

Programação em JavaScript

ServiceNow - Deloitte

Rodrigo Costa

Fundamentos Web

Módulo 1

HTML5 Semântico e Acessível

Sessão 2

Objetivo geral

Dotar os formandos dos conhecimentos fundamentais de HTML5, permitindo-lhes compreender a sua estrutura, o significado das tags semânticas e a importância da acessibilidade na construção de páginas web.

Objetivos específicos

Ao final da sessão, os formandos deverão ser capazes de:

- Estruturar um documento HTML5 corretamente, respeitando a hierarquia de títulos, a organização do conteúdo e a semântica dos elementos;
- Utilizar elementos HTML semânticos de forma adequada (header, main, section, article, nav, footer);
- Incorporar conteúdos multimédia (imagens, áudio, vídeo e iframes) em páginas web;
- Criar formulários HTML funcionais e acessíveis;
- Aplicar princípios básicos de acessibilidade (A11y) em documentos HTML,;
- Compreender a estrutura do DOM (Document Object Model) e a relação entre os elementos HTML numa página web.

Fundamentos do HTML Semântico

O que é HTML e para que ele serve

- HTML significa HyperText Markup Language;
- É uma linguagem de marcação, não de programação e serve para:
 - Estruturar conteúdos;
 - Organizar informação;
 - Dar significado aos elementos da página.

O papel do HTML numa página web

- É base de qualquer página web;
- Trabalha em conjunto com:
 - CSS → aparência;
 - JavaScript → comportamento.

O que é HTML semântico?

- HTML semântico usa elementos que descrevem o significado do conteúdo;
- Cada tag comunica:
 - O tipo de conteúdo;
 - A sua função na página.

Exemplo simples de semântica

```
html

<div class="header">
  <div class="menu">
    <div>Início</div>
    <div>Sobre</div>
    <div>Contacto</div>
  </div>
</div>

<div class="content">
  <div class="title">Bem-vindo ao site</div>
  <div class="text">
    Este é o conteúdo principal da página.
  </div>
</div>

<div class="footer">
  <div>© 2026</div>
</div>
```

HTML não semântico

```
html

<header>
  <nav>
    <ul>
      <li><a href="#">Início</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </nav>
</header>

<main>
  <section>
    <h1>Bem-vindo ao site</h1>
    <p>Este é o conteúdo principal da página.</p>
  </section>
</main>

<footer>
  <p>© 2026</p>
</footer>
```

HTML semântico

Por que a semântica é importante?

- Benefícios do HTML semântico:
 - Melhora a **acessibilidade**:
 - Ajuda leitores de ecrã;
 - Facilita manutenção do código.

HTML não semântico vs HTML semântico

HTML não semântico

- Uso excessivo de `<div>` e ``;
- Estrutura pouco clara;
- Dificulta a acessibilidade;
- Código menos legível e difícil de manter.

HTML semântico

- Utilização de elementos com significado;
- Estrutura clara e organizada;
- Facilita a acessibilidade;
- Código mais limpo e fácil de compreender e manter.

Elementos de texto e significado

- `<p>` — parágrafo;
- `` — importância;
- `` — ênfase;
- `<a>` — Ligação;

Estrutura Semântica de uma página HTML5

Estrutura base de um documento HTML5

<> index.html X

<> index.html > ...

```

1  <!DOCTYPE html> <!--Informa que é HTML5-->
2  <html lang="en"> <!--Elemento raiz do documento HTML-->
3  <head> <!--Informações para o navegador-->
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body> <!--Conteúdo visível da página-->
9
10 </body>
11 </html>

```

Layout semântico: pensar em blocos

- Uma página web pode ser vista como blocos de conteúdo:
 - Topo;
 - Navegação;
 - Conteúdo principal;
 - Conteúdo complementar;
 - Rodapé.

Principais elementos semânticos do HTML5

- **<header>** — Cabeçalho da página ou de uma secção (ex: logótipo, título);
- **<nav>** — Área de navegação;
- **<main>** — Conteúdo principal da página (deve existir apenas um por página);
- **<section>** — Agrupamento temático de conteúdo (normalmente tem um título);
- **<article>** — Conteúdo independente e reutilizável;
- **<aside>** — Conteúdo complementar;
- **<footer>** — Rodapé da página ou de uma secção (ex: copyright, contactos, links).

Exemplo de estrutura completa

```
<> index.html X
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head> ...
7  </head>
8  <body>
9      <header>...</header>
10     <nav>...</nav>
11
12     <main>
13         <section>
14             <article>...</article>
15         </section>
16     </main>
17
18     <footer>...</footer>
19
20 </body>
21 </html>
```

Embedding Media

O que é Embedding media?

