

COM222

DESENVOLVIMENTO DE SISTEMAS WEB

Aula 03: CSS Flexbox

□ Flexbox

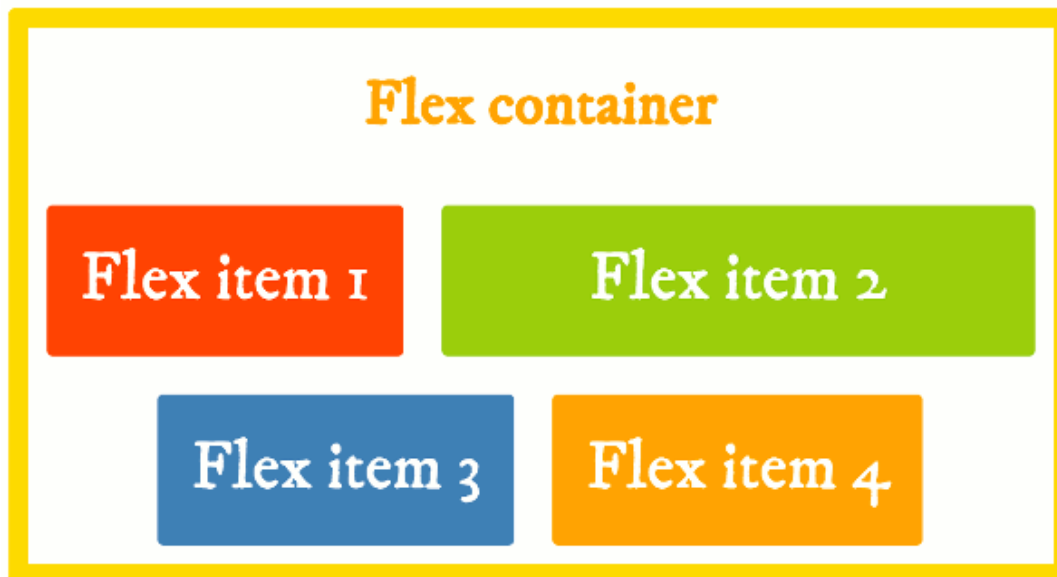
- Introdução
- Propriedades
- Criação de layout unidimensional
- Criação de layout bidimensional
- Media objects
- Galeria de fotos flexível

Flexbox

Introdução

3

- Permite que o desenvolvedor controle o posicionamento de elementos em containers
 - ▣ Possibilita especificar o alinhamento vertical e horizontal de elementos, a ordem em que os elementos aparecem e especificar de que forma cada elemento deve ocupar o espaço do container



Flexbox

Introdução

4

- Antes do Flexbox, fazer o layout de um website era extremamente trabalhoso em função das limitações dos modelos de layout existentes
 - ▣ Para obter um bom resultado era necessário utilizar uma combinação de layouts, tornando o design das aplicações web uma tarefa complexa
- CSS Flexible Box (nome correto)
 - ▣ Desenvolvido pelo CSS Working Group, do W3C
 - ▣ Em 2016 tornou-se uma “W3C Candidate Recommendation”
 - Desde então passou a ser amplamente suportada na maioria dos browsers

Propriedades Flexbox

5

- Flexbox consiste de várias propriedades “flex” e também de propriedades de alinhamento
 - ▣ As propriedades do Flexbox definem como elementos são dimensionados e de que forma ocupam o espaço do container
 - ▣ As propriedades de alinhamento definem o alinhamento vertical e horizontal dos elementos dentro de um container

Criando um container flex

6

- Neste primeiro exemplo veremos como criar um container flex com alguns itens flex

```
<!doctype html>
<title>Example</title>
<style>
.container {
  display: flex; }

.red {
  background: orangered;
  flex-grow: 1; }

.green {
  background: yellowgreen; }

.blue {
  background: steelblue; }

.container>div {
  font-size: 5vw;
  padding: .5em;
  color: white; }
</style>
```

```
<body>
  <div class="container">
    <div class="red">1</div>
    <div class="green">2</div>
    <div class="blue">3</div>
  </div>
</body>
</html>
```

```
.container {
  display: flex;
}
```

