Introdução ao HTML

Conceito De HTML

HTML é a linguagem de marcação padrão para criar páginas na WEB. O HTML não é exatamente uma linguagem de programação, seu significado é: Hyper Text Markup Language (Linguagem de Marcação de HiperTexto).

O HTML descreve a estrutura da página usando marcação através de alguns elementos, esses elementos são blocos de construção e são representados por tags. As tags HTML classificam partes do conteúdo como: titulo, parágrafo, tabela, imagem etc. Os navegadores não exibem as tags HTML, mas as usam para renderizar o conteúdo da página, e o que você vê quando acessa um site, é o resultado dessa codificação de tags HTML

Estrutura Básica de Um Documento HTML

```
1   <!DOCTYPE html>
2   <html>
3   <heade>
4       <title>Introdução ao HTML</title>
5   </heade>
6   <body>
7       <h1>Primeiro Cabeçalho</h1>
8       <p>O meu primeiro parágrafo</p>
9   </body>
10  </html>
11
```

Explicação:

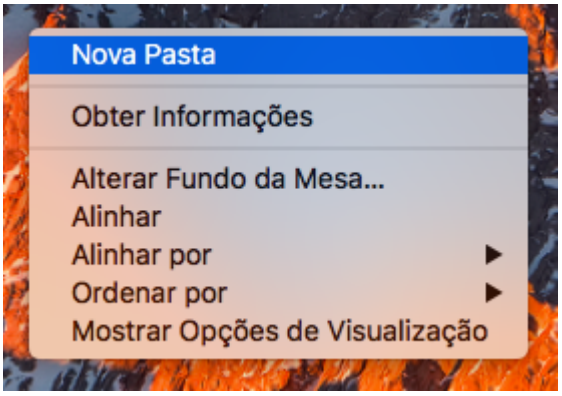
<!DOCTYPE html> define que o documento é HTML
<html> elemento raiz da página HTML
<heade> elemento que contém meta-informações sobre a página HTML
<title> elemento que especifica a título da página que aparece na aba do navegador
<body> O elemento que define o corpo da nossa página, é onde aparece o que visualizamos na página tais como parágrafos, imagens, links, etc.
<h1> elementos que define o nível de título mais alto

<p> elemento que define um parágrafo.

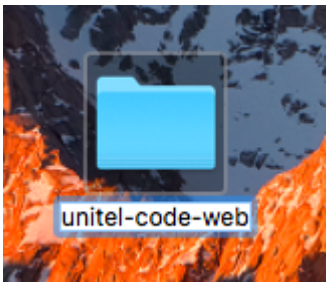
Essas são algumas das tags HTML, elas são nomes de elementos entre colchetes angulares. Ex: **<nome_da_tag>** conteúdo vai aqui... **</nome_da_tag>.**

Para começarmos a executar os nossos códigos em HTML vamos criar uma pasta onde vamos colocar os arquivos do nosso curso. Vamos no ambiente de trabalho (no Mac Mesa) criamos uma pasta com nome unitel-code-web dentro dessa pasta criamos uma outra pasta com nome aula1. Como mostram os passos a seguir:

