

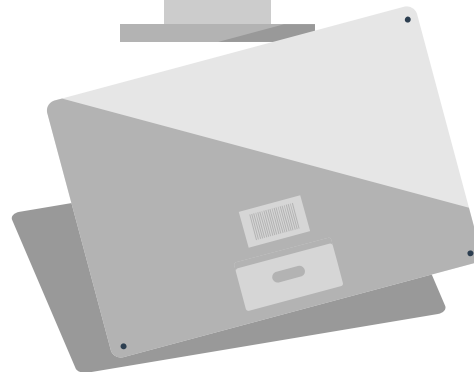
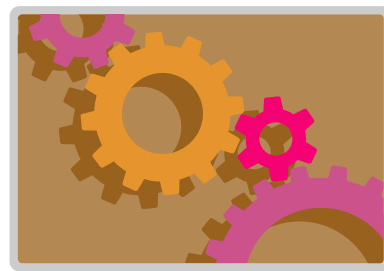
# SUMÁRIO

<b>Introdução a programação .....</b>	<b>3</b>
<b>Linguagem de Marcação de Hypertexto .....</b>	<b>4</b>
Introdução ao HTML5 .....	6
Definição e atributos de uma tag .....	9
Tags estruturais, textuais e semânticas .....	11
Tags estruturais.....	11
Tags textuais .....	15
Tags semânticas .....	18
<header> .....	22
<nav> .....	23
<section> .....	23
<article> .....	24
<aside> .....	24
<footer> .....	25
<b>Referências.....</b>	<b>26</b>

# INTRODUÇÃO A PROGRAMAÇÃO

O desenvolvimento web é dividido em duas partes: *back-end* e *front-end*. O desenvolvimento *front-end*, também chamado de programação do lado do cliente, é o que acontece no navegador. É tudo o que o usuário vê e com o qual interage. Já o desenvolvimento *back-end*, também chamado de programação do lado do servidor, ocorre no servidor e no servidor de banco de dados, por exemplo. O servidor é a máquina que funciona nos bastidores para potencializar os recursos sofisticados com os quais os usuários interagem ao lado do cliente.

## Back-end



## Front-end



Os fundamentos do desenvolvimento web de *back-end* incluem linguagens como Java, Ruby, Python, PHP, Net etc. Já as linguagens de *front-end* mais comuns são HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript, embora o JavaScript também possa ser utilizado no *back-end* quando se utiliza a plataforma Node.js, por exemplo.

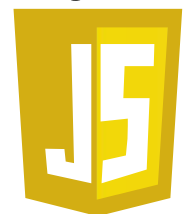
HTML



CSS



JS



Neste estudo, iremos focar no desenvolvimento *front-end*, mais especificamente com o HTML5. A seguir comece estudando a linguagem de marcação de hipertexto.

# LINGUAGEM DE MARCAÇÃO DE HYPERTEXTO

HTML é a linguagem de marcação que usamos para estruturar o conteúdo que consumimos na web. O HTML é servido ao navegador de diferentes maneiras. Ele pode ser gerado por uma aplicação do lado do servidor que o constrói dependendo da solicitação ou dos dados da sessão. Pode ser gerado por uma aplicação JavaScript do lado do cliente que gera HTML na hora. No caso mais simples, pode ser armazenada em um arquivo e servida ao navegador por um servidor web.

HTML é um padrão definido pelo WHATWG, um acrônimo para *Web Hypertext Application Technology Working Group*, uma organização formada por pessoas que trabalham nos mais populares navegadores da web. Isto significa que é basicamente controlado pelo Google, Mozilla, Apple e Microsoft. No passado, o W3C (*World Wide Web Consortium*) era a organização encarregada de criar o padrão HTML. O controle passou informalmente do W3C para o WHATWG quando ficou claro que o impulso do W3C em direção ao XHTML não era uma boa ideia (WHATWG, 2022).



## Curiosidade

Se você nunca ouviu falar em XHTML, aqui vai uma pequena história. No início dos anos 2000, todos nós acreditávamos que o futuro da web era o XML. Então o HTML passou para uma linguagem de marcação XML. Foi uma grande mudança. Tínhamos que conhecer e respeitar mais regras. Regras mais rígidas. Eventualmente, os fornecedores de navegadores perceberam que este não era o caminho certo para a web, e se afastaram, criando o que agora é conhecido como HTML5.

O W3C não concordou realmente em abrir mão do controle do HTML, e durante anos tivemos dois padrões concorrentes, cada um deles visando ser o padrão oficial. Eventualmente, em 28 de maio de 2019, o W3C oficializou que a versão HTML verdadeira era a publicada pela WHATWG. Tivemos a versão HTML 1 em 1993. Seguiu-se o HTML 2 em 1995. Tivemos o HTML 3 em janeiro de 1997, e o HTML 4 em dezembro de

1997. Mais de 20 anos se passaram, tivemos toda essa evolução do XHTML, e eventualmente chegamos a esta versão do HTML5, que na verdade não é mais apenas HTML. HTML5 é um termo que agora define todo um conjunto de tecnologias, que inclui HTML, mas acrescenta muitas APIs (*Application Programming Interface*) e padrões como WebGL, SVG (*Scalable Vector Graphics*) e muito mais (WHATWG, 2022).



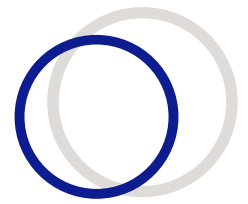
Por convenção, um arquivo HTML é salvo como um arquivo com extensão “.html” ou “.htm”. Dentro desse arquivo, nós organizamos o conteúdo usando tags. As tags envolvem o conteúdo, e cada tag dá um significado especial ao texto que ela envolve. Você confere esses conceitos mais à frente.