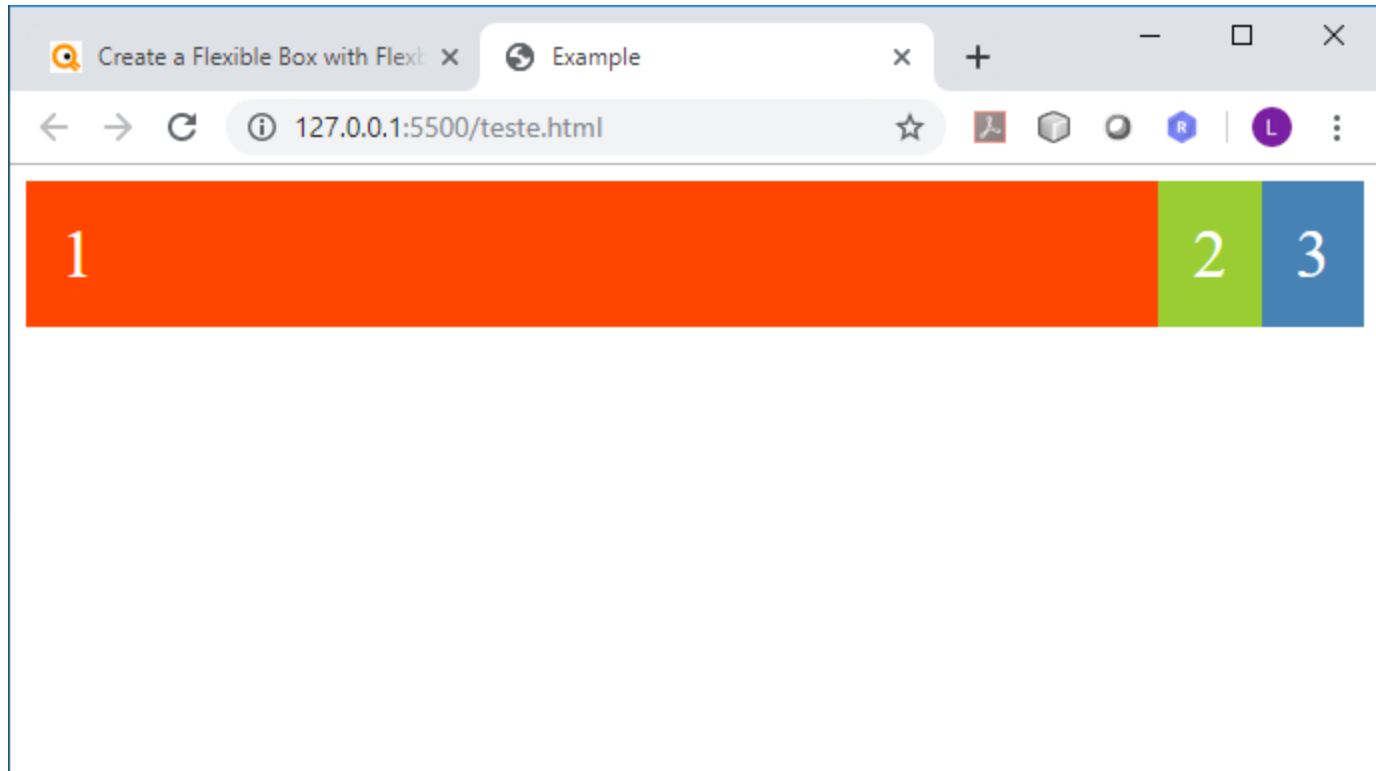
Este código declara o container como flex

Criando um container flex

7

```
.red {  
  background: orangered;  
  flex-grow: 1;  
}
```

A propriedade flex-grow ajusta o fator de crescimento, começando em zero. Neste caso, ajustando o valor 1 para .red fazemos com que o item ocupe todo o espaço disponível



Adicionando conteúdo

8

- Ao adicionar conteúdo a um item, este cresce de forma a acomodar o conteúdo

```
<!doctype html>
<title>Example</title>
<style>
  .container {
    display: flex; }

  .red {
    background: orangered;
    flex-grow: 1; }

  .green {
    background: yellowgreen; }

  .blue {
    background: steelblue; }

  .container>div {
    font-size: 5vw;
    padding: .5em;
    color: white; }
</style>
```

```
<body>
  <div class="container">
    <div class="red">1</div>
    <div class="green">2</div>
    <div class="blue">3
      <p>Flexbox items can grow and shrink as required to fit
their contents.</p>
    </div>
  </div>
</body>

</html>
```

1

2

3

Flexbox items can grow and shrink as required to fit their contents.