Passo: 1



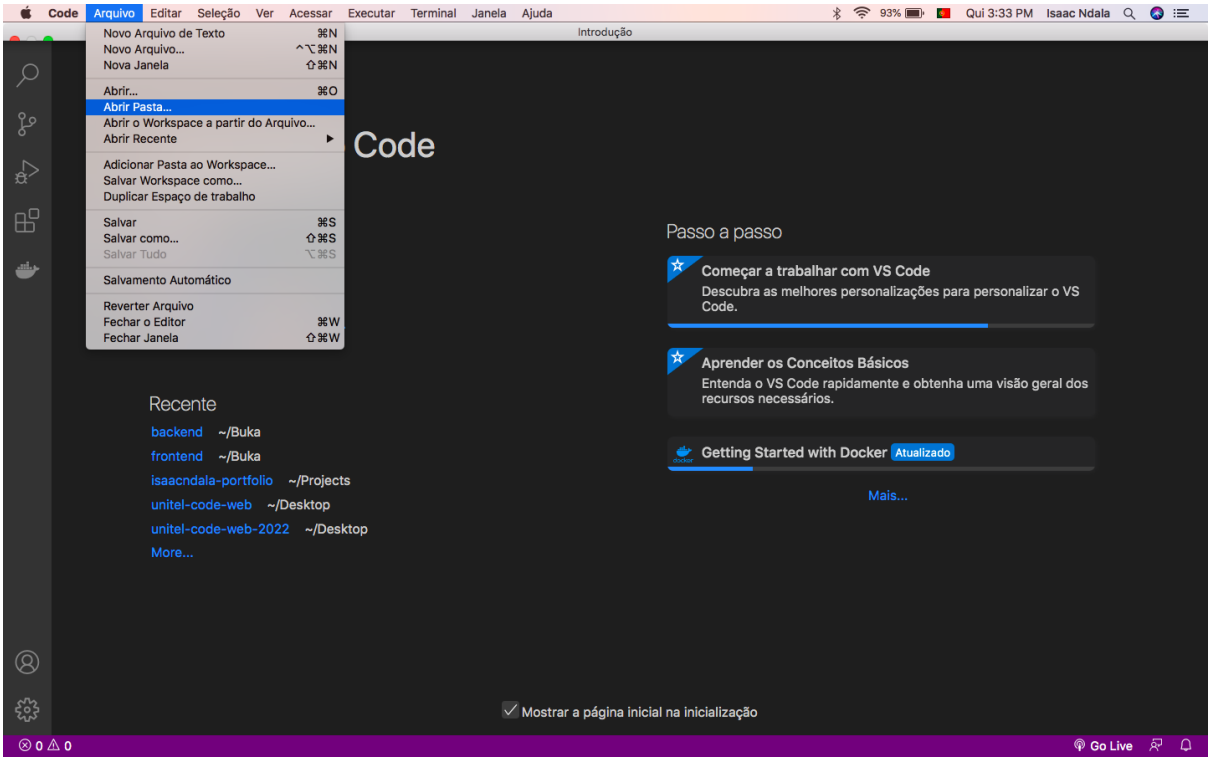
Passo: 2

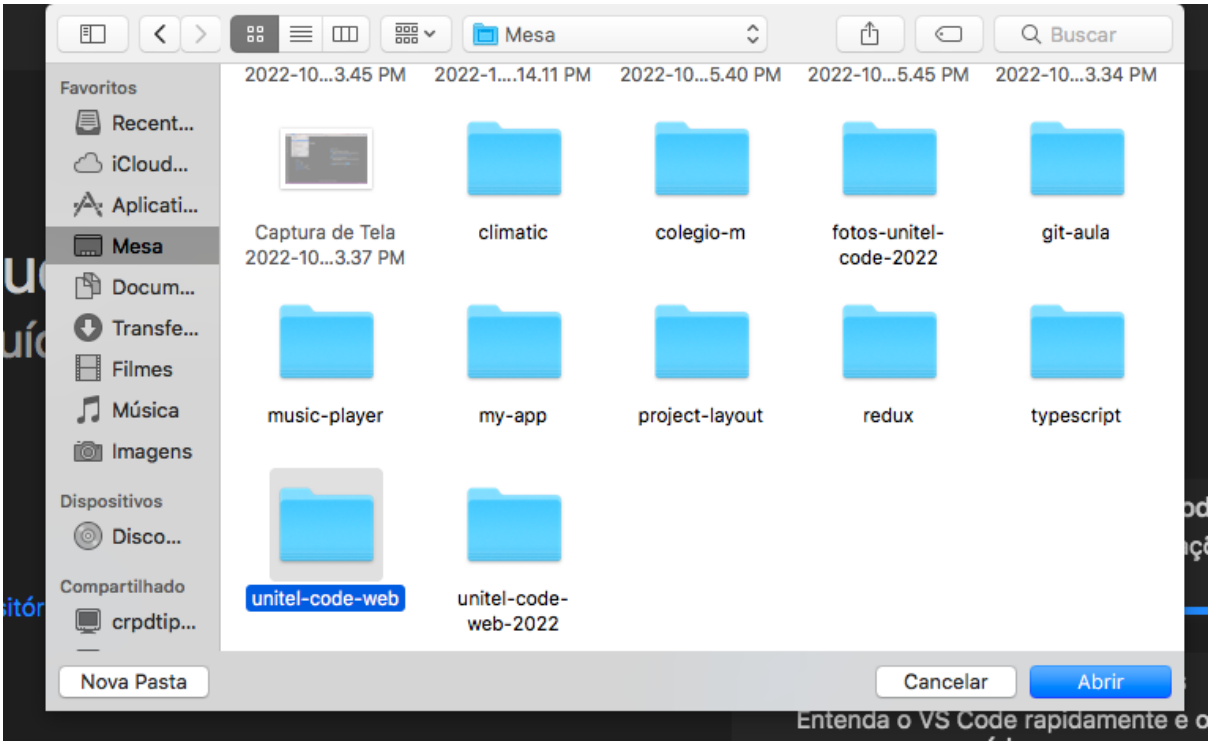


Passo: 3

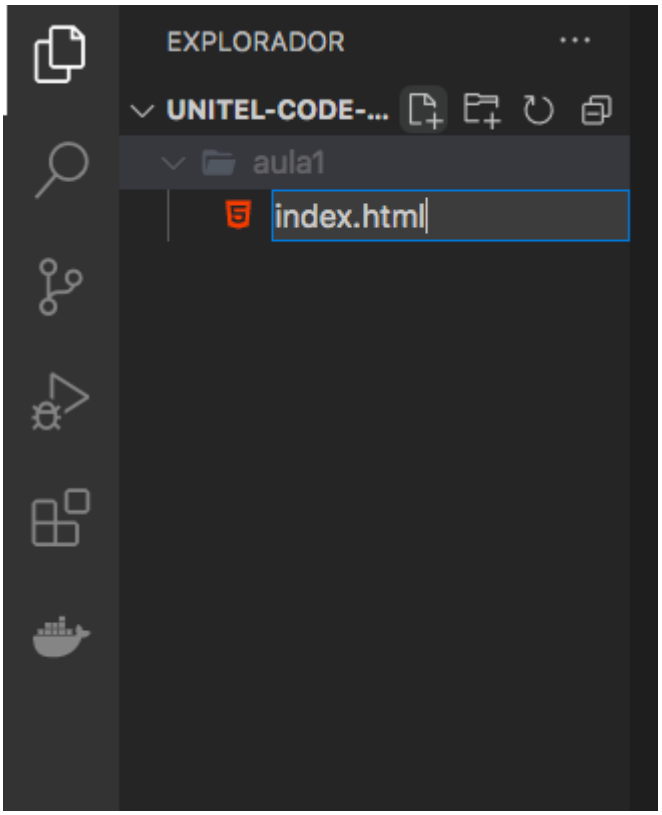
Agora vamos abrir o **Visual Studio Code** (pode usar um editor de sua escolha), no Mac usamos a combinação **Command + Tab** para abrir o **spotlight search** (no Windows no **ícone de pesquisa da barra de tarefa**) e digitamos **Visual Studio Code.**

Com o Visual Studio Code aberto vamos abrir a pasta que criamos no ambiente de trabalho. No menu na opção Arquivos>Abrir Pasta, vamos no ambiente de trabalho e selecionamos a pasta unitel-code-web e clicamos em abrir. Como mostram as imagens:

