

INTRODUÇÃO ÀS CSS

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 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.]

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.

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:

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 quasi! Eum saepe temporibus!

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.

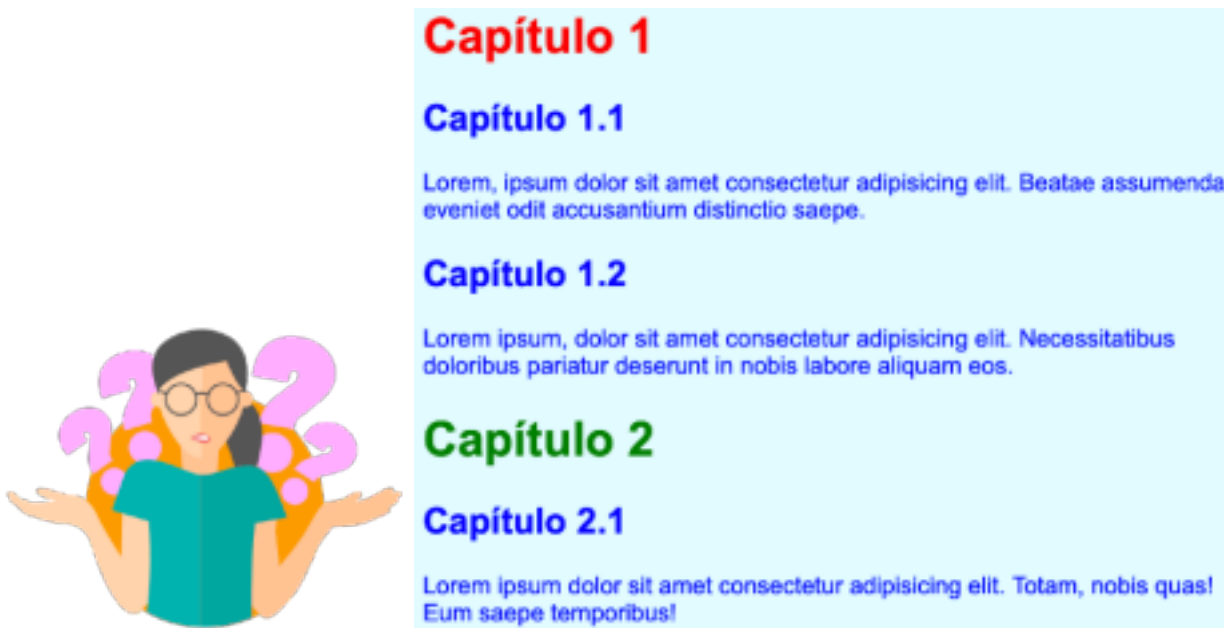
Estilizando de maneira mais interessante: CSS internal style

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,
6     initial-scale=1.0">
7   <title>Estilos pontuais</title>
8   <style>
9     body {
10       font-family: Arial, Helvetica, sans-serif;
11       background-color: lightcyan;
12       color: blue;
13     }
14     h1 {
15       color: green;
16     }
17   </style>
18 </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!

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



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.

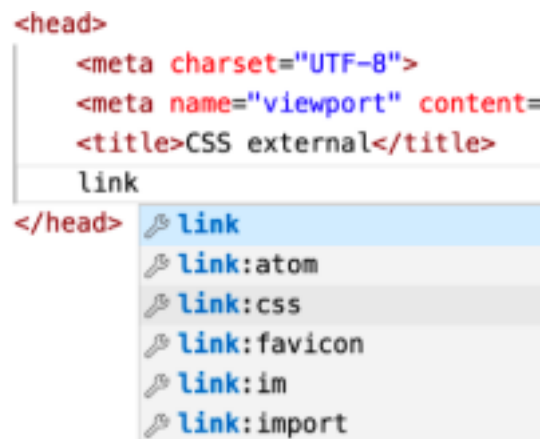
```
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS external</title>
  <link rel="stylesheet" href="style.css">
</head>
```

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.



```
<title>CSS external</title>
<link rel="stylesheet" href="style.css">
```

Siga o link (cmd + clique)

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.

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.

Qual técnica eu escolho pra usar?

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.

Usando Gradientes em CSS

Podemos gerar gradientes e aplicarmos a componentes visuais usando folhas de estilo. Vamos usar um exemplo simples no nosso exercício atual. Vá até o documento e modifique a declaração do nosso seletor body.

