

IMT - Instituto Mauá de Tecnologia

Engenharia de Computação

ECM252 – Linguagens de Programação II

Professor Rodrigo Bossini

Aula 01 - Introdução à Linguagem de Marcação HTML

1 Introdução

As páginas que visitamos ao navegar pela Internet possuem conteúdo estruturado em seções, cada qual com uma espécie de significado. A linguagem HTML disponibiliza elementos que viabilizam a especificação de páginas assim. Neste guia, iremos aprender as características básicas da linguagem HTML. Note que HTML **não é uma linguagem de programação**, pois não possui estruturas de seleção e repetição, por exemplo. Trata-se de uma linguagem de marcação.

Dê uma olha no código fonte das seguintes páginas. Para isso, usando o Google Chrome, você pode apertar CTRL + U ou clicar com o direito na página e escolher

www.google.com.br
www.cplusplus.com

Quando o navegador lida com uma página assim, ele se encarrega de construir uma **estrutura de dados** do tipo **árvore**, que viabiliza operações eficientes e faz parte do que chamamos de **renderização** da página.

Neste material iremos desenvolver uma página HTML simples com o objetivo de ilustrar os principais componentes disponíveis na linguagem.

NOTA: Sempre que quiser saber mais sobre HTML, visite os links a seguir.

w3.org
whatwg.org
developer.mozilla.org

2 Desenvolvimento

2.1 Comece criando um diretório que servirá para abrigar seus arquivos. Seu nome pode ser algo como “**desenvolvimento**”.

2.2 Iremos utilizar o IDE Visual Studio Code para desenvolver nossas páginas. Para colocar uma instância dele em execução, abra um terminal e navegue até o diretório criado no passado anterior. A seguir, use o comando

code .

que fará com que o VS Code fique vinculado ao diretório local, representado pelo símbolo “.”.

2.3 No VS Code, crie um arquivo chamado **minha_pagina.html**. Note que uma página é descrita por um arquivo textual simples. É responsabilidade do navegador renderizá-la: exibir o conteúdo da forma como seu conteúdo descreve.

2.4 Documentos HTML são constituídos por **elementos**, os quais possuem, em geral, uma **tag de abertura**, um **conteúdo**, uma **tag de fechamento**. Veja o exemplo da Listagem 2.4.1.

Listagem 2.4.1

```
<p> Um exemplo de paragrafo </p>
```

2.5 Para começar a escrever nossos documentos HTML, o primeiro passo é especificar um elemento html, como na Listagem 2.5.1.

Listagem 2.5.1

```
<html>  
  
</html>
```

Cada elemento HTML tem um **significado** ou **semântica**. Este que acabamos de utilizar, tem como significado o início e o término de um documento HTML. Abra esse arquivo no navegador e veja nenhum conteúdo será exibido.

2.6 A fim de especificar conteúdo visível, precisamos definir o **corpo** do documento, o que é feito com o elemento **body**. Por si só ele também não é visível. Mas ele permite que conteúdo visível seja especificado. Veja a Listagem 2.6.1. Note que a **indentação** não é obrigatória porém é fortemente recomendável, pois promove a legibilidade do código.

Listagem 2.6.1

```
<html>
  <body>

  </body>
</html>
```

Dizemos que **body** é **filha** de **html**, pois aparece aninhada a ela. Agora já podemos adicionar conteúdo que será visível no navegador, como na Listagem 2.6.2.

Listagem 2.6.2

```
<html>
  <body>
    Minha página HTML.
  </body>
</html>
```

2.7 Ocorre, porém, que documentos HTML são bem estruturados por meio de elementos apropriados. Uma página HTML possui diversas seções, cada qual com um significado, e este significado é tornado claro por meio do uso apropriado dos elementos que a linguagem disponibiliza. Assim, o texto exibido seria mais apropriadamente colocado como conteúdo de um elemento apropriado para texto, como um parágrafo. Veja a Listagem 2.7.1.

Listagem 2.7.1

```
<html>
  <body>
    <p>Minha página HTML.</p>
  </body>
</html>
```

Note que a exibição da página permanece a mesma. Porém, para quem lê o código fica mais clara a estrutura do documento. Além disso, o uso de elementos HTML apropriados torna mais interessante o uso de dispositivos voltados para a **acessibilidade**, como leitores de conteúdo utilizados por pessoas com deficiência visual.

