CURSO DE INFORMÁTICA PARA INTERNET

MÓDULO 01 ▪ CAPÍTULO 12

DevWeb

INTRODUÇÃO ÀS CSS

curso em vídeo

M01C12

INTRODUÇÃO ÀS CSS

Eu costumo sempre dizer que você pode ter o melhor conteúdo do mundo, mas se ele não for bem apresentado, o alcance dele diminui consideravelmente. Visitantes de sites gostam da beleza, mesmo que eles não conheçam nada de design. É uma sensação satisfatória ver um conteúdo organizado e bonito. Esse capítulo vai te mostrar os primeiros passos com o uso de CSS, aplicando esses conceitos em seus códigos.

Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público **https:// github.com/gustavoguanabara/**. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Você se lembra do que falamos sobre as CSS?

Nós já falamos sobre folhas de estilo em cascata, as famosas CSS no **capítulo 3**. Se por acaso você não se lembra direito, vale a pena voltar lá e dar uma segunda olhada nas definições. Vou considerar que você lembra claramente o que são as **Cascading Style Sheets** para podermos prosseguir.

Outro conteúdo muito importante que vimos está no **capítulo 8**, onde falamos sobre a íntima relação da HTML5 com a **semântica** das tags. Lá foi comentado que todo e qualquer efeito visual é responsabilidade das CSS. Vou partir daí e vamos trabalhar com os estilos, ok?

Caso você sinta qualquer dúvida a partir daqui, não se esqueça de revisitar os capítulos anteriores, pois a base foi dada gradativamente até esse momento. Vamos lá!

A forma mais simples de aplicar estilos: CSS inline style

Vamos começar pela técnica mais básica para aplicar estilos em áreas pontuais em nosso site, que é usando as CSS dentro de parâmetros de HTML5. Crie mais uma pasta dentro da sua área de **exercícios** e crie um arquivo `index.html` com aquele código base que já fizemos várias vezes. Dentro da área `<body>`, crie um código como apresentado a seguir:

```
8   <body>
9       <h1>Capítulo 1</h1>
10      <h2>Capítulo 1.1</h2>
11      <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Beatae
        assumenda eveniet odit accusantium distinctio saepe.</p>
12      <h2>Capítulo 1.2</h2>
13      <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit.
        Necessitatibus doloribus pariatur deserunt in nobis labore aliquam
        eos.</p>
14      <h1>Capítulo 2</h1>
15      <h2>Capítulo 2.1</h2>
16      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam,
        nobis quas! Eum saepe temporibus!</p>
17  </body>
```

Agora abra o arquivo recém criado no Google Chrome. O resultado visual deve ser semelhante ao apresentado a seguir:

Capítulo 1

Capítulo 1.1

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Beatae assumenda eveniet odit accusantium distinctio saepe.

Capítulo 1.2

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus doloribus pariatur deserunt in nobis labore aliquam eos.

Capítulo 2

Capítulo 2.1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam, nobis quas! Eum saepe temporibus!

RELEMBRANDO: Não se esqueça que você pode criar esses parágrafos automáticos com texto "Lorem ipsum" apenas digitando o atalho `lorem` no VSCode.

Vamos começar nos focando na tag `<body>` e aplicando um estilo diferente ao corpo da página. Adicione o parâmetro `style` e digite as duas declarações de `font-family` e `color`, conforme apresentado a seguir:

```
<body style="font-family: Arial, Helvetica, sans-serif; color: blue;">
```

Muito cuidado na hora de digitar esse código. Tudo deve ser seguido exatamente como fizemos acima, inclusive com letras maiúsculas e minúsculas. Não esqueça de adicionar os ponto e vírgulas para separar as declarações. Seu resultado visual deve ser esse:

Capítulo 1

Capítulo 1.1

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Beatae assumenda eveniet odit accusantium distinctio saepe.

Capítulo 1.2

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus doloribus pariatur deserunt in nobis labore aliquam eos.

Capítulo 2

Capítulo 2.1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam, nobis quas! Eum saepe temporibus!

