

Fundamentos de CSS

Conteudista

Prof. Me. Marco Antonio Sanches Anastacio

Revisão Textual

Prof.^a Dra. Selma Aparecida Cesarin



OBJETIVOS DA UNIDADE

- Entender as regras e a função do CSS na programação *web*;
- Compreender como as folhas de estilo organizam o conteúdo de uma página *web*, aplicando os conhecimentos adquiridos na construção de *layouts*.

Atenção, estudante! Aqui, reforçamos o acesso ao conteúdo *on-line* para que você assista à videoaula. Será muito importante para o entendimento do conteúdo.

Este arquivo PDF contém o mesmo conteúdo visto *on-line*. Sua disponibilização é para consulta *off-line* e possibilidade de impressão. No entanto, recomendamos que acesse o conteúdo *on-line* para melhor aproveitamento.

O Que é CSS?

A finalidade do HTML, desde sua criação, sempre foi lidar somente com a marcação e a estruturação de conteúdo. Entretanto, quando abrimos nosso navegador, vemos páginas atraentes e agradáveis, que extrapolam uma simples marcação de *tags* HTML.

No início do HTML, recorria-se a elementos e atributos específicos da própria Linguagem para definir a formatação das páginas. Por exemplo, o elemento `` foi introduzido na versão 3.2 do HTML para definir fonte, tamanho e cor de texto.

A partir da versão 4.0 do HTML, o *World Wide Web Consortium (W3C)* incorporou o CSS (*Cascading Style Sheet*, ou Folha de Estilo em Cascatas), com a finalidade de se encarregar de toda a apresentação visual de um *site*, alterar elementos como fonte, cor e tamanho de textos, espaçamento entre blocos, enfim, todo o aspecto estético de uma página *web*.

A utilização do CSS trouxe vários benefícios e, entre eles, podemos citar:

- Garantir flexibilidade e controle dos elementos na página;
- Simplificar a criação de páginas *web*;
- Permitir que a aparência e o *layout* de todas as páginas de um *site* sejam alteradas editando-se, com isso, um único arquivo (folhas de estilo externas);
- Garantir maior flexibilidade e controle nas especificações do *site*, entre outros.

Sintaxe CSS

O CSS utiliza uma estrutura de sintaxe simples, na qual a regra é a unidade básica de uma folha de estilo e é composta de duas partes principais: um seletor e uma ou mais declarações, como apresentado na Figura 1.

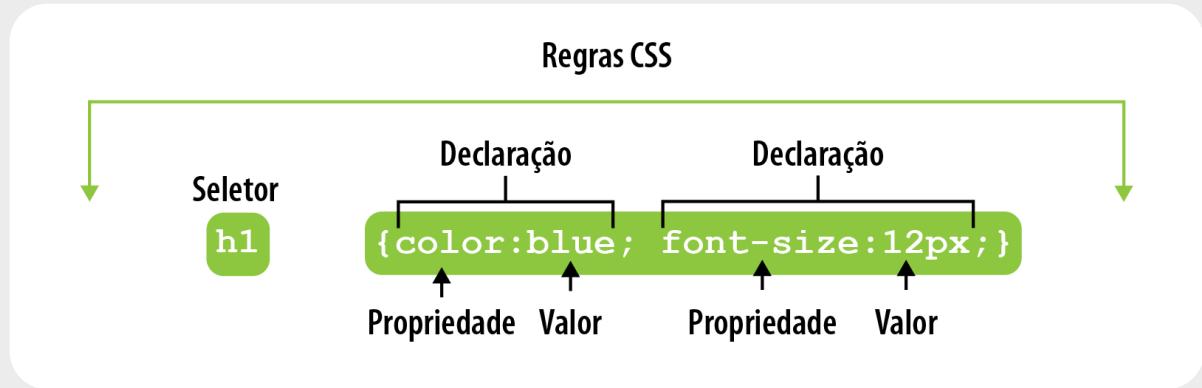


Figura 1 – Regra CSS

#ParaTodosVerem: imagem que descreve a utilização de uma Regra CSS. Da esquerda para a direita, são especificados o Seletor e as declarações. Na parte superior da imagem, há o título “Regra CSS”. A parte inferior da imagem contém textos destacando a Propriedade e o valor de cada uma das declarações. O seletor utilizado é o h1 e as declarações são “color: blue” e “font-size: 12px”. Fim da descrição.

Nessa definição dos componentes de uma regra CSS, temos:

- **Seletor:** é o elemento HTML que se deseja criar o estilo;
- **Declaração:** determina os parâmetros do estilo e consiste em uma **propriedade** e um **valor**;
- **Propriedade:** define a característica do seletor que será alterada;
- **Valor:** refere-se à quantificação ou qualificação da propriedade.

