

Fundamentos de Programação Web

UNIDADE 02

Olá!

Continuando com o desenvolvimento *front-ent*, o lado *cliente* da aplicação *web*, veremos o **CSS**, ou *Cascading Style Sheets*, uma linguagem que descreve como os elementos **HTML** devem ser formatados para exibição.

Vamos começar a entender como usar o CSS.

Confira a seguir os pontos de destaque dessa linguagem.

- Não é linguagem de programação, pois não conseguimos realizar comandos condicionais (*if/else*), nem laços de repetição (*for/while*) ou nem funções, por exemplo.
- É uma linguagem **descritiva** para **estilizar páginas** *web*, visualizadas em **navegadores**.
- Folhas de estilo externas são armazenadas em arquivos com extensão .CSS, para serem aproveitadas em diversas páginas HTML, economizando o trabalho de formatação.

O CSS permite alterar completamente o visual de uma página web, apenas trocando o conjunto de formatação CSS.

Em resumo: HTML + CSS

- O HTML estrutura o *layout* de um documento *web*, ou seja, organiza os elementos visuais de um documento.
- O CSS formata a exibição dos elementos HTML.

Antes de ler o material, vamos conhecer melhor que resultados podemos esperar ao trabalhar com CSS. Veja alguns exemplos apresentados neste vídeo.

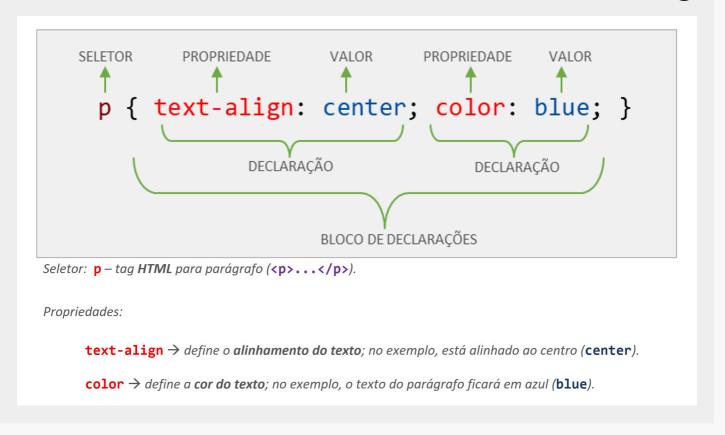


Regra CSS

Entendemos **CSS** por uma **linguagem descritiva** que permite a **formatação visual** de elementos de um documento **HTML**. Logo, **não é uma linguagem de programação**.

Basicamente, o **CSS** é composto por **regras** que indicam como formatar elementos **HTML**. É o **navegador** que interpreta as **regras** de associação entre um documento **HTML** e seu **CSS**, e as aplica ao documento, modificando sua aparência.

A composição de uma regra está exibida na Figura a seguir.



Fonte: Autor, 2023.

Sintaxe da regra CSS

- **SELETOR:** indica o <u>elemento HTML</u> que recebe o estilo definido no **bloco de declarações**.
- BLOCO DE DECLARAÇÃO: delimitado por chaves { e }, mantém declarações separadas por ;, onde temos: uma PROPRIEDADE CSS e seu VALOR, separados por :.

Tipos de CSS

As regras de estilo CSS podem ser definidas de três formas:

- 1. *Inline*: as regras são definidas dentro de uma tag HTML.
- 2. Interna: as regras são definidas no cabeçalho do documento HTML.
- 3. **Externa**: as regras são definidas em um documento separado, externo a todas as páginas HTML.

Vamos praticar cada uma dessas formas de definir estilo para os elementos HTML.

1. Sintaxe: Regra CSS inline

O código CSS é adicionado diretamente no elemento HTML por meio do atributo style:

```
texto.
```

Evite usar esta prática! Ela mistura o código HTML com CSS e não permite o reuso do CSS, tornando a manutenção e evolução do código mais complexa.



EXPERIMENTE

PRÁTICA: visualizando o CSS inline em um documento web

Vamos criar um documento **HTML** formatado com folha de estilo **CSS**, para visualização em um navegador. Também vamos utilizar a **ferramenta de inspeção** de página do **navegador Google Chrome**.

Siga os passos indicados a seguir para ver o CSS inline funcionando.

- 1. Defina uma pasta de trabalho, como por exemplo C:/Teste-CSS.
- 2. Dentro da pasta, crie um documento **estiloInline.html**¹, com o código exemplificado a seguir.

Arquivo C:/Teste-CSS/ estiloInline.html

```
<!DOCTYPE html>
1
     <html>
2
    <head>
3
         <meta charset="UTF-8">
4
         <title>Página HTML com CSS</title>
5
    </head>
6
     <body>
7
         <h1 style="color:rgb(17, 128, 35)">CSS inline</h1>
8
             style="color:deepskyblue; font-family:Verdana; text-align:center">
9
             style="color:brown; font-family:'Courier New'; text-align:right">T
10
     </body>
11
     </html>
12
```

¹Você pode utilizar um ambiente integrado de desenvolvimento, ou **IDE** – do inglês *Integrated Development Environment* –, como **o Visual Studio Code** da Microsoft (<u>https://code.visualstudio.com/download</u>).

3. Abra o seu **index.html** no navegador (*browser*): copie o caminho completo do arquivo e cole da barra de endereços.

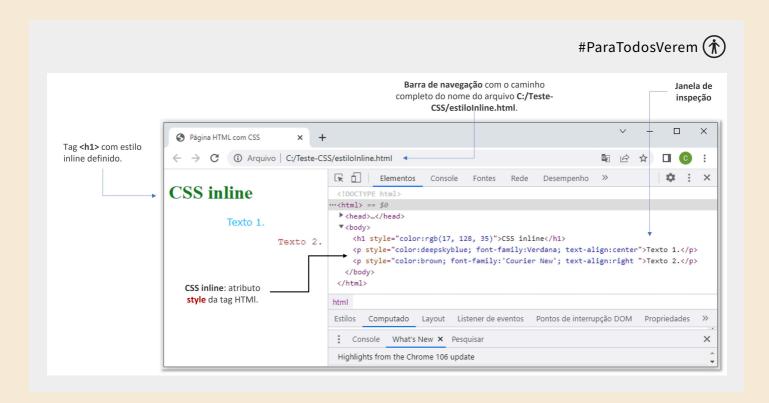


Figura 2: HTML com CSS *inline*. Fonte: A autora (2023).