- É o ato de incorporar conteúdos multimédia numa página web;
- Exemplos:
 - imagens;
 - Áudio;
 - Vídeo;

Embedding media: Imagens

```

12 <main>
13 <section>
14   <article>
15     
16   </article>
17 </section>
18 </main>

```

- src: caminho da imagem;
- alt: descrição textual da imagem;
- O alt é um elemento importante de acessibilidade, ele é lido pelos leitores de ecrã.

Uma boa descrição é importante

```

```

Má utilização do atributo alt

```

```

Boa utilização do atributo alt

Embedding media: Audio

```
<audio controls>  
  <source src="audio.mp3" type="audio/mpeg">  
  O seu navegador não suporta áudio.  
</audio>
```

- controls: habilita os controles acessíveis por teclado;
- O texto é um recurso em caso de falha na reprodução;
- O type define o tipo de ficheiro que será reproduzido.

Embedding media: Vídeo

```
<video controls width="400">  
  <source src="video.mp4" type="video/mp4" tracks="pt"  
  alt="Vídeo de apresentação">  
  O seu navegador não suporta vídeo.  
</video>
```

- controls: habilita os controles acessíveis por teclado;
- O texto é um recurso em caso de falha na reprodução;
- tracks permite a inclusão de legendas
- O type define o tipo de ficheiro que será reproduzido.

Embedding media: Iframe

```
<iframe  
  src="https://www.youtube.com/embed/VIDEO_ID"  
  title="Vídeo institucional"  
  width="560"  
  height="315"  
  allowfullscreen>  
</iframe>
```

- O <iframe> permite incorporar conteúdo externo;
- title atua no papel do alt nesse contexto, para a acessibilidade;
- allowfullscreen é um atributo que permite que essa janela seja exibida em fullscreen.

Embedding media: Cuidados com o iframe

- Por ser um recurso externo o iframe exige alguns cuidados, pois ele pode:
 - Afetar o desempenho;
 - Trazer risco de segurança;

Media e semântica

- Media deve estar:
 - Dentro de <figure>;
 - Com <figcaption> quando fizer sentido.

```
<figure>  
    
  <figcaption>Vendas em 2025</figcaption>  
</figure>
```

Formulários

Para que servem os formulários?

- Permitem recolher dados do utilizador e são usados em:
 - Login;
 - Registo;
 - Pesquisas;
 - Contactos;
 - Uploads.

Estrutura base de um formulário

```
<form action="/enviar" method="post">  
| <!-- campos do formulário -->  
</form>
```

- <form> define a área de submissão;
- action indica o destino dos dados;
- method define o método http que será usado para enviar os dados;
- <label> melhora acessibilidade e usabilidade.

A importância do <label>

- for liga o rótulo ao campo
 - Nota, o valor precisa ser idêntico ao ID do input;
- Ajuda leitores de ecrã;
- Aumenta a área clicável.

```
<label for="email">Email:</label>  
<input type="email" id="email" name="email">
```

Tipos comuns de inputs

- Vantagens:
 - Validações automáticas;
 - Teclados adaptados em mobile;
 - Melhora a experiência sem utilização de JavaScript.

```
<input type="email">  
<input type="date">  
<input type="tel">
```


Validações nativas em HTML

```
<input type="text" required>  
<input type="text" minlength="3" maxlength="20">  
<input type="text" pattern="[0-9]{9}">
```

- O atributo **required** impede submissões vazias;
- **minlength** e **maxlength** controlam o tamanho do input;
- **pattern** permite validações baseadas em expressões regulares;
- As mensagens de erro são geridas pelo próprio navegador.

Upload de ficheiros e botão de submit

```
<button type="submit">Enviar</button>  
<input type="file">  
<input type="file" accept=".pdf,.jpg,.png">  
<input type="file" multiple>
```

- **type="submit"**: Ativa o envio do formulário com o teclado;
- **type="file"** permite selecionar ficheiros do dispositivo;
- **accept** restringe os tipos de ficheiro permitidos;
- **multiple** permite selecionar mais do que um ficheiro;
- O **backend** é sempre responsável pela validação final.