Alterando a largura (width) dos elementos

9

- Vamos ajustar a largura da caixa azul para 40% do viewport (width: 40vw)

```
<!doctype html>
<title>Example</title>
<style>
  .container {
    display: flex; }

  .red {
    background: orangered;
    flex-grow: 1; }

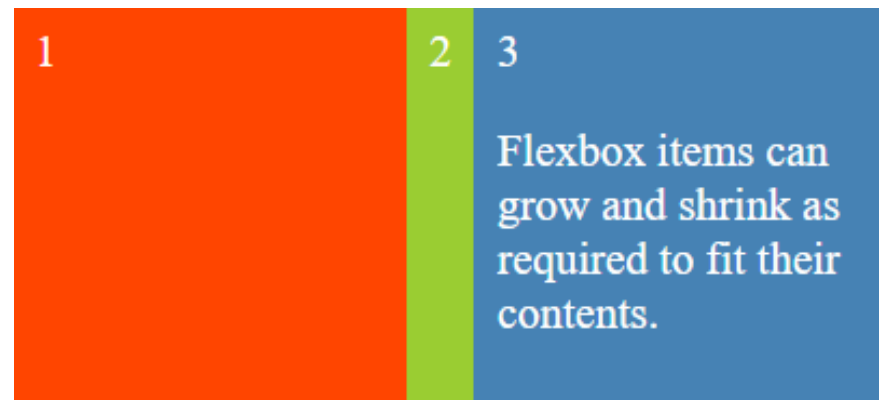
  .green {
    background: yellowgreen; }

  .blue {
    background: steelblue;
    width: 40vw; }

  .container>div {
    font-size: 5vw;
    padding: .5em;
    color: white; }
</style>
```

```
<body>
  <div class="container">
    <div class="red">1</div>
    <div class="green">2</div>
    <div class="blue">3
      <p>Flexbox items can grow and shrink as required to fit
their contents.</p>
    </div>
  </div>
</body>

</html>
```



Criando o layout de websites com Flexbox

10

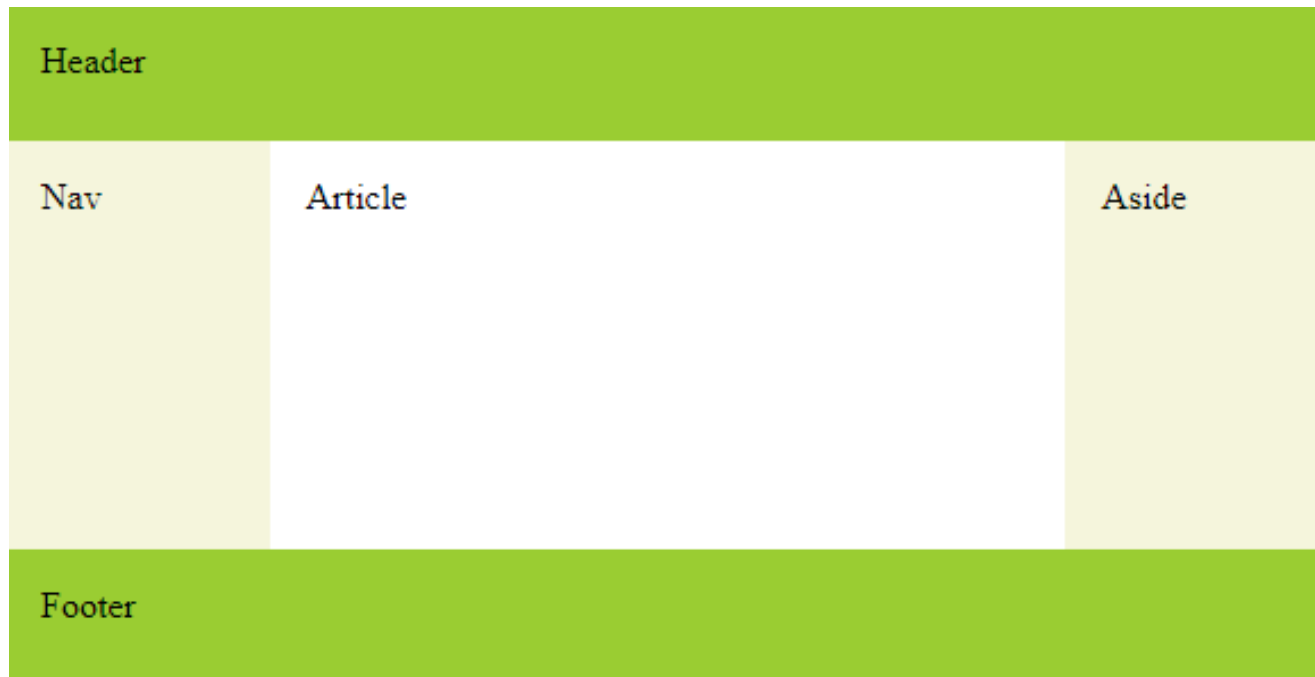
- Flexbox é ideal para criar layouts comumente usados por aplicações web
 - ▣ o famoso layout de 3 colunas, no qual as colunas devem ocupar todo espaço vertical, independentemente do conteúdo de cada coluna, é facilmente implementado



Criando o layout de websites com Flexbox

11

- Antes da existência do Flexbox, fazer um layout como este era extremamente trabalhoso



Criando o layout de websites com Flexbox

12

- Neste exemplo
 - ▣ #main é o container flex (parte central)
 - ▣ header e footer são elementos tipo bloco
- Assim:
 - ▣ Apenas a parte central é flex



