

## MÓDULO 02

### PROJETO PRATICO 1

Vamos dar início ao nosso primeiro projeto prático onde vamos colocar em prática todos os conceitos aprendidos no módulo anterior de HTML e CSS. Você irá perceber que com apenas com essas duas tecnologias já é possível construir aplicações reais. O projeto que iremos desenvolver nesse módulo trata-se de um site que simula a compra de ingressos para o filme Batman lançado em 2022.

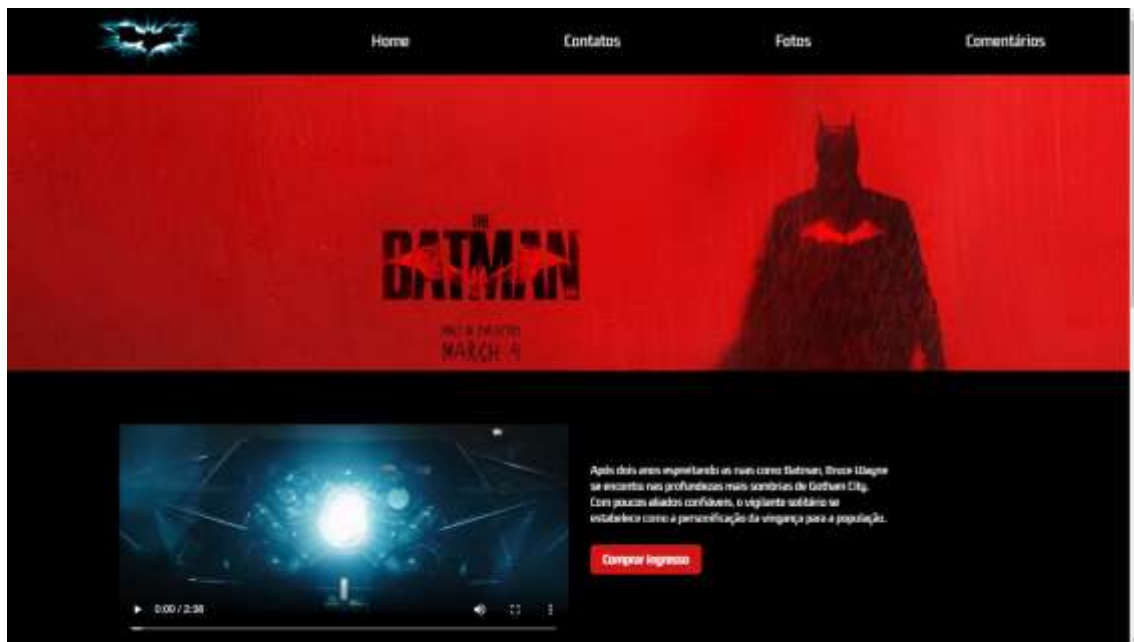


Imagem do projeto

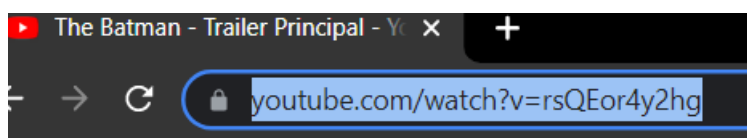
Vamos desenvolver o layout tanto para desktop quanto para mobile (conceito de responsividade). É muito importante que você desenvolva esse projeto durante as próximas aulas para que esse conhecimento seja fixado.

Para a criação desse projeto vamos precisar de duas coisas:

- Trailer oficial do filme
- Logotipo

Para realizar o download do trailer você pode seguir os seguintes passos:

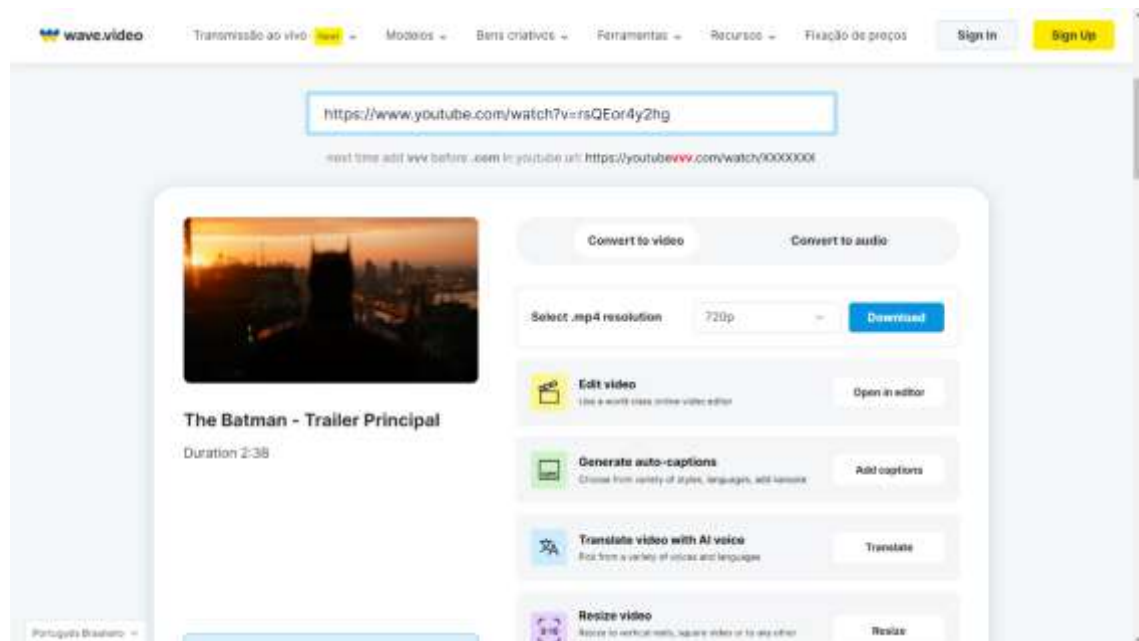
1. Procure o trailer no YouTube e copie o seu link



2. Acesse o site abaixo para realizar o download:

<https://wave.video/br/>

Basta colar o link do YouTube e clicar no botão download



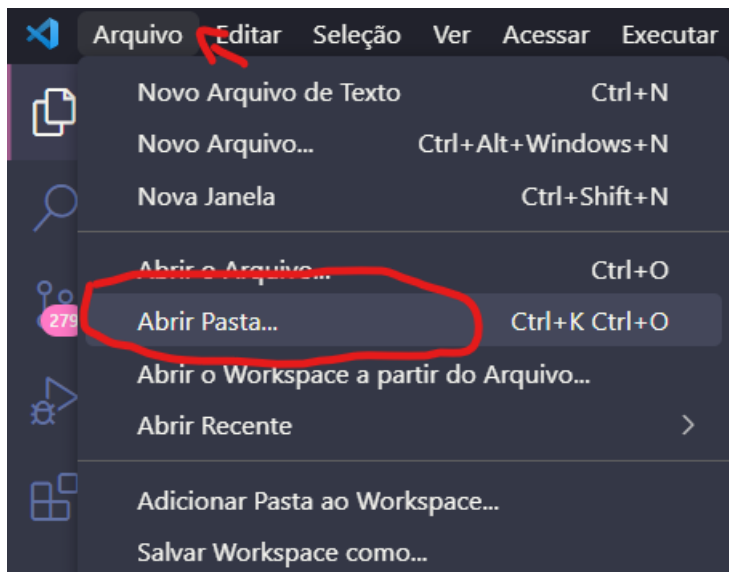
Caso não consiga usar esse site, existem vários outros que funcionam da mesma forma. Basta pesquisar no Google: converter YouTube MP4

O Logotipo deve ser baixado no formato JPG ou PNG. Caso o formato seja JPG é importante que a imagem possua um fundo preto, pois essa será a cor do nosso Header. Exemplo:

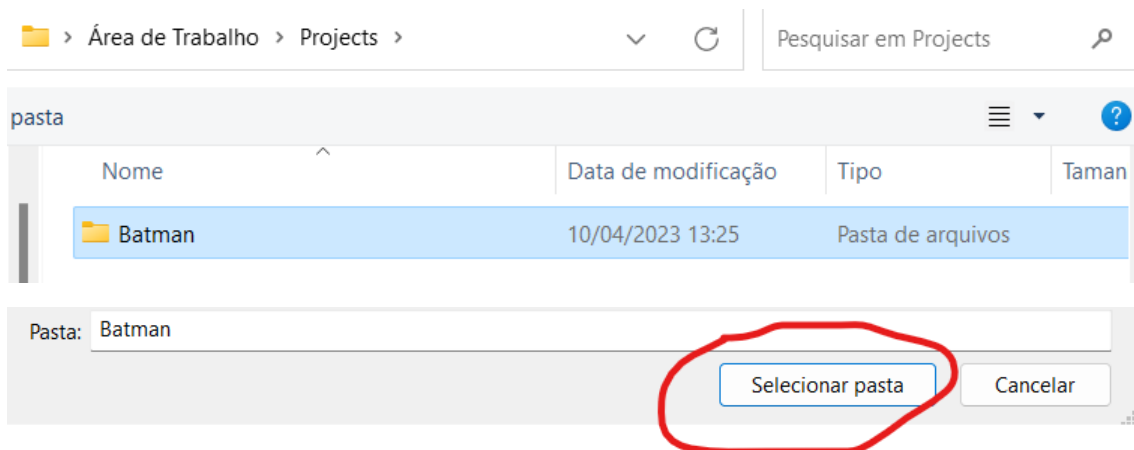


Caso tenha alguma imagem e queira remover o fundo dela você pode usar o site [remove.bg/pt-br](https://remove.bg/pt-br). É só fazer o upload da imagem que o fundo será removido automaticamente. Depois é só clicar no botão download e baixar a imagem já no formato PNG.