Note que o formato da letra mudou (era Times e ficou em Arial) e a cor da fonte também foi alterado para azul. Se por acaso alguma dessas duas alterações não funcionou corretamente com você, confira seu código, pois algo foi digitado incorretamente. Lembre-se que o computador não é tão inteligente quanto você pode pensar. Temos que dar ordens bem claras e seguindo sempre as regras para que ele nos obedeça.

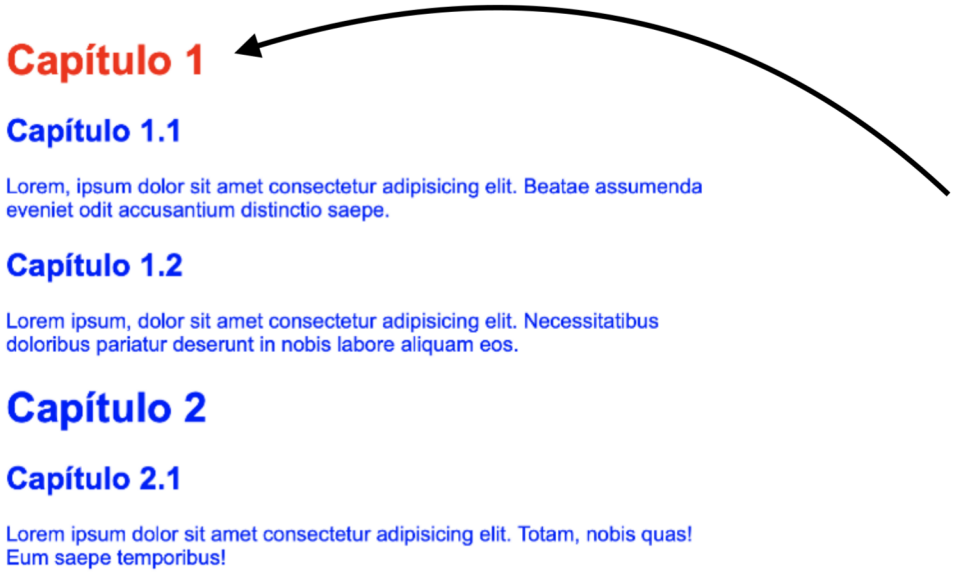
CUIDADO! Se por acaso você aprendeu em algum momento a tag `<font color="blue">` e acha muito mais prática usá-la, saiba que ela **NÃO É MAIS ACEITA** para as especificações da HTML5!

Nós falamos sobre especificações obsoletas de HTML no **capítulo 3**. Se precisar relembrar, volte lá e faça uma revisão do conteúdo.

Vamos fazer mais uma alteração, dessa vez na linha do primeiro título `<h1>` do nosso código:

```
<h1 style="color: red;">Capítulo 1</h1>
```

O resultado visual deve ser:



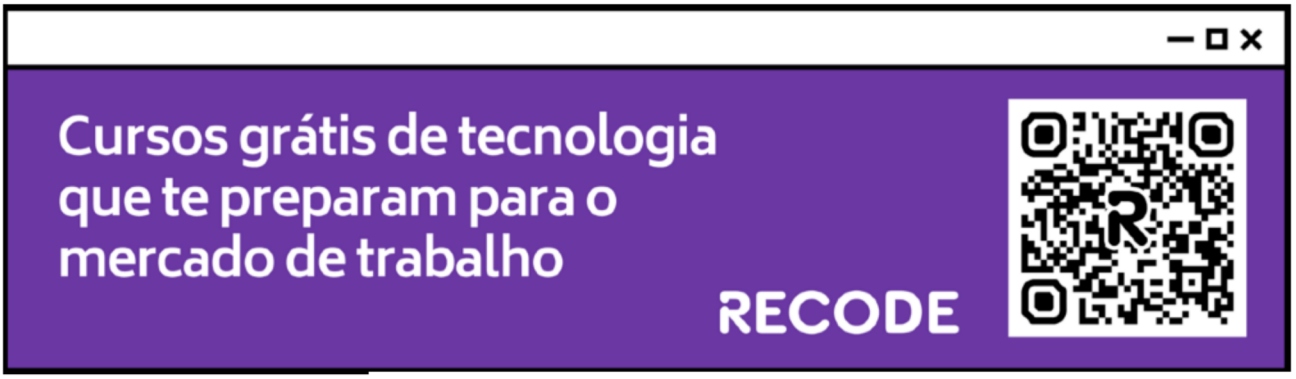
Note que apenas o **Capítulo 1** ficou vermelho, o **Capítulo 2** - que também é um `<h1>` - não teve alteração alguma. Isso acontece pois estamos fazendo **configurações pontuais** usando CSS.

