

Aula 2

Iniciando o projeto

Curso

HTML e CSS:
praticando HTML e CSS

O que vamos aprender?



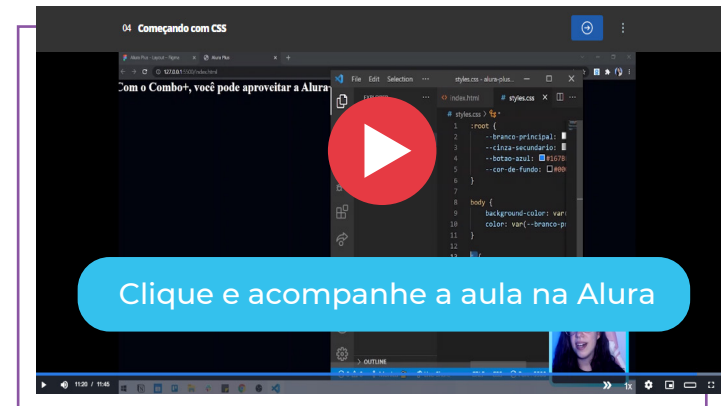
Criar variáveis no CSS.



Conhecer e aplicar o seletor universal.



Aplicar cores em elementos da página.



Começando com CSS

Os códigos essenciais que já estão adicionados em nosso projeto no VSCode nos ajudarão a manter uma estrutura organizada e de fácil leitura ao longo do desenvolvimento do projeto. Lembrando que os códigos que adicionamos são os seguintes:

```
!DOCTYPE html>
<html>
  <head>
    <title>Alura Plus</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
  </body>
</html>
```

Vamos agora entender a função de cada um dos códigos utilizados.

Em HTML, nós chamamos de *tags* os códigos que são escritos entre `<>`.

Ao abrirmos um novo arquivo no VSCode, precisamos informar qual será o tipo de documento que será escrito, neste caso, um arquivo HTML. Para isso, usamos a seguinte *tag*:

```
<!DOCTYPE html>
```

E então, para definir o início do conteúdo, precisamos abrir uma *tag* chamada `<html>`. Com exceção de algumas *tags*, sempre que abrirmos uma nova *tag*, devemos também fechá-la. Para isso, reescrevemos o nome da *tag* e acrescentamos a barra / para representar o fechamento. Essa barra deve ser adicionada antes do nome, mas dentro destes símbolos: `<>`

```
<!DOCTYPE html>  
<html>  
</html>
```

A *tag* que representa o primeiro conteúdo a ser inserido na página é **<head>**. Por meio dela, podemos definir o título da página, o conjunto de caracteres que será utilizado e outras configurações importantes do nosso *site*. *Head*, em inglês, significa “cabeça”.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
</html>
```

Para concluir o conjunto dessas *tags* está faltando a que representa o corpo do arquivo. Esta é chamada de **<body>**, responsável por exibir todo o conteúdo de uma página.

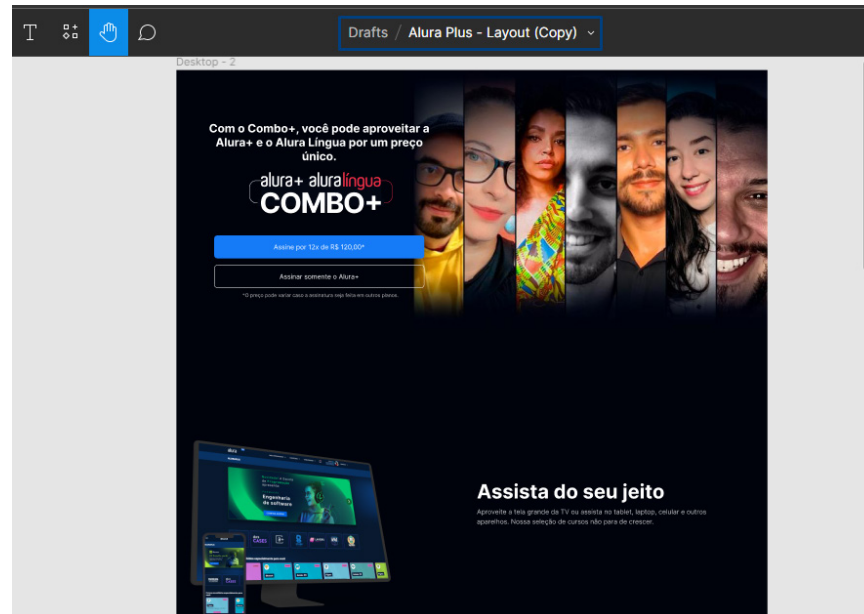
```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