Uma regra pode conter várias declarações separadas por ponto e vírgula. Veja o exemplo a seguir:

```
h1{color: red; font-size: 36px}
```

Seletores Válidos

Os seletores são componentes HTML que desejamos modificar e estilizar, por meio do uso do CSS. Dentre os vários existentes, o Quadro 1 apresenta três dos principais seletores utilizados para criar estilos para um elemento do HTML.

Quadro 1 – Principais seletores CSS

Tipo de seletor	Descrição	Exemplo
Seletor de elemento	Seleciona um elemento do HTML.	p Seleciona apenas os elementos p da página.
Seletor de id	Usado para especificar um estilo para um elemento único e seleciona apenas um elemento que possui o valor atribuído id.	#main Seleciona apenas o elemento com o id "main" .
Seletor de classe	Usado para especificar um estilo para um grupo de elementos, permitindo criar um estilo para vários elementos do HTML com a mesma classe.	.main Seleciona apenas os elementos que possuem a classe "main" .

Vejamos, a seguir, um exemplo contendo o uso desses seletores:

```

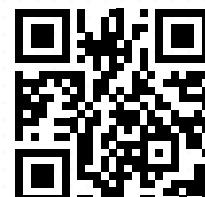
<html>
<head>
<style>
    #main {
        text-align: center;
        color: red;
    }
    .cor {color: blue;}
    p {font-size: 28px;}
    /*Efeito de sombra no texto*/
    h1{text-shadow: 5px 5px 5px #FF0000;}
    /*Efeito quebra de linha*/
    p {word-wrap:break-word;}
    @import url('https://fonts.googleapis.com/css2?family=Shan-
tell');
    p{font-family: 'Shantell Sans', cursive;}
    div{
        transform: rotate(20deg);
    }
</style>
</head>
<body>
    <h1 id="main">Texto alinhado ao centro na cor vermelha</
    h1>
    <h2 class="cor">Texto alinhado à esquerda na cor azul</h2>
    <p>Texto alinhado à esquerda no tamanho 28px</p>
</body>
</html>

```



Site

O CSS Validation Service é uma ferramenta in-
dispensável para verificar se o seu CSS é válido.



Vinculando Folhas de Estilo a Documentos

Não basta que você defina uma regra CSS. É preciso informar ao documento onde ela se encontra. As folhas de estilo podem ser utilizadas de três maneiras diferentes:

- Estilo externo;
- Estilo interno;
- Estilo *inline*.

O estilo externo é ideal quando a formatação deve ser aplicada a várias páginas. Com um arquivo externo, você pode alterar um formato e ele será aplicado automaticamente em todo o projeto.

Um estilo externo pode ser escrito por qualquer editor e não deve conter nenhuma tag HTML. O arquivo deve ser salvo com a extensão .css e pode ser vinculado a um documento HTML usando o elemento *link*, que deve estar contido na seção *head*.

Veja o exemplo a seguir:

```
<head>
    <link rel="stylesheet" type="text/css" href="estilo.css">
</head>
```

Lembre-se de que o atributo href deve apontar exatamente o endereço no qual se encontra o arquivo que contém a folha de estilo criada. Veja a seguir um exemplo de uma folha de estilo salva no arquivo **estilo.css**:

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

O estilo interno é usado na própria página HTML, sendo definido na seção *head* com o emprego do elemento *style*. As regras de estilo são definidas dentro das tags `<style></style>`, como mostra o exemplo a seguir:

```
<head>
    <style type="text/css">
        hr {color:sienna;}
        p {margin-left:20px;}
        body {background-image:url("images/back40.gif");}
    </style>
</head>
```

O estilo *inline* é o método mais simples e direto de aplicar formatação a um elemento HTML. Ele é usado diretamente no elemento que se quer alterar, dentro das tags de abertura e fechamento do elemento, por meio do emprego do atributo *style*.

Veja um exemplo a seguir:

```
<p style="color:sienna; margin-left:20px">
Este é um parágrafo formatado
</p>
```

Estilizando Documentos

Como vimos no começo desta Unidade, as propriedades são utilizadas para definir características que serão aplicadas aos elementos HTML. A seguir, veremos algumas dessas propriedades e como elas alteram o *layout* de uma página web.

