

Aula 5

A dupla HTML e CSS

Curso

HTML e CSS:
praticando HTML e CSS

O que vamos aprender?



Aprender a criar um botão transparente.



Aplicar o valor grid ao display da página.



Aprender a dividir a tela em colunas.



Grid e continuação

O botão foi adicionado à página da Alura Plus. Porém, observe que o seu tamanho ainda não está adequado pois ele está tomando uma largura grande da página, passando por cima da imagem, encobrindo-a.

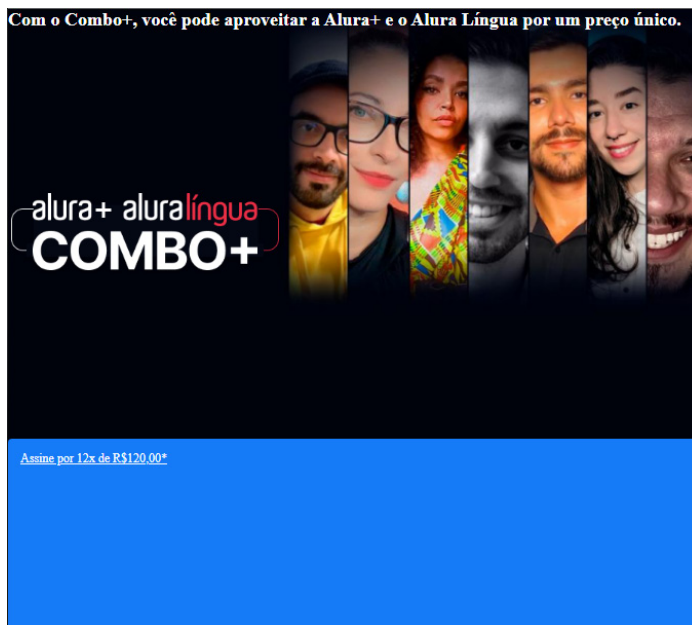


Para resolvermos esse problema, vamos utilizar a propriedade **display**. Vamos até o arquivo `style.css` e acrescentamos na classe `.container` a propriedade **display**. Com essa propriedade podemos definir como os elementos serão organizados em nossa página.

A propriedade **display** pode assumir diversos valores; entre eles, nós iremos adicionar o valor **grid**. *Grid*, em inglês, pode ser traduzido como rede ou algo parecido com uma tabela. Lembre-se, em uma tabela há colunas e linhas. Isso implica que os elementos com a propriedade `grid` ficam dispostos em linhas e colunas. Logo abaixo da propriedade de altura (**height**), vamos acrescentar o seguinte código:

```
.container {  
  height: 100vh;  
  display: grid;  
}
```

Ao carregarmos a página, vemos que a disposição dos elementos foi alterada, porém nosso botão ainda não está da forma que desejamos. Note pela imagem que ele está grande, ocupando espaço na página que poderia ser mais bem utilizado.



Para corrigir isso, no arquivo index.html, vamos inserir a tag **<div>**. Essa tag permite que os elementos sejam agrupados de acordo com sua função na página. Vamos inserir essa divisão logo após a nossa tag **<section>**.

```
<section class="container principal">  
  <div>
```

Não se esqueça de acrescentar o fechamento dessa tag antes da tag **</section>**.

```
    </div>  
  </section>
```

Ao carregarmos nossa página, parece que ainda não atingimos nosso objetivo, certo? Nosso botão continua ocupando toda a largura da página.



