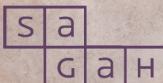


FERRAMENTAS DE DESENVOLVIMENTO WEB

Vanessa Stangerlin Machado Paixão



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Aplicação de *layout* HTML + CSS

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Demonstrar a utilização de *tags* HTML para a estruturação do conteúdo.
- Aplicar o CSS para controlar a formatação e o posicionamento de elementos HTML.
- Elaborar páginas e aplicativos para a internet utilizando HTML e CSS.

Introdução

Com o avanço da internet, cresceu muito o desenvolvimento de aplicações Web no decorrer dos últimos anos. A banda larga e os diferentes dispositivos computacionais popularizaram o acesso a conteúdos na World Wide Web, o que trouxe às empresas a necessidade de serem vistas pelos usuários por meio de *sites* e aplicativos.

Visando a essa ascensão, o HTML5, versão atual do HTML, teve como objetivo o desenvolvimento de conteúdos ricos sem a necessidade de *plug-ins* adicionais, criando um modelo que pode ser interpretado em diversas plataformas e sistemas diferentes. Essa versão foi recomendada em 2014 pela W3C (World Wide Web Consortium), organização mundial para padronização da World Wide Web. Os documentos HTML são compostos por *tags*, atributos e elementos, e suas respectivas formatações ficam sob a responsabilidade da linguagem de formatação CSS (folha de estilo em cascata).

Neste capítulo, você vai estudar a estrutura das páginas Web e ver como gerar *sites* padronizados e organizados. Também estudará a sintaxe e a semântica da linguagem de marcação HTML5 e da linguagem de formatação CSS3.

Utilizando tags HTML para estruturar conteúdo

HTML é a sigla de *hypertext markup language* (linguagem de marcação de hipertexto). É a linguagem utilizada para descrever a estrutura de páginas Web, que usa marcação para a publicação de conteúdo como texto, imagem, vídeo, áudio etc.

As tags funcionam dividindo a página em seções, que, em sua maioria, começam com uma tag de abertura <elemento> e terminam com uma tag de fechamento </elemento> (SANDERS, 2012). Os documentos em HTML5 começam sempre com <!DOCTYPE HTML>, que indica ao *browser web* que aquele é, de fato, um documento HTML5 (PRESCOTT, 2015). Veja um exemplo da tag de parágrafo, que faz a inserção de um parágrafo na página.

<p> Exemplo do uso da tag de parágrafo em HTML. </p>

Uso de elementos para a inserção de figuras

O elemento figure, <figure>, novo no HTML5, delimita um bloco de conteúdo independente que, normalmente, contém imagem, gráfico, diagrama ou ilustração que faz referência ao conteúdo principal do documento (FLATSCART, 2011). Além disso, o elemento figcaption, <figcaption>, permite a criação de uma legenda para o elemento figure.

Um exemplo do emprego do elemento figure é com a inserção de imagens, como uma foto ou gráfico, por meio do uso da tag . Essa tag não tem fechamento, apenas os atributos, que são:

- **src**: indica a url da imagem, ou seja, o caminho onde ela se encontra.
- **alt**: indica o texto alternativo a ser mostrado caso a imagem não carregue na página ou caso o usuário esteja usando um *software* de acessibilidade, como os leitores de tela (utilizados por pessoas com deficiência visual).

