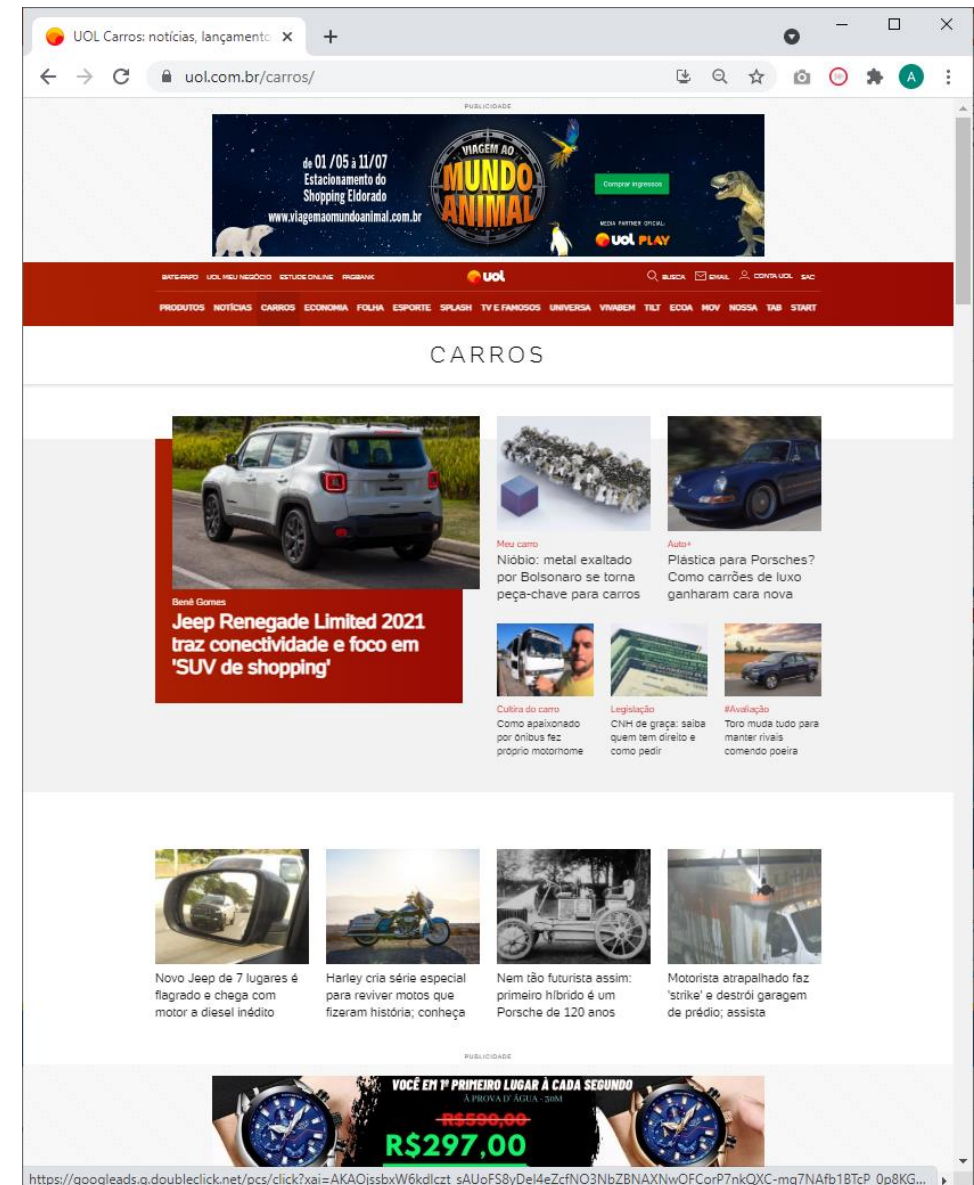


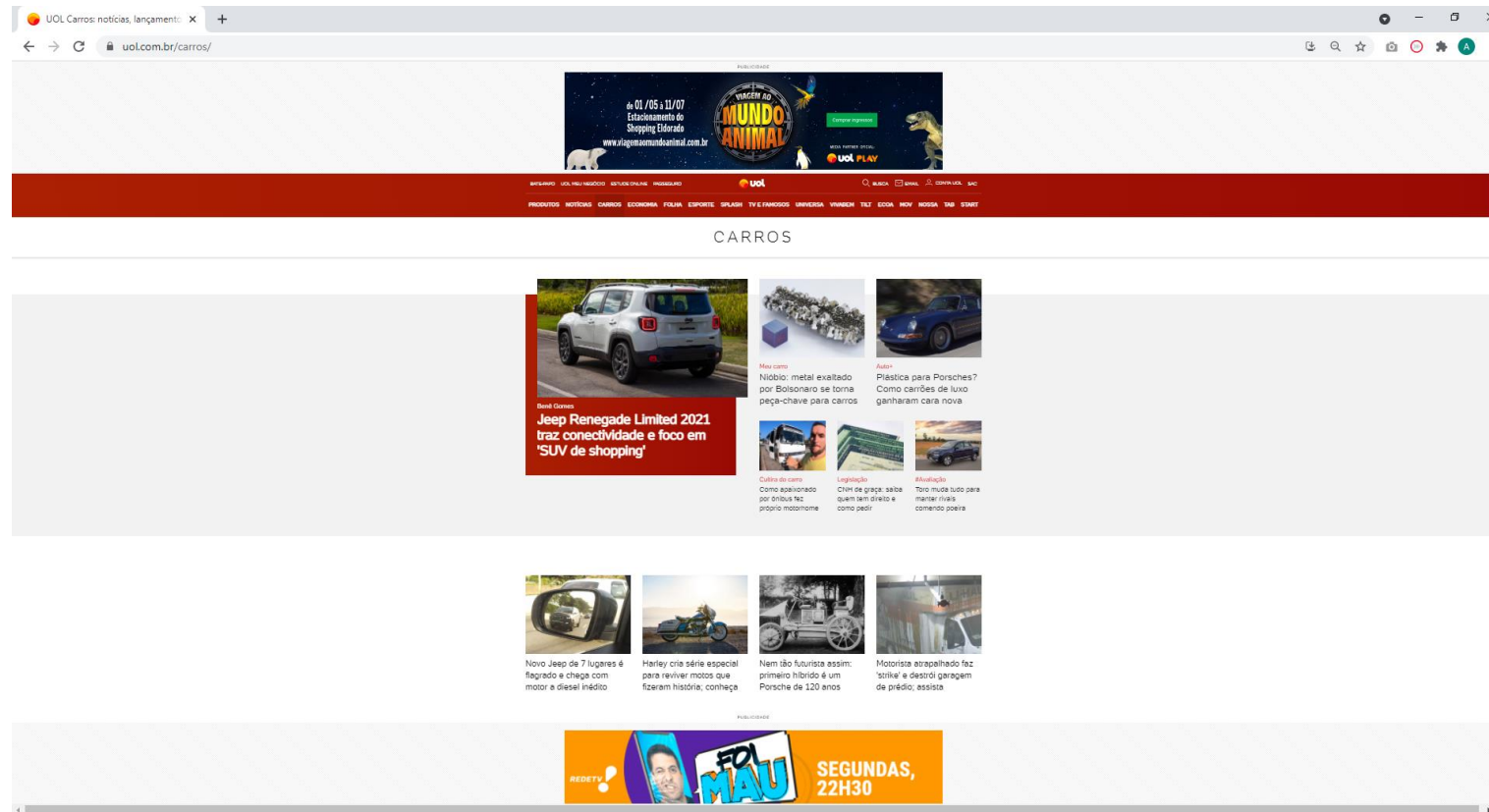
# Aplicações para Internet

- ✓ Layout líquido e congelado
- ✓ Layout com float
- ✓ Frameworks para layout em grid (introdução)
- ✓ Flex CSS e Grid CSS
- ✓ Exemplos

Trata-se de uma organização de conteúdo que permanece a mesma, mesmo que o tamanho da janela mude.

