



CSS

APOSTILA



São Paulo - 2022

Créditos

Organização e Produção

Davi Domingues

Douglas de Oliveira

Fernando Esquírio Torres

Rafael Lopes dos Santos



É proibida a duplicação ou reprodução deste volume, no todo ou em parte, em quaisquer formas ou por quaisquer meios (eletrônicos, mecânico, gravação, fotocópia, distribuição pela internet ou outros), sem permissão prévia do Instituto da Oportunidade Social.

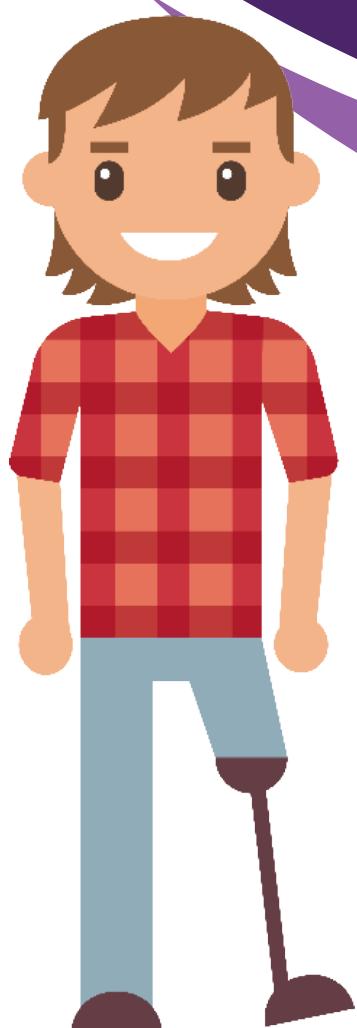
São Paulo
2022

Atenção

Em caso de dúvidas, sugestões ou reclamações. Entre em contato com a equipe de conteúdo.

educacional@ios.org.br

ESSE É SUA APOSTILA!



Ela vai te acompanhar durante todo o período do curso.

Cuide dela com carinho e responsabilidade.

Para que ela chegasse toda cheia de estilo do jeito que você está vendo, muita gente quebrou a cabeça para te entregar a melhor experiência de aprendizagem.

Ótimos estudos!

SUMÁRIO

Introdução às CSS	5
Configuração de texto	15
Class	28
ID e outros seletores.....	37
Elementos Visuais	45
Margin versus Padding.....	60
Medidas no CSS.....	70
Float display e box sizing.....	80
Pseudoclasses e Pseudoelementos	94
Media Query	109
Flexbox	123
Grid CSS Layout	137
Grid e Flexbox	180
Bootstrap Parte 01	190
Bootstrap Parte 02	208



Introdução às CSS

Os objetivos desta aula são:

- Conhecer as folhas de estilo;
- Compreender a aplicação das precedências de CSS;
- Aplicar seletores e declarações CSS.

Bons estudos!

Introdução às CSS

Cascading Style Sheets (CSS, em português folha de estilos) é uma **linguagem de estilos** usada para **descrever a apresentação** de um documento escrito em uma linguagem de marcação como o **HTML**. Enquanto o **HTML** é usado para definir a **estrutura** e o conteúdo do documento, o **CSS** é usado para configurar **texto, cor, layout da página e aplicar estilos**.

Dessa forma, a linguagem **CSS** configura a **aparência da página** e possibilita **separar o estilo de apresentação das informações de uma página web**. Com o **CSS**, é possível aplicar **estilos tipográficos** (tipo de letra, tamanho da fonte) e **instruções de layout** a uma página da web.

As **especificações CSS** são **mantidas** pelo grupo **World Wide Web Consortium (W3C)**, que também define os **padrões para HTML**. Assim como o **HTML**, a linguagem **CSS** passou por várias versões, a última das quais é **CSS Versão 3**, mais comumente conhecida como **CSS3**.

Precedência nas CSS

O design de um site geralmente não é o produto de uma única folha de estilo; é uma combinação de estilos começando na folha de estilo do navegador (**browser styles**) e, em seguida, substituída pela folha de estilo definida pelo usuário (**user-defined styles**), folhas de estilo externas (**external styles**), folhas de estilo incorporadas (**embedded styles**) e concluindo com estilos inline (**inline styles**).

Vamos examinar cada uma dessas fontes de estilo em mais detalhes.

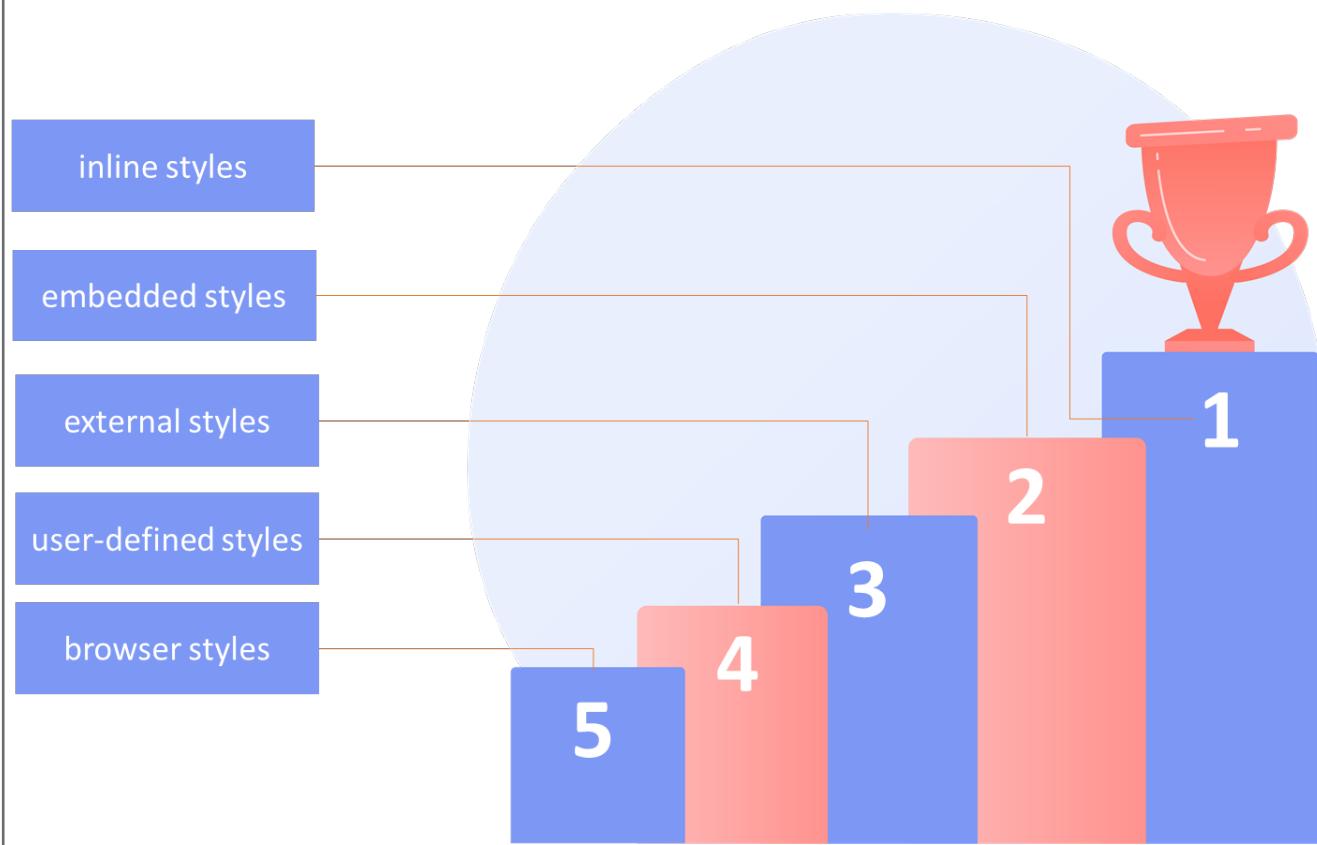


Figura 1 - Hierarquia dos estilos.

Os **primeiros** estilos a serem processados são os estilos do navegador (**browser styles**), que são os estilos criados no **próprio navegador**. Na ausência de estilos concorrentes de outras folhas de estilo, um estilo de navegador é aquele aplicado à página da web.

Os **próximos estilos** a serem processados são os estilos definidos pelo usuário (**user-defined styles**), que são estilos **definidos pelo usuário com base nas configurações que ele faz ao configurar o navegador**. Por exemplo, um usuário com deficiência visual pode alterar as configurações padrão do navegador para exibir texto com cores altamente contrastantes e um tamanho de fonte grande para facilitar a leitura. **Qualquer estilo definido pelo usuário tem precedência sobre sua contraparte de estilo de navegador.**

Os **estilos definidos pelo usuário** podem ser **substituídos por estilos externos (external styles)**, que são os estilos que o **autor do site cria e coloca dentro de um arquivo CSS**.

Acima dos **estilos externos** na hierarquia das folhas de estilo estão os **estilos incorporados (embedded styles)**, que são os estilos adicionados ao cabeçalho de um documento **HTML**. Os **estilos incorporados se aplicam apenas ao documento HTML no qual são criados** e não são acessíveis a outros documentos no site, mas **substituem quaisquer estilos** em uma folha de estilo externa.

Finalmente, **na ordem mais alta de precedência estão os estilos inline**, que são adicionados como **atributos de elemento em um documento HTML** e, portanto, **se aplicam apenas a esse elemento**. Estilos **incorporados e estilos inline não são considerados prática recomendada e seu uso deve ser evitado**, porque violam os princípios básicos do **HTML**, que é que o **HTML deve apenas descrever o conteúdo e a estrutura do documento e os estilos de design devem ser colocados fora do código HTML**.

Nesse primeiro documento, vamos trabalhar com o **estilo incorporado** e o **inline**, portanto, vamos aprofundar um pouco mais sobre esses estilos.



Estilo incorporado

Os estilos incorporados são inseridos diretamente no arquivo **HTML** como metadados dentro do elemento **<head>** na página web. A sintaxe de um estilo incorporado é:

```
<style>
    regras de estilo
</style>
```

Onde as regras de estilo são as diferentes regras que você deseja incorporar na página HTML.

Por exemplo, o seguinte estilo incorporado aplica regras de estilo para que o corpo do documento HTML atual apareça com a cor de fundo **LightGray (#D3D3D3)** e a cor do texto **Azul-Petróleo (#008080)**.

```
<!DOCTYPE html>
<html lang="pt-br">

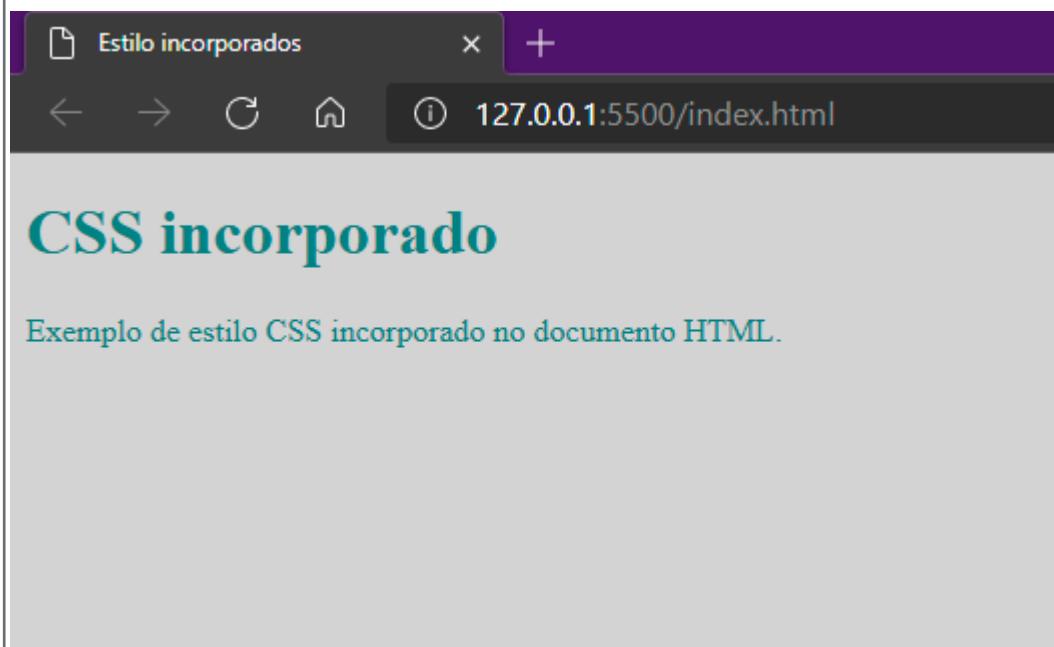
<head>
    <title>Estilo incorporado</title>
    <meta charset="utf-8">

    <style>
        body {
            background-color: #D3D3D3;
            color: #008080;
        }
    </style>

</head>

<body>
    <h1>CSS incorporado</h1>
    <p>Exemplo de estilo CSS incorporado no documento HTML.</p>
</body>

</html>
```



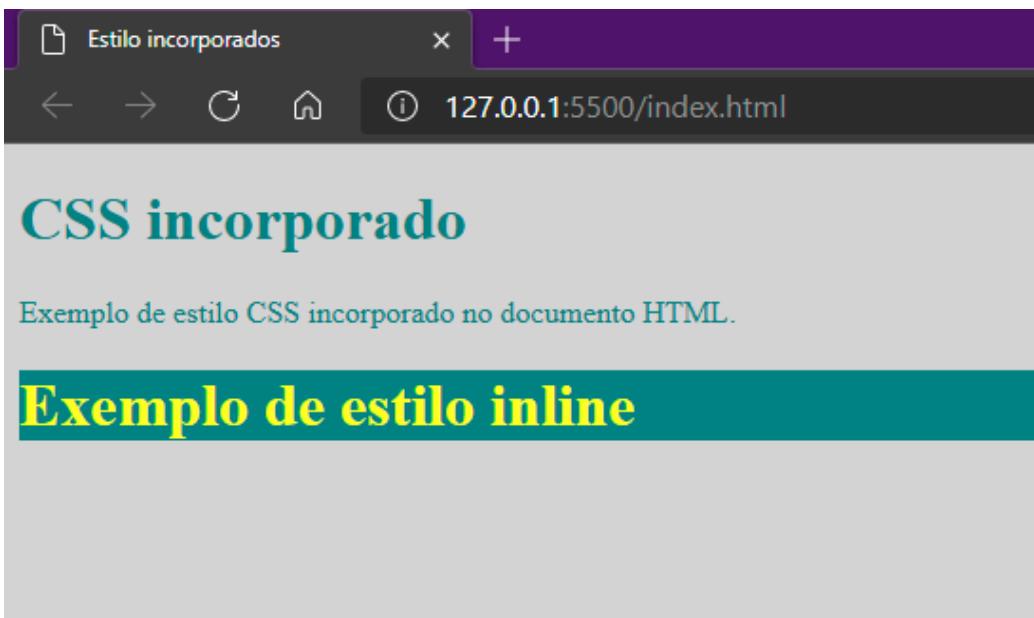
Estilo inline

O estilo **inline** é aplicado **diretamente a elementos específicos** e usado como atributo de estilo desse elemento. O valor do atributo **style** é **definido como a declaração da regra de estilo que você precisa configurar**. A sintaxe do estilo inline é:

```
<elemento style="propriedade1: valor1; propriedade2: valor2; ...">  
    conteúdo  
</elemento>
```

Onde as regras de **estilo** são **definidas** pelas **propriedades: valor e aplicadas apenas ao elemento a que elas foram atribuídas**. Por exemplo, o seguinte estilo inline aplica regras de estilo para que o esse elemento **h1** apareça com a **cor do texto Azul (blue)** e a **cor de fundo Azul-Petróleo (#008080)**.

```
<!DOCTYPE html>  
<html lang="pt-br">  
  
<head>  
    <title>Estilo incorporado e inline</title>  
    <meta charset="utf-8">  
    <style>  
        body {  
            background-color: #D3D3D3;  
            color: #008080;  
        }  
    </style>  
</head>  
  
<body>  
    <h1>CSS incorporado</h1>  
    <p>Exemplo de estilo CSS incorporado no documento HTML.</p>  
  
    <h1 style="color: yellow; background-color: #008080;">  
        Exemplo de estilo inline  
    </h1>  
</body>  
  
</html>
```



Seletores e declarações CSS

A linguagem **CSS** é definida por **regras de estilo**, que têm a função de **descrever o estilo a ser aplicado**. As regras de estilo funcionam para o CSS de forma semelhante aos elementos de marcação no HTML e **podem ser aplicadas em um elemento ou um grupo de elementos**. Essas regras são compostas de duas partes: **um seletor e uma declaração e possuem a sintaxe**:

```
seletor {
    propriedade1: valor1;
    propriedade2: valor2;
    ...
}
```

O **seletor** pode ser um **nome de elemento HTML**, um **nome de classe** ou um **nome de id**. No momento, vamos abordar os estilos aplicados a elementos de HTML. A **declaração** é formada **por pares propriedade: valores** que especificam as **propriedades de estilo e seus valores** aplicados ao elemento ou grupo de elementos. Veja o exemplo:

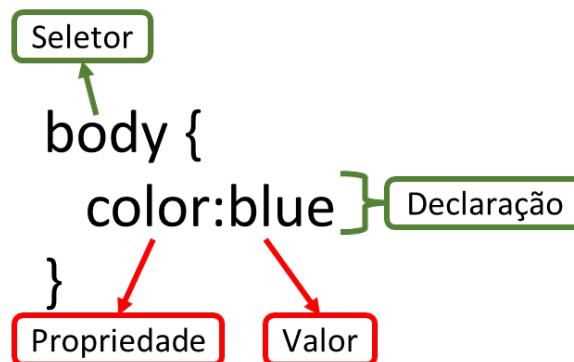


Figura 2 - Exemplo de estilo.

O estilo **CSS** mostra o elemento do HTML **body** como **seletor** da **regra de estilo** e a **declaração color: blue**, que define a **cor do texto do corpo da página como azul**.

Podemos colocar **mais de um par** **propriedade: valor** em um único **seletor** (**Figura 3**) e podemos também ter uma lista de **seletores separados por vírgulas** na mesma regra (**Figura 4**).

```
h1 {  
    color: red;  
    text-align: center;  
}
```

Figura 3

```
h1, h2 {  
    color: red;  
}
```

Figura 4

A **Figura 3** mostra uma regra de estilo com um seletor **h1** para corresponder a **todos** os elementos **h1** no documento **HTML** e essa regra possui **dois pares** **propriedade: valor**: o **primeiro par** contém a propriedade **cor e o valor vermelho** (**color: red**); o **segundo par** contém **o alinhamento de texto e o valor centralizado** (**text-align: center**). A regra de estilo da **Figura 3** instruirá o navegador a exibir todos os cabeçalhos **h1** em **vermelho** e **centralizado** na página **HTML**.

A **Figura 4** mostra uma regra de estilo com seletores **h1** e **h2 separados** por **vírgulas** e com a propriedade **cor e o valor vermelho** (**color: red**). Essa regra irá instruir o navegador a exibir os títulos **h1** e **h2** com a **cor da fonte em vermelho** na página **HTML**.

Propriedade background-color

A propriedade CSS **background-color** configura a **cor de fundo de um elemento**. Vamos testar essa propriedade para **mudar a cor de fundo do corpo de uma página web simples**. Por exemplo, a regra mostrada a seguir configura a cor de fundo amarela para o corpo do documento **HTML**.

```
body {  
    background-color: yellow;  
}
```

Propriedade color

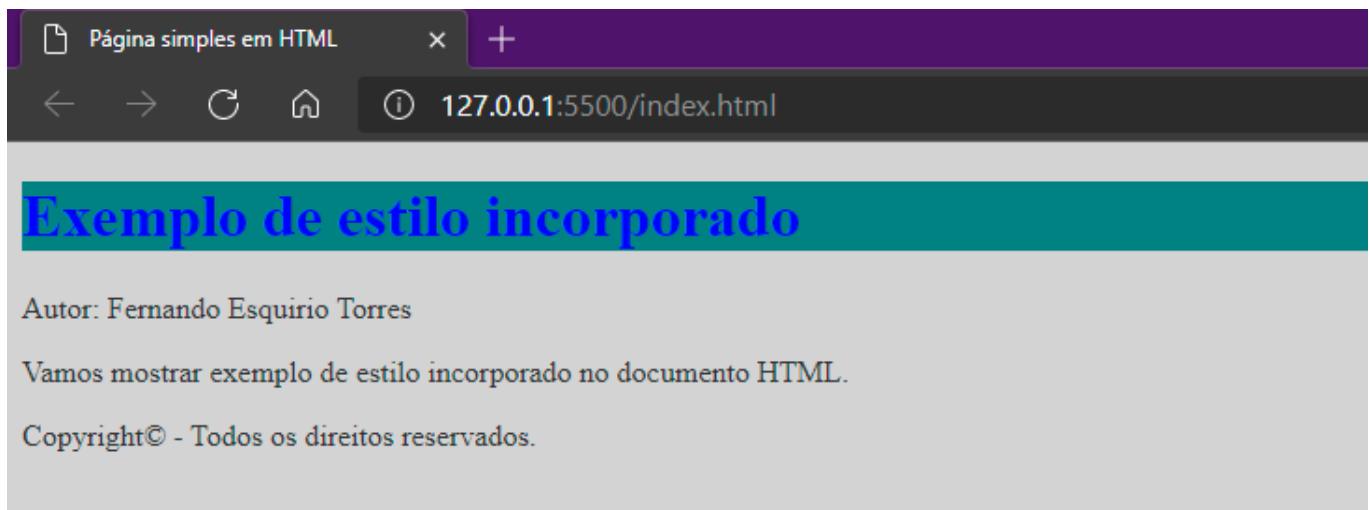
A propriedade CSS **color** configura a **cor do texto de um elemento**. Por exemplo, a regra mostrada a seguir configura a **cor do texto da página web como azul**.

```
body {  
    color: blue  
}
```

Configurando a cor do texto e de fundo com estilo inline

Vamos criar uma página web com **estilo inline aplicados** nos **elementos de marcação** de um documento HTML.

```
<!DOCTYPE html>  
<html lang="pt-br">  
  
<head>  
    <!-- conteúdo do cabeçalho -->  
    <title>Página simples em HTML</title>  
    <meta charset="utf-8" />  
</head>  
  
<body style="background-color:#D3D3D3; color:#008080;">  
  
    <header>  
        <h1 style="background-color:#008080; color:blue;">Exemplo de estilo inline</h1>  
        <p style="color:#333333">Autor: Fernando Esquírio Torres</p>  
    </header>  
    <article>  
        <p style="color:#333333">Vamos mostrar exemplos de estilos inline aplicados em elementos individuais do documento HTML.</p>  
    </article>  
    <footer>  
        <p style="color:#333333">Copyright&copy; - Todos os direitos reservados.<br />  
    </p>  
    </footer>  
  
</body>  
  
</html>
```



A screenshot of a web browser window titled "Página simples em HTML". The address bar shows the URL "127.0.0.1:5500/index.html". The main content area displays the text "Exemplo de estilo incorporado" in a large blue header, followed by "Autor: Fernando Esquirio Torres", "Vamos mostrar exemplo de estilo incorporado no documento HTML.", and "Copyright© - Todos os direitos reservados.".

Com esse exemplo foi possível ver como **aplicar regras de estilo CSS** usadas como se fossem **atributos de um elemento de marcação em HTML**. Foi possível perceber a **regra em um elemento específico que se sobrepõe a regra de um elemento geral**. Por exemplo, a **regra do elemento <h1> foi aplicada a esse elemento ao invés da regra geral configurada no corpo (elemento <body>)** no exemplo mostrado.

Outro ponto a ser destacado é se a regra de estilo é **válida apenas para o elemento a que ela está atribuída**. No caso do elemento <p>, tivemos que copiar a regra para todos os elementos para que ela fosse aplicada a todos os parágrafos do documento HTML.

Configurando a cor do texto e de fundo com estilo incorporado

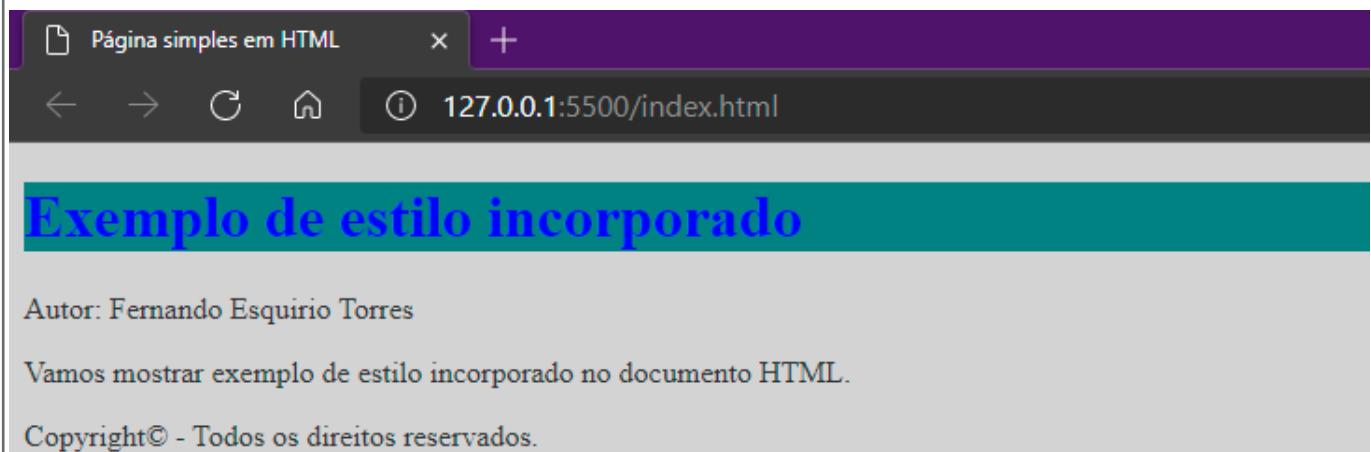
Vamos criar uma página web com **estilos incorporados** aplicados nos elementos de marcação de um documento HTML.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <!-- conteúdo do cabeçalho -->
    <title>Página simples em HTML</title>
    <meta charset="utf-8" />
    <style>
        body {
            background-color: #D3D3D3;
            color: #008080;
        }

        h1 {
            background-color: #008080;
            color: blue;
        }
    </style>
</head>
<body>
    <h1>Exemplo de estilo incorporado</h1>
    <p>Vamos mostrar exemplo de estilo incorporado no documento HTML.</p>
    <p>Copyright© - Todos os direitos reservados.</p>
</body>
</html>
```

```
p {  
    color: #333333;  
}  
</style>  
</head>  
  
<body>  
    <header>  
        <h1>Exemplo de estilo incorporado</h1>  
        <p>Autor: Fernando Esquírio Torres</p>  
    </header>  
    <article>  
        <p>Vamos mostrar exemplo de estilo incorporado no documento HTML.</p>  
    </article>  
    <footer>  
        <p>Copyright&copy; - Todos os direitos reservados. </p>  
    </footer>  
</body>  
  
</html>
```



Com esse exemplo foi possível ver como **aplicar regras de estilo CSS incorporadas** em um documento HTML. Como no **exemplo anterior**, foi possível perceber a **regra em um elemento específico que se sobrepõe a regra de um elemento geral**. Por exemplo, a **regra do elemento <h1>** foi aplicada a esse **elemento ao invés da regra geral configurada no corpo (elemento <body>)** no exemplo mostrado.

Outro ponto a ser destacado é que **a regra de estilo agora é válida para todos os elementos que aparecem no documento HTML**. Por exemplo, a regra de **estilo do elemento <p>** foi aplicada a **todos os parágrafos** da página web. De certo modo, **o estilo incorporado deixa o código mais limpo e organizado em relação ao estilo inline**. Além disso, o estilo incorporado é mais prático, pois é necessário configurar apenas uma regra a um determinado elemento de HTML para ela ser aplicada todas as vezes que esse elemento aparecer no documento HTML.



Configuração de texto

Os objetivos desta aula são:

- Conhecer algumas propriedades da CSS;
- Compreender a aplicação das folhas de estilo na formatação de páginas HTML.

Bons estudos!

Propriedades introdutórias do CSS

Como foi visto no capítulo anterior, o **CSS** é uma **linguagem de estilos** usada para **formatar e configurar** as propriedades das informações exibidas em um documento HTML. Existem diversas **propriedades no CSS** e as mais comuns são mostradas na tabela abaixo:

Propriedade	Descrição	Valores
background-color	Configura a cor de fundo de um elemento	Qualquer cor válida
color	Cor do primeiro plano de um elemento, por exemplo a cor de um texto	Qualquer cor válida
font-family	Nome de uma fonte ou de uma família de fonte	Quaisquer fontes válidas tais como serif, sans-serif, fantasym monospace, verdana, cursive, calibri, etc.
font-size	Tamanho da fonte	Variados; pode ser um valor numérico em pt (tamanhos de ponto de fonte padrão) ou px (pixels) ou a unidade em (que corresponde à largura do M maiúsculo da fonte atual); Pode ser, também, uma porcentagem numérica; e os valores disponíveis de texto xx-small, x-small, small, medium, large, x-large e xx-large
font-style	Estilo da fonte	Normal, itálica ou obliqua
font-weight	Negrito ou grossura da fonte	Variados; os valores de texto normal, negrito e mais claro ou escuro. Também, podem ser os valores numéricos 100, 200, 300, 400, 500, 600, 700, 800 e 900.
letter-spacing	Espaço entre caracteres	Um valor numérico (em px ou em) ou normal (configuração padrão)
line-height	O espaçamento permitido para a linha de texto	É mais comum usar um valor em porcentagem, por exemplo, um valor de 200% corresponderia a um espaçamento duplo.
margin	Notação abreviada para configurar a margem em torno de um elemento. Com essa propriedade você pode configurar as quatro margens: <code>margin: top right bottom left</code>	Um valor numérico (px ou em), por exemplo: <code>body { margin: 25px 50px 75px 100px; }</code> Definirá as margens da página Margem superior (top margin) será 25px Margem direita (right margin) será 50px Margem inferior (bottom margin) será 75px Margem esquerda (left margin) será 100px. Ao eliminar a margem, não use a unidade px ou em, por exemplo: <code>body {margem: 0}</code>
margin-top	Configura o espaço na margem superior do elemento.	Deve ser um valor numérico (px ou em), auto ou 0
margin-right	Configura o espaço na margem direita do elemento.	Deve ser um valor numérico (px ou em), auto ou 0

margin-bottom	Configura o espaço na margem inferior do elemento.	Deve ser um valor numérico (px ou em), auto ou 0
margin-left	Configura o espaço na margem esquerda do elemento.	Deve ser um valor numérico (px ou em), auto ou 0
text-align	Alinhamento do texto	center (centralizado), justify (justificado), left (esquerda) ou right (direita)
text-decoration	Determina se o texto está sublinhado; este estilo é mais frequentemente aplicado a hiperlinks	O valor “none” fará com que um hiperlink não seja sublinhado em um navegador que normalmente é processado dessa maneira
text-indent	Configura o recuo da primeira linha do texto	Valor numérico (px ou em) ou porcentagem
text-shadow	Configura uma sombra projetada no texto exibido em um elemento. Esta propriedade CSS3 não é compatível com todos os navegadores	De dois a quatro valores numéricos (px ou em) para indicar deslocamento horizontal, deslocamento vertical, raio de desfoque (opcional) e distância de propagação (opcional) e um valor de cor válido.
text-transform	Configura a capitalização (Caps lock ou Caps letter) do texto	none (padrão), capitalize (primeira letra de cada palavra maiúscula), uppercase (todos os caracteres maiúsculos) ou lowercase (todos os caracteres minúsculos)
white-space	Configura a exibição de espaços em branco	normal (default), nowrap, pre, pre-line, pre-wrap width
width	A largura do conteúdo de um elemento	Um valor numérico (px ou em), porcentagem numérica ou auto (padrão)
word-spacing	O espaço entre as palavras	Um valor numérico (px ou em) ou normal (padrão)

Usando Cores em páginas web

No capítulo anterior, vimos como configurar as **cores de plano de fundo e do conteúdo (texto)** de um elemento exibido em um documento HTML. Mas **como podemos descobrir um valor válido e cor?** Primeiramente, vamos entender **como funcionam as cores em um monitor usado no seu computador.**

Os monitores exibem uma cor como uma combinação de diferentes intensidades de **vermelho (red)**, **verde (green)** e **azul (blue)**, esse conceito é conhecido como padrão de cores **RGB (Red, Green, Blue)**.

A intensidade **RGB** é o valor numérico de **0** a **255**. Cada cor **RGB** tem três valores: **um valor para o vermelho (red)**, **um para verde (green)** e **um para o azul (blue)**. Esses valores sempre são listados na mesma ordem: **primeiro o valor de vermelho (red)**, **segundo o valor de verde (green)** e **terceiro o valor de azul (blue)**.

As cores em **RGB** de um elemento podem ser representadas em **dois sistemas numéricicos**:

1. **Sistema de numeração decimal**, que é o sistema que utiliza símbolos de **0, 1, 2, 3, 4, 5, 6, 7, 8 e 9** para representar os seus valores numéricos. Esse sistema é que estamos mais acostumados a usar no nosso dia a dia.
2. **Sistema de numeração hexadecimal**, que é o sistema que utiliza os símbolos **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F** para representar os seus valores numéricos. **Esse é o sistema mais comum usado para configurar cores em HTML e CSS.**

A imagem abaixo mostra as **duas formas** de **configuração da cor vermelha** nas duas representações **decimal** e **hexadecimal**. Na representação **decimal**, é **necessária a marcação `rgb`** e entre parênteses três valores: **o primeiro valor 255 define o valor total para a intensidade da cor vermelha, o segundo valor 0 define valor mínimo para a intensidade da cor verde e o terceiro valor 0 define valor mínimo para a intensidade da cor azul.**

Na representação **hexadecimal**, é necessário o símbolo **#** e **6 dígitos em hexadecimal**: os **dois primeiros dígitos FF definem o valor total para a intensidade da cor vermelha, dois dígitos seguintes 00 definem o valor mínimo para a intensidade da cor verde e os dois últimos dígitos 00 definem o valor mínimo para a intensidade da cor azul.**

Representação em decimal:

```
p { color: rgb(255, 0, 0); }
```

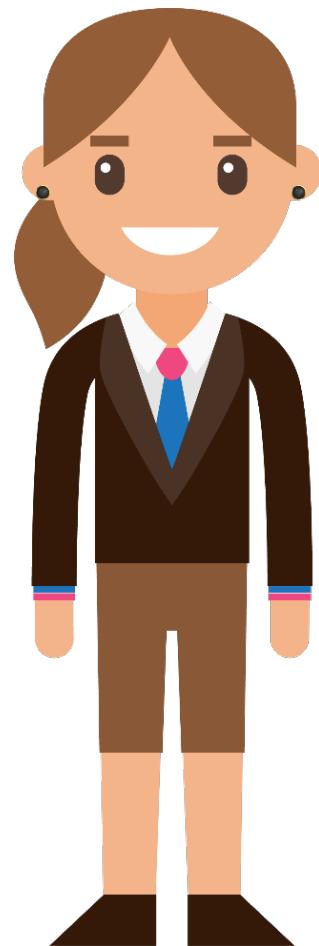


Representação em hexadecimal:

```
p { color: #FF0000; }
```



Sistema decimal e hexadecimal para configuração das cores.



Sintaxe de Cores no CSS

A sintaxe do CSS permite você configurar cores de formas variadas:

- Nome da cor
- Valor em hexadecimal
- Valor da cor abreviada hexadecimal
- Valor decimal (tripla RGB)
- Valor da cor em notação HSL (Matiz, Saturação e Intensidade), que foi disponibilizada no CSS3 e para saber mais acesse:
<http://www.w3.org/TR/css3-color/#hsl-color>

A tabela abaixo mostra exemplos de diferentes maneiras de **configurar a cor vermelha** para o elemento de marcação **<p>** no **CSS**.

Sintaxe CSS	Forma de configuração
p { color: red; }	Nome da cor
p { color: #FF0000; }	Valor hexadecimal
p { color: #F00; }	Hexadecimal abreviado (um caractere para cada par hexadecimal; usado apenas com cores seguras para a web)
p { color: rgb(255,0,0); }	Valor da cor em decimal (valor RGB)
p { color: hsl(0, 100%, 50%); }	Valor da cor em HSL

Para ajudar você no desenvolvimento e na escolha de cores da sua página HTML, você pode encontrar diversos códigos em decimal ou hexadecimal nos sites:

- Gráfico de Cores HTML: <https://html-color-codes.info/Codigos-de-Cores-HTML/>
- Tabela de Cores – WEB: https://www.flextool.com.br/tabela_cores.html

Propriedade font-family

A propriedade **font-family** configura o **tipo da fonte mostrada no texto apresentado**. O navegador mostra um texto usando alguma fonte que está instalada no seu computador. Quando a fonte escolhida não estiver instalada no computador do usuário, o navegador irá escolher uma fonte padrão para exibir o texto.

Em geral, as fontes **Verdana**, **Tahoma** e **Georgia** foram projetadas para **exibir um texto em um monitor**, o mais comum é usar fontes com **serif** (como **Georgia** ou **Times New Roman**) para os títulos (**cabeçalhos**) e **sans-serif** (como **Verdana** ou **Arial**) para os **outros conteúdos textuais da página**.

A sintaxe do CSS para a propriedade font-family é:

```
font-family: family-name|generic-family|initial|inherit;
```

Onde:

- **family-name|generic-family:** é uma lista priorizada com nomes da família de fontes e/ou nome da família genérica que o desenvolvedor web deseja utilizar;
- **initial:** Define esta propriedade com seu valor padrão;
- **inherit:** Herda essa propriedade de seu elemento pai.

Exemplos:

```
h2, h3 {  
    font-family: Georgia, "Times New Roman", serif;  
}  
  
body {  
    font-family: Verdana, Arial, sans-serif;  
}
```

A fonte **Times New Roman** teve que ser escrita **entre aspas** para ser reconhecida como uma **única fonte**, pois, como ela tem mais de uma palavra no nome, **se ela fosse escrita sem aspas o navegador interpretaria como três nomes de fontes**. Nesse caso, tanto faz usar aspas simples ou duplas.

Configurando um texto com o CSS configurado em um arquivo externo

Vamos aprender a configurar o **estilo CSS** em um **arquivo externo**, pois desse modo podemos **separar a configuração de estilo do código em HTML** em uma página web. Além disso, utilizar um arquivo externo para o CSS **possibilita reaproveitar a configuração para diversas páginas do seu código e é a forma mais comum de usar o CSS no desenvolvimento de programação web**.

Vimos anteriormente que o elemento **<link>** é usado para **vincular um arquivo externo a um documento HTML**, então vamos utilizar essa marcação do **HTML** para **linkar** o nosso arquivo **CSS** criado em nossa página web.

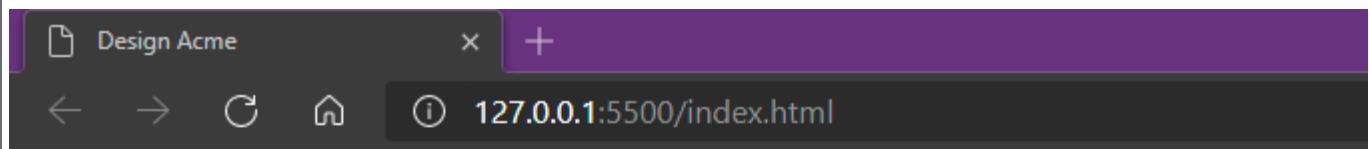
Vamos criar um **arquivo HTML** e o **arquivo de estilo CSS** e **visualizar como configurar uma fonte do texto exibida na página web**.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <title>Design Acme</title>  
    <meta charset="utf-8" />  
</head>  
<body>  
    <header>  
        <h1>Empresa de Design Acme</h1>  
    </header>  
    <article>  
        <h2>Novas mídias e novo design</h2>  
        <p>A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web.<br />  
            Nós oferecemos os seguintes serviços:</p>  
        <ul>
```

```

<li>Desenvolvimento de Websites</li>
<li>Animação Interativa</li>
<li>Soluções de E-Commerce</li>
<li>Estudos de Usabilidade</li>
<li>Otimização de Máquinas de Busca</li>
</ul>
<h2>Atendendo às suas necessidades de negócios</h2>
<p> Nossos designers especializados são criativos e prontos para trabalhar com você. <br /> Aproveite o crescimento da web!</p>
</article>
<footer>Copyright © 2020 Coloque seu nome aqui!</footer>
</body>
</html>

```



Empresa de Design Acme

Novas mídias e novo design

A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web. Nós oferecemos os seguintes serviços:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Nossos designers especializados são criativos e prontos para trabalhar com você.
Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!

Agora crie um **novo arquivo na mesma pasta do arquivo HTML** construído anteriormente e insira o seguinte **código CSS**.

```

@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
}

```

```

    font-family: Arial, Verdana, sans-serif;
}
h1 {
    background-color: #191970;
    color: #e6e6fa;
    font-family: Georgia, "Times New Roman", serif;
}
h2 {
    background-color: #aeaed4;
    font-family: Georgia, "Times New Roman", serif;
}
footer {
    font-family: 'Courier New', Courier, monospace;
}

```

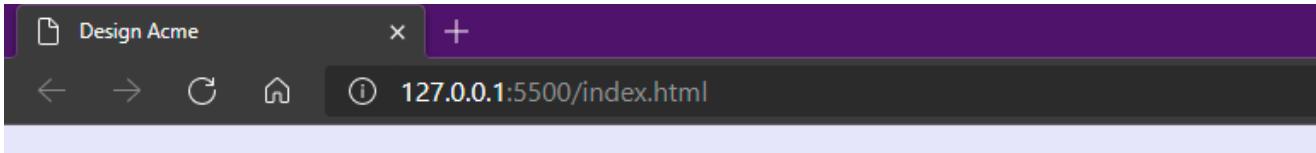
Salve o arquivo de **estilo CSS**, acessando o **Menu Arquivo** e a opção **Salvar como...** ou use as teclas de atalho **Ctrl+Shift+S**. Coloque o nome de: **estilo.css**

Volte ao seu arquivo **html** e no elemento **<head>** insira a marcação **<link>** para vincular o **arquivo CSS** ao seu documento **HTML**.

```

<head>
    <title>Design Acme</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="estilo.css">
</head>

```

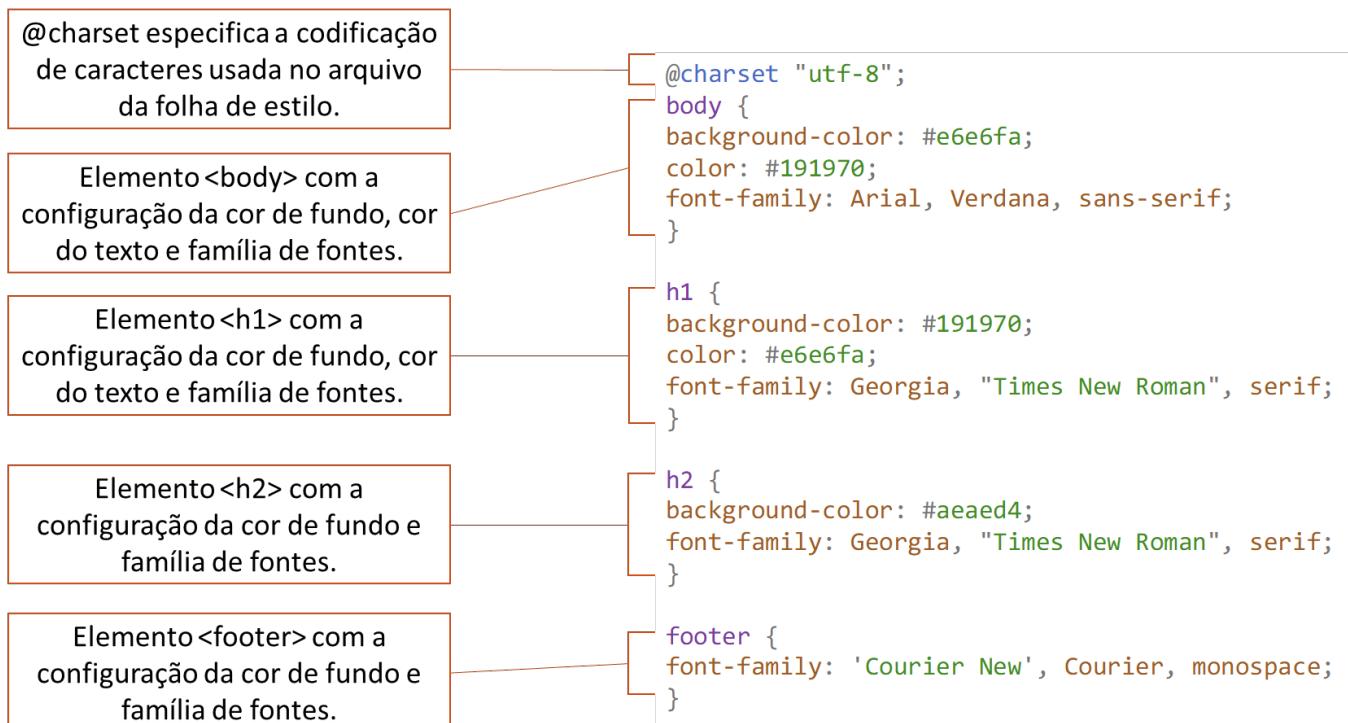


The screenshot shows a web browser window with a purple header bar. The title bar says 'Design Acme'. Below it, the address bar shows the URL '127.0.0.1:5500/index.html'. The main content area displays the 'Design Acme' website. The header has a dark blue background with white text. It features a large 'Empresa de Design Acme' logo. Below the header is a light blue navigation bar with the text 'Novas mídias e novo design'. The main content area has a white background with dark blue text. It lists services like 'Desenvolvimento de Websites', 'Animação Interativa', etc. There's also a section titled 'Atendendo às suas necessidades de negócios' with a message about creative designers. At the bottom, there's a copyright notice: 'Copyright © 2020 Coloque seu nome aqui!'

Identificando os estilos no arquivo CSS

O elemento **<body>** possui as configurações de **cor de fundo (background-color: #e6e6fa;)**, **cor do texto (color: #191970;)** e família da **fonte (font-family: Arial, Verdana, sans-serif;)**. O elemento **<h1>** possui as configurações de **cor de fundo (background-color: #191970;)**, **cor do texto (color: #e6e6fa;)** e **família da fonte (font-family: Georgia, "Times New Roman", serif;)**.

O elemento **<h2>** possui as configurações de **cor de fundo (background-color: #aeaed4;)** e família da **fonte (font-family: Georgia, "Times New Roman", serif;)**. O elemento **<footer>** possui as configurações família da **fonte (font-family: 'Courier New', Courier, monospace;)**.



Observe que quando **um elemento não possui alguma propriedade no estilo CSS configurada, ele vai utilizar a configuração do elemento hierarquicamente acima dele**, denominado **elemento pai**.

Por exemplo, o elemento **<h2>** **não possui a configuração** da propriedade **cor da fonte**, portanto ele **utilizará a configuração do elemento <body>**, que é o **primeiro elemento pai** da **marcação <h2>**.

O mesmo acontece com as configurações da cor de fundo e da cor do texto do elemento **<footer>** e com as configurações da fonte, da cor de fundo e da cor do texto do elemento **<p>** no documento HTML.

font-size, font-weight, font-style, line-height, text-align, text- shadow e text-indent

A propriedade **font-size** determina o **tamanho** da **fonte do texto**. Essas podem ser divididas em duas categorias, as **absolutas** e as **relativas**.

As **medidas absolutas** não tem seu tamanho relacionado com outro elemento.

Medidas absolutas	Valores
cm	Centímetros
in	Polegadas (1 polegada é = 2.54cm)
px	Pixel (1pixel é = 1/90 in)

As **unidades relativas**, elas se **adaptam conforme o elemento com o qual elas estão relacionadas**.

Medidas relativas	Valores
Valor do texto	Valores
em	xx-small, x-small, small, medium (padrão), large, x-large, xx-large
em	Valor numérico com unidade, seu tamanho está relacionado ao tamanho do elemento superior . Por exemplo, se o elemento superior tiver um tamanho de 16px, e no elemento inferior for definido o tamanho de 2em, o elemento inferior terá 32px, ou seja, o dobro.
rem	Valor numérico com unidade, seu tamanho está relacionado ao elemento root , ou seja, o tamanho padrão da fonte de um navegador é de 16px, se colocarmos o tamanho da fonte de 0.5em estamos definindo que o elemento tem 8px. Lembrando que é possível alterar o tamanho padrão do root no CSS.
Valor em porcentagem	Valor numérico com porcentagem, seu tamanho está relacionado ao tamanho do elemento superior . Por exemplo, se o elemento superior tiver um tamanho de 16px, e no elemento inferior for definido o tamanho de 50%, o elemento inferior terá 8px, ou seja, a metade.

Veremos mais sobre medidas nos próximos capítulos.

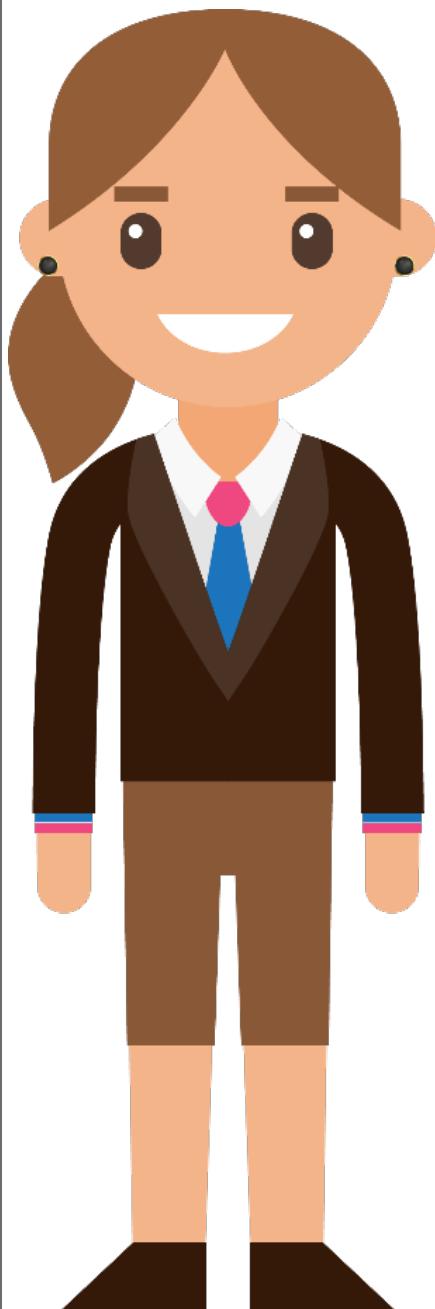
A propriedade **font-weight** configura a **grossura do texto**, por exemplo: **font-weight: bold** é **similar ao elemento no HTML** e configura o texto para ser **exibido em destaque**.

A propriedade **font-style** pode ser usada para exibir um **texto em itálico**. Ela é semelhante ao elemento ** no HTML**.

A propriedade **line-height** modifica a altura padrão da linha do texto e é comumente usada com valores percentuais. Por exemplo, **line-height: 200%** configura o **dobro do espaço para exibir o texto**.

A propriedade **text-align** configura o **alinhamento do texto exibido no documento HTML**. Por padrão, **todo texto em um documento HTML é exibido com alinhamento à esquerda**. Com a **propriedade text-align você pode alterar o alinhamento dele**.

A propriedade **text-shadow** adiciona **profundidade e dimensão a um texto exibido na página web**. Os navegadores mais atuais oferecem suporte para essa propriedade. **Para configurar uma sombra no texto, você deve especificar os valores para o deslocamento horizontal da sombra, deslocamento vertical da sombra, raio do borrado (opcional) e cor**:



- **Horizontal offset (Deslocamento horizontal)**: um valor numérico positivo em pixel para configurar uma sombra à direita do texto e um valor numérico negativo em pixel para configurar uma sombra à esquerda do texto.
- **Vertical offset (Deslocamento vertical)**: um valor numérico positivo em pixel para configurar uma sombra abaixo do texto e um valor numérico negativo em pixel para configurar uma sombra acima do texto.
- **Blur radius (Raio do borrado) (opcional)**: um valor numérico de pixel. Se omitido, o padrão é o valor 0, que configura uma sombra nítida. Valores maiores do que 0 (zero) configuram a sombra mais borrada.
- **Valor da cor**: configura um valor de cor válido para a sombra.

Por exemplo, o código abaixo configura uma sombra cinza escura com deslocamento horizontal de 3 pixels, deslocamento vertical de 2 pixels e raio do borrado de 5 pixels;

```
text-shadow: 3px 2px 5px #666;
```

Por fim, a propriedade **text-indent** configura a **indentação (o recuo)** da **primeira linha do texto exibido em determinado elemento**. O valor pode ser numérico com as devidas **unidades (px, pt ou em) ou uma porcentagem**.

Abra o arquivo **estilo.css criado anteriormente** no **VS Code**. Vamos **atualizá-lo** com outras configurações de estilos.

```
@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
    font-family: Arial, Helvetica, sans-serif;
}

h1 {
    background-color: #191970;
    color: #e6e6fa;
    font-family: Georgia, 'Times New Roman', Times, serif;
    line-height: 200%;
    text-shadow: 3px 3px 5px #cccccc;
}

h2 {
    background-color: #aeaed4;
    font-family: Georgia, 'Times New Roman', Times, serif;
    text-align: center;
}

p {
    font-size: 0.90em;
    text-indent: 3em;
}

ul {
    font-weight: bold;
}

footer {
    font-family: 'Courier New', Courier, monospace;
    color: #333333;
    font-size: 0.75em;
    font-style: italic;
}
```

Design Acme x +

127.0.0.1:5500/index.html

Empresa de Design Acme

Novas mídias e novo design

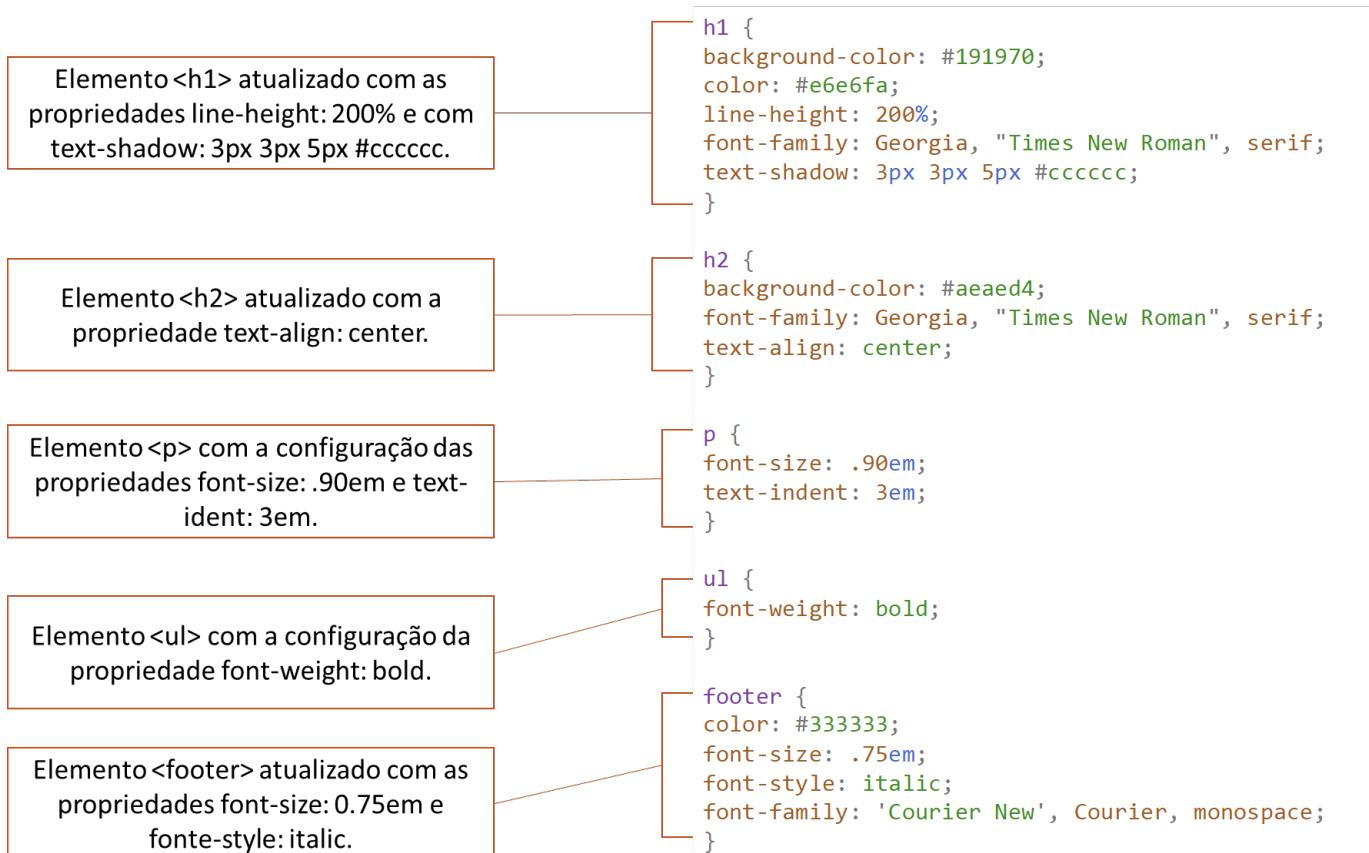
A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web.
Nós oferecemos os seguintes serviços:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Nossos designers especializados são criativos e prontos para trabalhar com você.
Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!





Os objetivos desta aula são:

- Compreender o uso de classes nas folhas de estilo;
- Aplicar uma ou mais formatações utilizando classes nas CSS.

Bons estudos!

Seletor de classe (class selector)

O **seletor de classe** é usado no CSS, quando queremos aplicar um estilo em um grupo de **elementos HTML** de página web e não em um único elemento como fizemos anteriormente. Com isso, é possível **configurar parte da informação exibida em um documento HTML**. A sintaxe para criar uma classe no CSS é:

```
.nome_da_classe {
    propriedade01: "valor01";
    propriedade02: "valor02";
}
```

Deve-se colocar um **ponto (.)** antes do nome para indicar que é uma classe na folha de estilo. Esse tipo de declaração de classe no CSS pode ser aplicado em qualquer elemento na sua página HTML. Vamos ver o exemplo abaixo, que cria uma classe com o nome cor_texto com a propriedade color configurada para a cor vermelho forte (#C70000).

```
.cor_texto {
    color: #C70000;
}
```

Vamos criar **um arquivo HTML** e o arquivo de **estilo CSS** e visualizar como configurar uma fonte do texto exibida na página web.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <title>Design Acme</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="estilo.css" />
</head>

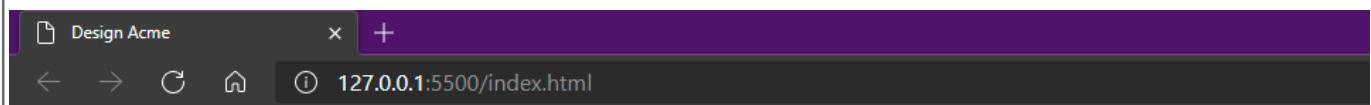
<body>
    <header>
        <h1>Empresa de Design Acme</h1>
    </header>
    <article>
        <h2>Novas mídias e novo design</h2>
        <p>A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web.<br />
            Nós oferecemos os seguintes serviços:</p>
        <ul>
            <li>Desenvolvimento de Websites</li>
            <li>Animação Interativa</li>
        </ul>
    </article>
</body>
```

```

<li>Soluções de E-Commerce</li>
<li class="cor_texto">Estudos de Usabilidade</li>
<li class="cor_texto">Otimização de Máquinas de Busca</li>
</ul>
<h2>Atendendo às suas necessidades de negócios</h2>
<p> Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.</p>
<p> Nossos designers especializados são criativos e prontos para trabalhar com você. <br /> Aproveite o crescimento da web!</p>
</article>
<footer>Copyright © 2020 Coloque seu nome aqui!</footer>
</body>

</html>

```



Empresa de Design Acme

Novas mídias e novo design

A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web. Nós oferecemos os seguintes serviços:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.

Nossos designers especializados são criativos e prontos para trabalhar com você. Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!

Agora crie um arquivo CSS na mesma pasta do arquivo HTML construído anteriormente e insira o seguinte código CSS. Observe, a criação da **classe** de seletor com o nome **cor_texto**.

```
@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
    font-family: Arial, Verdana, sans-serif;
}

h1 {
    background-color: #191970;
    color: #e6e6fa;
    line-height: 200%;
    font-family: Georgia, "Times New Roman", serif;
    text-shadow: 3px 3px 5px #cccccc;
}

h2 {
    background-color: #aeaed4;
    font-family: Georgia, "Times New Roman", serif;
    text-align: center;
}

p {
    font-size: .90em;
    text-indent: 3em;
}

ul {
    font-weight: bold;
    color: orange;
}

footer {
    color: #333333;
    font-size: .75em;
    font-style: italic;
    font-family: 'Courier New', Courier, monospace;
}

.cor_texto {
    color: #C70000;
}
```

Design Acme +
 ↵ → ⌂ ⌂ 127.0.0.1:5500/index.html

Empresa de Design Acme

Novas mídias e novo design

A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web. Nós oferecemos os seguintes serviços:

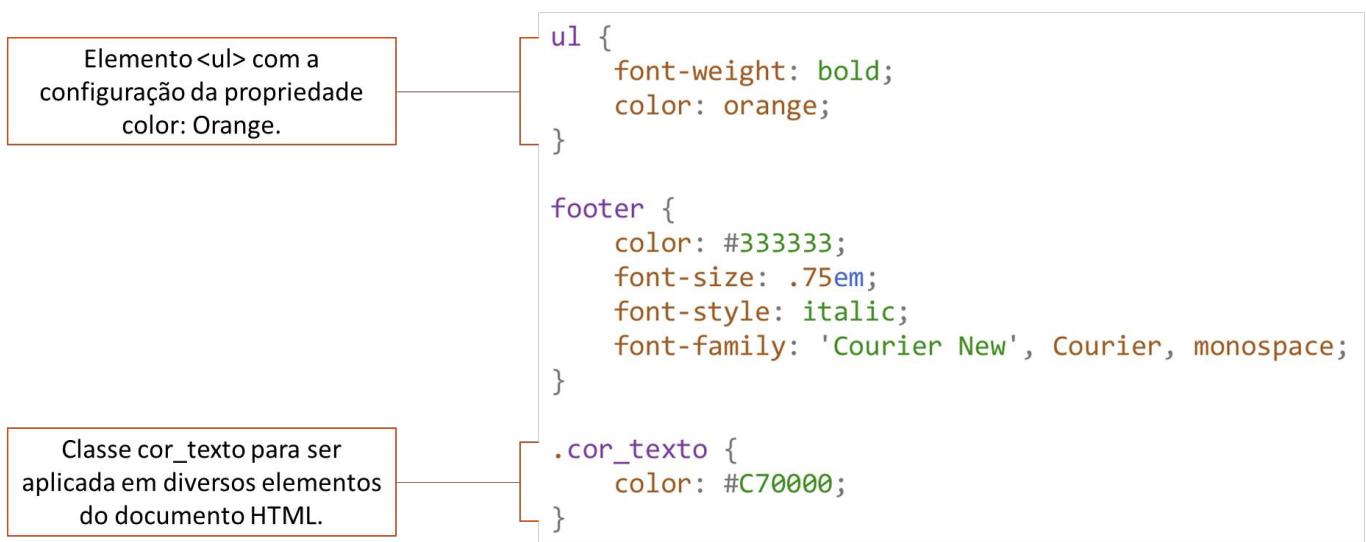
- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.

Nossos designers especializados são criativos e prontos para trabalhar com você. Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!



Observe que os **itens da lista sem a classe cor_texto utilizam a configuração de cor alaranjada** do elemento na folha de estilo <body> da lista não ordenada. Por sua vez, os **itens da lista não ordenada com a classe cor_texto estão destacados em vermelho**.

Classe em um elemento HTML

Podemos criar **uma classe associada a apenas um elemento do HTML** e, assim, **apenas as informações do determinado elemento configuradas com essa classe apresentarão o estilo definido**. Por exemplo:

```
p.fundo {
    background-color: yellow;
}
```

Nesse exemplo, o elemento **parágrafo <p>** possui uma **classe** com o nome **fundo** que **define a propriedade background para a cor amarelo (yellow)**. Esse tipo de declaração de classe é exclusivo ao elemento associado e não pode ser inserida em outro elemento da sua página HTML. Então, levando em consideração o exemplo mostrado, o atributo class="fundo" em qualquer outro elemento diferente do <p> o estilo não será aplicado.

Atualize o código do arquivo **estilo.css** para ficar idêntico ao mostrado abaixo.

```
@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
    font-family: Arial, Verdana, sans-serif;
}
h1 {
    background-color: #191970;
    color: #e6e6fa;
    line-height: 200%;
    font-family: Georgia, "Times New Roman", serif;
    text-shadow: 3px 3px 5px #cccccc;
}
h2 {
    background-color: #aeaed4;
    font-family: Georgia, "Times New Roman", serif;
    text-align: center;
}
p {
    font-size: .90em;
    text-indent: 3em;
    background-color: royalblue;
}
p.fundo {
    background-color: sandybrown;
}
ul {
    font-weight: bold;
    color: orange;
}
footer {
    color: #333333;
    font-size: .75em;
    font-style: italic;
    font-family: 'Courier New', Courier, monospace;
}
.cor_texto {
    color: #C70000;
}
```

Abra o arquivo **index.html** e **atualize** o código como mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <title>Design Acme</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="02.estilo.css" />
</head>

<body>
    <header>
        <h1>Empresa de Design Acme</h1>
    </header>
    <article>
        <h2>Novas mídias e novo design</h2>
        <p>A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web.<br />
            Nós oferecemos os seguintes serviços:</p>
        <ul>
            <li>Desenvolvimento de Websites</li>
            <li>Animação Interativa</li>
            <li>Soluções de E-Commerce</li>
            <li class="cor_texto">Estudos de Usabilidade</li>
            <li class="cor_texto">Otimização de Máquinas de Busca</li>
        </ul>
        <h2>Atendendo às suas necessidades de negócios</h2>
        <p class="fundo"> Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.</p>
        <p> Nossos designers especializados são criativos e prontos para trabalhar com você. <br /> Aproveite o crescimento da web!</p>
    </article>
    <footer>Copyright © 2020 Coloque seu nome aqui!</footer>
</body>

</html>
```



The screenshot shows a browser window titled 'Design Acme'. The address bar displays '127.0.0.1:5500/index.html'. The page content includes a header 'Empresa de Design Acme', a section titled 'Novas mídias e novo design', a list of services, a section titled 'Atendendo às suas necessidades de negócios', and a copyright notice.

Services Offered:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

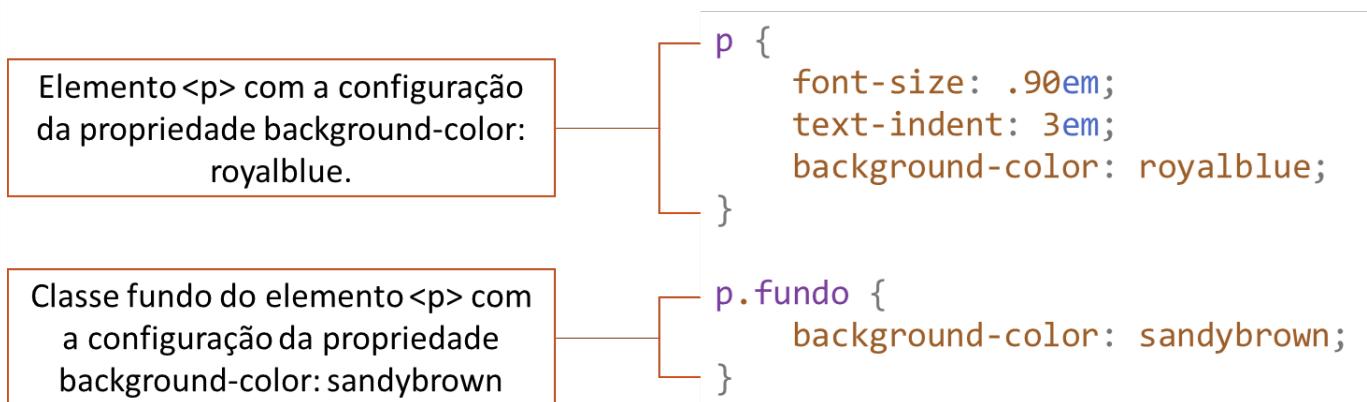
Text from the page:

A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web. Nós oferecemos os seguintes serviços:

Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.

