

PROGRAMAÇÃO FRONT END I

Diego César Batista Mariano



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Folha de estilo em cascatas (CSS)

Objetivos de aprendizagem

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

- Conceituar CSS.
- Descrever o funcionamento e as vantagens do CSS.
- Implementar leiautes em HTML com CSS.

Introdução

Arquivos CSS permitem uma melhor organização do código fonte de páginas da internet, armazenando a maior parte de códigos referentes à aparência da página.

Neste capítulo, você conhecerá o conceito de CSS e aprenderá a descrever seu funcionamento e suas vantagens. Além disso, saberá como implementar leiautes em HTML com CSS.

Conceito de CSS

CSS — do inglês, *cascading style sheet*, ou, na tradução, “folha de estilo em cascatas” — é um formato de arquivos que permite o armazenamento de estilos para documentos Web. Arquivos CSS foram projetados para trabalhar em conjunto com os HTML (Figura 1), separando o armazenamento do conteúdo de páginas HTML dos códigos relacionados à aparência e ao estilo, como tamanho de textos, espaçamentos, fontes e cores de elementos. Tal separação tem por objetivo facilitar a edição e manutenção de Websites com uma grande quantidade de páginas em que o leiaute é mantido.



Figura 1. Trabalho do CSS em conjunto com o HTML.

Fonte: jossnat/Shutterstock.com.



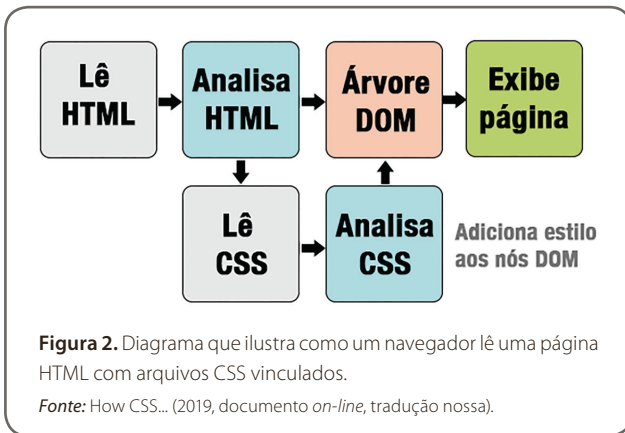
Fique atento

Enquanto arquivos de páginas de internet devem ser salvos com a extensão “.html”, arquivos de folhas de estilo em cascata são salvos com “.css”.

Funcionamento e vantagens do CSS

Para que você possa compreender como CSS funciona, é necessário, primeiramente, entender como um navegador lê uma página HTML. Segundo How CSS... (2019), o carregamento de uma página ocorre em seis etapas: (i) o navegador (*browser*) carrega o documento HTML de um servidor na Internet ou de um arquivo local; (ii) a seguir, converte o documento HTML em um modelo de objeto de documento (DOM); (iii) são carregados recursos vinculados, como imagens, vídeos incorporados e folhas de estilo CSS; (iv) é feita a análise do arquivo CSS, levando-se em consideração os seletores e as regras

de estilo que devem ser aplicadas a eles, como elementos, classes, IDs e assim por diante; (v) a árvore de renderização DOM é construída usando a estrutura do HTML e os estilos obtidos no arquivo CSS; e (vi) a página completa é, por fim, exibida na tela. A Figura 2, a seguir, mostra um esquema desse processo.



Na prática, um navegador carrega os dados e a estrutura da página no arquivo HTML. Depois, analisa arquivos CSS vinculados, buscando formatações e estilos que devem ser aplicados para visualização final do *site* na tela do computador. Por fim, ele junta todas as informações disponíveis em um documento armazenado na memória do computador (DOM) e exibe a página formatada.



Saiba mais

DOM — do inglês, *document object model*, ou, na tradução, modelo de objeto de documentos — é uma interface para representação e interação de documentos HTML e XML. O DOM pode ser visto com uma árvore em que cada elemento é representado como um nó. Por exemplo, em um arquivo HTML, a tag `<html>` pode ser vista como um nó. As tags `<head>` e `<body>` seriam nós filhos de `<html>`, enquanto, tags como `<div>` e `<p>` seriam filhas de `<body>`. Os objetos de uma árvore DOM podem ser manipulados pelo uso de diversos métodos.