Visualize o exemplo de uso desses atributos na Figura 1

(a)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="UTF-8"/>
5  <title>Título da página</title>
6  </head>
7  <body>
8  <!-- titulo-->
9  <h1>Imagens HTML</h1>
10
11 <!-- Imagem com legenda -->
12 <figure>
13   
14   <figcaption>Legenda da imagem</figcaption>
15 </figure>
16
17 </body>
18 </html>
```

(b)

Imagens HTML



Figura 1. (a) Código HTML com um exemplo de inserção de imagem utilizando as tags `<figure>`, `` e `<figcaption>`, da linha 12 à 16. (b) Conteúdo do código apresentado no navegador Web.

Links

Link em inglês significa ligação, elo, conexão. Em documentos Web, o *link* é um endereço, ou seja, uma referência de um local para outro dentro de um documento HTML. Para definir um *link*, basta utilizar a tag `<a>`, que tem o atributo obrigatório `href`, responsável pela definição do endereço de destino. Existem três categorias de *links*: internos, externos e âncora.

Links internos

Os *links* internos direcionam o usuário para arquivos presentes dentro do mesmo *site*. Para tal, é necessário indicar o nome do arquivo do documento Web ou da mídia que o *link* deverá apresentar ao usuário. Observe o exemplo a seguir. Este código direciona o usuário para outra página, chamada de `contato.html`, que esteja dentro do diretório em que se encontra o arquivo HTML:

```
<a href="contato.html"> Fale Conosco </a>
```

Já este código direciona o usuário para uma imagem `logotipo.png` que esteja dentro da pasta `img` do *site*:

```
<a href="img/logotipo.png"> Logo do HTML </a>
```

Links âncora

Os *links* do tipo âncora direcionam o usuário para um ponto específico dentro do mesmo documento. Normalmente, são utilizados para fazer *links* que voltam ao topo da página ou que direcionam o usuário a algum subtítulo da página.

Nesse caso, o valor do atributo `href` deve ser `#` (*hashtag* ou cerquilha), seguido pelo ID do elemento da página para o qual se quer direcionar. Segundo W3Schools (2019), o atributo `id` especifica um ID exclusivo (único) para um elemento HTML, cujo valor deve ser exclusivo dentro do documento HTML.

O código de exemplo a seguir define o texto “Voltar ao Topo” como um *link* que retornará à posição em que se encontra o `id= "topo"`, no título `h1` da página.

```
<h1 id="topo">Título Página</h1>
...
... Conteúdo da página ...
...
... Conteúdo da página ...
<a href="#topo"> Voltar ao Topo </a>
```

Tabelas

Tabelas são utilizadas como listas de dados distribuídos em células, formadas por linhas e colunas. São empregadas para apresentar dados de uma forma organizada, como em planilhas do Excel.

Para a criação de uma tabela no HTML são necessárias algumas *tags*:

- a tag `<table>` indica o início da tabela;
- a tag `<tr>` (*table row*) define o início de linha;
- a tag `<td>` (*table data*) marca a divisão de uma linha em células;
- a tag `<th>` (*table header*) marca a divisão da linha em células de cabeçalho. Essas células já vêm pré-formatadas (centralizadas e em negrito).

A tabelas podem conter todo tipo de elementos HTML, como texto, imagens, tabelas etc. Observe o exemplo: o código a seguir cria uma tabela com três linhas, sendo a primeira linha com células de cabeçalho `<th>`, com conteúdo centralizado e em negrito, e as outras duas com os dados de usuário `<td>`.