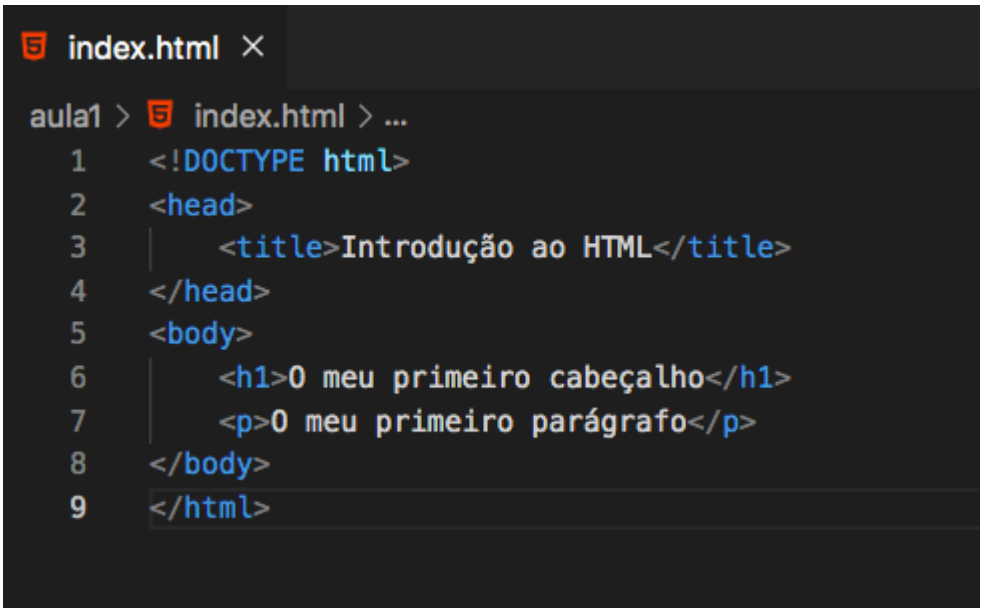




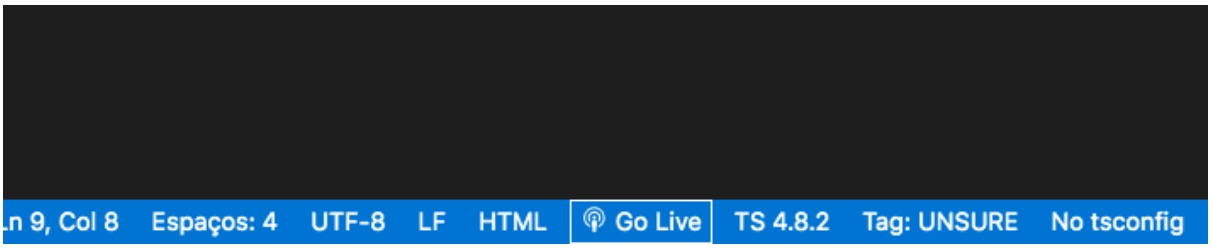
Depois de selecionarmos a pasta, agora podemos criar o nosso arquivo HTML. Na barra onde aparecem as pastas no canto esquerdo clicamos na pasta **aula1** e depois clicamos no ícone de documento com o sinal de mais para criar um novo arquivo e damos a ele o nome de index.html e clicamos enter.



No index.html digitamos o código que aparece abaixo e salvamos o arquivo clicando em CTRL + S



Para ver o resultado no navegador, no VS Code na parte inferior clicamos em Go Live (esta opção só aparece se instalou a extensão o Live Server).



Resultado:

Com isso temos a nossa primeira página HTML.

Cabeçalhos

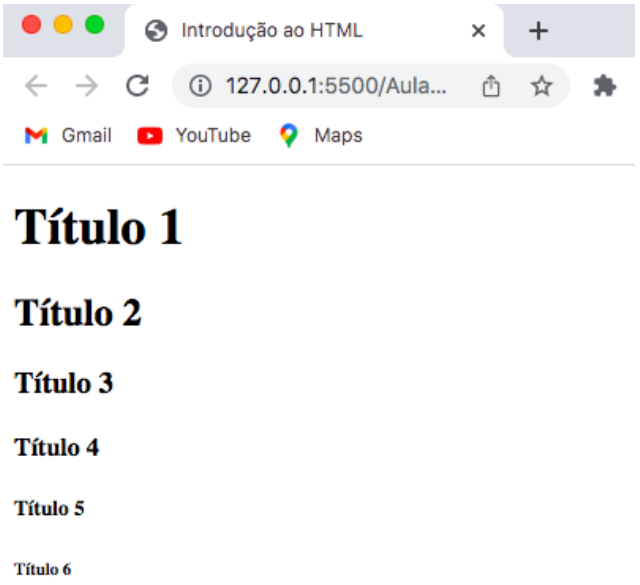
Os elementos de cabeçalhos permitem especificar que certas partes do seu conteúdo são títulos ou subtítulos. Da mesma forma que um livro tem o título principal e os capítulos possuem títulos e subtítulos, um documento HTML também tem. HTML contém 6 níveis de título que vai de <h1> à <h6>, onde <h1> é o nível de seção mais alto e <h6> é o mais baixo.

Exemplo:

```html
<!DOCTYPE html>
<html>
<heade>
    <title>Introdução ao HTML</title>
</heade>
<body>
    <h1>Título 1</h1>
    <h2>Título 2</h2>
    <h3>Título 3</h3>
    <h4>Título 4</h4>
    <h5>Título 5</h5>
    <h6>Título 6</h6>
</body>
</html>

```

Resultado:

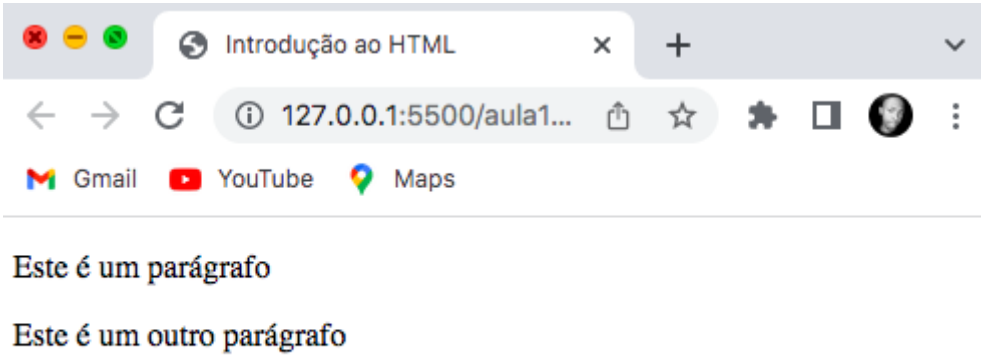


Parágrafos

Os elementos <p> são para conter parágrafos de texto; você os usará com frequência ao marcar um conteúdo de texto regular.

```
<p>Este é um parágrafo</p>
<p>Este é um outro parágrafo</p>
```