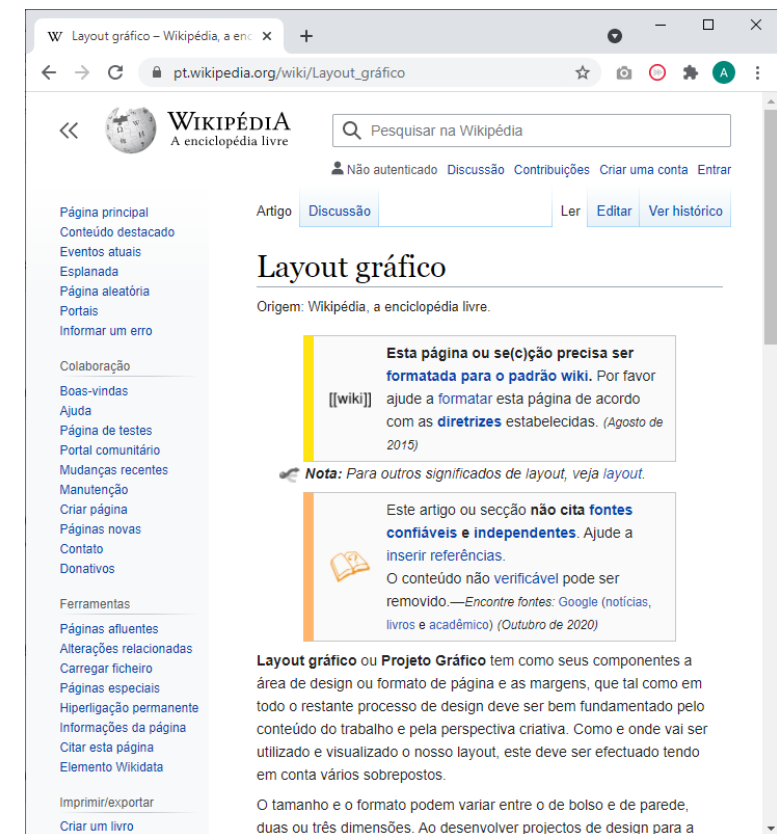
<http://www.uol.com.br/>



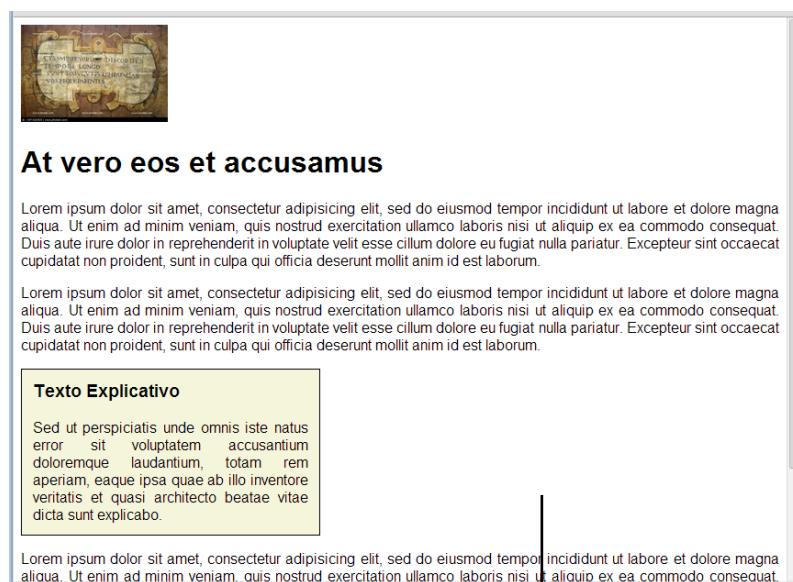


Mesmo a janela aumentando/diminuindo de tamanho o conteúdo permanece ocupando o mesmo espaço. Os conteúdos são cortados se a janela for pequena.

- ✔ Trata-se de uma organização de conteúdo que se adapta ou se molda ao tamanho da janela, **mas o layout normalmente não muda**, utiliza medidas em %. **Obs: Não é responsivo.**



- ✓ A propriedade **float** foi criada para fazer elementos flutuarem à direita ou à esquerda de um conteúdo



**At vero eos et accusamus**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

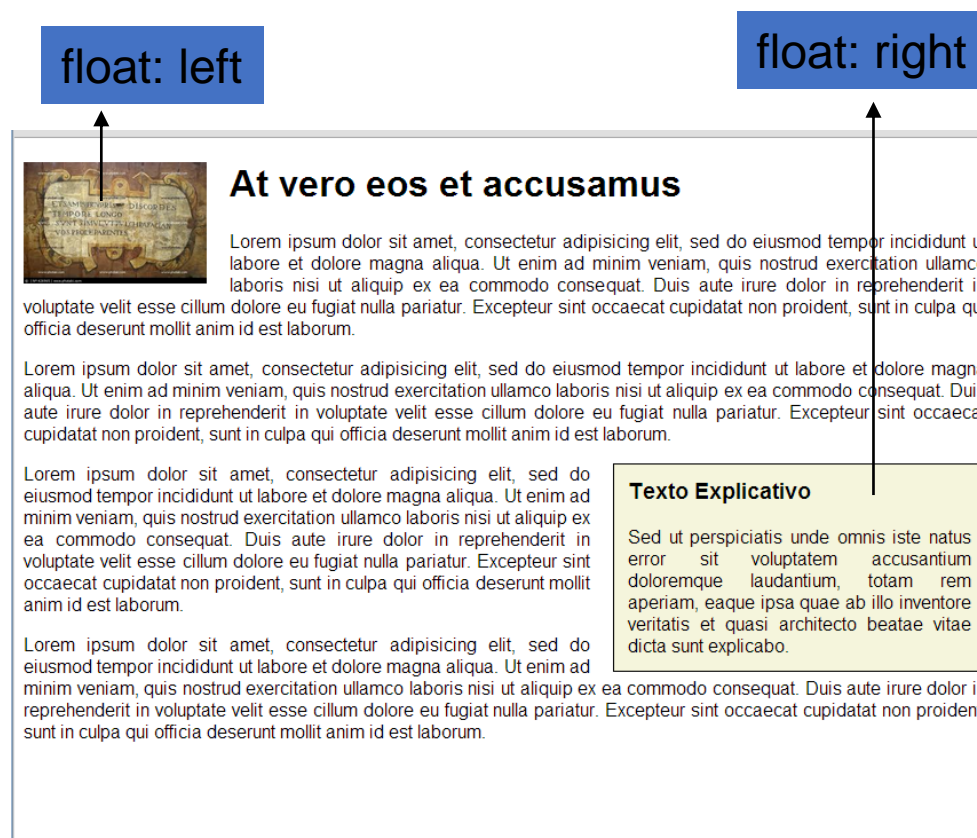
**Texto Explicativo**

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

sem usar float

Exemplo1.html, Exemplo1semFloat.html



**float: left**

**float: right**

**At vero eos et accusamus**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

**Texto Explicativo**

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

- ✓ Todos os blocos (divs ou blocos semânticos) têm seu tamanho definidos com porcentagem.
- ✓ Margens e "padding" devem ser definidos com porcentagens.
- ✓ Todos os blocos flutuam para a esquerda (float:left).
- ✓ O bloco que inicia uma nova linha deve ter a propriedade `clear: left;` ou `clear:both;`
- ✓ **Problema:**
  - pode não ser desejável que as margens, os 'padding' sejam de tamanho variável (proporcionais);
  - as bordas podem sumir.

- ✓ Todos os blocos (divs ou blocos semânticos) tem seu tamanho definidos em pixels, menos o bloco central.
- ✓ Margens, "padding" e bordas devem ser definidos em pixels.
- ✓ Há blocos que flutuam para a esquerda e outros para direita.
- ✓ O bloco que inicia uma nova linha deve ter a propriedade **clear: both;**
- ✓ **Problemas:**
  - Os blocos à direita tem que vir antes do bloco central no HTML. Em geral: confusão na ordem.
  - Definir a margem do bloco central pode ser confuso.

- ✓ Todos os blocos de layout devem estar dentro de um bloco que os contém, comumente chamado “wrapper” ou “container”
- ✓ Este bloco deve ter uma largura (**width**) determinada e ser centralizado. Para isso se define uma margem automática, o que faz o navegador colocar a mesma margem do lado direito e esquerdo: **margin: 0 auto;**



- ✓ Um framework é um conjunto de código para desenvolvimento de aplicações. Ele fornece uma base sobre a qual os desenvolvedores podem construir seus códigos.
- ✓ Simplificam o processo de desenvolvimento, já que os programadores não precisam reinventar a roda cada vez que forem desenvolver uma nova aplicação.
- ✓ Existem para diferentes tipos de linguagens e aplicações
- ✓ Veja alguns frameworks no link abaixo:
  - <http://usablica.github.io/front-end-frameworks/compare.html>