Quando uma página HTML é servida pelo navegador, as tags são interpretadas, e o navegador torna os elementos de acordo com as regras que definem sua aparência visual. Algumas dessas regras são embutidas, como por exemplo, como uma lista renderiza ou como um link é sublinhado em azul. Algumas outras regras são definidas com a utilização de CSS (*Cascading Style Sheets*).



O HTML não é apresentável. Ele não se preocupa com a aparência das coisas. Em vez disso, ele está preocupado com o que significam. Cabe ao navegador determinar como as coisas parecem, com as diretrizes definidas por quem constrói a página, com a linguagem de marcação CSS.





## Introdução ao HTML5

HTML5 é um tipo de linguagem de computador cuja principal função é apresentar o conteúdo da web de forma padronizada. Esta linguagem de marcação é usada para oferecer suporte a uma variedade de elementos em um site, desde a aparência até a funcionalidade. Essa linguagem é compreendida pelos navegadores e seus elementos servem como instruções para que as páginas da web sejam exibidas de maneira uniforme nos navegadores modernos.



**HTML5:**  
desenvolvimento  
do website

Para descrever a estrutura de uma página web, adicionamos código às palavras que queremos que apareçam na página. Você pode conferir um código HTML para uma página web na sequência. Não se preocupe com o que o código significa ainda. Comece a analisá-lo com mais detalhes.

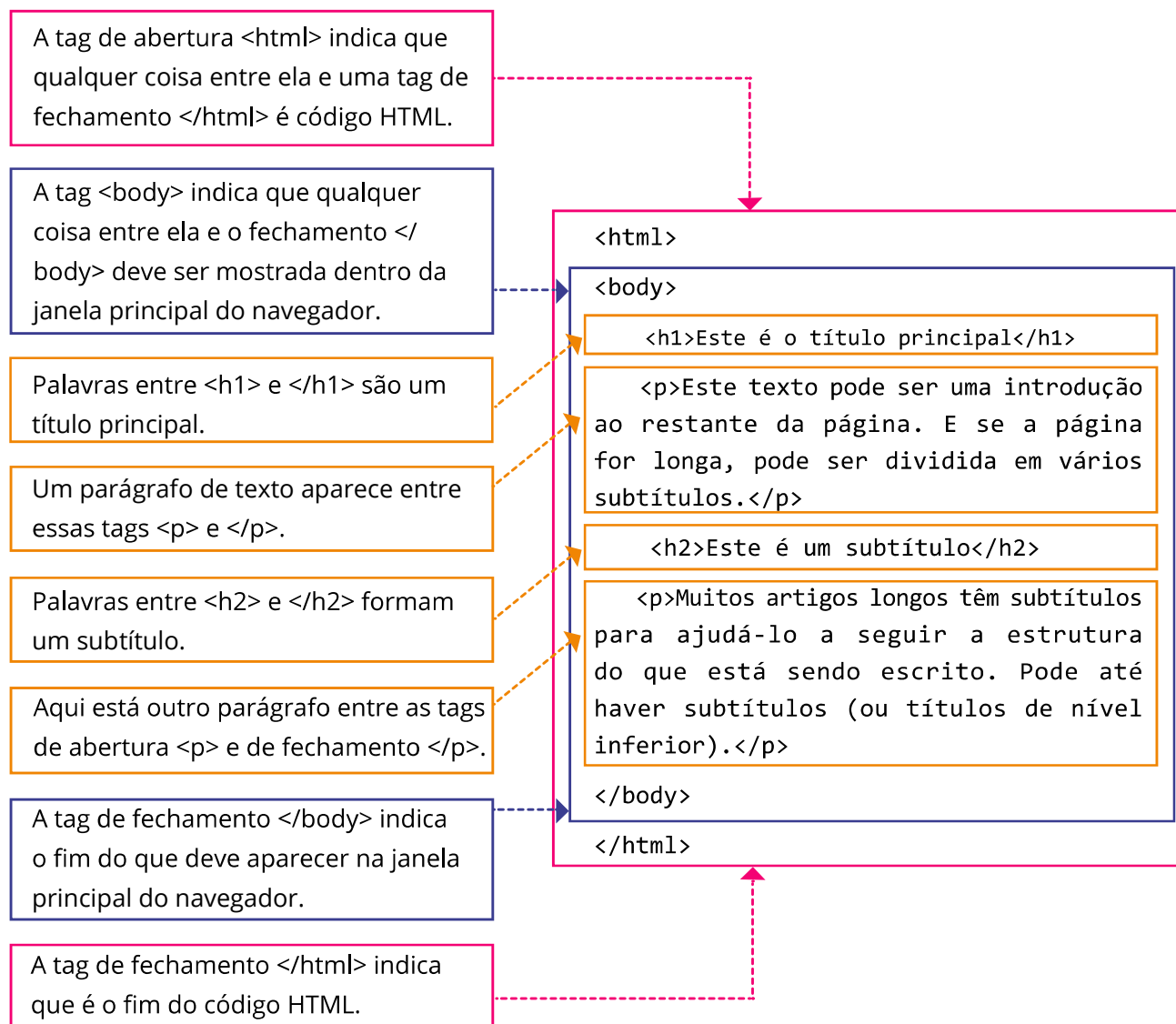


Figura 1 - Exemplo estrutural do HTML  
Fonte: do Autor (2022)

### Dica



Para facilitar a criação e execução de nossos projetos, iremos utilizar a ferramenta CodePen que você pode conferir clicando no link ou acessando seu celular do QR code:

<https://codepen.io/pen>

CodePen é um editor de código on-line popular usado por muitos desenvolvedores ao redor do mundo. CodePen é indicado apenas para projetos front-end e suporta HTML, CSS e JavaScript. Caso você deseje salvar seus arquivos e projetos de forma on-line, será necessário criar uma conta nesta plataforma. Para isso, você pode se inscrever com um endereço de e-mail em que criará um nome de usuário e senha ou, até mesmo, poderá autenticar-se por meio de redes sociais, por exemplo. A ferramenta possui três abas: HTML, CSS e JavaScript. Desta forma, iremos inserir nossos códigos diretamente na guia HTML para visualizarmos imediatamente o resultado renderizado da página na plataforma. Isso deixará nosso trabalho mais prático e rápido. Caso seja do seu interesse, você poderá utilizar outra ferramenta interessante e mais robusta: o *Visual Studio Code* que você pode conferir clicando no link ou acessando seu celular do QR code:



<https://code.visualstudio.com>

O *Visual Studio Code* é um editor de código aberto usado principalmente para corrigir e reparar erros de codificação de aplicativos da web e de nuvem. O VS Code é desenvolvido pela Microsoft e é compatível com os sistemas operacionais macOS, Linux e Windows. As ferramentas do VS Code podem ser usadas para aprimorar a funcionalidade de qualquer código escrito.



Confira na imagem a seguir, o resultado da inserção do código-fonte na aba HTML da ferramenta CodePen, bem como o resultado da renderização deste código-fonte no navegador por meio da plataforma.



```
1 <html>
2
3 <body>
4   <h1>Este é o título principal</h1>
5   <p>Este texto pode ser uma introdução ao restante da página. E se a página
   for longa, pode ser dividida em vários subtítulos.</p>
6   <h2>Este é um subtítulo</h2>
7   <p>Muitos artigos longos têm subtítulos para ajudá-lo a seguir a estrutura
   do que está sendo escrito. Pode até haver subtítulos (ou títulos de nível
   inferior).</p>
8 </body>
9
10 </html>
```

**Este é o título principal**

Este texto pode ser uma introdução ao restante da página. E se a página for longa, pode ser dividida em vários subtítulos.

**Este é um subtítulo**

Muitos artigos longos têm subtítulos para ajudá-lo a seguir a estrutura do que está sendo escrito. Pode até haver subtítulos (ou títulos de nível inferior).

Figura 2 - Renderização do conteúdo em HTML

Fonte: do Autor (2022)

A seguir, confira como definir atributos.

## Definição e atributos de uma tag

O código HTML é composto de caracteres que ficam dentro de colchetes angulares. Esses são chamados de elementos HTML. Os elementos são geralmente compostos por duas tags: uma **tag de abertura** e uma **tag de fechamento**. A tag de fechamento tem uma barra extra. Cada elemento HTML comunica ao navegador algo sobre as informações que ficam entre suas tags de abertura e fechamento.

A maioria das tags vem em pares com uma tag de abertura e uma tag de fechamento. A tag de fechamento é escrita da mesma forma que a tag de abertura, mas com uma barra "/". Existem algumas tags de fechamento automáticas, o que significa que elas não precisam de uma tag de fechamento separada, pois não contêm nada nelas.

Os elementos têm uma tag de partida e uma tag de fechamento. Portanto, um elemento constitui o pacote completo: tag de início, conteúdo de texto (e possivelmente outros elementos) e tag de fechamento. Se um elemento não



tiver uma tag de fechamento, ela é escrita apenas com a tag inicial, e não pode conter nenhum conteúdo de texto. Dito isto, o termo tag ou elemento significa o mesmo, exceto se mencionarmos explicitamente tag de início ou tag de fim.

A tag inicial de um elemento pode conter trechos especiais de informações que podemos anexar, chamados de atributos. Os atributos têm a sintaxe "chave="valor"", por exemplo:

```
<p class="nome-classe">Um parágrafo de conteúdo.</p>
```

Você também pode usar aspas simples, mas usar aspas duplas em HTML é uma boa convenção.

Os atributos "class" e "id" são os atributos mais comuns que você vai encontrar. Eles têm um significado especial, e são úteis tanto em CSS quanto em JavaScript. A diferença entre os dois é que um "id" é único no contexto de uma página web, logo ele não pode ser duplicado. As classes ou "class", por outro lado, podem aparecer múltiplas vezes em múltiplos elementos. Além disso, um "id" tem apenas um valor. Já uma "class" pode conter vários valores, separados por um espaço. Confira outro exemplo:

```
<p class="nome-classe outra-classe">Outro parágrafo.</p>
```

### Curiosidade

Os símbolos "<!--" e "-->" representam tags de comentário em HTML. Para comentar em HTML, insira informações entre as tags "<!--" e "-->". Perceba que os navegadores não mostrarão essas notas. Comentar em HTML permite que os desenvolvedores deixem notas sobre seu código, sua funcionalidade ou indiquem mudanças necessárias para o futuro. A sintaxe geral para um comentário HTML é assim: "<!-- Aqui vem um comentário -->".

# Tags estruturais, textuais e semânticas

Na sequência, iremos conhecer as principais tags do HTML que são divididas principalmente em tags estruturais, textuais e semânticas.

