Codificação Para leigos

Codificar envolve vocabulário, lógica e sintaxe. A princípio, a codificação parece intimidadora, mas, com a prática, sua terminologia, conceitos e estrutura se tornam familiares. Entendê-la é como aprender qualquer idioma: use-a com frequência para se tornar capaz de falar, pensar e escrever código. Ainda assim, é natural que os iniciantes tenham dúvidas. Existem muitos recursos de codificação disponíveis para você, tanto online quanto offline. Busque à sua volta e verá que não está sozinho — muitas outras pessoas estão aprendendo. Afinal, codificar é um eterno aprendizado. Domine uma faceta ou outra, e novas surgirão bem na sua frente.

VOCABULÁRIO DE CODIFICAÇÃO

A codificação tem um vocabulário extenso, que, para os iniciantes, às vezes parece uma bolha técnica impenetrável. Se estiver lendo um artigo online relacionado à codificação ou conversando com um desenvolvedor no trabalho, pode ouvir palavras que nunca escutou antes ou que têm um significado diferente no contexto da codificação. Aqui estão algumas palavras básicas:

- **Termos gerais de desenvolvimento web:**

  Servidor: Um computador que hospeda o código do site e o "envia" quando solicitado por um computador "cliente". Os servidores geralmente ficam em grandes galpões com milhares de outros, e são similares em tamanho e potência ao computador da sua casa.

  Cliente: Um dispositivo usado para acessar um site, incluindo computadores desktop ou laptop, tablets ou smartphones.

  Designer: Um profissional de arte que decide qual aparência e impressão o site transmitirá, além da maneira como os usuários interagirão com o site — como, por exemplo, clicar, deslizar, rolar, e assim por diante.

  Wireframe: Uma ilustração criada pelos designers que mostra em detalhes os layouts, imagens e esquemas de cores do site.

  Desenvolvedor: Um profissional de engenharia que escreve código para transformar wireframes em sites utilizáveis. Com base no tipo de código escrito, podem ser desenvolvedores de front-end, back-end ou full stack (ambos).

  Front-end: Tudo o que você vê e clica em um navegador. Desenvolvedores de front-end escrevem código em linguagens de front-end, como HTML, CSS e JavaScript, para criar a aparência do site.

  Back-end: Tudo o que acontece nos bastidores para fazer com que o front-end seja executado como planejado. Desenvolvedores de back-end escrevem código em linguagens de back-end, como Ruby ou Python, para criar funcionalidades como login de usuários, armazenamento de suas preferências e recuperação de dados, como comentários em uma foto.

RASCUNHO

- **Termos relacionados a linguagens de front-end:**

  ***HTML (Hypertext Markup Language):** Uma linguagem usada para posicionar textos, imagens e outros conteúdos em uma página.

  ***Tag HTML:** Instruções HTML, que geralmente aparecem em pares. Os navegadores aplicam efeitos especiais ao texto entre as tags HTML de abertura <elemento> e fechamento </elemento>. Por exemplo, a tag <h1> é renderizada no navegador como um grande título em negrito, e pode ser usada assim: <h1>Dewey vence Truman</h1>.

  ***Atributo HTML:** Atributos ou parâmetros modificam o comportamento de tags HTML. Os atributos são sempre posicionados na tag HTML de abertura. Por exemplo, href é o atributo da seguinte tag de âncora (usada para criar hyperlinks):

  ```
  <a href="http://www.google.com">Search engine</a>
  ```

  ***CSS (Cascading Style Sheets):** Código que modifica o HTML nas páginas e controla a aparência do conteúdo alterando tamanho de textos e imagens e outros atributos.

  ***JavaScript:** Código que adiciona interatividade e animação a páginas. O JavaScript também detecta eventos no navegador, como cliques do mouse, valida inserções de usuários, como as textuais, e recupera dados de sites externos.

  ***Variável:** Um local de armazenamento que recebe um nome e contém dados numéricos ou texto (chamados de strings) para uso posterior.

  ***Sentença if (condicional):** Uma instrução de código que testa uma condição, que comumente inclui variáveis, como x < 18, e executa o código escrito se a condição for verdadeira.