2.8 Além do elemento **body**, uma página HTML pode ter um elemento **head**. Neste elemento especificamos dados não visíveis pelo usuário final. Tratam-se de dados que indicam características da página, como ela deve funcionar, referências para outras páginas e arquivos entre muitas outras coisas. Os dados contidos na seção **head**, em geral, são chamados de **metadados**. Veremos muitos exemplos ao longo desse material. O título da página é algo que pode ser definido no elemento **head**. Veja a Listagem 2.8.1.

Listagem 2.8.1

```
<html>
  <head>
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <p>Minha página HTML.</p>
  </body>
</html>
```

Clique com o direito na página e escolha a opção para visualizar o código fonte da página.

2.9 Podemos associar atributos a elementos HTML. Eles são definidos na tag de abertura do elemento de interesse. Em geral, eles servem para personalizar o funcionamento do elemento. Na Listagem 2.9.1, definimos o idioma da página por meio de um atributo (chamado **lang**) aplicado ao elemento HTML.

Listagem 2.9.1

```
<html lang="pt-BR">
  <head>
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <p>Minha página HTML.</p>
  </body>
</html>
```

A configuração do idioma não altera o conteúdo da página (não tem nada a ver com traduzir a página, por exemplo), mas é algo muito útil para leitores aplicados à acessibilidade e para **motores de busca**, como o Google, embora, em geral, ele seja capaz de detectar por si só o idioma de uma página sob análise.

2.10 A Linguagem HTML possui diferentes versões. A última disponível se chama **HTML5** (<https://developer.mozilla.org/en-US/docs/Glossary/HTML5>). Ao longo dos anos a linguagem evoluiu muito e, ao criar páginas HTML, o desenvolvedor precisa especificar qual versão está utilizando para que o navegador saiba como lidar com ela adequadamente. Para especificar que estamos utilizando HTML5, usamos a construção exibida pela Listagem 2.10.1.