2. Sintaxe: Regra CSS interno

O código CSS é inserido em um elemento <style>, que geralmente fica dentro da seção de <head>:

</>>

```
<head>
1
        <title>Página HTML com CSS</title>
2
        <style>
3
             p {
4
                 color: blue;
5
                 font-family: Verdana;
6
7
        </style>
8
    </head>
9
```

Também evite usar esta prática! Ela deixa no mesmo arquivo o código HTML e o código CSS, o que não permite o reuso do CSS, podendo gerar duplicidade de código e tornar a manutenção e evolução do código mais complexa.



EXPERIMENTE

PRÁTICA: visualizando o CSS interno em um documento web

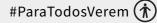
Vamos criar um documento **HTML** formatado com folha de estilo **CSS**, para visualização em um navegador.

Siga os passos indicados a seguir para ver o **CSS interno** funcionando.

- 1. Na pasta de trabalho, como, por exemplo C:/Teste-CSS.
- 2. Dentro da pasta, crie um documento **estiloInterno.html**, com o código exemplificado a seguir.

```
Arquivo C:/Teste-CSS/ estiloInterno .htmL
```

```
<!DOCTYPE html>
1
     <html>
2
     <head>
3
         <meta charset="UTF-8">
4
         <title>Página HTML com CSS</title>
5
         <style>
6
             p {
7
                 color:deepskyblue;
8
                 font-family:Verdana;
9
                 text-align:center;
10
             }
11
             h1{
12
                 color:rgb(17, 128, 35);
13
14
             }
15
         </style>
16
     </head>
17
     <body>
18
         <h1>CSS inline</h1>
19
         Texto 1.
20
         Texto 2.
21
     </body>
22
     </html>
23
```



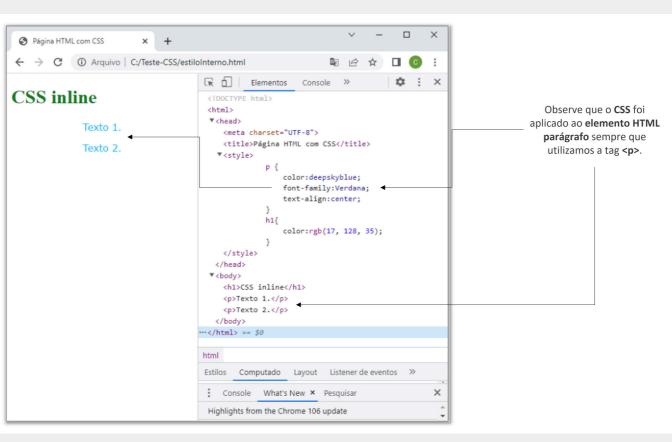


Figura 3: HTML com CSS interno. Fonte: A autora (2023).

3. Sintaxe da regra CSS externo

O código CSS é colocado em um arquivo .CSS, separadamente do arquivo .HTML.

Neste caso, precisamos ligar o arquivo .CSS ao arquivo .HTML, o que é realizado por um *link* adicionado ao elemento **<head>**:

Esta é uma boa prática! Ela separa os arquivos de HTML e CSS, permitindo o reuso do CSS, evitando duplicidade de código, o que tornar a manutenção e evolução do código menos complexa.



EXPERIMENTE

PRÁTICA: visualizando o CSS externo em um documento web

Vamos criar um documento **HTML** formatado com folha de estilo **CSS**, para visualização em um navegador.

Siga os passos indicados a seguir para ver o CSS interno funcionando.

- 1. Na pasta de trabalho, como por exemplo C:/Teste-CSS.
- Dentro da pasta, crie um documento estiloExterno.html, com o código exemplificado a seguir.
- 3. Crie uma subpasta css, como, por exemplo C:/Teste-CSS/css.

4. Crie os documentos **estiloExterno.html** e **css/style.css**, com os códigos a seguir.

```
Arquivo C:/Teste-CSS/ estiloExterno .html
     <!DOCTYPE html>
 1
     <html>
 2
     <head>
 3
         <meta charset="UTF-8">
 4
         <title>Página HTML com CSS</title>
 5
         <link href="css/style.css" rel="stylesheet" type="text/css "/>
 6
     </head>
 7
     <body>
 8
         <h1>CSS inline</h1>
 9
         Texto 1.
10
         Texto 2.
11
     </body>
12
     </html>
13
```

```
Arquivo C:/Teste-CSS/css/style.css
     p {
 1
         color:deepskyblue;
 2
         font-family:Verdana;
 3
         text-align:center;
 4
     }
 5
 6
     h1{
 7
         color:rgb(17, 128, 35);
 8
     }
 9
```

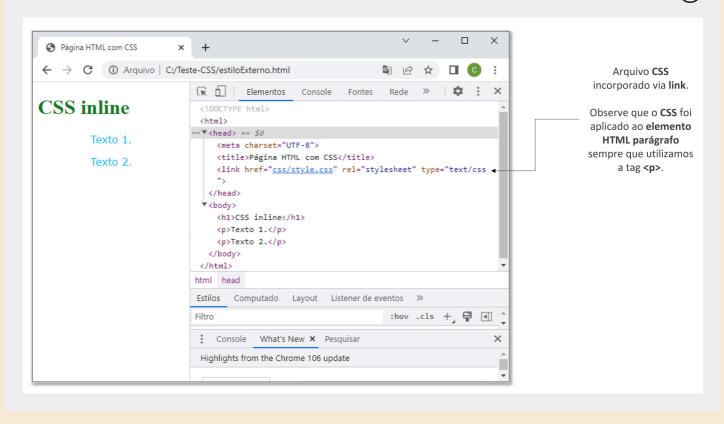


Figura 4: HTML com CSS externo. Fonte: A autora (2023).