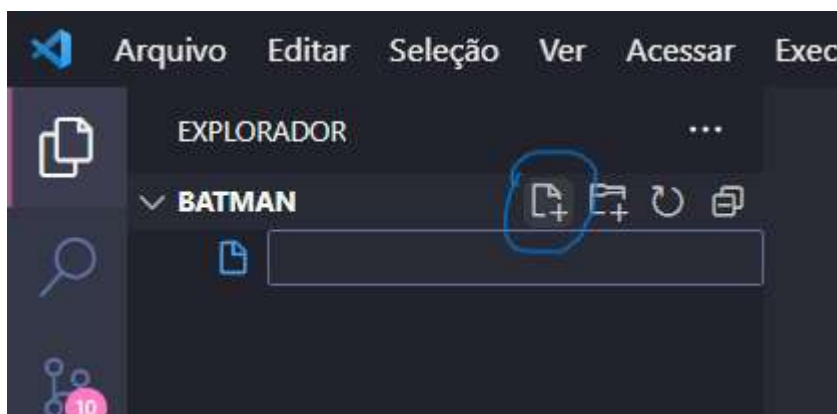
Agora como aprendemos no módulo anterior crie uma pasta para o seu projeto e coloque dentro dela a imagem logo e também o vídeo do trailer. Após isso abra o vs code, clique em arquivo e depois em abrir pasta:

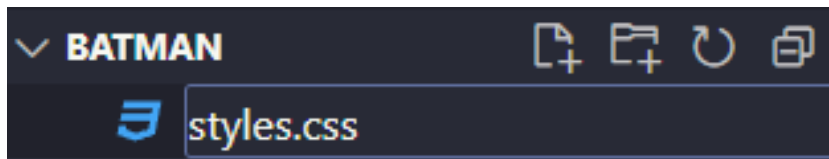
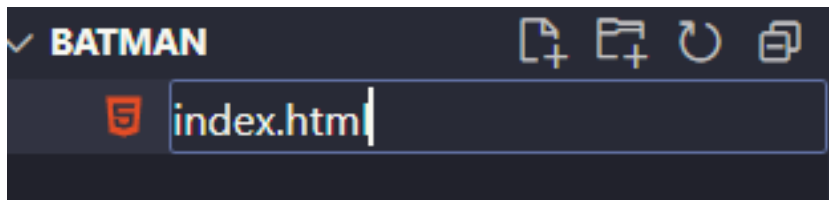


Agora procure a pasta do seu projeto e clique em selecionar pasta:



Após isso, clique no botão novo arquivo e crie os arquivos: index.html e styles.css.





No arquivo index.html, crie a estrutura inicial html e não se esqueça de linkar o arquivo styles.css.

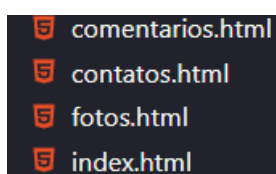
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Batman 2022 | Compra de ingressos</title>
  <link rel="stylesheet" href="./styles.css"/>
</head>
```

Agora já temos a estrutura inicial do nosso projeto e já estamos prontos para começar.

## CRIANDO O HEADER

A primeira coisa que vamos criar no nosso projeto é o Header, que nada mais é do que o cabeçalho da página. Mas antes de começar, vamos primeiro criar mais 3 arquivos, que serão as páginas do projeto:

contatos.html; fotos.html e comentarios.html.



Cada uma dessas páginas deverá ter uma estrutura inicial html e, por hora, contém um texto com o nome da página dentro de <body>.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     Fotos
11 </body>
12 </html>

```

Agora vamos criar o nosso Header.

```

<header>

</header>

```

Dentro de Header teremos:

- Logotipo (img)

```

```

- Navegação (nav)

```

<nav>

</nav>

```

- Lista não ordenada (ul)

```

<ul>

</ul>

```

- 4 itens (li). Home, Contatos, Fotos e Comentários

```

<li></li>
<li></li>
<li></li>
<li></li>

```

O código deverá ficar assim:

```
<header>
  
  <nav>
    <ul>
      <li>Home</li>
      <li>Contatos</li>
      <li>Fotos</li>
      <li>Comentários</li>
    </ul>
  </nav>
</header>
```

Lembrando que o nome do arquivo que você baixou pode ser diferente do exemplo acima, então procure colocar o nome do arquivo de acordo com o que você baixou.

Após isso, iremos envolver cada item com uma tag âncora `<a>` e direcioná-la para sua respectiva página conforme código abaixo:

```
<header>
  <a href="./index.html">
    
  </a>
  <nav>
    <ul>
      <a href="./index.html">
        <li>Home</li>
      </a>
      <a href="./contatos.html">
        <li>Contatos</li>
      </a>
      <a href="./fotos.html">
        <li>Fotos</li>
      </a>
      <a href="./comentarios.html">
        <li>Comentários</li>
      </a>
    </ul>
  </nav>
</header>
```

Observe que também envolvemos o logotipo com uma tag âncora e direcionamos para a página index.html, que é a nossa página Home do projeto. Assim, toda vez que o usuário clicar no logo ele será direcionado para a página Home. Este é um comportamento comum em praticamente todo site.

Após criar o Header, salvar e abrir o arquivo index.html no navegador, o resultado deverá ser assim:

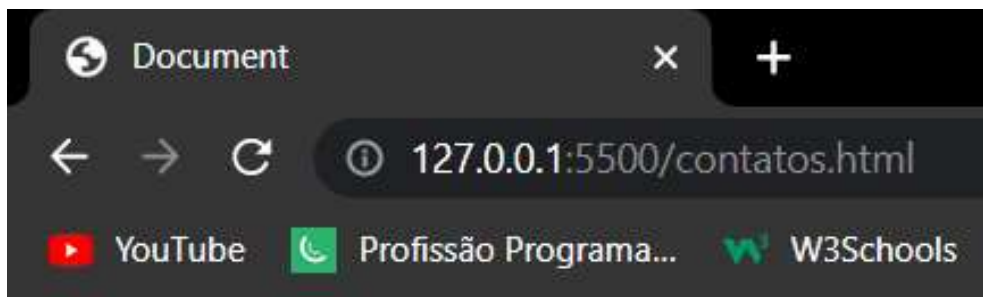


- [Home](#)
- [Contato](#)
- [Fotos](#)
- [Comentários](#)

Você também pode testar se a navegação está funcionando clicando nos links criados e ver se eles estão direcionando para a página correta.

- [Home](#)
- [Contatos](#)
- [Fotos](#)
- [Comentários](#)

Clique em contatos por exemplo e veja se ele redireciona para a página de contatos:

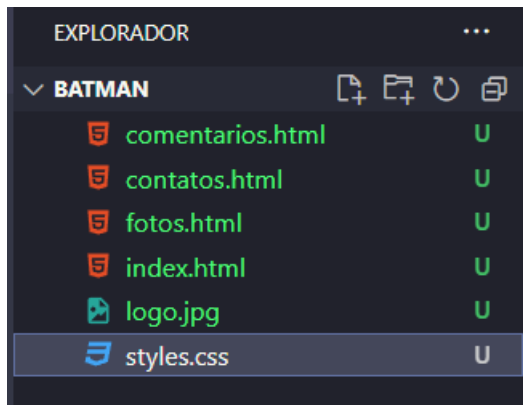


Contatos

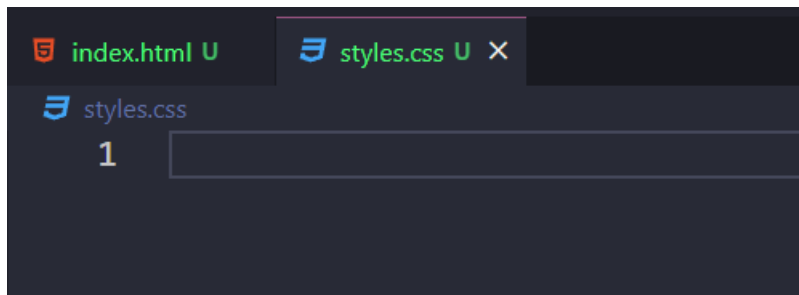
Faça isso com as demais páginas.

Agora que já temos o nosso header pronto no html, vamos começar a sua estilização. Abra o arquivo styles.css no seu vs code.

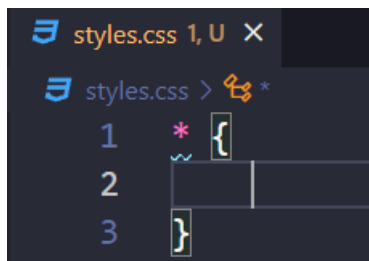




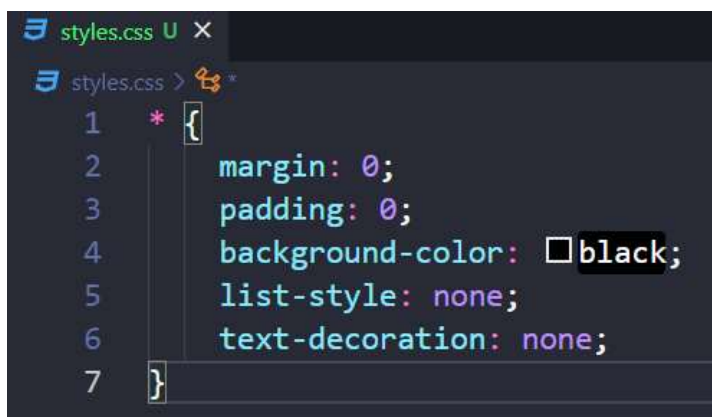
Basta clicar no arquivo e ele abrirá em uma nova aba.



A primeira coisa que iremos fazer é uma formatação geral para todo o nosso projeto. Para tanto utilizaremos asterisco `*` e chaves `{ }`.



Dentro dessas chaves iremos escrever o nosso código reset.



Uma observação. Talvez você tenha notado que a cor da fonte no seu vs code assim como ícones estão um pouco diferentes em relação ao que está

sendo apresentado na apostila. Fique tranquilo, isso são apenas extensões do vs code. Existem inúmeros tipos de extensões no vs code, cada uma com um objetivo específico. Durante o curso aprenderemos sobre essas extensões, então não se preocupe com isso por hora.

