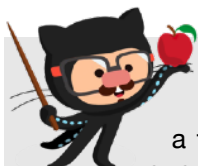


DevWeb

Capítulo 21

Conteúdos em Tabelas

Ao construir os conteúdos que vão compor o seu site, provavelmente você vai se deparar com a necessidade de criar uma ou mais tabelas. A HTML tem uma série de *tags* para a organização tabular de dados e precisamos dominá-las. Este capítulo vai te mostrar muitas peculiaridades das tabelas e suas várias possibilidades de personalização.

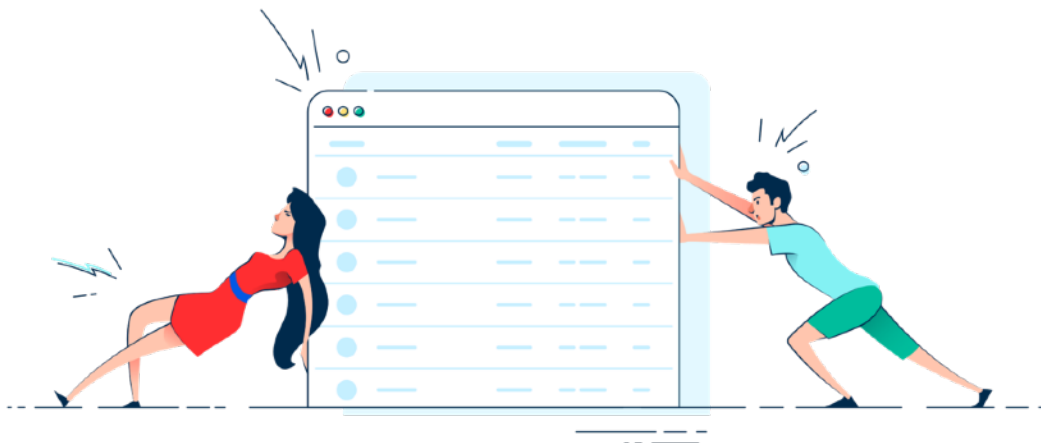


Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. 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.



Tableless: tabelas são para apresentar conteúdos, não para construir estruturas de sites

Antes de começar a estudar as tabelas, vamos retirar um assunto da frente. Houve uma época em que os sites eram desenvolvidos como se fossem tabelas. Foi um tempo sombrio e que ficou no passado!



Com a evolução das CSS e com o surgimento da HTML na sua versão 5, os sites podem ser criados de forma mais organizada e dinâmica e não dependem mais das tabelas para existir, um movimento do design que ficou conhecido como *tableless layout* (traçado sem tabelas).

Mas não vamos demonizar as tabelas! Elas são boas e importantes, só não precisam ser usadas para criar a estrutura do seu site. Deixe as tabelas para o seu objetivo principal: apresentar dados de forma tabular. Só isso!

Nossa primeira tabela

Vamos começar com uma tabela simples para podermos entender a estrutura básica dos dados tabulados:

População das Unidades Federativas	
Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791
Rio de Janeiro	17.264.943
Bahia	14.873.064
Paraná	11.433.957
Total	110.659.804



FONTE DOS DADOS: A tabela anterior foi construída com base nos dados disponíveis em Junho de 2020 no site da Wikipedia, no link a seguir.

https://pt.wikipedia.org/wiki/Lista_de_unidades_federativas_do_Brasil_por_população

Vamos começar analisando a anatomia básica da tabela, demarcando algumas áreas importantes:

População das Unidades Federativas	
Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791
Rio de Janeiro	17.264.943
Bahia	14.873.064
Paraná	11.433.957
Total	110.659.804

Em primeiro lugar, temos o **título** (*caption*) da tabela, marcado com o **número 1**. Em seguida, temos o **cabeçalho** (*thead*) da tabela com o **número 2**, que serve para identificar cada coluna de dados. Já o **corpo** (*tbody*) da tabela apresenta todos os dados de forma tabular na área demarcada com o **número 3**. Por fim, temos uma área final que totaliza todos os dados, que é o **rodapé** (*tfoot*) da tabela, delimitado pela área de **número 4**.

