

Aula 4

# A dupla HTML e CSS

Curso

**HTML e CSS:**  
**praticando HTML e CSS**

# O que vamos aprender?



Aprender a criar e estilizar um botão “clicável”.



Conhecer e aplicar diferentes unidades de medida em elementos, tais como em botões.



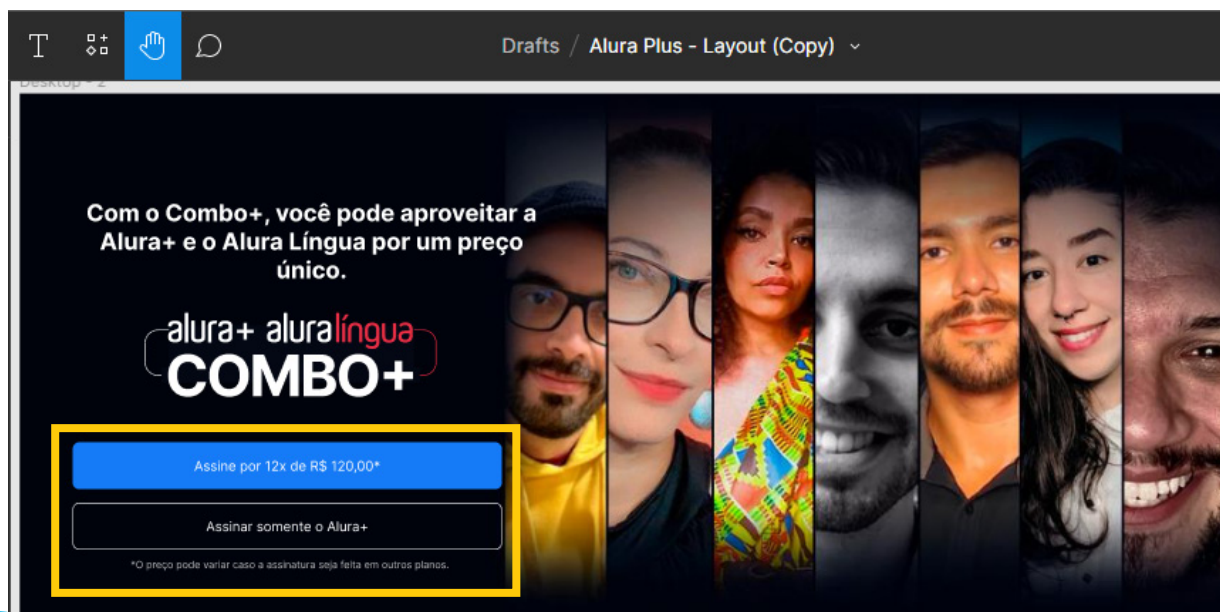
Conhecer e aplicar bordas em elementos.



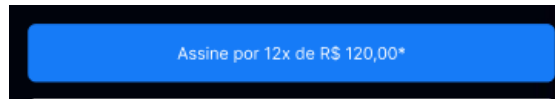
# Criando um botão

As imagens da primeira seção do site já estão adicionadas na página da Alura Plus. No entanto, alguns elementos essenciais da página ainda devem ser adicionados. Portanto, vamos até o protótipo da Alura Plus no figma.com para analisar o que devemos adicionar agora.

Ao abrirmos o protótipo, podemos observar que as imagens e o título já foram acrescentados em nosso projeto no VSCode. No entanto, os elementos que representam os botões da seção não estão adicionados ainda. Na imagem a seguir, podemos identificar os botões que faltam, os quais estão em destaque:



Vamos começar criando o primeiro botão, de cor azul.



Para isso, vamos até o arquivo index.html e, dentro do trecho da tag **<section>**, vamos acrescentar uma nova tag chamada **<a>**. A tag **<a>** é a abreviação para “*anchor*” do inglês, que significa âncora. Ela permite que um texto comum se transforme em um texto com link capaz de levar o usuário para algum site. Assim, a tag **<a>** funcionará da seguinte maneira: ao clicarmos no texto, nós seremos redirecionados para uma nova página da internet. É como a âncora do navio, utilizada para “conectar” a embarcação a um lugar fixo na água. Portanto, logo após a tag **<img>**, vamos adicionar a seguinte instrução:

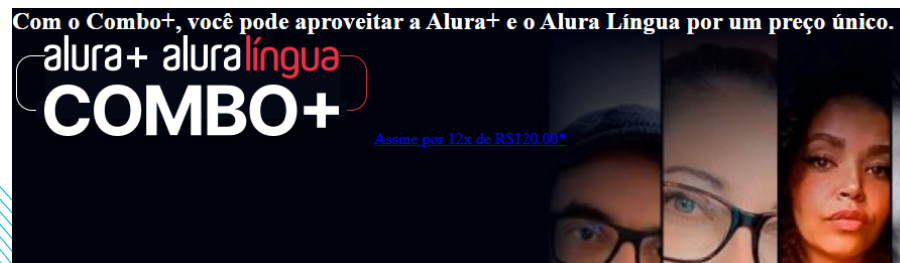
```
  
<a href="www.alura.com.br"></a>
```

O endereço que adicionamos é o site da Alura; isso significa na prática que, ao clicarmos no texto do botão azul, o site da Alura será aberto.

Vamos agora adicionar o texto que será exibido no botão.

```
<a href="www.alura.com.br">Assine por 12x de R$120,00*</a>
```

Ao carregarmos a página no navegador, podemos observar que o nosso texto foi adicionado e é “clicável”. No entanto, esse texto não se parece com um botão e não está posicionado corretamente.



Para realizarmos as modificações no elemento de texto para que fique com a aparência de um botão, vamos criar uma classe no arquivo style.css. Mas antes, acrescente essa classe à tag **<a>** no arquivo index.html que terá o nome de “**container\_\_botao**”.

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

É uma boa prática de programação nomear sugestivamente as classes para que elas possam ser identificadas com facilidade. Para o projeto da Alura Plus, podemos observar que muitas classes serão criadas com nomes que iniciam com “container”, que identifica a seção principal da página, enquanto o nome seguinte será o elemento que vamos estilizar, neste caso o botão.

Vamos até o arquivo style.css para iniciar as modificações e transformar o texto em um botão. O primeiro passo é acrescentar a classe **.container\_\_botao** ao arquivo style.css. Feito isso, vamos alterar a cor de fundo do texto que inserimos; neste caso, como estamos criando o botão azul, a cor será azul.

```
.container__botao {  
    background-color: var(--botao-azul);  
}
```

Lembre-se de que estamos usando os padrões de cores que foram definidos nas primeiras aulas por meio das variáveis.

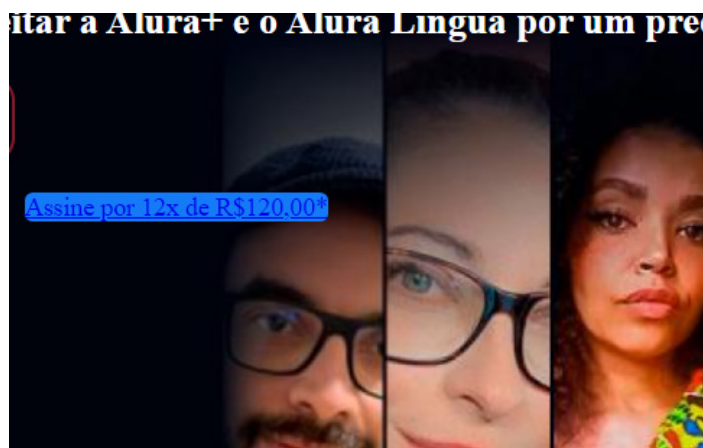
Ao carregarmos a página, podemos observar que a cor de fundo do texto foi modificada corretamente.



Vamos acrescentar uma borda arredondada ao nosso botão. Para isto, usaremos a propriedade **border-radius**. Essa propriedade exige um valor de medida. Portanto, vamos acrescentar **5px**. A unidade px significa pixels, portanto o raio da borda terá o valor de 5 pixels. Quanto maior o valor, mais arredondados os cantos dessa borda serão.

```
background-color: var(--botao-azul);  
border-radius: 5px;
```

Ao carregarmos a página, podemos observar que a borda foi aplicada ao botão.



