Use a Cabeça! Programação JavaScript

Não seria maravilhoso se houvesse um livro de JavaScript que fosse mais divertido que uma ida ao dentista e mais revelador do que um formulário do imposto de renda? É só uma fantasia, mesmo...



Eric T. Freeman Elisabeth Robson



Autores do Use a Cabeça Programação JavaScript







Eric é descrito pela co-criadora da série Use a Cabeça, Kathy Sierra, como "um daqueles raros indivíduos fluentes na língua, na prática e na cultura de múltiplos domínios como hipster hacker, VP corporativo, engenharia, think tank"

Profissionalmente, Eric recentemente encerrou quase uma década como executivo de companha de mídia — tendo sido CTO da Disney Online & Disney.com na The Walt Disney Company. Agora está devotando seu tempo à WickedlySmart, uma startup que ele criou junto com Elisabeth.

De formação, Eric é um cientista da computação, tendo estudado com o gênio da indústria David Gelernter durante seu trabalho de PhD na Universidade de Yale. Sua dissertação é creditada como um trabalho criativo nas alternativas para a metáfora de desktop e também como a primeira implementação de activity streams, um conceito desenvolvido por ele e pelo Dr. Gelernter.

Nas horas vagas, Eric é profundamente envolvido com música; você pode encontrar o último projeto dele, uma colaboração com o pioneiro da música ambiente Steve Roach, disponível no app store do iPhone com o nome de Immersion Station.

Eric vive com sua esposa e filha em Bainbridge Island. Sua filha sempre frequenta o estúdio dele, onde ela ama mexer nos botões dos sintetizadores e dos efeitos de áudio.

Visite o site dele em http://ericfreeman.com (em inglês).

Elisabeth é uma engenheira de software, escritora e treinadora. Ela é apaixonada por tecnologia desde seus dias como estudante na Universidade de Yale, onde obteve o título de Mestre em Ciências da Computação e projetou uma arquitetura de software e uma linguagem de programação visual concorrentes.

A Elisabeth está envolvida com a internet desde seus primeiros dias; ela é cocriadora do website premiado The Ada Project, um dos primeiros websites projetados para ajudar as mulheres da ciência da computação a encontrar informações sobre carreira e orientação online.

Ela é atualmente cofundadora da WickedlySmart, uma experiência de educação online centrada em tecnologias web, onde ela cria livros, artigos, vídeos e muito mais. Previamente, como Diretora de Projetos Especiais da O'Reilly Media, ela produziu workshops presenciais e cursos online de uma variedade de tópicos técnicos e desenvolveu sua paixão pela criação de experiências de aprendizado para ajudar as pessoas a entenderem a tecnologia. Antes de seu trabalho com a O'Reilly, ela passou um tempo espalhando pó de fadas na The Walt Disney Company, onde liderou esforços de pesquisa e desenvolvimento em mídia digital.

Quando não está na frente de seu computador, você vai encontrar a Elisabeth fazendo trilhas, andando de bicicleta ou de caiaque, com sua câmera por perto, ou então cozinhando comida vegetariana.

Visite o blog dela em http://elisabethrobson.com (em inglês).

viii

Conteúdo (Sumário)

	Introdução	XXV
1	um rápido mergulho no JavaScript Molhando os pés	1
2	escrevendo código de verdade Indo além	43
3	introduzindo as funções Tornando-se funcional	79
4	colocando um pouco de ordem nos seus dados Arrays	125
5	entendendo objetos Uma viagem a Objectville	173
6	interagindo com sua página web Conhecendo o DOM	229
7	tipos, igualdade, conversão e tudo mais Tipos sérios	265
8	juntando tudo Construindo um app	317
9	codificação assíncrona Manipulando eventos	381
10	funções de primeira classe Funções liberadas	429
11	funções anônimas, escopo e closures Funções sérias	475
12	construção avançada de objetos Criando objetos	521
13	usando protótipos Objetos extrafortes	563
Aр	êndice: Os Top Dez Tópicos (que não cobrimos)	
	as sobras	623

Conteúdo (a coisa real)

Introdução

Seu cérebro e o JavaScript. Aqui está você tentando aprender alguma coisa, enquanto seu cérebro lhe faz o favor de garantir que o aprendizado não ocorra. Seu cérebro está pensando: "Melhor deixar espaço para coisas mais importantes, como que animais selvagens evitar ou se fazer snowboarding pelado é uma má ideia." Então como você engana seu cérebro para que ele pense que sua vida depende de saber programação JavaScript?



Para quem é este livro?	xxvi
Sabemos o que você está pensando.	xxvii
Pensamos no leitor do "Use a Cabeça" como	
um aprendiz.	xxviii
Metacognição: pensando sobre pensar.	xxix
Eis o que NÓS fizemos:	XXX
Eis o que VOCÊ pode fazer para exercitar seu	
cérebro à sua vontade.	xxxi
Leia-me	xxxii
Revisores Técnicos	XXXV
Agradecimentos*	xxxvi

ix

um rápido mergulho no JavaScript

Molhando os pés

O JavaScript lhe dá superpoderes. A verdadeira linguagem de programação da web, o JavaScript, permite que você adicione comportamento às suas páginas web. Diga adeus às páginas chatas e estáticas que ficam apenas paradas olhando para você — com o JavaScript você será capaz de alcançar e tocar seus usuários, reagir a eventos interessantes, obter dados da web para usar em suas páginas, desenhar gráficos diretamente nas suas páginas web e muito mais. E uma vez que você domine JavaScript, será capaz de criar

Você estará em boa companhia também, o JavaScript não apenas é uma das linguagens de programação **mais populares**, também é **suportado** em todos os navegadores modernos (e na maioria dos antigos); o JavaScript está até se expandindo e sendo **embutido** em muitos ambientes fora do navegador. Falaremos sobre isso mais tarde; agora, vamos começar!

comportamentos totalmente novos para seus usuários.