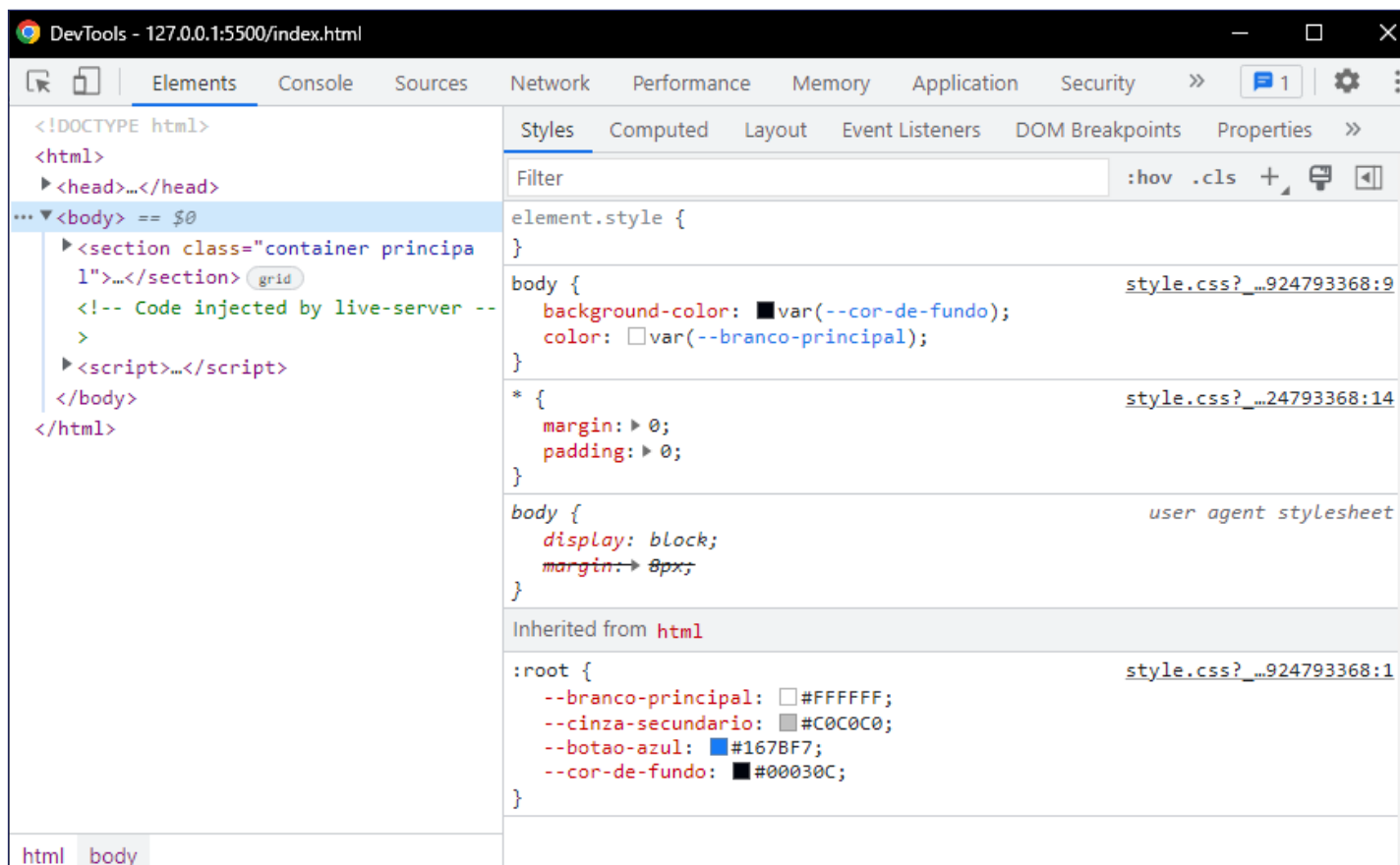
Ao usarmos o valor **grid** na propriedade **display**, precisamos informar para o CSS qual será a disposição dos elementos e qual o percentual da página que cada elemento ocupará. Vamos testar atribuindo o valor de 50% para os elementos. Para isso, vamos utilizar o seguinte código na classe: **.container**.

```
display:grid;  
grid-template-columns: 50% 50%;  
}
```

