

1 O que é o CSS?

O CSS (Cascading Style Sheets ou Folhas de Estilo em Cascata) é uma linguagem de estilo usada para aplicar visuais a uma página HTML. Ela é a principal linguagem responsável pela aparência de uma página de um site, e portanto, é muito importante aprender como utilizá-la.

1.1 Como aplicar o CSS?

Para trazermos CSS para nossa página HTML existem várias formas, porém cada uma tem vantagens e desvantagens que serão abordadas a seguir.

1.1.1 Estilização inline

Esse tipo de utilização do CSS já foi mostrada anteriormente, usamos o atributo global **style** presente em qualquer tag.

Com este atributo podemos aplicar uma estilização local e totalmente vinculada ao elemento HTML que recebe o atributo. Por exemplo: `<p style="color: red; font-size: 24px;">Olá, CSS!</p>` Essa forma é funcional, porém, se tivermos múltiplos parágrafos que desejamos aplicar os mesmos estilos, não será tão vantajoso. Se trata de uma estilização local e muito específica.

1.1.2 Tag style

O uso da tag `<style>` garante aplicar a sintaxe do CSS de maneira concreta, diferente da forma anterior. Veja o código a seguir:

Listagem 1: Tag style

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="utf-8" />
5 <title>Minha pagina de teste</title>
6 <style>
7 p {
8 color: red;
9 font-size: 24px;
10 }
11 </style>
12 </head>
13 <body>
14 <p>Hello, CSS1!</p>
15 <p>Hello, CSS2!</p>
16 <p>Hello, CSS3!</p>
17 </body>
18 </html>
```

Com este método, podemos utilizar totalmente o que o CSS dispõe: os **seletores**. Esta forma garante uma estilização dentro de um único arquivo HTML. Porém, caso tenhamos um site com múltiplas páginas acabaríamos tendo que repetir a estilização.

1.1.3 Arquivo CSS

A terceira forma garante uma reutilização possível para as páginas do site, pois é um arquivo separado apenas para o CSS. Observe o seguinte arquivo `styles.css`:

Listagem 2: `styles.css`

```
1 p {  
2 color: red;  
3 font-size: 24px;  
4 }
```

Como poderemos ter múltiplas estilizações dentro de um site, é recomendado criar uma pasta para abrigar os arquivos css, neste material iremos utilizar a nomenclatura **styles** para a pasta em questão.

Porém, você deve notar que esta estilização não funciona automaticamente em nossa página. Para que funcione devemos linkar o CSS e o HTML que irá utilizar esta estilização. Para isso usamos a tag **link**: `<link rel="stylesheet" href="styles/styles.css" />`

Este elemento `<link>` diz ao navegador que temos uma folha de estilo, usando o atributo `rel`, e a localização desse arquivo como o valor do atributo `href`.

1.2 Qual forma utilizar?

Foram apresentadas três formas diferentes de se aplicar CSS em uma página HTML, porém é importante saber quando utilizar cada forma, e a resposta é: **depende**.

A dica é: *quanto mais organizado o projeto do seu site, mais fácil será posteriormente de alterar e inserir novas funcionalidades. Devemos sempre pensar por esse lado, pois um site bem organizado garante uma facilidade maior de manutenção.*

Portanto, recorra à maioria dos casos para a estilização em **arquivos CSS**, pois não só será mais fácil de manutenção posterior, como também torna a estilização mais organizada e separada, pois podemos linkar múltiplos arquivos CSS usando a tag **link**.

Para situações muito específicas (mesmo), utilize as tags `<style>` ou o atributo `inline`, porém essas situações dificilmente acontecerão se você entender bem como funciona o **efeito cascata** do CSS.

2 Sintaxe e Seletores

No HTML, nós temos as populares tags que nos ajudam a construir a estrutura da página. Porém, as tags por si só não possuem visuais definidos, uma vez que estes visuais são definidos pelo navegador que está sendo usado para ler a página. Com o uso do CSS podemos garantir que o site será visualizado de maneira consistente em qualquer navegador.

No CSS, temos os **seletores**, que são um mecanismo que nos permite selecionar grupos de elementos existentes da página de diversas formas e assim aplicar um visual específico. Nas seções a seguir, você aprenderá os seletores mais essenciais e quando utilizá-los.

2.1 Sintaxe da linguagem

Antes de partir para qualquer seletor do CSS, é importante entender mais sobre a sintaxe desta linguagem. O CSS é baseado em regras, essas regras se aplicam aos elementos particulares ou grupos de elementos da página. Para escrever CSS, devemos sempre seguir a sintaxe a seguir:

Listagem 3: Sintaxe do CSS

