

ntander Coders 2024.1 | Front-End | #1178

Isaias Soares

Id: 1178008



# SS

CSS é uma linguagem que permite a estilização de páginas HTML. Com ela, podemos definir todas as configurações visuais dos elementos HTML.

O CSS é baseado em dois conceitos:

1. Seletores - Sintaxe que permite indicar qual elemento HTML será estilizado.
2. Regras de Estilo - Sintaxe que permite alterar os parâmetros visuais do elemento.

A sintaxe básica do CSS será:

```
seletor { regra1 : valor1; ... regraN : valorN; }
```

ou quebrando linhas:

```
seletor {  
  regra1 : valor1;  
  ...  
  regraN : valorN;  
}
```

Você pode ter quantas regras quiser para um mesmo seletor.

# Adicionando CSS ao HTML

---

Existem três formas de adicionar CSS em sua página HTML.

1. propriedade style do elemento HTML
2. elemento style (`<style></style>`)
3. carregar um arquivo css usando: `<link rel='stylesheet' type='text/css'>`

Recomendamos apenas a terceira opção, mas veremos todas:

## Propriedade style

Todo elemento HTML possui uma propriedade `style`, ela permite aplicar regras de CSS diretamente ao elemento, portanto, dispensa o seletor. Essa forma de aplicar CSS é considerada uma má prática, no entanto, existem situações em que só é possível estilizar um elemento usando essa técnica (mas são raras).

*Parágrafo com cor de texto vermelha:*

```
<p style="color:red">Teste</p>
```

*Resultado:*

Teste

**Imagem 1** - *Texto, aplicando a cor da fonte vermelha (Fonte da imagem: do autor)*

Demais regras sempre são separadas com ; .

*Parágrafo com cor de texto vermelha, "margem interna" e borda:*

```
<p style="color:red;padding:10px;border:1px solid black">Teste</p>
```

*Resultado:*

Teste

**Imagem 2** - *Texto, aplicando a cor da fonte vermelha, padding e bordas (Fonte da imagem: do autor)*

Observe que nesse último exemplo, a linha de HTML já contém mais CSS do que conteúdo. Para evitar essa "poluição" do código HTML, usamos essa forma de aplicar CSS raramente.

## Elemento style

Podemos aplicar diversos estilos em diversos elementos usando a *tag* `<style></style>`. Ela é uma *tag* HTML. Tenha em mente que o HTML renderiza linha a linha, portanto, a *tag* `style` deve estar em um local anterior aos componentes visuais para que eles renderizem aplicando os estilos. Se estiver depois, eles renderizarão com a aquela cara feia padrão e depois receberão o visual correto.

Sendo assim, `<style></style>` costuma aparecer, por questões de boas práticas, na última linha do `<head></head>`.

*Todo parágrafo deve ser vermelho*

```
<style type="text/css">
  p {
    color: red;
  }
</style>
```

Aqui vemos a divisão mencionada acima: `p` é o seletor, `color` a regra e `red` o valor.

Como antes, regras adicionais podem ser inseridas separadas por `;` com quebra de linha opcional, mas recomendável.

*Todo parágrafo deve ser vermelho, com 'margem interna' e borda.*

```
<style type="text/css">
  p {
    color: red;
    padding: 10px;
    border: 1px solid black;
  }
</style>
```

```
}  
</style>
```

Podemos usar o mesmo tag `style` para estilizar quantos elementos quisermos, basta quebrar uma linha ao final da chave e colocar um novo seletor e suas regras.

Aproveitando, observe a regra `border`. Seu valor é composto por várias partes, no caso: um tamanho (espessura da borda), um estilo (borda sólida) e uma cor.

Várias regras de CSS utilizam múltiplos valores, como o `border`, lembre-se de separá-los por um espaço em branco.

## Arquivo .CSS

O conteúdo de um arquivo de CSS é idêntico ao conteúdo da `tag style`, sendo assim, você pode usar a `tag` provisoriamente e quando estiver contente com o resultado, copiar todo o conteúdo para um arquivo com extensão `.css`.

Os arquivos de CSS devem ser "importados" para o HTML dentro do `tag head` usando o `tag link` como indicado abaixo:

*Importando um arquivo chamado `styles.css` presente no mesmo diretório*

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

Como a `tag link` tem múltiplas funções, no HTML é obrigatório indicar a propriedade `rel` com o valor `stylesheet` (folha de estilo) para que ele aplique o CSS na página. O `type` deve ser setado para `text/css` como boa prática.

Prefira sempre usar arquivos `.css` no lugar dos métodos mostrados anteriormente.

## Precedência

---

É importante mencionar que CSS trabalha com precedência, ou seja, se múltiplos CSS forem encontrados na página, especialmente se forem conflitantes, existem regras para definir qual deles se aplicará.

A ordem de aplicação do CSS será sempre Arquivo -> Elemento -> Propriedade.

O que quer dizer que arquivos .css são aplicados primeiro, se houver um `tag style`, ela será aplicada em seguida sobrescrevendo regras conflitantes (que existem tanto no arquivo quando no `style`), e por último aplica-se o CSS na propriedade `style` do elemento, sobrescrevendo regras conflitantes tanto do `tag style` como do arquivo .css.

As regras de CSS das três formas serão somadas, e todas serão aplicadas. Apenas regras conflitantes são sobrescritas.

Por exemplo, no arquivo `p` tem `color = black` e no tag `style p` tem `color = green` valerá então a `color green`. No entanto, se um elemento `p` tiver um `style` com `color = red` valerá `red`.

Isso vale também para múltiplos arquivos CSS, nesse caso, as regras do último arquivo a ser importado sobrescrevem regras conflitantes dos anteriores.

É uma prática comum no CSS fazer um arquivo que reseta todas as regras de CSS primeiro, para em seguida (em outro arquivo) aplicar apenas as regras desejadas.

Isso porque os navegadores aplicam algumas regras por padrão, que nem sempre (quase nunca) são regras que nós queremos em nossas páginas.

## Estilizando uma página

---

Se você olhar o arquivo HTML disponibilizado na aula passada, já colocamos uma indicação para um CSS nele.

O arquivo HTML da aula passada pode ser baixado [aqui](#).

Baixe o arquivo css [aqui](#), coloque no mesmo diretório onde está seu arquivo HTML, recarregue a página, e veja a diferença.

Fizemos algo básico, sem grande compromisso com design, apenas para posicionar os elementos que criamos nas aulas passadas, mas já é uma grande diferença, não!?

Agora substitua o conteúdo por [esse outro CSS](#), feito por um outro aluno, o Ricardo Sturk do curso Web Fullstack. Viu a diferença!?

HTMLs iguais podem produzir resultados visuais completamente diferentes quando aplicamos outros CSS.

## Referências e Links úteis

---

- [CSS](#). MDN Web Docs, 26/06/2021. Acesso em: 27/06/2022.
- [CSS Tutorial](#). W3 Schools. Acesso em: 27/06/2022.

Próximo Tópico