

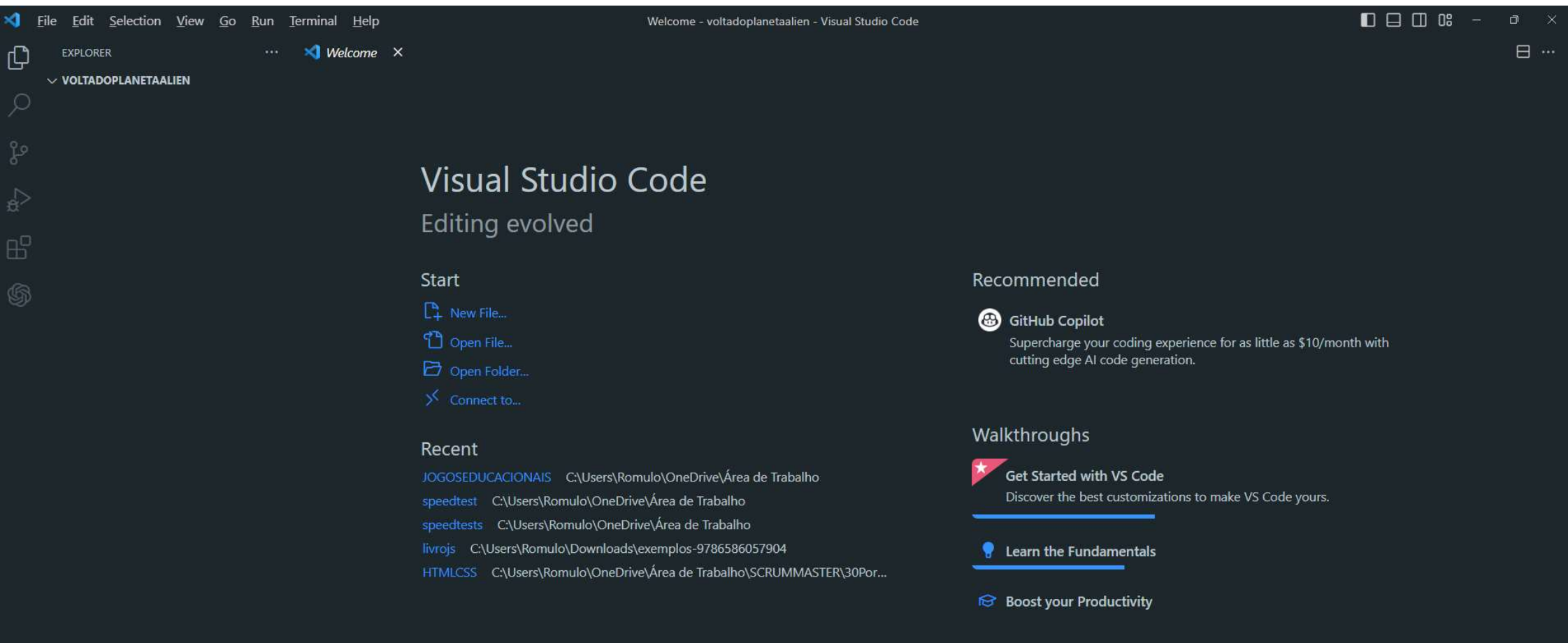


Esp. Rômulo C. Silvestre

Uma pasta foi criada →



A pasta **voltadoplanetaalien** foi aberta no VS CODE...





Foi criado um arquivo chamado index.html
Dentro da pasta "voltado planeta alien".

O arquivo index.html é o arquivo principal de um site,
página web ou sistema web ou aplicativo web.

A extensão é .html

Todo site que você conhece e já navegou usa
Essa linguagem de marcação nos bastidores.

Agora chegou a hora de você criar o seu site
e também aprender a aplicar algoritmos e lógica de programação
Com Java Script.



