

15 DE FEVEREIRO DE 2022 / #HTML5

Elementos semânticos do HTML5 explicados



Gustavo Goulart Baptista

Artigo original: [Semantic HTML5 Elements Explained](#)
Traduzido e adaptado por [Gustavo Goulart Baptista](#)

Elementos semânticos do HTML5 são aqueles que descrevem claramente seu significado de uma maneira que seja legível tanto para humanos quanto para máquinas.

Elementos como `<header>`, `<footer>` e `<article>` (cabeçalho, rodapé e artigo, respectivamente, traduzidos para o português) são considerados semânticos, pois eles descrevem com precisão o propósito do elemento e o tipo de conteúdo dentro dele.

O que são elementos semânticos?

O HTML foi originalmente criado como uma linguagem de marcação para descrever documentos no início da internet. Conforme a internet cresceu e foi sendo adotada por cada vez mais pessoas, algumas mudanças precisaram acontecer.



A internet, que foi originalmente pensada para compartilhamento de artigos científicos, agora se tornou um local onde as pessoas

Devido à web não ser ter sido inicialmente construída pensando em design, os programadores usaram alguns truques diferentes para arrumar tudo de diversas maneiras. Ao invés de usar a tag `<table>` `</table>` (tabela, em inglês) para descrever informações usando uma tabela, programadores a usavam para posicionar outros elementos dentro de uma página.

Conforme as interfaces progrediram visualmente, os programadores começaram a usar tags genéricas e não semânticas como `<div>` (divisão, em inglês). Eles costumavam dar a esses elementos atributos de classe (`class`) ou `id` que descrevessem seu propósito. Por exemplo, ao invés de `<header>` era comumente escrito algo como `<div class="header">`

Como o HTML5 é algo ainda relativamente novo, esse uso de elementos não semânticos ainda é muito comum em websites atualmente.

Lista de elementos semânticos:

Os elementos semânticos adicionados ao HTML5 foram:

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`

Aprenda a programar — currículo gratuito de 3 mil horas

- `<nav>`
- `<section>`
- `<summary>`
- `<time>`

Elementos como `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>` e `<footer>` funcionam mais ou menos como elementos `<div>`. Eles agrupam outros elementos juntos em seções de uma página. Uma tag `<div>` poderia conter qualquer tipo de informação. Contudo, é fácil identificar que tipo de informação seria incluída em uma tag semântica como `<header>`.

Um exemplo de uma interface com elementos semânticos da w3schools

Por que usar elementos semânticos?

Para ver os benefícios dos elementos semânticos, aqui temos dois trechos de código em HTML. Esse primeiro bloco de código utiliza elementos semânticos:

```
<header></header>
<section>
  <article>
    <figure>
      <img>
      <figcaption></figcaption>
    </figure>
  </article>
</section>
<footer></footer>
```

Aprenda a programar — currículo gratuito de 3 mil horas

semânticos:

```
<div id="header"></div>
<div class="section">
  <div class="article">
    <div class="figure">
      <img>
      <div class="figcaption"></div>
    </div>
  </div>
</div>
<div id="footer"></div>
```

O primeiro é muito mais **fácil de ler**. Essa, provavelmente, é a primeira coisa que você notará quando for ler o primeiro trecho de código, que está usando elementos semânticos. Esse é um exemplo pequeno. Porém, como programador, você lerá centenas ou milhares de linhas de código. Quanto mais fácil for ler e entender o código, mais fácil será seu trabalho.

Ele tem **maior acessibilidade**. Você não é o único que acha que elementos semânticos são mais fáceis de entender. Os algoritmos de indexação em sites de busca e as tecnologias assistivas (como leitores de tela para usuários com deficiência visual) também entenderão melhor o contexto e conteúdo do seu website, entregando uma melhor experiência para os usuários.