Resultado:



Listas

Muito do conteúdo da web é de listas e o HTML tem elementos especiais para elas. Listas de marcação sempre consistem em pelo menos 2 elementos. Os tipos mais comuns de lista são ordenadas e não ordenadas:

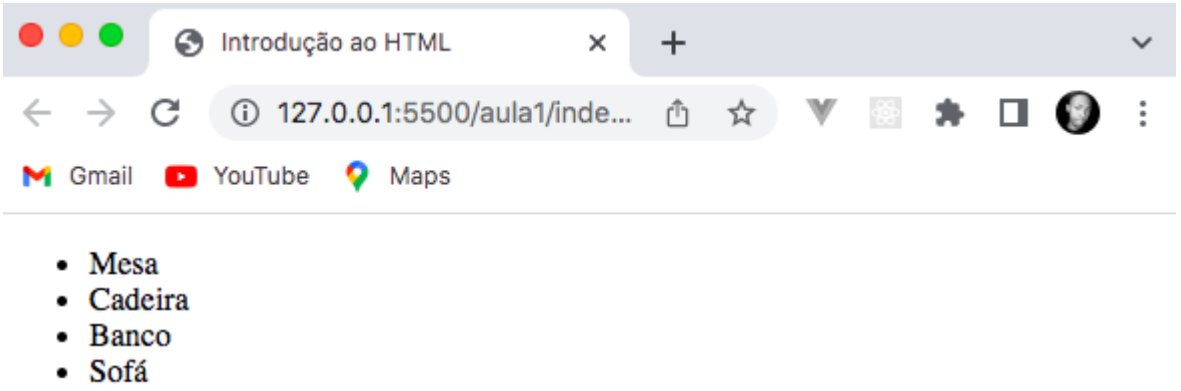
1. Listas não ordenadas são para listas onde a ordem dos itens não importa, como uma lista de compras, por exemplo. Essas são envolvidas em um elemento <ul>.
2. Listas Ordenadas são para listas onde a ordem dos itens importa, como uma receita. Essas são envolvidas em um elemento <ol>.

Cada item dentro das listas é posto dentro de um elemento <li> (item de lista).

Não ordernada:

```
<ul>
    <li>Mesa</li>
    <li>Cadeira</li>
    <li>Banco</li>
    <li>Sofá</li>
</ul>
```

Resultado:



Exercício:

1) Agora que já sabes como criar listas **não ordenadas** em HTML, como desafio vais usar os mesmos conhecimentos para criar uma lista **ordenada** contendo os seguintes elementos: **Café**, **Chá**, **Pão** e **Manteiga**.

2) Qual é a diferença que notou?

Imagens

As imagens em HTML são representados pela tag <img>. Essa tag incorpora uma imagem na nossa página na posição que aparece. Isso é feito pelo atributo src (*source*), que contém o caminho para nosso arquivo de imagem.

Incluímos também um atributo alt (*alternative*). Neste atributo, você especifica um texto descritivo para usuários que não podem ver a imagem, possivelmente devido aos seguintes motivos:

1. Eles são deficientes visuais. Usuários com deficiências visuais significativas costumam usar ferramentas chamadas leitores de tela para ler o texto alternativo para eles.
2. Algo deu errado, fazendo com que a imagem não seja exibida. Por exemplo, tente alterar deliberadamente o caminho dentro do atributo src para torná-lo incorreto. Se você salvar e recarregar a página, você deve ver algo assim no lugar da imagem:

Esta tag recebe atributos os atributos:

src: Onde se encontra o arquivo (internet, ou local)

alt: Texto caso a imagem não apareça (opcional)

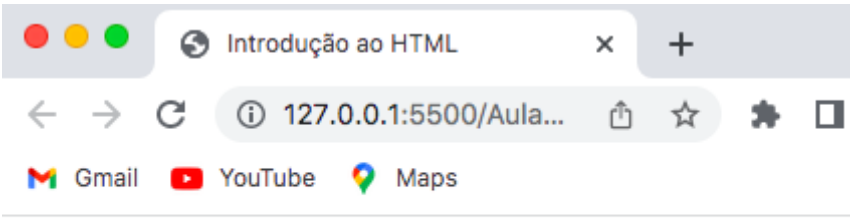
width: Largura da imagem (opcional)

height: Altura da imagem (opcional)

Exemplo:

```
<img src="./img/download.jpeg" alt="Uvas">
```

Resultado:



Alguns elementos não possuem conteúdo e são chamados de **elementos vazios.**

A tag de imagem não tem </img> de fechamento, e não há conteúdo interno. Isso acontece porque um elemento de imagem não envolve conteúdo para ter efeito em si mesmo. Sua proposta é incorporar uma imagem na página HTML no lugar que o código aparece.