A maneira como o JavaScript funciona	2
Como você vai escrever JavaScript	3
Como colocar JavaScript na sua página	4
JavaScript, você já avançou muito	6
Como fazer uma instrução	10
Variáveis e valores	11
Afaste-se desse teclado!	12
Expresse-se	15
Fazendo as coisas mais de uma vez	17
Como funciona o loop while	18
Tomando decisões com o JavaScript	22
E, quando você precisa tomar MUITAS decisões	23
Comunique-se com seu usuário	25
Um olhar mais de perto no console.log	27
Abrindo o console	28
Codificando uma Aplicação JavaScript Séria	29
Como adiciono código na minha página?	
(deixe-me contar as maneiras)	32
Vamos ter que separar vocês dois	33



X

44

44

escrevendo código de verdade

Indo além

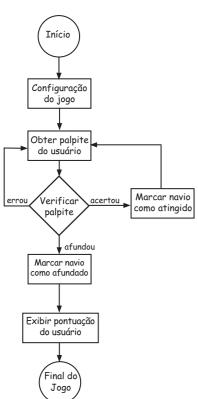
Você já sabe sobre variáveis, tipos, expressões...

e poderíamos continuar. A questão é: você já sabe algumas coisas sobre JavaScript. Na verdade, você sabe o suficiente para escrever algum código de verdade, algum código que faça algo interessante, algum código que alguém possa querer usar. O que lhe falta é a experiência real de escrever código, e nós vamos consertar isso aqui e agora. Como? Mergulhando de cabeça e codificando um jogo casual, todo escrito em JavaScript. Nosso objetivo é ambicioso, mas vamos passo a passo. Vamos lá, vamos começar e se você quiser lançar a nova startup casual, não haverá empecilhos, pois o código é seu.

Vamos construir um jogo de Batalha Naval

Nossa primeira tentativa...





Primeiro, um design de alto nível 45 Trabalhando o Pseudocódigo 47 49 Ah, antes de prosseguirmos, não se esqueça do HTML! Escrevendo o código do Batalha Naval Simples 50 Agora vamos escrever a lógica do jogo 51 Passo Um: preparando o loop, obtendo alguma entrada 52 Como o prompt funciona 53 Verificando o palpite do usuário 54 Então, acertamos? 56 Adicionando o código de detecção de acerto 57 Forneça alguma análise pós-jogo 58 E isto completa a lógica! 60 61 Fazendo um pouco de Controle de Qualidade 65 Podemos falar sobre sua verbosidade... Finalizando o jogo Batalha Naval Simples 66 Como atribuir localizações aleatórias 67 A receita mundialmente famosa para gerar 67 um número aleatório De volta para um pouco mais de CQ 69 Parabéns pelo seu primeiro e verdadeiro programa em JavaScript e uma palavrinha sobre a reutilização 71 de código

Whoa. A real flowchart.

introduzindo as funções

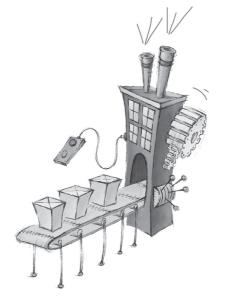
Tornando-se funcional

3

Prepare-se para seu primeiro superpoder. Você já sabe algumas coisas sobre programação, agora é hora de realmente avançar as coisas com funções. As funções lhe dão o poder de escrever código que pode ser aplicado em várias circunstâncias diferentes, código que pode ser reutilizado de novo e de novo, código que é muito mais gerenciável, código que pode ser abstraído e receber um nome simples de forma que você possa esquecer toda a complexidade e focar nas coisas importantes. Você vai descobrir não apenas que as funções são sua passagem de criador de scripts para programador, elas são a chave para o estilo de programação JavaScript. Neste capítulo, vamos começar com o básico: a mecânica, os detalhes de como as funções realmente funcionam e, então, você vai continuar afiando suas habilidades com funções pelo resto do livro. Vamos começar com uma boa base agora.



O que está errado com esse código?	81
Só por curiosidade, já mencionamos FUNÇÕES?	83
Ok, como isso funciona na verdade?	84
O que você pode passar para uma função?	89
JavaScript é passagem por valor	92
Funções podem retornar coisas também	95
Analisando uma função com uma instrução return	96
Variáveis globais e locais	99
Sabendo o escopo das suas variáveis locais e globais	101
As vidas curtas das variáveis	102
Não se esqueça de declarar suas variáveis locais!	103



хii

4

colocando um pouco de ordem nos seus dados

Arrays

Há mais no JavaScript do que números, strings e booleanos.



Até agora, você vinha escrevendo código JavaScript com **primitivas** — strings simples, números e booleanos como "Fido", 23 e true. E você pode fazer muito com tipos primitivos, mas em algum momento você vai ter que lidar com **mais dados**. Digamos, todos os itens de um carrinho de compras, ou todas as músicas de uma playlist, ou um conjunto de estrelas e sua magnitude aparente, ou um catálogo inteiro de produtos. Para isso precisamos de um pouco mais de *poder*. O tipo adequado para esse tipo de dados ordenados é um **array** JavaScript, e neste capítulo vamos ver como colocar seus dados em um array, como passá-los para outras estruturas e como operar em cima dele. Vamos olhar para algumas outras maneiras de **estruturar seus dados** em capítulos mais à frente, mas vamos começar com arrays.

Você pode ajudar a Bubbles-R-Us?	126
Como representar múltiplos valores em JavaScript	127
Como os arrays funcionam	128
Então qual é o tamanho desse array?	130
O Phrase-O-Matic	132
Enquanto isso, de volta à Bubbles-R-Us	135
Como iterar um array	138
Mas espere, há uma maneira melhor de	
iterar um array	140
Podemos falar sobre sua verbosidade?	146
Refazendo o loop com o operador de	
pós-incrementação	147
Um rápido test drive	147
Criando um array do zero (e adicionando a ele)	151
E os vencedores são	155
Uma rápida verificação no código	157
Escrevendo a função printAndGetHighScore	158
Refatorando o código usando printAndGetHighScore	159
Organizando as coisas	161



xiii

entendendo objetos



Uma viagem a Objectville

Até agora você tem utilizado primitivos e arrays

em seu código. E tem abordado a codificação de uma maneira bastante procedural utilizando instruções simples, condicionais e loops for/while com funções — isso não é exatamente orientado a objetos. Na verdade, não é nem um pouco orientado a objetos! Utilizamos alguns objetos aqui e ali sem realmente saber sobre eles, mas você não escreveu nenhum dos seus objetos ainda. Bem, está na hora de deixar essa cidade procedural chata para trás e criar alguns objetos próprios. Neste capítulo, você vai descobrir porque utilizar objetos vai tornar a sua vida muito melhor — bem, pelo menos no sentido da programação (não podemos realmente lhe ajudar com seu senso de estilo e sua programação JavaScript em um só livro). Só um aviso: uma vez que descobrir objetos você nunca mais vai querer voltar. Nos envie um cartão-postal quando chegar lá.

