



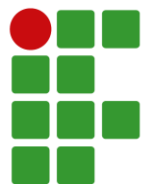
**Licenciatura  
em Computação**

Curso de Licenciatura em Computação  
Disciplina de Programação I  
Professor Tauã M. Cabreira

# AULA 13

---

CSS Layout



**INSTITUTO FEDERAL**  
Sul-rio-grandense  
Câmpus Pelotas

# Layout Centralizado

---

- Elementos nível de bloco podem ser centralizados horizontalmente através da propriedade **margin: auto**.
- Deve-se definir uma **largura específica** para o elemento através da propriedade **width** (em px ou %). O espaço restante é automaticamente distribuído igualmente em ambos os lados.
- O alinhamento centralizado não funciona sem a definição da propriedade **width** ou se ela for configurada em 100%.

# Layout Centralizado

## Topo da página

```
<header>  
  <h1>Topo da página</h1>  
</header>
```

Trecho HTML

Centralizando um elemento  
em relação à página.

```
header{  
  width: 50%;  
  margin: auto;  
  padding: 5px;  
  background: #036;  
  font-family: Tahoma;  
  color: #fff;  
}
```

Trecho CSS

# Layout Centralizado

## Topo da página

```
<header>  
  <h1>Topo da página</h1>  
</header>
```

Trecho HTML

Centralizando um texto em relação a um elemento.

```
header{  
  width: 50%;  
  margin: auto;  
  padding: 5px;  
  background: #036;  
  font-family: Tahoma;  
  color: #fff;  
  text-align: center;  
}
```

Trecho CSS

# Alinhamento Vertical

Topo da página

```
<header>  
  <h1>Topo da página</h1>  
</header>
```

Trecho HTML

O alinhamento vertical pode ser feito através da propriedade **padding**.

```
header{  
  width: 50%;  
  margin: auto;  
  padding: 20px 0;  
  background: #036;  
  font-family: Tahoma;  
  color: #fff;  
  text-align: center;  
}
```

Trecho CSS

# Propriedade **float**

---

- A propriedade **float** especifica como um elemento deve **flutuar** no layout.
- Esta propriedade é utilizada para posicionamento e formatação de conteúdo.
- Podem ser utilizados os seguintes valores: **left**, **right**, **none** e **inherit**.

# Propriedade **float**

```
<div class="bloco">
  <h1>Box de Conteúdo 1</h1>
  <p>Aqui vai algum conteúdo... </p>
</div>
<div class="bloco">
  <h1>Box de Conteúdo 2</h1>
  <p>Aqui vai algum conteúdo...</p>
</div>
```

Trecho HTML

```
.bloco{
  width: 200px;
  height: 200px;
  padding: 10px;
  background: #c1c1c1;
  margin: 20px;
  font-family: Tahoma;
  text-align: center;
  float: left;
}
```

Trecho CSS

# Propriedade **float**

---

## **Box de Conteúdo 1**

Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.

## **Box de Conteúdo 2**

Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.

Sem float

## **Box de Conteúdo 1**

Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.

## **Box de Conteúdo 2**

Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.  
Aqui vai algum conteúdo.

float: left



# Propriedade **overflow**

```
<div class="principal">
  <div class="bloco">
    <h1>Box de Conteúdo 1</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
  <div class="bloco">
    <h1>Box de Conteúdo 2</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
</div>
```

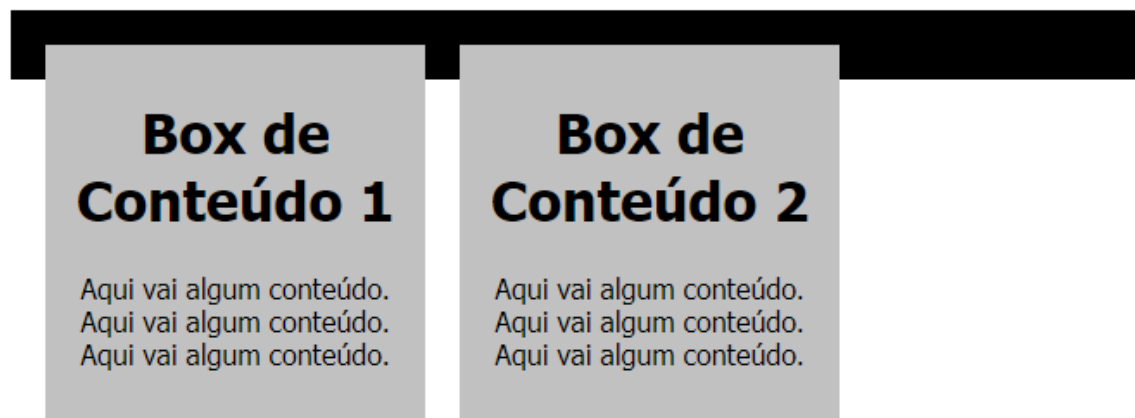
Trecho HTML

```
.bloco{
  width: 200px;
  height: 200px;
  padding: 10px;
  background: #c1c1c1;
  margin-right: 20px;
  font-family: Tahoma;
  text-align: center;
  float: left;
}

.principal{
  padding: 20px;
  background-color: #000;
}
```

