

# DESENVOLVIMENTO WEB

HTML, CSS, JAVASCRIPT, PHP, MYSQL E MUITO MAIS



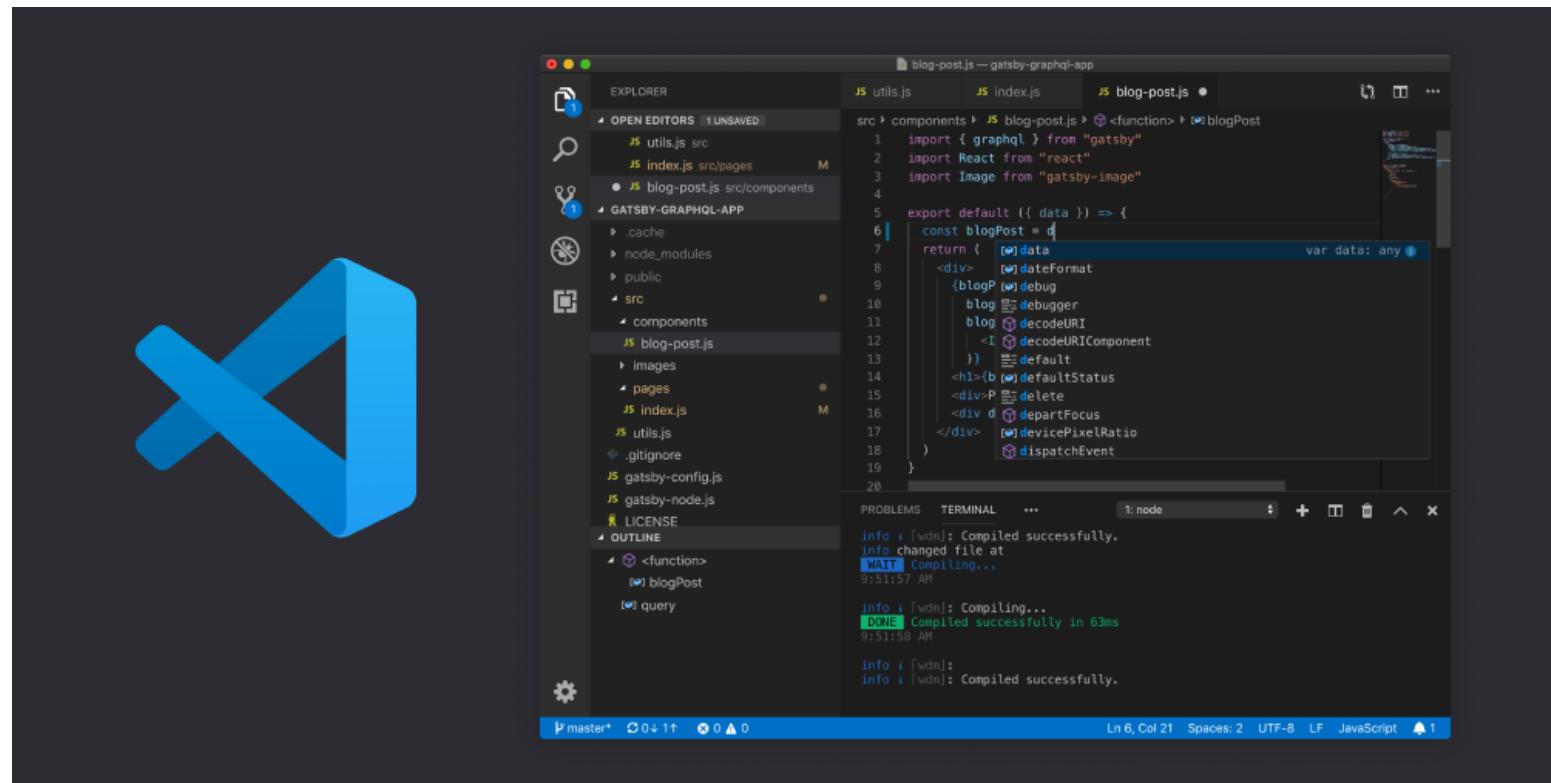
# HTML + CSS

## FUNDAMENTOS BÁSICOS

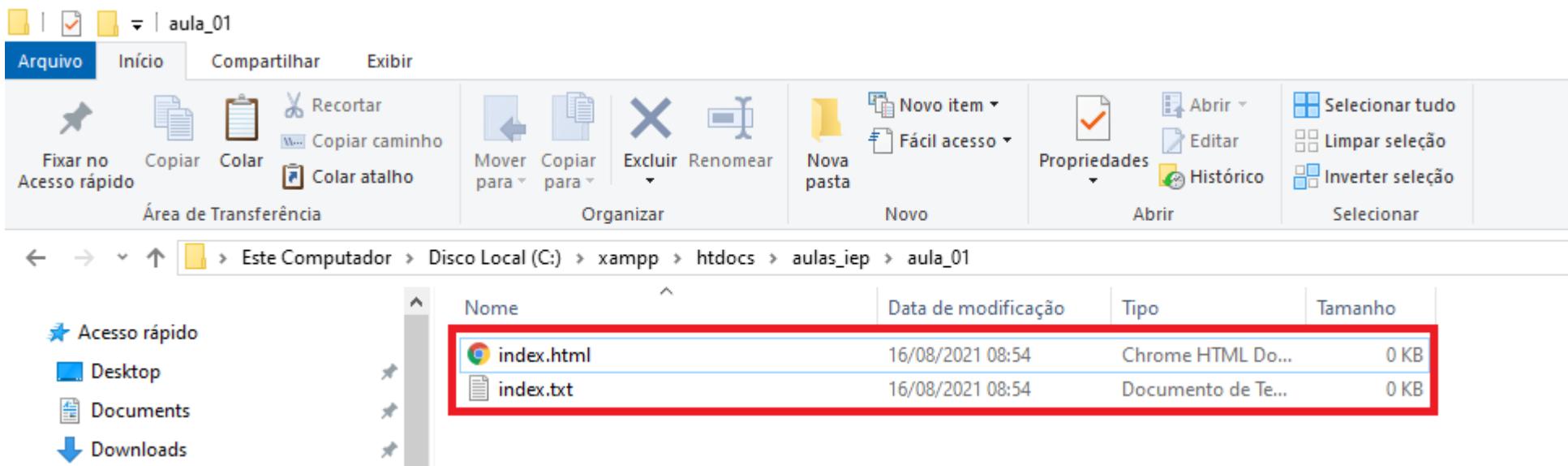
Para iniciarmos nosso curso, primeiramente será necessário instalar na maquina uma ferramenta de edição de código, no caso utilizaremos o **Visual Code Studio**, ferramenta gratuita da Microsoft.

Link para Download: <https://code.visualstudio.com/>

Tutoria de instalação: <https://www.youtube.com/watch?v=R6YsIWRUFk>



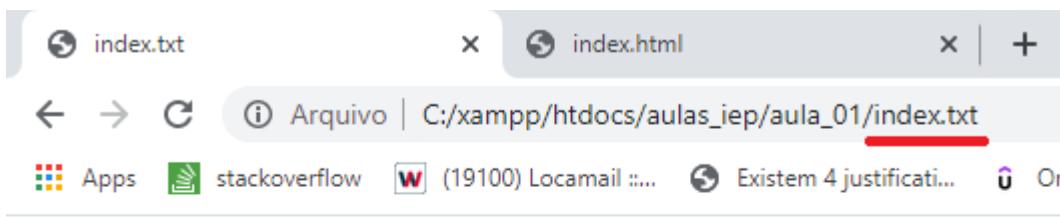
Para criarmos nossa primeira página **HTML** (HyperText Markup Language, que significa: Linguagem de Marcação de Hipertexto) é necessário criar um arquivo com a extensão **.html** para o navegador converter nosso código em arte visual. Exemplo:



Veja que o próprio sistema operacional já identifica que páginas com extensão **.html** deverão ser iniciadas através de um navegador da internet.

Neste outro exemplo vemos o que acontece quando um navegador faz a leitura de uma página com extensão **.txt** e outra com **.html**:

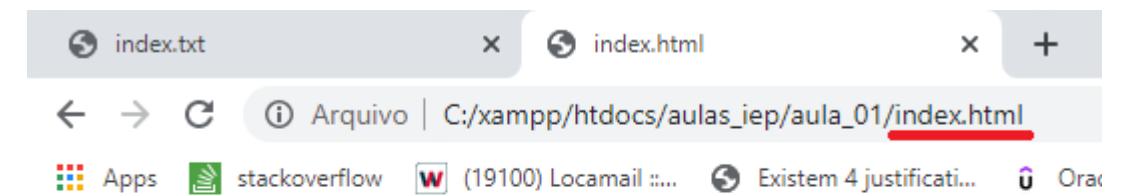
### Página com extensão **.txt**



A screenshot of a browser window showing the file 'index.txt'. The address bar shows 'Arquivo | C:/xampp/htdocs/aulas\_iep/aula\_01/index.txt'. The page content is:

```
<html>
  <!-- TRADUCAO DA PAGINA -->
  <meta charset="UTF-8">
  Olá Mundo!
</html>
```

### Página com extensão **.html**



A screenshot of a browser window showing the file 'index.html'. The address bar shows 'Arquivo | C:/xampp/htdocs/aulas\_iep/aula\_01/index.html'. The page content is:

```
Olá Mundo!
```

#### Nota

Quando nomeamos o arquivo como **index.html** estamos informando ao servidor de hospedagem que essa **será a primeira página a ser exibida em nosso site**.



# DICAS BÁSICAS DE PROGRAMAÇÃO

ORGANIZAÇÃO E ESTRUTURAÇÃO



Antes de programarmos, é necessário entendermos o conceito de organização do nosso código:

Sempre que abrirmos um **bloco** que receberá conteúdo, **dentro desse bloco damos um tab** (tecla tab) para que o código sempre esteja organizado, o próprio Visual Studio Code já nos sugere o uso do **tab** quando abrirmos um novo bloco de conteúdo, essa organização chamamos de **indentação**.

The screenshot shows an open HTML file named 'index.html' in Visual Studio Code. The code is as follows:

```
1 <html>
2
3   
4   <meta charset="UTF-8">
5
6   Olá Mundo!
7
8   <div style="border: solid 1px black; width: 200px">
9     Conteudo dentro da Div
10    </div>
11
12
13
14 </html>
```

Annotations explain the code structure:

- A red bracket on the left side of the first line is labeled **Bloco html principal (Pai)**.
- A green bracket on the left side of the second line is labeled **Conteúdo Pai**.
- A purple bracket on the left side of the eighth line is labeled **Bloco DIV (filho)**.
- A green bracket on the left side of the ninth line is labeled **Conteúdo Filho**.
- A green icon at the bottom left is labeled **Espaçamento (Tab)**, indicating the use of tabs for indentation.

Note que o próprio Visual Code Studio já identifica com um traço cinza-claro onde se inicia e finaliza o bloco e o conteúdo dentro dele:

The screenshot shows the Visual Studio Code interface with an HTML file named 'index.html' open. The code editor displays the following HTML structure:

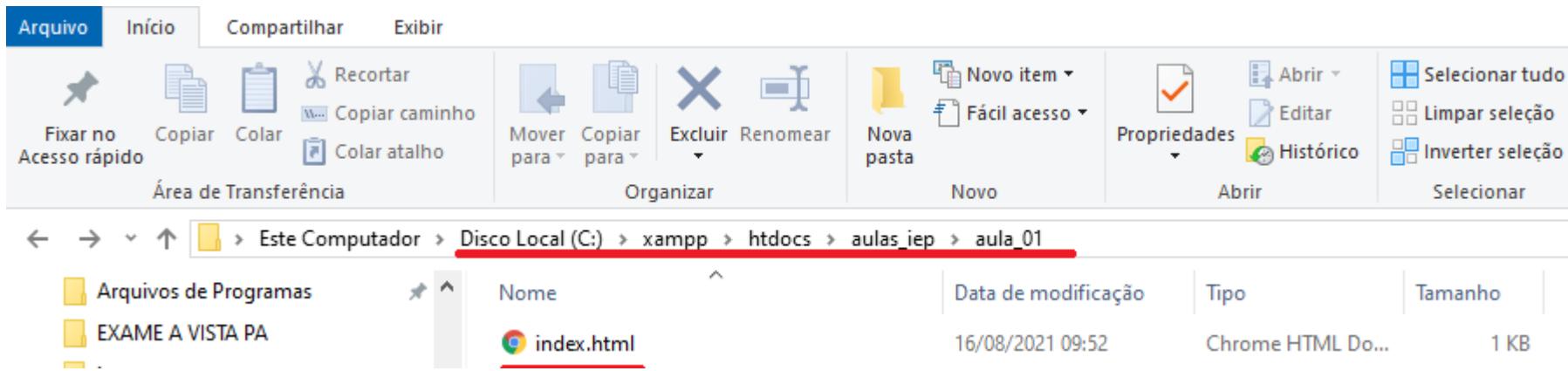
```
<html>
  <!-- TRADUCAO DA PAGINA -->
  <meta charset="UTF-8">
  Olá Mundo!
  <div style="border: solid 1px black; width: 200px">
    Conteudo dentro da Div
  </div>
</html>
```

Two red arrows point to the opening and closing tags of the 'div' element, specifically highlighting the 'border' and 'width' attributes. To the right of the code editor, a browser window titled 'index.html' shows the rendered content: 'Olá Mundo!' followed by a 200px wide black-bordered div containing the text 'Conteudo dentro da Div'.