```
<table>
  <tr>
    <th>Nome</th>
    <th>Média</th>
  </tr>
  <tr>
    <td>Borigodolfa Silva</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Felizberto Leite</td>
    <td>9,5</td>
  </tr>
</table>
```

O resultado é a página exibida no navegador que você pode ver a seguir.

Nome	Média
Borigodolfa Silva	10
Felizberto Leite	9,5

As tabelas criadas apenas com código HTML não têm formatação, não apresentam bordas ou cores — isso é feito com o uso do CSS.

Formatando o código HTML com CSS

CSS (*cascade style sheet*, ou “folha de estilo em cascata”, ou ainda “cascata de estilos”) é uma linguagem que aplica formatações de estilo e de *layout* aos documentos Web. Segundo Scheidt (2015), é comum encontrarmos programadores que trabalham apenas com *front-end*, ou seja, com a parte visível do sistema, com a qual o usuário interage; o CSS é a principal tecnologia para definir a aparência de um sistema Web.



Saiba mais

Existem três formas de formatar os elementos HTML usando o CSS:

- **Código CSS inline:** a código *inline* é desenvolvido dentro da tag HTML que receberá a formatação. Utiliza-se o atributo `style` e define-se a propriedade e valor para formatação.
- **Código CSS interno ou incorporado:** a aplicação dos estilos internos é feita dentro da seção `<head>` do HTML. Utiliza-se a tag `<style>` e define-se o seletor, a propriedade e o valor para a formatação.
- **Código CSS externo:** a aplicação dos estilos externos é feita dentro de um arquivo `.css`. Esse tipo de estilo é o mais apropriado, pois um arquivo `.css` é capaz de formatar vários arquivos `.html` ao mesmo tempo. Ele segue os padrões Web Standards. Para utilizar arquivos CSS externos em uma página HTML, o documento HTML deve conter um *link* para o arquivo de estilos, utilizando a tag `<link>` dentro da seção `<head>`.

A aplicação de formatação às páginas HTML é feita com regras do CSS, e é dividida em três partes (SCHEIDT, 2015):

- um **seletor**, que define quais elementos serão formatados ou estilizados;
- a **propriedade**, que é a característica que se deseja configurar;
- o **valor** que a propriedade assumirá.

Seletores CSS

Para selecionarmos algum elemento a ser formatado, é necessário marcá-lo. Para isso utilizamos os seletores do CSS, que podem ser elementos HTML baseados no seu ID, classes, tipos, atributos, valores de atributos e muito mais.

Seletores de elementos

O seletor de elementos seleciona os elementos baseado nos seus nomes, no nome da *tag*. Você pode selecionar todos os elementos `<p>` de uma página, por exemplo, e aplicar a cor azul (`blue`), como neste código:

```
p {  
    color: blue;  
}
```

Todo o seletor vem acompanhando de um bloco de chaves `{ }`; entre as chaves são definidas as propriedades e valores a serem recebidos.

Seletores agrupados

O seletor agrupado seleciona mais de um elemento ao mesmo tempo. Isso minimiza o tamanho do código CSS, pois muitos elementos acabam recebendo a mesma formatação, como mesma cor, fontes, alinhamentos, etc.

Para utilizar seletores agrupados, basta separar cada um deles com a vírgula `(,)`, acompanhe:

```
h1,h2 {  
    color: red;  
    text-align: center;  
}
```

Esse código irá selecionar todos os títulos `<h1>` e `<h2>` e aplicar a cor vermelha (`red`) e o alinhamento centralizado a todos eles.

Seletores `id`

O seletor `id` seleciona um elemento específico, apenas um item, que receberá o `id` especificado dentro de sua `tag`. Um `id` deve ser único dentro de uma página, não deve conter caracteres especiais e não pode iniciar com números.

Para utilizar o seletor `id`, você deve definir, no HTML, o `id` para a `tag` que receberá a formatação e, no CSS, chamá-lo com o uso de cerquilha ou `hashtag` (#), seguida pelo nome do `id`, como mostra a Figura 2.

(a) HTML

```
<body>
    <h1> Páginas WEB </h1>
    <p id="especial"> Internet </p>
</body>
```

(b) CSS

```
#especial {
    color: red;
    text-align: center;
}
```

Figura 2. (a) Código HTML contendo um título `<h1>` e um parágrafo com o `id = "especial"`. (b) Código CSS que seleciona esse ID e formata na cor vermelha e com o alinhamento centralizado.

Seletores `class`

O seletor `class` seleciona uma classe de elementos específicos. A classe pode ser utilizada em vários elementos, inclusive em páginas diferentes, não sendo única.

Para utilizar o seletor `class`, você deve definir, no HTML as `tags` que receberão a formatação por meio da classe e, no CSS, chamá-la com o uso de ponto (.), seguido pelo nome da classe, como mostra a Figura 3.

(a) HTML

```
<body>
  <h1 class="centro"> Páginas WEB </h1>
  <p class="centro"> Internet </p>
</body>
```

(b) CSS

