

Índice

CSS	2
Conceitos fundamentais	2
CSS Selectors	3
Propriedades	4
Elementos Semânticos	4
CSS - Uso	6
Início	6
#Id	7
.Class	8
Background-image	9
Ajustando posições com FlexBox	10
Margin	12
Unidades de medida	12
Botão	13
Cor no botão	13
Botão com animação de rotação	14
Comandos	15
Boas Práticas	16
Padronização de nomenclatura	20
Links e afins	21
Dúvidas	22

Assuntos: HTML Avançado

Abertura Alunos:

Apresentação pitch

06/02: Ana Eliza e Verônica Freitas.

07/02: Eduardo

CSS

(*Cascading Style Sheets* ou Folhas de Estilo em Cascata) é um mecanismo para adicionar estilo a um documento web. O código CSS pode ser aplicado diretamente nas tags ou ficar contido dentro das tags `<style>`. Também é possível, em vez de colocar a formatação dentro do documento, criar um link para um arquivo CSS que contém os estilos.

Esse módulo inicia com o funcionamento básico do CSS, incluindo seletores e propriedades, escrever regras CSS, aplicar CSS ao HTML, como especificar comprimento, cor, e outras unidades em CSS

Em CSS, o código é lido em cascata, ou seja, em ordem de escrita, de cima para baixo. Ele fica contido nas tags `<style>` em HTML

CSS, que significa *Cascading Style Sheets* (Folhas de Estilo em Cascata), é uma linguagem de estilo usada para definir a apresentação de documentos HTML (ou XML). Em termos mais simples, o CSS é o que dá estilo e design a uma página da web, como cores, fontes, margens, espaçamento e posicionamento de elementos.

Conceitos fundamentais

Seletores: São padrões usados para selecionar os elementos HTML aos quais você deseja aplicar estilos. Por exemplo, você pode usar o seletor de elemento para selecionar todos os parágrafos `<p>` ou um seletor de classe para selecionar elementos com uma determinada classe.

Propriedades e Valores: As propriedades são os aspectos específicos do estilo que você deseja alterar, como cor, tamanho da fonte, margem, etc. Os valores são os ajustes específicos para essas propriedades, como 'vermelho' para a cor ou '12px' para o tamanho da fonte.

Caixas: Em CSS, todos os elementos são representados como caixas retangulares, e você pode controlar as dimensões e o layout dessas caixas usando propriedades como `width`, `height`, `margin`, `padding`, `border`, etc.

Modelo de Caixa: Refere-se à forma como o tamanho total de um elemento é calculado, incluindo o conteúdo, o preenchimento (`padding`), a borda (`border`) e a margem (`margin`).

Flexbox e Grid: São técnicas de layout mais avançadas que permitem criar layouts complexos e responsivos com facilidade.

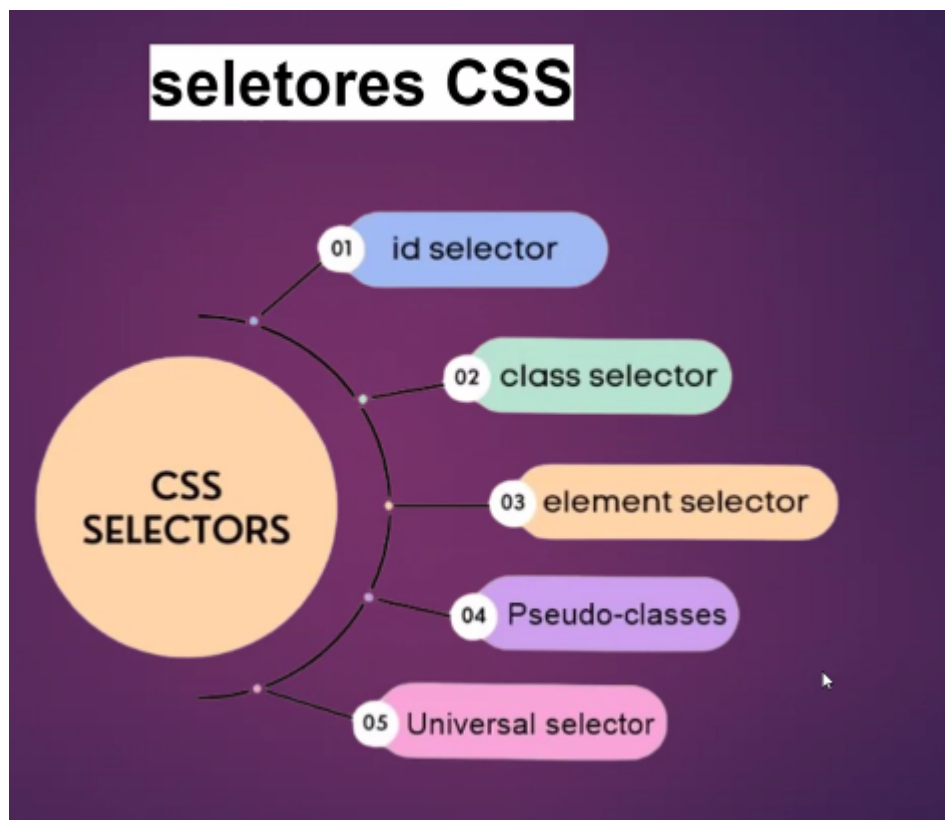
Responsividade: Refere-se à capacidade de criar páginas da web que se ajustam e são visualmente agradáveis em diferentes dispositivos e tamanhos de tela.

