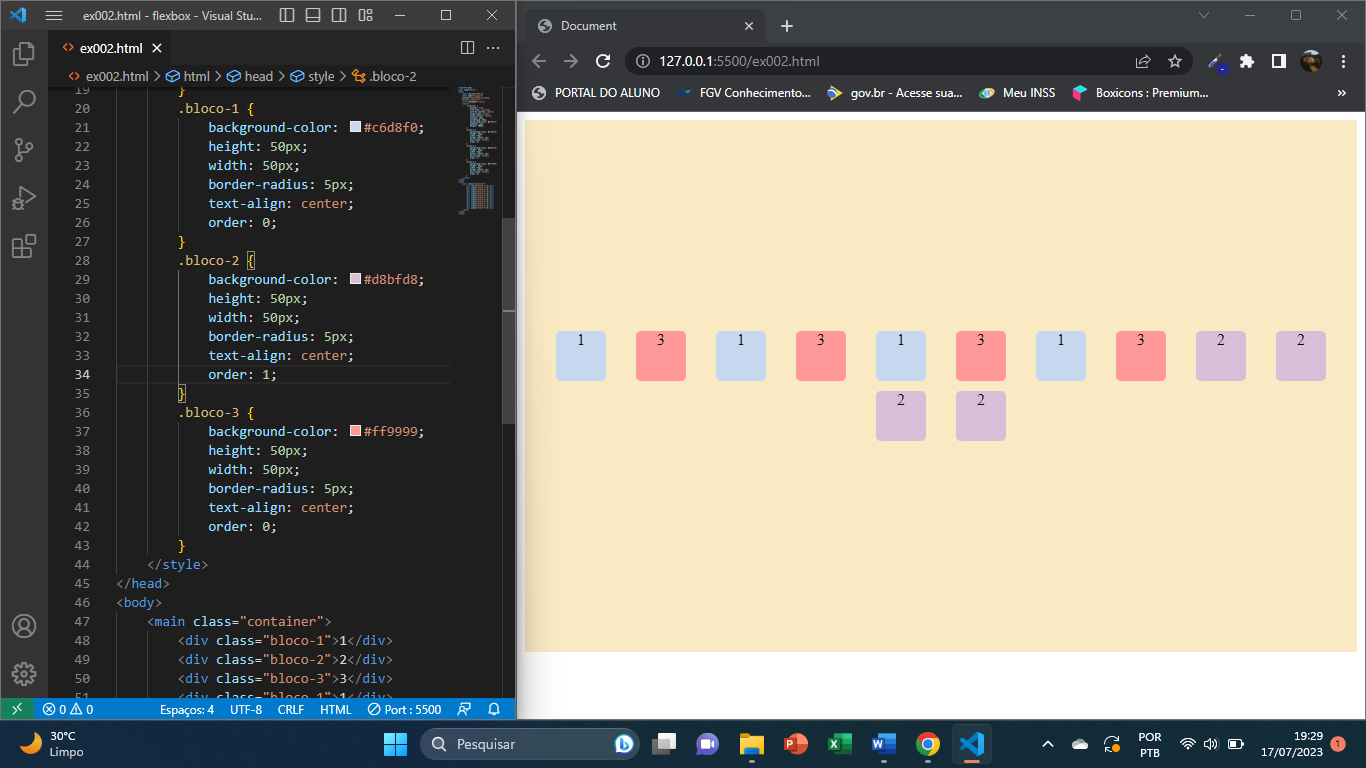
**Gap, row-gap e column-gap**

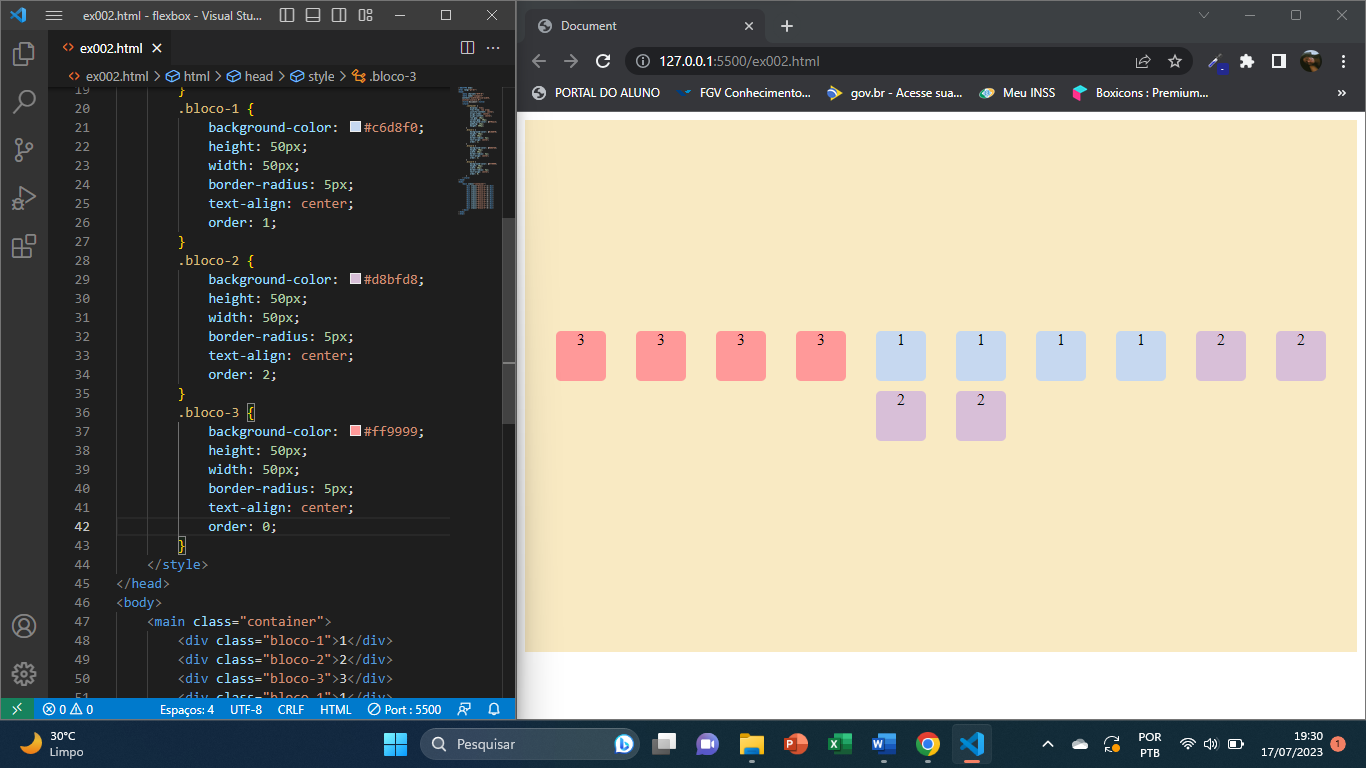
**PROPRIEDADES PARA ESTILIZAÇÃO DOS FILHOS**

**Order:** me permite alterar a ordem dos elementos filhos. Inicialmente é como se todos elementos fossem order: 0, sendo assim a ordem dependerá exclusivamente do conteúdo HTML. É possível usar números positivos e negativos.

