SOBRE ESTA APOSTILA

Esta apostila da Caelum visa ensinar de uma maneira elegante, mostrando apenas o que é necessário e quando é necessário, no momento certo, poupando o leitor de assuntos que não costumam ser de seu interesse em determinadas fases do aprendizado.

A Caelum espera que você aproveite esse material. Todos os comentários, críticas e sugestões serão muito bem-vindos.

Essa apostila é constantemente atualizada e disponibilizada no site da Caelum. Sempre consulte o site para novas versões e, ao invés de anexar o PDF para enviar a um amigo, indique o site para que ele possa sempre baixar as últimas versões. Você pode conferir o código de versão da apostila logo no ​nal do índice.

Baixe sempre a versão mais nova em: www.caelum.com.br/apostilas

Esse material é parte integrante do treinamento Desenvolvimento Web com HTML, CSS e JavaScript e distribuído gratuitamente exclusivamente pelo site da Caelum. Todos os direitos são reservados à Caelum. A distribuição, cópia, revenda e utilização para ministrar treinamentos são absolutamente vedadas. Para uso comercial deste material, por favor, consulte a Caelum previamente.

Caelum Sumário**Sumário**

**1 A estrutura dos arquivos de um projeto**

**1**

1.1 Web site ou aplicação Web? 1 1.2 Editores e IDEs 2

**2 Exercício: Baixando os arquivos do curso**

**4**

2.1 Objetivo 4 2.2 Passo a passo com código 5

**7**

**3 Introdução ao HTML**

3.1 Exibindo informações na Web 7 3.2 Sintaxe do HTML 12 3.3 Tags HTML 13 3.4 Imagens 14 3.5 Primeira página 15

**17**

**4 Exercício: Primeiros passos com HTML**

4.1 Objetivo 17 4.2 Passo a passo com código 18

**5 Estrutura de um documento HTML**

**21**

5.1 A tag <html> 21 5.2 A tag <head> 22 5.3 A tag <body> 22 5.4 A instrução DOCTYPE 23

**6 Exercício: A Especificação HTML**

**25**

6.1 Objetivo 25 6.2 Passo a passo com código 26

**7 Estilizando com CSS**

**29**

7.1 Sintaxe e inclusão de CSS 29 7.2 Propriedades tipográficas e fontes 32

Sumário Caelum

7.3 Alinhamento e decoração de texto 33 7.4 Imagem de fundo 34 7.5 Bordas 34 7.6 Cores na Web 35

**8 Exercício: Mudando o visual da nossa página**

**37**

8.1 Objetivo 37 8.2 Passo a passo com código 38

**9 Espaçamentos e dimensões**

**41**

9.1 Dimensões 41 9.2 Espaçamentos 42

**45**

**10 Exercício: Trabalhando com tamanhos e espaçamento**

10.1 Objetivo 45 10.2 Passo a passo com código 46

**11 Listas HTML**

**48**

11.1 Listas de definição 48 11.2 Links no HTML 49

**12 Exercício: Usando links e listas**

**51**

12.1 Objetivo 51 12.2 Passo a passo com código 52

**13 Seletores mais específicos e herança**

**55**

13.1 Para saber mais: o valor inherit 59

**61**

**14 Exercício: Selecionando elementos mais específicos e herança**

14.1 Objetivo 61 14.2 Passo a passo com código 62

**66**

**15 Desacoplamento com classes**

**68**

**16 Exercício: Usando classes para evitar acoplamento de estilos**

16.1 Objetivo 68 16.2 Passo a passo com código 69

**73**

**17 Elementos estruturais e semântica dos elementos**

**74**

**18 Exercício: Semantica em primeiro lugar**

18.1 Objetivo 74 18.2 Passo a passo com código 75

Caelum Sumário

**19 Conhecendo padrões de CSS**

**78**

19.1 Tipos de display 79

**20 Exercício: Treinando display e nomenclatura de classes**

**83**

20.1 Objetivo 83 20.2 Passo a passo com código 83

**21 Exercício: Um pouco sobre estilização com blocos**

**87**

21.1 Objetivo 87 21.2 Passo a passo com código 87

**22 Exercício: Modificadores**

**92**

22.1 Objetivo 92 22.2 Passo a passo com código 92

**23 Unidades relativas com EM e REM 24 Exercício: Medidas relativas em e rem**

**97 99**

24.1 Objetivo 99 24.2 Passo a passo com código 99

**25 Site mobile ou mesmo site?**

**104**

25.1 CSS media types 105 25.2 CSS3 media queries 106 25.3 Viewport 107 25.4 Responsive Web Design 108 25.5 Mobile-first 109

**110**

**26 Exercício: Responsividade com viewport e unidades relativas**

26.1 Objetivo 110 26.2 Passo a passo com código 110

**27 Exercício: Responsividade com `@media`**

**115**

27.1 Objetivo 115 27.2 Passo a passo com código 117

**28 O processo de desenvolvimento de uma tela**

**121**

28.1 Analisando o Layout 122

**29 Exercício: Estrutura da página principal e o cabeçalho**

**125**

29.1 Objetivo 125 29.2 Passo a passo com código 127 30.1 CSS Reset 132

Sumário Caelum

30.2 Fontes Próprias 133 30.3 Modularizando Componentes com CSS Isolados 134

**30 Exercício: Começando a estilizar o `header` da home**

**136**

31.1 Objetivo 136 31.2 Passo a passo com código 140

**31 Progressive Enhancement**

**143**

32.1 Condições, opções, limitações e restrições 143

**32 Exercício: Progressive Enhancement**

**146**

33.1 Objetivo 146 33.2 Passo a passo com código 147

**33 Display Flex**

**150**

34.1 Flex Container 150

**34 Exercício: Novos Layouts com Flexbox**

**152**

35.1 Objetivo 152 35.2 Passo a passo com código 153

**35 Exercício: Novos Layouts com Flexbox B**

**154**

36.1 Objetivo 154 36.2 Passo a passo com código 154

**36 Responsividade e Fallback**

**37 Exercício: Responsividade e fallback**

**156 158**

38.1 Objetivo 158 38.2 Passo a passo com código 159

**38 Exercício: Desafio: crie o HTML do rodapé**

**162**

39.1 Objetivo 162 39.2 Passo a passo com código 163

**39 Exercício: Conteúdo Rodapé**

**166**

40.1 Objetivo 166 40.2 Passo a passo com código 167

**171**

**40 Exercício: Espaçando o conteúdo do rodapé - os detalhes nas entrelinhas e margens**

41.1 Objetivo 171 41.2 Passo a passo com código 172

**41 Exercício: O formulário de newsletter do rodapé**

**176**

Caelum Sumário

42.1 Objetivo 176 42.2 Passo a passo com código 177

**42 Exercício: Embaralhando os itens do rodapé em telas maiores com Flexbox 181**

43.1 Objetivo 181 43.2 Passo a passo com código 182

**187**

**43 Exercício: Rodapé em telas grandes sem sobrescrever tudo o que veio antes**

44.1 Objetivo 187 44.2 Passo a passo com código 187

**44 Exercício: O componente container**

**190**

45.1 Objetivo 190 45.2 Passo a passo com código 190

**45 Display: grid**

**197**

46.1 grid-template-columns 202 46.2 grid-template-rows 202

**46 Exercício: Conteúdo principal - a vitrine de cursos**

**206**

47.1 Objetivo 206 47.2 Passo a passo com código 207

**47 Exercício: Conteúdo Principal**

**216**

48.1 Objetivo 216 48.2 Passo a passo com código 217

**48 Exercício: Conteúdo principal (opcional) - os planos**

**222**

49.1 Objetivo 222 49.2 Passo a passo com código 223

**49 Exercício: Conteúdo Principal - os benefícios**

**231**

50.1 Objetivo 231 50.2 Passo a passo com código 232

**50 Exercício: Conteúdo Principal (opcional) - chamada youtube**

**239**

51.1 Passo a passo com código 239

**51 Exercício: Conteúdo Principal - chamada dos aplicativos**

**247**

52.1 Objetivo 247 52.2 Passo a passo com código 248

**52 Bootstrap e frameworks de CSS**

**255**

53.1 Estilo e componentes base 255

Sumário Caelum

**53 Exercício: Primeiros passos com Bootstrap**

**258**

54.1 Objetivo 258 54.2 Passo a passo com código 258

**54 Exercício: Customizando o Bootstrap (opicional)**

**260**

55.1 Objetivo 260 55.2 Passo a passo com código 260

**55 Exercício: A tabela de cursos**

**262**

56.1 Objetivo 262 56.2 Passo a passo com código 262

**56 Um pouquinho da história do JavaScript**

**266**

57.1 História 266 57.2 Características da linguagem 267 57.3 Console do navegador 267 57.4 Sintaxe básica 268 57.5 A tag script 270 57.6 JavaScript em arquivo externo 271 57.7 Mensagens no console 271 57.8 DOM: sua página no mundo JavaScript 272 57.9 querySelector 272 57.10 Elemento da página como variável 273 57.11 querySelectorAll 273 57.12 Alterações no DOM 274 57.13 Funções e os eventos do DOM 274 57.14 Funções Anônimas 275 57.15 Manipulando strings 276 57.16 Imutabilidade 276 57.17 Conversões 277 57.18 Manipulando números 277 57.19 Concatenações 277

**57 Exercício: Calculando total de horas e cursos com JavaScript**

**280**

58.1 Objetivo 280 58.2 Passo a passo com código 280

**58 Propriedades CSS**

**284**

59.1 Propriedade font 284 59.2 Propriedade text 284

Caelum Sumário

59.3 Propriedade letter-spacing 284 59.4 Propriedade line-height 285 59.5 Propriedades de cor 285 59.6 Propriedade background 285 59.7 Propriedade border 285 59.8 Propriedade vertical-align 286 59.9 Propriedades width e height 286 59.10 Propriedade box-sizing 286 59.11 Propriedade overflow 287

Versão: 27.1.5

CAPÍTULO 1

A ESTRUTURA DOS ARQUIVOS DE UM PROJETO

Como todo tipo de projeto de software, existem algumas recomendações quanto à organização dos arquivos de um site. Não há nenhum rigor técnico quanto a essa organização e, na maioria das vezes, você vai adaptar as recomendações da maneira que for melhor para o seu projeto.

Como um site é um conjunto de páginas Web sobre um assunto, empresa, produto ou qualquer outra coisa, é comum todos os arquivos de um site estarem dentro de uma só pasta e, assim como um livro, é recomendado que exista uma "capa", uma página inicial que possa indicar para o visitante quais são as outras páginas que fazem parte desse projeto e como ele pode acessá-las, como se fosse o **índice** do site.

Ter esse índice, não por coincidência, é uma convenção adotada pelos servidores de páginas Web. Se desejamos que uma determinada pasta seja servida como um site e dentro dessa pasta existe um arquivo chamado **index.html**, esse arquivo será a página inicial, ou seja o índice, a menos que alguma configuração determine outra página para esse fim.

Dentro da pasta do site, no mesmo nível que o index.html , é recomendado que sejam criadas mais algumas pastas para manter separados os arquivos de imagens, as folhas de estilo e os scripts. Para iniciar um projeto, teríamos uma estrutura de pastas como a demonstrada na imagem a seguir:



Muitas vezes, um site é servido por meio de uma aplicação Web e, nesses casos, a estrutura dos arquivos depende de como a aplicação necessita dos recursos para funcionar corretamente. Porém, no geral, as aplicações também seguem um padrão bem parecido com o que estamos adotando para o nosso projeto.

1.1 WEB SITE OU APLICAÇÃO WEB?

Quando estamos começando no mundo do desenvolvimento Web, acabamos por conhecer muitos termos novos, que por muitas vezes não são claros ou nos causam confusão. Vamos entender um pouco mais agora, qual a diferença de um Web site e uma aplicação Web.

1 A ESTRUTURA DOS ARQUIVOS DE UM PROJETO **1**

**Web site**

Podemos considerar um Web site uma coleção de páginas HTML estáticas, ou seja, que não interagem com um banco de dados através de uma linguagem de servidor Web. Ou seja, aqui todo o conteúdo do site está escrito diretamente no documento HTML, assim como as imagens e outras mídias. Claro que, para qualquer página Web ser fornecida publicamente a mesma deve estar hospedada em um simples servidor Web (hospedagem de sites).

**Aplicação Web**

Uma aplicação Web pode conter uma coleção de páginas, porém o conteúdo destas páginas é montado dinamicamente, ou seja, é carregado através de solicitações (requisições) à um banco de dados, que conterá armazenado os textos e indicação dos caminhos das imagens ou mídias que a página precisa exibir. Porém um HTML não tem acesso direto à um banco de dados, e esta comunicação deve ser feita por uma linguagem de programação de servidor Web. Esta aplicação escrita com uma linguagem de servidor que tem o poder de acessar o banco de dados e montar a página HTML conforme o solicitado pelo navegador. Estas solicitações podem ser feitas de várias maneiras, inclusive utilizando JavaScript. Portanto uma aplicação Web é mais complexa porque precisa de uma linguagem de servidor para poder intermediar as solicitações do navegador, um banco de dados, e muitas vezes (porém não obrigatoriamente) exibir páginas HTML com estes conteúdos.

Exemplo de linguagens de servidor Web: Java EE, PHP, Python, Ruby on Rails, NodeJS etc...