Trecho CSS

# Propriedade **overflow**

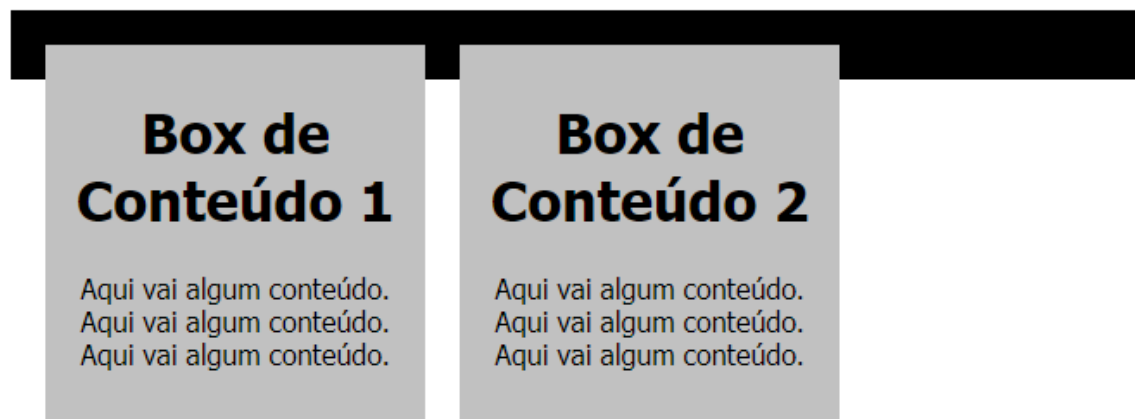


- As **divs** da classe **bloco** estão inseridas dentro da **div** com a classe **principal**.
- Estes elementos flutuam lado a lado com a propriedade **float: left**.

```
.bloco{  
    width: 200px;  
    height: 200px;  
    padding: 10px;  
    background: #c1c1c1;  
    margin-right: 20px;  
    font-family: Tahoma;  
    text-align: center;  
    float: left;  
}  
  
.principal{  
    padding: 20px;  
    background-color: #000;  
}
```