  ***Função:** O nome dado a um grupo de sentenças de programação para facilitar sua referenciação e uso.

- **Termos relacionados a linguagens de back-end:**

  ***Ruby:** Uma linguagem de programação de código aberto mais conhecida pelo uso na programação web.

  ***Rails:** Um framework desenhado para facilitar a criação de páginas com Ruby.

  ***Python:** Uma linguagem de programação de código aberto usada na web, em aplicativos científicos, e para análise de dados.

RASCUNHO

Codificação

Para leigos

RASCUNHO

por Nikhil Abraham

ALTA BOOKS
EDITORA
Rio de Janeiro, 2019

Sumário

INTRODUÇÃO 1

Sobre Este Livro 2
Penso que... 2
Ícones Usados Neste Livro 3
Além Deste Livro 3
De Lá para Cá, Daqui para Lá 4

PARTE 1: COMEÇANDO 5

CAPÍTULO 1: **O que É Codificação?** 7

Definindo o que São Códigos 8
Seguindo instruções 8
Codificando com Angry Birds 9
Descubra o que a Codificação Pode Fazer por Você 10
Devorando o mundo com softwares 10
Codificação no trabalho 12
Faça por conta própria (e fique rico e famoso) 13
Examinando os Tipos de Linguagem de Programação 13
Comparando linguagens de programação de baixo e alto níveis 15
Código compilado contrastante e código interpretado 16
Programando para a web 16
Percorrendo um Aplicativo da Web Construído com Código 16
Definindo o propósito e o escopo do aplicativo 17
Apoiando-se nos ombros de gigantes 17

CAPÍTULO 2: **Programação para a Web** 19

Exibindo Páginas da Web em Seu Desktop ou Dispositivo Móvel . . 20
Pirateando seu site de notícias favorito 20
Entendendo como a World Wide Web funciona 23
Observando seu front-end e back-end 24
Definindo aplicativos web e móveis 25
Codificação de Aplicativos Web 26
Começando com HTML, CSS e JavaScript 27
Adicionando lógica com Python, Ruby ou PHP 27
Codificação de Aplicativos Móveis 28
Construindo aplicativos móveis 29
Construindo aplicativos nativos móveis 30

CAPÍTULO 3: **Tornando-se Programador** 33

Escrevendo Códigos com um Método 34
Pesquisar o que deseja construir 36

RASCUNHO

Projetar seu app 36
Codificando seu app 38
Depurando seu código 38
Escolhendo Ferramentas para Trabalhar 39
Trabalhando offline 39
Trabalhando online com a Codecademy.com 40

PARTE 2: CONSTRUINDO PÁGINAS SILENCIOSAS E INTERATIVAS 43

CAPÍTULO 4: **Explorando o HTML Básico** 45
O que o HTML Faz? 46
Entendendo a Estrutura do HTML 46
Identificando elementos 47
Apresentando seu melhor atributo 48
Mantendo cabeçalho, título e corpo de texto acima de tudo . . . 50
Familiarizando-se com Tarefas e Tags HTML Comuns 51
Escrevendo títulos 53
Organizando textos em parágrafos 54
Vinculando seu conteúdo (a seu gosto) 54
Adicionando imagens 56
Formate-me Bem 57
Destacando com negritos, itálicos, sublinhados e riscados 57
Aumentando e diminuindo textos com sobrescrito e subscrito 58
Construindo Seu Primeiro Site com HTML 59

CAPÍTULO 5: **Conheça Melhor o HTML** 63
Organizando o Conteúdo na Página 64
Listando Dados 65
Crie listas ordenadas e não ordenadas 66
Aninhando listas 67
Colocando Dados em Tabelas 68
Estruturação básica de tabela 68
Expandindo colunas e linhas da tabela 70
Alinhando tabelas e células 71
Preenchendo Formulários 74
Entenda como formulários funcionam 74
Criando formulários básicos 76
Praticando Mais o HTML 77

CAPÍTULO 6: **Formatando com CSS** 79
O que o CSS Faz? 80
Estrutura CSS 81
Escolhendo o elemento a formatar 81
Minha propriedade tem valor 83
Pirateando seu site favorito com CSS 84