**Agora é a melhor hora de aprender algo novo**

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos** 

**online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento

com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e

Business, entre outros! Ex-estudante da Caelum tem 10% de desconto, siga o link! Conheça a Alura Cursos Online.

1.2 EDITORES E IDES

Os editores de texto são programas de computador leves e ideais para escrever e editar as páginas de um site, como *Visual Studio Code* (https://code.visualstudio.com/), *Sublime* (https://www.sublimetext.com/), *Atom* (https://atom.io/) e *Notepad++* (https://notepad-plus-plus.org), que possuem realce de sintaxe e outras ferramentas para facilitar o desenvolvimento de páginas.

**2** 1.2 EDITORES E IDES

Há também **IDE**s (*Integrated Development Environment*) que são editores mais robustos e trazem mais facilidades para o desenvolvimento de aplicações Web, se integrando com outras funcionalidades. São alguns deles: *WebStorm* (https://www.jetbrains.com/webstorm/) *Eclipse* (https://www.eclipse.org/) e *Visual Studio* (https://visualstudio.microsoft.com).

1.2 EDITORES E IDES **3**

CAPÍTULO 2

EXERCÍCIO: BAIXANDO OS ARQUIVOS DO CURSO

2.1 OBJETIVO

Nesse curso criaremos um site com várias páginas e criaremos vários arquivos pra isso. Para sua organização, recomendamos que crie uma pasta para colocar todos esses arquivos. Tome cuidado para não criar essa pasta em um lugar muito escondido no seu computador e acabar esquecendo onde estão os arquvios.

Nossa recomendação é que crie essa pasta com o nome **"projeto-musicdot"** na **"Área de Trabalho"**. Assim, durante os exercícios, os seus arquivos estarão sempre acessíveis à um clique e você não precisará ficar navegando pelas pastas do seu computador à procura dos seus arquivos. Se você estiver confortável com seu computador e tiver algum nome e lugar favorito para criar essa pasta, fique à vontade.

Antes de escrever qualquer código do projeto, você também já pode pegar todos os arquivos de imagens e estilo do curso que estão prontos e copiar eles para a pasta **"projeto-musicdot"** que acabou de criar.

Faça o download dos arquivos no seguinte endereço: https://www.caelum.com.br/download/caelum arquivos-wd43.zip

**4** 2 EXERCÍCIO: BAIXANDO OS ARQUIVOS DO CURSO

| Abrindo o projeto no Visual Studio Code:  Clique no menu **ARQUIVO > ABRIR PASTA** e escolha a pasta **"projeto-musicdot"** que você criou na **"Área de Trabalho"**. Agora o *Visual Studio Code* vai ter acesso a todos os arquivos do curso, e sempre que você criar novos arquivos pelo editor eles já vão ser criados dentro da pasta.  Confira na barra lateral da esquerda se todos os arquivos aparecem como na imagem abaixo:  Figura 2.1: Barra lateral do VS Code com os arquivos do curso |
| --- |

**Seus livros de tecnologia parecem do século passado?**

Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**. 

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

Casa do Código, Livros de Tecnologia.

2.2 PASSO A PASSO COM CÓDIGO

1. Há 43 arquivos a serem adicionados nas pastas css, img, js, textos e video. Note que esses arquivos foram disponibilizados já prontos para você.

Os 43 arquivos devem ser adicionados na seguinte estrutura de pastas:

2.2 PASSO A PASSO COM CÓDIGO **5**

**# novos arquivos do projeto**

└─ pasta\_raiz\_do\_projeto

├─ css

│ ├─ bootstrap.css

│ └─ reset.css

├─ img

│ ├─ familia-tupfeln.jpg

│ ├─ favicon.ico

│ ├─ footer-icone-facebook.svg

│ ├─ footer-icone-instagram.svg

│ ├─ footer-icone-youtube.svg

│ ├─ home-background-mobile.png

│ ├─ home-background.png

│ ├─ icone-android-logo.svg

│ ├─ icone-apple-logo.svg

│ ├─ icone-beneficios-didatica-cuidadosa.svg │ ├─ icone-beneficios-nenhuma-duvida.svg │ ├─ icone-beneficios-online.svg

│ ├─ icone-beneficios-tudo-para-voce.svg │ ├─ icone-cordas.svg

│ ├─ icone-teclado-piano.svg

│ ├─ icone-youtube.svg

│ ├─ icones-audio.svg

│ ├─ icones-baixo.svg

│ ├─ icones-bateria.svg

│ ├─ icones-canto.svg

│ ├─ icones-percussao.svg

│ ├─ icones-sopro.svg

│ ├─ icones-teoria.svg

│ ├─ icones-violao-guitarra.svg

│ ├─ logo.svg

│ ├─ matriz-musicdot-antiga.jpg

│ ├─ matriz-musicdot.png

│ ├─ mockup-app.png

│ ├─ musicdot-logo-light.svg

│ ├─ planos-background-mobile.png

│ ├─ planos-background.png

│ ├─ play-button.svg

│ ├─ simbolo-musidot-branco.svg

│ ├─ simbolo-musidot-dark.svg

│ ├─ video-principal-mobile.png

│ ├─ video-principal.png

│ ├─ videos-thumbnail-video1.png

│ └─ videos-thumbnail-video2.png

├─ js

│ └─ bootstrap.js

├─ textos

│ └─ sobre.txt

└─ video

└─ video-promocional.mp4

**6** 2.2 PASSO A PASSO COM CÓDIGO

CAPÍTULO 3

INTRODUÇÃO AO HTML

*"Quanto mais nos elevamos, menores parecemos aos olhos daqueles que não sabem voar." -- Friedrich Wilhelm Nietzsche*

3.1 EXIBINDO INFORMAÇÕES NA WEB

A única linguagem que um navegador Web consegue interpretar para a exibição de conteúdo é o HTML. Para iniciar a exploração do HTML, vamos imaginar o seguinte caso: o navegador realizou uma requisição e recebeu como corpo da resposta o seguinte conteúdo:

MusicDot

Bem-vindo à MusicDot, seu portal de cursos de música online.

Confira nossas promoções.

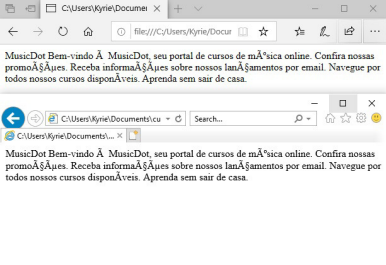
Receba informações sobre nossos lançamentos por email.

Navegue por todos nossos cursos disponíveis.

Aprenda sem sair de casa.

Para conhecer o comportamento dos navegadores quanto ao conteúdo descrito antes, vamos reproduzir esse conteúdo em um arquivo de texto comum, que pode ser criado com qualquer editor de texto puro. Salve o arquivo como **index.html** e abra-o a partir do navegador à sua escolha.

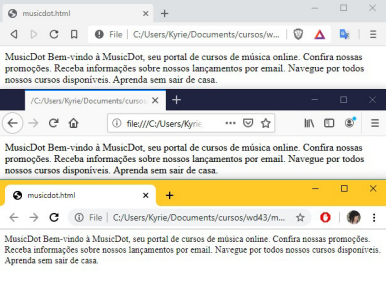
3 INTRODUÇÃO AO HTML **7**

****

Parece que obtemos um resultado um pouco diferente do esperado, não? Apesar de ser capaz de exibir texto puro em sua área principal, algumas regras devem ser seguidas caso desejemos que esse texto seja exibido com alguma formatação, para facilitar a leitura pelo usuário final.

Uma nota de atenção é que a imagem acima foi tirada dos navegadores: **Microsoft Edge e Microsoft Internet Explorer 11**. Veja o que acontece quando obtemos a mesma imagem porém com navegadores mais atuais:

**8** 3.1 EXIBINDO INFORMAÇÕES NA WEB



A imagem acima foi tirada nos navegadores: **Brave, Mozilla Firefox e Google Chrome**.

*Obs: existe a possibilidade de que mesmo nesses navegadores, se utilizada uma versão mais antiga, pode ser que o texto seja mostrado igual na foto dos navegadores da Microsoft*.

Usando os resultados acima podemos concluir que os navegadores mais antigos e até mesmo o **Microsoft Edge** por padrão:

Podem não exibir caracteres acentuados corretamente;

Mas até mesmo nos navegadores mais novos:

Não exibem quebras de linha.

Para que possamos exibir as informações desejadas com a formatação, é necessário que cada trecho de texto tenha uma **marcação** indicando qual é o significado dele. Essa marcação também influencia a maneira com que cada trecho do texto será exibido. A seguir é listado o texto com esta marcação esperada pelo navegador:

*<!DOCTYPE html>*

<html>

<head>

<title>MusicDot</title>

<meta charset="utf-8">

</head>

3.1 EXIBINDO INFORMAÇÕES NA WEB **9**

<body>

<h1>MusicDot</h1>

<h2>Bem-vindo à MusicDot, seu portal de cursos de música online.</h2>

<ul>

<li>Confira nossas promoções.</li>

<li>Receba informações sobre nossos lançamentos por email.</li>

<li>Navegue por todos nossos cursos disponíveis.</li>

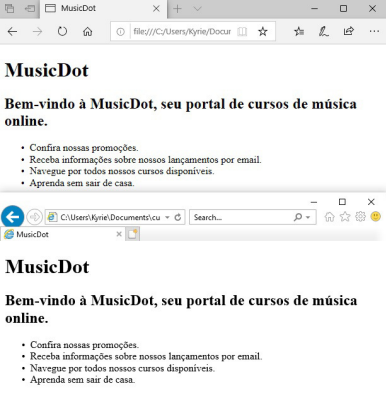
<li>Aprenda sem sair de casa.</li>

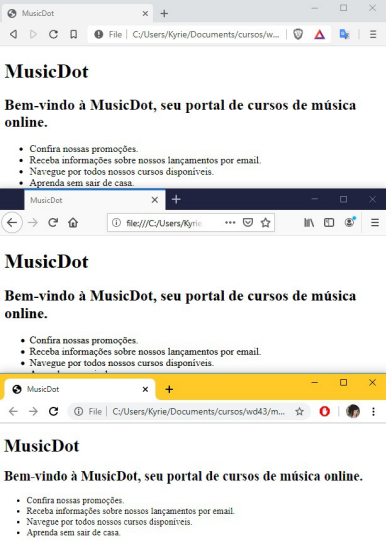
</ul>

</body>

</html>

O texto com as devidas marcações, comumente chamado de "código". Reproduza então o código anterior em um novo arquivo de texto puro e salve-o como **index-2.html**. Não se preocupe com a sintaxe, vamos conhecer detalhadamente cada característica destas marcações nos próximos capítulos. Abra o arquivo no navegador.

**10** 3.1 EXIBINDO INFORMAÇÕES NA WEB



Agora, uma página muito mais agradável e legível é exibida. Para isso, tivemos que adicionar as marcações que são pertencentes ao HTML. Essas marcações são chamadas de **tags**, e elas basicamente dão uma **representação** ao texto contido entre sua abertura e fechamento.

Apesar de estarem corretamente marcadas, as informações não apresentam pouco ou nenhum atrativo estético e, nessa deficiência do HTML, reside o primeiro e maior desafio de pessoas que desenvolvem para *front-end*.

3.1 EXIBINDO INFORMAÇÕES NA WEB **11**

O HTML (*Hypertext Markup Language*) ou linguagem de marcação de hipertexto foi desenvolvido para suprir a necessidade exibição de documentos científicos fornecidos por uma rede de Internet. Para termos uma comparação, é como se a Web fosse desenvolvida para exibir monografias redigidas e formatadas pela Metodologia do Trabalho Científico da ABNT. Porém, com o tempo e a evolução da Web e de seu potencial comercial, tornou-se necessária a exibição de informações com grande riqueza de elementos gráficos e de interação.

Começaremos por partes, primeiro entenderemos como o HTML funciona, para depois aprendermos estilos, elementos gráficos e interações.

**Agora é a melhor hora de aprender algo novo**

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos** 

**online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento

com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e

Business, entre outros! Ex-estudante da Caelum tem 10% de desconto, siga o link! Conheça a Alura Cursos Online.

3.2 SINTAXE DO HTML

O HTML é um conjunto de **tags** responsáveis pela marcação do conteúdo de uma página no navegador. No código que vimos antes, as tags são os elementos a mais que escrevemos usando a sintaxe <nomedatag> . Diversas tags são disponibilizadas pela linguagem HTML e cada uma possui uma funcionalidade específica.

No código de antes, vimos por exemplo o uso da tag <h1> . Ela representa o título principal da página.

<h1>MusicDot</h1>

Note a sintaxe. Uma tag é definida com caracteres *<* e *>* , e seu nome (**h1** no caso). Muitas tags possuem conteúdo, como o texto do título (*"MusicDot"*). Nesse caso, para determinar onde o conteúdo acaba, usamos uma *tag de fechamento* com a barra antes do nome: </h1> .

Algumas tags podem receber algum tipo de informação extra dentro de sua definição chamada de **atributo**. São parâmetros usando a sintaxe de atributo="valor" . Para definir uma imagem, por exemplo, usamos a tag <img> e, para indicar o caminho que está essa imagem, usamos o atributo

**12** 3.2 SINTAXE DO HTML

src :

<img src="../imagens/casa\_de\_praia.jpg">

Repare que a tag img não possui conteúdo por si só, e sim ela carrega ali o conteúdo de um arquivo externo (a imagem). Nesses casos, **não** é necessário usar uma tag de fechamento como antes no h1 .