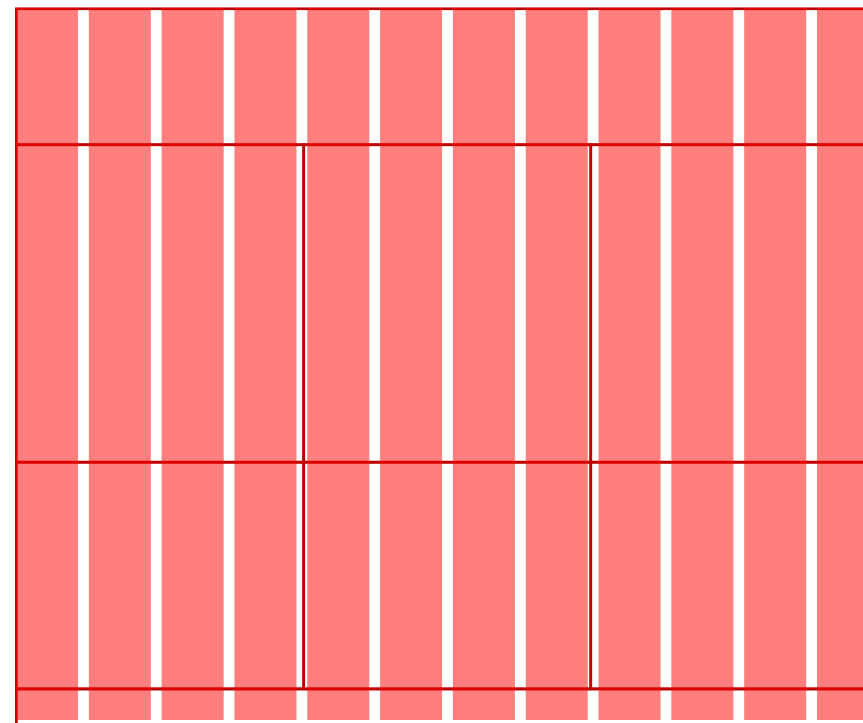


**Framework** é um conjunto de códigos (ou biblioteca) que estendem uma linguagem, ou seja, que 'criam' novos termos com novas funcionalidades.

Os **frameworks de grid** normalmente são disponibilizados em um arquivo CSS. Este arquivo deve ser anexado à página onde o Framework deve ser usado.

Na sua grande maioria, eles oferecem classes preparadas para criar o *wrapper* (que pode ser de tamanho fixo ou não) e blocos cuja largura ocupa uma certa quantidade de 'colunas' do grid.

Framework 960.css



O usual é que hajam 12 ou 16 colunas

- ✓ <http://960.gs/>
- ✓ <http://foundation.zurb.com/>
- ✓ <http://www.responsivegridsystem.com/>
- ✓ <http://unsemantic.com/>

- ★ ✓ <https://getbootstrap.com/>
- ★ ✓ <https://semantic-ui.com/>

- ✓ Baixe o framework (use o link do slide anterior).
- ✓ Crie a página HTML com **divs** para o layout dentro de uma **div wrapper** e dispostos na ordem correta, mas sem nenhuma informação de layout no CSS.
- ✓ Coloque o arquivo 960.css na pasta e faça o link para ele no arquivo HTML:

```
<link rel="stylesheet" href="960.css" />
```

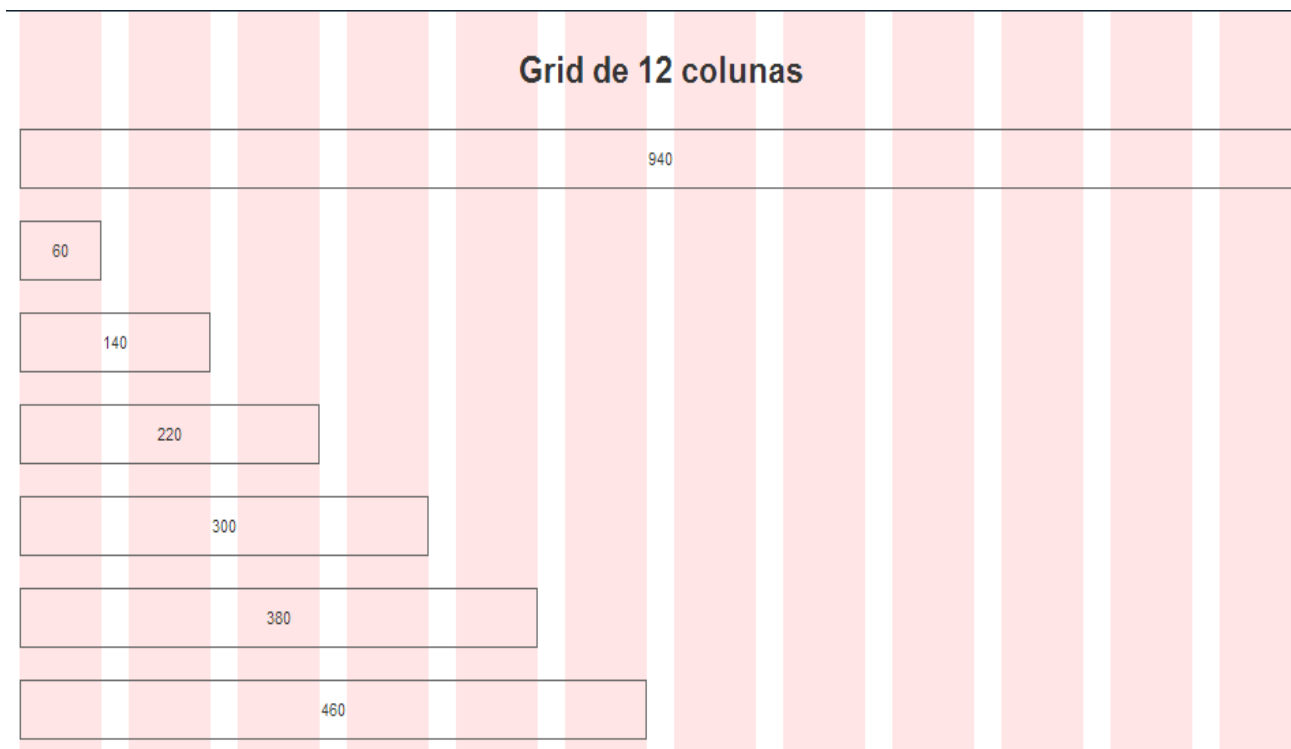
- ✓ No wrapper, adicione a classe **container\_12**  

```
<div id="wrapper" class="container_12">
```

Como o wrapper é **container\_12**, em cada **div** adicione uma classe de acordo com o espaço que o bloco irá ocupar no layout:

Classes:

- **grid\_12**
- **grid\_1**
- **grid\_2**
- **grid\_3**
- **grid\_4**
- **grid\_5**
- **grid\_6**



- ✓ Se, nas linhas do grid, o conteúdo fizer com que as alturas dos blocos fiquem diferentes, é necessário criar uma separação (para que uma linha não invada outra), colocando no HTML uma **div** sem conteúdo entre as linhas do grid. Essa **div** deve ser da classe "clear":

```
<div class="clear"></div>
```

**Importante:** Assegure-se de que cada linha ocupa exatamente doze colunas.

layout\_960.html

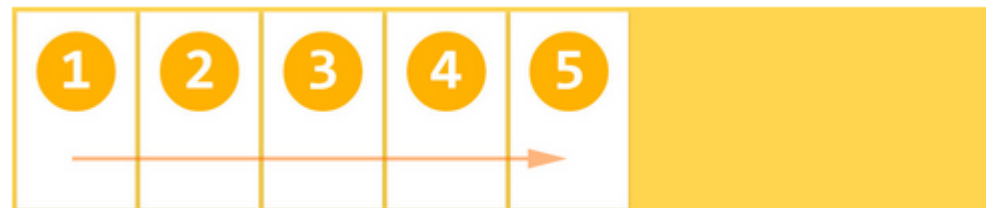
- ✓ Flex é uma especificação que estende o CSS, criada originalmente pela Mozilla e incorporada na versão 3.0 da linguagem.
- ✓ Em termos da construção de um grid, o Flex resolve a organização de uma linha, desde que seus blocos estejam envoltos em uma tag de caixa para este grupo.
- ✓ Para utilizar o Flex, devemos configurar a propriedade display do CSS:

**display: flex;**

- ✓ Essa configuração se faz no elemento raiz, ou seja, no container. Automaticamente, todos os filhos desse elemento se tornam flex.