```
.centro {
  text-align: center;
}
```

Figura 3. (a) Código HTML contendo um título `<h1>` e um parágrafo, ambos com o seletor de classe `class = "centro"`. (b) Código CSS que seleciona a classe “centro” e formata com o alinhamento centralizado.

Propriedades e valores no CSS

Após a seleção do elemento a ser formatado, é necessário definir as propriedades a serem alteradas, ou seja, a característica que será modificada nesse elemento. As características que podem ser alteradas são cor, alinhamento, tamanho, entre outros efeitos — isso depende do elemento que será selecionado.

Toda propriedade tem valores que definem o que exatamente será colocado na propriedade a ser formatada. Qual cor? Qual alinhamento? Qual tamanho? Assim, de acordo com a propriedade atribuída, há diversos valores a serem escolhidos. Acompanhe um exemplo:

```
p{
  text-align:justify;
  line-height:1.5;
  text-indent:20px;
  color:gray;
}
```

Esse código apresenta o elemento `<p>` como seletor, recebendo as propriedades `text-align` (definição de alinhamento), `line-height` (definição de espaçamento entre linhas), `text-indent` (definição de recuo de primeira linha) e `color` (definição de cor). O `text-align` recebe o valor `justify` (justificado), o `line-height` recebe o valor 1.5 (uma linha e meia de espaço), o `text-indent` recebe o valor 20 px (pixels) e o `color` recebe a cor `gray` (cinza).

Alinhando elementos no CSS

Elementos do HTML podem ser alinhados com a propriedade `float` do CSS, que permite que o elemento flutue na página. Vejamos, como exemplo, três propriedades para alinhar imagens em um documento: `none` (nenhuma), `left` (esquerda) e `right` (direita).

Alinhamento à esquerda

Veja na Figura 4 como fazer o alinhamento à esquerda com HTML e CSS.

(a) HTML

```
<body>
    
<p>
    Float é uma propriedade CSS de posicionamento. Se você está
    familiarizado com projetos para mídia impressa, você pode pensar,
    de forma semelhante, numa imagem em um layout onde o texto a
    circunda quando necessário. Além de simples exemplos, como
    posicionar uma imagem ao lado de um bloco de texto, o float também
    é utilizado para estruturar layouts Web.
</p>
</body>
```

(b) CSS

```
.esquerda{
    float: left;
    width: 200px;
}
```

Figura 4. (a) Código HTML contendo uma imagem e logo abaixo um parágrafo. (b) Código CSS.

Perceba que o código CSS seleciona a classe `esquerda`, definindo seu alinhamento flutuante para `left` (esquerda) e, ainda, utiliza a propriedade `width` para definir a largura da imagem para 200 px. Acompanhe o resultado final na Figura 5.



Float é uma propriedade CSS de posicionamento. Se você está familiarizado com projetos para mídia impressa, você pode pensar, de forma semelhante, numa imagem em um layout onde o texto a circunda quando necessário. Além de simples exemplos, como posicionar uma imagem ao lado de um bloco de texto, o float também é utilizado para estruturar layouts Web.

Figura 5. Alinhamento de imagens à esquerda.

Alinhamento à direita

Veja na Figura 6 como fazer o alinhamento à direita com HTML e CSS.

(a) HTML

```
<body>
    
<p>
    Float é uma propriedade CSS de posicionamento. Se você está
    familiarizado com projetos para mídia impressa, você pode pensar,
    de forma semelhante, numa imagem em um layout onde o texto a
    circunda quando necessário. Além de simples exemplos, como
    posicionar uma imagem ao lado de um bloco de texto, o float também
    é utilizado para estruturar layouts Web.
</p>
</body>
```

(b) CSS