```
1 <regra ou seletor> {  
2 propriedade1: valor;  
3 propriedade2: valor;  
4 ...  
5 }
```

Aplicar uma regra ou um seletor específico, abrimos um par de chaves e dentro aplicamos as diversas propriedades e valores possíveis para customizar seja quem for aplicável as regras definidas.

O uso do ponto e vírgula (;) é importante para separar as declarações de propriedade e valor que são as estilizações em si.

2.2 Seletores de tipo

Esse seletor nos permite selecionar elementos da página baseados em sua tag HTML, ou seja, com esse tipo de seletor, podemos aplicar estilo a todos os elementos de uma tag em particular. Imagine o seguinte HTML:

Listagem 4: Arquivo HTML

```
1 <h1>I am a level one heading</h1>  
2  
3 <p>This is a paragraph of text. In the text is a <span>span element</span> 4 and also a  
<a href="http://example.com">link</a>.</p>  
5
```

```

6 <p>This is the second paragraph. It contains an <em>emphasized</em> element.</p> 7
8 <ul>
9 <li>Item one</li>
10 <li>Item two</li>

```

3

Unidade I: Formas de Uso e Seletores soulcode.com

```

11 <li>Item <em>three</em></li>
12 </ul>

```

Podemos usar o seletor de tipo para estilizar as tags mostradas no código acima:

Listagem 5: Uso de seletores de tipo

```

1 h1 {
2 font-size: 32px;
3 }
4
5 p {
6 font-size: 18px;
7 font-style: italic;
8 }
9
10 li {
11 color: green;
12 }
13
14 h1, p {
15 color: red;
16 }

```

Observe a **linha 1**, que define uma regra CSS para os elementos <h1> de uma página, todos os elementos presentes terão um tamanho de 32px. Na **linha 5**, todos os <p> da página terão tamanho de 18px e estilo itálico. Na **linha 10**, os itens da lista terão cor verde.

Na **linha 14**, temos uma regra diferente, podemos separar por vírgula diversos seletores e assim eles terão a mesma estilização. Dessa forma tanto <h1> quanto <p> terão cor vermelha.

Exercício: Redefina as regras CSS de forma que os elementos h1, p e li possuam a mesma cor vermelha.

2.3 Seletores de ID

Um seletor por ID, define uma estilização para um único elemento da página. Para isso devemos definir o atributo **id** para um elemento da página, e criar o seletor da mesma forma no CSS. Por exemplo, imagine o seguinte elemento dentro do arquivo HTML mostrado anteriormente:

```
<p id="special">I am very special.</p>
```

No CSS devemos selecionar este elemento da seguinte maneira:

Listagem 6: Seletor por ID

```

1 #special {
2 color: orange;
3 font-size: 10px

```

4 }

Com este tipo de seletor, apenas o parágrafo com este ID receberá a estilização. Este seletor é um dos mais específicos junto da estilização inline, pois eles só podem ser aplicados a um único elemento específico da página.

4

Unidade I: Formas de Uso e Seletores soulcode.com

Vale ressaltar, que se existirem seletores de tipo para <p> no arquivo CSS, algumas destas propriedades serão sobrescritas pelo que estiver definido na declaração do seletor por ID (#special).

2.4 Seletores de classe

Um seletor de classe, é utilizado para estilizar grupos de elementos que definirem esta classe por meio do atributo **class**. Por exemplo, imagine o seguinte HTML:

Listagem 7: Uso do atributo class

```
1 <h1 class="marca-texto">Class selectors</h1>
2 <p>Veggies es bonus vobis, proinde vos postulo essum magis <span class="marca-texto">kohlrabi welsh onion</span> daikon amaranth tatsoi tomatillo 3 melon azuki bean garlic.</p>
4
5 <p class="marca-texto">Gumbo beet greens corn soko <strong>endive</strong> gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard 6 greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.</p>
```

Podemos observar, o uso múltiplo do atributo **class** com valor *marca-texto*. No CSS podemos criar uma regra que estiliza qualquer elemento que esteja com esta classe:

Listagem 8: Seletor de classe

```
1 .marca-texto {
2 background-color: yellow;
3 }
```

Exercício: Com base na leitura desta [seção](#), defina uma cor de texto diferente para o , <h1> e <p> que utilizarem a classe "marca-texto" usando a segmentação de classes.

2.5 Seletores descendentes

Os seletores descendentes permitem selecionar elementos que são descendentes diretos ou indiretos de outro elemento. Ou seja, podemos estilizar elementos que estão dentro de outros elementos.

Vamos considerar o seguinte HTML:

Listagem 9: Exemplo de HTML

```
1 <div class="container">
2 <h1>Titulo</h1>
```

```
3 <p>Paragrafo 1</p>
4 <div>
5 <p>Paragrafo 2</p>
6 <p>Paragrafo 3</p>
7 </div>
8 </div>
```

5

Unidade I: Formas de Uso e Seletores soulcode.com

Caso queiramos estilizar todos os elementos `<p>` que existirem dentro da `div` com a classe "container", podemos criar o seguinte seletor:

Listagem 10: Seletor de descendente