Tudo começou com o `<!DOCTYPE html>`

```
1 <!DOCTYPE html>
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

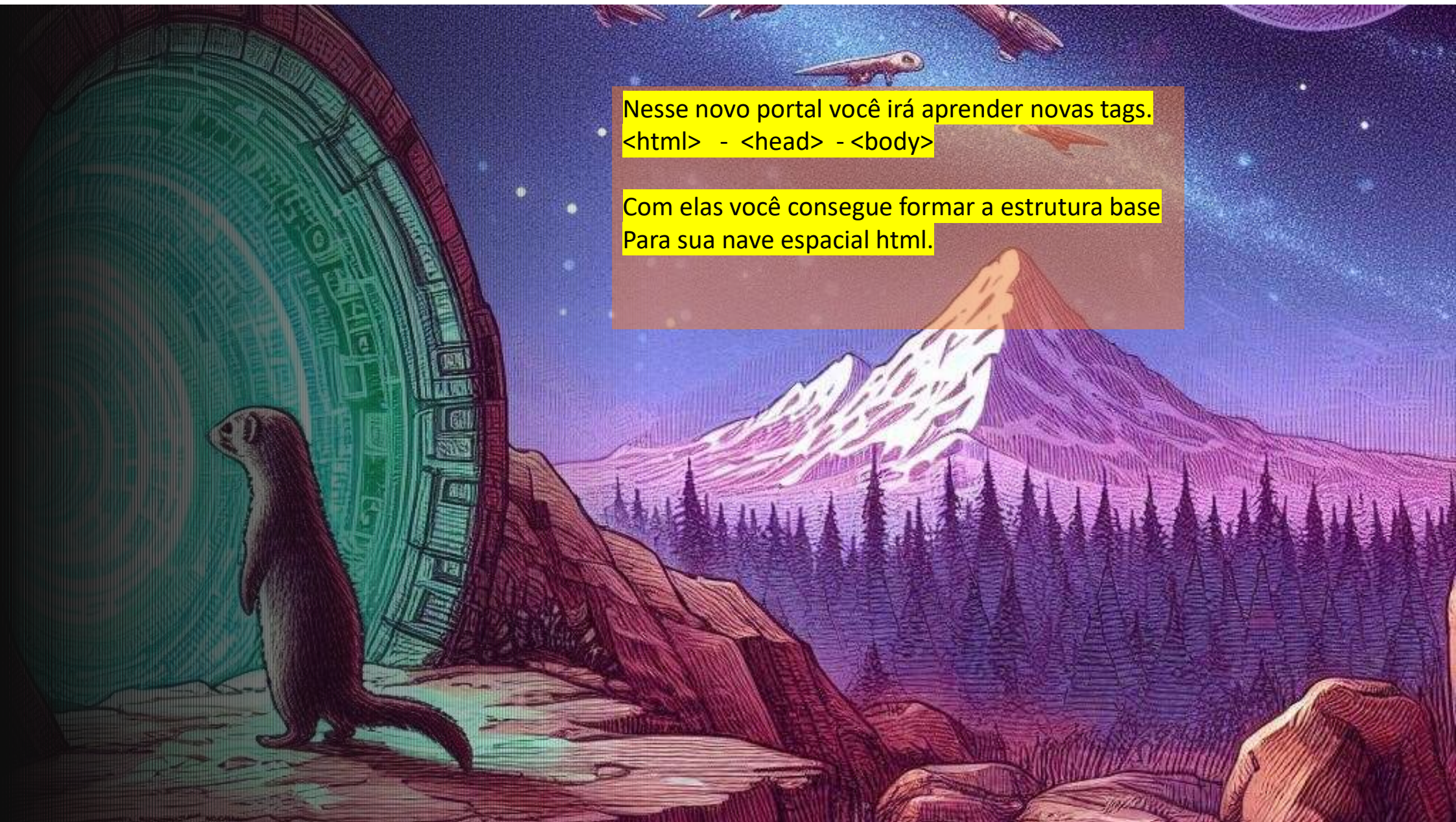
```
10 |
```

O DOCTYPE definia os tipos de documentos que eram utilizados no Planeta Alien.

A ferret is shown from the chest up, looking out of a large, circular window of a spaceship. The window reveals a vibrant green nebula with swirling patterns. The spaceship's interior is visible, showing various panels and lights. In the background, a large, orange planet and a smaller, grey planet are visible in the dark space.

Ao digitar a tag: `<!DOCTYPE html>`

O portal do HTML se abre para uma novo mundo de **TAGS**.



Nesse novo portal você irá aprender novas tags.
<html> - <head> - <body>

Com elas você consegue formar a estrutura base
Para sua nave espacial html.


```
1 <!DOCTYPE html>
```

```
2 <html>
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10 </html>
```

A tag é formada por um par ou não.
Nesse caso da tag html é um par:
<html>

</html>

Onde a primeira é o início (abertura) e a segunda o fechamento.
A linguagem de marcação HTML é baseada em TAGS. Quanto mais tags
aprender melhor e mais sites e aplicativos você pode construir.

2 <html>

3

4 <head>

5

6 </head>

7

8



```
2 <html>
```

```
3
```

```
4 <head>
```

```
5
```

```
6 </head>
```

```
7
```

```
8 <body>
```

```
9
```

```
10
```

```
11 </body>
```

```
12
```

```
13 </html>
```

Agora sim! Podemos explorar o Planeta Alien!

A exploração do Planeta Alien.

É baseada na exploração de tags.

Cada TAG possui um poder.

As funcionalidades das TAGS
São manipuladas por um motor
Java Script.

Essa manipulação no planeta Alien
Chamam de DOM
(Document Object Model)

Ao lado segue as primeiras tags
Que você deve explorar.
Com elas você pode construir
Uma NAVE HTML.

```
1 <!DOCTYPE html>
```

```
2 <html>
```

```
3
```

```
4 <head>
```

```
5
```

```
6 </head>
```

```
7
```

```
8 <body>
```

```
9
```

```
10
```

```
11 </body>
```

```
12
```

```
13 </html>
```





MEMORIZAR	COMPREENDER	APLICAR	ANALISAR	AVALIAR	CRIAR
Listar	Esquematizar	Utilizar	Resolver	Defender	Elaborar
Relembrar	Relacionar	Implementar	Categorizar	Delimitar	Desenhar
Reconhecer	Explicar	Modificar	Diferenciar	Estimar	Produzir
Identificar	Demonstrar	Experimentar	Comparar	Selecionar	Prototipar
Localizar	Parafrasear	Calcular	Explicar	Justificar	Traçar
Descrever	Associar	Demonstrar	Integrar	Comparar	Idear
Citar	Converter	Classificar	Investigar	Explicar	Inventar

Avaliação:

Questão 01

Listar as principais tags de uma estrutura base para construção de um documento HTML.

A tag que iremos explorar agora
É a HEAD

Essa tag ela é usada para fornecer
Informações sobre o documento
HTML.

Ela não exibe conteúdo.
Ela auxilia navegadores e motores
De busca entender e processar
O documento HTML.

Se você capturou essa TAG
Você consegue chamar os seguintes
Motores de busca:




```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Cadastro de Bloom</title>
7 </head>
```

- A tag **<meta charset="UTF-8">** em HTML é usada para especificar o conjunto de caracteres (character set) que o navegador deve usar para interpretar o texto contido na página da web .

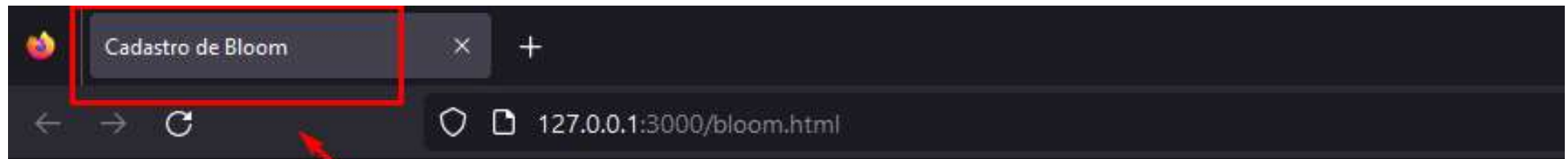


```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Responsividade

- A tag meta com os atributos name e content
- Tendo como valores:
 - name="viewport"
 - Content="width=device-width",initial-scale=1.0"
- Deixa o sistema web adaptativo para celulares e outros dispositivos.


```
<title>Cadastro de Bloom</title>
```



Cadastrar Bloom

Nome:

Lista de Bloom


ID Nome Ações

MEMORIZAR	COMPREENDER	APLICAR	ANALISAR	AVALIAR	CRIAR
Listar	Esquematizar	Utilizar	Resolver	Defender	Elaborar
Relembrar	Relacionar	Implementar	Categorizar	Delimitar	Desenhar
Reconhecer	Explicar	Modificar	Diferenciar	Estimar	Produzir
Identificar	Demonstrar	Experimentar	Comparar	Selecionar	Prototipar
Localizar	Parafrasear	Calcular	Explicar	Justificar	Traçar
Descrever	Associar	Demonstrar	Integrar	Comparar	Idear
Citar	Converter	Classificar	Investigar	Explicar	Inventar

Questão 02

Relembrar “tag ancora”, enfatizando a ancora entre páginas (hiperkink)

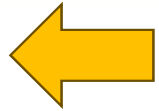
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>De Volta ao Planeta Alien</title>
7 </head>
8
9 <body>
10
11     <a href="bloom.html">Bloom</a>
12
13
14
15 </body>
16
17 </html>
```



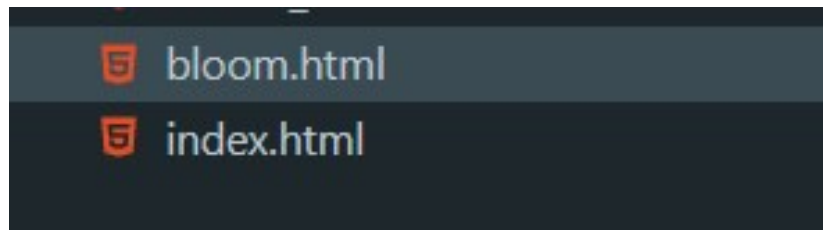


127.0.0.1:3000/index.html

Bloom



Cria-se um novo arquivo:





index.html bloom.html bloom_teste.html

bloom.html

1

!



Emmet Abbreviation



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Cadastro do Nível Bloom</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
12
13 |
```

Ao clicar no hiperlink ele irá direcionar para uma página

```
<a href="bloom.html">Bloom</a>
```

```
<!-- Formulário de Cadastro -->
<h2>Cadastrar Bloom</h2>
<form id="cadastroForm">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" required>
  <button type="submit">Cadastrar</button>
</form>
```



```
<!-- Formulário de Cadastro -->
```



```
<h2>Cadastrar Bloom</h2>
```

```
<form id="cadastroForm">
```

```
  <label for="nome">Nome:</label>
```

```
  <input type="text" id="nome" required>
```

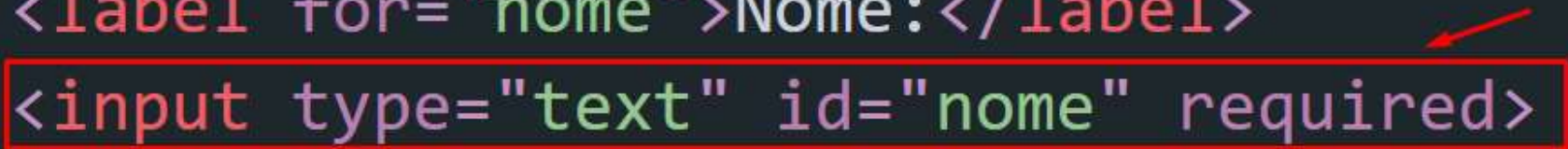
```
  <button type="submit">Cadastrar</button>
```

```
</form>
```

```
<!-- Formulário de Cadastro -->  
<h2>Cadastrar Bloom</h2>  
<form id="cadastroForm"> ←  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" required>  
  <button type="submit">Cadastrar</button>  
</form>
```

```
<!-- Formulário de Cadastro -->
<h2>Cadastrar Bloom</h2>
<form id="cadastroForm">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" required>
  <button type="submit">Cadastrar</button>
</form>
```

```
<!-- Formulário de Cadastro -->
<h2>Cadastrar Bloom</h2>
<form id="cadastroForm">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" required>
  <button type="submit">Cadastrar</button>
</form>
```




```
<!-- Formulário de Cadastro -->
<h2>Cadastrar Bloom</h2>
<form id="cadastroForm">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" required>
  <button type="submit">Cadastrar</button>
</form>
```



```
<!-- Formulário de Cadastro -->  
<h2>Cadastrar Bloom</h2>  
<form id="cadastroForm">  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" required>  
  <button type="submit">Cadastrar</button>  
</form>
```



```
8 <body>
9
10 <!-- Formulário de Cadastro -->
11 <h2>Cadastrar Bloom</h2>
12 <form id="cadastroForm">
13     <label for="nome">Nome:</label>
14     <input type="text" id="nome" required>
15     <button type="submit">Cadastrar</button>
16 </form>
17
18 </body>
19 </html>
```





127.0.0.1:3000/bloom.html

Cadastrar Bloom

Nome:

Cadastrar

MEMORIZAR	COMPREENDER	APLICAR	ANALISAR	AVALIAR	CRIAR
Listar	Esquematizar	Utilizar	Resolver	Defender	Elaborar
Relembrar	Relacionar	Implementar	Categorizar	Delimitar	Desenhar
Reconhecer	Explicar	Modificar	Diferenciar	Estimar	Produzir
Identificar	Demonstrar	Experimentar	Comparar	Selecionar	Prototipar
Localizar	Parafrasear	Calcular	Explicar	Justificar	Traçar
Descrever	Associar	Demonstrar	Integrar	Comparar	Idear
Citar	Converter	Classificar	Investigar	Explicar	Inventar


<table>

Demonstrar o uso de novas tags aplicando inovação com proatividade

```
20     <table>
21         <thead>
22             <tr>
23                 <th>ID</th>
24                 <th>Nome</th>
25                 <th>Ações</th>
26             </tr>
27         </thead>
28         <tbody id="bloomTableBody"></tbody>
29     </table>
```



```
28         <tbody id="bloomTableBody"></tbody>
29     </table>
30
31     <!-- JavaScript -->
32     <script>
33         // Array para armazenar os dados de Bloom
34         let blooms = [];
35
```



```
36 // Função para cadastrar Bloom
37 document.getElementById('cadastroForm').addEventListener('submit', function(e) {
38     e.preventDefault();
39     const nome = document.getElementById('nome').value;
40     const id = blooms.length + 1;
41     blooms.push({ id, nome });
42     exibirBloom();
43     document.getElementById('nome').value = '';
44 });
```

```
<script>
```

```
// Array para armazenar os dados de Bloom
```

```
let blooms = [];
```

```
// Função para cadastrar Bloom
```

```
document.getElementById('cadastroForm').addEventListener('submit', function(e) {
```

```
  e.preventDefault();
```

```
  const nome = document.getElementById('nome').value;
```

```
  const id = blooms.length + 1;
```

```
  blooms.push({ id, nome });
```

```
  exibirBloom();
```

```
  document.getElementById('nome').value = '';
```

```
});
```

```
// Função para exibir Bloom na tabela
function exibirBloom() {
  const tableBody = document.getElementById('bloomTableBody');
  tableBody.innerHTML = '';

  blooms.forEach(bloom => {
    const row = document.createElement('tr');
    row.innerHTML = `
      <td>${bloom.id}</td>
      <td>${bloom.nome}</td>
      <td>
        <button onclick="editarBloom(${bloom.id})">Editar</button>
        <button onclick="excluirBloom(${bloom.id})">Excluir</button>
      </td>
    `;
    tableBody.appendChild(row);
  });
}
```

```
// Função para editar Bloom
function editarBloom(id) {
  const nome = prompt('Editar Nome:', blooms.find(bloom => bloom.id === id).nome);
  if (nome !== null) {
    const index = blooms.findIndex(bloom => bloom.id === id);
    blooms[index].nome = nome;
    exibirBloom();
  }
}
```

```
// Função para excluir Bloom
function excluirBloom(id) {
  const confirmacao = confirm('Tem certeza que deseja excluir este Bloom?');
  if (confirmacao) {
    blooms = blooms.filter(bloom => bloom.id !== id);
    exibirBloom();
  }
}
```



```
// Inicializar a tabela com dados existentes, se houver  
exibirBloom();
```