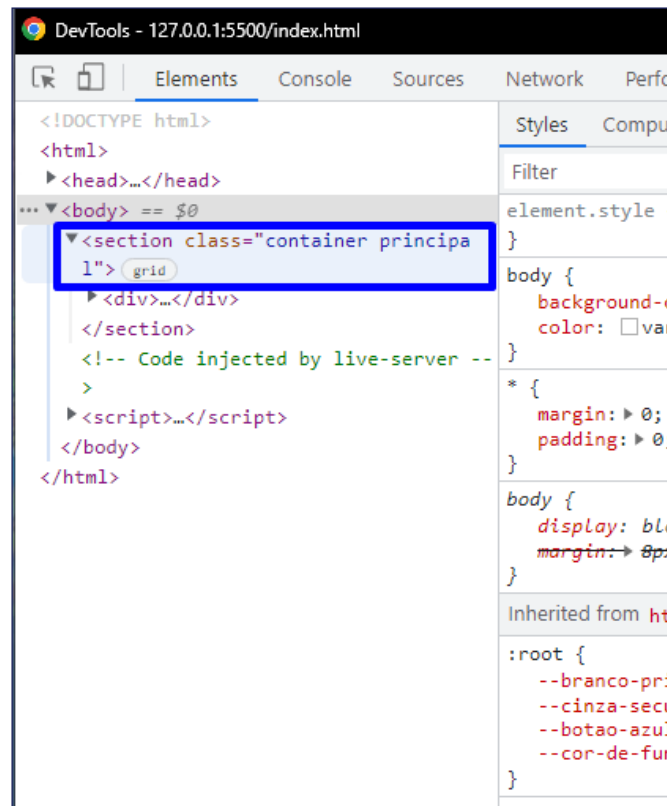
O código que escrevemos anteriormente diz que a nossa página será dividida em duas colunas, que ocupam cada uma 50% do tamanho total da página. E é exatamente o que podemos visualizar ao carregarmos a nossa página. Note na imagem abaixo que o botão ocupa metade da página.



Para visualizarmos a divisão feita pelo CSS, vamos até o navegador onde nossa página está carregada e pressionamos a tecla F12 para que seja aberta a janela das ferramentas do desenvolvedor, essa aqui:

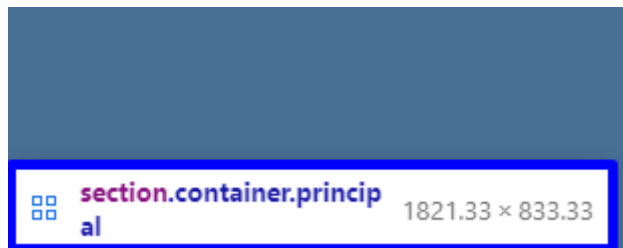


O primeiro passo consiste em você passar o mouse em cima da tag **<section>**.