Nossos designers especializados são criativos e prontos para trabalhar com você. Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!



Observe que os **parágrafos sem o atributo class** tem a **cor de fundo Azul Royal (royalblue)** configurada no seletor de estilo **p** e os **parágrafos** com o atributo **class="fundo"** tem a **cor de fundo Marrom Areia (sandybrown)** configurada na classe **fundo** do elemento **p**. Vale ressaltar que **se você inserir o atributo class="fundo" em qualquer outro elemento diferente do <p> o estilo não será aplicado**.

Aplicando mais de uma classe em um único elemento

Você pode aplicar **mais de uma classe em um único elemento**. Vamos fazer isso na prática: Abra o arquivo **index.html** e **atualize** o código como mostrado abaixo.

```
<p class="fundo cor_texto"> Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.</p>
```



Duas classes aplicadas a um elemento HTML. Você deve escrever os nomes das classes separadas por espaço.

```
<p class="fundo cor_texto"> Criar um site possibilita que sua empresa tenha maior credibilidade perante o seu público-alvo, garantindo que os clientes notem o seu esforço em possuir um relacionamento no ambiente virtual.</p>
```

Observe que o parágrafo com o atributo a **class="fundo cor_texto"** tem a cor de fundo **Marrom Areia (sandybrown)** configurada na classe **fundo** do **elemento p** e a cor do texto **vermelho** configurada na classe geral **cor_texto**.



ID e outros seletores

Os objetivos desta aula são:

- Diferenciar o uso de classes e id;
- Compreender o uso de id nas folhas de estilo;
- Aplicar formatações utilizando id nas CSS e o seletor *.

Bons estudos!

Seletor de id (id selector)

O seletor de id é usado no CSS para identificar e aplicar uma regra CSS exclusivamente a uma única área em uma página web. Ao contrário de um seletor de classe que pode ser aplicado várias vezes em uma página web, um id só pode ser aplicado uma vez em uma página.

Lembrando que, o atributo global id define um identificador exclusivo (ID) que deve ser único por todo o documento. Seu objetivo é identificar o elemento ao navegar por âncoras (usando um identificador de fragmento), quando utilizar scripts ou estilizando (com CSS). A sintaxe para criar um seletor de id no CSS é:

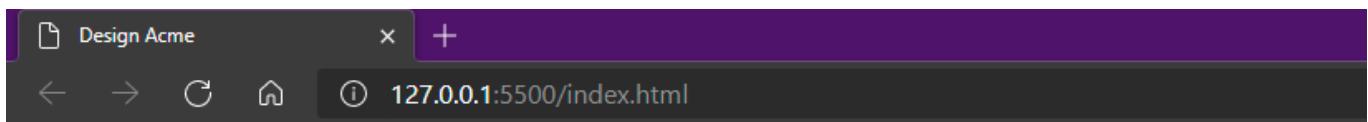
```
#valor_id {  
    propriedade01: "valor01";  
    propriedade02: "valor02";  
    ...  
}
```

Deve-se colocar o símbolo # antes do nome para indicar que é um seletor de id na folha de estilo. Um nome de id pode conter letras, números, hifens e sublinhados. Os nomes de identificação não podem conter espaços. Vamos ver o exemplo abaixo, que cria um seletor de ID primeiro_nome com a propriedade color de fundo configurada para a cor amarelo (yellow).

```
#primeiro_nome {  
    background-color: yellow;  
}
```

Vamos criar um arquivo HTML e o arquivo de estilo CSS na sua página web.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <title>Design Acme</title>  
    <meta charset="utf-8" />  
    <link rel="stylesheet" href="estilo.css" />  
</head>  
  
<body>  
    <header>  
        <h1>Empresa de Design Acme</h1>  
    </header>  
    <article>  
        <h2>Equipe:</h2>  
        <p id="primeiro_nome">Charlie</p>  
        <p id="sobrenome">Brown</p>  
    </article>  
    <footer>Copyright © 2020 Coloque seu nome aqui!</footer>  
</body>  
</html>
```



Empresa de Design Acme

Equipe:

Charlie

Brown

Copyright © 2020 Coloque seu nome aqui!

```
@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
    font-family: Arial, Verdana, sans-serif;
}

#primeiro_nome {
    background-color: yellow;
}

p#sobrenome {
    background-color: grey;
}

body {
    color: blue
}
```



Seletor id #primeiro_nome pode ser aplicado uma vez em qualquer elemento do HTML.

```
#primeiro_nome {
    background-color: yellow;
}
```

Seletor id p#sobrenome pode ser aplicado uma vez em apenas elementos de marcação <p>

```
p#sobrenome {
    background-color: grey;
}
```

Seletor id primeiro_nome

```
<article>
    <h2>Equipe:</h2>
    <p id="primeiro_nome">Charlie</p>
    <p id="sobrenome">Brown</p>
</article>
```

Seletor id sobrenome

Observe que o seletor **p#sobrenome** só funciona se for aplicado a um elemento de marcação **<p>**, caso você insira esse id em outro elemento de marcação não irá funcionar.

Seletor *

O seletor ***** é usado para aplicar uma **regra CSS em todos os elementos** de uma página web. Por exemplo:

```
* {
    background-color: blue;
}
```

O seletor de exemplo define a **cor de fundo amarelo para todos os elementos que não possuem cor de fundo definida**. Lembrando: **se você criar um seletor para algum elemento específico, ele irá sobrepor o seletor ***. Crie um arquivo para construir a sua página web e vamos **inserir** o código **CSS**.

```
@charset "utf-8";
* {
    background-color: #49274A;
    color: #F8EEE7;
}
table * {
    background-color: #94618E;
    text-align: center;
    width: 400px;
}
p {
    background-color: #F4DEC8;
    color: black;
}
```

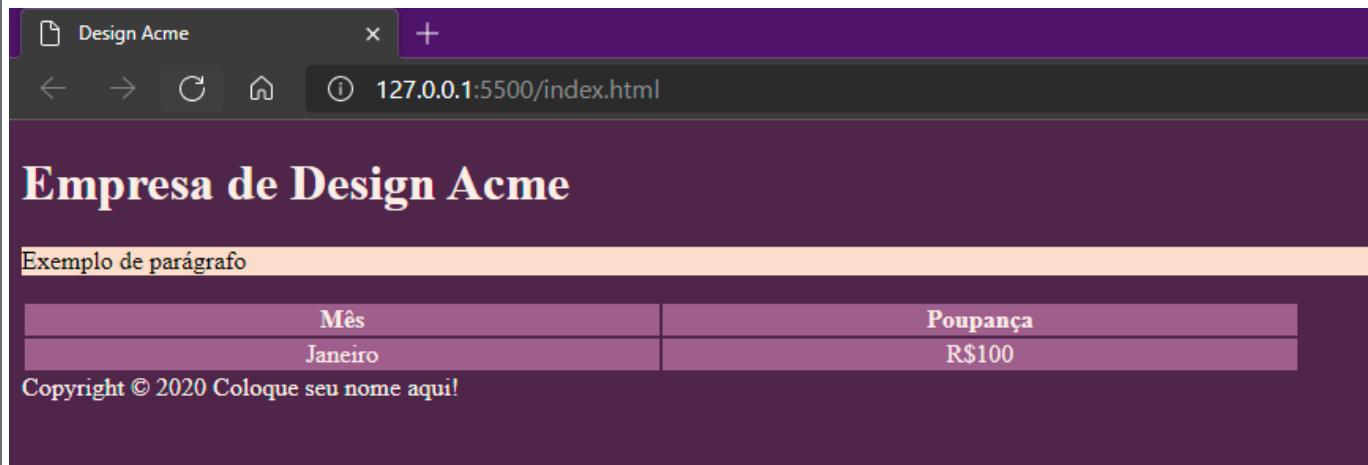
Crie um arquivo para construir a sua página web e vamos **inserir** o código na página **HTML**.

```
<!DOCTYPE html>

<html lang="pt">
  <head>
    <title>Design Acme</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="estilo.css">
  </head>

  <body>
    <header>
      <h1>Empresa de Design Acme</h1>
    </header>
    <article>
      <p>Exemplo de parágrafo</p>
      <table>
        <tr>
          <th>Mês</th>
          <th>Poupança</th>
        </tr>
        <tr>
          <td>Janeiro</td>
          <td>R$100</td>
        </tr>
      </table>
    </article>
    <footer>Copyright &copy; 2020 Coloque seu nome aqui!</footer>
  </body>

</html>
```



Seletor * que será aplicado em todos os elementos que não tem estilo CSS definido.

```
* {
background-color: #49274A;
color: #F8EEE7;
}
```

Seletor table * que será aplicado em todo elemento <table> da página web.

```
table * {
background-color: #94618E;
text-align: center;
width: 400px;
}
```

Seletor p que será aplicado em todo elemento <p> da página web.

```
p {
background-color: #F4DECB;
color: black;
}
```

Seletor elemento elemento

Você pode criar um **seletor para aplicar um estilo CSS em um elemento dentro de um outro elemento específico**. Por exemplo, podemos criar um **seletor para todo elemento parágrafo <p> dentro do elemento <article>**:

```
article p {
background-color: #F4DECB;
color: black;
}
```

Nesse exemplo, os **outros elementos parágrafos <p> não estarão configurados com o estilo exclusivo para os parágrafos, que estão dentro do elemento <article>**.

Não confunda:

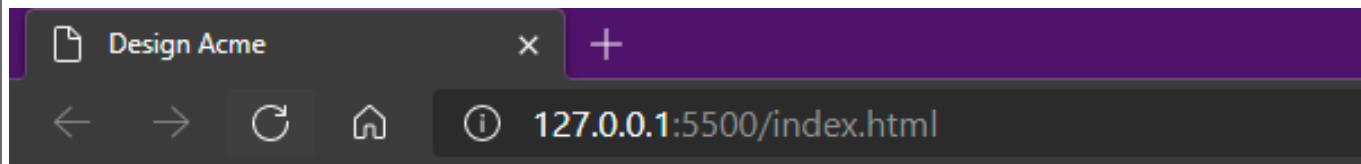
```
article p {
background-color: #F4DECB;
color: black;
}
(a)
```

```
h1, h2 {
color: red;
}
(b)
```

O estilo descrito em **(a)** é aplicado a todo elemento parágrafo dentro da estrutura `<article>`, por sua vez, o estilo descrito em **(b)** é aplicado aos elementos `<h1>` e `<h2>` dentro da página web. Crie um arquivo para construir a sua página web e vamos inserir o código na página **HTML**.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <title>Design Acme</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="estilo.css">
</head>

<body>
    <header>
        <h1>Empresa de Design Acme</h1>
        <p>Parágrafo dentro do &lt;header&gt;</p>
    </header>
    <article>
        <p>Exemplo de parágrafo dentro do &lt;article&gt;</p>
        <h2>H2 dentro do article</h2>
        <p>Outro parágrafo dentro do &lt;article&gt;</p>
    </article>
    <footer>
        <p>Exemplo de parágrafo dentro do &lt;footer&gt;</p>
    </footer>
</body>
</html>
```



Empresa de Design Acme

Parágrafo dentro do <header>

Exemplo de parágrafo dentro do <article>

H2 dentro do article

Outro parágrafo dentro do <article>

Exemplo de parágrafo dentro do <footer>

Crie um arquivo para construir a sua página web e vamos **inserir** o código **CSS**.

```
@charset "utf-8";
* {
    background-color: #49274A;
    color: #F8EEE7;
}

article p {
    background-color: #F4DECB;
    color: black;
}
```



Seletor * que será aplicado nos outros elementos da página web.

```
* {
    background-color: #49274A;
    color: #F8EEE7;
}
```

Seletor article p que será aplicado a todo parágrafo dentro da estrutura article.

```
article p {
    background-color: #F4DECB;
    color: black;
}
```



Elementos Visuais

Os objetivos desta aula são:

- Formatar linhas e bordas com folhas de estilo CSS;
- Formatar imagens com folhas de estilo CSS.

Bons estudos!

Configurando linhas e bordas

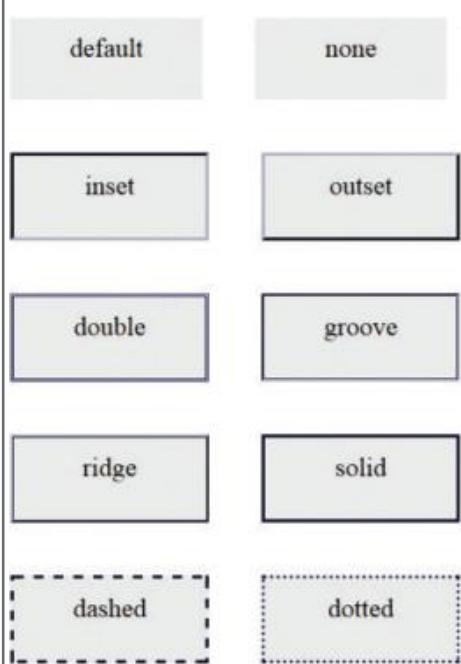
As pessoas **web designers** costumam usar elementos visuais como **linhas** e **bordas** para separar ou definir **áreas nas páginas da web**. Então, vamos implementar duas técnicas de codificação para configurar uma linha em uma página da web: o elemento de linha horizontal **<hr> HTML** e as propriedades de **borda e preenchimento CSS**.

Elementos **<hr>** e **<border>**

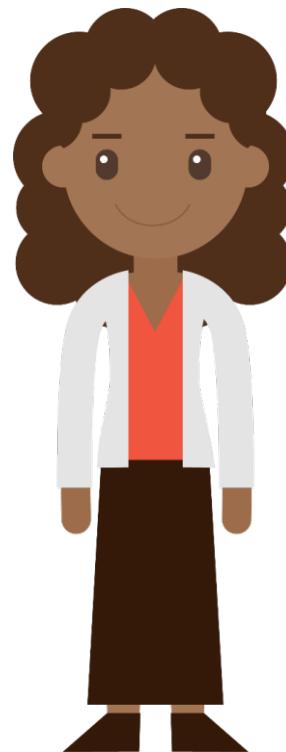
Um elemento de linha horizontal, **<hr>**, separa visualmente **áreas de uma página e configura uma linha horizontal** em uma página da web. Como o elemento de **linha horizontal não contém nenhum texto, ele é codificado como uma tag única e não em um par de tags de abertura e fechamento**.

A propriedade **border** configura a **borda**, ou **limite, em torno de um elemento**. Por **padrão**, a borda tem uma **largura** definida como **0** e **não é exibida**. Você pode definir a largura da borda (**border-width**), a cor da borda (**border-color**) e o estilo da borda (**border-style**) com a **propriedade border**. E ainda mais, você pode até **definir configurações individuais** para as bordas **superior, direita, inferior e esquerda** usando as propriedades **border-top, border-right, border-bottom e border-left**.

A propriedade **border-style** configura o **tipo de linha exibida na borda**. As **opções de formatação** incluem **inset, outset, double, groove, ridge, solid, dashed e dotted**. Esteja ciente de que esses valores de propriedade não são aplicados uniformemente pelos navegadores. A imagem abaixo, mostra como uma versão recente do Firefox renderiza vários valores de estilo de borda.



Exemplos de **border-style** no navegador Firefox.



A classe **bordatracejada** mostrada abaixo configura as bordas para uma largura de borda de **3 pixels, com estilo de borda tracejado (dashed) e uma cor de borda de #000033**:

```
.bordatracejada {  
    border-width: 3px;  
    border-style: dashed;  
    border-color: #000033;  
}
```

Você também pode criar a classe **bordatracejada** com uma **notação abreviada** com as propriedades em uma linha e listando os valores na ordem **border-width, border-style e border-color**, como mostrado abaixo.

```
.bordatracejada {  
    border: 3px dashed #000033;  
}
```

A propriedade **padding** configura o **espaço vazio entre o conteúdo do elemento HTML (geralmente texto) e a borda**. Por **padrão**, o **preenchimento** é definido como **0**. Se você configurar uma cor de fundo para um elemento, a cor é aplicada tanto ao preenchimento quanto ao conteúdo das áreas.

Vamos criar um arquivo **index.html**, **services.html** e o arquivo de **estilo.CSS** na sua página web.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Título</title>  
    <link rel="stylesheet" href="style.css">  
</head>  
<body>  
    <header>  
        <h1>Empresa de Design Acme</h1>  
    </header>  
    <nav>  
        <a href="index.html">Home</a>  
        <a href="services.html">Serviços</a>  
    </nav>  
    <main>  
        <article>  
            <h2>Novas mídias e novo design</h2>  
            <p>A <span class="company">Design Acme</span> está habilitada para construir  
a página da sua empresa e  
garantir  
sua presença na web.</p>
```

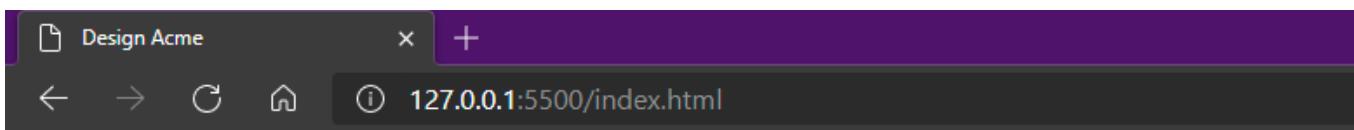
```

<p>Nós oferecemos os seguinte serviços</p>
<ul>
    <li>Desenvolvimento de Websites</li>
    <li>Animação Interativa</li>
    <li>Soluções de E-commerce</li>
    <li class="feature">Estudos de usabilidade</li>
    <li class="feature">Otimização de máquinas de buscas</li>
</ul>
<h2>Atendendo às suas necessidades de negócios</h2>
<p>Nossos designers especializados são criativos e prontos para trabalhar
com você.</p>
<p>Aproveite o crescimento da Web!</p>
</article>
</main>
<footer>Copyright © 2021 - Coloque seu nome aqui!</footer>

</body>
</html>

```

Salve sua página web com o nome **index.html**



Empresa de Design Acme

[Home](#) [Serviços](#)

Novas mídias e novo design

A Design Acme está habilitada para construir a página da sua empresa e garantir sua presença na web.
Nós oferecemos os seguintes serviços:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Nossos designers especializados são criativos e prontos para trabalhar com você!
Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!

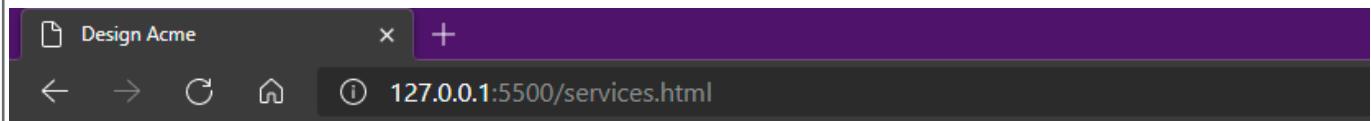
Vamos criar a página services.html

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Empresa de Design Acme</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Empresa de Design Acme</h1>
    </header>
    <nav>
        <a href="index.html">Home</a>
        <a href="services.html">Serviços</a>
    </nav>
    <main>
        <article>
            <h2>Nossos serviços encontram a necessidade do seu negócio </h2>
        </article>
        <dl>
            <dt>
                <strong>Desenvolvimento de Website</strong>
            </dt>
            <dd> Se suas necessidades são grandes ou pequenas, a Acme pode colocar sua empresa na Web! </dd>
            <dt>
                <strong>Animação Interativa</strong>
            </dt>
            <dd>Treinamentos multimídia e animações de marketing são nossa especialidade.</dd>
            <dt>
                <strong>Soluções de E-Commerce</strong>
            </dt>
            <dd> A Acme oferece entrada rápida e fácil no mercado de comércio eletrônico.</dd>
            <dt>
                <strong>Estudos de Usabilidade</strong>
            </dt>
            <dd>A Acme pode avaliar a usabilidade do seu site atual e sugerir melhorias.</dd>
            <dt>
                <strong>Otimização das ferramentas de busca por site na Internet</strong>
            </dt>
            <dd>A maioria das pessoas encontra sites usando mecanismos de pesquisa. A Acme pode ajudá-lo a divulgar seu site.</dd>
        </dl>
    </main>
</body>
</html>

```

Salve sua página web com o nome **services.html**



Empresa de Design Acme

[Home](#) [Serviços](#)

Nossos Serviços Encontram a Necessidade do seu Negócio

Desenvolvimento de Website

Se suas necessidades são grandes ou pequenas, a Acme pode colocar sua empresa na Web!

Animação Interativa

Treinamentos multimídia e animações de marketing são nossa especialidade.

Soluções de E-Commerce

A Acme oferece entrada rápida e fácil no mercado de comércio eletrônico.

Estudos de Usabilidade

AO Acme pode avaliar a usabilidade do seu site atual e sugerir melhorias.

Otimização das ferramentas de busca por site na Internet

A maioria das pessoas encontra sites usando mecanismos de pesquisa. A Acme pode ajudá-lo a divulgar seu site.

Copyright © 2020 Coloque seu nome aqui!

Vamos criar o estilo.css

```
@charset "utf-8";
body {
    background-color: #e6e6fa;
    color: #191970;
    font-family: Arial, Helvetica, sans-serif;
    margin-left: auto;
    margin-right: auto;
    width: 80%;
}

h1 {
    background-color: #191970;
    color: #e6e6fa;
    line-height: 200%;
    font-family: Georgia, 'Times New Roman', Times, serif;
    text-shadow: 3px 3px 5px #cccccc;
    padding: 1em;
}

h2 {
    background-color: #aeaed4;
    font-family: Georgia, 'Times New Roman', Times, serif;
    text-align: center;
    border-bottom: 2px dashed #191970;
```

```

}

p {
    font-size: 0.9em;
    text-indent: 3em;
}

ul {
    font-weight: bold;
}

footer {
    color: #333333;
    font-size: 0.75em;
    font-style: italic;
    font-family: 'Courier New', Courier, monospace;
}

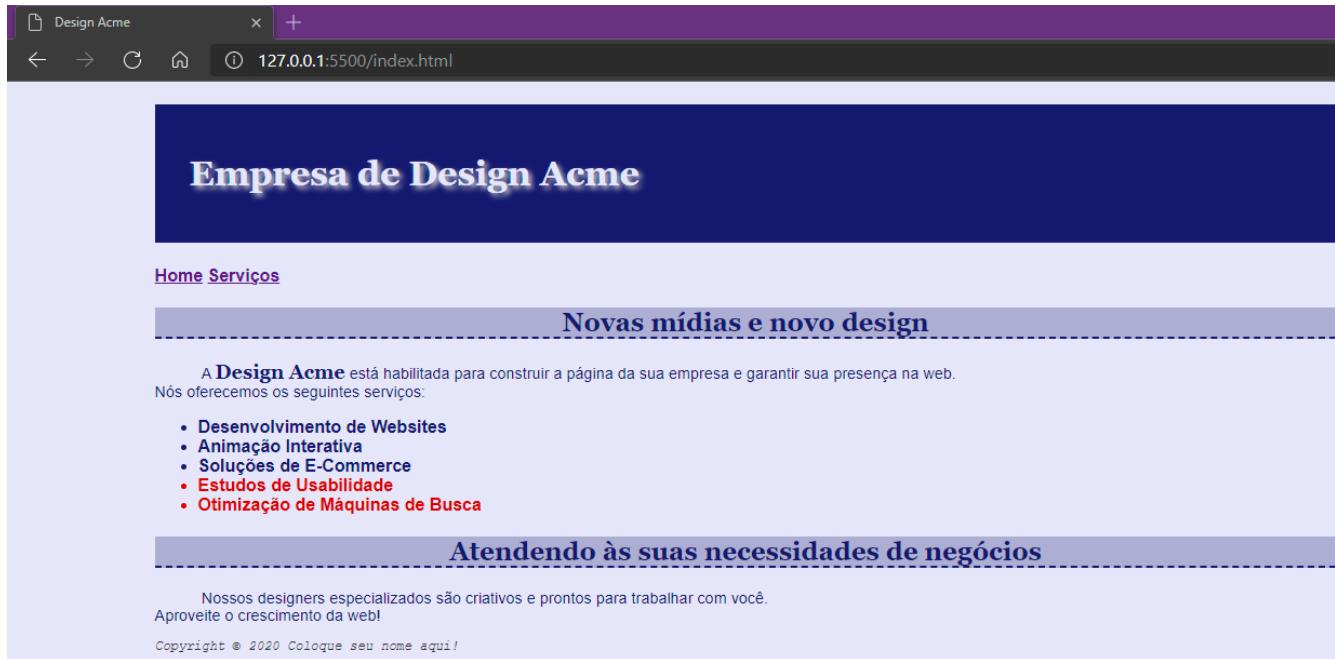
nav {
    font-weight: bold;
}

.feature {
    color: #c70000;
}

.company {
    font-weight: bold;
    font-family: Georgia, 'Times New Roman', Times, serif;
    font-size: 1.25em;
}

```

Salve o arquivo com o nome estilo.css



Design Acme 127.0.0.1:5500/index.html

Empresa de Design Acme

[Home](#) [Serviços](#)

Novas mídias e novo design

A **Design Acme** está habilitada para construir a página da sua empresa e garantir sua presença na web. Nós oferecemos os seguintes serviços:

- Desenvolvimento de Websites
- Animação Interativa
- Soluções de E-Commerce
- Estudos de Usabilidade
- Otimização de Máquinas de Busca

Atendendo às suas necessidades de negócios

Nossos designers especializados são criativos e prontos para trabalhar com você. Aproveite o crescimento da web!

Copyright © 2020 Coloque seu nome aqui!



Propriedade padding com
espaçamento 1em.

Propriedade border-bottom que
configura a borda inferior com os
valores de largura e de borda 2px,
estilo dashed e cor #191970.

Estilo CSS em elemento

Você pode configurar o **estilo** de **uma** ou **mais imagens exibidas na sua página web**. Para isso, podemos utilizar o elemento **<figure>** no HTML. O elemento **<figure>** representa o conteúdo independente, frequentemente com uma legenda (**<figcaption>**), e é normalmente referido como uma única unidade como, por exemplo, uma imagem. Desse modo, é possível configurar um seletor CSS para definir um estilo para a imagem e sua legenda (**<figcaption>**).

Importante: Antes de começar a construir o site faça o download da imagem.

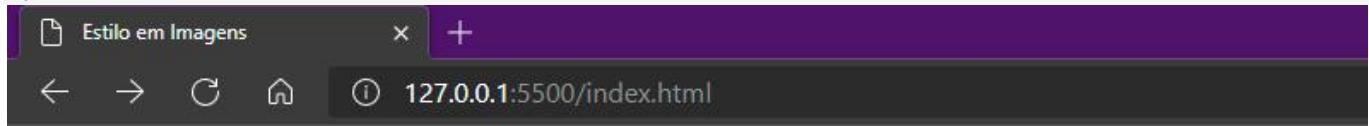
Vamos criar um arquivo **html** e o arquivo de **css** da sua página web.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <title>Estilo em Imagens</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="estilo.css">

```

```
</head>
<body>
  <header>
    <h1>Aplicando estilo em uma imagem</h1>
  </header>
  <article>
    <figure>
      
      <figcaption>Ilha Tropical</figcaption>
    </figure>
  </article>
  <footer>Copyright © 2020 Coloque seu nome aqui!</footer>
</body>
</html>
```



Aplicando estilo em uma imagem



Ilha Tropical

Copyright © 2020 Coloque seu nome aqui!

Vamos criar o **02.estilo.css**

```
@charset "utf-8";
body {
    background-color: #ffffb3;
    color: #191970;
    font-family: Arial, Verdana, sans-serif;
}

h1 {
    background-color: #ffcc00;
    color: black;
    text-align: center;
    line-height: 200%;
    font-family: Georgia, "Times New Roman", serif;
    padding: 0.5em;
}

img {
    height: 240px;
    width: 320px;
}

figure {
    width: 320px;
    padding: 5px;
    border: 2px solid #000000;
}

figcaption {
    text-align: center;
    font-family: Papyrus, fantasy;
}

footer {
    color: #333333;
    font-size: .75em;
    font-style: italic;
    font-family: 'Courier New', Courier, monospace;
}
```

Estilo em Imagens x + 127.0.0.1:5500/index.html

Aplicando estilo em uma imagem



Ilha Tropical

Copyright © 2020 Coloque seu nome aqui!

Elemento configurado com as propriedades alt: "Tropical Island", height: 240px e width: 320px.

Elemento <figure> configurado com as propriedades width: 320px, padding: 5px e border: 2px solid #000000.

Elemento <figcaption> configurado com as propriedades text-align: center e font-family: Papyrus, fantasy.

```
img {
    alt: "Tropical Island";
    height: 240px;
    width: 320px;
}

figure {
    width: 320px;
    padding: 5px;
    border: 2px solid #000000;
}

figcaption {
    text-align: center;
    font-family: Papyrus, fantasy;
}
```

Propriedade background-image

Você já aprendeu como configurar a cor do plano de fundo da área ocupada por um elemento do HTML utilizando a propriedade background-color no CSS, mas também é possível configurar uma imagem como plano de fundo. A propriedade **background-image** do CSS permite configurar uma imagem como plano de fundo na área de um elemento do HTML. Por exemplo, o trecho seguinte em **CSS** configura o seletor **body** do **HTML** com a imagem gráfica **texture1.png** como plano de fundo:

```
body {
    background-image: url('texture1.png');
}
```

Você pode configurar as duas propriedades de cor e imagem de fundo, para que o navegador mostre uma cor de fundo, enquanto a imagem é carregada na página web. Vejamos o exemplo:

```
body {
    background-color: #99cccc;
    background-image: url('texture1.png');
}
```

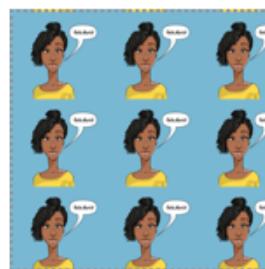
Outra propriedade é a **background-repeat**. Por padrão, o navegador repete a imagem por todo plano de fundo. Você pode alterar esse comportamento configurando a propriedade **background-repeat** com um dos seguintes valores: **repeat** (default), **repeat-y** (repete verticalmente), **repeat-x** (repete horizontalmente), **no-repeat** (não repete), **round**, **round repeat**, **space** ou **space repeat**.



repeat-x



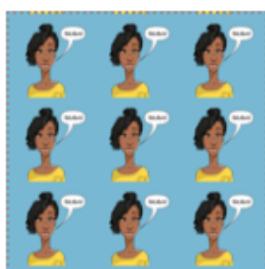
repeat-y



repeat



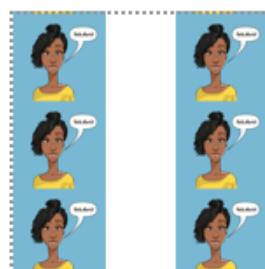
space



round



no-repeat



space repeat



round space

Valores possíveis da propriedade background-repeat

Existem outras propriedades

background-position: permite definir o local da sua imagem na página de acordo com valores numéricos (em pixels, porcentagem, etc.) ou pelos rótulos: left (padrão), top, center, bottom, and right.

background-attachment: configura se a imagem irá rolar com o resto da página (scroll) ou se irá permanecer fixa (fixed).

background-size: pode ser usada para redimensionar a imagem com valores de largura (**width**) e altura (**height**), em pixels ou porcentagem, ou com rótulos como: cover (cobre toda a área disponível, preservando o aspect radio) ou contain (escala a imagem para a maior medida entre a largura e altura, preservando o aspect radio).

Vamos Praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

Crie o arquivo **HTML**

Importante: Antes de começar a construir o site, faça o download da imagem!

```
<!DOCTYPE html>
<html lang="pt-br">

    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="estilo.css">
        <title>Exemplo background-image</title>

    </head>

    <body>
        <header class="showcase">
            <h1> Praia de São Miguel dos Milagres</h1>

            <p>O litoral de Alagoas é um dos mais bonitos do Brasil. Entre Maceió (AL) e Tamandaré (PE) fica localizada a APA Costa dos Corais, uma área de arrecifes e piscinas naturais. Apesar de Maragogi ser o destino mais famoso da Costa, a cidade acabou ficando muito cheia e pode assustar quem viaja procurando sossego.</p>

            <h2>História</h2>

            <p>Chamava-se, antes, Freguesia de Nossa Senhora Mãe do Povo. Mudou sua denominação, segundo a tradição, depois
```

que um pescador encontrou na praia uma peça de madeira coberta de musgos e algas marinhas. Ao levá-la para casa e fazer sua limpeza, descobriu que se tratava de uma imagem de São Miguel Arcanjo, provavelmente caída de alguma embarcação. Ao terminar o trabalho de limpeza, o pescador descobriu espantado, que uma ferida persistente que o afligia há tempos estava totalmente cicatrizada.

A notícia logo se espalhou, fazendo com que aparecessem pessoas em busca de cura para suas doenças e de novos milagres. Sua colonização tomou corpo durante o período da invasão holandesa, quando moradores da sofrida Porto Calvo fugiram em busca de um lugar seguro para abrigar suas famílias e de onde pudesse avistar com antecipação a chegada dos inimigos batavos. A capela inicial, que deu origem à freguesia estabelecida pela Igreja Católica, foi dedicada a Nossa Senhora Mãe do Povo.

Apesar de Maragogi ser o destino mais famoso da Costa, a cidade acabou ficando muito cheia e pode assustar quem viaja procurando sossego.

São Miguel dos Milagres virou uma das opções mais tranquilas para quem visita a Costa dos Corais.

Com pousadas pé na areia e vilas pequenas e rústicas, em São Miguel dos Milagres as praias são

praticamente selvagens e mais tranquilas do que em Maragogi.

A partir de lá, também é possível fazer um dos deliciosos passeios de jangada até as piscinas naturais.

</header>

</body>

</html>

Vamos criar o **estilo.css**

```
@charset "utf-8";
* {
    margin: 0;
    padding: 0;
}

body {
    margin: 0;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 17px;
    color: #333;
    line-height: 1.6;
}

.showcase {
    background-image: url('03.praia.jpg');
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
}
```

A screenshot of a web browser window titled "Exemplo background-image". The URL in the address bar is "127.0.0.1:5500/05.Elementos-visuais/código-fonte-material-educador/03.background-image.html". The page content includes a heading "Praia de São Miguel dos Milagres" and some descriptive text about the beach. The background of the page is a photograph of a beach with clear blue water and a small boat.

Isso é tudo, fique à vontade em fazer modificações na página e ver o resultado dessas modificações.



Margin versus Padding

Os objetivos desta aula são:

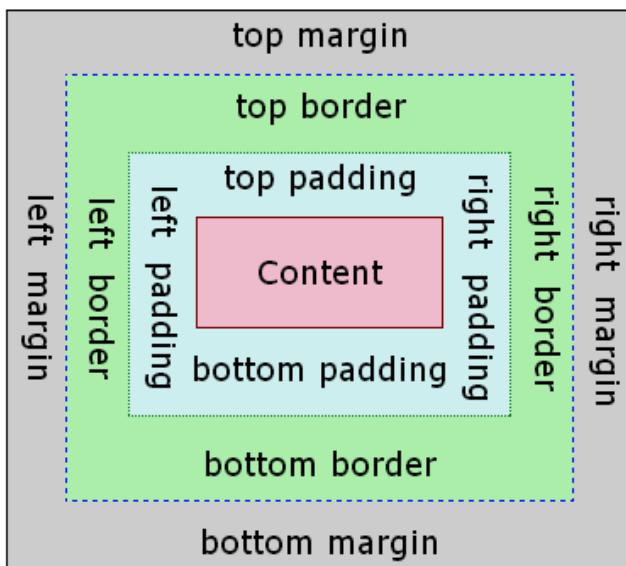
- Diferenciar o uso de classes e id;
- Compreender o uso de id nas folhas de estilo;
- Aplicar formatações utilizando id nas CSS e o seletor *.

Bons estudos!

Diferença entre margin e padding

Quando estamos começando em CSS geralmente podemos confundir os conceitos de **margin** e **padding**. Isso acontece, porque os dois são essenciais, nos permitindo dar espaçamento em elemento do HTML, mas existe importantes diferenças fundamentais no uso desses dois estilos.

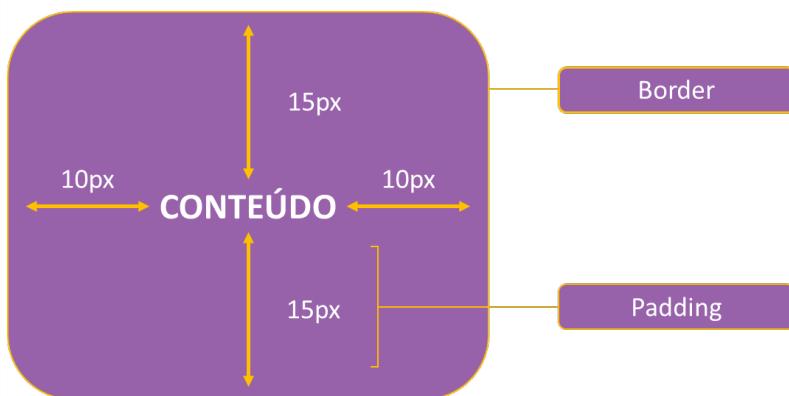
Antes de ver a diferença entre essas duas propriedades, vamos entender o conceito de **Box Model**. No CSS, o termo **Box Model** é usado quando estamos trabalho com o design ou o layout em uma página web. Cada elemento HTML em um documento é visto como uma **caixa (box) retangular** e o **CSS box model** é essencialmente essa caixa com diversas camadas em volta do conteúdo, que é formada por **padding**, **border** e **margin**.



Diferença entre margin e padding.

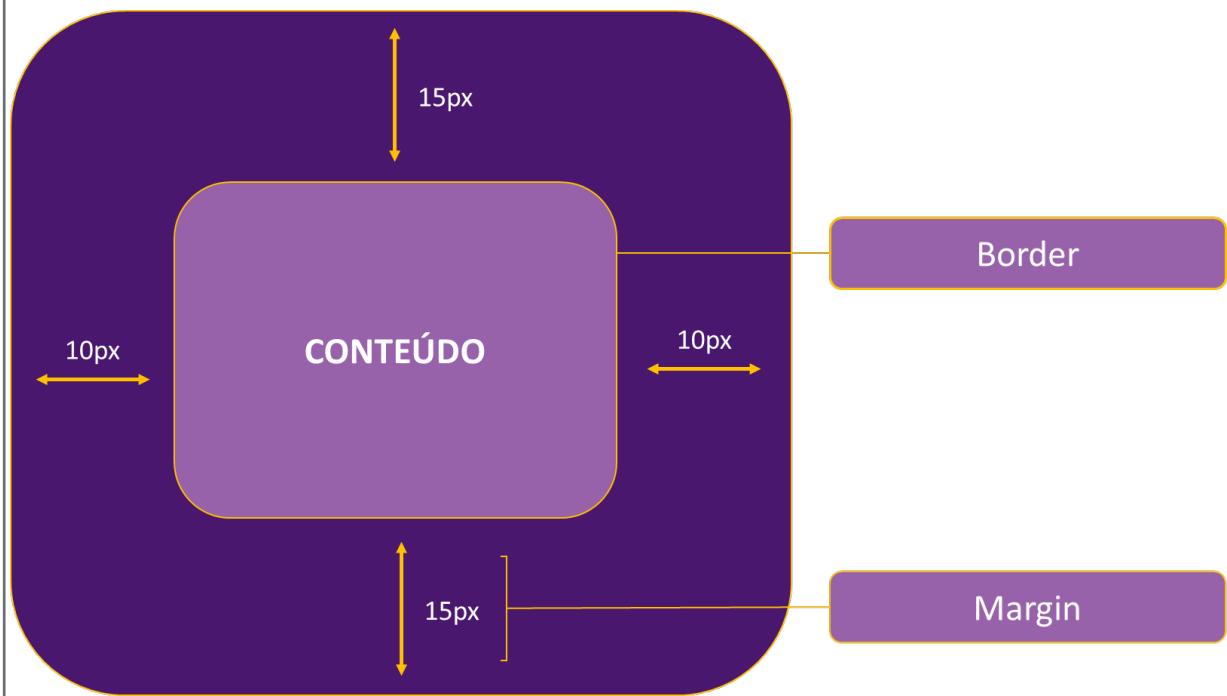
Basicamente, o conteúdo é a combinação dos textos e os elementos das páginas web, tais como: imagens, parágrafos, headings, listas, entre outros. Ele é o local onde os textos e as imagens irão aparecer.

Por sua vez, a propriedade **padding** é o **espaço entre o conteúdo e a borda**. Ou seja, **está do lado de dentro da borda**. Assim, a propriedade **padding** é usada para definir o espaço em torno do conteúdo e, também, contorna os quatro lados e pode ser configurado separadamente cada um dos lados.



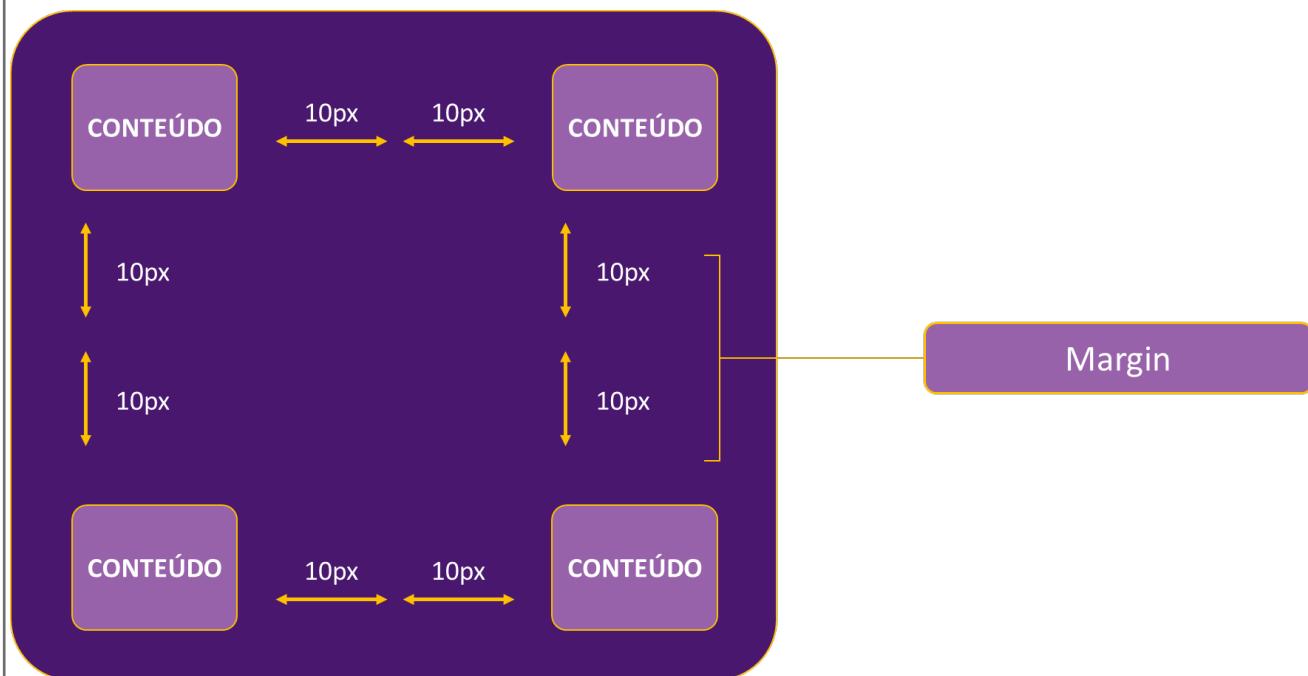
Padding.

A propriedade **margin** é o espaço entre a borda do elemento HTML e o próximo elemento da sua página web. Ou seja, está do lado de fora da borda.



Margin.

Assim, a propriedade **margin** é usada para definir a espessura entre a borda e outros elementos HTML e contorna os quatro lados do conteúdo e pode ser configurado separadamente cada um dos lados.



Margem entre outros elementos HTML da página web.

Configurando a propriedade margin

No CSS, a propriedade **margin** permite definir a margem com valores numéricos (**px, cm, pt, em, rem, etc.**) ou valores percentuais (**%**). Ambos podem ser **negativos** ou **positivos**. Você pode configurar cada uma das margens de um elemento HTML separadamente ou todas de uma vez.

Para configurar cada uma das margens você pode utilizar as instruções: **margin-top**, **margin-right**, **margin-bottom** e **margin-left**. Por exemplo, o trecho a seguir configura a margem superior com **100 pixels**, a inferior com **100 pixels**, a margem da esquerda com **80 pixels** e a da direita com **150 pixels** de um parágrafo.

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

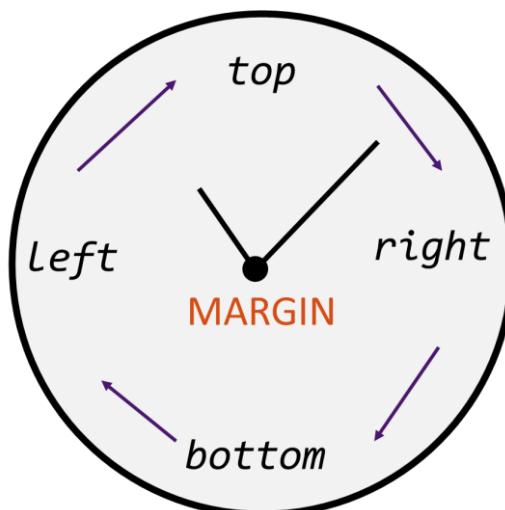
Para fazer a mesma configuração anterior, mas com uma única instrução você deve usar o seguinte trecho:

```
p {  
    /* margin: top right bottom left */  
    margin: 100px 150px 100px 80px;  
}
```



Dica!

Para memorizar essa notação da CSS, basta lembrar dos ponteiros de um relógio! Ela também pode ser usada em outros elementos, como o padding por exemplo.



Existem outros formatos de usar a instrução única. Por exemplo, caso você queira configurar as margens **top** e **bottom** com valores de **25 pixels** e **right** e **left** com **50 pixels**, basta usar o trecho:

```
p {  
    /* margin: (top e bottom) (right left) */  
    margin: 25px 50px;  
}
```

Ou configurar as margens **top** com **25 pixels** e **right** e **left** com **50 pixels** e **bottom** com **75 pixels**, basta usar o trecho:

```
p {  
    /* margin: (top) (right bottom) (left) */  
    margin: 25px 50px 75px;  
}
```

Ou configurar as **quatro margens** com **25 pixels**, basta usar o trecho:

```
p {  
    /* margin: top right bottom left */  
    margin: 25px;  
}
```

Você também pode usar o valor **auto**, que deixa para o navegador calcular a margem do elemento automaticamente ou usar o valor **inherit**, que especifica que a margem deverá ser herdada do **elemento pai** (primeiro elemento acima na hierarquia do design da página web).

Configurando a propriedade padding

No CSS, a propriedade **padding** permite definir o espaçamento com valores numéricos (**px, cm, pt, em, rem, etc.**), que podem ser **apenas positivos**. Você pode configurar separadamente **cada espaçamento de um elemento HTML ou todos de uma vez**.

Para configurar cada uma das margens você pode utilizar as instruções: **padding-top, padding-right, padding-bottom e padding-left**.

Por exemplo, o trecho a seguir configura o espaçamento **superior** com **100 pixels**, o **inferior** com **100 pixels**, o da **esquerda** com **80 pixels** e o da **direita** com **150 pixels** de um parágrafo.

```
p {  
    padding-top: 100px;  
    padding-bottom: 100px;  
    padding-right: 150px;  
    padding-left: 80px;  
}
```

Para fazer a mesma configuração anterior, mas com uma única instrução você deve usar o seguinte trecho:

```
p {  
    /* padding: top right bottom left */  
    padding: 100px 150px 100px 80px;  
}
```

Existem outros formatos de usar a instrução única. Por exemplo, caso você queira configurar os espaçamentos **top** e **bottom** com valores de **25 pixels** e **right** e **left** com **50 pixels**, basta usar o trecho:

```
p {  
    /* padding: (top e bottom) (right left) */  
    padding: 25px 50px;  
}
```

Ou configurar os espaçamentos **top** com **25 pixels** e **right** e **left** com **50 pixels** e **bottom** com **75 pixels**, basta usar o trecho:

```
p {  
    /* padding: (top) (right bottom) (left) */  
    padding: 25px 50px 75px;  
}
```

Ou configurar os **quatro espaçamentos** com **25 pixels**, basta usar o trecho:

```
p {  
    /* padding: top right bottom left */  
    padding: 25px;  
}
```

Você também pode usar o valor **auto**, que deixa para o navegador calcular o espaçamento do elemento automaticamente. Ou usar o valor **inherit**, que especifica que o espaçamento deverá ser herdado do **elemento pai** (**primeiro elemento acima na hierarquia do design da página web**).

Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

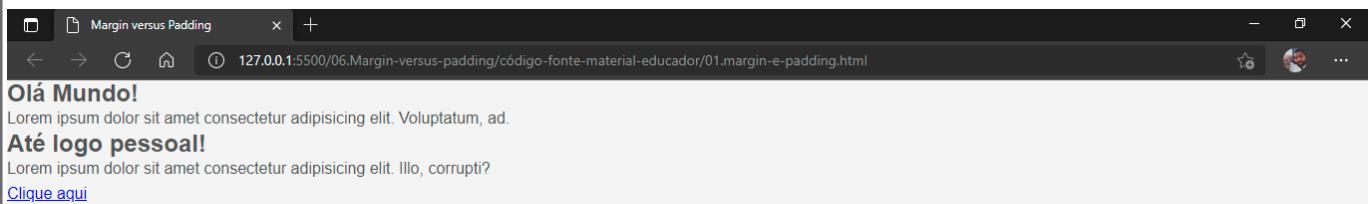
Crie o arquivo **HTML**

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" href="estilo.css">  
    <title>Margin versus Padding</title>
```

```
</head>
<body>
  <section class="container">
    <div class="box-1">
      <h1>Olá Mundo!</h1>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatum, ad.</p>
    </div>
    <div class="box-2">
      <h1>Até logo pessoal!</h1>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Illo, corrupti?</p>
      <a href="#" class="btn">Clique aqui</a>
    </div>
  </section>
</body>
</html>
```

Crie o arquivo **CSS**.

```
@charset "utf-8";
* {
  margin: 0;
  padding: 0;
  border: none;
  top: 0;
}
body {
  background-color: #f4f4f4;
  color: #555555;
  font: normal 16px Arial, Helvetica, sans-serif;
  line-height: 1.6em;
  margin: 0;
}
```



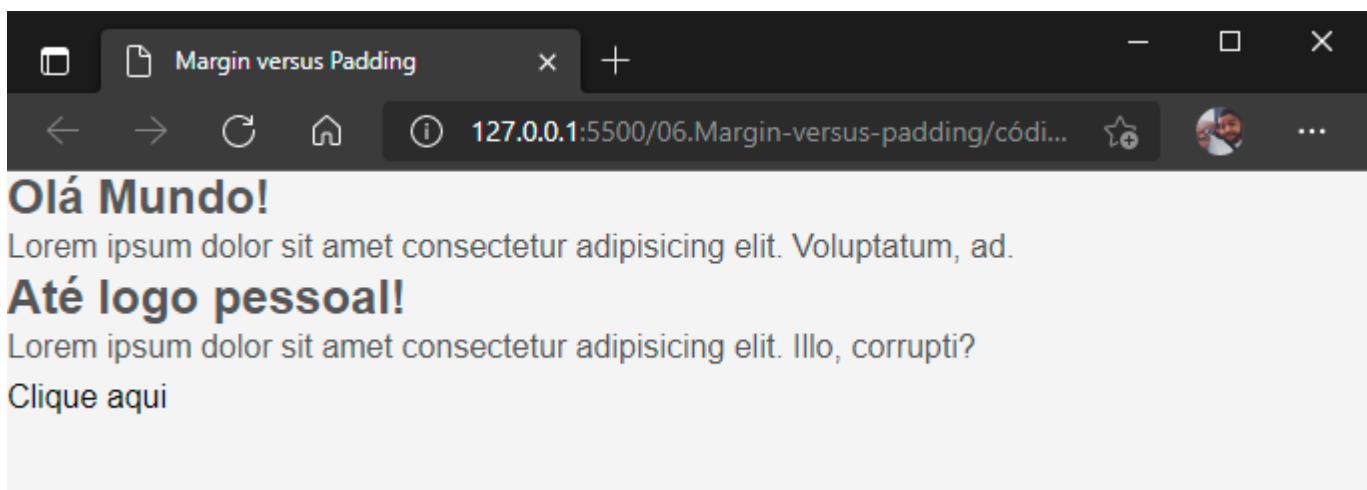
Observe que todos os elementos estão as propriedades de **margin** e **padding** zeradas, pois no arquivo **CSS** fizemos a seguinte configuração:

```
* {
    margin: 0;
    padding: 0;
    border: none;
}
```

Vamos, primeiramente, configurar o elemento `<a>`. No arquivo **CSS**, insira o seguinte trecho de código.

```
a {
    text-decoration: none;
    color: black;
}
```

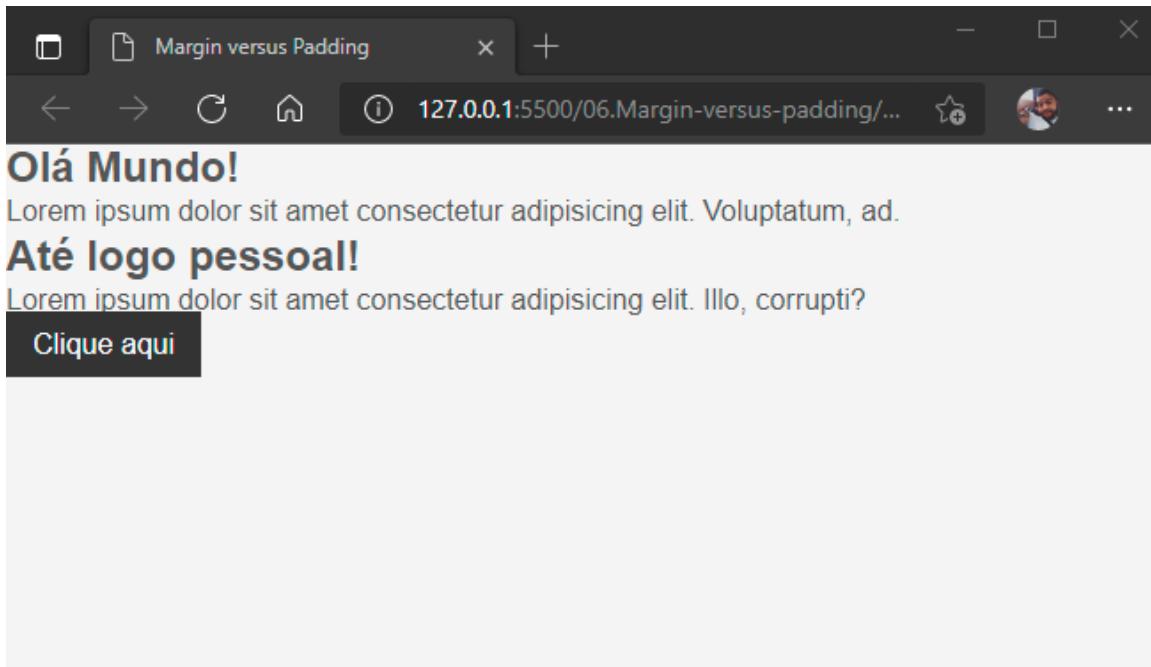
Esse trecho de código fará com que os links da página web não tenham nenhuma decoração (sublinhado, por exemplo) e fiquem na cor preta.



Agora vamos configurar a **classe btn** que está associada ao link com o rótulo **Clique aqui**. No arquivo **CSS**, insira o seguinte trecho de código.

```
.btn {
    background-color: #333;
    color: #fff;
    padding: 10px 15px;
    border: none;
}
```

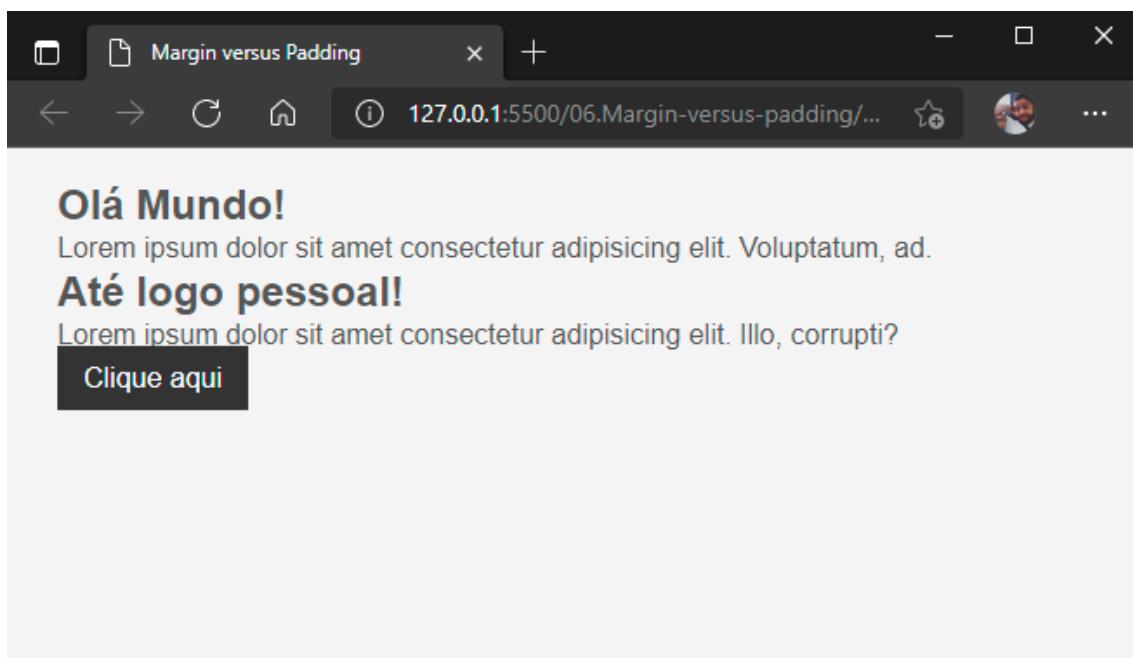
O resultado é a inserção de um **espaço** entre o **conteúdo** e o a **borda do elemento (padding 10px 15px)** de **10 pixels** na parte de **cima** e **embaixo** e de **15 pixels** na **esquerda** e **direita**.



Agora vamos configurar a classe container, que está associada ao elemento <section>. No arquivo **CSS**, insira o seguinte trecho de código.

```
.container {  
    width: 80%;  
    margin: 20px 30px;  
}
```

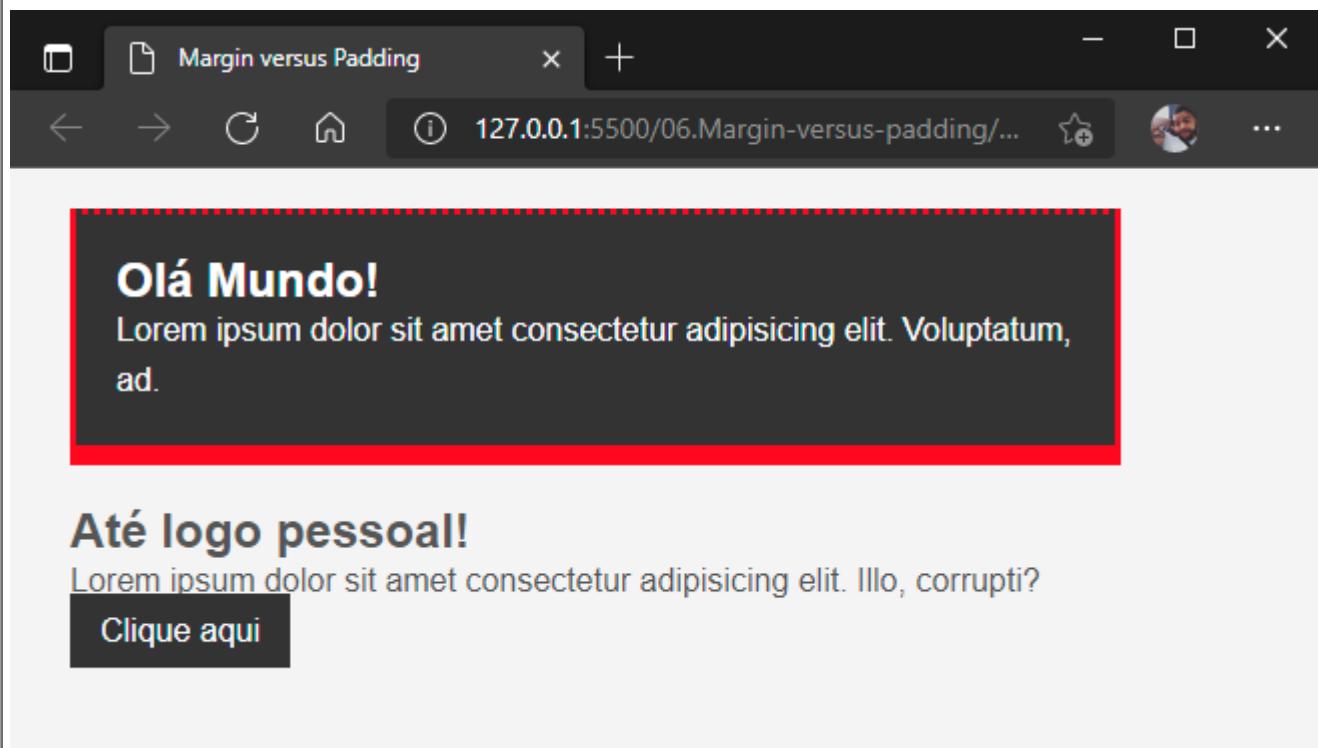
Observe que o conteúdo foi deslocado, pois agora toda a seção tem margin de 20 pixels em cima e embaixo e de 30 pixels na direita e esquerda.

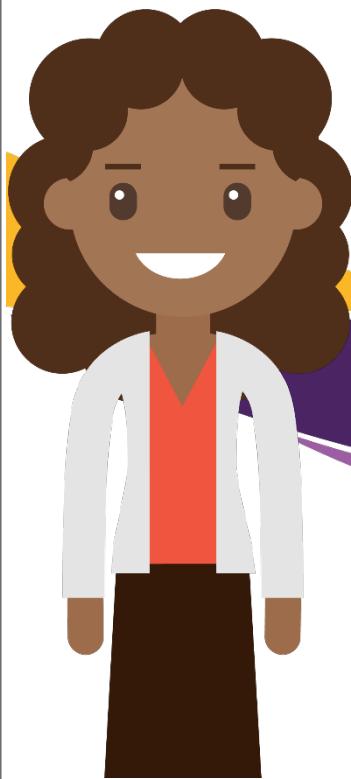


Agora, vamos configurar a **classe box-1**, que está associado ao primeiro elemento <div>. No arquivo **CSS**, insira o seguinte trecho de código:

```
.box-1 {  
    background-color: #333;  
    color: #fff;  
    border-right: 5px red solid;  
    border-left: 5px red solid;  
    border-top: 5px red solid;  
    border-bottom: 5px red solid;  
    border-width: 3px;  
    border-bottom-width: 10px;  
    border-top-style: dotted;  
    padding: 20px;  
}
```

O resultado é a inserção de uma margem de **20 pixels** em **cima e embaixo**, uma **borda vermelha** e um **espaçamento de 20 pixels em volta do conteúdo**.





Medidas no CSS

Os objetivos desta aula são:

- Conhecer as medidas absolutas e relativas;
- Aplicar as medidas absolutas e relativas com as folhas de estilo.

Bons estudos!

Medidas no CSS

O CSS possui diferentes medidas para expressar tamanho de algum elemento do HTML. Existem diversas propriedades que você precisa especificar o tamanho para configurar completamente o estilo desejado, tais como: **width**, **margin**, **padding**, **font-size**, etc. O tamanho sempre é representado por um número seguido de uma unidade, como por exemplo: **10px**, **3em**, **25rem**, etc.

Importante!



Não deixe espaço entre o número e a unidade escolhida, apenas se o valor for 0 (zero).

```
h1 {  
    font-size: 60px;  
}  
  
p {  
    font-size: 25px;  
    line-height: 50px;  
}
```

Algumas propriedades do CSS **permitem valores negativos e positivos** como, por exemplo, **margin**; e outros **somente positivo** como, por exemplo, **padding**.

Medidas absolutas

As medidas de absolutas são as mais **comuns** e **fixas**, ou seja, **não dependem de nenhum valor de referência**. Essas medidas não são alteradas de acordo com as especificações do dispositivo e normalmente são indicadas quando sabemos perfeitamente as características físicas e as configurações das mídias onde serão exibidos o website.

Elas não são recomendadas para um site que será exibido em telas de diferentes tamanhos. Porém podem ser muito úteis para a formatação do site para impressão, por exemplo. Algumas medidas absolutas são:

Unidade	Descrição
cm	centímetros
mm	milímetros
in	polegadas (1in = 96px = 2.54cm)
px*	pixels (1px = 1/96 de 1in)
pt	points (1pt = 1/72 de 1in)

pc

paica (1pc = 12 pt)

* **Pixels (px) são relativos do ponto de vista do dispositivo.** Para dispositivos de baixo dpi, 1px é um pixel de dispositivo (ponto) da tela. Para impressoras e telas de alta resolução, 1px implica vários pixels de dispositivo.

Pixels são os pequenos pontos luminosos da tela do seu monitor, celular, televisão, etc. A unidade **Points** é mais conhecida pelos designers, principalmente os que estudam tipografia e não é muito popular para os Web Designers. **Polegada** (do inglês, inch) é uma unidade de medida geralmente usada para definir tamanho de TVs, monitores, telas, etc. **Centímetros** e **milímetros** são as unidades do sistema métricos. Por fim, **Paica** é mais comum em tipografia.

Medidas relativas

As medidas relativas são calculadas tendo como base uma outra unidade de medida definida. O benefício de usar uma medida relativa é que você pode fazer o tamanho do texto ou de um elemento ser dimensionado em relação a todo o resto da página. Assim garantimos um layout consistente e fluido em diferentes mídias. Algumas medidas relativas são:

Unidade	Descrição
em	Em relação ao tamanho da fonte do elemento (2em significa 2 vezes o tamanho da fonte atual)
ex	Relativo à altura x da fonte atual (raramente usado)
ch	Relativo à largura do "0" (zero)
rem	Relativo ao tamanho da fonte o elemento raiz
vw	Em relação a 1% da largura do viewport*
vh	Em relação a 1% da altura do viewport*
vmin	Em relação a 1% da dimensão menor do viewport*
vmax	Em relação a 1% da dimensão maior do viewport*
%	Relativo ao elemento pai

viewport = tamanho da janela do navegador e pode ser traduzido como a área visível do website. Por exemplo, se a janela do navegador tem 50cm de largura, 1vw é igual a 0,5cm.

A unidade **em** altera o tamanho configurado em **uma propriedade dos elementos filhos multiplicando o tamanho da mesma propriedade do elemento pai**. Veja por exemplo o trecho de código:

```
<style>
    #div {
        font-size: 16px;
    }

    #filho {
        font-size: 2em;
    }
</style>

<div id="pai">
    div pai
    <div id="filho">
        div filho
    </div>
</div>
```

Nesse exemplo, a propriedade **font-size** do primeiro elemento **<div>** (elemento pai) é **16px**. Então o elemento filho (segundo elemento **<div>**) terá o tamanho da propriedade **font-size** multiplica por 2 vezes em relação ao pai. Ou seja, **2em** será igual a **32px**.

O **REM** é o sucessor do **EM**. Eles possuem a mesma lógica de funcionamento. Porém, a forma de implementação é diferente. No caso do **REM**, o tamanho relativo é o elemento raiz (root elemento **<html>**). Por padrão, o elemento raiz **<html>** tem tamanho de **16px**, portanto **1rem** é igual a **16px**, **2rem** é igual a **32px** e assim por diante. Porém, você pode alterar o tamanho padrão, por exemplo:

```
html {
    font-size: 62.5%; /*tamanho do elemento raiz agora é 0,625*16px = 10px*/
}

h1 {
    font-size: 1.2rem; /*equivalente a 12px*/
}

p {
    font-size: 2.4rem; /*equivalente a 24px*/
}
```

A porcentagem é bastante utilizada quando estamos **desenvolvendo um layout responsivo e fluido.**

Ela permite que os módulos das páginas web se readaptem para ocupar a quantidade especificada. Por exemplo, **se definirmos a propriedade width de uma imagem para 50% da tela, essa imagem será redimensionada sempre que alteramos a tela de exibição.**

Observação: Layout responsivo é aquele que se adapta automaticamente ao tamanho da tela do usuário.