Para definirmos o título da aba que será exibida no navegador, usamos a *tag* **<title>**. Já para definirmos o conjunto de caracteres que será utilizado na linguagem do *site*, usamos a *tag* **<meta>** com a propriedade `charset` e a definimos para “UTF-8”. Existem vários conjuntos de caracteres; como usaremos acentos e a letra com cedilha (ç), precisamos adicionar esse padrão. Para tornar nossa página responsiva, definimos que a largura da página será ajustada, independentemente do dispositivo acessado. Para isso, utilizamos a *tag* **<meta>** com a propriedade `viewport`.

Por fim, para agregarmos o arquivo `style.css` ao nosso arquivo `index.html`, usamos a *tag* **<link>** com o nome do arquivo criado.

```
<head>
  <title>Alura Plus</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="style.css">
</head>
```

Dadas essas definições, vamos entender para que serve o arquivo style.css em nosso código. Para começar, vamos até o protótipo da Alura Plus no Figma.com. Ao abrimos o projeto, a seguinte tela será mostrada:

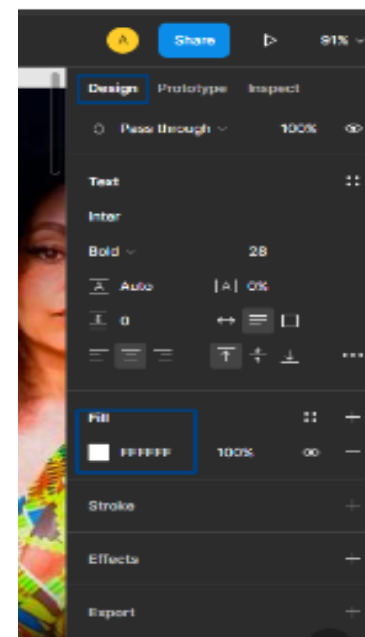


Quais padrões de estilo podemos visualizar nessas telas? Pois bem, os títulos possuem a cor branca, o fundo das telas é bem escuro, as descrições são breves e as cores possuem tons de cinza etc. Após analisarmos tais características, devemos incluí-las em nosso arquivo style.css. Portanto, vamos até o VSCode, para fazer-mos essas inclusões.

Vamos definir configurações iniciais e padronizadas para a nossa página, e, portanto, usaremos a pseudo-classe **:root** dentro do arquivo `style.css`.

```
:root {  
}
```

Feito isto, precisamos identificar qual é a cor exata dos títulos da nossa página, e, portanto, vamos até o figma.com, selecionamos qualquer título através do clique duplo do *mouse*, e, na coluna à direita da tela, chamada “Design”, podemos visualizar o código hexadecimal da cor de que precisamos.



Já identificamos que a cor do título possui o valor de “**FFFFFF**”, então, vamos até o arquivo style.css para adicionarmos esse valor aos títulos da página. Para que todos os títulos estejam padronizados com essa cor, criaremos uma variável chamada **--branco-principal**.

