

## Aula 2 – CSS3

### Objetivo da Aula

Conhecer as principais propriedades que compõem a Folha de Estilos em Cascata – CSS (Cascading Style Sheets).

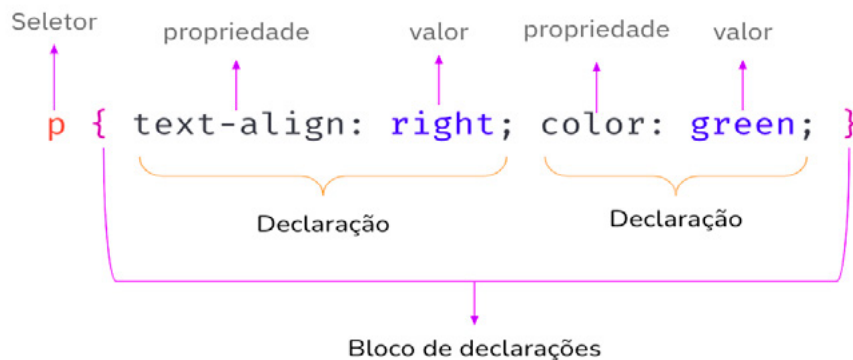
### Apresentação

Uma página ou um site feito apenas com HTML não é muito bonito, certo? Antes do CSS, toda a estilização tinha que ser incluída na marcação HTML, utilizando tags e atributos que caíram em desuso com a chegada do HTML5. Agora que já aprendemos alguns dos elementos que compõem a linguagem HTML, chegou o momento de aprendermos como estilizar estes elementos. A Folha de Estilos em Cascata (CSS) é uma linguagem baseada em regras utilizada para aplicar estilos (cor de fundo, borda, cor de texto, tamanho da fonte...) ao seu documento. As características visuais aplicadas aos elementos são representadas pelo que chamamos de propriedades e essas propriedades podem ser combinadas, trazendo possibilidades de personalização quase infinitas. Ao longo desta aula você vai aprender as principais propriedades do CSS.

### CSS (Cascading Style Sheets) – Folha de Estilo em Cascata

“Um estilo é essencialmente uma regra que instrui ao navegador como formatar algo em uma página web” (MCFARLAND, 2015). É como se o estilo estivesse falando o seguinte para o navegador: “Ei, navegador, faça isso se parecer com isto.” A estrutura da sintaxe do CSS é bem simples, porém, existem algumas regras que você precisa saber.

**Figura 1 | Sintaxe CSS**



Fonte: Webdev Book (<https://webdev.jesielviana.com/frontend/css>).

Como pode ser visto na Figura 1, sua sintaxe é composta por um seletor (elemento HTML que você deseja estilizar) e um bloco de declaração. O bloco de declarações pode conter uma ou mais declarações separadas por ponto e vírgula (;) e cada declaração contém uma propriedade CSS e um valor, separados por dois pontos (:). Resumindo, você seleciona um elemento e depois declara que tipo de efeito você deseja aplicar nele. Simples, não é mesmo? O exemplo da Figura 1, está definindo o alinhamento do texto à direita e a cor do texto verde para todos os parágrafos presentes no seu documento de HTML. Agora vamos entender detalhadamente cada um dos itens que compõem a sintaxe do CSS:

- **Seletor:** responsável por informar ao navegador qual elemento de HTML será estilizado. O CSS possui uma ampla variedade de seletores e sua principal função é “encontrar” o elemento HTML com base no nome do elemento (tag), da classe (class), do identificador (id) etc. Observe na Tabela 1 os tipos mais comuns de seletores:

**Tabela 1 | Seletores mais comuns**

| Seletor         | Exemplo                    | O que ele seleciona   |
|-----------------|----------------------------|---|
| tag ou elemento | <code>p { }</code>         | Todos os elementos HTML de determinado tipo (tag). Qualquer tag/elemento pode ser usada como seletor.   |
| class           | <code>.my-class { }</code> | O(s) elemento(s) na página com a classe especificada. Vários elementos podem possuir a mesma classe. O seletor de classe inicia com um ponto (.).       |
| id              | <code>#my-id { }</code>    | O(s) elemento(s) na página com o id especificado. Só deve haver um elemento com o mesmo id por página HTML. O seletor de id inicia com uma hashtag (#). |

| Seletor    | Exemplo      | O que ele seleciona   |
|------------|--------------|---|
| pseudoclas | a: hover { } | O(s) elemento(s) especificado(s), mas somente quando estiver no estado especificado. No exemplo é utilizado o hover que é um estado ativado quando o mouse está sobre o elemento. O seletor de pseudoclas é definido com o nome do seletor, dois pontos e o estado (ex.: seletor:estado). |

Fonte: Webdev Book (<https://webdev.jesielviana.com/frontend/css>).