Pode parecer esquisito no início, mas um gradiente é considerado pelo navegador como se fosse uma imagem, por isso usamos a propriedade background-image na declaração CSS. A função linear-gradient é auto-explicativa e gera um gradiente linear angular. O primeiro parâmetro da função, indica o ângulo de inclinação de 90 graus (90deg) e as seguintes indicam as cores do degradê a ser criado. Você pode indicar quantas cores quiser e o navegador vai saber se virar pra gerar seu degradê personalizado. Experimente na sua casa outros valores de ângulo também, incluindo negativos (45deg, -90deg, 25deg,...) e note as diferenças.

Também é possível gerar os chamados gradientes radiais, que também são meio auto- explicativos. Veja o exemplo:

Altere o tipo de gradiente do body para usar o formato radial circular e veja o resultado. Você também pode personalizar ainda mais seu degradê colocando uma porcentagem ao lado da cor como red 10%, yellow 40%, green 50%.

Fonte, letra e família

Fontes Serifadas

Esta é a categoria mais clássica de fontes, surgida lá na época das prensas que eu citei no início do capítulo. Tipicamente, os caracteres serifados sempre foram aplicados em grandes blocos de textos impressos em papel e se aproveitam de uma característica da nossa percepção: nós nunca lemos as palavras letra por letra, e sim por um conjunto. As serifas têm a capacidade de guiar nossos olhos graças aos pequenos prolongamentos que elas criam e fazem as letras “se juntarem” em palavras. A seguir, vemos quatro exemplos de fontes serifadas:

Atualmente, não usamos fontes serifadas para apresentar textos longos na Web pois as tendências atuais nos levam a usar fontes um pouco mais leves visualmente. Porém, as fontes serifadas são bastante usadas em títulos, pois acabam chamando mais atenção por conta das características que citei.

APRENDA MAIS SOBRE FONTES: As classificações não param por aqui. Existem também sub-categorias para cada fonte serifada, como as *old style*, *transitional*, *didone*, *slab*, *clarendon* e *glyphic*, que apresentam características detalhadas para cada uma. Nesse material, nós vamos nos limitar apenas às classificações gerais por motivos práticos. Provavelmente o seu professor de *design* vai falar sobre o assunto de maneira mais aprofundada.

Fontes não Serifadas

Mais conhecidas por seu “nome chique” em Francês *sans-serif* (significa “sem serifa”), são fontes que, como você já pode imaginar, não apresentam serifas. As primeiras fontes dessa categoria surgiram em 1816, mas foram consideradas avançadas demais para a época. Anos depois, ressurgiram em versão melhorada e vieram pra ficar, principalmente para a Web. Isso acontece porque elas são ótimas para a exibição em telas/monitores pois transmitem a sensação de limpeza, clareza e organização. Veja a seguir alguns exemplos de fontes não serifadas:

A grande maioria dos textos que você está lendo nesse material desde o início do curso estão sendo escritos com uma fonte não serifada muito popular: a **Verdana**.

Fontes Monoespaçadas

Essa é uma das categorias de fontes que vieram derivadas das duas categorias que vimos anteriormente, por isso existem fontes monoespaçadas com e sem serifas. A principal diferença desse tipo de fonte é o espaço horizontal (largura) ocupado por cada letra. Na maioria das fontes, a letra *i* ocupa muito menos espaço lateral do que a letra **M**, não é? Não para as fontes monoespaçadas. Elas possuem a mesma largura para todas as letras.

A principal vantagem no uso desse tipo de fonte é facilitar ao máximo a leitura das palavras, principalmente aquelas que requerem que você as reproduza. Usamos muito esse tipo de fonte para representar comandos de linguagens de programação de computadores. Por isso, nós também costumamos chamá-las de fonte de terminal ou fonte de console.

Fontes Script

Também chamadas de fontes *handwriting*, são aquelas que tentam imitar a escrita humana. Seu uso deve ser bem controlado e jamais será aplicado a textos muito longos, pois causam cansaço visual e tornam-se difíceis de ler, como já provamos anteriormente no início do capítulo, dentro do item “As fontes falam”.

Fontes Display

Toda fonte que foge completamente das definições feitas pelas classificações acima são consideradas fontes *display*. São fontes com bastante efeitos visuais, enfeitadas e até mesmo curiosas. Também são chamadas de fontes comemorativas e algumas delas sequer representam letras, podendo ser desenhos de animais, objetos, pessoas, personagens de quadrinhos, etc.