RASCUNHO

Tarefas e Seletores Comuns do CSS 86
Ginástica da fonte: Tamanho, cor, estilo, família e decoração . . 86
Personalizando links 90
Adicionando imagens de segundo plano e formatando as do primeiro 92
Formate-me Bem 96
Adicione CSS a seu HTML 97
Construa sua primeira página 99

CAPÍTULO 7: **Próximos Passos com CSS** 101
Formatando (Mais) Elementos em Sua Página 102
Formatando listas 102
Desenhando tabelas 106
Selecionando Elementos para Aplicar Estilo 108
Aplicando estilo a elementos específicos 108
Nomeando elementos HTML 112
Alinhando e Organizando Seus Elementos 114
Organizando dados na página 114
Modelando a div 116
Entendendo o modelo de caixas 117
Posicionando as caixas 119
Escrevendo CSS Mais Avançado 122

CAPÍTULO 8: **Trabalhando Mais Rápido com o Twitter Bootstrap** 123
Descobrindo o que o Bootstrap Faz 124
Instalando o Bootstrap 125
Entendendo as Opções de Layout 126
Alinhando no sistema de grade 127
Arrastando e soltando para um site 129
Usando templates predefinidos 130
Adaptando layout para dispositivos móveis, tablets e desktops 131
Codificando Elementos Básicos de uma Página 133
Desenhando botões 134
Navegando com barras de ferramentas 135
Adicionando ícones 137
Construindo a Página Inicial do Airbnb 138

CAPÍTULO 9: **Incluindo JavaScript** 139
O que o JavaScript Faz? 140
Entendendo a Estrutura do JavaScript 142
Usando Ponto e vírgula, Aspas, Parênteses e Chaves 142
Codificando Tarefas JavaScript Comuns 143
Armazenando dados com variáveis 143
Tomando decisões com sentenças if-else 144
Trabalhando com métodos de string e números 148
Alertando e solicitando inserções de usuários 150

RASCUNHO

Nomeando código com funções 151
Adicionando JavaScript às páginas. 152
Escrevendo Seu Primeiro Programa de JavaScript 153
Trabalhando com APIs 154
O que as APIs fazem? 154
Raspagem de dados sem API 157
Buscando e escolhendo uma API. 158
Usando as Bibliotecas JavaScript 158
jQuery 159
D3.js. 159
Procurando Vídeos com a API do YouTube 160

PARTE 3: CRIANDO UM APLICATIVO WEB 163

CAPÍTULO 10: **Construindo Seu App** 165
Criando um Aplicativo de Ofertas Baseadas na Localização 166
Entendendo a situação 166
Traçando os próximos passos 167
Seguindo o Processo de Desenvolvimento de um App 167
Planejando Seu Primeiro Aplicativo 168
Explorando o Processo Geral 169
Conheça as Pessoas que Dão Vida aos App Web 171
Criando com designers 171
Codificando com desenvolvedores de front-end e back-end . . 173
Gestão com gerentes de produto 174
Teste com garantia de qualidade. 175

CAPÍTULO 11: **Pesquisando Seu Primeiro Aplicativo Web** 177
Dividindo o App em Etapas 178
Encontrando a funcionalidade de seu app 178
Encontrando a funcionalidade de seu app: Minha versão 179
Encontrando o formato de seu app 180
Encontrando o formato de seu app: O design do app de ofertas do McDuck's. 184
Identificando Fontes de Pesquisa. 186
Pesquisando as Etapas para o App de Ofertas do McDuck's 188
Escolha uma Solução para Cada Etapa 190

CAPÍTULO 12: **Codificando e Depurando Seu Primeiro App Web** 193
A Preparação 193
Codificando Seu Primeiro App Web. 194
Ambiente de desenvolvimento. 194
Código pré-escrito 195
Etapas de codificação para você seguir 198
Depurando Seu App 201