Links

Links são muito importantes — eles são o que faz da web ser de fato uma REDE! Para adicionar um link, precisamos usar um elemento simples — <a> — "a" é a forma abreviada de "âncora".

Esta tag recebe atributos os atributos:

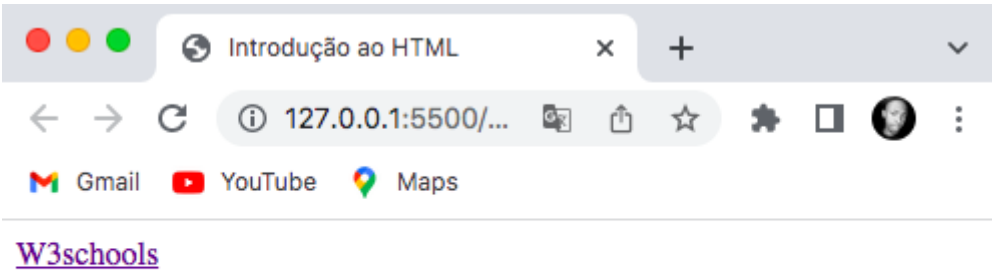
href: o endereço URL

target: especifica onde abrir o documento (opcional)
_self - Padrão. Abre o documento na mesma janela ou aba
_blank - Abre o documento em outra janela ou aba

Exemplo:

```
<a href="https://www.w3schools.com/" target="_blank">W3School</a>
```

Resultado:



Formulário

Formulários HTML são um dos principais pontos de interação entre um usuário e um web site ou aplicativo. Eles permitem que os usuários enviem dados para o web site. Na maior parte do tempo, os dados são enviados para o servidor da web, mas a página da web também pode interceptar para usá-los por conta própria.

Para construir o nosso formulário, vamos utilizar os seguintes elementos <form> , <label> , <input> , <textarea> , e <button> .

Todos formulários HTML começam com um elemento <form> como este:

```
<form action="/pagina-processa-dados-do-form" method="post">

</form>
```

Este elemento define um formulário. É um elemento de container como um elemento ou <p> , mas ele também suporta alguns atributos específicos para configurar a forma como o formulário se comporta. Todos os seus atributos são opcionais, mas é considerada a melhor prática sempre definir pelo menos o atributo action e o atributo method.

- O atributo *action* define o local (uma URL) em que os dados recolhidos do formulário devem ser enviados.
- O atributo *method* define qual o método HTTP para enviar os dados (ele pode ser "GET" ou "POST" (veja as diferenças aqui).

Se você quiser se aprofundar em como esses atributos funcionam, está detalhado no artigo Enviando e recebendo dados de um formulário (en-US).

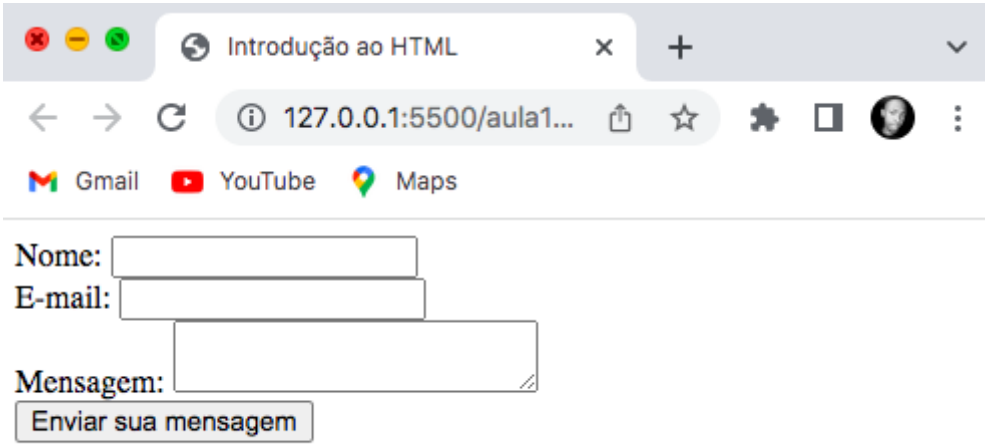
Adicionar campos com os elementos <label>, <input>, e <textarea>

O nosso formulário de contato é muito simples e contém três campos de texto, cada um com uma etiqueta. O campo de entrada para o nome será um campo básico texto de linha única("input"); o campo de entrada do e-mail será um campo de texto com uma única linha("input") que vai aceitar apenas um endereço de e-mail; o campo de entrada para a mensagem será um campo de texto de várias linhas("textarea").