Criando o layout de websites com Flexbox

13

```
#main {  
  display: flex;  
  min-height: calc(100vh - 40vh);  
}
```

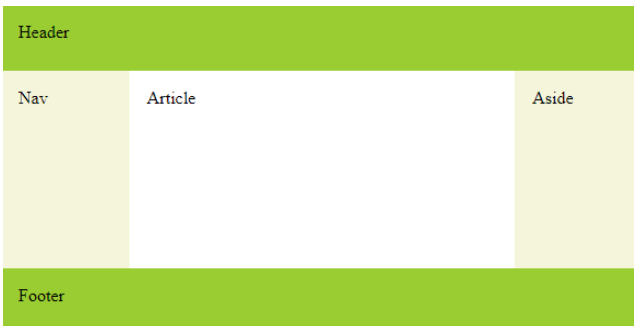
- Código que torna a div com id="main" flexível
 - ▣ min-height usa a função calc para fazer com que a altura mínima de main seja correspondente a 60% do viewport
 - vh = viewport height
 - ▣ Essa abordagem permite que as colunas ocupem toda a área do viewport, independentemente da quantidade do conteúdo exibido

Criando o layout de websites com Flexbox

14

```
#main > article {  
  flex: 1;  
}  
#main > nav,  
#main > aside {  
  flex: 0 0 20vw;  
  background: beige;  
}  
#main > nav {  
  order: -1;  
}
```

- O termo “*flex*” funciona como um atalho para 3 propriedades:
 - ▣ *flex-grow*, *flex-shrink* e *flex-basis*
- No primeiro caso
 - ▣ Somente *flex-grow* é ajustado
- No segundo caso
 - ▣ *flex-grow*=0, *flex-shrink*=0, *flex-basis*=20vw
 - *flex-basis* indica o tamanho inicial de um item *flex*. Então estamos ajustando o tamanho desses elementos a 20% do viewport



Criando o layout de websites com Flexbox

```
<!doctype html>
<title>Example</title>
<style>
  * {
    box-sizing: border-box; }

  body {
    margin: 0; }

  #main {
    display: flex;
    min-height: calc(100vh - 40vh); }

  #main>article {
    flex: 1; }

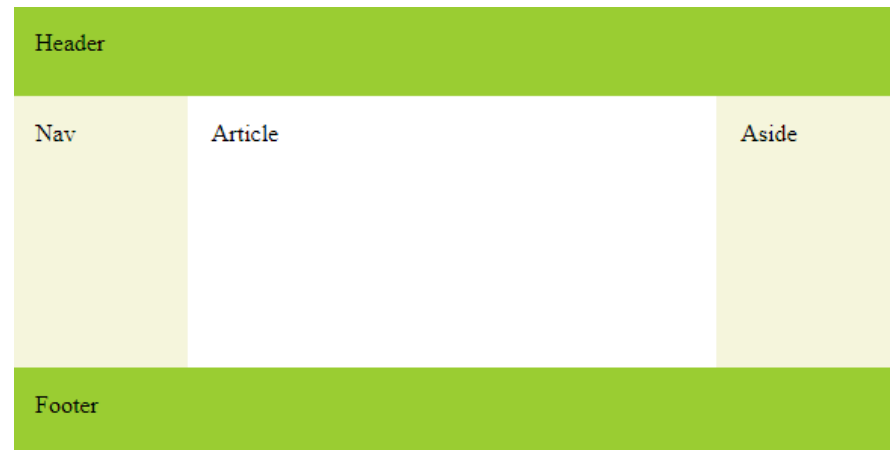
  #main>nav,
  #main>aside {
    flex: 0 0 20vw;
    background: beige; }

  #main>nav {
    order: -1; }

  header, footer, article, nav, aside {
    padding: 1em; }

  header, footer {
    background: yellowgreen;
    height: 20vh; }
</style>
```

```
<body>
  <header>Header</header>
  <div id="main">
    <article>Article</article>
    <nav>Nav</nav>
    <aside>Aside</aside>
  </div>
  <footer>Footer</footer>
</body>
```



Criando Layouts bidimensionais

16

- Flexbox é inerentemente unidirecional
 - ▣ Itens flex dentro de um container flex podem ser ajustados horizontal ou verticalmente, mas não ambos
- Para ajustar itens nas duas dimensões é necessário realizar aninhamento de containers flex
 - ▣ O exemplo a seguir ilustra essa abordagem



Criando Layouts bidimensionais

17

- Neste exemplo usamos *display: flex* tanto no container externo quanto no item vermelho. Porém no item vermelho usamos *flex-direction: column*, o que faz com os itens fiquem empilhados



Criando Layouts bidimensionais

18

```
<!doctype html>
<title>Example</title>
<style>
.container {
  display: flex; }

.red {
  background: orangered;
  display: flex;
  flex-direction: column; }

.green {
  background: yellowgreen; }

.blue {
  background: steelblue; }

.container div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  flex: 1; }

</style>
```