- Bloco de declarações: delimitado por { }, o bloco de declarações é responsável por incluir todas as opções de formatação que você deseja aplicar a um determinado seletor;
- Declaração: indica qual é a instrução de formatação através da propriedade e do valor associado a ela;
- Propriedade: css oferece uma ampla variedade de propriedades. Uma “propriedade é a forma pela qual você estiliza um elemento HTML. Cada propriedade está relacionada a uma característica que pode ser modificada no elemento HTML” (WEBDEV BOOK, 2019);
- Valor: “o valor da propriedade é a forma pela qual você define o estilo para determinada propriedade, afetando diretamente a aparência do elemento HTML” (WEBDEV BOOK, 2019).

Vale ressaltar que, uma única propriedade não irá fazer milagre e transformar a sua página em uma verdadeira obra de arte, o segredo aqui é combinar diferentes propriedades para obter páginas com um design incrível.

## Link

Aprenda mais sobre as propriedades e suas variações no site oficial da MDN (<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Reference>). Acesso em: 05/01/2023.). que separou e organizou as propriedades em ordem alfabética.

## Três Maneiras de Vincular o CSS ao HTML

Daí você deve estar se perguntando: como devo vincular o CSS a um documento HTML? Basicamente, existem três maneiras diferentes de fazer isso, são elas:

- Inline: inline significa em linha, então, neste modo as regras são declaradas dentro da tag através do atributo style. Vejamos:

```
<h1 style = "font-family: arial; text-align: center"> Gran Cursos </h1>
```

- Portanto, esta não é uma boa prática, pois mistura o código de CSS com HTML, tornando mais complexo a reutilização, evolução e manutenção do mesmo;
- Interno ou Incorporado: neste modo as regras são declaradas na área comportamental do nosso código HTML, ou seja, na tag <head> através da tag <style>. Vejamos:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Gran Cursos Online</title>
6      <style>
7          h1{
8              font-family: arial;
9              text-align: center;
10         }
11     </style>
12 </head>
```

Portanto, também não é considerado uma boa prática, pelos mesmos motivos descritos no inline. Imagine um site com muitas páginas, para padronizá-las seria necessário copiar e colar a folha de estilo interna em cada uma das páginas. Chato, né? Imagine se você resolve trocar, por exemplo, a fonte escolhida inicialmente, você precisaria editar o estilo em todas as páginas. Pior ainda, não é mesmo? Sendo assim, a melhor solução para estes problemas citados acima é utilizar a folha de estilo externa.

- Externo ou Importado: neste modo as regras são declaradas em um arquivo.css, ou seja, separadas do arquivo HTML. ATENÇÃO: aqui não é necessário utilizar <style>. Podemos vincular uma folha de estilo a uma página HTML de duas formas diferentes:

1) tag <link>: `<link rel="stylesheet" href="css/estilo.css">` onde rel="stylesheet" indica o tipo de link, que neste caso é uma folha de estilo e o href indica a localização do arquivo.css;

2) diretiva @import: a diretiva deve ser adicionada dentro da tag <style> HTML. Vejamos:

```
<style>
    @import url(css/estilo.css);
</style>
```

Ambos fazem a mesma coisa, porém a tag <link> é a mais utilizada; é uma questão de preferência.

## Principais Propriedades de Texto

Agora que já sabemos como integrar o CSS no HTML, aprenderemos as principais propriedades para a formatação de texto:

- **color:** altera a cor do texto. Podemos mencionar a cor de 4 maneiras diferentes: pelo nome em inglês, código em hexadecimal, código em RGB e código RGBA, o mais novo recurso do CSS3. Vejamos:

|  |                                      |
|--|--------------------------------------|
| <b>nome</b>                              | color: blue;<br>Exemplo              |
| <b>hexadecimal</b>                       | color: #0000FF;<br>Exemplo           |
| <b>RGB</b>                               | color: RGB(0,0,255);<br>Exemplo      |
| <b>RGBA</b><br>(red, green, blue, alpha) | color: RGBA(0,0,255,0.5);<br>Exemplo |

Repare que o RGBA é muito parecido com o RGB, seu diferencial é o canal alpha que permite controlar a opacidade (transparência) da cor. O valor assumido pelo canal alpha pode variar de 0 (transparente) a 1 (opaca).

- **font-family:** define uma lista de fontes que será utilizada para estilizar o elemento. Os nomes das fontes escolhidas devem estar separados por vírgulas, utilizando a ordem de preferência. Vejamos:

```
<p style="font-family: Trebuchet MS, Arial, Times, sans-serif;">Gran Cursos</p>
```

Se o usuário tiver a fonte Trebuchet MS, o navegador irá usá-la; caso contrário, usará a Arial; caso contrário, usará a Times; e caso o usuário não tenha nenhuma destas fontes

instaladas, a fonte padrão sem serifa do sistema operacional será utilizada. Uma fonte sem serifa é aquela que não possui os prolongamentos nas suas extremidades.

- `font-size`: define o tamanho da fonte. Esta propriedade permite tais unidades: px (pixels), em (unidade relativa), pt (pontos), %, entre outras;
- `font-weight`: define a intensidade ou espessura da fonte. Esta propriedade pode ter os seguintes valores: normal | bold | bolder | lighter, entre outros;
- `font-style`: define o estilo da fonte. Esta propriedade pode ter os seguintes valores: normal | italic | oblique;
- `font-variant`: transforma em maiúsculas de menor altura (**GRAN CURSOS**). Esta propriedade pode ter os seguintes valores: normal | small-caps;

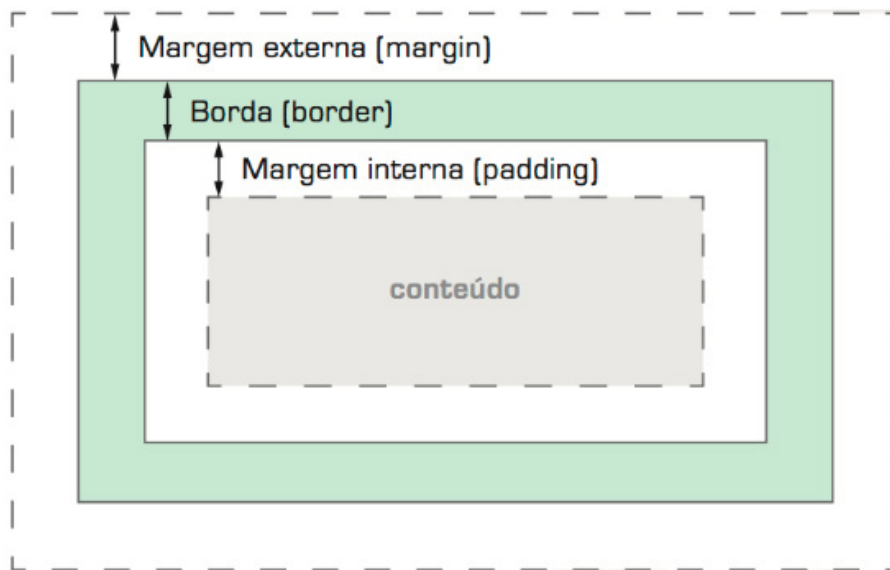
```
<p style="font-variant: small-caps">Gran Cursos</p>
```

- `line-height`: define o espaçamento entre linhas. Nesta propriedade é comum o uso da unidade em, pois com ela deixamos o espaçamento proporcional ao tamanho da fonte, por exemplo, `line-height: 2.5em`; (2.5 vezes o tamanho da fonte);
- `text-indent`: define a indentação (ou tabulação) da primeira linha do texto, por exemplo, `text-indent: 10px`;
- `text-transform`: define efeitos de capitalização do texto, são eles: caixa alta (uppercase), caixa baixa (lowercase) ou primeiras letras das palavras em maiúscula (capitalize);
- `Text-decoration`: define uma linha decorativa para o texto. Esta propriedade pode ter os seguintes valores: overline (superlinhado), underline (sublinhado), line-through (riscado), none (sem efeito). Vejamos:

|  |             |
|--|-------------|
| <code>&lt;p style="text-decoration: overline;"&gt; Gran Cursos &lt;/p&gt;</code>     | Gran Cursos |
| <code>&lt;p style="text-decoration: underline;"&gt; Gran Cursos &lt;/p&gt;</code>    | Gran Cursos |
| <code>&lt;p style="text-decoration: line-through;"&gt; Gran Cursos &lt;/p&gt;</code> | Gran Cursos |
| <code>&lt;p style="text-decoration: none;"&gt; Gran Cursos &lt;/p&gt;</code>         | Gran Cursos |

Outra coisa que devemos considerar é que no HTML tudo é uma caixa (box). Como assim? "A maioria dos elementos HTML podem ser pensados como caixas, que podem ser agrupadas de forma horizontal, vertical ou colocada dentro de outra" (WEBDEV BOOK, 2019).

**Figura 2 | Caixa de um elemento HTML**



Fonte: Webdev Book (<https://webdev.jesielviana.com/frontend/css>)

O box (caixa) que representa um elemento HTML (Figura 9) é composto por conteúdo, margem interna (padding), borda (border) e margem externa (margin). Todas essas partes que compõem o elemento é conhecida como box model (WEBDEV BOOK, 2019).

Além das propriedades de texto que aprendemos acima, vamos conhecer também algumas propriedades muito utilizadas para estilizar o espaço que o elemento ocupa, são elas (WEBDEV BOOK, 2019):

- **width** – define a largura de um elemento que pode ser definido com valor fixo usando pixel (px) ou valor dinâmico usando porcentagem (%);
- **height** – define altura de um elemento que pode ser definido com valor fixo usando pixel (px) ou valor dinâmico usando porcentagem (%);
- **margin** – define o espaçamento externo de um elemento;
- **border** – define a borda do elemento;
- **padding** – define o espaçamento interno, ou seja, espaço entre a borda e o conteúdo;
- **background-color** – define a cor de fundo do elemento.

**Destaque**

Quando as propriedades padding, margin e border recebem um único valor (padding: 2px;) alteram os quatro lados de um elemento: topo, direita, baixo e esquerda, porém é possível especificar o valor de forma individual e separadamente (padding-top: 2px; padding-right: 4px; padding-bottom: 6px; padding-left: 8px;) ou em conjunto (padding: 2px 4px 6px 8px;), passando os 4 valores de uma vez.

## Considerações Finais

Nesta unidade aprendemos a sintaxe utilizada na linguagem CSS, como ela pode ser aplicada e as suas principais propriedades. Além disso, vimos que o CSS foca em toda a parte estética, enquanto o HTML foca na estrutura e na semântica do documento. Vimos também que existe uma forte relação entre o HTML e o CSS, porém, é importante ressaltar que é uma boa prática separar o conteúdo da representação visual do site, pois isso facilita a reutilização, a evolução e a manutenção do código.

## Materiais Complementares

Lista de propriedade em CSS:

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Reference>

Principais Propriedades de Texto CSS:

<https://youtu.be/vtP3Vqo2Kog>

Aprenda Construir Layouts Elegantes com CSS Grid e Flexbox:

[https://youtu.be/1mf4mZE9o\\_4](https://youtu.be/1mf4mZE9o_4)

O que é CSS? Guia Básico para Iniciantes:

<https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>

Apostila de CSS – Curso W3C Escritório Brasil:

<https://www.w3c.br/pub/Cursos/CursoCSS3/css-web.pdf>

## Referências

CSS. Webdev Book, 2019. Disponível em: <<https://webdev.jesielviana.com/frontend/css>>. Acesso em: 02 de nov. de 2022.

MCFARLAND, David Sawyer. CSS3: o manual que faltava. Rio de Janeiro, RJ: AltaBooks, 2015.