A medida **vw (viewport width)** é uma das medidas mais atuais, que ditam o futuro do CSS. Essa medida usa como referência a **largura da área de visualização (viewport)** da tela, na qual o site está sendo exibido. Assim, **1vw corresponde a a 1% da largura área de visualização da tela.**

Por sua vez, a medida **vh (viewport height)** é semelhante a medida **vw**, porém **a medida vh usa como referência à altura da área de visualização da tela, na qual o site está sendo exibido.** Assim, **1vh corresponde a 1% da altura da área de visualização da tela.**

A medida **vmin (viewport minimun)** também se **relaciona com as dimensões da área de visualização da tela**, porém o **vmin** utiliza com base a **menor dimensão entre a altura e a largura da tela**. Já a medida **vmax (viewport maximum)** terá como valor de referência a **maior dimensão da viewport (altura x largura)**.

Por exemplo, se tiver uma tela de **1600px de altura e 900px de largura**, **1vmin** corresponderá a **9px (1vmin = 900px/100)** e **1vmax** corresponderá a **16px (1vmax = 1600px/100)**.

Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

Crie o arquivo **HTML**.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="estilo.css">
    <title>CSS - Unidade de medida</title>
</head>

<body>
    <header id="showcase">
        <h1>Bem-vindo à praia!</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis quia fuga, dolore unde reprehenderit animi<br/>eos expedita perspiciatis tenetur.</p>
        <a href="#" class="btn">Leia mais</a>
    </header>
</body>
```

```

</header>
<section id="section-a">
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit nam obcaecati
    optio, commodi necessitatibus
        ullam aspernatur id voluptatem molestiae nostrum. In nobis, consectetur opti
    o deleniti, fugit, consequatur
        placeat amet error corporis eaque quas quo officiis odit! Architecto, magnam
.</p>
</section>
</body>

</html>

```

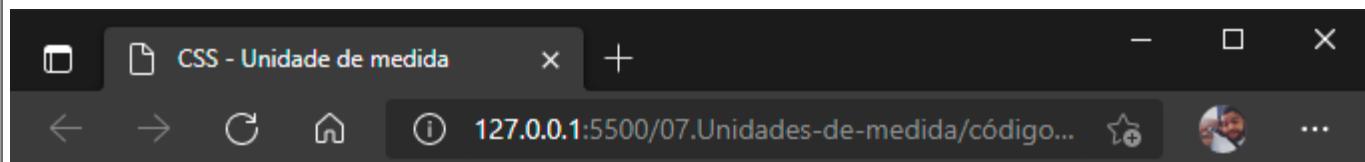
Vamos criar o arquivo **CSS**.

```

@charset "utf-8";
* {
    margin: 0;
    padding: 0;
}

body {
    margin: 0;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 17px;
    color: #926239;
    line-height: 1.6em;
}

```



Bem-vindo à praia

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quis quia fuga, dolore unde
 reprehenderit a animi eos expedita perspiciatis tenetur.

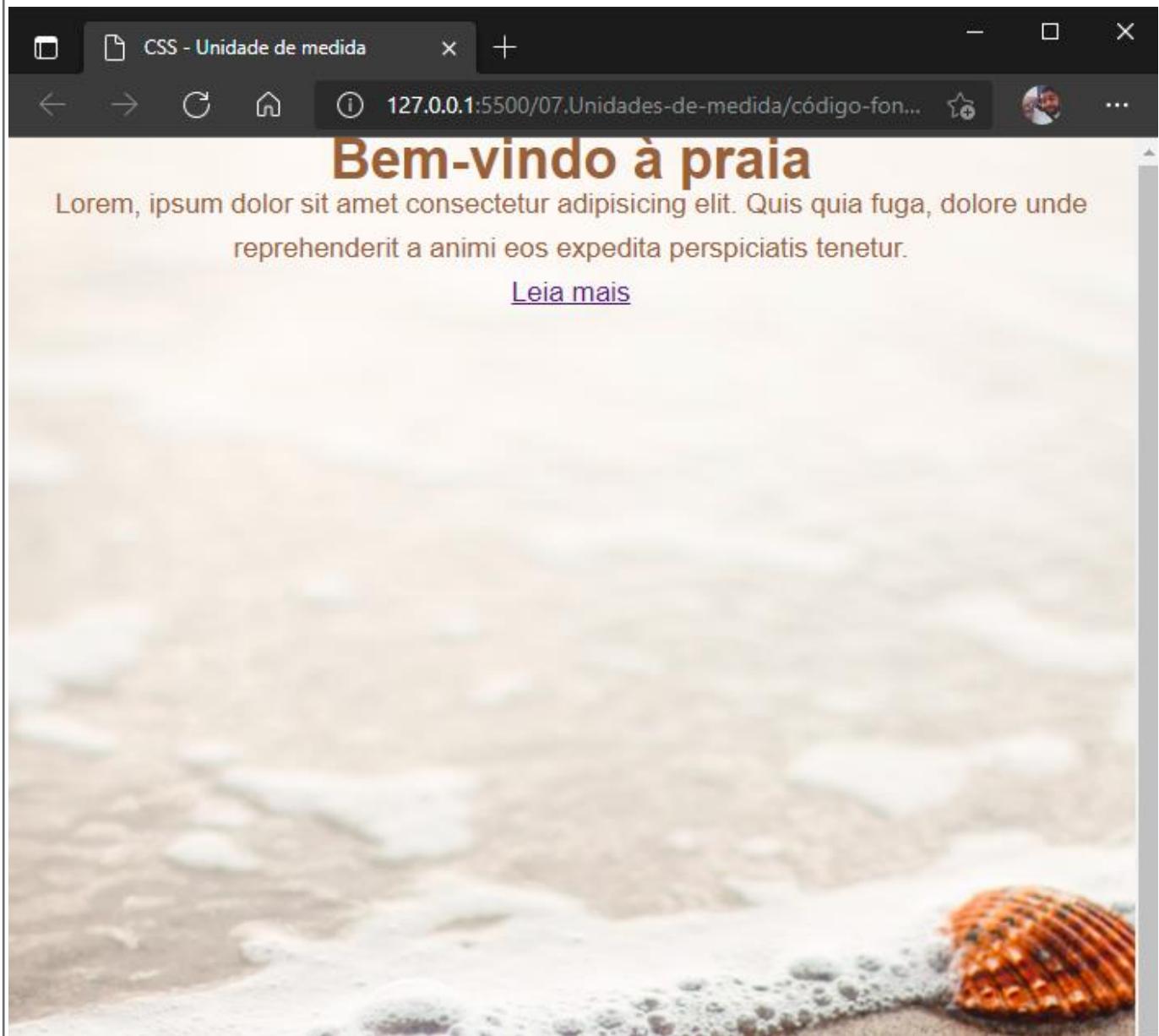
[Leia mais](#)

Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit nam obcaecati optio,
 commodi necessitatibus ullam aspernatur id voluptatem molestiae nostrum. In nobis,
 consectetur optio deleniti, fugit, consequatur placeat amet error corporis eaque quas quo
 officiis odit! Architecto, magnam.

Vamos, primeiramente, configurar a **id showcase** que está associada ao elemento **<header>** na página html. No arquivo **CSS**, insira o seguinte trecho de código:

```
#showcase {  
    background-image: url('01.beachshowcase.jpg');  
    background-size: cover;  
    background-position: center;  
    height: 100vh;  
    text-align: center;  
    padding: 0 20px;  
}
```

Esse trecho de código fará com que a imagem fique como plano de fundo (**propriedade background-image**) ocupe **100% da área de visualização da tela**. Isso acontece, devido a configuração da altura da **id #showcase** como **100vh** (propriedade **height: 100vh**).



Apenas para melhorar a visualização, vamos configurar os elementos **h1** e **parágrafo**, que pertencem a **id #showcase** no código HTML. No arquivo **CSS**, insira o seguinte trecho de código.

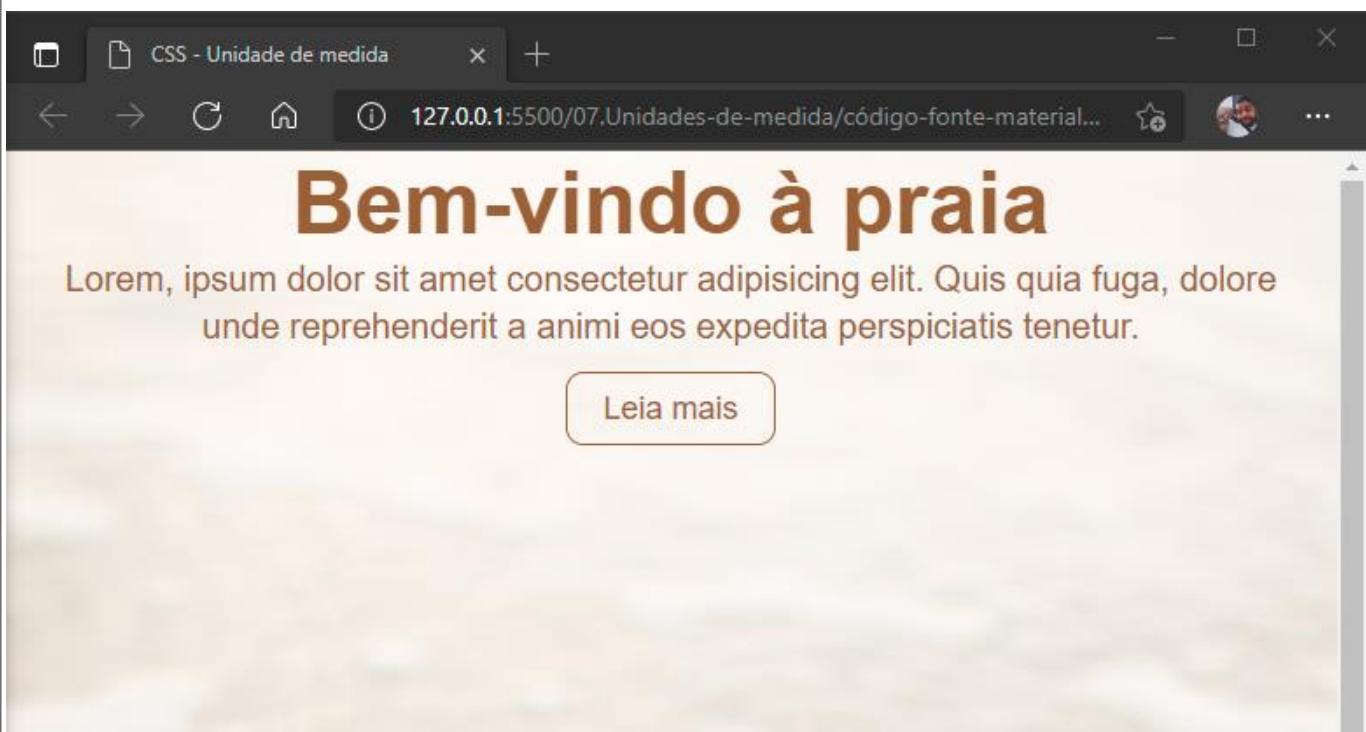
```
#showcase h1 {
    font-size: 50px;
    line-height: 1.2em;
}

#showcase p {
    font-size: 20px;
    margin-bottom: 20px;
}
```



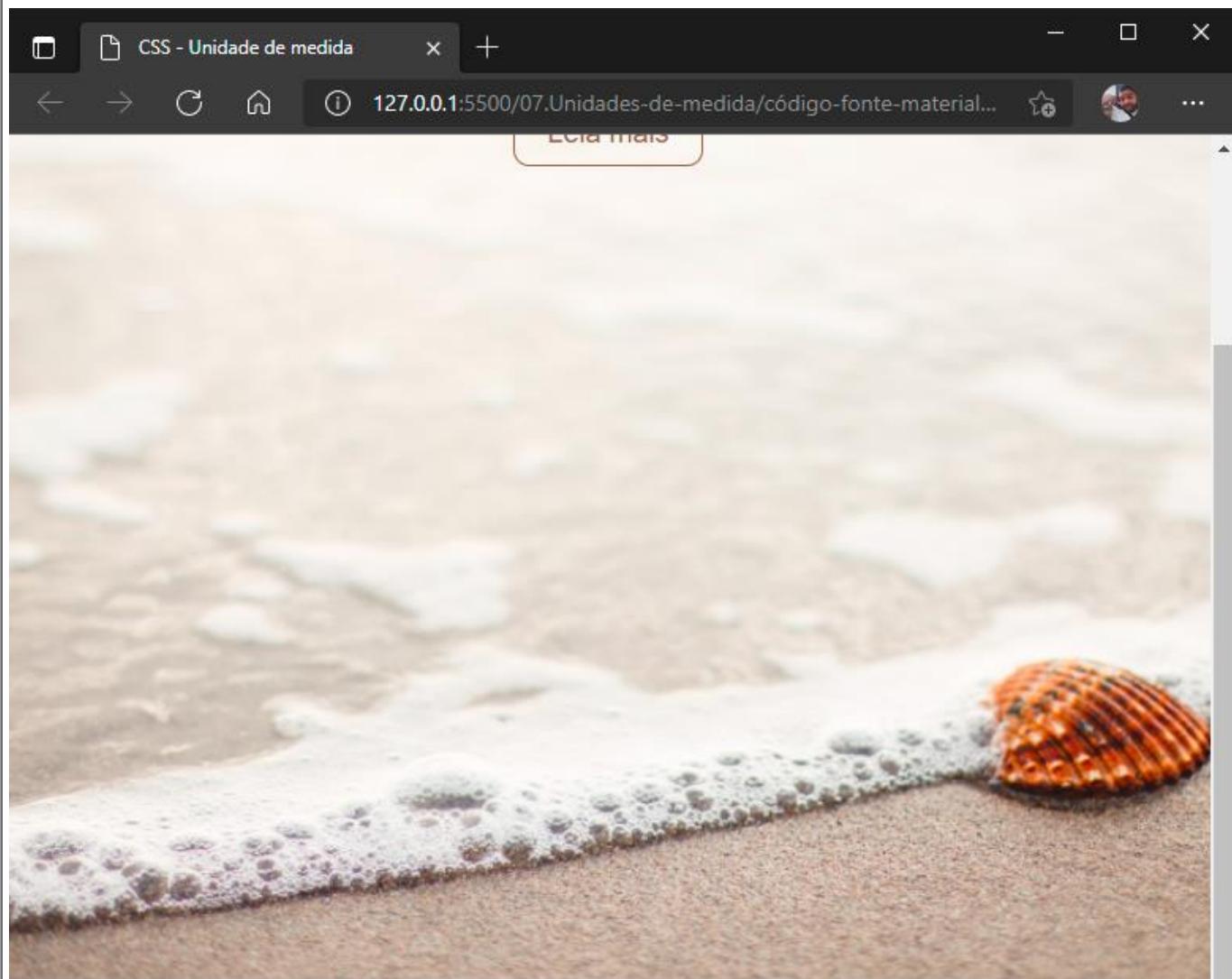
O resultado é a configuração do tamanho da fonte (propriedade **font-size**), a altura da linha do heading 1 (propriedade **line-height**) e a margem inferior do elemento parágrafo (propriedade **margin-bottom**). E a última configuração da **id #showcase** será a classe **.btn** que está associada ao link **Leia mais**. No arquivo CSS, insira o seguinte trecho de código:

```
#showcase .btn {
    font-size: 18px;
    text-decoration: none;
    color: #926239;
    border: #926239 1px solid;
    padding: 10px 20px;
    border-radius: 10px;
    margin-top: 20px;
}
```



Observe que o **link Leia mais** está **estilizado com um botão**. Por fim, vamos configurar a classe **.section-a**, que está associado ao primeiro elemento **<section>** na nossa página HTML. No arquivo **CSS**, insira o seguinte trecho de código.

```
#section-a {  
    padding: 20px;  
    background: #926239;  
    color: #fff;  
    text-align: center;  
}
```



CSS - Unidade de medida x

127.0.0.1:5500/07.Unidades-de-medida/código-fonte-material... ... mais

Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit nam obcaecati optio, commodi necessitatibus ullam aspernatur id voluptatem molestiae nostrum. In nobis, consectetur optio deleniti, fugit, consequatur placeat amet error corporis eaque quas quo officiis odit! Architecto, magnam.



Float display e box sizing

Os objetivos desta aula são:

- Compreender o uso do float e display CSS para posicionar elementos;
- Aplicar posicionamentos com float e display CSS.

Bons estudos!

CSS float

A propriedade **float** do **CSS** coloca um elemento do lado direito ou esquerdo de um **conteúdo**, permitindo que o texto e outros elementos alinhados envolvam esse elemento. A propriedade **float** pode ser configurada com alguns valores, por exemplo:

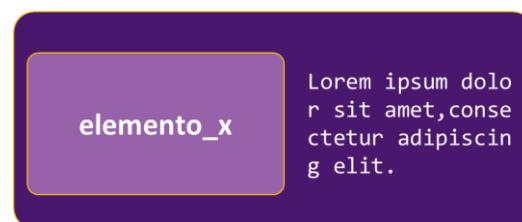
- **none**: esse é o valor padrão, não aplica a propriedade **float** no elemento.

```
elemento_x {
    float: none;
}
```



- **left**: o elemento é colocado à **esquerda** do **conteúdo**.

```
elemento_x {
    float: left;
}
```



- **right**: o elemento é colocado à **direita** do **conteúdo**.

```
elemento_x {
    float: right;
}
```



- **inherit**: o valor da propriedade **float** é o **mesmo** do **elemento pai**.

Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

Crie o arquivo **HTML**.

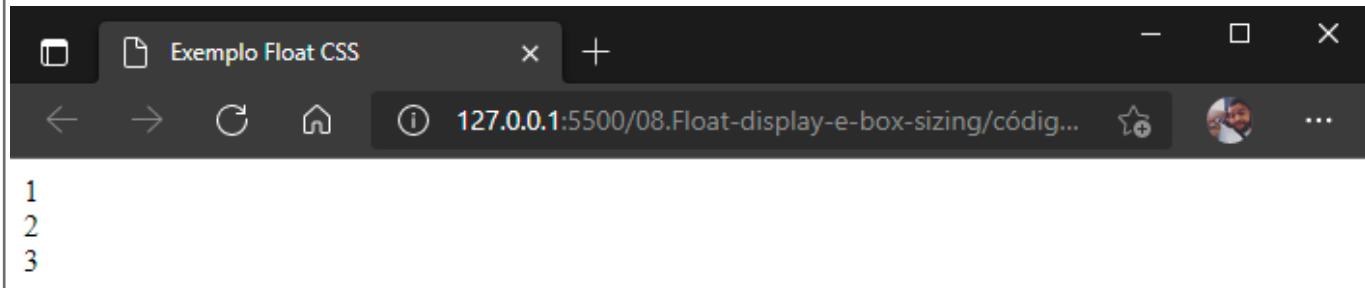
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="estilo.css">
    <title>Exemplo Float CSS</title>
</head>
<body>
    <section>
        <div class="esquerda">1</div>
        <div class="esquerda">2</div>
        <div class="direita">3</div>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  

            Morbi tristique sapien ac erat tincidunt, sit amet dignissim  

            lectus vulputate. Donec id iaculis velit. Aliquam vel  

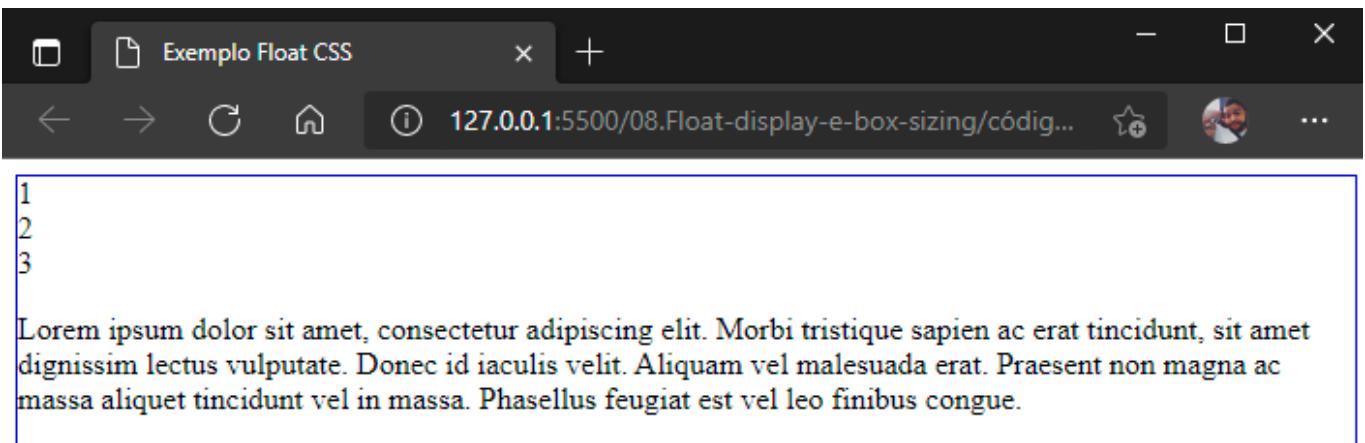
            malesuada erat. Praesent non magna ac massa aliquet tincidunt  

            vel in massa. Phasellus feugiat est vel leo finibus congue.</p>
    </section>
</body>
</html>
```



Vamos criar o arquivo **CSS** e começar a configurar os estilos dos elementos na página web para inserir **border**, definir a **largura** e a **propriedade float** do **elemento <section>**.

```
section {
    border: 1px solid blue;
    width: 100%;
    float: left;
}
```

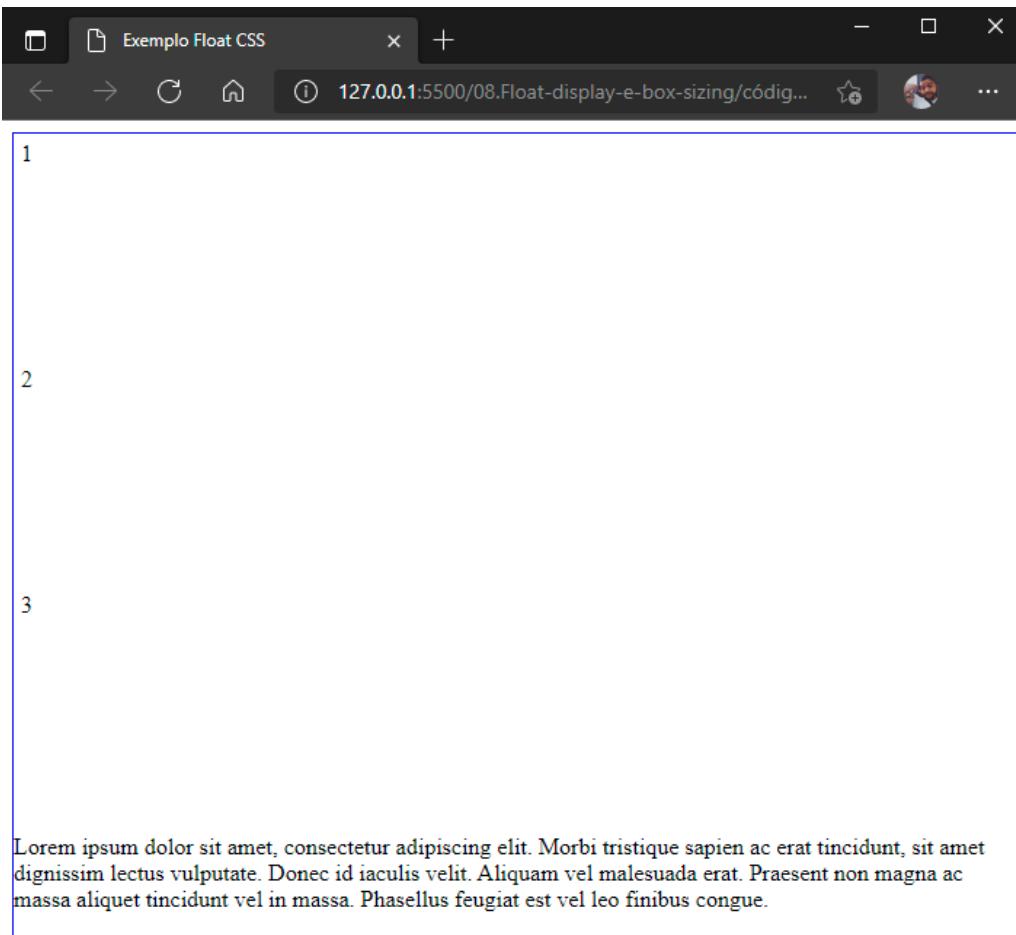


1
2
3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi tristique sapien ac erat tincidunt, sit amet dignissim lectus vulputate. Donec id iaculis velit. Aliquam vel malesuada erat. Praesent non magna ac massa aliquet tincidunt vel in massa. Phasellus feugiat est vel leo finibus congue.

Vamos, agora, completar o código do CSS para configurar as propriedades **margin**, **width** e **height** do elemento <div>.

```
div {  
    margin: 5px;  
    width: 50px;  
    height: 150px;  
}
```



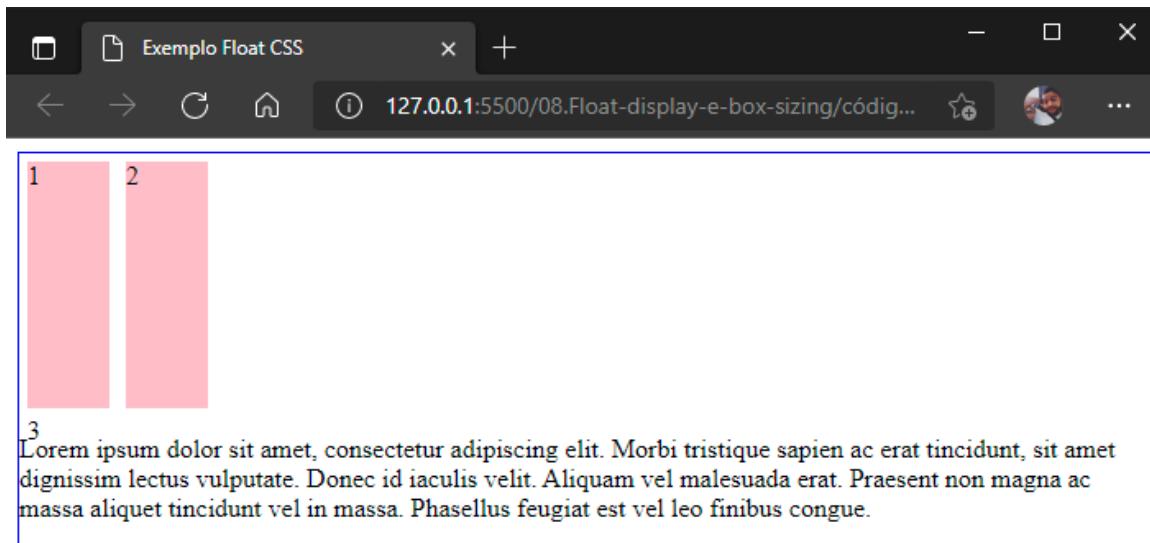
1
2
3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi tristique sapien ac erat tincidunt, sit amet dignissim lectus vulputate. Donec id iaculis velit. Aliquam vel malesuada erat. Praesent non magna ac massa aliquet tincidunt vel in massa. Phasellus feugiat est vel leo finibus congue.

Vamos, agora, configurar a classe **.esquerda** para colocar os elementos associados a essa classe à esquerda da página. No arquivo **CSS**, insira o seguinte trecho de código:

```
.esquerda {
    float: left;
    background: pink;
}
```

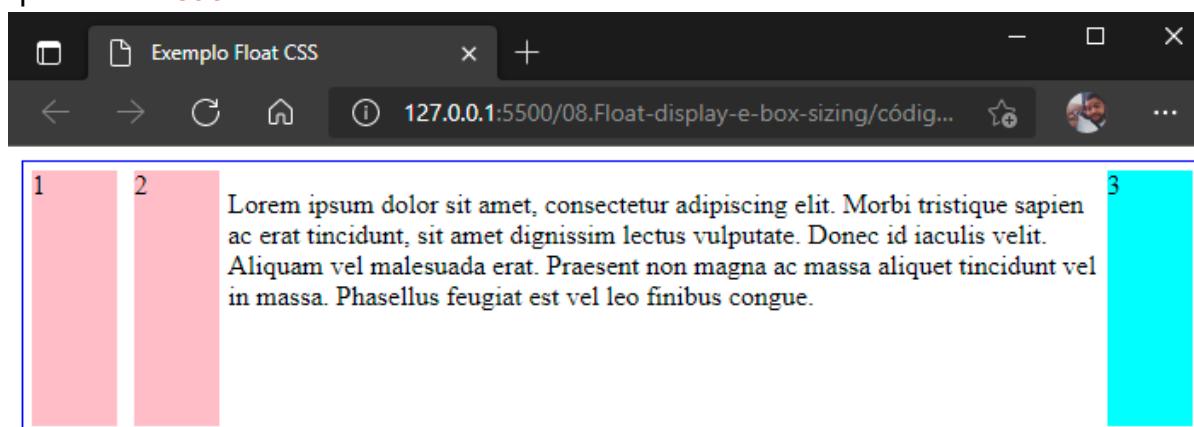
Salve o seu arquivo e veja o resultado no navegador dessa configuração.



Vamos, agora, configurar a classe **.direita** para colocar os elementos associados a essa classe à **esquerda** da página. No arquivo **CSS**, insira o seguinte trecho de código.

```
.direita {
    float: right;
    background: cyan;
}
```

Observe que os elementos estão posicionados na página de acordo com o valor de configuração da propriedade **float**.

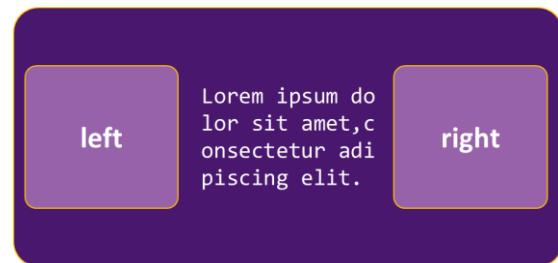


Clear float

Algumas vezes você pode querer forçar todos os elementos fiquem alinhados um embaixo do outros, para isso você pode usar a propriedade **clear**. A propriedade **clear** pode ser configurada com alguns valores, por exemplo:

- **none**: permite a propriedade **float dos dois lados**.

```
clear: none;
```



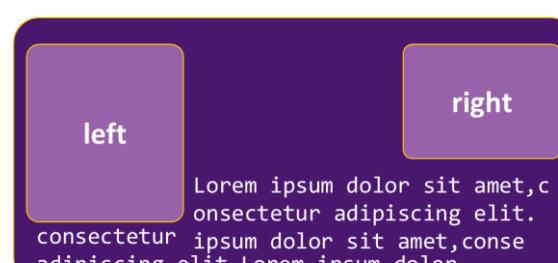
- **left**: sem float à esquerda.

```
clear: left;
```



- **right**: sem float à direita.

```
clear: right;
```



- **both**: sem float em ambos os lados.

```
clear: both;
```

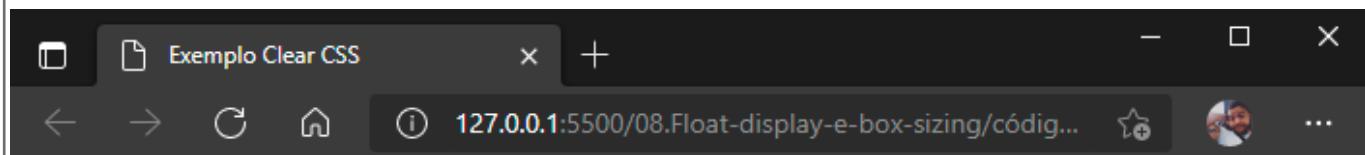


Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

Crie o arquivo **HTML**.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="estilo.css">
    <title>Exemplo Clear CSS</title>
</head>
<body>
    <h2>Sem clear</h2>
    <div class="div1">div1</div>
    <div class="div2">div2 - Note que div2 está depois do div1 no código HTML. Entretanto, uma vez que div1 floats está configurado como left, o texto div2 ficará em volta do div1.</div>
    <br><br>
    <h2>Com clear</h2>
    <div class="div3">div3</div>
    <div class="div4">div4 - Aqui, clear: left; move div4 para baixo do div3, que está com o float configurado com left.
        Isso acontece, porque a propriedade clear: left desabilita o float à esquerda.</div>
</body>
</html>
```



Sem clear

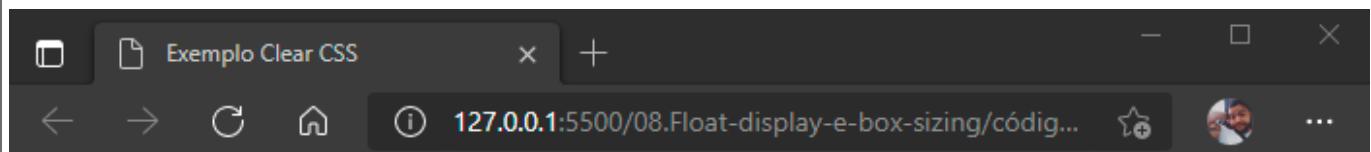
div1
 div2 - Note que div2 está depois do div1 no código HTML. Entretanto, uma vez que div1 floats está configurado como left, o texto div2 ficará em volta do div1.

Com clear

div3
 div4 - Aqui, clear: left; move div4 para baixo do div3, que está com o float configurado com left. Isso acontece, porque a propriedade clear: left desabilita o float à esquerda.

Vamos criar o arquivo **CSS** no **VS Code** e vamos utilizar a propriedade **clear**.

```
@charset "utf-8";
.div1 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}
.div2 {
    border: 1px solid red;
}
.div3 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}
.div4 {
    border: 1px solid red;
    clear: left;
}
```



Sem clear

div1

div2 - Note que div2 está depois do div1 no código HTML. Entretanto, uma vez que div1 floats está configurado como left, o texto div2 ficará em volta do div1.

Com clear

div3

div4 - Aqui, clear: left; move div4 para baixo do div3, que está com o float configurado com left. Isso acontece, porque a propriedade clear: left desabilita o float à esquerda.

CSS Box Sizing

A propriedade **box-sizing** permite incluir o espaçamento (**padding**) e a borda (**border**) na largura (**width**) e altura (**height**) total de um elemento, ou seja, essa propriedade faz com que o navegador entenda que o **padding** e a **border** fazem parte da largura e altura real do elemento na hora de calcular o tamanho do elemento. Os valores da propriedade **box-sizing** são:

conten-box: valor default, o valor de padding e de border são somados a altura e largura do elemento, assim ele ocupa um espaço maior.

```
largura (width) + padding + border = largura total do elemento  
altura (height) + padding + border = altura total do elemento
```

border-box: configura o navegador para considerar que padding e border já fazem parte da altura e da largura do elemento.

Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

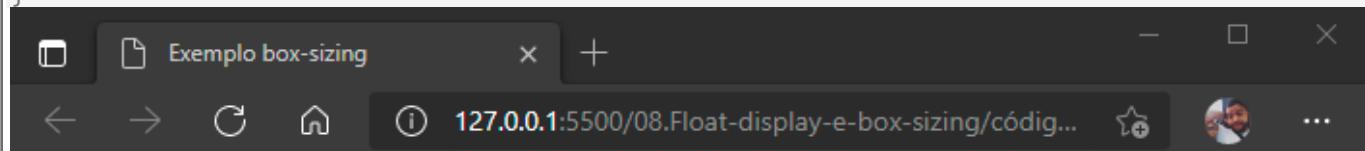
Crie o arquivo **HTML**.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Exemplo box-sizing</title>  
    <link rel="stylesheet" href="estilo.css">  
  
</head>  
<body>  
    <section class="section1">Esse elemento section foi configurado com largura 300px e  
    altura 200px.</section>  
    <br>  
    <section class="section2">Esse elemento section foi configurado com largura 300px e  
    altura 200px, mas o navegador  
        está calculando largura total = 351 (width = 300 + padding = 50 + border = 1) e  
    altura total = 251 (height = 200  
        + padding = 50 + border = 1).</section>  
</body>  
</html>
```

Vamos criar o arquivo **CSS** no **VS Code**.

```
@charset "utf-8";
.section1 {
    width: 300px;
    height: 200px;
    border: 1px solid blue;
}

.section2 {
    width: 300px;
    height: 200px;
    padding: 50px;
    border: 1px solid red;
}
```



Esse elemento section foi configurado com largura 300px e altura 200px.

Esse elemento section foi configurado com largura 300px e altura 200px, mas o navegador está calculando largura total = 351 (width = 300 + padding = 50 + border = 1) e altura total = 251 (height = 200 + padding = 50 + border = 1).

Observe que os dois elementos <section> tem valores iguais de **altura (height)** e **largura (width)**, mas como a **box-sizing** está com o valor default (**content-box**), o segundo <section> ocupa um valor bem maior do que o primeiro.

Para resolver esse problema basta atualizar os arquivos:

HTML

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Exemplo box-sizing</title>
    <link rel="stylesheet" href="estilo.css">

</head>

<body>
    <section class="section1">Esse elemento section foi configurado com largura 300px e altura 100px.</section>
    <br>
    <section class="section2">Esse elemento section foi configurado com largura 300px e altura 100px.</section>
</body>

</html>
```

CSS

```
@charset "utf-8";
.section1 {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
}

.section2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
    box-sizing: border-box;
}
```

CSS display

A propriedade display define a **forma como um elemento HTML será renderizado na página**, ou seja, configura o **comportamento de exibição de um elemento**. Essa propriedade determina o fluxo e estruturação de uma página web.

Existem diversos valores, que você pode usar para essa propriedade, alguns desse viram legado no CSS3, mas não significa que pararam de funcionar. A lista completa dos valores você pode conferir nos sites:

<https://developer.mozilla.org/en-US/docs/Web/CSS/display>

https://www.w3schools.com/cssref/pr_class_display.asp.

A tabela a seguir mostra alguns desses valores disponíveis para usar na propriedade display:

Valor	Descrição
inline	Exibe um elemento em linha (inline) a outro elemento.
block	Exibe um elemento com um bloco (block).
inline-block	Exibe o elemento como um bloco em linha.
none	Remove o elemento.
flex	Exibe um elemento como um container flex block-level
inline-flex	Exibe o elemento como um container flex block-level em linha
grid	Exibe um elemento como um container grid block-level
inline-grid	Exibe um elemento como um container grid block-level em linha

Vamos praticar

Vamos criar um arquivo **HTML** e o arquivo de estilo **CSS** na sua página web. Siga os passos para criarmos a página de internet e aplicarmos o estilo CSS:

Crie o arquivo **HTML**.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Exemplo Propriedade Display</title>
    <link rel="stylesheet" href="estilo.css">
```

```
</head>

<body>
    <h1>Propriedade display</h1>

    <h2>display: none:</h2>
    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at er
        at pulvinar, at pulvinar felis
        blandit. <p class="ex-
1">HELLO WORLD!</p>Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.
        </div>

    <h2>display: inline:</h2>
    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at er
        at pulvinar, at pulvinar felis
        blandit. <p class="ex-
2">HELLO WORLD!</p> Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut
        .
        </div>

    <h2>display: block:</h2>
    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at er
        at pulvinar, at pulvinar felis
        blandit. <p class="ex-
3">HELLO WORLD!</p> Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut
        .
        </div>

    <h2>display: inline-block:</h2>
    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at er
        at pulvinar, at pulvinar felis
        blandit. <p class="ex-
4">HELLO WORLD!</p> Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut
        .
        </div>
    </div>

</html>
```

Crie o arquivo **CSS**.

```
@charset "utf-8";
p {
    color: red;
}

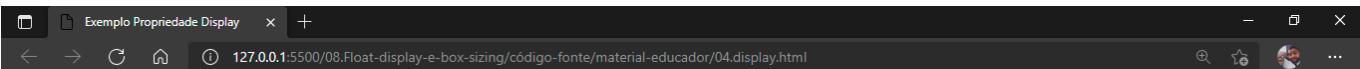
p.ex-1 {
    display: none;
}

p.ex-2 {
    display: inline;
}

p.ex-3 {
    display: block;
}

p.ex-4 {
    display: inline-block;
}
```

Observe que a propriedade **display** com o valor **none** ocultou o conteúdo **Hello World!**, com o valor **inline** manteve o conteúdo na mesma linha, com o valor **block** quebrou a linha e criou um bloco para o conteúdo e com o valor **inline-block** manteve na mesma linha, mas o espaçamento entre as linhas ficou diferente por causa do bloco.



Propriedade display

display: none:

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

display: inline:

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. **HELLO WORLD!** Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

display: block:

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit.

HELLO WORLD!

 Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

display: inline-block:

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. **HELLO WORLD!** Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.



Pseudoclasses e Pseudoelementos

Os objetivos desta aula são:

- Compreender o uso de pseudoclasses e pseudoelementos;
- Aplicar o conceito de pseudoclasses e pseudoelementos nas folhas de estilo.

Bons estudos!

Pseudoclasses

Por diversas vezes, você está em uma página da internet e encontra elementos da página, como **links**, que **mudam de cor no momento que passamos o mouse sobre eles**. Isso pode ser feito utilizando **pseudoclasses no CSS**.

Pseudoclasse é uma palavra-chave usada para **aplicar um efeito especial a um seletor**, ou seja, ela define um estado especial para um determinado elemento HTML. A pseudoclasse seleciona elementos para aplicar uma determinada ação, propriedade etc. A sintaxe de pseudoclasse é:

```
seletor:pseudoclasse {  
    property: value;  
}
```

As principais pseudoclasses do **CSS** são:

Seletor	Exemplo	Descrição do Exemplo
:active	a:active	Seleciona links ativos (não visitados)
:checked	input:checked	Seleciona todo elemento <input> ativado (checked)
:disabled	input:disabled	Seleciona todo elemento <input> desabilitado
:empty	p:empty	Seleciona todo elemento <p> não tem filhos
:enabled	input:enabled	Seleciona todo elemento <input> habilitado
:first-child	p:first-child	Seleciona todo elemento <p> que é o seu elemento pai
:first-of-type	p:first-of-type	Seleciona todo elemento <p> que é o primeiro elemento <p> de um determinado elemento pai
:focus	input:focus	Seleciona o elemento <input> que recebe foco
:hover	a:hover	Seleciona links que o mouse passar por cima.
:in-range	input:in-range	Seleciona elementos <input> com um valor dentro um uma faixa específica
:invalid	input:invalid	Seleciona todo elemento <input> com valores inválidos
:lang(<i>language</i>)	p:lang(it)	Seleciona todo elemento <p> com o atributo lang iniciado on "it"
:last-child	p:last-child	Seleciona o elemento <p> que é o último filho de um pai
:last-of-type	p:last-of-type	Seleciona todo elemento <p> que é o último elemento <p> do elemento pai
:link	a:link	Seleciona todos os links não visitados
:not(selector)	:not(p)	Seleciona todo elemento que não é um elemento <p>
:nth-child(n)	p:nth-child(2)	Seleciona todo elemento <p> que é o Segundo filho de um elemento pai
:nth-last-child(n)	p:nth-last-child(2)	Seleciona todo elemento <p> que é o segundo filho desse elemento pai, contando a partir do último elemento
:nth-last-of-type(n)	p:nth-last-of-type(2)	Seleciona todo elemento <p> que é o segundo elemento <p> de um determinado pai, contando a partir do último elemento
:nth-of-type(n)	p:nth-of-type(2)	Seleciona todo elemento <p> que é o segundo elemento de um determinado pai

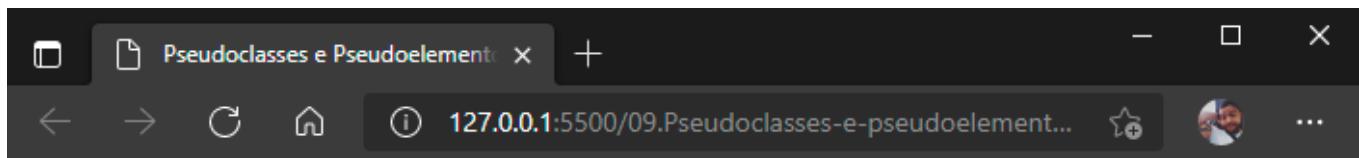
:only-of-type	p:only-of-type	Seleciona todo elemento <p> que não possui irmãos do mesmo tipo <p> de um determinado pai
:only-child	p:only-child	Seleciona todo elemento <p> sem nenhum elemento-irmão
:optional	input:optional	Seleciona elementos <input> sem o atributo "required"
:out-of-range	input:out-of-range	Seleciona elementos <input> com um valor for a de uma específica faixa
:read-only	input:read-only	Seleciona elementos <input> com o atributo "readonly" especificado
:read-write	input:read-write	Seleciona elementos <input> sem atributo "readonly" especificado
:required	input:required	Seleciona elementos <input> com o atributo "required" especificado
:root	root	Seleciona o elemento root do documento
:target	#news:target	Representa um único elemento (o elemento alvo) com um id (#news) correspondente ao fragmento da URL.
:valid	input:valid	Representa qualquer <input> ou outro elemento do <form> cujo conteúdo foi validado com sucesso.
:visited	a:visited	Seleciona todos os links visitados

Vamos ver alguns exemplos

O seguinte seletor marca todo elemento <p> que é o primeiro filho de outro elemento. Observe que o primeiro elemento <p> de qualquer elemento pai está com **a cor da fonte azul** por causa da **pseudoclasse** define no <style> CSS.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Pseudoclasses</title>
    <style>
        p:first-child {
            color: blue;
        }
    </style>
</head>
<body>
    <section>
        <p>Primeiro elemento &lt;p&gt; filho do elemento
           &lt;section&gt;.</p>
        <p>Parágrafo qualquer.</p>
    </section>
    <footer>
        <p>Primeiro elemento &lt;p&gt; filho do elemento
           &lt;footer&gt;.</p>
        <p>Parágrafo qualquer.</p>
    </footer>
</body>
</html>
```

Resultado:

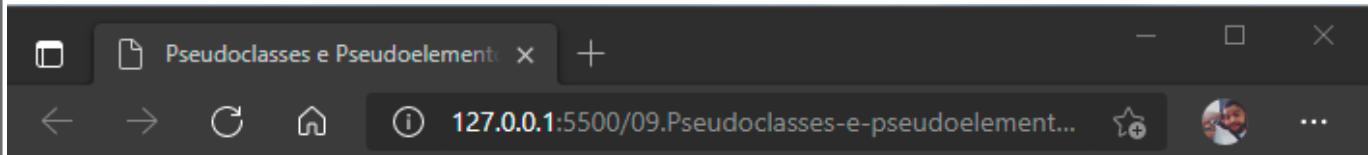


O seguinte seletor **altera a cor de plano de fundo dos elementos ** filhos da lista não ordenada que foram selecionados no estilo **CSS**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Pseudoclasses</title>
    <style>
        .mylist li:first-child {
            background: greenyellow;
        }
        .mylist li:nth-child(3) {
            background: grey;
        }

        .mylist li:last-child {
            background: yellow;
        }
    </style>
</head>
<body>
    <section>
        <p>Lista não ordenada colorida</p>
        <ul class="mylist">
            <li>Primeiro filho</li>
            <li>Segundo filho</li>
            <li>Terceiro filho</li>
            <li>Quarto filho</li>
            <li>Último filho</li>
        </ul>
    </section>
</body>
</html>
```

Resultado:



Lista não ordenada colorida

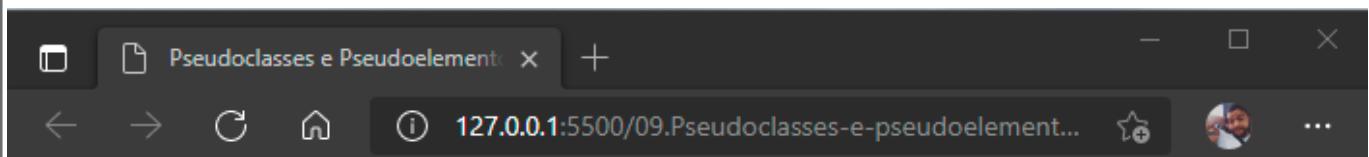
- Primeiro filho
- Segundo filho
- Terceiro filho
- Quarto filho
- Último filho

O seguinte seletor **altera a cor de plano de fundo do elemento <a> quando o mouse passa por cima dele.**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Pseudoclasses</title>
    <style>
        a:hover {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <a href="https://ios.org.br/">IOS</a>
    <a href="https://www.wikipedia.org">wikipedia.org</a>

    <p><strong>Nota:</strong> A pseudoclasse :hover aplica o estilo quando o mouse passa por cima do elemento.</p>
</body>
</html>
```

Resultado:



IOS [wikipedia.org](https://www.wikipedia.org)

Nota: A pseudoclasse :hover aplica o estilo quando o mouse passa por cima do elemento.

Pseudoelementos

Pseudoelemento é uma palavra-chave usada para estilizar uma parte específica de um determinado elemento. A sintaxe do pseudoelemento é:

```
seletor::pseudoelemento {  
    property: value;  
}
```

Os principais pseudoelementos do **CSS** são:

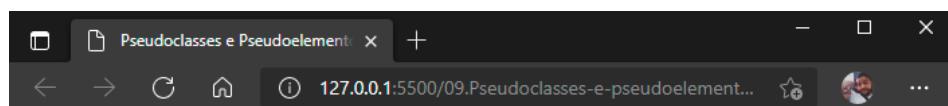
Seletor	Exemplo	Descrição do Exemplo
::after	p::after	Inserir um conteúdo depois de todo elemento <p>
::before	p::before	Inserir um conteúdo antes de todo elemento <p>
::first-letter	p::first-letter	Seleciona a primeira letra de todo elemento <p>
::first-line	p::first-line	Seleciona a primeira linha de todo elemento <p>
::selection	p::selection	Seleciona uma porção de um elemento que é selecionada pelo usuário.

Vamos ver alguns exemplos

O seguinte seletor **insere um conteúdo antes de depois de todo elemento <p>**.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Pseudoelementos</title>  
    <style>  
        p::before {  
            content: "Leia isso - ";  
        }  
  
        p::after {  
            content: " - Leia Mais";  
        }  
    </style>  
</head>  
<body>  
    <p>Meu nome é Donald</p>  
    <p>Eu moro em in Ducksburg</p>  
</body>  
</html>
```

Resultado:



Leia isso - Meu nome é Donald - Leia Mais

Leia isso - Eu moro em in Ducksburg - Leia Mais

O seguinte seletor **altera a primeira letra de todo parágrafo.**

```
<!DOCTYPE html>
<html lang="pt-br">

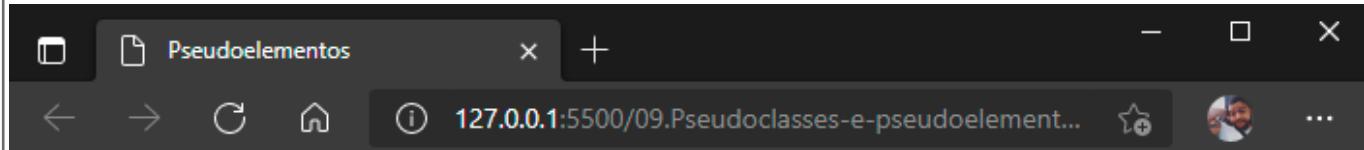
<head>
    <meta charset="UTF-8">
    <title>Pseudoelementos</title>
    <style>
        p::first-letter {
            font-size: 200%;
            color: #8A2BE2;
        }
    </style>
</head>

<body>
    <h1>Bem-vindo</h1>

    <p>Meu nome é Donald.</p>
    <p>Eu moro em Duckburg.</p>
    <p>Meu melhor amigo é o Mickey.</p>
</body>

</html>
```

Resultado:



Bem-vindo

Meu nome é Donald.

Eu moro em Duckburg.

Meu melhor amigo é o Mickey.

O seguinte seletor **altera a cor de plano de fundo da primeira linha de todo parágrafo.**

Código:

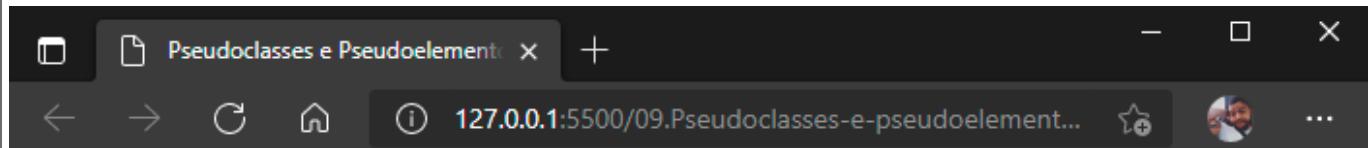
```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Pseudoelementos</title>
    <style>
        p::first-line {
            background-color: yellow;
        }
    </style>
</head>

<body>
    <h1>IOS</h1>
    <p>Lorem ipsum dolor sit amet consectetur, adipisicing
        elit. Veritatis dolores vero consectetur ea rem similique
        quas sint nostrum error tempore.</p>
</body>

</html>
```

Resultado:



IOS

Lore*rem ipsum dolor sit amet consectetur, adipisicing elit. Veritatis dolores vero consectetur ea rem similique quas sint nostrum error tempore.*

O seguinte seletor determina **a cor da fonte e do fundo para o conteúdo de um elemento selecionado com o mouse**.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Pseudoelementos</title>
    <style>
        ::selection {
            color: gold;
        }
    </style>
</head>
```

```

        background-color: red;
    }

    p::selection {
        color: white;
        background-color: blue;
    }

```

</style>

</head>

<body>

<h1>Selecione com o mouse parte do texto:</h1>

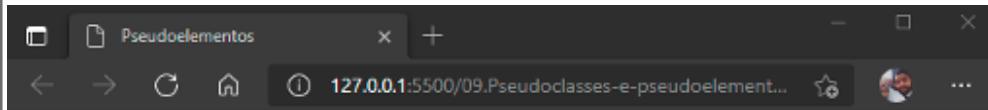
<p>Esse é um parágrafo.</p>

<div>Esse é um texto do elemento div.</div>

</body>

</html>

Resultado



Selecione com o mouse parte do texto:

Esse é um parágrafo.

Esse é um texto do elemento div.

Vamos praticar

Vamos criar um arquivo HTML, nesse exemplo vamos usar o estilo CSS declarado no documento HTML para ficar mais simples de explicar e atualizar o arquivo.

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Pseudoclasses e Pseudoelementos</title>
    <link rel="stylesheet" href="03.estilo.css">

```

</head>

<body>

<section class="container">

```

<div class="box">
    <h1>Olá pessoal!</h1>
    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Reiciendis,  

dolore necessitatibus molestias  

        assumenda soluta quis cum repellat quidem voluptas veritatis!  

Repudiandae, officia adipisci minima  

        cupiditate tempora consequuntur totam quibusdam eos delectus atque  

tenetur nesciunt sit! Maxime,  

        reiciendis perspiciatis praesentium quidem nostrum natus optio dolorem  

fugiat doloribus ullam expedita!  

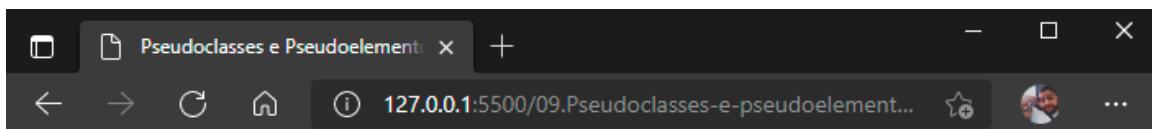
        Maxime, deserunt!  

    </p>
    <a class="btn" href="#">Leia Mais</a>
</div>

<div class="categorias">
    <h2>Categorias</h2>
    <ul>
        <li><a href="#">Categoria 1</a></li>
        <li><a href="#">Categoria 2</a></li>
        <li><a href="#">Categoria 3</a></li>
        <li><a href="#">Categoria 4</a></li>
    </ul>
</div>
</section>
</body>

</html>

```



Olá pessoal!

Placeholder text (Lorem ipsum) describing the page's purpose.

[Leia Mais](#)

Categorias

- [Categoria 1](#)
- [Categoria 2](#)
- [Categoria 3](#)
- [Categoria 4](#)

Observe que a página está totalmente sem formatação, vamos configurar alguns **estilos**.

Vamos criar o arquivo **CSS**.

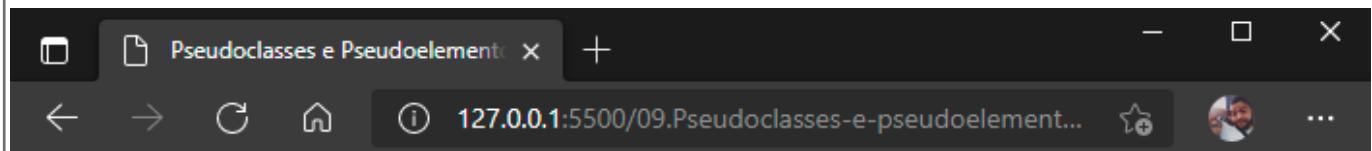
```
@charset "utf-8";
body {
    background-color: #f4f4f4;
    color: #555555;
    font: normal 16px Arial, Helvetica, sans-serif;
    line-height: 1.6em;
    margin: 0;
}
```

/ Observe que o usar a configuração da propriedade font dessa forma:*

```
font: normal 16px Arial, Helvetica, sans-serif;
```

É o mesmo que fazer separadamente cada uma das propriedades, dessa forma:

```
font-family: Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: normal; */
```



Olá pessoal!

Latin text:
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Reiciendis, dolore necessitatibus molestias assumenda soluta quis cum repellat quidem voluptas veritatis! Repudiandae, officia adipisci minima cupiditate tempora consequuntur totam quibusdam eos delectus atque tenetur nesciunt sit! Maxime, reiciendis perspiciatis praesentium quidem nostrum natus optio dolorem fugiat doloribus ullam expedita! Maxime, deserunt!

[Leia Mais](#)

Categorias

- [Categoria 1](#)
- [Categoria 2](#)
- [Categoria 3](#)
- [Categoria 4](#)

Agora, vamos atualizar o **estilo** dos **links** da página. Insira o seguinte código no arquivo `estilo.css`.

```
a {  
    text-decoration: none;  
    color: black;  
}  
  
a:hover {  
    color: red;  
}  
  
a:active {  
    color: green;  
}
```

Observe que os links **não estão mais sublinhados e nem na cor azul** e quando você passar o mouse sobre eles a cor da fonte muda para vermelho e quando você clica fica verde.

Categorias

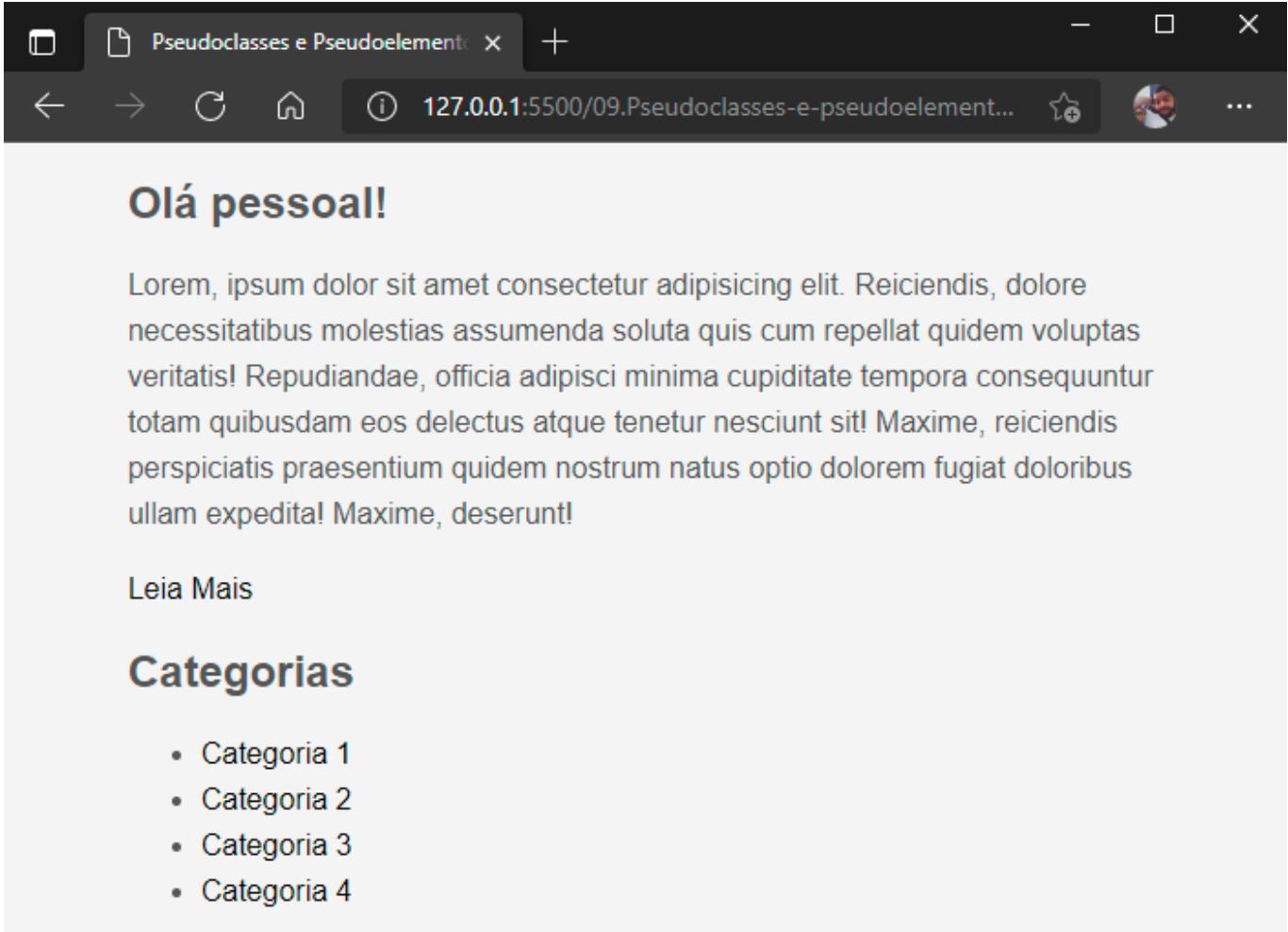
- Categoria 1
- Categoria 2
- Categoria 3
- Categoria 4

Categorias

- Categoria 1
- Categoria 2
- Categoria 3
- Categoria 4

Agora, vamos atualizar a classe **.container** que está associada ao elemento **<section>**.

```
.container {  
    width: 80%;  
    margin: auto;  
}
```



Olá pessoal!

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Reiciendis, dolore necessitatibus molestias assumenda soluta quis cum repellat quidem voluptas veritatis! Repudiandae, officia adipisci minima cupiditate tempora consequuntur totam quibusdam eos delectus atque tenetur nesciunt sit! Maxime, reiciendis perspiciatis praesentium quidem nostrum natus optio dolorem fugiat doloribus ullam expedital! Maxime, deserunt!

[Leia Mais](#)

Categorias

- Categoria 1
- Categoria 2
- Categoria 3
- Categoria 4

Observe que o conteúdo agora está com uma **margem e ocupa uma largura de 80% da área visível do navegador.**

Agora, vamos atualizar a classe **.btn** que está associada ao elemento **<a>** com o **conteúdo Leia Mais.**

```
.btn {  
background-color: #333;  
color: #fff;  
padding: 10px 15px;  
border: none;  
}  
  
.btn:hover {  
background: red;  
color: #FFF;  
}
```



Olá pessoal!

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Reiciendis, dolore necessitatibus molestias assumenda soluta quis cum repellat quidem voluptas veritatis! Repudiandae, officia adipisci minima cupiditate tempora consequuntur totam quibusdam eos delectus atque tenetur nesciunt sit! Maxime, reiciendis perspiciatis praesentium quidem nostrum natus optio dolorem fugiat doloribus ullam expedital! Maxime, deserunt!

[Leia Mais](#)

Categorias

- Categoria 1
- Categoria 2
- Categoria 3
- Categoria 4

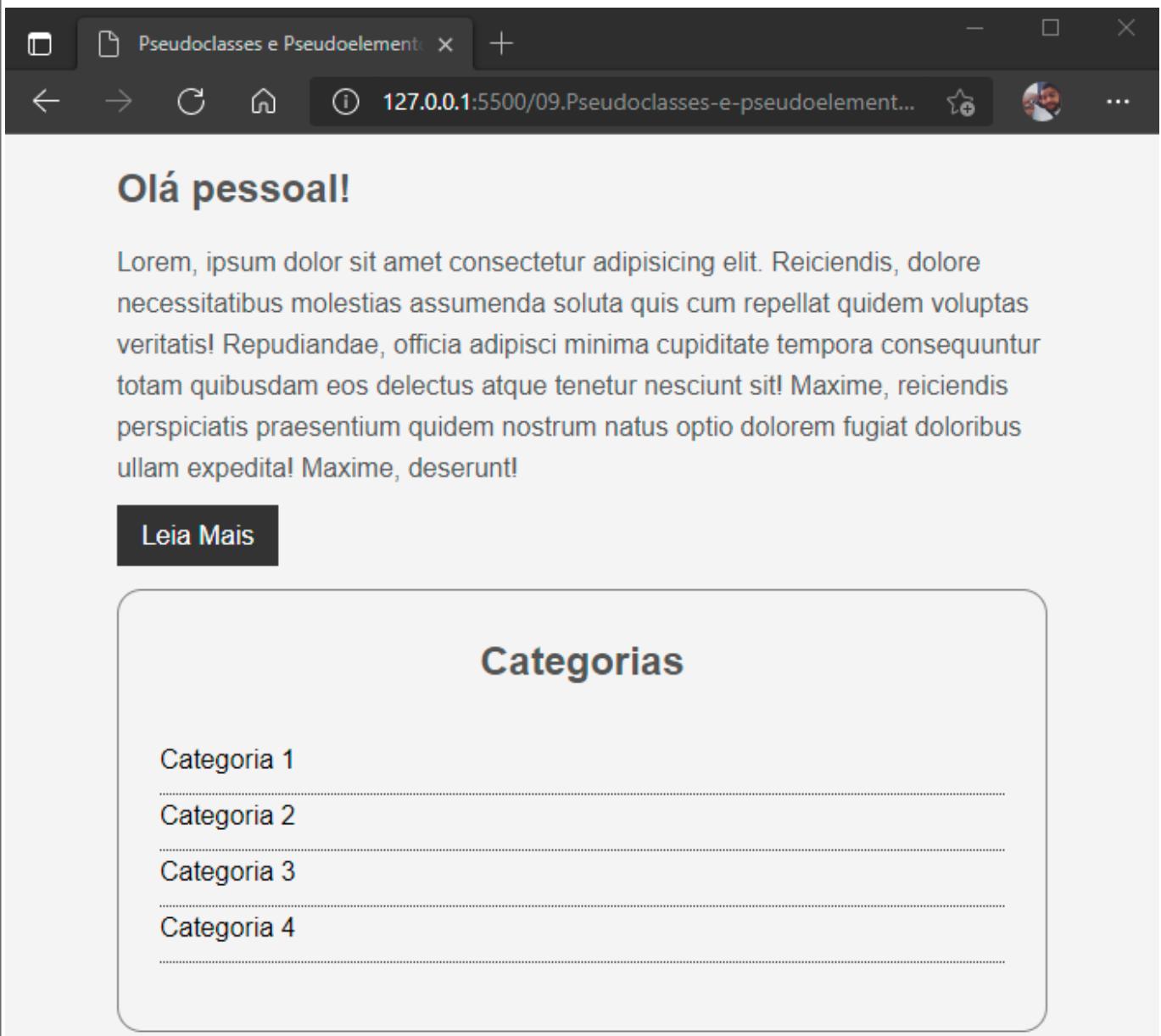
Observe que o conteúdo agora virou um **botão que muda a cor para vermelho**, quando você **passa o mouse por cima dele**. Por fim, vamos configurar a classe **.categorias**. Insira o seguinte código no arquivo **CSS**.

```
.categorias {
    margin-top: 20px;
    border: 1px gray solid;
    padding: 10px;
    border-radius: 15px;
}

.categorias h2 {
    text-align: center;
}

.categorias ul {
    padding: 15px;
    list-style: none;
}

.categorias li {
    padding-bottom: 8px;
    border-bottom: dashed 1px #333;
}
```



The screenshot shows a web browser window with the title "Pseudoclasses e Pseudoelementos". The page content includes a heading "Olá pessoal!" followed by a large amount of placeholder text (Lorem ipsum). Below this is a "Leia Mais" button. A rounded rectangular box contains the heading "Categorias" and a list of four categories: "Categoria 1", "Categoria 2", "Categoria 3", and "Categoria 4", each separated by a horizontal dotted line.