3.3 TAGS HTML

O HTML é composto de diversas tags, cada uma com sua função e significado. Desde 2013, com a atualização da linguagem para o HTML 5, muitas novas tags foram adicionadas, que veremos ao longo do curso.

Nesse momento, vamos focar em tags que representam **títulos**, **parágrafo** e **ênfase**. **Títulos**

Quando queremos indicar que um texto é um título em nossa página, utilizamos as tags de ***heading*** em sua marcação:

<h1>MusicDot</h1>

<h2>Bem-vindo à MusicDot, seu portal de cursos de música online.</h2>

As tags de *heading* são para exibir conteúdo de texto e contém 6 níveis, ou seja de <h1> à <h6> , seguindo uma ordem de importância, sendo <h1> o título principal, o mais importante, e <h6> o título de menor importância.

Utilizamos, por exemplo, a tag <h1> para o nome, título principal da página, e a tag <h2> como subtítulo ou como título de seções dentro do documento.

*Obs: a tag <h1> só pode ser utilizada uma vez em cada página porque não pode existir mais de um conteúdo mais importante da página.*

A ordem de importância tem impacto nas ferramentas que processam HTML. As ferramentas de indexação de conteúdo para buscas, como o Google, Bing ou Yahoo! levam em consideração essa ordem e relevância. Os navegadores especiais para acessibilidade também interpretam o conteúdo dessas tags de maneira a diferenciar seu conteúdo e facilitar a navegação do usuário pelo documento.

**Parágrafos**

Quando exibimos qualquer texto em nossa página, é recomendado que ele seja sempre conteúdo de alguma tag filha da tag <body> . A marcação mais indicada para textos comuns é a tag de **parágrafo**:

<p>

A MusicDot é a maior escola online de música em todo o mundo.

</p>

3.3 TAGS HTML **13**

Se você tiver vários parágrafos de texto, use várias dessas tags <p> para separá-los:

<p>

A MusicDot é a maior escola online de música em todo o mundo.

</p>

<p>

Nossa matriz fica em Mafra, em Santa Catarina. De lá, saem grande parte das gravações de nossos cu rsos.

</p>

**Marcações de ênfase**

Quando queremos dar uma ênfase diferente a um trecho de texto, podemos utilizar as marcações de ênfase. Podemos deixar um texto "mais forte" com a tag <strong> ou deixar o texto com uma "ênfase acentuada" com a tag <em> . Do mesmo jeito que a tag <strong> deixa a tag "mais forte", temos também a tag <small> , que diminui o "peso" do texto.

Por padrão, os navegadores exibem o texto dentro da tag <strong> em negrito e o texto dentro da tag <em> em itálico. Existem ainda as tags <b> e <i> , que atingem o mesmo resultado visualmente, mas as tags <strong> e <em> são mais indicadas por definirem nossa intenção de significado ao conteúdo, mais do que uma simples indicação visual. Vamos discutir melhor a questão do significado das tags mais adiante.

<p>Aprenda de um jeito rápido e barato na <strong>MusicDot</strong>.</p>

3.4 IMAGENS

A tag <img> indica para o navegador que uma imagem deve ser "renderizada" (mostrada/desenhada) naquele lugar e necessita dois atributos preenchidos: src e alt . O primeiro é um atributo obrigatório para exibir a imagem e aponta para a sua localização (pode ser um local do seu computador ou um endereço na Web), já o segundo é um texto alternativo que aparece caso a imagem não possa ser carregada ou visualizada.

O atributo alt não é obrigatório, porém é considerado um erro caso seja omitido, pois ele provê o entendimento da imagem para pessoas com deficiência que necessitam o uso de leitores de tela para acessar o computador, e também auxilia na indexação da imagem para motores de busca, como o Google etc.

O HTML 5 introduziu duas novas tags específicas para imagem: <figure> e <figcaption> . A tag <figure> define uma imagem em conjunto com a tag <img> . Além disso, permite adicionar uma legenda para a imagem por meio da tag <figcaption> .

<figure>

<img src="img/matriz-musicdot.png" alt="Foto da matriz da musicdot">

<figcaption>Matriz da MusicDot</figcaption>

</figure>

**14** 3.4 IMAGENS

**Editora Casa do Código com livros de uma forma diferente**

Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não dominam tecnicamente o assunto para revisar os livros a fundo. Não têm anos de experiência em didáticas com cursos. 

Conheça a **Casa do Código**, uma editora diferente, com curadoria da **Caelum** e obsessão por livros de qualidade a preços justos.

Casa do Código, ebook com preço de ebook.

3.5 PRIMEIRA PÁGINA

A primeira página que desenvolveremos para a *MusicDot* será a *Sobre*, que explica detalhes sobre a empresa, apresenta fotos e a história.

Recebemos o design já pronto, assim como os textos. Nosso trabalho, como pessoas desenvolvedoras de front-end, é codificar o HTML e CSS necessários para esse resultado.

| **BOA PRÁTICA - INDENTAÇÃO**  Uma prática sempre recomendada, ligada à limpeza e utilizada para facilitar a leitura do código, é o uso correto de **recuos**, ou **indentação**, no HTML. Costumamos alinhar elementos "irmãos" na mesma margem e adicionar alguns espaços ou um *tab* para elementos "filhos".  A maioria dos exercícios dessa apostila utiliza um padrão recomendado de recuos. |
| --- |

| **BOA PRÁTICA - COMENTÁRIOS**  Quando iniciamos nosso projeto, utilizamos poucas tags HTML. Mais tarde adicionaremos uma quantidade razoável de elementos, o que pode gerar uma certa confusão. Para manter o código mais legível, é recomendada a adição de comentários antes da abertura e após do fechamento de tags estruturais (que conterão outras tags). Dessa maneira, nós podemos identificar claramente quando um elemento está **dentro** dessa estrutura ou **depois** dela. |
| --- |

3.5 PRIMEIRA PÁGINA **15**

<!-- início do cabeçalho -->

<header>

<p>Esse parágrafo está <strong>dentro</strong> do cabeçalho.</p> </header>

<!-- fim do cabeçalho -->

<p>Esse parágrafo está <strong>depois</strong> do cabeçalho.</p> **16** 3.5 PRIMEIRA PÁGINA

CAPÍTULO 4

EXERCÍCIO: PRIMEIROS PASSOS COM HTML

4.1 OBJETIVO

Começaremos a fazer nossa primeira página do projeto. Escreveremos o conteúdo e marcaremos ele com as *tags* do *HTML* para que nossa página tenha cada conteúdo diferenciado pela sua natureza. Nesse começo, o resultado será que cada conteúdo marcado terá um visual diferente por padrão, por exemplo, títulos aparecerão com um tamanho maior que o dos textos comuns.

Os conteúdos que aparecerão na página por enquanto serão:

A logo da MusicDot

O título principal da página: "Sobre a empresa"

Os textos sobre a empresa, em parágrafos

O subtítulo: "História"

Os textos sobre a história, em parágrafos

As figuras da empresa e da família. Lembrando que chamamos de figura o conjunto de uma foto e uma legenda.

Todo o conteúdo de texto das páginas já está pronto na pasta **"textos"**. Essa pasta foi disponibilizada juntamente com os arquivos do curso.

O resultado ao final do exercício será o seguinte:

4 EXERCÍCIO: PRIMEIROS PASSOS COM HTML **17**

****

Figura 4.1: Visual da página ao final do exercício

**Já conhece os cursos online Alura?**

A **Alura** oferece centenas de **cursos online** em sua plataforma exclusiva de ensino que favorece o aprendizado com a **qualidade** reconhecida da Caelum. Você pode escolher um curso nas áreas de Programação, Front-end, Mobile, 

Design & UX, Infra, Business, entre outras, com um plano que dá acesso a todos os cursos. Ex estudante da Caelum tem 10% de desconto neste link!

Conheça os cursos online Alura.

4.2 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **sobre.html** na pasta **raíz do projeto** com o seguinte código:

**18** 4.2 PASSO A PASSO COM CÓDIGO

**# sobre.html**

+<img src="img/logo.svg" alt="MusicDot">

+

+<h1> Sobre a empresa </h1>

+

+<p>

+ A MusicDot é a maior escola online de música em todo o mundo.

+ Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de + participação em 118 deles.

+</p>

+

+<p>

+ Nossa matriz fica em Mafra, em Santa Catarina. De lá, saem grande parte das gravações de nosso s cursos. Nossa matriz:

+</p>

+

+<figure>

+ <img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

+ <figcaption>

+ Matriz MusicDot

+ </figcaption>

+</figure>

+

+<p>

+ Assine os cursos da MusicDot. Acesse nosso site ou entre em contato

+ se tiver dúvidas. Conheça também nossa história e nossos diferenciais. +</p>

+

+<h2> História </h2>

+

+<figure>

+ <img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

+ <figcaption>Família Tüpfeln</figcaption>

+</figure>

+

+<p>

+ A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no i nterior de Santa Catarina. A

+ família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova iniciativa ,

+ revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular visão + administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado + de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da família. +</p>

+

+<p>

+ O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 paí ses.

+ Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhões + de usuários diferentes, em bilhões de diferentes pedidos.

+</p>

+

+<p>

+ O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do + Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora d o

+ Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e + nos escritórios em todo país.

+</p>

+

+<p>

4.2 PASSO A PASSO COM CÓDIGO **19**

+ Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu diversos p rêmios,

+ homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Musi cDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. +</p>

**20** 4.2 PASSO A PASSO COM CÓDIGO

CAPÍTULO 5

ESTRUTURA DE UM DOCUMENTO HTML

Um documento HTML válido precisa seguir obrigatoriamente a estrutura composta pelas tags <html> , <head> e <body> e a instrução <!DOCTYPE> . Esta estrutura está informada em uma documentação que descreve todos os detalhes do HTML, no caso as tags e atributos, e como os navegadores devem considerar e interpretar estas tags, esta documentação é chamada de "especificação do HTML", e através do que está declarado nela que é possível entender se um documento HTML válido. Um documento HTML inválido é carregado pelo navegador, porém em um "*modo de compatibilidade*", vamos entender melhor sobre isto logo mais.

Abaixo, vamos conhecer em detalhes cada uma das tags estruturais obrigatórias:

5.1 A TAG <HTML>

Na estrutura do nosso documento, antes de começar a colocar o conteúdo, inserimos uma tag <html> . Dentro dessa tag, é necessário declarar outras duas tags: <head> e <body> . Essas duas tags são "irmãs", pois estão no mesmo nível hierárquico em relação à sua tag "mãe", que é <html> .

<html> <!-- mãe -->

<head></head> <!-- filha -->

<body></body> <!-- filha -->

</html>

**Você pode também fazer o curso data dessa apostila na Caelum**

Querendo aprender ainda mais sobre? Esclarecer dúvidas dos exercícios? Ouvir explicações detalhadas com um instrutor? 

A Caelum oferece o **curso data** presencial nas cidades de São Paulo, Rio de Janeiro e Brasília, além de turmas incompany.

Consulte as vantagens do curso *Desenvolvimento Web com HTML, CSS e JavaScript*

5 ESTRUTURA DE UM DOCUMENTO HTML **21**

5.2 A TAG <HEAD>

A tag <head> contém informações sobre o documento HTML que são de interesse somente do navegador e para outros serviços da web, e não para as pessoas que vão acessar nosso site. São informações que não serão exibidas diretamente no navegador, também podemos considerar um local onde informamos os metadados sobre a página.

A especificação do HTML obriga a presença da tag de conteúdo <title> dentro da <head> , permitindo definir o título do documento, que poder ser visto na *barra de título* ou *aba* da janela do navegador. Caso contrário, a página não será um documento HTML válido.

Outra configuração muito importante, principalmente em documentos HTML cujo conteúdo é escrito em um idioma como o português, que contém caracteres "especiais" (acentos e cedilha), é a codificação/conjunto de caracteres, chamada de **encoding** ou **charset**.

Podemos configurar qual codificação queremos utilizar em nosso documento por meio da configuração de charset na tag <meta> . Um dos valores mais comuns usados hoje em dia é o **UTF 8**, também chamado de **Unicode**. Há outras possibilidades, como o **latin1**, muito usado antigamente.

O **UTF-8** é a recomendação atual para encoding na Web por ser amplamente suportada em navegadores e editores de código, além de ser compatível com praticamente todos os idiomas do mundo. É o que usaremos no curso.

<html>

<head>

<meta charset="utf-8">

<title>MusicDot</title>

</head>

<body>

</body>

</html>

5.3 A TAG <BODY>

A tag <body> contém o corpo de um documento HTML, que é exibido pelo navegador em sua janela, ou seja, todo o conteúdo visível do site. É necessário que o <body> tenha ao menos um elemento "filho", ou seja, uma ou mais tags HTML dentro dele.

