01

Começando com CSS

Transcrição

Na aula anterior, vimos como criar o conteúdo de uma página usando a linguagem HTML. Porém, nossa página ainda está muito básica. Isso porque o navegador já sabe **o quê** queremos mostrar, mas não sabe **como** queremos mostrar, então mostra apenas o conteúdo, sem estilo algum.

Precisamos, então, ensiná-lo como queremos mostrar os elementos da nossa página. Imagine que queremos definir o tamanho da letra (fonte) como 16 pixels. O navegador nos permite especificar isso da seguinte forma:

```
font-size: 16px;
```

Essa forma de *declarar* como queremos mostrar um elemento é o que chamamos de linguagem **CSS**. Nessa linguagem, conseguimos configurar diversos aspectos da exibição dos elementos da nossa página.

Repare no formato da declaração: dizemos qual propriedade queremos mudar (`font-size`) e colocamos dois-pontos para separá-la do valor que lhe queremos dar, no nosso caso o tamanho da fonte (16 pixels). Por fim, colocamos um ponto-e-vírgula no final, para indicar que nossa declaração terminou.

Associando estilos a elementos

Na declaração que fizemos, dissemos que queríamos mudar o tamanho da fonte. Mas de qual elemento? Precisamos sempre **associar estilo a elemento**. Uma forma de fazer isso é usar o atributo `style` das tags do HTML. Por exemplo, podemos mudar a fonte do elemento `<h2>` da nossa página:

```
<h2 style="font-size: 16px;">Como trabalho</h2>
```

Apesar de funcionar, não é muito prático: se quisermos deixar todos os subtítulos da página com a mesma aparência, teremos que repetir esse código em todas as tags `<h2>` . Então, melhor que isso, podemos usar a tag `<style>` , colocando o estilo desejado dentro dela:

```
<style>
    font-size: 16px;
</style>
```

E onde colocamos a tag `<style>` ? No lugar onde vão as tags com as informações *para o navegador*: dentro da tag `<head>` .

Está faltando algo na declaração acima, não? Anteriormente, dissemos que todo estilo deveria ser associado a um elemento da página. A qual elemento a declaração acima está associada? Precisamos fazer associação **selecionando** um elemento da página para essa regra. É aí que entram os **seletores** da linguagem CSS.

```
<style>
    h2 {
        font-size: 16px;
```

```
    }
</style>
```

No exemplo acima, estamos **selecionando** os elementos `<h2>` da página e aplicando o estilo que queríamos, o tamanho de fonte. Repare que precisamos abrir chaves logo após o seletor e fechar chaves ao final do estilo. Podemos colocar mais regras dentro das chaves, se quisermos. Por exemplo, podemos alterar a fonte usada no subtitulo para "Arial":

```
<style>
    h2 {
        font-family: "Arial";
        font-size: 16px;
   }
</style>
```

Colocando o CSS no seu devido lugar

Com a tag `<style>` , conseguimos não só evitar repetição como também deixar nosso código mais organizado: tudo que é relativo à aparência da página fica agrupado num só lugar, facilitando a manutenção da página.

Mas ainda temos um problema: nosso site não vai ter uma página só. Essas outras páginas muito provavelmente terão uma cara parecida com a página que estamos desenvolvendo agora. Então o CSS também será bem parecido, com bastante coisa repetida. Pior: se quisermos alterar alguma característica visual do site inteiro, teremos que fazer a alteração em **todas as páginas** do site. Muito trabalho com muita chance de erro!

Como podemos evitar essa repetição? **Colocando o CSS num arquivo separado**! Podemos, então, criar um arquivo chamado `site.css` com o conteúdo que tínhamos colocado na tag `<style>` :

```
h2 {
    font-family: "Arial";
    font-size: 16px;
}
```

Agora, no arquivo HTML, precisamos dizer para o navegador que esse arquivo CSS precisa ser carregado. Para isso, usamos a tag `<link>` . Lembra dela?

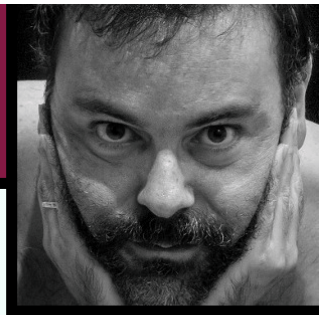
```
<link rel="stylesheet" href="site.css">
```

Assim como a tag `<link>` que criamos para escolher o ícone, essa tag também vai dentro da tag `<head>` . E agora, quando o navegador encontrar essa tag, ele vai baixar o arquivo `site.css` e, por causa do atributo `rel="stylesheet"` , ele sabe que esse arquivo tem o estilo da página.