- ✔ Configura a disposição dos itens, essa disposição pode ser em **linha** ou **coluna**.

```
.container {  
  flex-direction: row;  
}
```



```
.container {  
  flex-direction: column;  
}
```



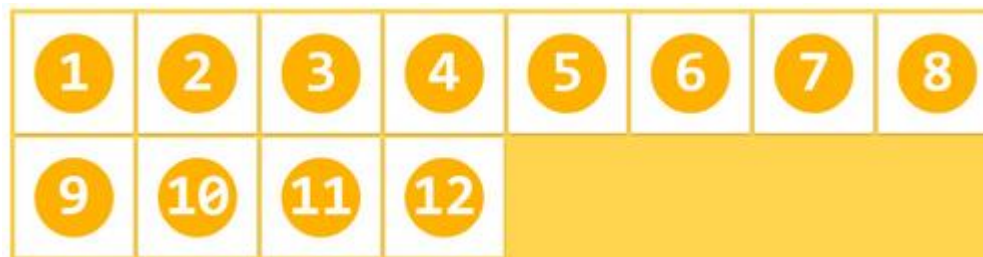
veja pasta **exemplo1Flex**

- ✔ Controla se os itens do container serão de uma linha simples ou terá quebra de linha.

```
.container {  
    flex-wrap: nowrap;  
}
```



```
.container {  
    flex-wrap: wrap;  
}
```



veja pasta **exemplo2Flex**



- ✔ Está propriedade **flex-flow** é um atalho para configurar simultaneamente as propriedades **flex-direction** e **flex-wrap**

```
.container {  
    flex-flow: <flex-direction> <flex-wrap>;  
}
```

Ex:

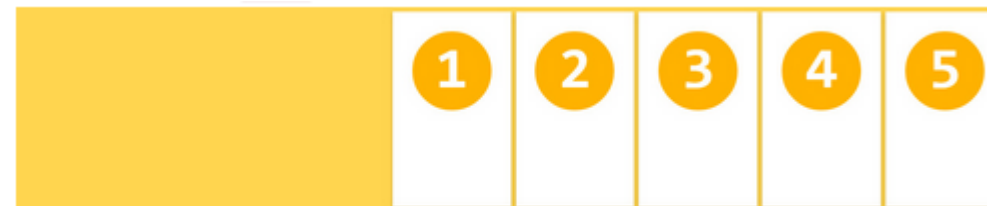
```
.container {  
    flex-flow: row wrap;  
}
```

- ✓ Alinha os itens no eixo principal, na **horizontal** (x)

```
.container {  
  justify-content: flex-start;  
}
```



```
.container {  
  justify-content: flex-end;  
}
```



```
.container {  
  justify-content: center;  
}
```



veja pasta **exemplo3Flex**

```
.container {  
    justify-content: space-between;  
}
```



```
.container {  
    justify-content: space-around;  
}
```

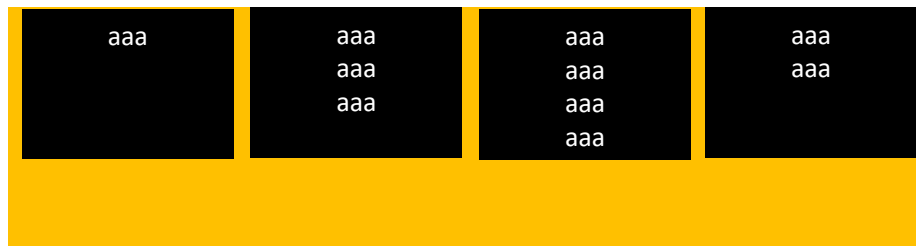


- ✓ Alinha os itens na **vertical**, semelhante à propriedade anterior, os itens são padronizados conforme a altura do maior item.

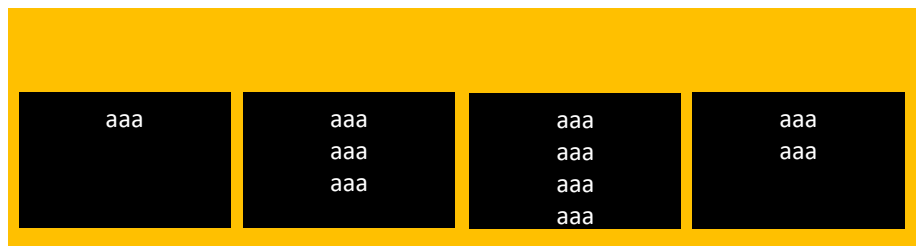
```
.container {  
  align-content: stretch;  
}
```



```
.container {  
  align-content: flex-start;  
}
```



```
.container {  
  align-content: flex-end;  
}
```



```
.container {  
  align-content: center;  
}
```



- ✓ Alinha os itens na **vertical**, semelhante à propriedade align-content, **cada item mantém seu tamanho de altura conforme conteúdo**

```
.container {  
  align-items: stretch;  
}
```



```
.container {  
  align-items: flex-start;  
}
```

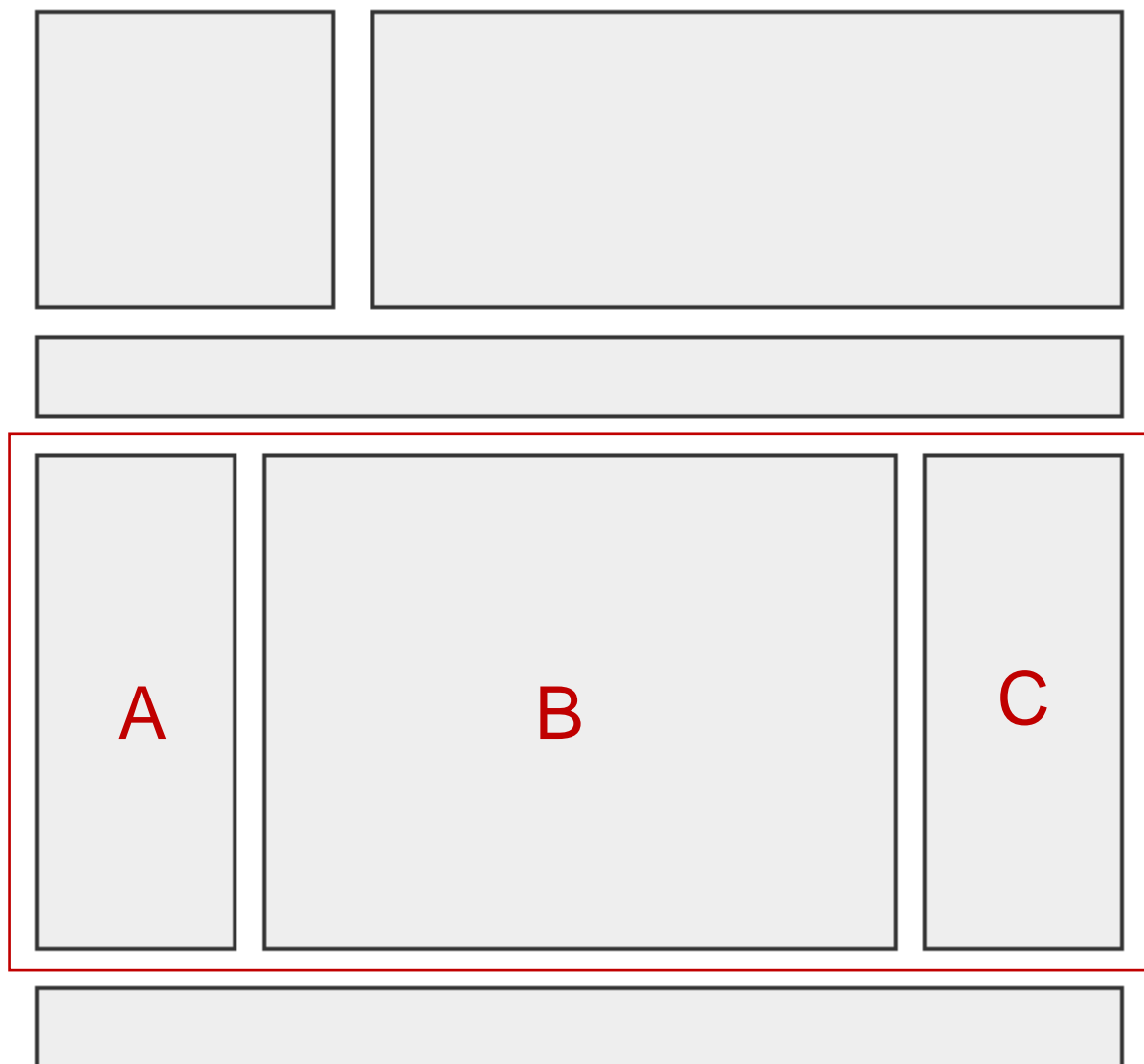


```
.container {  
  align-items: flex-end;  
}
```



```
.container {  
  align-items: center;  
}
```