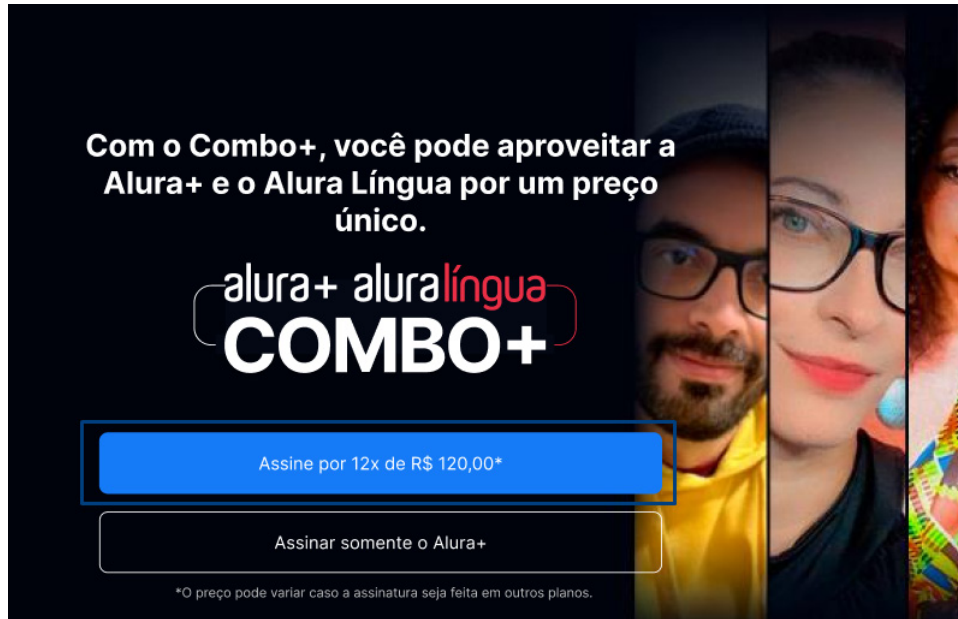
Lembrando que, por se tratar de um valor hexadecimal, devemos adicionar o símbolo # (*hashtag*) antes da sequência de letra F.

```
:root {  
  --branco-principal: #FFFFFF;  
}
```

Faremos o mesmo passo a passo para descobrir o código da cor cinza utilizado nas descrições. Portanto, novamente vá até o [figma.com](https://www.figma.com) e identifique qual é o código correspondente a essa cor. Uma vez que você tenha feito isto, retorne ao arquivo style.css e crie uma nova variável para armazenar tal cor. Essa variável terá o nome de **--cinza-secundario**.

```
--branco-principal: #FFFFFF;  
--cinza-secundario: #C0C0C0;
```

A próxima variável que deve ser criada será responsável por colorir o botão exibido na página:

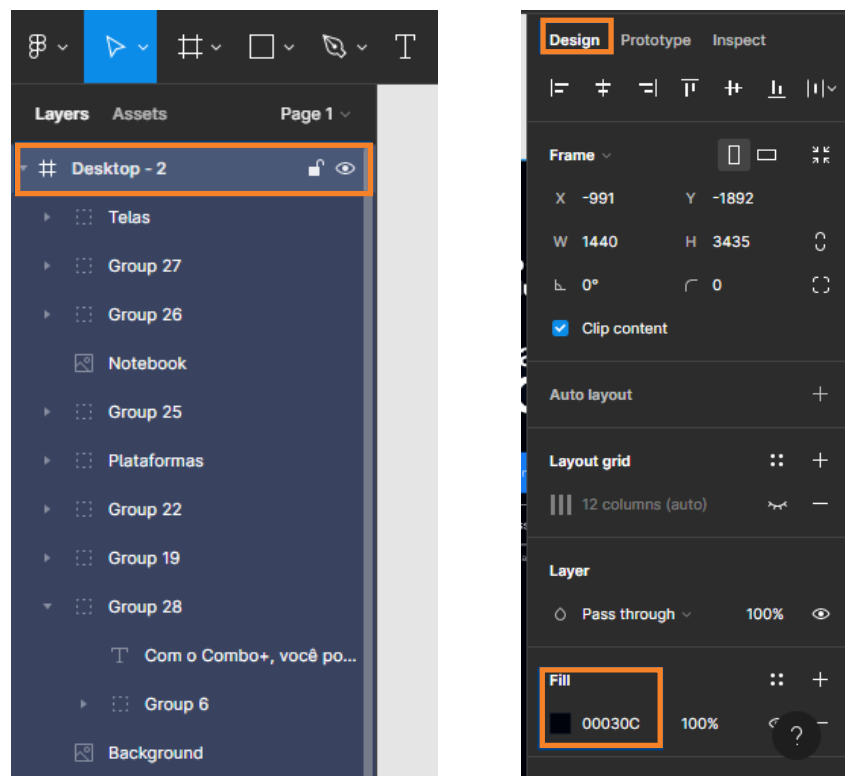


```
--cinza-secundario: #C0C0C0;  
--botao-azul: #167BF7;
```