Observe que a área delimitada pelo elemento `<div>` que está associado à classe `categorias` está com a barda e `padding` configurados e os elementos `<h2>`, `` e `` também são estilizados na sequência.



Media Query

Os objetivos desta aula são:

- Compreender o uso de At-Rules;
- Compreender o uso de Media Query;
- Aplicar diferentes formatações conforme o tamanho do Viewport.

Bons estudos!

At-rules

O CSS tem diversas regras, que são denominadas **At-Rules**, que são declarações com **instruções de como o CSS irá se comportar**. **At-rules** são iniciadas com o sinal de arroba (@), seguido por um identificador e **terminando com um ponto-e-vírgula (;)**. Existem muitas at-rules regulares, que foram implementadas com seu identificador, cada uma com uma sintaxe diferente, por exemplo:

- **@charset**: define o conjunto de caracteres (codificação) usado pelo CSS.
- **@import**: informa ao CSS para incluir uma folha de estilos externa.
- **@namespace** (en-US): informa ao CSS que todo o conteúdo deve ser considerado prefixado com um namespace XML.

Existe também um subconjunto de declarações, que pode ser usado como uma regra de uma folha de estilos e, também, dentro de um grupo de regras condicional:

- **@media**: um grupo de regras condicional que é aplicado ao conteúdo caso o dispositivo de exibição atenda com a condição de mídia definida (media query).
- **@document** (en-US): um grupo de regras condicional que é aplicado ao conteúdo caso o documento atenda aos critérios da condição definidos.
- **@page**: descreve o aspecto das mudanças do layout que serão aplicadas ao imprimir o documento
- **@font-face**: descreve o aspecto de uma fonte externa que deve ser baixada.
- **@supports** (en-US): um grupo de regras condicional que é aplicado ao conteúdo caso o navegador atenda aos critérios da condição definidos.
- **@keyframes**: descreve o aspecto das etapas intermediárias em uma sequência de animação CSS. Ele é muito usado para declarar variáveis em CSS (spoiler das próximas aulas).
- **@viewport**: descreve os aspectos da área de visualização para dispositivos com tela pequena.
- **@counter-style**: define estilos de contador específicos que não fazem parte do conjunto predefinido de estilos. (no estágio de recomendação do candidato, mas implementado apenas no Gecko no momento da escrita)
- **font-feature-values** (plus @swash, @ornaments, @annotation, @stylistic, @stylesheet w @character-variant): define nomes comuns na font-variant-alternates para recurso ativado de forma diferente no OpenType.
- **@property**: descreve o aspecto de propriedades e variáveis personalizadas.
- **@color-profile**: permite que um perfil de cores seja definido para uso da função color().

Media query

Atualmente, os sites devem se adaptar a diferentes mídias (computador, tablet, celulares, etc.) e é importante que a informação exibida no site seja enquadrada corretamente nas diferentes telas de exibição. Isso é o que chamamos de web design responsivo. Por exemplo, algumas larguras de tela comuns são:

Smartphones: 320px

Tablets: 768px

Netbook e modo de visualização landscape(paisagem) do tablet: 1024px

Desktop com tela mais larga: 1600px

Portanto media query é **uma definição do tipo de mídia (como telas) e de uma expressão lógica que determina a resolução e orientação (Portrait [retrato] ou landscape [paisagem]) da tela de um dispositivo para exibição de um conteúdo na página web.** Quando a media query é avaliada como **verdadeira**, ela direciona os **navegadores para a parte do código CSS** com a configuração específica para esse recurso.

@media

A regra **@media** é usada para aplicar estilos com base no resultado de uma ou mais consultas de mídia e é capaz de testar características específicas de um dispositivo, ou seja, é usada em media query para aplicar diferentes estilos para diferentes tipos de mídias e dispositivos.

Ela pode ser usada para testar muitas coisas, tais como:

Largura e altura da área de visualização.

Largura e altura de um dispositivo.

Orientação, por exemplo, em tablets e celulares, temos retrato (portrait) ou paisagem (landscape).

Resolução.

Você pode fazer regras de acordo com o tipo de mídia e/ou tipo de características. Os tipos de mídia aceitos são:

Valor	Descrição
all	Default. Usada para todos os dispositivos de tipo de mídia
print	Usada para impressora
screen	Usada para telas de computadores, tablets, smartphones, etc.

speech

Usada para telas de leitura (por exemplo: kindle), que “leem” a página para o usuário.

As características de mídia são:

Valor	Descrição
any-hover	Algum dispositivo de entrada disponível permite que o usuário passe o mouse sobre os elementos?
any-pointer	Algum mecanismo de entrada disponível é um dispositivo apontador? Em caso afirmativo, quão preciso é?
aspect-ratio	A proporção entre a largura e a altura da área de visualização
color	O número de bits por componente de cor para o dispositivo de saída
color-gamut	A faixa aproximada de cores que são suportadas pelo agente do usuário e dispositivo de saída
color-index	O número de cores que o dispositivo pode exibir
grid	Se o dispositivo é um grid ou bitmap
height	A altura da área de visualização
hover	O mecanismo de entrada principal permite que o usuário passe o mouse sobre os elementos?
inverted-colors	O navegador ou o sistema operacional subjacente está invertendo as cores?
light-level	Nível de luz ambiente atual
max-aspect-ratio	A proporção máxima entre a largura e a altura da área de exibição
max-color	O número máximo de bits por componente de cor para o dispositivo de saída
max-color-index	O número máximo de cores que o dispositivo pode exibir
max-height	A altura máxima da área de exibição, como uma janela do navegador
max-monochrome	O número máximo de bits por "cor" em um dispositivo monocromático (escala de cinza)
max-resolution	A resolução máxima do dispositivo, usando dpi ou dpcm
max-width	A largura máxima da área de exibição, como uma janela do navegador
min-aspect-ratio	A proporção mínima entre a largura e a altura da área de exibição
min-color	O número mínimo de bits por componente de cor para o dispositivo de saída

min-color-index	O número mínimo de cores que o dispositivo pode exibir
min-height	A altura mínima da área de exibição, como uma janela do navegador
min-monochrome	O número mínimo de bits por "cor" em um dispositivo monocromático (escala de cinza)
min-resolution	A resolução mínima do dispositivo, usando dpi ou dpcm
min-width	A largura mínima da área de exibição, como uma janela do navegador
monochrome	O número de bits por "cor" em um dispositivo monocromático (escala de cinza)
orientation	A orientação da área de visualização (modo paisagem ou retrato)
overflow-block	Como o dispositivo de saída lida com o conteúdo que ultrapassa a área de visualização ao longo do eixo do bloco
overflow-inline	O conteúdo que ultrapassa a área de visualização ao longo do eixo pode ser rolado
pointer	O mecanismo de entrada principal é um dispositivo apontador e, em caso afirmativo, qual é sua precisão?
resolution	A resolução do dispositivo de saída, usando dpi ou dpcm
scan	O processo de digitalização do dispositivo de saída
scripting	O script (por exemplo, JavaScript) está disponível?
update	Com que rapidez o dispositivo de saída pode modificar a aparência do conteúdo
width	A largura da área de visualização

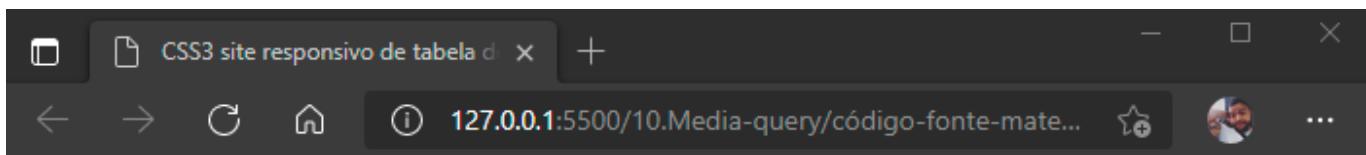
Vamos praticar

Vamos criar um arquivo HTML, nesse exemplo vamos usar o estilo CSS declarado no documento HTML para ficar mais simples de explicar e atualizar o arquivo:

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="01.estilo.css">
  <title>CSS3 site responsivo de tabela de preços</title>
```

```
</head>
<body>
    <h2>CSS3 site responsivo de tabela de preços</h2>
    <div class="col">
        <ul class="price-box">
            <li class="header">Simples</li>
            <li class="emph"><strong>R$ 5.99</strong> / ao Mês</li>
            <li><strong>20GB</strong> Espaço em Disco</li>
            <li><strong>10GB</strong> Transferência de dados</li>
            <li><strong>2</strong> Domínios</li>
            <li><strong>50</strong> Contas de Email</li>
            <li><strong>2</strong> Contas de FTP</li>
            <li><a href="#" class="btn">Cadastre-se</a></li>
        </ul>
    </div>
    <div class="col">
        <ul class="price-box best">
            <li class="header header-green">Padrão</li>
            <li class="emph"><strong>R$ 15.99</strong> / ao Mês</li>
            <li><strong>75GB</strong> Espaço em Disco</li>
            <li><strong>50GB</strong> Transferência de dados</li>
            <li><strong>10</strong> Domínios</li>
            <li><strong>100</strong> Contas de Email</li>
            <li><strong>Ilimitadas</strong> Contas de FTP</li>
            <li><a href="#" class="btn">Cadastre-se</a></li>
        </ul>
    </div>
    <div class="col">
        <ul class="price-box">
            <li class="header">Super</li>
            <li class="emph"><strong>R$ 25.99</strong> / ao Mês</li>
            <li><strong>120GB</strong> Espaço em Disco</li>
            <li><strong>1000GB</strong> Transferência de dados</li>
            <li><strong>Ilimitados</strong> Domínios</li>
            <li><strong>Ilimitadas</strong> Contas de mail</li>
            <li><strong>Ilimitadas</strong> Contas de FTP</li>
            <li><a href="#" class="btn">Cadastre-se</a></li>
        </ul>
    </div>
</body>
```

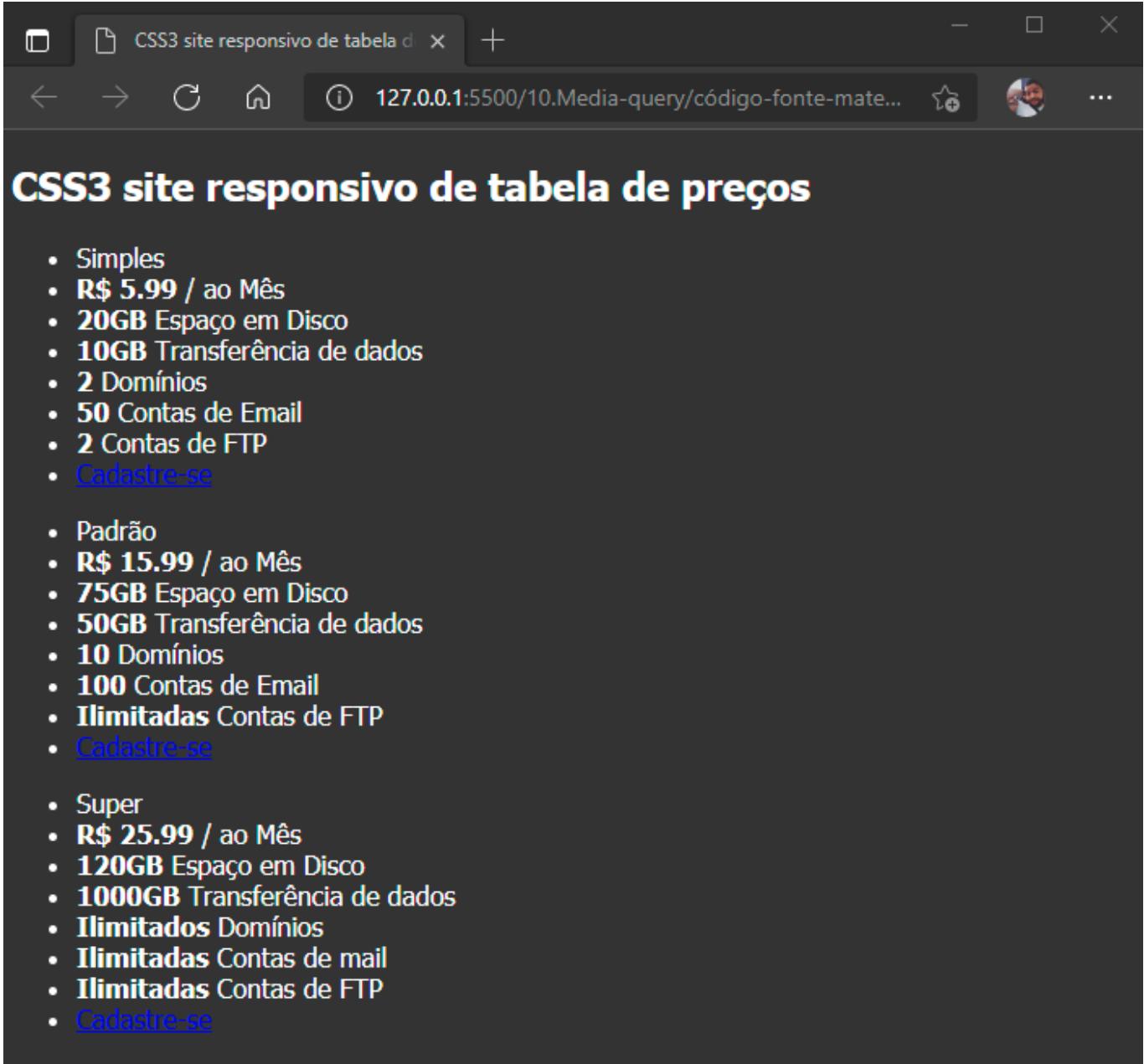


CSS3 site responsivo de tabela de preços

- Simples
 - R\$ 5,99 / ao Mês
 - 20GB Espaço em Disco
 - 10GB Transferência de dados
 - 2 Domínios
 - 50 Contas de Email
 - 2 Contas de FTP
 - [Cadastre-se](#)
- Padrão
 - R\$ 15,99 / ao Mês
 - 75GB Espaço em Disco
 - 50GB Transferência de dados
 - 10 Domínios
 - 100 Contas de Email
 - Ilimitadas Contas de FTP
 - [Cadastre-se](#)
- Super
 - R\$ 25,99 / ao Mês
 - 120GB Espaço em Disco
 - 1000GB Transferência de dados
 - Ilimitados Domínios
 - Ilimitadas Contas de mail
 - Ilimitadas Contas de FTP
 - [Cadastre-se](#)

Observe que a página está totalmente sem formatação, vamos configurar alguns estilos. Vamos começar a configurar o estilo do elemento <body>, para configurar **a família da fonte**, a **cor de plano de fundo** e a **cor do texto**. Insira o seguinte código no arquivo **CSS**.

```
@charset "utf-8";
body {
    font-family: "Tahoma", sans-serif;
    background: #333;
    color: #fff;
}
```

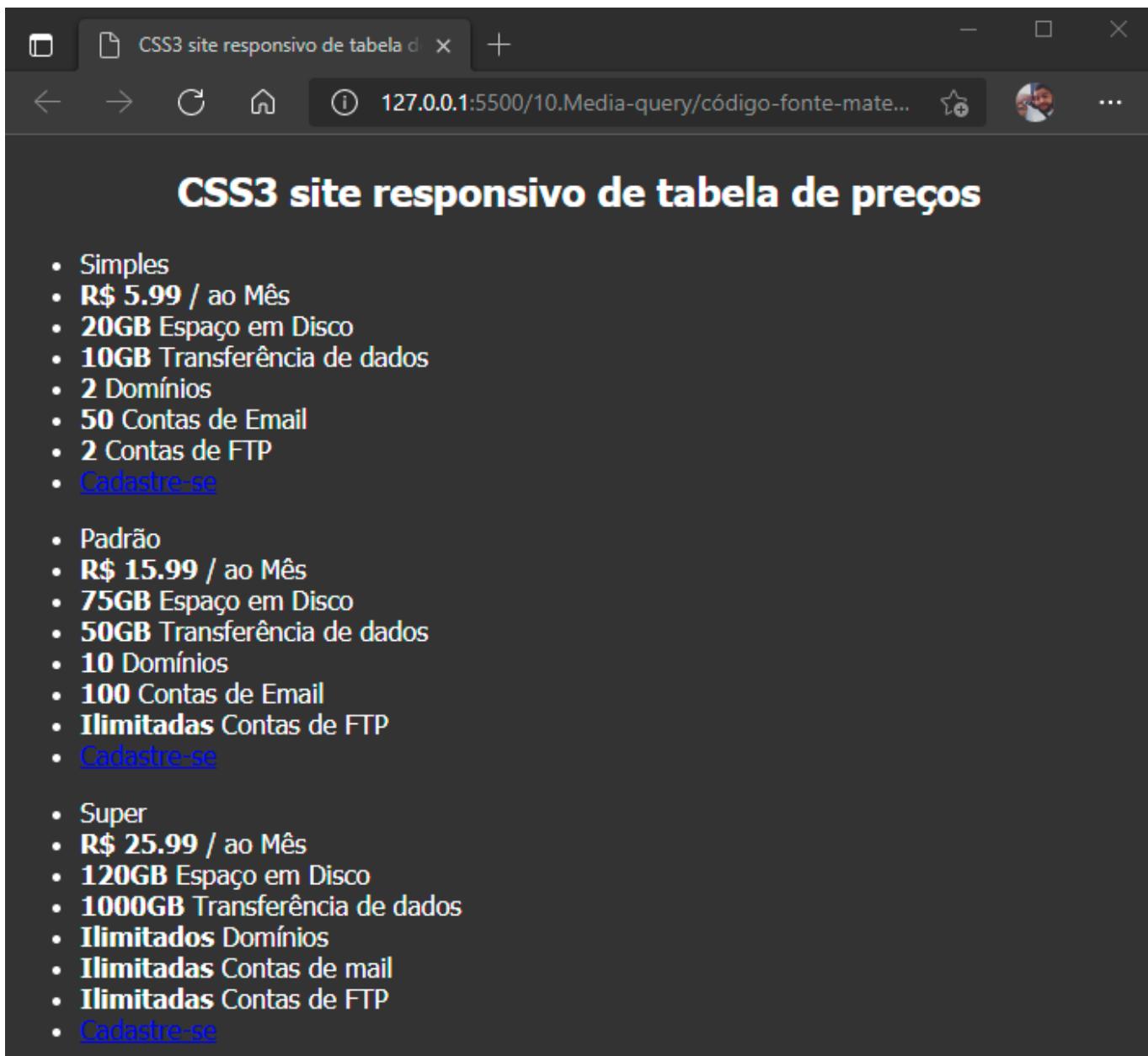


The screenshot shows a browser window with the title "CSS3 site responsivo de tabela de preços". The address bar indicates the URL is 127.0.0.1:5500/10.Media-query/código-fonte-mate... . The page content displays three pricing plans: Simple, Standard, and Super, each with its own set of features and a "Cadastre-se" button.

Plano	Preço	Espaço em Disco	Transferência de dados	Domínios	Contas de Email	Contas de FTP
Simples	R\$ 5,99 / ao Mês	20GB	10GB	2	50	2
Padrão	R\$ 15,99 / ao Mês	75GB	50GB	10	100	Ilimitadas
Super	R\$ 25,99 / ao Mês	120GB	1000GB	Ilimitados	Ilimitadas	Ilimitadas

Agora, vamos atualizar o estilo do elemento <h2> para ficar com o alinhamento centralizado.

```
h2 {  
    text-align: center;  
}
```



The screenshot shows a browser window with the title "CSS3 site responsivo de tabela de preços". The address bar indicates the URL is 127.0.0.1:5500/10.Media-query/código-fonte-mate... . The page content displays a price table with three columns: Simple, Standard, and Super, each with its own set of features and a "Cadastre-se" button.

	Simple	Standard	Super
Price	R\$ 5.99 / ao Mês	R\$ 15.99 / ao Mês	R\$ 25.99 / ao Mês
Space	20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
Data Transfer	10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
Domains	2 Domínios	10 Domínios	Ilimitados Domínios
Email Accounts	50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
FTP Accounts	2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Action	Cadastre-se	Cadastre-se	Cadastre-se

Agora, vamos atualizar a classe `.col` que está associada aos elementos <div> e configura o tipo de **flutuação**, a **largura**, o **padding** e o **box-sizing**.

```
.col {  
    float: left;  
    width: 30%;  
    padding: 8px;  
    box-sizing: border-box;  
}
```



The screenshot shows a browser window with the title "CSS3 site responsivo de tabela de preços". The page content is a responsive price table with three columns: "Simples", "Padrão", and "Super". Each column lists features like storage, transfer, domains, email accounts, and FTP accounts, with a "Cadastre-se" link at the bottom.

Simples	Padrão	Super
R\$ 5,99 / ao Mês	R\$ 15,99 / ao Mês	R\$ 25,99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Agora, vamos atualizar a classe `.price-box` que está associada às listas não ordenadas (**elemento ``**) e configurar a **cor de plano de fundo**, a **cor do texto**, o **estilo da lista**, a **borda**, a **margem** e o **padding**.

```
.price-box {  
    background: #fff;  
    color: #555;  
    list-style-type: none;  
    border: 1px solid #555;  
    margin: 0;  
    padding: 0;  
}
```



CSS3 site responsivo de tabela de preços

Simples	Padrão	Super
R\$ 5.99 / ao Mês	R\$ 15.99 / ao Mês	R\$ 25.99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Em seguida, a classe **.header** que está associada ao primeiro elemento das listas não ordenadas e configurar a cor de plano de fundo, a cor do texto e o tamanho da fonte.

```
.price-box .header {
    background-color: #111;
    color: #fff;
    font-size: 25px;
}
```

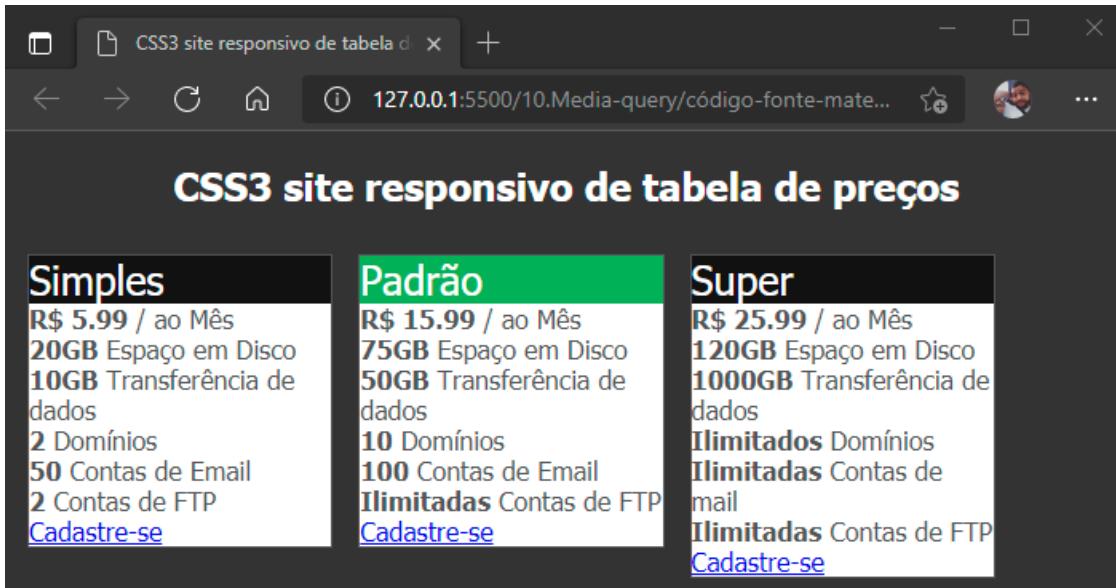


CSS3 site responsivo de tabela de preços

Simples	Padrão	Super
R\$ 5.99 / ao Mês	R\$ 15.99 / ao Mês	R\$ 25.99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Em seguida, a classe **.header-green** para alterar a **cor do plano de fundo do cabeçalho do plano Padrão**, como se ele fosse o recomendado pelo site.

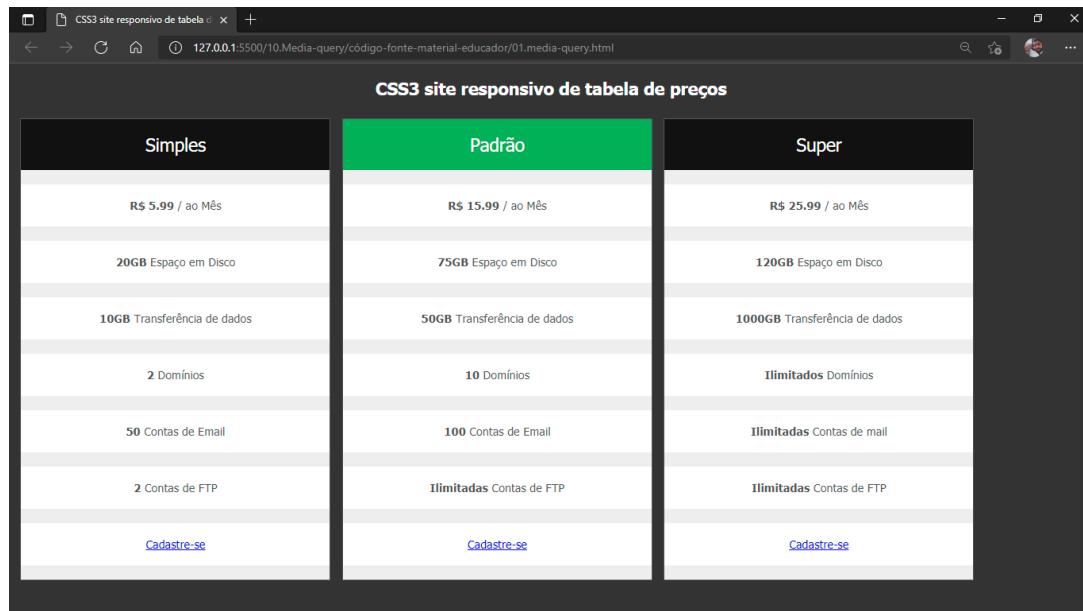
```
.price-box .header.header-green {
    background: #4caf50;
}
```



Simples	Padrão	Super
R\$ 5,99 / ao Mês	R\$ 15,99 / ao Mês	R\$ 25,99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Em seguida, vamos configurar **os itens das listas não ordenadas** (elementos ****).

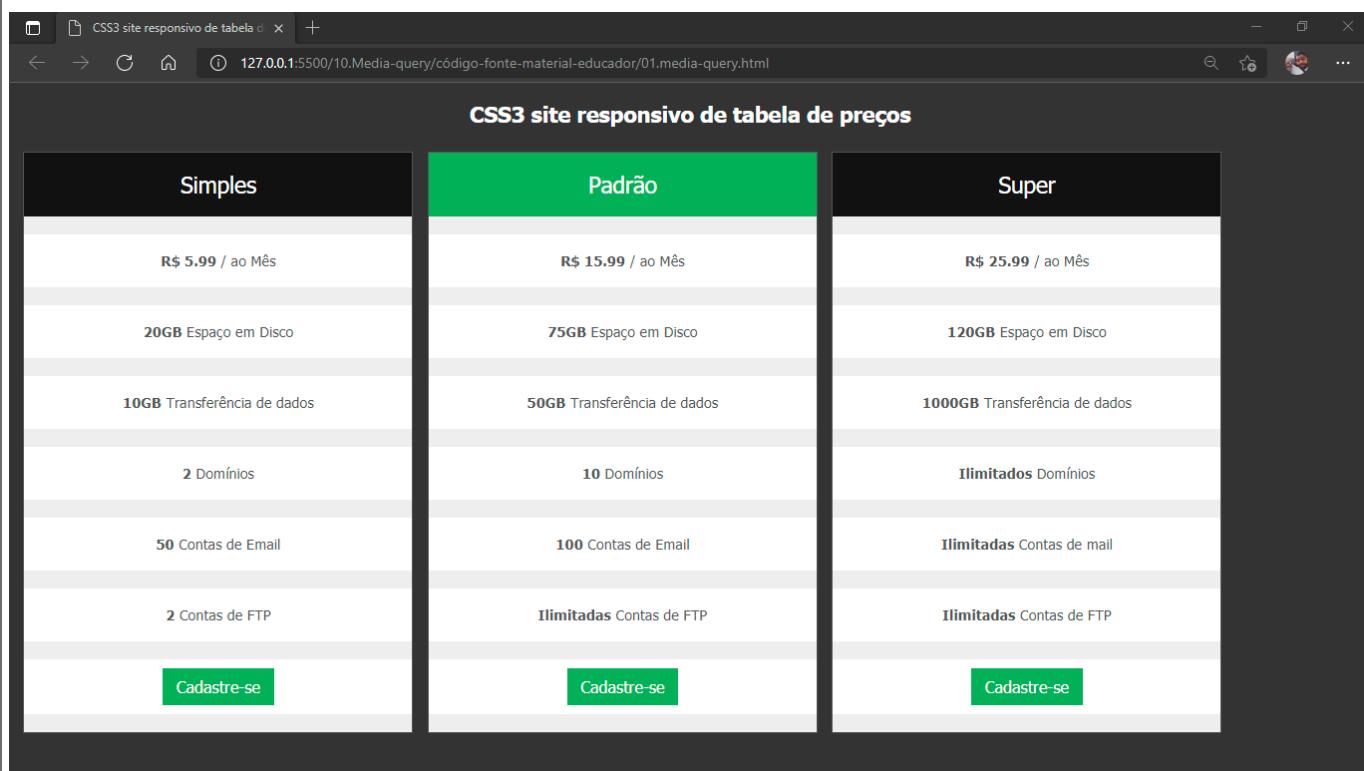
```
.price-box li {
    padding: 20px;
    border-bottom: 20px solid #eee;
    text-align: center;
}
```



Simples	Padrão	Super
R\$ 5,99 / ao Mês	R\$ 15,99 / ao Mês	R\$ 25,99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Por fim, vamos configurar a classe **.btn** que está associada aos links com o conteúdo cadastrado e definir a **cor de plano de fundo**, a **borda**, a **cor do texto**, **padding**, o **alinhamento do texto**, a **decoração do texto** e o **tamanho da fonte**.

```
.btn {
    background-color: #4caf50;
    border: none;
    color: #fff;
    padding: 10px 15px;
    text-align: center;
    text-decoration: none;
    font-size: 18px;
}
```



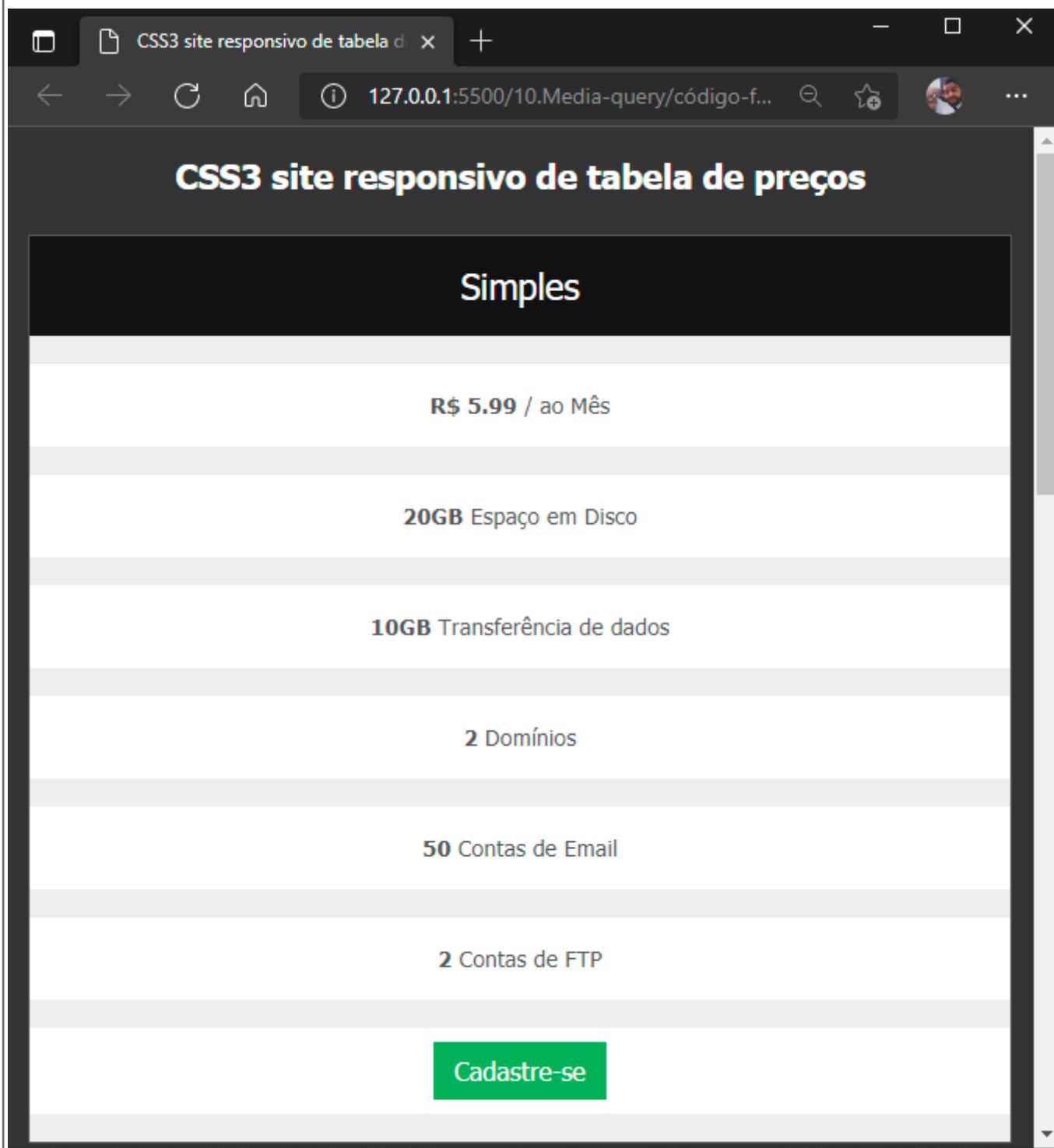
CSS3 site responsivo de tabela de preços

Simples	Padrão	Super
R\$ 5,99 / ao Mês	R\$ 15,99 / ao Mês	R\$ 25,99 / ao Mês
20GB Espaço em Disco	75GB Espaço em Disco	120GB Espaço em Disco
10GB Transferência de dados	50GB Transferência de dados	1000GB Transferência de dados
2 Domínios	10 Domínios	Ilimitados Domínios
50 Contas de Email	100 Contas de Email	Ilimitadas Contas de mail
2 Contas de FTP	Ilimitadas Contas de FTP	Ilimitadas Contas de FTP
Cadastre-se	Cadastre-se	Cadastre-se

Enfim, vamos configurar a **media query**, para **exibir as tabelas uma embaixo da outra, quando estivermos exibindo o site em tela de até 700px**. Insira o código no arquivo **CSS**. Nesse caso vamos configurar a classe **.col** para usar a largura da área de visualização em **100%**, para tela com até **700px de largura**.

```
@media(max-width:768px) {
    .col {
        width: 100%;
    }
}
```

Salve o arquivo e para visualizar o resultado, redimensione a janela do navegador com o mouse diminuindo a largura.



The screenshot shows a responsive web page titled "CSS3 site responsivo de tabela de preços". The page features a dark header with the word "Simples" in white. Below the header is a table with five rows, each containing a bolded value and a descriptive text. The rows are separated by horizontal grey lines. A green button at the bottom right of the table says "Cadastre-se". The browser's address bar shows the URL "127.0.0.1:5500/10.Media-query/código-f...".

R\$ 5.99	/ ao Mês
20GB	Espaço em Disco
10GB	Transferência de dados
2	Domínios
50	Contas de Email
2	Contas de FTP

Cadastre-se



Os objetivos desta aula são:

- Compreender a criação de diferentes layouts com o uso do flexbox;
- Conhecer as diversas propriedades do flexbox;
- Aplicar os recursos do flexbox nas folhas de estilo.

Bons estudos!

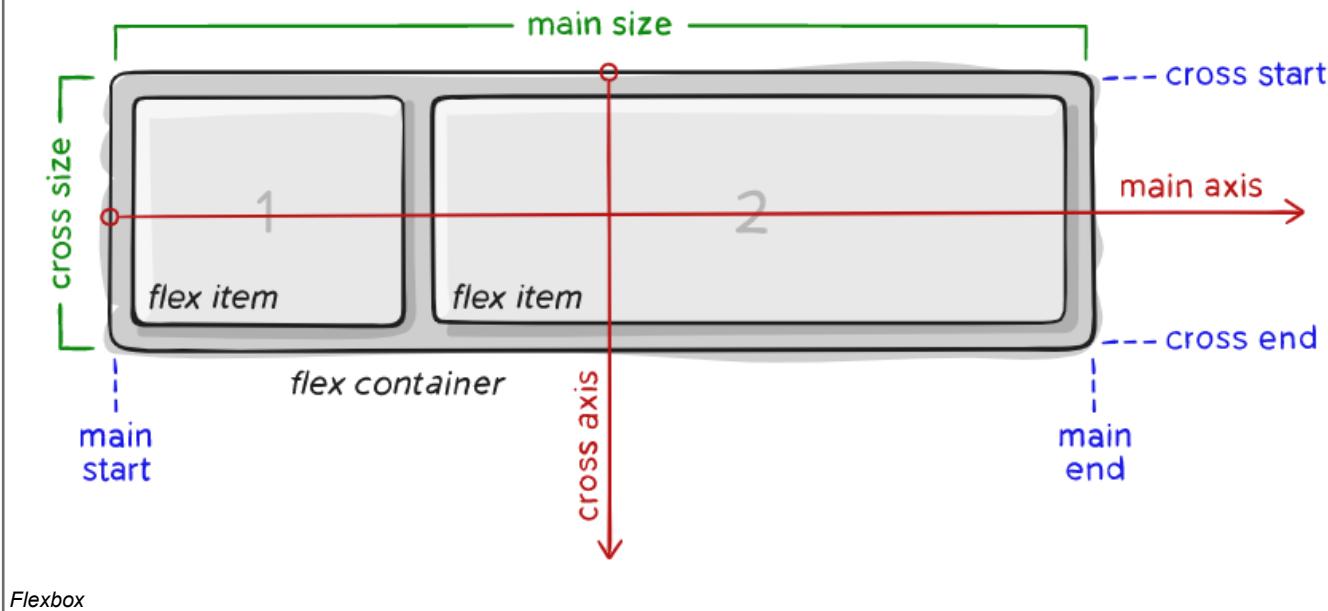
CSS Flexible Box Layout

Você tem usado a propriedade **float do CSS** para criar página web com duas ou três colunas, o que é uma técnica muito comum para esse tipo de organização no seu site. Porém, existe uma técnica lançada em 2017 pela W3C que oferece uma maneira mais eficiente de dispor, alinhar e distribuir o espaço entre os elementos em uma página web, essa técnica de chama **CSS Flexible Box Layout** ou apenas **Flexbox**.

O objetivo do **Flexbox** é propor um **layout flexível**, no qual os **elementos estão contidos em um container flexível e podem ser configurados em uma dimensão (horizontal ou vertical)** de uma maneira bem prática e simples. A ideia principal por trás do layout flexível é dar ao container flexibilidade para alterar a largura e altura dos itens e ordená-los para melhor encaixá-los no espaço disponível.

O container flex **expande os itens para preencher o espaço livre disponível ou encolhe-os para evitar overflow** (overflow em HTML é o conteúdo transbordar, ou seja, ocupar um espaço maior do que o visível e fazendo aparecer as barras de rolagem).

O Flexbox é um módulo no CSS que inclui um conjunto de propriedades. Algumas estão **atreladas ao elemento pai (container)** e outras aos **elementos filhos (itens)**. A imagem abaixo mostra a ideia principal por trás do flexbox.



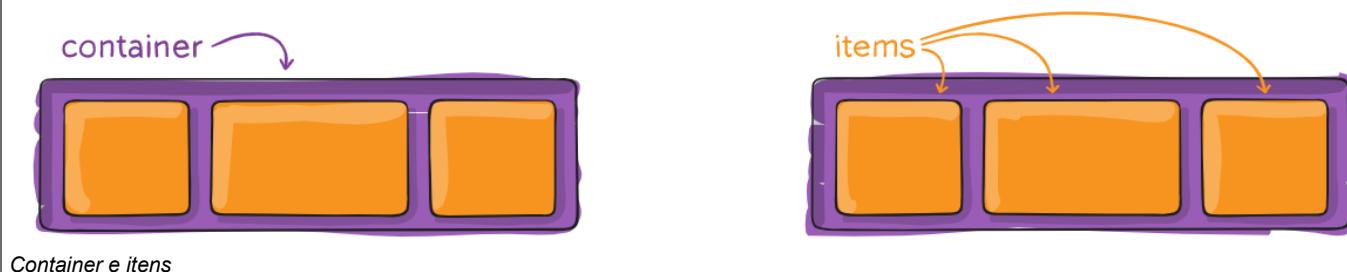
Os itens serão colocados seguindo o **main axis** (partindo do **main-start** até o **main-end**) ou, também, podem ser colocados seguindo o **cross axis** (partindo do **cross-start** até o **cross-end**). Vamos explicar cada uma dessas nomenclaturas.

- **main axis**: o main axis de um container flexível é o eixo principal ao longo do qual os itens flexíveis serão dispostos. Cuidado, não é necessariamente horizontal; depende da propriedade **flex-direction**.
- **main-start | main-end**: os itens flexíveis são colocados dentro do container começando a partir do **main-start** e indo até o **main-end**.

- **main-size**: a largura ou altura de um item flexível, que estiver na dimensão principal, é o **main-size** do item. O main-size de um item flexível pode ser a propriedade ‘width’ ou ‘height’ do item, pois depende de como foi configurado a propriedade **flex-direction**.
- **cross-axis**: o eixo perpendicular ao eixo principal é chamado **cross-axis**. Sua direção depende da direção do eixo principal.
- **cross-start | cross-end**: as linhas flexíveis são preenchidas com itens e colocadas no container, começando no **cross-start** e indo até o **cross-end**.
- **cross-size**: A largura ou altura de um item flexível é o **cross-size**. O cross-size de um item flexível pode ser a propriedade ‘width’ ou ‘height’ do item, pois depende de como foi configurado a propriedade **flex-direction**.

Propriedades do flexbox

Temos propriedades relacionadas ao elemento pai. Ou seja, container e propriedades relacionados aos elementos filhos, também chamados de itens.



Vamos começar pelas **propriedades** do **container do flexbox**. Elas são **display**, **flex-direction**, **flex-wrap**, **flex-flow**, **justify-content**, **align-items** e **align-content**.

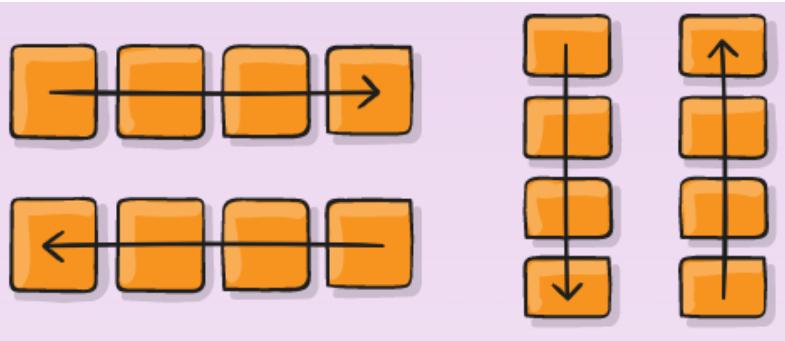
Propriedade display

A propriedade **display** pode definir um container como **flexível**, **inline** ou **block** dependendo do valor que é dado. Portanto ela habilita o contexto flex para todos os itens dentro do container.

```
@charset "utf-8";
.container {
  display: flex;
}
```

Propriedade flex-direction

A propriedade **flex-direction** configura a direção dos itens flexíveis, ou seja, ela estabelece o **main-axis** do container.



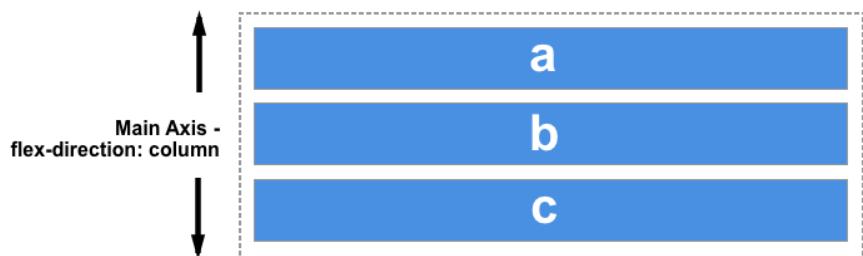
Direção dos itens em um container.

Os valores possíveis para configurar essa propriedade são:

- **row (default)**: da esquerda para a direita se a propriedade **direction** estiver com o valor padrão **ltr (left-to-right, é o padrão dos documentos HTML)**. Ou da direita para a esquerda se a propriedade **direction** estiver configura com o valor **rtl (right-to-left)**.



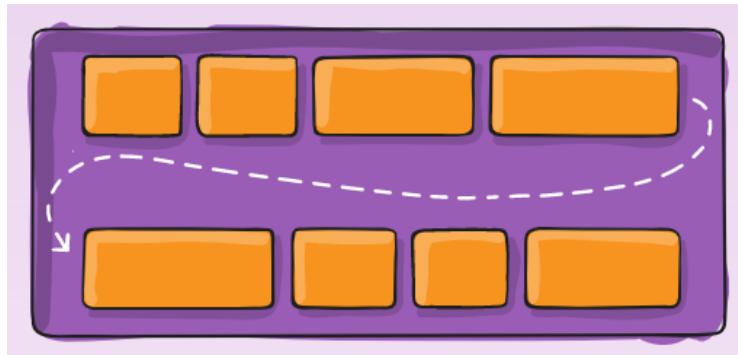
- **row-reverse**: da direita para a esquerda se a propriedade **direction** estiver com o valor padrão **ltr (left-to-right, é o padrão dos documentos HTML)**. Ou da esquerda para a direita se a propriedade **direction** estiver configura com o valor **rtl (right-to-left)**
- **column**: de cima para baixo se a propriedade **direction** estiver com o valor padrão **ltr (left-to-right, é o padrão dos documentos HTML)**. Ou de baixo para cima se a propriedade **direction** estiver configura com o valor **rtl (right-to-left)**.



- **column-reverse**: de baixo para cima se a propriedade **direction** estiver com o valor padrão **ltr (left-to-right, é o padrão dos documentos HTML)**. Ou de cima para baixo se a propriedade **direction** estiver configura com o valor **rtl (right-to-left)**

Propriedade flex-wrap

A propriedade **flex-wrap** configura se os itens serão exibidos em múltiplas linhas. Por padrão, os itens irão tentar encaixar em uma única linha, pois o valor default é **nowrap**. Você precisa modificar essa propriedade, caso precise exibir os itens em mais de uma linha.



Itens em múltiplas linhas de cima para baixo.

Os valores possíveis para configurar essa propriedade são:

- **nowrap** (default): todos os itens serão dispostos em uma única linha.
- **wrap**: os itens flex poderão ser dispostos em múltiplas linhas de cima para baixo.
- **wrap-reverse**: os itens flex poderão ser dispostos em múltiplas linhas de baixo para cima.

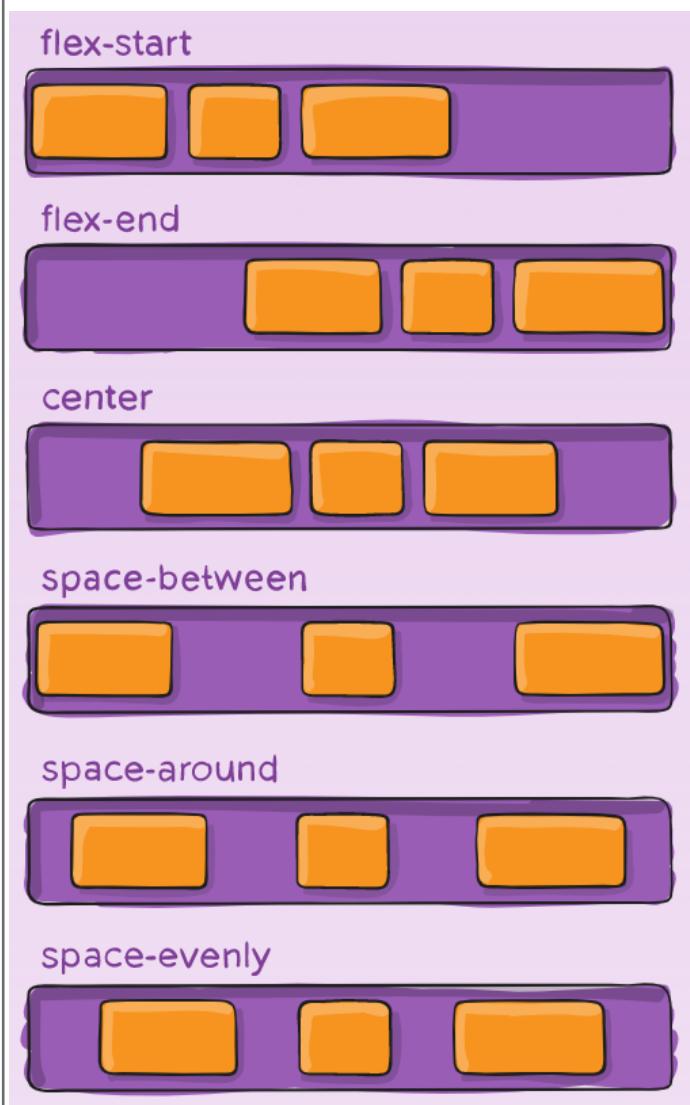
Propriedade flex-flow

Essa é uma abreviação das propriedades **flex-direction** e **flex-wrap**. Com uma única instrução você pode configurar a direção e a múltiplas linhas de uma vez.

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

Propriedade justify-content

A propriedade **justify-content** configura como o navegador irá exibir qualquer espaço extra que possa existir no container, ou seja, ela define o alinhamento ao longo do **main-axis** do container.



Opções de valores para a propriedade **justify-content**.

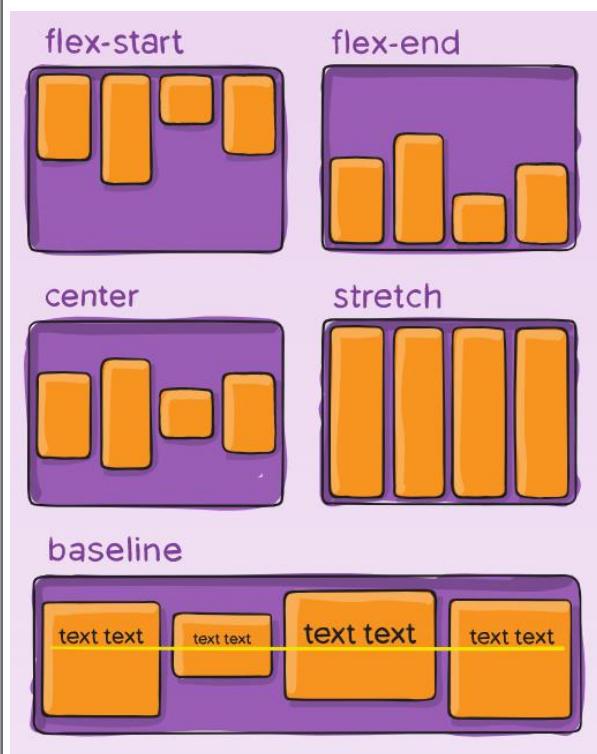
Os valores possíveis para configurar essa propriedade são:

- **flex-start** (default): os itens são exibidos alinhados no início do **flex-direction**.
- **flex-end**: os itens são exibidos alinhados no final do **flex-direction**.
- **start**: os itens são exibidos alinhados no início da direção do modo de escrita.
- **end**: os itens são exibidos alinhados no final da direção do modo de escrita.
- **left**: os itens são exibidos alinhados em direção da borda esquerda do container a menos que isso não faça sentido com o `flex-direction`, então se comporta como o valor **start**.
- **right**: os itens são exibidos alinhados em direção da borda direita do container a menos que isso não faça sentido com o `flex-direction`, então se comporta como o valor **end**.
- **center**: os itens são alinhados ao longo da linha.

- **space-between**: os itens são distribuídos uniformemente na linha. Ou seja, o primeiro item está no início da linha e o último está no final da linha.
- **space-around**: os itens são distribuídos uniformemente na linha com tamanhos de espaços iguais ao redor do item (na direita e esquerda). Note que visualmente os espaços não são iguais, uma vez que todos os itens têm espaço igual em ambos os lados. O primeiro item terá uma unidade de espaço contra a borda do container, mas duas unidades de espaço entre o próximo item, porque esse próximo item tem seu próprio espaçamento que se aplica.
- **space-evenly**: os itens são distribuídos de forma que o espaçamento entre quaisquer dois itens seja igual.

Propriedade align-itens

A propriedade **align-itens** define o comportamento padrão de como os itens flex estão dispostos ao longo do **cross-axis**. Ele faz a mesma função para o **cross-axis**, quanto o **justify-content** para o **main-axis**.



Propriedade align-items

Os valores possíveis para configurar essa propriedade são:

- **stretch** (default): esticar os itens para preencher todo o container (respeitando ainda a min-width/max-width);
- **flex-start / start / self-start**: os itens são colocados no início do **cross-axis**. A diferença entre eles é sutil e diz respeito às regras do flex-direction e do writing-mode;

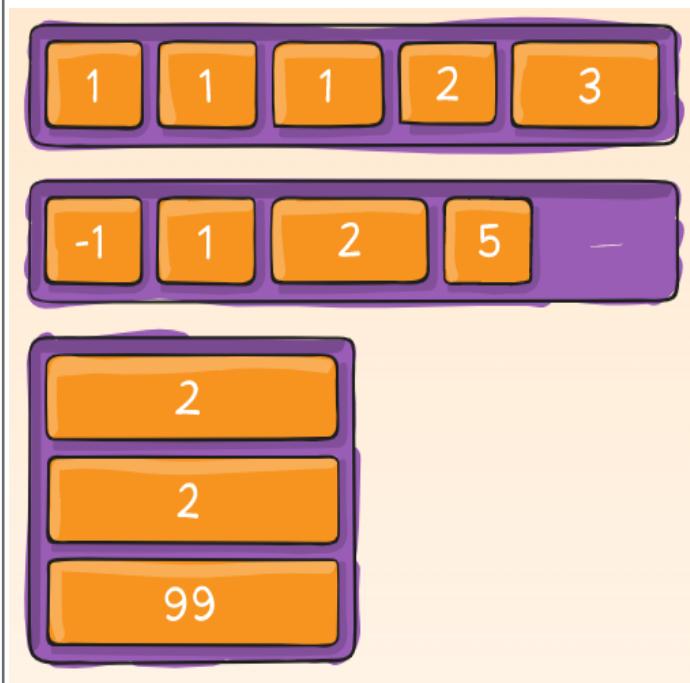
- **flex-end / end / self-end**: os itens são colocados no final do **cross-axis**. A diferença entre eles é sutil e diz respeito às regras do flex-direction e do writing-mode;
- **center**: os itens são centralizados no **cross-axis**;
- **baseline**: os itens são alinhados assim como suas linhas base são alinhadas.

Vamos, agora, para as propriedades dos itens do flexbox. Elas são **order**, **flex-grow**, **flex-shrink**, **flex-basis**, **flex** e **align-self**.

Propriedade **order**

A propriedade **order** controla a ordem onde cada item deverá aparecer no container. Você pode por exemplo alterar a ordem de um item por meio de uma classe.

```
.item {
  order: 5;
}
```

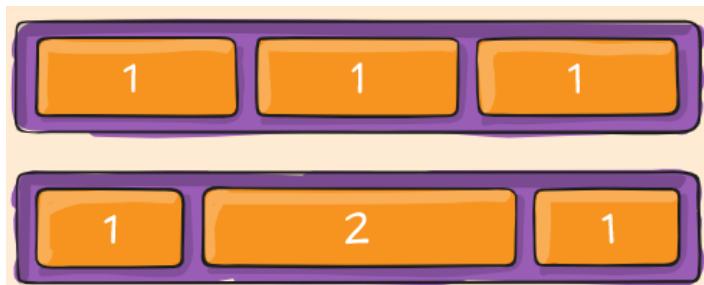


Propriedade **order**.

Propriedade flex-grow

A propriedade **flex-grow** define a capacidade de um item flex crescer, se necessário. Ela determina a quantidade de espaço disponível dentro do flex container que o item deve ocupar. Se todos os itens tiverem **flex-grow** definido como 1, o espaço restante no container será distribuído igualmente para todos os filhos. Se um dos filhos tiver o valor 2, o espaço restante ocupará o dobro do espaço dos outros (ou tentará, pelo menos).

```
.item {
  flex-grow: 4;
  /* default 0 */
}
```



Propriedade flex-grow.

Propriedade flex-shrink

A propriedade **flex-shrink** define a capacidade de um item flexível encolher, se necessário.

```
.item {
  flex-shrink: 3;
  /* default 1 */
}
```

Propriedade flex-basis

A propriedade **flex-basis** define o tamanho padrão de um elemento antes do espaço restante seja distribuído. Pode ser um tamanho (ex. 20%, 5rem, etc) ou uma keyword. A keyword **auto** significa “verifique a minha propriedade **width** e **height**”. A keyword **content** significa “dimensionar com base no conteúdo do item”. Existem outras Keywords que não iremos abordar para o conteúdo não ficar muito extenso.

```
.item {
  flex-basis: auto;
  /* default auto */
}
```

Propriedade flex

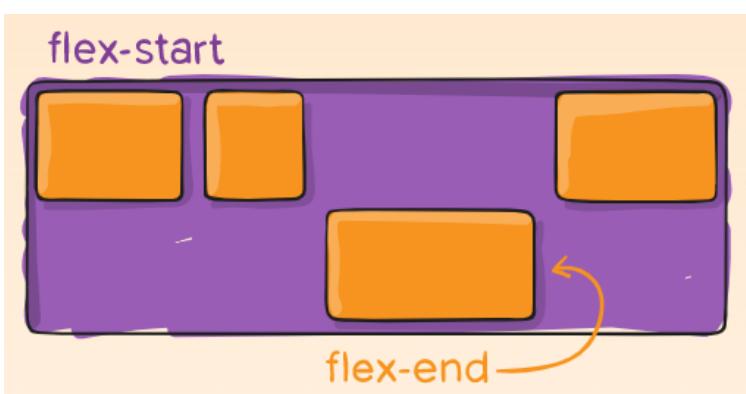
A propriedade **flex** é uma abreviação das propriedades **flex-grow**, **flex-shrink** e **flex-basis**. Com uma única instrução você pode configurar a capacidade de um item flex crescer ou diminuir e o tamanho padrão de um elemento.

```
.item {
  flex: [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'>];
}
```

Propriedade align-self

A propriedade **align-self** permite que o alinhamento padrão (ou aquele especificado por align-items) seja substituído nos itens flex individualmente.

```
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```



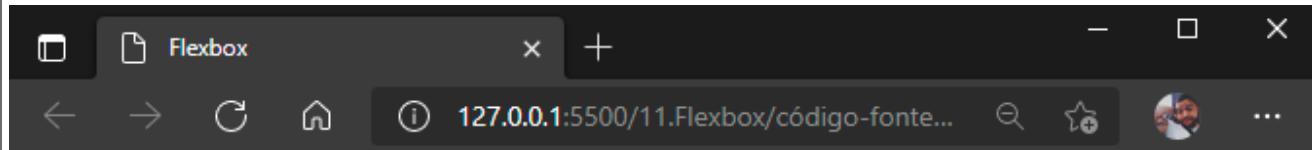
Propriedade align-self.

Vamos praticar

Vamos criar um arquivo HTML. Nesse exemplo vamos usar o estilo CSS declarado no documento HTML para ficar mais simples de explicar e atualizar o arquivo. Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="estilo.css">
  <title>Flexbox</title>
</head>
```

```
<body>
  <section class="container">
    <div class="item-box-1">
      <h3>Box um</h3>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus
voluptates dolore voluptatem,
          rerum eum omnis consequuntur iste libero fugit beatae sequi voluptas id
deleniti quibusdam eius, ad
          recusandae.</p>
    </div>
    <div class="item-box-2">
      <h3>Box dois</h3>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus
voluptates dolore voluptatem,
          rerum eum omnis consequuntur iste libero fugit beatae sequi voluptas id
deleniti quibusdam eius, ad
          recusandae.</p>
    </div>
    <div class="item-box-3">
      <h3>Box três</h3>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus
voluptates dolore voluptatem,
          rerum eum omnis consequuntur iste libero fugit beatae sequi voluptas id
deleniti quibusdam eius, ad
          recusandae.</p>
    </div>
  </section>
</body>
</html>
```



Box Um

Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit, beatae tenetur suscipit eaque accusantium asperiores alias quis eum repudiandae amet nemo fuga deleniti magni debitatis minima.

Box Dois

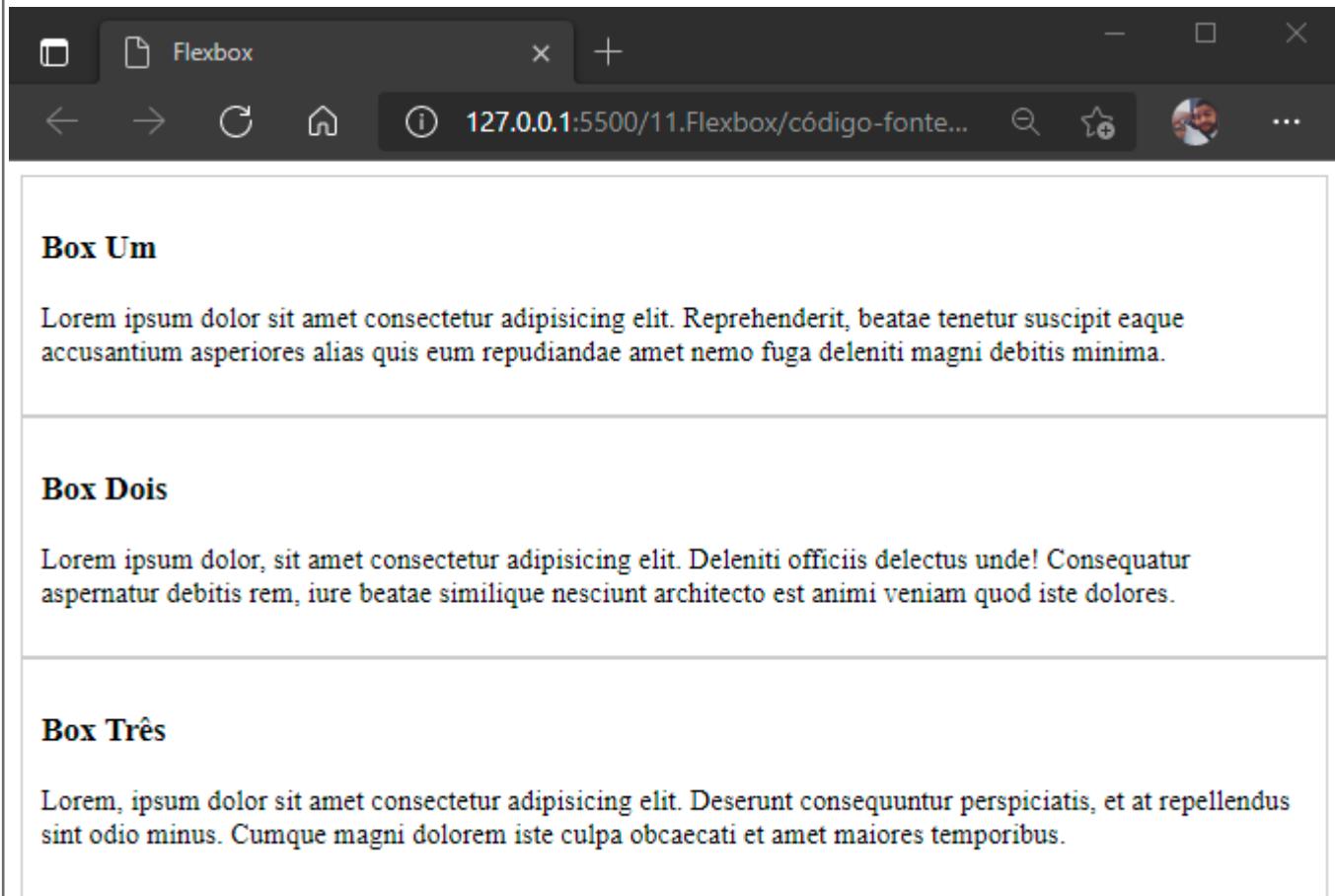
Consequatur aspernatur debitis rem, iure beatae similique nesciunt architecto est animi veniam quod iste dolores.

Box Três

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Deserunt consequuntur perspiciatis, et at repellendus sint odio minus. Cumque magni dolorem iste culpa obcaecati et amet maiores temporibus.

Observe que a página está totalmente sem formatação, vamos configurar alguns estilos. Vamos começar a configurar os elementos <div> dentro da classe **.container** e definir uma **borda** e o **espaçamento** desses elementos. Insira o seguinte código no arquivo **CSS**.

```
@charset "utf-8";
.container div {
    border: 1px #ccc solid;
    padding: 10px;
}
```



The screenshot shows a browser window with the title 'Flexbox'. The address bar displays '127.0.0.1:5500/11.Flexbox/código-fonte...'. The page content is as follows:

- Box Um**
Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit, beatae tenetur suscipit eaque accusantium asperiores alias quis eum repudiandae amet nemo fuga deleniti magni debitis minima.
- Box Dois**
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Deleniti officiis delectus unde! Consequatur aspernatur debitis rem, iure beatae similique nesciunt architecto est animi veniam quod iste dolores.
- Box Três**
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Deserunt consequuntur perspiciatis, et at repellendus sint odio minus. Cumque magni dolorem iste culpa obcaecati et amet maiores temporibus.

Agora, no **início** do arquivo **CSS** e configura a propriedade **display** na classe **.container** como **flex**.

```
@charset "utf-8";
.container {
    display: flex;
}

.container div {
    border: 1px #ccc solid;
    padding: 10px;
}
```



The screenshot shows a browser window with the title "Flexbox". The URL bar displays "127.0.0.1:5500/11.Flexbox/código-fonte...". The main content area contains three boxes arranged horizontally. The first box is labeled "Box Um" and contains the text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit, beatae tenetur suscipit eaque accusantium asperiores alias quis eum repudiandae amet nemo fuga deleniti magni debitis minima.". The second box is labeled "Box Dois" and contains the text: "Lorem ipsum dolor, sit amet consectetur adipisicing elit. Deleniti officiis delectus unde! Consequatur aspernatur debitis rem, iure beatae similique nesciunt architecto est animi veniam quod iste dolores.". The third box is labeled "Box Três" and contains the text: "Lorem, ipsum dolor sit amet consectetur adipisicing elit. Deserunt consequuntur perspiciatis, et at repellendus sint odio minus. Cumque magni dolorem iste culpa obcaecati et amet maiores temporibus."

Observe que por padrão a **flex-direction** dispõe os itens em **linha (row)**. Em seguida, vamos atualizar o estilo da classe **.item-box-1** que está associada ao **primeiro item do container**.

```
.item-box-1 {
  flex: 2;
  order: 2;
}
```

Vamos atualizar o estilo da classe **.item-box-2** que está associada ao **segundo item do container**.

```
.item-box-2 {
  flex: 1;
  order: 1;
}
```

Agora, vamos atualizar o estilo da classe **.item-box-3** que está associada ao **terceiro item do container**.