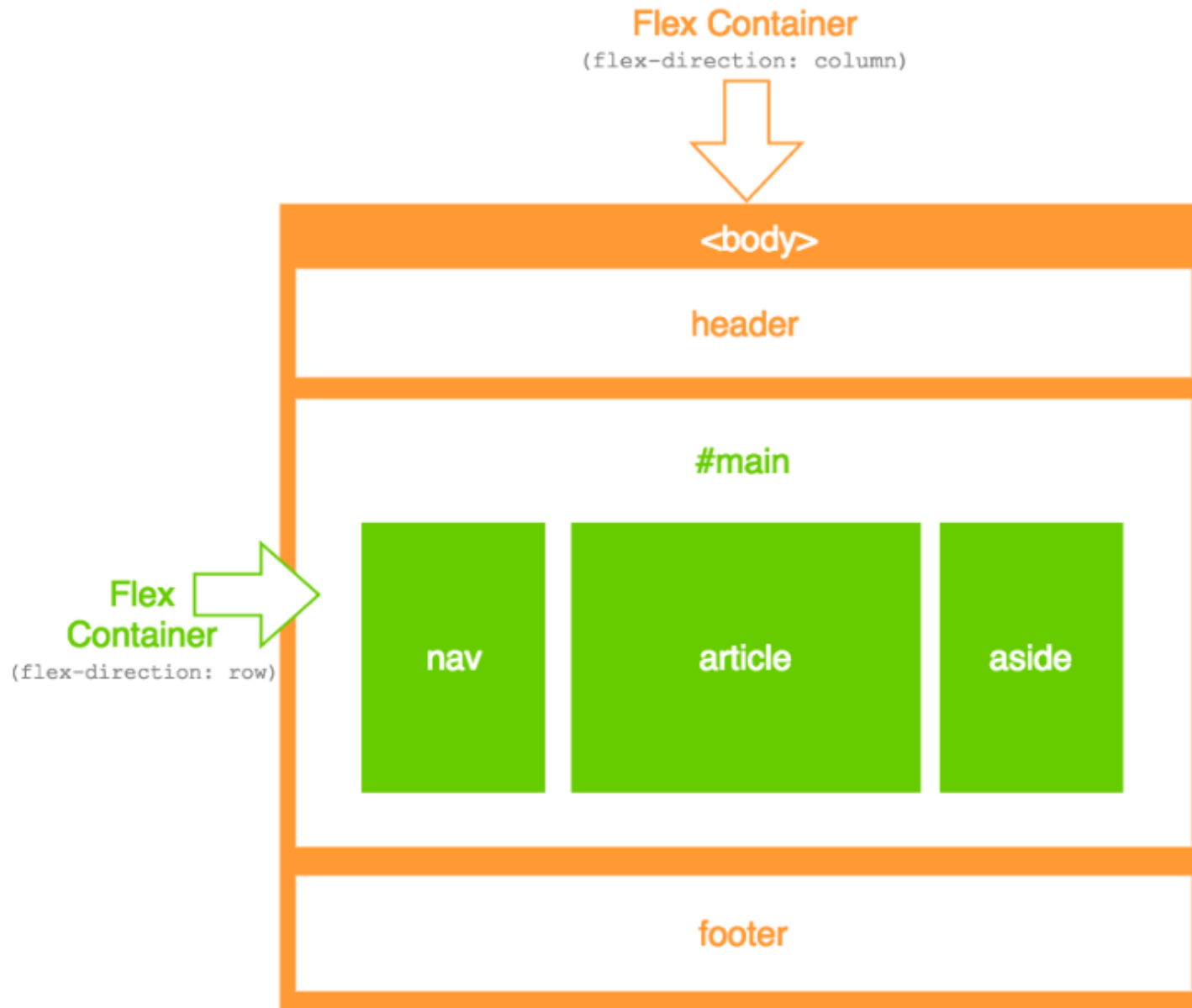
```
<body>
  <div class="container">
    <div class="red">1
      <div class="green">1a</div>
      <div class="blue">1b</div>
    </div>
    <div class="green">2</div>
    <div class="blue">3</div>
  </div>
</body>
```

Criando Layouts bidimensionais

19

- A página com layout de 3 colunas que construímos usa apenas Flexbox na seção do meio
 - ▣ Header e footer são blocos fixos
 - ▣ FlexBox foi usado apenas para fazer o layout horizontal de 3 itens flex
 - ▣ Nenhum item flex foi especificado na vertical (colunas)
 - ▣ Assim, usamos Flexbox de forma unidirecional
- Vamos modificar este exemplo usando Flexbox de forma bidirecional (horizontal e verticalmente)
 - ▣ Teremos um container flex para o **layout horizontal** e outro container flex para o **layout vertical**