Um pouco de cor

Agora que sabemos onde escrever o CSS, podemos começar a especificar o estilo que queremos que nossa página tenha, para chegarmos no resultado mostrado na imagem abaixo.

SOBRE MIM



JOÃO DA SILVA

home
portfolio
sobre mim
blog
contato

Moro em São Paulo mas atendo clientes do mundo todo. Sou conhecido por fazer produtos de *qualidade*, *durabilidade* e que *agregam valor* para meus clientes.

Trabalho usando a web como plataforma, ou seja, respiro HTML5, CSS3 e JavaScript (ou melhor: ECMASCript). Crio sites para todos, seguindo as principais diretivas de **acessibilidade, responsividade e web semântica**, sem descuidar da qualidade de código.

Como trabalho

Satisfazer meus clientes é prioridade. Para isso, garanto um processo de desenvolvimento altamente interativo, baseado em feedback contínuo. **Não trabalho com escopo fechado**: o cliente é que decide quando o produto está pronto.

Também não trabalho com prazos fechados: **qualidade é importante demais para ser sacrificada**.

João é o melhor desenvolvedor front-end com quem já trabalhei. Muito eficiente e muito capaz. Recomendo sem dúvidas!

José Souza, Fiat

Experiência

João domina as tecnologias como ninguém. Eu apresentava um problema, ele tinha na ponta da língua a solução mais adequada com as tecnologias mais recentes.

Manoel Santos, Petrobrás

Já desenvolvi projetos para grandes empresas como BMW, UOL e IBM. Neles, o foco principal era entregar uma experiência imersiva e impactante para o usuário final sem descuidar do desempenho e da acessibilidade da página.

Também já fui contratado para transformar grandes portais, como Terra e G1, em páginas responsivas. Fui responsável por renovar o layout, reorganizar o conteúdo e re-escrever o código de forma mais reaproveitável.

Comunidade

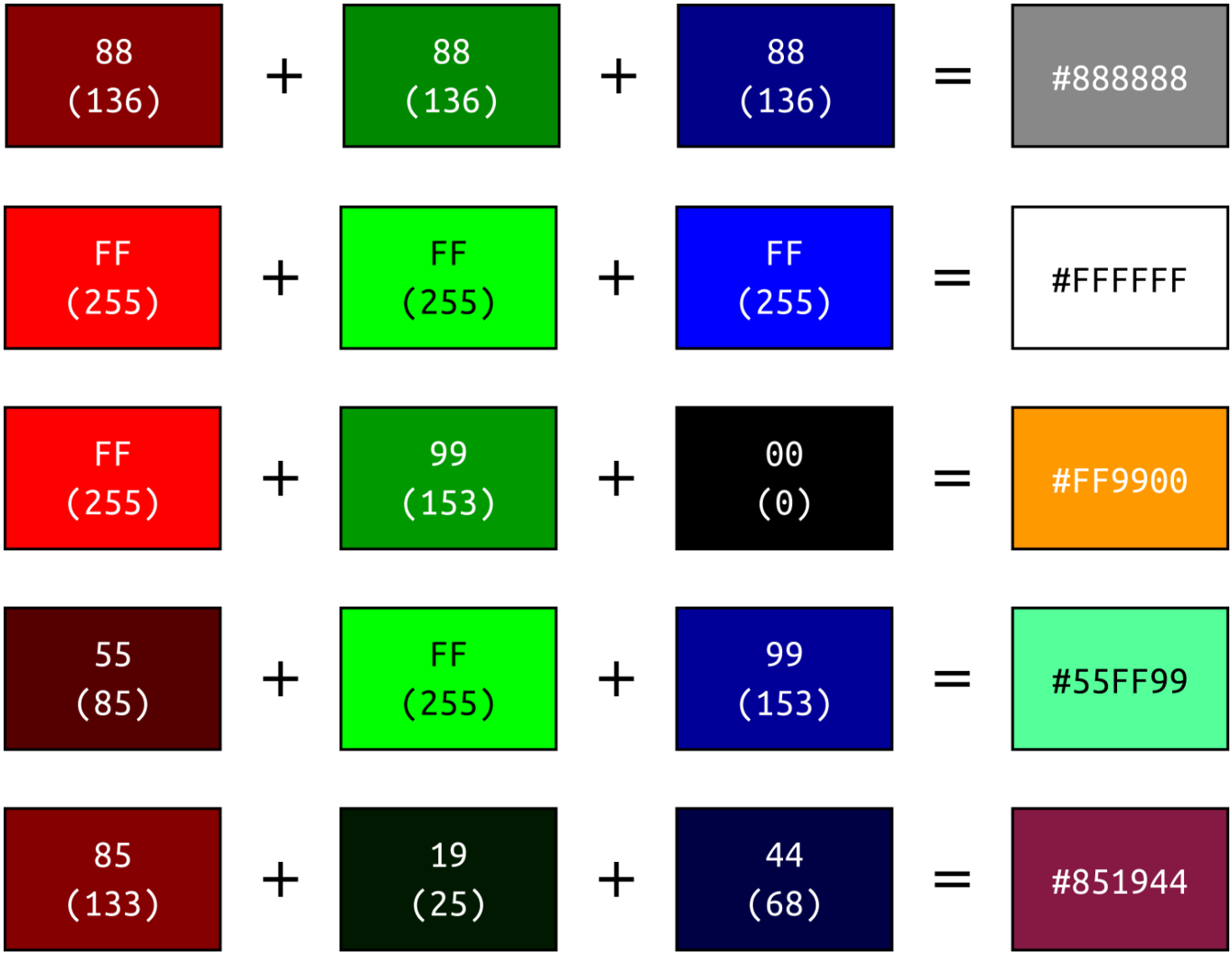
Procuro repassar meu conhecimento para a comunidade. Para isso, já dei diversas palestras e mantenho um blog.