Alguém disse "Objetos"?!	174
Pensando sobre propriedades	175
Como criar um objeto	177
O que é Orientação a Objetos?	180
Como as propriedades funcionam	181
Como uma variável contém um objeto?	
Mentes inquisitivas querem saber	186
Comparando primitivos e objetos	187
Fazendo ainda mais com objetos	188
O passo a passo da pré-qualificação	190
Vamos falar um pouco mais sobre passar	
objetos para funções	192
Comporte-se! Ou, como adicionar comportamento	
aos seus objetos	198
Melhorando o método drive	199
Por que o método drive não sabe sobre a	
propriedade started?	202
Como this funciona	204
Como o comportamento afeta o estado	
Adicionando um pouco de Gas-o-line	210
Agora vamos afetar o comportamento com o estado	211
Parabéns pelos seus primeiros objetos!	213
Adivinhe? Há objetos em todos lugares!	
(e eles vão facilitar muito sua vida)	214



230

interagindo com sua página web

O desafio "desvende o código"

6

Conhecendo o DOM

Você já caminhou bastante com o JavaScript. Na

verdade, você evoluiu de um novato para um criador de scripts para, bem, um **programador**. Mas há algo faltando. Para realmente começar a alavancar suas habilidades como programador JavaScript, você vai ter que saber como interagir com a página web onde seu código mora. Somente ao saber isso é que você vai ser capaz de escrever páginas que sejam **dinâmicas**, que reajam, que respondam, que atualizemse após serem carregadas. Então, como você interage com a página? Utilizando o **DOM**, também conhecido como **document object model**. Neste capítulo, vamos analisar o DOM e ver como podemos utilizá-lo, junto com o JavaScript, para ensinar alguns truques novos à sua página.

231 Então o que o código faz? Como o JavaScript realmente interage com sua página 233 Como fazer seu próprio DOM 234 Um primeiro gostinho do DOM 235 240 Obtendo um elemento com getElementById O que, exatamente, estou obtendo do DOM? 241 242 Encontrando seu HTML interno O que acontece quando você altera o DOM 244 247 Um test drive ao redor dos planetas Nem pense em rodar meu código até que a página 249 esteja totalmente carregada! 250 Você diz "event handler", eu digo "callback" Como definir um atributo com setAttribute 255 Mais diversão com atributos! (você pode PEGAR os atributos também) 256 Não se esqueça, getElementByld pode retornar null também! 256

head body

p id ="greenplanet" p id ="redplanet" p id ="blueplanet"

All is Nothing to report A-OK

document

ΧV

258

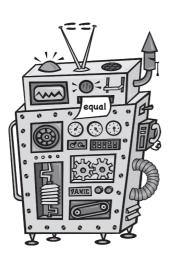
E para o que mais o DOM é bom?

tipos, igualdade, conversão e tudo mais

7

Tipos sérios

É hora de ser sério com relação a tipos. Uma das grandes coisas sobre o JavaScript é que você pode chegar longe sem saber um monte de detalhes da linguagem. Mas, para realmente tornar-se um mestre na linguagem, obter aquela promoção e seguir com as coisas que você realmente quer fazer na vida, você tem que mandar bem nos tipos. Lembra-se do que dissemos lá atrás sobre o JavaScript? Que ele não tinha o luxo de um berço de ouro, uma linguagem acadêmica e revisada por pares? Bem, isso é verdade, mas a vida acadêmica não parou o Steve Jobs e o Bill Gates, e não parou o JavaScript também. Isso não significa que o JavaScript não tenha o... bem, o mais pensado sistema de tipos, e vamos encontrar algumas idiossincrasias ao longo do caminho. Mas não se preocupe, neste capítulo vamos mostrar tudo isso e em breve você será capaz de evitar todos aqueles momentos embaraçosos com os tipos.



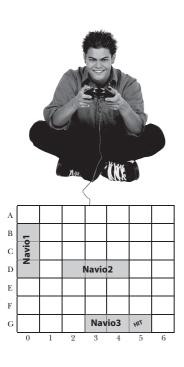
A verdade está lá fora	266
Cuidado, você pode tropeçar no undefined	
quando menos esperar	268
Como usar null	271
Lidando com NaN	273
E fica ainda mais estranho	273
Temos que fazer uma confissão	275
Entendendo o operador de igualdade (também	
conhecido como ==)	276
Como a igualdade converte seus operadores (soa mais	
perigoso do que realmente é)	277
Como ser rigoroso	
com a igualdade	280
Ainda mais conversões de tipos	286
Como determinar se dois objetos são iguais	289
A "verdade" está lá fora	291
O que o JavaScript considera falsey	292
A Vida Secreta das Strings	294
Como uma string pode se parecer com um	
primitivo e um objeto	295
Um tour de cinco minutos pelos métodos de	
string (e pelas propriedades)	297
Guerra das Cadeiras	301

juntando tudo

8

Construindo um app

Coloque seu cinto de ferramentas. Isto é, o cinto com todas as suas habilidades de programação, seu conhecimento do DOM e até mesmo algum HTML e CSS. Vamos juntar tudo neste capítulo para criar nossa primeira aplicação web de verdade. Sem mais jogos de brinquedos bobos com um navio e uma única linha de locais para se esconder. Neste capítulo, vamos construir a experiência inteira: um grande e bonito tabuleiro de jogo, múltiplos navios e entradas do usuário direto na página web. Vamos criar a estrutura da página para o jogo com HTML, estilizar o jogo visualmente com CSS e escrever em JavaScript para codificar o comportamento do jogo. Prepare-se: este é um capítulo de desenvolvimento acelerado, em que vamos mostrar bastante código de verdade.