Para sabermos o tom de azul utilizado, devemos seguir os mesmos passos anteriores. Clique duplo no botão “Verificar a coluna de Design” “Verificar o código hexadecimal da cor”.

Portanto, crie uma variável chamada **--botao-azul** e adicione o valor da cor.

A última variável que precisamos criar neste momento corresponde ao fundo de tela da página. Para descobrirmos o código da cor, devemos voltar até o figma.com, clicar em “Desktop”, na coluna à esquerda chamada “Layers”, e então, verificarmos o código da cor na coluna “Design”, à direita da tela.



Feito isso, criaremos a variável **--cor-de-fundo** no arquivo `style.css`:

```
--botao-azul: #167BF7;  
--cor-de-fundo: #00030C;
```

Ao criarmos variáveis que correspondem a cores diferentes, podemos facilmente manipular essas informações em nosso código e facilitar o entendimento do outro quando formos explicar esse código a ele. Imagine a seguinte situação: Você precisa que outra pessoa desenvolvedora adicione novas cores aos títulos. Se você utilizar o código da cor para se referir ao título, poderá deixar a pessoa confusa, visto que temos muitos códigos e decorá-los é uma tarefa complexa. Porém, se você relacionar essa mudança com o nome da variável, isto é, alterar a cor na variável branco-principal, isso significará que todo o texto principal da página, isto é, os títulos, devem ser alterados.

Já temos nossas cores iniciais, portanto, vamos adicionar a nossa cor de fundo à página; para isto, utilizaremos a *tag* **<body>** dentro do arquivo style.css. Lembrando que, no arquivo HTML, usamos o símbolo <> para as *tags*; já no arquivo CSS, excluímos esse símbolo, mas adicionamos as chaves {} para identificar as *tags*, identificadores e classes do meu projeto.

Portanto, ao adicionarmos a *tag* **<body>** no arquivo style.css, devemos também escrever a propriedade **background-color**, que define a cor do fundo de tela.

Como estamos trabalhando com variáveis, após os dois pontos : da propriedade, adicionamos também o prefixo **var**. Ao fazermos isto, todas as variáveis criadas serão exibidas em uma lista:

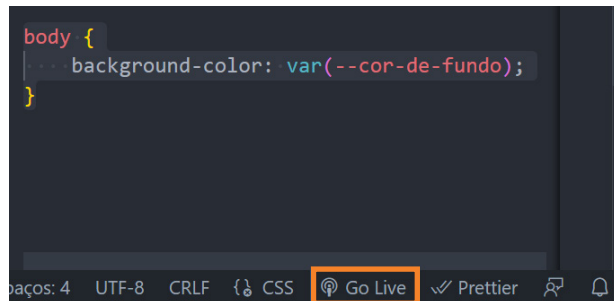
```
body {  
  background-color: var();  
}
```

■	--botao-azul	#167BF7
■	--branco-principal	#FFFFFF
■	--cinza-secundario	#C0C0C0
■	--cor-de-fundo	#00030C