Em termos de código HTML, teremos algo assim:

```html
<form action="/pagina-processa-dados-do-form" method="post">
    <div>
        <label for="name">Nome:</label>
        <input type="text" id="name" />
    </div>
    <div>
        <label for="mail">E-mail:</label>
        <input type="email" id="mail" />
    </div>
    <div>
        <label for="msg">Mensagem:</label>
        <textarea id="msg"></textarea>
    </div>
    <div class="button">
        <button type="submit">Enviar sua mensagem</button>
    </div>
</form>
```

Resultado:



Os elementos estão lá para estruturar nosso código e deixar a estilização mais fácil (ver abaixo). **Observe o uso do atributo *for* em todos os elementos <label> ; é uma maneira para vincular uma *label* à um campo do formulário**. Este atributo faz referência ao *id* do campo correspondente. Há algum benefício para fazer isso, é a de permitir que o usuário clique no rótulo para ativar o campo correspondente.

No elemento <input> , o atributo mais importante é o atributo **type**. Esse atributo é extremamente importante porque define a forma como o elemento <input> se comporta. Ele pode mudar radicalmente o elemento, então preste atenção a ele. Em nosso exemplo, nós usamos somente o **type="text"**, valor padrão para este atributo. Ele representa um campo de texto com uma única linha que aceita qualquer tipo de texto sem controle ou validação. Nós também usamos o **type="email"** que define um campo de texto com uma única linha que só aceita um endereço de e-mail bem-formados. O elemento <textarea> torna um campo de texto básico em uma espécie de campo "inteligente", que irá realizar alguns testes com os dados digitados pelo usuário.

Para permitir que o usuário envie seus dados depois de ter preenchido o formulário usamos o elemento <button> .

Um botão pode ser de três tipos: submit, reset, ou button.

1. Um clique sobre um botão de **submit** envia os dados do formulário para a página de web definida pelo atributo action do elemento <form> .
2. Um clique sobre um botão de ****reset** **redefine imediatamente todos os campos do formulário para o seu valor padrão.
3. Um clique em um botão do tipo **button** faz ...ops, nada! Mas é útil para construir botões personalizados com JavaScript, ou seja, ele pode assumir qualquer comportamento através desta linguagem.

Vídeos

A tag <video> é usada para embutir um vídeo na página.

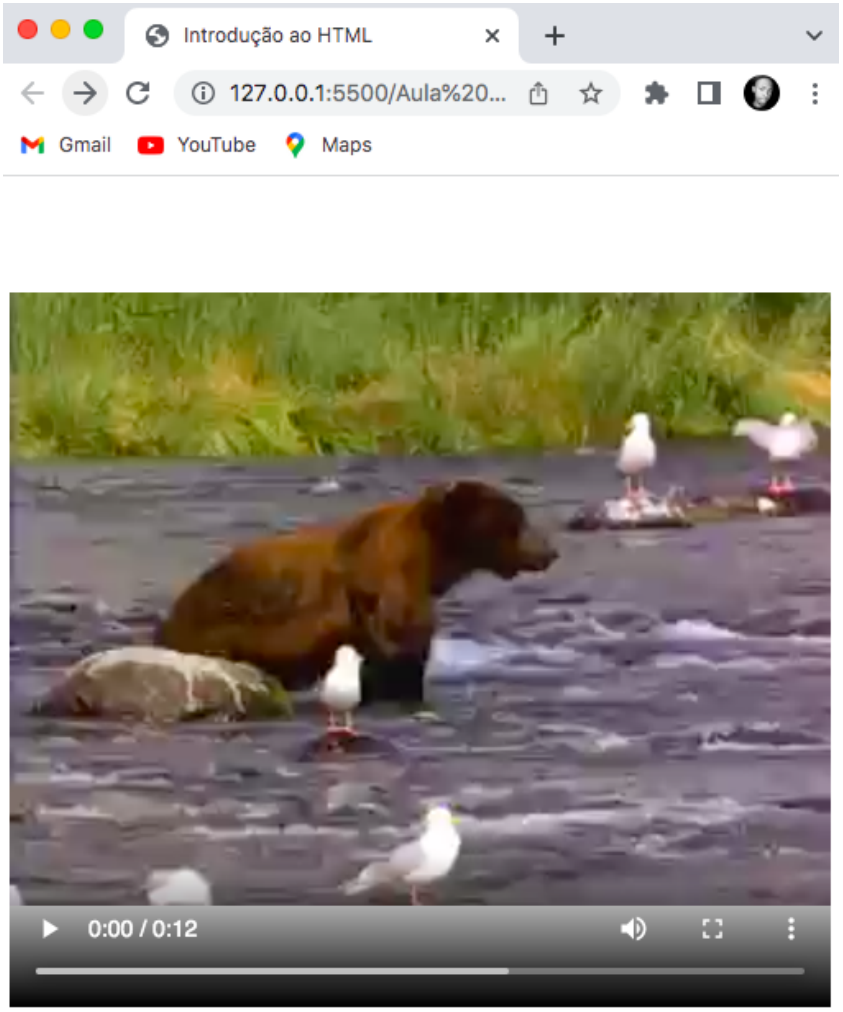
```
<video width="500px" height="500px" controls>
    <source src="movie.mp4" type="video/mp4">
    Navegador não suporta video
</video>
```