Background

As propriedades do *background* são usadas para definir uma cor ou uma imagem de fundo e estão listadas a seguir:

- *Background-color;*
- *Background-image;*
- *Background-repeat;*
- *Background-attachment;*
- *Background-position.*

A propriedade *background-color* especifica a cor de fundo do elemento. Por exemplo:

```
body {background-color: #b0c4de;}
```

Já a propriedade *background-image* define uma ou mais imagens a serem usadas como fundo para o elemento.

Por exemplo:

```
body {background-image:url("paper.gif");}
```



Importante

Ao se utilizar a *background-image*, recomenda-se utilizar uma propriedade auxiliar, como a *background-color*, visto que pode ocorrer alguma falha no carregamento da imagem.

A propriedade *background-repeat* pode ser utilizada conjuntamente para repetição da imagem de fundo:

- **Repeat:** a imagem é repetida na vertical e na horizontal, até o preenchimento da área do elemento;
- **Repeat-x:** a imagem repetida somente na horizontal;
- **Repeat-y:** a imagem repetida somente na vertical;

- **No-repeat:** a imagem é inserida apenas uma vez no plano de fundo. Pode ser utilizada de modo que a imagem não atrapalhe o texto, como o exemplo da Figura 2:

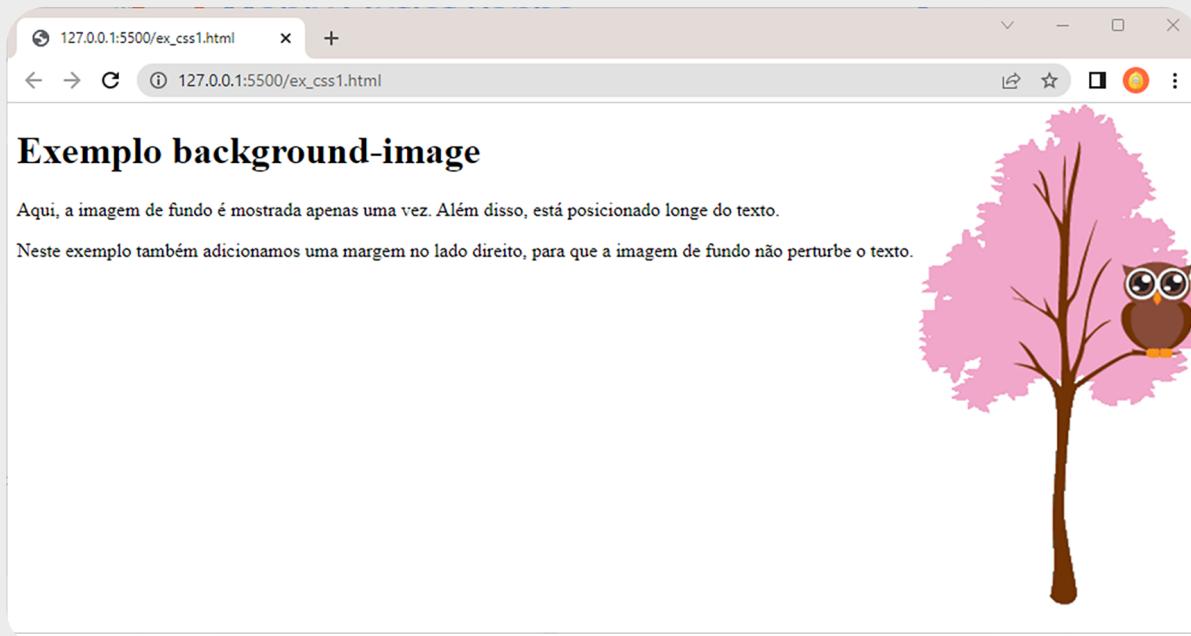


Figura 2 – Exemplo de uma página utilizando *background-image*

Fonte: Reprodução

#ParaTodosVerem: imagem de exemplo da utilização do *background-image*. Na parte superior, há um título escrito “Exemplo *background-image*”. Logo abaixo, dois parágrafos com textos que descrevem a estilização do conteúdo na página. À direita, há uma imagem de uma árvore nas cores rosa e marrom, sendo que em um dos galhos encontra-se uma coruja. Fim da descrição.