Então, escolhemos a variável que corresponde à cor de fundo da página.

```
body {  
  background-color: var(--cor-de-fundo);  
}
```

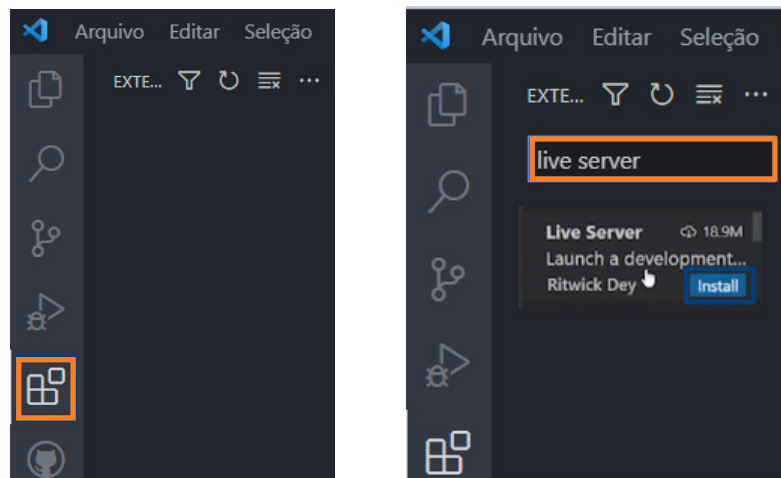
Após salvar o seu progresso, usando o atalho *Ctrl+S*, vá até o botão “Go Live”, na barra inferior do VSCode, para testar em seu navegador.



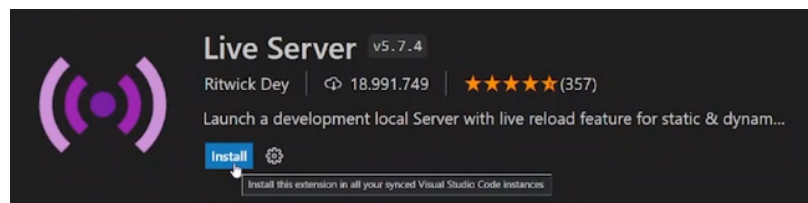
Assim podemos visualizar que o título da página está sendo aplicado e a cor de fundo também.



Caso o botão “Go Live” não esteja aparecendo para você, vá até o ícone de extensões no VSCode, e então procure por *live server*. Assim que aparecer a primeira opção, clique em “Install”.

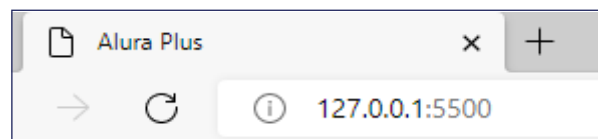


A janela abaixo será aberta, e novamente você deverá clicar em “Install”.



Feitos esses passos, agora você possui o botão “Go Live” para testar diretamente no navegador.

Perceba que, aos termos esta página no navegador, uma sequência de números é exibida no topo da nossa página.