Geralmente, elementos semânticos também resultam em um **código mais consistente**. Quando criamos um cabeçalho usando elementos não semânticos, diversos programadores podem escrever algo como `<div class="header">`, `<div id="header">`, `<div class="head">` ou, simplesmente, `<div>`. Existem várias maneiras

Aprenda a programar — currículo gratuito de 3 mil horas
programamos o conteúdo em partes com elementos semânticos,
facilitamos o trabalho para todos.

Em outubro de 2014 o HTML4 foi atualizado para o HTML5, que incluiu alguns novos elementos "semânticos". Até hoje, alguns de nós ainda nos confundimos com relação aos diversos elementos que não parecem demonstrar nenhuma grande mudança.

`<section>` e `<article>`

"Qual a diferença?", você pode estar se perguntando. Ambos os elementos são usados para seccionar o conteúdo – e, sim, eles definitivamente podem ser intercambiáveis. É uma questão de analisar cada situação. O HTML4 nos oferecia somente um tipo de elemento container, a `<div>`. Embora ela ainda seja usada em HTML5, ainda temos a possibilidade de substituí-la em duas maneiras, usando `<section>` e `<article>`.

Os elementos `<section>` e `<article>` são conceitualmente similares e intercambiáveis. Para decidir qual deles você deve escolher, observe o que citaremos abaixo:

1. Um `<article>` tem como objetivo ser independentemente distribuível ou reutilizável.
2. Uma `<section>` é um temático agrupamento de conteúdos.

```
<section>
  <p>Histórias novas</p>
  <section>
    <p>Notícias</p>
    <article>História 1</article>
    <article>História 2</article>
    <article>História 3</article>
  </section>
```

Aprenda a programar — currículo gratuito de 3 mil horas

```
<article>História 3</article>
</section>
</section>
```

<header> e <hgroup>

O elemento <header> é geralmente encontrado no topo de um documento, uma seção ou um artigo e geralmente contém o cabeçalho principal da página e algumas navegações e barras de busca.

```
<header>
  <h1>Company A</h1>
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/about">Sobre</a></li>
    <li><a href="/contact">Contato</a></li>
  </ul>
  <form target="/search">
    <input name="q" type="search" />
    <input type="submit" />
  </form>
</header>
```

O elemento <hgroup> deve ser usado quando você quer um cabeçalho principal com um ou mais subcabeçalhos.

```
<hgroup>
  <h1>Cabeçalho 1</h1>
  <h2>Subcabeçalho 1</h2>
  <h2>Subcabeçalho 2</h2>
</hgroup>
```

Aprenda a programar — currículo gratuito de 3 mil horas

cabeçalhos, ou seja, elementos de `<h1>` (*header1*, do inglês) a `<h6>` (*header6*, do inglês) e outros `<hgroup>` .

`<aside>`

O elemento `<aside>` é destinado a conteúdos que não fazem parte do fluxo do texto, mas que, ainda sim, estão relacionados de alguma forma. Esse `<aside>` poderia ser utilizado como uma barra lateral (*sidebar*) para o seu conteúdo principal.

```
<aside>
  <p>Essa é uma sidebar, por exemplo, uma definição de uma terminologi
</aside>
```



Antes do HTML5, nosso menus eram criados com as tags `` e `` . Agora, junto com esses elementos, nós podemos separar nossos itens no menu com um elemento `<nav>` para a navegação entre as páginas. Você pode ter mais de um elemento `<nav>` na página. É comum, por exemplo, ter uma navegação global percorrendo o topo da página (dentro do elemento `<header>`) e uma navegação local em uma sidebar (dentro de um elemento `<aside>`).

```
<nav>
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/about">Sobre</a></li>
    <li><a href="/contact">Contato</a></li>
  </ul>
</nav>
```