RASCUNHO

PARTE 4: AMPLIANDO SUAS HABILIDADES 203

CAPÍTULO 13: **Familiarizando-se com Ruby** 205
O que o Ruby Faz? 206
Definindo a Estrutura do Ruby 207
Entendendo os princípios do Ruby 207
Estilo e espaçamento 208
Codificando Tarefas e Comandos Ruby Básicos 209
Definindo tipos de dados e variáveis 210
Computação matemática simples e avançada 210
Usando strings e caracteres especiais 211
Decidindo com condicionais: if, elsif, else 212
Inserções e saídas 214
Formatando Suas Strings 215
Métodos de string: upcase, downcase e strip 215
Inserindo variáveis em strings com # 216
Construindo um Processador Simples de Texto com Ruby 217

CAPÍTULO 14: **Mergulhando no Python** 219
O que o Python Faz? 220
Definindo a Estrutura do Python 221
Entendendo o Zen do Python 221
Formatação e espaçamento 222
Codificando Tarefas e Comandos de Python Comuns 223
Definindo tipos de dados e variáveis 224
Computação matemática simples e avançada 225
Usando strings e caracteres especiais 226
Decidindo com condicionais: if, elif, else 227
Inserções e saídas 228
Formatando Suas Strings 229
Notação de ponto com upper(), lower(), capitalize() e strip() . . . 230
Formatando strings com % 230
Construindo uma Calculadora Simples com Python 231

PARTE 5: A PARTE DOS DEZ 233

CAPÍTULO 15: **Dez Recursos Gratuitos para Codificadores** 235
Sites que Ensinam a Codificar 236
Codecademy 236
Coursera e Udacity 236
Hackdesign.org 237
Code.org 238
Sites de Referência para Codificação 238
W3Schools 238
Mozilla Developer Network 239
Stack Overflow 240

RASCUNHO

Sites de Notícias de Tecnologia e Comunidades 240
TechCrunch 240
Hacker News 241
Meetup 242

CAPÍTULO 16: **Dez Dicas para os Novatos** 243

Escolha uma Linguagem, Qualquer uma 243
Defina um Objetivo 244
Divida Seu Objetivo em Pequenos Pedaços 245
Diferencie o Bolo da Cereja 246
O Google É Seu Melhor Amigo 246
Corrija Aqueles Bugs 247
Envie 248
Colete Feedback 249
Itere Seu Código 249
Compartilhe Seus Sucessos e Falhas 250

ÍNDICE 251

RASCUNHO

1 Começando

RASCUNHO

NESTA PARTE . . .

Compreenda códigos e o que se pode construir com eles.

Revise linguagens de programação usadas para escrevê-los.

Codifique para web com linguagens de front-end e back-end.

Siga o processo usado pelos programadores para criá-los.

Escreva seu primeiro programa com códigos.

RASCUNHO

NESTE CAPÍTULO

- **Descobrindo o que códigos são e o que podem fazer**
- **Passeando por seu primeiro programa com códigos**
- **Entendendo as linguagens de programação usadas para escrever códigos**

RASCUNHO

Capítulo 1
O que É Codificação?

Um milhão de dólares não é legal, sabe o que é legal? Um bilhão.

— SEAN PARKER, A REDE SOCIAL

Todas as semanas, os jornais informam sobre outra empresa de tecnologia que levantou capital ou foi vendida por milhões de dólares. Às vezes, no caso de empresas como Instagram, WhatsApp e Uber, o valor na chamada é de bilhões. Talvez, esses artigos agucem sua curiosidade e façam com que queira descobrir como os códigos são usados para construir aplicativos que promovem esses resultados financeiros. Alternativamente, seus interesses podem ser mais profissionais. Talvez trabalhe em um setor em declínio, como a mídia impressa, ou em uma função que a tecnologia se transforma rapidamente, como o marketing. Independente de pensar em mudar de carreira ou aprimorar a atual, entender programação ou "codificação" computacional alavanca seu desenvolvimento profissional. Por fim, seu interesse pode ser mais pessoal — talvez você tenha uma ideia, um grande desejo de criar algo, um site ou um aplicativo, para resolver um problema que vivenciou e sabe que ler e escrever códigos é o primeiro passo para chegar à solução. Qualquer que seja sua motivação, este livro dará uma luz sobre codificação e programadores, e o ajudará a pensar em ambos não como misteriosos e intricados, mas acessíveis e algo que você mesmo pode fazer.