```
.direita{
    float: right;
    width: 200px;
    padding-left: 20px;
}
p{
    text-align:left;
    text-indent:20px;
}
```

Figura 6. (a) Código HTML com um parágrafo e uma imagem que são posicionados segundo as propriedades do CSS (b).

Na Figura 6b, perceba que o código CSS seleciona a classe direita, definindo seu alinhamento flutuante para `right` (direita). Além disso, a propriedade `width` define a largura da imagem para 200 px e a propriedade chamada `padding-left` cria uma margem interna do lado esquerdo da imagem. Nesse caso, é possível ver na Figura 7 que o texto não ficou grudado na imagem. Por fim, no seletor `p`, há a seleção do parágrafo, deixando-o com alinhamento alinhado à esquerda (`left`) e recuo de primeira linha de 20 px.

Float é uma propriedade CSS de posicionamento. Se você está familiarizado com projetos para mídia impressa, você pode pensar, de forma semelhante, numa imagem em um layout onde o texto a circunda quando necessário. Além de simples exemplos, como posicionar uma imagem ao lado de um bloco de texto, o float também é utilizado para estruturar layouts Web.



Figura 7. Alinhamento de imagens à direita.

Alinhamento de `div` com a propriedade `float`

Uma *tag* `div` pode ser alinhada em relação à página Web. Pode-se alinhar à direita, ao centro ou à esquerda. Como vimos anteriormente, a propriedade `float` do CSS faz com que o elemento flutue na página. Acompanhe este exemplo:

(a) HTML:

```
<h1> Alinhamento de Div </h1>
<div class="esquerda"></div>
<div class="direita"></div>
<div class="centro"></div>
```

(b) CSS:

```
div {  
    border: 1px solid #000000;  
    height: 50px;  
    width: 50px;  
}  
  
.esquerda {  
    background-color: #0000FF;  
    float: left;  
}  
.direita {  
    background-color: #FF0000;  
    float: right;  
}  
.centro{  
    background-color:#C0C0C0;  
    float: none;  
    margin:0 auto;  
}
```

Aqui temos dois códigos: em (a) o código HTML contém três elementos de <div>, com diferentes classes: esquerda, direita e centro. Em (b) o código CSS seleciona os elementos div e coloca uma borda sólida na cor preta (#000000), além de definir uma altura de 50 px e uma largura também de 50 px. Logo após, seleciona o elemento de classe esquerda, formatando-o com plano de fundo na cor azul (#0000FF) e alinhamento flutuante à esquerda. Abaixo é selecionado o elemento de classe direita, que define a cor de plano de fundo vermelha (#FF0000) e o alinhamento flutuante à direita. A última formatação é para o elemento de classe centro, o qual recebe a cor de plano de fundo cinza (#C0C0C0), alinhamento flutuante none (nenhum) e margem 0 auto (centralizando o elemento de acordo com a página).

O resultado desse exemplo será uma página com três quadrados, cada um com uma cor e um alinhamento, conforme a Figura 8.



Figura 8. Página “Alinhamento de Div”.

Elaborando páginas com a semântica do HTML5

Para estruturar os conteúdos de documentos Web existem *tags* que fornecem semântica ao HTML5, que se referem aos tipos de conteúdo da página, como menus, artigos, cabeçalho, rodapé. Esse conteúdo de *layout* garante uma melhor visualização por parte do navegador e do usuário, além de proporcionar uma otimização para os processos de SEO (*search engine optimization*, ou “otimização para mecanismos de busca”).

Os elementos de semântica do HTML5 são utilizados como níveis de blocos, que são empregados para a criação de páginas, e sua visualização não muda — eles são iguais a uma *div*:

```
<header>, <footer>, <article>, <aside>, <section> e <nav>
```

Visualize esses elementos na Figura 9.

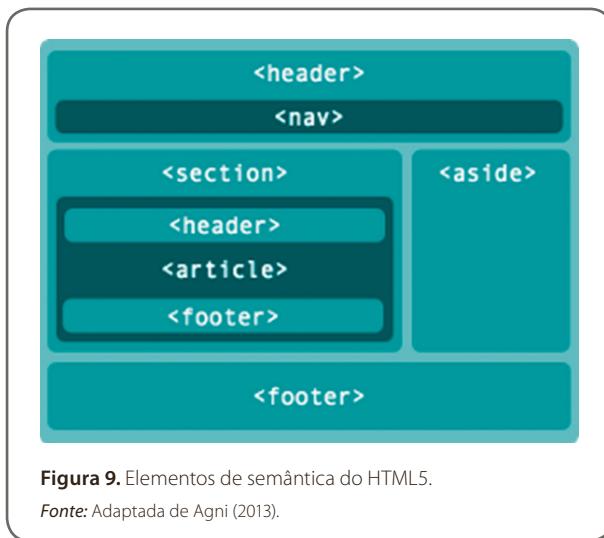


Figura 9. Elementos de semântica do HTML5.

Fonte: Adaptada de Agni (2013).

Segundo Ferreira ([201-?]), o HTML5 modifica a forma de escrever código e organizar a informação na página: mais semântica com menos código, mais interatividade sem precisar instalar *plug-ins* e perder *performance*. Trata-se da criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas.

Cabeçalho com o elemento <header>

O elemento <header> é utilizado para estruturar um cabeçalho de um documento HTML, podendo apresentar o nome do *site*, logomarca, *slogan*, títulos etc. Ele pode ser utilizado como cabeçalho principal do documento ou como cabeçalho de um artigo ou seção. Acompanhe um exemplo de código em HTML com um *container* de cabeçalho <header>, onde há um título <h1>.

```
<header id="topo">
  <h1>Design Gráfico </h1>
</header>
```

Navegação global com o elemento <nav>

O elemento <nav> é utilizado para agrupar uma lista de *links* para outras partes do *site*, como uma lista de navegação local ou global. Este exemplo mostra um código em HTML com um *container* de navegação <nav>, onde há uma lista, que, com a devida formatação em CSS, poderá ser um menu horizontal.

```
<nav id="menu"> <!--Espaço do menu-->
  <ul>
    <li> <a href="index.html">Agência</a></li>
    <li> <a href="#"> Promoções </a></li>
    <li> <a href="#"> Galeria </a></li>
    <li> <a href="#"> Contato </a></li>
  </ul>
</nav> <!--Fim MENU-->
```

Elemento <section>

O elemento <section> identifica a seção de conteúdo da página Web. Podemos utilizar várias *sections* separando assuntos da página.

A diferença entre <section> e <div> é que section divide em seções e assuntos, enquanto div é apenas para dividir qualquer conteúdo, sem uma finalidade específica.

O exemplo a seguir mostra um código em HTML com um *container* de seção <section>, o qual apresenta um título e um parágrafo sobre o assunto Agência de Viagens — simulando um *site* para uma agência turística.

```
<section id="conteudo"> <!--Espaço para conteúdo do site-->
  <h1> World Viagens – Conhecendo o Mundo! </h1>
  <p>
    A agência World Viagens traz uma imensidão de oportunidades para você
    conhecer o mundo e aumentar sua cultura.
  </p>
</section>
```

O conteúdo central com o elemento <article>

O elemento `<article>` identifica um conteúdo independente e altamente relevante, algo externo ao conteúdo do *site*, que pode ser um *post*, um artigo, um bloco de texto ou ainda um miniaplicativo embutido no conteúdo (*widget*). Podemos ter vários elementos `article` em mesmo documento.

Este exemplo mostra um código em HTML com um *container* de conteúdo `<article>`, que demonstra um título e um parágrafo sobre uma rede social — simulando um *post* de divulgação.

```
<article id="post"> <!--Espaço para conteúdo do site-->
  <h1> Facebook – entre ou cadastre-se </h1>
  <p>
    Crie uma conta ou entre no Facebook. Conecte-se com amigos, familiares
    e outras pessoas que você conheça. Compartilhe fotos e vídeos, envie
    mensagens.
  </p>
</article>
```

Conteúdos relacionados com o elemento <aside>

O elemento `<aside>` representa uma seção de uma página que consiste em conteúdo que é tangencialmente relacionado ao conteúdo do seu entorno, mas que poderia ser considerado separado do conteúdo, como barras laterais. Esse tipo de seção, muitas vezes, contém explicações laterais, como a definição de um glossário, ou conteúdo vagamente relacionado, como avisos. São exemplos a biografia do autor, ou, em aplicações Web, informações de perfil ou *links* de *blogs* relacionados (MOZILLA, 2019).

Neste exemplo, o código em HTML tem um *container* de conteúdo `<aside>`; dentro dele há um título e um parágrafo que traz o perfil do autor de um *site*, simulando um *blog* pessoal.