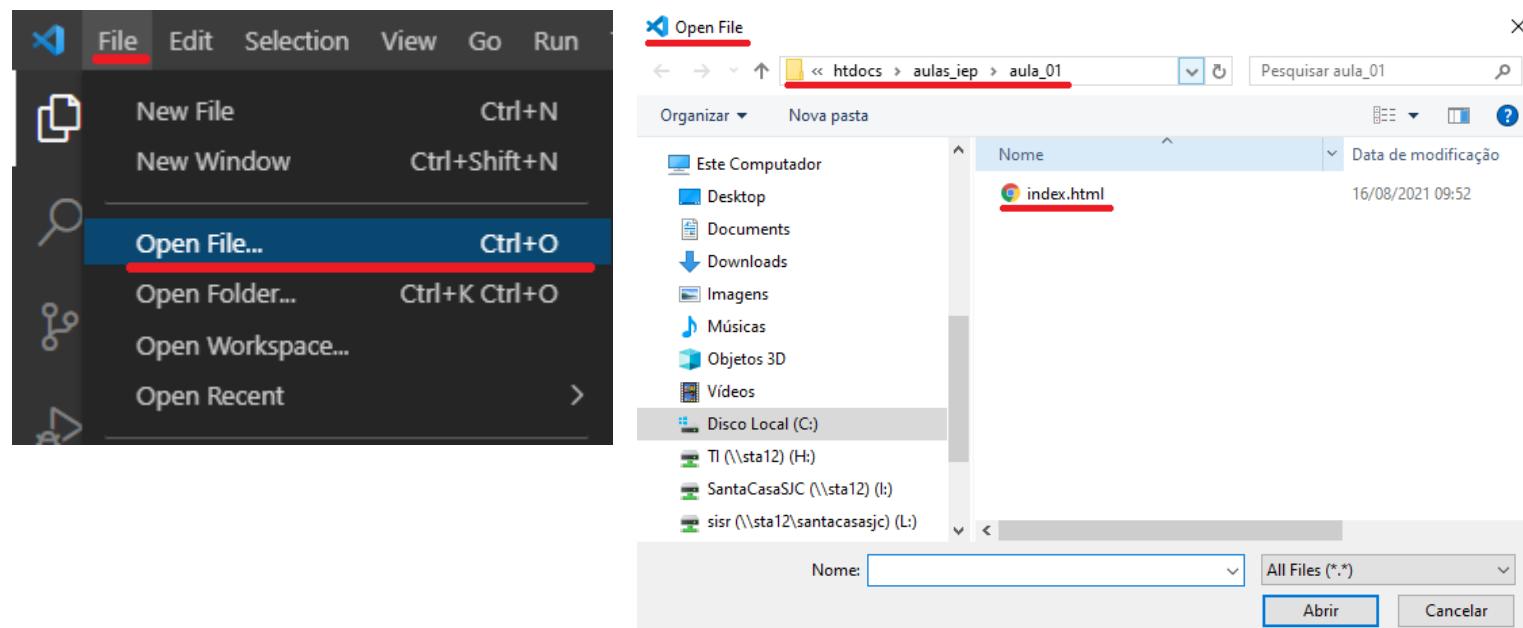
# PRIMEIRA PÁGINA WEB

HTML

Para criarmos nossa primeira página web, precisaremos criar um arquivo com extensão **.html**:



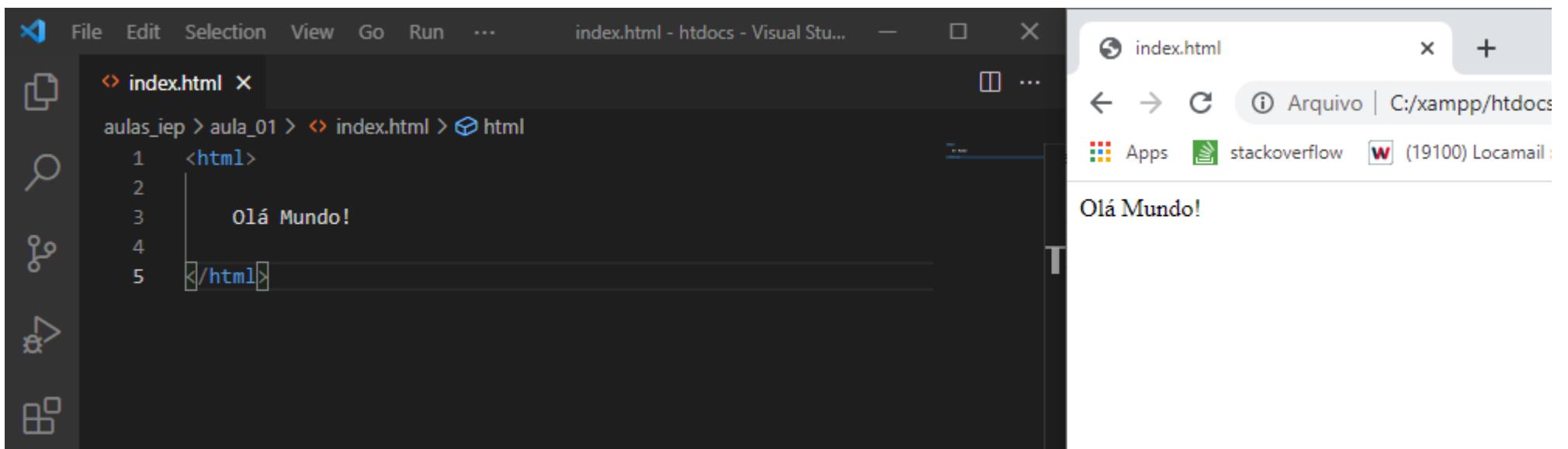
Agora precisamos abrir esse arquivo no Visual Studio Code:



Vamos iniciar com um simples Olá Mundo!

Veja que abrimos a tag `<html>` no inicio da página e no final fechamos essa tag `</html>`, note que ao fechar uma tag utilizamos a barra / dentro das chaves `</>`.

Não esqueçamos também da **indentação** e de **salvar** o arquivo antes de visualizar as alterações na página.

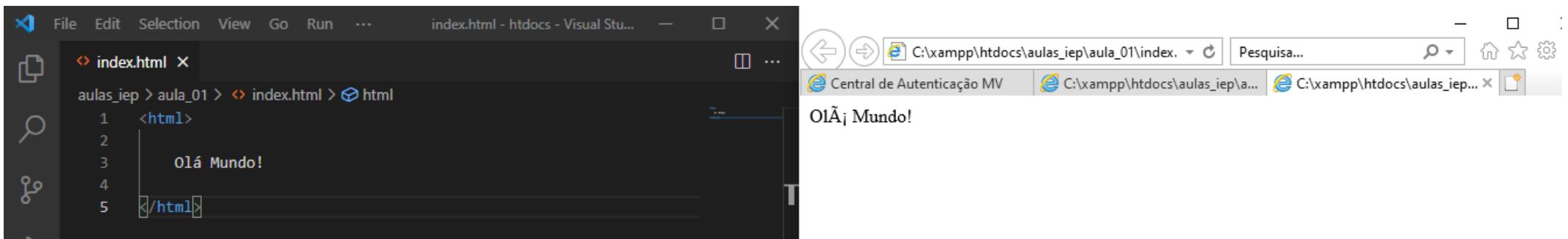


The image shows a screenshot of a computer interface. On the left is a dark-themed code editor window titled "index.html - htdocs - Visual Studio Code". The file contains the following HTML code:

```
File Edit Selection View Go Run ...
index.html - htdocs - Visual Studio Code
index.html x
aulas_iep > aula_01 > index.html > html
1 <html>
2   |
3   |   Olá Mundo!
4   |
5 </html>
```

To the right of the code editor is a web browser window titled "index.html". The address bar shows "Arquivo | C:/xampp/htdocs". The browser displays the text "Olá Mundo!".

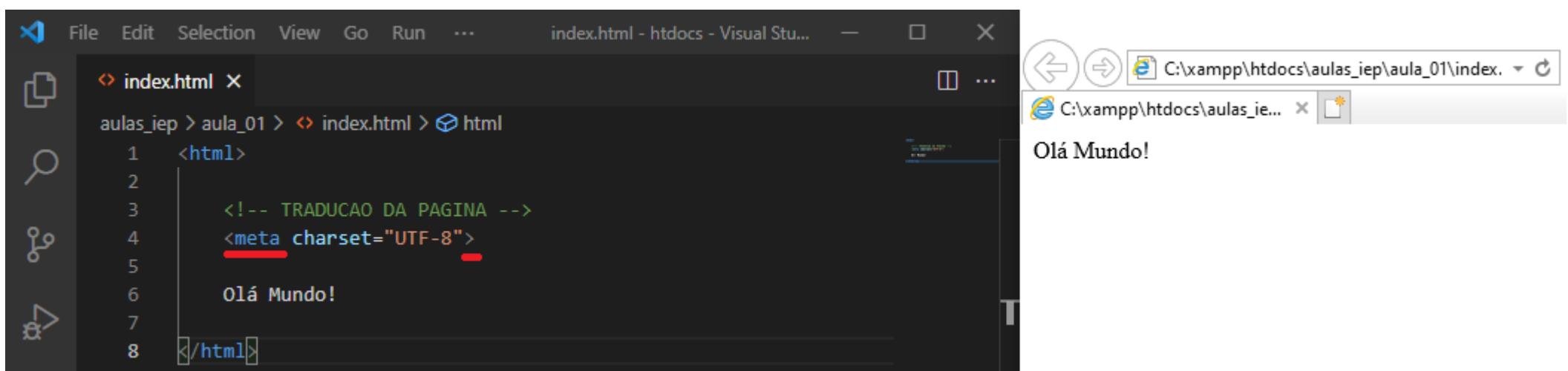
Veja que alguns navegadores não fazem a adaptação da escrita para português automaticamente, Então chamaremos a **tag meta** com a funcionalidade **charset**, para resolver esse problema.



```
File Edit Selection View Go Run ... index.html - htdocs - Visual Studio Code
index.html x
aulas_iep > aula_01 > index.html > html
1 <html>
2
3     Olá Mundo!
4
5 </html>
```

OlÃ¡ Mundo!

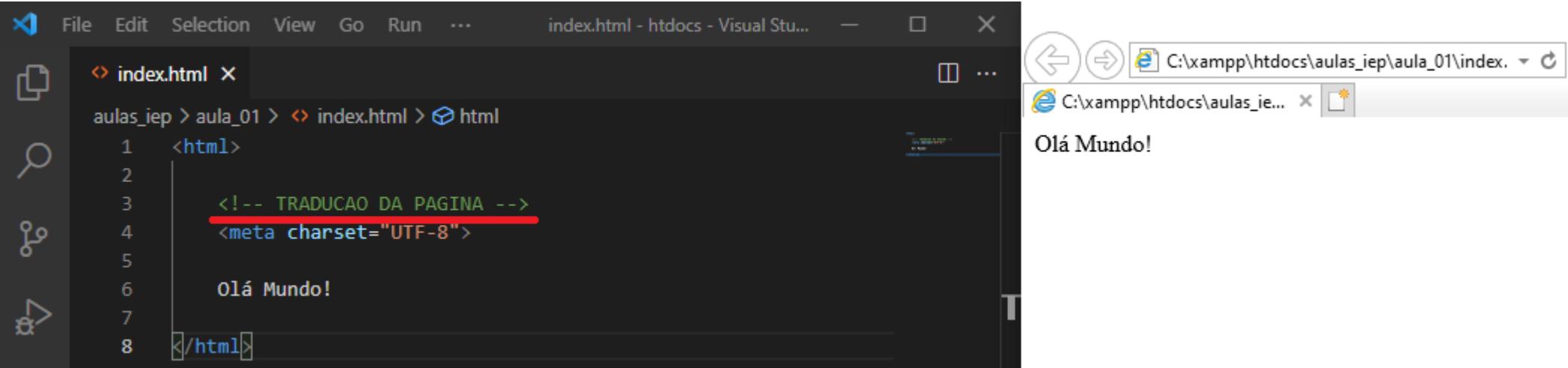
Note que abrimos a **tag <meta>** e na própria **abertura** inserirmos a informação do **charset**, e finalizamos a **abertura** da **tag** com apenas um **>**, nesse caso **não precisamos encerrar com </meta>**, pois esse tipo de **tag** não aceita conteúdos dentro dela.



```
File Edit Selection View Go Run ... index.html - htdocs - Visual Studio Code
index.html x
aulas_iep > aula_01 > index.html > html
1 <html>
2
3     
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8 </html>
```

Olá Mundo!

Para organizar melhor nosso código podemos inserir um **comentário**, para inserir o **comentário** utilizamos o código: `<!-- TEXTO QUALQUER -->`, com isso podemos organizar e identificar o que estamos fazendo em nosso código para consultas posteriores, **sem interferir na página** a ser exibida para o usuário.



```
File Edit Selection View Go Run ... index.html - htdocs - Visual Studio Code
index.html X
aulas_iep > aula_01 > index.html > html
1 <html>
2
3     
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8 </html>
```

C:\xampp\htdocs\aulas\_iep\aula\_01\index.html

Olá Mundo!

### Nota

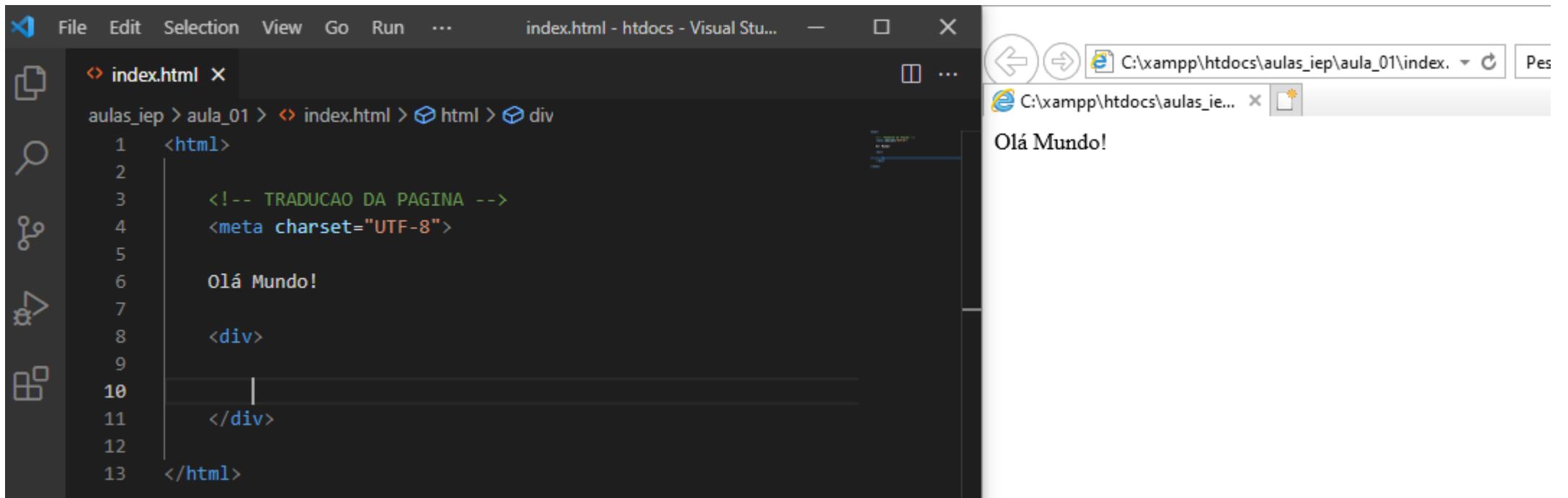
Recomendamos a utilização do `<meta charset="UTF-8">` para todas as aplicações, com isso podemos utilizar caracteres especiais e acentuação da nossa linguagem em todos os navegadores sem maiores problemas.



# UTILIZAÇÃO DE DIVS

HTML + CSS

Vamos aprender um pouco sobre **div**, que consideramos uma das **tags** mais importantes para o desenvolvimento web.



```
File Edit Selection View Go Run ... index.html - htdocs - Visual Studio Code
index.html x
aulas_iep > aula_01 > index.html > html > div
1 <html>
2
3     <!-- TRADUCAO DA PAGINA -->
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8     <div>
9
10    </div>
11
12
13 </html>
```

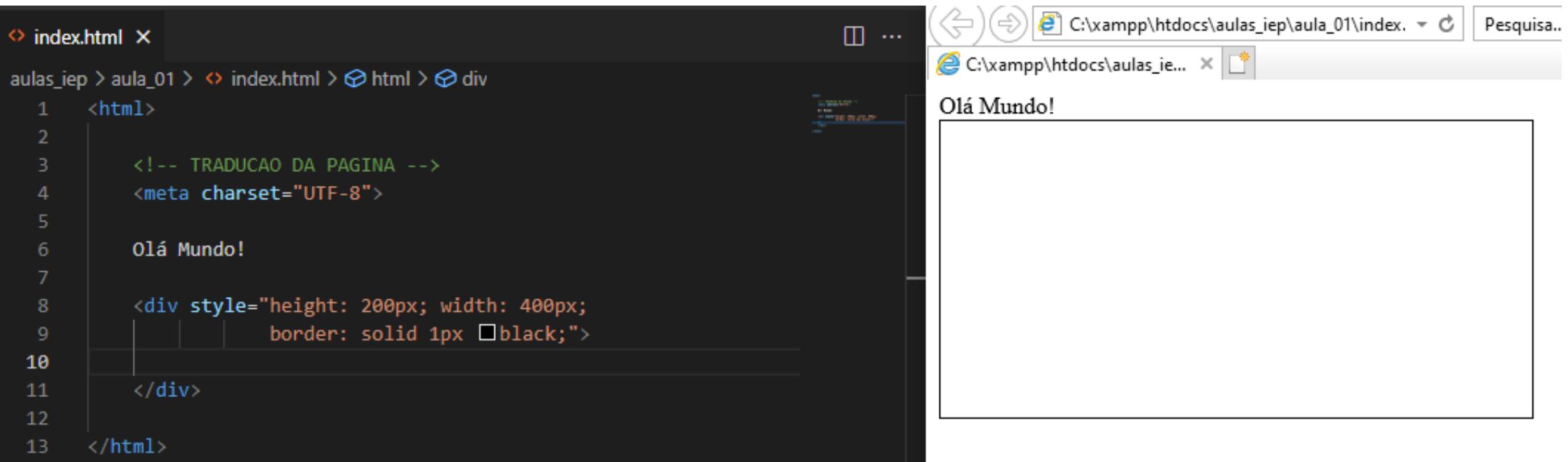
Olá Mundo!