Nesse exemplo, a imagem de fundo é mostrada numa posição de modo que ela não perturbe o texto. O código CSS utilizado é mostrado a seguir:

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    margin-right: 200px;
}
```

Perceba que, para obtermos o efeito desejado, foi necessário utilizar uma nova propriedade, o *background-position* que indica a posição na qual a imagem de fundo será inserida.

Alguns dos valores possíveis dessa propriedade são:

- **Left top:** à esquerda superior;
- **Right center:** à direita e ao centro;
- **Center bottom:** ao centro inferior.

Fontes e Texto

Quando o desejamos formatar a exibição de textos, as propriedades CSS geralmente se enquadram em duas categorias:

- **Estilos de fontes:** afetam a fonte do texto, como, por exemplo, tamanho, negrito e itálico, entre outros;
- **Estilos de layout de texto:** afetam o espaçamento e outros recursos de layout do texto, como, por exemplo, o espaço entre linhas e o alinhamento do texto dentro de uma caixa de conteúdo.

Estilos de Fontes

A propriedade *font* define família , tamanho e estilo do texto:

- *Font-family*;
- *Font-style*;
- *Font-size*.

A família da fonte de um texto é definida com a propriedade *font-family*, cujo valor (nome da fonte) é uma *string* ou uma lista de *strings* separadas por vírgula. Procure sempre terminar a lista com uma fonte genérica, que se destina a uma possível falha nas escolhas do programador.

Veja o exemplo a seguir:

```
p{font-family: "Lucida Grande", Lucida, Verdana, Sans-serif;
```

A propriedade *font-style* tem 3 valores:

- **Normal:** a própria fonte;
- ***Italic:*** itálico;
- **Oblique:** muito similar ao itálico, porém, menos utilizado.

```
p.normal {font-style:normal;}  
p.italic {font-style:italic;}  
p.oblique {font-style:oblique;}
```

Já o tamanho do texto é definido pela propriedade *font-size*, que pode ter um valor absoluto ou relativo:

Valor Absoluto

- Define o texto a um tamanho especificado;
- Não permite que o usuário altere o tamanho do texto em todos os navegadores (ruim por razões de acessibilidade);
- O tamanho absoluto é útil quando o tamanho físico da saída é conhecido.

Valor Relativo

- Define o tamanho relativo aos elementos;
- Permite que o usuário altere o tamanho do texto em navegadores.
- A seguir, temos um exemplo que se utiliza de valores relativos e absolutos:

```
/*Tamanho absoluto*/  
h1 {font-size:40px;}  
h2 {font-size:30px;}  
p {font-size:14px;}  
/*Tamanho relativo*/  
h1 {font-size:2.5em;} /* 40px/16=2.5em */  
h2 {font-size:1.875em;} /* 30px/16=1.875em */  
p {font-size:0.875em;} /* 14px/16=0.875em */  
body {font-size:100%;}  
h1 {font-size:2.5em;}  
h2 {font-size:1.875em;}  
p {font-size:0.875em;}
```

Estilos de *Layout* de Texto

Completando a formatação de *layout* de texto, as propriedades que veremos são:

- **Color:** é utilizada para configurar a cor do texto:

```
body {color:blue;} h1 {color:#00ff00;}  
h2 {color:rgb(255,0,0);}
```

- **Text-align:** é usada para especificar o alinhamento do texto à esquerda, direita, centralizado ou justificado:

```
h1 {text-align:center;}  
p.date {text-align:right;}  
p.main {text-align:justify;}
```

- **Text-decoration:** é frequentemente utilizada nos links para configurar ou remover decorações do texto:

```
a {text-decoration:none;}  
h1 {text-decoration:overline;}  
h2 {text-decoration:line-through;}  
h3 {text-decoration:underline;}  
h4 {text-decoration:blink;}
```

- **Text-transform:** é usada para especificar o *uppercase* e *lowercase* nos textos:

```
p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}
```

- **Text-indentation:** é usada para definir um espaçamento horizontal antes do início da primeira linha do texto, em outras palavras, uma indentação da primeira linha ou parágrafo:

```
p {text-indent:50px;}
```

Importante

Quando utilizamos a marcação `p.main {text-align:justify;}` estamos justificando o texto do parágrafo (`<p>`) cuja classe é main.

Estilos de Borda de um Elemento

A propriedade *border* é responsável por adicionar contorno ao redor de seu conteúdo e pode ter diferentes estilos e tamanhos, podendo ser utilizada em diferentes elementos HTML, como, por exemplo, o `<input>`. As especificações do CSS permitem definir espessura, cor e estilo para a borda, além de ser possível especificar cada uma das quatro bordas individualmente.

A sintaxe resumida dessa propriedade é apresentada na Figura 3, a seguir.