Media Queries: Permitem aplicar estilos específicos com base em características do dispositivo, como largura da tela, tipo de dispositivo, orientação, etc.

Transições e Animações: São recursos que permitem criar efeitos de transição suaves e animações em elementos HTML usando CSS.

CSS Selectors

Id, Class, Element, Pseudo Class, Universal



ID: Um ID é um seletor exclusivo para um elemento específico na página. Ele é definido no HTML usando o atributo id. Em CSS, você pode selecionar um elemento com um ID específico prefixando-o com

Classe: Uma classe é um seletor que pode ser aplicado a vários elementos na página. Ele é definido no HTML usando o atributo class. Em CSS, você pode selecionar elementos com uma classe específica prefixando-a com “.”.

Elemento: Um seletor de elemento seleciona todos os elementos de um tipo específico. Por exemplo, p seleciona todos os parágrafos na página. Não há prefixo especial para seletores de elementos.

Pseudo Classe: As pseudo classes são usadas para selecionar elementos com base em um estado específico que não está presente no DOM. Por exemplo, :hover seleciona um elemento quando o cursor do mouse está sobre ele. Não há prefixo especial para pseudo classes.

Seletor universal: Ele é usado para selecionar todos os elementos dentro de um documento HTML. Isso significa que quando você aplica o seletor universal em uma regra de estilo, ela afetará todos os elementos na página, a menos que sejam especificamente anulados por outras regras de estilo.

Propriedades

As propriedades são características visuais que os elementos html possuem. Uma declaração pode conter uma ou mais propriedades, cada uma com seu respectivo valor e separada por ponto e vírgula.

#	.	:	[]	*
ID	CLASSE	ESTADO	ATRIBUTO	TOTAL

Exemplo de uso de CSS

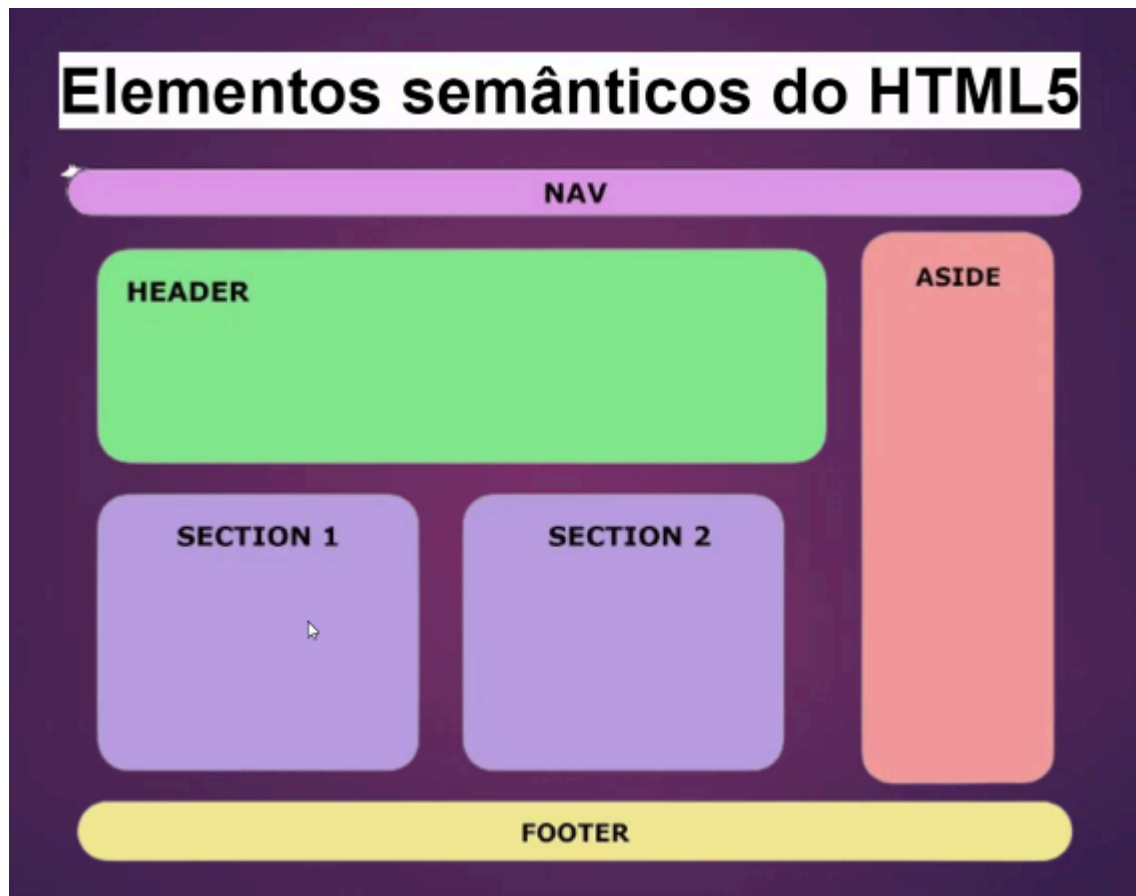
```
body {background-color : rgb (255,0,0);}
```

Seletor	Propriedade	Valor
body	{background-color:	rgb(255,0,0);}

Sempre vai ser seletor, propriedade e valor.