```
<aside id="post"> <!--Espaço para conteúdo do site-->
  <h1> Perfil do autor </h1>
  <p>
    Steve é um experiente desenvolvedor de software. Ele domina linguagens
    de marcação HTML e de formatação CSS na criação de páginas web.
  </p>
</aside>
```

Rodapé com o elemento <footer>

O elemento <footer> marca a área inferior do documento, normalmente conhecida como rodapé. Ela pode se referir ao conteúdo geral do documento ou ao conteúdo de uma seção específica à qual está subordinada.

O exemplo a seguir mostra um código em HTML com um *container* de rodapé <footer>, que tem o endereço eletrônico “worldagencia@worldagencia.com”.

```
<footer id="rodape"> <!--Espaço para rodapé-->
    <address>
        worldagencia@worldagencia.com
    </address>
</footer>
```

A Figura 10 é gerada através da estrutura do HTML5 e a formatação CSS3 apresentadas na Figura 11.

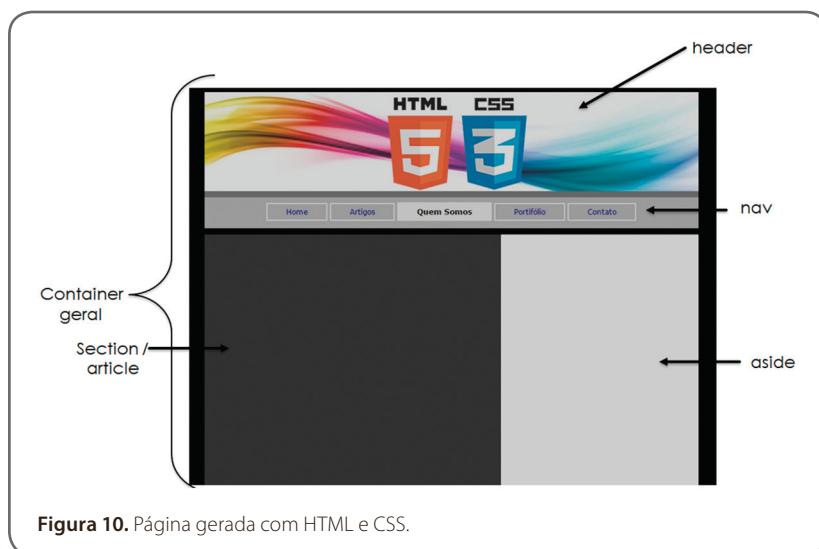


Figura 10. Página gerada com HTML e CSS.

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4  <meta charset="utf-8">
5  <title> Layout com div </title>
6  <link rel="stylesheet" type="text/css" href="estilo.css">
7  </head>
8  <body>
9  <div id="geral">
10 <header></header>
11 <nav> ←←←
12 <ul>
13 <li><a href="#">Home</a></li>
14 <li><a href="#">Artigos</a></li>
15 <li><a href="#">Quem Somos</a></li>
16 <li><a href="#">Portifólio</a></li>
17 <li><a href="#">Contato</a></li>
18 </ul>
19 </nav> ←←←
20 <aside> </aside>
21 <article> </article> ←←←
22 </div>
23 </body>
24 </html>

```

Figura 11. Código HTML e CSS que gerou a página da Figura 10.