Trecho CSS

# Propriedade **overflow**

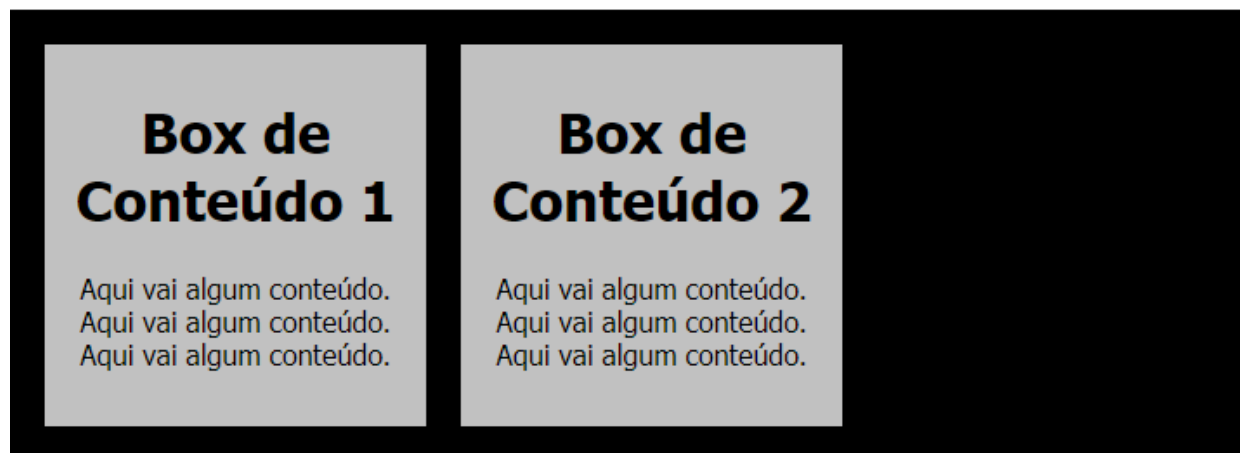


- Se os elementos internos (**.bloco**) forem maiores que o elemento container (**.principal**) e estiverem **flutuando**, eles irão **ultrapassar** o elemento container.

```
.bloco{  
    width: 200px;  
    height: 200px;  
    padding: 10px;  
    background: #c1c1c1;  
    margin-right: 20px;  
    font-family: Tahoma;  
    text-align: center;  
    float: left;  
}  
  
.principal{  
    padding: 20px;  
    background-color: #000;  
}
```

Trecho CSS

# Propriedade **overflow**



- Este problema pode ser solucionado através da propriedade **overflow: auto**.
- O elemento container se ajusta ao tamanho dos elementos internos.

```
.bloco{  
    width: 200px;  
    height: 200px;  
    padding: 10px;  
    background: #c1c1c1;  
    margin-right: 20px;  
    font-family: Tahoma;  
    text-align: center;  
    float: left;  
}  
  
.principal{  
    padding: 20px;  
    background-color: #000;  
    overflow: auto;  
}
```

Trecho CSS

# Propriedade **clear**

```
<div class="principal">
  <div class="bloco">
    <h1>Box de Conteúdo 1</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
  <div class="bloco">
    <h1>Box de Conteúdo 2</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
  <footer>
    <h1>Box do rodapé com informações.</h1>
  </footer>
</div>
```

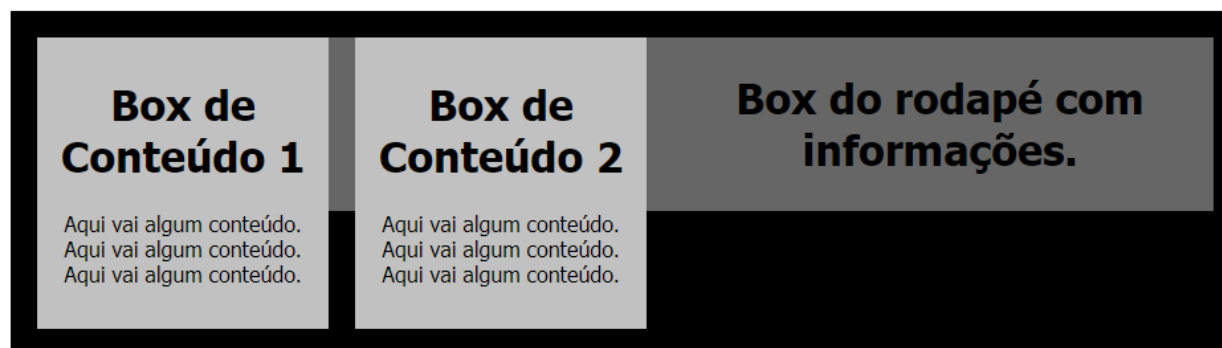
Trecho HTML

```
/* ... Estilização da
classe .principal e .bloco */

footer{
  padding: 5px;
  background: #666;
  font-family: Tahoma;
  text-align: center;
}
```