Regra de estilização

A linguagem CSS permite a manipulação de estilos de arquivos por meio da chamada regra de estilização (SILVA, 2015). Nessa regra, uma formatação de estilos é composta por um seletor seguido por uma ou mais declarações separadas por ponto e vírgula. Declarações CSS, por sua vez, são compostas por uma determinada propriedade e um valor que pode ser aplicado a ela, conforme esquema da Figura 3.



Seletores indicam elementos em que as regras CSS serão aplicadas na marcação HTML. Um exemplo simples de aplicação da linguagem CSS seria a alteração de textos de uma página (armazenados na *tag* <p>) para a cor azul. No exemplo anterior, a propriedade `color` (cor) foi aplicada a todos os textos presentes em *tags* <p>, recebendo o valor `blue` (azul).



Saiba mais

Qual é a diferença entre `id` e `class`?

Ambos servem para identificar elementos em documentos Web indicados pelo desenvolvedor. Entretanto, há diferenças sutis, como as que seguem.

- Identificadores (`id`) devem possuir um nome único, declarado uma única vez na página. IDs devem ser declarados no arquivo CSS, precedidos do caractere *hash*, também conhecido como jogo-da-velha, cerquilha ou sustenido (`#`).
- Classes (`class`) podem ser utilizadas diversas vezes em um documento Web. Elas devem ser declaradas no arquivo CSS, precedidas pelo caractere ponto (`.`)

Leiautes em HTML com CSS

Há três formas de se realizar declarações CSS (MARIANO; MELO-MINARDI, 2017), apresentadas no Quadro 1, a seguir.

Quadro 1. Maneiras de realizar declarações CSS

Tipo	Onde é aplicado?	Definição
<i>Inline</i>	Arquivo HTML	Adotado diretamente na <i>tag</i> em que se deseja aplicar o estilo por meio do atributo <code>style</code> . Exemplo: <pre><p style="color: blue">Meu texto</p></pre>
Incorporado	Arquivo HTML	Declarações são realizadas dentro da <i>tag</i> <code><style></code> , geralmente declarada no cabeçalho <code><head></code> . Exemplo: <pre><style> p { color: blue; } </style></pre>

(Continua)

(Continuação)

Quadro 1. Maneiras de realizar declarações CSS

Tipo	Onde é aplicado?	Definição
Importado	Arquivo CSS	<p>Método mais recomendado. Declarações são inseridas em um arquivo CSS externo que, depois, é interligado ao arquivo HTML. A referência é realizada por meio da <i>tag</i> <code><link></code>. No arquivo CSS, são permitidas apenas declarações na sintaxe CSS.</p> <p>Exemplo:</p> <pre><!-- Arquivo index.html --> <link href="estilo. css" rel="stylesheet"> /* Arquivo estilo.css */ p { color: blue }</pre>

Fonte: Adaptado de Mariano e Melo-Minardi (2017).

Um navegador define o estilo aplicado de acordo com a ordem de prioridade em que a declaração foi aplicada (efeito cascata). Por exemplo, declarações *inline* têm prioridade sobre as incorporadas, que têm prioridade sobre as importadas. Declarações dos tipos *inline* e incorporadas são realizadas diretamente no arquivo HTML e, por isso, devem ser evitadas.

Se nenhuma declaração for aplicada a um determinado elemento, o navegador carrega sua folha de estilos-padrão. Por exemplo, se apenas um texto for inserido na *tag* `<body>`, o navegador define por padrão que a cor do texto deve ser preta (`color: black`) e fundo deve ser branco (`background-color: white`).



Saiba mais

Comentários

Comentários são trechos de código que não são lidos pelo navegador durante a análise do arquivo CSS. Eles permitem ao desenvolvedor organizar e documentar o código. Em CSS, comentários são declarados entre os símbolos de `/*` e `*/`. Observe que comentários HTML são declarados de forma diferente aos CSS:

```
/* Exemplo de comentário CSS: este trecho não será lido
pelo navegador */
<!-- Comentário HTML -->
```

Importação de folhas de estilo

Agora, você verá, na prática, como criar sua primeira folha de estilos e vinculá-la a um arquivo HTML. Primeiro, crie o arquivo HTML com o seguinte conteúdo (nomeie o arquivo como “index.html”):



Exemplo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Minha primeira página HTML + CSS</title>

    <!-- importando arquivos CSS -->
    <link rel="stylesheet" href="estilo.css">
  </head>