Essas fontes também são recomendadas para criar títulos em destaque e devem ser evitadas para textos médios ou longos.

Como aplicar isso na prática?

Para configurar a família tipográfica que será aplicada a um determinado texto, usamos a propriedade `font-family` das CSS. Se indicarmos mais de uma família na sequência, estamos indicando ao navegador que dê preferência para a primeira. Caso ela não seja encontrada, tente a próxima. E essa estratégia se seguirá até a última, que geralmente é a família genérica `serif`, `sans-serif` ou `monospaced`.

Vamos fazer alguns exemplos aplicando famílias bem simples às nossas fontes. Vá até o seu exercício atual e aplique algumas declarações de `font-family` aos seletores de cada componente formatável do seu documento HTML.

SEQUÊNCIAS SEGURAS: Existem as chamadas sequências seguras para especificações de famílias de fontes. Para ver quais são elas, abra o Google e faça uma rápida busca por **CSS Web Safe Font Combinations**.

No código acima, seus títulos principais `<h1>` usarão preferencialmente a fonte **Franklin Gothic Medium**, uma fonte sem serifa e que tem seu espaço horizontal bem limitado. Porém, essa fonte geralmente não existe em smartphones, que possuem a fonte **Arial Narrow** que é bem parecida mas é menos densa. Caso nenhuma delas seja encontrada no aparelho do visitante, o navegador vai selecionar a fonte **Arial** normal. Em último caso, se tudo der errado, o sistema selecionará uma fonte genérica sem serifa.

Vamos falar de tamanhos

Além da família, podemos configurar tamanhos e estilos extras de qualquer componente textual do nosso documento HTML5.

Para especificar tamanho de fontes, existem várias medidas como **cm** (centímetros), **in** (polegadas), **pt** (pontos), **pc** (paicas), **px** (pixels), etc. Para tamanhos de fonte a serem exibidos na tela, o W3C recomenda o uso do **px** ou do **em**.

EU GOSTO DE USAR PT, MAS: A medida **pt** é aquela usada em editores de texto como o **Microsoft Word**. A recomendação oficial é de usar **pt** apenas para referenciar conteúdos que serão impressos.

A medida **em** é uma das que gera mais dúvida nos alunos. Ela é uma medida referencial em relação ao tamanho original da fonte. O tamanho padrão de uma fonte é geralmente **16px**, isso equivale a **1em**. A partir daí, podemos configurar o tamanho de um título, por exemplo, como sendo 2 vezes maior que a fonte padrão usando o valor **2em** para a propriedade.

No exemplo acima, todo título `<h2>` do nosso documento será 1.5x o tamanho padrão da fonte de referência.

MAIS INFORMAÇÃO: Para saber mais sobre as medidas suportadas pelas CSS, acesse o site oficial da W3C em: https://www.w3.org/Style/Examples/007/units.pt_BR.html

Outros estilos

Existem outras formatações muito usadas em CSS, que são as propriedades `font style` para aplicar o itálico e `font-weight` para aplicar o negrito, sem contudo existir o fator semântico discutido no **capítulo 08**.

O padrão para essas duas propriedades é o valor normal, mas podemos aplicar o valor itálico ao `font-style` usando `italic` (mais compatível) ou `oblique` (menos compatível). Já o negrito, pode ser aplicado por nomes como `lighter`, `bold` e `bolder` ou pelo peso numérico, como indicado na imagem.

As formatações de fontes são tão importantes e tão usadas em CSS, que existem “atalhos” para usá-las. São as chamadas *shorthands*.

Existe uma shorthand para fontes que é a propriedade `font`. No lugar de fazer várias configurações em múltiplas linhas, podemos simplificar tudo de maneira muito simples.

Podemos usar a shorthand `font` que vai simplificar tudo:

A ordem dos atributos de uma *shorthand* em CSS é importante. No caso da propriedade `font`, devemos informar, na ordem:

- `font-style`
- `font-variant`
- `font-weight`
- `font-size/line-height`
- `font-family`

Alinhamentos

Existem quatro tipos de alinhamento de textos:

`text-align: left;`

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

`text-align: right;`

`text-align: center;`

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

`text-align: justify;`

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.