Vamos agora a explicação do porquê estamos usando cada um desses resets.

```
margin: 0; /*Remove todas as margens do nosso documento HTML*/
```

```
padding: 0; /*Remove todo o espaçamento interno dos elementos*/
```

```
background-color: black;  
/*Define a cor de fundo do documento inteiro como PRETO*/
```

```
list-style: none;  
/*Remove qualquer estilização das listas.  
Remove as "bolinhas" que tem padrão*/
```

```
text-decoration: none;  
/*Remove o sublinhado dos links (tags âncora)*/
```

Certo. Após fazer o reset, podemos prosseguir com a estilização no nosso header que deverá ficar assim:

```
header {  
  height: 90px;  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
}
```

Estilo da tag nav:

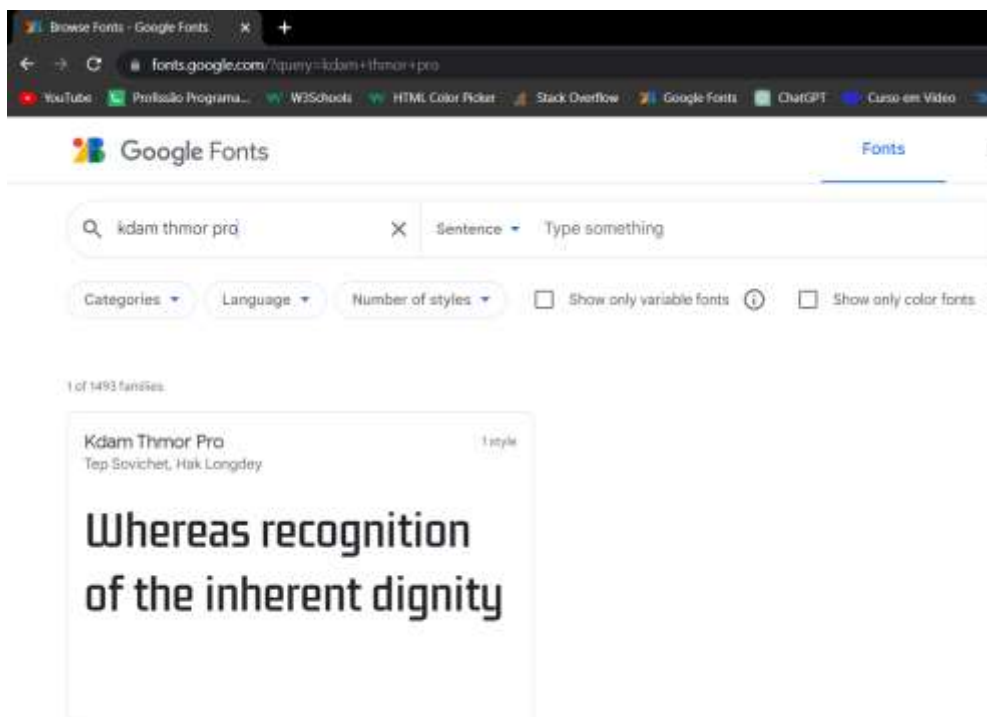
```
nav {  
  width: 60%;  
}
```

Estilo da tag ul:

```
ul {
  display: flex;
  width: 100%;
  justify-content: space-between;
}
```

Antes de irmos para o estilo dos nossos “li”, vamos primeiramente importar duas fontes que iremos usar no nosso projeto. Faremos a importação diretamente do google fonts.

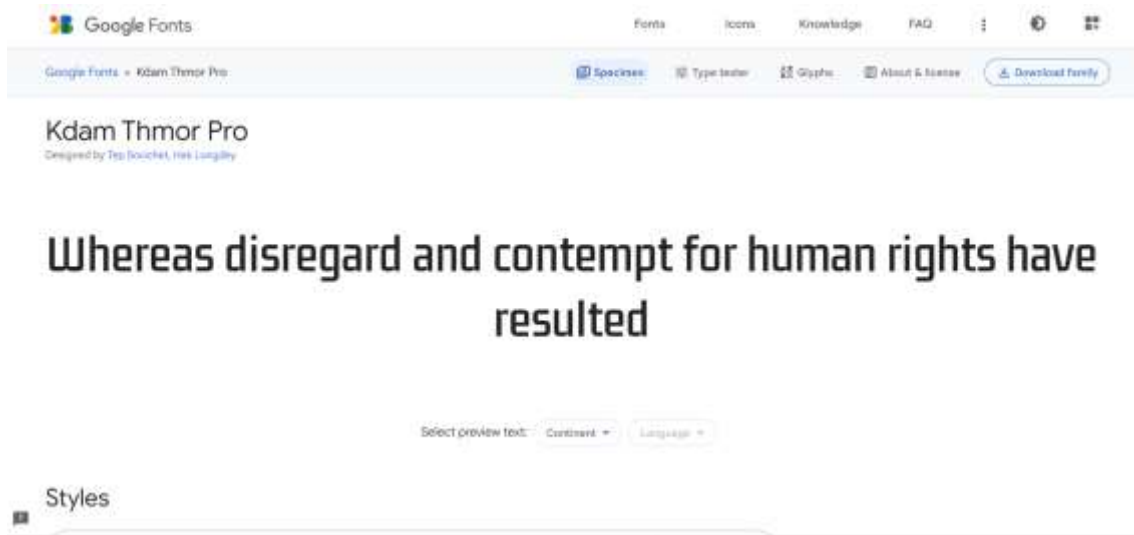
Acesse o site <https://fonts.google.com/> e na barra de pesquisa procure pela fonte kdam thmor pro conforme exemplo abaixo:



Clique na fonte kdam thmor pro:



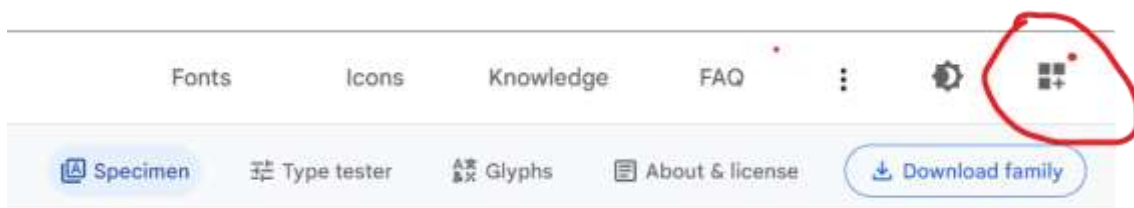
Ao clicar na fonte iremos para outra página onde vamos escolher qual a largura da fonte que queremos:



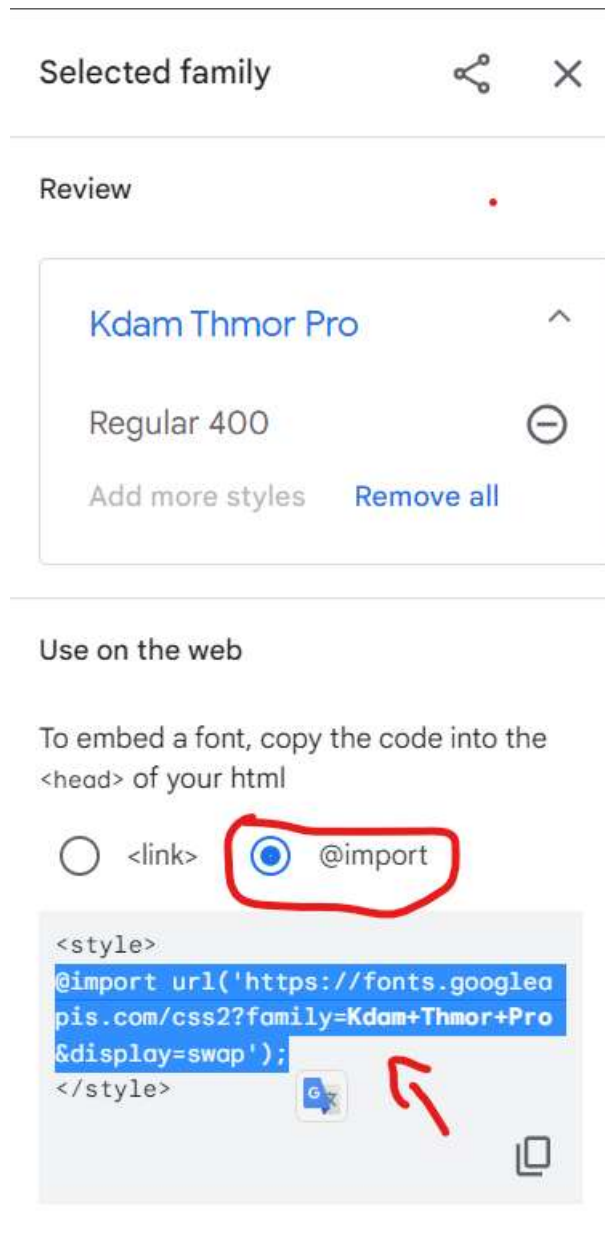
Role a página para baixo e escolha: Regular 400



Ao clicar na largura desejada abrirá um menu lateral onde vamos copiar a importação. Caso o menu não abra automaticamente, basta clicar no seguinte botão do cabeçalho:

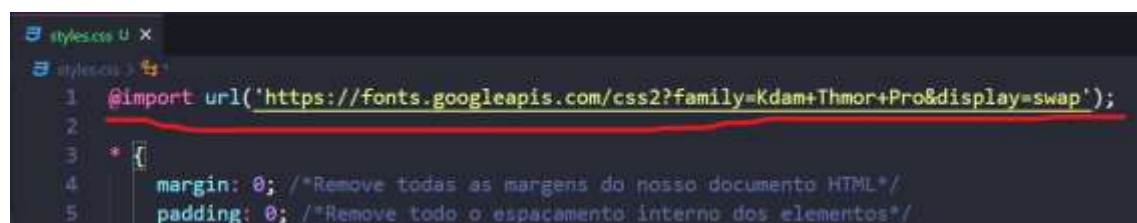


No menu, basta selecionar o @import e copiar o trecho selecionado no exemplo.



Note que **NÃO** é preciso copiar a tag <style> pois já estamos usando um arquivo css para estilizar o projeto. Copie apenas o trecho selecionado no exemplo.

Uma vez copiado @import, cole-o no topo do arquivo styles.css no vs code.



Dando continuidade à estilização do projeto:

Estilo das tags li:

```
li {  
  list-style: none;  
  color: ■ aliceblue;  
  cursor: pointer;  
  font-size: 19px;  
  transition: .5s;  
  font-family: "Kdam Thmor Pro", sans-serif;  
}
```

Note que em font-family já estamos chamando a fonte que acabamos de importar. Colocamos o nome da fonte entre aspas e após a vírgula inserimos uma fonte alternativa para caso essa fonte principal não funcionar por algum motivo, ela fica como segunda opção.