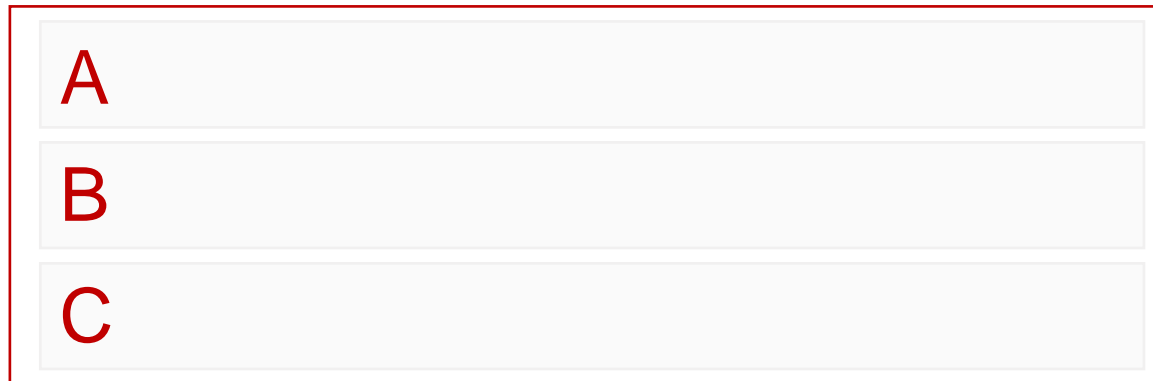




```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

OBS: utilizamos a tag DIV para exemplo, contudo, podemos usar qualquer tag de bloco.

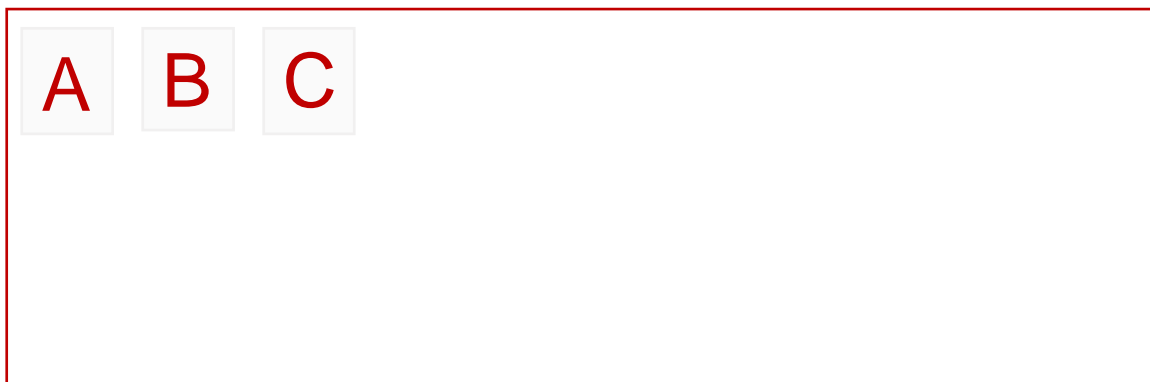
Normalmente...



```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

## CSS com Flex:

```
#linha {  
  display: flex;  
}
```



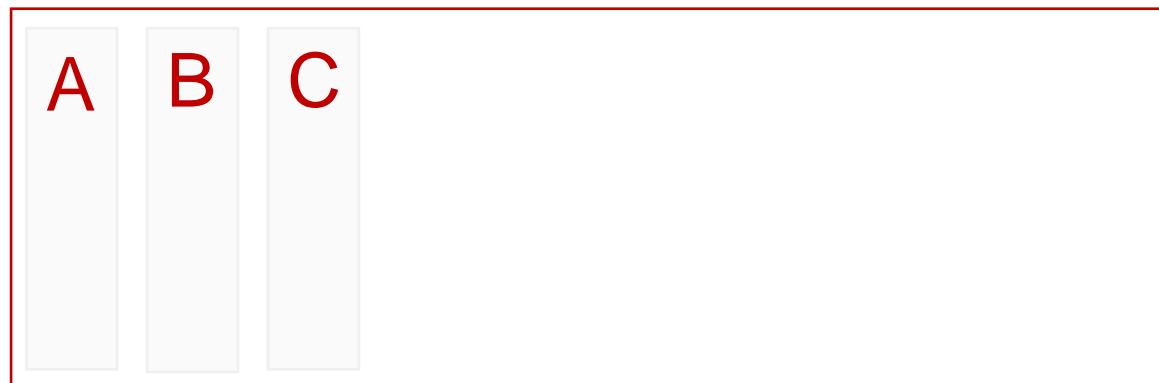
```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```



## CSS com FlexBox:

```
#linha {  
  display: flex;  
  align-items: stretch;  
  height: 200px;  
}
```

```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

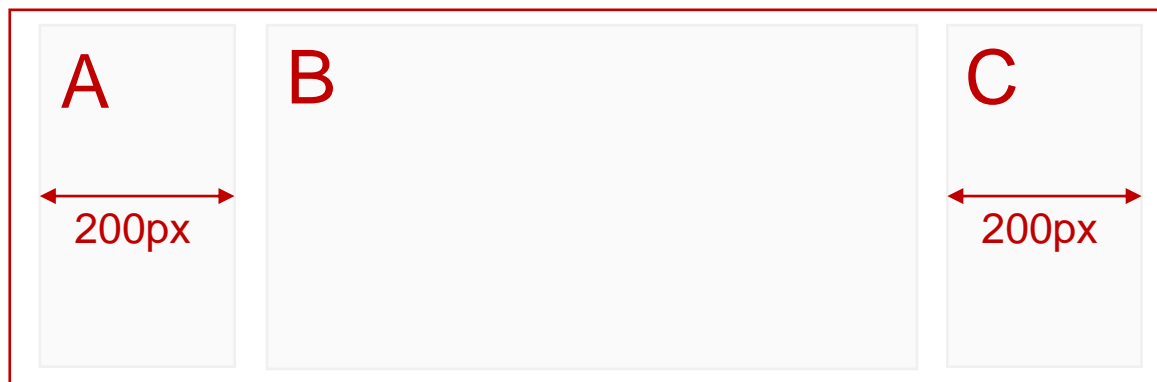


## CSS com FlexBox:

```
#linha {  
  display: flex;  
  align-items: stretch;  
  height: 200px;  
}
```

```
#A, #C {  
  min-width: 200px;  
}  
  
#B {  
  flex-grow: 1;  
}
```

```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```



```
#linha {  
  display: flex;  
  align-items: stretch;  
  height: 200px;  
}
```

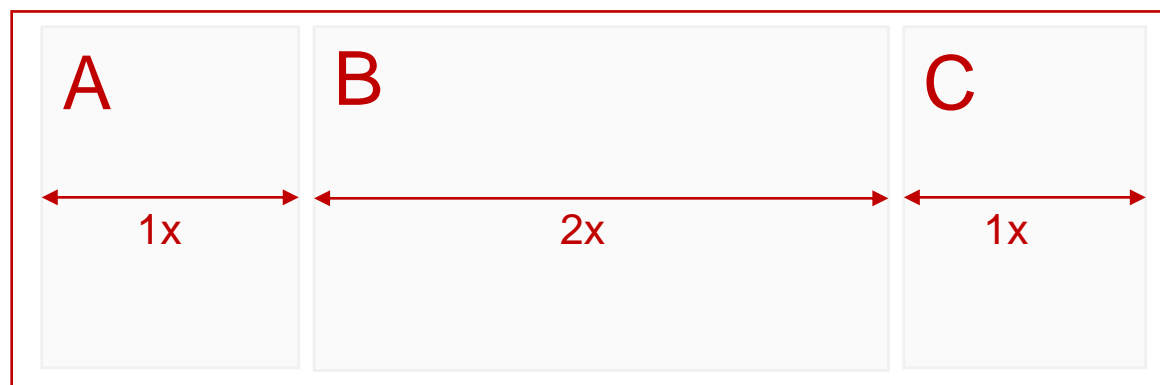
```
#A, #C { flex-grow : 1; }  
  