<html>

<head>

<meta charset="utf-8">

<title>MusicDot</title>

</head>

<body>

<h1>A MusicDot</h1>

</body>

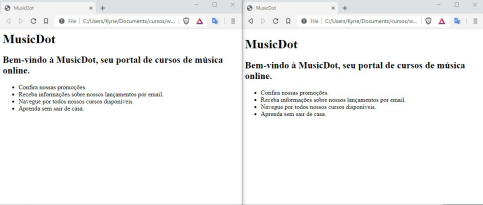
</html>

**22** 5.2 A TAG

Nesse exemplo, usamos a tag <h1> , que indica o título principal da página.

5.4 A INSTRUÇÃO DOCTYPE

O DOCTYPE não é uma tag HTML, mas uma instrução especial. Ela indica para o navegador qual **versão do HTML** deve ser utilizada para exibir a página. Quando não colocamos essa instrução a página é exibida numa espécie de "*modo de compatibilidade*" na qual algumas tags e estilizações não funcionam 100% corretamente. Principalmente as tags e estilizações mais atuais (lançadas na versão 5 do HTML). Inclusive é possível ver a diferença na folha de estilos padrão que o navegador usa quando não colocamos essa instrução.



A imagem da esquerda é a página **sem** Doctype e a imagem da direita é a página **com** Doctype . Dá para ver que existe uma leve diferença entre as duas páginas, principalmente com relação aos espaçamentos.

Utilizaremos <!DOCTYPE html> , que indica para o navegador a utilização da versão mais recente do HTML - a versão 5, atualmente\*.

Há muitas possibilidades mais complicadas nessa parte de DOCTYPE que eram usados em versões anteriores do HTML e do XHTML. Hoje em dia, nada disso é mais importante. O recomendado é **sempre usar a última versão do HTML**, usando a declaração de DOCTYPE simples:

*<!DOCTYPE html>*

A declaração do DOCTYPE, pode ser escrita toda em maiúsculo ou toda em minúsculo ou com a primeira letra maiúscula: <!DOCTYPE HTML> , <!DOCTYPE html> , <!Doctype HTML> , <!Doctype html> , <!doctype html> , <!doctype HTML> . O resultado será o mesmo para todos os casos.

5.4 A INSTRUÇÃO DOCTYPE **23**

\**Obs: desde maio de 2019 o desenvolvimento do HTML é mantido pelo W3C (World Wide Web Consortium) https://www.w3.org/, WHATWG e comunidade de desenvolvedores, e sua especificação é aberta no Github https://github.com/whatwg/html, e desde este movimento o HTML é considerado um "padrão vivo" (living standard) onde sua versão a partir da 5 é atualizada continuamente*.

**Seus livros de tecnologia parecem do século passado?**

Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**. 

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

Casa do Código, Livros de Tecnologia.

**24** 5.4 A INSTRUÇÃO DOCTYPE

CAPÍTULO 6

EXERCÍCIO: A ESPECIFICAÇÃO HTML

6.1 OBJETIVO

Nesse momento a aba do navegador está exibindo o caminho e o nome do arquivo "sobre.html" 

Figura 6.1: Aba do Firefox com a página "sobre" aberta

Nesse exercício mudaremos algumas informações da nossa página. A diferença é que essas informações serão apenas para o navegador e não devem ser exibidas como conteúdo da página.

As informações que passaremos para o navegador são:

O texto que aparecerá na aba do navegador: **"MusicDot | Sobre a empresa"**

O ícone que aparecerá na aba do navegador: **"img/favicon.ico"**

Qual o "charset" que será usado para o conteúdo da aba e da página: **"utf-8"**

Qual o idioma principal da página: **"pt-BR"**

Qual a versão do HTML está sendo usada: **"a mais atual suportada pelo navegador"**

Ao final do exercício deve ser possível escrever e ler caractéres acentuados no conteúdo da página. Ao final, também, a aba do navegador deve apresentar o texto e o ícone:



Figura 6.2: Aba do Firefox com a página "sobre" aberta

6 EXERCÍCIO: A ESPECIFICAÇÃO HTML **25**

**Agora é a melhor hora de aprender algo novo**

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos** 

**online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento

com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e

Business, entre outros! Ex-estudante da Caelum tem 10% de desconto, siga o link! Conheça a Alura Cursos Online.

6.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **sobre.html** na pasta **raíz do projeto** faça as seguintes alterações: **# sobre.html**

-<̶i̶m̶g̶ s̶r̶c̶=̶"̶i̶m̶g̶/̶l̶o̶g̶o̶.̶s̶v̶g̶"̶ a̶l̶t̶=̶"̶M̶u̶s̶i̶c̶D̶o̶t̶"̶>̶

+*<!doctype html>*

+<html lang="pt-BR">

+ <head>

+ <meta charset="utf-8">

+ <title>

+ MusicDot | Sobre a empresa

+ </title>

+ <link href="img/favicon.ico" rel="icon">

+ </head>

-<̶h̶1̶>̶ S̶o̶b̶r̶e̶ a̶ e̶m̶p̶r̶e̶s̶a̶ </̶h̶1̶>̶

+ <body>

-<̶p̶>̶

+ <img src="img/logo.svg" alt="MusicDot">

+

+ <h1> Sobre a empresa </h1>

+

+ <p>

A MusicDot é a maior escola online de música em todo o mundo.

Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

-</̶p̶>̶

+ </p>

-<̶p̶>̶

+ <p>

Nossa matriz fica em Mafra, em Santa Catarina. De lá, saem grande parte das gravações de n ossos cursos. Nossa matriz:

-</̶p̶>̶

+ </p>

-<̶f̶i̶g̶u̶r̶e̶>̶

+ <figure>

**26** 6.2 PASSO A PASSO COM CÓDIGO

<img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

<figcaption>

Matriz MusicDot

</figcaption>

-</̶f̶i̶g̶u̶r̶e̶>̶

+ </figure>

-<̶p̶>̶

+ <p>

Assine os cursos da MusicDot. Acesse nosso site ou entre em contato se tiver dúvidas. Conheça também nossa história e nossos diferenciais. -</̶p̶>̶

+ </p>

-<̶h̶2̶>̶ H̶i̶s̶t̶ó̶r̶i̶a̶ </̶h̶2̶>̶

+ <h2> História </h2>

-<̶f̶i̶g̶u̶r̶e̶>̶

+ <figure>

<img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

<figcaption>Família Tüpfeln</figcaption>

-</̶f̶i̶g̶u̶r̶e̶>̶

+ </figure>

-<̶p̶>̶

+ <p>

A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no interior de Santa Catarina. A

família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova inicia tiva,

revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular vi são

administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da famí lia.

-</̶p̶>̶

+ </p>

-<̶p̶>̶

+ <p>

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países.

Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhõe s

de usuários diferentes, em bilhões de diferentes pedidos.

-</̶p̶>̶

+ </p>

-<̶p̶>̶

+ <p>

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora d o

Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fo ra do

Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e nos escritórios em todo país.

-</̶p̶>̶

+ </p>

-<̶p̶>̶

+ <p>

Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu divers os prêmios,

6.2 PASSO A PASSO COM CÓDIGO **27**

homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da MusicDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. -</̶p̶>̶

+ </p>

+

+ </body>

+</html>

**28** 6.2 PASSO A PASSO COM CÓDIGO

CAPÍTULO 7

ESTILIZANDO COM CSS

Quando escrevemos o HTML, marcamos o conteúdo da página com tags que melhor representam o significado daquele conteúdo. Quando abrimos a página no navegador é possível perceber que ele mostra as informações com estilos diferentes.

Um h1, por exemplo, por padrão é apresentado em negrito numa fonte maior. Parágrafos de texto são espaçados entre si, e assim por diante. Isso quer dizer que o navegador tem um *estilo padrão* para as tags que usamos. Porém para fazer sites bonitos, ou com o visual próximo de uma dada identidade visual (design), vamos precisar *personalizar a apresentação padrão dos elementos* da página.

Antigamente, isso era feito no próprio HTML. Caso houvesse a necessidade de um título ser vermelho, era só fazer:

<h1><font color="red">MusicDot anos 90</font></h1>

Além da tag <font> , várias outras tags de estilo existiam. Mas isso é passado. Hoje em dia **tags HTML para estilo são má prática** e jamais devem ser usadas, são interpretadas apenas para o modo de compatibilidade.

Em seu lugar, surgiu o **CSS** (*Cascading Style Sheet* ou folha de estilos em cascata), que é uma outra linguagem, separada do HTML, com objetivo único de cuidar da estilização da página. A vantagem é que o CSS é bem mais robusto que o HTML para estilização, como veremos. Mas, principalmente, escrever formatação visual misturado com conteúdo de texto no HTML se mostrou algo impraticável. O CSS resolve isso separando as coisas; regras de estilo não aparecem mais no HTML, apenas no CSS.

7.1 SINTAXE E INCLUSÃO DE CSS

A sintaxe do CSS tem estrutura simples: é uma declaração de propriedades e valores separados por um sinal de dois pontos ":", e cada propriedade é separada por um sinal de ponto e vírgula ";" da seguinte maneira:

color: blue;

background-color: yellow;

O elemento que receber essas propriedades será exibido com o texto na cor azul e com o fundo amarelo. Essas propriedades podem ser declaradas de três maneiras diferentes.

7 ESTILIZANDO COM CSS **29**

**Atributo style**

A primeira delas é com o atributo style no próprio elemento:

<p style="color: blue; background-color: yellow;">

O conteúdo desta tag será exibido em azul com fundo amarelo no navegador!

</p>

Mas tínhamos acabado de discutir que uma das grandes vantagens do CSS era manter as regras de estilo fora do HTML. Usando esse atributo style não parece que fizemos isso. Justamente por isso não se recomenda esse tipo de uso na prática, mas sim os que veremos a seguir.

**A tag style**

A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma tag <style> .

Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a qual elemento nos referimos. Fazemos isso utilizando um **seletor CSS**. É basicamente uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos.

No exemplo a seguir, usaremos o seletor que pega todas as tags p e altera sua cor e background:

*<!DOCTYPE html>*

<html>

<head>

<meta charset="utf-8">

<title>Sobre a MusicDot</title>

<style>

p {

color: blue;

background-color: yellow;

}

</style>

</head>

<body>

<p>

O conteúdo desta tag será exibido em azul com fundo amarelo!

</p>

<p>

<strong>Também</strong> será exibido em azul com fundo amarelo!

</p>

</body>

</html>

O código dentro da tag <style> indica que estamos alterando a cor e o fundo de todos os elementos com tag p . Dizemos que selecionamos esses elementos pelo nome de sua tag, e aplicamos certas propriedades CSS apenas neles.

Revisando então a estrutura de uso do CSS:

seletor {

propriedade: valor;

**30** 7.1 SINTAXE E INCLUSÃO DE CSS

}

Algumas propriedades contém "subpropriedades" que modificam uma parte específica daquela propriedade que vamos trabalhar, sendo sua sintaxe:

seletor {

propriedade-subpropriedade: valor;

}

No exemplo abaixo, em ambos os casos, trabalhamos com a propriedade text , que estiliza a aparência do texto do seletor informado. Podemos especificar quais propriedades específicas do texto queremos modificar, no caso text-align o alinhamento do texto, e com text-decoration colocamos o efeito de sublinhado.

p {

text-align: center;

text-decoration: underline;

}

**Arquivo externo**

A terceira maneira de declararmos os estilos do nosso documento é com um arquivo externo com a extensão .css . Para que seja possível declarar nosso CSS em um arquivo à parte, precisamos indicar em nosso documento HTML uma ligação entre ele e a folha de estilo (arquivo com a extensão .css ).

Além da melhor organização do projeto, a folha de estilo externa traz ainda as vantagens de manter nosso HTML mais limpo e do reaproveitamento de uma mesma folha de estilos para diversos documentos.

A indicação de uso de uma folha de estilos externa deve ser feita dentro da tag <head> de um documento HTML:

*<!DOCTYPE html>*

<html>

<head>

<meta charset="utf-8">

<title>MusicDot | Sobre a empresa</title>

<!-- Inclusão do arquivo CSS -->

<link rel="stylesheet" href="estilos.css">

</head>

<body>

<p>

O conteúdo desta tag será exibido em azul com fundo amarelo!

</p>

<p>

<strong>Também</strong> será exibido em azul com fundo amarelo!

</p>

</body>

</html>

E dentro do arquivo estilos.css colocamos apenas o conteúdo do CSS:

p {

7.1 SINTAXE E INCLUSÃO DE CSS **31**

color: blue;

background-color: yellow;

}

**Editora Casa do Código com livros de uma forma diferente**

Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não dominam tecnicamente o assunto para revisar os livros a fundo. Não têm anos de experiência em didáticas com cursos. 

Conheça a **Casa do Código**, uma editora diferente, com curadoria da **Caelum** e obsessão por livros de qualidade a preços justos.

Casa do Código, ebook com preço de ebook.

7.2 PROPRIEDADES TIPOGRÁFICAS E FONTES

Da mesma maneira que alteramos cores, podemos alterar o texto. Podemos definir fontes com o uso da propriedade font-family .

A propriedade font-family pode receber seu valor com ou sem aspas dependendo da sua composição, por exemplo, quando uma fonte tem o nome separado por *espaço*.

Por padrão, os navegadores mais conhecidos exibem texto em um tipo que conhecemos como "serif". As fontes mais conhecidas (e comumente utilizadas como padrão) são "Times" e "Times New Roman", dependendo do sistema operacional. Elas são chamadas de **fontes serifadas** pelos pequenos ornamentos em suas terminações.