Neste capítulo, você entenderá o que são códigos, quais setores são afetados por softwares computacionais, os tipos de linguagem de programação em que são escritos e fará um percurso por um aplicativo da web construído com códigos.

Definindo o que São Códigos

Códigos computacionais não são uma atividade enigmática reservada a gênios e oráculos. Na verdade, em poucos minutos você mesmo escreverá um! A maioria dos códigos computacionais executa uma variedade de tarefas em nossas vidas, do mundano ao extraordinário. Códigos são responsáveis por nossos semáforos, elevadores em nossos edifícios, torres que transmitem sinal para nossos celulares e naves aeroespaciais. Também interagimos com códigos em um nível mais pessoal, em nossos smartphones e computadores, e eventualmente para conferir o e-mail ou o clima.

Seguindo instruções

Códigos computacionais são um conjunto de instruções, frases geralmente em inglês, e cada uma direciona o computador para executar uma etapa ou instrução. Cada uma dessas etapas é muito precisa e seguida à risca. Por exemplo, se estiver em um restaurante e pedir a um garçom que lhe indique o banheiro, ele poderia dizer: "Siga até os fundos, é a porta do meio." Para um computador, essas instruções são tão vagas que são inúteis. Em vez disso, se o garçom lhe orientasse como a um programa de computador, diria: "A partir desta mesa, caminhe 40 passos para o nordeste. Em seguida, vire 90º à direita, dê cinco passos, vire 90º à esquerda e dê mais cinco passos. Abra a porta imediatamente à frente e entre no banheiro." A Figura 1-1 mostra linhas de código do famoso jogo Pong. Não se preocupe em entender o que cada linha significa nem se sinta intimidado. Em breve, estará lendo e escrevendo seu próprio código.

```
launchPong(function () {
    function colour_random() {
        var num = Math.floor(Math.random() * Math.pow(2, 24));
        return '#' + ('00000' + num.toString(16)).substr(-6);
    }


    pongSettings.ball.size = 15;
    pongSettings.ball.color = colour_random();
    pongSettings.ball.velocity[0] = 15;
    pongSettings.ball.velocity[1] = 15;

});
```

FIGURA 1-1: Código do jogo Pong.

RASCUNHO

Uma maneira difícil de mensurar a complexidade de um programa é contar suas sentenças ou linhas de código. Aplicativos básicos, como o Pong, têm 5 mil linhas de código, enquanto aplicativos mais complexos, como o Facebook, atualmente possuem mais de 10 milhões. Independente de haver poucas ou muitas linhas de código, o computador segue precisamente cada instrução sem esforço, não fica cansado como o garçom após lhe perguntarem pela centésima vez onde fica o banheiro.

DICA

Tenha cuidado ao usar somente as linhas de código para mensurar a complexidade de um programa. Como escrever um texto, 100 boas linhas de código agem com a mesma funcionalidade que 1.000 mal escritas.

Codificando com Angry Birds

Se nunca escreveu códigos, agora é sua chance de tentar! Vá a: `http://csedweek.org/learn` e, abaixo de "Tutorials for Beginners", clique em "Write Your First Computer Program", o link com o ícone do Angry Birds, como mostrado na Figura 1-2 [conteúdo em inglês]. Esse tutorial se destina àqueles sem experiência em programação e apresenta os blocos de construção básicos usados por todos os programas de computador. O objetivo principal desse tutorial é entender que os programas usam códigos para dizer literal e exatamente ao computador para executar um conjunto de instruções.

FIGURA 1-2: Escreva seu primeiro programa de computador com um tutorial de jogo similar ao Angry Birds.

DICA

O Computer Science Education Week é um programa anual dedicado a destacar a imagem da ciência da computação, que dura uma semana em dezembro. No passado, Obama, Bill Gates, o atleta do basquete Chris Bosh e a cantora Shakira, entre outros, apoiaram e encorajaram pessoas de todo o mundo a participar.