Explicação:

<video> A tag de vídeo
width atributo que define a largura do vídeo

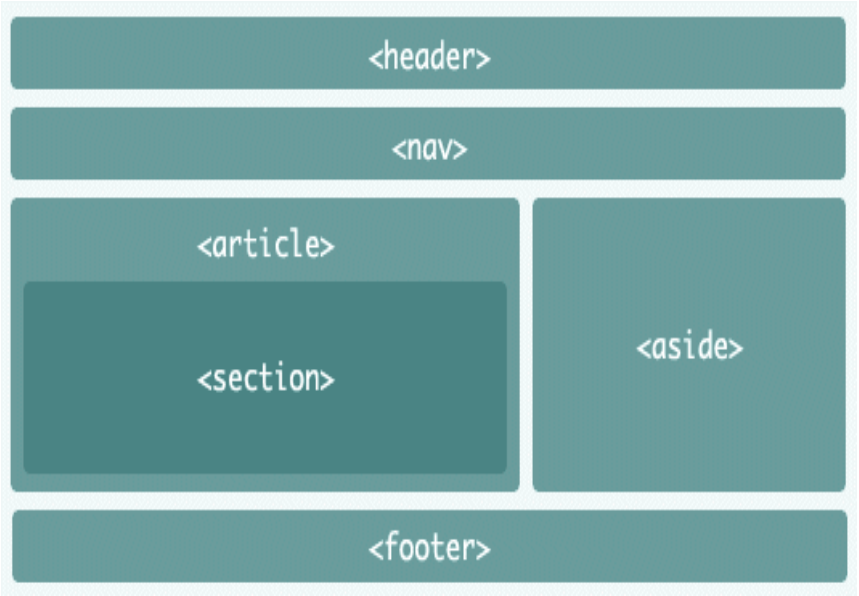
height atributo altura do vídeo
<source> tag que define o local onde está o vídeo
src atributo que que define o local do vídeo
type a extensão ou o formato do vídeo
O texto aparece caso o navegador não suporte o vídeo

Resultado:



Tags Semânticas

O HTML semântico tem como objetivo descrever o significado do conteúdo presente em documentos HTML, tornando-o mais claro tanto para programadores quanto para browsers e outras engines que processam essa informação.



<header>: é utilizado para representar o cabeçalho de um documento ou seção declarado no HTML.

<nav>: O elemento <nav> é utilizado quando precisamos representar um agrupamento de links de navegação, que, por sua vez, são criados com os elementos <ul>, <li> e <a>.

<article>: Utilizamos o elemento <article> quando precisamos declarar um conteúdo que não precisa de outro para fazer sentido em um documento HTML, por exemplo, um artigo em um blog.

<section>:O elemento <section> representa uma seção dentro de um documento e geralmente contém um título, o qual é definido por meio de um dos elementos entre <h1> e <h6>.

<footer>: O elemento <footer> representa um rodapé de um documento, como a área presente no final de uma página web. Normalmente é utilizado para descrever informações de autoria, como nome e contato do autor, e data de criação do conteúdo.

Links Úteis:

https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started

https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/Heading_Elements

https://developer.mozilla.org/pt-BR/docs/Learn/Forms/Your_first_form

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

https://www.w3schools.com/tags/default.asp