Trecho CSS

# Propriedade **clear**

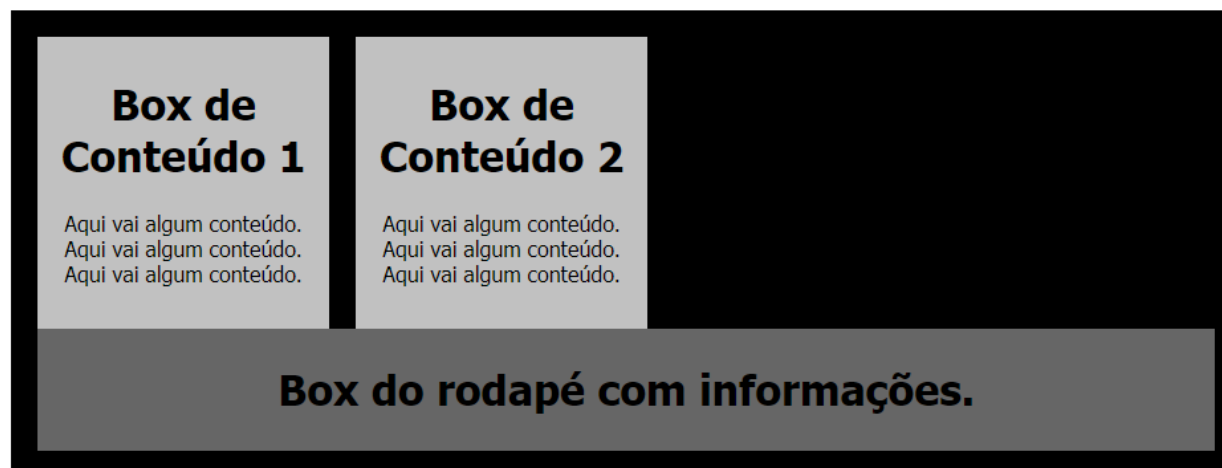


- Ao adicionar um novo elemento (**footer**) após os elementos flutuantes da classe **.bloco**, ocorre uma **sobreposição**.

```
/* ... Estilização da  
classe .principal e .bloco */  
  
footer{  
    padding: 5px;  
    background: #666;  
    font-family: Tahoma;  
    text-align: center;  
}
```

Trecho CSS

# Propriedade **clear**



- Para solucionar este problema, deve-se usar a propriedade **clear: left** no elemento **footer**.

```
/* ... Estilização da  
classe .principal e .bloco */  
  
footer{  
    padding: 5px;  
    background: #666;  
    font-family: Tahoma;  
    text-align: center;  
}
```

Trecho CSS

# Propriedade **clear**

---

- A propriedade **clear** especifica como elementos subsequentes aos elementos flutuantes devem se comportar.
- A propriedade **clear** deve ser usada em concordância com o float. Se **float: left**, então **clear: left**.
- Podem ser utilizados os seguintes valores: **left**, **right**, **none**, **both** e **inherit**.



# Imagem Centralizada



```

```

Trecho HTML

```
img{  
  width: 50%;  
  display: block;  
  margin: auto;  
}
```

Trecho CSS

Centralizando uma imagem.

# Imagem e conteúdo

Londres, capital da Inglaterra e do Reino Unido, é uma cidade do século 21 com uma história que remonta à era romana. Seu centro abriga as sedes imponentes do Parlamento, a famosa torre do relógio do Big Ben e a Abadia de Westminster, local de coroação dos monarcas britânicos. Do outro lado do rio Tâmisa, a roda gigante London Eye tem vista panorâmica do complexo cultural da margem sul e de toda a cidade.



```
<div class="bloco">
  
  <span>Londres, capital da Inglaterra...</span>
</div>
```