Elementos Semânticos

Elementos semânticos do HTML5



HTML5 introduziu uma série de elementos semânticos que ajudam a estruturar e descrever o conteúdo de uma página web de uma maneira mais significativa. Esses elementos fornecem informações adicionais sobre o conteúdo da página, facilitando a compreensão tanto para os desenvolvedores quanto para os motores de busca. Aqui estão alguns dos elementos semânticos mais comuns em HTML5:

<header>: Define o cabeçalho de uma seção ou de todo o documento. Geralmente contém elementos de introdução, como logotipos, menus de navegação e títulos.

<nav>: Define uma seção de navegação, que normalmente contém links para outras páginas ou partes do site.

<main>: Define o conteúdo principal de uma página. Deve conter apenas o conteúdo diretamente relacionado ao propósito central da página.

<article>: Define um conteúdo independente e autocontido, como postagens de blog, artigos de notícias ou comentários.

<section>: Define uma seção de conteúdo tematicamente relacionado, como capítulos em um livro ou seções de uma página.

<aside>: Define um conteúdo que é tangencial ao conteúdo principal da página, como barras laterais, anúncios ou caixas de informações adicionais.

<footer>: Define o rodapé de uma seção ou de todo o documento. Geralmente contém informações de contato, links para políticas de privacidade, direitos autorais, etc.

<figure> e <figcaption>: <figure> é usado para encapsular conteúdo que é referenciado no texto principal, como imagens, gráficos, vídeos, etc. <figcaption> fornece uma legenda para o conteúdo encapsulado em <figure>.

<time>: Define uma data ou hora. Isso pode ser útil para máquinas (como mecanismos de pesquisa) e humanos entenderem o significado da data/hora.

<mark>: Define um texto marcado ou destacado, geralmente com a cor de fundo amarela padrão do navegador.

<header>, <footer>, <nav>, <aside> e <section>: Embora esses elementos não sejam exclusivos do HTML5, eles foram redefinidos com significados semânticos mais claros e específicos no HTML5.

Ao usar esses elementos semânticos, você pode criar uma estrutura mais clara e significativa para suas páginas da web, tornando-as mais acessíveis, compreensíveis para os motores de busca e facilitando a manutenção do código.

CSS – Uso

Início

A propriedade que indica o uso de CSS é a propriedade **style**, ela costuma ser usada da seguinte maneira:

<p style = “font-size: 20px;”> TEXTO </p>

Perceba, que dentro da tag de início de parágrafo, colocamos a propriedade **style**, essa propriedade tem dentro de si diversos atributos, nesse caso, escolhemos o **font-size**, que determinará o tamanho da fonte, colocamos o ‘ : ‘ para indicar que agora determinaremos o tamanho da fonte, e finalizamos com o ‘ ; ‘ para indicar que terminamos de colocar informações para o tamanho de fonte.

Por ser uma tag de parágrafo, prosseguimos escrevendo o texto de parágrafo, e depois fechamos a tag.

Esse uso geralmente é útil para se mudar um atributo único.

Outro exemplo:

```
<style> p {  
    font-size: 20px;  
}  
</style>
```

Nesse caso, se abre a tag **<style>** para determinar que estamos usando o CSS, e indicamos que, “todos os parágrafos (**<p>**)”, abrimos colchetes para iniciar nossos comandos, e indicamos lá dentro que **font-size : 20px;** ou seja, que a fonte deve ter 20px de tamanho. Finalizamos fechando tudo.

O que esse exemplo faz, é indicar que TODOS os parágrafos deverão ter fonte de tamanho 20px.

Outra forma de se usar, é **referenciando um arquivo externo**, como por exemplo:

```
<link rel="stylesheet" href="assets/css/style.css">
```

Nesse caso, no html, usamos essa tag para indicar que estamos linkando um arquivo de **stylesheet**, e a referência de link é o arquivo **style.css**, que está na pasta css, que por sua vez está na pasta assets.

Assim, podemos adicionar as propriedades de estilo, sem criá-las uma a uma no HTML, além de ser possível referenciar o arquivo **style.css** em mais e uma página HTML.

#Id

Em CSS, o símbolo **#** é usado para representar um seletor de ID. Quando você vê **#** seguido de um nome, como **#meuld**, isso indica que o seletor está visando um elemento específico no HTML pelo seu ID.

Assim, qualquer estilo que você defina dentro do seletor **#meuld** será aplicado apenas ao elemento que tem o ID “meuld” no HTML. É importante notar que o ID deve ser único na página; ou seja, não deve haver mais de um elemento com o mesmo ID em um documento HTML válido.

Implementação no Index:

```
<header>
  <h1 id="titulo">Filmes, séries e muito mais, sem limites</h1>
```

<h1 id = “titulo”> Texto h1 </h1>

Aqui, inserimos dentro da tag desejada uma propriedade de ‘**id**’, e indicamos que essa id deve ser identificada pelo nome “**titulo**”. Id só pode ser usado para um único elemento.

Exemplo no CSS doc:

```
#titulo{
  color: azure;
}
```

```
#titulo {
    color : azure;
}
```

Lá no documento de CSS, puxamos o **titulo**, colocando uma **#** na frente para indicar que se trata de um id, e dentro dos colchetes determinamos todas as mudanças que desejamos fazer lá na tag que possui esse id em específico.

.Class

Em CSS, uma classe é um seletor que permite aplicar estilos a um ou mais elementos HTML. Ao contrário dos identificadores (IDs), que devem ser únicos em um documento HTML, as classes podem ser aplicadas a múltiplos elementos.

Para definir uma classe em CSS, você usa um seletor precedido por um ponto . seguido pelo nome da classe.