## Tags estruturais

Estas são as tags básicas que devem fazer parte de cada página da web. Elas informam ao navegador que o documento é uma página da web. Existem quatro conjuntos de tags HTML que são necessários para formar a estrutura básica de cada arquivo HTML:

Tag	Propósito
<code>&lt;html&gt;...&lt;/html&gt;</code>	Esta é a tag pai (elemento raiz) para qualquer documento HTML. Todo o bloco de código HTML está incorporado nesta tag. Isso basicamente define o documento como página da web. Também identifica o início e o fim do documento HTML. Todas as outras tags devem estar entre as tags <code>&lt;html&gt;</code> .
<code>&lt;head&gt;...&lt;/head&gt;</code>	Esta tag fornece informações gerais sobre o documento, como seu título e links para folhas de estilo (se houver). Essas informações não são exibidas na página da web. O cabeçalho contém informações sobre o documento que não aparecerão na página real, como o título do documento, o autor, qual folha de estilo usar e também metatags.
<code>&lt;title&gt;...&lt;/title&gt;</code>	A tag <code>&lt;title&gt;</code> define o título que aparecerá na barra de título do seu navegador. O <code>&lt;title&gt;</code> deve aparecer entre a tag <code>&lt;head&gt;</code> .
<code>&lt;body&gt;...&lt;/body&gt;</code>	A tag <code>&lt;body&gt;</code> contém todas as informações e outros conteúdos visíveis na página. Todas as suas imagens, links e texto simples devem ficar entre as tags <code>&lt;body&gt;</code> e <code>&lt;/body&gt;</code> .

Quadro 1 - Tags estruturais

Fonte: do Autor (2022)

A tag de início `<html>` é usada no início do documento, logo após a declaração do tipo de documento. Dentro do elemento `<html>` temos dois elementos: `<head>` e `<body>`.

Dentro da tag `<head>` teremos tags que são essenciais para criar uma página, como o título, os metadados e o CSS interno ou externo e JavaScript. A maioria dos itens não aparecem diretamente na página, mas apenas ajudam o

navegador a exibi-las corretamente. Dentro do corpo da página, ou seja, a tag `<body>`, teremos o conteúdo da página. Aquilo que é visível.

Essas quatro tags são especiais. Deve haver apenas um conjunto de cada e elas devem estar na ordem correta, como no exemplo seguinte:

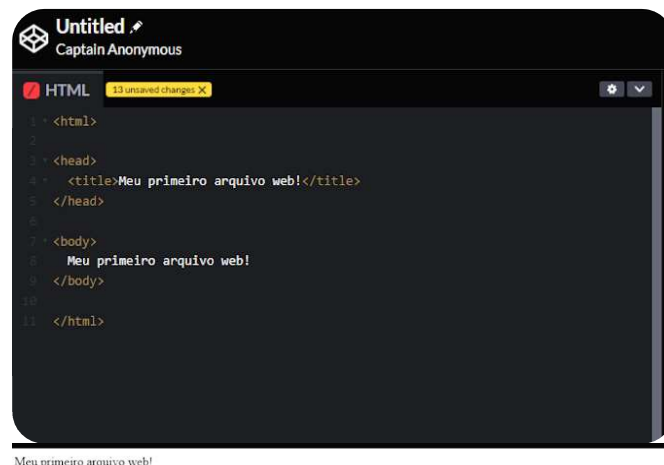


Figura 3 - Exemplos de tags estruturais do HTML

Fonte: do Autor (2022)

Todo arquivo HTML precisa ter um DOCTYPE declarado. O DOCTYPE não é realmente um elemento ou tag HTML. Ele permite que o navegador saiba como o documento deve ser interpretado, indicando qual versão ou padrão de HTML (ou outra linguagem de marcação) está sendo usado. Isso precisa ser a primeira coisa em seu documento, antes dos elementos `<html>` ou `<head>`. Além disso, não há tag de fechamento. Confira um exemplo para documentos HTML5:



Como complemento, sempre use um atributo de idioma na tag <html> para declarar o idioma padrão do texto na página. Isso é herdado por todos os outros elementos. De acordo com a recomendação do W3C, é necessário indicar o idioma principal de cada página da web com o atributo “lang” dentro da tag <html>, como <html lang=“en”>, que define a linguagem em inglês, enquanto <html lang=“pt-br”> define a linguagem em português do Brasil. Essa definição não é exclusiva do HTML5, pode ser utilizado por exemplo no XHTML.

As Metatags estão ocultas na página, mas elas, assim como todo o código HTML em uma página, podem ser visualizadas selecionando alguma opção relacionada à visualização do código-fonte da página no menu do navegador. Metatags contém uma descrição geral da página, palavras-chave e informações de direitos autorais, por exemplo. Abaixo, estão as metatags de informações utilizadas em uma página HTML:

Tag	Propósito
<base>	Isso é usado para especificar a URL base do site.
<meta>	Contém informações como data de publicação, nome do autor etc.
<style>	Contém as informações relacionadas à aparência da página da web.
<link>	É usado para indicar links externos, principalmente folhas de estilo. É uma tag vazia e contém apenas atributos.
<script>...</script>	Usado para adicionar trechos de código em JavaScript para tornar uma página da web dinâmica.

Quadro 2 - Metatags

Fonte: do Autor (2022)

Confira também, no exemplo, algumas metatags.



```
1 <html>
2
3 <head>
4   <meta name="Autor" content="Edinilson Vida">
5   <meta charset="UTF-8">
6   <base href="https://www.site.com.br" target="_blank">
7   <title>Página Inicial</title>
8   <link rel="stylesheet" href="estilo.css">
9   <script src="codigo.js">
10     var a = 10;
11   </script>
12 </head>
13
14 <body>
15   Este é o conteúdo do corpo do documento HTML.
16 </body>
17
18 </html>
```

Este é o conteúdo do corpo do documento HTML.