RASCUNHO

Descubra o que a Codificação Pode Fazer por Você

A codificação é usada para executar tarefas e resolver problemas que você vivencia todos os dias. As situações "cotidianas" às quais programas e aplicativos assistem continuam a crescer exponencialmente, mas nem sempre foi assim. O aumento de aplicativos da web, conectividade com a internet e telefones celulares inseriu programas de software no dia a dia e reduziu a barreira para que você se torne um criador, resolvendo com códigos problemas pessoais e profissionais.

Devorando o mundo com softwares

Em 2011, Marc Andreessen, criador do Netscape Navigator e, agora, capitalista de risco, observou que: "Softwares estão devorando o mundo." Ele previu que as empresas de software perturbariam as outras em um ritmo acelerado. De modo geral, softwares escritos para desktops e laptops. O software precisava primeiro ser instalado, então você fornecia os dados para o programa. Três tendências aumentaram drasticamente o uso dos códigos no cotidiano:

- **Softwares baseados na web:** Funcionam no navegador sem necessidade de instalação. Por exemplo, se quisesse verificar seu e-mail, primeiro teria que instalar um cliente de e-mail, baixando o software ou por meio de um CD-ROM. Às vezes, surgiam problemas quando o software não estava disponível para seu sistema operacional ou quando havia conflitos com a versão do sistema. O Hotmail, um cliente de e-mail baseado na web, em parte se popularizou porque permitia que os usuários visitassem `www.hotmail.com` para conferir instantaneamente o e-mail sem se preocuparem com instalação de softwares ou compatibilidade. Aplicativos da web aumentaram o apetite do consumidor para experimentar mais aplicativos, e os desenvolvedores, por sua vez, foram incentivados a escrever mais deles.
- **Banda larga:** A conectividade de banda larga se expandiu, proporcionando uma conexão rápida à internet para mais pessoas nos últimos anos mais do que na última década. Hoje, mais de dois bilhões de pessoas podem acessar softwares baseados na web, contra aproximadamente 50 milhões apenas uma década atrás.
- **Telefones celulares:** Os smartphones atuais carregam programas com você aonde quer que vá, e ajudam a fornecer dados a esses programas. Muitos programas de software se tornam mais práticos quando acessados em movimento do que limitados a um desktop. Por exemplo, o uso de aplicativos de mapas aumentou bastante graças aos telefones celulares porque os

RASCUNHO

usuários precisam de orientações principalmente quando estão perdidos, não apenas ao planejar uma viagem em casa, no computadores. Além disso, telefones celulares são equipados com sensores que medem e fornecem dados para programas de orientação, aceleração e localização atual via GPS. Agora, em vez de você mesmo ter que inserir todos os dados nos programas, os dispositivos móveis o ajudam. Por exemplo, o aplicativo de atividades físicas RunKeeper não requer que você insira horários de início e término para acompanhar a execução. Basta pressionar o botão de começar no início e o telefone rastreará automaticamente sua distância, velocidade e tempo.

A combinação dessas tendências criou empresas de software que levaram os operadores históricos a praticamente todos os setores, em particular aos tipicamente imunes à tecnologia. Alguns exemplos notórios incluem:

- **Airbnb:** O Airbnb é uma empresa de hospedagem peer-to-peer que não possui nenhum quarto, mas reserva mais noites que a Hilton e a Intercontinental, maiores cadeias hoteleiras do mundo. (Veja a Figura 1-3.)
- **Uber:** A Uber é uma empresa de transporte que não possui veículos, faz mais corridas e tem mais motoristas nas principais 200 cidades do que qualquer outro serviço de táxi ou automóveis.
- **Groupon:** O Groupon, uma empresa de e-commerce, gerou quase $1 bilhão depois de apenas dois anos de existência, crescendo mais rápido do que qualquer outra empresa na história e mais ainda do que qualquer outra tradicional de marketing direto.

FIGURA 1-3: O Airbnb reservou 5 milhões de noites após 3,5 anos de existência, e os outros 5 milhões, 6 meses depois.