Desta vez, vamos construir um VERDADEIRO jogo	
de Batalha Naval	318
Dando um passo para trás para o HTML e o CSS	319
Criando a página HTML: o Quadro Geral	320
Adicionando um pouco mais de estilo	324
Usando as classes hit e miss	327
Como projetar o jogo	329
Implementando o View	331
Como o displayMessage funciona	331
Como funcionam displayHit e displayMiss	333
O Modelo	336
Como vamos representar os navios	338
Implementando o objeto model	341
Preparando o método fire	342
Implementando o Controller	349
Processando o palpite do jogador	350
Planejando o código	351
Implementando o parseGuess	352
Contando palpites e disparando	355
Como adicionar um event handler ao botão Fire!	359
Passando a entrada para o controller	360
Como colocar os navios	364
Escrevendo o método generateShip	365
Gerando a posição inicial para o novo navio	366
Completando o método generateShip	367

xvii

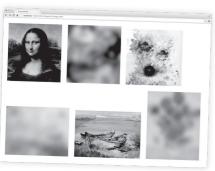
codificação assíncrona

9

Manipulando eventos

Depois deste capítulo você vai perceber que não está mais no Kansas. Até agora você vinha escrevendo código que tipicamente executava de cima para baixo — claro, seu código pode ser um pouco mais complexo que isso e fazer uso de algumas funções, objetos e métodos, mas em algum momento o código simplesmente segue seu curso. Bem, lamentamos muito falar isto para você tão tarde nesse livro, mas não é assim que, tipicamente, se escreve código JavaScript. Em vez disso, a maioria do JavaScript é escrito para reagir a eventos. Que tipos de eventos? Bem, que tal um usuário clicando na sua página, dados chegando da rede, timers expirando no navegador, alterações ocorrendo no DOM e isso são apenas alguns exemplos. Na verdade, todos os tipos de eventos estão ocorrendo o tempo todo, nos bastidores do seu navegador. Neste capítulo, vamos repensar nossa abordagem com relação à codificação JavaScript e aprender como e porque devemos escrever código que reaja a eventos.





O que sao eventos:	383
O que é um event handler?	384
Como criar seu primeiro event handler	385
Faça o test drive do seu evento	386
Entendendo eventos criando um jogo	388
Implementando o jogo	389
Test drive	390
Vamos adicionar mais algumas imagens	394
Agora precisamos atribuir o mesmo event handler	
para a propriedade onclick de cada imagem	395
Como reutilizar o mesmo handler para	
todas as imagens	396
Como o objeto event funciona	399
Colocando o objeto event para trabalhar	401
Faça o test drive do objeto event e do target	402
Eventos e filas	404
Ainda mais eventos	407
Como funciona o setTimeout	408
Finalizando o jogo das imagens	412
Faça o test drive do timer	413

xviii

clas Fui

funções de primeira classe



Funções liberadas

Conheça funções, então mande ver. Cada arte, artesanato ou disciplina tem um princípio chave que separa os praticantes medianos das estrelas do rock virtuosas — e, em se tratando do JavaScript, o real entendimento de funções é o que faz a diferença. Funções são fundamentais para o JavaScript e muitas técnicas que utilizamos para projetar e organizar o código dependem do conhecimento e do uso avançado das funções. O caminho para o aprendizado de funções nesse nível é interessante, então prepare-se... Este capítulo será um pouco como o Willy Wonka fazendo um tour pela fábrica de chocolate — você encontrará algumas coisas selvagens, loucas e maravilhosas a medida que aprender mais sobre as funções em JavaScript.

A misteriosa vida dupla da palavra-chave function	430
Declarações de funções versus expressões de funções	431
Analisando a declaração de função	432
O que vem a seguir? O navegador executa o código	433
Seguindo em frente O condicional	434
Como funções são valores também	439
Já mencionamos que funções têm status de	
Primeira Classe no JavaScript?	442
Voando de Primeira Classe	443
Escrevendo código para processar e fazer a	
verificação dos passageiros	444
Iterando os passageiros	446
Passando uma função para uma função	447
Teste de voo	447
Retornando funções de funções	450
Escrevendo o código do pedido de bebida do	
comissário de bordo	451
O código do pedido de bebidas do comissário de	
bordo: uma abordagem diferente	452
Recebendo pedidos com funções de primeira classe	454
Como o método sort do array funciona	459
Colocando tudo junto	460
Levando a ordenação para um test drive	462



xix

funções anônimas, escopo e closures

11

Funções sérias

Você deu uma boa olhada nas funções, mas ainda

há mais a aprender. Neste capítulo vamos além; seremos hardcore. Vamos mostrar como realmente manipular as funções. Este não será um capítulo super longo, mas será intenso e, no fim, você será mais expressivo com seu JavaScript do que pensava ser possível. Você também estará preparado para trabalhar com o código de um companheiro de trabalho, ou mergulhar em uma biblioteca JavaScript de código aberto, porque vamos falar sobre algumas expressões idiomáticas e convenções das funções. E se você nunca ouviu falar de uma função anônima ou de um closure, cara, você está no lugar certo.

Dando uma olhada no outro lado das funções	476
Como usar uma função anônima	477
Precisamos falar sobre sua verbosidade, de novo	479
Quando uma função é definida? Depende	483
O que aconteceu? Por que fly não foi definida?	484
Como aninhar funções	485
Como o aninhamento afeta o escopo	486
Uma neguena revisão sobre escono lévico	188



revisão sobre escopo léxico	48
Quando as coisas ficam interessantes	
com o escopo léxico	489
Funções revisitadas	49
Chamando uma função (revisitada)	49
O que diabos é um closure?	49.
Fechando uma função	49
Usando closures para implementar um	
contador mágico	498
Test drive do seu contador mágico	499
Olhando os bastidores	499
Criando um closure ao passar uma	
expressão de função como argumento	50
O closure contém o ambiente	
verdadeiro, não uma cópia	503
Criando um closure com um	
event handler	50
Como o closure Click me! funciona	50

XX

522

construção avançada de objetos

12

Criando objetos

Até agora você vinha criando objetos à mão.

Criando objetos com objetos literais

Para cada objeto, usamos um **objeto literal** para especificar toda e qualquer propriedade. Tudo certo em uma pequena escala, mas para código sério precisamos de algo melhor. É aí que entram os **construtores de objetos**. Com os construtores, podemos criar objetos muito mais facilmente e podemos criar objetos que sigam a mesma **planta de projeto** — significando que podemos usar os construtores para garantir que cada objeto tenha as mesmas propriedades e inclua os mesmos métodos. E com construtores podemos escrever código de objetos que seja muito mais **conciso** e muito menos sujeito a erros quando estivermos criando muitos objetos. Então, vamos começar e depois desse capítulo você falará sobre construtores como se tivesse crescido em Objectville.