Exemplo de um formulário completo

```
<form>
  <label for="email">Email:</label>
  <input type="email" id="email" required>

  <label for="data">Data de nascimento:</label>
  <input type="date" id="data">

  <button type="submit">Enviar</button>
</form>
```

Acessibilidade

O que é acessibilidade (A11y)

- Acessibilidade na web, o que significa criar páginas que:
 - Todas as **pessoas** conseguem usar;
 - Independentemente de limitações físicas, sensoriais ou cognitivas;
 - Incluem utilizadores que:
 - Usam leitores de ecrã;
 - Navegam apenas com teclado;
 - Têm baixa visão ou daltonismo.

Por que a acessibilidade é importante?

- Inclusão social;
- Requisitos legais (em muitos contextos);
- Melhor experiência para todos;
- Código mais limpo e estruturado.

Tecnologias assistivas

- Leitores de ecrã;
- Navegação por teclado;
- Ampliadores de ecrã.

HTML semântico e acesibilidade

- Elementos semânticos:
 - Ajudam leitores de ecrã;
 - Melhoram a compreensão do conteúdo;
 - Criam pontos de navegação.



Introdução ao ARIA – Accessible rich internet applications

- ARIA fornece informações adicionais às tecnologias assistivas;
- aria-label descreve elementos sem texto visível;
- Deve ser utilizado apenas quando não existe um elemento HTML semântico adequado;
- O uso incorreto de ARIA pode prejudicar a acessibilidade.

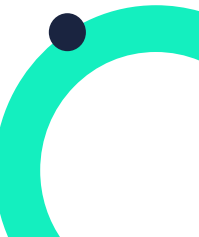




Aria-label

- Fornece descrição a elementos sem texto visível;
- Muito usado em ícones;
- Invisível para quem vê, essencial para quem ouve.

```
<button aria-label="Fechar">  
  ✕  
</button>
```



Role e estados ARIA

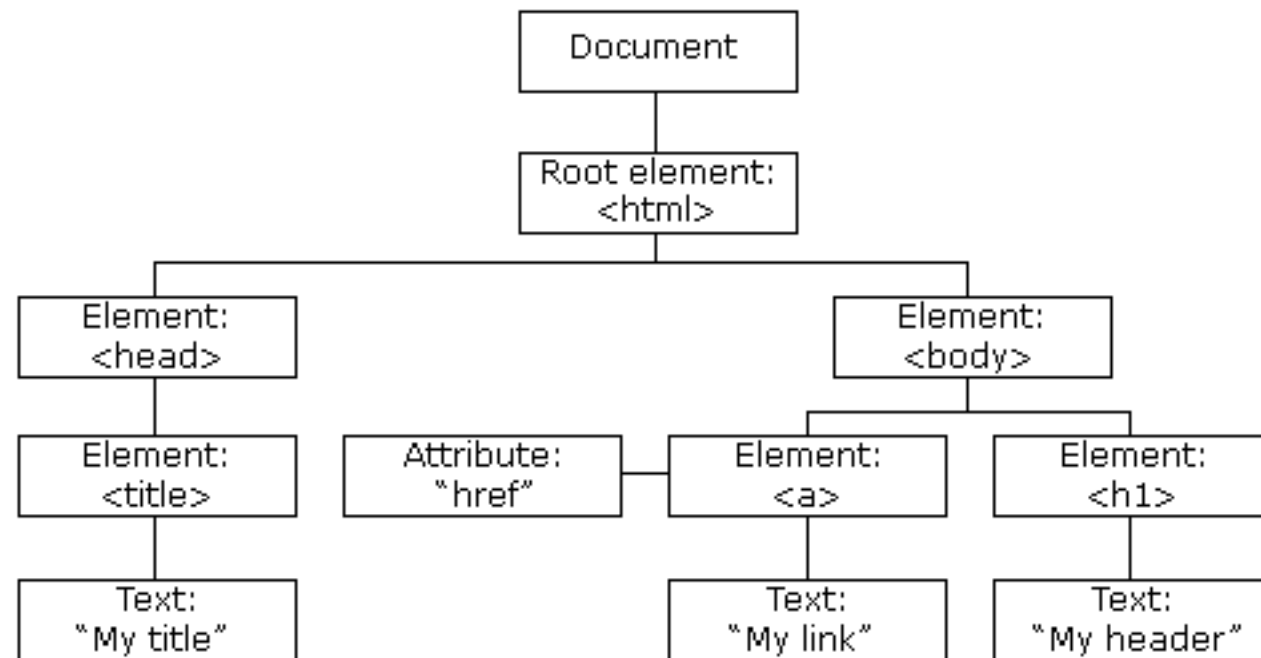
```
<div role="button">Enviar</div>  
<div aria-hidden="true"></div>
```

- role: define a função do elemento;
- aria-hidden: oculta de leitores de ecrã.