Veja que ao utilizar uma **div** abrindo e fechando ela (**<div> </div>**) não há efeito nenhum na página, pois a **div** serve para organizar conteúdo e aplicarmos os efeitos estilizados que chamados de **CSS**.

Para podermos visualizar uma div precisamos passar três valores para ela:

Altura (**height**), largura (**width**) e borda (**border**)

Para passar esses valores para um **div**, utilizaremos uma funcionalidade da **div** chamado **style**, lembrando que os valores desse atributos serão passados dentro de aspas duplas “”, separados por ponto e vírgula ; exemplo:



```
index.html x
aulas_iep > aula_01 > index.html > html > div
1  <html>
2
3      <!-- TRADUCAO DA PAGINA --&gt;
4      &lt;meta charset="UTF-8"&gt;
5
6      Olá Mundo!
7
8      &lt;div style="height: 200px; width: 400px;
9          border: solid 1px black;"&gt;
10
11     &lt;/div&gt;
12
13 &lt;/html&gt;</pre>

C:\xampp\htdocs\aulas_iep\aula_01\index. Pesquisa..



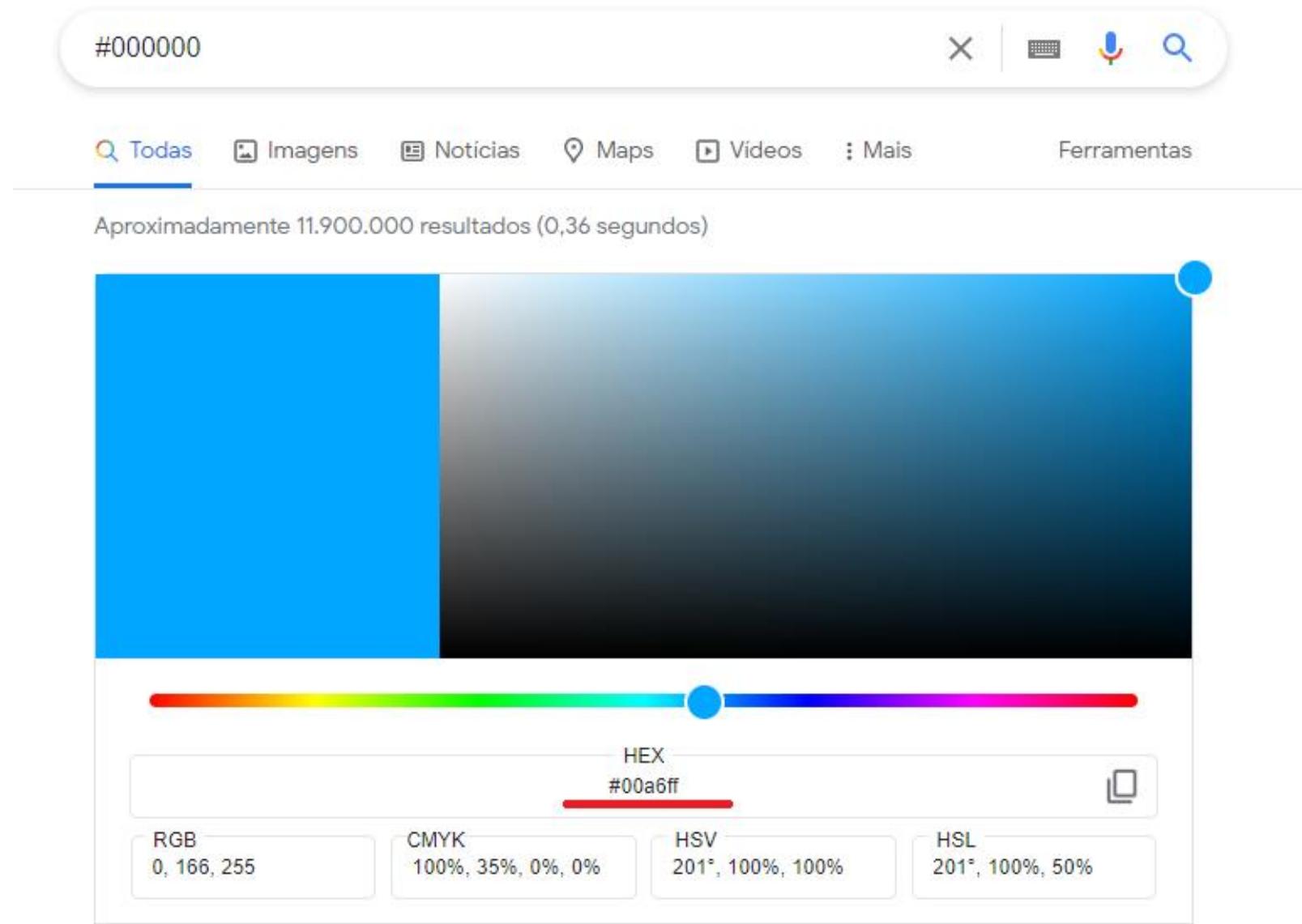
C:\xampp\htdocs\aulas_iep... * Olá Mundo!


```

Veja que quando passamos a **borda** desejada temos **três** valores, o primeiro **solid** se trata do tipo da borda que no caso é uma borda solida, o segundo é a **espessura da borda** que no caso escolhemos a largura de **1px** (1 pixel) e o terceiro se refere a **cor da borda**, que pode ser passado pelo nome da cor em inglês ou pelo valor hexadecimal da borda que no caso do black se trata do hexadecimal **#000000**.

```
<div style="height: 200px; width: 400px;  
           border: solid 1px black;">  
    1   2   3  
</div>
```

Podemos **consultar os valores hexadecimais das cores** em ferramentas de edição de imagem ou no próprio google:



Existe uma lista com os diversos tipos de borda que podemos utilizar descritas no site da w3schools:  
[https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)

Demonstration of the different border styles:

```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}
```

Result:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

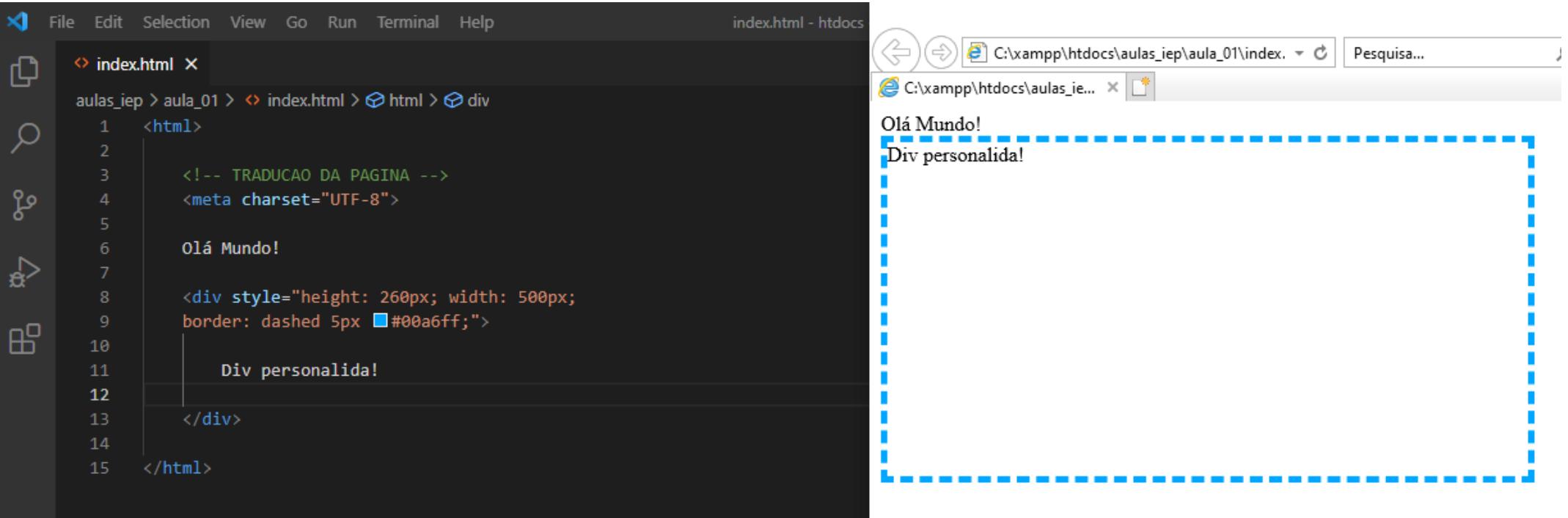
A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

Podemos definir o formato da div livremente modificando os valores no **style**, veja o exemplo abaixo:



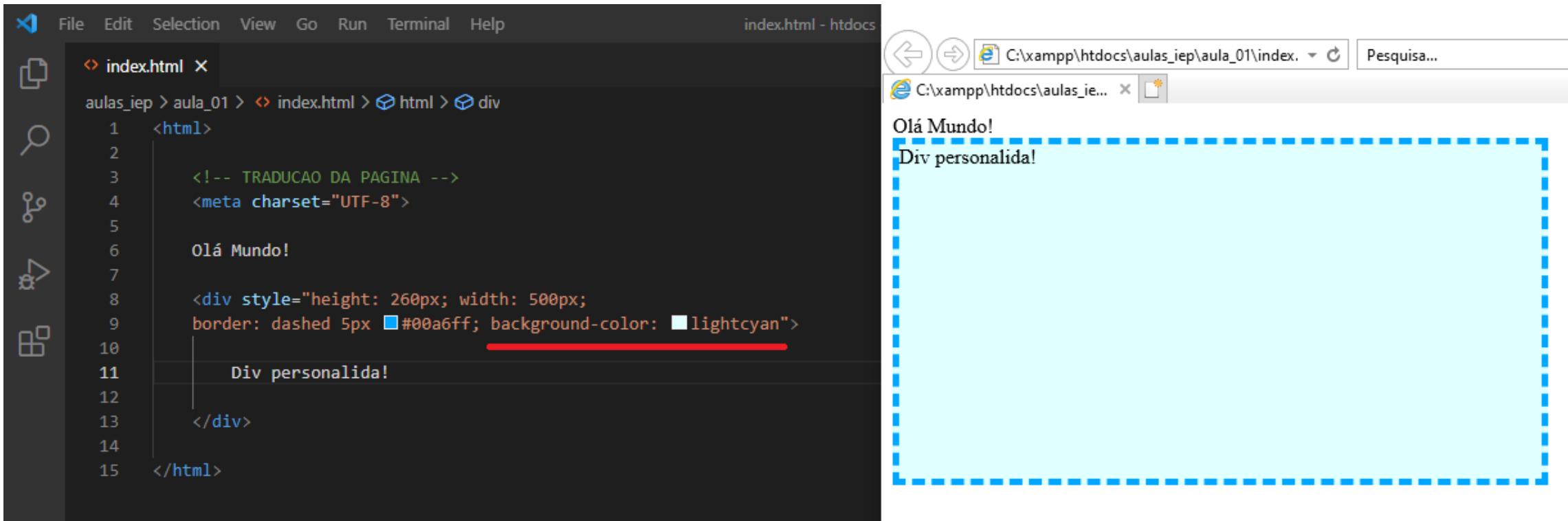
The image shows a code editor on the left and a web browser on the right. The code editor displays the file `index.html` with the following content:

```
File Edit Selection View Go Run Terminal Help
index.html - htdocs
index.html
aulas_iep > aula_01 > index.html > html > div
1 <html>
2
3     <!-- TRADUCAO DA PAGINA --&gt;
4     &lt;meta charset="UTF-8"&gt;
5
6     Olá Mundo!
7
8     &lt;div style="height: 260px; width: 500px;
9         border: dashed 5px #00a6ff;"&gt;
10
11         Div personalida!
12
13     &lt;/div&gt;
14
15 &lt;/html&gt;</pre>

The browser window on the right shows the rendered HTML. It contains the text "Olá Mundo!" and "Div personalida!". The "Div personalida!" text is contained within a <div> element that has a dashed blue border, matching the styling defined in the style attribute of the <div> tag in the code editor.


```

Para inserirmos uma cor de fundo na nossa div podemos utilizar o atributo **background-color** no style da nossa div (**CSS**):



The image shows a code editor on the left and a web browser on the right. The code editor displays the file 'index.html' with the following content:

```
File Edit Selection View Go Run Terminal Help
index.html - htdocs
index.html X
aulas_iep > aula_01 > index.html > html > div
1 <html>
2
3     
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8     <div style="height: 260px; width: 500px;
9         border: dashed 5px #00a6ff; background-color: lightcyan">
10        Div personalida!
11    </div>
12
13 </html>
```

A red horizontal bar highlights the line containing the CSS rule for the div element. The browser window on the right shows the rendered page with a blue dashed border and a light cyan background color for the div.

Depois de entendermos melhor o conceito básico de uma página **html** fica fácil de realizarmos as alterações que desejamos de forma simples, clara e objetiva.

Podemos inserir vários conteúdos dentro de uma **div**, inclusive outra **div**, veja exemplo:

The screenshot shows the Visual Studio Code interface. On the left is the code editor with the file `index.html` open. The code contains HTML and CSS. The browser preview on the right shows the resulting page.

```
File Edit Selection View Go Run Terminal Help
index.html - htdocs - Visual Studio Code
index.html X
aulas_iep > aula_01 > index.html > html > div
1 <html>
2
3     
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8     <div style="height: 260px; width: 500px;
9         border: dashed 5px #00a6ff; background-color: lightcyan">
10
11         Div personalida!
12
13         <!-- Pular linha-->
14         <br></br>
15
16         <div style="height: 120px; width: 300px; background-color: lightgreen">
17
18             Div dentro da div!
19
20         </div>
21
22     </div>
23
24 </html>
```

The browser preview shows the following content:

Olá Mundo!

Div personalida!

Div dentro da div!

The outer `div` has a dashed blue border and a light cyan background. The inner `div` has a solid green background.

Não foi necessário o uso de borda na nova `div` pois visualizamos apenas com o `background-color`.

Utilizamos dois `<br>` para pularmos as linhas entre o texto `Div personalidade` e o inicio da nova `div`.

Para inserir uma imagem na página, primeiro vamos criar uma pasta no nosso projeto chamado fotos:

Disco Local (C:) > xampp > htdocs > aulas_iep > aula_01 >			
Nome	Data de modificação	Tipo	Tamanho
fotos	16/08/2021 14:00	Pasta de arquivos	
index.html	16/08/2021 13:35	Chrome HTML Do...	1 KB

Feito isso, salvaremos alguma imagem nessa pasta, precisamos observar bem o **nome da pasta**, o **nome do arquivo** de imagem e a **extensão** dele:

The screenshot shows a Windows File Explorer window with a sidebar. In the main pane, there is a single file named "logo\_coracao.png". In the details pane, the file's properties are displayed in a dialog box titled "Propriedades de logo\_coracao.png". The "Geral" tab is selected, showing the file name "logo\_coracao.png", the type "Arquivo PNG (.png)", and the size "30 KB". The "Tipo de arquivo:" field is highlighted with a red box, showing "Arquivo PNG (.png)". The "Abre com:" field shows "Photos" and has a "Alterar..." button.