Listagem 2.10.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <p>Minha página HTML.</p>
  </body>
</html>
```

2.11 Todo conteúdo digital é representado por uma sequência de bits (zeros e uns), inclusive nossas páginas HTML. Um **encoding** é um mapeamento de um conjunto de caracteres para o conjunto de sequências de bits que representam cada um deles. Um dos mais comuns e utilizados hoje é o **UTF-8**. Quando um arquivo é salvo (qualquer arquivo, HTML, txt ou qualquer outro), o editor de texto utilizado faz uso de um encoding para armazenar o conteúdo digitado. Assim, o conteúdo armazenado será a sequência de bits que representa o conteúdo digitado, de acordo com o encoding. Para entender melhor, considere os mapeamentos das tabelas 2.11.1 e 2.11.2. São mapeamentos fictícios, apenas para exemplificar. Ali, mostramos o código binário para três possíveis caracteres.

Tabela 2.11.1 – Encoding 1

Carectere	Codificação
a	111
b	100
c	101

Tabela 2.11.1 – Encoding 2

Carectere	Codificação
a	001
b	110
c	011

Agora, considere um arquivo textual cujo conteúdo é dado na Listagem 2.11.1.

Listagem 2.11.1

aabac

A forma como ele é armazenado, segundo cada um dos encodings de exemplo é exibida nas listagens 2.11.2 e 2.11.3.

Listagem 2.11.2 – Representação com Encoding 1

111111100111101

Listagem 2.11.3 – Representação com Encoding 2

001001110001011

Para saber mais sobre encodings e, em particular, sobre o UTF-8, veja, por exemplo, o Link 2.11.1.

Link 2.11.1

<https://en.wikipedia.org/wiki/UTF-8>

Nosso objetivo agora é testar o seguinte. O VS Code usou um Encoding para salvar o arquivo. O que ocorre caso diferentes encodings sejam utilizados para exibir o conteúdo no Google Chrome.

O Google Chrome, desde sua versão 55, removeu a opção de escolha manual do encoding a ser utilizado, segundo a documentação oficial por conta de a opção ser utilizada por pouquíssimos usuários. Segundo sua documentação, ele é capaz de detectar com sucesso o encoding correto a ser usado para tratar o conteúdo das páginas. Assim, para testar o conteúdo deste passo, precisamos instalar uma extensão.

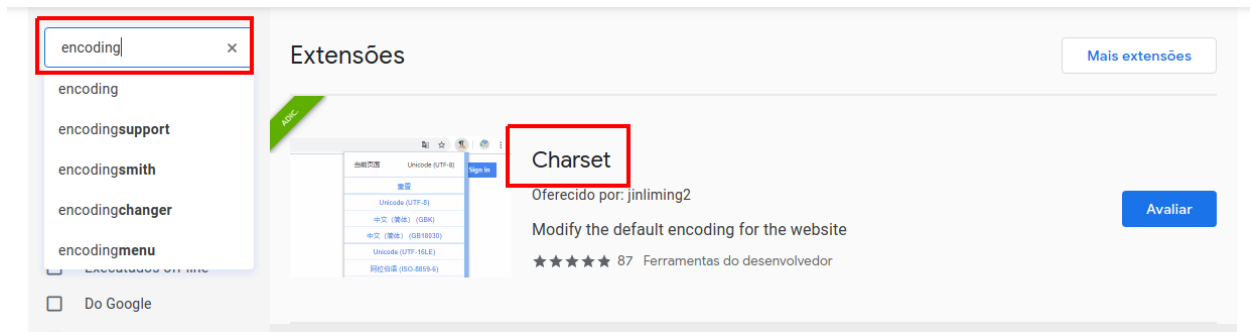
Comece visitando o Link 2.11.2.

Link 2.11.2

<https://chrome.google.com/webstore/category/extensions>

A seguir, faça a busca por “encoding” e instale o plugin chamado “Charset”. Veja a Figura 2.11.1.

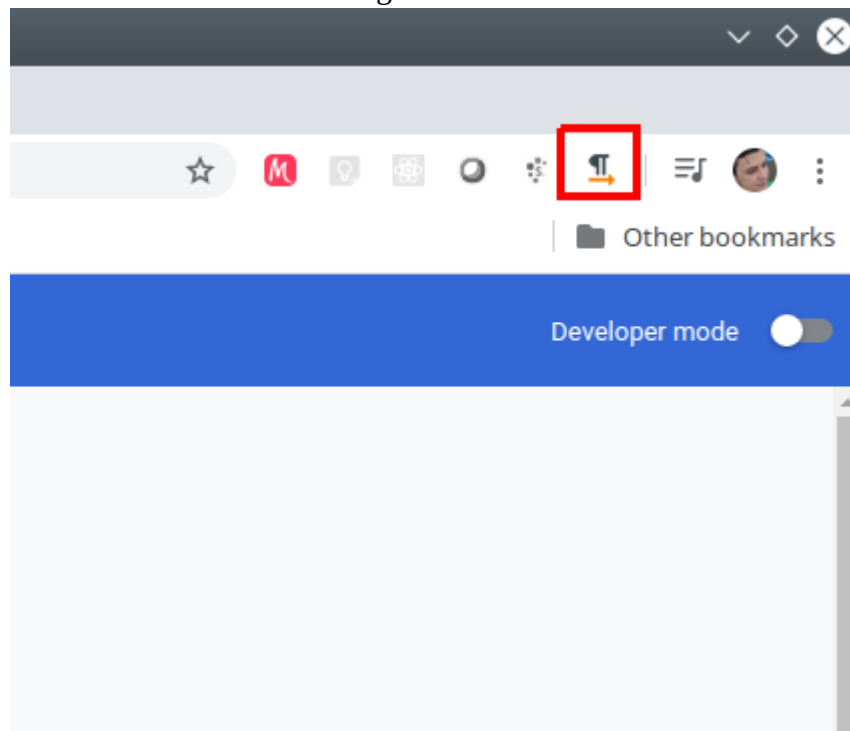
Figura 2.11.1



Feita a instalação, podemos usar o plugin para alterar o encoding a ser utilizado pelo navegador para interpretar nossa página. Antes disso, porém, é necessário permitir que o plugin acesse arquivos em nosso sistema de arquivos. Para isso, clique no **símbolo de três bolinhas no canto superior direito do Chrome** e então **More Tools >> Extensions**. Na extensão Charset, clique em **Details** e então escolha a opção **Allow access to File URLs**.

Depois disso, no Chrome, no canto superior direito, haverá um ícone da extensão, como na Figura 2.11.2.

Figura 2.11.2



Antes de clicar na extensão, vá até a aba em que o seu arquivo .HTML está aberto. Clique no ícone da extensão e escolha um encoding qualquer, como **Arabic (ISO-8859-6)**. Repare que os caracteres com acento ficam com exibição incorreta. Para voltar ao normal, escolha **UTF-8**. Porque **UTF-8**? Pois, por padrão, o VS Code salvou nosso arquivo com essa condificação. Dá pra ver isso na parte inferior direita dele, como na Figura 2.11.3.

Figura 2.11.3



Clique sobre UTF-8 e veja que dá para reabrir o arquivo com outros tipos de encodings. Faça alguns testes para ver os resultados.

A fim de auxiliar o navegador a identificar o encoding utilizado para salvar seu arquivo, podemos especificar um meta dado, como mostra a Listagem 2.11.1.

Listagem 2.11.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <p>Minha página HTML.</p>
  </body>
</html>
```

