	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Bacharelado em Sistemas de Informação</p>	
<p><i>Disciplina</i> Laboratório de Aplicações Web</p>	<p align="center">MATERIAL COMPLEMENTAR</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

O CSS (Cascading Style Sheets (do inglês, Folha de Estilo em Cascata)) formata a informação presente no HTML. Sua sintaxe é composta por um seletor e um bloco de declaração.



O seletor representa a estrutura utilizada para determinar quais elementos do HTML serão formatados, podemos utilizar o seletor encadeado:

```
table tr th {
    text-align: center
}
```

Este seletor formata o texto da coluna th que está dentro da tag tr que está dentro da tag table.

Podemos utilizar o seletor agrupado:

```
strong, em, span {
    color: #FF0000;
}
```

Neste os elementos do HTML strong, em e span terão a formatação aplicada, todos eles.

É possível ainda colocar um atributo class em qualquer tag do HTML dentro da tag body e selecioná-la:

```
.azul {
    color: #0000FF;
}
```

Ou ainda colocar um atributo id em qualquer tag do HTML dentro da tag body e selecioná-la:

```
#pequeno: {
    font-size: 0.8rem;
}
```

A diferença no uso do class para o id é que o id deve ter o seu valor único. Com o class é possível aplicar a mesma definição de css em vários locais do HTML.

Existem ainda os seletores complexos que devem ser utilizados permitindo aplicar configurações em diversas partes do HTML.

Imagine que você tenha um título (h1) seguido de um parágrafo (p). Você precisa selecionar todos os parágrafos que vem depois de um título h1.

```
h1 + p {
    color: red;
}
```

Esses seletores são um dos mais simples e mais úteis.

Padrão	Significado	CSS
*	qualquer elemento	2
E	um elemento do tipo E	1
E[foo]	um elemento E com um atributo "foo"	2
E[foo="bar"]	um elemento E cujo valor do atributo "foo" é exatamente igual a "bar"	2
E[foo~="bar"]	um elemento E cujo valor do atributo "foo" é uma lista de valores separados por espaços em branco, um dos quais é exatamente igual a "bar"	2
E[foo^="bar"]	um elemento E cujo valor do atributo "foo" começa exatamente com a string "bar"	3
E[foo\$="bar"]	um elemento E cujo valor do atributo "foo" termina exatamente com a string "bar"	3
E[foo*="bar"]	um elemento E cujo valor do atributo "foo" contém a substring "bar"	3
E[foo = "en"]	um elemento E cujo atributo "foo" tem uma lista de valores separados por hífen começando (da esquerda) com "en"	2
E:root	um elemento E, raiz do documento	3
E:nth-child(n)	um elemento E, o n-ésimo filho de seu pai	3
E:nth-last-child(n)	um elemento E, o n-ésimo filho de seu pai, contando a partir do último	3
E:nth-of-type(n)	um elemento E, o n-ésimo irmão de seu tipo	3
E:nth-last-of-type(n)	um elemento E, o n-ésimo irmão de seu tipo, contando a partir do último	3
E:first-child	um elemento E, primeiro filho de seu pai	2
E:last-child	um elemento E, último filho de seu pai	3
E:first-of-type	um elemento E, primeiro irmão de seu tipo	3
E:last-of-type	um elemento E, último irmão de seu tipo	3
E:only-child	um elemento E, único filho de seu pai	3
E:only-of-type	um elemento E, único irmão de seu tipo	3
E:empty	um elemento E que não tem filhos (incluindo nós de texto)	3
E:link E:visited	um elemento E sendo a âncora de origem de um hiperlink cujo alvo ainda não foi visitado (:link) ou já foi visitado (:visited)	1
E:active E:hover E:focus	um elemento E durante certas ações do usuário	1 e 2
E:target	um elemento E sendo o alvo do URI de referência	3
E:lang(fr)	um elemento do tipo E na linguagem "fr" (a linguagem do documento especifica como a linguagem é determinada)	2
E:enabled E:disabled	um elemento da interface do usuário E que é habilitado ou desabilitado	3
E:checked	um elemento da interface do usuário E que é marcado (por exemplo, um botão de opção ou caixa de seleção)	3
E::first-line	a primeira linha formatada de um elemento E	1
E::first-letter	a primeira letra formatada de um elemento E	1
E::before	conteúdo gerado antes de um elemento E	2
E::after	conteúdo gerado depois de um elemento E	2

E.warning	um elemento E cuja classe é "warning" (a linguagem do documento especifica como a classe é determinada).	1
E#myid	um elemento E com ID igual a "myid".	1
E:not(s)	um elemento E que não corresponde ao seletor simples s	3
E F	um elemento F descendente de um elemento E	1
E > F	um elemento F filho de um elemento E	2
E + F	um elemento F imediatamente precedido por um elemento E	2
E ~ F	um elemento F precedido por um elemento E	3

<https://www.w3.org/TR/selectors-3/#selectors>

As unidades de medida também são importantes, principalmente para o desenvolvimento responsivo de sistemas Web. Unidades de comprimento absoluto não são recomendadas para uso na tela, porque os tamanhos de tela variam muito. No entanto, elas podem ser usadas se o meio de saída for conhecido, como para layout de impressão.