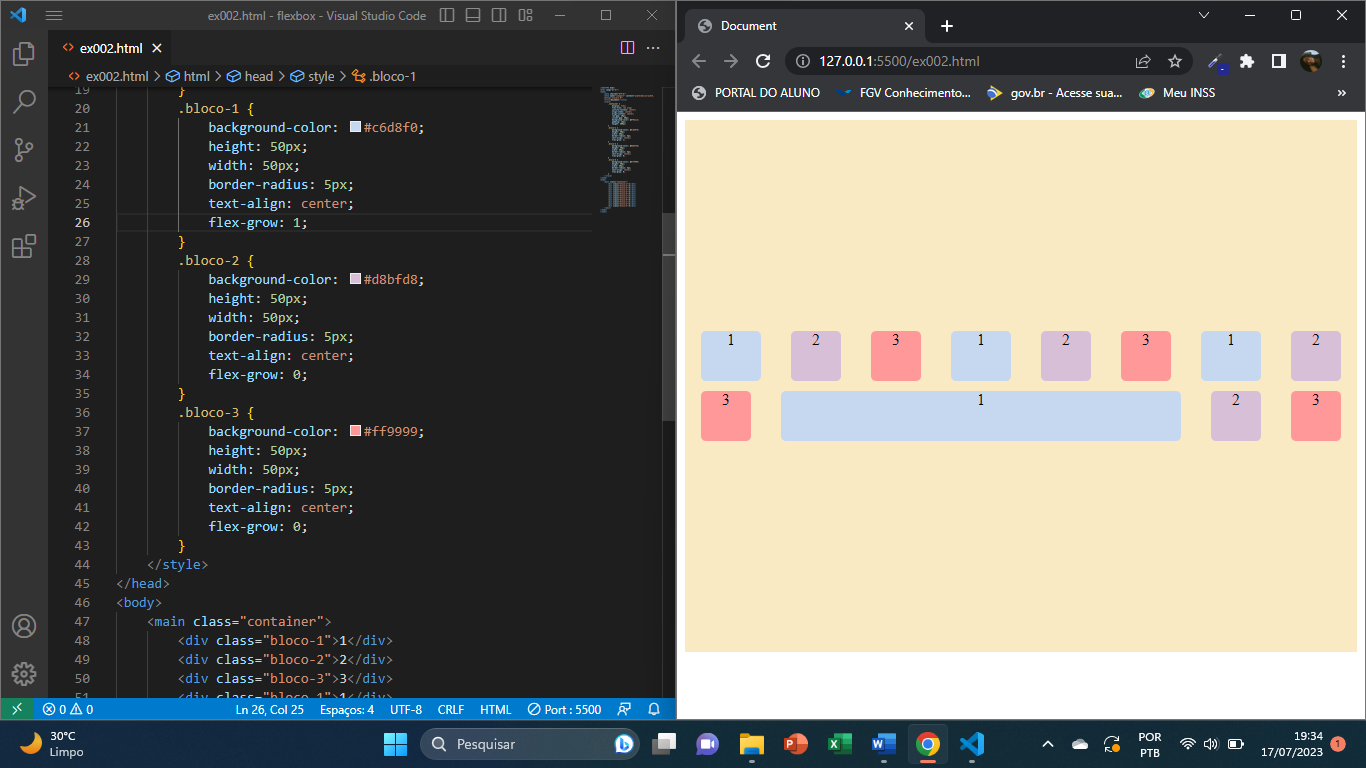




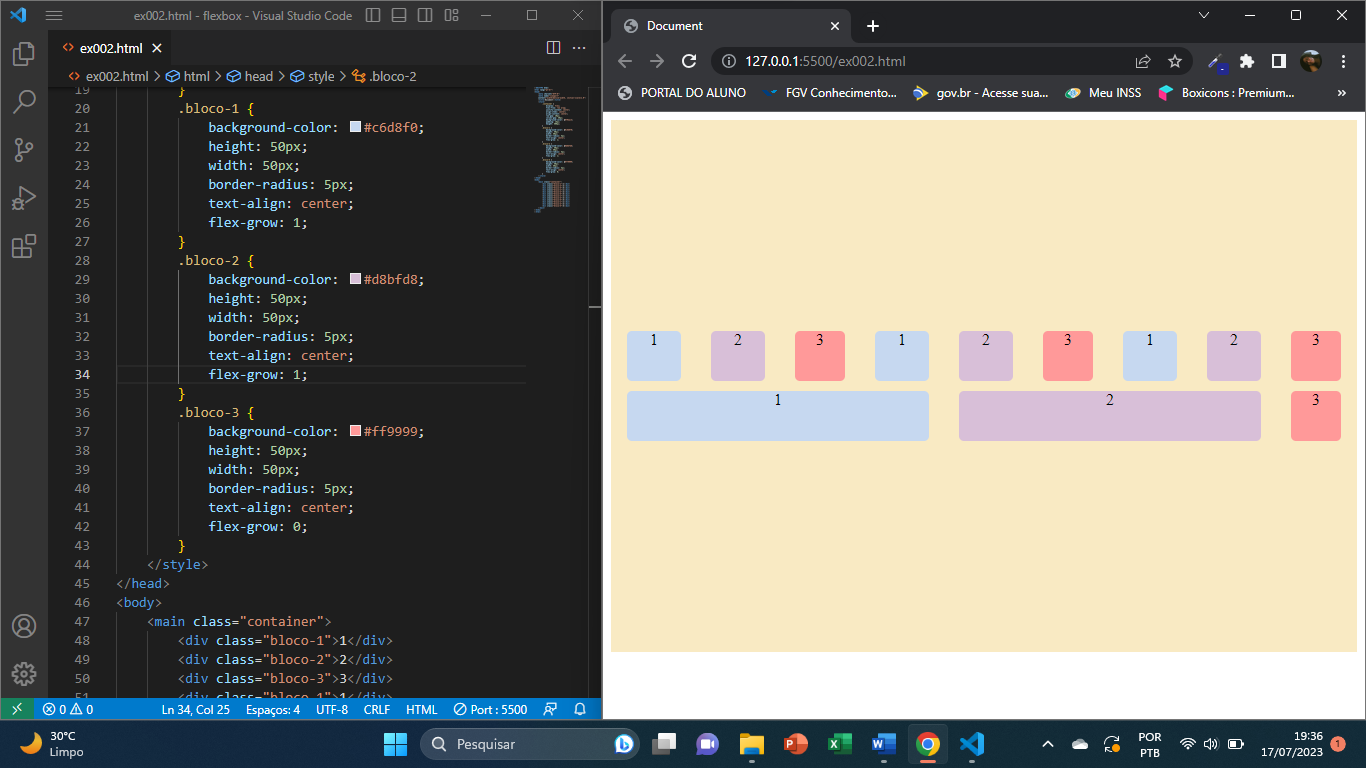




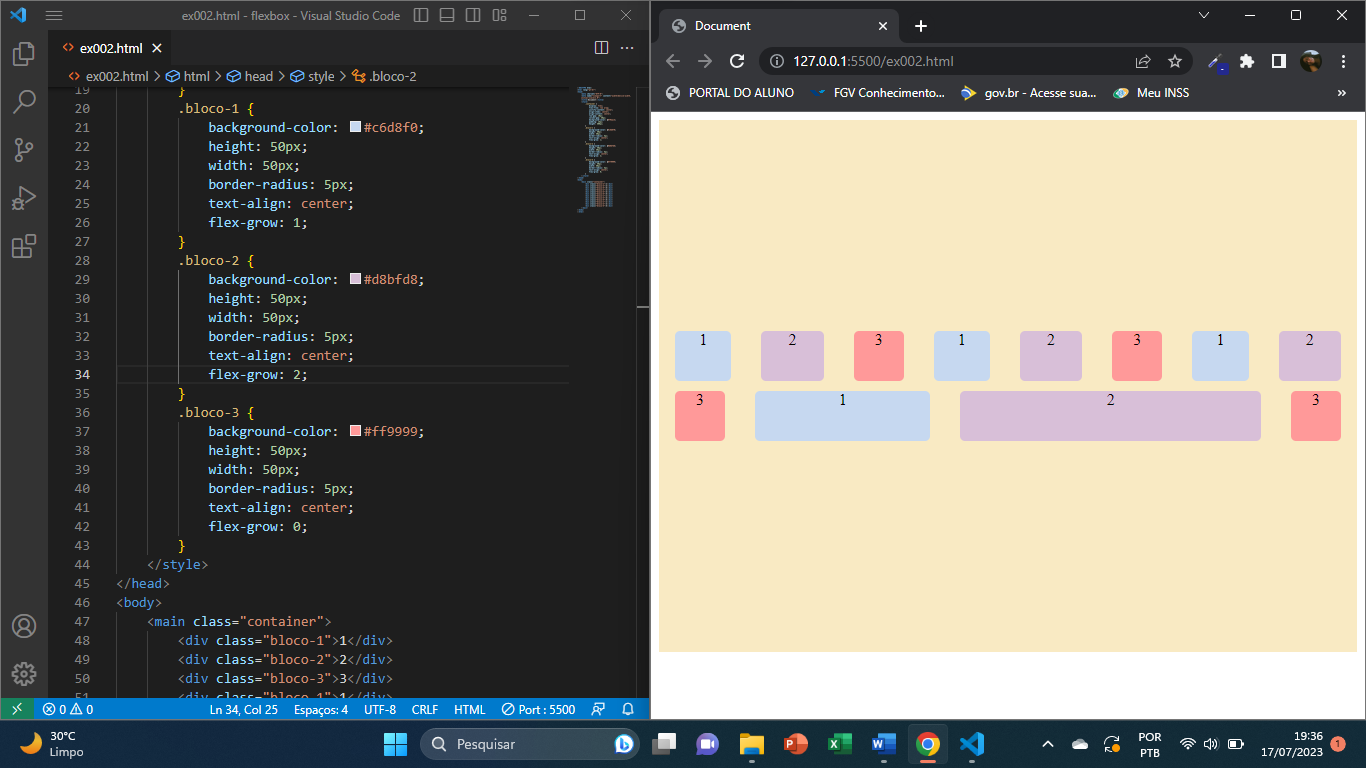
**Flex-grow:** define a possibilidade de um item/filho poder crescer caso seja necessário. Por padrão todos possuem valor 0 (não crescerão/ausência da possibilidade).