  <body>
    <p>Meu texto</p>
    <p class="azul">Estilo múltiplo</p>
    <p id="vermelho">Estilo único</p>
    <p class="azul">Estilo múltiplo</p>
  </body>
</html>
```


Observe que, na linha 7, é realizada uma referência a um arquivo denominado “estilo.css”. Este, ainda não criado, armazenará todas as declarações do tipo CSS. Para que o navegador compreenda que a *tag* `<link>` é interligada a um arquivo de folhas de estilo, é necessário adicionar o atributo `rel` que deve receber o valor `stylesheet`, além de um atributo `href` com o nome do arquivo CSS (desde que os arquivos HTML e CSS estejam na mesma pasta – caso contrário, é necessário indicar o endereço completo).

Crie, agora, o arquivo CSS com o seguinte conteúdo:



Exemplo

```
* {  
    margin: 0;  
    padding: 0;  
}  
  
body {  
    background-color: black;  
}  
  
p {  
    color: green;  
}  
  
.azul {  
    color: cyan; /* azul ciano */  
}  
  
#vermelho {  
    color: red;  
}
```

Esses códigos utilizam diversos seletores. O primeiro é o seletor universal (*), que é usado para aplicar estilos a elementos de qualquer tipo. Nesse caso, você usou o seletor universal para remover margens inseridas por padrão em certos tipos de navegadores. O próximo seletor foi usado na *tag* `body`, que armazena todo o corpo da página, ou seja, o conjunto de códigos que gera elementos visuais. Na *tag* `body`, foi inserida a propriedade `background-color`, que altera a cor do plano de fundo para preto. A seguir, a propriedade `color` com diferentes valores é aplicada a três seletores: *tag* `<p>`, classe “azul” e *id* “vermelho”. Ao executar o arquivo HTML em um navegador, a seguinte página da Figura 4 será carregada.

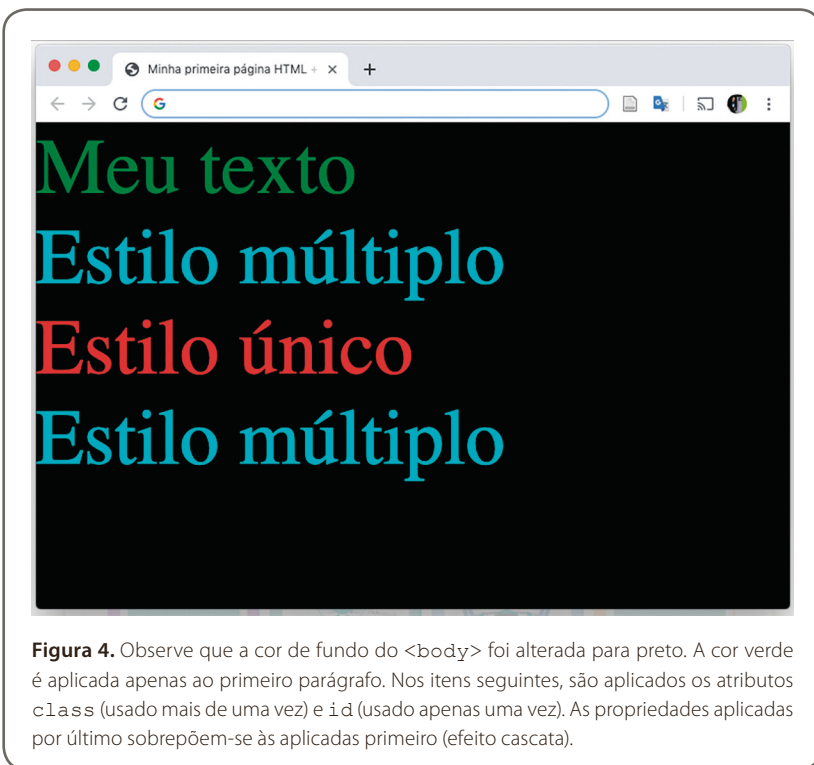


Figura 4. Observe que a cor de fundo do `<body>` foi alterada para preto. A cor verde é aplicada apenas ao primeiro parágrafo. Nos itens seguintes, são aplicados os atributos `class` (usado mais de uma vez) e `id` (usado apenas uma vez). As propriedades aplicadas por último sobrepõem-se às aplicadas primeiro (efeito cascata).

Nas próximas seções, você conhecerá diversas propriedades e possíveis valores que podem ser aplicados a elas.

Tipografia

As propriedades tipográficas permitem a formatação dos textos em uma determinada página. Em geral, elas iniciam pelos prefixos “font-” ou “text-”, como a propriedade `font-size`, que define o tamanho da fonte a ser utilizada, ou a propriedade `text-align`, que define o alinhamento do texto (alinhado à direita, à esquerda ou centralizado).

No Quadro 2, a seguir, você pode ver algumas das propriedades mais utilizadas para formatação de textos.

Quadro 2. Propriedades de tipografia

Propriedade	Função	Exemplo
<code>color</code>	Aplica cor ao texto; recebe nome de cor em inglês (ex.: <code>blue</code> , <code>red</code> , <code>black</code>), código hexadecimal RGB (ex.: <code>#000000</code>), etc.;	<pre>/* Cor branca */ color: #FFFFFF;</pre>