As unidades de comprimento absoluto mais comuns são:

Unidade	Descrição
cm	Centímetros
mm	Milímetros
in	Polegadas (1 in = 96px = 2.54cm)
px	Pixels (1 px = 1/96 de 1 in) são relativos ao dispositivo de visualização. Para dispositivos de baixo dpi, 1 px é um pixel de dispositivo (ponto) da tela. Para impressoras e telas de alta resolução, 1 px implica em vários pixels de dispositivo.
pt	(1 pt = 1/72 de 1 in)

Unidades de comprimento relativo especificam um comprimento relativo a outra propriedade de comprimento. Unidades de comprimento relativo escalam melhor entre diferentes mídias de renderização.

As unidades de comprimento relativo mais comuns são:

Unidade	Descrição
rem	Relativo ao tamanho da fonte do elemento raiz
em	Relativo ao tamanho da fonte do elemento (2em significa 2 vezes o tamanho da fonte atual)
vw	Relativo a 1% da largura do viewport.
vh	Relativo a 1% da altura do viewport
vmin	Relativo a 1% da menor dimensão do viewport
vmax	Relativo a 1% da maior dimensão do viewport
%	Relativo ao elemento pai
fr	É uma unidade fracionária e 1fr é para 1 parte do espaço disponível.

https://www.w3schools.com/cssref/css_units.php

Tags semânticas, são tags que não apresentam nenhum efeito visual, mas são importantes para a compreensão do código do HTML.

https://www.w3schools.com/html/html5_semantic_elements.asp

Estas tags devem ser utilizadas para criar a estrutura do seu site. Após desenhar o seu site, divida-o em quadros, de forma que você consiga dar um sentido para cada tag utilizada.

Feito isso, agora vamos montar o CSS para posicionar os elementos. Inicialmente vamos estudar como incluir uma fonte no CSS. Inicialmente, escolha sua fonte no site google fonts:

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

Clique nela, depois clique em Get font, depois em Get embed code, depois marque a opção @import. A primeira linha do arquivo do print abaixo é o import da fonte Inter.


No seletor * colocamos configurações que devem ser aplicadas em todas as tags. Todas as tags tem configurações padrão. As propriedades colocadas no * retiram as configurações do box model e faz com que as bordas componham o tamanho das tags.

O seletor :root permite criar variáveis. Utilizamos elas no estilo criado para definir as cores. Para utilizá-las colocamos as chaves dentro da função var().

Flex box é um recurso do CSS muito interessante para posicionar os elementos na tela.

<https://origamid.com/projetos/flexbox-guia-completo/>

Utilizamos o header para colocar uma borda abaixo, o display flex e o justify-content foram utilizados para separar a logo do menu. O height define a altura. O padding espaçamento interno. A configuração colocada no nav alinha o a que está dentro dele no centro e o a: hover configura ao passar o mouse no link.

```
css >  estilo.css > ...
1  @import url('https://fonts.googleapis.com/css2?family=Inter:wght@100..900&display=swap');
2  * {
3      box-sizing: border-box;
4      margin: 0;
5      padding: 0;
6      outline: 0;
7      font-family: 'Inter';
8  }
9  :root {
10     --primary: #78909C;
11     --secondary: #d9d9d9;
12     --black: #000000;
13     --white: #ffffff;
14 }
15 header {
16     display: flex;
17     justify-content: space-between;
18     border-bottom: var(--primary);
19     border-bottom-width: 0.5rem;
20     border-bottom-style: solid;
21     height: 5rem;
22 }
23 header img {
24     padding: 0.3rem;
25 }
26 header nav {
27     display: flex;
28     align-items: center;
29 }
```

Vamos estudar o CSS do footer. Definimos a altura e alinhamos as imagens no centro. Foi importante aplicar uma margem nas imagens para que elas não fiquem grudadas.

```

footer {
  height: 3.5rem;
  display: flex;
  align-items: center;
  background-color: var(--primary);
}
footer img {
  margin: 0 0.5rem;
}

```

Definidos os tamanhos do header e do footer, podemos fazer o cálculo do main para que a página ocupe todos os espaços da tela.

```

main {
  height: calc(100vh - 8.5rem);
}

```

Outro recurso interessante é o grid

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

Utilizando o main pode-se configurar o display grid para posicionar os itens dentro dele em 4 colunas com espaçamento automático e de tamanho igual. O justify-content coloca espaçamento entre os quadros e o align-content coloca eles no início da página.

O section foi utilizado para configurar o quadrado colocando fundo, espaçamento interno e externo, borda arredondada e largura mínima.

```

/* Comentário */
main.comentario {
  display: grid;
  grid-template-columns: auto auto auto auto;
  justify-content: space-around;
  align-content: baseline;
}
main.comentario section {
  background-color: var(--secondary);
  padding: 0.5rem;
  margin: 0.5rem 0;
  border-radius: 0.5rem;
  min-width: 12rem;
}

```

Outra possibilidade seria utilizar o fr

```

main.comentario {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 20px;
  padding: 20px;
}

```

<https://developer.mozilla.org/en-US/docs/Web/CSS/minmax>

Alguns comandos básicos para trabalhar com o github.

Iniciando um projeto com git

```
git init
```

Configurando a sua pasta local com uma pasta remota. É importante que a pasta remota esteja vazia.

```
git remote add origin https://endereco_do_seu_git
```

Acrescentando todos os arquivos criados localmente para serem enviados para a pasta remota

```
git add .
```

Configurando seu nome de usuário para ser identificado pelo github como autor dos arquivos.

```
git config user.name seu_usuario_git
```

Configurando seu e-mail para ser identificado pelo github como autor dos arquivos.

```
git config user.email seu_email_git
```

Definindo uma mensagem que identifica o código implementado.

```
git commit -m 'uma mensagem'
```

Enviando os arquivos para o origin na branch master

```
git push origin master
```