Trecho HTML

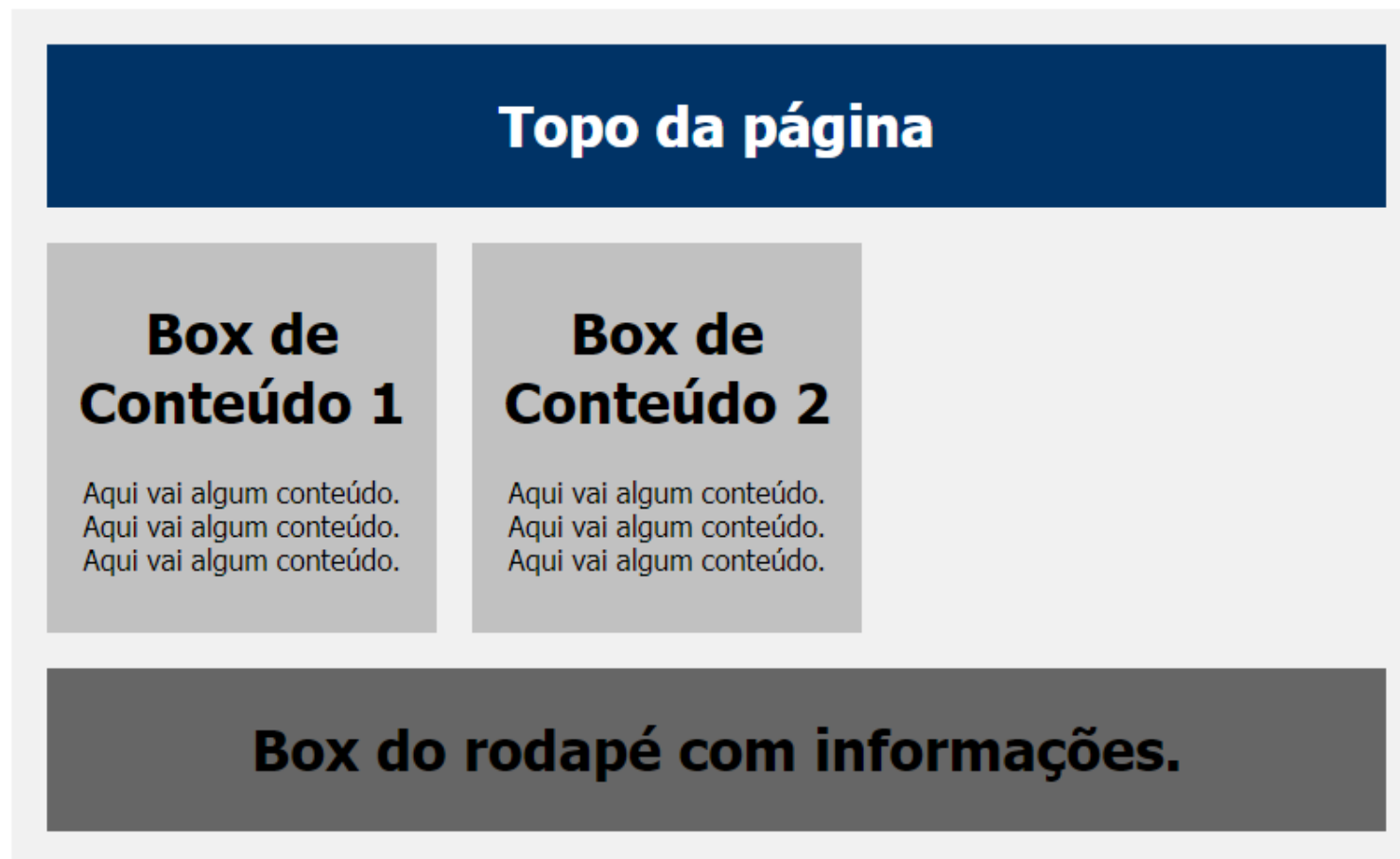
```
img{
  width: 50%;
  margin-left: 20px;
  float: right;
}

.bloco{
  width: 50%;
  padding: 20px;
  background: #c1c1c1;
  font-family: Tahoma;
  text-align: justify;
  overflow: auto;
}
```

Trecho CSS

# Grid Box

---



# Grid Box

```
<div class="principal">
  <header>
    <h1>Topo da página</h1>
  </header>
  <div class="bloco">
    <h1>Box de Conteúdo 1</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
  <div class="bloco">
    <h1>Box de Conteúdo 2</h1>
    <p>Aqui vai algum conteúdo...</p>
  </div>
  <footer>
    <h1>Box do rodapé com informações.</h1>
  </footer>
</div>
```

Trecho HTML

# Grid Box

```
.principal{  
    width: 75%;  
    margin: auto;  
    padding: 20px;  
    background-color: #f1f1f1;  
    font-family: Tahoma;  
    text-align: center;  
    overflow: auto;  
}
```

```
header{  
    padding: 5px;  
    background: #036;  
    color: #fff;  
}
```