```
.item-box-3 {
  flex: 1;
  order: 3;
}
```



The screenshot shows a browser window with the title "Flexbox". The address bar displays the URL "127.0.0.1:5500/11.Flexbox/código-fonte-mat...". The page content is a flexbox layout with three items:

- Box Um**: The largest box, containing placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit, beatae tenetur suscipit eaque accusantium asperiores alias quis eum repudiandae amet nemo fuga deleniti magni debitis minima."
- Box Dois**: A medium-sized box, containing placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti officiis delectus unde! Consequatur aspernatur debitis rem, iure beatae similique nesciunt architecto est animi veniam quod iste dolores."
- Box Três**: A small box, containing placeholder text: "Lorem, ipsum dolor sit amet consectetur adipisicing elit. Deserunt consequuntur perspiciatis, et at repellendus sint odio minus. Cumque magni dolorem iste culpa obcaecati et amet maiores temporibus."

Observe que o **Box Um** está com o **dobro** do **tamanho do outros boxes** e a **ordem para eles aparecerem foi alterada**.



Grid CSS Layout

Os objetivos desta aula são:

- Compreender a criação de diferentes layouts com o uso do Grid CSS;
- Conhecer as diversas propriedades do Grid CSS;
- Aplicar os recursos do Grid CSS nas folhas de estilo.

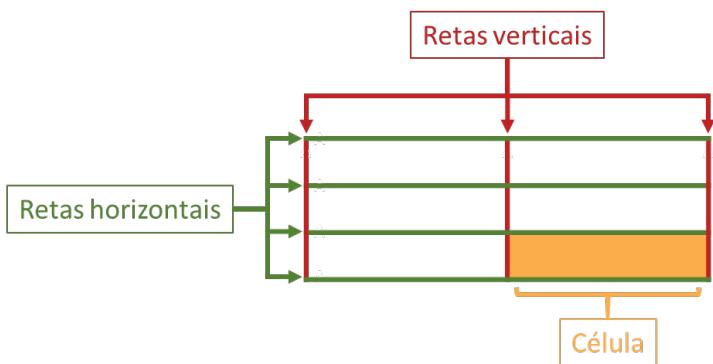
Bons estudos!

CSS Grid Layout

Você tem usado a propriedade **CSS Float** e o **CSS Flexible Box Layout** para criar a aparência de páginas web com várias colunas, porém existe uma técnica bastante eficiente para isso: o **CSS Grid Layout**.

O CSS Grid Layout é uma excelente forma de **dividir uma página em regiões principais ou definir o relacionamento em termos de medidas, posicionamento e camadas entre os diferentes componentes da marcação HTML**.

Você deve entender o grid como um layout formado por **retas verticais e horizontais**, que **formam as células do grid**. Por exemplo, a imagem abaixo mostra um grid **3x2** (**3 linhas e 2 colunas**) formado por três retas verticais e quatro retas horizontais, formando assim seis células.



Exemplo de grid com reta horizontais e verticais.

A proposta do Grid é definir o layout bidimensional de uma página web criando itens fixos ou flexíveis que podem ser configurados individualmente. Como tabelas, o layout Grid permite alinhar elementos em linha e colunas por meio de um método bidimensional para criação de layout utilizando CSS.

O **Grid tem vantagem** sobre tabelas por **não se basear em estruturação de conteúdo**. Possibilita **uma enorme variedade de layouts**. O Grid layout é o primeiro módulo no CSS criado especificamente para resolver problemas de layout.

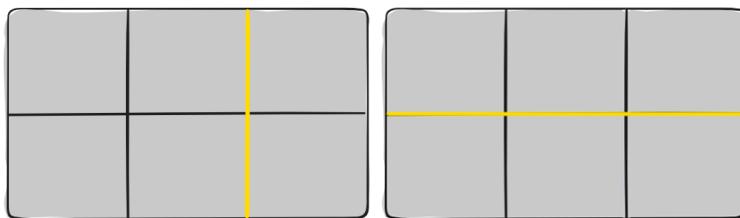
Você deve ficar atento a termologia usada no Grid Layout e se familiarizar com os termos para entender melhor como utilizar essa técnica. Os principais termos são:

- **Grid Container**: é o elemento ou classe que definimos a propriedade **display: grid**. Então, esse elemento será o pai de todos os itens do Grid. Por exemplo, podemos configurar uma classe **.container** em um elemento **<section>** e vários itens dentro dessa seção.

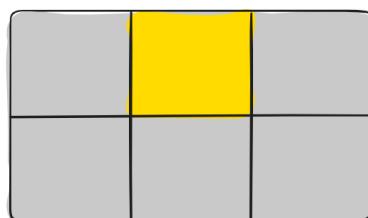
```
<section class="container">
  <div class="item item-1"> </div>
  <div class="item item-2"> </div>
  <div class="item item-3"> </div>
</section>
```

Ao definir o CSS da classe `.container` com a propriedade `display: grid`, então o elemento `<section>` passará a ser o **pai direto dos itens do container**. Você sempre precisa definir um container para trabalhar com Grid layout.

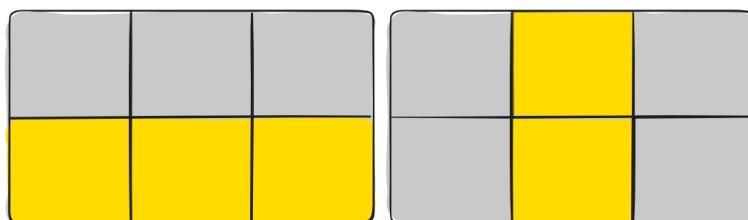
- **Grid Item:** são os filhos do grid container.
- **Grid Line:** são as linhas divisórias que formam a estrutura do grid. Elas podem ser as retas verticais ou horizontais e estar em qualquer lado da linha ou coluna.



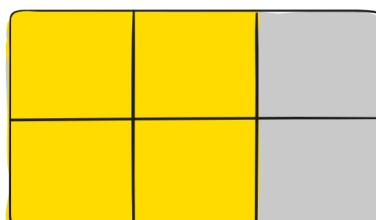
- **Grid Cell:** é a célula do grid (unidade única do grid), ou seja, o espaço entre duas retas horizontais ou verticais adjacentes. Abaixo é mostrado o Grid Cell entre as retas 1 e 2 horizontais e as retas 2 e 3 verticais.



- **Grid Track:** é uma trilha do grid, que pode ser uma linha ou uma coluna inteira.



- **Grid Area:** O espaço total rodeado por quatro retas do grid. Grid Area pode ser composta de qualquer número de células do grid. Por exemplo, abaixo a Grid Area está entre as retas horizontais 1 e 3 e as retas verticais 1 e 3.



Propriedades do Grid Layout

Como no flexbox, temos **propriedades relacionadas ao elemento pai**, ou seja, o **container**, e **propriedades relacionados aos elementos filhos**, ou seja os **itens**.

Propriedades do elemento pai (Grid Container)

Vamos começar pelas propriedades do container do **Grid Layout**. Elas são **display**, **grid-template-columns**, **grid-template-rows**, **grid-template-areas**, **grid-template**, **column-gap**, **row-gap**, **grid-column-gap**, **grid-row-gap**, **gap**, **grid-gap**, **justify-items**, **align-items**, **place-items**, **justify-content**, **align-content**, **place-content**, **grid-auto-columns**, **grid-auto-rows**, **grid-auto-flow** e **grid**.

Propriedade display

A propriedade **display** configura um elemento do HTML como um container grid e determina um novo grid formatando o contexto para o seu conteúdo. Os valores possíveis são:

- **grid**: gera um block grid
- **inline-grid**: gera um grid alinhado

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

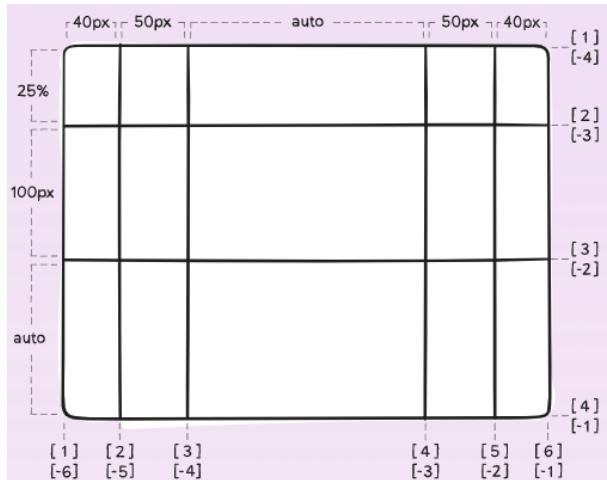
```
.container {  
    display: inline-grid;  
}
```

Propriedades grid-template-columns e grid-template-rows

As propriedades **grid-template-columns** e **grid-template-rows** configura a largura e altura das colunas e linhas do grid. Você deve colocar os valores desejados separados por espaços e os valores representam o tamanho do Grid Track (trilha). Por exemplo:

```
.container {  
    grid-template-columns: 40px 50px auto 50px 40px;  
    grid-template-rows: 25% 100px auto;  
}
```

Quando você deixa um espaço vazio entre os valores da trilha, as linhas do grid são atribuídas automaticamente aos números na ordem que aparecem. Resultado da configuração é mostrado na imagem abaixo:



Resultado das trilhas criadas no Grid.

Mas você pode escolher explicitamente os valores de cada trilha. Por exemplo, o trecho abaixo produz o mesmo resultado mostrado na imagem acima.

```
.container {
    grid-template-columns: [first] 40px [line2] 50px [line3] auto [col4-start] 50px
    [five] 40px [end];
    grid-template-rows: [row1-start] 25% [row1-end] 100px [third-line] auto [last-line];
}
```

Observe que pode haver mais de um rótulo. No exemplo a seguir, a segunda linha terá dois nomes: **row1-end** e **row2-start**:

```
.container {
    grid-template-rows: [row1-start] 25% [row1-end row2-start] 25% [row2-end];
}
```

Caso você queira **dividir em partes iguais**, pode usar a notação **repeat()**. Por exemplo:

```
.container {
    grid-template-columns: repeat(3, 20px [col-start]);
}
.container {
    grid-template-columns: repeat(4, 1fr);
}
```

É equivalente a:

```
.container {
    grid-template-columns: 20px [col-start] 20px [col-start] 20px [col-start];
}
.container {
    grid-template-columns: 1fr 1fr 1fr 1fr;
```

A unidade **fr** permite você definir o tamanho da trilha como uma fração do espaço livre dentro do container do grid. No exemplo anterior, **repeat(4, 1fr)** ou **1fr 1fr 1fr 1fr** divide o espaço livre do container em quatro parte iguais. Outro exemplo:

```
.container {
    grid-template-columns: 1fr 2fr 1fr;
}
```

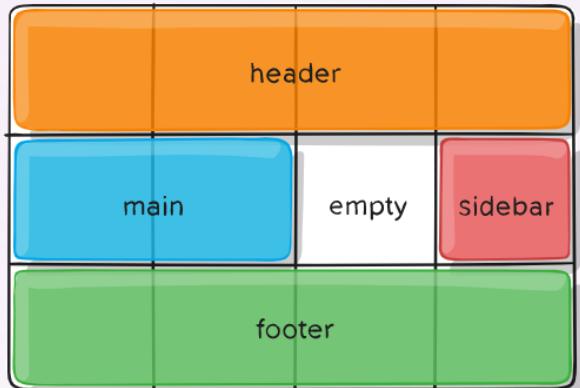
Esse último trecho de código divide o espaço livre em três partes, sendo que a parte do meio ocupa o dobro de espaço (**2fr**) do que as outras duas partes.

Propriedade **grid-template-areas**

A propriedade **grid-template-areas** define um modelo de grid referenciando os nomes das áreas do grid que são especificadas pela propriedade **grid-area**. Repetir o nome de uma área do grid faz com que o conteúdo se espalhe por essas células com o nome repetido. Um ponto significa uma célula vazia. Exemplo:

```
.item-a {
    grid-area: header;
}
.item-b {
    grid-area: main;
}
.item-c {
    grid-area: sidebar;
}
.item-d {
    grid-area: footer;
}
.container {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-template-rows: auto;
    grid-template-
    areas: "header header header header" "main main . sidebar" "footer footer footer footer"
;
```

Esse exemplo cria um grid com quatro colunas e três linhas de mesmo tamanho. Toda a trilha da parte de superior tem o mesmo rótulo **header** e será ocupado por ele. A trilha do meio será composta de duas áreas **main** (rótulo **main**), uma célula vazia (indicada pelo **ponto**) e uma área **sidebar** e a última trilha será toda o **footer** (rótulo **footer**). O resultado é mostrado na imagem abaixo:



Resultado das divisões configuradas no exemplo `grid-template-areas`.

Propriedade `grid-template`

A propriedade **grid-template** é uma abreviação das propriedades **grid-template-rows**, **grid-template-columns** e **grid-template-areas**, ou seja, com uma única declaração você pode configurar a largura e altura das colunas e linhas do grid e modelo de grid referenciando os nomes das áreas do grid. Por exemplo:

```
.container {
    grid-template: [row1-start] "header header header" 25px [row1-end] [row2-start] "footer footer footer" 25px [row2-end] / auto 50px auto;
}
```

Esse exemplo seria equivalente a:

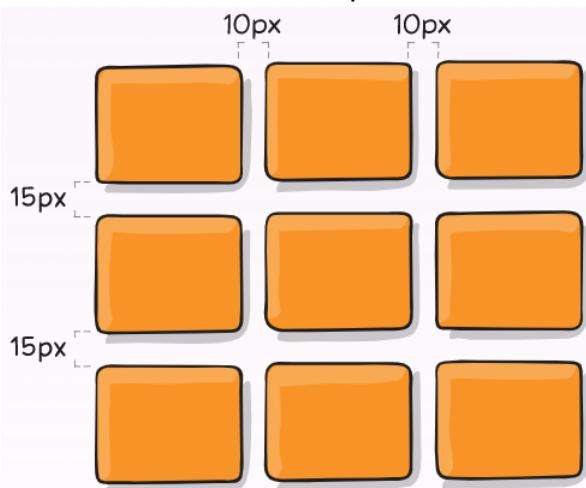
```
.container {
    grid-template-rows: [row1-start] 25px [row1-end row2-start] 25px [row2-end];
    grid-template-columns: auto 50px auto;
    grid-template-areas: "header header header" "footer footer footer";
}
```

Propriedades `column-gap`, `row-gap`, `grid-column-gap` e `grid-row-gap`

Elas especificam o espaçamento (gap) entre as linhas e as colunas do grid, ou seja, a espessura das retas verticais ou horizontais. O prefixo `grid-` pode ser omitido. Ao invés de usar **grid-row-gap**, podemos usar somente **row-gap**. Vejamos o exemplo:

```
.container {
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: 80px auto 80px;
    column-gap: 10px;
    row-gap: 15px;
}
```

O resultado desse exemplo é mostrado na imagem seguinte:



Resultado do exemplo `row-gap` e `column-gap`.

Propriedade gap ou grid-gap

A propriedade **grid-gap** é uma abreviação das propriedades **row-gap** e **column-gap**. Os valores de ser na ordem: primeiro `<grid-row-gap>` e em segundo `<grid-column-gap>`. Por exemplo:

```
.container {
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: 80px auto 80px;
    grid-gap: 15px 10px;
}
```

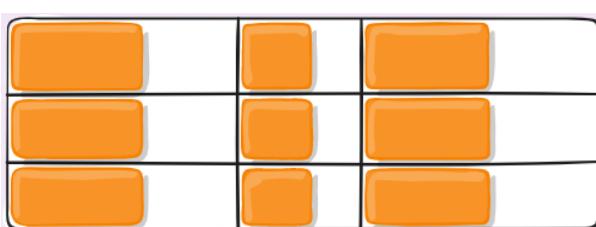
Propriedade justify-items

A propriedade **justify-items** alinha horizontalmente os itens do grid. Os valores possíveis para configurar essa propriedade são:

- **start**: os itens ficam alinhados na reta esquerda da célula do grid.

```
.container {
    justify-items: start;
}
```

Resultado:



- **end**: os itens ficam alinhados na reta direita da célula do grid.

```
.container {
  justify-items: end;
}
```

Resultado:



- **center**: os itens ficam centralizados horizontalmente na célula do grid.

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

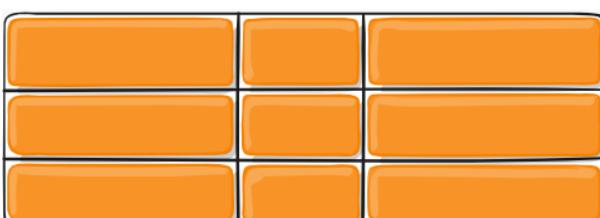
Resultado:



- **stretch**: os itens preenchem todo o espaço da célula do grid.

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

Resultado:



Esse mesmo comportamento para o alinhamento horizontal dos itens pode ser definido individualmente para cada item do grid com a propriedade **justify-self**.

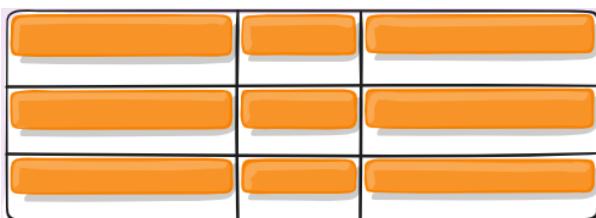
Propriedade align-items

A propriedade **align-items** alinha verticalmente os itens do grid. Os valores possíveis para configurar essa propriedade são:

- **start**: os itens ficam alinhados na reta superior da célula do grid.

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

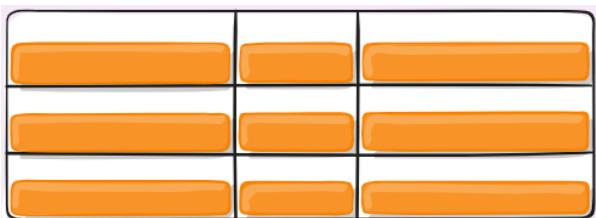
Resultado:



- **end**: os itens ficam alinhados na reta inferior da célula do grid.

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

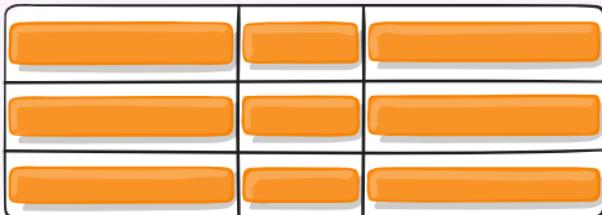
Resultado:



- **center**: os itens ficam centralizados verticalmente na célula do grid.

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

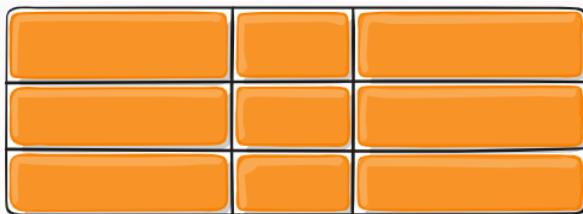
Resultado:



- **stretch**: os itens preenchem todo o espaço da célula do grid.

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

Resultado:



Esse mesmo comportamento para o alinhamento vertical dos itens pode ser definido individualmente para cada item do grid com a propriedade **align-self**.

Propriedade place-items

A propriedade **place-items** define os dois alinhamentos vertical (**align-items**) e horizontal (**justify-items**) com uma única declaração. Os valores possíveis para configurar essa propriedade são:

<align-items> / <justify-items>: O primeiro valor define o **align-items** e o segundo o **justify-items**. Se o segundo valor for omitido, o primeiro valor é atribuído para ambos os alinhamentos.

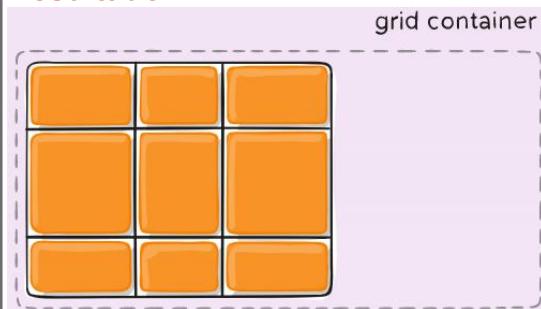
Propriedade justify-content

A propriedade **justify-content** alinha o grid completo ao longo do eixo horizontal do container do grid. Os valores possíveis para configurar essa propriedade são:

- **start**: o grid fica alinhado na reta esquerda do container.

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

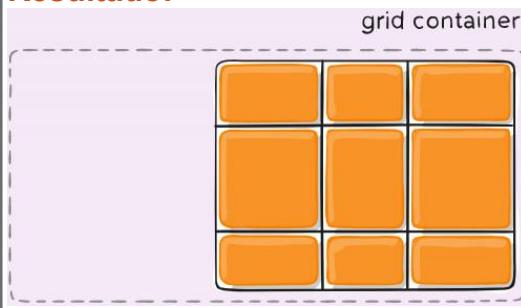
Resultado:



- **end**: o grid fica alinhado na reta direita do container.

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

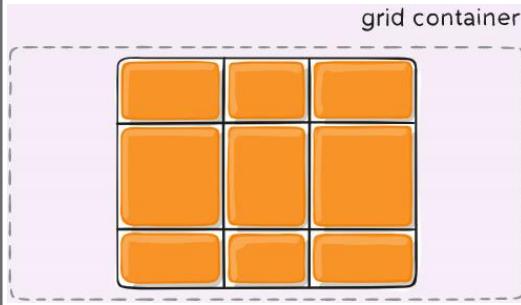
Resultado:



- **center**: o grid fica centralizado horizontalmente no container.

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

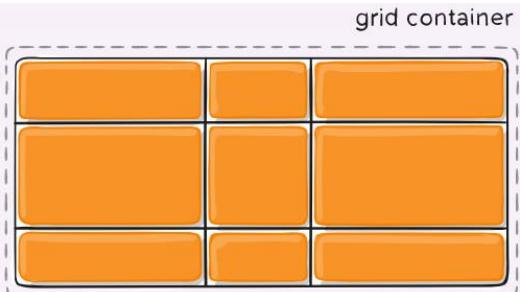
Resultado:



- **stretch**: o grid preenche todo o espaço do container.

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

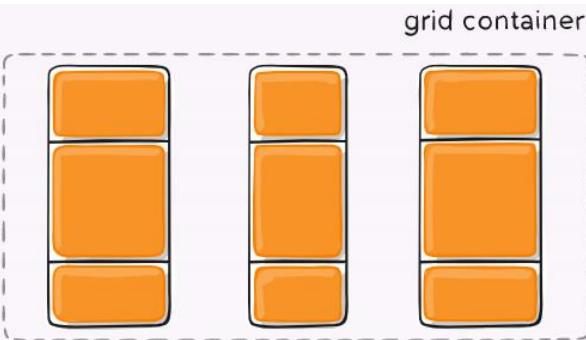
Resultado:



- **space-around**: coloca uma quantidade uniforme de espaço entre cada item do grid e metade do tamanho do espaço nas extremidades da esquerda e da direita.

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

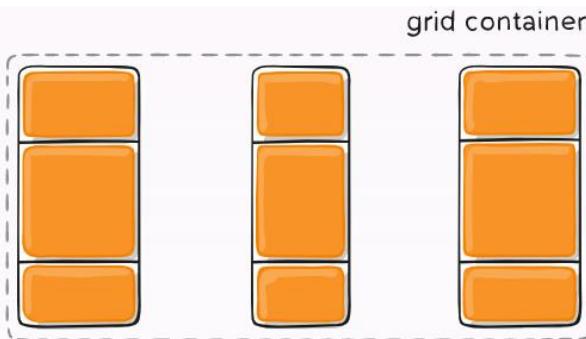
Resultado:



- **space-between**: coloca uma quantidade uniforme de espaço entre cada item do grid e sem espaço nas extremidades da esquerda e da direita.

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

Resultado:

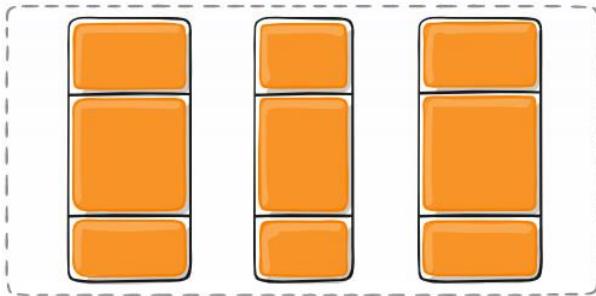


- **space-evenly**: coloca uma quantidade uniforme de espaço entre cada item do grid, incluindo as extremidades da esquerda e da direita com mesmo tamanho de espaço.

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

Resultado:

grid container



Propriedade align-content

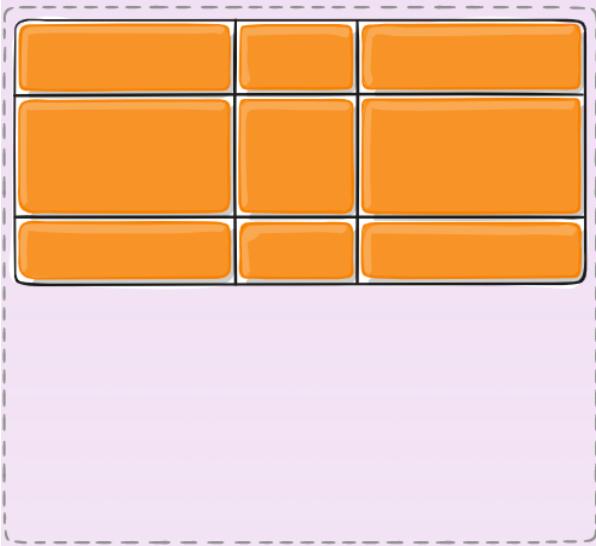
A propriedade **align-content** alinha o grid completo ao longo do eixo vertical do container do grid. Os valores possíveis para configurar essa propriedade são:

- **start**: o grid fica alinhado na borda superior do container.

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

Resultado:

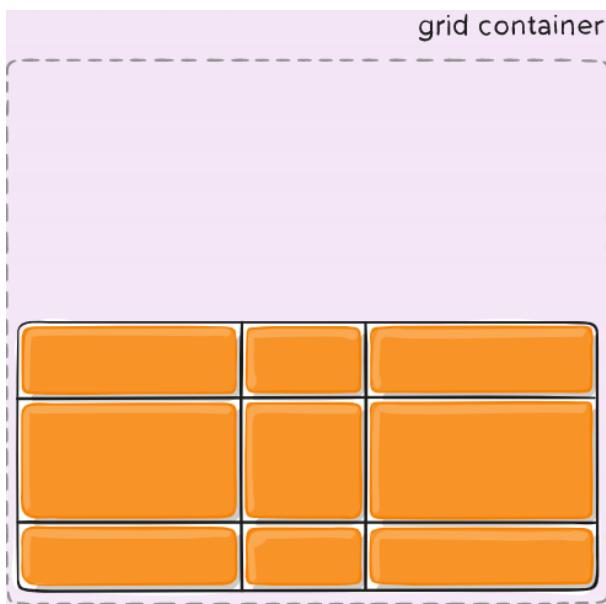
grid container



- **end**: o grid fica alinhado na borda inferior do container.

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

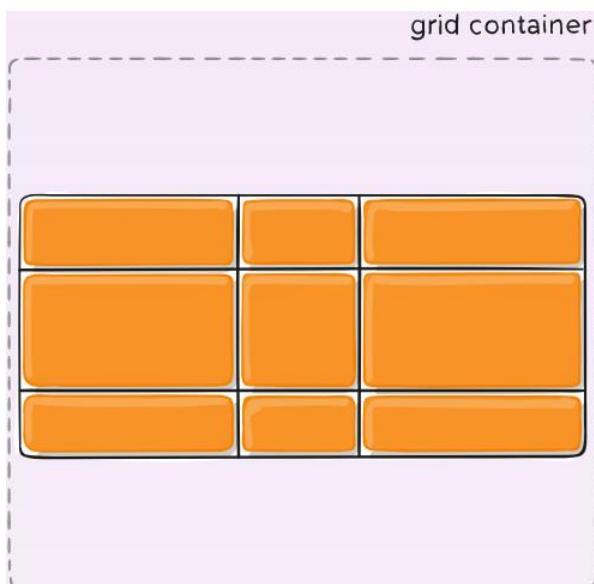
Resultado:



- **center**: o grid fica centralizado verticalmente no container.

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

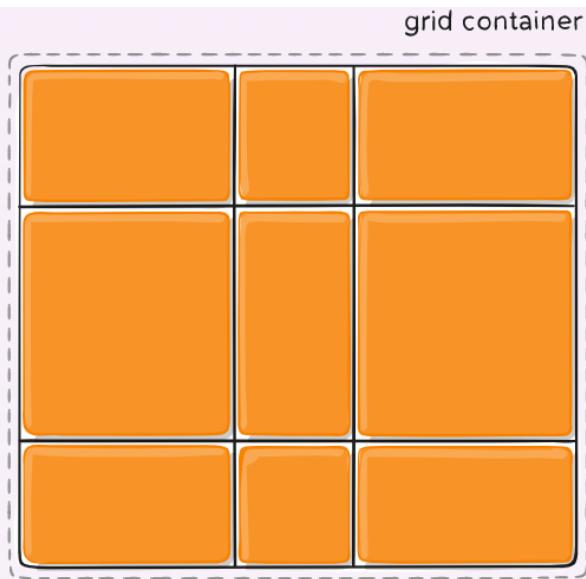
Resultado:



- **stretch**: o grid preenche todo o espaço do container.

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

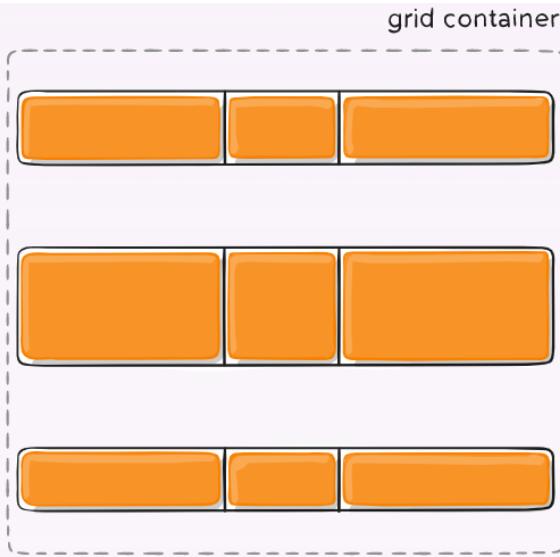
Resultado:



- **space-around**: coloca uma quantidade uniforme de espaço entre cada item do grid e metade do tamanho do espaço nas extremidades superior e inferior.

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

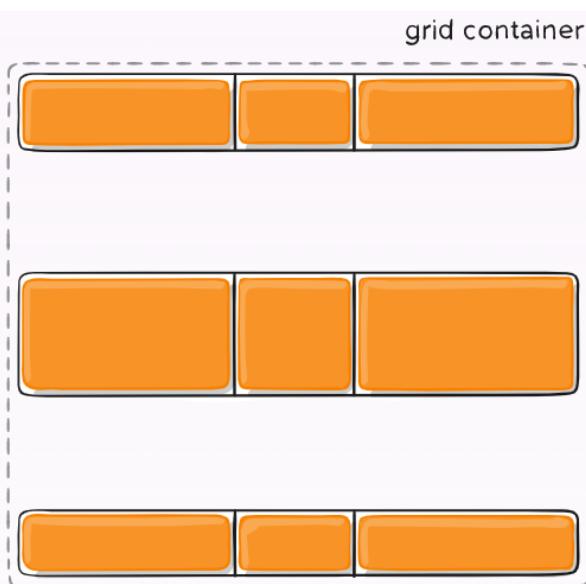
Resultado:



- **space-between**: coloca uma quantidade uniforme de espaço entre cada item do grid e sem espaço nas extremidades superior e inferior.

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

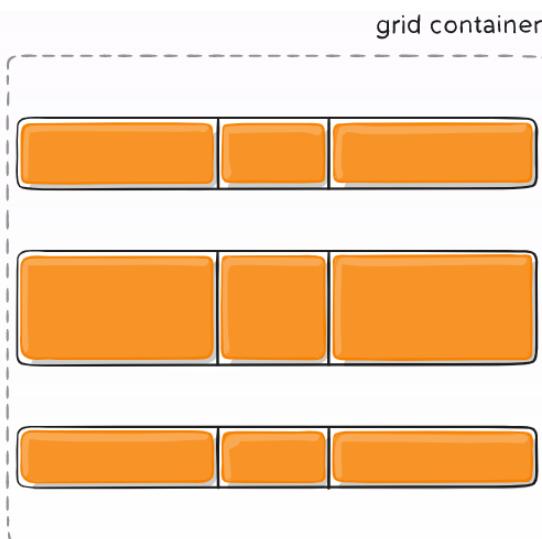
Resultados:



- **space-evenly**: coloca uma quantidade uniforme de espaço entre cada item do grid, incluindo as extremidades superior e inferior com mesmo tamanho de espaço.

```
.container {
  align-content: space-evenly;
}
```

Resultado:



Propriedade place-content

A propriedade **place-content** define os dois alinhamentos **align-content** e **justify-content** com uma única declaração. Os valores possíveis para configurar essa propriedade são:

<align-content> / <justify-content>: O primeiro valor define o **align-content** e o segundo o **justify-content**. Se o segundo valor for omitido, o primeiro valor é atribuído para ambos os alinhamentos.

Propriedades grid-auto-columns e grid-auto-rows

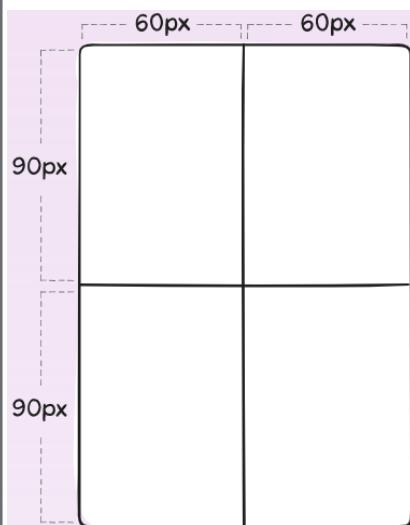
As propriedades **grid-auto-columns** e **grid-auto-rows** especificam o tamanho de quaisquer trilhas do grids geradas automaticamente, que podem também ser chamadas de trilha implícitas do grid. Trilhas implícitas são criadas quando há mais itens no grid do que células disponíveis ou quando um item do grid é colocado fora da grade explícita. Os valores possíveis para configurar essa propriedade são:

<track-size>: que pode ser um tamanho, uma porcentagem ou uma fração (usando a unidade **fr**) do espaço livre no grid.

Para ilustrar melhor, vamos ver como trilhas implícitas são criadas. Vamos inicialmente criar um grid de 2x2:

```
.container {
    grid-template-columns: 60px 60px;
    grid-template-rows: 90px 90px;
}
```

Resultado:

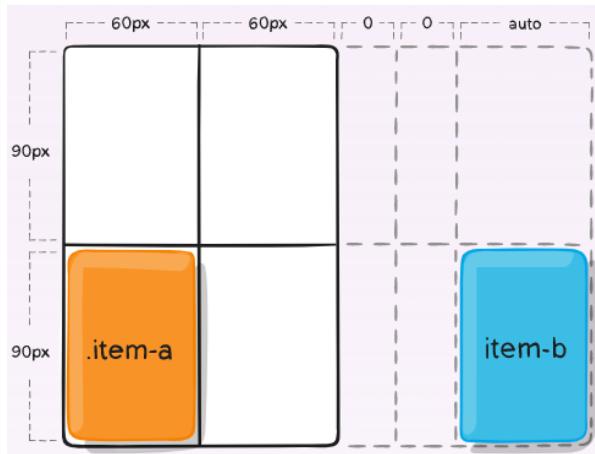


Agora, imagine que você use as propriedades **grid-column** e **grid-row** para posicionar os itens da seguinte forma.

```
.item-a {
    grid-column: 1 / 2;
    grid-row: 2 / 3;
}

.item-b {
    grid-column: 5 / 6;
    grid-row: 2 / 3;
}
```

Resultado:



Nesse caso, o item-b começa na reta da coluna 5 e termina na reta coluna 6, mas nós não definimos anteriormente essa parte (nossa grid é 2x2). Sendo assim, ao referenciar uma reta não existente teremos trilhas implícitas criadas com tamanho 0 para o gap de trilhas.

Caso quisermos, nós podemos mudar a primeira declaração, quando criamos o grid 2x2 na classe **.container** para que sejam criadas trilhas implícitas com tamanho especificado, isso é o que as propriedades **grid-auto-columns** e **grid-auto-rows** fazem. Por exemplo:

```
.container {
    grid-auto-columns: 60px;
}
```



Assim, as trilhas implícitas são criadas com tamanho fixo definido de 60px.

Propriedade grid-auto-flow

Se você tiver itens do grid que não foram colocados explicitamente no grid, o algoritmo de auto posicionamento entra em ação para colocar os itens automaticamente. A propriedade **grid-auto-flow** controla como o algoritmo funciona. Os valores possíveis para configurar essa propriedade são:

- **row** (default): define que o algoritmo de auto posicionamento irá preencher cada linha por vez, adicionando novas linhas se necessário.
- **end**: define que o algoritmo de auto posicionamento irá preencher cada coluna por vez, adicionando novas colunas se necessário.
- **dense**: define que o algoritmo de auto posicionamento use um formato de compactação "denso", ou seja, ele irá tentar preencher primeiro as lacunas no grid se itens menores surgirem.

Vamos entender melhor. Suponha que tenhamos o seguinte trecho no HTML:

```
<section class="container">
  <div class="item-a">item-a</div>
  <div class="item-b">item-b</div>
  <div class="item-c">item-c</div>
  <div class="item-d">item-d</div>
  <div class="item-e">item-e</div>
</section>
```

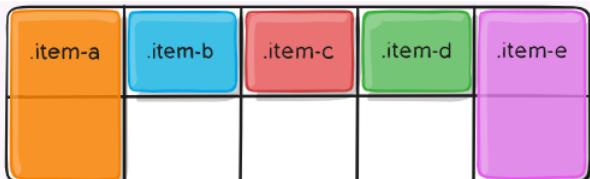
E no CSS, tenhamos o grid com cinco colunas e duas linhas e define a propriedade **grid-auto-flow** com o valor **row**.

```
.container {
  display: grid;
  grid-template-columns: 60px 60px 60px 60px 60px;
  grid-template-rows: 30px 30px;
  grid-auto-flow: row;
}
```

Então, quando colocamos os itens no grid, especificamos os pontos para dois desses itens.

```
.item-a {
  grid-column: 1;
  grid-row: 1 / 3;
}
.item-e {
  grid-column: 5;
  grid-row: 1 / 3;
}
```

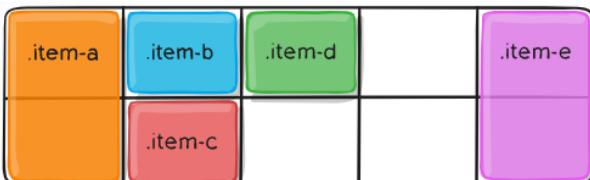
Como a propriedade **grid-auto-flow** foi configurada com o valor **row**, o nosso grid será exibido com os três itens (item-b, item-c e item-d) no grid, que nós não especificamos, posicionados pela linha.



Os itens não especificados posicionados na linha.

Se ao invés do valor **row** colocássemos o valor **column**, os itens seriam posicionados nas colunas.

```
.container {
    display: grid;
    grid-template-columns: 60px 60px 60px 60px 60px;
    grid-template-rows: 30px 30px;
    grid-auto-flow: column;
}
```



Os itens não especificados posicionados na coluna.

Propriedade grid

A propriedade **grid** é uma abreviação para configurar todas as seguintes propriedades usando uma única declaração: **grid-template-rows**, **grid-template-columns**, **grid-template-areas**, **grid-auto-rows**, **grid-auto-columns** e **grid-auto-flow**.

Por exemplo, os dois trechos de código definem a mesma configuração:

```
.container {
    grid: 100px 300px / 3fr 1fr;
}

.container {
    grid-template-rows: 100px 300px;
    grid-template-columns: 3fr 1fr;
}
```

Outro exemplo equivalente é:

```
.container {  
    grid: auto-flow / 200px 1fr;  
}  
  
.container {  
    grid-auto-flow: row;  
    grid-template-columns: 200px 1fr;  
}
```

Mais um exemplo de trecho de código equivalente:

```
.container {  
    grid: auto-flow dense 100px / 1fr 2fr;  
}  
  
.container {  
    grid-auto-flow: row dense;  
    grid-auto-rows: 100px;  
    grid-template-columns: 1fr 2fr;  
}
```

Vamos para um último exemplo de trecho equivalentes:

```
.container {  
    grid: 100px 300px / auto-flow 200px;  
}  
  
.container {  
    grid-template-rows: 100px 300px;  
    grid-auto-flow: column;  
    grid-auto-columns: 200px;  
}
```

Propriedades dos elementos filhos (Grid Items)

Vamos, agora, para as propriedades dos itens do Grid Layout. Elas são **grid-column-start**, **grid-column-end**, **grid-row-start**, **grid-row-end**, **grid-column**, **grid-row**, **grid-area**, **align-self** e **place-self**.

Propriedades **grid-column-start**, **grid-column-end**, **grid-row-start** e **grid-row-end**

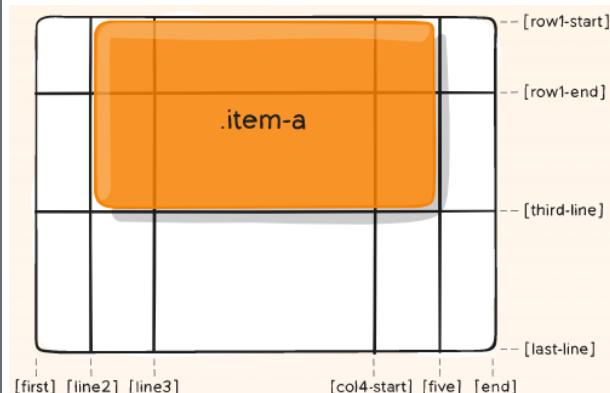
Essas propriedades determinam a localização de um item de acordo com uma reta específica no grid. As propriedades **grid-column-start** e **grid-row-start** definem a reta onde a localização do

item irá começar e as propriedades **grid-column-end** e **grid-row-end** onde a localização do item irá terminar. Os valores possíveis são:

- **<reta>**: pode ser o número da reta no grid ou o nome dela.
- **span <number>**: o item irá ocupar a quantidade de trilhas que for especificado no parâmetro **<number>**.
- **span <name>**: o item irá ocupar a quantidade de trilhas até a reta com o nome especificada no parâmetro **<name>**.
- **auto**: indica auto posicionamento.

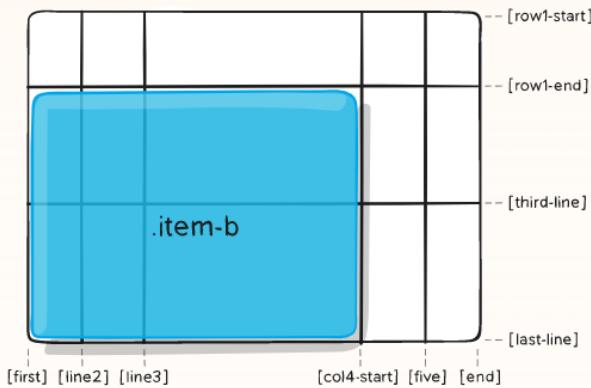
Vejamos alguns exemplos e seus respectivos resultados. No primeiro exemplo, o item-a tem seu início na reta vertical **dois** (**grid-column-start: 2**) e na **primeira** reta horizontal (**grid-row-start: row1-start**) e seu final na reta vertical **cinco** (**grid-column-end: five**) e na reta horizontal **três** (**grid-row-end: 3**). Note, que você pode usar valores numéricos, nome por extenso e o rótulo da reta.

```
.item-a {
    grid-column-start: 2;
    grid-column-end: five;
    grid-row-start: row1-start;
    grid-row-end: 3;
}
```



Nesse outro exemplo, vamos definir o item-b com seu início na **primeira** reta vertical (**grid-column-start: 1**) e na reta horizontal **dois** (**grid-row-start: 2**) e seu final na reta vertical **quatro** (**grid-column-end: span col4-start**) e na reta horizontal ocupa duas trilhas (**grid-row-end: span 2**). Note, que nesse exemplo foi usado o span das duas formas descritas acima com o rótulo da reta e com um valor numérico.

```
.item-b {
    grid-column-start: 1;
    grid-column-end: span col4-start;
    grid-row-start: 2;
    grid-row-end: span 2;
}
```

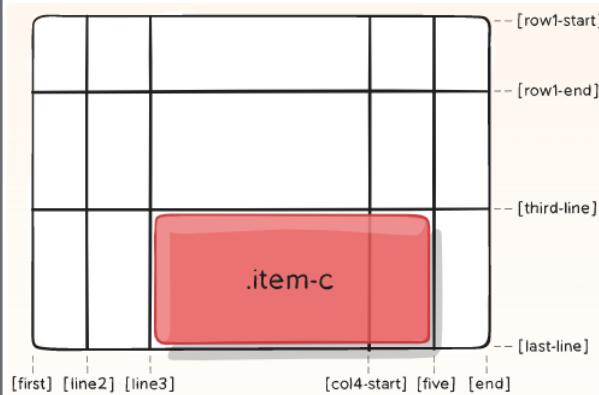


Propriedades `grid-column` e `grid-row`

Essas propriedades são abreviações para a `grid-column-start` + `grid-column-end` e `grid-row-start` e `grid-row-end` respectivamente. Os valores possíveis são `<start-line>` / `<end-line>` (reta inicial / reta final).

Vejamos o exemplo e seu respectivo resultado. Nesse exemplo, o item-c tem ocupado verticalmente **dois** espaços a partir da reta vertical **dois** (`grid-column: 2 / span 2`) e horizontalmente vai da **terceira** reta horizontal até a reta **quatro** (`grid-row: third-line / 4`).

```
.item-c {
    grid-column: 2 / span 2;
    grid-row: third-line / 4;
}
```



Propriedade `grid-area`

A propriedade `grid-area` define um nome/rótulo para o item do grid para que ele possa ser usado com a propriedade `grid-template-area`. Essa propriedade pode também ser usada como uma abreviação para configurar `grid-row-start` + `grid-column-start` + `grid-row-end` + `grid-column-end` em uma única declaração. Os valores possíveis são:

- `<name>`: nome que você escolher para o item.

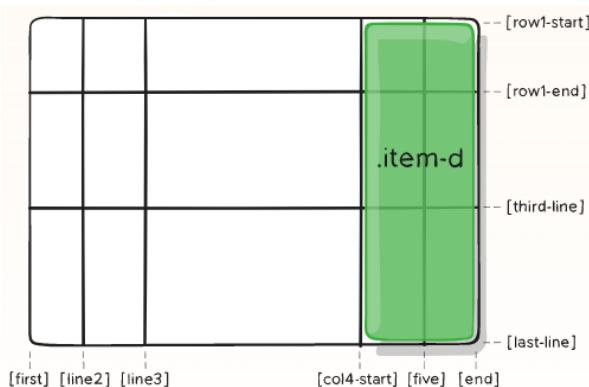
- **<row-start>** / **<column-start>**: retas horizontal e vertical iniciais. Você pode usar o nome ou número das retas.
- **<row-end>** / **<column-end>**: retas horizontal e vertical finais. Você pode usar o nome ou número das retas.

Vejamos o exemplo abaixo, onde o item-d é configurado com o rótulo header:

```
.item-d {
    grid-area: header;
}
```

No exemplo a seguir, é definido os valores **grid-row-start + grid-column-start + grid-row-end + grid-column-end** do item-d. Veja o resultado da configuração.

```
.item-d {
    grid-area: 1 / col4-start / last-line / 6;
}
```

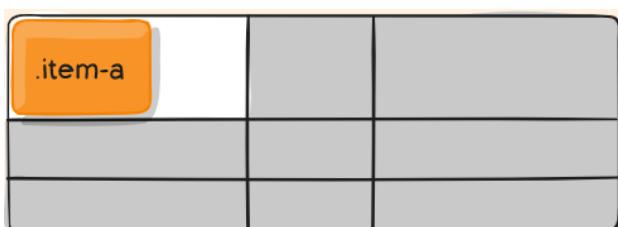


Propriedade justify-self

A propriedade **justify-self** alinha o item horizontalmente na célula. A diferença entre a **justify-self** e a **justify-items** é que a primeira configura o alinhamento horizontal de um item e a segunda de todos os itens do grid. Os valores possíveis para configurar essa propriedade são:

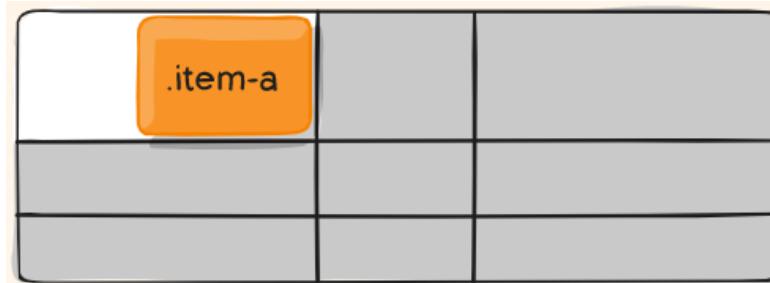
- **start**: o item fica alinhado na reta esquerda da célula do grid.

```
.item-a {
    justify-self: start;
}
```



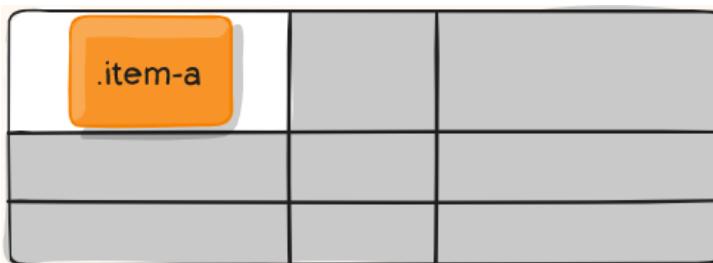
- **end**: o item fica alinhado na reta direita da célula do grid.

```
.item-a {  
  justify-self: end;  
}
```



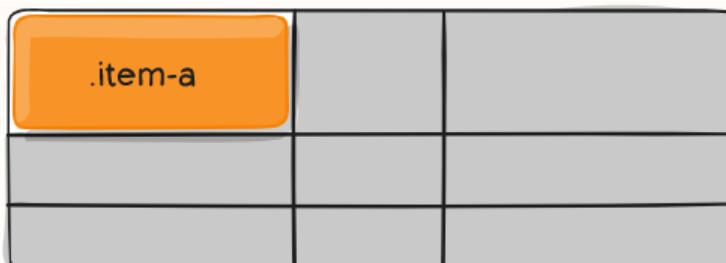
- **center**: o item fica centralizado horizontalmente na célula do grid.

```
.item-a {  
  justify-self: center;  
}
```



- **stretch**: o item preenche todo o espaço da célula do grid.

```
.item-a {  
  justify-self: stretch;  
}
```

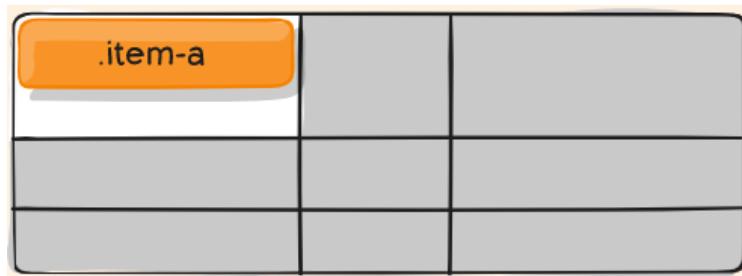


Propriedade align-self

A propriedade **align-self** alinha o item verticalmente na célula. A diferença entre a **align-self** e a **align-items** é que a primeira configura o alinhamento vertical de um item e a segunda de todos os itens do grid. Os valores possíveis para configurar essa propriedade são:

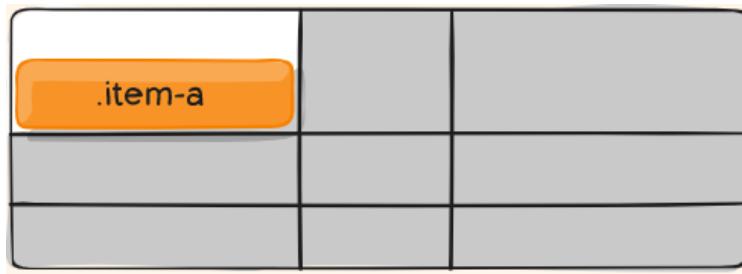
- **start**: o item fica alinhado na reta superior da célula do grid.

```
.item-a {
  align-self: start;
}
```



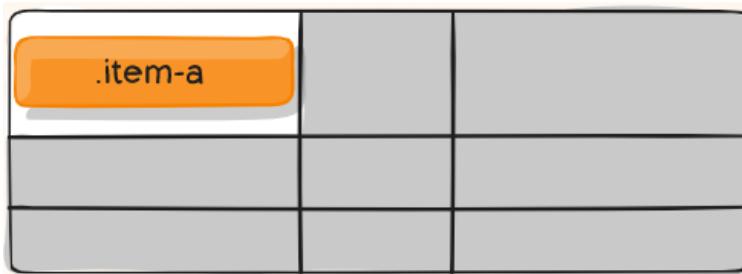
- **end**: o item fica alinhado na reta inferior da célula do grid.

```
.item-a {
  align-self: end;
}
```



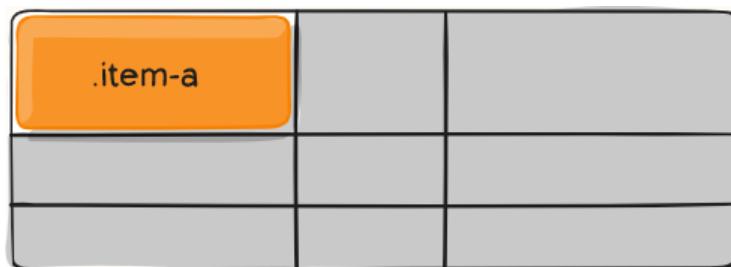
- **center**: o item fica centralizado verticalmente na célula do grid.

```
.item-a {
  align-self: center;
}
```



- **stretch**: o item preenche todo o espaço da célula do grid.

```
.item-a {
  align-self: stretch;
}
```



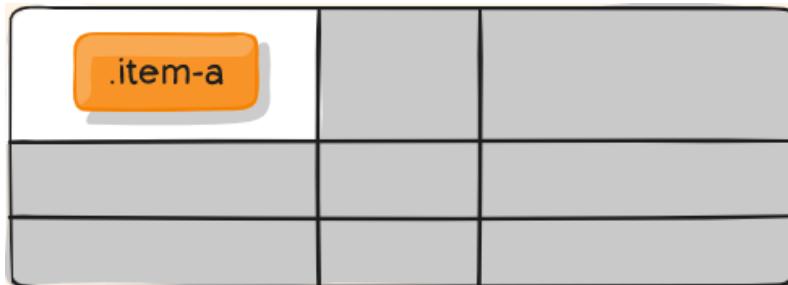
Propriedade place-self

A propriedade **place-self** configura ambas as propriedades **align-self** e **justify-self** em uma única declaração. Os valores possíveis são:

- **auto**: usa o alinhamento padrão para o modelo de layout.
- **<align-self> / <justify-self>**: o primeiro valor é para o alinhamento vertical e segundo valor é para o alinhamento horizontal dos itens. Se o segundo valor for omitido, o primeiro valor será atribuído para ambos os alinhamentos.

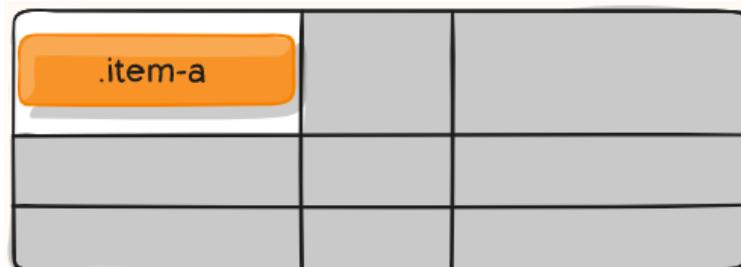
Vejamos um primeiro exemplo para centralizar o item verticalmente e horizontalmente na célula.

```
.item-a {
  place-self: center;
}
```



O próximo exemplo centraliza verticalmente o item na célula e o faz ocupar todo o espaço horizontal.

```
.item-a {  
    place-self: center stretch;  
}
```



Algumas considerações finais sobre Grid Layout

Ao dimensionar o tamanho das linhas e das colunas, você pode usar não somente as medidas comuns do CSS (px, rem, %, etc), mas você pode também usar Keywords, como: min-content, max-content, auto. E talvez unidade fracionárias mais úteis. Por exemplo, grid-template-columns: 200px 1fr 2fr min-content;

Você também pode acessar uma função, que pode auxiliar para definir os limites das linhas e das colunas. Por exemplo, para definir que uma coluna irá ocupar uma fração do espaço livre (1fr), mas que não irá passar de 200px, você pode usar a declaração **grid-template-columns: 1fr minmax(200px, 1fr);**

Existe também a função repeat(), que economiza digitar vários tamanhos de colunas iguais. Por exemplo, para definir 10 colunas de uma fração do espaço livre do grid, podemos fazer grid-template-columns: repeat(10, 1fr);

Combinar todas essas coisas pode ser extremamente produtivo, como grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

Vamos praticar I

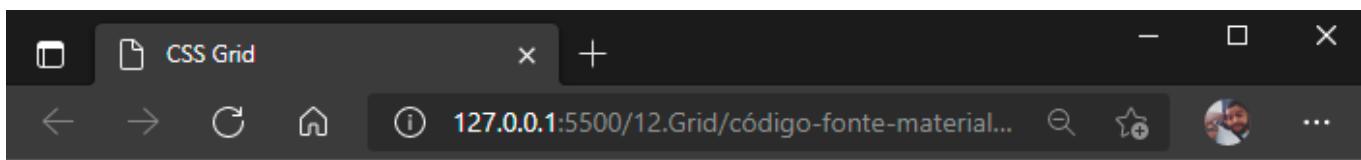
Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>  
<html lang="pt-br">  
  
<head>  
    <meta charset="UTF-8">  
    <title>CSS Grid</title>  
    <link rel="stylesheet" href="estilo.css">  
</head>  
  
<body>
```

```
<section class="wrapper">
  <article>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae iure bl
    anditiis omnis, tenetur eum
      impedit sint tempora repellat minus adipisci nemo autem inventore eveniet am
      et odit, deserunt vel animi
        perferendis soluta totam nesciunt magnam officia? Minima pariatur aperiam mo
        lestias veritatis odio adipisci
          laborum, ipsum consequatur soluta deleniti sapiente voluptatum ullam enim hi
          c similiqe nobis et earum qui
            quos libero suscipit consectetur, amet numquam aspernatur? Asperiores minima
            in magnam odit maiores veniam
              atque unde, error quibusdam commodi repellendus similiqe reprehenderit labo
              rum.
    </article>
    <article>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam minus, ex v
      ero inventore magni harum.
      Placeat tempore iste itaque cupiditate fugiat corrupti assumenda reiciendis
      at, adipisci, perspiciatis
        dicta? Omnis, atque!
    </article>

    <article>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae iure bl
      anditiis omnis, tenetur eum
        impedit sint tempora repellat minus adipisci nemo autem inventore eveniet am
        et odit, deserunt vel animi
          perferendis soluta totam nesciunt magnam officia? Minima pariatur aperiam mo
          lestias veritatis odio adipisci
            laborum, ipsum consequatur soluta deleniti sapiente voluptatum ullam enim hi
            c similiqe nobis et earum qui
              quos libero suscipit consectetur, amet numquam aspernatur? Asperiores minima
              in magnam odit maiores veniam
                atque unde, error quibusdam commodi repellendus similiqe reprehenderit labo
                rum.
    </article>
    <article>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam minus, ex v
      ero inventore magni harum.
      Placeat tempore iste itaque cupiditate fugiat corrupti assumenda reiciendis
      at, adipisci, perspiciatis
        dicta? Omnis, atque!
    </article>
  </section>
</body>

</html>
```



...adipisicing elit. Repudianda iure blanditiis omnis, tenetur eum impedit sint tempora repellat minus adipisci nemo autem inventore eveniet amet odit, deserunt vel animi perferendis soluta totam nesciunt magnam officia? Minima pariatur aperiam molestias veritatis odio adipisci laborum, ipsum consequatur soluta deleniti sapiente voluptatum ullam enim hic similique nobis et earum qui quos libero suscipit consectetur, amet numquam aspernatur? Asperiores minima in magnam odit maiores veniam atque unde, error quibusdam commodi repellendus similique reprehenderit laborum.

Ipse dicitur: *Quod si quis dicit deus est in mundo, non videt quod mundus in eo non est?* Quod si quis dicit deus in me, non videt quod et ipse in me non est? *Si ergo dicitis deum in vobis, non credam vobis.*

tempore iste haque cupiditate fugit, corrupti assumenda referendis ut, adipisci, perspicillans dicta. Omnis, atque
Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae iure blanditiis omnis, tenetur eum impedit sint
tempora repellat minus adipisci nemo autem inventore eveniet amet odit, deserunt vel animi perferendis soluta totam
nesciunt magnam officia? Minima pariatur aperiam molestias veritatis odio adipisci laborum, ipsum consequatur soluta
deleniti sapiente voluptatum ullam enim hic similique nobis et earum qui quos libero suscipit consectetur, amet
numquam aspernatur? Asperiores minima in magnam odit maiores veniam atque unde, error quibusdam commodi
repellendus similique reprehenderit laborum.

... ipsum dolor sit amet consectetur adipisicing elit. Aliquam minus, ex vero inventore magni harum. Placeat tempore iste itaque cupiditate fugiat corrupti assumenda reiciendis at, adipisci, perspiciatis dicta? Omnis, atque!

Observe que a página está totalmente sem formatação, vamos configurar alguns estilos. Para isso, abra um novo arquivo **CSS**.

Vamos começar a configurar a classe `.wrapper` que está associada ao elemento `<section>`. Insira o seguinte código no arquivo `CSS`:

```
@charset "utf-8";
.wrapper {
    display: grid;
    grid-template-columns: 70% 30%;
    grid-gap: 1em;
    overflow-x: hidden;
}
```

Nesse trecho, fazemos a configuração do Grid Layout como forma de exibição da página web (**display: grid;**), da largura das colunas (**grid-template-columns: 70% 30%;**), do espaçamento entre as linhas e colunas do grid (**grid-gap: 1em;**) e para não exibir nenhuma barra de rolagem no **eixo X (overflow-x: hidden;)**.

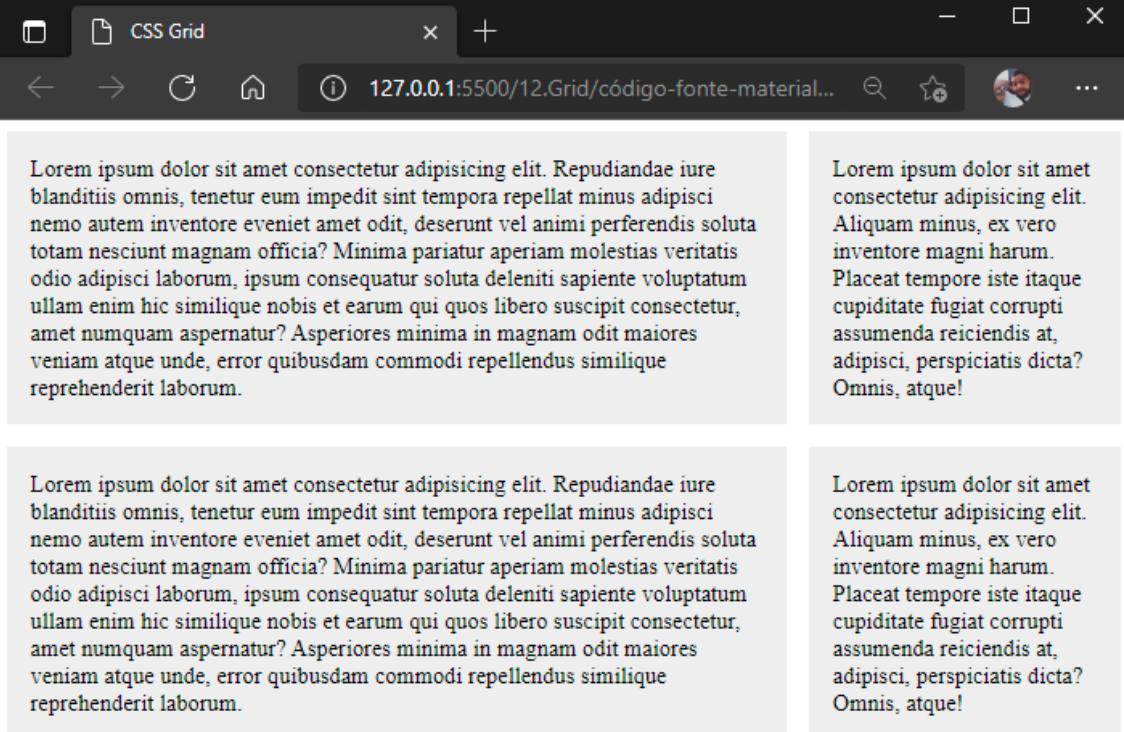


The screenshot shows a browser window with the title "CSS Grid". The URL is "127.0.0.1:5500/12.Grid/código-fonte-material...". The page displays a 2x2 grid of text blocks. The top-left block contains a long paragraph of placeholder text. The top-right block contains a shorter paragraph. The bottom-left block contains a long paragraph of placeholder text. The bottom-right block contains a shorter paragraph. The top-left block has a light gray background (#eee) and 1em padding.

Agora, vamos configurar todos os elementos **<article>** dentro da classe **.wrapper**, ou seja, que estão contidos no elemento **<section>**. Insira o seguinte código no arquivo estilo.css:

```
.wrapper>article {
    background: #eee;
    padding: 1em;
}
```

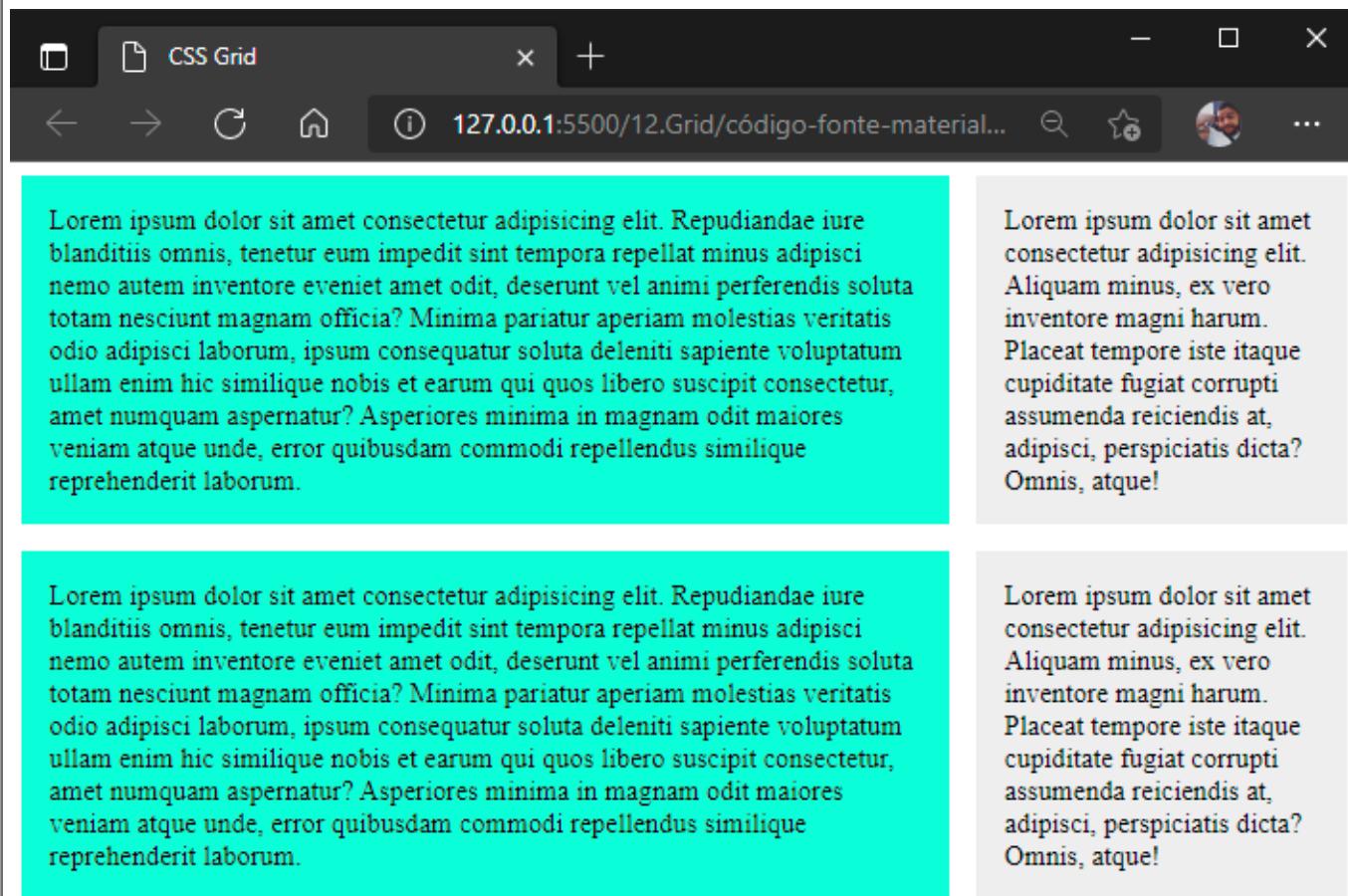
Nesse trecho, fazemos a configuração da **cor de plano de fundo do item do grid (background: #eee;)** e do espaçamento do elemento **<article>** (**padding: 1em;**).