Você pode aplicar a mesma classe a vários elementos no seu HTML, permitindo que você defina um estilo consistente para esses elementos. As classes são extremamente úteis para aplicar estilos a grupos de elementos que compartilham características semelhantes, como estilos de layout, cores ou tamanhos de fonte.

.subTitulo

Em CSS, você pode estilizar um subtítulo usando uma classe ou um seletor específico.

font-size define o tamanho da fonte do subtítulo.

font-weight define a espessura da fonte (neste caso, negrito).

color define a cor do texto.

margin-bottom adiciona um espaçamento na parte inferior do subtítulo, para garantir que haja espaço entre o subtítulo e o próximo elemento..

Aplicação no HTML:

```
<h2 class="subTitulo">Assista onde quiser. Cancele quando quiser.</h2>
<h3 class="subTitulo">Quer assistir? Informe seu email para criar ou reiniciar sua assinatura.</h3>
</header>
```

<h2 class = “nomeClasse”> Texto h2 </h2>

<h3 class = “nomeClasse”> Texto h3 </h3>

Aqui, determinamos no html, que as tags h2 e h3 devem receber uma classe de CSS chamada **“nomeClasse”**.

Configuração no CSS doc:

```
.subTitulo{
  color: rgb(255, 255, 255);
}
```

```
.nomeClasse {
  color: rgb (255, 255, 255);
}
```

Aqui no CSS doc, chamamos a classe **nomeClasse**, colocando um ‘.’ na frente para indicar que se trata de uma classe, e dentro dos colchetes nós determinamos as mudanças que serão feitas, que nesse caso seria mudar a cor, essas mudanças serão aplicadas a **todas** as tags que herdam essa classe.

Background-image

O background precisa estar dentro do seletor <body>, Background: é o atributo que você vai definir.

Por definição padrão a imagem se repete automaticamente, por padrão também, uma imagem de fundo é colocada no canto superior esquerdo de um elemento e repetida vertical e horizontalmente.

A imagem de fundo também pode ser definida para elementos específicos, como o elemento <p>:

Para adicionar uma imagem com link devemos usar o url (“ link”)

Exemplos:

```
background: url("https://i.imgur.com/0  
background-repeat: no-repeat;  
background-color: black;  
background-size: 100%;
```

FlexBox

https://www.w3schools.com/css/css3_flexbox.asp

Flexbox é um modelo de layout em CSS que foi introduzido no CSS3 para facilitar o design responsivo e flexível de layouts de página complexos. Ele oferece uma maneira eficiente de organizar, alinhar e distribuir espaços entre os itens de um contêiner, mesmo quando o tamanho dos itens é desconhecido ou dinâmico.

O Flexbox opera em dois níveis: o contêiner e os itens dentro dele.

Contêiner Flex: O contêiner flex é simplesmente um elemento que tem seus filhos diretos exibidos como itens flexíveis. Para definir um contêiner como um contêiner flex, você pode aplicar a propriedade `display: flex` ou `display: inline-flex` ao elemento pai.

Itens Flexíveis: Os itens flexíveis são os filhos diretos do contêiner flex. Eles se comportam de acordo com as propriedades e os valores definidos no contêiner flex para controlar o layout e o comportamento dos itens.

Aqui estão algumas das principais características do Flexbox:

Alinhamento: O Flexbox oferece um controle preciso sobre o alinhamento dos itens em relação ao eixo principal (principal) e ao eixo transversal (cruzado).

Distribuição de espaço: Ele permite distribuir o espaço restante dentro do contêiner entre os itens de maneira flexível, permitindo criar layouts responsivos e dinâmicos.

Reorganização: Os itens flexíveis podem ser reorganizados e reordenados facilmente, alterando a ordem em que aparecem no código HTML, sem modificar a estrutura do DOM.

Tamanho Flexível: Os itens flexíveis podem expandir, encolher e se adaptar dinamicamente ao tamanho do contêiner e ao conteúdo dentro deles.

Flexbox é amplamente suportado pelos navegadores modernos e é uma ferramenta poderosa para criar layouts complexos de forma mais simples e eficiente do que os métodos tradicionais de layout CSS, como floats e posicionamento absoluto. Ele é especialmente útil para criar layouts responsivos e adaptáveis em dispositivos de diferentes tamanhos e resoluções de tela.

Em um contêiner, usamos a tag <div> </div> para separar a tela em divisões:



o código estaria:

```
<div class = "flex-container">  
<div> 1 </div>  
<div> 2 </div>  
<div> 3 </div>  
</div>
```

No CSS doc:

```
.menu{  
  display: flex;  
  justify-content: space-between;  
}
```

```
.menu{  
  display: flex;  
  justify-content: space-between;  
}
```

```
.centro{  
  display: flex;  
  justify-content: center;  
}
```

```
.centro {  
  display: flex;  
  justify-content: center;  
}
```

- Display é uma propriedade CSS usada para definir o tipo de exibição de um elemento. No contexto do Flexbox, você usaria display: flex para transformar um elemento em um contêiner flexível, o que significa que seus filhos diretos (itens flexíveis) seguirão as regras do Flexbox.