#B      { flex-grow: 2; }  
  
#A, #B, #C { flex-basis : 0; }
```

**flex-grow**: especifica quanto o item crescerá com relação aos outros itens no container (default: 0)

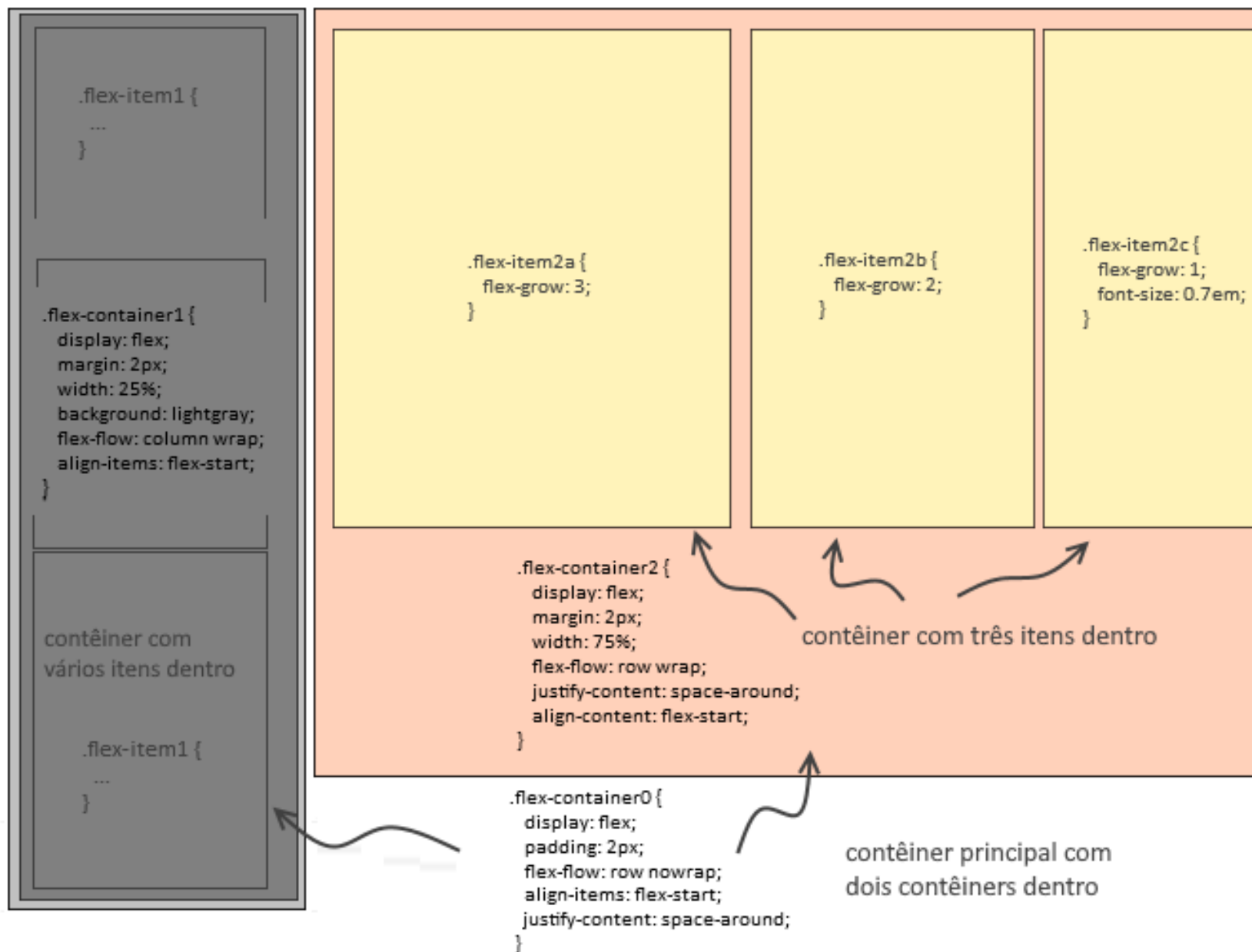
**flex-basis**: tamanho inicial do item

flexBoxLayout.html

```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```



- ✔ Existem muito mais comandos que os apresentados aqui, que permitem trabalhar com coluna em vez de linha, alterar a ordem e fazer todo tipo de alinhamento dos blocos internos
- ✔ Tudo que foi escrito nos exemplos com “-webkit-” na frente, deve ser escrito também com “-moz-”, com “-ms” para que rode em todos os navegadores que tem implementação
- ✔ Para saber sobre o suporte dos navegadores ao FlexBox, veja o site <http://caniuse.com/flexbox>
- ✔ Mais ajuda:
  - <http://tableless.com.br/flexbox-organizando-seu-layout/>
  - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>





## Exemplo Flexbox - um contêiner principal e dois internos

**Lorem Ipsum.** Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

**Lorem Ipsum.** Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

**Lorem Ipsum.** Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

**Lorem Ipsum** é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

**Lorem Ipsum** é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos.



Um gato

Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Elemento com flex-grow: 2;

**Lorem Ipsum** é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Elemento com flex-grow: 1;

```
<div class="flex-container0">

  <article class="flex-container1">
    <section class="flex-item1">...</section>
    <section class="flex-item1">...</section>
    <section class="flex-item1">...</section>
  </article>

  <article class="flex-container2">
    <section class="flex-item2a">...</section>
    <section class="flex-item2b">...</section>
    <section class="flex-item2c">...</section>
  </article>

</div>
```

```
.flex-container0 {
  display: flex;
  padding: 2px;
  margin: 0;
  background: #EEE;
  flex-flow: row nowrap;
  justify-content: space-around;
  align-items: flex-start;
}

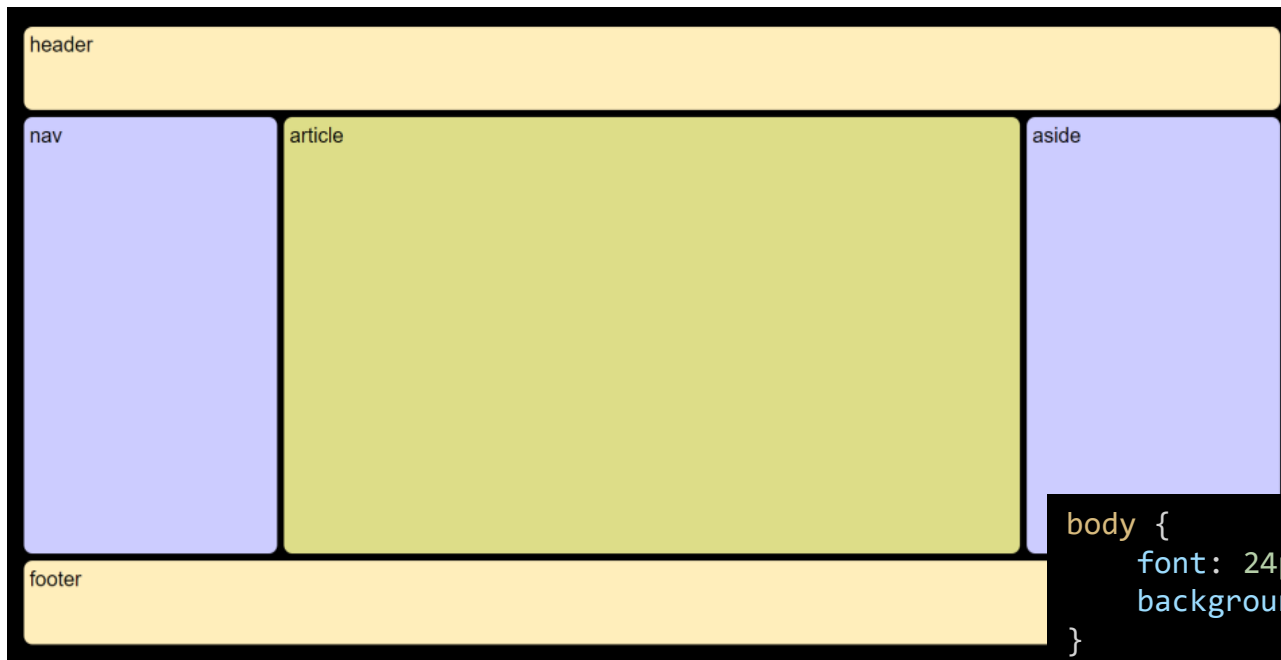
.flex-container1 {
  display: flex;
  margin: 2px;
  width: 25%;
  background: lightgray;
  flex-flow: column wrap;
  align-items: flex-start;
  justify-content: space-around;
}

.flex-container2 {
  display: flex;
  padding: 0;
  margin: 2px;
  width: 75%;
  border: solid 2px lightgray;
  background: #f4ead9;
  flex-flow: row wrap;
  justify-content: space-around;
  align-content: flex-start;
}
```

```
.flex-item1 {
  color: white;
  background: #7c7777;
  font-size: 0.7em;
  font-family: 'Calibri';
  text-align: justify;
  margin: 4px;
  padding: 5px;
}

.flex-item2a,
.flex-item2b,
.flex-item2c {
  color: black;
  background: #ffebc5;
  font-family: 'Calibri';
  text-align: justify;
  margin: 4px;
  padding: 5px;
  flex-basis: 0;
  border: solid 2px #d9d9d9;
}

.flex-item2a { flex-grow: 3; }
.flex-item2b { flex-grow: 2; }
.flex-item2c { flex-grow: 1; }
h2 {
  display: flex;
  align-items: center;
}
```



```
<header>header</header>
<section id='main'>
  <nav>nav</nav>
  <article>article</article>
  <aside>aside</aside>
</section>
<footer>footer</footer>
```

```
body {
  font: 24px Helvetica;
  background: #000;
}

*{
  margin: 4px;
  padding: 7px;
  border-radius: 10px;
}

#main {
  height: 500px;
  margin: 0px;
  padding: 0px;
  display: flex;
  flex-flow: row;
}
```

```
#main>article {
  background: #dddd88;
  flex: 3 1 60%;
  order: 2;
}

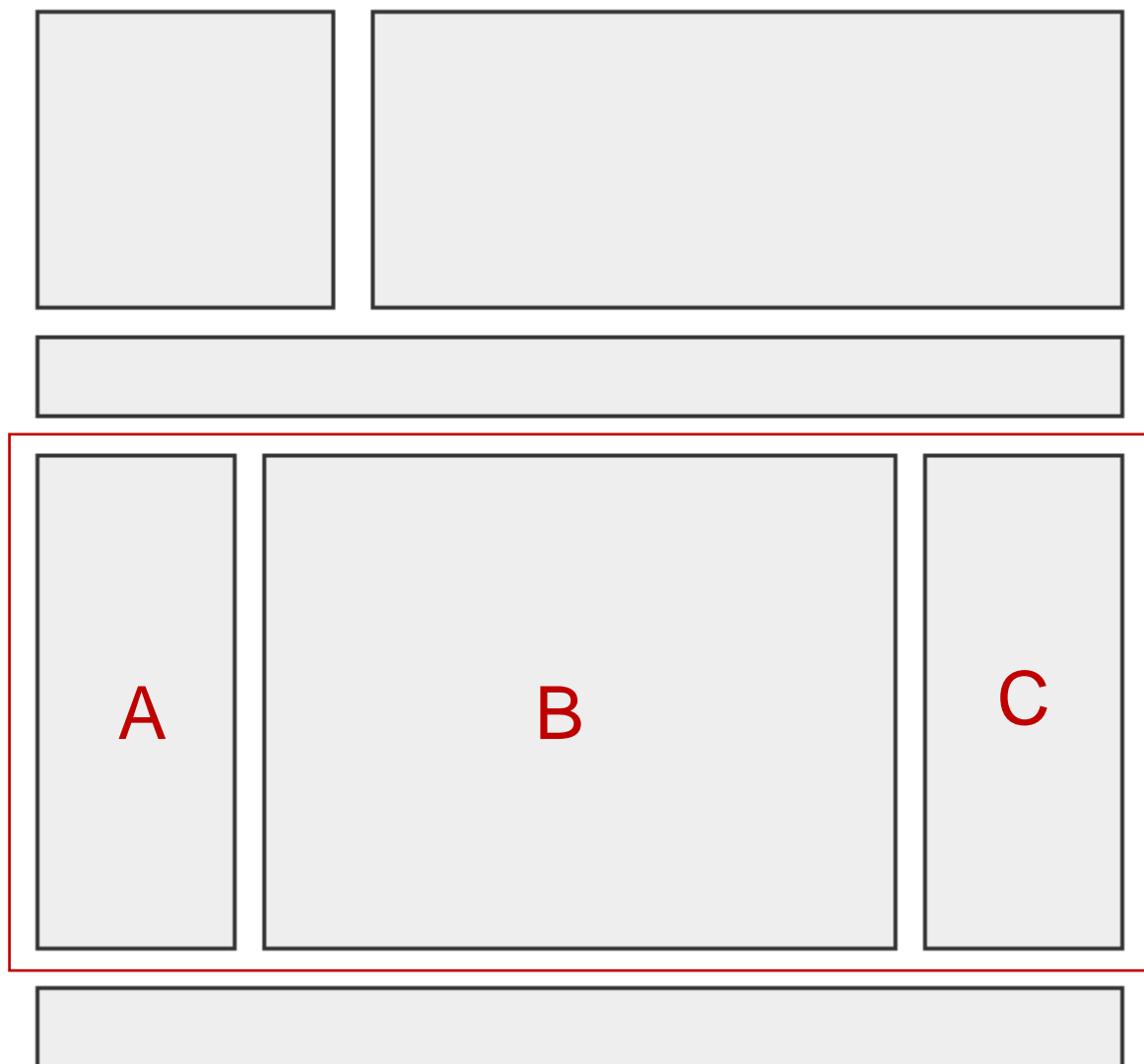
#main>nav {
  background: #ccccff;
  flex: 1 1 20%;
  order: 1;
}

#main>aside {
  background: #ccccff;
  flex: 1 1 20%;
  order: 3;
}

header, footer {
  height: 80px;
  background: #ffeebb;
}
```