<code>font-size</code>	Altera o tamanho do texto.	<pre>/* Fonte em tamanho grande */ font-size: 40px;</pre>
<code>font-style</code>	Altera o estilo da fonte; recebe os valores <code>italic</code> , <code>normal</code> ou <code>oblique</code> ;	<pre>/* Itálico */ font-style: italic;</pre>
<code>font-family</code>	Define a família da fonte utilizada, nome da família. Ex.: <code>arial</code> , <code>helvetica</code> , <code>times new roman</code> .	<pre>/* Definindo fontes */ font-family: helvetica, arial;</pre>
<code>font-weight</code>	Realça palavras de um texto (negrito); recebe um valor que vai de 100 (mais fino) a 900 (mais grosso), ou <code>bold</code> (negrito).	<pre>/* Negrito */ font-weight: bold;</pre>

(Continua)

(Continuação)

Quadro 2. Propriedades de tipografia

Propriedade	Função	Exemplo
<code>text-align</code>	Define o alinhamento do texto; pode ser à esquerda (<code>left</code>), à direita (<code>right</code>), centralizado na tela (<code>center</code>) ou justificado (<code>justify</code>);	<pre>/* Centralizar */ text-align: center;</pre>
<code>line-height</code>	Espaçamento entre linhas.	<pre>/* Dobrando o espaço entre linhas */ line-height: 2;</pre>
<code>text-decoration</code>	Permite a inserção de sublinhado; <code>overline</code> : insere linha superior; <code>line-through</code> : insere linha no meio do texto; <code>underline</code> : insere sublinhado; <code>none</code> : remove decoração.	<pre>/* Removendo sublinhado de links */ text-decoration: none;</pre>

Fonte: Mariano e Melo-Minardi (2017).

Background

O advento da popularização de Internet banda larga, aliado ao uso de folhas de estilo, permitiu uma evolução no uso de *backgrounds*, também conhecidos como planos de fundo (ADAMS *et al.*, 2009).

A propriedade `background` permite diversas configurações de planos de fundo, como definição de cor ou, até mesmo, inserção e posicionamento de imagens.

No Quadro 3, a seguir, você pode ver algumas das propriedades para definição de *backgrounds*.

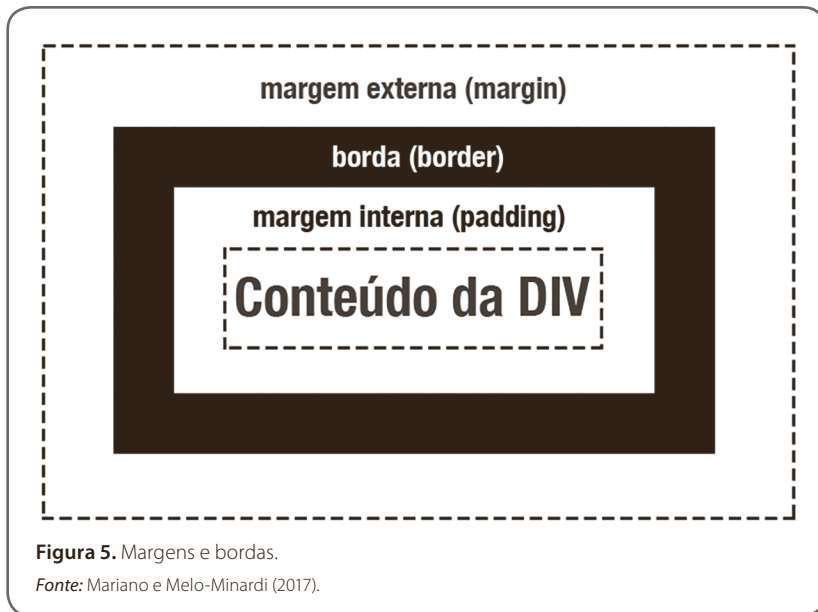
Quadro 3. Backgrounds

Propriedade	Função	Exemplo
background-color	Insere uma cor no <i>background</i> .	<pre>/* Cor azul */ background-color: #0000FF;</pre>
background-image	Insere uma imagem como <i>background</i> . Requer o endereço de uma imagem.	<pre>/* Imagem de fundo */ background-image: url("imagem.jpg");</pre>
background-repeat	Define se a imagem se repetirá no <i>background</i> . Requer a propriedade <i>background-image</i> declarada anteriormente. Recebe os valores: <i>repeat</i> (repetição nos eixos x e y), <i>no-repeat</i> (sem repetição), <i>repeat-x</i> e <i>repeat-y</i> .	<pre>/* Impedir repetição */ background-repeat: no-repeat;</pre>
background-position	Define o posicionamento de uma imagem no <i>background</i> . Requer a propriedade <i>background-image</i> declarada anteriormente. Recebe as distâncias x e y ou as palavras-chave: <i>top left</i> , <i>top center</i> , <i>top right</i> , <i>center left</i> , <i>center center</i> , <i>center right</i> , <i>bottom left</i> , <i>bottom center</i> , <i>bottom right</i> , <i>top</i> , <i>right</i> , <i>bottom</i> , <i>left</i> .	<pre>/* Margem para background */ background-position: 100px 200px;</pre>