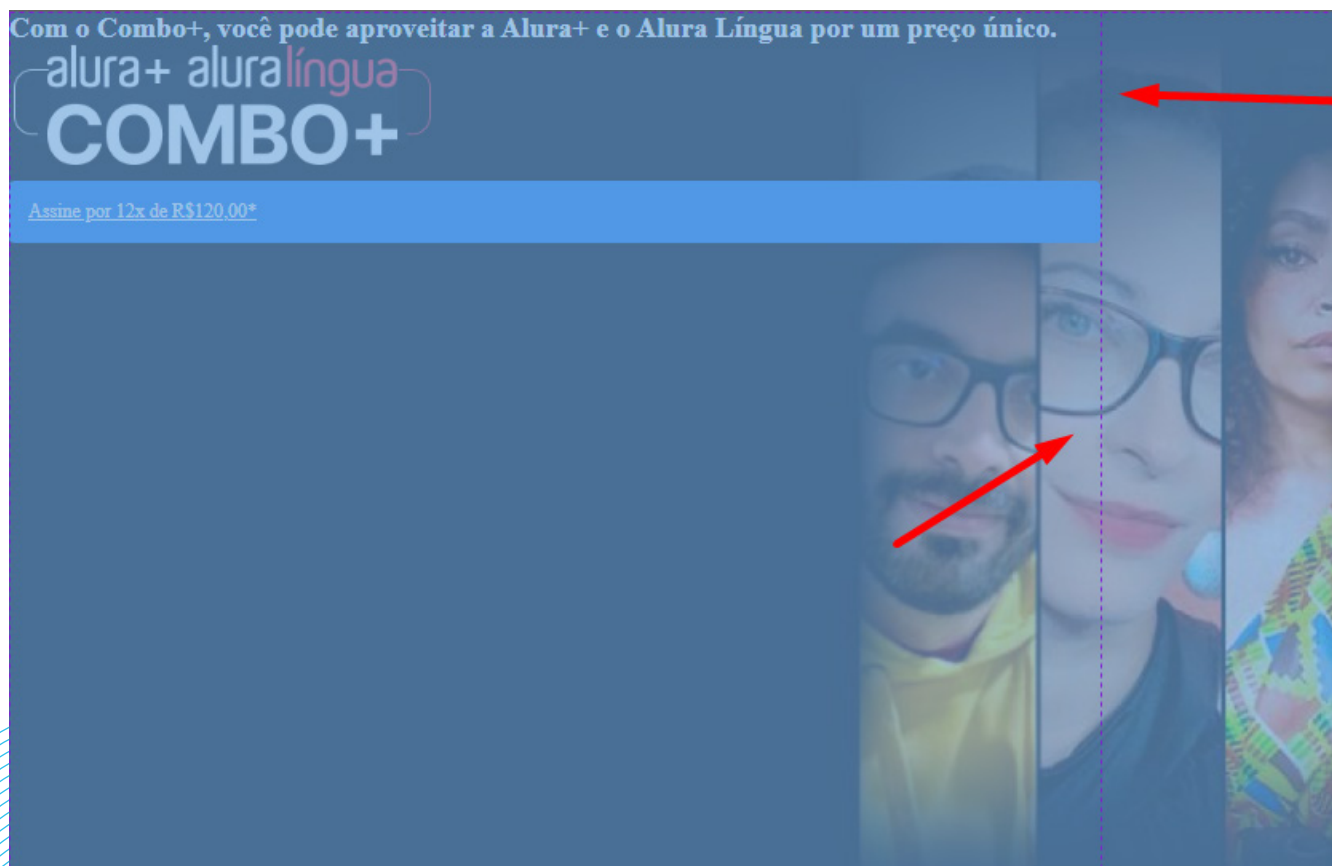


Ao fazermos isto, podemos verificar algumas coisas interessantes. Vamos descobrir quais são elas.

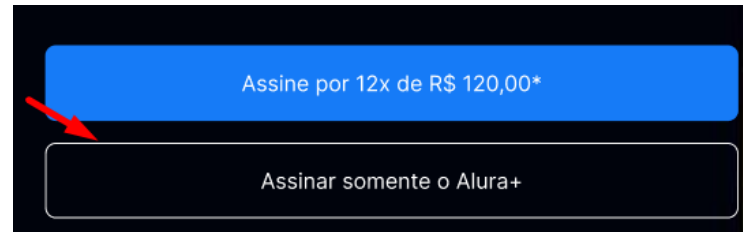
No canto inferior esquerdo da página, podemos verificar as tags e a classe que adicionamos.



Bem no meio da página, podemos verificar a divisão feita pelo valor **grid** da nossa propriedade **display**.



Nossa página já possui a divisão correta; agora, vamos iniciar a criação de outro botão. Vamos até o protótipo da Alura Plus, no site [figma.com](https://www.figma.com), para saber quais são as características desse novo botão e qual texto deverá ser escrito.



Esse botão possui uma borda branca e um texto na mesma cor com a seguinte frase: "Assinar somente o Alura+". Nosso objetivo agora é criar esse botão.

Vamos até o arquivo `index.html` para criar um novo botão. Do mesmo modo como criamos o botão azul, vamos usar os mesmos códigos, como você pode observar abaixo:

```
<a href="www.alura.com.br" class="container__botao">Assine por 12x de R$120,00*</a>  
<a href="www.alura.com.br" class="container__botao">Assinar somente o Alura+</a>
```

Note que a única diferença até o momento é o texto "Assinar somente o Alura+".