Figura 4 - Exemplos de metatags

Fonte: do Autor (2022)



### Dica

Caso as acentuações das palavras não estejam sendo exibidas corretamente na página web, inclua a tag "<meta charset='UTF-8'>" no código-fonte da página. UTF-8 é a codificação para Unicode, que atribui um valor único para todos os caracteres e emojis. UTF-8 é definido como a codificação de caracteres padrão para HTML5 usada para exibir uma página HTML perfeitamente. O atributo charset é usado para realizar uma codificação de caracteres para o HTML. Assim, a tag meta "<meta charset='UTF-8'>" especifica a codificação de caracteres para o documento HTML, habilitando o uso dos caracteres universais padrão do Unicode, sendo também compatível com o ASCII (tabela com acentos e uso de ç).

Qualquer informação dentro das tags <head> é ocultada e usada por navegadores e rastreadores da web. Isso é conhecido como metadados. Observe também que algumas tags estão aninhadas dentro de outras tags. Por exemplo: <title> está aninhado dentro <head> e <head> dentro de <html>. A imagem seguinte mostra como os elementos podem existir dentro de outros elementos:

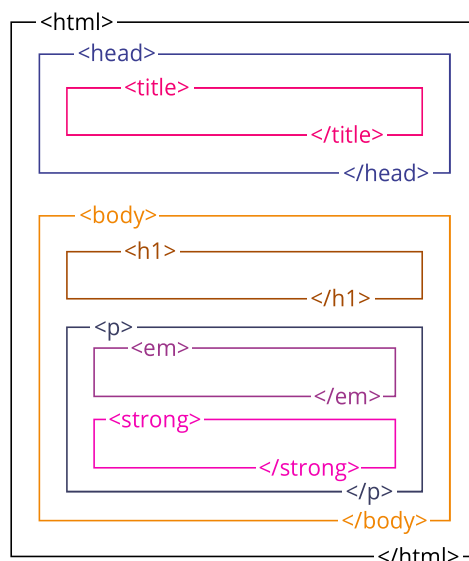


Figura 5 - Aninhamento de tags

Fonte: do Autor (2022)

## Tags textuais

Existem várias tags HTML que você pode usar para imprimir texto em uma página. Confira a seguir as tags de formatação de texto:

Tag	Propósito
<p>...</p>	Parágrafo do texto.
 	Quebra de linha (forçar uma nova linha).
<b>...</b>	Deixa o texto em negrito.
<i>...</i>	Deixa o texto em itálico.
<u>...</u>	Sublinha o texto.
<strike>...</strike>	Risca o texto.
<strong>...</strong>	Deixa o texto em negrito (igual às tags <b>...</b>).
<em>...</em>	Deixa o texto em itálico (igual às tags <i>...</i>).
<h#>...</h#>	Tag de título (# pode variar de 1 a 6).
<small>...</small>	Torna o texto pequeno.
<sup>...</sup>	Exibe o texto como sobrescrito.
<sub>...</sub>	Exibe o texto como subscrito.
<blockquote>...</blockquote>	Exibe o texto como um bloco de código distinto.

Quadro 3 - Tags textuais

Fonte: do Autor (2022)

Confira a seguir um exemplo utilizando as tags de formatação textuais.

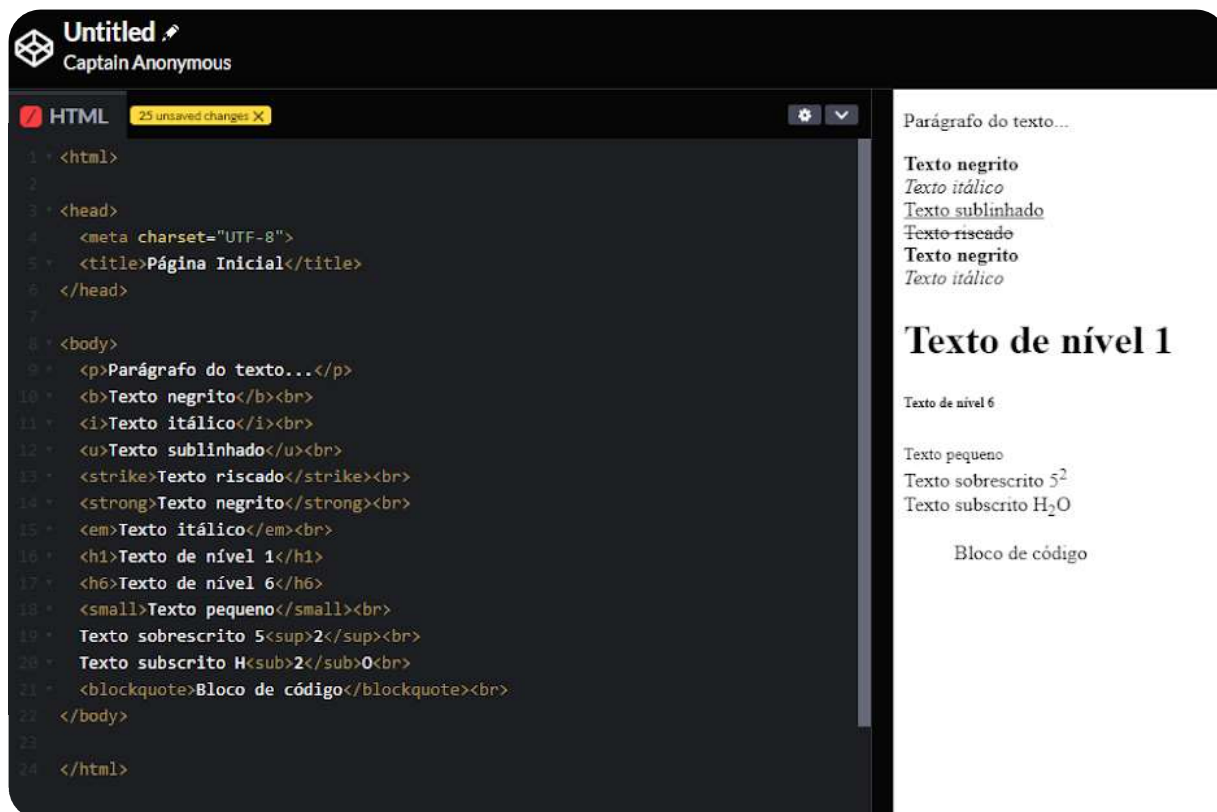


Figura 6 - Exemplos de tags textuais

Fonte: do Autor (2022)