O código base para criar a tabela acima será:

```
<table>
  <caption>
    <!-- título da tabela -->
  </caption>
  <thead>
    <!-- cabeçalho da tabela -->
  </thead>
  <tbody>
    <!-- corpo da tabela -->
  </tbody>
  <tfoot>
    <!-- rodapé da tabela -->
  </tfoot>
</table>
```

Compare o código HTML com a estrutura da imagem com as quatro áreas delimitadas e tudo ficará bem claro. Esse será o "esqueleto" da nossa tabela que vai receber os dados.

Agora vamos nos focar na estrutura das linhas de cada área:

População das Unidades Federativas	
Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791

Em primeiro lugar, cada área é dividida em **linhas** (*table rows - tr*), que é a área marcada com o **número 5** e ocupa a largura total da tabela. Dentro das linhas, temos **células de dados** (*table data - td*) ou **células de cabeçalho** (*table header - th*). Sendo assim, para fazer o código relativo ao trecho que marcamos na imagem acima, vamos escrever:

```
<tbody> <!-- corpo da tabela -->
  <tr> <!-- primeira linha -->
    <td>São Paulo</td> <!-- primeira célula da primeira linha -->
    <td>45.919.049</td> <!-- segunda célula da primeira linha -->
  </tr>
  <tr> <!-- segunda linha-->
    <td>Minas Gerais</td> <!-- primeira célula da segunda linha -->
    <td>21.168.791</td> <!-- segunda célula da segunda linha -->
  </tr>
  ...
</tbody>
```



QUANDO USAR <TD> OU <TH>? Uma dúvida muito comum é quando usar cada uma dessas *tags*. De forma resumida, vamos usar o <td> quando a célula contiver um dado (**São Paulo**, por exemplo) e usaremos o <th> quando a célula contiver um identificador de dados (**Cidade**, por exemplo). Porém, ambos estarão dentro de linhas <tr>.

Minha tabela ficou feia, e agora?

Antes de falar em aplicar folhas de estilo a tabelas, precisamos entender que a hierarquia dos componentes de uma tabela é algo **primordial**! Qualquer erro na hierarquia, qualquer *tag* no lugar errado, vai causar a desorganização dos dados. Dessa maneira, se por acaso os dados não estiverem aparecendo na organização correta, é sinal de que existe um erro na estrutura da tabela.

Se a sua tabela estiver sendo apresentada na tela de forma organizada, mas sem bordas, sem cores e sem graça, chegou a hora de criar as configurações de CSS.

A primeira configuração será configurar tamanhos e bordas:

```
table {  
    width: 600px;  
    border: 1px solid black;  
    border-collapse: separate;  
}  
  
td, th {  
    border: 1px solid black;  
}
```

O resultado será uma tabela com as células delimitadas por bordas finas. Note que existe um pequeno espaço entre as células, que talvez incomode um pouco. Essa distância pode ser personalizada através da propriedade `border-spacing`.

Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791
Rio de Janeiro	17.264.943
Bahia	14.873.064
Paraná	11.433.957
Total	110.659.804

Para isso, mude o parâmetro `border-collapse` do valor `separate` para o valor `collapse`. A tabela vai mudar um pouco:

Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791
Rio de Janeiro	17.264.943
Bahia	14.873.064
Paraná	11.433.957
Total	110.659.804

Agora vamos adicionar um pequeno padding (8px) às configurações de `td` e `th` que fizemos anteriormente e adicionar mais alguns seletores bem simples e intuitivos:

```
caption {  
    font-weight: bolder;  
    font-size: 1.5em;  
    text-align: center;  
    background-color: #ccc;  
    padding: 5px;  
}
```

```
thead {
  background-color: #636363;
  color: white;
}

tfoot {
  background-color: #636363;
  color: white;
  font-weight: bolder;
}
```

Todas as configurações acima já foram explicadas nos capítulos anteriores. Se surgir alguma dúvida, consulte o material impresso ou assista os vídeos relativos a cada capítulo. Como eu sempre digo, não tente fazer tudo na correria ou o resultado do seu aprendizado pode ser bem frustrante.

O resultado visual do código CSS acima aplicado à nossa tabela será:

População das Unidades Federativas	
Cidade	População
São Paulo	45.919.049
Minas Gerais	21.168.791
Rio de Janeiro	17.264.943
Bahia	14.873.064
Paraná	11.433.957
Total	110.659.804

Como criar o efeito zebrado?