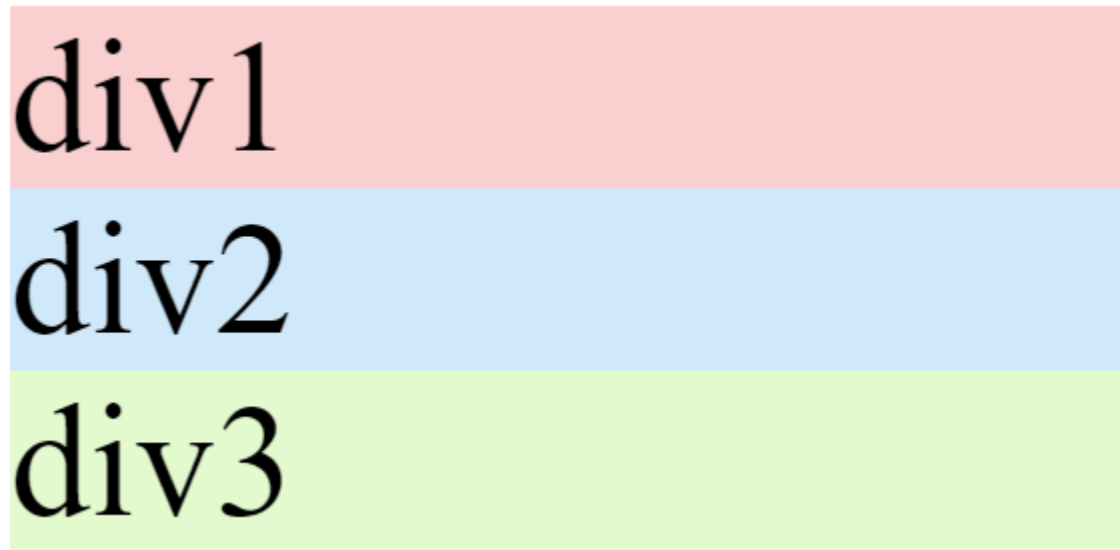
- ✓ Além do Flex, este é outro recurso novo no CSS.
- ✓ Introduz o sistema de grid (grades)
- ✓ Os grids são basicamente um conjunto de linhas/colunas que facilitam a criação dos layouts.
- ✓ Com este recurso, não precisamos utilizar os grids de frameworks, usamos CSS puro.
- ✓ Grids podem ter dimensões fixas (em pixel) ou flexíveis (em % ou usando a unidade fr). FR representa uma fração do espaço restante no contêiner de grade.
- ✓ Assim como no Flex, devemos declarar o container pai como grid e todos seus filhos diretos, automaticamente serão grid item.
- ✓ Vamos fazer os mesmos exemplos do Flex, mas agora com GRID.



```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

OBS: utilizamos a tag DIV para exemplo, contudo, podemos usar qualquer tag de bloco.

Normalmente...



```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

## CSS com Grid:

```
#linha{  
    display: grid;  
}
```

div1

div2

div3

```
<div id="linha">  
  <div id="A">A</div>  
  <div id="B">B</div>  
  <div id="C">C</div>  
</div>
```

OBS: não veremos diferença, contudo, todas as divs são grid itens.