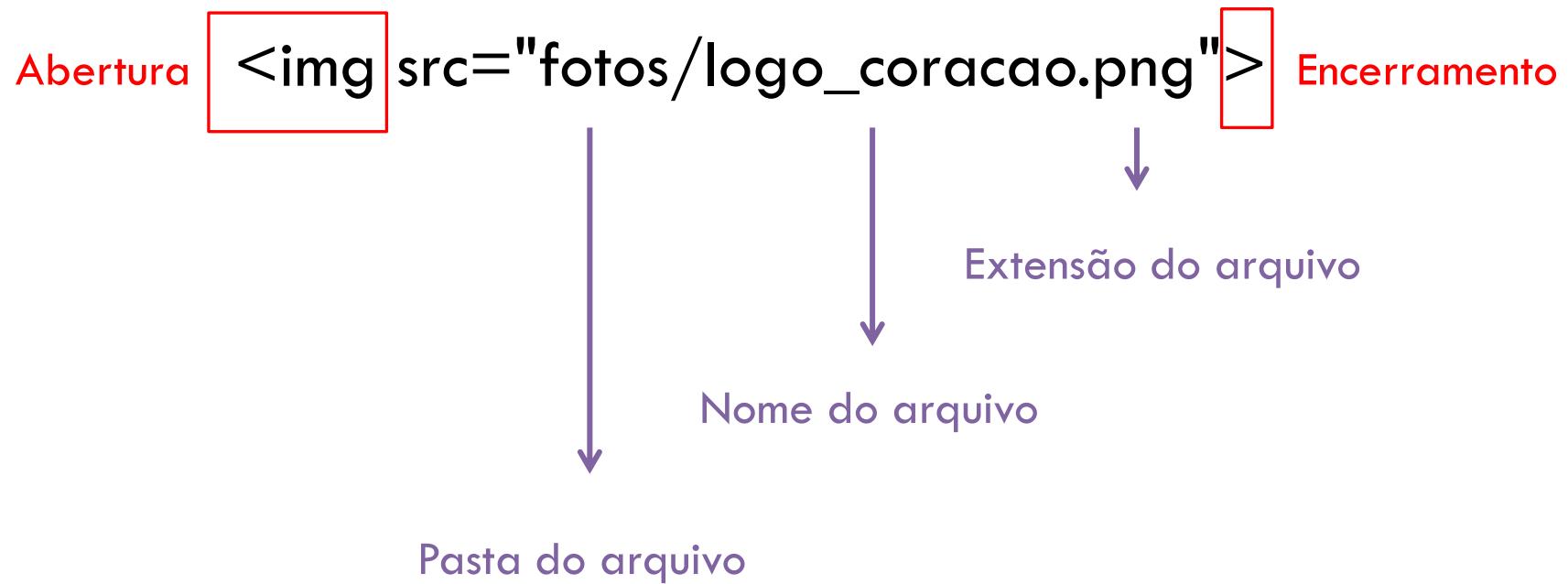
Com o arquivo de imagem no local correto e com as informações necessárias, vamos inclui-la em nossa página .html com o comando ``:

The image shows a screenshot of a computer interface. On the left is a dark-themed code editor for Visual Studio Code, displaying an HTML file named 'index.html'. The code contains standard HTML tags like `<html>`, `<head>`, and `<body>`, along with some CSS-like styles in the `<div>` tags. A specific line of code, ``, is highlighted with a red underline. To the right of the code editor is a web browser window showing the result of the code execution. The browser's title bar says 'index.html - htdocs - Visual Studio'. The main content area of the browser shows the text 'Olá Mundo!' and 'Div personalida!', followed by a large green rectangular area containing a blue heart-shaped logo.

```
File Edit Selection View Go Run Terminal Help
index.html - htdocs - Visual Studio
index.html X
aulas_iep > aula_01 > index.html > html
1  <html>
2
3      <!-- TRADUCAO DA PAGINA -->
4      <meta charset="UTF-8">
5
6      Olá Mundo!
7
8      <div style="height: 260px; width: 500px;
9          border: dashed 5px #00a6ff; background-color: lightcyan">
10
11         Div personalida!
12
13         <!-- Pular linha-->
14         <br><br>
15
16         <div style="height: 120px; width: 300px; background-color: lightgreen">
17
18             <!-- Inserindo uma imagem -->
19             
20
21         </div>
22
23     </div>
C:\xampp\htdocs\aulas_iep\au... Pesquisa...
Olá Mundo!
Div personalida!

```

Vamos observar os detalhes da **tag img**:



Note que dentro da **tag img** utilizamos o comando **src=""** que significa source (fonte), dentro das **aspas duplas ""** será informado o diretório do nosso arquivo de imagem.

Nota

Nesse exemplo utilizamos uma imagem com extensão **.png**, com esse tipo de imagem podemos inserir um fundo transparente, existem vários efeitos que podemos criar com imagens de fundo transparente em nossas páginas.

Na **tag img** dentro do **src=""** não precisamos passar todo o diretório do arquivo que no caso seria:

C:/xampp/htdocs/aulas\_iep/aula\_01/**fotos/logo\_coracao.png**

Pois a página que estamos chamando a imagem já consta no diretório:

C:/xampp/htdocs/aulas\_iep/aula\_01/

Então foi necessário informar apenas **fotos/logo\_coracao.png**

```
<!-- Inserindo uma imagem -->

```

Note que o **resultado** é o **mesmo** com a extensão do arquivo por completo:

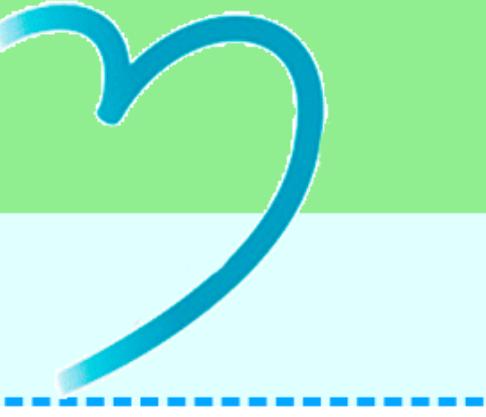
The image shows a comparison between the code editor and the browser output.

**Code Editor (Visual Studio Code):**

- File:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Project:** index.html - htdocs - Visual Studio
- Code:**

```
<!-- Arquivo index.html -->
<html>
    <head>
        <title>aulas_iep</title>
    </head>
    <body>
        <h1>Olá Mundo!</h1>
        <p>Div personalizada!</p>
        <!-- Pular linha-->
        <br><br>
        <div style="height: 120px; width: 300px; background-color: lightgreen">
            <!-- Inserindo uma imagem -->
            
        </div>
    </body>
</html>
```

**Browser Preview:**

- Address Bar:** C:\xampp\htdocs\aulas\_iep\al...
- Content:** Olá Mundo!  
Div personalizada!  


Para ajustarmos o tamanho da imagem podemos utilizar dentro da tag img o style="" para estilizarmos essa imagem (CSS) , então vamos aumentar um pouco o tamanho da div e redimensionar o tamanho da imagem:

The image shows a comparison between a code editor and a web browser. On the left, the Visual Studio Code interface displays the file 'index.html' with the following code:

```
File Edit Selection View Go Run Terminal Help
index.html - index.html - Visual Studio Code
index.html x
aulas_iep > aula_01 > index.html > html > div
1 <html>
2
3     <!-- TRADUCAO DA PAGINA -->
4     <meta charset="UTF-8">
5
6     Olá Mundo!
7
8     <div style="height: 260px; width: 500px;
9         border: dashed 5px #00a6ff; background-color: lightcyan">
10
11         Div personalida!
12
13         <!-- Pular linha-->
14         <br><br>
15
16         <div style="height: 200px; width: 300px; background-color: lightgreen">
17
18             <!-- Inserindo uma imagem -->
19             
20
21         </div>
22
23     </div>
24
25 </html>
```

The code defines a main 

with a height of 260px and a width of 500px, featuring a blue dashed border and a cyan background. Inside this, there's a smaller 

with a height of 200px and a width of 300px, colored lightgreen. An tag is placed within this green 

, pointing to a file named 'logo\_coracao.png'. The style attribute for the tag specifies a height of 200px and a width of 200px.

On the right, a screenshot of a web browser window titled 'index.html - htdocs - Visual Studio Code' shows the rendered HTML. The page contains the text 'Olá Mundo!' and 'Div personalida!', followed by a large blue heart logo on a green background, all contained within the specified div elements. The browser's address bar shows the local path 'C:\xampp\htdocs\aulas\_ie...'.

Note que abrimos a **tag img**, inserimos o **src=""** abrindo e fechando ele com o diretório da imagem, e depois abrimos o **style=""** separando os estilos com ponto e vírgula ; e depois fechamos a **tag img**.

```
<!-- Inserindo uma imagem -->

```

Podemos inserir várias **funcionalidades** dentro de uma **tag**, devemos informar a **funcionalidade** abrindo com **aspas duplas “ ”**, por exemplo **src=“ ”** e depois de inserir as informações dela e fechar com outra **aspas duplas “ ”**, por exemplo **src=“foto/foto.jpg”**, para depois utilizarmos outra funcionalidade após um espaço dentro da mesma **tag**, por exemplo **style=“background-color: yellow”**.

```
<!-- Exemplo tag img -->

```

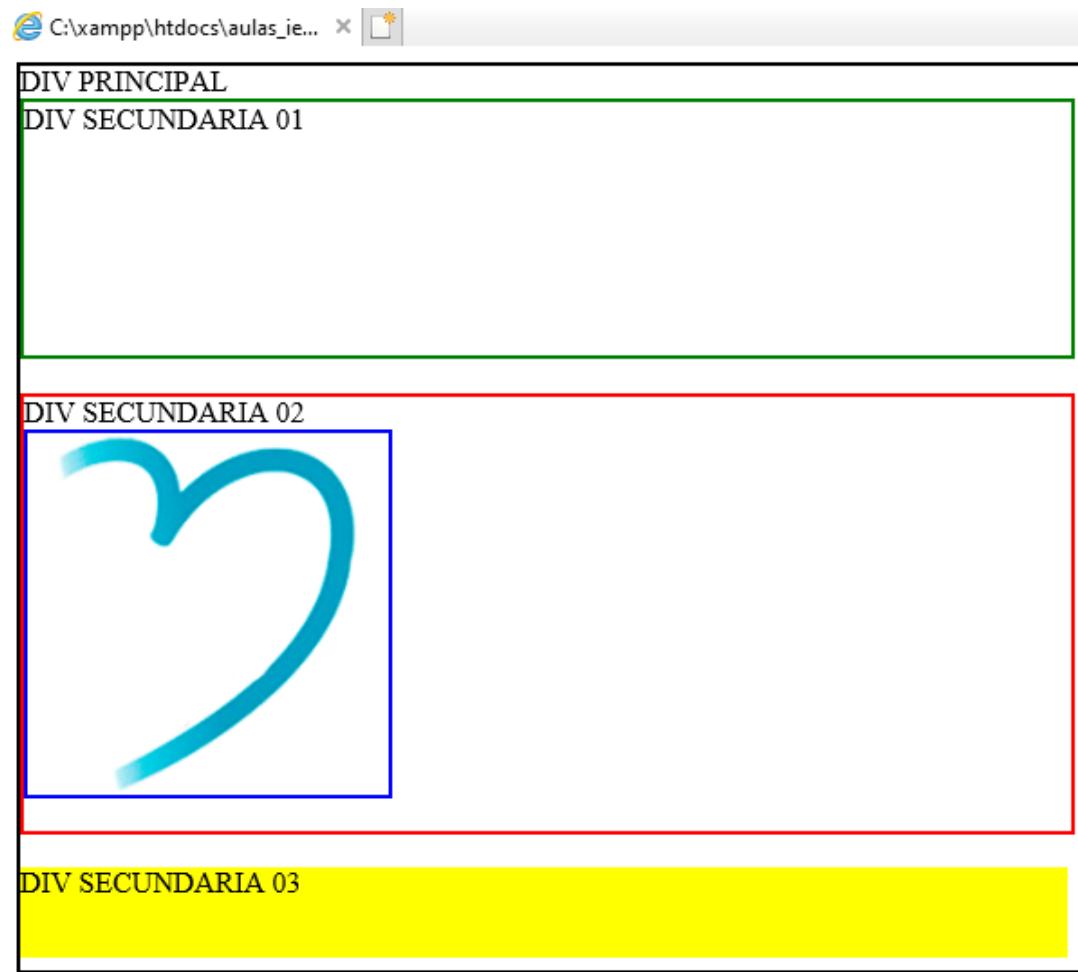


# EXERCÍCIO

HTML + CSS



Utilizando o que já aprendemos, monte uma página em .html conforme exemplo abaixo:

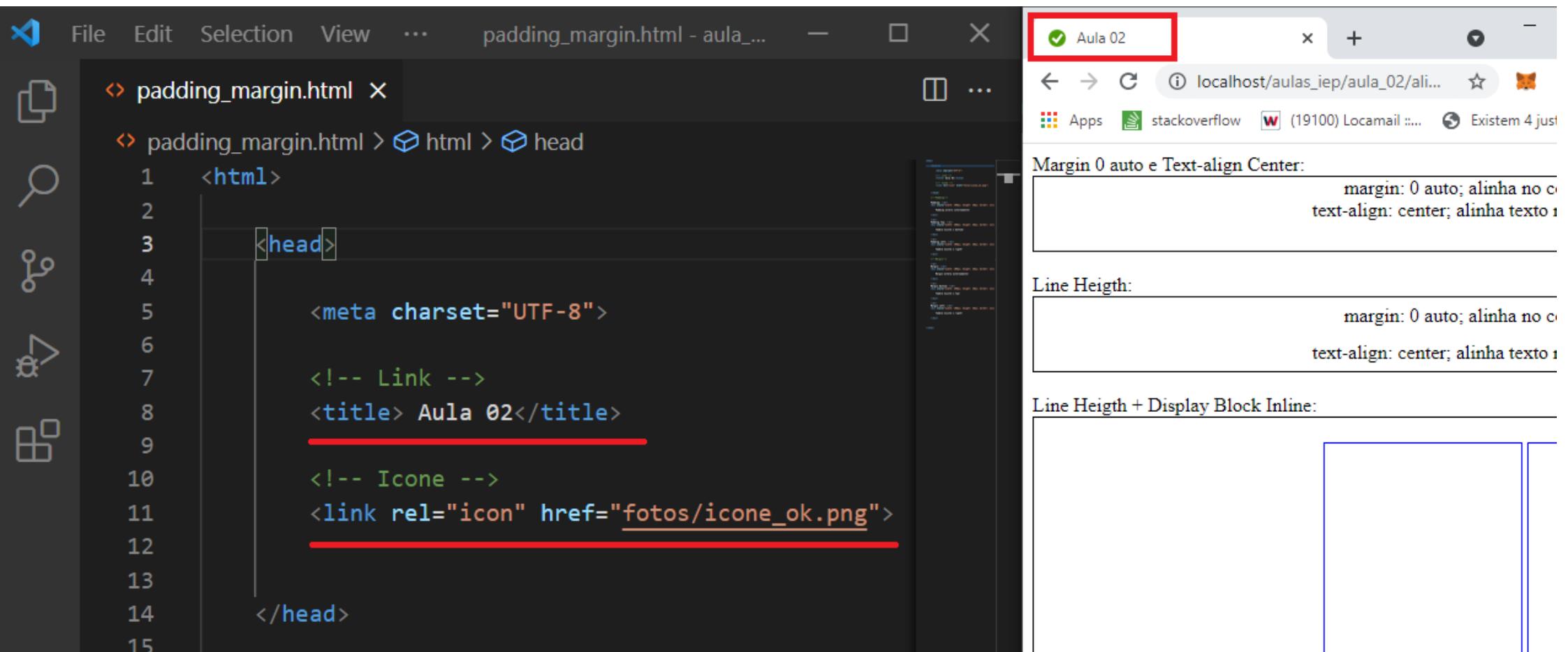


A imagem, as cores e o tamanho das divs ficam na escolha de cada aluno, o importante é entender o conceito das funcionalidades **HTML** e **CSS**.