3	
Usando convenções para objetos	523
Introduzindo construtores de objetos	525
Como criar um Construtor	526
Como usar um Construtor	52'
Como os construtores funcionam	528
Você pode colocar métodos em um	
construtor também	530
É Hora Da Produção!	536
Vamos fazer o test drive de alguns carros novos	538
Não descarte ainda os objetos literais	539
Refazendo os argumentos como um objeto literal	540
Retrabalhando o construtor Car	54
Entendendo Instâncias de Objetos	543
Mesmo objetos construídos podem ter	
suas propriedades independentes	546
Construtores do Mundo Real	548
O objeto Array	549
Ainda mais diversão com objetos internos	55
•	



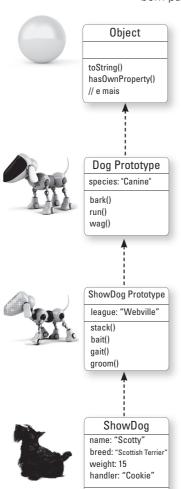
xxi

usando protótipos

Objetos extrafortes

Aprender como criar objetos foi apenas o começo.

É hora de colocar um pouco de músculo em seus objetos. Precisamos de mais maneiras de criar **relacionamentos** entre objetos existentes e de **compartilhar código** entre eles. Também precisamos de maneiras de estender e aprimorar objetos existentes. Em outras palavras, precisamos de mais ferramentas. Neste Capítulo, você verá que o JavaScript tem um **modelo de objetos** muito poderoso, mas que é um tanto quanto diferente da linguagem orientada a objetos. Ao invés do típico sistema orientado a objetos baseado em classes, o JavaScript opta por um modelo de **protótipo** mais poderoso, em que os objetos possam herdar e estender o comportamento de outros objetos. Isso é bom para quê? Você verá em breve. Vamos começar...



565
566
568
569
570
571
573
576
582
584
585
589
591
592
594
598
602
604
607
608
610
612
612
613
613

xxii

14

Apêndice: Sobras

Os Top Dez Tópicos (que não cobrimos)

Já cobrimos muita coisa, e você quase terminou este

livro. Vamos sentir sua falta, mas antes de deixarmos você ir, não nos sentiríamos bem em mandá-lo para o mundo antes de um pouco mais de preparação. Não é possível colocar tudo o que você vai precisar saber neste capítulo relativamente pequeno. Na verdade, originalmente tínhamos incluído



tudo o que você precisava saber sobre Programação JavaScript (e que não foi coberto nos outros capítulos) reduzindo o tamanho da letra para 0,00004. Coube tudo, mas ninguém conseguia ler. Então jogamos tudo fora e mantivemos as melhores partes para esse apêndice Top Dez. Esse é realmente o final do livro. Exceto pelo índice, claro (que é uma leitura imperdível!)

Nº1 jQuery	624
Nº2 Fazendo mais com o DOM	626
Nº3 O Objeto Window	627
Nº4 Argumentos	628
Nº5 Manipulando exceções	629
Nº6 Adicionando manipuladores de eventos com	
addEventListener	630
Nº7 Expressões Regulares	632
Nº8 Recursão	634
Nº9 JSON	636
Nº10 JavaScript no lado servidor	637



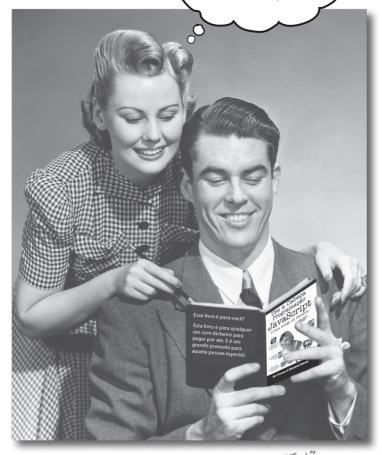
Índice 639

xxiii

como usar este livro

Introdução

Não posso acreditar que eles colocaram **isso** em um livro de JavaScript!

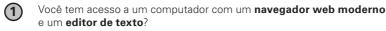


Nesta seção, respondemos àquela questão urgente: "Então, POR QUE eles colocaram isso em um livro de JavaScript?"

xxv

Para quem é este livro?

Se você puder responder "sim" a todas:



Você quer aprender, entender e se lembrar como programar em JavaScript usando as melhores técnicas e os padrões mais recentes?

Você prefere conversas informais estimulantes às palestras acadêmicas secas e tolas?

este livro é para você.

Consideramos uma versão atualizada do Safari, Chrome, Firefox ou IE versão 9 ou mais recente como sendo modernos.

ENota do marketing: este livro é para qualquer um com um cartão de crédito. J

Quem provavelmente deveria evitar esse livro?

Se você puder responder "sim" a qualquer uma dessas:

Você é <u>completamente</u> novato em desenvolvimento web?

HTML e CSS são conceitos estranhos para você? Se forem, você provavelmente vai preferir começar com o *Use a Cabeça HTML e CSS* para entender como fazer páginas web antes de lidar com JavaScript.

Você é um desenvolvedor web experiente procurando por um livro de referência?

Você tem **medo de tentar algo completamente diferente?** Você prefere fazer tratamento de canal a misturar listras com xadrez? Você acredita que um livro técnico não pode ser sério se os objetos JavaScript forem antropomorfizados?

este livro não é para você.



xxvi introdução

Seu cérebro acha que

ISSO é importante

Sabemos o que você está pensando.

"Como isso pode ser um livro sério?"

"Pra que todas essas imagens?"

"Dá para aprender algo assim?"

E sabemos o que seu cérebro está pensando

Seu cérebro anseia por novidades. Ele está sempre buscando, olhando, aguardando alguma coisa não usual. Ele foi feito dessa maneira e isso lhe ajuda a se manter vivo.

Hoje em dia é menos provável que você se torne almoço de tigre. Mas seu cérebro continua procurando. Nunca se sabe.

Então o que seu cérebro faz com todas aquelas coisas rotineiras, ordinárias e normais que você encontra? Tudo o que ele puder fazer para evitar que elas interfiram com o trabalho real do cérebro — gravar coisas que importem. Ele nem se importa em guardar coisas chatas; elas nunca passam do filtro "isso obviamente não é importante."

Como seu cérebro sabe o que é importante? Suponha que você está passeando e um tigre pula na sua frente. O que acontece com sua cabeça e seu corpo?

Os neurônios disparam. Emoções a flor da pele. A química inunda você.

E é assim que seu cérebro sabe...



Mas imagine que você está em casa, ou numa biblioteca. É um local seguro, aconchegante e sem tigres. Você está estudando, se preparando para uma prova ou tentando aprender algum tópico técnico difícil que seu chefe acha que vai levar uma semana, dez dias no máximo.