The screenshot shows a browser window with the title "CSS Grid". The URL is "127.0.0.1:5500/12.Grid/código-fonte-material...". The page displays a 2x2 grid of text blocks. All four blocks now have a light gray background (#eee) and 1em padding. The text content remains the same as in the previous screenshot.

Em seguida, vamos utilizar a pseudoclasse **nth-child()** para configura a **cor de fundo dos itens ímpares do grid**.

```
.wrapper>article:nth-child(odd) {
    background: aquamarine;
}
```



Vamos praticar II

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

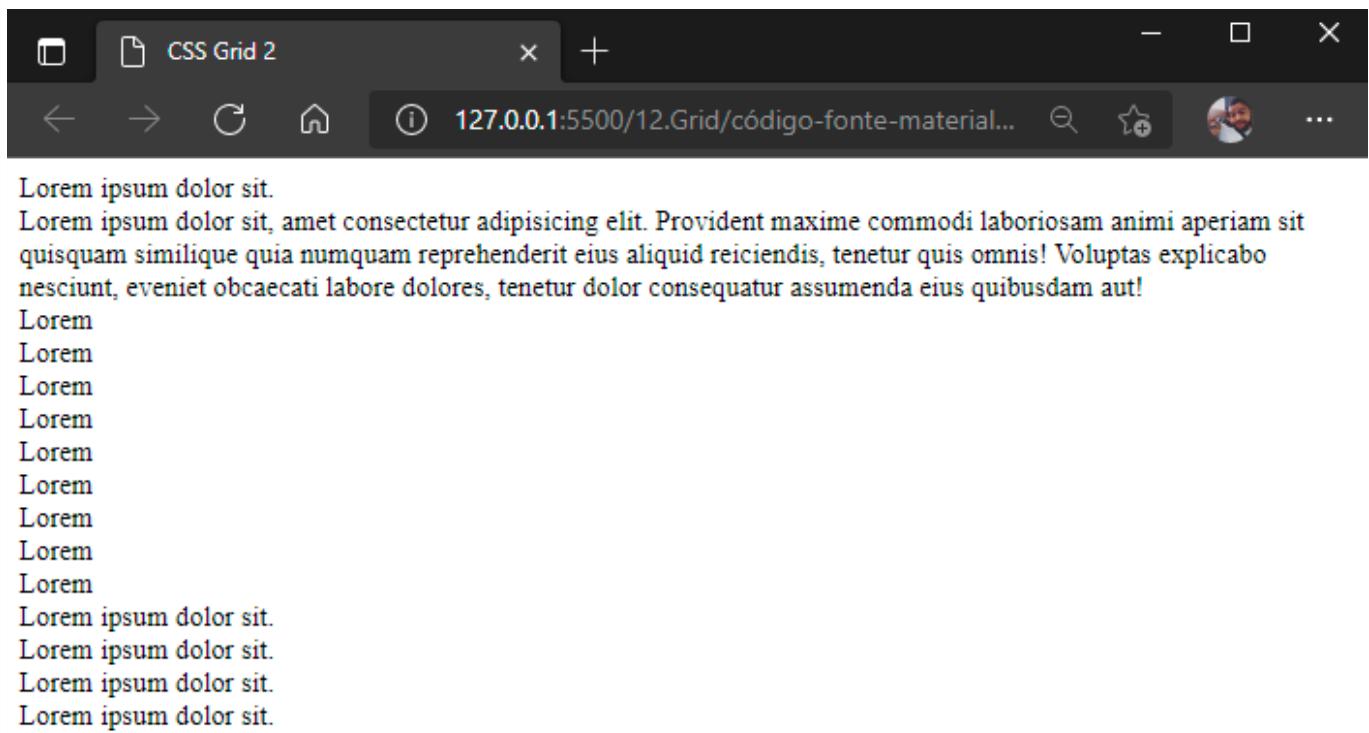
```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>CSS Grid 2</title>
    <link rel="stylesheet" href="02.estilo.css">
</head>

<body>
```

```
<section class="wrapper">
  <div>
    Lorem ipsum dolor sit.
  </div>
  <div>
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. Provident maxime c
ommodi laboriosam animi aperiam
      sit quisquam similique quia numquam reprehenderit eius aliquid reiciendis, t
enetur quis omnis! Voluptas
      explicabo nesciunt, eveniet obcaecati labore dolores, tenetur dolor consequa
tur assumenda eius quibusdam
      aut!
  </div>
  <div class="nested">
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
    <div>Lorem</div>
  </div>
  <div>
    Lorem ipsum dolor sit.
  </div>
</section>
</body>

</html>
```



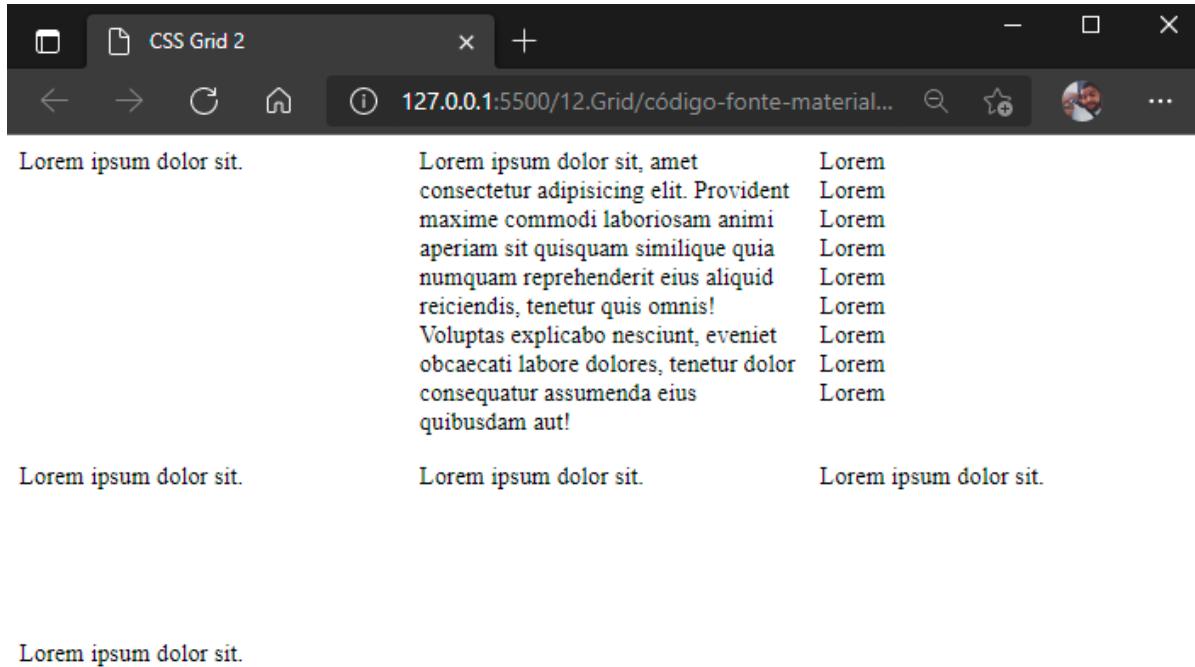
The screenshot shows a browser window with the title "CSS Grid 2". The address bar indicates the URL is 127.0.0.1:5500/12.Grid/código-fonte-material... . The page content consists of several lines of placeholder text ("Lorem ipsum dolor sit.") repeated across the screen, demonstrating a basic grid layout.

Observe que a página está totalmente sem formatação, vamos configurar alguns estilos. Para isso, abra um novo arquivo **CSS**. Vamos começar configurando a classe **.wrapper**. Insira o seguinte código no arquivo estilo.css.

```
@charset "utf-8";
.wrapper {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-gap: 1em;
    grid-auto-rows: minmax(100px, auto);
}
```

Nesse trecho, fazemos a configuração do Grid Layout como forma de exibição da página web (**display: grid;**), da largura das três colunas (**grid-template-columns: repeat(3, 1fr);**), do espaçamento entre as linhas e colunas do grid (**grid-gap: 1em;**) e para que o tamanho das linhas seja determinando automaticamente, mas que não seja menor do que 100px (**grid-auto-rows: minmax(100px, auto);**).

Salve o arquivo e visualize resultado da sua página web.

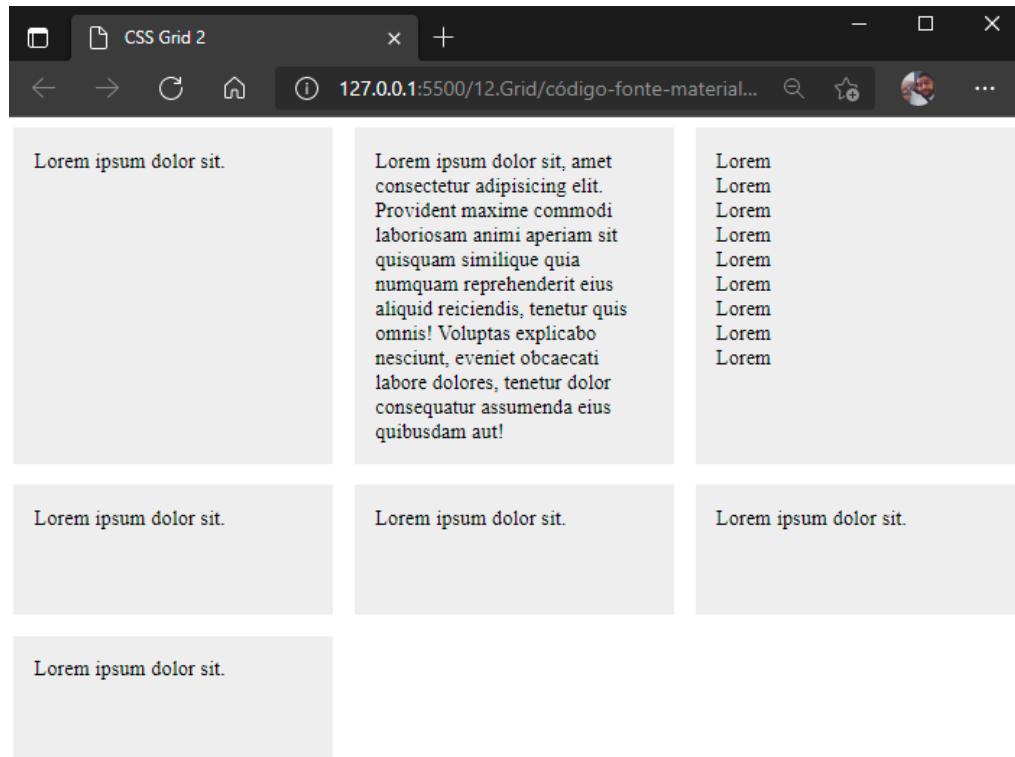


The screenshot shows a browser window titled "CSS Grid 2" with the URL "127.0.0.1:5500/12.Grid/código-fonte-material...". The page displays a CSS grid layout with 5 columns and 4 rows of content. The first row contains three items: "Lorem ipsum dolor sit.", a large block of Latin placeholder text, and another "Lorem ipsum dolor sit." item. The second row contains one item per column, all labeled "Lorem ipsum dolor sit.". The third row contains one item per column, all labeled "Lorem ipsum dolor sit.". The fourth row contains one item per column, all labeled "Lorem ipsum dolor sit.". The entire grid is contained within a container with a light gray background and 1em padding.

Em seguida, a **cor de fundo** e o **padding** de cada item do grid.

```
.wrapper>div {  
    background: #eee;  
    padding: 1em;  
}
```

Salve o arquivo e visualize resultado da sua página web.



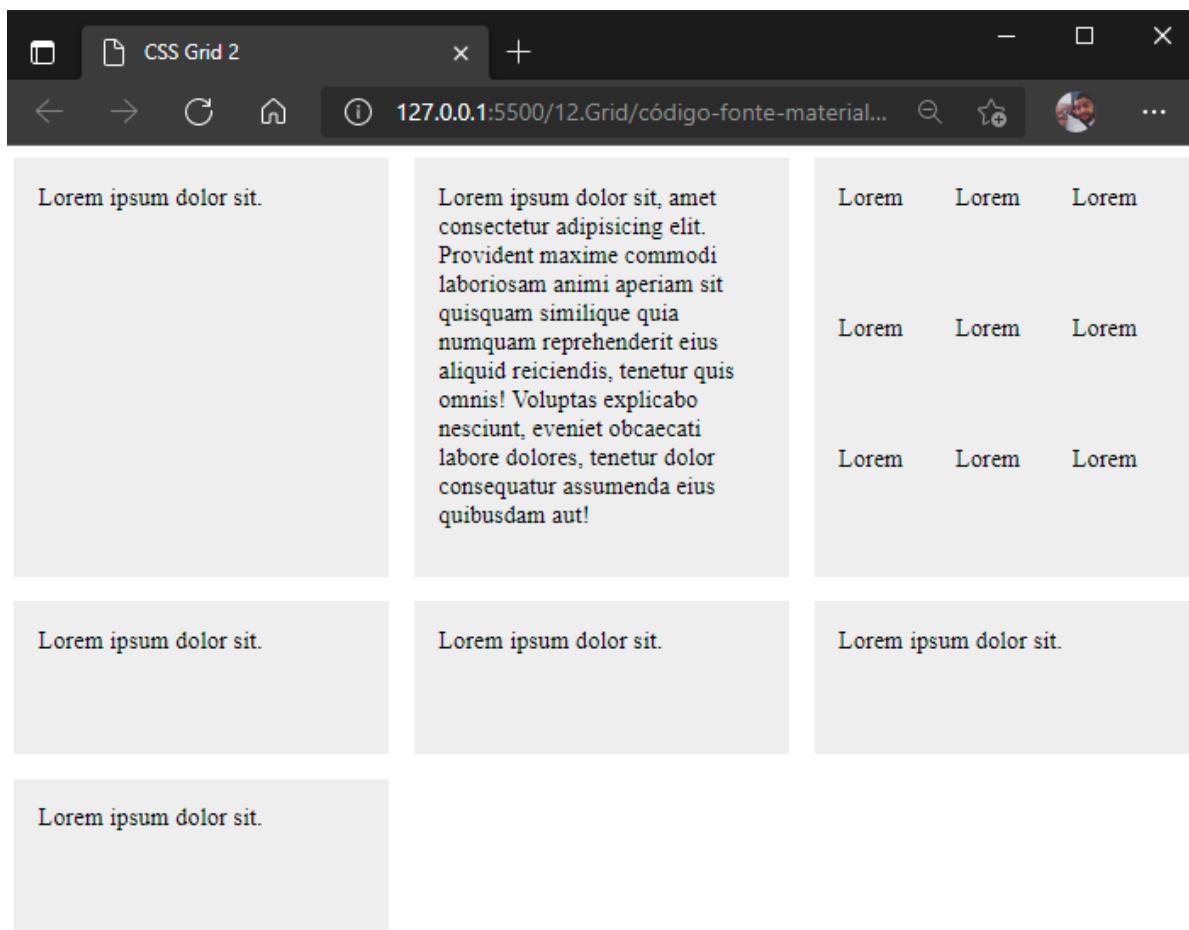
The screenshot shows a browser window titled "CSS Grid 2" with the URL "127.0.0.1:5500/12.Grid/código-fonte-material...". The page displays a CSS grid layout with 5 columns and 4 rows of content. The first row contains three items: "Lorem ipsum dolor sit.", a large block of Latin placeholder text, and another "Lorem ipsum dolor sit." item. The second row contains one item per column, all labeled "Lorem ipsum dolor sit.". The third row contains one item per column, all labeled "Lorem ipsum dolor sit.". The fourth row contains one item per column, all labeled "Lorem ipsum dolor sit.". The items in the grid now have distinct colors: the first item in each row is light red, the second is light blue, and the third is light green. The entire grid is contained within a container with a light gray background and 1em padding.

Em seguida, vamos configurar a classe **.nested**, que está associada ao terceiro elemento <div> da página. Vamos definir um grid dentro do grid já existente.

```
.nested {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-auto-rows: 70px;
    grid-gap: 1em;
}
```

Nesse trecho, fazemos a configuração do **Grid Layout** como forma de exibição da página web (**display: grid;**), da largura das três colunas (**grid-template-columns: repeat(3, 1fr);**), para que o tamanho das linhas (**grid-auto-rows: 70px**) e do espaçamento entre as linhas e colunas do grid (**grid-gap: 1em;**).

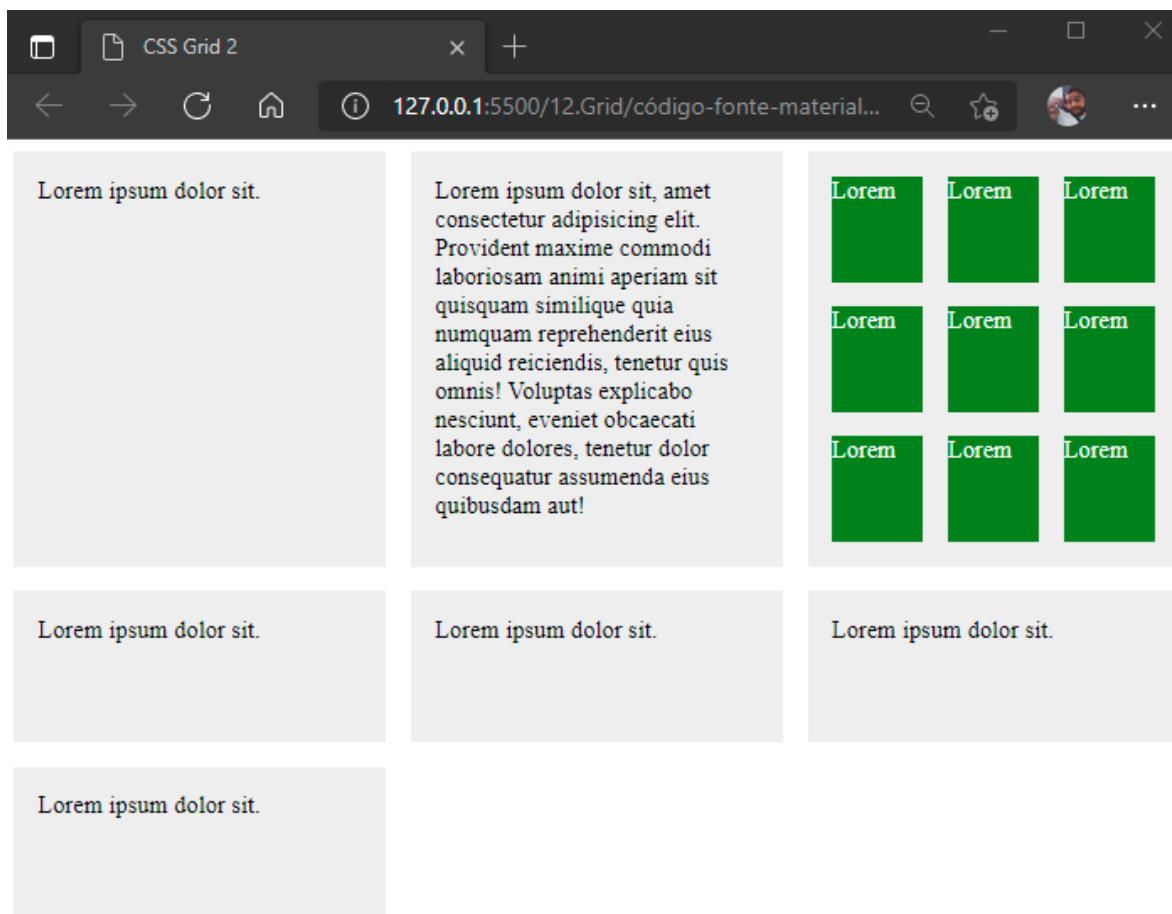
Salve o arquivo e visualize resultado da sua página web.



Por fim, vamos configurar a **cor de fundo** e a **cor da fonte de cada item de segundo grid**.

```
.nested>div {
    background: green;
    color: #fff;
}
```

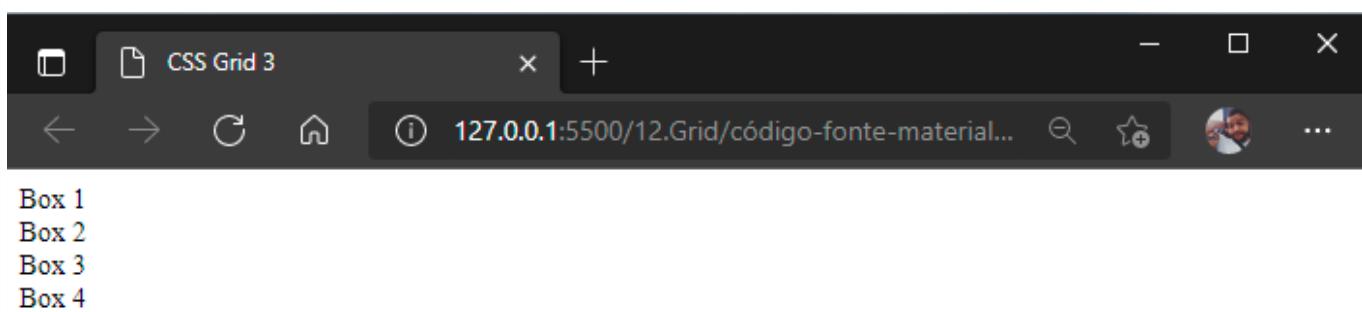
Salve o arquivo e visualize resultado da sua página web.



Vamos praticar III

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>CSS Grid 3</title>
    <link rel="stylesheet" href="estilo.css">
</head>
<body>
    <section class="wrapper">
        <div class="box box1">Box 1</div>
        <div class="box box2">Box 2</div>
        <div class="box box3">Box 3</div>
        <div class="box box4">Box 4</div>
    </section>
</body>
</html>
```

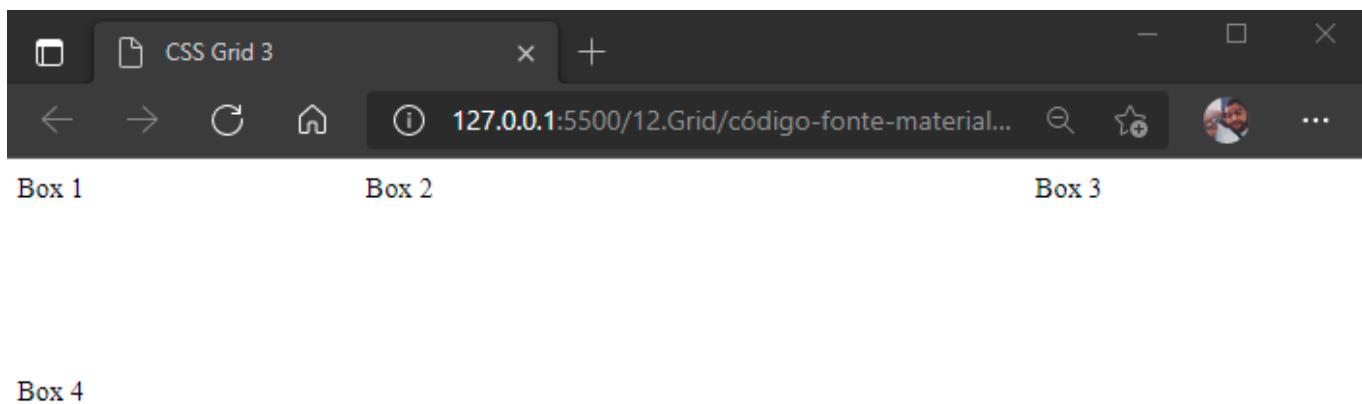


Observe que a página está totalmente sem formatação, vamos configurar alguns estilos. Para isso, abra um novo arquivo **CSS**. Vamos começar configurando a classe **.wrapper**. Insira o seguinte código no arquivo estilo.css.

```
@charset "utf-8";
.wrapper {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-auto-rows: minmax(100px, auto);
  grid-gap: 1em;
  justify-items: stretch;
  align-items: stretch;
}
```

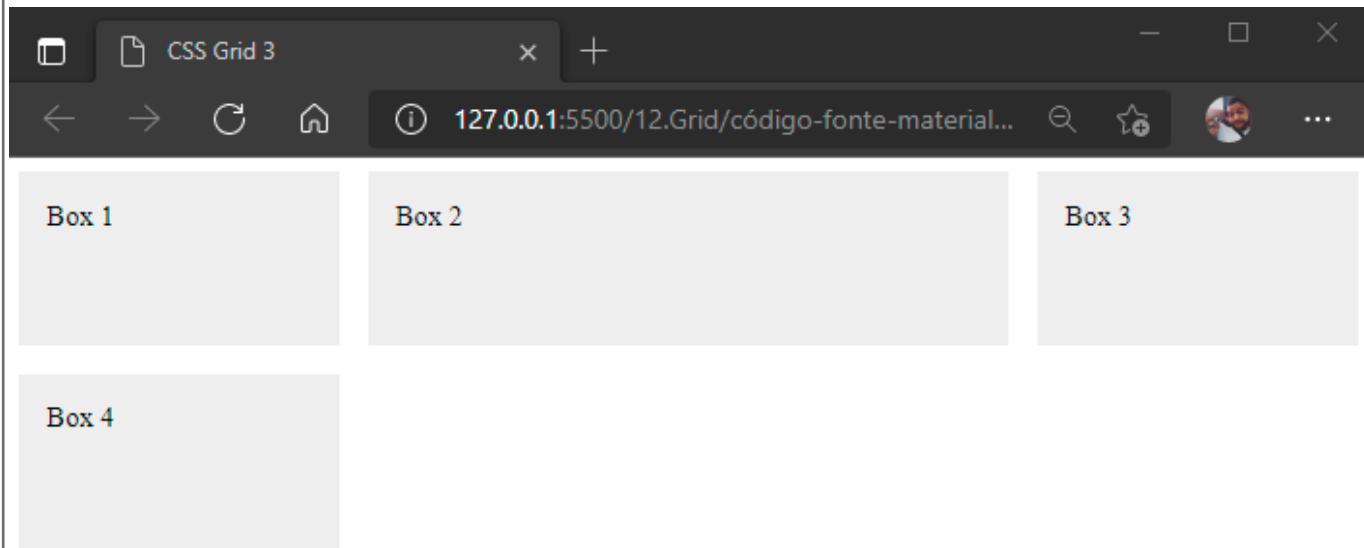
Nesse trecho, fazemos a configuração do Grid Layout como forma de exibição da página web (**display: grid;**), da largura das três colunas (**grid-template-columns: 1fr 2fr 1fr;**), para que o tamanho das linhas seja determinado automaticamente, mas que não seja menor do que 100px (**grid-auto-rows: minmax(100px, auto);**), do espaçamento entre as linhas e colunas do grid (**grid-gap: 1em;**) e dos alinhamentos horizontal (**justify-items: stretch;**) e vertical (**align-items: stretch;**) do itens no grid. Desse modo, definimos um grid de duas linhas e três colunas.

Salve o arquivo e visualize resultado da sua página web.



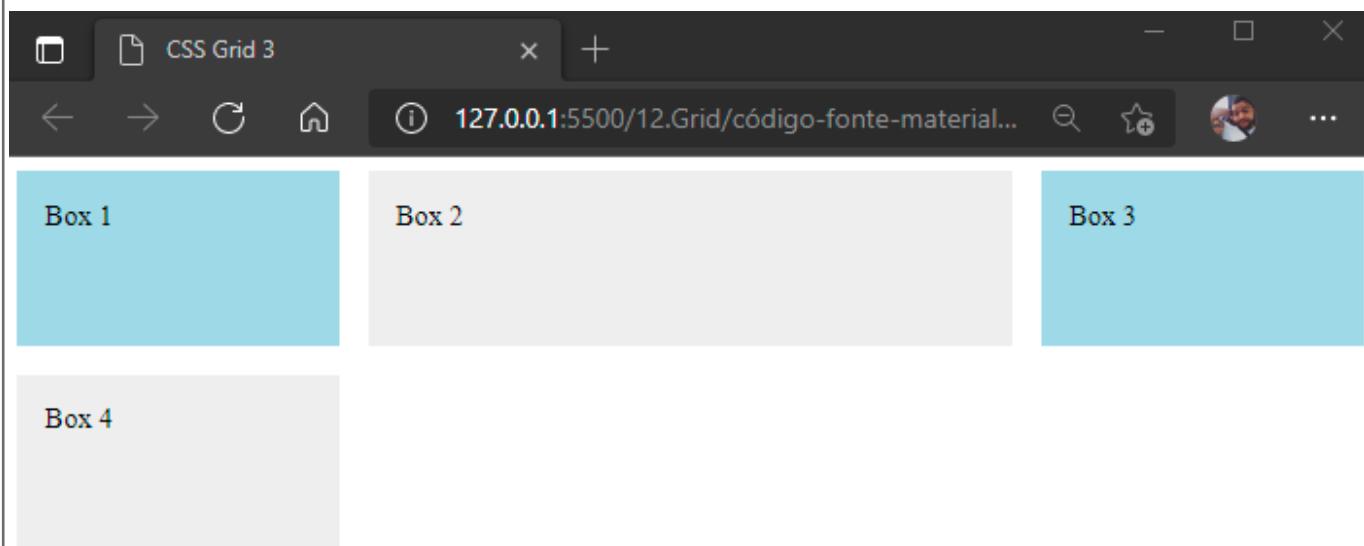
Agora, vamos configurar a **cor de fundo** e o **padding** de cada item do grid, ou seja, dos elementos `<div>` dentro da classe `.wrapper`.

```
.wrapper>div {  
    background: #eee;  
    padding: 1em;  
}
```



Em seguida, vamos utilizar a pseudoclasse **`nth-child()`** para configura a **cor de fundo dos itens ímpares do grid**.

```
.wrapper>div:nth-child(odd) {  
    background: lightblue;  
}
```



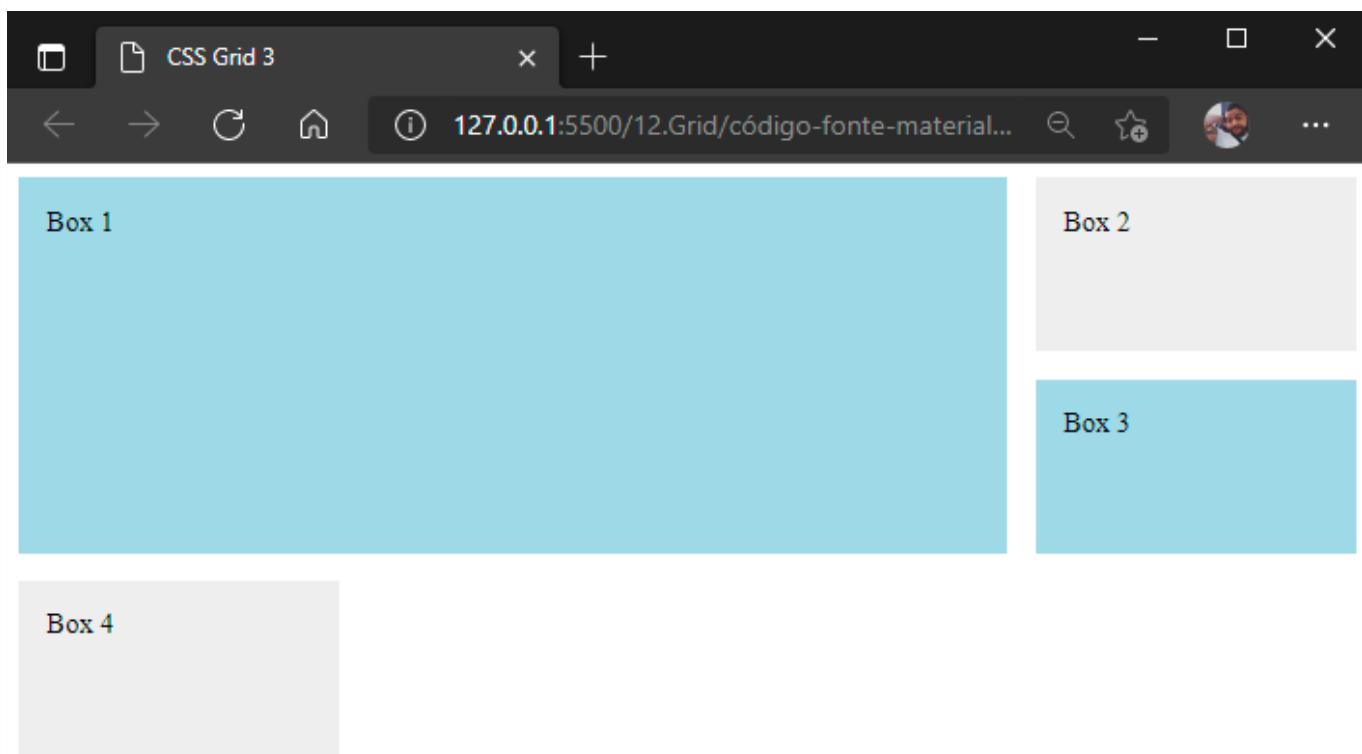
Continuando, vamos configurar a classe **.box1**, que está associada ao primeiro elemento **<div>** do elemento **<section>**.

```
.box1 {
    grid-column: 1 / 3;
    grid-row: 1 / 3;
}
```

Nesse trecho, fazemos a configuração da trilha ocupada pelo **primeiro item do grid**, o **Box 1**.

Nesse caso, estamos definindo que ele **inicie na reta vertical 1 e termina na reta vertical 3** (**grid-column: 1 / 3;**), assim, o **Box 1 irá ocupar duas colunas do grid**.

Também configuramos que ele inicie na **reta horizontal 1** e termina na **reta horizontal 3** (**grid-row: 1 / 3;**). Assim o **Box 1 irá ocupar duas linhas do grid**. Desse modo, **o Box 1 está ocupando quatro células do grid**.

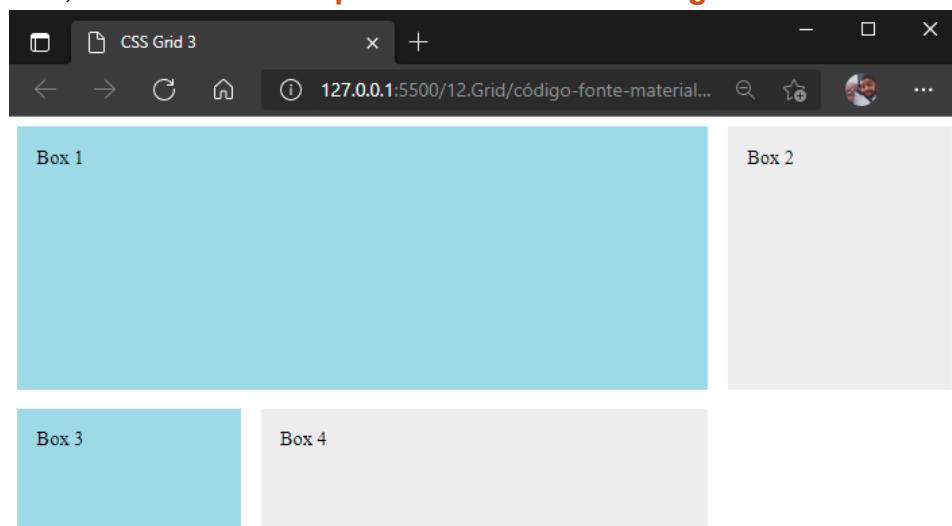


Note que os itens do grid vamos se reorganizando em trilhas implícitas não configuradas inicialmente (o **Box 4 ocupa a terceira linha do grid**). Isso acontece por causa do valor padrão da propriedade **grid-auto-flow**.

Depois, vamos configurar a classe **.box2**, que está associada ao segundo elemento **<div>** do elemento **<section>**.

```
.box2 {
    grid-column: 3;
    grid-row: 1 / 3;
}
```

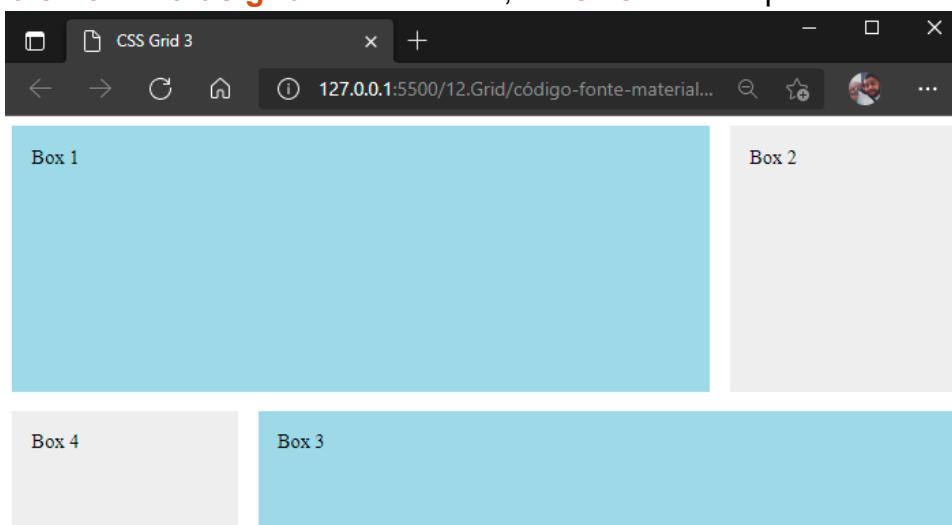
Nesse trecho, fazemos a configuração da trilha ocupada pelo **segundo item do grid**, o **Box 2**. Nesse caso, estamos definindo que ele **inicie na reta vertical 3** (`grid-column: 3;`), assim, o **Box 2 irá ocupar a última coluna do grid**. Também, configuramos que ele **inicie na reta horizontal 1 e termina na reta horizontal 3** (`grid-row: 1 / 3;`), assim, o **Box 2** irá ocupar **duas linhas do grid**. Desse modo, o **Box 2** está ocupando duas células do grid.



Em seguida, vamos configurar a classe `.box3`, que está associada ao terceiro elemento `<div>` do elemento `<section>`.

```
.box3 {
    grid-column: 2 / 4;
    grid-row: 3;
}
```

Nesse trecho, fazemos a configuração da trilha ocupada pelo **terceiro item do grid**, o **Box 3**. Nesse caso, estamos definindo que ele **inicie na reta vertical 2 e termine na reta vertical 4** (`grid-column: 2 / 4;`), assim, o **Box 3** irá ocupar a **segunda e terceira colunas do grid**. Também, configuramos para que ele **inicie na reta horizontal 1** (`grid-row: 1 / 3;`), assim, o **Box 3** irá ocupar a **última linha do grid**. Desse modo, o **Box 3** está ocupando duas células do grid.

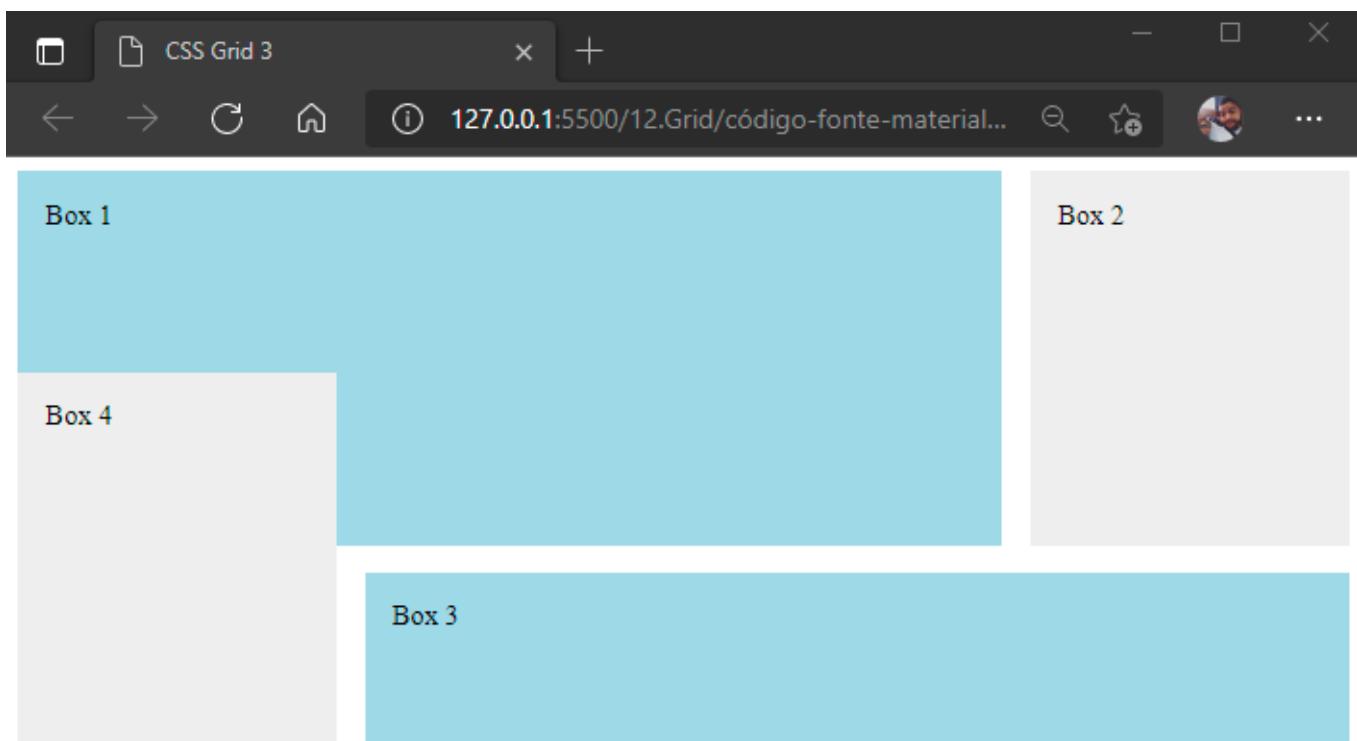


Note que o Box 3 trocou de lugar com o Box 4, pois essa é uma forma de trocar a ordem dos elementos do grid. Por fim, vamos configurar a classe **.box4**, que está associada ao quarto elemento **<div>** do elemento **<section>**.

```
.box4 {  
    grid-column: 1;  
    grid-row: 2 / 4;  
}
```

Nesse trecho, fazemos a configuração da trilha ocupada **pelo quarto item do grid**, o **Box 4**. Nesse caso, estamos definindo que **ele inicie na reta vertical 1 (grid-column: 1;)**. O **Box 4 irá ocupar a primeira coluna do grid**.

Também, configuramos para que ele **inicie na reta horizontal 2 e termine na reta horizontal 4 (grid-row: 2 / 4;)**. O **Box 4** irá ocupar a **segunda e terceira linhas do grid**. Desse modo, o **Box 4 está ocupando duas células do grid e sobrepondo o Box 1**. Ou seja, você pode personalizar o grid como desejar.





Grid e Flexbox

Os objetivos desta aula são:

- Compreender a criação de diferentes layouts com o uso do Grid CSS e flexbox;
- Conhecer as diversas propriedades do Grid CSS e flexbox;
- Aplicar os recursos do Grid CSS e flexbox nas folhas de estilo.

Bons estudos!

CSS Grid Layout e Flexbox

Você pode e deve usar o **CSS Grid Layout** e o **Flexbox juntos** no desenvolvimento de páginas web. As duas ferramentas irão contribuir para que você desenvolva layouts mais profissionais. Por isso, nessa aula você vai aprender a utilizar o CSS Layout e o Flexbox para criar o layout de uma página web.

Vamos praticar

Vamos criar um arquivo **HTML** e iniciar o desenvolvimento do site de uma **Galeria de arte fictícia**. Antes de começar, faça o download das imagens utilizadas no projeto e salve-as no dentro de um diretório chamado **img** na pasta que você irá criar o site da aula desse capítulo.

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/css2?family=Oswald&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="estilo.css">
    <title>Galeria de Arte</title>
</head>
<body>
</body>
</html>
```

Note, que usamos o elemento **<link>** duas vezes: uma vez para **vincular a fonte Oswald**, que está disponível no **Google Fonts (site útil para desenvolvedor web)** e a outra vez para vincular o nosso arquivo de estilo.

Agora, insira restante do código no arquivo **HTML**. Note que inserimos o elemento **<header>**, como as informações rápidas da galeria, e o elemento **<main>**, que contém cinco elementos **<section>** com as pinturas da galeria e algumas informações sobre as pinturas.



Dica!

Boa parte do texto aqui foi gerado automaticamente com a função lorem no Visual Studio Code. Essa função gera palavras aleatórias sem sentido. Você pode testar gerar 10 palavras aleatórias com o comando Lorem10 no editor do VS Code e pressione Enter ou TAB e veja o resultado.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/css2?family=Oswald&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="01.estilo.css">
    <title>Galeria de Arte</title>
</head>

<body>

    <header class="main-header">
        <h1>Galeria de <span>Pinturas em Acrílico</span></h1>
        <p>
            Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias,
            quam!
        </p>
    </header>

    <main class="container">
        <section class="card">
            
            <div>
                <h3>Pintura em Acrílico Um</h3>
                <p>
                    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error
                    enim laudantium, animi veniam libero eveniet culpa unde preferendis
                    illo fugit corporis, voluptatibus totam dolorum, maiores magnam
                    officia. Ab, delectus.
                </p>
                <a href="#" class="btn">Compre Agora</a>
            </div>
        </section>

        <section class="card" data-aos="fade-left">
            
            <div>
                <h3>Pintura em Acrílico Dois</h3>
                <p>
                    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error
                    enim laudantium, animi veniam libero eveniet culpa unde preferendis
                    illo fugit corporis, voluptatibus totam dolorum, maiores magnam
                    officia. Ab, delectus.
                </p>
                <a href="#" class="btn">Compre Agora</a>
            </div>
        </section>
    </main>
</body>
```

```

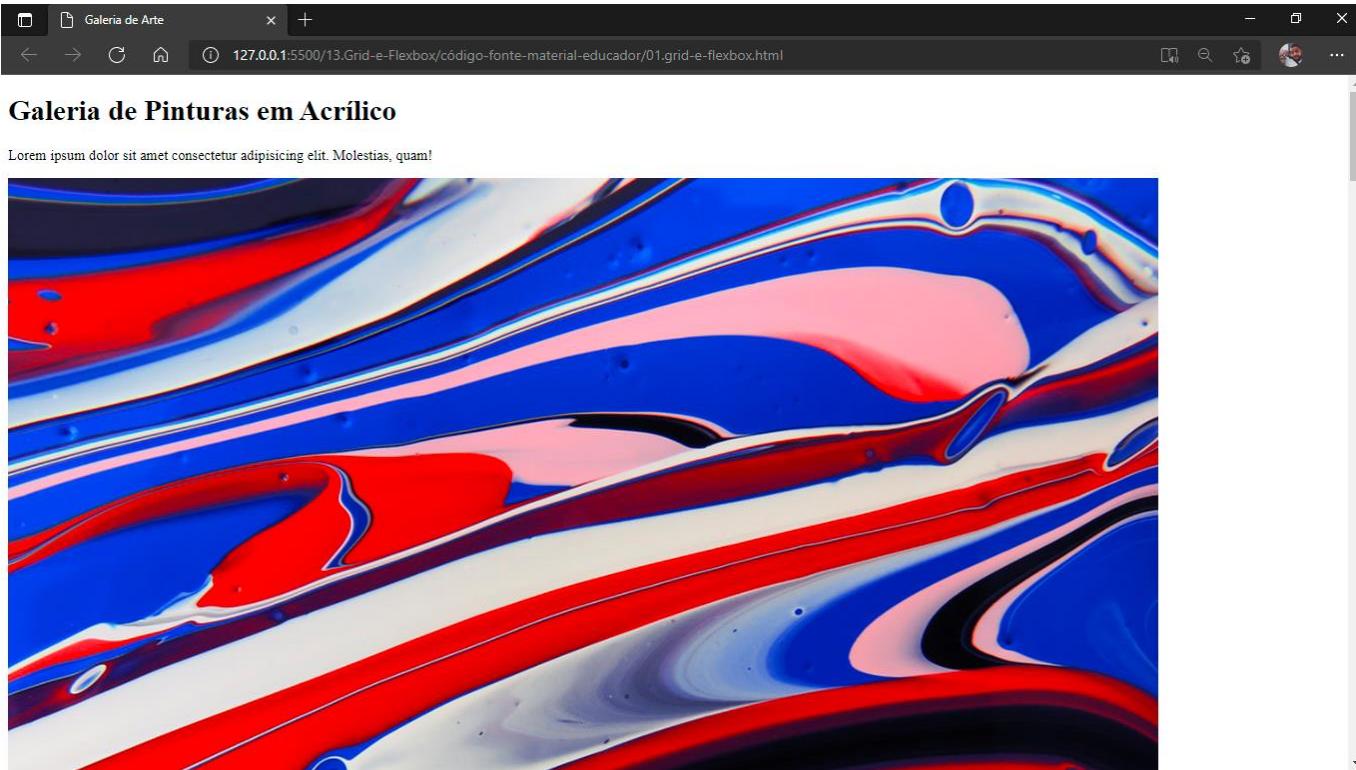
</section>

<section class="card" data-aos="fade-right">
    
    <div>
        <h3>Pintura em Acrílico Três</h3>
        <p>
            Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error
            enim laudantium, animi veniam libero eveniet culpa unde preferendis
            illo fugit corporis, voluptatibus totam dolorum, maiores magnam
            officia. Ab, delectus.
        </p>
        <a href="#" class="btn">Compre Agora</a>
    </div>
</section>

<section class="card" data-aos="fade-left">
    
    <div>
        <h3>Pintura em Acrílico Quatro</h3>
        <p>
            Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error
            enim laudantium, animi veniam libero eveniet culpa unde preferendis
            illo fugit corporis, voluptatibus totam dolorum, maiores magnam
            officia. Ab, delectus.
        </p>
        <a href="#" class="btn">Compre Agora</a>
    </div>
</section>

<section class="card" data-aos="fade-right">
    
    <div>
        <h3>Pintura em Acrílico Cinco</h3>
        <p>
            Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error
            enim laudantium, animi veniam libero eveniet culpa unde preferendis
            illo fugit corporis, voluptatibus totam dolorum, maiores magnam
            officia. Ab, delectus.
        </p>
        <a href="#" class="btn">Compre Agora</a>
    </div>
</section>
</main>
</body>
</html>

```



Observe que a página está totalmente sem formatação e as figuras estão gigantes. Precisamos configurar alguns estilos. Para isso, abra um novo arquivo **CSS**.

Vamos configurar a **margin**, o **padding** e **box-sizing** de todos os elementos. Também vamos configurar o tamanho e a família da fonte e a cor de plano de fundo do site e o elemento de criar vínculos **<a>**. Insira o seguinte código no arquivo **CSS**.

```
@charset "utf-8";
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-size: 1.3rem;
    font-family: 'Oswald', sans-serif;
    background: #f9f9f9;
}

a {
    color: #333;
    text-decoration: none;
}
```

Se você quiser pode ir visualizar a página, enquanto vamos atualizando o arquivo de **CSS**. Por exemplo, uma das mudanças realizadas foi a fonte ‘**Oswald**’, que agora está aplicada em todo o texto da página. A partir de agora, vamos colocar imagens do site, quando houver uma mudança for significativa.

Agora, vamos configurar **.main-header** que está associada ao elemento **<header>**. Insira o seguinte código no arquivo estilo.css:

```
.main-header {
    height: 55vh;
    padding: 2rem;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    text-align: center;
}
```

Nesse trecho, definimos as propriedades da classe **.main-header** associada ao elemento **<header>**, tais como: a altura (**height: 55vh**), o espaçamento (**padding: 2rem**), usar o Flexbox para exibir os itens (**display: flex**), a direção do flexbox em colunas (**flex-direction: column**), o alinhamento vertical dos itens dentro do flexbox (**align-items: center**), o alinhamento horizontal dos itens dentro do flexbox (**justify-items: center**) e o centralizar o texto (**text-align: center**).

Em seguida, vamos configurar os elementos **<h1>**, **** e **<p>**, que são filhos (estão dentro) do elemento **<header>** na página.

```
.main-header h1 {
    font-size: 4rem;
    margin-bottom: 2rem;
    line-height: 1.2rem;
}

.main-header h1 span {
    color: #b50d10;
}

.main-header p {
    font-size: 2rem;
}
```

Nesse trecho, definimos propriedades como: **tamanho e cor da fonte**, **margem e altura da linha**.



Vamos, então, configurar a classe `.container`, que está associada ao elemento `<main>`. Insira o seguinte código no arquivo **CSS**:

```
.container {  
    max-width: 1100px;  
    margin: auto;  
    overflow: auto;  
    padding: 0 2rem;  
}
```

Nesse trecho, configuramos a classe `.container` associada ao elemento `<main>` definindo as seguintes propriedades: **largura máxima (max-width: 1100px)**, **margin (margin: auto)**, **overflow (overflow: auto)** e o **espacamento superior, inferior, esquerda e direita (padding: 0 2rem)**.

Continuando as configurações no arquivo **CSS**, vamos definir as propriedades das imagens. Insira o seguinte código no arquivo `estilo.css`:

```
img {  
    width: 100%;  
}
```

Em seguida, vamos configurar a classe **.card** que está associada a todas as seções criadas com o elemento <**section**>. Insira o seguinte código no arquivo **CSS**:

```
.card {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    grid-gap: 2rem;
    background: #f1f1f1;
    margin-bottom: 2rem;
}
```

Nesse trecho, configuramos a classe **.card** associada ao elemento <**section**> definindo as seguintes propriedades: usar o CSS Grid Layout para exibir os itens (**display: grid**), a divisão do espaço livre em duas colunas iguais (**grid-template-columns: repeat(2, 1fr)**), o espaçamento entre os itens (**grid-gap: 2rem**), a cor de plano de fundo (**background: #f1f1f1**) e margem inferior (**margin-bottom: 2rem**).



Continuando, vamos definir algumas propriedades dos elementos <**h3**>, <**img**> e <**div**>, que estão dentro da classe **.card**. Insira o seguinte código no arquivo **CSS**:

```
.card h3 {
    margin-bottom: 2rem;
}

.card img {
    height: 400px;
}

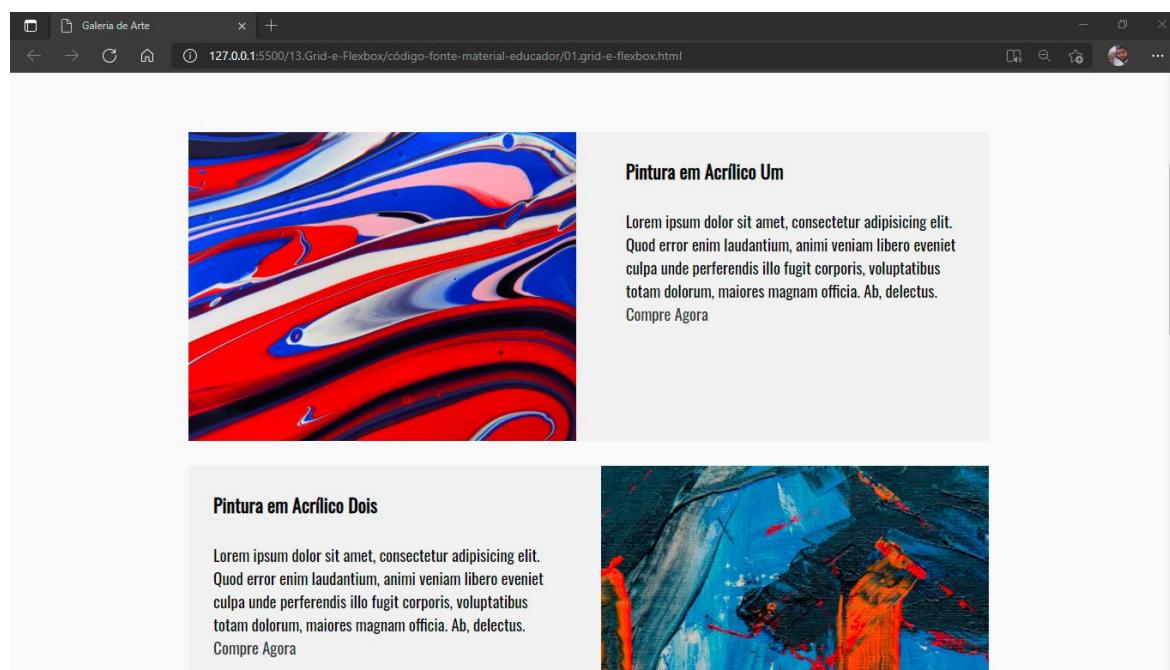
.card>div {
    padding: 2rem;
}
```

O símbolo **>** (maior que) é chamado **child selector**. Ele indica que o estilo vai se aplicado em cada elemento que é um filho diretamente de um elemento ou classe em particular. No nosso caso, da classe `.card`



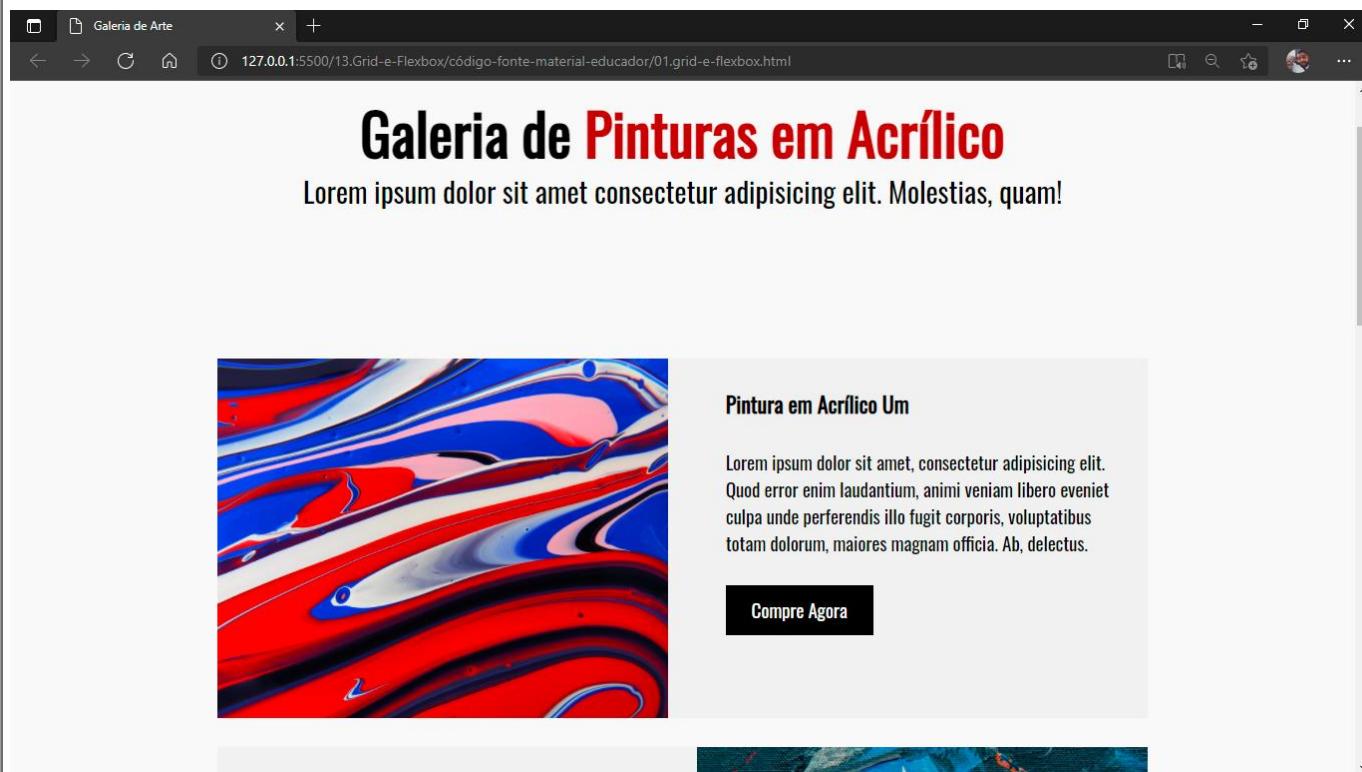
Depois, vamos configurar para que as **imagens** dos **itens ímpares do grid alterem a sua ordem de exibição**. Insira o seguinte código no arquivo **CSS**:

```
.card:nth-child(even) img {
    order: 2;
}
```



E por fim, vamos definir algumas propriedades da classe **.btn**, que está associada aos elementos **<a>** da página. Insira o seguinte código no arquivo **CSS**:

```
.btn {  
    display: inline-block;  
    background: #000;  
    color: #fff;  
    padding: 0.8rem 1.8rem;  
    margin-top: 2rem;  
    cursor: pointer;  
}  
  
.btn:hover {  
    opacity: 0.8;  
}
```



Galeria de Pinturas em Acrílico

Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias, quam!



Pintura em Acrílico Um

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod error enim laudantium, animi veniam libero eveniet culpa unde perferendis illo fugit corporis, voluptatibus totam dolorum, maiores magnam officia. Ab, delectus.

Compre Agora



Bootstrap Parte 01

Os objetivos desta aula são:

- Conhecer o framework Bootstrap;
- Aplicar o framework Bootstrap na criação de páginas web.

Bons estudos!

Bootstrap

Bootstrap é um framework web livre e de código aberto (open-source) para o desenvolvimento em HTML, CSS e JavaScript, que você pode usar para o projetar os componentes da interface e do front-end de sites e aplicações web. Com o Bootstrap é possível criar protótipos rapidamente ou aplicações completas de sites amigáveis e responsivos.

O site do Bootstrap é <https://getbootstrap.com/> e nele você encontra toda a documentação e exemplos dos estilos que podem ser usados no desenvolvimento da sua página web. No momento da produção desse material a versão atual é a 5.0, e é nela que iremos basear os nossos exemplos.

Existe um site traduzido com a versão 4.1 (disponível em <https://getbootstrap.com.br/>), que é uma versão bem mais antiga e é possível que alguns exemplos que vamos mostrar aqui estejam bem diferentes nessa versão traduzida.

Vamos praticar

Usar o bootstrap é muito simples, você precisa vinculá-lo ao seu projeto e saber o nome das classes para associá-las aos elementos HTML criados na sua página. Por isso, vamos sempre que possível colocar o link da documentação do bootstrap e, então, você poderá consultar e verificar outras variações possíveis para o estilo aplicado.

Também, vamos explicar os pontos mais importantes, coisas como cor de plano de fundo, tamanho da fonte, deixaremos para você consultar na documentação, pois caso contrário a prática ficará muito longa e cansativa.

Vamos criar um arquivo HTML e iniciar o desenvolvimento do nosso site.

Abra o VS Code, acesse o menu Arquivo e escolha Novo Arquivo ou use as teclas de atalho Ctrl+N para criar o arquivo HTML. Insira o código mostrado abaixo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-wEIJIV1mKuiNpC+IObjI7aAzPcEZeedi5yW5f2y0q55WLwNGmvvx4Um1vskeMj0" crossorigin="anonymous">
    <title>Exemplo Bootstrap</title>
</head>
<body>
</body>
</html>
```

Vamos optar por vincular os arquivos de estilos do bootstrap através do link disponibilizado na documentação oficial. Você pode copiar essa instrução:

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

No final do elemento body, vamos inserir o código em **JavaScript** para funcionar o **Bootstrap**.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet"
          integrity="sha384-EEm1mKuiNpC+IOBjI7aAzPcEZeedi5yw5f2y0q55WwLwNGmvvx4Um1vskeMj0" crossorigin="anonymous">
        <title>Exemplo Bootstrap</title>
</head>
<body>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
           integrity="sha384-p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
           crossorigin="anonymous"></script>
</body>
</html>
```

Para organização, deixe esse código no final do elemento body, antes da tag de fechamento , mas não coloque fora do body. Você pode copiar essa instrução:

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Agora, podemos começar a desenvolver a página web. Vamos começar pelo **menu de navegação**. Primeiro, vamos inserir o elemento <nav> com as classes **.navbar** (classe do bootstrap para aplicar estilos em barras de navegação), **.navbar-expand-lg** (tamanho da barra de navegação), **.navbar-dark** (estilo da barra de navegação escura) e **.bg-dark** (cor do background escura).

```
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        </nav>
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
               integrity="sha384-p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
               crossorigin="anonymous"></script>
</body>
```

O exemplo completo está disponível na documentação, fizemos apenas algumas modificações:
<https://getbootstrap.com/docs/5.0/components/navbar/>

Em seguida, vamos colocar um elemento `<div>` e associar à classe `.container-fluid` (<https://getbootstrap.com/docs/5.0/layout/containers/#how-they-work>). Vamos colocar a parte para a logomarca ou nome da empresa, no nosso exemplo vamos colocar apenas um elemento `<a>` com o texto **Bootstrap Navbar**.

```

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

    <div class="container-fluid">
      <a class="navbar-brand" href="#">Bootstrap Navbar</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
             data-bs-target="#navbarSupportedContent" aria-
             controls="navbarSupportedContent" aria-expanded="false"
             aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
    </div>
  </nav>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.m
in.js"
         integrity="sha384-
p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
         crossorigin="anonymous"></script>
</body>

```

Continuando, vamos inserir os elementos do menu dentro de elemento `<div>` e em uma lista não ordenada ``.