Aprenda a programar — currículo gratuito de 3 mil horas

Se existe um `<header>` (cabeçalho, em inglês), deve existir um `<footer>` (rodapé, em inglês). Uma tag `<footer>` é geralmente encontrada no final de um documento, uma seção ou um artigo. Assim como no `<header>` o conteúdo é geralmente uma metainformação, por exemplo, com detalhes sobre o autor, informações legais, e/ou links e informações relacionadas. Também é válido incluir elementos `<section>` dentro de um *footer*.

```
<footer>&copy; Empresa A</footer>
```

`<small>`

O elemento `<small>` (pequeno, na tradução para o português) sempre aparece dentro de um elemento `<footer>` ou `<aside>`, normalmente incluindo informações de direitos autorais ou avisos legais, além de outras "letras miúdas". No entanto, esse elemento não é feito para deixar seu texto menor. É apenas uma descrição de seu conteúdo e não faz nenhuma prescrição visual.

```
<footer><small>&copy; Empresa A</small> Data</footer>
```

`<time>`

O elemento `<time>` permite que uma data no formato ISO 8601 não ambígua seja adicionada a uma versão mais amigável para leitura humana daquela data.

Aprenda a programar — currículo gratuito de 3 mil horas

Por que se preocupar com `<time>` ? Embora humanos possam eliminar ambiguidades e ler o tempo de maneiras normais, os computadores podem ler datas em formato ISO 8601 e ver a data, a hora e o fuso horário.

`<figure>` e `<figcaption>`

`<figure>` serve para inserir conteúdos de imagem, enquanto `<figcaption>` é destinado às legendas dessa imagem.

```
<figure>
  Arte de capa de Middle Earth: Shadow of Mordor</figcapti
</figure>
```

Aprenda mais sobre os novos elementos do HTML5:

- A [w3schools](#) mantém uma descrição simples e clara de muitos dos novos elementos e de como/onde eles devem ser usados.
- A [MDN](#) também é uma grande referência para todos os elementos HTML e vai mais a fundo em cada um deles.



Gustavo Goulart Baptista

Ler [mais publicações](#).

Aprenda a programar — currículo gratuito de 3 mil horas

Aprenda a programar gratuitamente. O plano de estudos em código aberto do freeCodeCamp já ajudou mais de 40.000 pessoas a obter empregos como desenvolvedores.

[Comece agora](#)

O freeCodeCamp é uma organização sem fins lucrativos 501(c)(3), isenta de impostos e apoiada por doações (Número de identificação fiscal federal dos Estados Unidos: 82-0779546).

Nossa missão: ajudar as pessoas a aprender a programar de forma gratuita. Conseguimos isto criando milhares de vídeos, artigos e lições de programação interativas, todas disponíveis gratuitamente para o público. Também temos milhares de grupos de estudo do freeCodeCamp em todo o mundo.

As doações feitas ao freeCodeCamp vão para nossas iniciativas educacionais e ajudam a pagar servidores, serviços e a equipe.

Você pode fazer [uma doação dedutível de impostos aqui](#).

Guias de tendências

Nova aba em HTML	Jogo do dinossauro
Máscaras de sub-rede	Menu iniciar
40 projetos em JavaScript	Arrays vazios em JS
Tutorial de button onClick	Caracteres especiais
Bot do Discord	Python para iniciantes
Centralizar em CSS	Provedores de e-mail
Excluir pastas com o cmd	15 portfólios
Imagens em CSS	Node.js no Ubuntu
25 projetos em Python	10 sites de desafios
Excluir branches	Clonar branches

Aprenda a programar — currículo gratuito de 3 mil horas

[Axios em React](#)[SQL em Python](#)[ForEach em JavaScript](#)[Interpretadas x compiladas](#)[Fotos do Instagram](#)[Imagens SVG em HTML e CSS](#)

Nossa organização sem fins lucrativos

[Sobre](#) [Rede de ex-alunos](#) [Código aberto](#) [Loja](#) [Apoio](#) [Patrocinadores](#)[Honestidade acadêmica](#) [Código de conduta](#) [Política de privacidade](#) [Termos de serviço](#)[Política de direitos de autor](#)