- ✔ Para estruturar nosso grid, podemos usar as propriedades **grid-template-columns** e **grid-template-rows**.
- ✔ Vamos criar um grid com 3 colunas de tamanho 300px

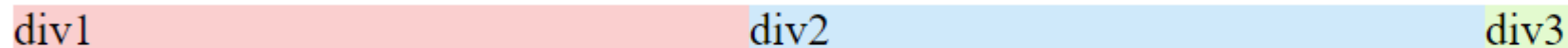
div1

div2

div3

```
#linha{  
    display: grid;  
    grid-template-columns: 300px 300px 300px;  
}
```

- ✔ Usando a unidade FR, tornamos nossas colunas flexíveis conforme o espaço disponível. (fr trabalha com a ideia de proporção de tela, dividindo a tela conforme a soma das quantidades de fr e atribuindo a quantidade para cada coluna ou linha)



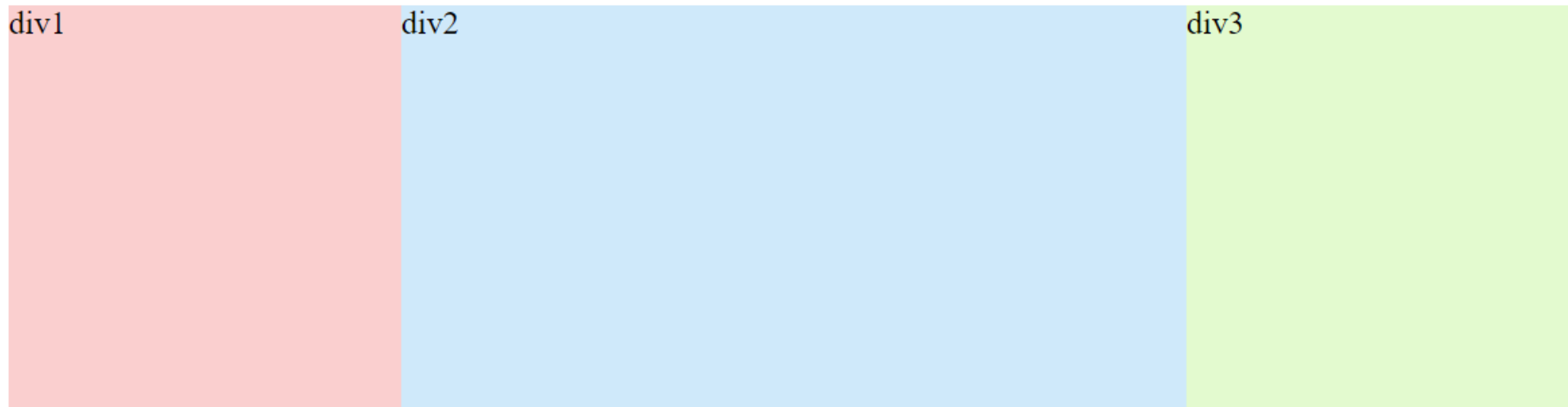
div1 div2 div3

```
#linha{  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
}
```



div1 div2 div3

```
#linha{  
  display: grid;  
  grid-template-columns: 3fr 1fr 1fr;  
}
```



```
#linha{  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr;  
  height: 200px;  
}
```



```
<header>header</header>
<section id='main'>
  <nav>nav</nav>
  <article>article</article>
  <aside>aside</aside>
</section>
<footer>footer</footer>
```

```
#main {
  height: 500px;
  margin: 0px;
  padding: 0px;
  display: grid;
  grid-template-columns: 1fr 3fr 1fr;
}
```

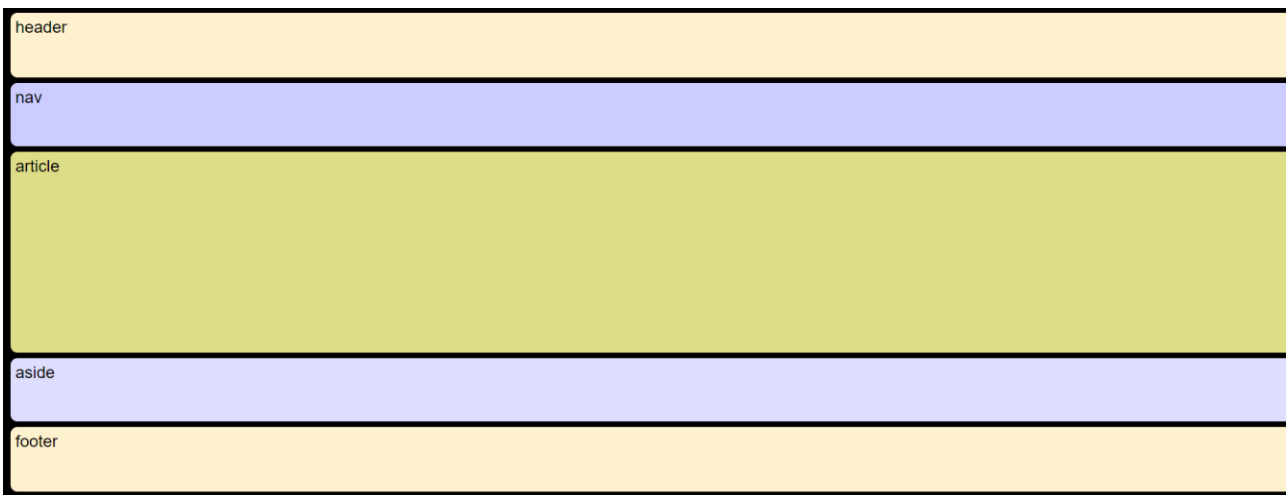
```
body {
  font: 24px Helvetica;
  background: #000;
}

*{
  margin: 4px;
  padding: 7px;
  border-radius: 10px;
}
```

```
#main>article { background: #dddd88; }
#main>nav { background: #ccccff;}
#main>aside { background: #dddffd; }

header,footer {
  height: 80px;
  background: #fdf1ce;
}
```





```
<header>header</header>
<section id='main'>
  <nav>nav</nav>
  <article>article</article>
  <aside>aside</aside>
</section>
<footer>footer</footer>
```

```
main {
  height: 500px;
  margin: 0px;
  padding: 0px;
  display: grid;
  grid-template-rows: 1fr 3fr 1fr;
}
```

```
body {
  font: 24px Helvetica;
  background: #000;
}

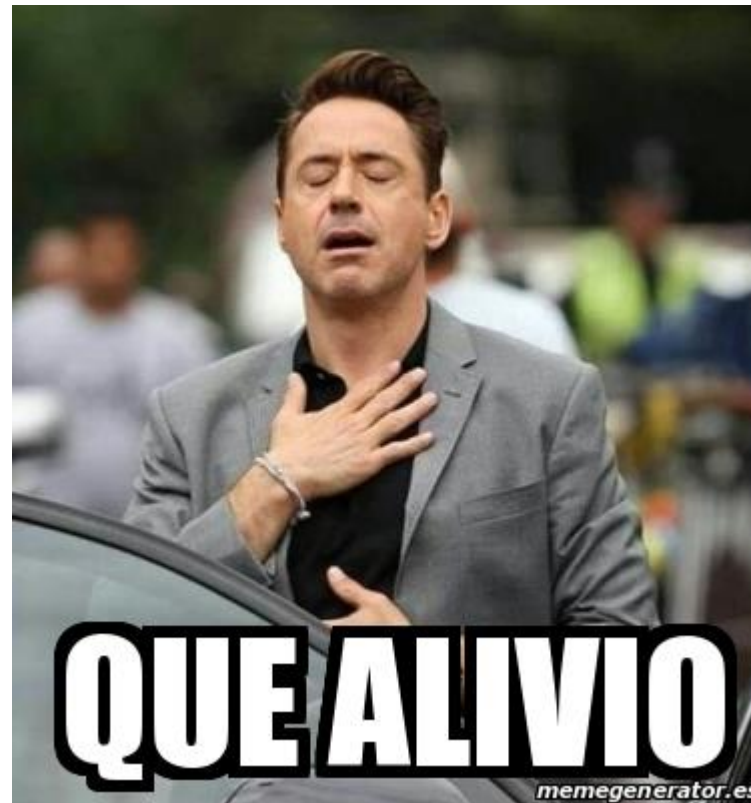
*{
  margin: 4px;
  padding: 7px;
  border-radius: 10px;
}
```

```
#main>article { background: #dddd88; }
#main>nav { background: #ccccff;}
#main>aside { background: #dddfd; }

header,footer {
  height: 80px;
  background: #fdf1ce;
}
```

- ✓ [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout)
- ★ [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)
- ★ <https://css-tricks.com/snippets/css/complete-guide-grid/>
- ✓ <https://css-tricks.com/things-ive-learned-css-grid-layout/>
- ✓ <https://gridbyexample.com/examples/>
- ✓ <https://tableless.com.br/um-pouco-sobre-css-grid-layout/>

Nesta aula não haverá entrega de exercícios!



- ✓ <https://tobiasahlin.com/blog/common-flexbox-patterns/>
- ✓ <https://css-tricks.com/designing-a-product-page-layout-with-flexbox/>
- ✓ <https://demos.scotch.io/visual-guide-to-css3-flexbox-flexbox-playground/demos/>
- ✓ <https://the-echoplex.net/flexyboxes/>
- ✓ <https://loading.io/flexbox/>
- ✓ <https://flexbox.webflow.com/#holygrail>
- ✓ [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Basic\\_Concepts\\_of\\_Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)
- ✓ [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Typical\\_Use\\_Cases\\_of\\_Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Typical_Use_Cases_of_Flexbox)
- ✓ <https://origamid.com/projetos/flexbox-guia-completo/>