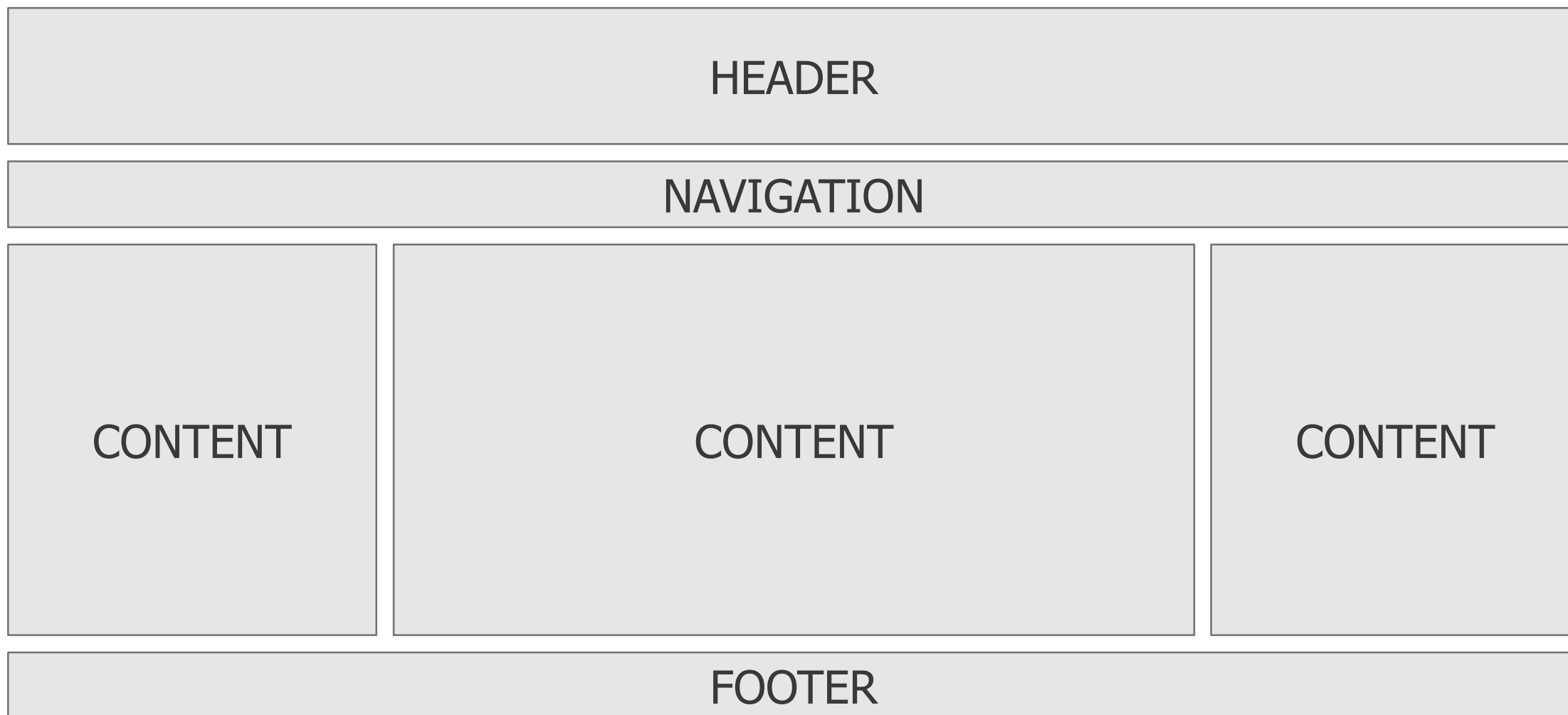
```
.bloco{  
    width: 200px;  
    height: 200px;  
    padding: 10px;  
    background: #c1c1c1;  
    margin: 20px 20px 20px 0;  
    float: left;  
}
```

```
footer{  
    padding: 5px;  
    background: #666;  
    clear: left;  
}
```