```

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

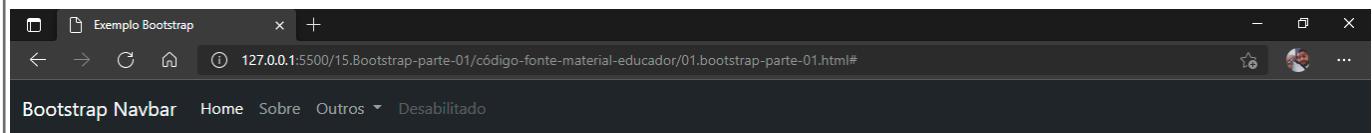
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Bootstrap Navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
             data-bs-target="#navbarSupportedContent" aria-
             controls="navbarSupportedContent" aria-expanded="false"
             aria-label="toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>

```

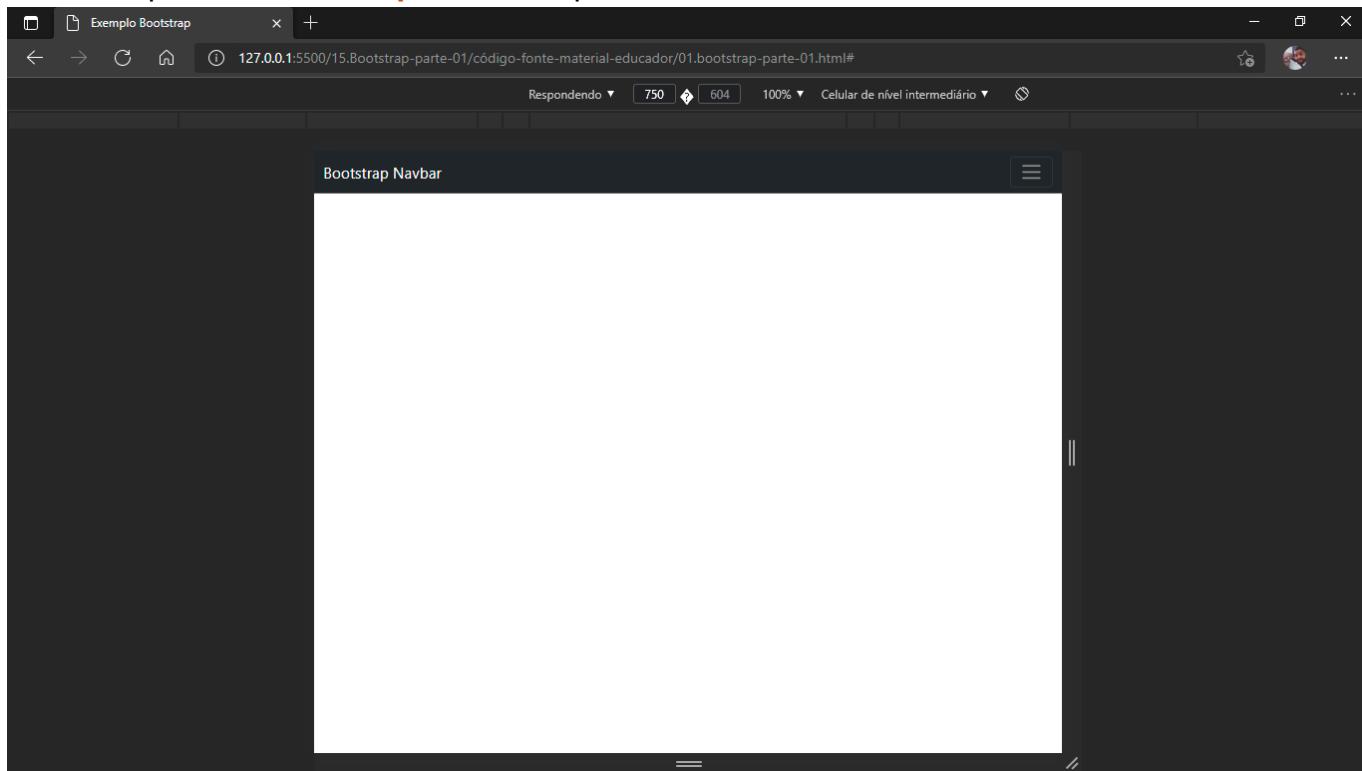
```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">Sobre</a>
        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdown" role="button"
               data-bs-toggle="dropdown" aria-expanded="false">
                Outros
            </a>
            <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
                <li><a class="dropdown-item" href="#">Projeto</a></li>
                <li><a class="dropdown-item" href="#">Contato</a></li>
                <li>
                    <hr class="dropdown-divider">
                </li>
                <li><a class="dropdown-
item" href="#">Qualquer coisa</a></li>
            </ul>
        </li>
        <li class="nav-item">
            <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Desabilitado</a>
        </li>
    </ul>
</div>
</div>
</nav>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.m
in.js"
       integrity="sha384-
p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
       crossorigin="anonymous"></script>
</body>
```

Note que criamos um elemento `<div>` associado às classes **collapse** e **navbar-collapse** e uma lista não ordenada associada à classe **navbar-nav**, **me-auto**, **mb-2** e **mb-lg-0**. Cada item da lista não ordenada deve estar associado à classe **nav-item**.



Observe que o **menu** é **responsivo**. Experimente visualizá-lo em diferentes dimensões.



Para não ficar muito grande o código, vamos mostrar apenas a parte que deve ser atualizada e indicar onde você deve inserir essa atualização.

Vamos agora criar um cabeçalho para ser visualizado na página. Insira esse trecho de código abaixo da barra de navegação, após o fechamento do elemento **nav** (**</nav>**).

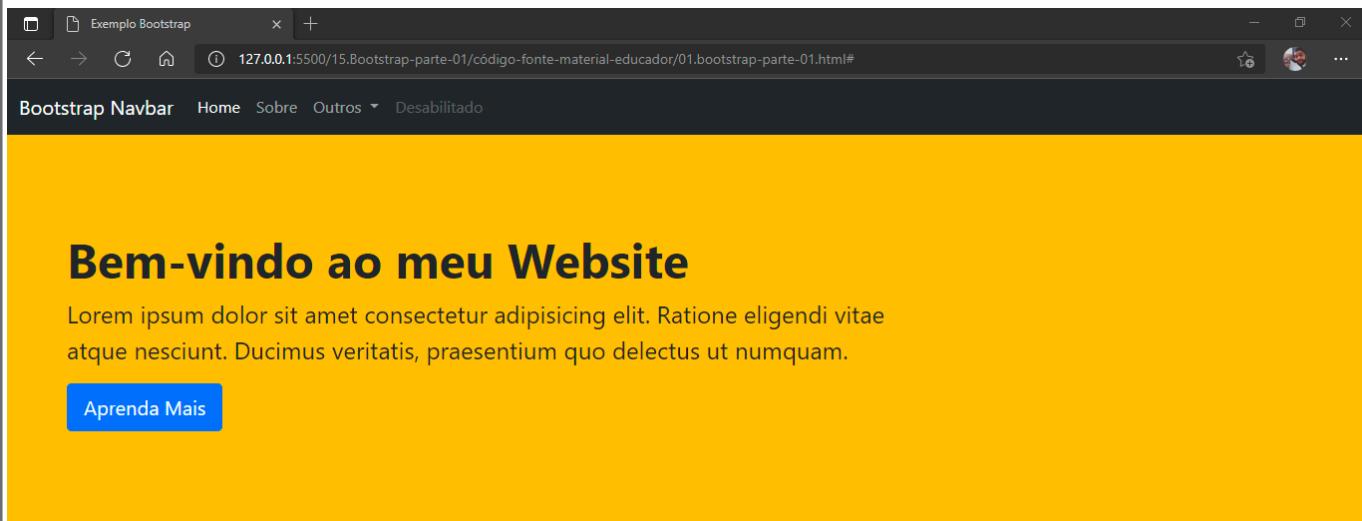
```
<header class="p-5 bg-warning">
  <div class="container-fluid py-5">
    <h1 class="display-5 fw-bold">Bem-vindo ao meu Website</h1>
    <p class="col-md-8 fs-
4">Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione
      eligendi vitae atque nesciunt. Ducimus veritatis, praesentium quo delectus u
t numquam.</p>
    <a class="btn btn-primary btn-lg" href="#" role="button">Aprenda Mais</a>
  </div>
</header>
```

Note que o elemento **<header>** possui a classe **p-5** faz o posicionamento **horizontal (eixo-x)** e **vertical (eixo-y)** do conteúdo do **header** (**p-5 igual a 5rem**). A classe **bg-warning** define a cor do plano de fundo (<https://getbootstrap.com/docs/5.0/customize/color/#theme-colors>) e a classe **row**, que define a direção do itens do grid layout.

Note, também, que o elemento **<div>** possui a classe **container-fluid**, que define a **largura do elemento** (<https://getbootstrap.com/docs/5.0/layout/containers/#how-they-work>).

Observe o elemento <h1> com a classe **display-5**, que define o tamanho da fonte do elemento e a classe **fw-bold**, que faz fica em negrito (**font-weight**).

Por fim, note que os elementos <p> tem as classes: **col-md-X**, que define quantas colunas do elemento vai ocupar e **fs-X**, que define o tamanho da fonte (**font-size**). X é um valor numérico no exemplo usado: **col-md-8** e **fs-4**. (<https://getbootstrap.com/docs/5.0/utilities/text/>)



Vamos agora trabalhar com a **tipografia do site** (documentação completa sobre tipografia no bootstrap 5.0 está disponível em: <https://getbootstrap.com/docs/5.0/content/typography/>). Vamos criar uma seção e ver como o bootstrap aplica os estilos em diversos elementos.

```
<!-- Tipografia -->
<section class="container text-center">
  <h1 class="pb-2 mt-4 mb-2 border-bottom">Hello, world! <small>Secondary text</small></h1>
  <p class="lead">Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati, laborum? Placeat non est
    facere
    recusandae fugiat voluptatum</p>

  <p><mark>Lorem ipsum dolor sit amet</mark> consectetur adipisicing elit. Obcaecati, laborum? Placeat non est.
    Delectus consequuntur <del>laudantium repellat dolorum labore</del> tenetur, <u>voluptatem
      necessitatibus</u> rerum eveniet. Esse deserunt ducimus atque iure consequatur at cum reiciendis nihil
      soluta labore preferendis.</p>
  <hr>
</section>
```

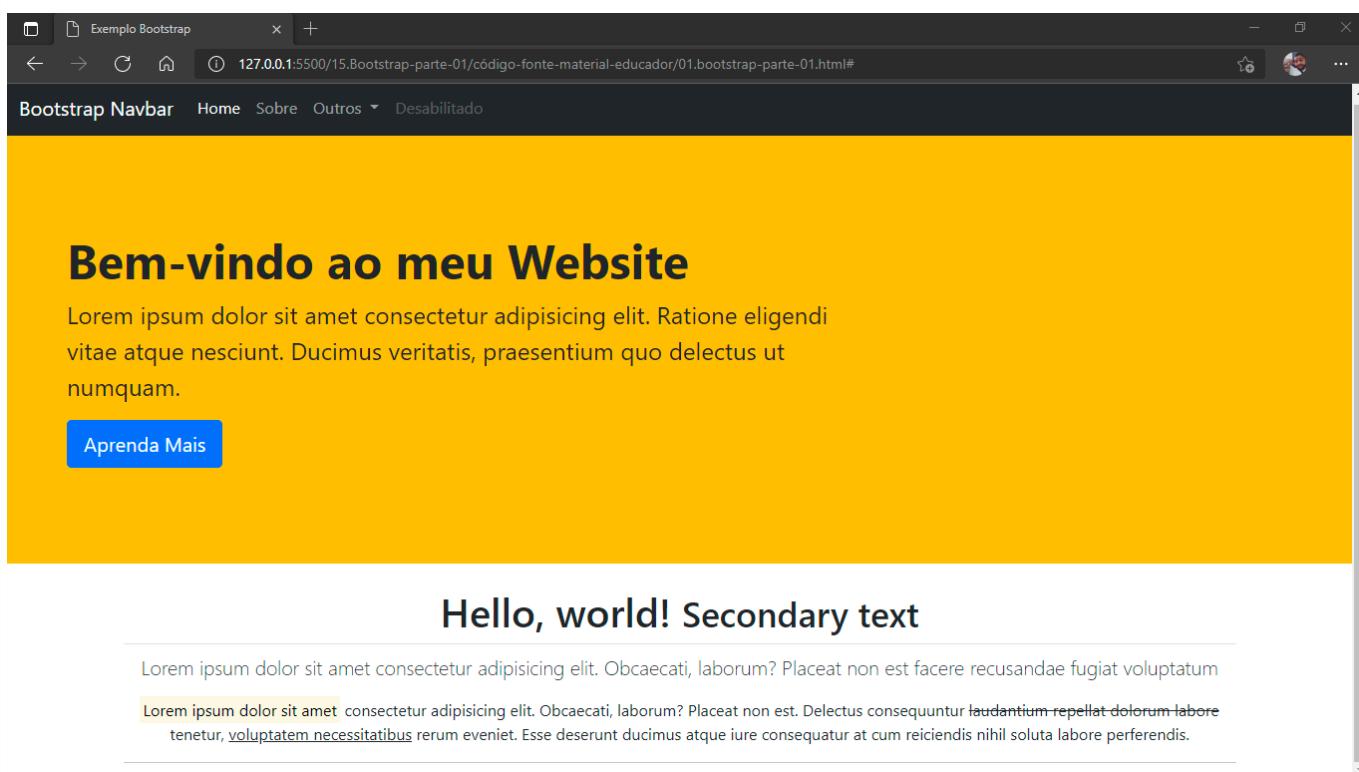
Note que o elemento <section> possui a classe **container** (<https://getbootstrap.com/docs/5.0/layout/containers/#how-they-work>) e a classe **text-center**, que centraliza o texto.

Note, também, que o elemento <**h1**> possui:

- a classe **pb-2**, que define espaçamento inferior (padding-bottom) onde p = padding e b = bottom,
- a classe **mt-4**, que define a margem superior (margin-top), onde m = margin e t = top,
- a classe **mb-2**, que define a margem inferior (margin-bottom) e
- a classe border-bottom, que inseri uma borda sólida de 1px embaixo do elemento <**h1**> (<https://getbootstrap.com/docs/5.0/utilities/borders/>).

Confira a notação do bootstrap em <https://getbootstrap.com/docs/5.0/utilities/spacing/>

Observe o segundo elemento <**p**> tem vários elementos textuais do **HTML** e o bootstrap aplica estilos personalizados em cada uma desses elementos.



Bem-vindo ao meu Website

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione eligendi
vitae atque nesciunt. Ducimus veritatis, praesentium quo delectus ut
numquam.

Aprenda Mais

Hello, world! Secondary text

Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati, laborum? Placeat non est facere recusandae fugiat voluptatum
Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati, laborum? Placeat non est. Delectus consequuntur ~~laudantium repellat dolorum labore~~
tenetur, ~~voluptatem necessitatibus~~ rerum eveniet. Esse deserunt ducimus atque iure consequatur at cum reiciendis nihil soluta labore perferendis.

Vamos ver agora como funciona o alinhamento e a transformação do texto. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

```
<!-- Alinhamento & Transformação-->
<section class="container">
  <p class="text-start text-lowercase">Texto alinhado à esquerda da área de visualização.</p>
  <p class="text-center text-uppercase">Texto alinhado centralizado na área de visualização.</p>
  <p class="text-end text-newrap">Texto sem quebra de linha. Lorem ipsum dolor sit amet consectetur, adipisicing elit.
```

```

        Aliquid perspiciatis sed quia expedita? Id cupiditate dicta magni fugiat nulla v
oluptatibus ratione
accusantium!

```

</p>

<hr>

</section>

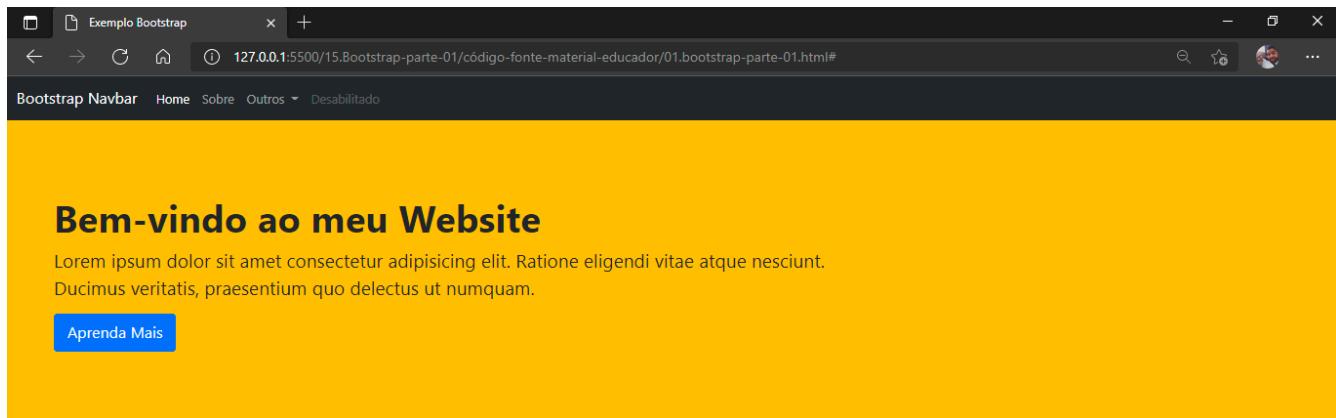
Note, também, que o primeiro elemento <p> possui a classe **text-start**, que define o alinhamento do texto à esquerda; e a classe **text-lowercase**, que transforma o texto escrito com letras minúsculas. Confira mais sobre as classes associadas a elementos textuais em <https://getbootstrap.com/docs/5.0/utilities/text/>

Já o segundo elemento <p> possui a classe **text-center**, que define centraliza o texto, e a classe **text-uppercase**, que transforma o texto escrito com letras maiúsculas.

O terceiro elemento <p> possui a classe **text-end**, que define o alinhamento do texto à direita. Por fim, o quarto parágrafo está com a classe **nowrap**, que define para não quebrar linha, assim todo o texto é exibido na mesma linha.

O bootstrap 5.0 não fornece uma classe para justificar apenas o texto. De acordo com a documentação: “embora, esteticamente, o texto justificado possa parecer mais atraente, ele torna o espaçamento entre palavras mais aleatório e, portanto, mais difícil de ler”.

Visualize resultado da sua página web.



Hello, world! Secondary text

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione eligendi vitae atque nesciunt.

Ducimus veritatis, praesentium quo delectus ut numquam.

Aprenda Mais

... Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati, laborum? Placeat non est facere recusandae fugiat voluptatum

... Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati, laborum? Placeat non est. Delectus consequuntur ludentium repellat dolorum labore tenetur, voluptatem necessitatibus rerum eveniet. Esse deserunt ducimus atque iure consequatur at cum reiciendis nihil soluta labore perferendis.

texto alinhado à esquerda da área de visualização.

TEXTO ALINHADO CENTRALIZADO NA ÁREA DE VISUALIZAÇÃO.

Texto alinhado à direta da área de visualização.

Texto sem quebra de linha. Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquid perspiciatis sed quia expedita? Id cupiditate dicta magni fugiat nulla voluptatibus ratione accusantium!

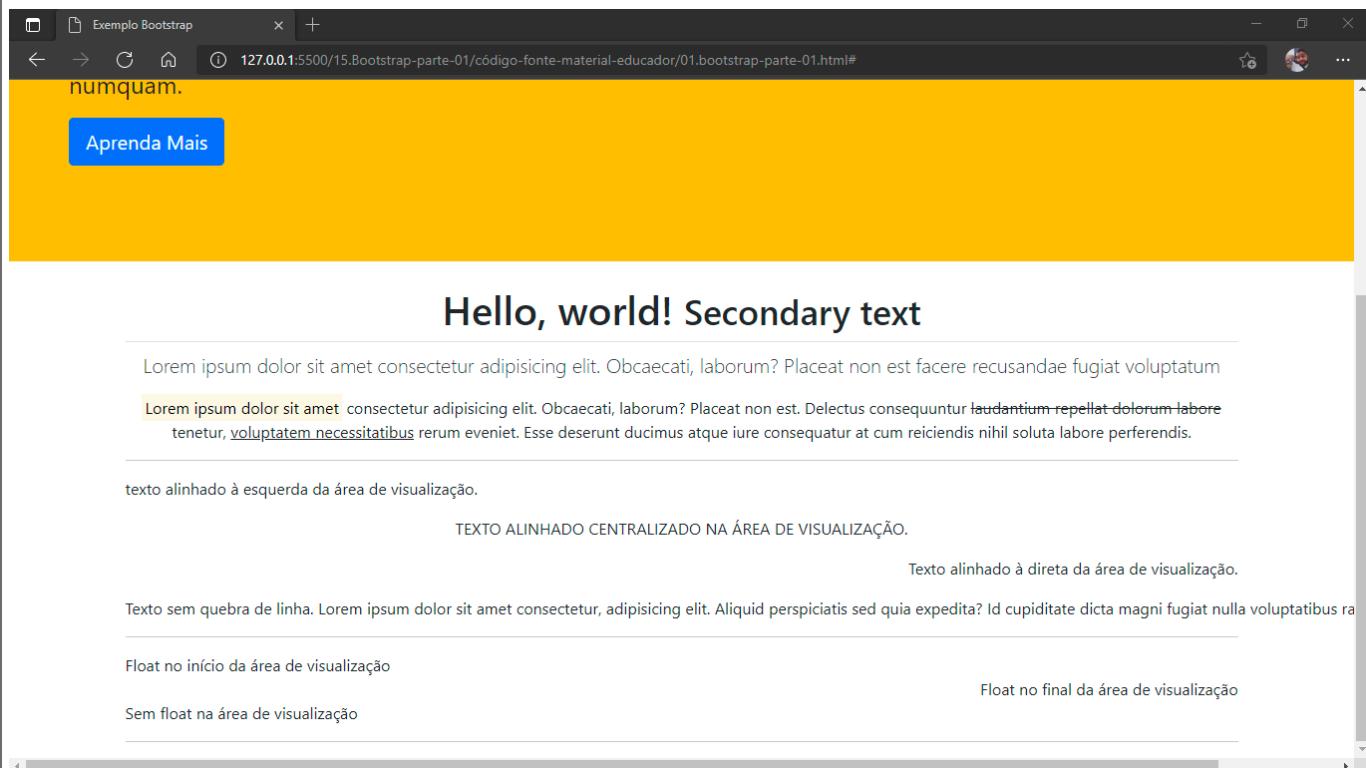
Vamos agora testar a flutuação dos elementos usando o **Float** no bootstrap. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

```
<section class="container">
  <!-- Floats -->
  <div class="float-start">Float no início da área de visualização</div><br>
  <div class="float-end">Float no final da área de visualização</div><br>
  <div class="float-none">Sem float na área de visualização</div>
  <!-- Limpa o Float -->
  <div class="clearfix"></div>
  <hr>
</section>
```

Note que o primeiro elemento `<div>` possui a classe **float-start**, que define a flutuação do elemento no início da área de visualização. Confira mais sobre as classes associadas a elementos textuais em <https://getbootstrap.com/docs/5.0/utilities/float/>

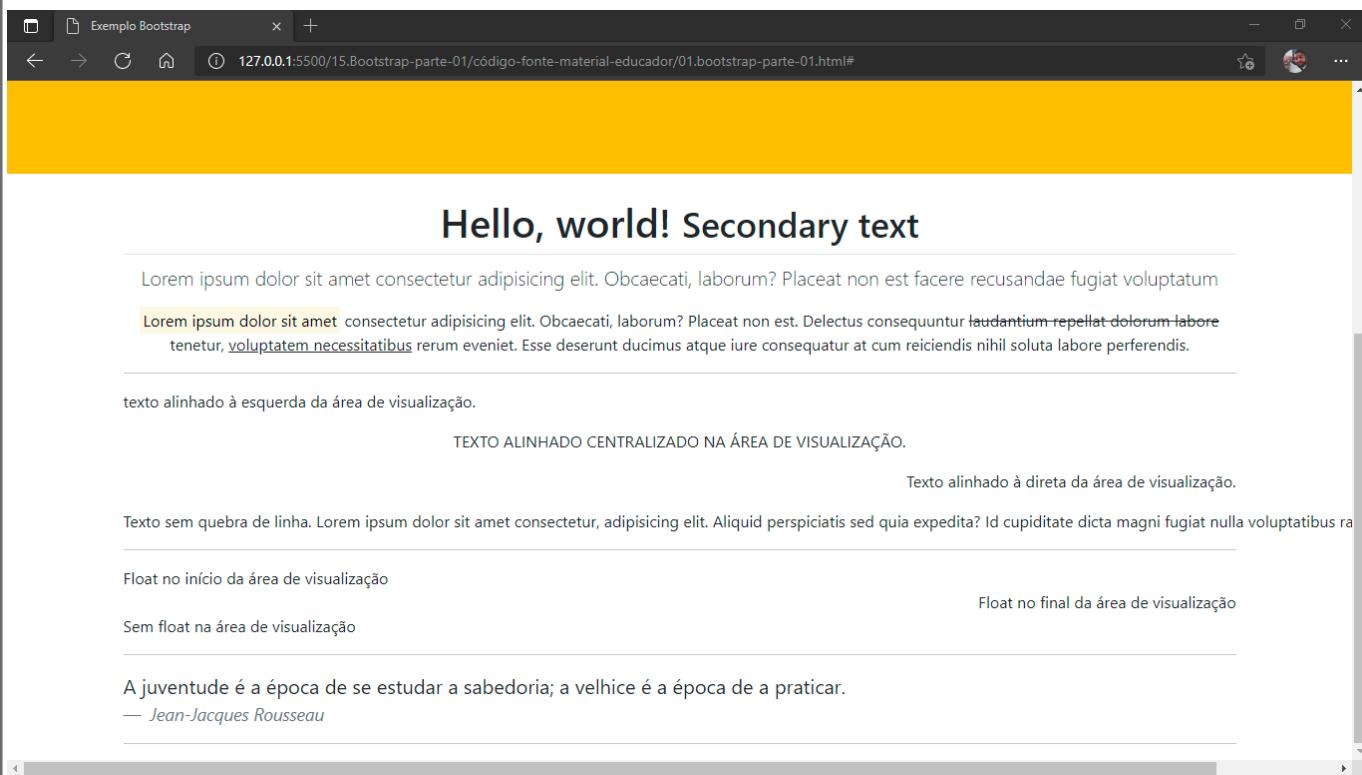
Já o segundo elemento `<div>` possui a classe **float-end**, que define a flutuação do elemento no final da área de visualização. Por fim, o terceiro elemento `<div>` possui a classe **float-none**, que define nenhuma flutuação será aplicada a área de visualização.

Importante: você deve “limpar” a flutuação dos elementos sempre que for utilizar essa propriedade, para isso inclua a instrução `<div class="clearfix"></div>` quando acabar de usar a propriedade.



Vamos agora testar o elemento **<blockquote>** e ver como o bootstrap o configura. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

```
<!-- Blockquote -->
<section class="container">
  <blockquote class="blockquote">
    <p>A juventude é a época de se estudar a sabedoria; a velhice é a época de a praticar.</p>
    <figcaption class="blockquote-footer">
      <cite title="Source Title">Jean-Jacques Rousseau</cite>
    </figcaption>
  </blockquote>
  <hr>
</section>
```



Continuando, vamos ver como as listas são exibidas. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

```
<!-- Listas -->
<section class="container">
  <ul class="list-unstyled">
    <li>Item um</li>
    <li>Item dois</li>
    <li>Item três</li>
    <li>Item quatro</li>
  </ul>
  <ul class="list-inline">
    <li class="list-inline-item">Item um</li>
    <li class="list-inline-item">Item dois</li>
```

```

<li class="list-inline-item">Item três</li>
<li class="list-inline-item">Item quatro</li>
</ul>
<ol>
  <li>Item um</li>
  <li>Item dois</li>
  <li>Item três</li>
  <li>Item quatro</li>
</ol>
<hr>
</section>

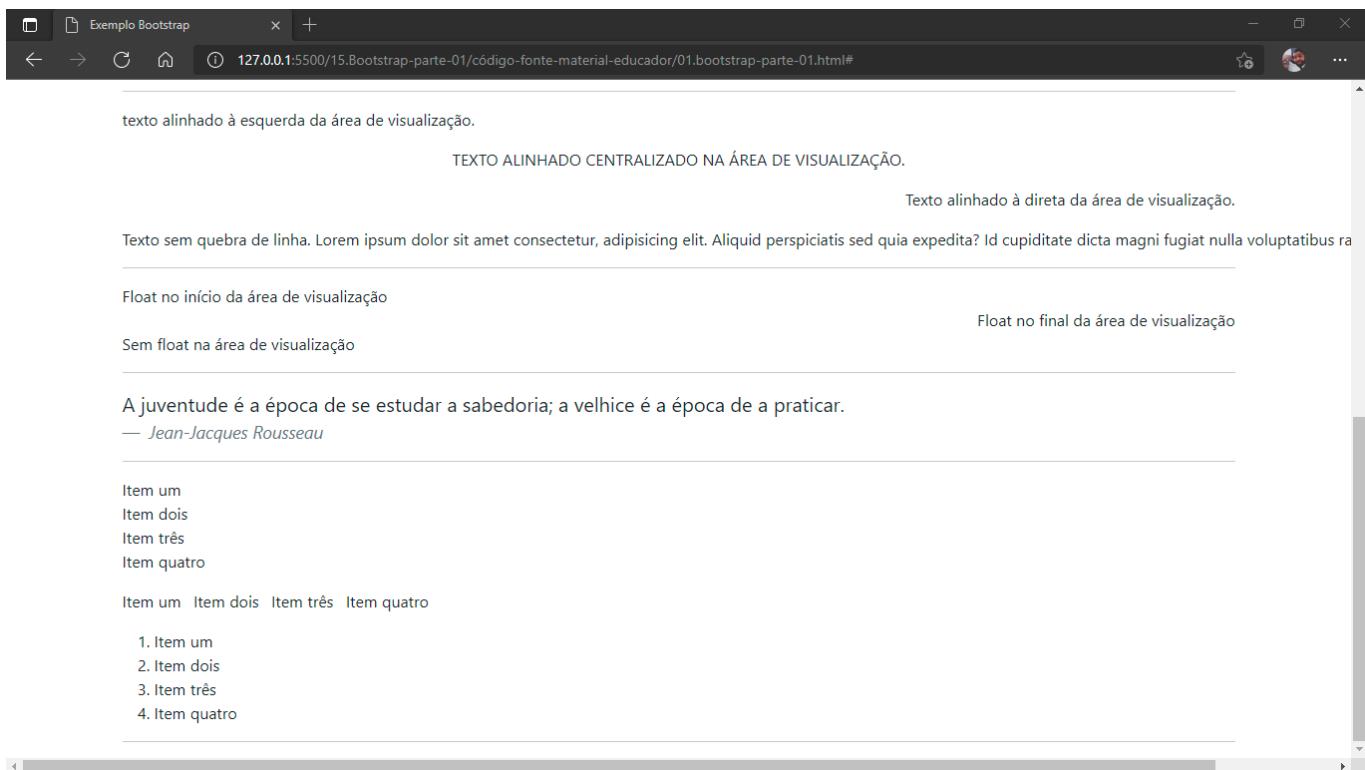
```

Note que a classe **list-unstyled** define que a lista não terá nenhum marcador (bullet point, disc, square etc.) e a classe **list-inline** define que os elementos da lista sejam exibidos em linha.

A documentação sobre os estilos de listas está disponível em:

<https://getbootstrap.com/docs/5.0/content/typography/#lists>

Visualize resultado da sua página web.



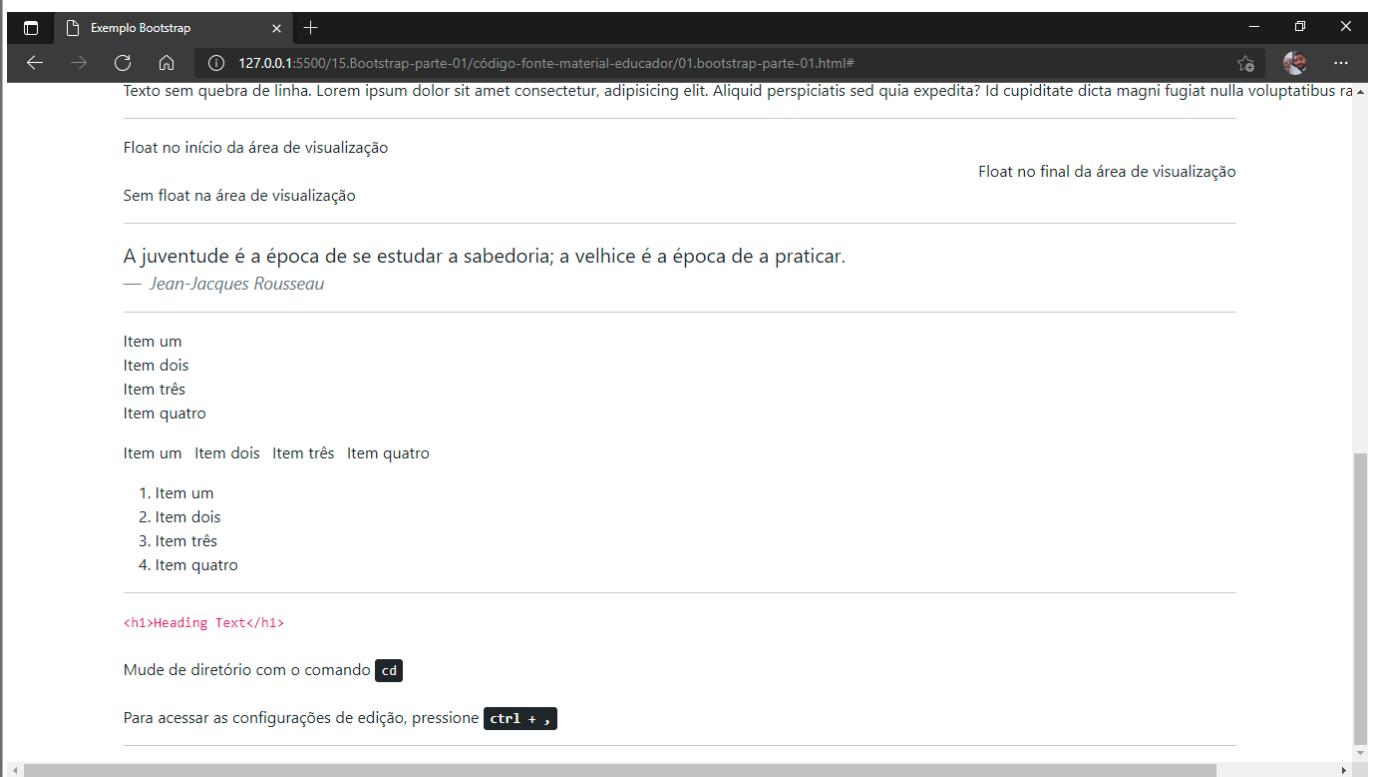
The screenshot shows a browser window titled "Exemplo Bootstrap" with the URL "127.0.0.1:5500/15.Bootstrap-parte-01/código-fonte-material-educador/01.bootstrap-parte-01.html#". The page displays various Bootstrap styling examples:

- Text aligned to the left of the visualization area.
- Text centered in the visualization area.
- Text aligned to the right of the visualization area.
- Text without line break. Placeholder: "Text sem quebra de linha. Lorem ipsum dolor sit amet consectetur, adipisciing elit. Aliquid perspiciatis sed quia expedita? Id cupiditate dicta magni fugiat nulla voluptatibus ra"
- Float at the beginning of the visualization area.
- Float at the end of the visualization area.
- Text without float in the visualization area.
- A quote by Jean-Jacques Rousseau: "A juventude é a época de se estudar a sabedoria; a velhice é a época de a praticar." — Jean-Jacques Rousseau
- Unordered list items: Item um, Item dois, Item três, Item quatro.
- Unordered list items: Item um, Item dois, Item três, Item quatro.
- Numbered list items: 1. Item um, 2. Item dois, 3. Item três, 4. Item quatro.

Existe outra forma de trabalhar com listas no Bootstrap, utilizando a classe **list-group**, mas isso **será assunto da próxima aula**.

O Bootstrap estiliza os elementos <**code**> e <**kbd**>, que são úteis para escrever códigos na sua página web. Vamos criar outra seção logo depois da anterior e inserir o seguinte código para ver como é exibido o conteúdo desses dois elementos utilizando o Bootstrap.

```
<!-- Código -->
<section class="container">
    <code>&lt;h1&gt;Heading Text&lt;/h1&gt;</code>
    <br><br>
    Mude de diretório com o comando <kbd>cd</kbd>
    <br><br>
    Para acessar as configurações de edição, pressione <kbd><kbd>ctrl</kbd> + <kbd>,</kb
d></kbd>
    <hr>
</section>
```



Agora vamos ver como aplicar o estilo de cores textuais ou de plano de fundo disponíveis no **Bootstrap**. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

```
<section class="container">
    <!-- Cores textuais -->
    <p class="text-primary">Olá Mundo com a cor do texto: text-primary</p>
    <p class="text-success">Olá Mundo com a cor do texto: text-success</p>
    <p class="text-info">Olá Mundo com a cor do texto: text-info</p>
    <p class="text-warning">Olá Mundo com a cor do texto: text-warning</p>
    <p class="text-danger">Olá Mundo com a cor do texto: text-danger</p>
    <p class="text-muted">Olá Mundo com a cor do texto: text-muted</p>
    <p class="text-black-50">Olá Mundo com a cor do texto: text-black-50</p>

    <!-- Cores de plano de fundo -->
    <p class="bg-primary">Olá Mundo com a cor de fundo: bg-primary</p>
    <p class="bg-success">Olá Mundo com a cor de fundo: bg-success</p>
```

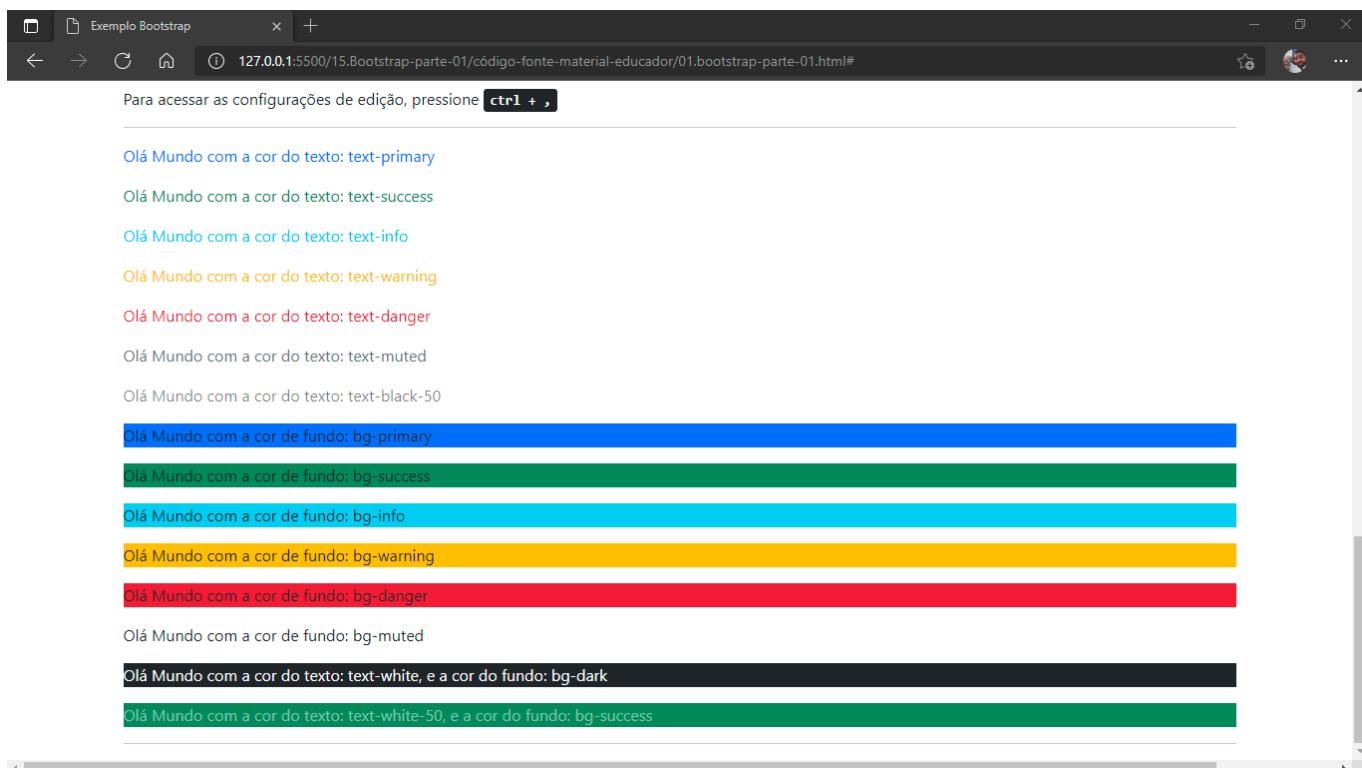
```

<p class="bg-info">Olá Mundo com a cor de fundo: bg-info</p>
<p class="bg-warning">Olá Mundo com a cor de fundo: bg-warning</p>
<p class="bg-danger">Olá Mundo com a cor de fundo: bg-danger</p>
<p class="bg-muted">Olá Mundo com a cor de fundo: bg-muted</p>

<!-- Combinação dos estilos -->
<p class="text-white bg-dark">Olá Mundo com a cor do texto: text-
white, e a cor do fundo: bg-dark</p>
<p class="text-white-50 bg-success">Olá Mundo com a cor do texto: text-white-
50, e a cor do fundo: bg-success
</p>
<hr>
</section>

```

Note que o Bootstrap já possui diversas cores predefinidas em seu CSS e podemos utilizá-las como bem quisermos ou definir nossas próprias cores personalizadas. A documentação sobre os estilos de cores está disponível em: <https://getbootstrap.com/docs/5.0/utilities/colors/>



O último assunto dessa aula será **botões**. Vamos entender com criar e estilizar botões utilizando o Bootstrap. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

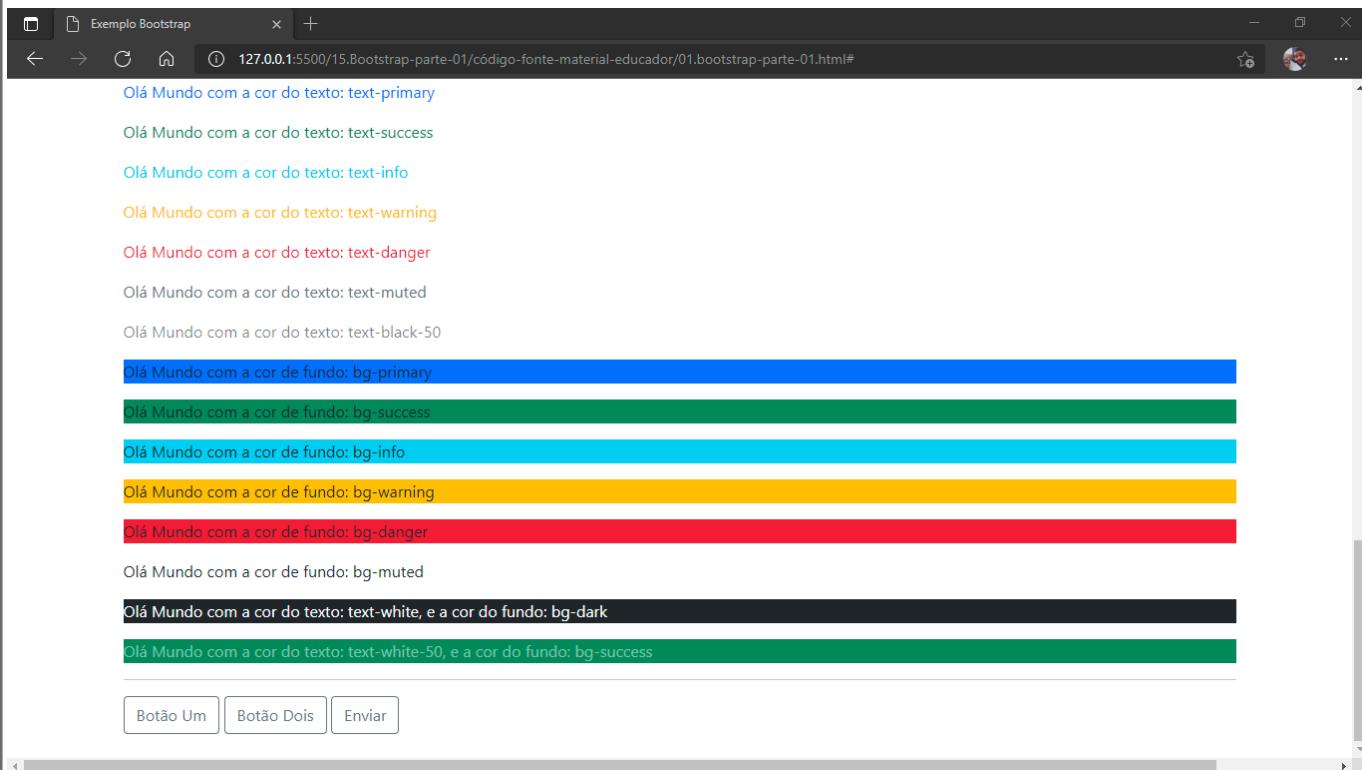
```
<section class="container">
```

```
<!-- Botões -->
<button class="btn btn-outline-secondary">Botão Um</button>
<a href="#" class="btn btn-outline-secondary" role="button">Botão Dois</a>
<input type="submit" class="btn btn-outline-secondary" value="Enviar">
<br><br>
</section>
```

Podemos criar botões utilizando os elementos `<button>`, `<a>` ou `<input>`. Nesse exemplo criamos botões idênticos utilizando as classes `.btn` e `.btn-outline-secondary` do Bootstrap. A documentação sobre os estilos de botões está disponível em:

<https://getbootstrap.com/docs/5.0/components/buttons/>

Visualize resultado da sua página web.



Em seguida, vamos ver as cores predefinidas para estilizar os botões. Vamos continuar na mesma seção anterior e inserir o seguinte código.

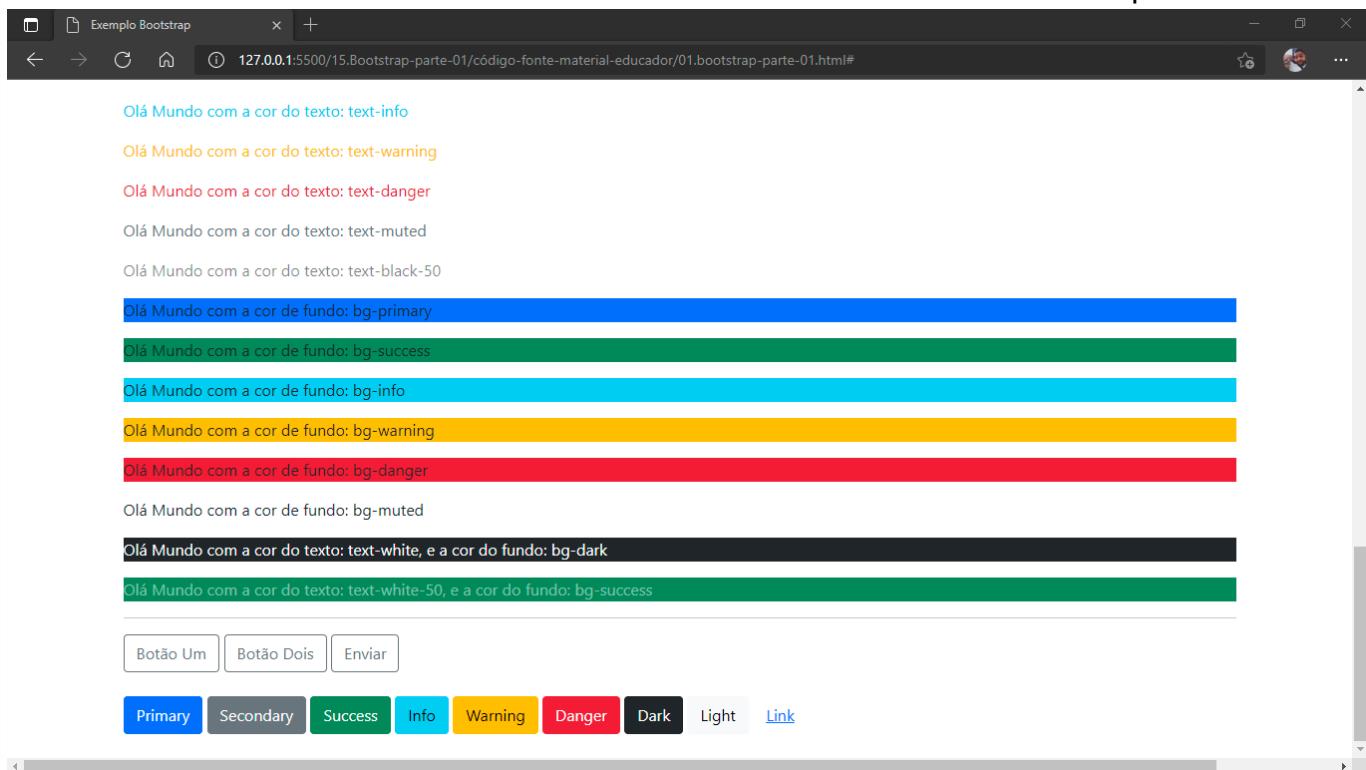
```
<section class="container">
  <!-- Botões -->
  <button class="btn btn-outline-secondary">Botão Um</button>
  <a href="#" class="btn btn-outline-secondary" role="button">Botão Dois</a>
  <input type="submit" class="btn btn-outline-secondary" value="Enviar">
  <br><br>
  <!-- Botões Contextuais -->
  <button class="btn btn-primary">Primary</button>
  <button class="btn btn-secondary">Secondary</button>
```

```

<button class="btn btn-success">Success</button>
<button class="btn btn-info">Info</button>
<button class="btn btn-warning">Warning</button>
<button class="btn btn-danger">Danger</button>
<button class="btn btn-dark">Dark</button>
<button class="btn btn-light">Light</button>
<button class="btn btn-link">Link</button>
<br><br>
</section>

```

Criamos oito diferentes botões e um link com as classes existentes no Bootstrap.



Podemos alterar os tamanhos dos botões utilizando as classes **.btn-lg**, **.btn-sm** do Bootstrap e desabilitar um botão com o atributo **disabled**. Vamos continuar na mesma seção anterior e inserir o seguinte código.

```

<section class="container">
  <!-- Botões -->
  <button class="btn btn-outline-secondary">Botão Um</button>
  <a href="#" class="btn btn-outline-secondary" role="button">Botão Dois</a>
  <input type="submit" class="btn btn-outline-secondary" value="Enviar">
  <br><br>
  <!-- Botões Contextuais -->
  <button class="btn btn-primary">Primary</button>
  <button class="btn btn-secondary">Secondary</button>
  <button class="btn btn-success">Success</button>
  <button class="btn btn-info">Info</button>

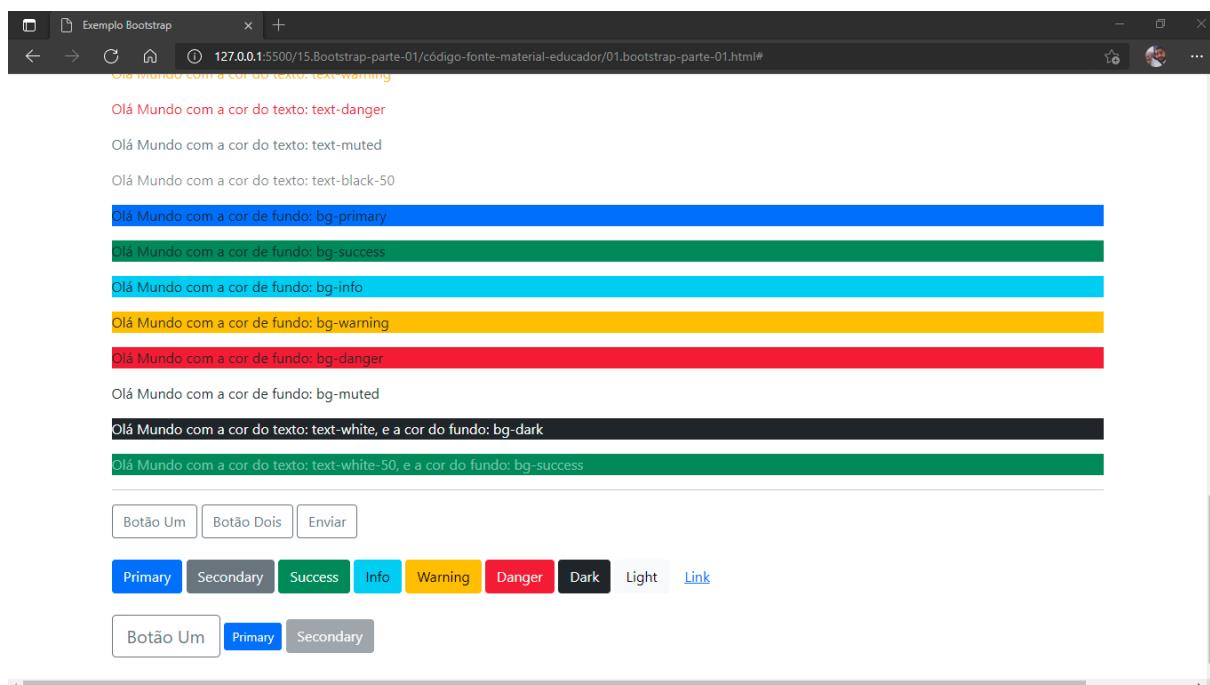
```

```

<button class="btn btn-warning">Warning</button>
<button class="btn btn-danger">Danger</button>
<button class="btn btn-dark">Dark</button>
<button class="btn btn-light">Light</button>
<button class="btn btn-link">Link</button>
<br><br>
<!-- Tamanho de Botões -->
<button class="btn btn-outline-secondary btn-lg">Botão Um</button>
<button class="btn btn-primary btn-sm">Primary</button>
<button class="btn btn-secondary" disabled="disabled">Secondary</button>
<br><br>
</section>

```

Visualize resultado da sua página web.



E a última configuração é utilizar a classe `.btn-outline-*`, que além de definir o estilo do botão também já configura as pseudoclasses `:hover` e `:active`, alterando o estilo do botão quando você passar o mouse ou clicar no botão. Vamos continuar na mesma seção anterior e inserir o seguinte código.

```

<section class="container">
<!-- Botões -->
<button class="btn btn-outline-secondary">Botão Um</button>
<a href="#" class="btn btn-outline-secondary" role="button">Botão Dois</a>
<input type="submit" class="btn btn-outline-secondary" value="Enviar">
<br><br>
<!-- Botões Contextuais -->
<button class="btn btn-primary">Primary</button>
<button class="btn btn-secondary">Secondary</button>

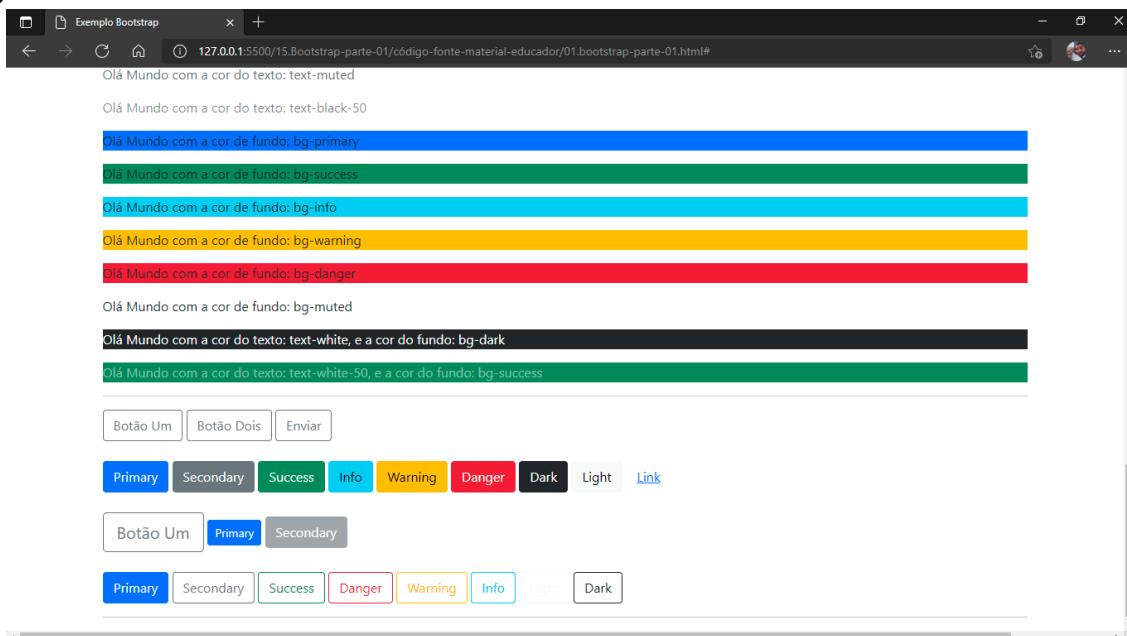
```

```

<button class="btn btn-success">Success</button>
<button class="btn btn-info">Info</button>
<button class="btn btn-warning">Warning</button>
<button class="btn btn-danger">Danger</button>
<button class="btn btn-dark">Dark</button>
<button class="btn btn-light">Light</button>
<button class="btn btn-link">Link</button>
<br><br>
<!-- Tamanho de Botões -->
<button class="btn btn-outline-secondary btn-lg">Botão Um</button>
<button class="btn btn-primary btn-sm">Primary</button>
<button class="btn btn-secondary" disabled="disabled">Secondary</button>
<br><br>
<!-- Efeito dos botões -->
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
<hr>
</section>

```

Experimente passar o mouse sobre os botões e clicar em um dos botões. Visualize resultado da sua página web.



Vamos parar por aqui nessa aula, veremos mais sobre o bootstrap na próxima aula.



Bootstrap Parte 02

Os objetivos desta aula são:

- Conhecer o framework Bootstrap;
- Aplicar o framework Bootstrap na criação de páginas web.

Bons estudos!

Vamos praticar

Vamos criar um arquivo **HTML** e iniciar o desenvolvimento do site. Antes de começar, faça o download das imagens utilizadas no projeto e salve-as no dentro de um diretório chamado **img** na pasta que você irá criar o site da aula desse capítulo.

Abra o **VS Code**, acesse o menu **Arquivo** e escolha **Novo Arquivo** ou use as teclas de atalho **Ctrl+N** para criar o arquivo **HTML**. Insira o código mostrado abaixo.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet"
          integrity="sha384-
wEmEV1mKuiNpC+IOBjI7aAzPcEZeedi5yW5f2y0q55WWLwNGmvvx4Um1vskeMj0" crossorigin="anonymous">
    <title>Exemplo Bootstrap – Parte 02</title>
</head>
<body>
</body>
</html>
```

Vamos optar por vincular os arquivos de estilos do Bootstrap através do link disponibilizado na documentação oficial, você pode copiar essa instrução:

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

No final do elemento **<body>**, vamos inserir o código em **JavaScript** para funcionar o Bootstrap.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet"
          integrity="sha384-
wEmEV1mKuiNpC+IOBjI7aAzPcEZeedi5yW5f2y0q55WWLwNGmvvx4Um1vskeMj0" crossorigin="anonymous">
    <title>Exemplo Bootstrap – Parte 02</title>
</head>
<body>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
           integrity="sha384-
p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
           crossorigin="anonymous"></script>
</body>
</html>
```

Para organização, deixe esse código no final do elemento <body>, antes da tag de fechamento </body>, mas não coloque fora do <body>. Você pode copiar essa instrução: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Agora podemos começar a desenvolver a página web. Iremos começar criando formulários na nossa página web. Vamos criar uma seção e inserir o trecho de código abaixo.

```
<body>

    <section class="container">
        <!-- Formulários -->
        <form action="#">
            <label class="form-label">Nome Completo</label>
            <input type="text" class="form-control" placeholder="Adicione o Nome">

            <label class="form-label">Profissão</label>
            <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">

            <label class="form-label">E-mail</label>
            <input type="email" class="form-control form-control-lg" placeholder="Adicione o Email">

            <label>Mensagem</label>
            <textarea class="form-control" placeholder="Adicione uma Mensagem"></textarea>
            <br>
        </form>
    </section>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-f0vrtt0+5tMqPtfy8B6Jz7H+r0Spf/wXNqD8hLwZ9r+q5STQf4PbzWZdJN" crossorigin="anonymous"></script>

</body>
```

Note que criamos três elementos <input> e um <textarea> no formulário com seus respectivos elementos <label>. No elemento <label>, utilizamos a classe **.form-label**, que define tamanho da fonte, espaçamento, margem, entre outras propriedades do rótulo do <input>.

No primeiro elemento <input>, usamos a classe **.form-control**, que define as propriedades da caixa onde o texto será inserido, inclusive sendo totalmente responsivo, sendo redimensionado para diversas telas de visualização. Esse primeiro <input> tem o tamanho default definido pelo Bootstrap.

No segundo elemento <input>, além da classe **.form-control** usamos a classe **.form-control-sm**, que diminuiu o tamanho da fonte e a altura da caixa de texto.

No terceiro elemento **<input>**, além da classe **.form-control** usamos a classe **.form-control-lg**, que aumentou o tamanho da fonte e a altura da caixa de texto.

A documentação sobre os estilos de formulários está disponível em:

<https://getbootstrap.com/docs/5.0/forms/overview/>



Continuando, vamos **inserir a opção de selecionar** uma formação. Atualize o seu código como mostra abaixo.

```

<body>

    <section class="container">
        <!-- Formulários -->
        <form action="#">
            <label class="form-label">Nome Completo</label>
            <input type="text" class="form-control" placeholder="Adicione o Nome">

            <label class="form-label">Profissão</label>
            <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">

            <label class="form-label">E-mail</label>
            <input type="email" class="form-control form-control-lg" placeholder="Adicione o Email">

            <label>Mensagem</label>
            <textarea class="form-control" placeholder="Adicione uma Mensagem"></textarea>
            <br>
            <hr>
            <label>Formação</label>
            <select name="" id="" class="form-control">
                <option value="fundamental">Ensino fundamental</option>
                <option value="medio">Ensino médio</option>
                <option value="superior">Ensino superior</option>
                <option value="superior">Outro</option>
            </select>
        </form>
    </section>

```

```

        </select>
        <br>
        <hr>

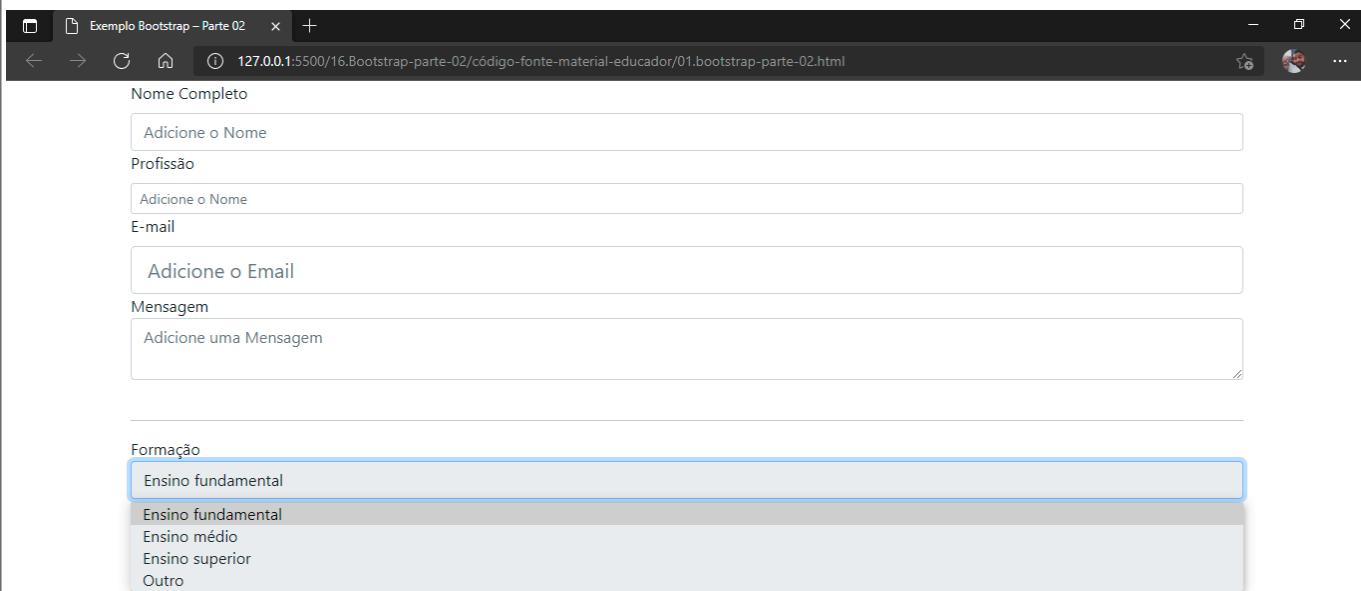
    </form>
</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-p34f1KxgkxW/XTkc9rOVtEJgkHtOQXG9wHmSsX8anwXqQ0eXtBhKUo+xtqj8h" crossorigin="anonymous"></script>

</body>

```

Visualize resultado da sua página web, clique na lista para ver as opções como mostra a figura a seguir.



Em seguida vamos uma **lista de dados** para selecionar uma cidade. Atualize o seu código como mostra abaixo.

```

<body>
<section class="container">
    <!-- Formulários -->
    <form action="#">
        <label class="form-label">Nome Completo</label>
        <input type="text" class="form-control" placeholder="Adicione o Nome">
        <label class="form-label">Profissão</label>
        <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">
        <label class="form-label">E-mail</label>
    </form>
</section>

```

```

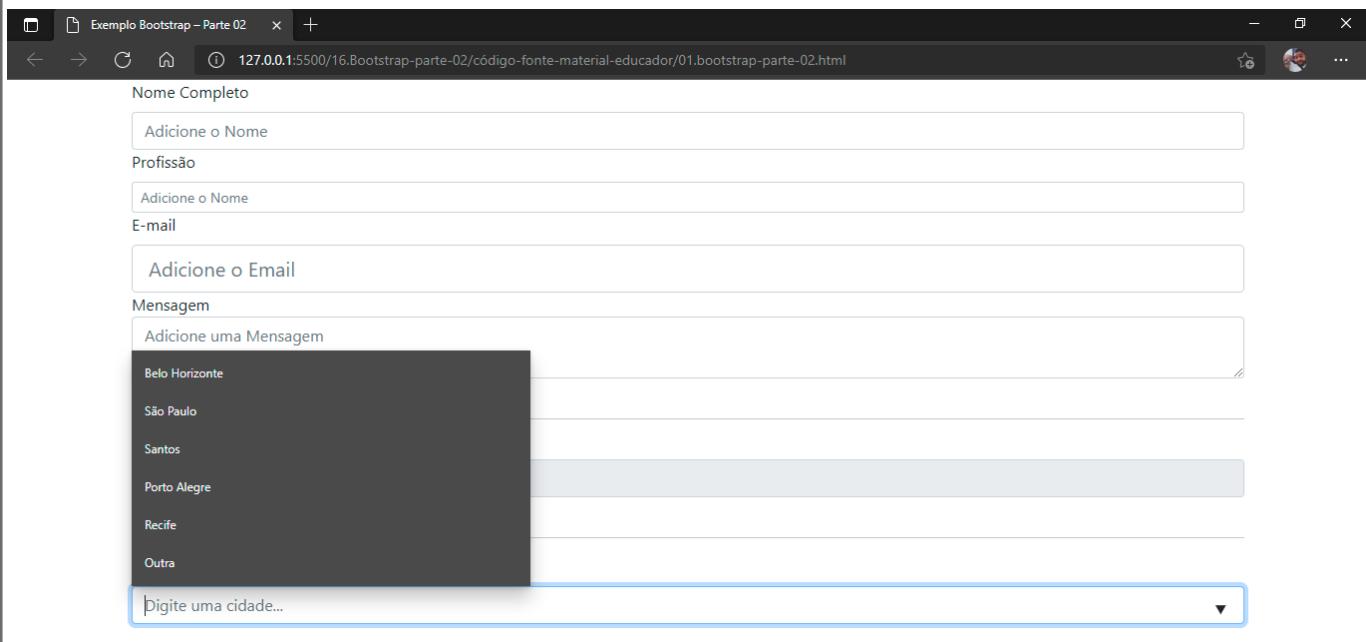
        <input type="email" class="form-control form-control-
lg" placeholder="Adicione o Email">
        <label>Mensagem</label>
        <textarea class="form-
control" placeholder="Adicione uma Mensagem"></textarea>

        <br>
        <hr>
        <label>Formação</label>
        <select name="" id="" class="form-control">
            <option value="fundamental">Ensino fundamental</option>
            <option value="medio">Ensino médio</option>
            <option value="superior">Ensino superior</option>
            <option value="superior">Outro</option>
        </select>
        <br>
        <hr>
        <label for="exampleDataList" class="form-label">Cidade</label>
        <input class="form-
control" list="datalistOptions" id="exampleDataList" placeholder="Digite uma cidade...">
        <datalist id="datalistOptions">
            <option value="Belo Horizonte"></option>
            <option value="São Paulo"></option>
            <option value="Santos"></option>
            <option value="Porto Alegre"></option>
            <option value="Recife"></option>
            <option value="Outra"></option>
        </datalist>

        <br>
        <hr>
    </form>
</section>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.m
in.js"
        integrity="sha384-
p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
        crossorigin="anonymous"></script>
</body>

```

Visualize resultado da sua página web, clique na lista para ver as opções ou comece a digitar uma cidade.



The screenshot shows a web browser window with the title "Exemplo Bootstrap – Parte 02". The URL in the address bar is "127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html". The page contains a form with fields for Nome Completo, Profissão, E-mail, and Mensagem. Below the Mensagem field is a dropdown menu titled "Cidade" with the following options: Belo Horizonte, São Paulo, Santos, Porto Alegre, Recife, and Outra. The input field for "Digite uma cidade..." is visible at the bottom.

Vamos então inserir uma opção de **envio de arquivo**. Atualize o seu código como mostra abaixo.