# ORGANIZANDO OBJETOS

HTML + CSS

Antes de iniciarmos os alinhamentos dos objetos vamos inserir um título para a nossa página, para isso utilizamos a **tag title** e para inserir uma imagem no título utilizamos a **tag link**:



The screenshot shows a code editor and a browser side-by-side. The code editor displays the following HTML code:

```
<html>
<head>
    <meta charset="UTF-8">
    <!-- Link -->
    <title> Aula 02</title>
    <!-- Icone -->
    <link rel="icon" href="fotos/icon_ok.png">
</head>
```

The browser window shows the rendered page with the title "Aula 02" and a green checkmark icon. The browser's address bar shows "localhost/aulas\_iep/aula\_02/ali...".

Para darmos **espaçamentos internos** dentro de objetos utilizamos o **padding**:

```
13 </head>
14
15
16 <!--Padding-->
17
18 Padding: <br>
19 <div style="width: 200px; height: 20px;
20 border: solid 1px blueviolet; padding: 20px;">
21
22     Padding altera internamente!
23
24 </div>
25
26 <br>
27 Padding Top: <br>
28 <div style="width: 200px; height: 20px;
29 border: solid 1px blueviolet; padding-top: 20px;">
30
31     Também existe o bottom!
32
33 </div>
34
35 <br>
36 Padding Left: <br>
37 <div style="width: 200px; height: 20px;
38 border: solid 1px blueviolet; padding-left: 50px;">
39
40     Também existe o right!
41
42 </div>
```

Padding:

Padding altera internamente!

Padding Top:

Também existe o bottom!

Padding Left:

Também existe o right!

Margin:

Margin altera externamente!

Margin Bottom:

Também existe o top!

Margin Left:

Também existe o right!

Existem 5 comandos que podemos utilizar no **padding**, veja exemplos:

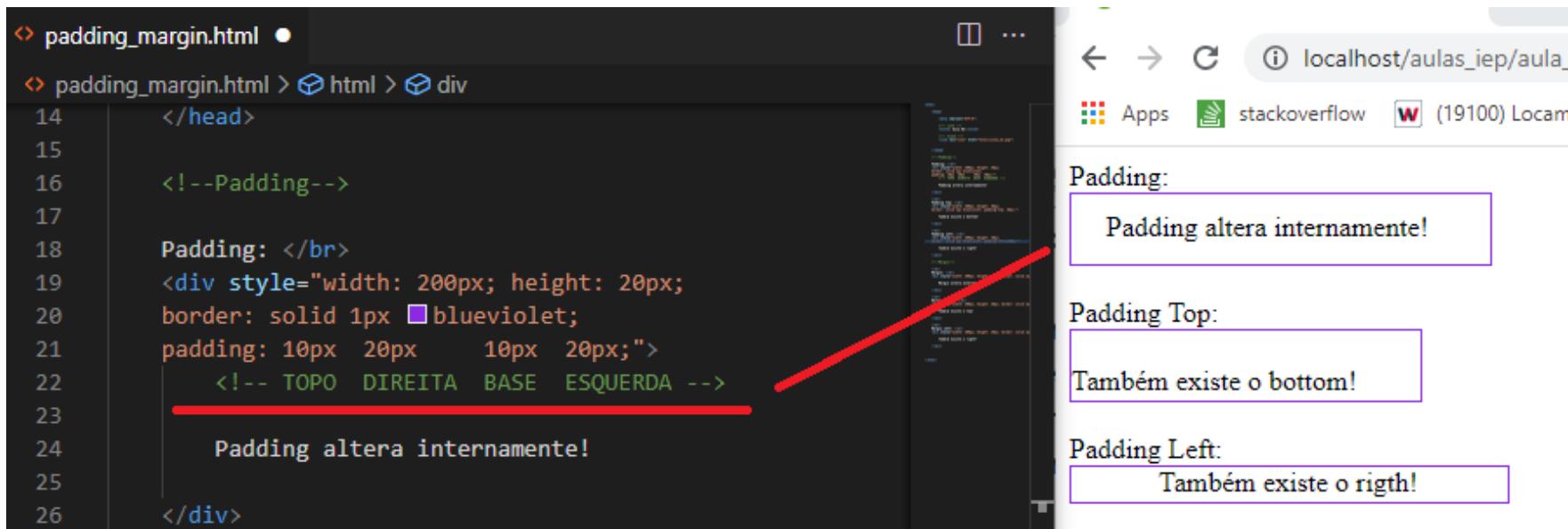
**padding-top** : Espaçamento interno apenas do **topo**;

**padding-right** : Espaçamento interno apenas da **direita**;

**padding-bottom** : Espaçamento interno apenas da **base**;

**padding-left** : Espaçamento interno apenas da **esquerda**;

Podemos utilizar todas as funcionalidades apenas com o **padding**, a direção é feita no **sentido horário**:



The screenshot shows a browser window displaying a local file named 'padding\_margin.html'. The code in the file is as follows:

```
14     </head>
15
16     <!--Padding-->
17
18     Padding: <br>
19     <div style="width: 200px; height: 20px;
20     border: solid 1px blueviolet;
21     padding: 10px 20px 10px 20px;">
22         <!-- TOPO DIREITA BASE ESQUERDA -->
23
24         Padding altera internamente!
25
26     </div>
```

A red arrow points from the line 'Padding altera internamente!' in the code to the text 'Padding altera internamente!' in the browser's right-hand panel. The browser panel also contains three other sections: 'Padding Top:' with 'Também existe o bottom!', 'Padding Left:' with 'Também existe o right!', and a navigation bar at the top.

Para mais exemplos: [https://www.w3schools.com/css/css\\_padding.asp](https://www.w3schools.com/css/css_padding.asp)

Para darmos **espaçamentos externos** dentro de objetos utilizamos o **margin**:

The screenshot shows a code editor with a dark theme. On the left, there's a vertical toolbar with icons for file operations like Open, Save, and Find. The main area contains the following CSS code:

```
45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
```

Margin altera externamente!

```
<!--Margin-->
<br>
Margin: <br>
<div style="width: 200px; height: 20px;
border: solid 1px green; margin: 20px;">
    Margin altera externamente!
</div>
```

Também existe o top!

```
<br>
Margin Bottom: <br>
<div style="width: 200px; height: 20px;
border: solid 1px green; margin-bottom: 20px;">
    Também existe o top!
</div>
```

Também existe o rigth!

```
<br>
Margin Left: <br>
<div style="width: 200px; height: 20px;
border: solid 1px green; margin-left: 50px;">
    Também existe o rigth!
</div>
```

On the right side of the slide, there are several callout boxes with arrows pointing to specific parts of the code or the resulting visual effect:

- A purple box labeled "Padding" points to the margin declaration in the first example.
- A purple box labeled "Padding altera internamente!" points to the inner padding area of the first div.
- A purple box labeled "Padding Top:" points to the margin-bottom declaration in the second example.
- A purple box labeled "Também existe o bottom!" points to the bottom margin of the second div.
- A purple box labeled "Padding Left:" points to the margin-left declaration in the third example.
- A purple box labeled "Também existe o rigth!" points to the right margin of the third div.
- A green box labeled "Margin:" points to the margin declaration in the first example.
- A green box labeled "Margin altera externamente!" points to the outer margin space of the first div.
- A green box labeled "Margin Bottom:" points to the margin-bottom declaration in the second example.
- A green box labeled "Também existe o top!" points to the top margin of the second div.
- A green box labeled "Margin Left:" points to the margin-left declaration in the third example.
- A green box labeled "Também existe o rigth!" points to the right margin of the third div.

O **margin** segue as **mesmas regras** do **padding**, para mais exemplos:

[https://www.w3schools.com/css/css\\_margin.asp](https://www.w3schools.com/css/css_margin.asp)

Para alinharmos horizontalmente um objeto no centro da página ou no centro de outros objetos utilizamos o **margin: 0 auto** e para alinharmos um texto horizontalmente utilizamos o **text-align: center**:

The screenshot shows a Visual Studio Code editor on the left and a web browser window on the right. The code editor displays an HTML file named 'alinhamento.html' with the following content:

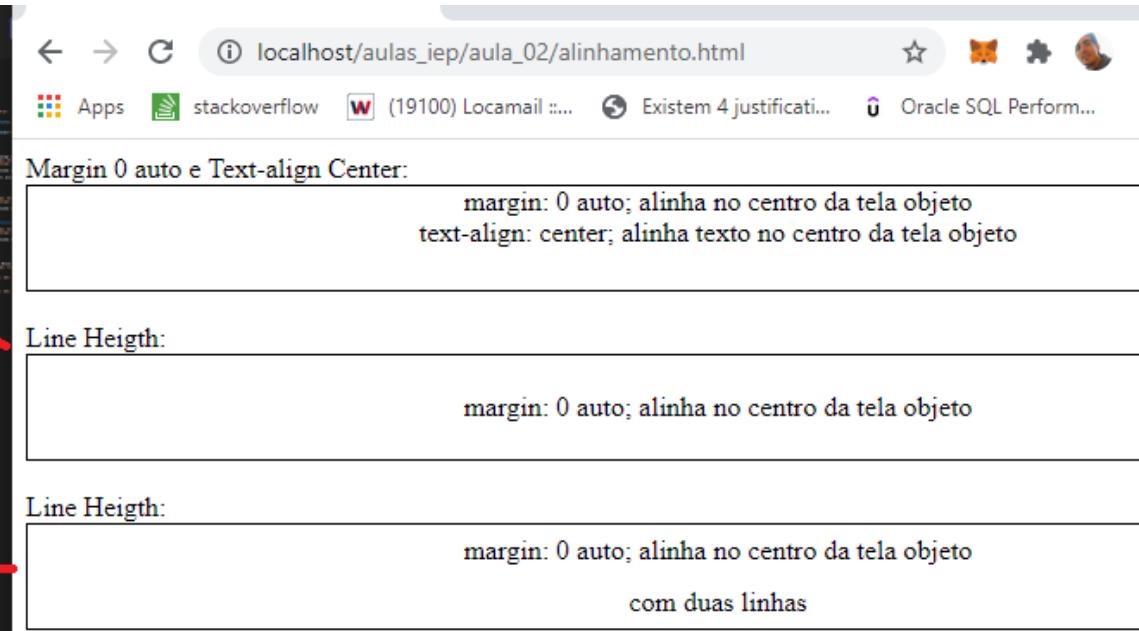
```
<head>
</head>
<!--Alinhamento-->
Margin 0 auto e Text-align Center:
<div style="width:800px; height: 60px; border: solid 1px black; margin: 0 auto; text-align: center;">
    margin: 0 auto; alinha no centro da tela objeto
    <br>
    text-align: center; alinha texto no centro da tela objeto
</div>
```

The browser window shows the rendered HTML. It has three sections labeled 'Margin 0 auto e Text-align Center:', 'Line Height:', and 'Line Height + Display Block Inline:'. The first section contains the text 'margin: 0 auto; alinha no centro da tela objeto' and 'text-align: center; alinha texto no centro da tela objeto'. The second section contains the text 'margin: 0 auto; alinha no centro da tela objeto'. The third section is currently empty.

Existem também a funcionalidade de **text-align: left;** e **text-align: right;** que alinham da **esquerda** e **direita respectivamente.**

Para alinhar um texto verticalmente dentro de um objeto utilizamos o **line-height**, informando exatamente a altura do objeto (**height**):

```
<br>
27
28    </br>
29    Line Height:
30    <div style="width:800px; height: 60px; border: solid 1px black;
31    margin: 0 auto; text-align: center; line-height: 60px;">
32
33        margin: 0 auto; alinha no centro da tela objeto
34
35    </div>
36
37    <br>
38    Line Height:
39    <div style="width:800px; height: 60px; border: solid 1px black;
40    margin: 0 auto; text-align: center; line-height: 30px;">
41
42        margin: 0 auto; alinha no centro da tela objeto
43
44    <br>
```



Quando existe mais de uma linha de texto, dividimos o **line-height** pela quantidade de linhas de texto.

Lembrando que existem **outras formas** de alinhar objetos verticalmente, que serão explicadas posteriormente.



# EXERCÍCIO

HTML + CSS



**Dica importante**, para não se perder no código e ser mais claro quando uma div precisa estar dentro de outra comentamos onde a div se inicia e ela termina, veja exemplo:

The image shows a code editor on the left and a web browser on the right. The code editor displays the following HTML code:

```
<!--Inicio da div Blue-->
<div style="width: 600px; height: 400px;
border: solid 3px blue; padding: 20px;">

    <!--Inicio da div Green-->
    <div style="width: 400px; height: 300px;
border: solid 3px green; padding: 20px;">

        <!--Inicio da div Red-->
        <div style="width: 200px; height: 200px;
border: solid 3px red; padding: 20px;">

            </div>
        <!--Fim da div Red-->

    </div>
    <!--Fim da div Green-->

</div>
<!--Fim da div Blue-->
```

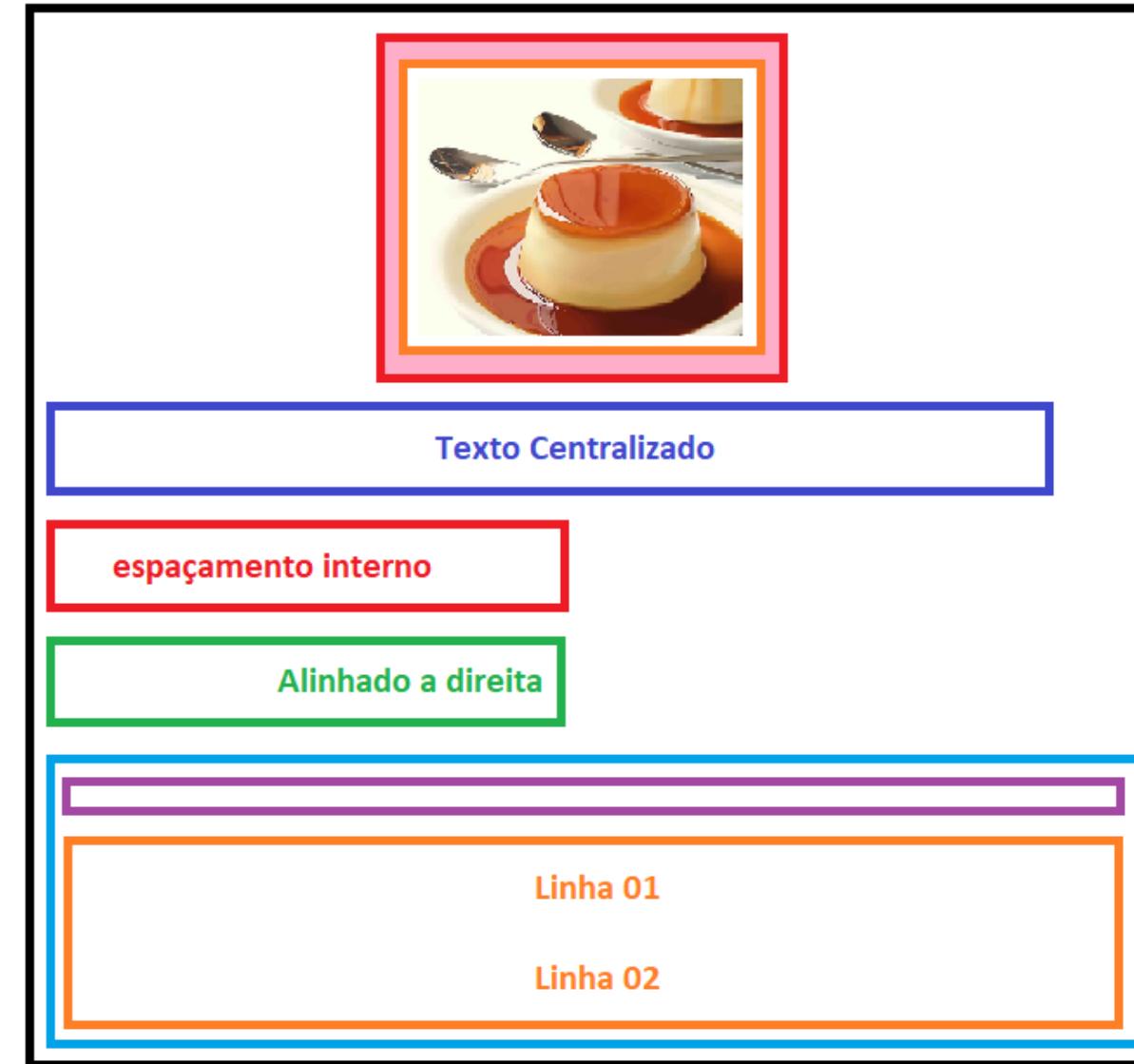
The browser window on the right shows the rendered HTML structure. It consists of three nested divs:

- An outermost div with a blue border (600px width, 400px height).
- A middle div with a green border (400px width, 300px height) centered within the blue div.
- An innermost div with a red border (200px width, 200px height) centered within the green div.