Vamos começar ajustando as cores. O título tem um fundo roxo. Como alterar a cor de fundo? Como queremos mexer na *cor de fundo* do título, a propriedade que vamos alterar chama-se `background-color`. Faz sentido, não? A maior parte das propriedades que podemos alterar no CSS tem um nome bem explicativo. Veremos mais exemplos adiante.

Nosso título está dentro de uma tag `<h1>`. Então precisamos *selecioná-lo* para alterar a cor de fundo. Teremos algo assim no CSS, então:

```
h1 {
    background-color: ????;
}
```

A questão que resta é: como dizer que queremos a cor *roxa*? Aliás, qual tom de roxo queremos? Existem vários "roxos". E também existem várias formas de identificar uma cor no CSS. A mais popular é especificar uma mistura das cores primárias **vermelho**, **verde** e **azul** para chegar à cor que queremos, dizendo o "quanto" queremos de cada cor numa escala de 0 a 255. Por exemplo, nosso roxo tem 133 de vermelho, 25 de verde e 68 de azul.



Sabendo disso, basta escrever essa mistura no CSS. Existem duas formas para isso. A mais popular e compacta é usar a notação *hexadecimal* para representar essas quantidades. Nessa notação, "0" continua sendo "0", mas "10" vira "A", "15" vira a letra "F" e "16" vira "10". Seguindo nessa lógica, "133" vira "85", "25" vira "19" e "68" vira "44". Juntando as cores na ordem *vermelho*, *verde* e *azul* chegamos no formato usado no CSS:

```
h1 {
    background-color: #851944;
}
```

Repare que colocamos um cerquilha (#) na frente dos números. Ele faz parte desse formato de cor do CSS e serve para deixar claro para quem ler o código que aquela sequência de números e letras é uma cor.

Outra forma de especificar essa mistura é usando a forma abaixo:

```
h1 {
    background-color: rgb(133, 25, 68);
}
```

Precisamos mudar também a cor da letra do nosso título para branco. Para isso, usamos a propriedade `color`. Nosso CSS fica, então:

```
h1 {
    background-color: rgb(133, 25, 68);
    color: rgb(255, 255, 255);
}
```

Ou então, se você preferir a notação hexadecimal:

```
h1 {
    background-color: #851944;
    color: #FFFFFF;
}
```

Algumas cores não precisam de tanta precisão assim (de 0 a 255) para serem especificadas; dezesseis níveis de cor (de 0 a 15) já bastariam, economizando um pouco de código para nós. Pois bem: temos uma notação *hexadecimal compacta* no CSS também! Podemos escrever a cor branca usando essa sintaxe:

```
h1 {
    background-color: #851944;
    color: #FFF;
}
```

A cor roxa, por ser mais específica, precisa ser especificada na notação hexadecimal normal. Porém, se a cor fosse `#882244`, poderíamos escrevê-la como `#824`.