```
<body>

<section class="container">
    <!-- Formulários -->
    <form action="#">
        <label class="form-label">Nome Completo</label>
        <input type="text" class="form-control" placeholder="Adicione o Nome">

        <label class="form-label">Profissão</label>
        <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">

        <label class="form-label">E-mail</label>
        <input type="email" class="form-control form-control-lg" placeholder="Adicione o Email">

        <label>Mensagem</label>
        <textarea class="form-control" placeholder="Adicione uma Mensagem"></textarea>

        <br>
        <hr>

        <label>Formação</label>
```

```

<select name="" id="" class="form-control">
    <option value="fundamental">Ensino fundamental</option>
    <option value="medio">Ensino médio</option>
    <option value="superior">Ensino superior</option>
    <option value="superior">Outro</option>
</select>

<br>
<hr>

<label for="exampleDataList" class="form-label">Cidade</label>
<input class="form-
control" list="datalistOptions" id="exampleDataList" placeholder="Digite uma cidade...">
    <datalist id="datalistOptions">
        <option value="Belo Horizonte"></option>
        <option value="São Paulo"></option>
        <option value="Santos"></option>
        <option value="Porto Alegre"></option>
        <option value="Recife"></option>
        <option value="Outra"></option>
    </datalist>

<br>
<hr>

<label class="form-label">Enviar Arquivo</label>
<input type="file">
<br>
<span class="form-text">
    Apenas arquivos png e jpg.
</span>

<br>
<hr>

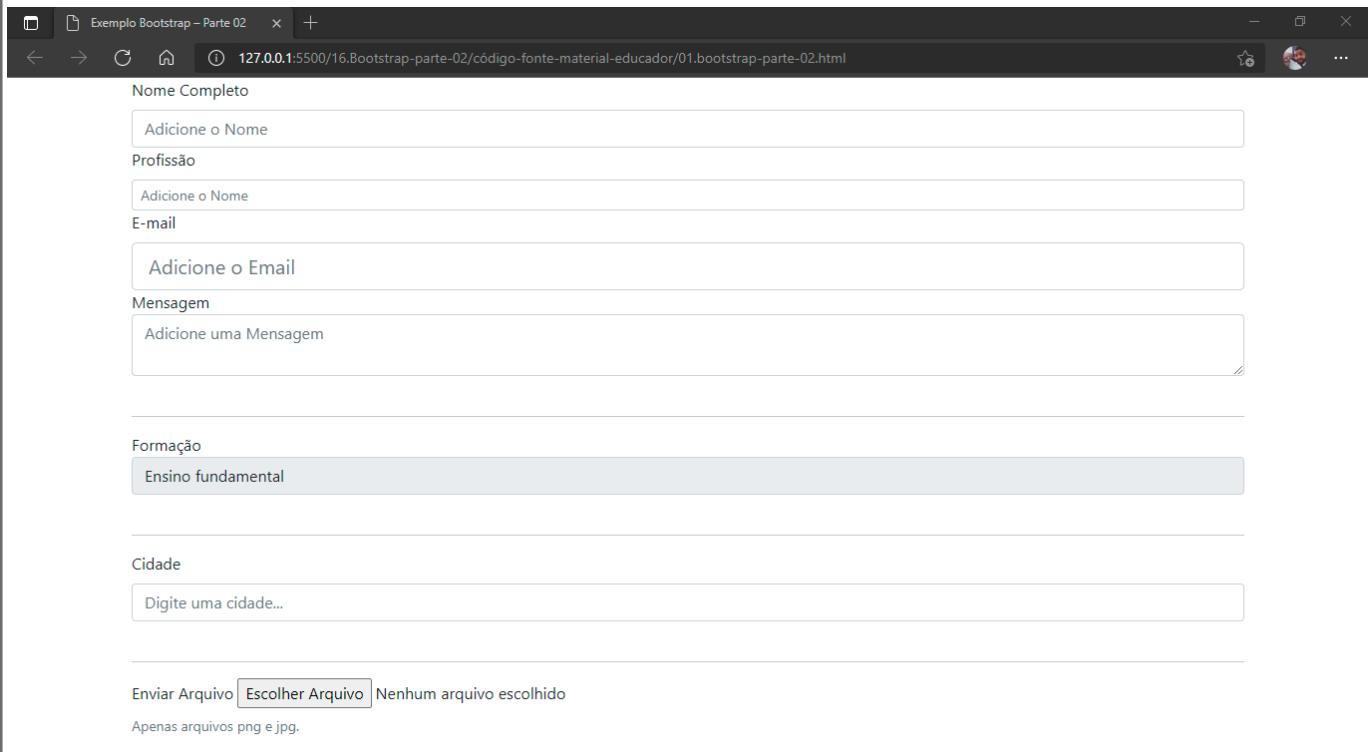
</form>
</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.m
in.js"
    integrity="sha384-
p34f1UUtsS3wqzfto5wAAmvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
    crossorigin="anonymous"></script>

</body>

```

Visualize resultado da sua página web.



The screenshot shows a web browser window titled "Exemplo Bootstrap – Parte 02". The URL is "127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html". The page displays a Bootstrap form with the following fields:

- Nome Completo:** A text input field placeholder "Adicione o Nome".
- Profissão:** A text input field placeholder "Adicione o Nome".
- E-mail:** A text input field placeholder "Adicione o Email".
- Mensagem:** A textarea placeholder "Adicione uma Mensagem".
- Formação:** A dropdown menu showing "Ensino fundamental" selected.
- Cidade:** A text input field placeholder "Digite uma cidade...".
- File Upload:** Buttons for "Enviar Arquivo" and "Escolher Arquivo". Below them, it says "Nenhum arquivo escolhido" and "Apenas arquivos png e jpg.".

Vamos inserir uma **paleta de cores**. Atualize o seu código como mostra abaixo.

```
<body>

    <section class="container">
        <!-- Formulários -->
        <form action="#">
            <label class="form-label">Nome Completo</label>
            <input type="text" class="form-control" placeholder="Adicione o Nome">

            <label class="form-label">Profissão</label>
            <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">

            <label class="form-label">E-mail</label>
            <input type="email" class="form-control form-control-lg" placeholder="Adicione o Email">

            <label>Mensagem</label>
            <textarea class="form-control" placeholder="Adicione uma Mensagem"></textarea>

            <br>
            <hr>
```

```

<label>Formação</label>
<select name="" id="" class="form-control">
    <option value="fundamental">Ensino fundamental</option>
    <option value="medio">Ensino médio</option>
    <option value="superior">Ensino superior</option>
    <option value="superior">Outro</option>
</select>

<br>
<hr>

<label for="exampleDataList" class="form-label">Cidade</label>
<input class="form-
control" list="datalistOptions" id="exampleDataList" placeholder="Digite uma cidade...">
<datalist id="datalistOptions">
    <option value="Belo Horizonte"></option>
    <option value="São Paulo"></option>
    <option value="Santos"></option>
    <option value="Porto Alegre"></option>
    <option value="Recife"></option>
    <option value="Outra"></option>
</datalist>

<br>
<hr>

<label class="form-label">Enviar Arquivo</label>
<input type="file">
<br>
<span class="form-text">
    Apenas arquivos png e jpg.
</span>

<br>
<hr>

<label for="exampleColorInput" class="form-label">Seletor de Cores</label>
<input type="color" class="form-control form-control-
color" id="exampleColorInput" value="#021BFC"
        title="Escolha a sua Cor">

<br>
<hr>

</form>
</section>

```

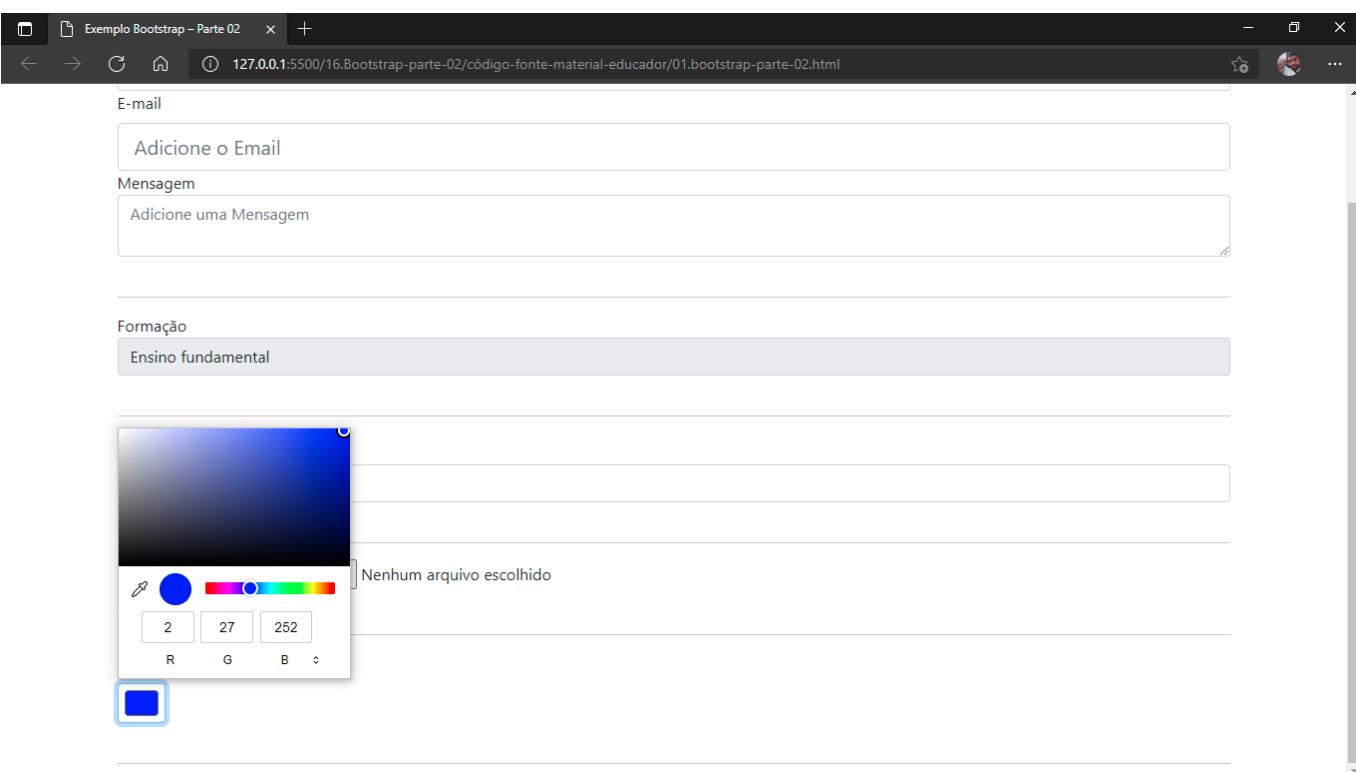
```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.js"
integrity="sha384-p34f1Kx8z8vEIwvJqE9cZoC2O7GnWd13eVgQI4m5MhD0N7tqEJLWZyvHhjGScy+8"
crossorigin="anonymous"></script>

</body>

```

Visualize resultado da sua página web.



Para finalizar o formulário, vamos inserir um **botão de enviar**.

```

<body>

<section class="container">
    <!-- Formulários -->
    <form action="#">
        <label class="form-label">Nome Completo</label>
        <input type="text" class="form-control" placeholder="Adicione o Nome">

        <label class="form-label">Profissão</label>
        <input type="text" class="form-control form-control-sm" placeholder="Adicione o Nome">

```

```

<label class="form-label">E-mail</label>
<input type="email" class="form-control form-control-
lg" placeholder="Adicione o Email">

    <label>Mensagem</label>
    <textarea class="form-
control" placeholder="Adicione uma Mensagem"></textarea>

    <br>
    <hr>

    <label>Formação</label>
    <select name="" id="" class="form-control">
        <option value="fundamental">Ensino fundamental</option>
        <option value="medio">Ensino médio</option>
        <option value="superior">Ensino superior</option>
        <option value="superior">Outro</option>
    </select>

    <br>
    <hr>

    <label for="exampleDataList" class="form-label">Cidade</label>
    <input class="form-
control" list="datalistOptions" id="exampleDataList" placeholder="Digite uma cidade...">
    <datalist id="datalistOptions">
        <option value="Belo Horizonte"></option>
        <option value="São Paulo"></option>
        <option value="Santos"></option>
        <option value="Porto Alegre"></option>
        <option value="Recife"></option>
        <option value="Outra"></option>
    </datalist>

    <br>
    <hr>

    <label class="form-label">Enviar Arquivo</label>
    <input type="file">
    <br>
    <span class="form-text">
        Apenas arquivos png e jpg.
    </span>

    <br>
    <hr>

    <label for="exampleColorInput" class="form-label">Seletor de Cores</label>

```

```
    <input type="color" class="form-control form-control-color" id="exampleColorInput" value="#021BFC"
           title="Escolha a sua Cor">

    <br>
    <hr>

    <button type="submit" class="btn-outline-secondary btn-lg">Enviar</button>

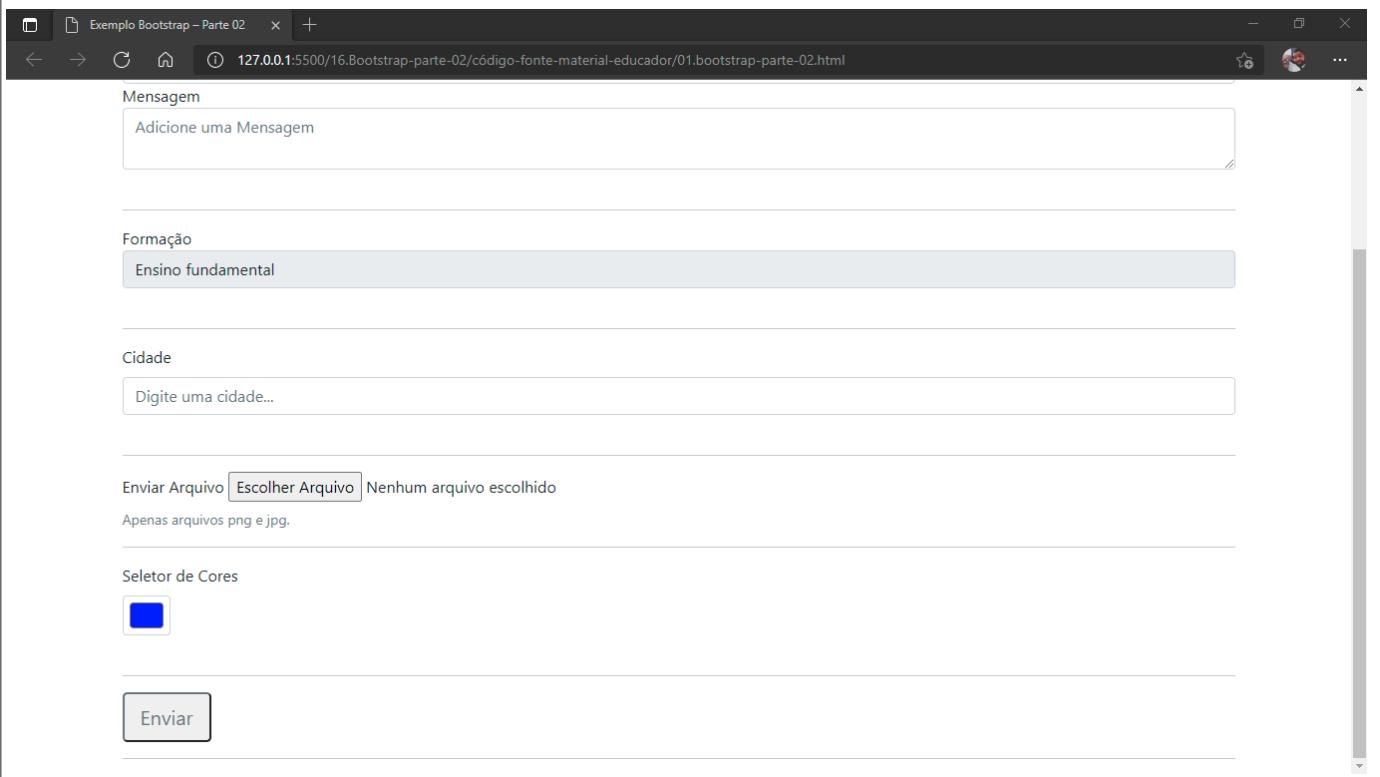
    <br>
    <hr>

</form>
</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"
        integrity="sha34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6o0ypYoCJhGGScy+8"
        crossorigin="anonymous"></script>

</body>
```

Visualize resultado da sua página web.



The screenshot shows a web browser window with the title "Exemplo Bootstrap – Parte 02". The address bar displays the URL "127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html". The page content is a form built with Bootstrap:

- Mensagem:** A text input field with placeholder "Adicione uma Mensagem".
- Formação:** A dropdown menu showing "Ensino fundamental".
- Cidade:** A text input field with placeholder "Digite uma cidade...".
- Enviar Arquivo:** Buttons for "Enviar Arquivo" and "Escolher Arquivo". Below them, it says "Nenhum arquivo escolhido" and "Apenas arquivos png e jpg.". A small note indicates "Apenas arquivos png e jpg.".
- Seletor de Cores:** A color picker showing a blue square.
- Enviar:** A submit button.

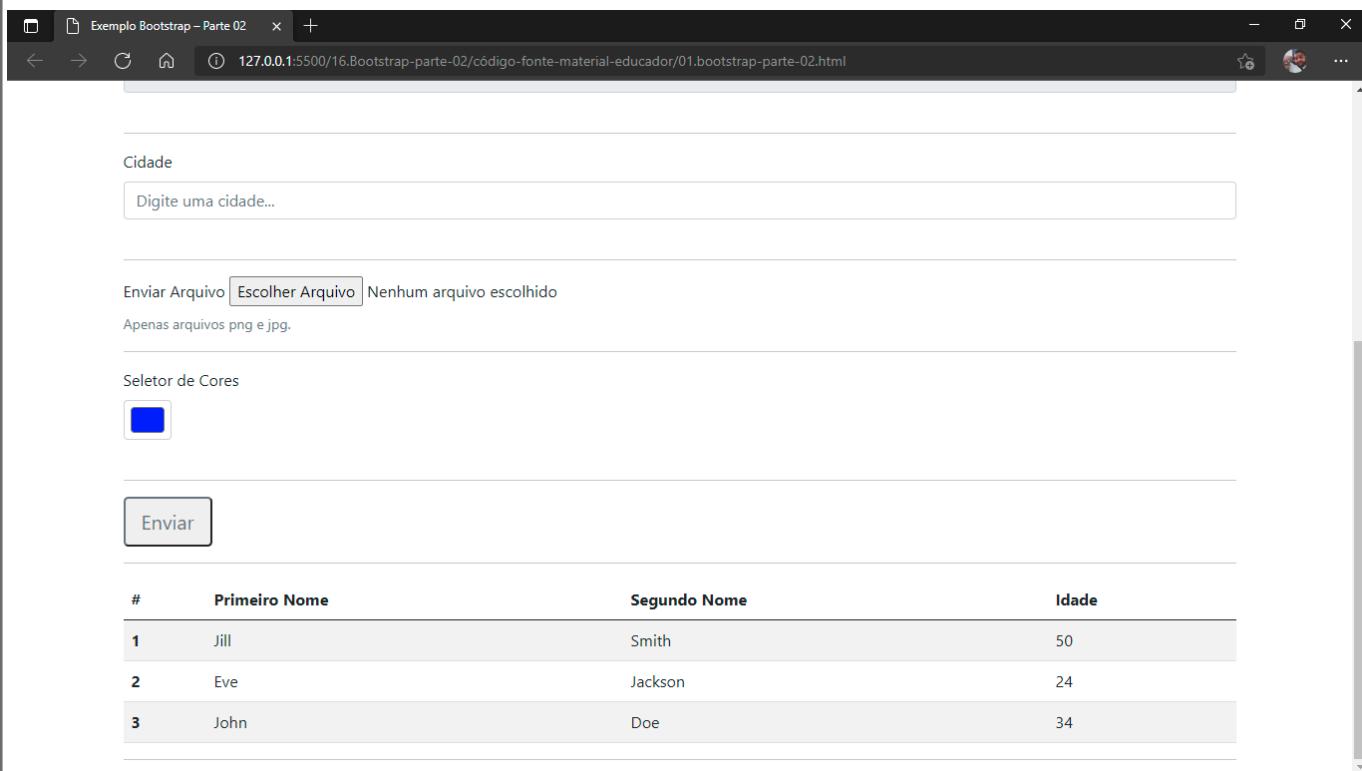
Em seguida, vamos aprender a trabalhar com **tabelas** utilizando o **Bootstrap**. Para não ficar muito grande o código, **vamos mostrar apenas a parte que deve ser atualizada e indicar onde você deve inserir essa atualização**. Vamos criar uma nova seção logo abaixo da anterior e inserir o trecho a seguir.

```
<section class="container">
    <!-- Tabela Primeiro Exemplo -->
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Primeiro Nome</th>
                <th scope="col">Segundo Nome</th>
                <th scope="col">Idade</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope="row">1</th>
                <td>Jill</td>
                <td>Smith</td>
                <td>50</td>
            </tr>
            <tr>
                <th scope="row">2</th>
                <td>Eve</td>
                <td>Jackson</td>
                <td>24</td>
            </tr>
            <tr>
                <th scope="row">3</th>
                <td>John</td>
                <td>Doe</td>
                <td>34</td>
            </tr>
        </tbody>
    </table>

    <br>
    <hr>
</section>
```

Observe que utilizamos a classe **.table**, que define o estilo de uma tabela no Bootstrap, com o cabeçalho (elemento `<thead>` e `<th>`) em destaque, uma linha sólida separando o cabeçalho do restante dos dados, entre outros estilos.

A classe **.table-striped** faz o corpo da tabela (**<tbody>**) ficar zebraado. As linhas são definidas com cores de fundo alternadas. O atributo **scope** faz com o conteúdo ficar em destaque dos outros valores da tabela. A documentação sobre os estilos de formulários está disponível em: <https://getbootstrap.com/docs/5.0/content/tables/>



The screenshot shows a web browser window titled "Exemplo Bootstrap – Parte 02". The URL in the address bar is "127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html". The page contains the following elements:

- Cidade**: An input field with placeholder "Digite uma cidade...".
- Enviar Arquivo** and **Escolher Arquivo** buttons, with a message "Nenhum arquivo escolhido". Below it says "Apenas arquivos png e jpg."
- Seletor de Cores**: A color picker with a blue square preview.
- Enviar**: A button with a rounded rectangle border.
- Table**:

#	Primeiro Nome	Segundo Nome	Idade
1	Jill	Smith	50
2	Eve	Jackson	24
3	John	Doe	34

Vamos fazer um segundo exemplo de tabela e utilizar outros estilos do Bootstrap. **Na mesma seção anterior, insira o código a seguir logo após a tag <hr>**.

```
<!-- Tabela Segundo Exemplo -->





```

```

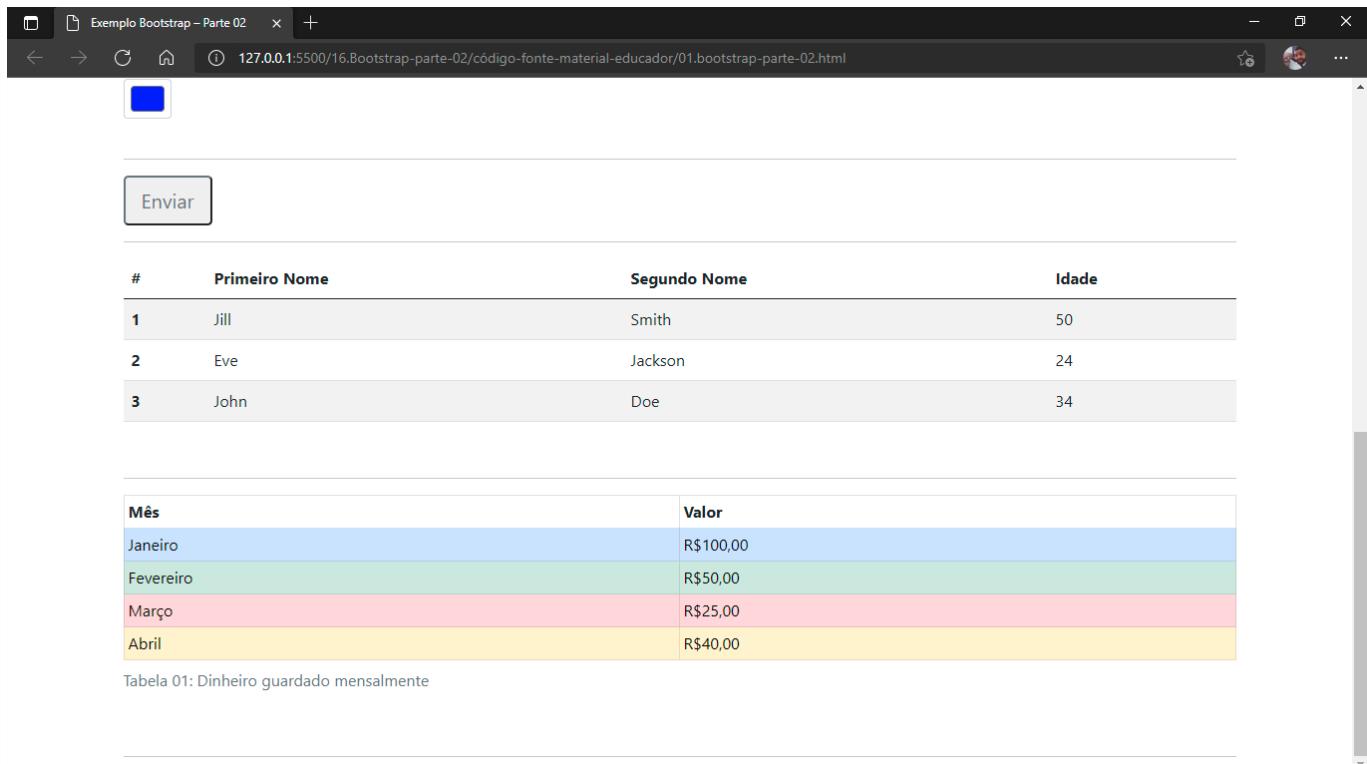
</tr>
<tr class="table-danger">
    <td>Março</td>
    <td>R$25,00</td>
</tr>
<tr class="table-warning">
    <td>Abril</td>
    <td>R$40,00</td>
</tr>
</tbody>
</table>

<br>
<hr>

```

Nesse exemplo, definimos a borda da tabela com a classe **.table-bordered**, o efeito ao passar o mouse pelas linhas do corpo da tabela com a classe **.table-hover** e o tamanho pequeno para a fonte e a célula com a classe **.table-sm**.

Também usamos os estilos de cores (classes **.table-primary**, **.table-success**, **.table-danger**, **.table-warning**, etc.) para alterar a cor da fonte de do plano de fundo das linhas. Você pode usar esses estilos nas colunas ou em uma célula também. Visualize resultado da sua página web.



#	Primeiro Nome	Segundo Nome	Idade
1	Jill	Smith	50
2	Eve	Jackson	24
3	John	Doe	34

Mês	Valor
Janeiro	R\$100,00
Fevereiro	R\$50,00
Março	R\$25,00
Abril	R\$40,00

Tabela 01: Dinheiro guardado mensalmente

Agora, vamos aprender como trabalhar de forma mais elaborada com **listas** no Bootstrap. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

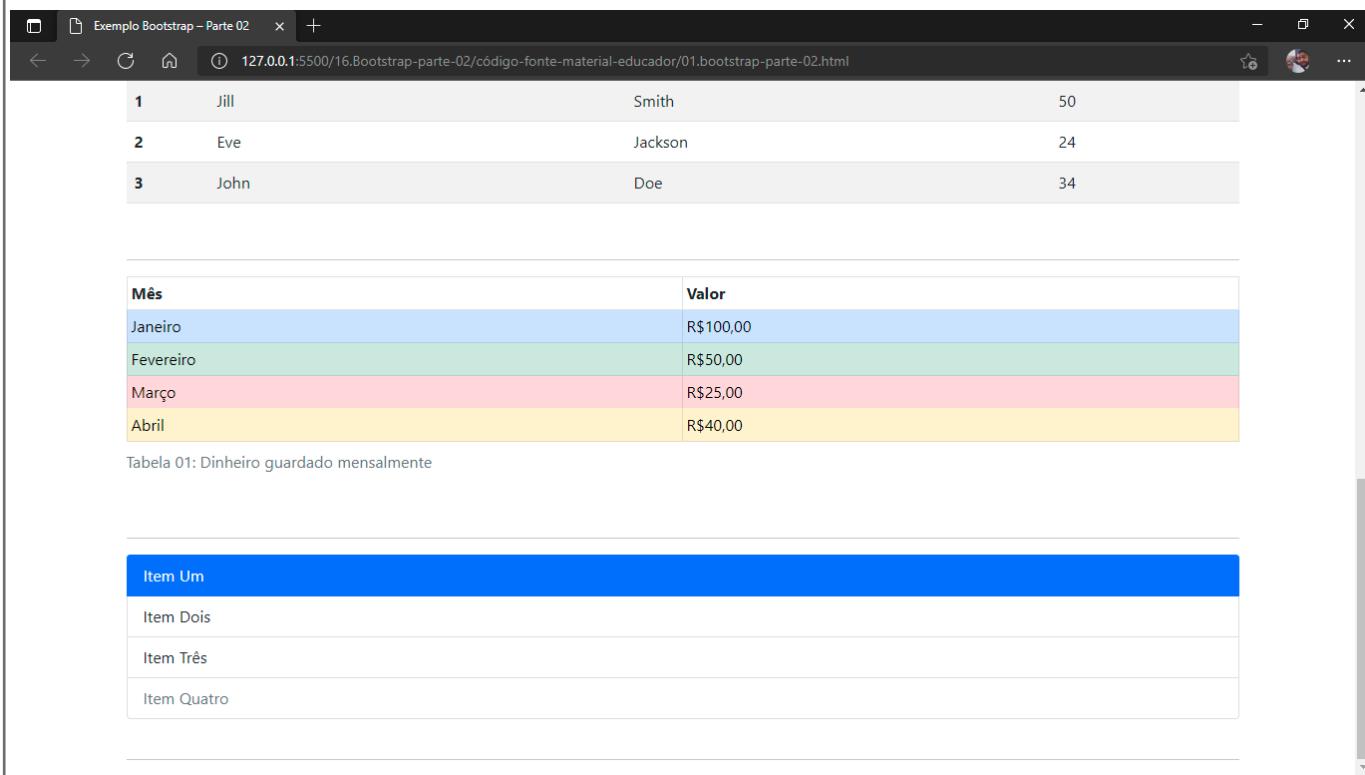
```
<!-- Exemplo de Listas com a classe List Group -->
<section class="container">
    <!-- Lista básica -->
    <ul class="list-group">
        <li class="list-group-item active" aria-current="true">Item Um</li>
        <li class="list-group-item">Item Dois</li>
        <li class="list-group-item">Item Três</li>
        <li class="list-group-item disabled" aria-disabled="true">Item Quatro</li>
    </ul>

    <br>
    <hr>

</section>
```

A classe **list-group** define a lista no Bootstrap e cada item da lista você deve colocar a classe **.list-group-item**. Se você quiser deixar um item ativo, você pode usar a classe **.active** e a classe **.disabled** para desabilitar um item. A documentação sobre os estilos de listas está disponível em: <https://getbootstrap.com/docs/5.0/components/list-group/>

Visualize resultado da sua página web.



The screenshot shows a browser window with the title "Exemplo Bootstrap – Parte 02". The address bar indicates the URL is 127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html.

Table 1:

1	Jill	Smith	50
2	Eve	Jackson	24
3	John	Doe	34

Table 2:

Mês	Valor
Janeiro	R\$100,00
Fevereiro	R\$50,00
Março	R\$25,00
Abril	R\$40,00

Tabela 01: Dinheiro guardado mensalmente

List Group:

- Item Um
- Item Dois
- Item Três
- Item Quatro

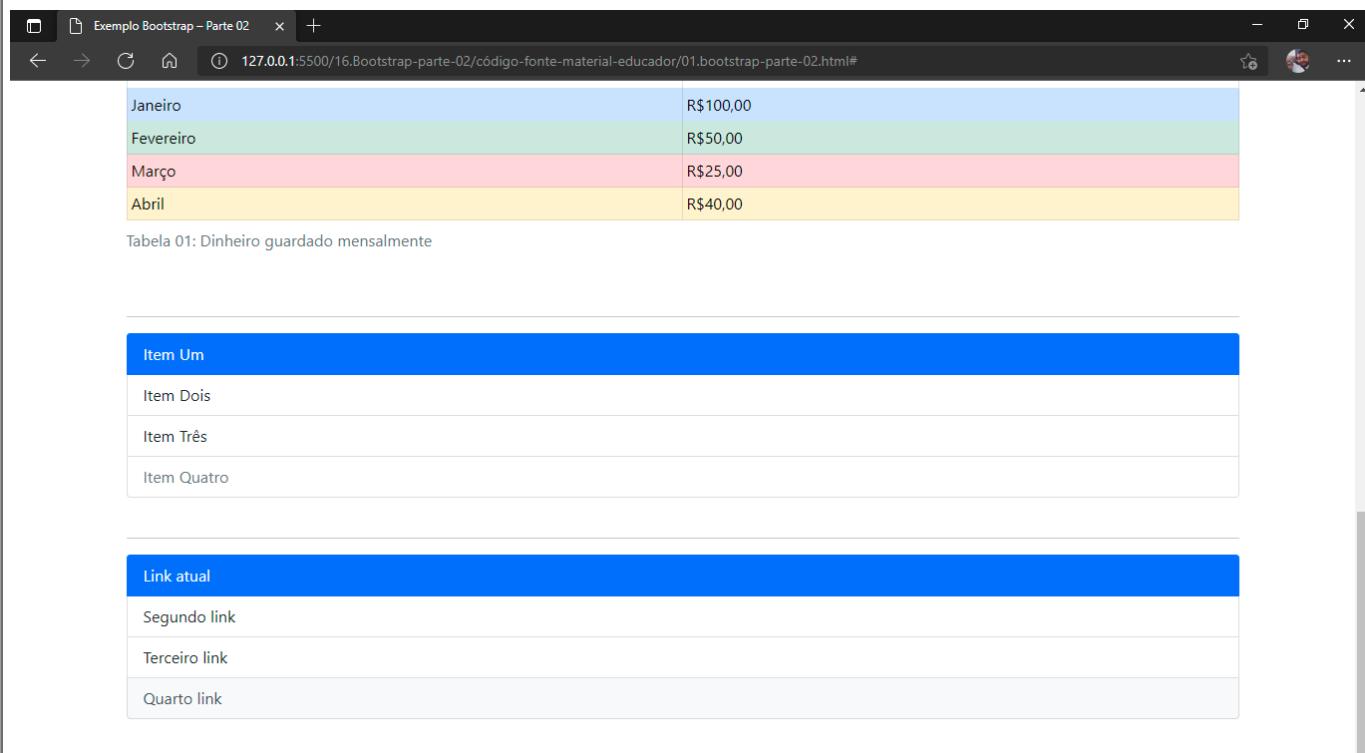
Vamos fazer um **segundo exemplo de listas** e utilizar outros estilos do Bootstrap. Na mesma seção anterior, insira o código a seguir logo após a tag <hr>.

```
<!-- Lista com Links -->
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action active" aria-current="true">
    Link atual
  </a>
  <a href="#" class="list-group-item list-group-item-action">Segundo link</a>
  <a href="#" class="list-group-item list-group-item-action">Terceiro link</a>
  <a href="#" class="list-group-item list-group-item-action">Quarto link</a>
</div>

<br>
<hr>
```

Nesse exemplo, definimos uma lista de links que podem ser usadas com um menu de navegação no seu site. Observe que a classe **.list-group-item-action** produz um efeito quando o mouse passa por cima do link.

Visualize resultado da sua página web.



The screenshot shows a web browser window with the title "Exemplo Bootstrap – Parte 02". The address bar indicates the URL is 127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html#.

Table:

Janeiro	R\$100,00
Fevereiro	R\$50,00
Março	R\$25,00
Abril	R\$40,00

Caption: Tabela 01: Dinheiro guardado mensalmente

List Group:

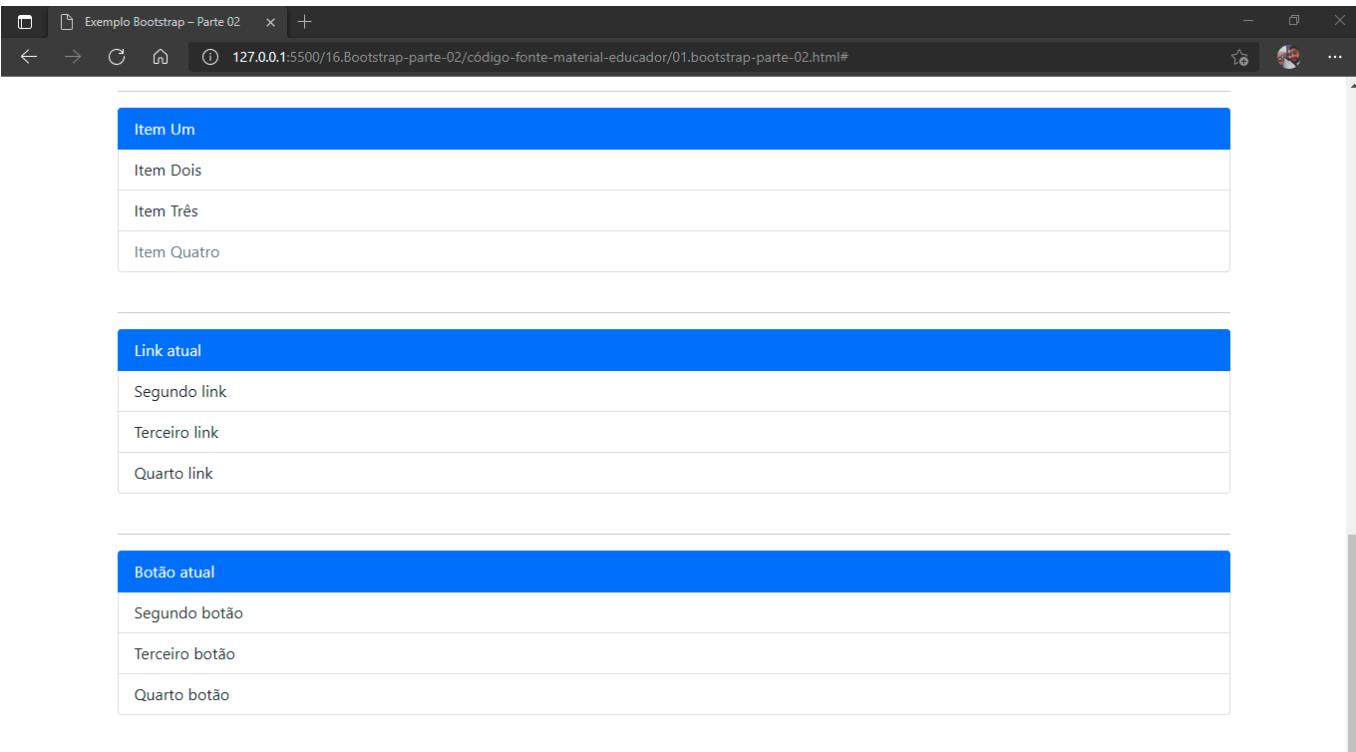
- Link atual
- Segundo link
- Terceiro link
- Quarto link

Também podemos criar uma **lista de botões no Bootstrap**. Vamos ver o trecho de código a seguir. Na mesma seção anterior, insira o código a seguir logo após a tag <hr>.

```
<!-- Lista com botões -->
<div class="list-group">
    <button type="button" class="list-group-item list-group-item-action active" aria-current="true">
        Botão atual
    </button>
    <button type="button" class="list-group-item list-group-item-action">Segundo botão</button>
    <button type="button" class="list-group-item list-group-item-action">Terceiro botão</button>
    <button type="button" class="list-group-item list-group-item-action">Quarto botão</button>
</div>

<br>
<hr>
```

Visualize resultado da sua página web.



The screenshot shows a browser window titled "Exemplo Bootstrap – Parte 02". The address bar indicates the URL is 127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html#.

The page displays three separate examples of button lists:

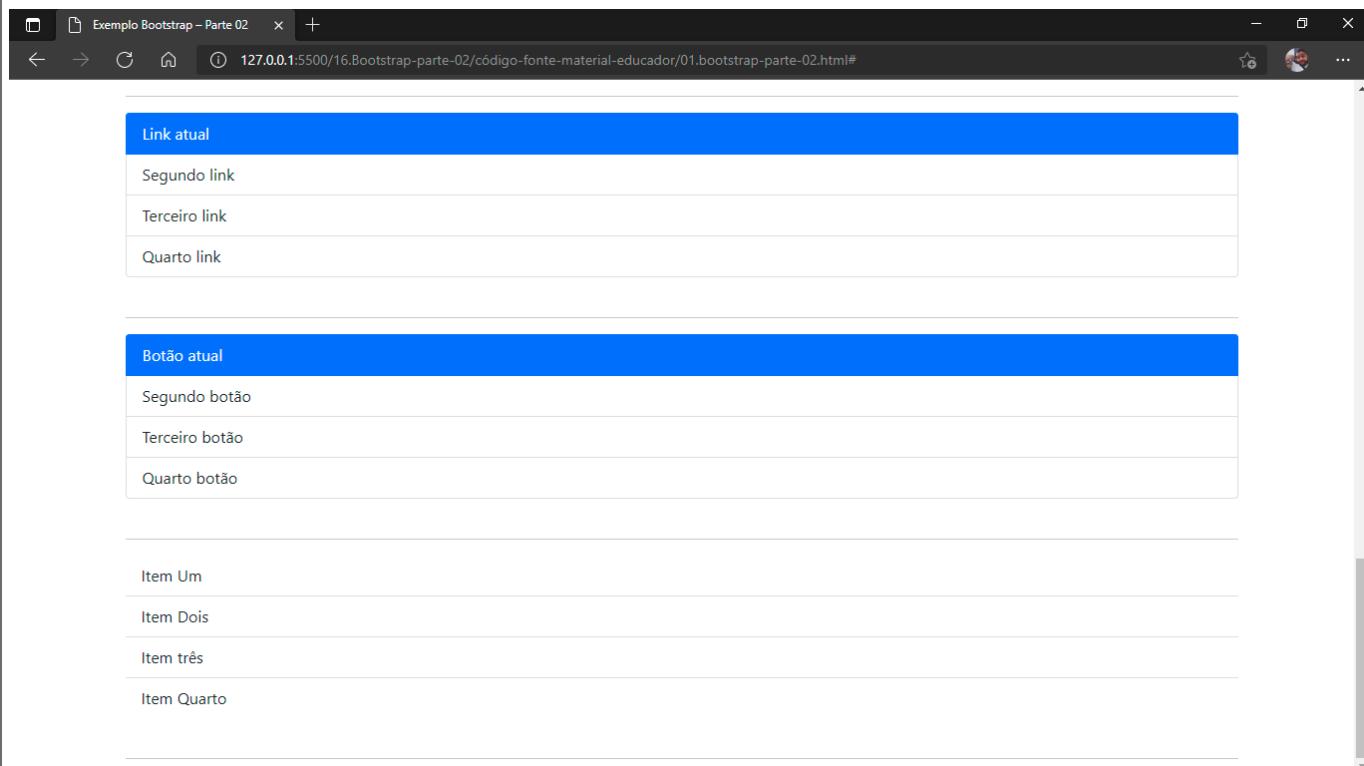
- Example 1:** A list group with four items. The first item is highlighted with a blue background and the class "active".
 - Item Um
 - Item Dois
 - Item Três
 - Item Quatro
- Example 2:** A list group with four items. The first item is highlighted with a blue background and the class "active".
 - Link atual
 - Segundo link
 - Terceiro link
 - Quarto link
- Example 3:** A list group with four items. The first item is highlighted with a blue background and the class "active".
 - Botão atual
 - Segundo botão
 - Terceiro botão
 - Quarto botão

Podemos utilizar um efeito **Flush**, que **remove algumas bordas da lista** através da classe **.list-group-flush**. Na mesma seção anterior, insira o código logo após a tag **<hr>**.

```
<!-- Lista com efeito flush -->
<ul class="list-group list-group-flush">
  <li class="list-group-item">Item Um</li>
  <li class="list-group-item">Item Dois</li>
  <li class="list-group-item">Item três</li>
  <li class="list-group-item">Item Quarto</li>
</ul>

<br>
<hr>
```

Visualize resultado da sua página web.



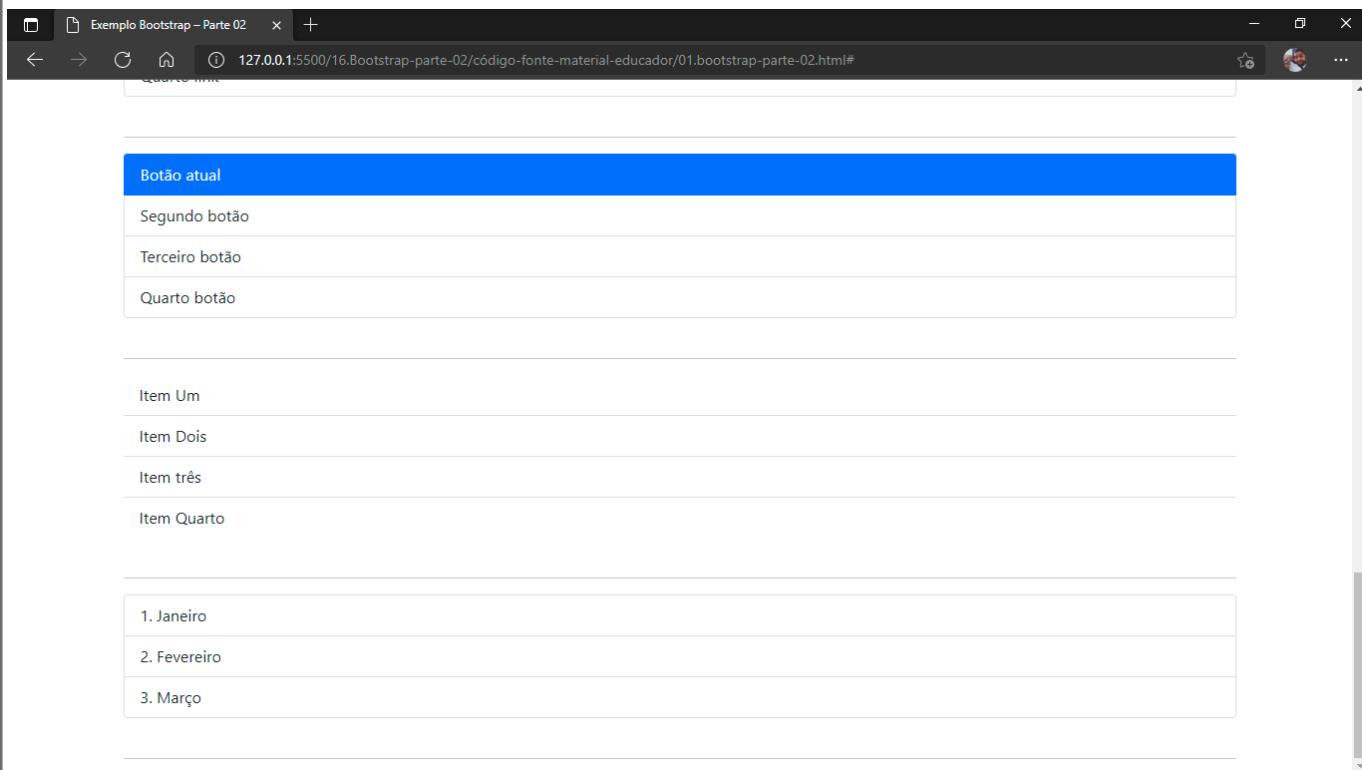
The screenshot shows a browser window with the title "Exemplo Bootstrap - Parte 02". The page content is divided into three main sections:

- Link atual**: Contains four items: "Segundo link", "Terceiro link", and "Quarto link".
- Botão atual**: Contains four items: "Segundo botão", "Terceiro botão", and "Quarto botão".
- Lista com efeito flush**: Contains four items: "Item Um", "Item Dois", "Item três", and "Item Quarto".

Se quisermos, também, existe **lista numerada** com a classe **.list-group-numbered**. Na mesma seção anterior, insira o código a seguir logo após a tag **<hr>**.

```
<!-- Lista numerada -->
<ol class="list-group list-group-numbered">
  <li class="list-group-item">Janeiro</li>
  <li class="list-group-item">Fevereiro</li>
  <li class="list-group-item">Março</li>
</ol>
<br>
<hr>
```

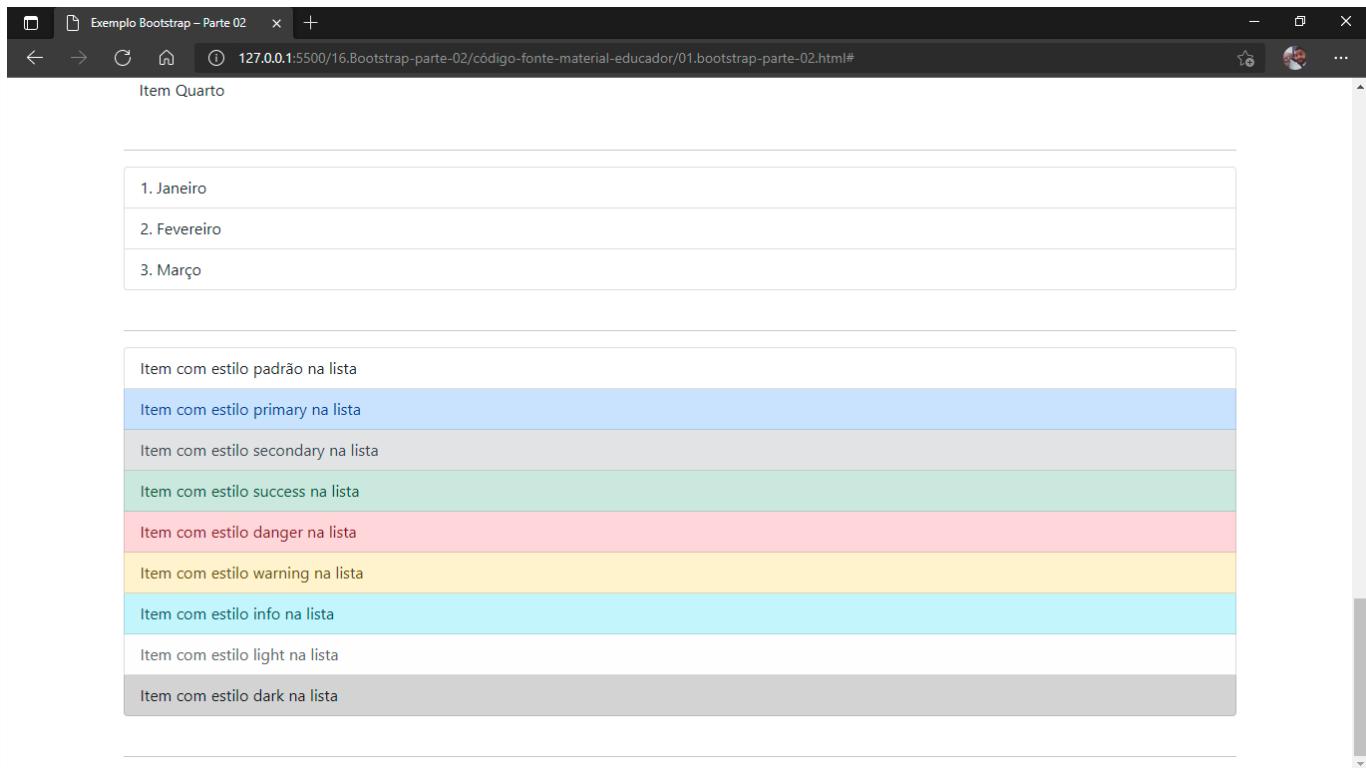
Visualize resultado da sua página web.



E como último exemplo de listas dessa aula, podemos **colorir cada item da lista**. Na mesma seção anterior, insira o código a seguir logo após a tag **<hr>**.

```
<!-- Lista colorida -->
<ul class="list-group">
    <li class="list-group-item">Item com estilo padrão na lista</li>
    <li class="list-group-item list-group-item-primary">Item com estilo primary na lista</li>
    <li class="list-group-item list-group-item-secondary">Item com estilo secondary na lista</li>
    <li class="list-group-item list-group-item-success">Item com estilo success na lista</li>
    <li class="list-group-item list-group-item-danger">Item com estilo danger na lista</li>
    <li class="list-group-item list-group-item-warning">Item com estilo warning na lista</li>
    <li class="list-group-item list-group-item-info">Item com estilo info na lista</li>
    <li class="list-group-item list-group-item-light">Item com estilo light na lista</li>
    <li class="list-group-item list-group-item-dark">Item com estilo dark na lista</li>
</ul>
<br>
<hr>
```

Visualize resultado da sua página web.



The screenshot shows a web browser window titled "Exemplo Bootstrap – Parte 02". The URL is 127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html#. The content displays a list of months in a card, followed by a vertical stack of colored cards representing different Bootstrap list styles: standard, primary, secondary, success, danger, warning, info, light, and dark.

Cards

Vamos fazer **Cards** na página web utilizando o Bootstrap. Vamos criar outra seção logo depois da anterior e inserir o seguinte código.

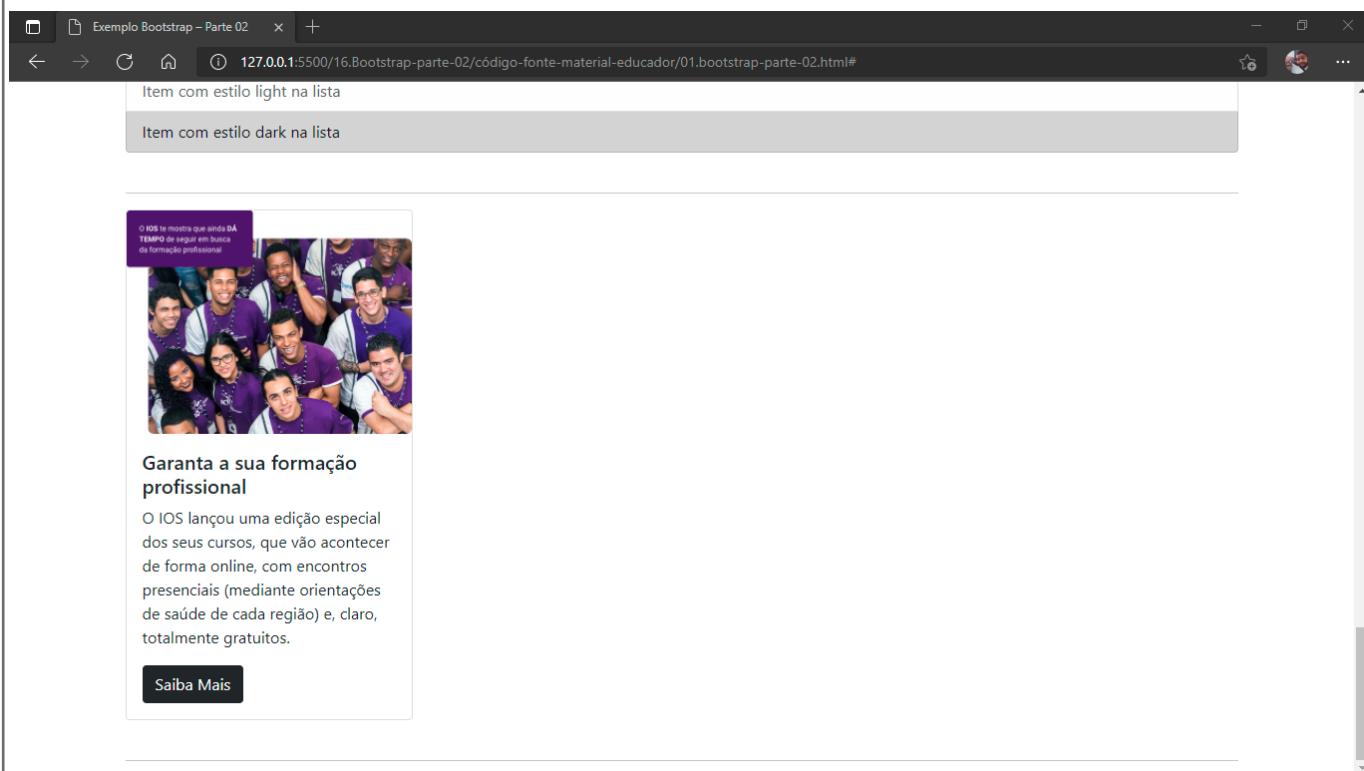
```
<!-- Cards -->
<section class="container">
    <div class="card" style="width: 18rem;">
        
        <div class="card-body">
            <h5 class="card-title">Garanta a sua formação profissional</h5>
            <p class="card-text">O IOS lançou uma edição especial dos seus cursos, que vão acontecer de forma online,
                com
                encontros presenciais (mediante orientações de saúde de cada região) e,
                claro, totalmente gratuitos.</p>
            <a href="#" class="btn btn-dark">Saiba Mais</a>
        </div>
    </div>
    <br>
    <hr>
```

Card é um **container de conteúdo flexível e extensível**. Inclui opções de cabeçalhos e rodapés, uma ampla variedade de conteúdo, cores de fundo contextuais e opções de exibição poderosas. Você pode fazer um Card simples com apenas textos ou uma lista ou até mais elaborado como no exemplo mostrado, que usa uma imagem, títulos etc.

No exemplo, usamos a classe **.card-body** para definir o corpo do Card, a classe **.card-title** para estilizar o título do Card e a classe **.card-text** para definir estilo ao conteúdo textual do Card. A documentação sobre os estilos de cards está disponível em:

<https://getbootstrap.com/docs/5.0/components/card/>

Visualize resultado da sua página web.

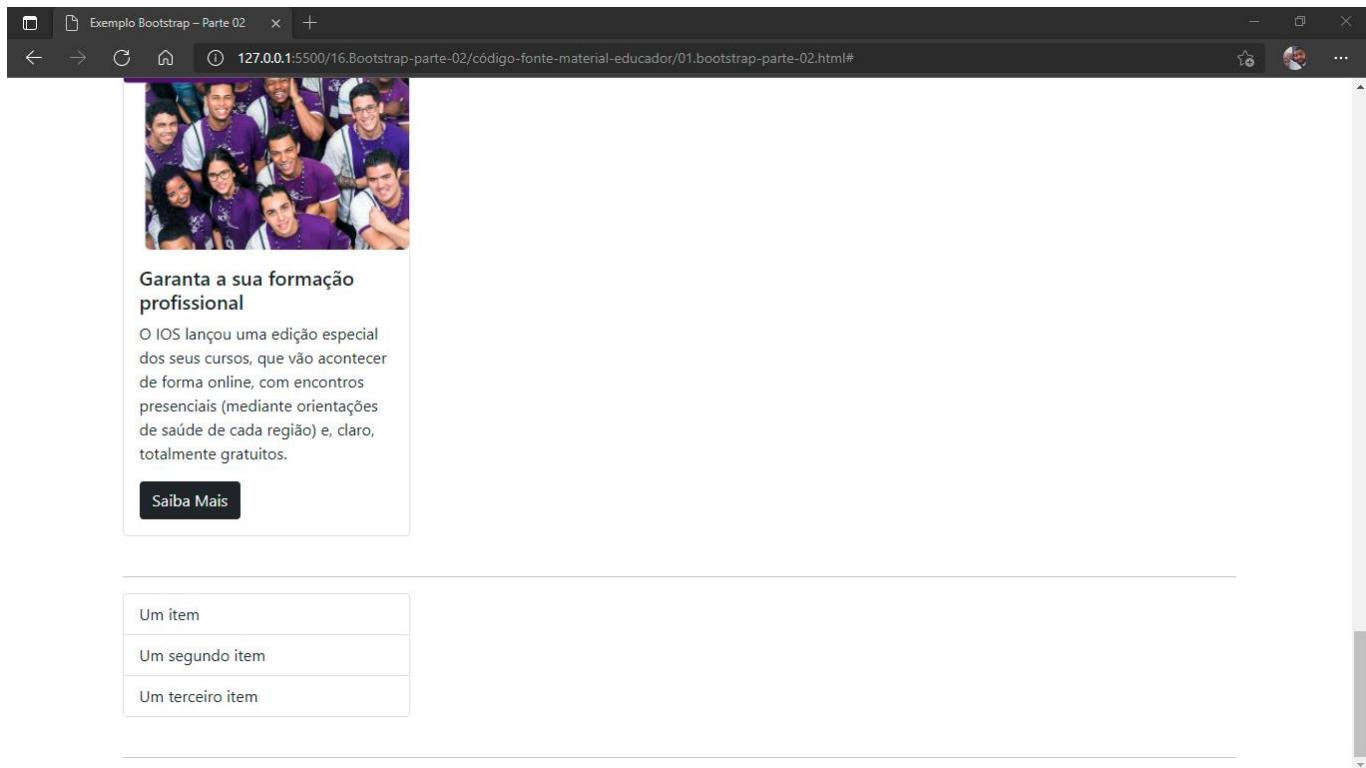


Vamos ver um exemplo de **Card** com **listas não ordenadas**. Na mesma seção anterior, insira o código a seguir logo após a tag **<hr>**.

```
<!-- Cards com Listas exemplo 01 -->
<div class="card" style="width: 18rem; >
  <ul class="list-group list-group-flush">
    <li class="list-group-item">Um item</li>
    <li class="list-group-item">Um segundo item</li>
    <li class="list-group-item">Um terceiro item</li>
  </ul>
</div>

<br>
<hr>
```

Visualize resultado da sua página web.

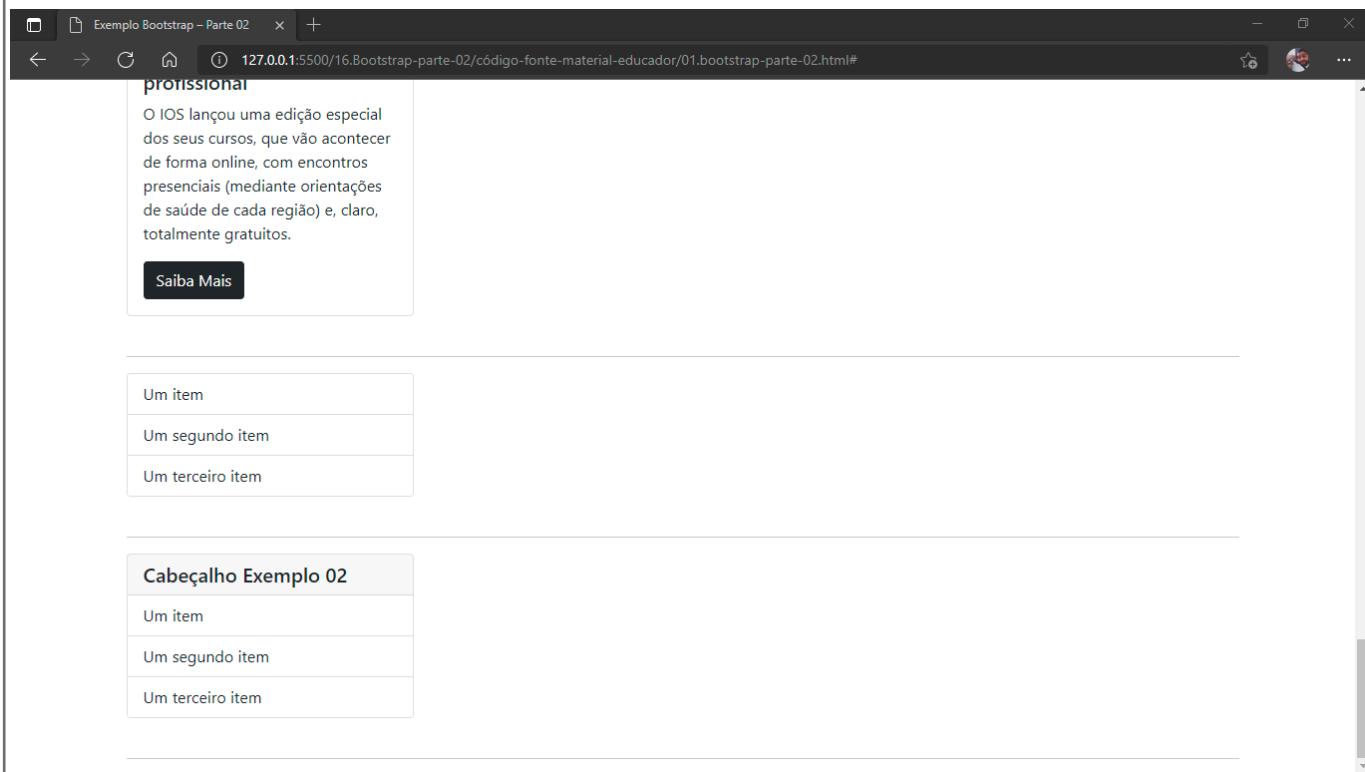


Mais um exemplo com listas, só que agora definimos o estilo do cabeçalho da lista com a classe **.card-header** e com rodapé com a classe **.card-footer**. Na mesma seção anterior, insira o código a seguir logo após a tag <hr>.

```
<!-- Cards com Listas exemplo 02 -->
<div class="card" style="width: 18rem;">
  <h5 class="card-header">Cabeçalho Exemplo 02</h5>
  <ul class="list-group list-group-flush">
    <li class="list-group-item">Um item</li>
    <li class="list-group-item">Um segundo item</li>
    <li class="list-group-item">Um terceiro item</li>
  </ul>
</div>

<br>
<hr>
```

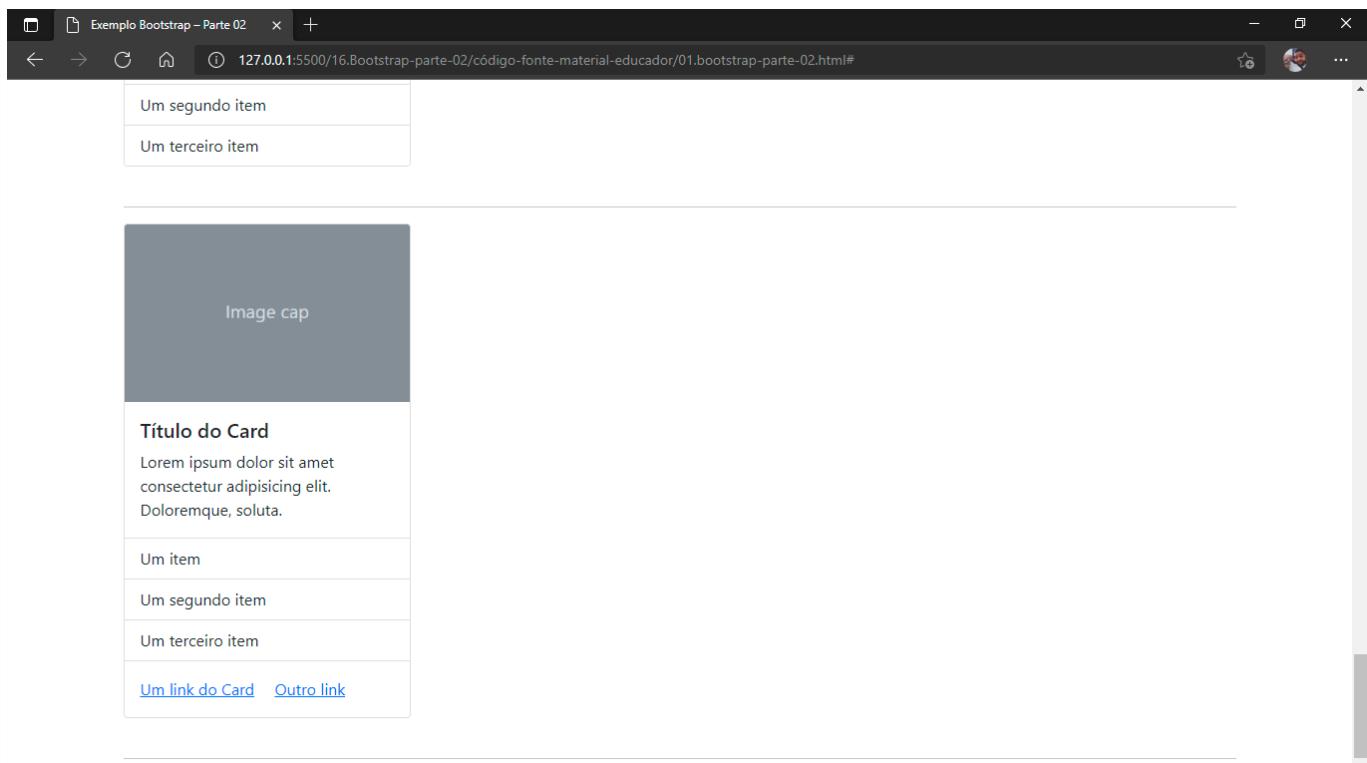
Visualize resultado da sua página web.



Por fim, podemos ter **muitos elementos dentro do Card** como mostra o código a seguir. Na mesma seção anterior, insira o código a seguir logo após a tag <**hr**>.

```
<!-- Card com diversos elementos-->
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Título do Card</h5>
    <p class="card-text">Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloremque, soluta.</p>
  </div>
  <ul class="list-group list-group-flush">
    <li class="list-group-item">Um item</li>
    <li class="list-group-item">Um segundo item</li>
    <li class="list-group-item">Um terceiro item</li>
  </ul>
  <div class="card-body">
    <a href="#" class="card-link">Um link do Card</a>
    <a href="#" class="card-link">Outro link</a>
  </div>
</div>
<br>
<hr>
```

Visualize resultado da sua página web.



The screenshot shows a web browser window with the title "Exemplo Bootstrap – Parte 02". The URL in the address bar is "127.0.0.1:5500/16.Bootstrap-parte-02/código-fonte-material-educador/01.bootstrap-parte-02.html#". The page displays a Bootstrap card component. The card has a gray header section with the text "Image cap". Below the header, there is a title "Título do Card" followed by a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloremque, soluta.". The card also contains three horizontal list items: "Um item", "Um segundo item", and "Um terceiro item". At the bottom of the card, there are two blue links: "Um link do Card" and "Outro link".