- **Justify-content** é uma propriedade CSS usada para alinhar os itens flexíveis ao longo do eixo principal do contêiner flexível. O eixo principal é definido pela direção em que o contêiner flexível é exibido, que é controlada pela propriedade `flex-direction`

Exemplos de Flex:

- **flex-start:** Alinha os itens no início do contêiner.
- **flex-end:** Alinha os itens no final do contêiner.
- **center:** Alinha os itens no centro do contêiner.
- **space-between:** Distribui uniformemente os itens ao longo do eixo principal, com espaço entre eles.
- **space-around:** Distribui uniformemente os itens ao longo do eixo principal, com espaço ao redor deles.

Margin

A propriedade `margin` pode ser usada para definir o espaçamento em torno de um elemento. Existem quatro valores que você pode definir para a margem: superior, direita, inferior e esquerda. Eles podem ser definidos de uma só vez ou separadamente.

Valores individuais: Se você deseja definir o espaçamento em todos os lados do elemento, você pode simplesmente definir um valor único para a propriedade `"margin"`.

Valores separados: Se você deseja definir espaçamento diferente para cada lado, pode especificar quatro valores separados para a propriedade `"margin"`, representando a margem superior, direita, inferior e esquerda, respectivamente.

Unidades de medida: `px` (pixels) é uma das unidades de medida mais comuns para definir o tamanho da margem. 1 pixel é uma unidade mínima de medida na tela. Quando você define `margin: 10px;`, está dizendo ao navegador para adicionar um espaço de 10 pixels em torno do elemento.

Unidades de medida

Pixels (px): É uma unidade de medida comum usada para dimensões precisas em telas de dispositivos. Um pixel é a menor unidade de exibição em uma tela.

Porcentagem (%): É uma unidade de medida relativa ao elemento pai. Por exemplo, definir `width: 50%;` em um elemento filho fará com que ele ocupe metade da largura de seu elemento pai.

Em (em): É uma unidade de medida relativa ao tamanho da fonte do elemento pai. 1em é igual ao tamanho da fonte do elemento pai.

Rem (rem): Semelhante ao em, mas relativo ao tamanho da fonte do elemento raiz do documento HTML (<html>).

Centímetros (cm): Unidade de medida absoluta baseada em unidades físicas do mundo real, como centímetros.

Milímetros (mm): Unidade de medida absoluta baseada em milímetros, outra unidade de medida física.

Polegadas (in): Unidade de medida absoluta baseada em polegadas, também usada em contextos físicos.

Pontos (pt): Unidade de medida absoluta frequentemente usada em contextos de impressão e design gráfico.

Picopolegadas (pc): Outra unidade de medida absoluta, geralmente equivalente a 12 pontos.

Viewport Width (vw): Unidade de medida relativa ao tamanho da janela de visualização (viewport) do navegador, representando uma porcentagem da largura da viewport.

Viewport Height (vh): Unidade de medida relativa ao tamanho da janela de visualização (viewport) do navegador, representando uma porcentagem da altura da viewport.

Viewport Minimum (vmin): Unidade de medida relativa ao tamanho da viewport, representando o menor valor entre a largura e a altura da viewport.

Viewport Maximum (vmax): Unidade de medida relativa ao tamanho da viewport, representando o maior valor entre a largura e a altura da viewport.

Botão

Para estilizar um botão em CSS, você pode usar várias propriedades para controlar sua aparência, como cor de fundo, cor do texto, borda, sombra e muito mais.

A tag usada para criar um botão é o <button> e fechamento </button>.

Adicionando link no botão :

A tag não tem atributo href para link, por isso para inserir link ao botão precisamos criar uma tag de hiperlink ou <a> juntamente do atributo href="URL" que ira especificar o url de direcionamento.

```
<a href="URL"><button>botão</button></a>
```

Cor no botão

Para mudar a cor de fundo do botão em CSS, você pode usar a propriedade

background-color : XXXXX

Você pode substituir esse valor por qualquer outra cor hexadecimal, nome de cor, valor RGB, valor HSL ou outro formato de cor suportado pelo CSS.

Editando a cor do texto do botão:

Para mudar a cor do texto no botão em CSS, você pode usar a propriedade color:

color: #ffffff;

Como alterar o tamanho do botão

Para alterar o tamanho do botão em CSS, você pode ajustar as propriedades de largura (**width**) e altura (**height**). Você pode ajustar esses valores conforme necessário para obter o tamanho desejado para o seu botão em px.

Lembre-se de que, ao ajustar o tamanho do botão, é importante garantir que o conteúdo do botão permaneça legível e que o botão não fique muito pequeno ou muito grande em relação ao restante do design da sua página. Experimente diferentes tamanhos e unidades de medida até encontrar o que funciona melhor para o seu caso de uso.

Como adicionar sombras ao botão

Para adicionar sombras ao botão em CSS, você pode usar a propriedade **box-shadow**.

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

Os valores especificados representam a posição horizontal, a posição vertical, o desfoque, o espalhamento e a cor da sombra, respectivamente.

- O primeiro valor (0) representa a posição horizontal da sombra.
- O segundo valor (4px) representa a posição vertical da sombra.
- O terceiro valor (8px) representa o desfoque da sombra.
- O quarto valor (rgba(0, 0, 0, 0.1)) representa a cor da sombra, especificada como uma cor RGBA (red, green, blue, alpha).

Botão com transição de cor

Para criar um botão com uma transição suave de cor ao passar o mouse sobre ele, você pode usar a propriedade `transition` em conjunto com a pseudo-classe `:hover`

Tem que ser adicionada no botão principal com `background-color` o tempo de duração da transição (0.3s) e o tipo de transição (ease, para uma transição suave).

```
transition: background-color 0.3s ease
```

Exemplo no Doc do CSS

```
.botao:hover {  
    background-color: #45a049;
```

Botão com animação de rotação

Para criar um botão com uma animação de rotação ao passar o mouse sobre ele, você pode usar a propriedade `transform` em conjunto com a pseudo-classe `:hover`.