2.12 Como mencionado, uma página HTML pode ser estruturada utilizando muitos elementos que tem como principal finalidade deixar clara a forma como a página está organizada, destacar suas seções etc. Veja o que ocorre com a página quando escrevemos um texto sem utilizar elementos, diretamente em seu corpo. A Listagem 2.12.1 ilustra isso.

DICA: No VS Code, use o atalho **lorem** para gerar o texto usado.

Listagem 2.12.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
    unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum adipisci
    odio itaque obcaecati quis.
  </body>
</html>
```

O navegador exibe todo o texto de uma vez só, sem nenhuma organização ou estrutura.

Note também que várias linhas e espaços em branco são ignorados.

2.13 Suponha que agora desejamos montar uma página que tem um título e um parágrafo. Podemos usar o elemento **h1** para solicitar ao navegador que se preocupe com esse destaque, como na Listagem 2.13.1.

Listagem 2.13.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <h1>Vejamoss como fazer um título</h1>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
    unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
    adipisci odio itaque obcaecati quis.
  </body>
</html>
```

Nota: A Linguagem HTML possui também os elementos **h2**, **h3**, **h4**, **h5** e **h6**. Substitua o elemento **h1** no exemplo anterior por cada um deles e veja o resultado.

2.14 Suponha que nossa página possui diferentes parágrafos e que desejamos pular uma linha entre cada um deles. Há também um elemento apropriado para isso, chamado **p**. Ajuste o exemplo como na Listagem 2.14.1.

Listagem 2.14.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <h1>Vejamos como fazer um título</h1>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum
      veldistinctio unde mollitia laudantium ut sint architecto deserunt reprehenderit
      eum molestias earum adipisci odio itaque obcaecati quis.
    </p>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel
      distinctio unde mollitia laudantium ut sint architecto deserunt reprehenderit eum
      molestias earum adipisci odio itaque obcaecati quis.
    </p>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel
      distinctio unde mollitia laudantium ut sint architecto deserunt reprehenderit eum
      molestias earum adipisci odio itaque obcaecati quis.
    </p>
  </body>
</html>
```

2.15 No exemplo da Listagem 2.15.1 adicionamos um subtítulo à página, usando um elemento **h2**.

Listagem 2.15.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <h1>Vejamos como fazer um título</h1>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci ovdio itaque obcaecati quis.
    </p>
    <h2>Subtítulo da página</h2>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci odio itaque obcaecati quis.
    </p>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci odio itaque obcaecati quis.
    </p>
  </body>
</html>
```

NOTA: Perceba que cada tag tem sua forma de exibição pré determinada pelo navegador. É importante observar que essa forma pode variar em navegadores diferentes. Até então, o conceito que mais nos preocupa é a **semântica do documento HTML**. Ou seja, o uso das tags por si só. Sua aparência não é o mais importante e ela pode variar. Inclusive, podemos (e iremos) personalizar a sua forma de exibição usando CSS em aulas futuras. Resumindo, o uso dos elementos HTML apropriados é importante, principalmente para quem lê o código, seja um ser humano, seja um assistente de hardware. A aparência pode ser personalizada de acordo com diferentes critérios, de acordo com algum tema escolhido etc.

2.16 Páginas HTML também podem conter comentários, ou seja, trechos de texto que não serão renderizados pelo navegador. Veja a Listagem 2.16.1.