Nunca ouviu falar no "efeito zebrado" em tabelas? Pois saiba que o conceito é muito simples e consiste em intercalar as cores das células pares e ímpares, criando um efeito "listrado" (tá aí a referência a uma zebra) que facilita na leitura dos dados em tabelas muito grandes e com muito conteúdo.



Para obter esse resultado na tabela que estamos criando, vamos acrescentar os seguintes seletores ao estilo:

```
tbody > tr:nth-child(even) {
  background-color: #dddddd;
}

tbody > tr:nth-child(odd) {
  background-color: #ffffff;
}
```

A pseudo-klasse `nth-child()` vai permitir selecionar as linhas **pares** (*even*) ou as linhas **ímpares** (*odd*), configurando uma cor para cada uma delas. No exemplo acima, inclusive, se o fundo da página já está branco, o segundo seletor (para linhas ímpares) torna-se desnecessário.

Mudando o alcance das células

É possível modificar o alcance das células para obter resultados ainda mais organizados. Isso é feito através das propriedades HTML nas células: `colspan` e `rowspan`. Dê uma boa olhada na tabela a seguir:

Grupo	Nomes	Filmes Favoritos		
Mulheres	Ana Maria Santos	Alien	Rambo	Vingadores
	Beatriz Souza	Hulk	Inception	Batman
	Cláudia Melo	Oblivion	Matrix	Big Hero
Homens	Bruno Mendonça	Intocáveis	Amnésia	Gladiador
	Daniel Lourenço	Wall-E	Oldboy	Dangal
	Fabiano Mota	Star Wars 5	Taxi Driver	Toy Story

Note que, no cabeçalho da tabela, temos uma célula para "*Filmes Favoritos*" que está ocupando o espaço de três colunas. Enquanto isso, na lateral esquerda da tabela, as células com as palavras "*Mulheres*" e "*Homens*", que estão ocupando três linhas cada. Para efetuar essas mesclagens, basta usar as propriedades `colspan` e `rowspan`, respectivamente.

```
<table>
  <thead>
    <tr>
      <th>Grupo</th>
      <th>Nomes</th>
      <th colspan="3">Filmes Favoritos</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="3">Mulheres</td>
      <td>Ana Maria Santos</td>
      <td>Alien</td>
      <td>Rambo</td>
      <td>Vingadores</td>
    </tr>
    <tr>
      <td>Beatriz Souza</td>
      <td>Hulk</td>
      <td>Inception</td>
      <td>Batman</td>
    </tr>
    <tr>
```

Para treinar um pouco mais, crie aí no seu ambiente de testes as seguintes tabelas: a primeira contendo números e com algumas mesclagens pontuais.

1	2	3	4
5	6		7
8			9
10	11		12
	13	14	15

A segunda tabela vai conter letras e vai fazer as seguintes mesclagens:

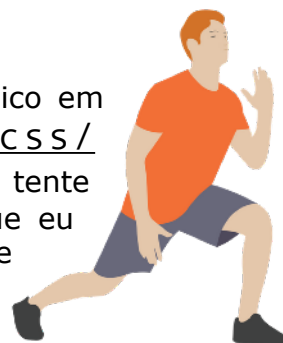
A			B	
C	D	E		F
	G	H		I
	J	L		M
N				

Se você conseguiu fazer as tabelas acima, vamos ao desafio final e reproduza a seguinte tabela:

Área	Disciplinas	Notas		Média
		Nota 1	Nota 2	
Exatas	Matemática	0.0	0.0	0.0
	Física	0.0	0.0	0.0
	Química	0.0	0.0	0.0
	Biologia	0.0	0.0	0.0
Média de Exatas				0.0
Humanas	História	0.0	0.0	0.0
	Geografia	0.0	0.0	0.0
Média de Humanas				0.0

Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/desafios/> e executar o **desafio 010** 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**.



Eu já falei sobre isso no YouTube?

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 tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.



Curso em Vídeo: https://www.youtube.com/playlist?list=PLHz_AreHm4dlAnJ_jJtV29RFxnPHDuk9o

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.



Questão...

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Questão...

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Questão...

- ☐ A
- ☐ B
- ☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D

Questão...

☐ A

☐ B

☐ C

☐ D