Seletores CSS

Os **seletores CSS** são usados para "encontrar" (ou selecionar) os elementos HTML que desejamos estilizar.

Podemos organizar os seletores CSS nas seguintes categorias:

- 1. Seletor de elemento.
- 2. Seletor de id.
- 3. Seletor de classe.
- 4. Seletores de combinação (agrupa elementos).
- 5. Seletores de pseudoclasse (seleciona elementos com base em um determinado estado).

Vamos conhecer melhor cada um deles.

+ Seletor de elemento CSS

Seleciona que elemento HTML receberá o estilo, baseado no seu nome HTML, ou *tag*.

Exemplos de seletores:

- Seletor p serve para formatar um elemento HTML para parágrafo, da tag
- Seletor * serve para formatar elementos HTML de qualquer tipo

PRÁTICA: seletor de elemento CSS no navegador

```
Estilo no elemento da tag  .. 
    <!DOCTYPE html>
 1
    <html>
 2
       <head>
 3
          <style>
 4
             * { /* formatação para qqr elemento HTML */
 5
             font-family: Cambria, serif;
 6
             color: darkolivegreen;
 7
 8
             p { /* formatação para elemento HTML
 9
             >
10
                */
11
                text-align: center;
12
                color: red;
13
14
          </style>
15
       </head>
16
       <body>
17
       <h2>H2 é afetado pelo seletor *</h2>
18
       <h3>H3 também é afetado pelo seletor *</h3>
19
       Todo parágrafo é afetado por estilo próprio. 
20
       Eu também! 
21
       E eu idem! 
22
       </body>
23
    </html>
24
```

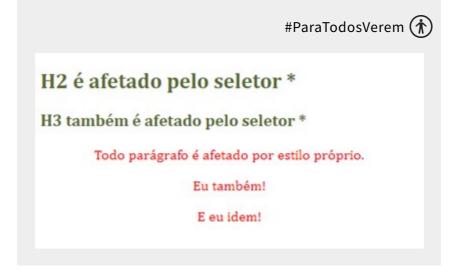


Figura 5: Seletor de elemento CSS. Fonte: A autora (2023).

+ Seletor de id CSS

O seletor usa o atributo **id** (de identificador) de um elemento HTML para selecionar um elemento específico.

O **id** de um elemento é **único** dentro de uma página HTML, logo o seletor de id é aplicado em um elemento único elemento da página.

USO CORRETO: escreva um caractere hash (#), seguido do id do elemento.

PRÁTICA: seletor de id CSS no navegador

Estilo no id #param1	l.		

```
<html>
1
       <head>
2
          <style>
3
             #para1 {
4
             text-align: center;
5
             color: red;
6
             }
7
          </style>
8
       </head>
9
       <body>
10
          Olá mundo!
11
          Este parágrafo não é afetado pelo estilo.
12
       </body>
13
    </html>
14
```

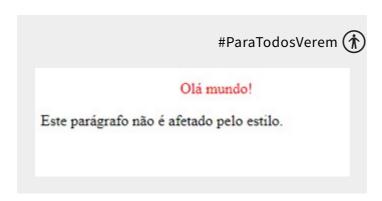


Figura 6: Seletor de id CSS. Fonte: A autora (2023).

+ Seletor de classe CSS

O seletor de classe seleciona elementos HTML com um atributo de *class* específico. Para selecionar elementos de uma classe, escrevemos um caractere ponto (.), seguido do **nome da classe**.

PRÁTICA: seletor de classe CSS no navegador

Estilo na class="center"		

```
<!DOCTYPE html>
1
    <html>
2
       <head>
3
          <style>
4
             .center {
 5
             text-align: center;
6
             color: red;
7
             }
8
          </style>
9
       </head>
10
       <body>
11
          <h1 class="center"> Cabeçalho vermelho e centralizado
12
           Parágrafo vermelho e centralizado.
13
       </body>
14
    </html>
15
```



Figura 7: Seletor de classe CSS. Fonte: A autora (2023).

Elemento	s HTML	também	podem	se	referir	а	mais	de	uma	classe.

```
<!DOCTYPE html>
1
    <html>
2
       <head>
3
          <style>
4
            p.center {
5
            text-align: center;
6
             color: red;
7
8
             p.large {
9
             font-size: 300%;
10
             }
11
          </style>
12
       </head>
13
       <body>
14
          <h1 class="center">Este cabeçalho não será afetado</k
          Este parágrafo será vermelho e cent
16
          Este parágrafo será vermelho;
17
       </body>
18
    </html>
19
```



Figura 7: Seletor de classe CSS. Fonte: A autora (2023).

+ Seletor de pseudoclasse CSS

Uma **pseudoclasse** é usada para definir um **estado** especial de um elemento. Pode ser usado para estilizar um elemento quando um usuário **passa o mouse** sobre ele, ou quando é um *link* **visitado**, ou quando o elemento estiver **em foco**, por exemplo.

Sintaxe para pseudo-classe:

selector:pseudo-class { property: value; }

PRÁTICA: seletor de pseudoclasse CSS no navegador

	Estilo na pseudoclasse
I	

```
<!DOCTYPE html>
1
    <html>
2
       <head>
3
          <style>
4
              a:link { /* pseudo-classe p/ link NÃO VISITADO
5
              color: red;
6
              }
7
              a:visited { /* pseudo-classe p/ link VISITADO*/
8
             color: green;
9
              }
10
              a:hover { /* pseudo-classe p/ MOUSE SOBRE o link
11
             color: hotpink;
12
              }
13
             a:active { /* pseudo-classe p/ link SELECIONADO'
14
             color: blue;
15
              }
16
          </style>
17
       </head>
18
       <body>
19
          <h2>Estilo para link, dependendo do seu estado</h2>
20
          <b><a href="default.asp" target="_blank">Este é ur
21
          <dl>
22
             <dt>
23
                 <b>Importante 1:</b>
24
             </dt>
25
              <dd><b>a:hover</b> DEVE vir depois de <b>a:link</t
26
             <dt>
27
                 <b>Importante 2:</b>
28
             </dt>
29
              <dd><b>a:active</b> DEVE vir depois de <b>a:hover
30
31
          </dl>
       </body>
32
    </html>
33
```

Estilo para link, dependendo do seu estado

Este é um link

Importante 1:

a:hover DEVE vir depois de a:link e a:visited, para ser efetivo no CSS.