Estilizando de maneira mais interessante: CSS internal style

Para aplicar estilos de forma mais dinâmica e prática, podemos adicionar uma tag `<style>` dentro da área `<head>` do nosso documento HTML local. Volte lá no seu VSCode, e adicione o código dentro de `<head>`.

```
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
6       <title>Estilos pontuais</title>
7       <style>
8           body {
9               font-family: Arial, Helvetica, sans-serif;
10              background-color: lightcyan;
11              color: blue;
12          }
13          h1 {
14              color: green;
15          }
16      </style>
17  </head>
```

ATENÇÃO! A tag `<style>` deve estar dentro da área <head> do seu documento HTML5. Se você colocá-la em qualquer outro local, como dentro da tag `<body>`, o resultado até pode funcionar, mas seu código estará fora dos padrões estabelecidos pela W3C. Siga sempre as regras!



QUERO MAIS CORES: Se você está achando que essa coisa de colocar cor pelo nome em Inglês é algo limitado, você está coberto de razão! No próximo capítulo, vou te mostrar outras técnicas de representar cores em CSS. Aguarde e confie!

Feitas as alterações, vamos ver o resultado e uma dúvida vai surgir:

Capítulo 1

Capítulo 1.1

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Beatae assumenda eveniet odit accusantium distinctio saepe.

Capítulo 1.2

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus doloribus pariatur deserunt in nobis labore aliquam eos.

Capítulo 2

Capítulo 2.1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam, nobis quas! Eum saepe temporibus!

Você sabe explicar por que o **Capítulo 1** ficou **vermelho** e não **verde**, como solicitamos?

Isso acontece porque as configurações pontuais (HTML style) vão prevalecer sobre as configurações gerais (CSS style). Volte ao seu código e remova todas as configurações de estilo que fizemos nas tags `<body>` e `<h1>` no início do capítulo.

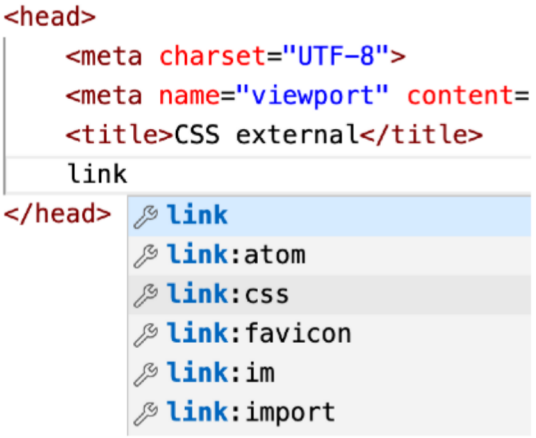
A técnica mais versátil: CSS external style

Manter as folhas de estilo fora do código HTML, além de uma maior organização faz com que tudo seja reaproveitado de maneira mais eficiente nas outras páginas do nosso site. Para isso, utilizamos a tag `<link>` especialmente configurada para trabalhar com arquivos externos de estilo. Essa tag deve ser colocada dentro da área `<head>` do seu documento HTML.

```html
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS external</title>
    <link rel="stylesheet" href="style.css">
</head>
```

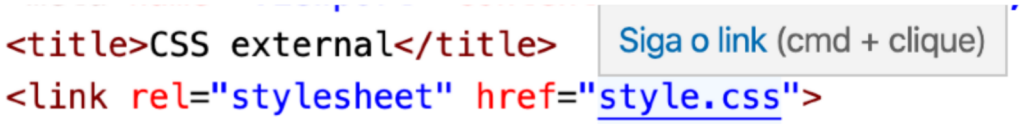
A linha com o `<link>` pode estar em qualquer linha, contanto que seja dentro da área `<head>`. Particularmente, sempre procuro adicionar essa configuração após a tag `<title>` do documento atual.