Podemos alterar a família de fontes que queremos utilizar em nosso documento para a família "sans serif" (sem serifas), que contém, por exemplo, as fontes "Arial" e "Helvetica". Podemos também declarar que queremos utilizar uma família de fontes "monospace" como, por exemplo, a fonte "Courier".

*Obs: Fontes monospace podem ser tanto com serifa ou sem serifa. Monospace quer dizer apenas que todas as letras possuem o mesmo tamanho*

h1 {

font-family: serif;

}

h2 {

font-family: sans-serif;

}

**32** 7.2 PROPRIEDADES TIPOGRÁFICAS E FONTES

p {

font-family: monospace;

}

É possível, e muito comum, declararmos o nome de algumas fontes que gostaríamos de verificar se existem no computador, permitindo que tenhamos um controle melhor da forma como nosso texto será exibido.

Em nosso projeto, as fontes não têm ornamentos, vamos declarar essa propriedade para todo o documento por meio do seu elemento body :

body {

font-family: "Helvetica", "Lucida Grande", sans-serif;

}

Nesse caso, o navegador verificará se a fonte "Helvetica" está disponível e a utilizará para exibir os textos de todos os elementos do nosso documento que, por cascata, herdarão essa propriedade do elemento body .

Caso a fonte "Helvetica" não esteja disponível, o navegador verificará a disponibilidade da próxima fonte declarada, no nosso exemplo a "Lucida Grande". Caso o navegador não encontre também essa fonte, ele solicita qualquer fonte que pertença à família "sans-serif", declarada logo a seguir, e a utiliza para exibir o texto, não importa qual seja ela.

Temos outras propriedades para manipular a fonte, como a propriedade font-style , que define o estilo da fonte que pode ser: normal (normal na vertical), italic (inclinada) e oblique (oblíqua).

7.3 ALINHAMENTO E DECORAÇÃO DE TEXTO

Já vimos uma série de propriedades e subpropriedades que determinam o tipo e estilo da fonte. Vamos conhecer algumas maneiras de alterarmos as disposições dos textos.

No exemplo a seguir vamos mudar o alinhamento do texto com a propriedade text-align .

p {

text-align: right;

}

O exemplo determina que todos os parágrafos da nossa página tenham o texto alinhado para a direita. Também é possível determinar que um elemento tenha seu conteúdo alinhado ao centro ao definirmos o valor center para a propriedade text-align , ou então definir que o texto deve ocupar toda a largura do elemento aumentando o espaçamento entre as palavras com o valor justify . Por padrão o texto é alinhado à esquerda, com o valor left , porém é importante lembrar que essa propriedade propaga-se em cascata.

É possível configurar também uma série de espaçamentos de texto com o CSS: 7.3 ALINHAMENTO E DECORAÇÃO DE TEXTO **33**

p {

line-height: 3px; /\* tamanho da altura de cada linha \*/

letter-spacing: 3px; /\* tamanho do espaço entre cada letra \*/

word-spacing: 5px; /\* tamanho do espaço entre cada palavra \*/

text-indent: 30px; /\* tamanho da margem da primeira linha do texto \*/

}

7.4 IMAGEM DE FUNDO

A propriedade background-image permite indicar um arquivo de imagem para ser exibido ao fundo do elemento. Por exemplo:

h1 {

background-image: url(sobre-background.jpg);

}

Com essa declaração, o navegador vai requisitar um arquivo sobre-background.jpg , que deve estar na mesma pasta do arquivo CSS onde consta essa declaração. Mas podemos também passar um endereço da web para pegar imagens remotamente:

body {

background-image: url(https://i.imgur.com/uAhjMNd.jpg);

}

**Já conhece os cursos online Alura?**

A **Alura** oferece centenas de **cursos online** em sua plataforma exclusiva de 

ensino que favorece o aprendizado com a **qualidade** reconhecida da Caelum.

Você pode escolher um curso nas áreas de Programação, Front-end, Mobile,

Design & UX, Infra, Business, entre outras, com um plano que dá acesso a todos os cursos. Ex estudante da Caelum tem 10% de desconto neste link!

Conheça os cursos online Alura.

7.5 BORDAS

As propriedades do CSS para definirmos as **bordas** de um elemento nos apresentam uma série de opções. Podemos, para cada borda de um elemento, determinar sua cor, seu estilo de exibição e sua largura. Por exemplo:

body {

border-color: red;

border-style: solid;

border-width: 1px;

}

**34** 7.4 IMAGEM DE FUNDO

A propriedade border tem uma forma resumida para escrever os mesmos estilos que adicionamos acima, mas de uma maneira mais simples:

body {

border: 1px solid red;

}

Para que o efeito da cor sobre a borda surta efeito, é necessário que a propriedade border-style tenha qualquer valor diferente do padrão none .

Podemos também falar em qual dos lados do nosso elemento queremos a borda usando a subpropriedade que indica lado:

h1 {

border-top: 1px solid red; /\* borda vermelha em cima \*/

border-right: 1px solid red; /\* borda vermelha à direita \*/

border-bottom: 1px solid red; /\* borda vermelha embaixo \*/

border-left: 1px solid red; /\* borda vermelha à esquerda \*/

}

| Conseguimos fazer também comentários no CSS usando a seguinte sintaxe:  /\* deixando o fundo amarelo ouro \*/  body {  background-color: gold;  } |
| --- |

7.6 CORES NA WEB

Propriedades como background-color , color , border-color , entre outras aceitam uma cor como valor. Existem várias maneiras de definir cores quando utilizamos o CSS.

A primeira, mais simples, é usando o nome da cor:

h1 {

color: red;

}

h2 {

background-color: yellow;

}

O difícil é acertar a exata variação de cor que queremos no design e também cada navegador tem o seu padrão de cor para os nomes de cores. A W3C obriga que todos os navegadores tenham pelo menos 140 nomes de cores padronizados, mas existem mais de 16 milhões de cores na web e seria extremamente complicado nomear cada uma delas. Por isso, é bem incomum usarmos cores com seus nomes. O mais comum é definir a cor com base em sua composição RGB.

RGB é o sistema de cor usado nos monitores, já que cada pixel nos monitores possuem 3 leds (um 7.6 CORES NA WEB **35**

vermelho, um verde e um azul) e a combinação dessas 3 cores formam todas as outras 16 milhões de cores que vemos nos monitores. Podemos escolher a intensidade de cada um desses três leds básicos, numa escala de 0 a 255.

Um amarelo forte, por exemplo, tem 255 de Red, 255 de Green e 0 de Blue (255, 255, 0). Se quiser um laranja, basta diminuir um pouco o verde (255, 200, 0). E assim por diante.

No CSS, podemos escrever as cores tendo como base sua composição RGB. Aliás, no CSS3 - que veremos melhor depois - há até uma sintaxe bem simples pra isso:

h3 {

color: rgb(255, 200, 0);

}

Essa sintaxe funciona nos browsers mais modernos e até alguns browsers super antigos mas não é a mais comum na prática, por questões de compatibilidade. O mais comum é a **notação hexadecimal**. Essa sintaxe tem suporte universal nos navegadores e é mais curta de escrever, apesar de ser mais enigmática.

h3 {

background-color: #f2eded;

}

No fundo, porém, as duas formas são baseadas no sistema RGB. Na notação hexadecimal (que começa com **#**), temos 6 caracteres, os primeiros 2 indicam o canal Red, os dois seguintes, o Green, e os dois últimos, Blue; ou seja, RGB. E usamos a matemática pra escrever menos, trocando a base numérica de decimal para hexadecimal.

Na base hexadecimal, os algarismos vão de zero a quinze (ao invés do zero a nove da base decimal comum). Para representar os algarismos de dez a quinze, usamos letras de A a F. Nessa sintaxe, portanto, podemos utilizar números de 0-9 e letras de A-F.

Há uma conta por trás dessas conversões, mas seu editor de imagens deve ser capaz de fornecer ambos os valores para você sem problemas. Um valor 255 vira FF na notação hexadecimal. A cor **#f2eded**, por exemplo, é equivalente a **rgb(242, 237, 237)**, um cinza claro.

Vale aqui uma dica quanto ao uso de cores hexadecimais, toda vez que os caracteres presentes na composição da base se repetirem, estes podem ser simplificados. Então um número em hexadecimal **3366ff**, pode ser simplificado para **36f**.

*Obs: os 3 pares de números devem ser iguais entre si, ou seja, se tivermos um hexadecimal #33aabc não podemos simplificar nada do código.*

**36** 7.6 CORES NA WEB

CAPÍTULO 8

EXERCÍCIO: MUDANDO O VISUAL DA NOSSA PÁGINA

8.1 OBJETIVO

Começaremos a alterar o visual da nossa página para deixá-la mais agradável.

Primeiro precisamos criar nosso arquivo de estilo e importá-lo no nosso html. Dentro desse arquivo utilizaremos os **seletores CSS** para escolher as tags/elementos que queremos estilizar e depois aplicaremos as **propriedades do CSS**.

As mudanças de estilo serão:

Na página toda: trocar a família da fonte de toda a página

No título principal: alterar a cor da fonte para um cinza-escuro **"#3D3D3D"**

No título principal: colocar uma cor de fundo cinza-azulado **"#E1EDF2"**

Nos subtítulos: colocar uma borda na parte de baixo

Nas figuras: centralizar o conteúdo

Nas figuras: colocar uma cor de fundo chamada **"whitesmoke"**

Nas figuras: alterar a cor da fonte para um cinza-claro **"lightgrey"**

A página ficará assim:

8 EXERCÍCIO: MUDANDO O VISUAL DA NOSSA PÁGINA **37**

****

Figura 8.1: Visual da página ao final do exercício

**Você pode também fazer o curso data dessa apostila na Caelum**

Querendo aprender ainda mais sobre? Esclarecer dúvidas dos exercícios? Ouvir explicações detalhadas com um instrutor? 

A Caelum oferece o **curso data** presencial nas cidades de São Paulo, Rio de Janeiro e Brasília, além de turmas incompany.

Consulte as vantagens do curso *Desenvolvimento Web com HTML, CSS e JavaScript*

8.2 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **sobre.css** na pasta **css** com o seguinte código:

**# css/sobre.css**

+body {

+ font-family: "Helvetica", "Lucida Grande", sans-serif;

+ color: #3D3D3D;

+}

+

+h1 {

**38** 8.2 PASSO A PASSO COM CÓDIGO

+ background-color: #E1EDF2;

+}

+

+h2 {

+ border-bottom: 2px solid black;

+}

+

+figure {

+ text-align: center;

+

+ background-color: whitesmoke;

+ border: 1px solid lightgrey;

+}

2. No arquivo **sobre.html** na pasta **raíz do projeto** faça as seguintes alterações: **# sobre.html**

*<!doctype html>*

<html lang="pt-BR">

<head>

<meta charset="utf-8">

<title>

MusicDot | Sobre a empresa

</title>

<link href="img/favicon.ico" rel="icon">

+ <link rel="stylesheet" href="css/sobre.css">

</head>

<body>

<img src="img/logo.svg" alt="MusicDot">

<h1> Sobre a empresa </h1>

<p>

A MusicDot é a maior escola online de música em todo o mundo.

Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

</p>

<p>

Nossa matriz fica em Mafra, em Santa Catarina. De lá, saem grande parte das gravações de n ossos cursos. Nossa matriz:

</p>

<figure>

<img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

<figcaption>

Matriz MusicDot

</figcaption>

</figure>

<p>

Assine os cursos da MusicDot. Acesse nosso site ou entre em contato

se tiver dúvidas. Conheça também nossa história e nossos diferenciais. </p>

<h2> História </h2>

<figure>

<img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

8.2 PASSO A PASSO COM CÓDIGO **39**

<figcaption>Família Tüpfeln</figcaption>

</figure>

<p>

A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no interior de Santa Catarina. A

família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova inicia tiva,

revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular vi são

administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da famí lia.

</p>

<p>

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países.

Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhõe s

de usuários diferentes, em bilhões de diferentes pedidos.

</p>

<p>

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora d o

Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fo ra do

Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e nos escritórios em todo país.

</p>

<p>

Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu divers os prêmios,

homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da MusicDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. </p>

</body>

</html>

**40** 8.2 PASSO A PASSO COM CÓDIGO

CAPÍTULO 9

ESPAÇAMENTOS E DIMENSÕES

Temos algumas maneiras de trabalhar com dimensões e espaçamentos. Para espaçamento interno e externo usamos respectivamente padding e margin , e para redimensionar elementos podemos usar as propriedades de largura e altura ou width e height . Vamos ver mais a fundo essas propriedades.

9.1 DIMENSÕES

É possível determinar as dimensões de um elemento, por exemplo:

p {

background-color: red;

height: 300px;

width: 300px;

}

Todos os parágrafos do nosso HTML ocuparão 300 pixels de altura e de largura, com cor de fundo vermelha.

| Se usarmos o inspetor de elementos do navegador veremos que o restante do espaço ocupado pelo elemento vira margin |
| --- |

**Seus livros de tecnologia parecem do século passado?**

Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**. 

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

Casa do Código, Livros de Tecnologia.