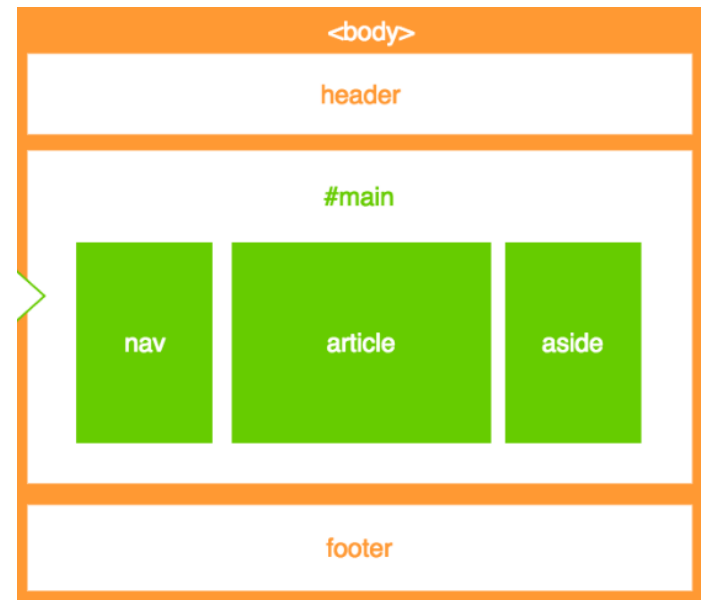
Criando Layouts bidimensionais



Criando Layouts bidimensionais

21

- Neste caso, o layout vertical vai envolver *header*, *main* e *footer*, como se toda a página fosse uma única grande coluna contendo 3 linhas, uma para o *header*, outra para o conteúdo principal (*main*) e a última para o *footer*
- O layout horizontal vai envolver as áreas *nav*, *article* e *aside*, as quais estão contidas na segunda linha (*main*), mas precisam ser dispostas horizontalmente. Assim, vamos tornar a segunda linha um container flex e fazer com que seu conteúdo flua horizontalmente



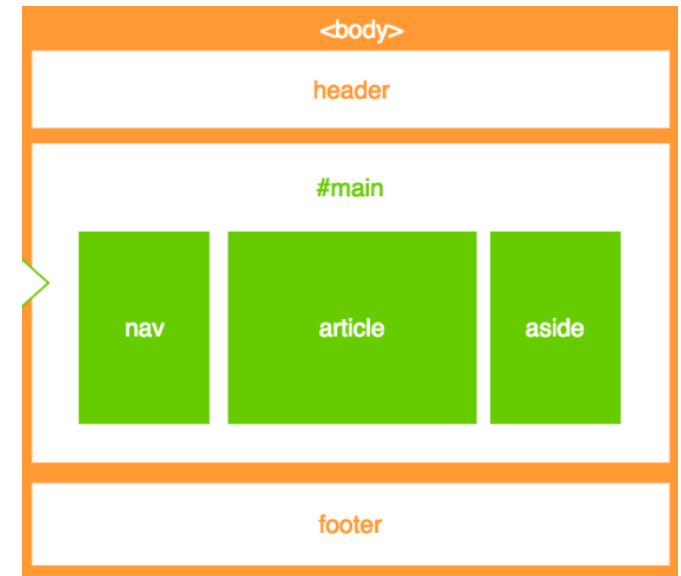
Criando Layouts bidimensionais

22

□ Código para o elemento <body>

```
body {  
  display: flex;  
  min-height: 100vh;  
  flex-direction: column;  
  margin: 0;  
}
```

- display: flex
 - ▣ torna <body> um container flex
- flex-direction: column
 - ▣ faz com que seus itens flex sejam mostrados numa coluna



Criando Layouts bidimensionais

23

□ Código para o container aninhado

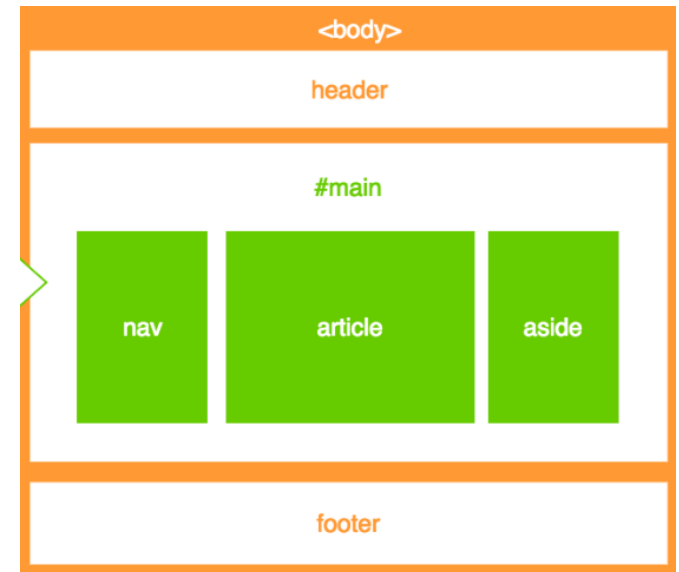
```
#main {  
  display: flex;  
  flex: 1;  
}
```

□ display: flex

- ▣ torna o container flex

□ flex: 1

- ▣ garante que o container cresça até ocupar todo espaço disponível



Criando Layouts bidimensionais

```
<!doctype html> <title>Example</title>
<style>
  * {
    box-sizing: border-box; }

  body {
    display: flex;
    min-height: 100vh;
    flex-direction: column;
    margin: 0; }

  #main {
    display: flex;
    flex: 1; }

  #main>article {
    flex: 1; }

  #main>nav,
  #main>aside {
    flex: 0 0 20vw;
    background: beige; }

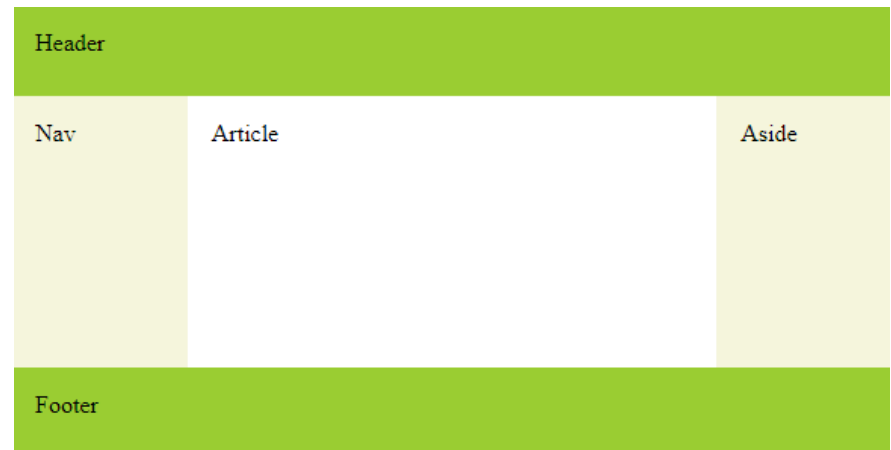
  #main>nav {
    order: -1; }

  header, footer {
    background: yellowgreen;
    height: 20vh;}

  header, footer, article, nav, aside {
    padding: 1em; }

</style>
```

```
<body>
  <header>Header</header>
  <div id="main">
    <article>Article</article>
    <nav>Nav</nav>
    <aside>Aside</aside>
  </div>
  <footer>Footer</footer>
</body>
```



Exercício 1

Entrega: 31/08

25

- Refaça sua página pessoal (exercício proposto na Aula 01) de forma que ela utilize o layout flex bidimensional apresentado nesta aula.

Criando objetos de mídia com Flexbox

26

- Objetos de mídia (media objects) é um nome dado a um componente UI que normalmente apresenta uma imagem de um lado e um texto do outro.
 - ▣ Objetivo é fazer um layout que permita que o texto fique de um lado e a imagem do outro
- Objetos de mídia precisam ser flexíveis
 - ▣ Expandir e contrair de acordo com o tamanho do container



Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.

Criando objetos de mídia com Flexbox

27

- Para o exemplo exibido, temos que criar um documento html com classes para facilitar a estilização

```
<div class="media">
  <img class="media-object">
  <div class="media-body">
    <h3 class="media-heading">...</h3>
    <p>...</p>
  </div>
</div>
```



Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.

Criando objetos de mídia com Flexbox

28

```
<div class="media">
  <img class="media-object">
  <div class="media-body">
    <h3 class="media-heading">...</h3>
    <p>...</p>
  </div>
</div>
```



Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.

- Fazemos flex: 1 na classe media-body, para assegurar que ela cresça ocupando todo o espaço restante
- Não fazemos o mesmo com a classe media-object já que queremos que a imagem ocupe exatamente o espaço de seu conteúdo
 - Mas ajustamos uma margem à direita para evitar que o texto “cole” na imagem

Criando objetos de mídia com Flexbox

29

```
<div class="media">
  <img class="media-object">
  <div class="media-body">
    <h3 class="media-heading">...</h3>
    <p>...</p>
  </div>
</div>
```



Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.

- Na div media (container) fazemos align-items: flex-start
 - ▣ Isso alinha os itens flex na parte de cima da linha do container
 - ▣ Por esse motivo a imagem aparece no topo
- Download das imagens:
 - ▣ <https://baldochi.unifei.edu.br/COM222/Imagens.zip>

Criando objetos de mídia com Flexbox

30

```
<!doctype html>
<title>Example</title>
<style>
  .media {
    display: flex;
    align-items: flex-start;
    background: #F6F3EB;
    padding: 1em;
    border-radius: 3px;
    max-width: 40em;
  }

  .media-object {
    margin-right: 1em;
  }

  .media-body {
    flex: 1;
  }

  .media-heading {
    margin: 0 0 .5em;
  }
</style>
```

```
<div class="media">
  
  <div class="media-body">
    <h3 class="media-heading">Organically Synthesized</h3>
    <p>Leverage agile frameworks to provide a robust synopsis for high level
    overviews. Iterative approaches to corporate strategy foster collaborative
    thinking to further the overall value proposition.</p>
  </div>
</div>
```



Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.

Exercício 2

31

- Refaça o exemplo posicionando a imagem à direita

Organically Synthesized

Leverage agile frameworks to provide a robust synopsis for high level overviews. Iterative approaches to corporate strategy foster collaborative thinking to further the overall value proposition.



Galeria de fotos com Flex

32



Galeria de fotos com Flex

flex-direction: row

33

```
<!doctype html>
<title>Example</title>
<style>
  .cards {
    display: flex;
    flex-wrap: wrap;
    align-items: flex-start;
    flex-direction: row;
    max-height: 100vh;
  }

  .cards img {
    margin: 10px;
    border: 3px solid #000;
    box-shadow: 3px 3px 8px 0px rgba(0, 0, 0, 0.3);
    max-width: 23vw;
  }
</style>
```

```
<main class="cards">
  
  
  
  
  
  
  
  
  
</main>
```



Galeria de fotos com Flex

flex-direction: column

34

```
<!doctype html>
<title>Example</title>
<style>
  .cards {
    display: flex;
    flex-wrap: wrap;
    align-items: flex-start;
    flex-direction: column;
    max-height: 100vh;
  }

  .cards img {
    margin: 10px;
    border: 3px solid #000;
    box-shadow: 3px 3px 8px 0px rgba(0, 0, 0, 0.3);
    max-width: 23vw;
  }
</style>
```

```
<main class="cards">
  
  
  
  
  
  
  
  
  
</main>
```



Galeria de fotos com Flex

align-items: stretch

35

```
<!doctype html>
<title>Example</title>
<style>
  .cards {
    display: flex;
    flex-wrap: wrap;
    align-items: stretch;
  }

  .cards img {
    margin: 10px;
    border: 3px solid #000;
    box-shadow: 3px 3px 8px 0px rgba(0, 0, 0, 0.3);
    max-width: 25vw;
  }
</style>
```

```
<main class="cards">
  
  
  
  
  
  
</main>
```



Galeria de fotos com Flex

align-items: flex-start

36

```
<!doctype html>
<title>Example</title>
<style>
.cards {
  display: flex;
  flex-wrap: wrap;
  align-items: flex-start;
}
.cards img {
  margin: 10px;
  border: 3px solid #000;
  box-shadow: 3px 3px 8px 0px rgba(0,0,0,0.3);
  max-width: 25vw;
}
</style>
```

```
<main class="cards">
  
  
  
  
  
  
</main>
```



Galeria de fotos com Flex

align-items: flex-end

37

```
<!doctype html>
<title>Example</title>
<style>
.cards {
  display: flex;
  flex-wrap: wrap;
  align-items: flex-end;
}
.cards img {
  margin: 10px;
  border: 3px solid #000;
  box-shadow: 3px 3px 8px 0px rgba(0,0,0,0.3);
  max-width: 25vw;
}
</style>
```

```
<main class="cards">
  
  
  
  
  
  
</main>
```



Galeria de fotos com Flex

align-items: center

38

```
<!doctype html>
<title>Example</title>
<style>
  .cards {
    display: flex;
    flex-wrap: wrap;
    align-items: center;
  }
  .cards img {
    margin: 10px;
    border: 3px solid #000;
    box-shadow: 3px 3px 8px 0px rgba(0,0,0,0.3);
    max-width: 25vw;
  }
</style>
```

```
<main class="cards">
  
  
  
  
  
  
</main>
```



Exercício 3 -- entregar

39

- Implemente uma galeria flex contendo seis imagens de locais turísticos
 - ▣ Sua galeria deve exibir, por padrão, 3 imagens por linha (colocando `max-width: 25vw` deve dar o resultado esperado)
 - ▣ Ao clicar na foto da galeria, deve-se carregar uma nova página, na qual a foto é mostrada como um media object, contendo informação textual a respeito da foto
 - Dica: Use texto do Wikipedia

Exercício 3

Entrega: 31/08

40

□ Exemplo



Cristo Redentor

Cristo Redentor é uma estátua art déco que retrata Jesus Cristo, localizada no topo do morro do Corcovado, a 709 metros acima do nível do mar, no Parque Nacional da Tijuca, com vista para a maior parte da cidade do Rio de Janeiro, Brasil. Em 2007 foi eleito informalmente como uma das sete maravilhas do mundo moderno.