Só um problema. Seu cérebro está tentando lhe fazer um grande favor. Ele está tentando se certificar que esse conteúdo obviamente não importante não atrapalhe os recursos escassos. Recursos que são mais bem gastos armazenando coisas realmente importantes. Tipo tigres. Tipo os perigos do fogo. Tipo como você nunca mais

deve fazer snowboard de shorts.

E não há uma maneira simples de dizer ao seu cérebro, "Ei, cérebro, muito obrigado, mas não importa o quão chato é esse livro e o quão baixo eu estou marcando na escala Richter emocional, eu realmente preciso que você guarde isso."



você está aqui

Pensamos no leitor do "Use a Cabeça" como um aprendiz.

Então o que é necessário para se aprender algo? Primeiro, você tem que entender, depois tem que se certificar de não esquecer. Não é apenas enfiar fatos na sua cabeça. Baseado nas últimas pesquisas em ciência cognitiva, neurobiologia e psicologia educacional, o aprendizado requer mais do que texto numa página. Nós sabemos Ao contrário de outras linguagens,

o que liga seu cérebro.

Alguns dos princípios de aprendizado do Use a Cabeça:

Torne visual. Imagens são muito mais memorizáveis que palavras sozinhas e tornam o aprendizado mais efetivo (até 89% de melhoria em estudos de lembrança e transferência). Elas também tornam as coisas mais compreensíveis. Coloque as palavras dentro ou perto das suas imagens às quais elas se referem, em vez de na parte de baixo ou em outra página, e os aprendizes vão ser até duas vezes mais capazes de resolver problemas relacionados com o conteúdo.



Realmente acho que o JavaScript devia estar no elemento <head:

Use um estilo personalizado e coloquial. Em estudos recentes, os estudantes foram até 40% melhores em testes de pós-aprendizado se o conteúdo falasse diretamente ao leitor usando o estilo coloquial em primeira pessoa em vez de um tom formal. Conte histórias em vez de palestrar. Use linguagem casual. Não se leve tão a sério. Ao que você prestaria mais atenção: a uma conversa informal ou a uma palestra?

Faça o aprendiz pensar mais profundamente. Em outras palavras, a menos que você flexione ativamente seus neurônios, nada mais acontece na sua cabeça. Um leitor tem que ser motivado, engajado, curioso e inspirado a resolver problemas, tirar conclusões e gerar novos conhecimentos. E para isso você precisa de desafios, exercícios e questões que façam pensar em atividades que envolvam ambos os lados do cérebro e múltiplos sentidos.

Não tão rápido! mplicações na formance e no carregamento da página mais cuidadoso ao usar variáveis globais.



Agora que tenho sua atenção, você deveria ser

Obtenha — e mantenha — a atenção do leitor.

Todos já tivemos a experiência "eu realmente tenho que aprender isso mas não consigo ficar acordado além da página um". Seu cérebro presta atenção a coisas que são fora do ordinário, interessantes, estranhas, chamativas, inesperadas. O aprendizado de um tópico técnico novo e difícil não tem que ser chato. Seu cérebro vai aprender mais rapidamente se não for.

Toque as emoções. Nós agora sabemos que sua habilidade de se lembrar de algo é largamente dependente de seu conteúdo emocional. Você se lembra daquilo com o que se importa. Você se lembra quando sente algo. Não, nós não estamos falando sobre uma história de dar nó no coração de um garoto e seu cachorrinho. Estamos falando sobre emoções como surpresa, curiosidade, diversão, "o que é ... ?" e do sentimento "Sou demais!" que vem quando você resolve um quebra-cabeça, aprende algo que todos acham que é difícil ou percebe que sabe algo que Bob "Sou Mais Técnico Que Você" da engenharia não sabe.



xxviii introdução

Metacognição: pensando sobre pensar.

Se você realmente quer aprender e quer aprender mais rapidamente e mais profundamente, preste atenção em como você presta atenção. Pense sobre como você pensa. Aprenda como você aprende.

A maioria de nós não faz cursos de metacognição ou teoria do aprendizado quando estamos crescendo. *Esperavam* que nós aprendêssemos, mas raramente nos *ensinavam* como aprender.

Mas assumimos que se você está segurando esse livro, realmente quer aprender como criar programas em JavaScript. E você provavelmente não quer gastar muito tempo. E quer *se lembrar* do que leu e ser capaz de aplicar. E, para isso, você tem que *entender*. *P*ara tirar o máximo desse livro ou *qualquer* livro ou experiência de aprendizado, seja responsável pelo seu cérebro. Seu cérebro *neste* conteúdo.

O truque é fazer seu cérebro ver o novo material que você está aprendendo como algo Realmente Importante. Crucial para seu bem-estar. Tão importante quanto um tigre. Caso contrário, você vai se encontrar em uma batalha permanente com seu cérebro fazendo o melhor dele para evitar que o novo conteúdo se fixe.

Me pergunto como enganar meu cérebro para se lembrar desta matéria...



Então *COMO* você faz seu cérebro pensar que JavaScript é tão importante quanto um tigre?

Há a maneira lenta e tediosa ou a maneira rápida e mais efetiva. A maneira lenta é por pura repetição. Obviamente você sabe que *é capaz* de aprender e se lembrar mesmo dos tópicos mais tediosos se você continuar batendo na mesma tecla. Com repetições suficientes, seu cérebro diz "Isso não *parece* importante, mas ele continua olhando a mesma coisa *de novo* e *de novo* e *de novo*, então suponho que seja."

A maneira mais rápida é *fazer qualquer coisa para aumentar sua atividade cerebral*, especialmente diferentes *tipos* de atividade cerebral. Os tópicos na página anterior são uma grande parte da solução e elas são todas coisas que comprovadamente ajudam seu cérebro a trabalhar a seu favor. Por exemplo, estudos mostram que colocando palavras *dentro* das imagens que elas descrevem (ao contrário de em algum outro lugar da página, como numa legenda ou no corpo do texto) faz seu cérebro tentar entender como as palavras e a imagem se relacionam e isso faz com que mais neurônios disparem. Mais neurônios disparando = mais chances para seu cérebro *perceber* que isso é algo que vale a pena prestar atenção e possivelmente gravar.

Um estilo coloquial ajuda porque as pessoas tendem a prestar mais atenção quando elas percebem que estão em uma conversa, uma vez que espera-se que elas acompanhem e sustenham uma conversa. O incrível é que seu cérebro não necessariamente *se importa* se a "conversa" é entre você e um livro! Por outro lado, se o estilo de escrita for formal e seco, seu cérebro se sente da mesma maneira que você ao estar em um auditório cheio de espectadores passivos ouvindo uma palestra. Nem precisa ficar acordado.