9 ESPAÇAMENTOS E DIMENSÕES **41**

9.2 ESPAÇAMENTOS

**Padding**

A propriedade **padding** é utilizada para definir um espaçamento interno em elementos (por espaçamento interno queremos dizer a distância entre o limite do elemento, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:

padding-top

padding-right

padding-bottom

padding-left

Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, abaixo e à esquerda respectivamente. Essa ordem é importante para entendermos como funciona a *shorthand property* (encurtamento) do padding.

Podemos definir todos os valores para as subpropriedades do padding em uma única propriedade, chamada exatamente de padding , e seu comportamento é descrito nos exemplos a seguir:

Se passado somente um valor para a propriedade padding , esse mesmo valor é aplicado em todas as direções.

p {

padding: 10px;

}

Se passados dois valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para padding-top e padding-bottom ) e o segundo será aplicado à direita e à esquerda (equivalente ao mesmo valor para padding-right e padding-left ).

p {

padding: 10px 15px;

}

Se passados três valores, o primeiro será aplicado acima (equivalente a padding-top ), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para padding-right e padding-left ) e o terceiro valor será aplicado abaixo do elemento (equivalente a padding-bottom ).

p {

padding: 10px 20px 15px;

}

Se passados quatro valores, serão aplicados respectivamente a padding-top , padding-right , padding-bottom e padding-left . Para facilitar a memorização dessa ordem, basta lembrar que os valores são aplicados em **sentido horário**.

p {

**42** 9.2 ESPAÇAMENTOS

padding: 10px 20px 15px 5px;

}

| Uma dica para omitir valores do padding:  Quando precisar omitir valores, sempre omita no sentido anti-horário começando a partir da subpropriedade -left .  Como os valores tem posicionamento fixo na hora de declarar os espaçamentos, o navegador não sabe quando e qual valor deve ser omitido. Por exemplo:  Se tivermos um padding:  h1 {  padding: 10px 25px 10px 15px;  }  O código não pode sofrer o encurtamento porque por mais que os valores de top e bottom sejam iguais, os valores right e left não são e eles são os primeiros a serem omitidos. Veja o que acontece quando vamos omitir o valor de 10px do bottom:  h1 {  padding: 10px 25px 15px;  }  O navegador vai interpretar da seguinte maneira:  h1 {  padding: top right bottom;  }  Que no final vai ficar igual a:  h1 {  padding-top: 10px;  padding-right: 25px;  padding-bottom: 15px;  padding-left: 25px;  }  E esses valores não são os que nós colocamos no começo com padding: 10px 25px 10px 15px; |
| --- |

**Margin**

A propriedade margin é bem parecida com a propriedade padding , exceto que ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si (espaçamento externo). Além das subpropriedades listadas a seguir, há a *shorthand property* margin que se comporta da mesma maneira que a *shorthand property* do padding vista no tópico anterior.

9.2 ESPAÇAMENTOS **43**

margin-top

margin-right

margin-bottom

margin-left

Há ainda uma maneira de permitir que o navegador defina qual será a dimensão da propriedade padding ou margin conforme o espaço disponível na tela: definimos o valor auto para os espaçamentos que quisermos.

No exemplo a seguir, definimos que um elemento não tem nenhuma margem acima ou abaixo de seu conteúdo e que o navegador define uma margem igual para ambos os lados de acordo com o espaço disponível:

p {

margin: 0 auto;

}

**44** 9.2 ESPAÇAMENTOS

CAPÍTULO 10

EXERCÍCIO: TRABALHANDO COM TAMANHOS E ESPAÇAMENTO

10.1 OBJETIVO

Nesse exercício trabalharemos com os tamanhos das caixas, com o espaço vazio entre essas caixas e com o espaço de respiro entre a caixa e seu conteúdo.

Em resumo, iremos:

diminuir a largura do conteúdo da página para **940px**

Para que o conteúdo não fique colado no canto da página, afaste ele dos cantos direito e esquerdo da tela, até que fique centralizado.

criar um espaço de respiro interno de **10px** para o conteúdo do título principal

criar um espaço de respiro interno de **15px** para o conteúdo das figuras

afastar **30px** as figuras dos cantos da tela e das caixas que vêm antes e depois delas dentro das figuras, afastar **10px** a legenda da imagem

A página ficará assim:

10 EXERCÍCIO: TRABALHANDO COM TAMANHOS E ESPAÇAMENTO **45**

****

Figura 10.1: Visual da página ao final do exercício

**Agora é a melhor hora de aprender algo novo**

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e Business, entre outros! Ex-estudante da Caelum tem 10% de desconto, siga o link! 

Conheça a Alura Cursos Online.

10.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **sobre.css** na pasta **css** faça as seguintes alterações:

**# css/sobre.css**

body {

+ width: 940px;

+ margin-left: auto;

+ margin-right: auto;

+

**46** 10.2 PASSO A PASSO COM CÓDIGO

font-family: "Helvetica", "Lucida Grande", sans-serif;

color: #3D3D3D;

}

h1 {

+ padding: 10px;

+

background-color: #E1EDF2;

}

h2 {

border-bottom: 2px solid black;

}

figure {

+ margin: 30px;

+ padding: 15px;

+

text-align: center;

background-color: whitesmoke;

border: 1px solid lightgrey;

}

+

+figcaption {

+ margin-top: 10px;

+}

10.2 PASSO A PASSO COM CÓDIGO **47**

CAPÍTULO 11

LISTAS HTML

Não são raros os casos em que queremos exibir uma listagem em nossas páginas. O HTML tem algumas tags definidas para que possamos fazer isso de maneira correta. A lista mais comum é a lista não ordenada definida pela tag <ul> (*unordered list*).

<ul>

<li>Primeiro item da lista</li>

<li>

Segundo item da lista:

<ul>

<li>Primeiro item da lista aninhada</li>

<li>Segundo item da lista aninhada</li>

</ul>

</li>

<li>Terceiro item da lista</li>

</ul>

Note que, para cada item da lista não-ordenada, utilizamos uma marcação de item de lista <li> (*list item*). No exemplo acima, utilizamos uma estrutura composta na qual o segundo item da lista contém uma nova lista. A mesma tag de item de lista <li> é utilizada quando demarcamos uma lista ordenada.

<ol>

<li>Primeiro item da lista</li>

<li>Segundo item da lista</li>

<li>Terceiro item da lista</li>

<li>Quarto item da lista</li>

<li>Quinto item da lista</li>

</ol>

As listas ordenadas ( <ol> - *ordered list*) também podem ter sua estrutura composta por outras listas ordenadas como no exemplo que temos para as listas não-ordenadas. Também é possível ter listas ordenadas aninhadas em um item de uma lista não-ordenada e vice-versa.

11.1 LISTAS DE DEFINIÇÃO

Existe um terceiro tipo de lista que devemos utilizar para demarcar um glossário, quando listamos termos e seus significados. Essa lista é a **lista de definição** (*definition list*).

<dl>

<dt>HTML</dt>

<dd>

HTML é a linguagem de marcação de textos utilizada

**48** 11 LISTAS HTML

para exibir textos como páginas na Internet.

</dd>

<dt>Navegador</dt>

<dd>

Navegador é o software que requisita um documento HTML

através do protocolo HTTP e exibe seu conteúdo em uma

janela.

</dd>

</dl>

Para estilizar o formato padrão das listas ordenadas e não-ordenadas, podemos utilizar a propriedade list-style-type no CSS:

ul {

/\* alterar para circulo antes de cada <li> da lista não-ordenada \*/

list-style-type: circle;

}

ol {

/\* alterar para uma sequência alfabética antes de cada <li> da lista ordenada \*/ list-style-type: upper-alpha;

}

| Também podemos usar a *shorthand property* list-style: circle |
| --- |

**Editora Casa do Código com livros de uma forma diferente**

Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não dominam tecnicamente o assunto para revisar os livros a fundo. Não têm anos de experiência em didáticas com cursos. 

Conheça a **Casa do Código**, uma editora diferente, com curadoria da **Caelum** e obsessão por livros de qualidade a preços justos.

Casa do Código, ebook com preço de ebook.

11.2 LINKS NO HTML

Quando precisamos indicar que um trecho de texto se refere a um outro conteúdo, seja ele no mesmo documento ou em outro endereço (por exemplo uma página na web), utilizamos a tag de âncora <a> .

Existem três diferentes usos para as âncoras. Um deles é a definição de links:

<p>

11.2 LINKS NO HTML **49**

Visite o site da <a href="https://www.caelum.com.br">Caelum</a>.

</p>

Note que a âncora está demarcando apenas a palavra **Caelum** de todo o conteúdo do parágrafo exemplificado. Isso significa que, ao clicarmos com o cursor do mouse na palavra **Caelum**, o navegador redirecionará o usuário para o site da **Caelum**, indicado no atributo href .

| Podemos adicionar o atributo target="" para especificar onde que essa página irá carregar. Por padrão a página irá abrir na mesma aba da página que tem o link, mas se quisermos que a página abra em outra aba podemos colocar o valor \_blank dentro desse atributo:  <a href="https://www.caelum.com.br" target="\_blank"> |
| --- |

Outro uso para a tag de âncora é a demarcação de destinos para links dentro do próprio documento, o que chamamos de ***bookmark***.

<p>Mais informações <a href="#info">aqui</a>.</p>

<p>Conteúdo da página...</p>

<h2 id="info">Mais informações sobre o assunto:</h2>

<p>Informações...</p>

De acordo com o exemplo acima, ao clicarmos sobre a palavra **aqui**, demarcada com um link, o usuário será levado à porção da página onde o *bookmark* **info** é visível. *Bookmark* é o elemento que tem o atributo id .

É possível, com o uso de um link, levar o usuário a um *bookmark* presente em outra página.

<a href="http://www.caelum.com.br/curso/wd43/#contato">

Entre em contato sobre o curso

</a>

O exemplo acima fará com que o usuário que clicar no link seja levado à porção da página indicada no endereço, especificamente no ponto onde o *bookmark* **contato** seja visível.

O outro uso para a tag de âncora é a demarcação de destinos para links dentro do próprio site, mas não na mesma página que estamos. Por exemplo, estamos na página **sobre.html** e queremos um link para a página **index.html**.

<p>Acesse <a href="index.html">nossa loja</a>.</p>

**50** 11.2 LINKS NO HTML

CAPÍTULO 12

EXERCÍCIO: USANDO LINKS E LISTAS

12.1 OBJETIVO

Criaremos uma nova seção no final da página que listará todos os diferenciais da MusicDot. Os textos de cada item dessa lista estão em **textos/sobre.txt**. Lembre-se que a ordem dos itens não é importante.

Agora que temos mais seções e bastante conteúdo, podemos melhorar a navegação e a experiência das pessoa que acessarem essa página. Adicionaremos vários links úteis no conteúdo que já temos.

No segundo parágrafo da página adicione um link ao redor de "Mafra, em Santa Catarina" que aponta para a localização ficticia da MusicDot no Google Maps: "http://maps.google.com.br/? q=190,GabrielDequech,Mafra,SC".

No terceiro parágrafo da página, adicione os seguintes links:

"nosso site" apontando para a página principal do nosso site em "index.html"

"nossa história" apontando para a página sobre, porém tendo como alvo o subtítulo <h2> História </h2>

"nossos diferenciais" apontando para a página sobre, porém tendo como alvo o subtítulo <h2> Diferenciais </h2>

Figura 12.1: Links com destaque nos parágrafos do início da página

12 EXERCÍCIO: USANDO LINKS E LISTAS **51**

****Figura 12.2: Lista de diferenciais no final da página

**Já conhece os cursos online Alura?**

A **Alura** oferece centenas de **cursos online** em sua plataforma exclusiva de ensino que favorece o aprendizado com a **qualidade** reconhecida da Caelum. Você pode escolher um curso nas áreas de Programação, Front-end, Mobile, 

Design & UX, Infra, Business, entre outras, com um plano que dá acesso a todos os cursos. Ex estudante da Caelum tem 10% de desconto neste link!

Conheça os cursos online Alura.

12.2 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **index.html** na pasta **raíz do projeto** com o seguinte código: **# index.html**

+<h1> Página principal temporária da MusicDot </h1>

2. No arquivo **sobre.html** na pasta **raíz do projeto** faça as seguintes alterações: **# sobre.html**

*<!doctype html>*

**52** 12.2 PASSO A PASSO COM CÓDIGO

<html lang="pt-BR">

<head>

<meta charset="utf-8">

<title>

MusicDot | Sobre a empresa

</title>

<link href="img/favicon.ico" rel="icon">

<link rel="stylesheet" href="css/sobre.css">

</head>

<body>

<img src="img/logo.svg" alt="MusicDot">

<h1> Sobre a empresa </h1>

<p>

A MusicDot é a maior escola online de música em todo o mundo.

Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

</p>

<p>

- N̶o̶s̶s̶a̶ m̶a̶t̶r̶i̶z̶ f̶i̶c̶a̶ e̶m̶ M̶a̶f̶r̶a̶,̶ e̶m̶ S̶a̶n̶t̶a̶ C̶a̶t̶a̶r̶i̶n̶a̶.̶ D̶e̶ l̶á̶,̶ s̶a̶e̶m̶ g̶r̶a̶n̶d̶e̶ p̶a̶r̶t̶e̶ d̶a̶s̶ g̶r̶a̶v̶a̶ç̶õ̶e̶s̶ d̶e̶ n̶ o̶s̶s̶o̶s̶ c̶u̶r̶s̶o̶s̶.̶ N̶o̶s̶s̶a̶ m̶a̶t̶r̶i̶z̶:̶

+ Nossa matriz fica em <a href="https://maps.google.com.br/?q=190,GabrielDequech,Mafra,SC"> Mafra, em Santa Catarina </a>. De lá, saem grande parte das gravações de nossos cursos. Nossa matr iz:

</p>

<figure>

<img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

<figcaption>

Matriz MusicDot

</figcaption>

</figure>

<p>

- A̶s̶s̶i̶n̶e̶ o̶s̶ c̶u̶r̶s̶o̶s̶ d̶a̶ M̶u̶s̶i̶c̶D̶o̶t̶.̶ A̶c̶e̶s̶s̶e̶ n̶o̶s̶s̶o̶ s̶i̶t̶e̶ o̶u̶ e̶n̶t̶r̶e̶ e̶m̶ c̶o̶n̶t̶a̶t̶o̶ - s̶e̶ t̶i̶v̶e̶r̶ d̶ú̶v̶i̶d̶a̶s̶.̶ C̶o̶n̶h̶e̶ç̶a̶ t̶a̶m̶b̶é̶m̶ n̶o̶s̶s̶a̶ h̶i̶s̶t̶ó̶r̶i̶a̶ e̶ n̶o̶s̶s̶o̶s̶ d̶i̶f̶e̶r̶e̶n̶c̶i̶a̶i̶s̶.̶ + Assine os cursos da MusicDot. Acesse <a href="index.html"> nosso site </a> ou entre em con tato

+ se tiver dúvidas. Conheça também <a href="#historia"> nossa história </a> e <a href="#dife renciais"> nossos diferenciais </a>.

</p>

- <̶h̶2̶>̶ H̶i̶s̶t̶ó̶r̶i̶a̶ </̶h̶2̶>̶

+ <h2 id="historia"> História </h2>

<figure>

<img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

<figcaption>Família Tüpfeln</figcaption>

</figure>

<p>

A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no interior de Santa Catarina. A

família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova inicia tiva,

revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular vi são

administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da famí

12.2 PASSO A PASSO COM CÓDIGO **53**

lia.

</p>

<p>

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países.

Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhõe s

de usuários diferentes, em bilhões de diferentes pedidos.

</p>

<p>

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora d o

Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fo ra do

Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e nos escritórios em todo país.

</p>

<p>

Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu divers os prêmios,

homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da MusicDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. </p>

+ <h2 id="diferenciais"> Diferenciais </h2>

+ <ul>

+ <li> Menor preço do mercado, garantido </li>

+ <li> Se você achar um serviço mais barato, leva 1 mês de graça </li>

+ <li> Pague em reais, dólares, euros, libras, dodgecoins, litecoins ou bitcoins </li> + <li> Todas as compras acompanham acesso à Alura </li>

+ <li> Maior escola de música online do mundo </li>

+ <li> Atendimento via telefone, email, chat, twitter, facebook, instagram, ICQ, WhatsApp, S MS, carta, fax, sinal de fumaça e telegrama </li>

+ <li> Presente em 124 países </li>

+ <li> Mais de um milhão de funcionários em todo o mundo </li>

+ </ul>

</body>

</html>

**54** 12.2 PASSO A PASSO COM CÓDIGO

CAPÍTULO 13

SELETORES MAIS ESPECÍFICOS E HERANÇA

Durante o curso veremos outros tipos de seletores. Por hora veremos um seletor que deixa nossa estilização um pouco mais precisa do que fazemos agora.

Vamos com o exemplo a seguir:

**HTML:**

<img src="img/logo-musicdot" atl="Logo da MusicDot">

<figure>

<img src="img/matriz-musicdot" alt="Foto da matriz da MusicDot">

<figcaption>Matriz da MusicDot</figcaption>

</figure>

<figure>

<img src="img/familia-tupfeln" alt="Foto da família Tupfeln">

<figcaption>Família Tüpfeln</figcaption>

</figure>

**CSS:**

img {

width: 300px;

}

No código acima estamos aplicando uma largura de 300px para todas as tags <img> . Mas e se nós só quisermos aplicar essa largura apenas para as imagens que estão nas figuras? É aí que entra o seletor mais específico:

figure img {

width: 300px;

}

Agora estamos aplicando a largura de 300px apenas às imagens que são filhas de uma tag <figure> .

Outra forma de selecionar elementos mais específicos é usando o atributo id="" nos elementos que queremos estilizar e depois fazer a chamada de seletor de id :

**HTML:**

13 SELETORES MAIS ESPECÍFICOS E HERANÇA **55**

<img src="img/logo-musicdot" atl="Logo da MusicDot">

<figure>

<img src="img/matriz-musicdot" alt="Foto da matriz da MusicDot" id="matriz-musicdot"> <figcaption>Matriz da MusicDot</figcaption>

</figure>

<figure>

<img src="img/familia-tupfeln" alt="Foto da família Tupfeln" id="familia-tupfeln"> <figcaption>Família Tüpfeln</figcaption>

</figure>

**CSS:**

#matriz-musicdot {

width: 300px;

}

#familia-tupfeln {

width: 300px;

}

Só que não é recomendado o uso de id para a estilização de elementos já que a idéia do atributo é para fazer uma referência única na página como fizemos na parte dos links. Quando queremos estilizar elementos específicos é melhor utilizar o atributo class="" . O comportamento no CSS será idêntico ao do atributo id="" , mas class foi feito para ser usado no CSS e no JavaScript.

Arrumando o exemplo anterior, usando classes:

**HTML:**

<img src="img/logo-musicdot" atl="Logo da MusicDot">

<figure>

<img src="img/matriz-musicdot" alt="Foto da matriz da MusicDot" class="matriz-musicdot"> <figcaption>Matriz da MusicDot</figcaption>

</figure>

<figure>

<img src="img/familia-tupfeln" alt="Foto da família Tupfeln" class="familia-tupfeln"> <figcaption>Família Tüpfeln</figcaption>

</figure>

**CSS:**

.matriz-musicdot {

width: 300px;

}

.familia-tupfeln {

width: 300px;

}

**Grau de especificidade de um seletor**

Existe uma coisa muito importante no CSS que precisamos tomar cuidado é o **grau de especificidade de um seletor**. Isto é, a prioridade de interpretação de um seletor pelo navegador. Para

**56** 13 SELETORES MAIS ESPECÍFICOS E HERANÇA

entender estas regras de especificidade de um selector, ao criarmos um seletor de tag a sua pontuação se torna **1**. Quando usamos um seletor de classe sua pontuação se torna **10**. Quando usamos um seletor de id sua pontuação se torna **100**. Ao fim, o navegador soma a pontuação dos seletores aplicados à um elemento, e as propriedades com o seletor de maior pontuação são as que valem.

<body>

<p class="paragrafo" id="paragrafo-rosa">Texto</p>

</body>

p { /\* Pontuação 1 \*/

color: blue;

}

.paragrafo { /\* Pontuação 10 \*/

color: red;

}

#paragrafo-rosa { /\* Pontuação 100 \*/

color: pink;

}

No exemplo acima o parágrafo vai ficar com a cor rosa porque o seletor que tem a cor rosa é o seletor de maior pontuação.

Quando elementos possuem a mesma pontuação quem prevalece é a propriedade do último seletor:

p { /\* Pontuação 1 \*/

color: blue;

}

p { /\* Pontuação 1 \*/

color: red;

}

No exemplo acima a cor do parágrafo será vermelha.

Podemos também somar os pontos para deixar nosso seletor mais forte:

body p { /\* Seletor de tag + outro seletor de tag = 2 pontos \*/

color: brown;

}

p { /\* Pontuação 1 \*/

color: blue;

}

No exemplo acima nós deixamos nosso seletor mais específico para os <p> que estão dentro de uma tag <body> , portanto a cor do parágrafo será marrom.

Em resumo:

Quanto mais específico é o nosso seletor, maior sua pontuação no nível de especificidade do CSS. Portanto devemos sempre trabalhar com uma baixa especificidade, para que não seja impossível sobrescrever valores quando necessário em uma situação específica.

13 SELETORES MAIS ESPECÍFICOS E HERANÇA **57**

**Herança**

A cascata do CSS, significa justamente a possibilidade de elementos filhos herdarem características de estilização de elementos superiores, estas definidas por suas propriedades, que podem ou não passar aos seus descendentes seus valores.

Vamos ver o exemplo de código a seguir:

*<!DOCTYPE html>*

<html>

<head>

<meta charset="UTF-8">

<title>Um exemplo</title>

</head>

<body>

<p>Uma breve explicação de algo com um <a href="https://google.com.br">link</a> para uma referênc ia de outra página</p>

<figure>

<img src="img/foto" alt="Uma foto">

<figcaption>Uma foto</figcaption>

</figure>

</body>

</html>

Vamos mudar a família da fonte de toda a página. Uma maneira que podemos fazer é selecionar todas as tags que contém text ( <p> , <a> e <figcaption> ) e colocar a família de fonte que queremos:

p {

font-family: 'Helvetica', sans-serif;

}

a {

font-family: 'Helvetica', sans-serif;

}

figcaption {

font-family: 'Helvetica', sans-serif;

}

Mas isso dá muito trabalho e estamos repetindo código. Ao invés de colocar essa propriedade em cada um dos elementos textuais da nossa página, podemos colocar no elemento superior a estas tags, neste caso é o elemento <body> .

body {

font-family: 'Helvetica', sans-serif;

}

No exemplo acima todos os elementos filhos da tag <body> vão receber a propriedade font family: e isso é o que nós chamamos de **herança**. Herança acontece quando elementos herdam propriedades dos elementos acima deles (elementos pai).

*Obs: Para saber se uma propriedade deixa herança ou não, é possível consultar na sua especificação ou no site MDN https://developer.mozilla.org/.*

**58** 13 SELETORES MAIS ESPECÍFICOS E HERANÇA

13.1 PARA SABER MAIS: O VALOR INHERIT

Imagine que temos a seguinte divisão com uma imagem:

<div>

<img src="box-model.png" alt="box model">

</div>

div {

border: 2px solid;

border-color: red;

width: 30px;

height: 30px;

}

Queremos que a imagem preencha todo o espaço da <div> , mas as propriedades width e height não são aplicadas em cascata, sendo assim, somos obrigados a definir o tamanho da imagem manualmente:

img {

width: 30px;

height: 30px;

}

Esta não é uma solução sustentável, porque, caso alterarmos o tamanho da <div> , teremos que lembrar de alterar também o tamanho da imagem. Uma forma de resolver este problema é utilizando o valor **inherit** para as propriedades width e height da imagem:

img {

width: inherit;

height: inherit;

}

O valor inherit indica para o elemento filho que ele deve utilizar o mesmo valor presente no elemento pai, sendo assim, toda vez que o tamanho do elemento pai for alterado, automaticamente o elemento filho herdará o novo valor, facilitando assim, a manutenção do código.

Lembre-se de que o inherit também afeta propriedades que não são aplicadas em cascata. 13.1 PARA SABER MAIS: O VALOR INHERIT **59**

**Você pode também fazer o curso data dessa apostila na Caelum**

Querendo aprender ainda mais sobre? Esclarecer dúvidas dos exercícios? Ouvir explicações detalhadas com um instrutor? 

A Caelum oferece o **curso data** presencial nas cidades de São Paulo, Rio de Janeiro e Brasília, além de turmas incompany.

Consulte as vantagens do curso *Desenvolvimento Web com HTML, CSS e JavaScript*

**60** 13.1 PARA SABER MAIS: O VALOR INHERIT

CAPÍTULO 14

EXERCÍCIO: SELECIONANDO

ELEMENTOS MAIS ESPECÍFICOS E HERANÇA

14.1 OBJETIVO

No momento as figuras da página ocupam praticamente a largura da tela inteira. Nesse exercício, diminuiremos a largura das duas tags <figure> da página. Além das larguras, também alteraremos a disposição de cada figura na página. Cada uma terá uma largura e disposição diferentes.

A <figure> da matriz deverá ter **550px** de largura e deve estar centralizada em relação à página. A <figure> da família devera ter **275px** de largura, deve ficar alinhada à direita e envolta pelos textos que no HTML vêm abaixo dela.

Ao alterar a largura das tags <figure> você notará que as tags <img> dentro delas são maiores e ultrapassam a caixa da figura:



Figura 14.1: A imagem da matriz é maior do que os 550px da <figure>

Para resolver isso podemos dizer que toda tag <img> dentro de uma tag <figure> deve ocupar todo o espaço que definimos, mas não mais que isso.

O resultado final será assim:

14 EXERCÍCIO: SELECIONANDO ELEMENTOS MAIS ESPECÍFICOS E HERANÇA **61**

****

Figura 14.2: Resultado final da figura da matriz



Figura 14.3: Resultado final da figura da família

**Seus livros de tecnologia parecem do século passado?**

Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**. 

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

Casa do Código, Livros de Tecnologia.