Também vamos adicionar um hover para os nossos “li” para que mudem de cor quando o mouse passar por cima deles:

```
li:hover {  
  color: ■ #D21617;  
}
```

Vamos usar uma cor hexadecimal aqui: #D21617

A última coisa que falta estilizar agora é o logo. Atribuímos um id a essa imagem e como aprendemos anteriormente o seletor de id é #.

Então:

```
#Logo {  
  width: 150px;  
  height: auto;  
}
```

Pronto. Isso é tudo que precisamos fazer no header. O resultado esperado é este a seguir:



Veja que a cor dos itens da navegação é branco e quando passar o mouse por cima fica essa cor avermelhada.

Para conferência, segue o css completo até agora:

```
1  @import url('https://fonts.googleapis.com/css2?family=Kdam+Thmor+Pro&display=swap');
2
3  * {
4      margin: 0; /*Remove todas as margens do nosso documento HTML*/
5      padding: 0; /*Remove todo o espaçamento interno dos elementos*/
6      background-color: black;
7      /*Define a cor de fundo do documento inteiro como PRETO*/
8      list-style: none;
9      /*Remove qualquer estilização das listas.
10     Remove as "bolinhas" que tem padrão*/
11     text-decoration: none;
12     /*Remove o sublinhado dos links (tags âncora)*/
13 }
14
15 header {
16     height: 90px;
17     display: flex;
18     justify-content: space-around;
19     align-items: center;
20 }
21
22 nav {
23     width: 60%;
24 }
25
26 ul {
27     display: flex;
28     width: 100%;
29     justify-content: space-between;
```

Continua na próxima página.



```

26  ul {
27      display: flex;
28      width: 100%;
29      justify-content: space-between;
30  }
31
32  li {
33      list-style: none;
34      color: ■ aliceblue;
35      cursor: pointer;
36      font-size: 19px;
37      transition: .5s;
38      font-family: "Kdam Thmor Pro", sans-serif;
39  }
40
41  li:hover {
42      color: ■ #D21617;
43  }
44
45  #Logo {
46      width: 150px;
47      height: auto;
48  }

```

## CRIANDO BANNER E TRAILER

A seguir vamos criar a seção onde ficará o nosso banner e também o trailer e a sinopse do filme. É muito importante que você pratique todo o conteúdo passado nas aulas para um melhor aprendizado.

Voltando no arquivo index.html, começaremos criando uma div com id="banner". Por hora isso é suficiente, pois iremos fazer o restante lá no CSS.

```

31  </header>
32
33  <div id="banner"></div>
34  _____
35  </body>

```



No arquivo styles.css, vamos chamar essa div por meio do id e dar uma background-image:

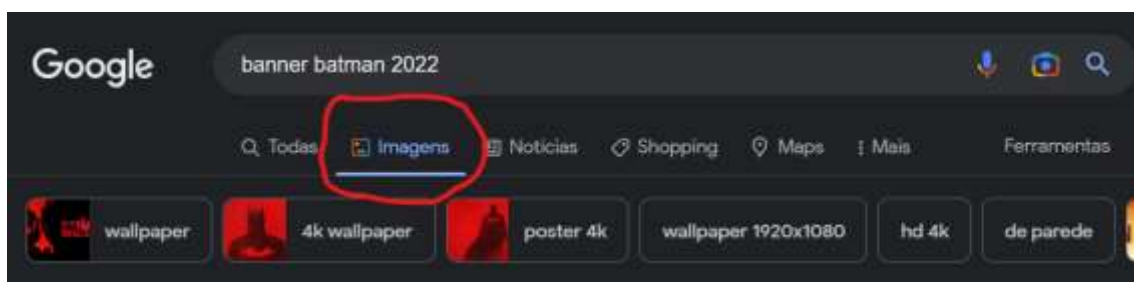
```
45 #Logo {  
46     width: 150px;  
47     height: auto;  
48 }  
49  
50 #banner {  
51     background-image: url();  
52 }
```

Por enquanto a url está vazia mas já vamos inserir uma url ai dentro dos parênteses.

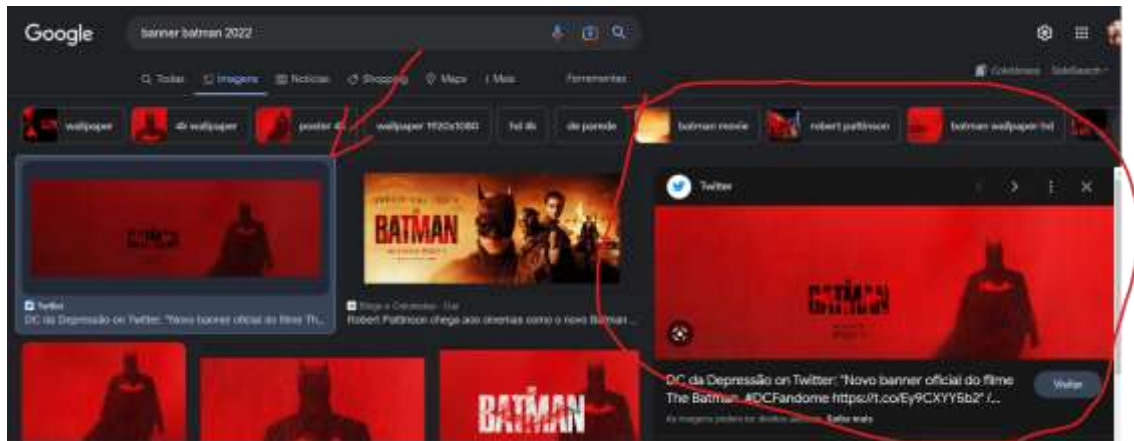
Pesquise no google por banner batman 2022:



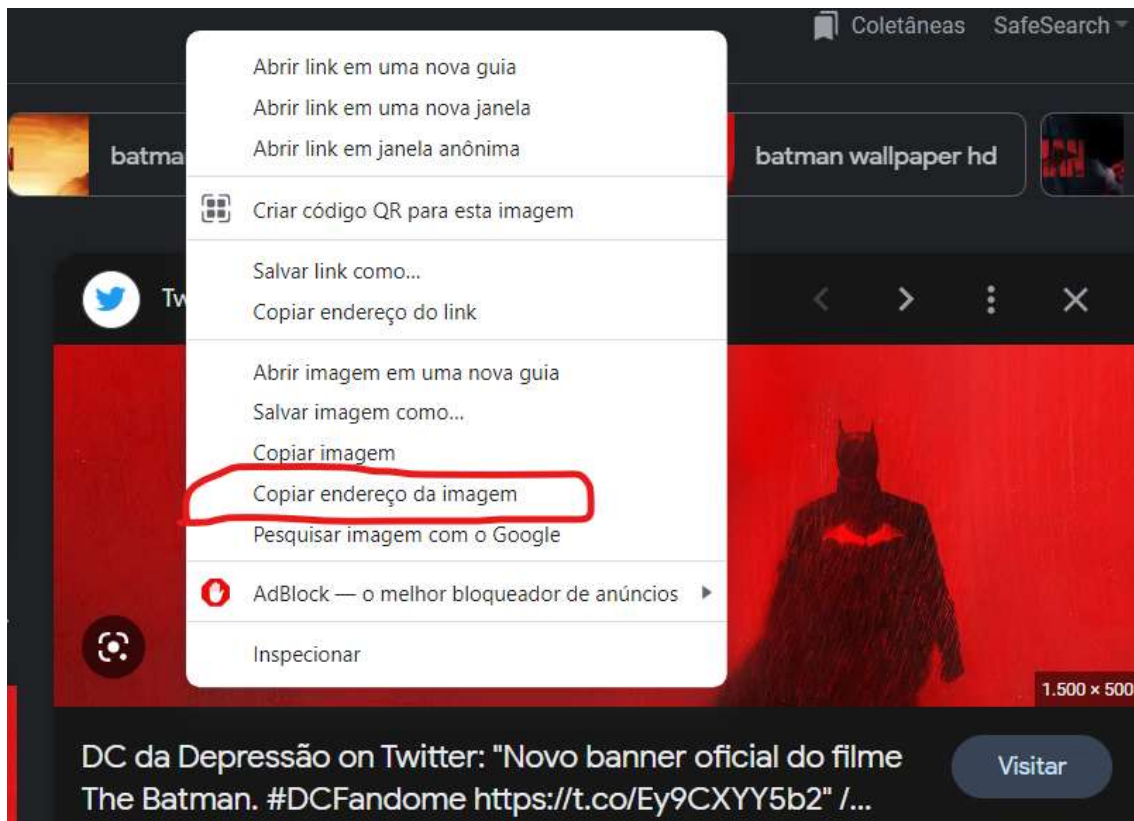
Clique na aba de imagens:



Clique na imagem escolhida e ela abrirá em um menu lateral:



Clique com o botão direito do mouse nessa imagem que foi aberta e clique em copiar endereço da imagem:



No vs code, vá no arquivo styles.css e cole esse endereço lá na url do background-image do nosso banner. Importante: o endereço deve estar entre aspas:

```
#banner {  
  background-image: url('');  
}
```

Deverá ficar assim:

```
#banner {  
  background-image: url('https://pbs.twimg.com/media/FBwJWeQXIAItfyH.jpg:large');  
}
```

Lembrando que estamos usando esse banner no exemplo mas você pode usar outro. Inclusive você pode usar até outro tema. Estamos fazendo um projeto baseado no filme Batman 2022 mas você pode fazer de outro filme, algum game ou série, etc.

Continuando:

Vamos dar mais alguns estilos ao nosso banner:

```
50 #banner {  
51   background-image: url('https://pbs.twimg.com/media/FBwJWeQXIAItfyH.jpg:large');  
52   background-position: 50% 50%;  
53   background-size: cover;  
54   width: 100%;  
55   height: 600px;  
56 }
```

O resultado deverá ficar assim:



OBS: Lembre-se sempre de salvar as alterações no vs code e atualizar a página no navegador.

Dando continuidade, lá no arquivo index.html criaremos mais uma div, agora com o id="trailer-container".

```
<div id="trailer-container">  
  
</div>
```

Dentro dessa div, criaremos outra div com uma classe="content".

```
<div id="trailer-container">  
  <div class="content">  
  
  </div>  
</div>
```

Dentro dessa div crie uma tag <vídeo> com uma classe="trailer" e também o atributo controls:

```
<div id="trailer-container">  
  <div class="content">  
    <video class="trailer" controls>  
  
    </video>  
  </div>  
</div>
```

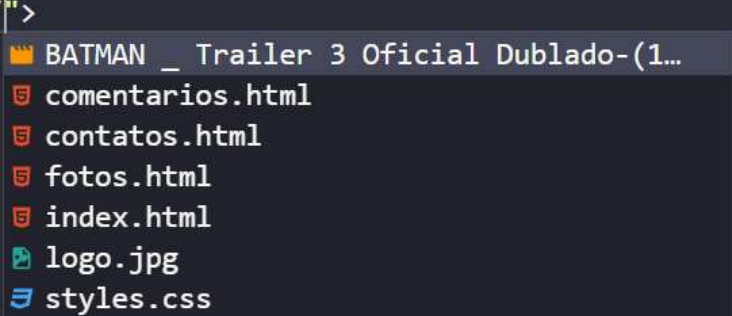
Dentro da tag <vídeo> adicione <source /> que deverá ter o atributo src:

```
<div id="trailer-container">  
  <div class="content">  
    <video class="trailer" controls>  
      <source src="._BATMAN _ Trailer 3 Oficial Dublado-(1080p).mp4">  
    </video>  
  </div>  
</div>
```

Veja que no src estamos colocando o nome do arquivo.

Como aprendido anteriormente, ao abrir aspas e digitar ./ o próprio vs code já te mostra as opções de arquivos presentes na pasta raiz do projeto, basta escolher o arquivo e teclar ENTER.

```
<div id="trailer-container">
  <div class="content">
    <video class="trailer" controls>
      <source src="./">
    </video>
  </div>
</div>
</body>
</html>
```



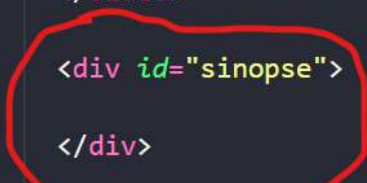
O bom de fazer dessa forma é evitar erros de digitação que farão com que o arquivo não possa ser acessado.

Também iremos adicionar ao source uma mensagem para caso aconteça do vídeo não ser renderizado.

```
<div id="trailer-container">
  <div class="content">
    <video class="trailer" controls>
      <source src="./BATMAN _ Trailer 3 Oficial Dublado-(1080p).mp4">
        Seu navegador não possui suporte para videos
      </video>
    </div>
  </div>
</div>
```

Logo abaixo da tag de vídeo, adicionaremos mais uma div com id="sinopse":

```
<video class="trailer" controls>
  <source src="./BATMAN _ Trailer 3 Oficial Dublado-(1080p).mp4">
    Seu navegador não possui suporte para videos
  </video>
  <div id="sinopse">
  </div>
```





Dentro dessa div colocaremos dois elementos. Uma tag <p> com uma classe="description" e abaixo dessa tag um <button> com uma classe="button".

```
<div id="sinopse">
  <p class="description">

  </p>
  <button class="button"></button>
</div>
```

Agora basta acrescentar os respectivos textos em ambos. Dentro de <p> vamos colocar a sinopse do filme (você pode encontrar alguma na internet ou copiar do exemplo. Lembre-se que a sinopse é um resumo curto. Se estiver fazendo o projeto com outro tema use a sinopse adequada para o seu tema)

O texto do nosso button será: Comprar ingresso

```
<div id="sinopse">
  <p class="description">
    Após dois anos espreitando as ruas como Batman,
    Bruce Wayne se encontra nas profundezas
    mais sombrias de Gotham City, com poucos
    aliados confiáveis, o vigilante solitário se
    estabelece como a personificação da vingança
    para a população.
  </p>
  <button class="button">Comprar ingresso</button>
```

Pronto. Agora vamos para o arquivo styles.css para podermos estilizar essa seção. Vamos chamar primeiramente a div com o id="trailer-container" e dar os seguintes estilos:

```
57 #trailer-container {
58   width: 100%;
59   height: 400px;
60   display: flex;
61   justify-content: center;
62 }
```

Agora chamamos a classe="content":

```
63  .content {
64    width: 60%;
65    display: flex;
66    align-items: center;
67    justify-content: space-between;
68  }
```

Note que quando chamamos pelo id usamos "#" e quando chamamos pela classe usamos "."

Estilo trailer:

```
69  .trailer {
70    width: 50%;
71  }
```

Estilo sinopse:

```
72  #sinopse {
73    width: 50%;
74    display: flex;
75    flex-direction: column;
76    justify-content: space-between;
77    margin-left: 30px;
78  }
```

Estilo description:

```
79  .description {
80    color: #fff;
81    font-size: 19px;
82    font-family: "Kdam Thmor Pro", sans-serif;
83    width: 90%;
84  }
```

Usamos aqui a mesma fonte que importamos anteriormente.

Caso você tenha problemas com o alinhamento do texto em relação ao trailer, você pode aumentar ou diminuir o width para ajustar conforme a necessidade. No caso do texto usado no exemplo, 90% ficou bom.

Estilo do button:

```
85  .button {
86    width: 200px;
87    height: 60px;
88    background-color: #D21617;
89    color: #fff;
90    padding: 8px 10px;
91    border-radius: 5px;
92    cursor: pointer;
93    transition: .8s;
94    font-size: 18px;
95    display: flex;
96    justify-content: center;
97    align-items: center;
98    font-weight: bold;
99    border: none;
100   outline: none;
101   margin-top: 25px;
102 }
```

O padding: 8px 10px; é usado dessa forma pois a primeira medida refere-se à encima e embaixo e a segunda medida refere-se à direita e esquerda como aprendido nas aulas passadas. Caso queira saber sobre algum estilo específico basta pesquisar no google e rapidamente você encontrará uma explicação sobre o que aquela propriedade faz na prática.

Vamos adicionar um efeito hover ao nosso button para que seu estilo seja alterado toda vez que o mouse passar por cima dele.

```
103  .button:hover {
104    background-color: #fff;
105    color: #000;
106  }
```

#000 equivale a cor preta (black) assim como #fff equivale a cor branca (White).



O resultado esperado é esse:



Caso o seu tenha ficado diferente, revise o seu código comparando com o exemplo, faça e refaça, até chegar no resultado esperado. Faz parte da rotina de um desenvolvedor encontrar solução para os problemas que surgem durante a construção de um projeto. Pratique e reproduza tudo o que está sendo passado para seu próprio aprendizado.

## CRIANDO CARDS

Criaremos a seguir a seção que conterá os cards com as fotos dos atores principais do filme.

Vamos para o arquivo index.html onde criaremos uma div com a classe="actor-cards-container"

```
<div id="sinopse">
  <p class="description">
    Após dois anos espreitando as ruas como Batman,
    Bruce Wayne se encontra nas profundezas
    mais sombrias de Gotham City, com poucos
    aliados confiáveis, o vigilante solitário se
    estabelece como a personificação da vingança
    para a população.
  </p>
  <button class="button">Comprar ingresso</button>
</div>
</div>
</div>

<div class="actor-cards-container">

</div>
```

Dentro dessa div criaremos uma outra div com a classe="cards-content"

```
<div class="actor-cards-container">
  <div class="cards-content">

  </div>
</div>
```

Agora dentro da div cards-content iremos adicionar 3 divs, uma para cada ator. O texto de cada div corresponde ao nome do ator e seu personagem. Ficará assim:

```
<div class="actor-cards-container">
  <div class="cards-content">

    <div class="card banner1">
      Robert Pattison - Batman
    </div>

    <div class="card banner2">
      Zoe Kravitz - Mulher gato
    </div>

    <div class="card banner3">
      Jeffrey Wrigth - Gordon
    </div>

  </div>
</div>
```

Aqui tem um detalhe muito legal sobre classes. Podemos adicionar duas classes diferentes para o mesmo elemento. Basta ter um espaço entre as classes como fizemos acima. Veja no detalhe:

```
<div class="card banner1">
```

Observe que "card" corresponde à uma classe e "banner1" é a segunda classe desse mesmo elemento.

## Por que fazer dessa forma?

Fazemos assim pois quando estilizarmos a classe card, isso afetará todos os 3 cards que criamos, pois todos eles tem a mesma classe "card"

```
<div class="card banner1">  
  Robert Pattison - Batman  
</div>  
  
<div class="card banner2">  
  Zoe Kravitz - Mulher gato  
</div>  
  
<div class="card banner3">  
  Jeffrey Wrigth - Gordon  
</div>
```

E ao utilizarmos a segunda classe podemos alterar o estilo de cada card individualmente. Por isso temos classes diferentes para cada um dos cards.

```
<div class="card banner1">  
  Robert Pattison - Batman  
</div>  
  
<div class="card banner2">  
  Zoe Kravitz - Mulher gato  
</div>  
  
<div class="card banner3">  
  Jeffrey Wrigth - Gordon  
</div>
```

Também poderíamos fazer isso atribuindo uma mesma classe para todos elementos e um id diferente para cada um deles. Assim, quando quiséssemos estilizar todos de uma vez usaríamos a classe, e quando quiséssemos estilizar individualmente usaríamos o id do elemento específico. Nesse exemplo usamos duas classes para que você aprenda que é possível fazer dessa forma.

Isso é tudo que precisamos fazer por hora no HTML. Vamos para o arquivo styles.css para estilizarmos esses novos elementos.

Começando pelo estilo do nosso actor-cards-container:

```
107  .actor-cards-container {  
108      width: 100%;  
109      display: flex;  
110      justify-content: center;  
111      margin-top: 50px;  
112  }
```

Em seguida cards-content:

```
.cards-content {  
    width: 90%;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    row-gap: 20px;  
    column-gap: 40px;
```

Usamos display: grid; aqui. Se você não se lembra muito bem pra que serve ou como funciona, volte no módulo 01 que lá tem uma aula específica sobre isso. Aqui usamos 3 colunas, por isso: repeat(3, 1fr) ou seja, repita 1 coluna 3 vezes.

row-gap: espaço entre linhas

column-gap: espaço entre colunas

Prosseguimos com o estilo da classe card:

```
.card {  
  width: 100%;  
  height: 600px;  
  background-size: cover;  
  background-position: 50% 50%;  
  border-radius: 5px;  
  cursor: pointer;  
  transition: 1s;  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-end;  
  padding: 0 0 20px 10px;  
  color: #fff;  
  font-size: 17px;  
  font-family: "Kdam Thmor Pro", sans-serif;  
}
```

Background-size: cover; é usado para que a imagem de fundo ocupe todo o espaço do card.

Como estamos usando um flex-direction: column, a direção dos elementos dentro dessa div será de cima para baixo e não da esquerda para direita como é o padrão. Então quando colocamos o justify-content como flex-end, significa que o texto ficará no final, ou seja, na parte de baixo.

Vamos agora fazer a estilização de cada card individualmente por meio da segunda classe que adicionamos a eles:

```
.banner1 {  
  background-image: url('');  
}  
.banner2 {  
  background-image: url('');  
}  
.banner3 {  
  background-image: url('');  
}
```

A princípio não tem nenhuma url. Isso será explicado a seguir.

Você ficará responsável por colocar as urls em cada um dos cards. Lembre-se que nós já fizemos esse mesmo processo com o banner. Então, caso precise recordar é só voltar naquela parte e seguir o passo a passo. Basicamente você vai procurar a foto do ator no google imagens, abrir a foto desejada e clicando com o botão direito sobre a imagem aberta copie o endereço da imagem. Após copiar, basta ir no css e colar esse endereço entre as aspas de background-image:

```
.banner1 {  
  background-image: url('https://preview.redd.it/  
  robert-pattinson-is-the-best-live-action-versio  
  n-of-batman-v0-5quh0wv3nzp91.jpg?width=640&  
  crop=smart&auto=webp&  
  s=e83b3194079005ec2d5bbd1474347083aad4ed18');  
}
```

Faça isso para cada um dos banners:

```
.banner1 {  
  background-image: url('https://preview.redd.it/  
  robert-pattinson-is-the-best-live-action-versio  
  n-of-batman-v0-5quh0wv3nzp91.jpg?width=640&  
  crop=smart&auto=webp&  
  s=e83b3194079005ec2d5bbd1474347083aad4ed18');  
}  
.banner2 {  
  background-image: url('https://  
  ellebrasil-wp-images.s3.amazonaws.com/2022/08/  
  origin-118.jpg');  
}  
.banner3 {  
  background-image: url('https://2.bp.blogspot.  
  com/-N9DtMsBCAwI/XbtdLT8foBI/AAAAAACzVU/  
  bTyfrBq-ApwVG8DUfRmWYV1vKyzXJ-08wCK4BGAYYCw/  
  s1600/jeffreywrights.jpg');  
}
```

Não esqueça de colocar a url entre aspas.

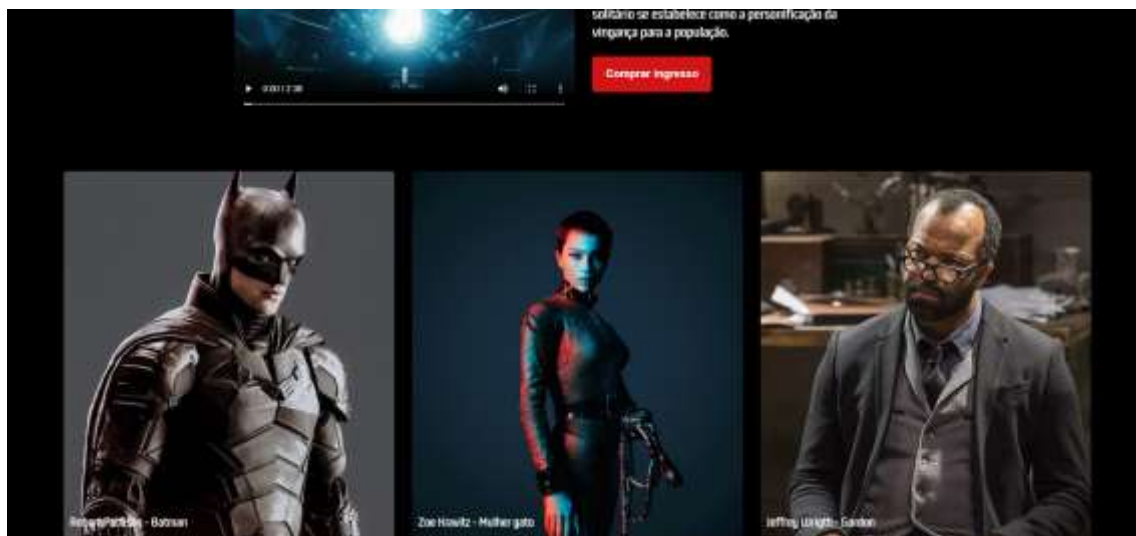


Você pode escolher as fotos que achar melhor. Não necessariamente precisam ser as do exemplo. A única coisa que você deve se atentar é que existe uma ordem que nós estabelecemos. Então em “banner1” deve ser a foto do Batman, em “banner2” deve ser a foto da Mulher gato e em “banner3” a foto do Gordon.

A última coisa a ser feita, é adicionar um hover a nossa classe “card” para que quando o mouse passar por cima dos cards ele aumente um pouco de tamanho. Faremos assim:

```
.card:hover {  
  transform: scale(1.02);  
}
```

Esse é o resultado esperado para essa seção:



## CRIANDO O FOOTER

Iremos implementar a ultima seção do nosso projeto que é o footer, ou rodapé. Ele terá basicamente o mesmo logo do header, um texto de copyright e o nome do desenvolvedor. Também terá uma navegação que por hora iremos deixar oculta e só vamos estiliza-la na hora de fazer a responsividade do projeto (adaptar para dispositivos móveis: celulares, tablets).

Começaremos como sempre, adicionando as tags lá no arquivo index.html.

Adicione a tag footer logo abaixo da seção de cards:

```
67     <div class="card banner3">
68         Jeffrey Wrigth - Gordon
69     </div>
70
71 </div>
72 </div>
73
74 <footer>
75
76 </footer>
77
```

Vamos acrescentar o logo dentro de footer através da tag img:

```
<footer>
  
</footer>
```

Veja que a tag img geralmente abre e fecha em si mesma.

Aqui atribuímos o estilo diretamente pra facilitar. Basicamente esse estilo é para que a imagem de adegue no melhor tamanho sem perder a qualidade.

Colocamos id="logo" pois ela terá os mesmos estilos do header.

E o src é para localizarmos a imagem logo dentro da pasta como já vimos anteriormente. Lembre que só de colocar "./" o vs code já te mostra quais arquivos estão presentes na pasta raiz.

Vamos agora colocar o primeiro texto:

```
<footer>
  
  <span>Todos os direitos reservados &copy;</span>
</footer>
```



Repare que estamos usando &copy;

Isso se chama esc2. É uma forma de adicionarmos caracteres em formato de imagens. Assim, nós teremos o símbolo de copyright no site.

Ficará assim na prática:



No seu projeto talvez você ainda não consiga ver, pois precisamos primeiro estilizar o footer lá no css.

Vamos adicionar mais um texto ao footer:

```
<footer>
  
  <span>Todos os direitos reservados &copy;</span>
  <span>Desenvolvido por: Coloque seu nome aqui</span>
</footer>
```

Substitua a parte em vermelho pelo seu nome.

Agora vamos colocar a navegação que será a mesma do header. Então basta copiar e colar a mesma navegação do header no footer:

```
<footer>
  
  <span>Todos os direitos reservados &copy;</span>
  <span>Desenvolvido por: Paulo Borges</span>

  <nav class="footer-navigation">
    <ul class="footer-list">
      <a href="./index.html">
        <li>Home</li>
      </a>
      <a href="./contatos.html">
        <li>Contatos</li>
      </a>
      <a href="./fotos.html">
        <li>Fotos</li>
      </a>
      <a href="./comentarios.html">
        <li>Comentários</li>
      </a>
    </ul>
  </nav>
</footer>
```

A única mudança em relação a essa navegação é que ela terá uma classe="footer-navigation" que vamos usar para deixar essa navegação oculta para desktop (computador) e disponível apenas no mobile (dispositivos móveis) e também a ul que terá a classe="footer-list":

```
<nav class="footer-navigation">
  <ul class="footer-list">
    <a href="./index.html">
      <li>Home</li>
    </a>
    <a href="./contatos.html">
      <li>Contatos</li>
    </a>
    <a href="./fotos.html">
      <li>Fotos</li>
    </a>
    <a href="./comentarios.html">
      <li>Comentários</li>
    </a>
  </ul>
</nav>
```

Pronto. Vamos para o styles.css estilizar esses novos elementos.

Começando pelo footer:

```
148 footer {
149   height: 300px;
150   margin-top: 50px;
151   display: flex;
152   flex-direction: column;
153   align-items: center;
154   justify-content: space-around;
155 }
```

Chamamos o elemento diretamente pelo nome da tag pois só temos um footer no projeto então não há necessidade de ter uma classe para ele. Logo não colocamos "." antes de chamar no css, já chamamos o nome da tag diretamente.

Em seguida, daremos estilo aos textos:

```
span {  
  color: #fff;  
  font-family: "Kdam Thmor Pro", sans-serif;  
}
```

Também já chamamos diretamente pelo nome da tag pois assim estilizamos todos os textos do footer de uma só vez.

Agora estilizaremos as classes da navegação:

```
.footer-navigation {  
  display: none;  
}  
.footer-list {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```

Perceba que colocamos `display: none;` na classe “`footer-navigation`”. Isso significa que ela ficará oculta. Mas então por que criamos essa navegação se ela ficará oculta? Ela ficará oculta apenas quando o projeto for acessado pelo computador. Quando for acessada pelo celular por exemplo ela ficará disponível.

Como “`footer-list`” fica dentro de “`footer-navigation`” então não precisamos atribuir `display: none;` novamente, pois ela já ficará oculta. Então podemos colocar os estilos nessa classe normalmente que quando o elemento “pai” que no caso é a navegação ficar disponível, ela também ficará. O resultado esperado para o footer é o seguinte:



No detalhe:



## TRABALHANDO A RESPONSIVIDADE

Estamos quase concluindo a home page do nosso projeto. Só falta fazermos as devidas adaptações no css para que o nosso projeto seja responsivo. Já falamos sobre o conceito mobile first, que tem sido amplamente adotado pelo mercado. Hoje em dia a maioria das pessoas acessam sites através do celular, então é muito importante que qualquer projeto desenvolvido por você seja totalmente responsivo. Para tanto, iremos usar a ferramenta do css chamada media queries.

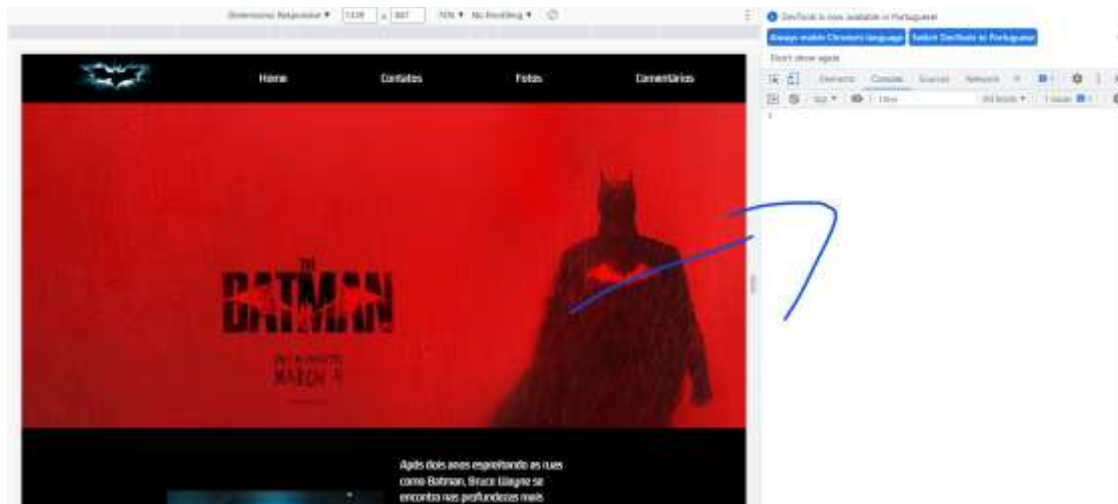
No arquivo styles.css, vamos criar dessa forma:

```
168
169  @media (max-width: 768px) {
170
171  }
```

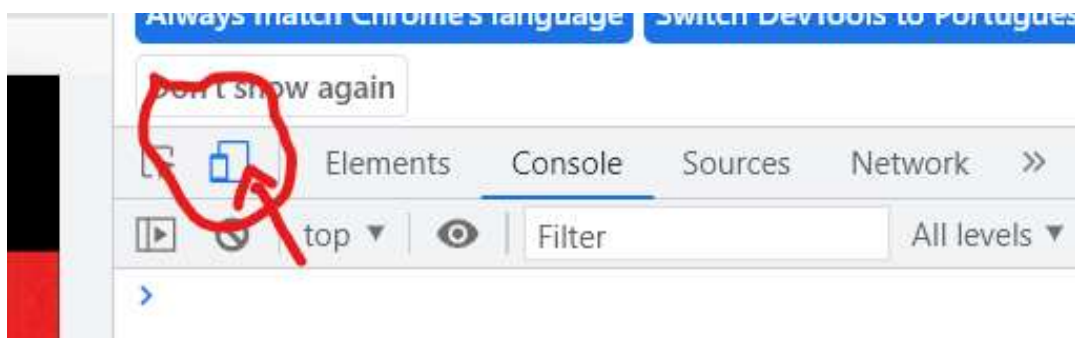
Observe que entre parênteses estamos estabelecendo uma regra: max-width: 768px. Isso significa que os estilos que forem adicionados entre chaves { } só serão válidos caso a largura do

dispositivo seja de 768px ou inferior. Então se a largura for superior a essa medida, o css adotado será o que criamos nas aulas anteriores. Porém, se a largura do dispositivo for igual ou inferior, o css adotado será esse que vamos acrescentar dentro de @media. Essa regra pode ser manipulada de acordo com a necessidade.

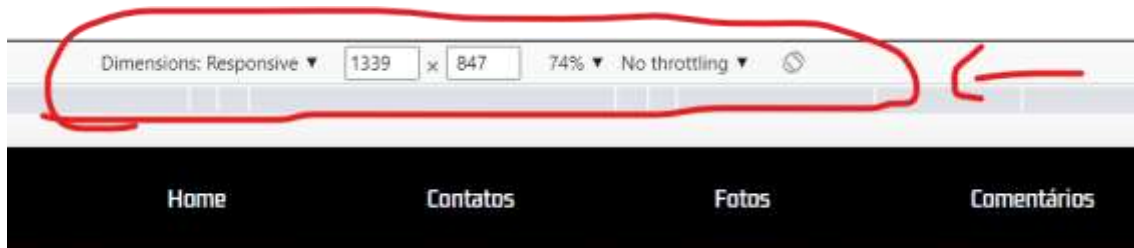
Existe uma ferramenta no navegador Google Chrome que irá nos ajudar bastante durante o processo de adaptação do projeto para mobile. Abra o seu projeto no navegador Google Chrome e digite o atalho: Ctrl + shift + i para abrir as ferramentas de desenvolvedor.



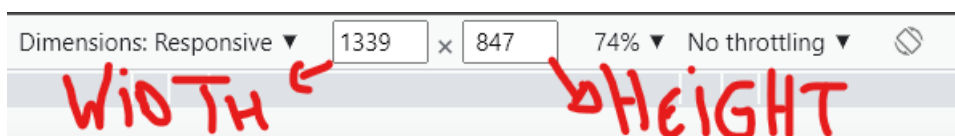
Nessa parte o botão toogle device toolbar deve estar selecionado. Ele fica com uma cor azulada:



Com ele selecionado, deverá aparecer a barra de tarefas do mobile:



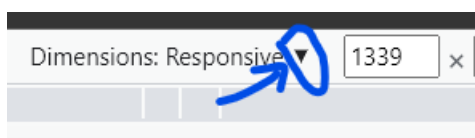
A medida da esquerda é a largura e a medida da direita é a altura em pixels (px).



Você pode usar a barra lateral para ir mudando o tamanho:

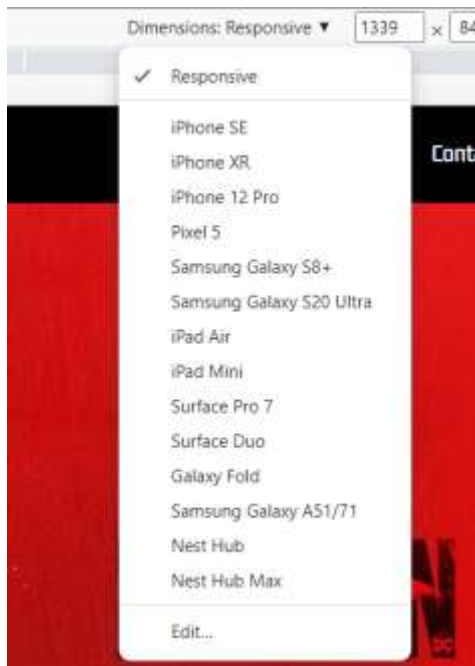


Ou mudar o tipo de dispositivo clicando na seta dimensions:





Você pode escolher qualquer um desses dispositivos que variam de celulares a tablets, cada um com sua medida:



Vamos usar como exemplo o iphone 12 pro:



Perceba que nosso projeto ao ser aberto em um celular fica com o estilo todo “quebrado”. Então vamos começar a adaptação. Voltando para o css, a primeira coisa que vamos adaptar é a classe footer-navigation:

```
@media (max-width: 768px) {  
  .footer-navigation {  
    display: block;  
  }  
}
```

Se você atualizar o projeto no navegador verá que a navegação que antes estava ausente no footer, agora está disponível:



A segunda coisa é remover a navegação do header. Pois no mobile nós teremos a navegação no footer, então não há necessidade de tela no header:

```
@media (max-width: 768px) {  
  .footer-navigation {  
    display: block;  
  }  
  nav {  
    display: none;  
  }  
}
```



No mobile, nosso header terá apenas o logo centralizado:



Agora vamos estilizar o banner por meio do seu id para que ele também fique centralizado:

```
@media (max-width: 768px) {  
  .footer-navigation {  
    display: block;  
  }  
  nav {  
    display: none;  
  }  
  #banner{  
    background-position: 40%;  
  }  
}
```



Vamos também adaptar o nosso trailer-container por meio do id:

```
#trailer-container {  
  width: 100%;  
  height: 550px;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```

O flex-direction: column, faz com que um elemento fique embaixo do outro e não mais lado a lado.

E também o content, só que por meio da classe:

```
.content {  
  width: 95%;  
  display: flex;  
  flex-direction: column;  
}
```

Vamos estilizar agora o trailer e a sinopse:

```
.trailer {  
  width: 100%;  
}
```

```
#sinopse {  
  width: 95%;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: space-between;  
}
```

Também a description:

```
.description {  
  width: 100%;  
}
```

Agora o button:

```
.button {  
  width: 100%;  
  margin-top: 30px;  
}
```

E por fim o cards-content:


```
.cards-content {  
  grid-template-columns: 1fr;  
}
```

Aqui não precisamos mais ter 3 colunas pois estamos na versão module. Então por isso estamos usando apenas 1fr, sem usar o repeat pois assim teremos uma única coluna e os cards ficarão um abaixo do outro.

Bom, a nossa página inicial já está totalmente adaptada para mobile.

Veja que foi simples, mas nós ainda vamos evoluir bastante. Criamos esse projeto até agora com apenas html e css mas vamos futuramente aprender a criar com Javascript tanto o front-end como o back-end.


A seguir estará o resultado esperado para a responsividade.



▶ 0:00 / 2:38

Após dois anos espreitando as ruas como Batman, Bruce Wayne se encontra nas profundezas mais sombrias de Gotham City, com poucos aliados confiáveis, o vigilante solitário se estabelece como a personificação da vingança para a população.

[Comprar ingresso](#)



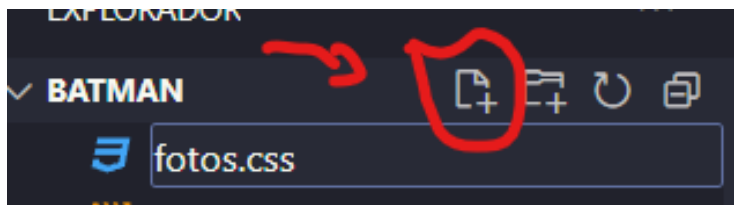
Caso o seu tenha ficado diferente revise o código a partir da apostila. Procura atentar-se aos detalhes. Também é importante ir testando soluções, modificando as medidas até obter o resultado desejado. A seguir você terá alguns desafios a cumprir.

Nesse projeto, desenvolvemos juntos apenas a página home. Você terá o desafio de criar as demais páginas que por enquanto estão em branco. A página de fotos, a página de contato e a página de comentários. A melhor maneira de você aprender de fato é praticando. Use sua criatividade e construa páginas bonitas assim como a home.

## DICAS

Você pode copiar o mesmo header e o mesmo footer para as outras páginas e assim manter um padrão.

Uma boa prática é ter um arquivo css para cada página, por exemplo: contatos.css, fotos.css e comentarios.css



Não se esqueça de linkar o devido arquivo css em cada uma das páginas:

```
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="./fotos.css">
9 </head>
10 <body>
11   fotos
12 </body>
13 </html>
```

Faça isso em todas as outras páginas. Crie um arquivo css e faça o link dele na sua respectiva página.

Em cada umas das páginas você pode copiar o reset que fizemos e também a importação da mesma fonte:

```
1 @import url('https://fonts.googleapis.com/css2?family=Kdam+Thmor+Pro&display=swap');
2
3 *{
4   margin: 0;
5   padding: 0;
6   background-color: black;
7   list-style: none;
8   text-decoration: none;
9 }
10
```

É recomendado usar a mesma fonte em todo o projeto para manter um padrão.

## SITES E FERRAMENTAS

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



Nesse site você encontra tutoriais sobre HTML e CSS, pode realizar exercícios para praticar e muitas outras ferramentas. O site está em inglês mas você pode usar a ferramenta do navegador e traduzir.



<https://developer.mozilla.org/pt-BR/docs/Web/HTML>

A documentação oficial do HTML.

<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

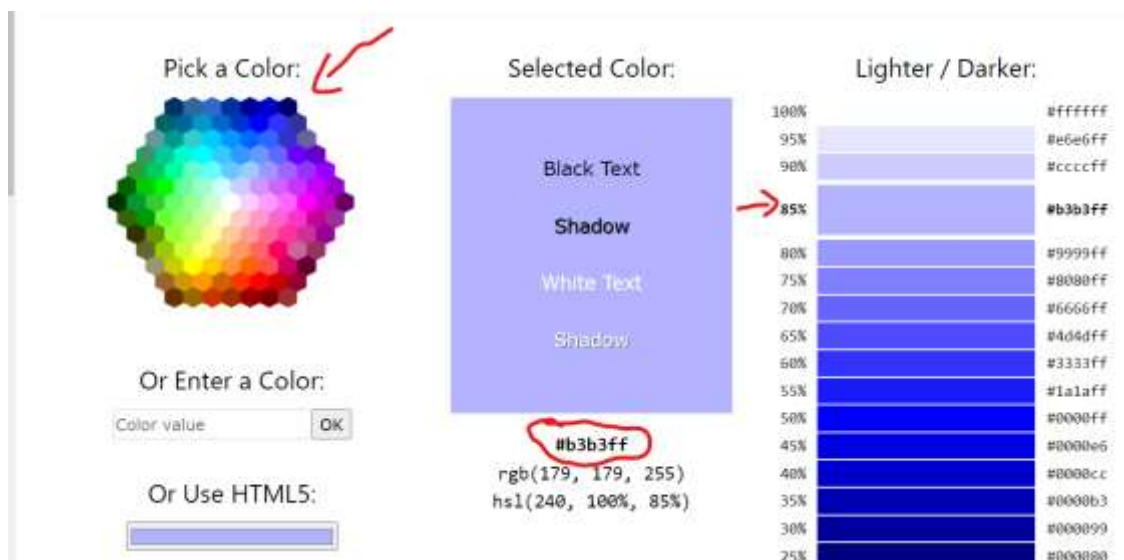
A documentação do CSS.

Vimos muita coisa durante essas aulas sobre HTML e CSS mas não vimos tudo, isso seria impossível. Caso queira, você pode pesquisar e se aprofundar mais, essas são as documentações.

## CORES:

[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)

No próprio site da w3schools temos uma ferramenta que nos ajuda com as cores. Basta escolher a cor desejada e sua tonalidade que ele já nos dá o código hexadecimal daquela cor para podermos usar no nosso código:



## MODELOS:

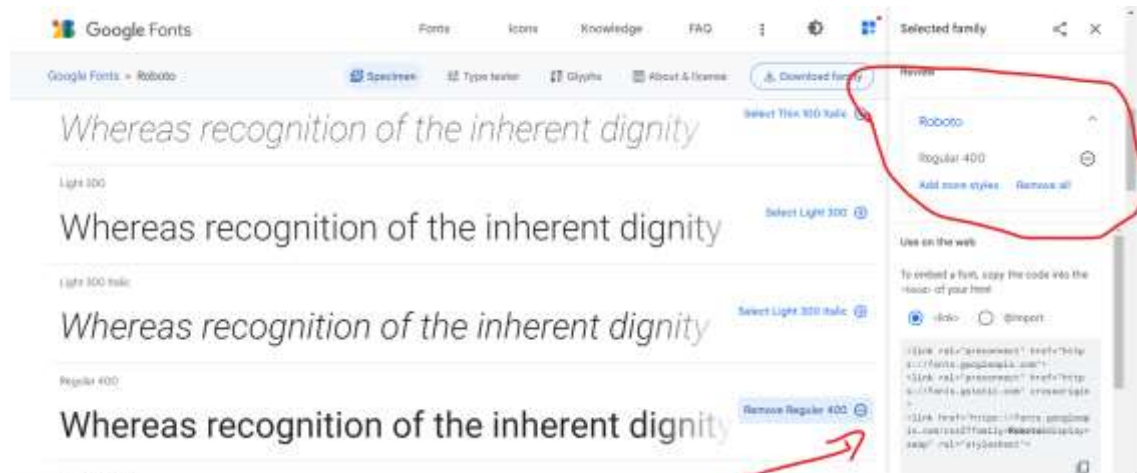
<https://br.pinterest.com/>

Esse você já deve conhecer. Basta acessar, criar sua conta e pesquisar por UI Design ou Landing Page e se inspirar nesses modelos para criar o seu projeto. Não é errado fazer isso. No mundo da programação é normal criar projetos que são parecidos com outros.

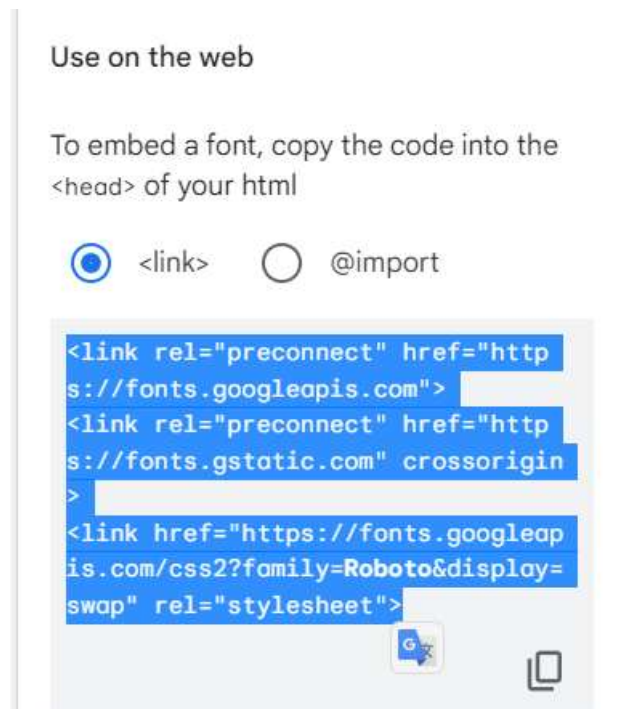
## FONTES:

<https://fonts.google.com/>

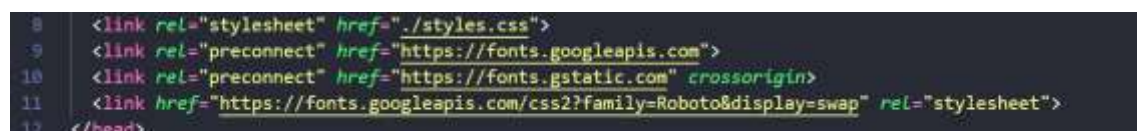
Já utilizamos o Google Fonts anteriormente e é bem simples. Você pesquisa por alguma fonte e adiciona ela ao menu:



Em seguida é só colocar dentro do HTML ou CSS. Se for HTML basta copiar o link:



E colar dentro das meta tags:



Se for no CSS é só copiar o @import e colar no início do arquivo css como fizemos ao longo desse projeto:

☐ <link>    ☒ @import

```
<style>
@import url('https://fonts.googlea
pis.com/css2?family=Roboto&display
=swap');
</style>
```

Com essas ferramentas você já conseguirá criar as demais páginas desse primeiro projeto prático e também criar outros projetos por conta própria.

## CONCLUSÃO

Dev, não pule etapas. Um erro que a maioria das pessoas cometem é pular os projetos práticos e focar só na teoria. Entenda algo de cara, programação é prática! Não pule os projetos práticos nem os seus desafios. Tudo isso foi colocado no curso para o seu aprendizado. Sua tarefa agora é concluir o projeto criando as demais páginas e também fazer a responsividade das mesmas, adaptando-as para mobile.

Após concluir tudo, você pode demonstrar mandando fotos e vídeos do seu projeto no whatsapp do nosso suporte:

85 8146-9819

E também marcando @pauloborges.tech no instagram com as hashtags #projeto pratico1 e #profissaoprogramador

Boa sorte e foco nos estudos!