background-attachment	Permite a fixação ou rolagem de uma imagem no <i>background</i> . Requer a propriedade <i>background-image</i> declarada anteriormente.	<pre>/* Imagem fixa */ background-attachment: fixed;</pre>

Fonte: Mariano e Melo-Minardi (2017).

Dimensão de elementos

CSS fornece uma série de propriedades para definição de margens e espaçamentos, como bordas (*border*), margens internas (*padding*), margens externas (*margin*), além das propriedades de altura (*height*) e largura (*width*) (Figura 5). Tais propriedades podem ser aplicadas a diversos tipos de *tags*, como divisórias (`<div>`), imagens (``), dentre outras.



A linguagem CSS permite uma grande gama de unidades de medida para expressar comprimento, como ponto (pt), paica (pc), centímetro (cm), milímetro (mm) e polegada (in). Além disso, há unidades que dependem do tamanho da fonte, como as de medida em e ex. Entretanto, CSS possui uma unidade de medida própria, o px, que se deriva do *pixel*, que a menor unidade de medida de uma imagem digital. As unidades baseadas em pixels são bastante utilizadas, por exemplo, na resolução de monitores, como o HD (1280 x 720 pixels), Full HD ou FHD (1920 x 1080 pixels), 4K ou Ultra HD (3840 x 2160 pixels) e 8K (7680 x 4320 pixels). Quanto maior a resolução, maior a quantidade de pixels que a tela terá e, consequentemente, melhor qualidade de imagem.

Tamanhos também podem ser representados em porcentagens (%), o que tem uma grande utilidade para as chamadas páginas que se adequam ao *design* responsivo (aquelas que se ajustam ao tamanho da tela de quem acessa o *site*).

No Quadro 4, a seguir, você verá algumas propriedades para definições de tamanhos e espaçamentos.

Quadro 4. Dimensão de elementos e diferentes tipos de margens

Propriedade	Função	Exemplo
height	Define a altura de elemento.	height: 200px;
max-height	Define a altura máxima. Usado em <i>sites</i> com <i>design</i> responsivo, caso a janela seja reajustada.	max-height: 300px;
min-height	Define a altura mínima. Usado em <i>sites</i> com <i>design</i> responsivo, caso a janela seja reajustada.	min-height: 100px;
width	Define a largura de um elemento.	width: 800px;
max-width	Define a largura máxima. Usado em <i>sites</i> com <i>design</i> responsivo, caso a janela seja reajustada.	max-width: 1000px;
min-width	Define a largura mínima. Usado em <i>sites</i> com <i>design</i> responsivo, caso a janela seja reajustada.	width: 800px;
margin	Define as margens externas. Quando recebe quatro valores, correspondem à distância para topo (1º), lado direito (2º), rodapé (3º) e lado esquerdo (4º); três valores indicam (1º) topo, (2º) esquerda e direita, e (3º) rodapé; dois valores indicam distância para o (1º) topo e rodapé e (2º) lado direito e esquerdo; um valor é válido para todos os outros.	/* Margens superior, direita, inferior e esquerda */ margin: 5px 0 10px 0; /* Margens superior, lados e inferior */ margin: 5px 0 10px; /* Margens topo/rodapé e lados */ margin: 10px 0; /* Todas iguais */ margin: 10px;