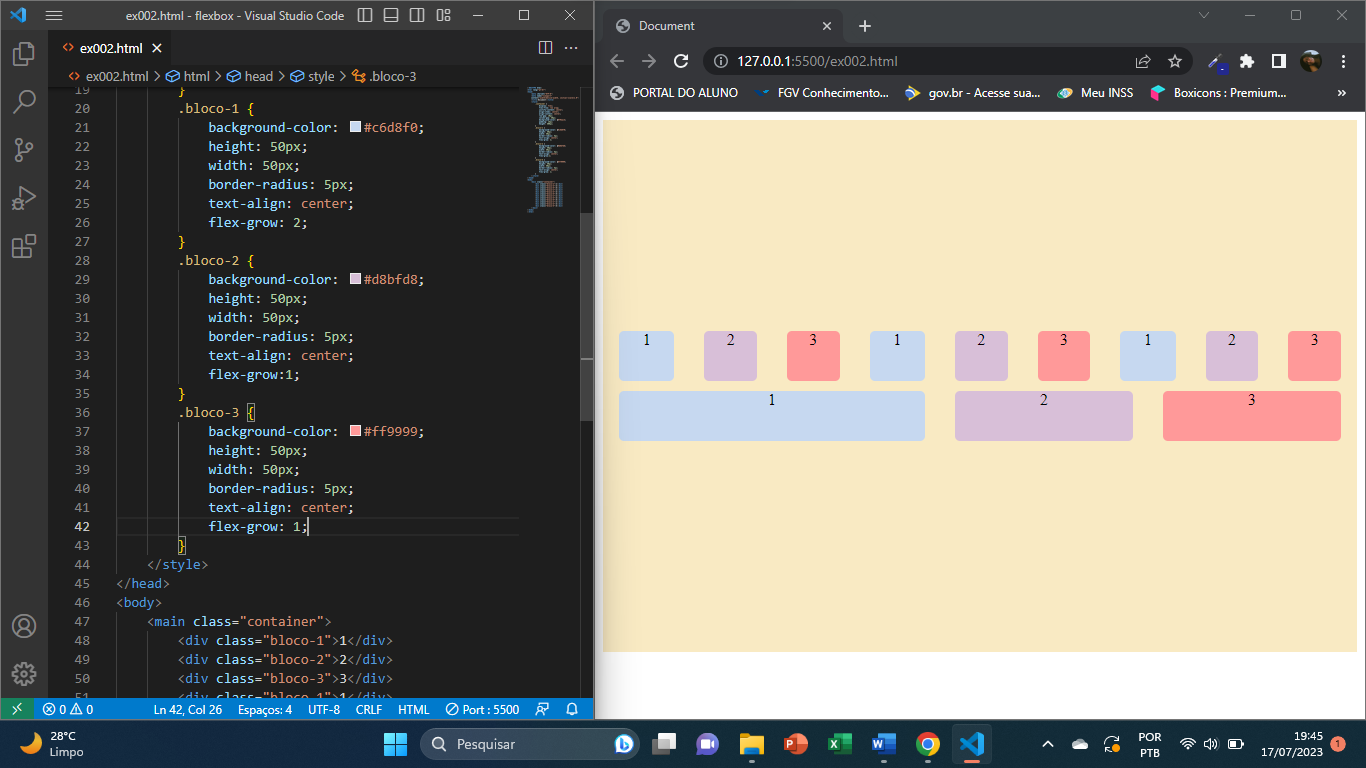


A numeração que definimos é como se fosse uma proporção



Com dois blocos definidos com valor 1 crescerão igualmente.



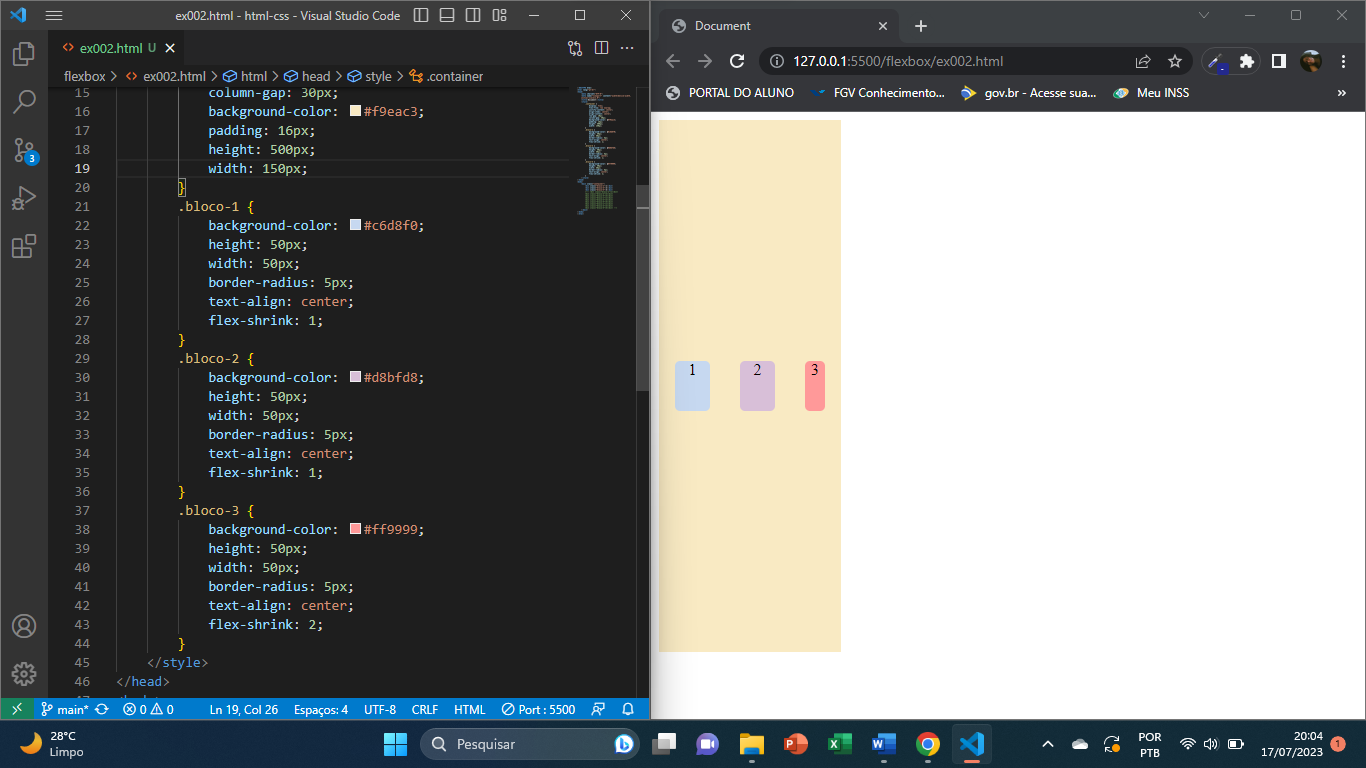


Observação: os blocos estão definidos com width de 50px. Se for acrescentado 10px para aqueles que possuem valor 1 de flex-grow, para o valor 2 será o dobro desse acréscimo – 20px. Logo, ficaria [ valor 1: 60px ], [ valor 2: 70px ].

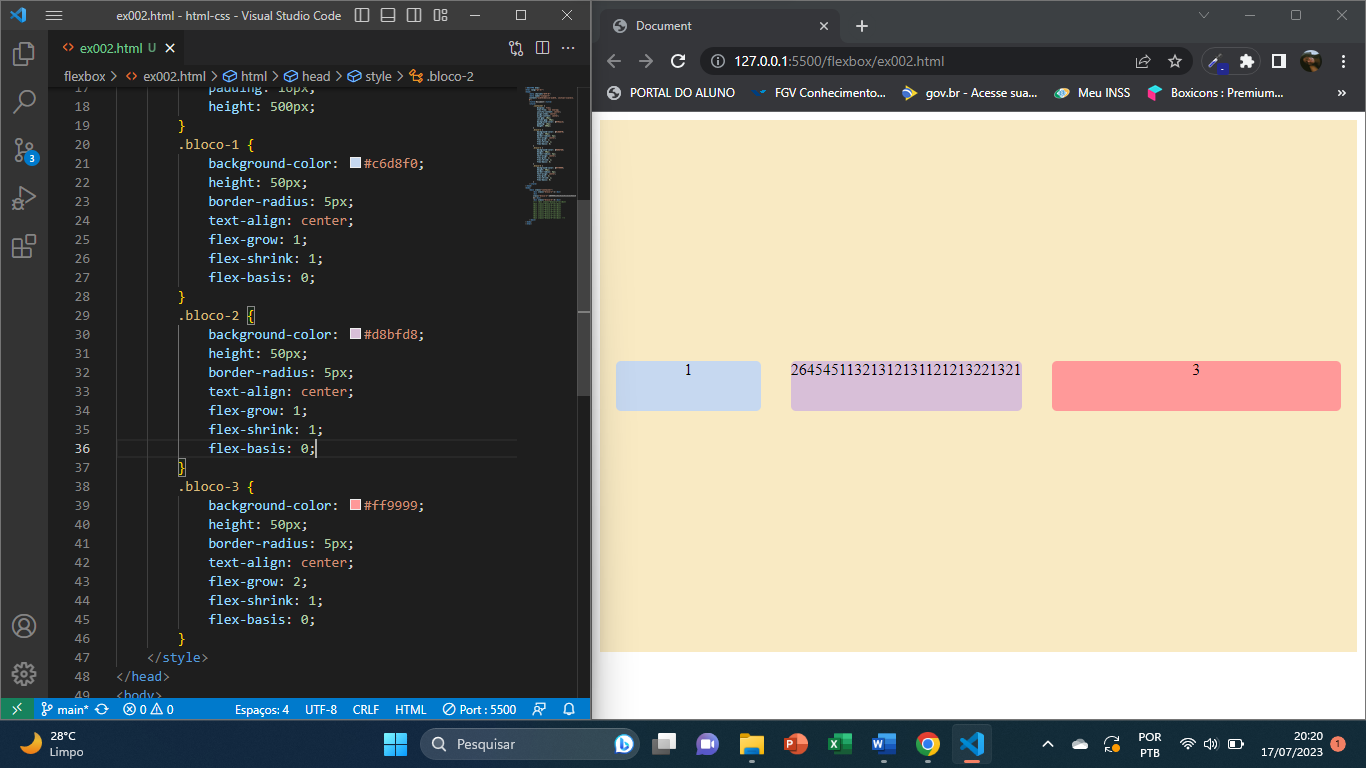
Não recebe valores negativos.

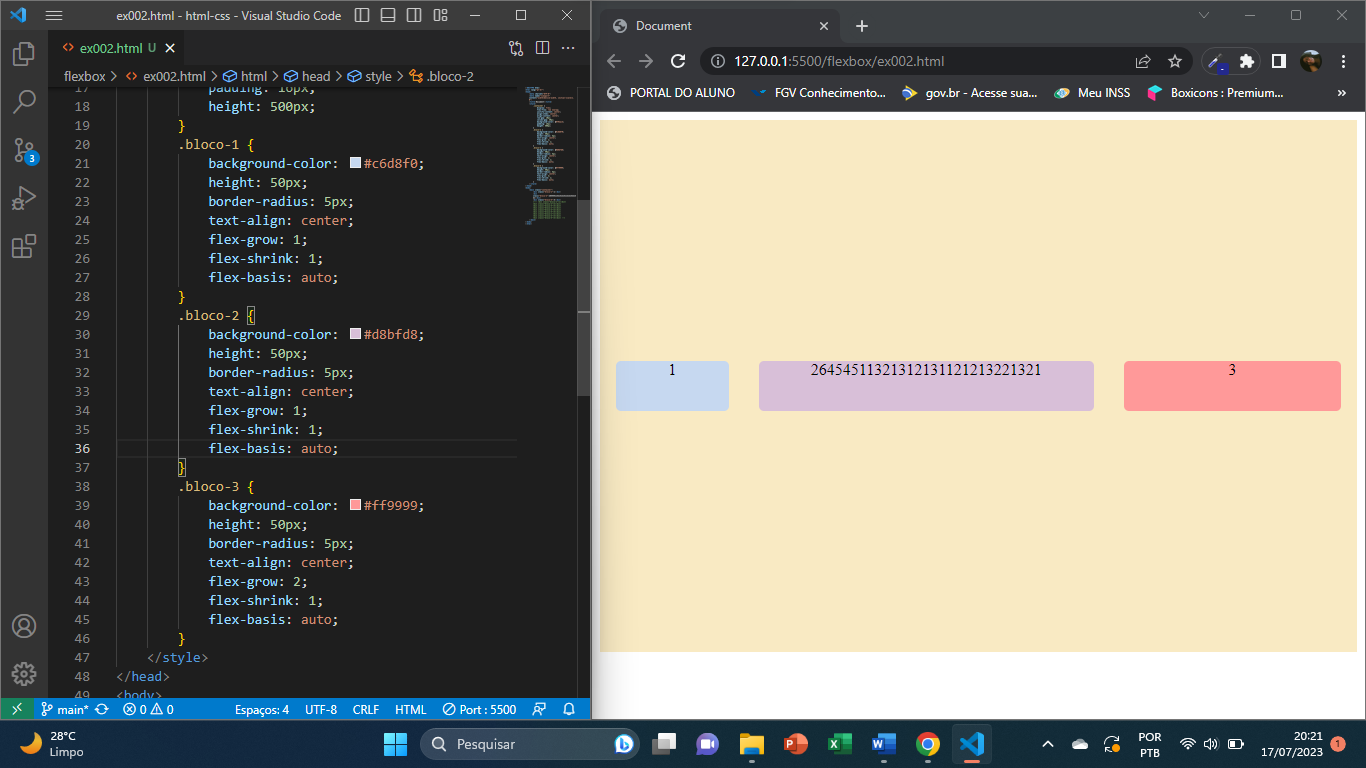
**Flex-shrink:** me permite encolher os itens para caber no container pai. É obrigatório não ter wrap, deve ser sempre nowrap para que os elementos não quebrem e sim diminuam. Outra coisa, deve-se definir uma largura senão nunca irão encolher.