A função `rotate(360deg)` aplicada ao `transform` faz com que o botão gire 360 graus ao passar o mouse sobre ele.

```
.botao:hover {  
    transform: rotate(360deg);
```

Para criar um botão com uma animação de pulsação ao passar o mouse sobre ele, você pode usar `keyframes` de animação CSS em conjunto com a propriedade `animation`.

```
@keyframes pulse {
```

```
    0% {  
        transform: scale(1);  
    }
```

```
    50% {  
        transform: scale(1.1);  
    }
```

```
    100% {  
        transform: scale(1);  
    }
```

No `hover` vamos anular a animação:

```
.botao:hover {  
    animation: none;
```

Assim um `keyframe` de animação chamado `pulse` é definido, que especifica a escala do botão em diferentes porcentagens da animação. A animação aumenta ligeiramente o tamanho do botão em 50% da duração da animação e depois retorna ao tamanho original.

A propriedade animation é aplicada ao botão, especificando a animação de pulsação (pulse), a duração da animação (1s) e o número de vezes que a animação deve ser repetida (infinite, para repetição infinita).

Quando o cursor do mouse passa sobre o botão, a animação de pulsação é interrompida removendo-se a animação. Isso é feito definindo animation: none; na pseudo-classe :hover

Comandos

<p style = “font-size: 20px;”> TEXTO </p>

Exemplo de tag com uso singular de CSS.

```
<style> p {  
    font-size: 20px;  
}  
</style>
```

Exemplo de tag de CSS para estilizar múltiplas tags de um mesmo tipo, nesse caso p (parágrafo).

<link rel=”stylesheet” href=”assets/css/style.css”>

Exemplo de tag que linkar um arquivo externo de CSS.

<h1 id = “titulo”> Texto h1 </h1>

Exemplo tag com id CSS

```
#titulo {  
    color : azure;  
}
```

Exemplo de definição de id em CSS.

<h2 class = “nomeClasse”> Texto h2 </h2>

<h3 class = “nomeClasse”> Texto h3 </h3>

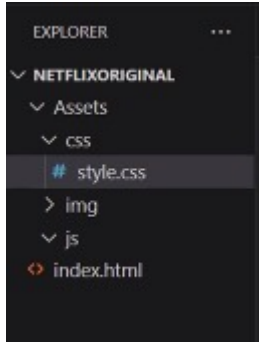
Exemplo de tag com uso de classe CSS.

```
.nomeClasse {  
    color: rgb (255, 255, 255);  
}
```

Exemplo de definição de classe em CSS.

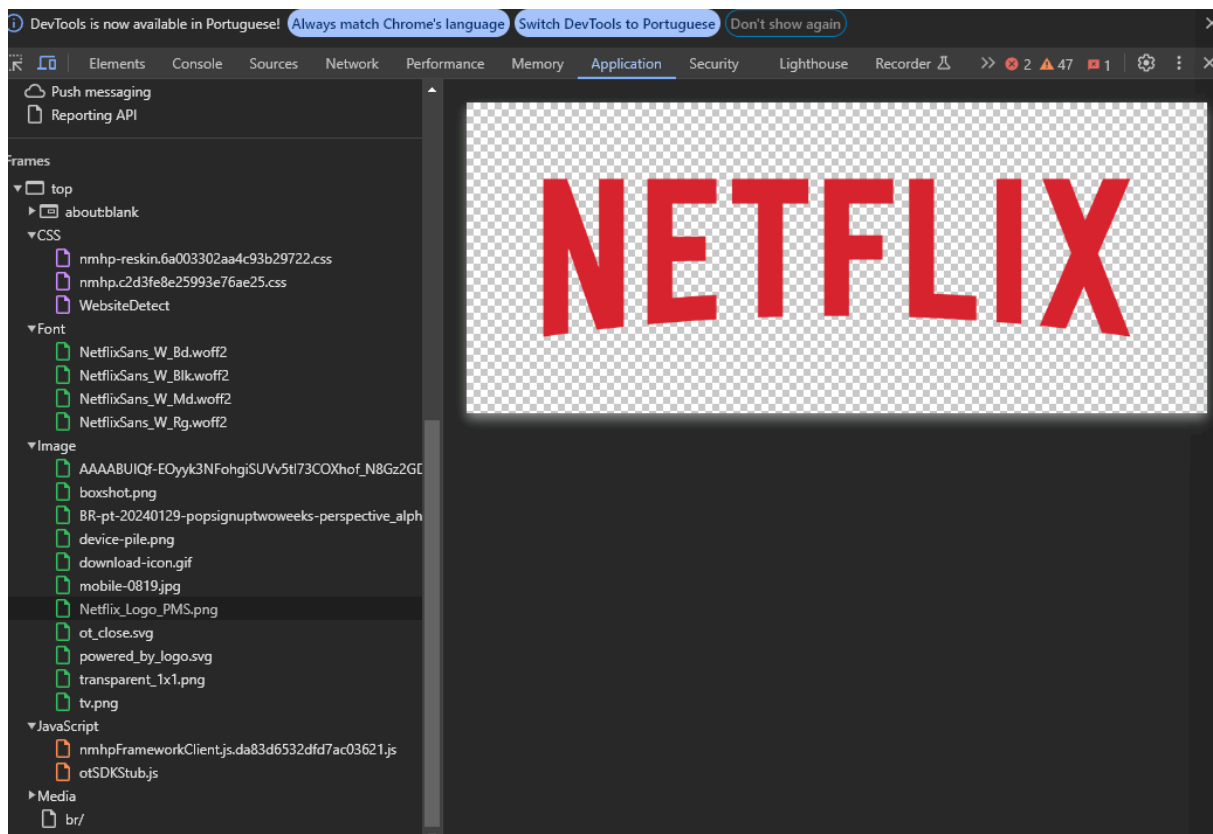
Boas Práticas

- Criar uma pasta Asset e organizar a estrutura de conteúdos como css, js e img:

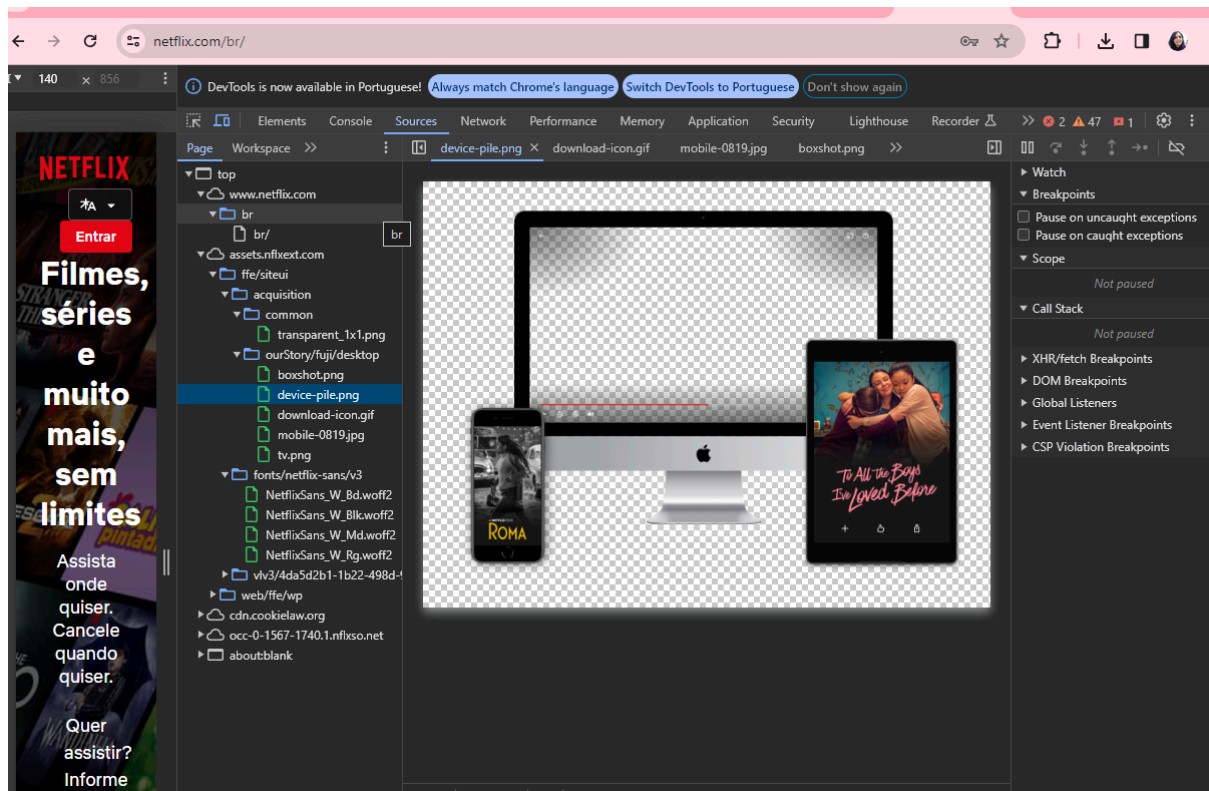


- Para pegar as fotos, gifs entre outros você pode entrar dentro das configurações de dev do site, “Ferramentas do programador”