The browser's address bar shows the URL: `localhost/aulas_iep/aula_02/dica_importante_div.html`.

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Utilizando o que já aprendemos, monte uma página em [.html](#) conforme exemplo abaixo. Seja criativo com as cores, e não esqueça da indentação, do título da página com imagem e das margins de distanciamento!



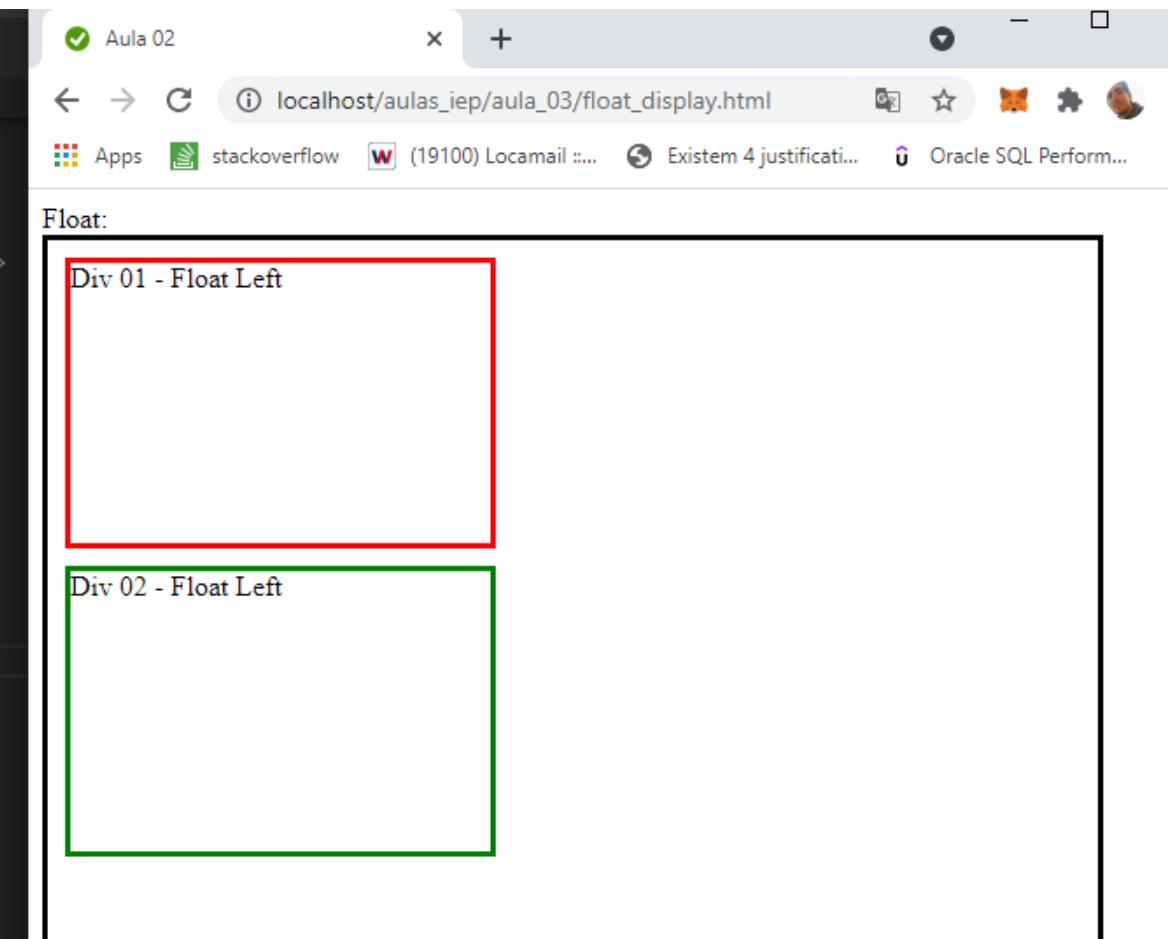


# FLOAT

HTML + CSS

Para inserir uma div ao lado da outra utilizamos a funcionalidade CSS **Float**, abaixo vemos uma div pai (div black) e duas divs filhas (div red e div green), **sem o Float** elas ficam organizadas dessa maneira:

```
float_display.html x
aulas_iep > aula_03 > float_display.html > html > div > div
15
16     Float:
17
18     <!--INICIO DIV BLACK-->
19     <div style="width:600px; height: 400px; border: solid 3px black;">
20
21         <!--INICIO DIV RED-->
22         <div style="width: 40%; height: 40%; border: solid 3px red; margin:10px;">
23
24             Div 01 - Float Left
25
26         </div>
27         <!--FIM DIV RED-->
28
29         <!--INICIO DIV GREEN-->
30         <div style="width: 40%; height: 40%; border: solid 3px green; margin:10px;">
31
32             Div 02 - Float Left
33
34         </div>
35         <!--FIM DIV GREEN-->
36
37     </div>
38     <!--FIM DIV BLACK-->
39
40
41
42
```



Mas ao adicionarmos o **Float: Left;** nas divs filhas elas se alinham na horizontal sempre iniciando da esquerda para a direita:

The image shows a comparison between the code editor and the browser output. On the left, the code editor displays the HTML file `float_display.html`. The code defines a large black container div and two smaller child divs (red and green) that are floated to the left. On the right, a browser window titled "Aula 02" shows the rendered result. The red div is on the left and the green div is on the right, demonstrating the left-to-right flow of floated elements.

```
File Edit Selection View Go Run Terminal Help float_display.html - htdocs - Visual Studio ...
File Edit Selection View Go Run Terminal Help float_display.html - htdocs - Visual Studio ...
aulas_iep > aula_03 > float_display.html > html > div > div
15
16     Float:
17
18     <!--INICIO DIV BLACK-->
19     <div style="width:600px; height: 400px; border: solid 3px black;">
20
21         <!--INICIO DIV RED-->
22         <div style="width: 40%; height: 40%; float: left;
23         border: solid 3px red; margin:10px;">
24
25             Div 01 - Float Left
26
27         </div>
28         <!--FIM DIV RED-->
29
30         <!--INICIO DIV GREEN-->
31         <div style="width: 40%; height: 40%; float: left;
32         border: solid 3px green; margin:10px;">
33
34             Div 02 - Float Left
35
36         </div>
37         <!--FIM DIV GREEN-->
38
39     </div>
40     <!--FIM DIV BLACK-->
41
42
```

Aula 02

localhost/aulas\_iep/aula\_03/float\_display.html

Float:

Div 01 - Float Left

Div 02 - Float Left

Para iniciar a regra da horizontal pela direita utilizamos o **Float: Right;**:

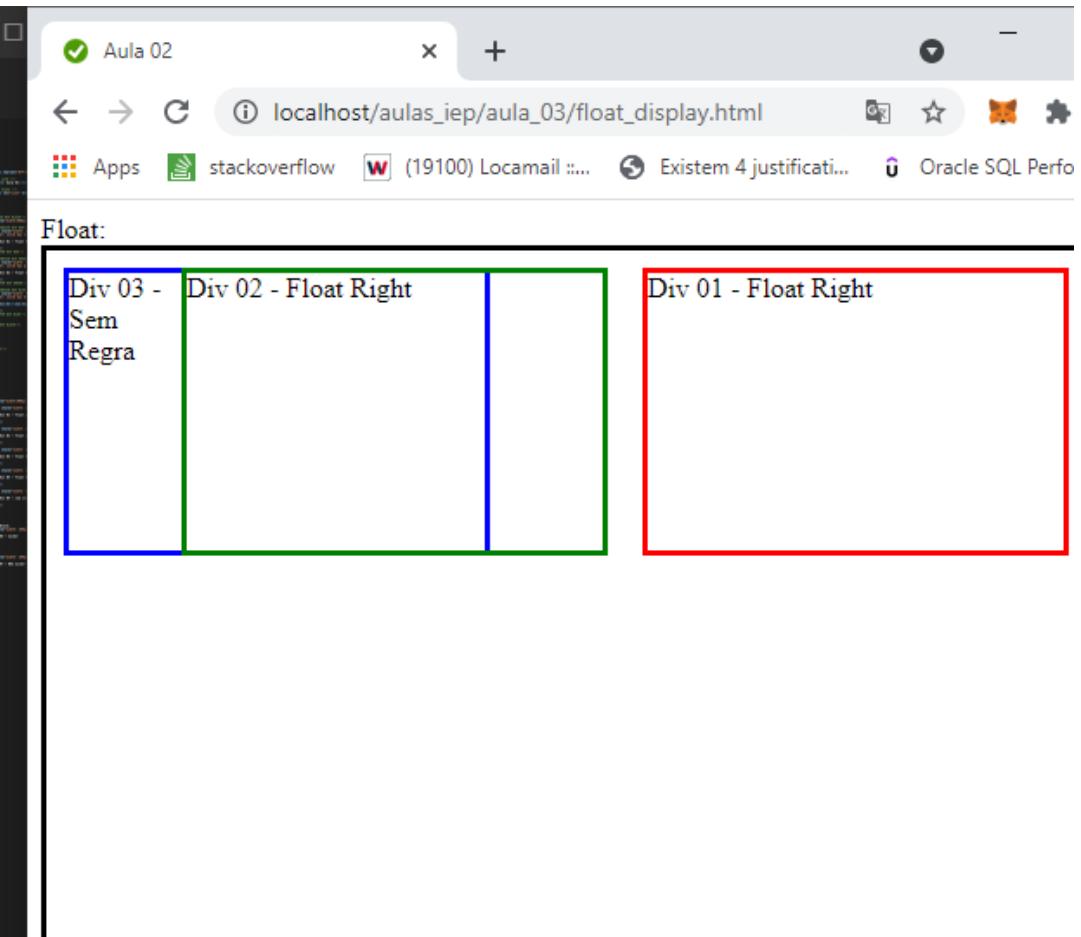
The image shows a screenshot of a development environment. On the left is a code editor window titled "float\_display.html" showing the following HTML and CSS code:

```
15
16     Float:
17
18     <!--INICIO DIV BLACK-->
19     <div style="width:600px; height: 400px; border: solid 3px black;">
20
21         <!--INICIO DIV RED-->
22         <div style="width: 40%; height: 40%; float: right;
23         border: solid 3px red; margin:10px;">
24
25             Div 01 - Float Right
26
27         </div>
28         <!--FIM DIV RED-->
29
30         <!--INICIO DIV GREEN-->
31         <div style="width: 40%; height: 40%; float: right;
32         border: solid 3px green; margin:10px;">
33
34             Div 02 - Float Right
35
36         </div>
37         <!--FIM DIV GREEN-->
38
39     </div>
40     <!--FIM DIV BLACK-->
```

On the right is a browser window titled "Aula 02" showing the rendered output. It displays a large black rectangular container labeled "Float:". Inside, there are two floating red boxes: one on the right labeled "Div 01 - Float Right" and one on the left labeled "Div 02 - Float Right".

Se desejarmos inserir uma div abaixo das filhas mas **sem a regra do float**, veja o que acontece:

```
File Edit Selection View Go Run Terminal Help float_display.html - htdocs - Visual Studio ...
float_display.html X
aulas_iep > aula_03 > float_display.html > html > div > div
23   border: solid 3px red; margin:10px;">
24
25     Div 01 - Float Right
26
27   </div>
28   <!--FIM DIV RED-->
29
30   <!--INICIO DIV GREEN-->
31   <div style="width: 40%; height: 40%; float: right;
32   border: solid 3px green; margin:10px;">
33
34     Div 02 - Float Right
35
36   </div>
37   <!--FIM DIV GREEN-->
38
39   <!--INICIO DIV BLUE-->
40   <div style="width: 40%; height: 40%;
41   border: solid 3px blue; margin:10px;">
42
43     Div 03 - Sem Regra
44
45   </div>
46   <!--FIM DIV BLUE-->
47
48 </div>
49 <!--FIM DIV BLACK-->
50
51
52
```



Para resolvemos esse problema utilizamos o **Clear: Both;**:

The image shows a development environment with two main windows. On the left is a code editor in Visual Studio Code displaying the file `float_display.html`. The code contains several `<div>` elements with different styles and floating properties. On the right is a web browser window titled "Aula 02" showing the rendered HTML. The browser window has a tab bar with "localhost/aulas\_iep/aula\_03/float\_display.html". Below the tabs, there are several icons for "Apps", "stackoverflow", "Locamail", "Existem 4 justificati...", and "Oracle SQL Perform". The rendered content in the browser shows three boxes: "Div 02 - Float Right" (green border), "Div 01 - Float Right" (red border), and "Div 03 - Sem Regra" (blue border). The red border of Div 01 overlaps the green border of Div 02, illustrating the issue of floating elements not respecting each other's vertical space.

```
File Edit Selection View Go Run Terminal Help float_display.html - htdocs - Visual Studio ... aulas_iep > aula_03 > float_display.html > html > div > div
23     border: solid 3px red; margin:10px; >
24
25     Div 01 - Float Right
26
27     </div>
28     <!--FIM DIV RED-->
29
30     <!--INICIO DIV GREEN-->
31     <div style="width: 40%; height: 40%; float: right;
32     border: solid 3px green; margin:10px;">
33
34         Div 02 - Float Right
35
36     </div>
37     <!--FIM DIV GREEN-->
38
39     <!--INICIO DIV BLUE-->
40     <div style="width: 40%; height: 40%; clear: both;
41     border: solid 3px blue; margin:10px;">
42
43         Div 03 - Sem Regra
44
45     </div>
46     <!--FIM DIV BLUE-->
47
48     </div>
49     <!--FIM DIV BLACK-->
50
```

Também podemos inserir arredondamento nas bordas das divs com o **border-radius** e alterar a cor dos textos das divs com o **color**:

The image shows a split-screen view. On the left is a screenshot of Visual Studio Code with the file `float_display.html` open. The code defines a black container `<div>` with a width of 600px and height of 400px, containing two floating red `<div>` elements (Div 01 and Div 02) and one green `<div>` element. The red divs have a border radius of 15px and a white background color. The green div has a white background color and a black border radius of 15px. On the right is a screenshot of a web browser window titled "Aula 02" showing the rendered HTML. The browser's address bar indicates the URL is `localhost/aulas_iep/aula_03/float_display.html`. The page title is "Float:". It displays three floating right-hand `<div>` elements: "Div 02 - Float Right" (green, rounded corners), "Div 01 - Float Right" (red, sharp corners), and "Div 03 - Float Right" (green, sharp corners).

```
File Edit Selection View Go Run Terminal Help float_display.html - htdocs - Visual Studio ...
aulas_iep > aula_03 > float_display.html > html > div > div
13
14     </head>
15
16     Float:
17
18     <!--INICIO DIV BLACK-->
19     <div style="width:600px; height: 400px; border: solid 3px black;">
20
21         <!--INICIO DIV RED-->
22         <div style="width: 40%; height: 40%; float: right; border: solid 3px red; margin:10px; border-radius:15px;">
23             Div 01 - Float Right
24         </div>
25         <!--FIM DIV RED-->
26
27         <!--INICIO DIV GREEN-->
28         <div style="width: 40%; height: 40%; float: right; background-color: green; margin:10px; color: white; border-radius: 15px 0px 15px 0px;">
29             Div 02 - Float Right
30         </div>
31         <!--FIM DIV GREEN-->
32
33     </div>
34     <!--FIM DIV BLACK-->
35
36
37
38
39
40
41
42
43
44
45
```

Border-Radius: [https://www.w3schools.com/cssref/css3\\_pr\\_border-radius.asp](https://www.w3schools.com/cssref/css3_pr_border-radius.asp)

Color: [https://www.w3schools.com/html/html\\_colors.asp](https://www.w3schools.com/html/html_colors.asp)

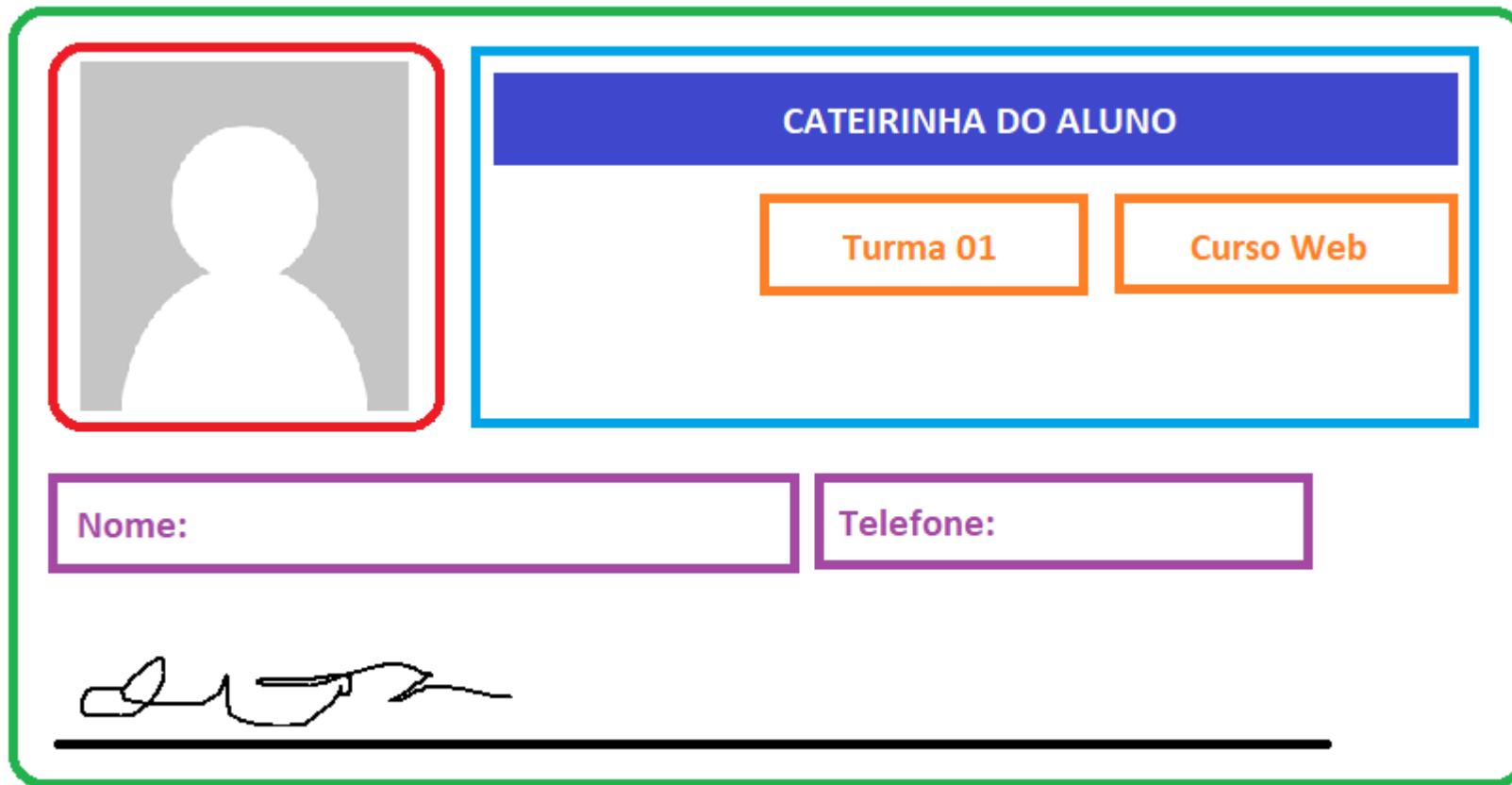


# EXERCÍCIO

HTML + CSS



Utilizando o que já aprendemos, monte uma página em .html conforme exemplo abaixo.



Dica: Para pintar apenas a borda de baixo da div utilizamos o **border-bottom**:

[https://www.w3schools.com/cssref/pr\\_border-bottom.asp](https://www.w3schools.com/cssref/pr_border-bottom.asp)

Não esqueça de subir os arquivos no **github**: <https://www.youtube.com/watch?v=Fj3gtbaF8WA>



# ARQUIVO CSS

HTML + CSS

Ao invés de abrirmos a **tag style** e passarmos todos os atributos para nossos objetos no próprio objeto como a **div** por exemplo, podemos criar um arquivo com a extensão **.css** e realizar todos os estilos dos objetos em uma **classe**, na qual essa **classe** será passada para todos os objetos que desejamos:

The screenshot shows a Visual Studio Code interface with two files open: `style.css` and `index.html`, and a browser preview window.

**style.css:**

```
# style.css
aulas_iep > aula_04 > # style.css > .div_estilizada
1 .div_estilizada{
2
3   width: 300px;
4   height: 100px;
5   line-height: 100px;
6   text-align: center;
7   border-top: solid 2px black;
8   border-bottom: solid 2px black;
9
10 }
```

**index.html:**

```
index.html - index.html - visual Studio Code
index.html
aulas_iep > aula_04 > index.html > html
1 <html>
2
3   <!--VINCULAR HTML COM CSS-->
4   <link rel="stylesheet" href="style.css">
5
6   <!--DIV COM A CLASSE CSS-->
7   <div class="div_estilizada">
8
9     Caixa CSS
10
11   </div>
12
13 </html>
```

**Browser Preview:**

The browser preview shows a single `div` element with the class `div_estilizada`. The `div` has a width of 300px, a height of 100px, and a line-height of 100px. It is centered and has a solid black border of 2px at the top and bottom. The text "Caixa CSS" is centered inside the `div`.

Para criarmos uma tabela utilizamos a **tag table**, para criar uma linha utilizando a **tag tr**, para coluna de titulo utilizamos a **tag th** e para coluna de resultado utilizamos a **tag td**, e todas elas podem ser individualmente estilizadas no nosso arquivo **.css**, lembrando que para **classes do sistema não utilizamos o . (ponto)** apenas utilizamos o **.** (ponto) nas classes personalizadas:

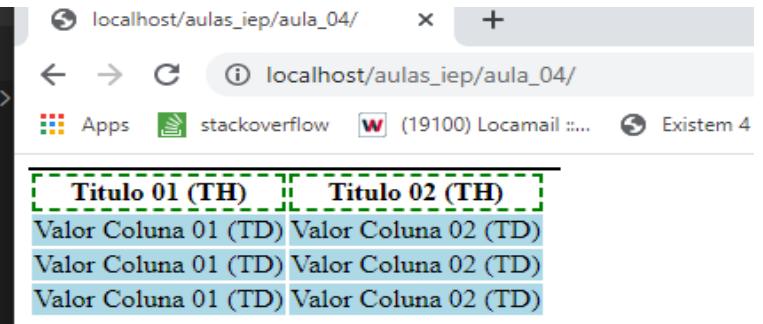
The screenshot shows a code editor with two tabs: style.css and index.html.

**style.css:**

```
# style.css  X
aulas_iep > aula_04 > # style.css > td
1   .div_estilizada{
2     width: 300px;
3     height: 100px;
4     line-height: 100px;
5     text-align: center;
6     border-top: solid 2px □black;
7     border-bottom: solid 2px □black;
8
9
10 }
11
12 /*TITULO*/
13 th{
14   border: dashed 2px □green;
15 }
16
17
18 /*VALORES*/
19 td{
20   background-color: □lightblue;
21 }
22
23
24 }
```

**index.html:**

```
aulas_iep > aula_04 > index.html > html > div.div_estilizada >
1   <html>
2     <!--VINCULAR HTML COM CSS-->
3     <link rel="stylesheet" href="style.css">
4     <!--DIV COM A CLASSE CSS-->
5     <div class="div_estilizada">
6
7       <table>
8
9         <tr>
10        <th>Titulo 01 (TH)</th>
11        <th>Titulo 02 (TH)</th>
12      </tr>
13
14      <tr>
15        <td>Valor Coluna 01 (TD)</td>
16        <td>Valor Coluna 02 (TD)</td>
17      </tr>
18
19      <tr>
20        <td>Valor Coluna 01 (TD)</td>
21        <td>Valor Coluna 02 (TD)</td>
22      </tr>
23
24      <tr>
25        <td>Valor Coluna 01 (TD)</td>
26        <td>Valor Coluna 02 (TD)</td>
27      </tr>
28
29    </table>
30
31  </div>
32 </html>
```





# EXERCÍCIO

HTML + CSS



Utilizando o que já aprendemos, monte uma página em `.html` e `.css` conforme exemplo abaixo.

TIULO 01	TITULO 02	TITULO 03
CONTEUDO 01	CONTEUDO 02	CONTEUDO 03
<input type="radio"/>	<input type="text"/>	<input type="checkbox"/>



# FORMS E FONT AWESOME

HTML + CSS

Utilizamos a **tag form** para poder receber e passar informações para os usuários que utilizam nossos sites, existem diversos **inputs types** para diferenciar o tipo de entrada, veja alguns:

The screenshot shows a browser window with a sidebar containing code snippets and a main panel showing the resulting form fields.

**Code Snippet:**

```
aula_05 > index.html > html > form > input
<!--VINCULAR HTML COM CSS-->
<link rel="stylesheet" href="style.css">

<form>

    Text <br>
    <input type="text">

    <br><br> Password <br>
    <input type="password">

    <br><br> Number <br>
    <input type="number">

    <br><br> Email <br>
    <input type="email">

    <br><br> Date <br>
    <input type="date">

    <br><br> Checkbox <br>
    <input type="checkbox">

    <br><br> File <br>
    <input type="file">

    <br><br> Submit <br>
    <input type="submit">

</form>
```

**Resulting Form Fields:**

- Text
- Password
- Number
- Email
- Date
- Checkbox
- File
- Submit

Dentro da **tag input** temos alguns comando que podemos utilizar além do **class** e o **style** que já conhecemos, um dos comandos mais usados para estilizar nosso projeto se chama **placeholder**, veja:

The image shows a split-screen view. On the left is a code editor in Visual Studio Code displaying an HTML file named 'index.html'. The code includes various input types with placeholder attributes, such as 'text', 'password', 'number', 'email', 'date', 'checkbox', 'file', and 'submit'. Lines 9, 15, and 18 are highlighted with red boxes. On the right is a browser window showing the rendered HTML. The 'Text' input has 'Insira um texto' placeholder. The 'Number' input has '(xx) xxxxx-xxxx' placeholder. The 'Email' input has 'seu-email@iep.com.br' placeholder. The 'Date' input has 'dd/mm/aaaa' placeholder. The 'File' input has 'Escolher arquivo Nenhum arquivo selecionado' placeholder. The 'Submit' button has 'Enviar' label.

```
<!-- VINCULAR HTML COM CSS -->
<link rel="stylesheet" href="style.css">

<form>
    Text <br>
    <input type="text" placeholder="Insira um texto">

    <br><br> Password <br>
    <input type="password">

    <br><br> Number <br>
    <input type="number" placeholder="(xx) xxxxx-xxxx">

    <br><br> Email <br>
    <input type="email" placeholder="seu-email@iep.com.br">

    <br><br> Date <br>
    <input type="date">

    <br><br> Checkbox <br>
    <input type="checkbox">

    <br><br> File <br>
    <input type="file">

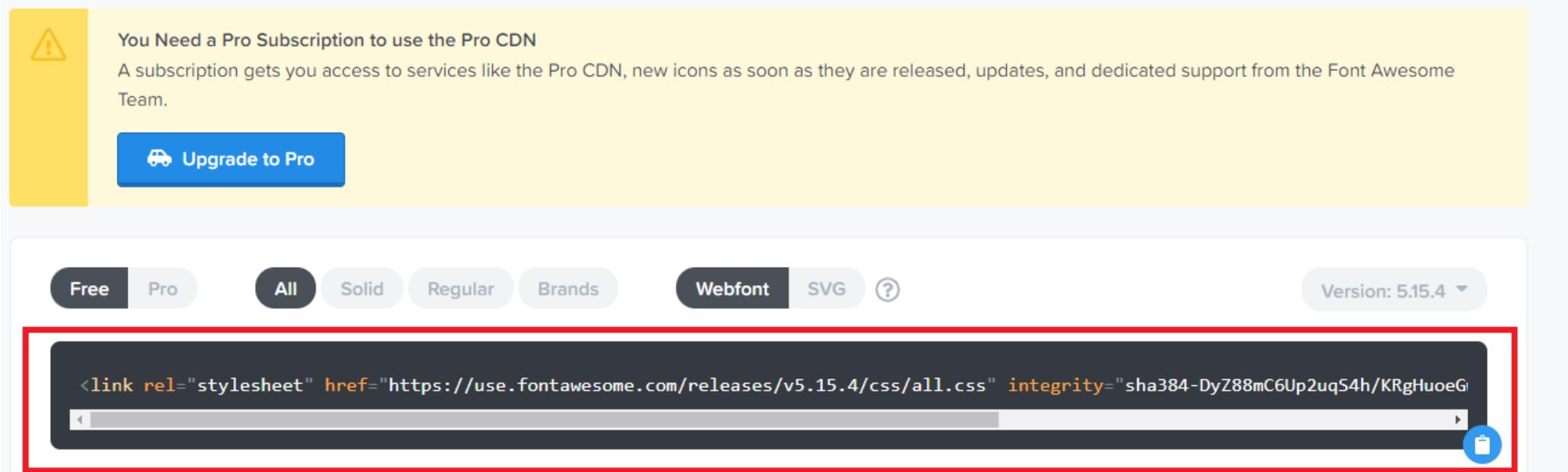
    <br><br> Submit <br>
    <input type="submit">
</form>
```

Obs.: Outros exemplos e mais detalhes serão vistos no modulo de **bootstrap**

Para estilizarmos ainda mais nosso site, utilizamos o **Font Awesome**, se trata de uma plataforma que oferece muitos ícones gratuitamente, e podemos utiliza-los com simples comando, o primeiro passo é se cadastrar na plataforma, feito isso será disponibilizado um código para ser utilizado em nosso site:

## Font Awesome CDN

Font Awesome's original CDN is still around and available for existing users.



You Need a Pro Subscription to use the Pro CDN  
A subscription gets you access to services like the Pro CDN, new icons as soon as they are released, updates, and dedicated support from the Font Awesome Team.