RASCUNHO

Codificação no trabalho

A codificação também é útil no local de trabalho. Fora do setor tecnológico, é comum no ambiente de trabalho de algumas profissões, como negociantes financeiros, economistas e cientistas. Entretanto, para a maioria dos profissionais fora do setor tecnológico, a codificação está começando a chegar e, gradualmente, a ganhar relevância. Aqui estão as áreas em que ela desempenha um papel maior:

- **Publicidade:** Padrões de consumo passam de campanhas impressas e televisivas para digitais, e publicidade em mecanismos de pesquisa e otimização dependem de palavras-chave para levar visitantes aos sites. Anunciantes que entendem de codificação veem palavras-chave usadas com sucesso pelos concorrentes e utilizam os dados para criar campanhas mais eficazes.
- **Marketing:** Ao promover produtos, personalizar a comunicação é uma estratégia que costuma incrementar os resultados. Marqueteiros que codificam consultam bancos de dados de clientes e adaptam as comunicações para incluir seus nomes e produtos adaptados aos seus interesses.
- **Vendas:** O processo de vendas sempre começa com chamadas. Vendedores que codificam recuperam as próprias chamadas de páginas da web e diretórios e depois as classificam e qualificam.

  DICA

  A recuperação de informações copiando textos de páginas da web ou de diretórios se chama *raspagem de dados*.
- **Design:** Depois de criar uma página ou um design digital, designers persuadem outros e, finalmente, desenvolvedores a realmente programar seus desenhos em um produto. Designers que codificam podem dar vida a seus projetos mais facilmente, e defender projetos específicos com mais eficácia criando protótipos de trabalho com que os outros possam interagir.
- **Relações-públicas:** As empresas mensuram constantemente como os clientes e o público reagem a anúncios e notícias. Por exemplo, se o porta-voz de um figurão de uma empresa faz ou diz algo ofensivo, a empresa deveria demiti-lo? Relações-públicas que codificam investigam redes sociais como Twitter ou Facebook e analisam milhares de mensagens para entender o sentimento do mercado.
- **Operações:** O lucro adicional pode ser gerado, em parte, pela análise de custos de uma empresa. Profissionais de operações que codificam escrevem programas para testar milhões de combinações que otimizem métodos de embalagem, rotinas de carregamento e rotas de entrega.

RASCUNHO

Faça por conta própria (e fique rico e famoso)

Usar códigos construídos pelos outros e a codificação no local de trabalho pode levá-lo a pensar que os problemas que deve encarar pessoalmente seriam melhor resolvidos com um código próprio. Você pode ter uma ideia para um site de mídia social, um aplicativo melhorado de atividades físicas ou algo completamente novo. O caminho da ideia ao protótipo funcional, usado pelos outros, envolve uma boa quantidade de tempo e trabalho, mas pode ser mais viável do que você pensa. Por exemplo, veja o Coffitivity, um site de produtividade que transmite som ambiente de cafeterias para criar ruído branco. O site foi criado por duas pessoas que tinham aprendido a programar alguns meses antes. Pouco depois do lançamento do Coffitivity, a Time Magazine o nomeou como um dos 50 Best Websites de 2013, e o *Wall Street Journal* também o avaliou. Embora nem toda startup ou aplicativo receba essa cobertura da mídia inicialmente, é útil saber que ela acontece quando uma solução realmente resolve um problema.

Ter um objetivo, como um site ou aplicativo que deseja construir, é uma das melhores maneiras de aprender a codificar. Quando se enfrenta um bug difícil ou um conceito complexo, a ideia de dar vida a seu site o motivará a continuar. Tão importante quanto isso, não aprenda a codificar para ficar rico e famoso, já que a possibilidade de seu site ou aplicativo ser bem-sucedido se deve amplamente a fatores fora do seu controle.

DICA

As características que tornam um site ou aplicativo viciantes são descritas usando o Hook Model aqui `http://techcrunch.com/2012/03/04/how-to-manufacture-desire`. Os produtos geralmente são feitos por empresas e as características de uma empresa duradoura estão aqui `http://www.sequoiacap.com/grove/posts/yal6/elements-of-enduring-companies`, baseadas em uma avaliação das que são financiadas pela Sequoia, uma das empresas de capital aberto mais bem-sucedidas do mundo e investidoras iniciais da Apple, do Google e do PayPal.