A tag `<big>` foi usada no HTML4 para definir um texto maior, mas não é mais suportada no HTML5. A tag `<q>...</q>` é usada para fornecer pequenas citações ao conteúdo HTML, neste caso, o navegador simplesmente coloca uma citação ao redor do conteúdo dentro da tag. Essa tag também é conhecida como citação em linha, o que significa que ela não passará para a próxima linha.

Por fim, caso você deseje inserir um link em alguma palavra ou expressão de um documento HTML, devemos usar a tag `<a>`. A tag `<a>` permite que o usuário defina um link para um documento, página web etc. Entenda alguns de seus atributos no quadro:



Atributos	Propósito	Valor
href	Menciona o local para onde o link deve redirecionar.	URL de destino
target	Menciona onde o link deve abrir.	<ul style="list-style-type: none"><li>_self (abre na mesma janela)</li><li>_blank (abre em uma nova guia ou janela)</li><li>_top (abre no modo de tela cheia na parte superior da janela)</li><li>_parent (abre o link no quadro pai)</li></ul>

Quadro 4 - Atributos

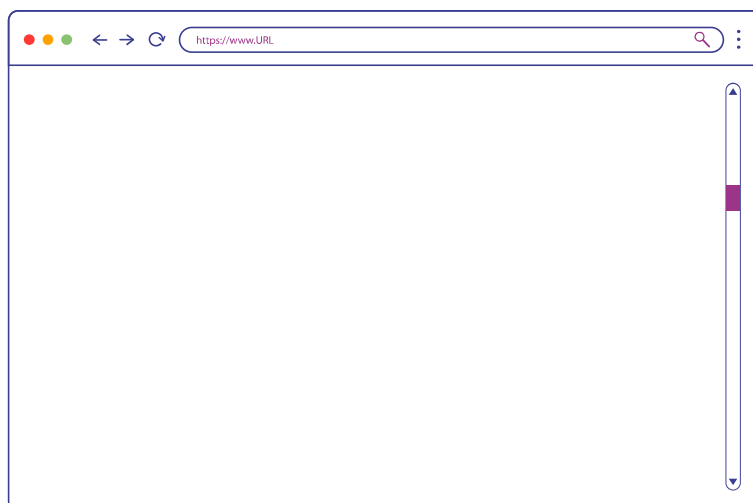
Fonte: do Autor (2022)

Confira um exemplo:



```
<a href="https://www.site.com.br" target="_blank">Acessar</a>
```

A maneira como você estrutura seu URL e links é absolutamente essencial para a função do seu site. Isso é importante para garantir que seu site possa ser rastreado e indexado adequadamente. Você tem duas opções básicas quando se trata de seleção de URLs: URL absoluto e URL relativo. Se você escolher o tipo errado, isso não apenas dificultará o rastreamento do site pelos mecanismos de pesquisa, mas também poderá apresentar links de acesso inválidos.





Um URL absoluto contém o endereço completo do protocolo (HTTP – *Hypertext Transfer Protocol*) ao nome de domínio (www.site.com.br) e inclui o local em seu site no sistema de pastas (/pasta1 ou /pasta2, por exemplo) dentro do URL. Basicamente, é o URL completo da página que você vincula. Um exemplo de um URL absoluto é:

```
<a href="http://www.site.com.br/pasta1/contato.html">Contato</a>
```

O URL relativo, por outro lado, não usa o endereço web completo e contém apenas a localização após o domínio. Ele assume que o link que você adiciona está no mesmo site e faz parte do mesmo domínio raiz. O caminho relativo começa com a barra e leva o navegador a permanecer no site atual. Um exemplo de um URL relativo é:

```
<a href="/contato.html">Contato</a>
```

Perceba que links absolutos requerem o endereço completo, o que pode fazer com que links se quebrem com qualquer alteração. Links podem ser editados, mas isso não pode ser feito facilmente dependendo do tamanho do site. Além disso, links absolutos não levam em consideração o local do arquivo.

## Tags semânticas

Existem bilhões de sites na internet. Muitos deles são feitos há muito tempo e não foram atualizados com recursos HTML. Como tal, os mecanismos de pesquisa passam por esses sites e os usuários identificam esses sites como difíceis de ler. A maioria desses sites utilizam em sua estruturação a **<div>**.

A tag de divisão HTML, chamada de **<div>**, é um elemento especial que permite agrupar conjuntos semelhantes de conteúdo em uma página da web. Você pode usá-la como um contêiner genérico para associar conteúdo semelhante.

A aplicação de uma tag `<div>` permite identificar a seção definida pela tag, para que você possa aplicar formatação ou script ao seu conteúdo. Por exemplo, você pode aplicar estilo CSS ou script do lado do cliente, como JavaScript, ao conteúdo de uma tag `<div>`. Observe que você pode colocar qualquer conteúdo que desejar em uma tag `<div>` em HTML. A tag `<div>` é simplesmente um contêiner HTML. Seu único propósito é permitir que você agrupe outros elementos do documento aos quais aplicar o mesmo estilo CSS ou efeito de script. Você também pode usar estilos CSS para alinhar e posicionar a tag `<div>` na página, se necessário. Entretanto, sua utilização pode trazer problemas de acessibilidade, legibilidade, consistência e padrões (WHATWG, 2022).