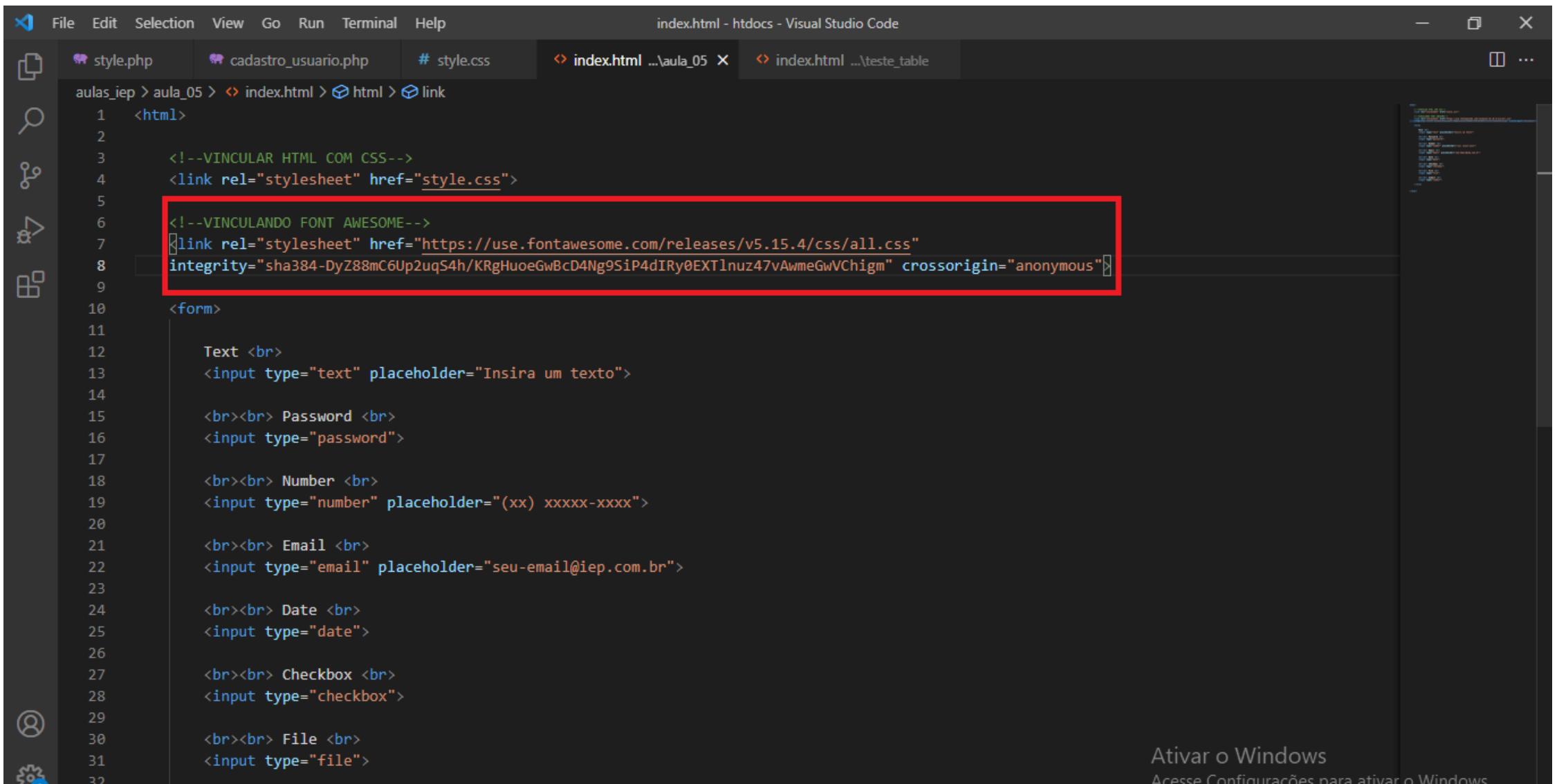
 Upgrade to Pro

Free Pro All Solid Regular Brands Webfont SVG ? Version: 5.15.4

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css" integrity="sha384-DyZ88mC6Up2uqS4h/KRgHuoeG" data-bbox="154 708 904 728"/>
```

Site: <https://fontawesome.com/>

## Veja o Font Awesome integrado:



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** index.html - htdocs - Visual Studio Code
- Left Sidebar:** Shows files: style.php, cadastro\_usuario.php, # style.css, index.html (active), and index.html (test\_table).
- Code Editor:** Displays an HTML form with various input fields. A red box highlights the following line of code:

```
<!--VINCULANDO FONT AWESOME-->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css"
integrity="sha384-DyZ88mC6Up2uq54h/KRgHuoeGwBcD4Ng9SiP4dIRy0EXTlnuz47vAwmeGwVChigm" crossorigin="anonymous" />
```
- Bottom Right Corner:** Ativar o Windows, Acesse Configurações para ativar o Windows.

Para utilizar um ícone basta pesquisar os ícones que são gratuitos e copiar a **tag i** na descrição do ícone:

The screenshot shows the Font Awesome website's "Icons" section. At the top, there is a navigation bar with links for "Start", "Icons", "Docs", "Support", "Upgrade", and "Blog". A search bar with the placeholder "Search icons..." and a magnifying glass icon is also present. Below the navigation, a banner announces "Font Awesome 6's beta release is out now!" and encourages users to "Subscribe to [Font Awesome Pro](#) and get instant access to the v6 beta." On the right side of the banner is a link to "Explore the Icons".

The main content area features a large heading "users" with a subtitle "Solid Style (fas)". Below this, a series of icons are displayed, each with its corresponding class name: "fas fa-users". One of these class names, "`<i class="fas fa-users"></i>`", is highlighted with a red rectangular box.

On the left, there is a decorative graphic consisting of a grid of squares with green semi-circles at the bottom. On the right, there is a graphic of several grey user icons.

At the bottom right of the page, there is a link to "Social" and a note indicating "Updated: Version 5.0.11".

## Veja resultado com o **Font Awesome**:

```
section View Go Run Terminal Help
cadastro_usuario.php # style.css index.html - h
aula_05 > index.html > html > form

<form>

    <i class="far fa-user"></i> Text <br>
    <input type="text" placeholder="Insira um texto">

    <br><br> <i class="fas fa-key"></i> Password <br>
    <input type="password">

    <br><br> <i class="fas fa-phone"></i> Number <br>
    <input type="number" placeholder="(xx) xxxxx-xxxx">

    <br><br> <i class="far fa-envelope-open"></i> Email <br>
    <input type="email" placeholder="seu-email@iep.com.br">

    <br><br> <i class="far fa-clock"></i> Date <br>
    <input type="date">

    <br><br> <i class="far fa-check-square"></i> Checkbox <br>
    <input type="checkbox">

    <br><br> <i class="far fa-file-excel"></i> File <br>
    <input type="file">

    <br><br><i class="far fa-paper-plane"></i> Submit <br>
    <input type="submit">

</form>
```

The screenshot shows a web browser window with the URL `localhost/aulas_iep/aula_05/`. The page displays a form with the following fields:

- Text:** An input field with placeholder text "Insira um texto".
- Password:** An input field for a password.
- Number:** An input field for a number, with placeholder "(xx) xxxxx-xxxx".
- Email:** An input field for an email address, with placeholder "seu-email@iep.com.br".
- Date:** A date input field with placeholder "dd/mm/aaaa".
- Checkbox:** A checked checkbox input field.
- File:** A file input field with placeholder "Escolher arquivo" and a message "Nenhum arquivo selecionado".
- Submit:** A submit button labeled "Enviar".



# EXERCÍCIO

HTML + CSS



Utilizando o que já aprendemos, monte uma página em **.html** e **.css** conforme exemplo abaixo:



Cadastro Aluno

+ Cadastro

Nome: Nome do aluno

Data Nascimento: dd/mm/aaaa

Cpf: xxx.xxx.xxx-xx

Senha:

Cadastrar

para

This image shows a screenshot of a web application interface. At the top right, there is a header with the text "Cadastro Aluno" and a "Cadastrar" button. Below the header, there is a form with several input fields. The first field is labeled "Nome:" and has the placeholder "Nome do aluno". The second field is labeled "Data Nascimento:" and has a date input field with the placeholder "dd/mm/aaaa" and a calendar icon. The third field is labeled "Cpf:" and has a text input field with the placeholder "xxx.xxx.xxx-xx". The fourth field is labeled "Senha:" and has a password input field. At the bottom of the form is a blue button with the text "Cadastrar". On the left side of the page, there is a large watermark or background image of the IEP Santa Casa de São José dos Campos logo, which features the letters "IEP" in blue with a heart shape integrated into the letter "P", and the text "Santa Casa de São José dos Campos" below it.



# BOOTSTRAP

Para utilizarmos o **bootstrap** em nosso site basta adicionar o link css que está na página de introdução do **bootstrap**:

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

## Quick start

Looking to quickly add Bootstrap to your project? Use jsDelivr, a free open source CDN. Using a package manager or need to download the source files? [Head to the downloads page](#).

### CSS

Copy-paste the stylesheet <link> into your <head> before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet">Copy
```

### JS

Para montarmos um site responsivo, precisamos primeiramente estruturar o layout, através de **containers, grid e columns**:

The screenshot shows the Bootstrap documentation website. At the top is a purple header bar with the Bootstrap logo and navigation links: Home, Docs, Examples, Icons, Themes, and Blog. Below the header is a search bar with placeholder text "Search docs..." and a keyboard shortcut "Ctrl + /". On the left side, there's a sidebar with a tree-like navigation menu. The "Layout" section is expanded, showing "Getting started", "Customize", and "Layout". Under "Layout", "Containers" and "Grid" are listed under "Breakpoints", while "Columns" is listed separately and highlighted with a red box. Other items in the sidebar include "Gutters", "Utilities", "Z-index", "CSS Grid", and sections for "Content" and "Forms". The main content area has a large title "Columns" and a subtext explaining how to modify columns using a flexbox grid system. A callout box contains the text "Heads up! Be sure to [read the Grid page](#) first before diving into how to modify and customize your grid columns." Below this, the heading "How they work" is visible.

Home Docs Examples Icons Themes Blog

Search docs... Ctrl + /

> Getting started  
> Customize  
▼ Layout  
  Containers (highlighted)  
  Grid  
  Columns (highlighted)  
Breakpoints  
Gutters  
Utilities  
Z-index  
CSS Grid  
> Content  
> Forms

# Columns

View on GitHub

Learn how to modify columns with a handful of options for alignment, ordering, and offsetting thanks to our flexbox grid system. Plus, see how to use column classes to manage widths of non-grid elements.

**Heads up!** Be sure to [read the Grid page](#) first before diving into how to modify and customize your grid columns.

## How they work

Também podemos utilizar os formulários e componentes do **bootstrap** para facilitar as nossas criações:

The screenshot shows the Bootstrap documentation website with a purple header. The header includes a logo, navigation links for Home, Docs, Examples, Icons, Themes, and Blog, and a search bar with a placeholder 'Search docs...' and a keyboard shortcut 'Ctrl + /'. A 'View on GitHub' button is also in the header.

The main content area has a sidebar on the left with links: Getting started, Customize, Layout, Content, Forms (which is highlighted with a red box), Components, Helpers, Utilities, Extend, About, and Migration.

The main content title is 'Forms'. Below it is a subtext: 'Examples and usage guidelines for form control styles, layout options, and custom components for creating a wide variety of forms.'

The page is divided into several sections with blue headings:

- Form control**: Style textual inputs and textareas with support for multiple states.
- Checks & radios**: Use our custom radio buttons and checkboxes in forms for selecting input options.
- Input group**: Attach labels and buttons to your inputs for increased semantic value.
- Layout**: Create inline, horizontal, or complex grid-based layouts.
- Select**: Improve browser default select elements with a custom initial appearance.
- Range**: Replace browser default range inputs with our custom version.
- Floating labels**: Create beautifully simple form labels that float over your input fields.
- Validation**: Validate your forms with custom or native validation.



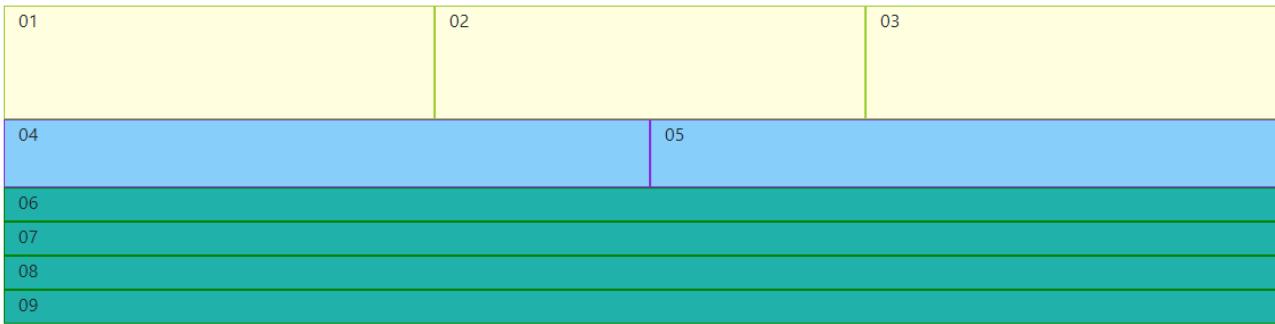
# EXERCÍCIO

HTML + CSS

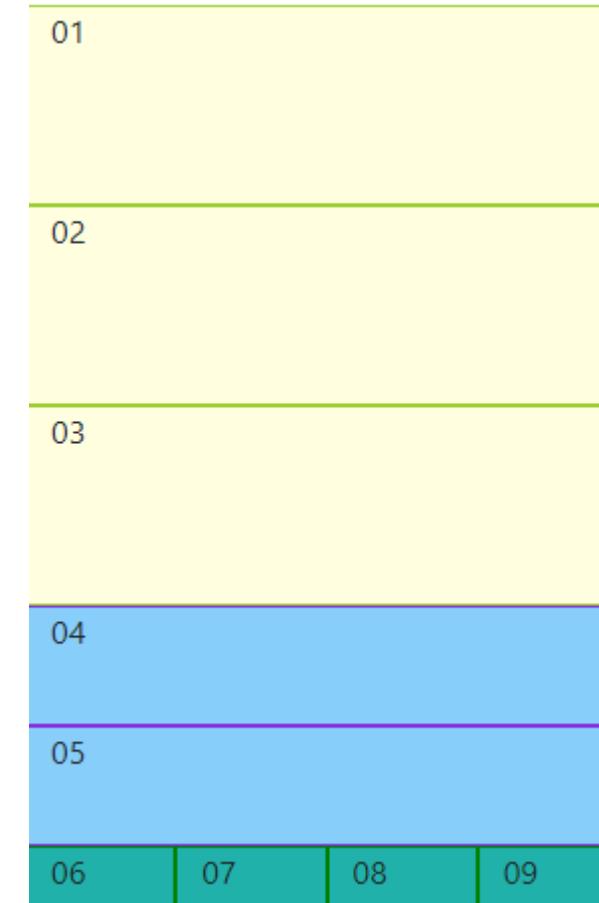


Utilizando o que já aprendemos, monte uma página em **.html**, **.css** e **bootstrap** conforme exemplo abaixo, lembrando que a página é a mesma só muda a visualização, no caso do google chrome utilizamos o inspecionar:

Desktop:



Mobile:





# REFERÊNCIAS

## Referências:

<https://code.visualstudio.com/>

<https://www.youtube.com/watch?v= R6YsIWUUFk>

<https://www.w3schools.com/>

<https://www.youtube.com/watch?v=Fj3gtbaF8WA>

<https://fontawesome.com/>

<https://getbootstrap.com/>