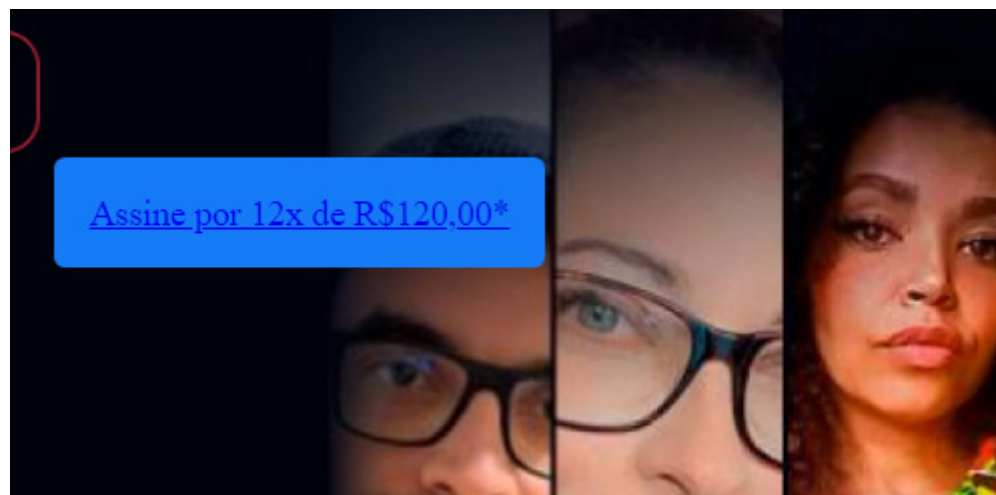


Pela imagem da página anterior, podemos notar que o texto está muito “grudado” nas bordas do fundo azul do botão. Podemos melhorar essa estética adicionando um espaçamento interno nesse elemento. Fazemos isso utilizando a propriedade **padding**. Assim como a borda, a propriedade **padding** exige uma medida para definir quanto espaçamento é necessário.

Vamos utilizar a medida **em**. Cada 1 **em** equivale a 16 px, portanto 2 **em** equivalem a 32 px, assim por diante.

```
border-radius: 5px;  
padding: 1em;
```

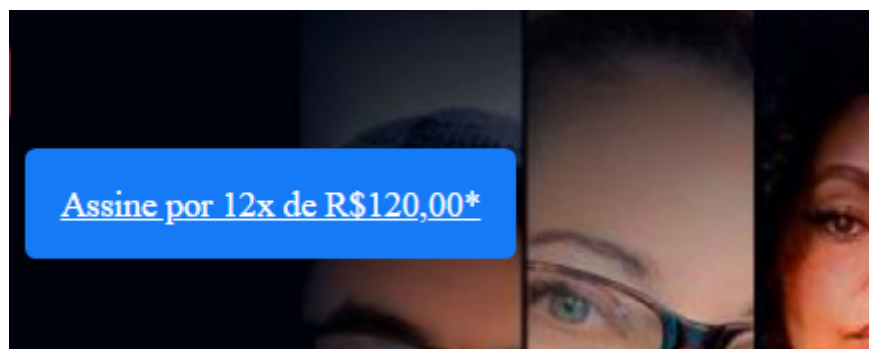
Ao carregarmos a página, o espaçamento interno foi adicionado.



O texto do nosso botão deve ser na cor branca. Dessa forma, vamos alterar a cor utilizando a propriedade **color**.

```
padding: 1em;  
color: var(--branco-principal);
```

Ao recarregarmos a página, nosso texto estará na cor branca.



No protótipo, o botão é exibido logo abaixo da imagem “Combo”, porém neste momento ele está sendo exibido à esquerda da imagem. Para resolvermos isso, vamos até o arquivo `style.css`.

Os elementos no HTML possuem um *display* automático que é o modo de visualização (como os elementos vão aparecer para nós, tal como tamanho, posicionamento, cor etc). Se não fizermos alterações, os elementos se comportam como programados originalmente. No caso do nosso elemento botão, seu **display** automático é o **inline**. Isso significa que o espaço que esse elemento ocupa na tela permite que mais elementos sejam adicionados ao lado dele.



Porém, nós queremos mudar isso, afinal, os elementos devem estar dispostos um abaixo do outro. Para isso, vamos adicionar à propriedade **display** o valor **block**.

```
color: var(--branco-principal);  
display: block;
```

O valor **block**, como o nome diz, bloqueia toda a linha em que o elemento está posicionado, fazendo com que obrigatoriamente outros elementos “pulem” para a próxima linha.

Ao carregarmos a página, podemos observar que o botão foi corretamente posicionado.



Nosso botão está se tornando cada vez mais semelhante ao exibido no protótipo, no entanto, ainda há alguns ajustes para serem feitos. Por exemplo, a largura do nosso botão que atualmente está ocupando toda a largura na tela. Essas modificações serão feitas ao longo das próximas aulas.

# Desafio

Para praticar nossos aprendizados desta aula sobre estilização de elementos, tente trocar a cor de fundo e a cor do texto do nosso botão. Lembre-se de que para realizar tais modificações, precisamos alterar o arquivo CSS. Além disso, as propriedades a serem modificadas são: **background-color** e **color**, respectivamente.