Uma última cor que precisamos ajustar é a cor de fundo da página. É uma cor quase branca: `#F2FFFC`. Como fazer para mudar a cor da página toda? Mudando a cor da tag que contém a página toda, que é a tag `<body>`:

```
body {
    background-color: #F2FFFC;
}
```

Ajustando o texto

Uma outra coisa que precisamos ajustar no cabeçalho é o alinhamento do texto. Queremos deixar o título principal centralizado na tela. Para isso, usamos a propriedade `text-align` com o valor `center`:

```
h1 {
    background-color: #851944;
    color: #FFF;
    text-align: center;
}
```

Precisamos ajustar o alinhamento dos parágrafos de texto, também, deixando as linhas todas com a mesma largura. Podemos, então, *selecionar os parágrafos* e usar novamente a propriedade `text-align`, dessa vez com o valor `justify`:

```
p {
    text-align: justify;
}
```

Outra coisa que precisamos ajustar no nosso texto é o tipo da fonte usada na página. Dependendo do navegador e do sistema que você está usar, você verá o texto numa fonte diferente.

Como trabalho

Como trabalho

Se você olhar com atenção a imagem que estamos usando como referência para estilizar nossa página, verá que estamos usando duas fontes diferentes: uma para o título e os subtítulos (Arial) e outra para o texto (Times New Roman). Para mudar as fontes para esses elementos da página, podemos usar a propriedade `font-family`, passando o nome da fonte que queremos usar para aquele elemento:

```
h1 {
    background-color: #851944;
    color: #FFF;
    text-align: center;
    font-family: "Arial";
}

h2 {
    font-family: "Arial";
}

p {
    text-align: justify;
    font-family: "Times New Roman";
}
```

Podemos melhorar um pouco o código CSS, tirando a repetição do nome da fonte. Para isso, podemos selecionar o título e os subtítulos **ao mesmo tempo**. Conseguimos fazer isso separando os nomes das tags por vírgula no nosso seletor:

```
h1, h2 {
    font-family: "Arial";
}
```

Fazendo isso, podemos retirar a declaração de dentro do seletor `h1`, e tanto os `<h1>` como os `<h2>` da página serão exibidos com a fonte Arial.

Só precisamos tomar cuidado com uma coisa: nossa página será exibida nos mais diversos navegadores e sistemas. Será que todos eles têm essas fontes que escolhemos? Provavelmente não. E, se o navegador não encontra a fonte que pedimos, ele vai exibir nossa página com as fontes que ele quiser, possivelmente deixando nossa página bem estranha.

Felizmente, podemos controlar um pouco qual será a fonte usada caso a fonte que escolhemos não seja encontrada, bastando especificar uma fonte alternativa. Por exemplo:

```
p {
    font-family: "Times New Roman", "Baskerville";
}
```

Com essa declaração, dizemos para o navegador usar a fonte Baskerville se a fonte Times New Roman não estiver disponível. Agora, será que especificar outra fonte é suficiente? E se a fonte Baskerville não estiver disponível também? Podemos explorar mais e mais possibilidades, ou então abrir mão do controle total da fonte: em vez de escolher a fonte específica, escolher apenas o *tipo* da fonte que queremos usar. As fontes Times New Roman e Baskerville, por exemplo, são fontes *serifadas*. Por outro lado, as fontes Arial e Verdana são fontes *sem serifa*. A diferença pode ser vista na imagem abaixo: as fontes serifadas possuem traços em algumas pontas.