Mas imagens e estilo coloquial são apenas o começo.

você está aqui ► xxix

03/05/2016 09:11:49

Eis o que NóS fizemos:

Usamos *imagens* porque seu cérebro é ajustado para o visual, não o textual. Na percepção do seu cérebro, uma imagem *realmente* vale por 1024 palavras. E quando texto e imagens funcionam juntas, nós embutimos o texto *nas* imagens porque seu cérebro funciona mais efetivamente quando o texto está *dentro* da coisa à qual ele se refere, ao contrário de em alguma legenda ou enterrado no texto em algum lugar.

Utilizamos *redundância*, dizendo a mesma coisa de *diferentes* maneiras e com diferentes tipos de mídia e múltiplos *sentidos*, para aumentar a chance de que o conteúdo seja codificado naquela área do cérebro.

Usamos conceitos e imagens de maneiras *inesperadas* porque seu cérebro está ajustado para novidades e usamos imagens e ideias com, pelo menos, algum *conteúdo emocional* porque seu cérebro é ajustado para prestar atenção à bioquímica das emoções. Aquilo que faz com que você *sinta* algo é mais passível de ser lembrado, mesmo que esse sentimento não seja mais do que um pouco de *humor*, *surpresa* ou *interesse*.

Usamos um *estilo conversacional* personalizado porque seu cérebro presta mais atenção quando acredita que está em uma conversa do que quando pensa que está ouvindo passivamente uma apresentação. Seu cérebro faz isso mesmo quando está *lendo*.

Incluímos mais de 100 *atividades*, porque seu cérebro está ajustado para aprender e se lembrar mais quando você *faz* coisas do que quando $l\hat{e}$ sobre coisas. E fizemos os exercícios desafiadores-mas-factíveis porque isso é o que a maioria das *pessoas* preferem.

nentos | primeiro, | primeiro, | primeiro | primeiro | primeiro | primeiro

Usamos *múltiplos estilos de aprendizagem* porque *você* pode preferir procedimentos passo a passo, enquanto alguma outra pessoa quer entender o quadro geral primeiro, enquanto outra apenas quer ver um exemplo de código. A despeito da sua preferência de aprendizado, *todos* se beneficiam ao ver o mesmo conteúdo representado de diversas maneiras.

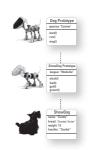
Incluímos conteúdo para *ambos os lados do seu cérebro*, porque quanto mais seu cérebro se engaja, mais provavelmente você deve aprender e se lembrar e mais consegue se manter focado. Uma vez que trabalhar um lado do cérebro frequentemente significa dar ao outro lado uma chance de descansar, você pode ser mais produtivo no aprendizado durante longos períodos de tempo.

E incluímos *histórias* e exercícios que apresentam *mais de um ponto de vista*, porque seu cérebro está ajustado para aprender mais profundamente quando é forçado a fazer avaliações e julgamentos.

Incluímos *desafios*, com exercícios e *questões* que nem sempre têm uma resposta direta; porque seu cérebro está ajustado para aprender e se lembrar quando ele tem que *trabalhar* em cima de algo. Pense nisso — não dá fazer seu *corpo* ficar em forma apenas *olhando* as pessoas na academia. Mas fizemos nosso melhor para garantir que quando você estiver trabalhando pesado seja nas coisas *certas*. Ou seja, que *você não esteja gastando nenhum neurônio extra* processando um exemplo difícil de entender ou analisando textos exageradamente concisos, difíceis ou carregados de jargões.

Usamos *pessoas* em histórias, exemplos, imagens, etc, porque, bem, *você* é uma pessoa. E seu cérebro presta mais atenção em *pessoas* do que em *coisas*.

Usamos uma abordagem *80/20*. Assumimos que, se você vai se tornar um desenvolvedor JavaScript genial, esse não vai ser seu único livro. Então não falamos sobre *tudo*. Apenas sobre as coisas que você realmente *precisa*.





Seja o Navegador





xxx introdução



Eis o que VOCÊ pode fazer para exercitar seu cérebro à sua vontade.

Então, fizemos nossa parte. O resto é com você. Estas dicas são um começo; ouça seu cérebro e descubra o que funciona e o que não funciona para você. Tente coisas novas.

Corte isto e cole na Porta da geladeira

Desacelere. Quanto mais você entender, menos vai ter que memorizar.

Não leia *apenas*. Pare e pense. Quando o livro fizer uma pergunta imagine que alguém realmente está perguntando aquilo. Quanto mais profundamente você forçar seu cérebro a pensar, melhores serão suas chances de aprender e se lembrar.

Faça os exercícios. Escreva suas próprias anotações.

Nós os colocamos aqui, mas se os fizéssemos para você, seria como ter alguém fazendo seu dever de casa. E não apenas *olhe* para os exercícios, **use uma caneta**. Há muitas evidências que atividade física *durante* o aprendizado pode aumentá-lo.

Leia o "Não Existem Questões Bobas"

Isso significa: leia todos eles. Eles não são barras laterais opcionais — *eles fazem parte do conteúdo principal!* Não os ignore.

Faça disso a última coisa que vai ler antes de dormir. Ou, pelo menos, a última coisas desafiante.

Parte do aprendizado (especialmente a transferência para a memória de longo prazo) acontece *depois* de você fechar o livro. Seu cérebro precisa de um tempo para si para fazer mais processamentos. Se você incluir alguma coisa nova antes desse tempo de processamento, algo que você acabou de aprender vai ser perdido.

🖺 Beba água. Muita água.

Seu cérebro funciona melhor em uma boa banheira de fluído. Desidratação (que pode ocorrer antes de você ficar com sede) diminui suas funções cognitivas.

Fale sobre isso. E em voz alta.

Falar ativa uma parte diferente do cérebro. Se você estiver tentando entender algo ou aumentar suas chances de se lembrar mais tarde, fale em voz alta. Melhor ainda, tente explicar isso em voz alta para outra pessoa. Você vai aprender mais rapidamente e pode descobrir novas ideias que você nem sabia que estavam ali enquanto estava lendo sobre elas.

Ouça seu cérebro.

Preste atenção se seu cérebro está ficando sobrecarregado. Se você se pegar lendo por alto ou se esquecendo do que acabou de ler, é hora para uma pausa. Depois de um determinado ponto, você não aprende mais rápido tentando enfiar tudo de uma vez e pode até se prejudicar no processo.