Ao carregarmos a nossa página, podemos observar que os dois botões foram criados e são iguais. Também não possuem espaçamento entre eles. Para criarmos um botão com propriedades diferentes, precisamos adicionar uma classe ao novo botão. Portanto, adicione a classe **.botao_secundario** ao elemento botão, no arquivo index.html.

```
<a href="www.alura.com.br" class="container_botao botao_secundario">Assinar somente o Alura+</a>
```

E no arquivo style.css vamos estilizar essa classe.

Definimos a cor de fundo do botão para transparente, afinal, ele não possui cor.

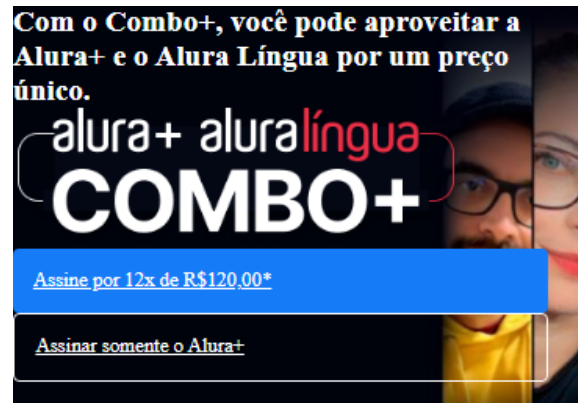
```
.botao_secundario {  
    background-color: transparent;  
}
```

Vamos agora aprender a aplicar a borda no novo botão transparente utilizando um dos padrões de cores que criamos nas aulas iniciais. Os valores das medidas e das propriedades da borda são:

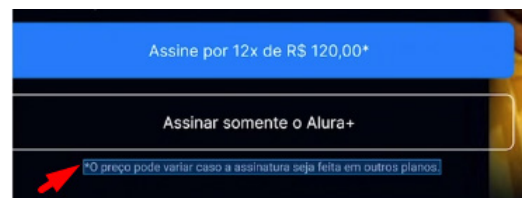
- **2px**: espessura da borda
- **Solid**: é uma linha reta, sem efeitos
- **var()**: padrão de cor armazenado em uma variável definida.

```
background-color: transparent;  
border: 2px solid var(--branco-principal);
```

Ao carregarmos nossa página, nossa classe foi aplicada ao botão.



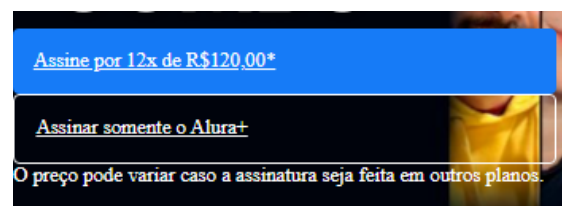
O próximo elemento que vamos adicionar representa o texto exibido após os botões:



Esse texto será adicionado por meio da tag `<p>`, que exibe um parágrafo na página. Além disso, vamos criar uma nova classe para estilizar esse novo elemento de texto. O nome da classe será **.container__aviso**.

```
<p class="container__aviso">O preço pode variar caso a assinatura seja feita em outros planos.</p>
```

Ao carregarmos a página, nosso texto está sendo exibido sem estilização alguma.



Desafio

Nesta aula, aprendemos como aplicar a propriedade `display` com alguns valores. A partir do valor `grid`, podemos dividir nossa página em colunas de diferentes tamanhos. O desafio de hoje é conhecer a disposição dos objetos em diferentes percentuais da largura total da página. Para isso, você deverá ir até o arquivo `style.css` e alterar os valores do `percentual` variando entre 10% e 100%. Varie também a quantidade de colunas (30% 30% 30% = 3 colunas; 60% 40% = 2 colunas; e assim por diante). Analise o resultado na aparência do site. Em qual delas fica mais agradável de navegar?