Uma das características mais importantes do HTML5 é sua semântica. O HTML semântico refere-se à sintaxe que torna o HTML mais compreensível, definindo melhor as diferentes seções e o layout das páginas da web. Torna as páginas da web mais informativas e adaptáveis, permitindo que navegadores e mecanismos de pesquisas interpretem melhor o conteúdo. Por exemplo, em vez de usar `<div id="header">` você pode usar uma tag `<header>` (WHATWG, 2022).

Um dos principais avanços do HTML5 foi a introdução de um conjunto padronizado de elementos semânticos. O termo semântico refere-se ao significado de uma palavra ou item, então elementos semânticos são elementos projetados para marcar a estrutura de um documento de uma forma mais significativa, de uma forma que deixe claro para que servem (WHATWG, 2022).

Existe um padrão super comum que pode ser encontrado em sites, tutoriais e até bibliotecas CSS por toda a internet, e por boas razões. Frequentemente, dividimos uma página em seu nível superior em três regiões: header, main e footer e, em seguida, dividimos essas regiões em seções conforme necessário (WHATWG, 2022).



Os elementos não semânticos não têm nenhum significado especial, as relações hierárquicas entre eles são meramente ilusórias. Os exemplos mais usados de tags HTML não semânticas são as tags `"<div></div>"` e as tags `"<span></span>"`.

Vamos primeiro considerar um modelo de página HTML básico, escrito em HTML não semântico:

```
<html>

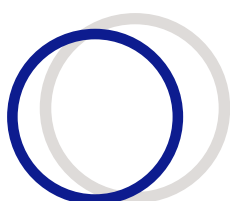
<head>
  <title>Exemplo</title>
</head>

<body>
  <div id="header">
    Aqui são inseridos logo, itens de navegação etc.
  </div>
  <div id="main">
    Aqui é um lugar no website onde vem o conteúdo principal.
  </div>
  <div id="footer">
    Aqui são inseridas informações relacionadas ao rodapé,
    como mensagens e links.
  </div>
</body>

</html>
```

Esse padrão foi e ainda é utilizado por décadas, e faz muito sentido estruturar um documento dessa maneira, tanto para legibilidade do HTML quanto para facilitar o estilo da página em CSS.

Agora considere o exemplo de HTML semântico mostrado a seguir:



```
<html>

<head>
  <title>Exemplo</title>
</head>

<body>
  <header>
    Aqui são inseridos logo, itens de navegação etc.
  </header>
  <main>
    Aqui é um lugar no website onde vem o conteúdo principal.
  </main>
  <footer>
    Aqui são inseridas informações relacionadas ao rodapé,
    como mensagens e links.
  </footer>
</body>

</html>
```

A principal diferença é que substituímos as tags `<div>` por três novas tags: `<header>`, `<main>` e `<footer>`. Estas tags são semânticas porque são usadas para representar diferentes seções em uma página HTML. Elas são mais descritivas do que as tags `<div>` que dificultam o particionamento de páginas da web em seções.

Os elementos `<header>` e `<footer>` são definidos de maneira muito semelhante na especificação e seguem o mesmo conjunto de regras sobre onde podem ser usados, com a única diferença sendo seus propósitos semânticos: os cabeçalhos vão no início e os rodapés vão nas extremidades. O terceiro elemento da região primária, `<main>`, é especial. Assim, o `<main>` é onde você coloca os itens principais, as partes importantes de uma página (WHATWG, 2022).

Assim, o HTML5 oferece novos elementos semânticos que definem as diferentes partes de uma página web, confira a imagem a seguir.