Examinando os Tipos de Linguagem de Programação

A codificação tem diferentes sabores, chamados de *linguagens de programação.* Algumas populares são mostradas na Figura 1-4.

RASCUNHO

FIGURA 1-4: Algumas linguagens de programação populares.

Você pode pensar em linguagens de programação como na língua falada, pois ambas compartilham características em comum, tais como:

- **Funcionalidade em todos os idiomas:** Linguagens de programação criam as mesmas funcionalidades, similar à forma como os idiomas expressam os mesmos objetos, frases e emoções.
- **Sintaxe e estrutura:** Comandos em linguagens de programação se sobrepõem como palavras na língua falada. Para colocar texto na tela em Python ou Ruby, você usa o comando `print`, assim como imprimir e imprimir são os verbos para "print" em francês e português.
- **Ciclo natural de vida:** Uma linguagem de programação nasce quando um programador pensa em uma maneira mais nova ou fácil de expressar um conceito computacional. Se outros programadores a aprovarem, adotam para os próprios programas e ela se espalha. No entanto, assim como o latim ou o aramaico, se não for empregada por outros programadores ou se uma melhor vier, ela morre lentamente pela falta de uso.

Apesar dessas semelhanças, as linguagens de programação diferem dos idiomas falados em alguns aspectos-chave:

- **Criador único:** Diferente da língua falada, linguagens de programação podem ser criadas por uma pessoa em um curto período de tempo, às vezes em pouco dias. Linguagens populares com um único criador incluem JavaScript (Brendan Eich), Python (Guido van Rossum) e Ruby (Yukihiro Matsumoto).

RASCUNHO

» **Escrita em inglês:** Ao contrário dos idiomas (exceto, é claro, o inglês), quase todas as linguagens de programação são escritas em inglês. Se estiverem programando em HTML, JavaScript, Python ou Ruby, programadores brasileiros, franceses e chineses usam as mesmas palavras-chave e sintaxa em inglês em seus códigos. Existem linguagens de programação que não são em inglês, como algumas em hindi ou árabe, mas nenhuma delas é popular ou convencional.

Comparando linguagens de programação de baixo e alto níveis

Uma maneira de classificar linguagens de programação é como linguagem de baixo ou alto níveis. Linguagens de baixo nível interagem diretamente com o processador do computador, ou CPU, são capazes de executar comandos muito básicos e, geralmente, difíceis de ler. Codificação de máquina, um exemplo de linguagem de baixo nível, usa códigos que consistem em apenas dois números 0 e 1. A Figura 1-5 mostra um exemplo de codificação de máquina. Linguagem de montagem, mais uma de baixo nível, usa palavras-chave para executar comandos básicos, como ler, mover e armazenar dados.

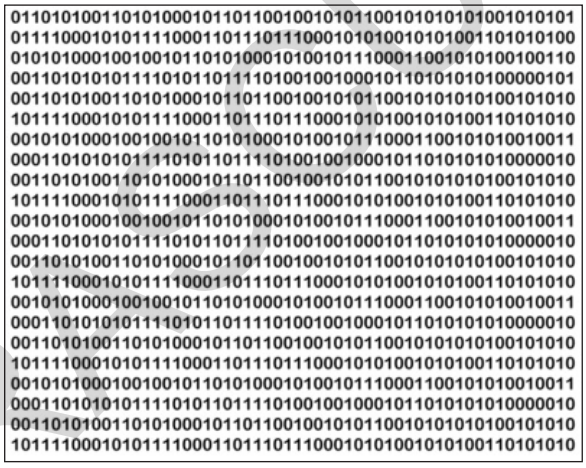

FIGURA 1-5: A codificação de máquina consiste em 0s e 1s.

Por outro lado, linguagens de alto nível usam a linguagem natural, por isso é mais fácil que as pessoas leiam e escrevam. Quando o código é escrito em uma linguagem de nível alto, como C++, Python ou Ruby, um interpretador ou compilador a traduz para um código de baixo nível para que o computador a entenda.

PRELIMINAR