Importante 2:

a:active DEVE vir depois de a:hover, para ser efetivo no CSS.

Estilo para link, dependendo do seu estado

Este é um link

Importante 1:

a:hover DEVE vir depois de a:link e a:visited, para ser efetivo no CSS.

Importante 2:

a:active DEVE vir depois de a:hover, para ser efetivo no CSS.

Estilo para link, dependendo do seu estado

Este é um link

Importante 1:

a:hover DEVE vir depois de a:link e a:visited, para ser efetivo no CSS.

Importante 2:

a:active DEVE vir depois de a:hover, para ser efetivo no CSS.

Figura 8: Seletor de agrupamento CSS. Fonte: A autora (2023).

+ Seletor de agrupamento CSS

Para minimizar o código, podemos agrupar os seletores, separando cada seletor com uma vírgula.

PRÁTICA: seletor de agrupamento CSS no navegador

```
Estilo no agrupamento de elementos
     <!DOCTYPE html>
 1
     <html>
 2
        <head>
 3
           <style>
 4
              h1, h2, p {
 5
              text-align: center;
 6
              color: red;
 7
              }
 8
           </style>
 9
        </head>
10
        <body>
11
           <h1>01á mundo! </h1>
12
           <h2>Cabeçalho menor! </h2>
13
           Este é um parágrafo. 
14
        </body>
15
     </html>
16
```

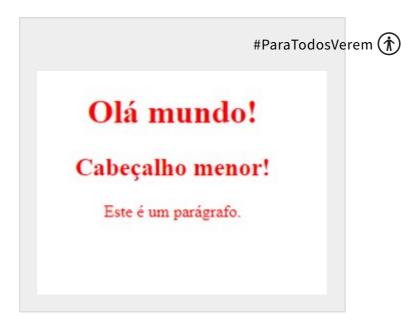


Figura 9: Seletor de agrupamento CSS. Fonte: A autora (2023).

CSS para a "box" (caixa) de elemento HTML

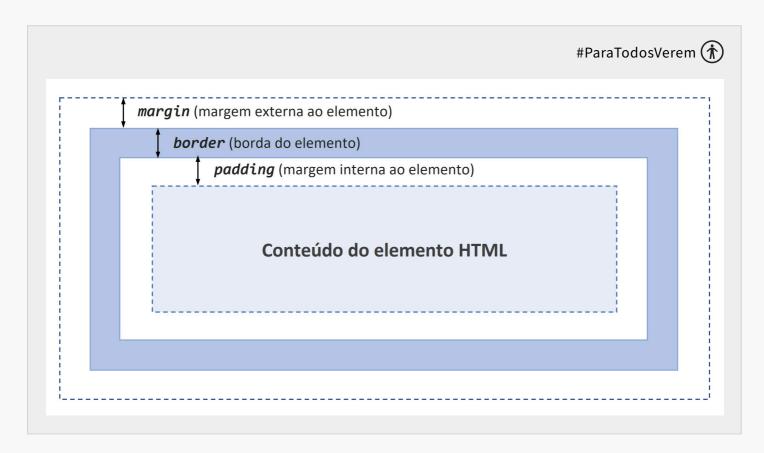
Quase todos os elementos HTML podem ser compreendidos como uma *box* (caixa), permitindo agrupamentos horizontal ou vertical, ou ainda inserida uma dentro da outra.

As **propriedades CSS** evidenciam essa compreensão, pois você precisará indicar a dimensão dos elementos **HTML**, como **largura** e **altura**, ou sua **cor**, **posição relativa** aos demais elementos, ou ainda **posição do seu conteúdo interno**, e assim por diante.

A **box** que representa um **elemento HTML** tem algumas propriedades que definem o espaço que ocupam na página web, como relacionado abaixo.

- height: altura do elemento; é definida em pixel (ex: 50px) ou em porcentagem (ex: 30%).
- width: largura do elemento; é definida em pixel (ex: 80px) ou em porcentagem (ex: 50%).
- border: linha que envolve o elemento.
- margin: espaço externo ao elemento.
- padding: espaço interno ao elemento, entre a borda e o conteúdo.
- background-color: cor de fundo do elemento; é definida pelo nome em inglês da cor, ou no padrão hexadecimal RGB (*Red-Green-Blue*), como por exemplo #3040AF.

Veja como é compreendida a *box* de um elemento HTML, exibida na Figura a seguir.





PRÁTICA: elemento HTML como uma box

Tag HTML <div> e Tag HTML

As tags <div> e definem uma divisão ou parte de um documento HTML. Qualquer tipo de conteúdo (elementos HTML) pode ser colocado dentro delas!

Ambas são usadas como um contêiner para elementos HTML, e podem ser:

- Estilizados com CSS usando o atributo class ou id; ou
- Manipulados com JavaScript.

A diferença entre elas é que:

- A tag <div> faz o agrupamento de elementos a nível de bloco.
- A tag faz o agrupamento de elementos a nível de linha.

Ou seja, se você tem algum conteúdo em texto que precise ser agrupado e/ou destacado, você deve usar a *tag* .

Veja no exemplo da Figura a seguir, em que a *tag* <div> é usada para estilizar elementos na página HTML.