Dica para criar CSS externo com VSCode



O Visual Studio Code sempre trás algumas facilidades para o nosso dia-a-dia. Vá até o final da linha com o `<title>` e pressione Enter para criar uma nova linha. Depois comece digitando a palavra `link`, sem as marcas de tag (veja a imagem ao lado).

No menu de contexto que vai aparecer, escolha a opção `link:css` e a linha apresentada abaixo será magicamente preenchida.



Agora passe o mouse sobre o nome do arquivo `style.css` e veja que existe um atalho *"Siga o link"*, bastando pressionar `Ctrl+clique` (ou `Cmd+clique`, se estiver usando o sistema MacOS). A primeira vez que você segurar o `Ctrl/Cmd` e clicar sobre o link, o VSCode vai perguntar se você quer que ele crie o arquivo pra você. Clique em `Sim` ou `Ok` para aceitar a ajuda e seu arquivo será criado automaticamente.

Agora é só adicionar os seletores e todas as suas respectivas declarações nesse arquivo separado para que elas possam ser aplicadas ao documento que contiver um `<link>` para ele.

```
1   @charset "UTF-8";
2   /* Regra @charset caso haja problemas com acentuação */
3
4   body {
5       background-color: lightgoldenrodyellow;
6       color: chocolate;
7       font-size: 12pt;
8       font-family: Arial;
9   }
10
11  h1 {
12      color: brown;
13  }
14
15  a {
16      color: brown;
17      text-decoration: none;
18  }
```

Na **linha 1**, colocamos uma **regra** em CSS, que vai indicar a compatibilidade de codificação com o padrão UTF-8, assim como fizemos com o arquivo HTML5. Essa linha não é obrigatória e normalmente nem vai aparecer na maioria dos seus arquivos de configurações de estilo, mas caso você comece a ter problemas de compatibilidade com alguns caracteres, saiba que ela existe.

Na **linha 2**, adicionamos um **comentário** para facilitar a documentação do arquivo. Os comentários - assim como vimos em HTML - só servirão de explicação para que o desenvolvedor entenda o funcionamento de uma determinada linha ou trecho de código. O navegador não vai considerar nada que está entre os símbolos `/*` e `*/` em CSS.

Nas demais linhas, fizemos as configurações dos seletores, da mesma maneira que criamos com as outras duas técnicas apresentadas no capítulo.

TÁ CONFUSO? Se você não está entendendo claramente todas as declarações, não se preocupe! Nos próximos capítulos nós vamos nos aprofundar nelas. Foque agora nas técnicas de uso das CSS.

Qual técnica eu escolho pra usar?

Nesse capítulo, aprendemos as três técnicas de uso de folhas de estilo em cascata: **inline**, **interna** e **externa**. Mas em que situações devemos escolher cada um dos formatos?

Para falar com toda a sinceridade, a técnica **CSS inline style** deve ser evitada ao máximo. Ela acaba deixando seu código meio confuso, misturando a parte HTML e CSS em uma mesma linha. Mas se é para citar um momento específico em que podemos aplicar estilos inline em nossos códigos, use apenas em configurações muito pontuais e que não serão mais usadas em nenhum outro momento.

Já a técnica **CSS internal style** organiza melhor seu código, separando conteúdo e estilo em duas áreas bem definidas do seu documento. Use essa técnica quando for criar páginas isoladas com estilos próprios, que não serão replicados em outras páginas. Opte também por essa técnica apenas se a quantidade de configurações de estilo for pequena/média. Usar muitos seletores com muitas declarações vai fazer com que seu arquivo `.html` fique muito grande e seu conteúdo seja visualmente jogado lá pra baixo, dificultando manutenções futuras.

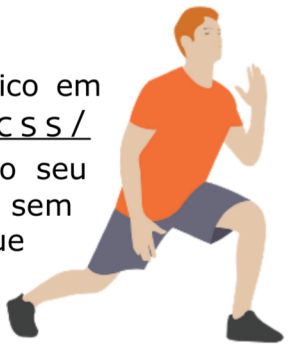
Por fim, opte sempre pela técnica **CSS external style** sempre que seu estilo for usado em várias páginas dentro do seu site. Usando a tag `<link>` em várias páginas, você pode compartilhar o mesmo estilo entre elas e não vai precisar ficar alterando vários arquivos quando o seu cliente solicitar uma pequena mudança no tom de uma determinada cor, por exemplo.