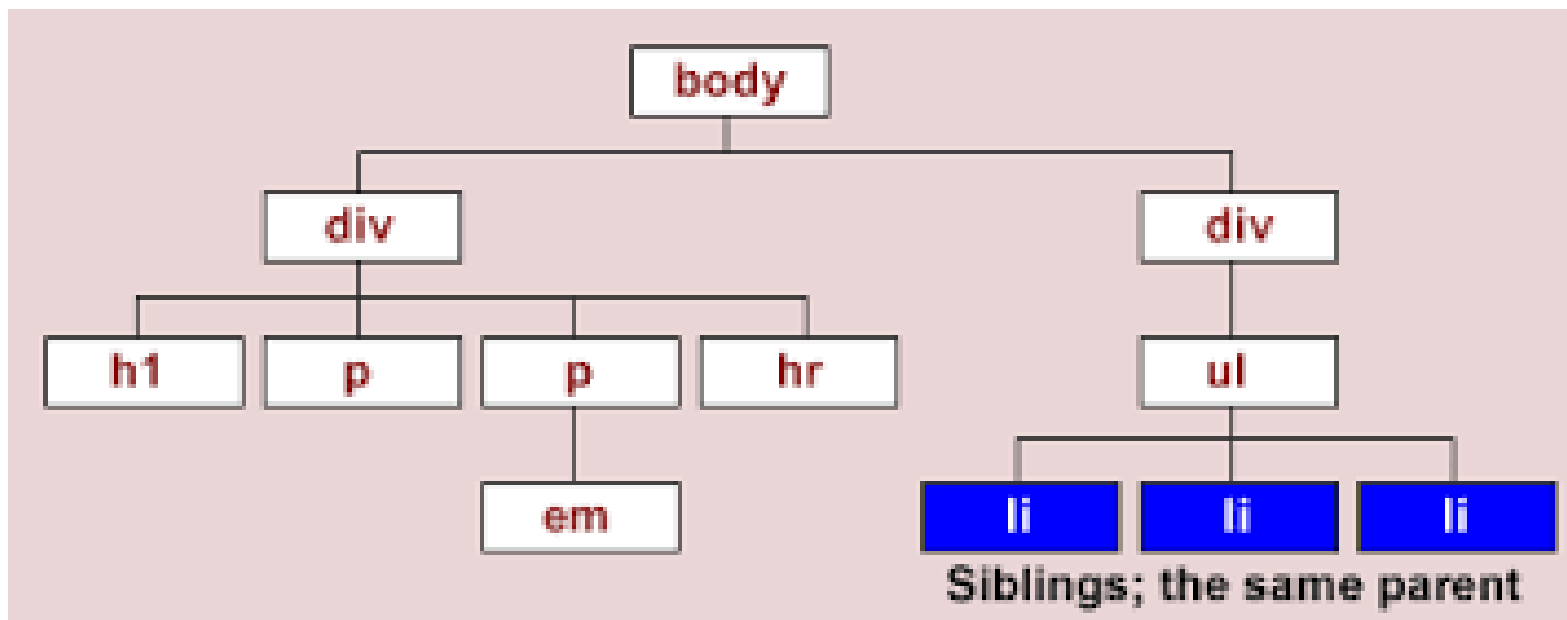
Document Object Model (DOM)

O que é o DOM

- O DOM representa a página como uma estrutura em árvore;
- Cada elemento HTML torna-se um nó do DOM;
- Os elementos possuem relações de pai, filho e irmãos;
- A estrutura do DOM influencia acessibilidade e manipulação com JavaScript.



Estrutura do DOM



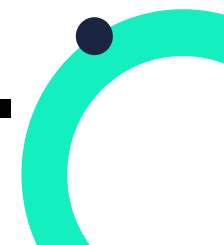
Por que o DOM é importante?

- JavaScript interage com O DOM;
- Acessibilidade depende do DOM;
- Estrutura mal feita, gera um problema em cadeia.



Síntese

- Semântica do HTML;
- Acessibilidade;
- Embedding Media;
- Formulários;
- DOM.



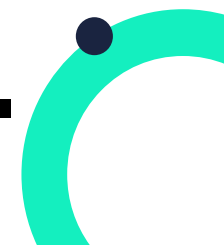
Conclusão



Conclusão

“O poder da Web está na sua universalidade. O acesso por todos, independentemente da deficiência, é um aspeto essencial.”

Tim Berners-Lee



Programação em JavaScript

ServiceNow - Deloitte

Rodrigo Costa