Na estrutura HTML é utilizada a semântica do HTML5, que cria o topo para a página, uma barra de navegação (menu), um espaço de uma barra lateral de conteúdos e um espaço de conteúdo de artigos. Veja na Figura 12 o código que define os estilos definidos para cada *container*.

```

1  @charset "utf-8";
2  body{
3      background-color:#000000;
4      color:#ffffff;
5      font-family:Verdana,Arial,Helvetica;
6      font-size:12px;
7      text-align:justify;
8  }
9  #geral{
10     background-color:#666666;
11     margin:auto;
12     width:800px;
13  }
14  header{ ←←←
15     background-image:url(topo.png);
16     height:160px;
17     opacity:0.7
18  }
19  nav{ ←←←
20     background-color:#999999;
21     height:50px;
22     margin-top:10px;
23     text-align:center;
24  }
25  aside{ ←←←
26     background-color:#cccccc;
27     width:320px; float: left;
28     height:640px;
29     margin-top:10px;
30  }
31  article{ ←←←
32     background-color:#333333;
33     width:480px;
34     float:right;
35     height:640px;
36     margin-top:10px;
37  }

```

Figura 12. Semântica do HTML5.

Perceba na Figura 12:

- No espaço do cabeçalho `<header>` foi colocada como plano de fundo uma imagem de *logo* para uma empresa, a altura de 160 px e uma leve transparência de 70%.
- No espaço do menu `<nav>` foi colocada uma cor de plano e fundo, uma altura de 50 px, uma margem no topo de 10 px e o alinhamento do texto centralizado.
- No conteúdo de `<aside>` foi empregada uma cor de plano de fundo, uma largura de 320 px e altura de 640 px, além do posicionamento à esquerda.
- No conteúdo de artigo `<article>` foi utilizada uma cor de plano de fundo, uma largura de 480 px e altura de 640 px, além do posicionamento à direita.



Link

O artigo “Layout CSS passo a passo”, disponível no *link* a seguir, explica como criar um arquivo HTML e um arquivo CSS e como integrá-los para que funcionem juntos, ajudando a construir *sites* mais complexos.

<https://qrgo.page.link/7kywq>



Referências

AGNI, E. *HTML5: entendendo a estrutura e a semântica*. [S. l.], 2013. Disponível em: <https://medium.com/@eduagni/html5-entendendo-a-estrutura-e-a-sem%C3%A2ntica-db5f17808c7>. Acesso em: 30 dez. 2019.

FERREIRA, E.; EIS, D. *HTML5: curso W3C escritório Brasil*. [S. l., 201-?]. Disponível em: <http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>. Acesso em: 30 dez. 2019.

FLATSCHART, F. *HTML 5: embarque imediato*. Rio de Janeiro: Brasport, 2011.

MOZILLA. *Elementos HTML*. [S. l.], 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>. Acesso em: 30 dez. 2019.

PRESSCOTT, P. *HTML 5*. [S. l.]: Babelcube Inc., 2015.

SANDERS, B. *Smashing HTML5*: técnicas para a nova geração da web. Porto Alegre: Bookman, 2012.

SCHEIDT, F. A. *Fundamentos de CSS*: criando design para sistemas web. [S. l.]: Outbox Livros Digitais, 2015.

W3SCHOOLS. *Blocos HTML*. [S. l.], 2019. Disponível em: http://www.w3schools.com/html/html_blocks.asp. Acesso em: 30 dez. 2019.

Leituras recomendadas

ADAMS, C. et al. *A arte e a ciência da CSS*: crie web designs inspiradores baseados em padrões. Porto Alegre: Bookman, 2009.

MAZZA, L. *HTML5 e CSS3*: domine o Web do futuro. São Paulo: Casa do Código, 2014.

W3C. *About W3C Standards*. [S. l.], 2012. Disponível em: <http://www.w3.org/standards/about.html>. Acesso em: 30 dez. 2019.

W3C. *HTML & CSS*. [S. l.], 2016. Disponível em: <http://www.w3.org/standards/webdesign/htmlcss>. Acesso em: 30 dez. 2019.

W3SCHOOLS. *Tabelas CSS*. [S. l.], 2014. Disponível em: http://www.w3schools.com/css/css_table.asp. Acesso em: 30 dez. 2019.



Fique atento

Os *links* para sites da Web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

