

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**INSTITUTO METRÓPOLE DIGITAL**

# **AULA 7**

HTML LAYOUT

**PROF. JANIHERYSON FELIPE**

---

# CONTEÚDO DESSA AULA

- CONHECER TAGS SEMÂNTICAS DO HTML E SUA IMPORTÂNCIA
- CONHECER AS PROPRIEDADES FLEX APLICADAS AOS CONTÊINERES
- CONHECER AS PROPRIEDADES FLEX APLICADAS AOS ITENS
- DISCUSSÕES E DÚVIDAS GERAIS.

## **ELEMENTOS DE LAYOUT DO HTML 5**

- As tags de layout foram criadas para facilitar a estruturação do site semanticamente.
- A estruturação semântica é importante, pois fica possível entender melhor o que cada parte do código HTML contém e significa. Dessa forma, outros sistemas podem entender melhor seu documento a partir da análise do código. Além disso, a estruturação semântica permite reaproveitar melhor estilos CSSs, uma vez que as tags são padronizadas.

# ELEMENTOS DE LAYOUT DO HTML 5

## <header>

- A tag <header> foi criada para conter conteúdo introdutório do seu documento HTML. Nela, normalmente ficam informações gerais como logomarca e navegação.

## <nav>

- A tag <nav> foi criada para conter as informações de navegação do documento HTML. É nessa tag que deve conter os *links* para outras páginas do seu site/sistema.

# ELEMENTOS DE LAYOUT DO HTML 5

## **<section>**

- A tag `<section>` define uma seção de um documento HTML.

## **<article>**

- A tag `<article>` foi criada para indicar o conteúdo principal do documento HTML. O conteúdo do `<article>` deve ser distribuído independentemente do restante do site. Exemplos de conteúdos que podem ficar na tag `<article>`

# ELEMENTOS DE LAYOUT DO HTML 5

## **<aside>**

- A tag `<aside>` define algum conteúdo complementar ao conteúdo principal. Caso não seja exibido, não deve prejudicar o entendimento do conteúdo principal. O conteúdo `<aside>` geralmente é colocado como uma barra lateral em um documento.

## **<footer>**

- A tag `<footer>` tem o objetivo de indicar o rodapé do documento HTML.

```

<body>
  <header>
    Cidades
  </header>
  <section>
    <nav>
      <ul>
        <li><a href="#">Londres</a></li>
        <li><a href="#">Paris</a></li>
        <li><a href="#">Rio de Janeiro</a></li>
      </ul>
    </nav>
    <article>
      <h2>Londres</h2>
      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
        Inventore qui sed numquam nam quo obcaecati,
        doloremque praesentium similique sequi quia nihil e
        aque fuga optio cumque beatae voluptatem at
        architecto ipsa.</p>
    </article>
  </section>
  <footer>
    <p><a href="http://imd.ufrn.br">www.imd.ufrn.br</a></p>
  </footer>
</body>

```

# ELEMENTOS DE LAYOUT DO HTML 5

# ELEMENTOS DE LAYOUT DO HTML 5

Cidades

[Londres](#)  
[Paris](#)  
[Rio de Janeiro](#)

## Londres

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Inventore qui sed numquam nam quo obcaecati, doloremque praesentium similique sequi quia nihil eaque fuga optio cumque beatae voluptatem at architecto ipsa.

[www.imd.ufrn.br](http://www.imd.ufrn.br)



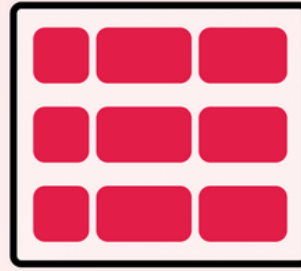
# FLEXBOX CONTÊINER

O *flex* é uma ferramenta do CSS que facilita a organização dos conteúdos de um contêiner. Para ativar o *flex* em determinado contêiner, basta utilizar na propriedade "display" o valor "flex".

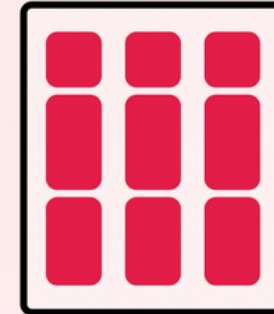
```
section {  
  display: flex;  
}
```

# CSS Flexbox

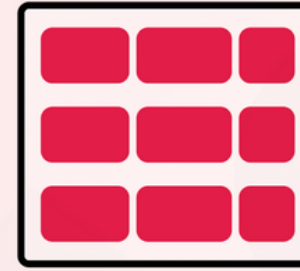
## flex-direction



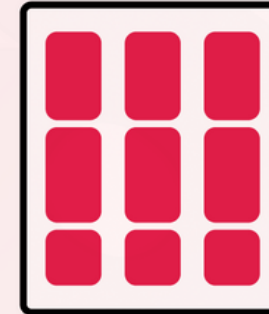
row



column



row-reverse

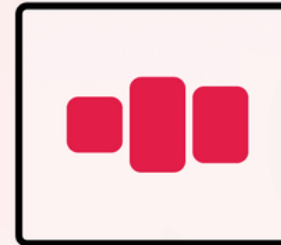


column-reverse

## align-items



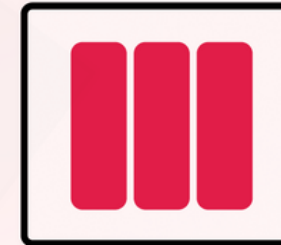
flex-start



center



flex-end



stretch

## justify-content



flex-start



center



flex-end



space-between

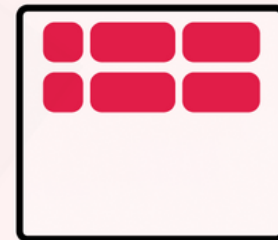


space-around

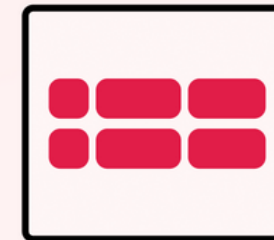


space-evenly

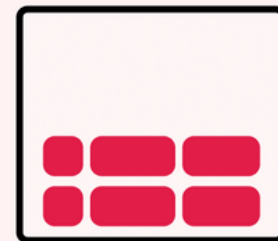
## align-content



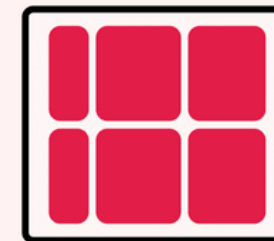
flex-start



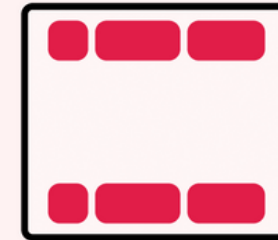
center



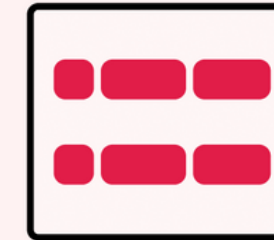
flex-end



stretch



space-between



space-around

# FLEXBOX CONTÊNER

# FLEX-DIRECTION

O "flex-direction" indica a direção de como os conteúdos de um container com `display flex`, irão se distribuir. Veja algumas opções:

Valor	Descrição
row	Dispõe os elementos um ao lado do outro
row-reverse	Dispõe os elementos um ao lado do outro em ordem reversa (da direita para a esquerda)
column	Dispõe os elementos um abaixo do outro (começando de cima)
column-reverse	Dispõe os elementos um acima do outro (começando de baixo)

## FLEX-WRAP

Essa propriedade controla como se dará a quebra, quando não há espaço suficiente para mostrar todos os itens em uma mesma linha (ou coluna, para o caso da "flex-direction" ser "column").

Valor	Descrição
wrap	Quebra a linha posicionando os itens na linha seguinte (ou coluna)
wrap-reverse	Quebra a linha, posicionamento os itens na linha seguinte em ordem reversa
no-wrap	Não quebra a linha

# FLEX-JUSTIFY

O "flex-justify" é a propriedade responsável por controlar como os elementos serão distribuídos dentro do contêiner.

Valor	Descrição
flex-start	Distribui os itens do início para o fim: da esquerda para a direita, em caso de linha, e, de cima para baixo, em caso de coluna.
flex-end	Distribui os itens do fim para o início: da direita para a esquerda, em caso de linha e, de baixo para cima, em caso de coluna.
center	Distribui os itens a partir do centro
space-between	Os itens terão espaço entre eles
space-around	Os itens terão espaço antes, depois e entre eles
space-evenly	Os itens terão espaço igual ao seu redor

# ALIGN-ITEMS

A propriedade "align-items" controla o comportamento do alinhamento dos itens, com relação ao eixo vertical, no caso de linhas, e horizontal, no caso de colunas.

Valor	Descrição
stretch	Força o item a ocupar todo o espaço disponível (padrão)
flex-start	Ocupa apenas o espaço necessário, alinhando de cima para baixo (no caso de linha).
flex-end	Ocupa apenas o espaço necessário, alinhando de baixo para cima (no caso de linha).
center	Ocupa apenas o espaço necessário, alinhando pelo centro.
baseline	Ocupa apenas o espaço necessário, alinhando o conteúdo do contêiner.

# ALIGN-CONTENT

A propriedade "align-content" controla o comportamento de como as linhas/colunas estarão alinhadas entre si.

Valor	Descrição
stretch	Força as linhas/colunas a ocupar todo o espaço disponível (padrão)
flex-start	As linhas são compactadas em direção ao início do flex container
flex-end	As linhas são compactadas em direção ao fim do flex container
center	As linhas são compactadas em direção ao centro do flex container
space-between	As linhas são distribuídas uniformemente no flex container
space-around	As linhas são distribuídas uniformemente no flex container, com metade do espaço nas extremidades
space-evenly	As linhas são distribuídas uniformemente no flex container, com espaço igual ao redor delas

# INTRODUÇÃO

- Agora veremos algumas propriedades para os itens de um contêiner flex. Ou seja, as propriedades que devem ser aplicadas aos itens e não ao contêiner, como as vistas até aqui.





## **FLEX-GROW**

- O "flex-grow" é uma propriedade que controla o tamanho dos itens e como se comportarão em relação ao tamanho da tela. Trata-se de um valor numérico em que o valor padrão é o 0. No caso de todos os itens possuírem "flex-grow" 0, indica que ocuparão apenas a área suficiente para apresentar seu conteúdo. Se o valor diferir de 0, o espaço total será dividido de forma a respeitar a proporção estabelecida por cada valor de "flex-grow".

## **FLEX-SHRINK**

- A propriedade "flex-shrink" especifica como o item será reduzido em relação ao restante dos itens flexíveis no mesmo contêiner. Trata-se de um valor numérico que possui 1 como valor padrão

## **FLEX-BASIS**

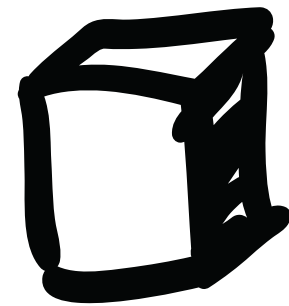
- A propriedade "flex-basis" especifica o comprimento inicial de um item flexível. Trata-se de um valor numérico que pode ser definido em pixels.

# FLEX

- O atributo "flex" é um atalho para todos os outros atributos vistos nesta aula

```
section>div {  
  flex: 0 1 auto;  
}
```

- No exemplo acima, a propriedade flex foi configurada com "flex-grow" com valor "0", o "flex-shrink" com valor "1" e o "flex-basis" com valor "auto". Perceba que a ordem é fundamental.



DÚVIDAS?