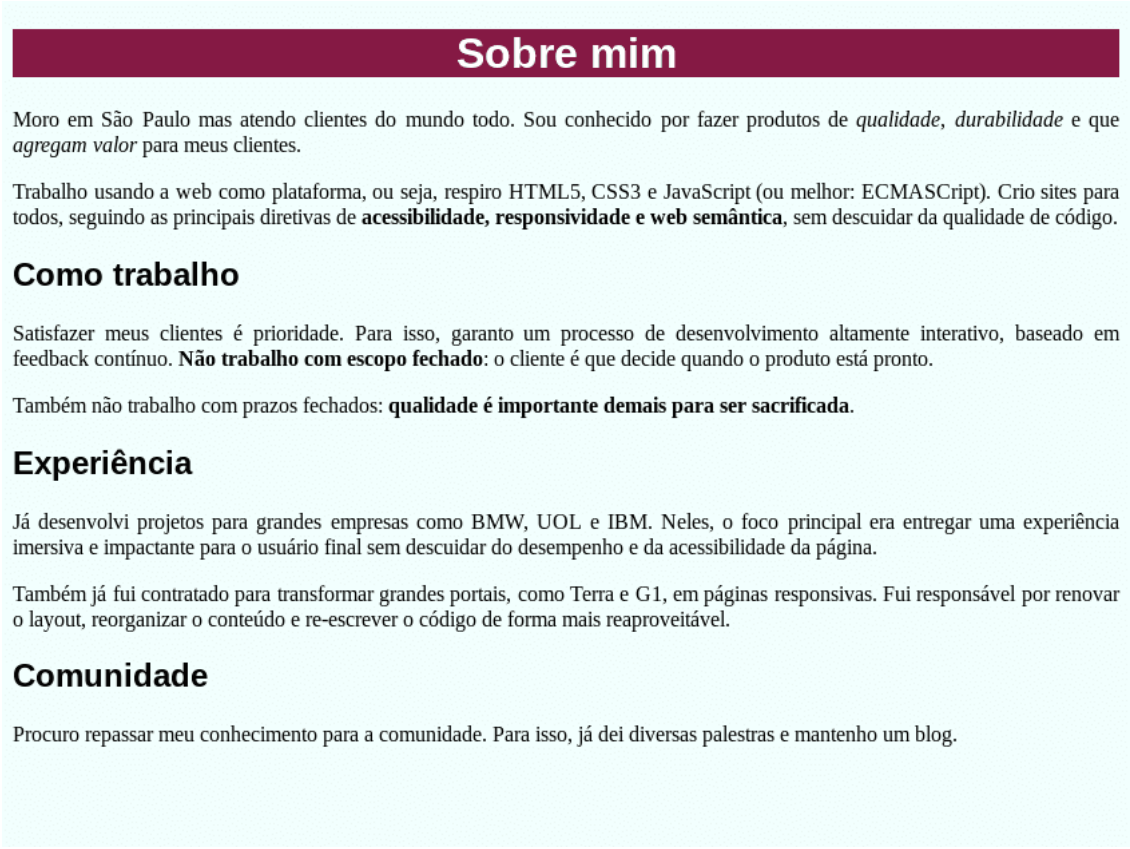
Então podemos dizer para o navegador: se a fonte Times New Roman não estiver disponível, use uma fonte qualquer, desde que ela seja serifada; fazemos isso passando o valor `serif` para o `font-family`. Podemos fazer algo semelhante para os títulos e subtítulos, mas dizendo que queremos uma fonte sem serifa com o valor `sans-serif`.

```
h1, h2 {
    font-family: "Arial", sans-serif;
}

p {
    font-family: "Times New Roman", serif;
}
```

Próximos passos

Com o que já vimos, nossa página deve ficar assim:



Sobre mim

Moro em São Paulo mas atendo clientes do mundo todo. Sou conhecido por fazer produtos de *qualidade*, *durabilidade* e que *agregam valor* para meus clientes.

Trabalho usando a web como plataforma, ou seja, respiro HTML5, CSS3 e JavaScript (ou melhor: ECMASCript). Crio sites para todos, seguindo as principais diretivas de **acessibilidade, responsividade e web semântica**, sem descuidar da qualidade de código.

Como trabalho

Satisfazer meus clientes é prioridade. Para isso, garanto um processo de desenvolvimento altamente interativo, baseado em feedback contínuo. **Não trabalho com escopo fechado**: o cliente é que decide quando o produto está pronto.

Também não trabalho com prazos fechados: **qualidade é importante demais para ser sacrificada**.

Experiência

Já desenvolvi projetos para grandes empresas como BMW, UOL e IBM. Neles, o foco principal era entregar uma experiência imersiva e impactante para o usuário final sem descuidar do desempenho e da acessibilidade da página.

Também já fui contratado para transformar grandes portais, como Terra e G1, em páginas responsivas. Fui responsável por renovar o layout, reorganizar o conteúdo e re-escrever o código de forma mais reaproveitável.

Comunidade

Procuro repassar meu conhecimento para a comunidade. Para isso, já dei diversas palestras e mantenho um blog.

Nosso texto começa a ter um pouco mais da aparência que queremos no final. No entanto, o espaçamento entre os elementos da página ainda não está legal. Além disso, antes de podermos nos preocupar com a foto e a barra lateral, faltam alguns elementos no próprio texto. No próximo capítulo cuidaremos desses detalhes.