14.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **sobre.css** na pasta **css** faça as seguintes alterações:

**# css/sobre.css**

body {

width: 940px;

margin-left: auto;

**62** 14.2 PASSO A PASSO COM CÓDIGO

margin-right: auto;

font-family: "Helvetica", "Lucida Grande", sans-serif;

color: #3D3D3D;

}

h1 {

padding: 10px;

background-color: #E1EDF2;

}

h2 {

border-bottom: 2px solid black;

}

figure {

margin: 30px;

padding: 15px;

text-align: center;

background-color: whitesmoke;

border: 1px solid lightgrey;

}

figcaption {

margin-top: 10px;

}

+

+.matriz-musicdot {

+ width: 550px;

+ margin-left: auto;

+ margin-right: auto;

+}

+

+.familia-tupfeln {

+ width: 275px;

+ float: right;

+}

+

+figure img {

+ width: 100%;

+}

2. No arquivo **sobre.html** na pasta **raíz do projeto** faça as seguintes alterações: **# sobre.html**

*<!doctype html>*

<html lang="pt-BR">

<head>

<meta charset="utf-8">

<title>

MusicDot | Sobre a empresa

</title>

<link href="img/favicon.ico" rel="icon">

<link rel="stylesheet" href="css/sobre.css">

</head>

<body>

14.2 PASSO A PASSO COM CÓDIGO **63**

<img src="img/logo.svg" alt="MusicDot">

<h1> Sobre a empresa </h1>

<p>

A MusicDot é a maior escola online de música em todo o mundo.

Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

</p>

<p>

Nossa matriz fica em <a href="https://maps.google.com.br/?q=190,GabrielDequech,Mafra,SC"> Mafra, em Santa Catarina </a>. De lá, saem grande parte das gravações de nossos cursos. Nossa matr iz:

</p>

- <̶f̶i̶g̶u̶r̶e̶>̶

+ <figure class="matriz-musicdot">

<img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

<figcaption>

Matriz MusicDot

</figcaption>

</figure>

<p>

Assine os cursos da MusicDot. Acesse <a href="index.html"> nosso site </a> ou entre em con tato

se tiver dúvidas. Conheça também <a href="#historia"> nossa história </a> e <a href="#dife renciais"> nossos diferenciais </a>.

</p>

<h2 id="historia"> História </h2>

- <̶f̶i̶g̶u̶r̶e̶>̶

+ <figure class="familia-tupfeln">

<img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

<figcaption>Família Tüpfeln</figcaption>

</figure>

<p>

A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no interior de Santa Catarina. A

família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova inicia tiva,

revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular vi são

administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da famí lia.

</p>

<p>

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países.

Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhõe s

de usuários diferentes, em bilhões de diferentes pedidos.

</p>

<p>

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora d o

Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fo **64** 14.2 PASSO A PASSO COM CÓDIGO

ra do

Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e nos escritórios em todo país.

</p>

<p>

Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu divers os prêmios,

homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da MusicDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. </p>

<h2 id="diferenciais"> Diferenciais </h2>

<ul>

<li> Menor preço do mercado, garantido </li>

<li> Se você achar um serviço mais barato, leva 1 mês de graça </li> <li> Pague em reais, dólares, euros, libras, dodgecoins, litecoins ou bitcoins </li> <li> Todas as compras acompanham acesso à Alura </li>

<li> Maior escola de música online do mundo </li>

<li> Atendimento via telefone, email, chat, twitter, facebook, instagram, ICQ, WhatsApp, S MS, carta, fax, sinal de fumaça e telegrama </li>

<li> Presente em 124 países </li>

<li> Mais de um milhão de funcionários em todo o mundo </li>

</ul>

</body>

</html>

14.2 PASSO A PASSO COM CÓDIGO **65**

CAPÍTULO 15

DESACOPLAMENTO COM CLASSES

Muitas vezes quando estamos declarando os estilos de uma página HTML, achamos mais fácil usar o seletor de nome da tag ao invés de usar classes. Porém isto pode causar problemas imprevistos se não usado com cautela. Vamos analisar o seguinte código para entender a situação:

**HTML:**

<h1>MusicDot</h1>

<h2>História</h2>

<p>Texto</p>

<h2>Diferenciais</h2>

<ul>

<li>Diferencial 1</li>

<li>Diferencial 2</li>

<li>Diferencial 3</li>

</ul>

**CSS:**

h2 {

font-size: 24px;

font-weight: bold;

border-bottom: 1px solid #000000;

}

No exemplo acima adicionamos estilo nas tags <h2> para ter um tamanho de fonte maior, uma fonte mais grossa e uma borda abaixo para fazer o efeito de linha divisória. Até aqui tudo certo. Mas agora vamos colocar um outro título na página chamada "Sobre a MusicDot" e esse título tem relevância maior do que os dois outros títulos que temos (História e Diferenciais), portanto vamos ter que modificar suas tags para uma de menos importância:

<h1>MusicDot</h1>

<h2>Sobre a MusicDot</h2>

<p>Introdução</p>

<!-- Mudamos para h3 pois queremos que tenham menos relevância que o título "Sobre a MusicDot" --> <h3>História</h3>

<p>Texto</p>

<!-- Mudamos para h3 pois queremos que tenham menos relevância que o título "Sobre a MusicDot" --> <h3>Diferenciais</h3>

<ul>

<li>Diferencial 1</li>

**66** 15 DESACOPLAMENTO COM CLASSES

<li>Diferencial 2</li>

<li>Diferencial 3</li>

</ul>

Agora com essa mudança da estrutura do HTML o nosso CSS está alterando um elemento que não é o que nós inicialmente queríamos mudar. Vamos ter que fazer a mudança no CSS para usar as nossas alterações no elemento certo.

Neste exemplo a solução é relativamente simples, porém imagine que temos seletores bem mais complexos, com heranças etc... a mudança não seria tão simples. Por isso o ideal é declarar estilos com classes ao invés de nomes de tags. Uma dica pra dar nome às classes é elas representarem o papel que estas tags estão exercendo em conjunto com os estilos declarados, no nosso caso, estamos declarando um conjunto de estilo para subtítulos.

Veja como fica o resultado do desacoplamento do conjunto de estilos do nome da tag, para ser agora com classes:

**HTML:**

<h1>MusicDot</h1>

<h2>Sobre a MusicDot</h2>

<p>Introdução</p>

<!-- Adicionamos a classe subtitulo-->

<h3 class="subtitulo">História</h3>

<p>Texto</p>

<!-- Adicionamos a classe subtitulo-->

<h3 class="subtitulo">Diferenciais</h3>

<ul>

<li>Diferencial 1</li>

<li>Diferencial 2</li>

<li>Diferencial 3</li>

</ul>

**CSS:**

.subtitulo {

font-size: 24px;

font-weight: bold;

border-bottom: 1px solid #000000;

}

Usando classes, podemos alterar toda a estrutura HTML sem nos preocupar se estas alterações afetarão a estilização que fizemos no começo.

15 DESACOPLAMENTO COM CLASSES **67**

CAPÍTULO 16

EXERCÍCIO: USANDO CLASSES PARA EVITAR ACOPLAMENTO DE ESTILOS

16.1 OBJETIVO

Adicionaremos uma nova seção com as informações de contato e começaremos colocando e estilizando apenas o título dessa nova seção.

O novo título de contato é na verdade um subtítulo da página e por isso usaremos uma tag <h2> . Porém, o estilo desse <h2> deve ser o mesmo da tag <h1> que temos na página. O desafio aqui é estilizar tanto o <h1> Sobre a empresa </h1> quanto o <h2> Contato </h2> sem repetir o código CSS para cada um e sem depender do nome da tag que usamos.

Uma dica é que podemos usar os tipos de seletores que vimos anteriormente para desacoplar nosso código CSS do HTML da página. Traduzindo, deixaremos de depender do nome das tags que escolhemos e passaremos a selecionar ambas as tags no CSS de uma forma mais específica e aplicaremos o mesmo estilo nas duas.

Para que esse estilo funcione, não poderemos mais estilizar todos os <h2> como fazemos agora. Será preciso também alterar os estilos genéricos de todo <h2> para que estilizem apenas os subtítulos <h2> História </h2> e <h2> Diferenciais </h2>

No final teremos o novo título aparecendo na página com o seguinte estilo:



Figura 16.1: Novo subtítulo "Contato" com mesmo estilo do título principal

**68** 16 EXERCÍCIO: USANDO CLASSES PARA EVITAR ACOPLAMENTO DE ESTILOS

**Agora é a melhor hora de aprender algo novo**

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos** 

**online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento

com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e

Business, entre outros! Ex-estudante da Caelum tem 10% de desconto, siga o link! Conheça a Alura Cursos Online.

16.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **sobre.css** na pasta **css** faça as seguintes alterações:

**# css/sobre.css**

body {

width: 940px;

margin-left: auto;

margin-right: auto;

font-family: "Helvetica", "Lucida Grande", sans-serif;

color: #3D3D3D;

}

-h̶1̶ {̶

+.titulo {

padding: 10px;

+ font-size: 32px;

+

background-color: #E1EDF2;

}

-h̶2̶ {̶

+.subtitulo {

border-bottom: 2px solid black;

}

figure {

margin: 30px;

padding: 15px;

text-align: center;

background-color: whitesmoke;

border: 1px solid lightgrey;

}

figcaption {

margin-top: 10px;

16.2 PASSO A PASSO COM CÓDIGO **69**

}

.matriz-musicdot {

width: 550px;

margin-left: auto;

margin-right: auto;

}

.familia-tupfeln {

width: 275px;

float: right;

}

figure img {

width: 100%;

}

2. No arquivo **sobre.html** na pasta **raíz do projeto** faça as seguintes alterações: **# sobre.html**

*<!doctype html>*

<html lang="pt-BR">

<head>

<meta charset="utf-8">

<title>

MusicDot | Sobre a empresa

</title>

<link href="img/favicon.ico" rel="icon">

<link rel="stylesheet" href="css/sobre.css">

</head>

<body>

<img src="img/logo.svg" alt="MusicDot">

- <̶h̶1̶>̶ S̶o̶b̶r̶e̶ a̶ e̶m̶p̶r̶e̶s̶a̶ </̶h̶1̶>̶

+ <h1 class="titulo"> Sobre a empresa </h1>

<p>

A MusicDot é a maior escola online de música em todo o mundo.

Fundada em 1932, possui estúdios em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

</p>

<p>

Nossa matriz fica em <a href="https://maps.google.com.br/?q=190,GabrielDequech,Mafra,SC"> Mafra, em Santa Catarina </a>. De lá, saem grande parte das gravações de nossos cursos. Nossa matr iz:

</p>

<figure class="matriz-musicdot">

<img src="img/matriz-musicdot.png" alt="Matriz da musicdot">

<figcaption>

Matriz MusicDot

</figcaption>

</figure>

<p>

Assine os cursos da MusicDot. Acesse <a href="index.html"> nosso site </a> ou entre em con tato

se tiver dúvidas. Conheça também <a href="#historia"> nossa história </a> e <a href="#dife **70** 16.2 PASSO A PASSO COM CÓDIGO

renciais"> nossos diferenciais </a>.

</p>

- <̶h̶2̶ i̶d̶=̶"̶h̶i̶s̶t̶o̶r̶i̶a̶"̶>̶ H̶i̶s̶t̶ó̶r̶i̶a̶ </̶h̶2̶>̶

+ <h2 id="historia" class="subtitulo"> História </h2>

<figure class="familia-tupfeln">

<img src="img/familia-tupfeln.jpg" alt="Foto da família tüpfeln">

<figcaption>Família Tüpfeln</figcaption>

</figure>

<p>

A fundação em 1932 ocorreu no momento da descoberta econônica de cursos por stream online no interior de Santa Catarina. A

família Tüpfeln, tradicional da região, investiu todas as suas economias nessa nova inicia tiva,

revolucionária para a época. A fundadora frau Dagmar Olaf Tüpfeln, dotada de particular vi são

administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho Ernst Noten Tüpfeln, atual CEO. O nome da empresa é inspirado no nome da famí lia.

</p>

<p>

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países.

Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 2 bilhõe s

de usuários diferentes, em bilhões de diferentes pedidos.

</p>

<p>

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora d o

Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fo ra do

Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Mafra e nos escritórios em todo país.

</p>

<p>

Dada a importância econômica da empresa para o Brasil, a família Tüpfeln já recebeu divers os prêmios,

homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da MusicDot, além de presidentes da União Européia, Ásia e o secretário-geral da ONU. </p>

- <̶h̶2̶ i̶d̶=̶"̶d̶i̶f̶e̶r̶e̶n̶c̶i̶a̶i̶s̶"̶>̶ D̶i̶f̶e̶r̶e̶n̶c̶i̶a̶i̶s̶ </̶h̶2̶>̶

+ <h2 id="diferenciais" class="subtitulo"> Diferenciais </h2>

<ul>

<li> Menor preço do mercado, garantido </li>

<li> Se você achar um serviço mais barato, leva 1 mês de graça </li> <li> Pague em reais, dólares, euros, libras, dodgecoins, litecoins ou bitcoins </li> <li> Todas as compras acompanham acesso à Alura </li>

<li> Maior escola de música online do mundo </li>

<li> Atendimento via telefone, email, chat, twitter, facebook, instagram, ICQ, WhatsApp, S MS, carta, fax, sinal de fumaça e telegrama </li>

<li> Presente em 124 países </li>

<li> Mais de um milhão de funcionários em todo o mundo </li>

</ul>

+

+ <h2 class="titulo"> Contato </h2>

+

16.2 PASSO A PASSO COM CÓDIGO **71**