```
1 .container p {
2 color: red;
3 }
```

Observe que existe um espaço entre `.container` e `p`. Essa regra não aplicará cor vermelha em parágrafos fora da `div` com classe "container". Podemos trabalhar mais ainda e fazer estilizações mais específicas, de forma que os parágrafos mais internos tenham uma estilização diferente:

Listagem 11: Seletor de descendente

```
1 .container div p {
2 color: red;
3 }
```

Esta regra é diferente da anterior no sentido de que o parágrafo da linha 3 não se encaixa nela. A regra diz: "selecione todos os parágrafos dentro das `div` que estão dentro de `.container`".

Caso queiramos estilizar apenas o parágrafo da linha 3, podemos usar um seletor de descendente direto, ou filho direto. Veja como seria:

Listagem 12: Seletor de descendente direto

```
1 .container > p {
2 color: green;
3 }
```

O maior que (`>`) determina que apenas os elementos `<p>` filhos diretos de `.container` receberão a estilização. Este seletor funciona para outras classes e ID também não apenas seletores por tipo.

2.6 Seletores de pseudo-classes

Os seletores de pseudo-classe em CSS são usados para selecionar elementos com base em um estado ou contexto específico.

As pseudo-classes são adicionadas aos seletores para estilizar elementos em situações especiais, como quando um link está sendo visitado, quando um elemento está sendo passado com o mouse ou quando um elemento está em um estado de foco.

Listagem 13: Pseudo-classes simples

```
1 button:hover {  
2 background-color: yellow;  
3 }  
4  
5 button:active {  
6 background-color: green;  
7 }
```

6

Unidade I: Formas de Uso e Seletores soulcode.com

Estes estilos serão aplicados quando o mouse passar por cima do botão (ho ver), e quando o usuário clicar no botão (active) em particular. Você pode aprender mais sobre pseudo-classes [aqui](#).

2.7 Mais sobre seletores

Caso queira aprender mais sobre outros tipos de seletores mais complexos, recomendo a leitura da documentação sobre [Seletores CSS](#) do MDN Web Docs. Existem muitos tópicos interessantes sobre o CSS, como:

- [Pseudo-elementos](#);
- [Seletores de atributo](#);
- [Seletor universal](#);
- [Seletor de elementos irmãos](#).

Caso queira praticar mais o uso dos seletores há um jogo muito interessante e didático: [CSS Diner](#).

3 Características da linguagem

3.1 Especificidade dos seletores

A especificidade de seletores é o mecanismo que o CSS utiliza para determinar qual estilo será aplicado a um elemento quando existem regras de estilo conflitantes. Cada seletor possui um valor de especificidade atribuído, que é calculado com base nos diferentes componentes do seletor.

Quanto mais específico for o seletor, maior será sua especificidade e maior será a probabilidade de que seu estilo seja aplicado. A seguir, estão os componentes que afetam a especificidade de seletores, em ordem crescente de importância:

1. Seletor universal (*) e pseudo-elementos (::before, ::after): Especificidade mais baixa;
2. Seletor de tipo (por exemplo, p, div): Especificidade moderada;

3. Seletor de classe, seletor de atributo e pseudo-classes (por exemplo, `.classe`, `[atributo]`, `:hover`): Especificidade média;
4. Seletor de ID (`#id`): Especificidade alta;
5. Seletor inline (atributo `style`): Especificidade máxima.

Este [site](#) mostra de maneira visual a especificidade de cada seletor e sua prioridade no CSS. Também recomendo a leitura deste [artigo](#) do MDN Web Docs.

7

Unidade I: Formas de Uso e Seletores soulcode.com

3.2 Herança de estilos

A herança de estilos é um princípio do CSS que permite que as propriedades de estilo sejam transmitidas dos elementos pais para os elementos filhos. Isso significa que um elemento filho pode herdar propriedades como cor, fonte e tamanho do texto do seu elemento pai. No entanto, nem todas as propriedades são herdadas.

3.3 Princípios de cascata

O princípio de [cascata](#) no CSS refere-se à maneira como as regras de estilo são aplicadas quando existem várias regras para o mesmo elemento. O CSS segue uma ordem específica na avaliação das regras:

- Primeiro, são aplicadas as **regras mais específicas**. Isso significa que uma regra com um seletor mais específico, como um seletor de ID, terá prioridade sobre uma regra com um seletor de classe ou seletor de tipo;
- Em seguida, as **regras de igual especificidade** são avaliadas pela ordem em que aparecem no código. A regra que aparece por último terá prioridade sobre as demais;
- Por fim, as regras que têm **menor especificidade** são aplicadas quando não há regras mais específicas.