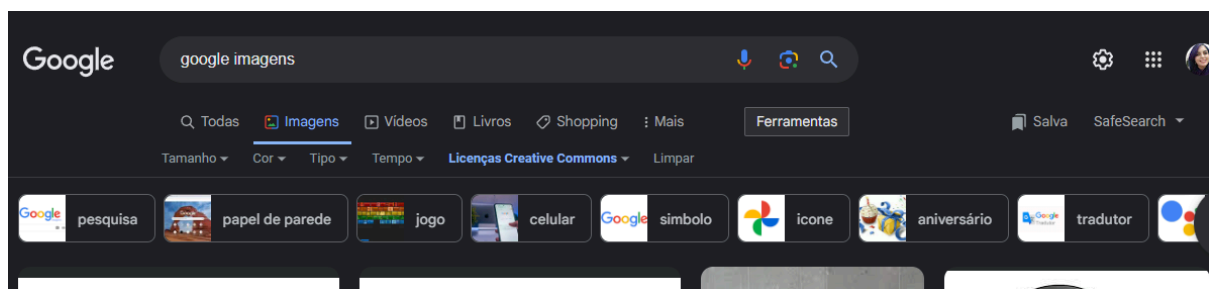
Application -> Frames



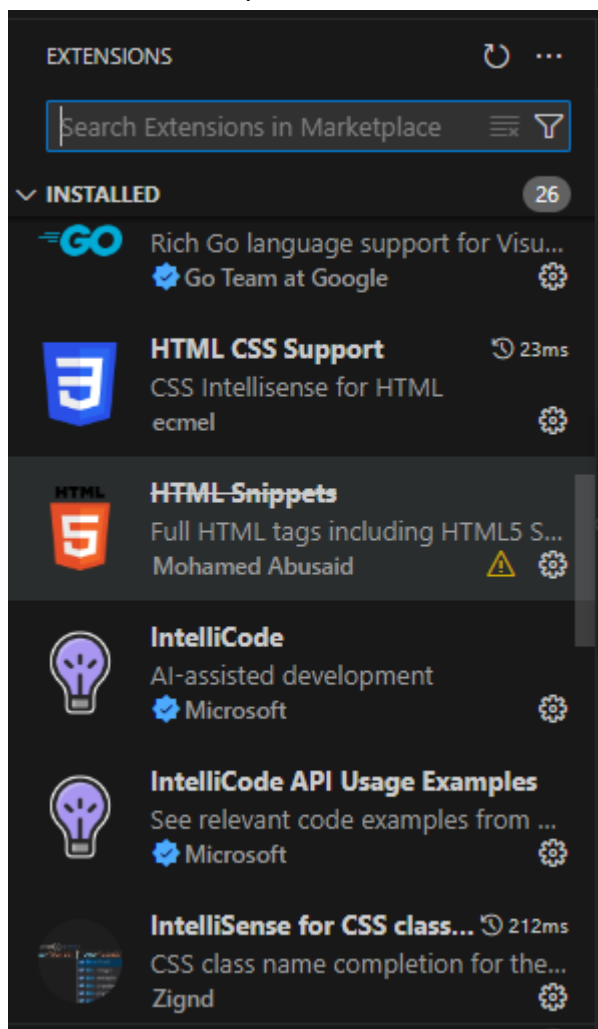
Source - > dentro das páginas



- **ctrl + space** para ligar caminhos dentro do código
- Use esse site para pegar imagens : <https://imgur.com/>
- Pela pesquisa do Google temos a opção de licença de uso:



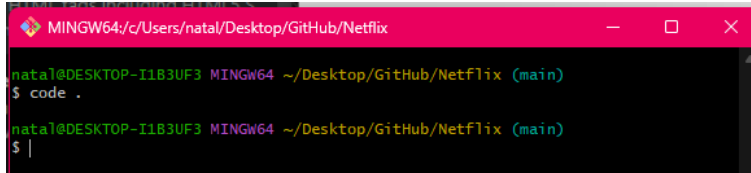
- Extensões para baixar:



(Dica, baixem a extensão live server pra toda vez que modificarem o código a página atualiza sozinha. A extensão do prettier e seta um atalho no teclado pra fazer quebra de linha sozinho)

- Dica para abrir o VScode na pasta que você deseja salvar :

Abra o git bash e digite **code .**



```
MINGW64/c/Users/natal/Desktop/GitHub/Netflix
natal@DESKTOP-I1B3UF3 MINGW64 ~/Desktop/GitHub/Netflix (main)
$ code .
natal@DESKTOP-I1B3UF3 MINGW64 ~/Desktop/GitHub/Netflix (main)
$ |
```

Padronização de nomenclatura

arquivos, pastas, classes, atributos e variáveis

Aqui está um resumo:

1. Pastas e pacotes: São escritos em minúsculo. Exemplo: "assets", "css".
2. Arquivos em geral: Com exceção de classes, são escritos em camelCase. Exemplo: "arquivoDeExemplo.js".
3. Classes: São escritas com a primeira letra maiúscula e usando CamelCase. Exemplo: "MinhaClasse".
4. Atributos e Variáveis: São escritos em camelCase. Exemplo: "minhaVariavel".

Este resumo reflete a convenção adotada na conversa, onde pastas, pacotes e a maioria dos arquivos seguem a convenção de minúsculas e camelCase, enquanto classes começam com letra maiúscula e também seguem o CamelCase.

Links e afins

Youtube Playlist com aulas de HTML,CSS e JS :

https://youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n&si=GG-pFR0d46pRbFJh

Library W3Schools:

https://www.w3schools.com/css/css_intro.asp

Imgur:

<https://imgur.com/>

Flexbox Froggy, para treinar habilidades de flexbox:

<https://flexboxfroggy.com/#pt-br>

Elementos semânticos do HTML5:

<https://medium.com/reprogramabr/semanticahtml5-5252b4937f0a>

Seletores CSS:

<https://www.atatus.com/blog/powerful-css-selectors/#id-selector>

Medidas no CSS:

https://www.alura.com.br/artigos/guia-de-unidades-no-css?utm_term=&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=20987928442&hsa_grp=157916200306&hsa_ad=689395782879&hsa_src=q&hsa_tgt=dsa-19959388920&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQiAzoeuBhDqARIsAMdH14FnWSz4GBiFHwoKKCLIMIDDpu2c1AWBNEI9n_iqbsliTO5EoDy4SR0aApj0EALw_wcB

Tipos de Cursor :

<https://www.webcis.com.br/mudando-o-cursor-do-mouse-com-css.html>

Dúvidas

Propiedades

As propriedades são características visuais que os elementos html possuem. Uma declaração pode conter uma ou mais propriedades, cada uma com seu respectivo valor e separada por ponto e vírgula.

REGRAS CSS

id . classe : estado [] atributo * total