De forma resumida, guarde isso na sua cabeça:

CSS externo = use sempre que puder
CSS interno = use para pequenas configurações
CSS inline = procure evitar

Ainda é possível misturar as três técnicas, criando um CSS externo para as configurações globais, CSS interno para as configurações locais de um documento e CSS inline para pequenas configurações pontuais.

Hora de exercitar



Chegou a hora de acessar o endereço do nosso repositório público em https://gustavoguanabara.github.io/html-css/exercicios/ e executar os **exercícios 013**, **014** e **015** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra os **Módulos 1 e 2** do **Curso de HTML5 e CSS3**, completamente gravado com base nesse material.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Módulo 2 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dlUpEXkY1AyVLQGcpSgVF8s

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.

1. A sigla CSS significa:

A Content Style Setup
B Content Smart Sheets
C Cascading Style Sheets
D Cascading Setup Style

2. Ao configurar os estilos com a técnica inline, as propriedades CSS são colocadas em que local?

A Dentro da própria tag HTML, em um parâmetro chamado style
B Dentro da área <head> do documento HTML, na área delimitada por <style>
C Em um arquivo separado dentro da própria pasta onde o documento HTML está
D Dentro de uma pasta especial chamada /style que fica no diretório atual do site

3. O que aconteceu com configurações como <body bgcolor="blue"> e <font color="yellow"> que eram usadas em sites mais antigos?

A Continuam funcionando perfeitamente e podem ser usadas normalmente
B Continuam funcionando, mas não é nada recomendável pois as CSS são mais atuais
C Não funcionam mais de forma alguma
D Ainda são a única opção de mudar cores

4. Qual das opções a seguir é a única que configura corretamente a cor da letra de um título usando CSS inline?

🅐 <h1 color="blue">Título</h1>
🅑 <h1 style="blue">Título</h1>
🅒 <h1 style="font-color: blue">Título</h1>
🅓 <h1 style="color: blue">Título</h1>

5. Para usar a técnica de estilos internos, a tag <style> deve estar em que local?

🅐 logo depois da tag HTML que queremos configurar
🅑 dentro da área <head> do seu documento HTML
🅒 no final do arquivo, logo depois do </body>
🅓 em qualquer lugar do arquivo HTML

6. Em um mesmo documento HTML, utilizamos as técnicas de CSS interno e CSS inline para mudar a cor de um elemento e acabamos usando duas cores diferentes. Quais configurações vão prevalecer na versão final do site?

🅐 as configurações duplicadas feitas nas CSS inline vão prevalecer
🅑 as configurações duplicadas feitas nas CSS internas vão prevalecer
🅒 as duas configurações vão prevalecer e o texto será exibido duplicado com cores diferentes
🅓 o seu navegador vai decidir qual cor utilizar. Não existe uma regra para esses casos.

7. Para utilizar estilos CSS externos, devemos usar a tag _____ no documento HTML para fazer o carregamento:

🅐 <style>
🅑 <css>
🅒 <external>
🅓 <link>

8. A tag de carga de estilos externos deve estar localizada em que local do documento HTML?

🅐 deve ser a primeira linha do arquivo HTML
🅑 deve estar dentro da área <body> do documento
🅒 deve estar dentro da área <head> do documento
🅓 pode estar em qualquer lugar, desde que esteja dentro do documento HTML

9. Para que os arquivos externos em CSS tenham compatibilidade com caracteres acentuados, devemos adicionar a regra:

🅐 @UTF-8;
🅑 @charset "UTF-8";
🅒 @charset "PT-BR";
🅓 @PT-BR;

10. No fim das contas, vamos dar preferência à técnica de _____ sempre que for possível e vamos tentar evitar a técnica de _____ em nossos sites. A opção que apresenta os termos que completam as lacunas na ordem correta é:

🅐 CSS externo / CSS interno
🅑 CSS inline / CSS interno
🅒 CSS externo / CSS inline
🅓 CSS interno / CSS externo

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 🌎