```
{border: border-width border-style border-color;}
```

Define a espessura
da borda

Determina o estilo
da borda

Indica a cor
utilizada na borda

Figura 3 – Sintaxe da propriedade *border*

#ParaTodosVerem: imagem que descreve a sintaxe da propriedade *border*. Da esquerda para a direita é especificada a propriedade e suas definições: “*border: border-width border-style border-color*”. Na parte inferior da imagem, existem três quadros que descrevem cada um dos estilos aplicados à borda. Fim da descrição.

A declaração da borda pode ser feita resumidamente, como visto na Figura 5, porém, também é possível fazer a declaração individualmente de cada propriedade.

Os exemplos a seguir produzem o mesmo efeito:

```
.main{border: 5px solid red;}
```

Ou

```
.main{  
    border-width: 5px;  
    border-style: solid;  
    border-color: red;  
}
```

Margens e Espaçamentos

A propriedade *margin* permite adicionar ao redor do elemento um distanciamento na parte externa. A margem não tem cor de fundo e é completamente transparente. As margens superior, inferior, direita e esquerda podem ser alteradas de forma independente, usando suas propriedades separadas:

- **Margin-top:** determina o espaçamento superior;
- **Margin-right:** determina o espaçamento no lado direito;
- **Margin-bottom:** determina o espaçamento inferior;
- **Margin-left:** determina o espaçamento no lado esquerdo.

Para centralizar um elemento na tela ou dentro de outro elemento, podemos utilizar o valor auto, como mostra o exemplo a seguir:

```
<html>
<head>
    <title>CSS</title>
    <style>
        .box {
            width: 400px;
            border: 2px solid rgb(186, 184, 228);
        }
        .box p {
            width: 250px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="box">
        <p>Parágrafo centralizado dentro de um box de largura de 400px.</p>
    </div>
</body>
</html>
```

Observe, no exemplo, que, como queremos centralizar o parágrafo, a propriedade *margin*: auto foi aplicada somente a esse elemento. Poderíamos aplicar a mesma formatação ao box para centralizá-lo, como foi feito anteriormente.

Veja a seguir como ficaria essa formatação:

```
.box p {  
    width: 400px;  
    border: 2px solid rgb(186, 184, 228);  
    margin: 0 auto;  
}
```



Importante

A propriedade *margin* pode desestruturar o *layout* da página HTML, pois altera o posicionamento dos elementos. Portanto, use-a com cuidado!

Outra propriedade utilizada para definir o espaçamento ao redor de um elemento é o *padding*, que define o espaço interno entre um elemento e sua respectiva borda. Assim como na propriedade *margin*, também é possível determinar os espaçamentos superior, inferior, direito e esquerdo de forma independente, usando suas propriedades separadas.

Vamos alterar o exemplo anterior, para aplicar um *padding* de 20px ao parágrafo:

```
.box {  
    width: 250px;  
    margin: 0 auto;  
    padding: 20px;  
}
```

A Figura 4 mostra a comparação entre a formatação do parágrafo antes e depois da aplicação do *padding* de 20px ao parágrafo.

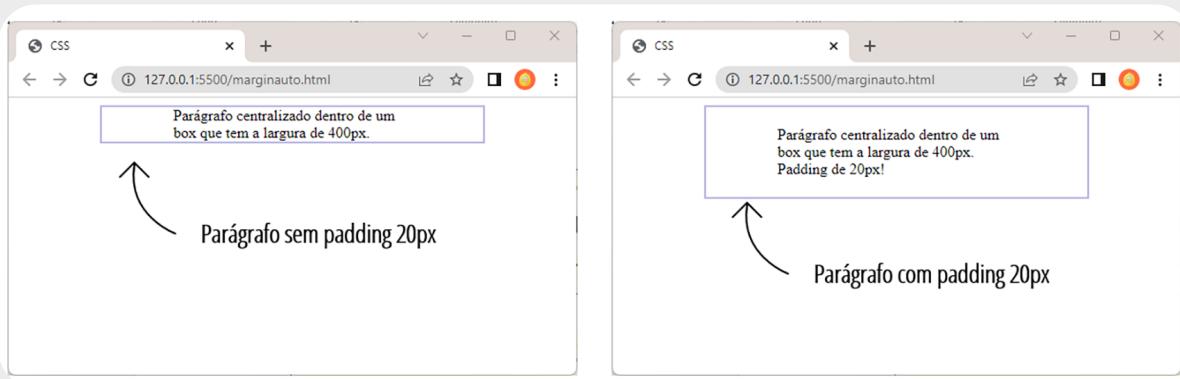


Figura 4 – Comparação entre os exemplos propostos

Fonte: Reprodução

#ParaTodosVerem: imagem que compara a utilização da formatação do parágrafo. À esquerda, há uma caixa contendo o texto “Parágrafo centralizado dentro de um box que tem largura de 400px.”. Logo abaixo, encontra-se a descrição “Parágrafo sem *padding* 20px” e uma seta aponta para a caixa que contém o texto. À direita, há uma caixa contendo o texto “Parágrafo centralizado dentro de um box que tem largura de 400px. *Padding* de 20px!”. Logo abaixo, encontra-se a descrição “Parágrafo com *padding* 20px” e uma seta aponta para a caixa que contém o texto. Fim da descrição.

Na Figura 5, temos a renderização de um elemento HTML, que exibe um determinado conteúdo, de acordo com a formatação CSS. Perceba a diferença entre *margin* e *padding*, bem como a função do *width* e do *height*, que determinam a largura e a altura do elemento, respectivamente.

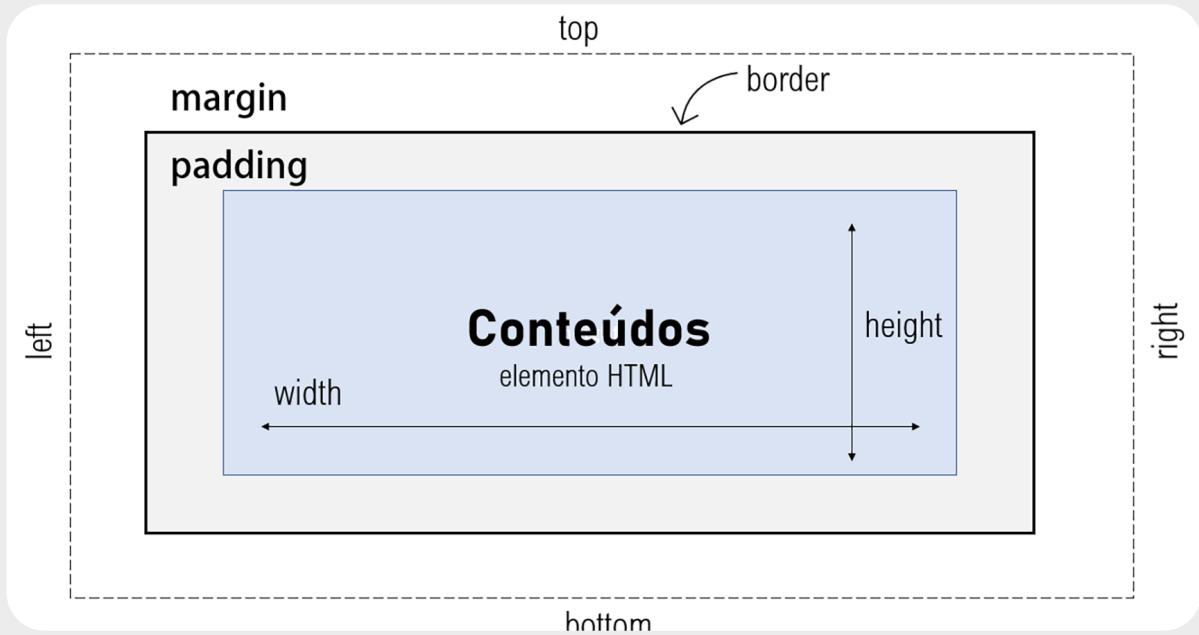


Figura 5 – Apresentação visual de um elemento de acordo com a formatação CSS

Fonte: Reprodução

#ParaTodosVerem: imagem ilustrativa de um elemento formatado com CSS. A Figura é composta por três retângulos de tamanhos diferentes, sendo o maior transparente, com borda tracejada, o seguinte na cor cinza com uma borda sólida na cor preta e, no centro, encontra-se o terceiro retângulo na cor azul com o texto “Conteúdos” e, abaixo, “elemento HTML”. No retângulo do centro, há duas setas: uma horizontal, indicando a largura do retângulo (*width*), e outra vertical, indicando a altura (*height*). No retângulo maior, com bordas tracejadas, há a indicação de cada um dos lados, contendo os textos, em sentido horário: “*right*”, “*bottom*”, “*left*” e “*top*”. Fim da descrição.

Módulos CSS3

A versão mais recente das folhas de estilo é o CSS3 que, diferentemente de suas versões anteriores, foi desenvolvido em módulos que introduziram novos recursos que mudaram amplamente a forma como as páginas web são vistas pelos usuários.

Vejamos, a seguir, alguns dos principais módulos e como utilizá-los.

Bordas e Sombras com CSS3

A partir do CSS3, podemos criar bordas arredondadas, adicionar sombras nas caixas e usar imagens como borda, sem a necessidade de um programa de edição.

O Quadro 2 apresenta as principais propriedades das bordas e suas compatibilidades com os navegadores:

Quadro 2 – Principais propriedades e compatibilidade com navegadores

Propriedade	Chrome	Explorer	Mozilla	Safari	Opera
border-radius	5.0 4.0 -webkit-	9.0	4.0 3.0 -moz-	5.0 3.1 -webkit-	10.5
box-shadow	10.0 4.0 -webkit-	9.0	4.0 3.5 -moz-	5.1 3.1 -webkit-	10.5



Importante

Na Tabela 1, os números seguidos por -webkit-, -moz- ou -o- especificam a primeira versão que funcionou com um prefixo.

A propriedade *border-radius* define um raio de curvatura nos quatro cantos de um *box* e permite adicionar o efeito de arredondamento ao elemento, independentemente do *box* ter ou não uma borda declarada.

Veja a Figura 6, a seguir:

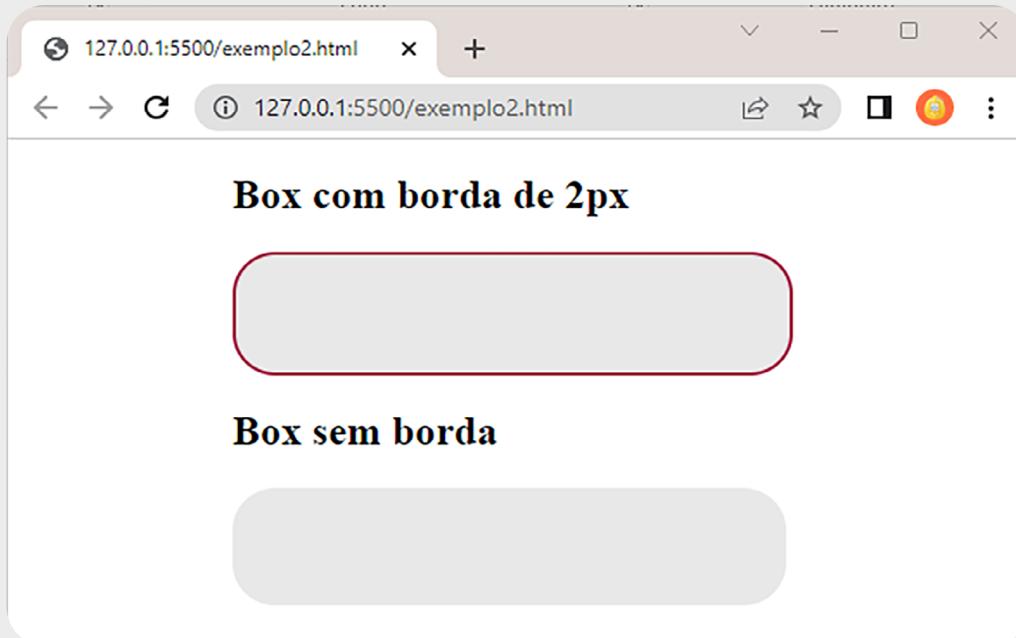


Figura 6 – Comparação entre o arredondamento com e sem a borda declarada.

Fonte: Reprodução

#ParaTodosVerem: imagem comparando o arredondamento com e sem bordas declaradas. A figura é composta por dois retângulos, sendo o superior com borda na cor vermelha e fundo cinza e o inferior sem bordas e com o fundo também cinza. Acima do primeiro retângulo, está o título “Box com borda de 2px” e acima do segundo retângulo está o título “Box sem borda”. Fim da descrição.

A seguir, temos o código utilizado para o box com borda declarada:

```
.comBorda {  
    border: 2px solid #991e1e;  
    padding: 10px 40px;  
    background: #ecebeb;  
    width: 300px;  
    border-radius: 25px;  
    -moz-border-radius: 25px;  
}
```

A propriedade *box-shadow* define uma ou mais sombras para um elemento:

```
.sombra {  
    border: 1px solid;  
    padding: 10px;  
    box-shadow: 5px 10px 8px #888888;  
}
```

Outras Formatações com CSS3

A partir do CSS3, a estilização de textos passou a contar com novas propriedades, entre as quais vamos destacar, nos exemplos a seguir, o *text-shadow* e *word-wrap*:

```
/*Efeito de sombra no texto*/  
h1{text-shadow: 5px 5px 5px #FF0000;}  
/*Efeito quebra de linha*/  
p {word-wrap: break-word;}
```

Como vimos anteriormente, a família da fonte de um texto é definida com a propriedade *font-family*. Entretanto, você pode querer utilizar uma fonte que não está instalada no computador do usuário. Isso é possível por meio da regra *@font-face*, que permite incluir uma fonte a partir da *Web*.

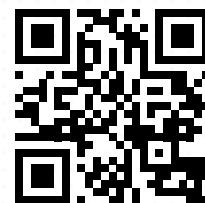
Veja o exemplo a seguir, no qual importamos a fonte “*Shantell Sans*”, a partir da biblioteca do *Google*, e utilizamos na formatação do parágrafo:

```
@import url('https://fonts.googleapis.com/css2?family=Shantell');  
p{font-family: 'Shantell Sans', cursive;}
```



Site

O *Google* tem uma Biblioteca para facilitar o uso de fontes remotas. Visite o *Google Fonts* e explore os tutoriais e funcionalidades disponíveis na ferramenta.



Com a propriedade CSS *transforms* é possível esticar, girar, aumentar e torcer um elemento HTML.

Os principais métodos que veremos são:

- **Rotate(x):** gira um elemento no sentido horário ou anti-horário, de acordo com um determinado ângulo (x);
- **Translate(x,y):** permite mover um elemento de uma posição para a outra, dependendo do parâmetros de posição passados dos eixos x e y;
- **Scale(x):** permite aumentar ou diminuir o tamanho de um elemento.

A seguir, temos um exemplo completo utilizando a propriedade *transform*.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Imagen transform</title>
    <style>
        .imagem {
            transform: scale(0.2) rotate(360deg);
            /*transition: all 2s*/;
        }
        .imagem:hover {
            transform: scale(0.7);
        }
    </style>
</head>
<body>
    
</body>
</html>
```

Se você digitou o código acima, pôde perceber que seu navegador exibe uma imagem (logo do CSS) e, quando passamos o *mouse* sobre ela, a imagem gira e aumenta de tamanho.

Você entendeu o que código faz?

Não!

Sem problema, pois, no Quadro 3, vamos analisar o que está acontecendo.

Quadro 3 – Exemplo comentado

Linha(s)	Descrição
Linha 7	Definimos o tamanho da imagem para 20% (scale(0.2)) e uma rotação de 360 graus (rotate(360deg)).
Linha 10	A pseudo-classe: hover corresponde ao usuário mover o cursor do <i>mouse</i> sobre a imagem.
Linha 11	O tamanho da imagem muda para 70% (scale(0.7)).
Linha 8	Adiciona um efeito de transição de 2s a todos os elementos (transition: all 2s).

Para finalizar esta Unidade, devemos detalhar a propriedade *transition*, que permite definir a transição entre dois estados de um elemento HTML. Experimente excluir a linha 11 do código acima e perceberá que a imagem aumenta de tamanho quando movemos o cursor do *mouse* sobre ela quase instantaneamente, sem o efeito de rotação!

Observe, nesse último exemplo, o efeito de transição aplicado para alterar o comprimento de uma <div>:

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    transition: width 2s;  
}  
div:hover {  
    width: 300px;  
}
```



Site

O CSS3 Generator é uma ferramenta que auxilia desenvolvedores iniciantes na geração de códigos CSS3. Dentre as opções disponíveis, é possível encontrar: *border-radius*, *box-shadow* e *text-shadow*, entre outras.



MATERIAL COMPLEMENTAR

Sites

W3school

É um site voltado à aprendizagem de conteúdos de tecnologias como HTML, CSS, Javascript e Python entre outros. Visite o tutorial de CSS.
<https://bit.ly/3Lh4Arj>

Code Adademy

É uma plataforma interativa *on-line* que oferece conteúdos em diversas Linguagens de Programação.
<https://bit.ly/3Pz12Dm>

Solearn

É uma plataforma *on-line* que disponibiliza material de diversas Linguagens de Programação.
<https://bit.ly/3R7J2RA>

30 Dias de CSS3

É uma plataforma que oferece desafios diários que visam a ajudá-lo(a) a melhorar as habilidades de codificação, utilizando HTML5 e CSS3.
<https://bit.ly/460CSa0>

REFERÊNCIAS BIBLIOGRÁFICAS

CLARK, R. *et al.* **Introdução ao HTML5 e CSS3:** a evolução da *web*. Rio de Janeiro: Alta Books, 2014.

SILVA, M. S. **Construindo sites com CSS e (X) HTML:** sites controlados por folhas de estilo em cascata. São Paulo: Novatec, 2008.

SILVA, M. S. **CSS3:** Desenvolva aplicações *web* profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2012.

TERUEL, E. C. **HTML 5:** guia prático. 2. ed. São Paulo: Erica, 2013.

W3C CSS. Disponível em: <<https://www.w3.org/community/webed/wiki/CSS/Training>>. Acesso em: 27/02/2023.