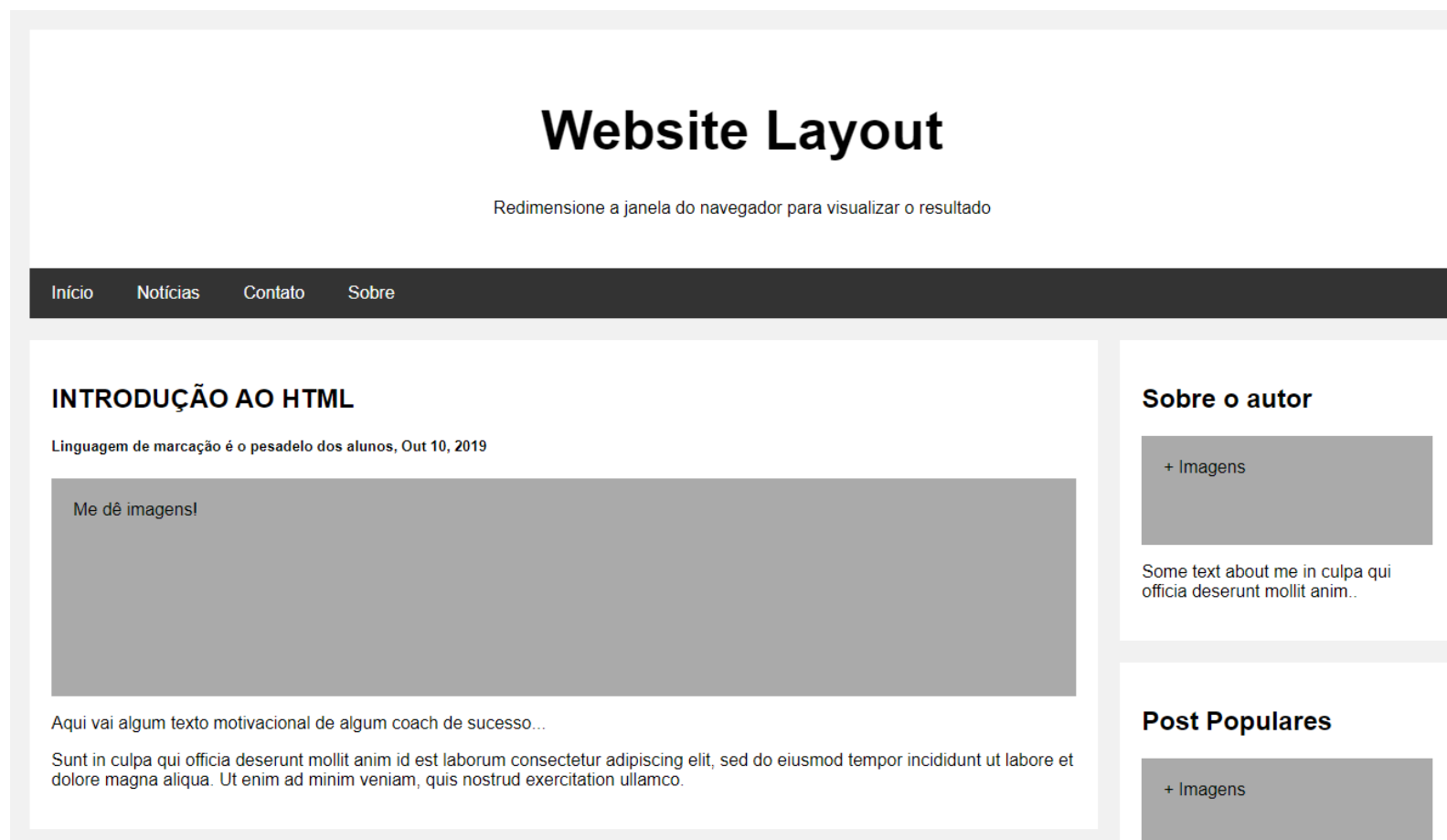
Trecho CSS

# Website Layout

---



# Responsive Web Design (RWD)



# Responsive Web Design (RWD)

## Website Layout

Redimensione a janela do navegador para visualizar o resultado

[Início](#) [Notícias](#) [Contato](#) [Sobre](#)

### INTRODUÇÃO AO HTML

Linguagem de marcação é o pesadelo dos alunos, Out 10, 2019

Me dê imagens!

## Website Layout

Redimensione a janela do navegador para visualizar o resultado

[Início](#)

[Notícias](#)

[Contato](#)

[Sobre](#)

### INTRODUÇÃO AO HTML

Linguagem de marcação é o pesadelo dos alunos, Out 10, 2019

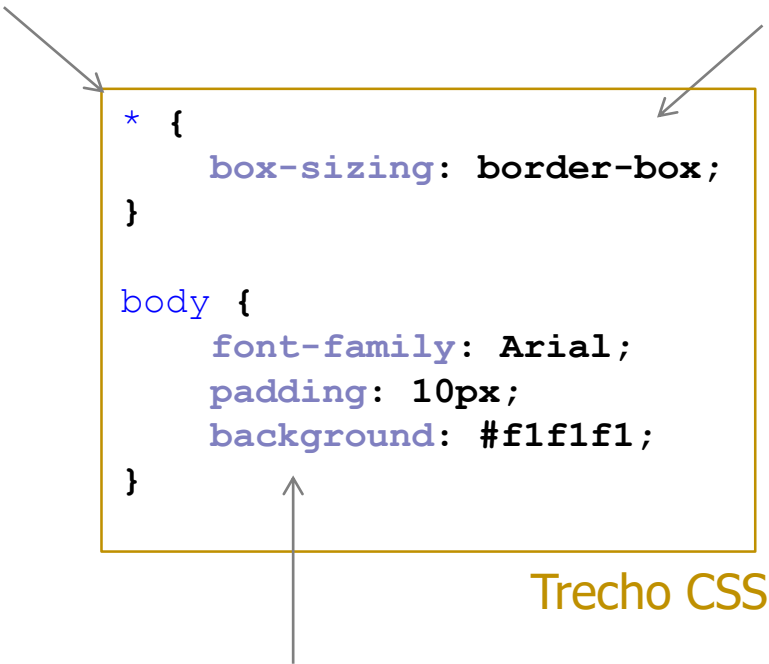
Me dê imagens!