Listagem 2.16.1

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>
      Testando a linguagem HTML.
    </title>
  </head>
  <body>
    <!-- A seguir temos o título da página-->
    <h1>Vejamos como fazer um título</h1>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci odio itaque obcaecati quis.
    </p>
    <h2>Subtítulo da página</h2>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci odio itaque obcaecati quis.
    </p>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Ea inventore ad, illum vel distinctio
      unde mollitia laudantium ut sint architecto deserunt reprehenderit eum molestias earum
      adipisci odio itaque obcaecati quis.
    </p>
  </body>
</html>
```

2.17 Podemos enfatizar algum trecho de texto de nossa página. Note que elementos enfatizados tendem a ser renderizados de forma diferente pelo navegador. Além disso, navegadores diferentes podem fazer sua renderização de forma diferente. Mas novamente, o mais importante é a semântica associada ao uso do elemento. Veja a Listagem 2.17.1. Adicione o conteúdo ao body da página desenvolvida até então.

Listagem 2.17.1

```
<p>
  <em>Estudando fundamentos</em>
</p>
```

2.18 Quando queremos destacar algo importante em uma frase, isso pode ser feito com o elemento **strong**, como na Listagem 2.18.1.

Listagem 2.18.1

```
<p>  
  A linguagem HTML é uma linguagem <strong>de marcação</strong>.  
</p>
```

2.19 O elemento **mark** permite que seja feita a marcação de trechos de texto de interesse. Veja a Listagem 2.19.1.

Listagem 2.19.1

```
<p>  
  Javascript roda no lado <mark>cliente</mark> ou <mark>servidor</mark>?  
</p>
```

2.20 Há também a possibilidade de inserir imagens em documentos HTML. A tag **img** é apropriada para isso, como mostra a Listagem 2.20.1. Note que especificamos atributos para essa tag. Um deles indica a localização da figura a ser exibida. Coloque a figura no mesmo diretório em que está armazenado o seu arquivo .html. O outro contém um texto alternativo que o navegador deve exibir caso a imagem não esteja disponível.

Listagem 2.20.1

```

```

Também é possível usar imagens hospedadas em servidores remotos, como mostra a Listagem 2.20.2.

Listagem 2.20.2

```

```

É possível organizar as figuras em uma pasta ou pastas diferentes da principal. Para tal, basta referenciá-las adequadamente no código, como na Listagem 2.20.3. Para esse exemplo, crie uma pasta chamada img e coloque a figura lá.

Listagem 2.20.3

```

```

Os atributos **width** e **height** permitem especificar largura e altura de uma imagem. Note, porém, que pode não ser uma boa ideia usar ambos simultaneamente. Veja os exemplos das listagens 2.20.4 e 2.20.5.

Listagem 2.20.4

```

```

Listagem 2.20.5

```

```

2.21 O elemento **âncora** permite definir um link para uma outra página ou recurso na Internet. Ele é definido usando a tag **a**. Veja a Listagem 2.21.1.

Listagem 2.21

```
<p>  
A linguagem <a href="https://html.spec.whatwg.org/" target="_blank">HTML</a> é uma linguagem  
<strong>de marcação</strong>.  
</p>
```

O atributo target configurado como “blank_” indica que o link deve ser aberto em uma nova aba.

Imagens também podem ter links associados a elas, como na listagem 2.21.2.

Listagem 2.21.2

```
<a href="http://www.google.com"></a>
```

Para navegar para uma página local, basta especificar seu nome (precedido pela pasta em que ela estiver, caso não seja a mesma da atual). Veja a Listagem 2.21.3.

Listagem 2.21.3

```
<p>Entre em <a href="contato.html">contato</a> conosco.</p>
```

A Listagem 2.21.4 mostra o conteúdo do arquivo contato.html. Note o prefixo **mailto** precedendo o endereço de email. Ele instrui o navegador a enviar uma mensagem ao sistema operacional dizendo que o programa de email padrão deve ser aberto já contendo aquele email como destinatário.

Listagem 2.21.4

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<title>Contato</title>
</head>
<body>
<h1>Contato</h1>
<p><a href="mailto:email@email.com">Entre em contato conosco</a></p>
</body>
</html>
```

2.22 A ideia de **elementos estruturais** consiste em especificar **seções** de nossa página. Em geral, isso é muito importante quando a página é relativamente grande. Exemplos de seções são **cabeçalho**, **rodapé**, **menu lateral** etc.

Digamos que queremos adicionar um logo para a página. E desejamos fazer isso especificando uma seção na página para ele. Isso poderia ser feito com um elemento HTML chamado **div** (de division). Veja a Listagem 2.22.1. Coloque esse conteúdo como **primeiro filho de body**.

Listagem 2.22.1

```
<div>
  <p>
    Nossas impressões sobre HTML.
  </p>
</div>
```

Note que o elemento **div** não tem uma semântica clara. Quem lê o código (em particular motores de busca e dispositivos de apoio acessível) não necessariamente entende que se trata de uma seção apropriada para a logomarca.

A fim de tornar a página mais legível, uma prática é adicionar um **id** ou até uma **class** ao elemento, como na Listagem 2.22.2.

Listagem 2.22.2

```
<div id="cabeçalho">
<p>
Nossas impressões sobre HTML.
</p>
</div>
```

Isso ajuda em partes, pois motores de busca e dispositivos de leitura não necessariamente interpretarão os nomes escolhidos pelos desenvolvedores como desejado.

Depois do cabeçalho, vamos adicionar uma nova div, cujo id será “principal”. Ela engloba todo o conteúdo restante. Veja a Listagem 2.22.3.

Listagem 2.22.3

```
<div id="principal">
<!-- A seguir temos o título da página-->
<!-- Coloque todo o conteúdo da página aqui-->

</div>
```

Note que a seção principal pode ter seu próprio cabeçalho, que pode ser definido usando uma outra div. Veja a Listagem 2.22.4.

Listagem 2.22.4

```
<div id="cabecalho-principal">
<!-- A seguir temos o título da página-->
<h1>Vejam como fazer um título</h1>
<a href="http://www.google.com"></a>
</div>
```

Por fim, a página poderia ter um rodapé, contendo o link de contato, como na Listagem 2.22.5.

Listagem 2.22.5

```
<div id="rodape">
<p>Entre em <a href="contato.html">contato</a> conosco.</p>
</div>
```

2.23 A fim de promover a legibilidade das páginas HTML, em particular simplificando a operação de motores de busca e leitores acessíveis, a linguagem HTML5 introduziu elementos semânticos, que ilustraremos agora. A div com id cabeçalho, por exemplo, pode ser substituída por um elemento **header**, como na Listagem 2.23.1.

Listagem 2.23.1

```
<header>
<p>
Nossas impressões sobre HTML.
</p>
</header>
```

A ideia é que o significado do elemento já é previamente definido e qualquer dispositivo irá interpretá-lo da mesma forma. Veja, por exemplo, o Link 2.23.1.

Link 2.23.1

<https://html.spec.whatwg.org/#the-header-element>

2.24 A seção principal do documento pode ser substituída por um **article** ou uma **section**. Veja a Listagem 2.24.1.

Listagem 2.24.1

```
<article>
<!-- A seguir temos o título da página-->
<!-- Coloque todo o conteúdo da página aqui-->

</article>
```

2.25 Também faz sentido alterar o cabeçalho da seção principal para **header** também. Veja a Listagem 2.25.1.

Listagem 2.25.1

```
<header>
<!-- A seguir temos o título da página-->
<h1>Vejamos como fazer um título</h1>
<a href="http://www.google.com"></a>
</header>
```

2.26 Caso a página tenha algum conteúdo relacionado com o principal, podemos colocá-lo em um elemento **aside**. Depois ele pode ser posicionado ao lado do conteúdo principal com CSS, por exemplo. Veja a Listagem 2.26.1. Coloque esse conteúdo antes do rodapé.

Listagem 2.26.1

```
<aside>
<p>
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Libero dolores placeat necessitatibus, aliquid
nulla odit quia, tenetur laudantium, adipisci voluptatum deleniti. Nemo odio rerum sed iusto corrupti
enim. Inventore, repudiandae!
</p>
</aside>
```

2.27 Há também um elemento apropriado para rodapés, chamado **footer**. Seu uso é ilustrado na Listagem 2.27.1. Ele deve substituir a div usada para rodapé feita anteriormente.

Listagem 2.27.1

```
<footer>  
<p>Entre em <a href="contato.html">contato</a> conosco.</p>  
</footer>
```

Referências

- Web Hypertext Application Technology Working Group (WHATWG). 2020. Disponível em <<https://whatwg.org/>>. Acesso em março de 2020.
- Web Hypertext Application Technology Working Group (WHATWG). 2020. Disponível em <<https://developer.mozilla.org/en-US/>>. Acesso em março de 2020.