Sinta algo!

Seu cérebro precisa saber que isso *importa*. Envolva-se nas histórias. Faça suas próprias legendas para as imagens. Reclamar sobre uma piada ruim *ainda* é melhor do que não sentir nada.

Crie algo!

Aplique isso em algo novo que esteja projetando, ou trabalhe novamente em um projeto antigo. Faça *alguma* coisa para conseguir alguma experiência para além dos exercícios e atividades neste livro. Tudo que você precisa é uma caneta e um problema para resolver... Um problema que possa se beneficiar do uso do JavaScript.

Durma.

Liberado por: Bianca Teodoro

Você tem que criar um monte de novas conexões cerebrais para aprender a programar. Durma com frequência; isso ajuda.

você está aqui ► xxxi

Leia-me

Essa é uma experiência de aprendizado, não um livro de referência. Deliberadamente eliminamos tudo que pudesse ficar no caminho do aprendizado do que quer que estivéssemos trabalhando naquele ponto do livro. E na primeira vez que você o utiliza, precisa começar no início, porque o livro faz suposições sobre o que você já viu e aprendeu.

Nós ensinamos as partes BOAS do JavaScript e lhe avisamos sobre as partes RUINS.

O JavaScript é uma linguagem de programação que não veio das grandes universidades com tempo suficiente para revisão acadêmica. O JavaScript foi jogado no mundo por necessidade e cresceu na vizinhança dos navegadores antigos. Então esteja avisado: ele tem umas partes ótimas e algumas não tão boas assim. Mas, de modo geral, o JavaScript é brilhante, se você o utilizar de forma inteligente.

Neste livro, ensinamos a usar as partes ótimas a seu favor e vamos lhe apontar as partes ruins e mostrar como contorná-las.

Não cobrimos exaustivamente cada pequeno aspecto da linguagem.

Há muito que você pode aprender sobre o JavaScript. Este livro não é de referências; é um livro de aprendizado, então não cobre tudo que há para se saber sobre o JavaScript. Nosso objetivo é ensinar os fundamentos da utilização do JavaScript de forma que você possa pegar qualquer livro de referência antigo e fazer o que quiser.

Este livro ensina o JavaScript no navegador.

O navegador não apenas é o ambiente mais comum no qual o JavaScript roda, é também o mais conveniente (todo mundo tem um computador com um navegador e um editor de textos e é tudo o que você precisa para se iniciar no JavaScript). Rodar o JavaScript no navegador significa que você obtém gratificação instantânea: você pode escrever código e tudo que tem que fazer é recarregar sua página web para ver o que ele faz.

Esse livro advoga códigos bem estruturados e legíveis baseados nas melhores práticas.

Você quer escrever códigos que você e outras pessoas possam ler e entender, códigos que vão funcionar nos navegadores no ano que vem. Você quer escrever códigos da maneira mais direta de forma que possa realizar o trabalho e seguir em frente. Neste livro, vamos ensinar a escrever código claro, bem organizado e que antecipe mudanças desde o início. Um código do qual você possa se orgulhar, um código que você vai querer emoldurar e pendurar na parede (mas tire da parede antes de levar seu namorado ou namorada para sua casa).

Encorajamos você a usar mais do que um navegador neste livro.

Ensinamos você a escrever JavaScript que é baseado em padrões, mas possivelmente você ainda encontrará pequenas diferenças na maneira que os navegadores web interpretam o JavaScript. Mesmo fazendo nosso melhor para garantir que todo o código no livro funcione em todos os navegadores modernos, e mesmo mostrando alguns truques para garantir que seu código seja

xxxii introdução

suportado por esses navegadores, encorajamos você a pegar pelo menos dois navegadores para testar seu JavaScript. Isso vai lhe dar a experiência em ver as diferenças entre eles e em criar código que funcione bem em uma variedade de navegadores com resultados consistentes.

Programação é um negócio sério. Você vai ter que trabalhar, e algumas vezes trabalhar duro.

Se você já tem alguma experiência em programação, então sabe sobre o que estamos falando. Se você estiver vindo direto do *Use a Cabeça HTML e CSS*, então você vai descobrir que escrever código é um pouco diferente. Programação requer uma maneira diferente de pensar. Programação é lógica, às vezes muito abstrata, e requer que você pense de uma maneira algorítmica. Mas não se preocupe, vamos fazer tudo isso de uma maneira amigável ao seu cérebro. Apenas pegue um pouco de tempo, certifique-se de estar bem alimentado e com o sono em dia. Dessa forma, estes novos conceitos sobre programação vão realmente ser entendidos.

As atividades NÃO são opcionais.

Os exercícios e as atividades neste livro $n\bar{a}o$ são add-ons; elas são parte do conteúdo central do livro. Algumas delas são para ajudar com a memória, algumas para o entendimento e algumas vão ajudar a aplicar o que você aprendeu. Não pule os exercícios. As palavras-cruzadas são as únicas coisas que você não tem que fazer, mas elas são boas para dar ao seu cérebro uma chance de pensar sobre as palavras em contextos diferentes.

A redundância é intencional e importante.

Uma diferença distinta de um livro Use a Cabeça é que realmente queremos que você entenda. E queremos que você termine o livro se lembrando do que aprendeu. A maioria dos livros de referência não têm retenção e lembrança como objetivos, mas esse livro é para o aprendizado, então você vai ver alguns dos conceitos aparecerem de novo e de novo.

Os exemplos são os mais simples possíveis.

Nossos leitores nos dizem que é frustrante passar por 200 linhas de um exemplo procurando pelas duas linhas que eles precisaram entender. A maioria dos exemplos neste livro são mostrados dentro do menor contexto possível, de forma que a parte que você está tentando aprender seja clara e simples. Não espere que todos os exemplos sejam robustos ou mesmo completos — eles foram escritos especificamente para o aprendizado e nem sempre são plenamente funcionais.

Colocamos todos os arquivos com os exemplos no site da editora para que você faça o download. Você vai encontrá-los em www.altabooks.com.br (procure pelo título do livro).

Os exercícios "Poder do Cérebro" geralmente não têm respostas.

Para alguns deles não há uma resposta correta e, para outros, parte da experiência de aprendizado das atividades Poder do Cérebro é que você decida se e quando suas respostas estão corretas. Em alguns exercícios você vai encontrar dicas apontando na direção certa.

você está aqui ► xxxiii