# RWD

A **seletor** `*` é um seletor universal que aplica as propriedades CSS a todos os elementos HTML de um documento.

A propriedade **`box-sizing: border-box`** é utilizada para incluir o **padding** e o **border** na largura total do elemento.



```
* {  
    box-sizing: border-box;  
}  
  
body {  
    font-family: Arial;  
    padding: 10px;  
    background: #f1f1f1;  
}
```

Trecho CSS

## Estilização Geral

O seletor **`body`** define as propriedades CSS que serão aplicadas no corpo do documento. Todos os elementos HTML contidos nele terão a **fonte Arial**.

# RWD

O topo do website é estilizado através do seletor **header** com texto centralizado e cor de fundo branca.

```
header {  
  padding: 30px;  
  text-align: center;  
  background: white;  
}  
  
header h1 {  
  font-size: 50px;  
}
```

Trecho CSS

## Estilização do Topo

O elemento **h1** que está dentro do **header** é estilizado com o tamanho de fonte de 50px.

## Estilização do Menu


```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333;  
}  
  
li {  
    float: left;  
}  
  
li a {  
    color: #fff;  
    text-decoration: none;  
    padding: 14px 20px;  
    display: block;  
}  
  
li a:hover {  
    background-color: #ccc;  
    color: #000;  
}
```

Trecho CSS

# RWD

As áreas de conteúdo são definidas pelos elementos **section** e **aside**. O tamanho destes elementos é definido em porcentagem para que eles se ajustem ao tamanho da tela.


## Estilização de Conteúdo



```
/* Conteúdo principal */
section {
  float: left;
  width: 75%;
}

/* Conteúdo lateral */
aside {
  float: left;
  width: 25%;
  padding-left: 20px;
}
```

Trecho CSS



A propriedade **float: left** é utilizada para flutuar os dois elementos de nível de bloco lado a lado.

# RWD

Os elementos com a classe **.fakeimg** simulam o espaço de uma imagem.

```
/* Simulando o espaço de uma imagem */  
.fakeimg {  
    background-color: #aaa;  
    width: 100%;  
    padding: 20px;  
}  
  
/* Bloco de conteúdo */  
.bloco {  
    background-color: white;  
    padding: 20px;  
    margin-top: 20px;  
}
```

Os elementos com a classe **.bloco** definem cada trecho de conteúdo dentro das áreas principais (**section** e **aside**).

Trecho CSS

# RWD

- O seletor **::after** é utilizado para inserir conteúdo após elementos HTML. Ele é empregado com a propriedade CSS **content**.

```
/* Limpa os floats após  
   as colunas de conteúdo */  
.conteudo::after {  
  content: "";  
  display: table;  
  clear: both;  
}  
  
/* Footer */  
footer {  
  padding: 20px;  
  text-align: center;  
  background: #ddd;  
  margin-top: 20px;  
}
```

A propriedade **display: table** permite apresentar um elemento HTML como uma tabela. Já a propriedade **clear: both** não permite elementos flutuantes após a área de conteúdo.

Desta forma, o rodapé é “jogado” para baixo da área de conteúdo.

Trecho CSS

# RWD

- **Media query** é uma técnica para incluir um conjunto de propriedades CSS se uma determinada **condição for verdadeira**.

```
@media screen and (max-width: 800px) {  
  section, aside {  
    width: 100%;  
    padding: 0;  
  }  
}  
  
@media screen and (max-width: 500px) {  
  li {  
    float: none;  
    width: 100%;  
  }  
}
```

Trecho CSS

Se a tela tiver **no máximo 800px** de largura, as duas colunas de conteúdo (section e aside) ocupam 100% da tela e ficam uma em cima da outra.

Se a tela tiver **no máximo 500px** de largura, os links de navegação deixam de flutuar lado a lado e ficam um em cima do outro ocupando 100% da largura.



**Licenciatura  
em Computação**

Curso de Licenciatura em Computação  
Disciplina de Programação I  
Professor Tauã M. Cabreira

# AULA 13

---

CSS Layout



**INSTITUTO FEDERAL**  
Sul-rio-grandense  
Câmpus Pelotas