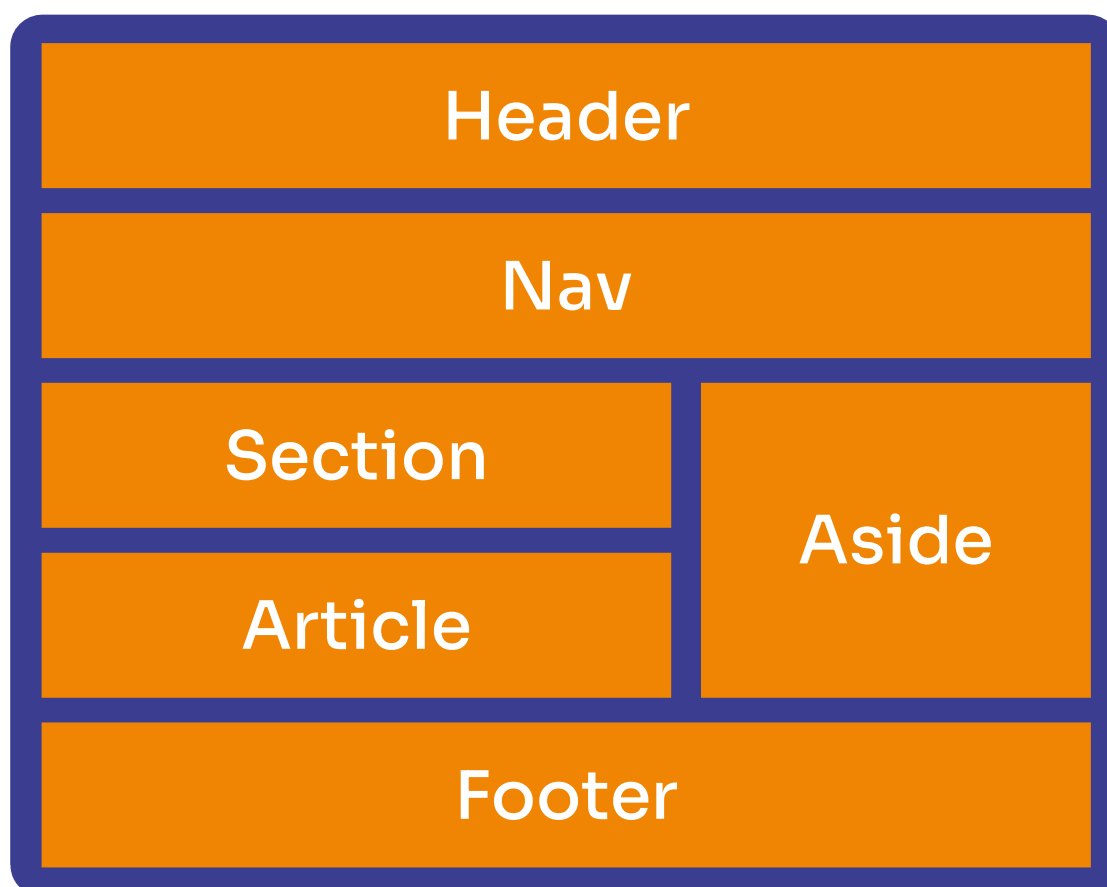


Figura 7 - Elementos semânticos do HTML5

Fonte: do Autor (2022)

Elementos como `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>` e `<footer>` agem mais ou menos como elementos `<div>`. Eles agrupam outros elementos em seções de página (WHATWG, 2022).

Usar esses elementos não é tão complicado quanto parece à primeira vista, e a maioria é bastante autoexplicativa. Faremos uma rápida passagem por cada novo elemento.

## **`<header>`**

A tag de cabeçalho é um dos elementos da seção, sua função é agrupar outros elementos HTML de acordo com sua função na página. O elemento `<header>` contém todo o conteúdo introdutório da página, normalmente um título e um slogan ou elementos de navegação. Contém informações das marcas, como o logotipo. Confira o trecho de código a seguir:

```
<header>
  <p>Slogan e logotipo da empresa...</p>
  <nav>
    <ul>
      <li>Início</li>
      <li>Sobre</li>
    </ul>
  </nav>
</header>
```

## <nav>

Contém a seção de navegação do site. Além disso, apresenta links para outras páginas ou para determinadas seções dentro da página. Acompanhe um exemplo:

```
<nav>
  <ul>
    <li><a href="#">Início</a></li>
    <li><a href="#">Contato</a></li>
    <li><a href="#">Sobre</a></li>
  </ul>
</nav>
```

## <section>

A tag “section” é usada para agrupar elementos. Na maioria das vezes, deve incluir uma tag de cabeçalho na parte superior. Por exemplo, uma página inicial pode ter uma seção para apresentar a empresa, outra para notícias e outra para informações de contato. Na sequência, é apresentado um exemplo:

```
<section>
  <h1>Título 1</h1>
  <p>Conteúdo 1</p>
</section>
<section>
  <h1>Título 2</h1>
  <p>Conteúdo 2</p>
</section>
```

## <article>

Representa um conteúdo independente que pode ser usado e distribuído separadamente do restante da página ou site. Pode ser uma postagem no fórum, um artigo de jornal ou revista, uma entrada de blog, um comentário ou qualquer outra parte independente do conteúdo. Acompanhe o exemplo a seguir:

```
<article>
  <h1>Título</h1>
  <p>Conteúdo...</p>
</article>
```

## <aside>

Contém informações extras e conteúdo ou links relacionados. Pode ser considerado separado, mas indiretamente relacionado ao conteúdo principal. Isso geralmente aparece como barras laterais. Quando uma tag <aside> é usada dentro de uma tag <article>, o conteúdo do <aside> deve ser especificamente relacionado a esse artigo. Conheça um exemplo:

```
<aside>
  <h1>Título 1</h1>
  <p>Conteúdo 1</p>
</aside>
<aside>
  <h1>Título 2</h1>
  <p>Conteúdo 2</p>
</aside>
```

## <footer>

Deve organizar o conteúdo final da página, como os créditos ou informações de contato. Informações sobre direitos autorais, política de privacidade, termos de uso também são encontradas na tag de rodapé. Acompanhe o trecho de código apresentado a seguir:

```
<footer>
  <p>© 2022 Todos os Direitos Reservados.</p>
</footer>
```

Esses exemplos apresentados não são de forma alguma os únicos elementos semânticos em HTML5. Existem muitos elementos adicionais que ajudam a marcar e estruturar seu conteúdo de texto, mídia incorporada etc.





Durante seus estudos, você pôde perceber que o HTML5 é um tópico amplo que evoluiu muito ao longo dos anos. Assim, o HTML5 foi desenvolvido para conceder aos desenvolvedores um meio de construir aplicativos mais flexíveis, poderosos e eficientes. Ele supera os problemas relacionados ao HTML 4.0 e traz muitas novas tags e aprimoramentos para uma ampla gama de recursos. Durante seus estudos, você aprendeu sobre os conceitos básicos do HTML, teve a oportunidade de conhecer os principais recursos do HTML5, viu a definição de atributos e tags e, por fim, conheceu as tags estruturais, textuais e semânticas.

Continue seu caminho de aprendizado, pois ainda serão abordados mais assuntos complementares na sequência! Fique ligado!

## REFERÊNCIAS

WHATWG. HTML Living Standard. Disponível em: <https://html.spec.whatwg.org>. Acesso em: 30 jun. 2022.