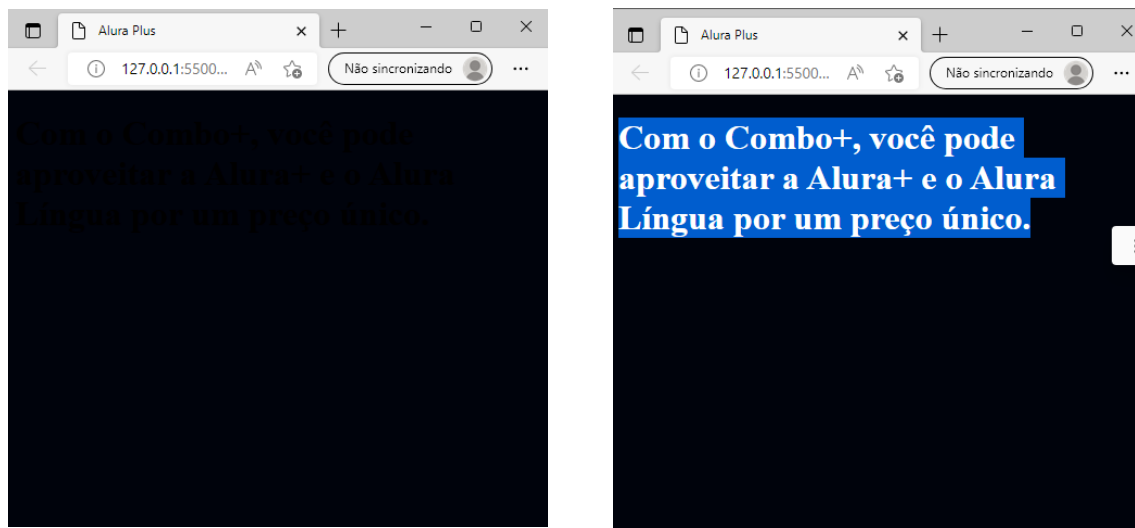


Estes números correspondem ao endereço do nosso servidor local, e é através dele que conseguimos testar a nossa página pelo navegador. Cada alteração feita nos arquivos no VSCode é refletida na página que está aberta do navegador. Por isso, é sempre importante lembrar-nos de salvar o arquivo toda vez que alterarmos algo no código.

Vamos realizar um teste: adicione no arquivo index.html, dentro da tag **<body>** o título da página Alura Plus, aquele escrito em branco, no protótipo do figma.com. Para isto, vamos adicionar também a tag de título **<h1>**.

```
<body>
  <h1>Com o Combo+, você pode aproveitar a Alura+ e o Alura Língua por um preço único.</h1>
</body>
```

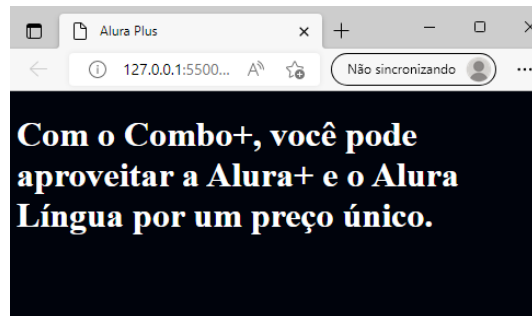
Ao executarmos, observamos que nada foi escrito na tela. Porém, ao passarmos o *mouse* em cima do fundo, segurando o cursor, podemos visualizar que o texto foi escrito corretamente, apenas possui uma fonte escura.



Para aplicarmos o padrão de cor que criamos no início desta aula, vamos acrescentar a propriedade `color` em nosso CSS e definirmos o valor dessa propriedade como a variável `--branco-principal`.

```
background-color: var(--cor-de-fundo);  
color: var(--branco-principal);  
}
```


Ao visualizarmos a página no navegador, podemos observar que o estilo foi aplicado corretamente.



Para melhorarmos a visualização do texto, precisaremos alterar as margens e o espaçamento das letras; mas, antes disso, precisamos definir que o ponto de partida seja 0.

Para fazermos isso, usaremos o seletor universal *. Ele permite configurar muitos elementos com o mesmo padrão. Portanto, vamos até o arquivo style.css e definiremos a margem e o espaçamento interno padrão como 0.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

Avançamos nosso conhecimento em CSS, atribuindo seletores e criando variáveis com o objetivo de facilitar a leitura e a implementação em nosso projeto.

Desafio

Ao longo desta aula, você criou algumas variáveis para armazenar códigos hexadecimais de cores e aplicá-los em sua página da Alura Plus. Além do sistema hexadecimal, podemos adicionar cores aos elementos usando o sistema RGB e o nome das cores em inglês. O desafio desta aula será realizar uma pesquisa sobre os códigos identificadores de cores, e então aplicar dois ou mais códigos diferentes no fundo de tela da página. Ao realizar este desafio, você praticará como estilizar a página de diferentes maneiras.