Por padrão eles possuem o valor 1. Os itens não crescem por padrão, é necessário usar o flex-grow para isso. Porém, diminuir é padrão, por isso o valor do flex-shrink é 1, mas pode ser 0.



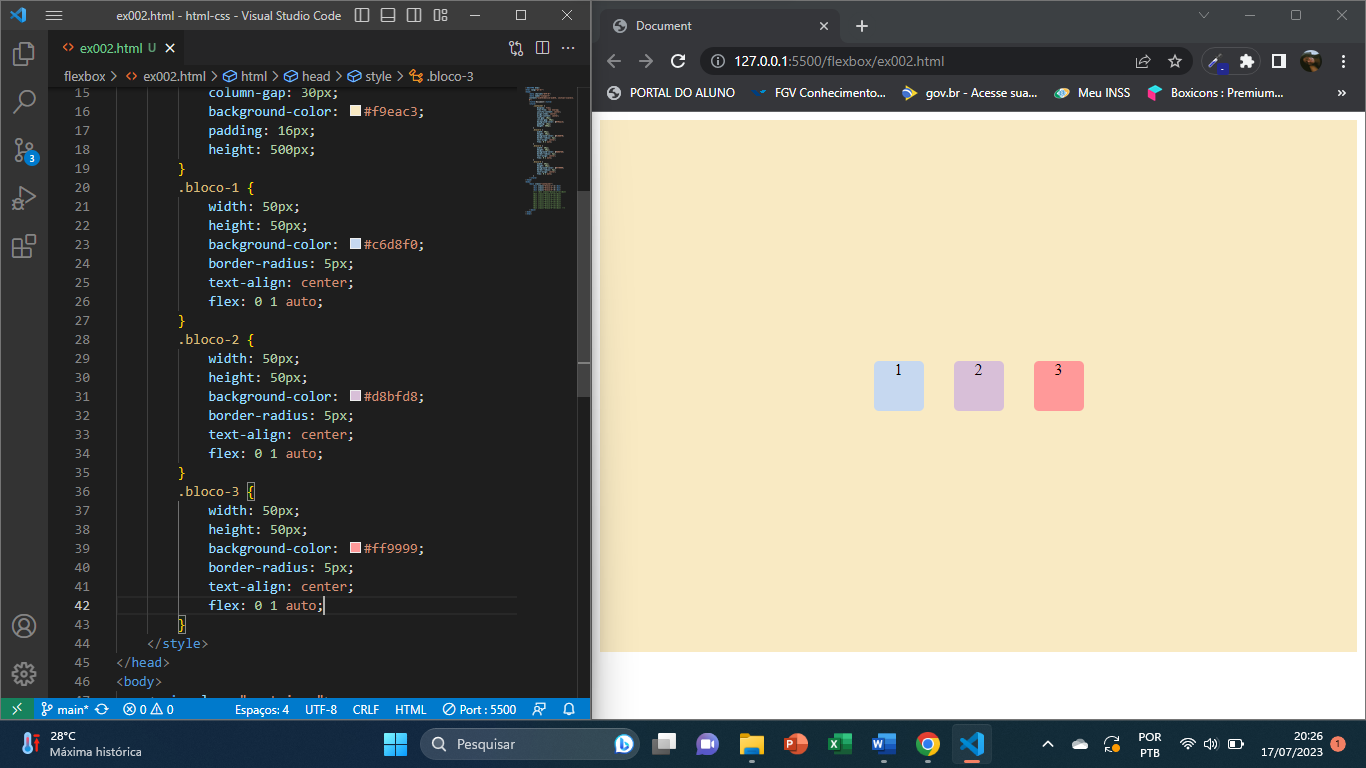
**Flex-basis:** define o tamanho dos itens (altura e largura). Por padrão é um valor auto (palava-chave).





Colocando valor auto, o mesmo tamanho acrescentado no bloco 1 foi também para o bloco 2, e o dobro para o 3.

**Flex:** junção das 3 propriedades anteriores (flex-grow, flex-shrink e flex-basis).



**Align-self:** me permite alterar um elemento específico (desobedece o pai).