(Continua)

(Continuação)

Quadro 4. Dimensão de elementos e diferentes tipos de margens

Propriedade	Função	Exemplo
margin-top	Margem superior.	margin-top: 10px;
margin-bottom	Margem inferior.	margin-bottom: 10px;
margin-left	Margem esquerda.	margin-left: 10px;
margin-right	Margem direita.	margin-right: 10px;
padding	Margem interna. Mesmas regras aplicadas à <i>margin</i> são válidas.	padding: 10px;
padding-top	Margem interna superior.	padding-top: 10px;
padding-bottom	Margem interna inferior.	padding-bottom: 10px;
padding-left	Margem interna esquerda.	padding-left: 10px;
padding-right	Margem interna direita.	padding-right: 10px;
Border	Delimita espessura, tipo e cor da linha de borda.	<pre>/* Borda sólida, tamanho 1px, preta */ border: 1px solid black;</pre>

Fonte: Mariano & de Melo-Minardi (2017).

Posicionamento

CSS permite a definição de posicionamento de um elemento pela propriedade `position`, que pode receber três possíveis valores: `fixed`, `relative` e `absolute` (Figura 6). O valor `relative` posiciona elementos com base em seu canto superior esquerdo. Sendo assim, elementos `relative` são inseridos abaixo de outros elementos `relative`. Por padrão, novos elementos declarados recebem essa configuração.

O valor `fixed` permite que elementos sejam posicionados em uma posição fixa da tela, enquanto o valor `absolute`, que elementos sejam posicionados com base em um elemento referência, possibilitando sobreposições entre elementos.

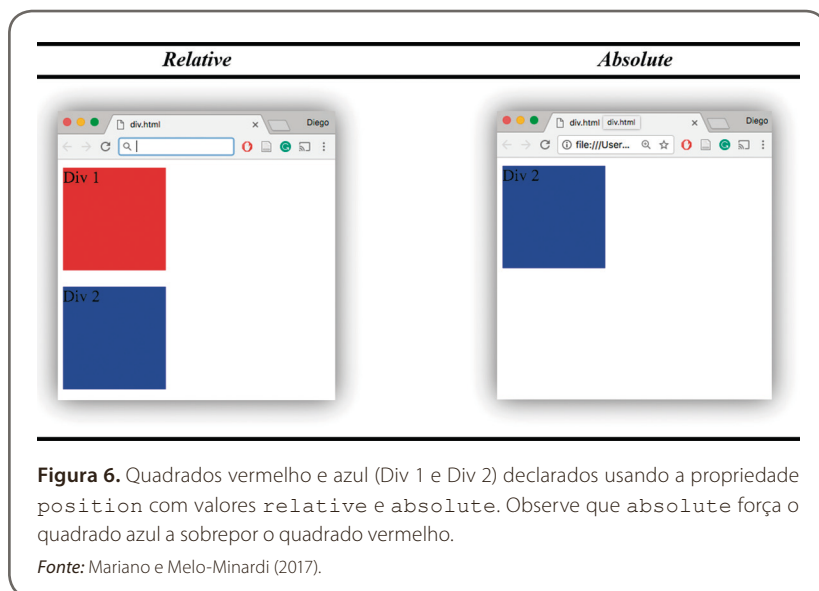


Figura 6. Quadrados vermelho e azul (Div 1 e Div 2) declarados usando a propriedade `position` com valores `relative` e `absolute`. Observe que `absolute` força o quadrado azul a sobrepor o quadrado vermelho.

Fonte: Mariano e Melo-Minardi (2017).

A posição de elementos do tipo `fixed` e `absolute` pode ser definida com base em quatro propriedades: (i) `top`: distância do elemento até o topo da página; (ii) `right`: distância do elemento até a margem direita; (iii) `bottom`: distância do elemento até a parte inferior da página; e (iv) `left`: distância do elemento até a margem esquerda.



Referências

ADAMS, C. *et al.* *A arte e a ciência da CSS: crie web designs inspiradores baseados em padrões*. Porto Alegre: Sitepoint; Bookman, 2009. 248 p.

HOW CSS works. *MDN Web Docs*, Mountain View, 2019. Disponível em: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works. Acesso em: 28 set. 2019.

MARIANO, D.; MELO-MINARDI, R. *Introdução à programação web para bioinformática: HTML, CSS, PHP & JavaScript*. North Charleston: CreateSpace Independent Publishing Platform, 2017. 410 p.

SILVA, M. S. *Fundamentos de HTML5 e CSS3*. São Paulo: Novatec, 2015. 304 p.

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

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS