

Apostila de Desenvolvimento Front-End I

Semana 3 – Estrutura e Semântica do HTML

Prof. Cleziel Franzoni da Costa

2025

Sumário

1	Passo 1: Criando a Estrutura Fundamental do HTML5	2
2	Passo 2: Dando Significado com Tags Semânticas	3
3	Passo 3: Validando o Código	4
4	Dica Extra: Extensões para VS Code	5

Introdução

Olá, turma! Bem-vindos a mais uma aula de **Desenvolvimento Front-End I**. Sou o professor Cleziel Franzoni da Costa, e hoje vamos mergulhar na prática, aplicando os conceitos de estrutura e semântica HTML que vimos na nossa apresentação.

O nosso grande objetivo é construir juntos um código HTML5 que sirva como um alicerce sólido e profissional para qualquer projeto web. Vamos criar uma página com uma estrutura semântica, clara e que siga as boas práticas que o mercado tanto valoriza. Lembrem-se sempre: um código de qualidade não é apenas aquele que funciona, mas aquele que é fácil de ler, de dar manutenção e que já nasce otimizado para os mecanismos de busca e para a acessibilidade.

Então, vamos lá! Abram o *Visual Studio Code* e vamos começar a criar nosso primeiro arquivo, o `index.html`.

1 Passo 1: Criando a Estrutura Fundamental do HTML5

Todo documento HTML5 profissional se inicia com uma estrutura padrão. Essa base garante compatibilidade entre os navegadores, melhora a acessibilidade e a otimização para os buscadores. Vamos começar com o esqueleto do nosso documento.

Nosso código inicial

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale
    =1.0">
  <title>Minha Página Institucional</title>
</head>
<body>
</body>
</html>
```

Analizando o código

- `<!DOCTYPE html>`: declaração fundamental que instrui o navegador a renderizar a página em HTML5.
- `<html lang="pt-BR">`: tag raiz, com atributo `lang` informando o idioma principal (Português-Brasil).
- `<head>`: contém metadados importantes para o navegador.
- `<meta charset="UTF-8">`: garante exibição correta de caracteres especiais.
- `<meta name="viewport"...>`: torna a página responsiva em celulares e tablets.
- `<title>`: define o título exibido na aba do navegador.
- `<body>`: abriga todo o conteúdo visível da página.

2 Passo 2: Dando Significado com Tags Semânticas

Agora, vamos elevar o nível do nosso código. Em vez de usar apenas `<div>`, aplicaremos as tags semânticas do HTML5 para descrever cada seção da nossa página.

Código semântico

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Minha Página Institucional</title>
</head>
<body>

  <header>
    <h1>Logo da Empresa</h1>
    <nav>
      <ul>
```

```

        <li><a href="#home">Home</a></li>
        <li><a href="#sobre">Sobre</a></li>
        <li><a href="#contato">Contato</a></li>
    </ul>
</nav>
</header>

<main>
    <article>
        <h2>Sobre a Nossa Empresa</h2>
        <p>Este parágrafo apresenta a nossa empresa. Utilizamos a
            tag <code><p></code> para blocos de texto.</p>
        <p>Lembrem-se: nunca usem <code><br></code> para espaçamento.
            Usem CSS!</p>
    </article>
</main>

<footer>
    <p>&copy; 2025 Minha Empresa. Todos os direitos reservados.<
        /p>
</footer>

</body>
</html>

```

Analizando a semântica aplicada

- <header>: agrupa logo e menu inicial.
- <h1>: título principal da página (um por página).
- <nav>: destaca a área de navegação.
- <main>: delimita o conteúdo central da página.
- <article>: agrupa um conteúdo independente (neste caso, “Sobre a Empresa”).
- <footer>: rodapé com direitos autorais.

3 Passo 3: Validando o Código

Um passo crucial é a validação no W3C Validator, que garante padrões corretos e ausência de erros.

Procedimento

1. Acesse <https://validator.w3.org/>.
2. Clique em *Validate by Direct Input*.
3. Cole o código e clique em *Check*.

Se tudo estiver correto, a mensagem exibida será: *"Document checking completed. No errors or warnings to show."*

4 Dica Extra: Extensões para VS Code

- **Live Server:** cria servidor local com atualização automática.
- **Prettier:** mantém o código organizado e padronizado.
- **Auto Rename Tag:** altera abertura/fechamento de tags em conjunto.
- **HTML CSS Support:** preenchimento inteligente para classes e IDs.
- **W3C Validation:** validação dentro do próprio VS Code.

Próxima Aula

Na próxima aula, vamos iniciar a estilização com **CSS**, transformando nossa estrutura semântica em um design profissional.