Estilo de classe na tag <di< th=""><th>v></th><th></th></di<>	v>	

```
<!DOCTYPE html>
1
    <html>
2
        <head>
3
           <style>
4
              .texto1 {
5
              font-family: Arial, Helvetica, sans-serif;
6
              background-color: LightSeaGreen;
7
              color: white;
8
              border: 2px solid black;
9
              margin: 20px;
10
              padding: 20px;
11
              }
12
              .texto2 {
13
              font-family: "Lucida Console", "Courier New", monospace;
14
              background-color: Chocolate;
15
              color: white;
16
              border: 4px solid black;
17
              margin: 40px;
18
              padding: 40px; /* um valor para toda a borda */
19
              }
20
           </style>
21
        </head>
22
        <body>
23
           <div class="texto1">
24
              <h2>Box 1</h2>
25
              Verde marinho claro, borda 2px, margem 20px e padding 20px.
26
           </div>
27
           <div class="texto2">
28
              <h2>Box 2</h2>
29
              Chocolate, borda 4px, margem 40px e padding 40px.
30
31
           </div>
        </body>
32
     </html>
33
```

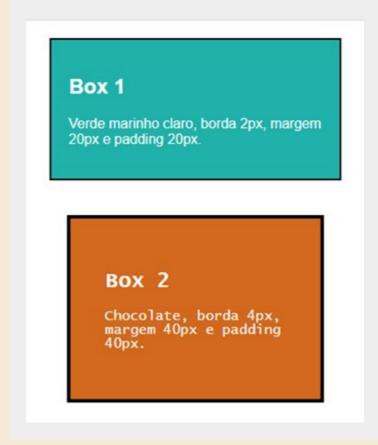


Figura 11: Box de elemento HTML, exemplificado com a tag <div>. Fonte: A autora (2023).

As propriedades padding, margin e border são aplicadas aos quatro lados de um elemento HTML: superior, direita, inferior e esquerdo.

No exemplo da Figura anterior, essas propriedades receberam apenas um valor, que foi aplicado a todos os lados do elemento:

```
</>>
      .texto2 {
  1
        font-family: "Lucida Console", "Courier New", monospace;
  2
        background-color: Chocolate;
  3
       color: white;
  4
       border: 4px solid black; /* apenas um valor para border */
  5
                                 /* apenas um valor para margin */
       margin: 40px;
  6
                                 /* apenas um valor para padding */
        padding: 40px;
 7
  8
  9
      }
```

Contudo, podemos especificar um valor para cada lado do elemento. Detalhamos esse tratamento para *padding*, *margin* a seguir.

Propriedade padding

Podemos atribuir valores diferente para cada lado da *box* do elemento HTML, para o *padding*. Veja os exemplos na tabela e figura a seguir.

Propriedade padding	
Exemplo	Descrição
div { padding: 25px 50px 75px 100px; }	padding superior é de 25px padding direito é de 50px padding inferior é de 75px padding esquerdo é de 100px
div { padding: 25px 50px 75px; }	padding superior é de 25px padding direito e esquerdo são de 50px padding inferior é de 75px
div { padding: 25px 50px; }	padding superior e inferior são de 25px padding direito e esquerdo são de 50px
div { padding: 25px; }	padding de todos os lados são de 25px

egador		

```
<!DOCTYPE html>
1
     <html>
2
        <head>
3
           <style>
4
              div {
5
              border: 1px solid black;
6
              padding: 25px 50px 75px 100px;
7
              background-color: lightgreen;
8
               }
9
           </style>
10
        </head>
11
        <body>
12
           <h2>Propriedade Padding</h2>
13
           <div>O elemento div tem padding superior = 25px, padding direito =
14
        </body>
15
     </html>
16
```

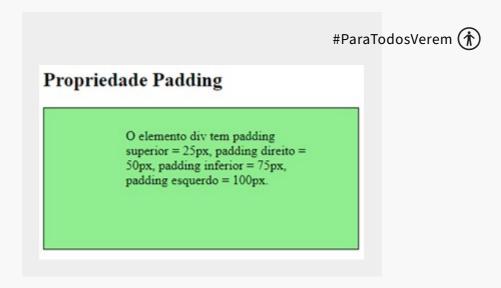


Figura 12: Exemplos de utilização da propriedade padding. Fonte: A autora (2023).

Uma alternativa para configurar o *padding* é com suas propriedades específicas para cada lado, que podem ser utilizadas em conjunto ou individualmente.

- padding-top
- padding-right
- padding-bottom
- padding-left

Propriedade margin

Podemos atribuir valores diferente para cada lado da Box do elemento HTML, para a *margin*. Veja os exemplos nas Figuras a seguir.

Propriedade margin	
Exemplo	Descrição
div { margin: 25px 50px 75px 100px; }	margin superior é de 25px margin direito é de 50px margin inferior é de 75px margin esquerdo é de 100px
div { margin: 25px 50px 75px; }	margin superior é de 25px margin direito e esquerdo são de 50px margin inferior é de 75px
div { margin: 25px 50px; }	margin superior e inferior são de 25px margin direito e esquerdo são de 50px
div { margin: 25px; }	margin de todos os lados são de 25px

```
Propriedade margin no navegador
     border: 1px solid black;
 1
     margin: 25px 50px 75px;
 2
     background-color: lightgreen;
 3
     }
 4
     </style>
 5
     </head>
 6
     <body>
 7
        <h2>Propriedade Margin</h2>
 8
        <div>O elemento div tem margin superior = 25px, margin direito e esquer
 9
        <hr>>
10
     </body>
11
     </html>
12
```

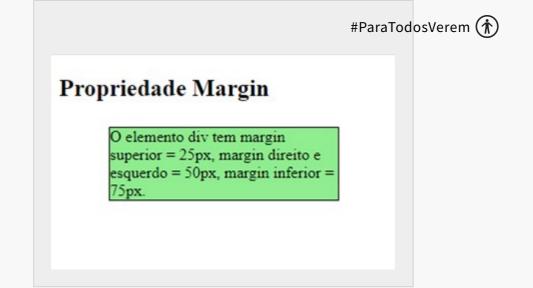


Figura 13: Exemplos de utilização da propriedade margin. Fonte: A autora (2023).

Uma alternativa para configurar o margin é com suas propriedades específicas para cada lado, que podem ser utilizadas em conjunto ou individualmente.

- margin-top
- margin-right
- margin-bottom
- margin-left

CSS para texto

Existem várias propriedades CSS para formatar um texto. Veja alguns exemplos na imagem a seguir.

Formatando texto	
Exemplo	Descrição
h1 { background-color: black; color: white; }	background-color: cor de fundo do elemento color: cor do texto
h2 { text-align: left; }	text-align: alinhamento horizontal de um texto. Pode assumir os valores: center, right, left ou justify.

```
h3 {
                                                            text-decoration: adiciona uma linha de decoração
text-decoration: underline;
                                                            ao texto. Pode assumir os valores: overline, line-
}
                                                            through ou underline.
                                                            text-transform: especifica letras maiúsculas e
p {
text-transform: uppercase;
                                                            minúsculas em um texto. Pode assumir os valores:
                                                            uppercase, lowercase ou capitalize.
}
                                                            font-family: especificar a fonte de um texto. Pode
p {
                                                            assumir os valores: Arial, Helvetica, "Lucida
font-family: Arial, Helvetica, sans-serif;
                                                            Console"....
}
                                                            Deve conter várias fontes para garantir
                                                            compatibilidade entre diferentes navegadores.
```

```
Formatando texto no navegador
```

```
<!DOCTYPE html>
1
    <html>
2
       <head>
3
          <style>
4
              body {
5
             background-color: lightgrey;
6
             color: blue;
7
              }
8
              .p1 {
9
              font-family: "Lucida Console", "Courier New", monospace;
10
             text-transform: uppercase;
11
             text-decoration: overline underline;
12
              background-color: lightgreen;
13
             color: black;
14
              }
15
          </style>
16
       </head>
17
        <body>
18
          <h1>CSS Formatação de Texto</h1>
19
          Parágrafo formatado. 
20
          Parágrafo sem formatação própria
21
        </body>
22
    </html>
23
```

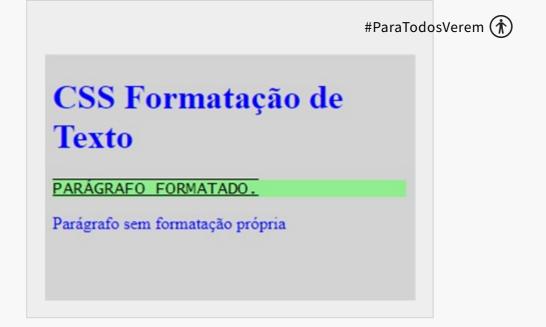
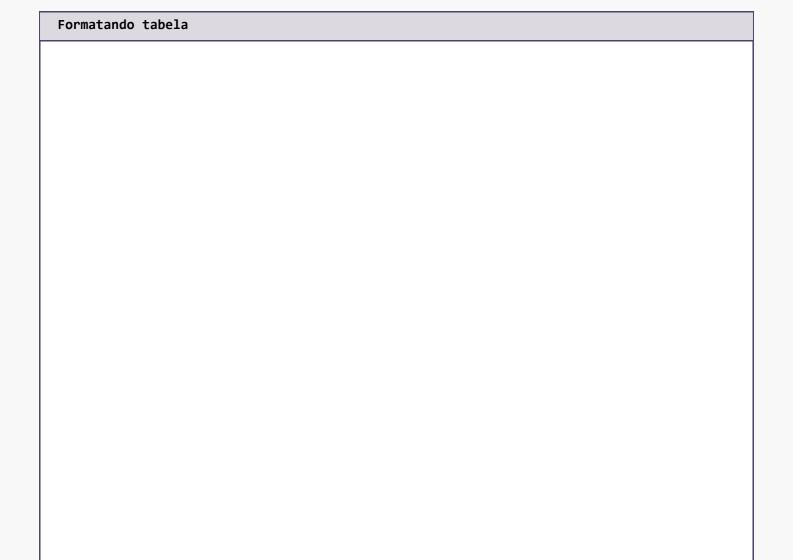


Figura 14: Exemplos de utilização da formatação para texto. Fonte: A autora (2023).

CSS para tabela

Existem várias propriedades CSS para formatar uma tabela. Veja alguns exemplos na imagem a seguir.



```
<!DOCTYPE html>
1
    <html>
2
    <head>
3
   <style>
4
    #clientes {
5
      font-family: Arial, Helvetica;
6
      border-collapse: collapse;
7
      width: 100%;
8
    }
9
    #clientes td, #clientes th {
10
      border: 1px solid #ddd;
11
      padding: 8px;
12
    }
13
14
    #clientes tr:nth-child(even){
15
      background-color: #e0ebeb;
16
    }
17
    #clientes tr:hover {
18
      background-color: #ddd;
19
    }
20
    #clientes th {
21
      padding-top: 12px;
22
      padding-bottom: 12px;
23
      text-align: center;
24
      background-color: #3d5c5c;
25
      color: white;
26
    }
27
    </style>
28
    </head>
29
    <body>
30
31
    <h1>Tabela Personalizada</h1>
    32
      >
33
       Nome
34
       País
35
      36
      37
       Maria Flores
38
       Brasil
39
      40
      41
       Caroline Summers
42
       EUA
43
      44
      45
       Francisco Lopez
46
```

```
47
     Argentina
48
    49
    >
50
     Arthur Wood
51
     Ingalterra
52
    53
54
   </body>
55
   </html>
```

Observe alguns pontos importantes nesta formatação:

- 1. O identificador id="clientes" identifica a tabela que queremos formatar.
- 2. Os seletores do identificador (#clientes td, #clientes tr:hover, ...) podem receber formatações exclusivas.
- 3. As linhas pares (#clientes tr:nth-child(even)) também recebem formatação exclusiva.
- 4. As demais *tags* de tabela (, , etc.) não têm propriedade interna formatada. Assim, podemos mudar totalmente a personalização da tabela, apenas editando o CSS.



Figura 15: Exemplos de formatação de tabela. Fonte: A autora (2023).

Existem várias propriedades CSS para formatar um formulário. Veja alguns exemplos na imagem a
seguir.
Inserir título aqui

```
<!DOCTYPE html>
1
     <html>
2
     <style>
3
     input[type=text], select {
4
       width: 100%;
5
       padding: 12px 20px;
6
       margin: 8px 0;
7
       display: inline-block;
8
       border: 1px solid #ccc;
9
       border-radius: 4px;
10
       box-sizing: border-box;
11
12
     input[type=submit] {
13
       width: 100%;
14
       background-color: #3d5c5c;
15
       color: white;
16
       padding: 14px 20px;
17
       margin: 8px 0;
18
       border: none;
19
       border-radius: 4px;
20
       cursor: pointer;
21
     }
22
     input[type=submit]:hover {
23
       background-color: #73978C;
24
25
     }
     div {
26
       border-radius: 5px;
27
       background-color: #f2f2f2;
28
       font-family: Arial, Helvetica;
29
       padding: 20px;
30
     }
31
     </style>
32
     <body>
33
     <h3>Usando CSS em Formulário</h3>
34
     <div>
35
       <form action="action page.php">
36
         <label for="nome">Nome</label>
37
         <input type="text" id="nome" name="nome" placeholder="Seu nome..">
38
39
         <label for="snome">Sobrenome</label>
40
         <input type="text" id="snome" name="snome" placeholder="Seu sobrenome.</pre>
41
42
         <label for="pais">País</label>
43
         <select id="pais" name="pais">
44
           <option value="brasil">Brasil</option>
45
           <option value="eua">EUA</option>
46
```

```
47 </select>
48 <br>
49 <input type="submit" value="Enviar">
50 </form>
51 </div>
52 </body>
53 </html>
```

Observe que o CSS foi aplicado ao seletor input e seus tipos e pseudo-classes.

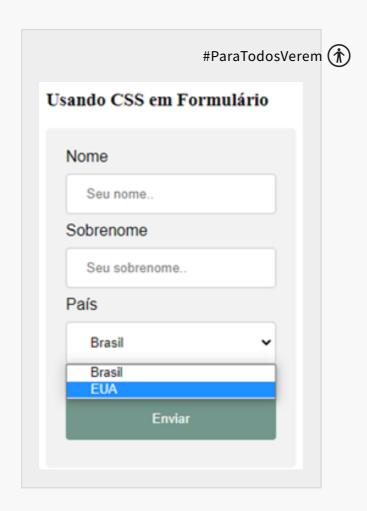


Figura 16: Exemplos de formatação de formulário. Fonte: A autora (2023).

Layout flexível: basta alterar o CSS

Vamos verificar na prática como facilitar a alteração de *layout* de uma página **HTML**. Para isso, basta alterar apenas o **CSS**, portanto, o **HTML** deve ser utilizado apenas para estruturar os elementos no documento. Toda formatação será feita com **CSS externo** – também para facilitar a manutenção e reutilização de código –, de modo a estilizar os elementos **HTML** ou as classes (atributo **class**) definidas no **HTML**.

Veja o exemplo na Figura a seguir.

Observe que o CSS foi aplicado ao seletor *input* e seus tipos e pseudoclasses.

```
Arquivo: form_action.html
     <!DOCTYPE html>
 1
     <head>
 2
         <meta charset="UTF-8">
 3
         <meta name="viewport" content="width=device-width, initial-scale=1.0">
 4
         <link rel="stylesheet" href="css/styleForm1.css">
 5
         <title>Formulário com CSS</title>
 6
     </head>
 7
     <body>
 8
         <div class="formulario">
 9
             <h2>Formatação de Formulário</h2>
10
                 <div class="campo">
11
                      <h4>Parabéns, você preencheu o formulário! </h4>
12
                 </div>
13
         </div>
14
     </body>
15
     </html>
16
```



```
<!DOCTYPE html>
1
     <head>
2
         <meta charset="UTF-8">
3
         <meta name="viewport" content="width=device-width, initial-scale=1.0">
4
         <link rel="stylesheet" href="css/styleForm1.css">
 5
         <title>Formulário com CSS</title>
6
     </head>
7
8
     <body>
9
         <div class="formulario">
10
             <h2>Formatação de Formulário</h2>
11
             <form action="form action.html">
12
                  <div class="campo">
13
                      <span class="tag"><label for="nome">Nome:</label></span>
14
                      <input type="text" id="nome" name="nome" placeholder="Seu</pre>
15
                  </div>
16
                  <div class="campo">
17
                      <span class="tag"><label for="s name">Sobrenome:</label></</pre>
18
                      <input type="text" id="s nome" name="s nome" placeholder="</pre>
19
                  </div>
20
                  <div class="campo">
21
                      <span class="tag"><label for="pais">País:</label></span>
22
                      <select id="pais" name="pais" required>
23
                          <option value="" disabled selected>Seu país.. </option</pre>
24
                          <option value="brasil">Brasil</option>
25
                          <option value="eua">EUA</option>
26
                          <option value="argentina">Argentina</option>
27
                          <option value="outro">Outro</option>
28
                      </select>
29
                  </div>
30
                  <div class="campo">
31
                      <span class="tag"><label for="email">E-mail:</label></span</pre>
32
                      <input type="e-mail" id="email" name="email" placeholder="</pre>
33
                  </div>
34
                  <div class="campo">
35
                      <span class="tag"><label for="pwd">Senha:</label></span>
36
                      <input type="password" id="senha" name="senha" placeholder</pre>
37
                  </div>
38
                  <div class="campo">
39
                      <input type="submit" value="Enviar">
40
                  </div>
41
             </form>
42
         </div>
43
44
     </body>
45
46
```

Arquivo: css/styleForm1.css	

```
1
         margin: 2px;
2
         padding: 2px;
3
         font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
4
5
     }
6
     body{
7
         background-color:#eeeeee ;
8
         justify-content: center;
9
     }
10
     h2{
11
         color: #697b98;
12
         text-align: center;
13
14
     }
     input[type=submit] {
15
         width: 100%;
16
         background-color: #697b98;
17
         color: white;
18
         padding: 14px 20px;
19
         margin: 8px 0;
20
         border: none;
21
         border-radius: 8px;
22
         text-transform: uppercase;
23
         text-align: center;
24
         font-weight: 700;
25
26
         letter-spacing: 1px;
         cursor: pointer;
27
28
     }
29
     .formulario {
30
31
         position: relative;
         width: 100%;
32
         max-width: 500px;
33
         min-height: 300px;
34
         margin: 50px;
35
         background: #fff;
36
         box-shadow: 0 35px 55px rgba(0, 0, 0, 0.3);
37
     }
38
     .campo{
39
         margin: 10px;
40
41
     }
42
     .campo .tag{
43
         width: 100px;
44
         display: inline-block;
45
         background: rgb(210 230 244);
46
```



Ai quivo.	CSS/StyleForm2.CSS

```
1
         margin: 2px;
2
         padding: 2px;
3
         font-family:'Courier New', Courier, monospace;
4
         font-size: medium;
5
6
7
     }
     body{
8
         background-color:#efd5d5 ;
9
         justify-content: center;
10
     }
11
     h2{
12
         color: #753636;
13
14
         font-size:xx-large;
         text-align: center;
15
     }
16
     input{
17
         padding: 6px 6px;
18
         background-color: #f7eded;
19
         border: none;
20
         border-bottom: solid 2px #753636;
21
     }
22
     select {
23
         padding: 6px 6px;
24
         background-color: #f7eded;
25
         border: none;
26
         border-bottom: solid 2px #753636;
27
     }
28
29
     input[type=submit] {
30
31
         width: 60%;
         background-color: #753636;
32
         color: white;
33
         padding: 15px 15px;
34
         margin: 8px 0;
35
         border: none;
36
         border-radius: 30px;
37
         text-align: center;
38
         font-weight: 500;
39
         letter-spacing: 10px;
40
         margin-left: 70px;
41
     }
42
     .formulario {
43
         position:
                     relative;
44
         width: 100%;
45
         max-width: 500px;
46
```

```
min-height: 300px;
47
48
         margin: 50px;
         background: #fff;
49
         box-shadow: 0 35px 55px rgba(0, 0, 0, 0.3);
50
51
     }
52
     .campo{
53
         margin: 10px;
54
     }
55
     .campo .tag{
56
         width: 100px;
57
58
         padding: 6px 6px;
         display: inline-block;
59
         background: rgb(212 141 141);
60
         color: white;
61
62
         margin-left: 70px;
63
         font-weight: 700;
64
         border: none;
         border-radius: 20px;
65
66
     }
```



	Nome:	Seu nome
País: E-mail: Senha:	Sobrenome:	Seu sobrenome
		Seu país v
		nome@email.com Sua senha
	Selma:	Sua senna

Flgura 17: Mesmo formulário HTML com 2 formatações diferentes de CSS. Fonte: A autora (2023).

ATIVIDADE FORMATIVA

Execute o exercício proposto na **videoaula-tutorial** a seguir, para criar um *curriculum vitae* em HTML, estilizado com CSS. Nesta prática, além de exercitar diferente maneiras de formatar elementos HTML, você poderá verificar como um ambiente de desenvolvimento adequado, como o VS Code (https://code.visualstudio.com/download), auxilia bastante na escrita dos códigos.

Além disso, um navegador com **janela de inspeção** vai permitir até testes de *layout* para visualizar qual é o mais interessante para a sua página.

Criando páginas HTML com CSS para definição de estilo

Vamos criar um *curriculum vitae* com um *layout* bem diferenciado, para treinar diferentes tipos de formatação possíveis. Observe que o HTML não tem formatação, que é feita totalmente em um arquivo .CSS em separado. Vamos também preparar nossa página para outros *devices*, utilizando a responsividade do CSS, que é a capacidade de adaptar o tamanho das páginas *web* de acordo com o tamanho das telas em que elas são exibidas. Mãos à obra!

Criando páginas HT...





SAIBA MAIS

CSS no W3School

Pesquise mais sobre os temas a seguir.

- Tutorial CSS: https://www.w3schools.com/css
- CSS Formulários: https://www.w3schools.com/css/css form.asp
- CSS Fontes: https://www.w3schools.com/css/css font.asp
- CSS Cores: https://www.w3schools.com/css/css colors.asp
- CSS Avançado: https://www.w3schools.com/css/css3 borders.asp
- CSS Responsivo: https://www.w3schools.com/css/css_rwd intro.asp

Conclusão

Olá!

Também, precisamos ressaltar que, para maior flexibilidade na alteração do *layout* das nossas páginas, devemos:

- Utilizar o CSS externo, incluído como link> na seção <head>.
- As páginas HTML devem evitar utilizar formatação com o atributo **style**. Em vez disso, devemos criar atributos **class** para os elementos HTML, que serão formatados no **CSS**.

Na próxima unidade, ainda, continuaremos no *front-end*, contudo com programação, pois utilizaremos o JavaScript. Realize as práticas indicadas aqui para o HTML juntamente com CSS, pois assim você aproveitará melhor nossa próxima unidade!

Até lá!

Referências Bibliográficas

ALVES, W. P. Desenvolvimento e design de sites. São Paulo: Erica, 2014.

MILETTO, E. M.; BERTAGNOLLI, S. C. **Desenvolvimento de software II**: Introdução ao desenvolvimento web com HTML, CSS, Java Script e PHP. Porto Alegre: Bookman, 2014.

TERUEL, E. C. HTML 5: Guia prático. 2. ed. Porto Alegre: Bookman, 2014.



© PUCPR - Todos os direitos reservados.