

*Para sua conveniência Aprima colocou algumas das frente questão material depois do índice. Por favor use os marcadores e o conteúdo num ápice links para acessá-los.*



**Visão geral do conteúdo**

Sobre o Autor [xiii](#page178)

Introdução [xv](#page4)

■ Capítulo 1: Fundamentos de CSS3 [1](#page6)

■ Capítulo 2: CSS3 transforma e Transições [9](#page14)

■ Capítulo 3: transições CSS3 para imagens [31](#page35)

■ Capítulo 4: CSS3 transições para elementos de UI [57](#page61)

■ Capítulo 5: CSS3 Keyframe animações [75](#page79)

■ Capítulo 6: CSS3 Animações Keyframe para conteúdo da Web [85](#page88)

■ Capítulo 7: Integrando CSS3 animações com SVG e filtros [103](#page106)

■ Capítulo 8: Integração CSS3 animação com resposta Web

Design e JavaScript [117](#page119)

■ Capítulo 9: CSS3 3D transforma, transições e animações [135](#page136)

■ Capítulo 10: Ferramentas, tecnologias e o futuro da animação CSS [155](#page156)

Índice [167](#page167)

**Introdução**

Bem-vindo ao Pro CSS3 Animação. Seu livro ensina como usar todo o poder do CSS para trazer seu conteúdo da web para a vida com interatividade e uma nova abordagem visual. Nos capítulos que se seguem, você aprenderá a usar os padrões da indústria de ponta para aumentar o apelo visual, a acessibilidade e a popularidade do seu site.

Que o seu livro é para

Seu livro é projetado para designers e codificadores com pelo menos alguns anos de experiência em desenvolvimento web que desejam atualizar rapidamente as suas competências para o novo W3C normas, ou que o desejo de tomar as suas explorações de CSS transforma, transições e animações em negrito novas direções. Ele *não é* um texto introdutório web design: o livro assume pelo menos uma compreensão básica de HTML, CSS e JavaScript. Como o desenvolvimento da web é um processo multidisciplinar, vou abordar também questões como a acessibilidade e a semântica, conceitos que o leitor deve estar familiarizado com a.

Como seu livro está estruturado

Eu dividir o livro em dez capítulos. Ele irst capítulos introduzir as componentes fundamentais do CSS animação, enquanto capítulos posteriores integrar animação com outras tecnologias da web.

**O capítulo 1**apresenta o CSS3, detalhando sua sintaxe e desenvolvimento e em contraste com as tecnologias anteriores.

**O capítulo 2**abrange as CSS3 transforma e transições.

**Capítulo 3**mostra como usar transições CSS3 com imagens, incluindo os efeitos da galeria.

**O capítulo 4**integra as transições com o site a interação do usuário elementos como botões e menus. O**Capítulo 5**apresenta o módulo de animação CSS.

**Capítulo 6**utiliza CSS3 animações sobre todos os tipos de conteúdo da web.

**Capítulo 7**mostra como integrar CSS3 transições e animações com gráficos vetoriais escaláveis (SVG) e filtros CSS.

**Capítulo 8**traz resposta web design e JavaScript juntamente com CSS animações.**Capítulo 9**toma transforma, transições e animações em terceira dimensão.

**Capítulo 10**olha para o futuro da web standards para efeitos visuais e as várias ferramentas que podem ser usadas para otimizar o CSS animação web hoje.

Baixar o código

Ele código para os exemplos mostrados neste livro está disponível no web site, [www.apress.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.apress.com). Aprima Um link pode ser encontrado na página de informações do livro sob o código fonte/guia Downloads. Seu guia está localizado sob o títulos relacionados seção da página.

Entrar em contato com o autor

Se você tiver quaisquer perguntas ou comentários ou mesmo local um erro você acha que eu deveria saber sobre-Por favor sinta-se à vontade para entrar em contato comigo através de e-mail ([dudley.storey@gmail.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\dudley.storey@gmail.com)) ou Twitter (@dudleystorey). Saúdo os vossos pensamentos e feedback.

**Capítulo 1**

**Fundamentos CSS3**

Por quase duas décadas a Folhas de Estilo em Cascata (CSS) norma tem sido utilizada para controlar a apresentação das páginas da web. HTML define o que algo *é:* uma rubrica, um parágrafo, um endereço, uma imagem, etc CSS descreve como elemento que é *apresentado* ao usuário, incluindo tais qualidades como sua cor, borda e dimensões. Inclui controles de apresentação CSS que alguns Web designers mesmo considerar, tais como a maneira de texto-para-voz serviços pronunciar o conteúdo da página da web.

Todas as regras de apresentação do CSS original foram projetados para  conteúdo estático; isto é, elementos HTML que não mudam com o passar do tempo. Até recentemente, se você queria uma imagem para fade in em uma página da web, existiam poucos tecnologias da web que você pode usar a mais popular das quais foram JavaScript e Flash. nes tas tecnologias não são soluções completas, no entanto, eles têm várias desvantagens graves, como eu vou discutir no final do capítulo.

Agora temos o CSS3 transforma, transições e módulos de animação. stas são extensões de sintaxe de CSS que são suportados em todos os browsers modernos, sobreposição, e em alguns casos, a substituição dos papéis tradicionais de JavaScript e Flash. Enquanto CSS3 não está sem suas limitações, a tecnologia é o caminho a seguir para um lote de conteúdo web dinâmico.

Para entender como é que chegamos aqui, você precisa saber onde estivemos. seu capítulo introdutório irá fornecer uma visão geral do processo de desenvolvimento de CSS e desenvolvimento web onde se encontra agora, olhando para o futuro.

Desenvolvimento de CSS

Ele independente de evolução das tecnologias da web teve algo de uma história irregular: navegador fornecedores têm por vezes a tecnologia automotriz frente, enquanto outras implementações tecnológicas têm complicado o desenvolvimento web através da adopção de abordagens incompatíveis.

Ele World Wide Web Consortium (W3C) foi formada em uma efort para tentar sintetizar e padronizar tecnologias da web em uma série de especificações que foram amplamente apoiadas pelo setor de desenvolvimento web. Ele W3C pode ser chamado de as Nações Unidas de desenvolvimento web: como um organismo de normalização independentes, ele pode avaliar diferentes propostas; criar fóruns de discussões entre a indústria e a academia, desenvolvedores e outros interesses; negociar e resolver desentendimentos; e um martelo para fora as especificações finais que todos podem concordar em seguir.

Ele padrão CSS foi desenvolvido pelo Grupo de Trabalho (CSSWG CSS), um subgrupo no âmbito do W3C. Ao longo do tempo, o CSSWG extended CSS para fornecer maior controle sobre mais aspectos do conteúdo da página da web. Como CSS 2.1 abordou o seu estatuto final, terminou, a continuação do desenvolvimento da especificação foi dividido em vários módulos. Muitos desses módulos começou como "nível 3" propostas, levando os desenvolvedores a utilizar o termo catch-all *CSS3* para qualquer coisa que se seguiram Sheets CSS2.1. Tecnicamente, as tecnologias da web de que me ocupo neste livro de animações e transições, transforma-são completamente novas especificações de nível 1, como eles não têm precedentes na CSS1 ou CSS2. Fora de muito debate formal, o setor de desenvolvimento web se refere a eles coletivamente como *CSS3*, e vou continuar a fazê lo neste livro.

Ao mesmo tempo, navegador desenvolvedores continuou a inovar. Muitas das propriedades CSS vou discutir neste livro foram propostos pela Apple, Google e Mozilla, não o W3C ou o grupo de trabalho do CSS. Seu levou a um

Problema: Os desenvolvedores queriam que seus navegadores para apoiar estas tecnologias incríveis *hoje,* sem ter de esperar que o longo processo de Recomendação da W3C, discussão e aprovação fina.

Todos sabiam o amargo lições do navegador guerras da década e o confronto de tecnologias associadas conflitantes. Como poderia navegadores oferecem suporte a muito mais recentes tecnologias propostas pelas suas empresas enquanto torna claro que estas novas propriedades foram experimental e sem entrar em conflito com as declarações oficiais do W3C que podem surgir mais tarde?

Solução ele revelado viáveis, mas controverso: fornecedor de CSS prefixos.

Prefixos de fornecedor de CSS

Para permitir que o CSS3 inovação por navegador web programadores, a comunidade de desenvolvimento concordaram que cada navegador teria o seu próprio prefixo exclusivo para a proposta ou experimentais propriedades CSS (consulte a Tabela 1-1).

**Tabela 1-1.***Prefixos de navegador exclusivo*

**Prefixo** **Navegador**

-moz- Firefox

-o- Opera

-webkit- Safari/Chrome/Konqueror

-ms- Internet Explorer 9+

■ **Observe**o fornecedor prefixos mostrados aqui não são os únicos na existência, apenas aquelas que você precisará para a maioria dos efeitos. Uma lista completa de prefixos do fornecedor pode ser encontrada em <http://alrra.github.com/little-helpers/vendor-prefixes/>).

Cada navegador destinados a apoiar uma propriedade CSS experimental pode fazer isso colocando o seu próprio prefixo de fornecedor na frente dele. Observe que essas propriedades são *despadronizados* até alcançarem a aprovação final pelo W3C. Até esse momento, eles estão abertos a modificação e interpretação tanto pelos fornecedores e o próprio W3C. Tanto o nome da propriedade e o seu valor é especificado pode alterar rapidamente, mesmo no mesmo navegador, como diferentes abordagens são consideradas normas e funcionou. Por exemplo, até o lançamento do Safari 5.1/iOS 5.0, a equipe de desenvolvimento Webkit propôs o seguinte texto como a maneira de o fazer em CSS gradientes lineares:

Corpo { background-image: -webkit-gradiente( linear,

No canto inferior esquerdo, canto superior esquerdo,

Cor-stop(0,11, rgb(167,9,246)), cor-stop(0,56, rgb(194,242,242) );

Outros navegadores implementado gradientes pressóricos em caminhos diferentes. Por exemplo, aqui como ele foi feito no Firefox:

Corpo {

Imagem de fundo: -moz-linear-gradiente(parte inferior, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

}

Estas duas abordagens produzido o mesmo resultado em cada navegador; sob alegação foi que era a melhor forma de código. No caso dos gradientes, o W3C teve um terceiro modo, mais estreitamente relacionados com o método do Firefox:

Capítulo 1 ■ FUNDAMENTOS CSS3

Corpo {

Imagem de fundo: linear-gradiente(para baixo, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

}

No entanto*,* porque os navegadores não podem ser forçados a atualizar retroactivamente, é ainda necessário incluir o fornecedor anterior-métodos prefixados para habilitar o suporte a versões mais antigas. No caso dos gradientes, isso inclui ambos os métodos para navegadores baseados em Webkit, comutado para apoiar o método padrão-agora mas manteve o prefixo do fornecedor durante um período de tempo.

Convenção dita que o W3C (método final, padrão esperado) apaga a última na declaração e que o fornecedor de versões prefixadas preceder. Ele toda a declaração de todos os navegadores seria como segue:

Corpo {

Imagem de fundo: -o-linear-gradiente(parte inferior, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

Imagem de fundo: -moz-linear-gradiente(parte inferior, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

Imagem de fundo: -webkit linear-gradiente(parte inferior, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

Imagem de fundo: -ms-linear-gradiente(parte inferior, rgb(167,9,246) 11%, rgb(194,242,242) 56 %); imagem de fundo: -webkit-gradiente(linear, canto inferior esquerdo, canto superior esquerdo,

Cor-stop(0,11, rgb(167,9,246)), cor-stop(0,56, rgb(194,242,242) );

Imagem de fundo: linear-gradiente(para cima, rgb(167,9,246) 11%, rgb(194,242,242) 56%);

}

Como navegadores pagar apenas a atenção para o CSS eles compreender e ignorar as CSS não, Safari e o Chrome irá ler a -webkit linha da declaração *apropriado para essa versão do navegador* e implementá-lo. Versões de navegador mais tarde que entendem a versão final da especificação irá ler a última linha em vez.

É inteiramente possível para navegadores para suportar tanto o prefixo e unprefixed CSS propriedades ao mesmo tempo. Regras de aparição em uma declaração são lidos da esquerda para a direita e de cima para baixo. No caso de um conflito, regras especificadas *mais tarde* têm precedência sobre os escritos *anteriores.* Colocando o padrão W3C na última declaração garante que ele sempre terá precedência se o navegador suporta.

Enquanto este código pode aparecer um pouco assustadora, é imediatamente evidente que existe uma grande quantidade de repetição dentro dela. Com a excepção do método desaprovada Webkit, a maioria dos CSS declaração poderia ser facilmente criados copiando e colando a primeira linha e fornecedor de prefixos a antecedendo cópias. Aqui também são ferramentas e técnicas para a geração automática de código de fornecedor prefixados, que vou discutir no capítulo 10.

A fim de obter suporte para CSS experimental de propriedades em um navegador específico, você *deve* incluir o prefixo do fornecedor apropriado e valor em seu stylesheet. Aqui estão apenas duas excepções:

 Ele browser permite aliases de prefixo (discutido na próxima seção).

 Navegador ele segue o padrão W3C final e não exigem um prefixo.

Hankfully, propriedades e valores para o CSS transforma, transições e animações foram amplamente acordado desde o início dos módulos; que este artigo foi escrito, cada navegador atual, implementa o código da mesma maneira, embora com o fornecedor do prefixo.

■ **Nota**em 6 de Junho de 2012 o W3C finalizado a especificação para as transições, animações e transforma e concordou em deixar que todos os fornecedores de navegador apoiar sem prefixos de fornecedor. Internet Explorer 10 é o primeiro navegador a fazê-lo, com outros navegadores deverá seguir o exemplo em breve. Versões mais antigas de navegadores ainda precisarão de prefixos de fornecedor.

Fornecedor do prefixo de questões

Enquanto o fornecedor do prefixo de sistema funciona, tem várias questões. Excepções e casos de borda pode ser difícil de controlar e lembrar. Por exemplo, actualmente a melhor solução para implementar a hifenização para pontos em todos os navegadores é como segue:

P { -ms-word-break: quebra-todos; palavra-break: quebra-todos; palavra-break: quebra-palavra; -moz-hífens: auto; -webkit-hífens: auto; hífens: auto; hifenizar: auto; }

Como você pode ver, alguns do número anterior declaração CSS usa prefixos de fornecedor, mas os nomes de propriedade e valores não correspondem ao W3C proposta no final e diferentes navegadores utilizam outras propriedades.

Além disso, alguns fornecedores de navegador têm tendência a manter seus prefixos de proprietários e não lhes preterido depois foram acordadas normas para desenvolvedores, exigindo para manter o Legacy o prefixo do código CSS.

Por último, lazy os desenvolvedores têm tendência para aplicar apenas um ou dois prefixos de fornecedor, ignorando outros navegadores que ofer igual apoio sob a sua própria versão da spec. Por exemplo, muitos desenvolvedores incluirá -moz e -webbkit propriedades prefixados em suas folhas, mas se esquecer de adicionar -ms ou -o. Por este motivo alguns navegadores mais notavelmente, versões recentes do Opera - têm a capacidade de reconhecer outro fornecedor prefixos.

No caso de ópera, isso significa que alguns -webkit prefixado propriedades.

■ **Nota**porque uma completa declaração CSS que inclui cada fornecedor prefixo pode ser muito longo, exemplos de código neste livro muitas vezes usam apenas a especificação final esperado. Na maioria dos casos você não deve limitar o prefixo propriedades para apenas os exemplos que você vê aqui se você quiser obter compatibilidade completa em todos os navegadores.

CSS3 Suporte do navegador

CSS3 transforma são totalmente suportados pelas seguintes versões de navegador, com o fornecedor do prefixo:

 O Internet Explorer 9 (IE10 não exigem prefixos)

 Firefox 3.5 e acima

 Chrome 4 e superior

 Safari 3.1 e acima

 Opera 10.5 e acima

 IOS Safari 3.2 e acima

 Opera Mobile 11

 Android Market 2.1 e acima

4

Capítulo 1 ■ FUNDAMENTOS CSS3

CSS3 As transições são totalmente suportados pelas seguintes versões de navegador, com o fornecedor do prefixo:

 O Firefox 4 e superior

 Chrome 4 e superior

 Safari 3.1 e acima

 Opera 10.5 e acima

 IOS Safari 3.2 e acima

 Opera Mobile 10

 Android Market 2.1 e acima

Internet Explorer 10 suporta transições sem prefixação; você pode esperar muito as versões recentes de outros navegadores a fazer o mesmo.

Ele CSS3 Animação Rascunho de trabalho é totalmente suportado pelos seguintes versões de navegador, com o fornecedor do prefixo:

 O Firefox 5 e acima

 Chrome 4 e superior

 Safari 4 e superior

 Opera 12 e acima

 IOS Safari 3.2 e acima

 Android Market 4.0 e acima (suporte parcial a partir de 2.1)

Novamente, o Internet Explorer 10 suporta CSS animações sem prefixação.

■ O **TIP**[www.caniuse.com](http://www.caniuse.com)é um excelente recurso para manter o controle de navegador suporte para CSS3.

Limitações do CSS3 Animação

Enquanto CSS3 transforma, transições e animações são muito poderosos, existem algumas propriedades que eles não podem afect, pelo menos não atualmente:

 CSS3 não pode controlar barras de rolagem ou "rolar" todo o corpo do documento

 Gradientes não pode ser animado (embora este seja possível alcançar com SVG ou JavaScript).

Princípios de projeto: melhoramento progressivo

E degradação harmoniosa

Uma das metodologias de concepção comum a ambas as CSS e JavaScript é *uma degradação*, também conhecido como *melhoramento progressivo*. Coloque simplesmente a idéia é usar CSS para*melhorar*um site, não para o local a ser totalmente dependentes dele.

Seu se torna especialmente importante quando você começar a animar o conteúdo. Porque CSS3 é suportado apenas em determinados navegadores, quando a aplicação de tecnologias avançadas, você deverá sempre perguntar, "Se o navegador não mostrar o que estou a tentar conseguir, o site pode ainda ser utilizados?"

A sua é problemático para algumas pessoas, especialmente clientes, que insistem que um site deve procurar e execute "exactamente o mesmo em cada navegador." Na idade do iPhone e ágil web design, no entanto, isso não é mais uma expectativa realista. Em vez disso, você precisa para tratar a adição de recursos avançados de CSS para um site como uma série de cenários what-if:

 Se você usar CSS3 para criar uma apresentação de slides de imagens em uma página da web (como mostrado no Capítulo 6), e o navegador não oferece suporte a ele, o que acontece? Se o usuário vê uma imagem de espaço reservado estática no lugar do diaporama, é que bem? Ou você precisa para usar o JavaScript como um fallback?

 Se você usar CSS3 para aprimorar uma barra de navegação do site com animação, é normal que alguns usuários não ver a animação? Eles podem ainda usar a barra de navegação sem ele?

Ele vários exemplos e tutoriais em todo este livro irá demonstrar diferentes soluções para alcançar a compatibilidade com versões anteriores, como não há uma prática ou técnica que é aplicável em todos os casos. Ao mesmo tempo, eu também vou ser enfatizando a acessibilidade; que é, permitindo que os utilizadores com diferentes necessidades e capacidades (tais como o cego, ou os visitantes do site que use um teclado sem um rato) para acessar seu trabalho.

Por que razão CSS3 em vez de JavaScript ou Flash?

Você pode encontrar uma lista completa das vantagens e desvantagens do uso do CSS3 em vez de JavaScript ou Flash na Tabela 2.

**Tabela 1-2.***CSS3 versus JavaScript e Flash*

**CSS3**

**Vantagens** **Desvantagens**

Se baseia na terra; usa um familiar estabelecido CSS sintaxe. Simples de entender.

Enquanto historicamente bem suportado no Firefox e Safari e Chrome, compatibilidade no IE e Opera está limitado a recente constrói.

Mais rápido e mais suave forma de animação em um navegador, com maiores taxas de quadros de JavaScript.

Manipula conteúdo HTML existentes, melhorando a SEO.

Existem relativamente poucas ferramentas gráficas para criar código de animação, forçando um grau de codificação manual. (embora isso está mudando; consulte o Capítulo 10).

|  |  |
| --- | --- |
|  | **Flash** |
|  |  |
| **Vantagens** | **Desvantagens** |
| Bem-estabelecida GUI para criar animações. | Não pode ser reproduzida em muitos dispositivos móveis |
|  | (mais significativamente o Apple iPhone e iPad juntos |
|  | Com Android e Windows 8 dispositivos). |

Loops, variáveis e funções fazer ActionScript (linguagem de script do Flash) mais potente do que o CSS.

Depende da presença de um plug-in que deve ser regularmente actualizado pelo usuário.

Tende a "bloquear" conteúdo, tornando inacessíveis para CSS, motor de pesquisa robôs, dispositivos de acessibilidade, tais como leitores de tela, leal e casos de uso

**CSS3**

**Vantagens** **Desvantagens**

Se baseia na terra; usa um familiar estabelecido CSS sintaxe. Simples de entender.

Enquanto historicamente bem suportado no Firefox e Safari e Chrome, compatibilidade no IE e Opera está limitado a recente constrói.

Mais rápido e mais suave forma de animação em um navegador, com maiores taxas de quadros de JavaScript.

Manipula conteúdo HTML existentes, melhorando a SEO.

Existem relativamente poucas ferramentas gráficas para criar código de animação, forçando um grau de codificação manual. (embora isso está mudando; consulte o Capítulo 10).

|  |  |
| --- | --- |
|  | **Flash** |
|  |  |
| **Vantagens** | **Desvantagens** |
| Bem-estabelecida GUI para criar animações. | Não pode ser reproduzida em muitos dispositivos móveis |
|  | (mais significativamente o Apple iPhone e iPad juntos |
|  | Com Android e Windows 8 dispositivos). |

Loops, variáveis e funções fazer ActionScript (linguagem de script do Flash) mais potente do que o CSS.

Depende da presença de um plug-in que deve ser regularmente actualizado pelo usuário.

Tende a "bloquear" conteúdo, tornando inacessíveis para CSS, motor de pesquisa robôs, dispositivos de acessibilidade, tais como leitores de tela, leal e casos de uso.

**JavaScript**

**Vantagens** **Desvantagens**

Bem apoiado e com um host de quadros.

A utilização de um quadro exige uma solicitação HTTP extra, diminuindo o tempo de carregamento da página.

|  |  |
| --- | --- |
| Funciona através da manipulação de elementos HTML via | Sintaxe pode ser um pouco obtuso, |
| O DOM e CSS, conhecida de qualquer desenvolvedor da Web. | Mesmo com um quadro. |
| Loops, variáveis e funções fazer | O conteúdo gerado por JavaScript não está |
| O idioma mais potente do que o CSS. | Indexado por motores de busca. |
|  |  |

Outras tecnologias

Existe aqui uma certa confusão sobre os papéis de outras novas tecnologias da web, especialmente em clientes, pelo que vale a pena discutir o CSS3 A animação não é*.*

 CSS3 não é HTML5. Enquanto as duas tecnologias tendem a ser falado no mesmo fôlego, CSS3 não está relacionado com o HTML5. Markup não apresentação: CSS3 pode ser igualmente aplicado ao XHTML ou HTML3.1. Neste livro, você estará usando HTML5 como seu markup, mas você não precisa.

 CSS3 não é Canvas. <lona> é um elemento HTML5 que cria um JavaScript-acessível "área de desenho" em uma página da web. Ele localização do <galpão de> superfície é definida pelo CSS, mas qualquer animação que ocorre dentro é controlada pelo JavaScript.

7

Capítulo 1 ■ FUNDAMENTOS CSS3

 CSS 3D transforma são parte do CSS transforma o módulo, não animação. O CSS transforma são utilizadas para manipular a perspectiva visual de elementos HTML. stas transformações pode ser animado, mas CSS 3D transforma (discutido no Capítulo 9) não são animações em si próprias.

 WebGL não é CSS animação. WebGL é um JavaScript API 3D que manipula os desenhos no elemento <lona>.

Resumo

Neste capítulo você aprendeu como CSS é desenvolvido e padronizado, e o papel da W3C nesse processo. Início os desenvolvedores da web por vezes ver o W3C como uma espécie de Overlord benigna, a ditar as normas do alto; a realidade é que a organização e os seus diferentes grupos de trabalho é realmente um integrador e caráter uniformizador de inovações criadas por fornecedores de navegador.

Enquanto eles permanecem em estado experimental, novas propriedades CSS são até para ganhar quando se trata de como eles são implementados. Para este fim, fornecedor prefixos, específicos para cada navegador, são utilizados para distinguir um navegador maker da interpretação de uma nova propriedade CSS. É apenas quando a propriedade é padronizada pelo W3C e suporte é integrado ao software que um navegador vai interpretar uma versão nonprefixed.

Enquanto CSS3 Animação, transições e transformações têm um certo número de vantagens significativas sobre as soluções tradicionais de animação web de Flash e JavaScript, sua relativa novidade lhes limites bastante recente navegador constrói, mais especialmente no caso do Internet Explorer e Opera. É importante considerar as técnicas de regressão durante o desenvolvimento de modo a que os utilizadores com versões mais antigas de navegadores ou que estão dependentes de dispositivos de assistência tais como leitores de tela não perca o seu conteúdo do site. (é igualmente importante para comunicar essas questões para clientes e outros desenvolvedores de web, que freqüentemente pode escolher a partir de uma "palavra" de salada periférico ou tecnologias independentes, tais como HTML5, ao tentar falar sobre animação web).

No próximo capítulo, vou introduzir a sintaxe para CSS transforma e como criar transições CSS, a forma mais simples de animação CSS3. Embora haja um pouco de matemática, vamos fermento isso comparando as CSS3 animações com exemplos do mundo real de movimento, incluindo animação clássica técnicas empregadas pela Disney.

**Capítulo 2**

**CSS3 transforma e Transições**

Enquanto CSS animações podem ser usadas para alterar quase todos os aspectos de um elemento HTML (com excepção das propriedades enumeradas no capítulo anterior), alguns dos mais poderosos meios de manipular a apresentação de páginas da web reside no CSS transforma e transições de todos os módulos que são inteiramente nova no CSS3.

Transições CSS são muito a forma mais simples de animação: um movimento entre dois Estados. Uma vez que você domina a sintaxe fundamental para transições descritas neste capítulo, você será capaz de aplicar animações simples e eficaz de imagens (descrito no capítulo 3 e os elementos da interface de usuário (descrita no Capítulo 4), e então começar a criar animações mais complexas keyframed (descrito no capítulo 5 e mais além).

O CSS transforma

Existem quatro principais funções de tradução CSS: traduzir, gire, escala e inclinação. As funções são combinadas em função da transformação da matriz. Você vai aplicar estas transformações para um layout de página da web padrão de uma imagem exposta junto a um parágrafo de texto conforme mostrado na cotação de 2-1.

**Lista 2-1.**Código HTML5 para uma imagem lançada

<!DOCTYPE HTML> <html>

<cabeça>

<título > simples transformação CSS3</title></head>

<body>

<p> < img src = "dudley andares-estatueta.jpg" alt = "aluno-feitas estatueta de Dudley Andares"

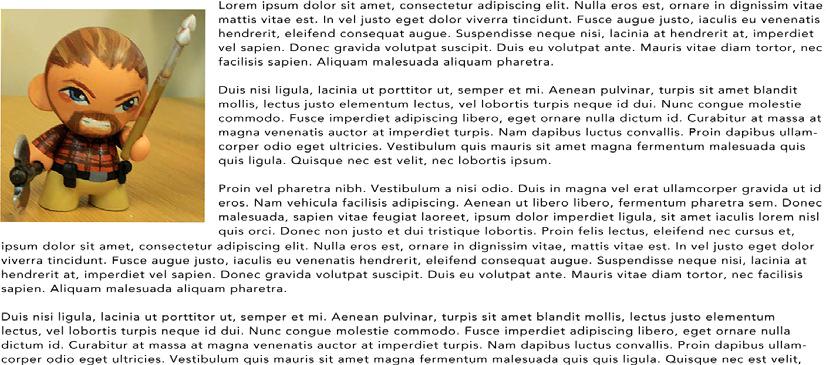
Estilo = "Largura: 300px; altura: 300px; flutuação: margem esquerda;: 0 2em 1.4em 0;" > Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ue mi tellus. Vestibulum tortor erat auctor facilisis, em semper, "Pharetra" quis mi...</p>

</body>

</html>

O código mostrado na listagem 2-1 irá produzir o esquema mostrado na Figura 2-1.

Capítulo 2 ■ CSS3 transforma e transições



**Figura 2-1.***Uma imagem lançada com o texto do parágrafo*

Com esta página básico no lugar, você pode começar a aplicar transforma a imagem do elemento.

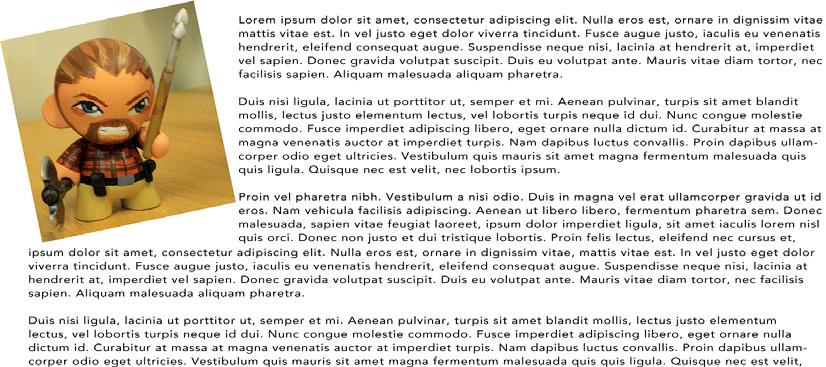
Gire

Primeiro, você vai transformar a imagem girando (consulte a listagem de 2-2). Valores para CSS3 transformações de rotação pode ser especificado em graus, gradianos, voltas ou radianos, usando positivo ou negativo valores de ponto flutuante para criar no sentido dos ponteiros do relógio ou no sentido contrário ao dos ponteiros do relógio a rotação. Você deve incluir prefixos de fornecedor para cobrir todos os navegadores.

**Lista 2-2.**Inline CSS para girar uma imagem

<img src = "dudley andares-estatueta.jpg" alt = "estatueta de Dudley estilo andares" = "Largura: 300px; altura: 300px; flutuação: margem esquerda;: 0 2em 1.4em 0; -moz-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg); ">

O resultado do código na lista 2-2 é mostrado na Figura 2-2.



**Figura 2-2.***Uma imagem exposta com um CSS gire transformação*

Durante a medição de rotação em graus é a abordagem mais comum quando escrever o CSS transformações, CSS3 permite uma grande variedade de unidades, mostrado na Tabela 2-1:

**Tabela 2-1.***Sistemas de unidade possível para o tipo de dados do ângulo do CSS*

|  |  |  |  |
| --- | --- | --- | --- |
| **Unidade** | **CSS** | **Descrição** | **Exemplo** |
|  |  |  |  |
| Graus | Deg | 360 graus em um círculo | Gire(90graus) |
| Gradianos | Grad | Também conhecido como "gons" ou "qualidades". 400 gradianos | Gire(100grad) |
|  |  | Em um círculo, tornando mais fácil o cálculos. |  |
| Radianos | Rad | 2p radianos em um círculo completo, igual ao 6.2831853rad. | Gire(1,57rad) |
| Voltas | Rodar | Uma rotação completa = 1 volta completa. | Gire(.25volta) |
|  |  |  |  |

Há algumas coisas a nota quando flutuante uma imagem usando o comando Girar:

 Outros conteúdos HTML na página não é afetado por transformações: o layout dos parágrafos não muda em resposta à rotação da imagem; maior rotação da imagem poderia fazer com que ele se sobrepõem ao texto. (Conteúdo que reage a transforma é suportado no módulo de regiões CSS).

 O Modelo de Objeto de documento (DOM) é igualmente afectada; o valor de propriedades para o elemento transformado como offsetWidth também será inalterada.

 CSS transformações essencialmente impor um estado de posicionamento relativo no elemento afetado; o espaço original usado pelo elemento é retido.

 Se o valor da propriedade de transbordo é de deslocamento ou auto, barras aparecerão como necessário para permitir que você visualize o conteúdo que é transformado fora da área visível.

 A rotação ocorre a partir do *centro de tomografia* do elemento, a transformar a origem.

 Outro aspecto CSS regras aplicadas ao elemento, como caixa-sombra, são aplicados *antes*da transformação, para que elas possam ser girados com o efeito.

Capítulo 2 ■ CSS3 transforma e transições

 Rodar a imagem 180 graus não flip ou espelho; que pode ser alcançado com o uso de um hack da transformação da escala, discutido posteriormente neste capítulo ou uma rotação 3D, discutido no Capítulo 9.

 Você pode rodar qualquer conteúdo HTML você quiser, mas a partir de uma perspectiva de design não é recomendável que você gire o texto: isso reduz a legibilidade e induz uma dolorosa crick no pescoço para seus leitores.

 A unidade de medida precisa estar presente, mesmo se a quantidade de rotação é 0. Na maioria das medições de CSS, 0 é 0 para qualquer unidade, (ou seja, largura: 0 funciona como uma alternativa à largura: 0px.) Mas quando girar a 0, você deve especificar a transformar: gire(0ºC); transformar: gire(0) não funcionará.

Como você pode ver, estilos inline para transformações pode ser prolongado devido à obrigação de o fornecedor incluir prefixos. É muito mais comum para criar transformações separadamente, como uma classe ou id em uma folha de incorporado ou vinculado (consulte a listagem de 2 a 3).

**Lista 2-3.**Uma folha de estilos CSS incorporado para transformar uma imagem

<!DOCTYPE HTML> <html>

<cabeça>

<título > simples transformação CSS3</title> <style>

Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

}

</Estilo>

</cabeça>

<body>

<p > <img src = "dudley andares-estatueta.jpg" alt = "estatueta de Dudley estilo andares" = "margem: 0 2em 1.4em 0; classe = "tilt" > Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ue mi tellus. Vestibulum tortor erat auctor facilisis, em semper, "Pharetra" quis mi...</p>

</body>

</html>

Para girar a imagem como se fosse travado no seu canto superior direito, você deve mover o transformar do elemento de origem para que local, como mostrado na lista 2-4.

**Lista 2-4.**Girar uma imagem a partir de um canto

Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda; -moz-transformar-origem: No canto superior direito;

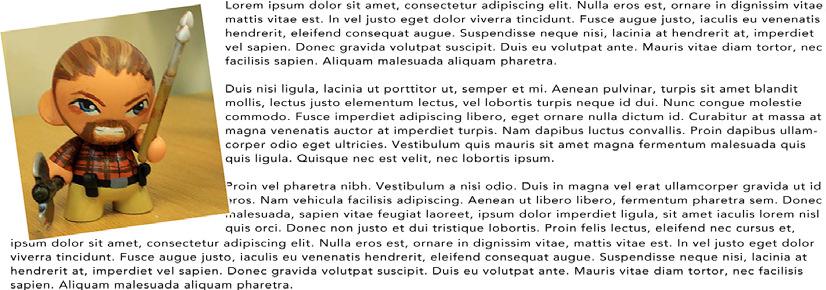
-o-transformar-origem: No canto superior direito; -ms-transformar-origem: No canto superior direito; -webkit-transformar-origem: No canto superior direito; transformar-origem: No canto superior direito;

-moz-transformar: gire(-10graus); -o-transformar: gire(-10graus); -ms-transformar: gire(-10graus); -webkit-transformar: gire(-10graus); transformar: gire(-10graus);

}

Capítulo 2 ■ CSS3 transforma e transições

O código na listagem 4 irá criar o resultado mostrado na Figura 2-3; observe que eu já tinha de mudar a imagem em linha do estilo ligeiramente para fornecer mais espaço para a margem do lado direito para compensar o novo ângulo da imagem.



**Figura 3.***Girado, flutuado imagem com o texto do parágrafo*

Transformar os valores de origem leva a mesma forma de fundo-posição e outras propriedades que combinam um deslocamento horizontal e vertical. Os valores são especificados como a posição horizontal do ponto de origem seguido a posição vertical *em relação ao próprio elemento.* Os valores podem ser especificados como palavras-chave (central superior, inferior, esquerda e direita), numericamente, ou como uma combinação dos dois. Eles também podem ser fora da área do elemento em si mesmo (por exemplo, para criar uma transformação eixo origem acima ou abaixo do elemento , como mostrado no cartão "fan" galeria de imagens de exemplo no Capítulo 3).

O Webkit CSS3 Transformar Aliasing questão

As primeiras versões do Chrome e Safari (até à versão 5.1) contêm um bug de renderização: quando transformando alguns elementos, os navegadores não antialias as bordas do rodado ou inclinadas conteúdo HTML, resultando no chamado "jaggies" ou "staircasing" nas bordas das imagens, como mostrado na Figura 4.

Capítulo 2 ■ CSS3 transforma e transições



**A Figura 2-4.***Imagem ampliada, mostrando aliasing de bordas em uma imagem girada com CSS transformar*

Existem várias técnicas para contornar este bug :

 Aplicar um 1-pixel borda branca em volta do elemento.

 Aplicar webkit-rejeição de fundo de face de visibilidade: oculto; para o elemento.

 Adicionar outro elemento para a transformação, tais como -webkit-transformar: gire(-10graus) translateZ(0);.

Não há nenhuma técnica única que atende melhor a renderização bug em todas as circunstâncias, no entanto; a eficácia de cada técnica depende do contexto do elemento sendo prestados.

Escala

A escala transformar é algo de um oddity quando aplicado às imagens: dado que a alteração de uma altura e largura da imagem terá muito o mesmo resultado visual, talvez ele não me parece útil. A diferença é que a escala pode ser aplicada a *qualquer* elemento HTML: alterando a largura de um parágrafo será reflow o conteúdo do texto, mas alterando sua escala fará com que o texto fisicamente maior ou menor.

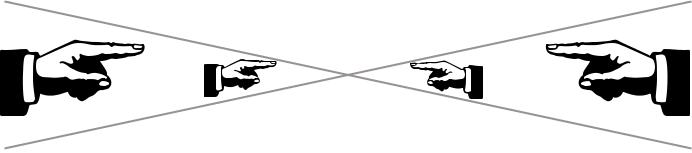
O valor de escala é um multiplicador: escala(2) aplicado a um elemento irá fazer com que apareça o dobro de largura e duas vezes mais alta (em outras palavras, quatro vezes o seu tamanho normal), enquanto escala (.5) resultará em uma imagem que é um quarto

Capítulo 2 ■ CSS3 transforma e transições

Seu tamanho original. escala irá transformar o elemento igualmente em todas as direções. Você também pode aplicar escala ordinal de direcções: X (horizontal), Y (vertical) e Z (profundidade).

Inversão de Imagens com scaleX

Você pode usar a escala de transformação de CSS para efetivamente espelho elementos HTML (geralmente imagens, embora em princípio esta técnica pode ser aplicada a qualquer elemento). Se a escala começa em 1, como mostrado no lado esquerdo do diagrama na Figura 2-5, o elemento afectados irá crescer menor como você baixar o valor de escala até chegar a 0, quando a imagem desaparece. Se você empurrar o valor em território negativo, a imagem vai começar a crescer novamente, mas aparecerá movido horizontalmente, como mostrado no lado direito da Figura 2-5.



ScaleX(0)

ScaleX(1) ScaleX(0,5) ScaleX(-0,5) ScaleX(-1)

(original) (invertidas)

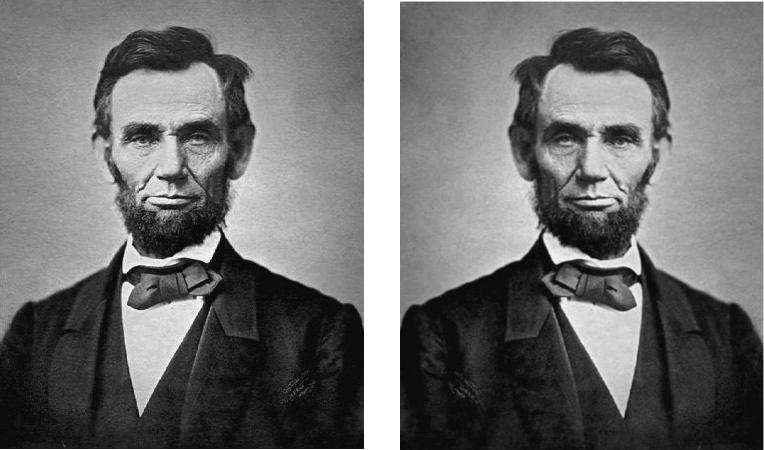
**Figura 2-5.***Efeito do uso de pequenas e valores negativos de scaleX()*

Você pode usar escala para flip rapidamente uma imagem em uma página, em vez de sua transformação através de um aplicativo como o Adobe PhotoShop para gerar uma nova cópia. Lista 2-5 mostra como a transformação poderia ser aplicada para inverter uma imagem de Abraham Lincoln.

**Lista 2-5.**Inverter uma imagem com um estilo de transformar em linha

<img src="lincoln.jpg" alt="Abraham Lincoln, 1863" style="Largura: Pinceâis 389px; altura: 480px; -moz-transformar: scaleX( 1); -o-transformar: scaleX( 1); -ms-transformar: scaleX( 1); -webkit-transformar: scaleX( 1); transformar: scaleX( 1);">

A figura 6 mostra o resultado do código na cotação de 2-5.



**Figura 2-6.***Fotografia original de Abraão Lincoln (esquerda) movido usando CSS3 escala transformar (direita)*

Capítulo 2 ■ CSS3 transforma e transições

Usando técnicas de CSS como este, você pode ajustar as imagens sobre a voar em vez de ter de voltar para PhotoShop, faça as alterações e clique em Salvar e carregar o arquivo para o local e sem a necessidade de modificar qualquer

Código HTML.

Traduzir

Como a escala, o modificador de traduzir pode parecer um pouco redundante em primeira: ele usa o mesmo sistema de coordenadas (e visualmente, produz o mesmo resultado) como aplicação superior, esquerda, inferior e direita propriedades para um elemento relativamente posicionados. No entanto, como você verá, traduzir pode tornar mais fácil para animar o conteúdo HTML.

Traduzir(x,y) move o elemento em direcções horizontal e vertical através de valores positivos ou negativos. translateX() move o elemento no plano horizontal, enquanto translateY() move verticalmente.

Por exemplo, se você queria para mover a imagem estatueta mostrado na Figura 2-4 até 4em e para a direita pelo 50px, você usaria o código mostrado na cotação de 2-6.

**Lista 2-6.**O código CSS para traduzir uma imagem

Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transformar: traduzir(50px, -4em); -o-transformar: traduzir(50px, -4em); -ms-transformar: traduzir(50px, -4em); -webkit-transformar: traduzir(50px, -4em); transformar: traduzir(50px, -4em);

}

Inclinação

Aplicando a inclinação um elemento "tesoura" na horizontal ou na vertical e pode ser útil para transmitir uma sensação de velocidade extra ou de movimento de um elemento. Imagine tirar os lados opostos de um retângulo (as bordas superior e inferior, por exemplo, ou os lados direito e esquerdo) e puxando-os em diferentes direções, garantindo que eles permaneçam paralelos.

Os valores introduzidos para inclinação consulte o ângulo em que as *outras* partes será definida. Por exemplo, "analisando" uma imagem à direita é uma transformação skewX. transformar: skewX(21graus) significará que as bordas esquerda e direita da imagem será ajustada a 21 graus da vertical (consulte a listagem de 2-7). Inclinar a imagem para a esquerda ainda usa skewX, mas com um valor negativo: skewX(-21graus), por exemplo, irá definir as mesmas bordas 21 graus negativos (isto é, esquerda) a partir da vertical. skewY leva os   lados esquerdo e direito de um elemento da caixa de mudanças e eles para cima e para baixo.

**Lista 2-7.**O código CSS para inclinação de uma imagem

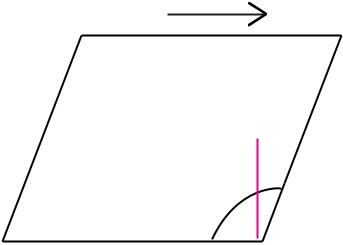
Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transformar: skewX(21graus); -o-transformar: skewX(21graus); -ms-transformar: skewX(21graus); -webkit-transformar: skewX(21graus); transformar: skewX(21graus);

}

Você pode ver o resultado da Listagem de 2-7 na Figura 2-7.



21 °



SkewX(21graus)

**Figura 2-7.***Um elemento retangular inclinada com CSS*

Uma combinação de ambos a inclinação horizontal e vertical com valores adequados ao converter o elemento no itinerário correspondente pode fornecer a impressão de que o elemento formas de um lado de uma caixa, como mostrado na lista 2-8.

**Lista 2-8.**O código CSS para transformar uma imagem em um lado de uma caixa de isométrico

Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transformar: skewY(30graus); -o-transformar: skewY(30graus); -ms-transformar: skewY(30graus); -webkit-transformar skewY(30graus); transformar: skewY(30graus);

}

Combinando transformações individuais (por exemplo, uma rotação e translação) fornece muito mais energia para seu CSS e oferece a você muitos mais possibilidades de animação.

Combinando Transformações

Você pode mesclar as transformações de um elemento em uma de duas formas: como espaço de valores separados de uma propriedade transformar, ou como os valores de uma matriz propriedade.

Para mesclar transformações como espaço de valores separados de um transformar propriedade, use o código mostrado na cotação de 2-9.

**Lista 2-9.**Vários transforma em uma única declaração de CSS

Img.incline { Largura: 300px; altura: 300px; bóia: esquerda; -moz-transformar: traduzir(50px, -4em) gire(15graus); -webkit-transformar: traduzir(50px, -4em) gire(15graus); -o-transformar: traduzir(50px, -4em) gire(15graus); -ms-transformar: traduzir(50px, -4em) gire(15graus); transformar: traduzir(50px, -4em) gire(15graus); }

O processo para a fusão de transformações como os valores de uma matriz propriedade é significativamente mais complicada. Transformações de matriz são um pouco além do escopo deste livro; é mais fácil usar uma ferramenta para gerar o código. A matriz Useragentman Conjunto de Construção ([www.useragentman.com/matrix/](http://www.useragentman.com/matrix/)) e CSS3 Transformar Calculadora Matriz ([www.leeourand.com/test/transform/test/transform.html](http://www.leeourand.com/test/transform/test/transform.html)) oferecem duas maneiras de o fazer. Uma explicação da matriz transformações pode ser encontrado na matriz de transformação para o CSS matematicamente desafiados ([www.useragentman.com/blog/2011/01/07/css3-matrix-transform-for-the-mathematically-challenged/](http://www.useragentman.com/blog/2011/01/07/css3-matrix-transform-for-the-mathematically-challenged/))

Capítulo 2 ■ CSS3 transforma e transições

E no site do desenvolvedor da Web Opera (<http://dev.opera.com/articles/view/understanding-the-css-transforms-matrix>). Enquanto eles têm a vantagem de ser mais curto e mais eficiente, matrix transforma não são legíveis por humanos, então eu não os utilizar para os exemplos neste livro.

■ **nota**por escrito separar transforma não irá funcionar para criar uma combinação de transformação.

Img.incline { Largura: 300px; altura: 300px; bóia: esquerda; transformar: traduzir(50px, -4em);

Transformar: gire(15graus); -webkit-transformar: traduzir(50px, -4em); -webkit-transformar: gire(15graus); }

Com o CSS acima o navegador irá seguir a última linha de código aplicável; que é, a imagem será girado, mas não traduzida.

Transições CSS

Transições CSS são exactamente que: a transição de um estado para outro, visual mais frequentemente do início por alguns evento de utilizador, tais como um mouseover em um elemento. Transições, em outras palavras, são ponto-a-ponto. Se você precisar para animar entre mais de um Estado e outro que você vai encontrar o CSS Keyframes são mais adequadas para o trabalho. (CSS Keyframes será discutido no Capítulo 5. )

Note que para os exemplos neste capítulo eu vou estar usando o mouse para iniciar as transições :mas tecnicamente *qualquer* modificação do valor de uma propriedade do elemento irá disparar uma transição para essa propriedade.

Vamos voltar para o primeiro exemplo e criar uma simples rotação transição para a imagem na página. Quando o usuário coloca o mouse sobre a imagem que deseja girar o elemento de 7,5 graus. Você vai fazer isso adicionando um mouse pseudo classe de :o .seletor da inclinação (:foco pode ser aplicada a cada elemento, não apenas links), como mostrado na listagem 2-10.

**Listando 2-10.**CSS transformar em pairar, nenhuma transição

<style> img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

}

</Estilo>

O código mostrado na listagem 2-10 funcionará, mas se você tente exibir a página em um navegador que você verá que não há animação sobre mouseover, apenas um movimento instantâneo entre um Estado e outro. Você vai criar uma animação entre estes estados usando a propriedade de transição (consulte a listagem 2-11).

**Listando 2-11.**Suavizando uma CSS transformar usando uma transição

Img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

-moz-transição: 2s todos; -webkit-transição: 2s todos; -o-transição: 2s todos; transição: 2s todos;

}

O código mostrado na listagem 2-11 é muito mais bem sucedida: quando você passa o mouse sobre a imagem que você verá que ele agora gira suavemente para a sua nova posição. A sintaxe, repetido com prefixos de vários fornecedores, é fácil de compreender, demasiado. O elemento é girada mais de dois segundos, e todas as suas propriedades podem ser alteradas durante a transição. Note que a ordem dos valores não importa: você pode usar 2s todos ou todos os 2s.

Se você for animar elementos sobre períodos de tempo que incluem frações de segundos, você pode especificar o período de tempo como valores de ponto flutuante em segundos, ou como milissegundos (milésimos de segundo), como mostrado na listagem 2-12.

**Listando 2-12.**Uma transição CSS medido em segundos

Img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

-moz-transição: 2,35s todos; -webkit-transição: 2,35s todos; -o-transição: 2,35s todos; transição: 2,35s todos;

}

Isso também poderia ser expressa como mostrado na listagem 2-13.

**Listando 2-13.**Uma transição CSS medido em milissegundos

Img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

-moz-transição: 2350ms todos;

-webkit-transição: 2350ms todos; -o-transição: 2350ms todos; transição: 2350ms todos;

}

Durante a sincronização de animação em milissegundos permite uma maior precisão, as duas declarações acima de alcançar o mesmo resultado de milissegundos não criar uma sequência de animação mais suave. Muito poucas animações exigirá precisão para baixo para um milésimo de segundo, e especificando o tempo em milissegundos geralmente requer mais digitação, por isso fico com o formato mais familiar segundos (mesmo para os valores de menos de um segundo: transição:

.35s todos, por exemplo). Você deve usar o sistema que você se sente mais confortável.

■ **Nota**Se você tiver animado com JavaScript, observe a diferença aqui: CSS3 pode usar valores de ponto flutuante em segundos ou milissegundos para sincronização de animações. JavaScript usa milissegundos exclusivamente, embora muitas estruturas JavaScript usado para criar animações podem usar intervalos de tempo medido em segundos.

Não é apenas mais uma melhoria para fazer. Você notará que depois do elemento ter sido girado, movendo o mouse para fora da imagem retorna instantaneamente para o seu estado inicial. Enquanto que pode ser o efeito visual que procura para elementos de página da web em algumas circunstâncias, na maioria dos casos é melhor para mostrar o elemento regresse à sua orientação inicial tão bem como ele chegou no seu estado girado.

A solução é ligeiramente contraditório: mova a transição porção do código CSS da declaração de passe :  estado padrão para a imagem, mantendo apenas a transformar no :hover declaração (consulte a listagem 2-14).

**Listando 2-14.**Criando uma transição suave para e a partir de um Estado Padrão

<style> img.incline {

Largura: 300px; altura: 300px; bóia: esquerda; -moz-transição: 2s todos; -webkit-transição: 2s todos; -o-transição: 2s todos; transição: 2s todos;

}

Img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

}

</Estilo>

A ideia é simples: colocação da propriedade de transição sobre a declaração de classe implica que qualquer transição no efeito tanto *de* e de *volta para* este estado. O exemplo anterior colocou a transição no :hover declaração, o que significa que ele só foi eficaz no mouse pairar, não durante o retorno ao estado normal.

Também vais notar que a transição pode ser interrompido se você mover seu mouse para e a partir da área da imagem; seu movimento será invertida suavemente. Você pode atalho o código mais por apenas especificar o tempo de transição (consulte a listagem 2-15).

**Listando 2-15.**Rotação temporizada em um CSS Transformação

Img.incline {

Largura: 300px; altura: 300px; bóia: esquerda; -moz-transição: 2s;

-webkit-transição: 2s; -o-transição: 2s; transição: 2s; }

Img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

}

Como você viu, criando uma suave e animação simples é fácil com transições CSS3. Você pode modificar e animar a quase todos os aspectos de um elemento da aparência que fornecem acesso a propriedades de CSS. As transições tenho mostrado até agora mudaram apenas um aspecto de um elemento por vez e sempre da mesma maneira.

Para criar animações mais ricos pode combinar várias transições de propriedade para o mesmo elemento que ocorrem em momentos diferentes e velocidades.

Atrasar e combinando os efeitos de transição

Um evento de transição pode ser atrasada por adição de uma transição de atraso, quer como uma propriedade separada ou anexados aos valores para transformar:

-moz-transição: 2s 4s;

-webkit-transição: 2s 4s; -o-transição: 2s 4s; transição: 2s 4s;

Note que o atraso tem efeito tanto no início da animação e o início da reversão do elemento para o seu ponto de início. A animação irá começar até quatro segundos após o cursor tem sido realizada sobre a imagem; uma vez que esteja totalmente rodado, o elemento irá permanecer no lugar durante quatro segundos antes de retornar à sua orientação padrão.

(nota também que a animação não começa até que o mouse foi realizada sobre a imagem por *pelo menos* quatro segundos).

Você pode animar várias propriedades CSS ao mesmo tempo adicionando ao estado o mouse (ver :listando 2-16).

**Listando 2-16.**Várias propriedades CSS migrou Simultaneosly

<style> img.incline {

Largura: 300px; altura: 300px; bóia: esquerda; -moz-transição: 2s;

-ms-transição: 2s; -o-transição: 2s; -webkit-transição: 2s; transição: 2s;

}

Img.incline:foco { -moz-transformar: gire(15graus);

-o-transformar: gire(15graus); -ms-transformar: gire(15graus); -webkit-transformar: gire(15graus); transformar: gire(15graus); opacidade: .3;

}

</Estilo>

As propriedades podem ser dadas as temporizações separada na animação declarando transição duração como uma propriedade com valores separados por vírgula. Digamos que você queria para mover a imagem para a direita sobre o mouse e fade-out ao mesmo tempo, mas com o fade tendo metade do tempo que o movimento não (consulte a listagem 2-17).

**Listando 2-17.**Uma CSS3 Transição de várias propriedades com temporizações diferentes para cada

<style> img.incline {

Largura: 300px; altura: 300px; bóia: esquerda; POSIÇÃO: relativos; -moz-transição de propriedade: opacidade, esquerda; -o-transição de propriedade: opacidade, esquerda; -webkit-transição de propriedade: opacidade, esquerda; propriedade de transição: opacidade, esquerda;

-moz-transição duração: 2s, 4s; -o-transição duração: 2s, 4s; -webkit-transição duração: 2s, 4s; transição duração: 2s, 4s;

}

Img.incline:foco { opacidade: .2; esquerda: 60px;

}

</Estilo>

Eu adicionei posição: em relação a fim de ser capaz de mover o elemento alterando o valor da sua esquerda e melhorou a eficiência da animação, declarando claramente as propriedades para ser animado. (Obviamente você não ter como prefixo para propriedades que são bem aceitos em todos os navegadores, tais como opacidade.) Você notará que a esquerda-direita animação podem não ser particularmente bom em alguns navegadores. Vamos alterar a propriedade animada para traduzir, como mostrado na listagem 2-18.

**Listando 2-18.**Uma CSS3 Transição de tradução

<style> img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transição de propriedade: opacidade, translateX;-o-propriedade de transição: opacidade, translateX; -webkit-transição de propriedade: opacidade, translateX;

Propriedade de transição: opacidade, translateX; -moz-transição duração: 2s, 4s; -o-transição duração: 2s, 4s; -webkit-transição duração: 2s, 4s; transição duração: 2s, 4s;

}

Img.incline:foco { opacidade: .2;

-webkit-transformar: translateX(60px);

-moz-transformar: translateX(60px); -ms-transformar: translateX(60px); -o-transformar: translateX(60px); transformar: translateX(60px);

}

</Estilo>

Você pode achar que o movimento está agora mais suave; traduzir é uma boa alternativa para animar a circulação de elementos HTML através da manipulação do posicionamento absoluto ou relativo.

Apresentando as funções de facilitar

Observar atentamente o movimento da imagem no mouseover nas animações que você criou até agora: há algo um pouco especial sobre ele (prolongamento o valor de tempo para a animação poderá ajudar a tornar mais claro). O movimento da imagem não é mecânica, mas biológicos: a partir de sua posição padrão, acelera a imagem conforme ela gira, atinge uma velocidade constante para um momento e abranda antes se trata de resto.

Na animação, este tipo de movimento é referido como a *facilidade em/facilidade out*. É o movimento de objetos no mundo quotidiano. Por exemplo, nenhum carro, não importa quão poderosas, pode atingir uma velocidade de 0-60 recorde de 0 segundos. Cada objeto em movimento a uma determinada velocidade acelera; no fim do seu curso (fora de uma situação extrema, tais como um carro colidir com uma parede de tijolos em velocidade superior) o objeto irá desacelerar antes de chegar a uma paragem.

Em CSS3 animação, *facilitar* as transições são o padrão; não há necessidade de estado que você estiver usando. Se você deseja usar a animação com um mais "mecânica", você pode começar especificando uma transição linear (consulte a listagem 2-19).

Capítulo 2 ■ CSS3 transforma e transições

**Listando 2-19.**CSS para uma transição de Rotação Linear

<style> img.incline {

Largura: 300px; altura: 300px; bóia: esquerda;

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

-moz-transição: 2s transformar linear;

Webkit-transição: 2s transformar linear; -o-transição: 2s transformar linear; transição: 2s transformar linear;

}

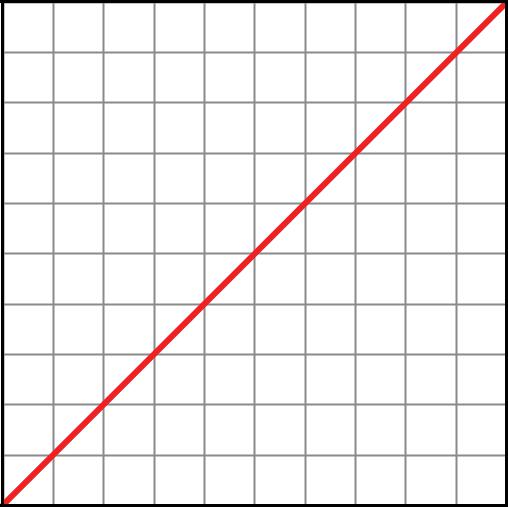
</Estilo>

Você verá que o movimento da imagem no mouseover é agora muito mais mecânica.

Funções de distribuição de transição e linhas vectoriais assim curvas

Linear e facilidade são apenas duas formas de quais são referidas como *funções de distribuição*, que é, descrições do caminho em que um objeto fica a partir de um ponto a para um ponto B em uma linha reta. Estas funções de distribuição podem ser colocados em gráfico em uma expressão matemática conhecida como linhas vectoriais assim curvas.

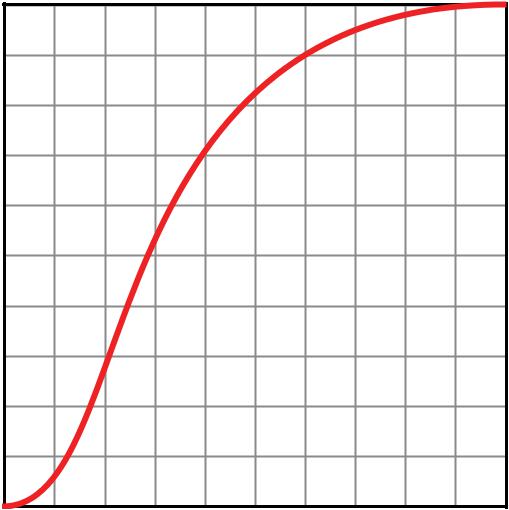
Por exemplo, se você em gráfico o movimento de um elemento durante a sua transição de 0 a 15 graus sob condições linear, com o tempo atribuído para o eixo horizontal e o ângulo da imagem atribuída a vertical do gráfico para animação linear seria semelhante a Figura 2-8. À medida que o tempo avança, o ângulo de rotação aumenta em sintonia com o passar dos segundos, criando uma taxa constante de movimento.



**Figura 2-8.***Gráfico linear de animação: tempo no eixo x (horizontal), distância/ângulo no eixo y (vertical). Nota a relação directa*

Remover a palavra-chave linear a partir da declaração retorna a animação para o seu estado natural facilitando, que quando plotados sobre os mesmos eixos pareça mais como Figura 2-9.

apítulo 2 ■ CSS3 transforma e transições



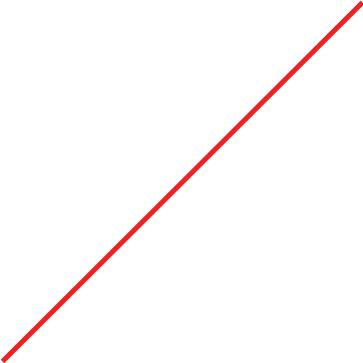
**Figura 2-9.***Facilitando a função de sincronização: tempo no eixo x, distância no eixo y*

Como você pode ver, o ângulo de rotação muda lentamente nos primeiros momentos de animação mais ventajosas; para o meio da transição, a taxa de mudança aumenta acentuadamente, atinge uma velocidade de "Terminal", depois diminui até que a seqüência atinge a sua conclusão.

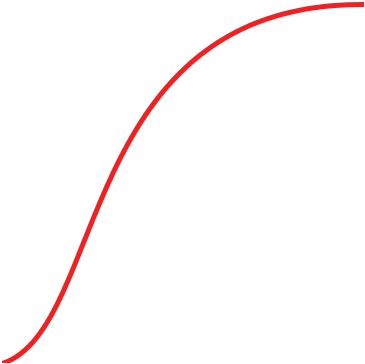
Existem várias palavras-chave que pode ser usada como um atalho para os movimentos de transição comum (ver quadro 2-2).

**A Tabela 2-2.***Palavra-chave alternativas para funções de distribuição Cubic-Bezier Comum*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Palavra-chave** | **Gráfico** | | | | | | | | | | | **Cubic-Bezier** | **Descrição** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Linear |  |  |  |  |  |  |  |  |  |  |  | 0, 0, 1, 1 | Arranque e paragem instantânea; constante |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Através da faixa de velocidade |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Movimento. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



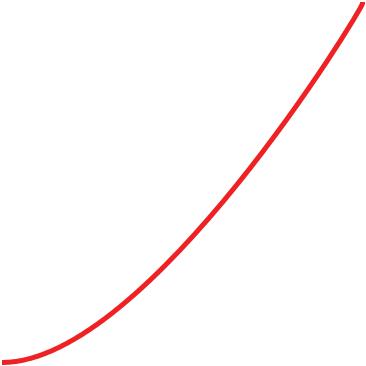
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Facilidade |  |  |  |  |  |  |  |  |  |  | 0,25, 0,1, 0,25, 1 Swift Start, acelerando rapidamente; |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | Transição lenta para parar no final. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



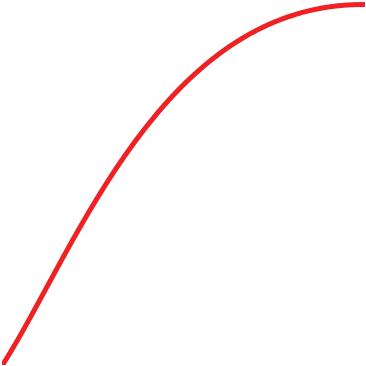
(*cont.*)

24

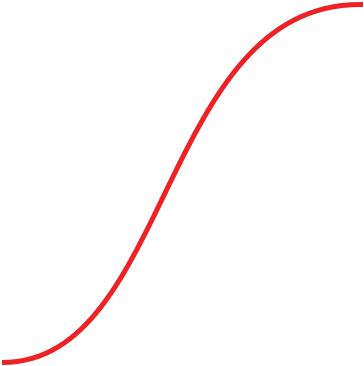
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Capítulo 2 ■ CSS3 transforma e transições |  |
| **A Tabela 2-2.**(*cont.*) | | | | | | | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Palavra-chave** | **Gráfico** | | | | | | | | | | | **Cubic-Bezier** | **Descrição** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Facilidade em |  |  |  |  |  |  |  |  |  |  |  | 0,42, 0, 1, 1 | Arranque lento, escalada de aceleração |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Para uma paragem súbita. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Facilidade para fora |  |  |  |  |  |  |  |  |  |  | 0, 0, 0,58, 1 | Instant começar a animação, motion |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Desacelera para fim. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

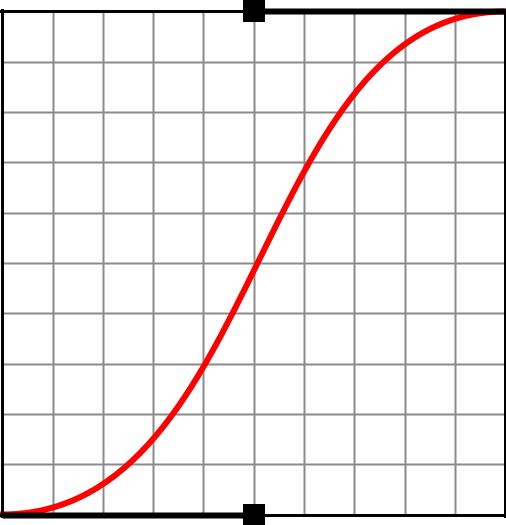


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Facilidade em out |  |  |  |  |  |  |  |  |  |  | 0,42, 0, 0,58, 1 | Elemento é facilitada para dentro e para fora |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Durante a animação: uma lenta e suave |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Iniciar brevemente atingir uma constante |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Velocidade durante o meio do |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Transição antes de abrandar para uma paragem. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |



Como você pode ver todas as curvas, facilitando a ter um modelo matemático equivalente sob forma de um metro cúbico de linhas vectoriais assim expressão: um número par em que cada conjunto de dígitos de ponto flutuante descreve um ponto no espaço de coordenadas, formando uma linha que cria uma curva de transição. Observe que os pontos de terminação no da curva no final não pode ser movido ou definido).

Para uma facilidade de entrada-saída curva, os pontos a aparência Figura 2-10.

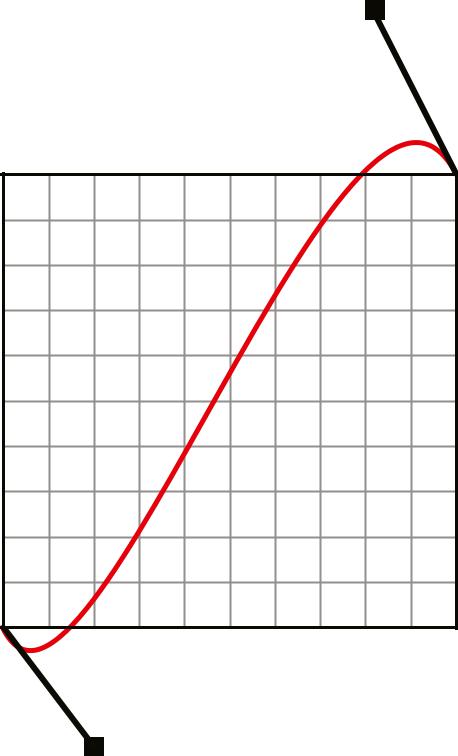


**Figura 2-10.***Curva de linhas vectoriais assim para uma facilidade no efeito de animação de saída*

Expressos em CSS, Figura 2-10 esta aparência:

Transição de função de distribuição: metros cúbicos linhas vectoriais assim(0,42, 0, 0,58, 1);

Compreender a função de linhas vectoriais assim cúbicos permite que você crie uma quase infinita variedade de curvas para o desagravamento personalizado CSS animações. É ainda possível dar os pontos os valores negativos ou valores maiores do que 1 para criar extreme facilitando curvas, que você pode ver na Figura 2-11.



**Figura 2-11.***Linhas vectoriais assim curva com negativo e maior que 1 valores, criando um "push-pull" do efeito de animação*

26

Capítulo 2 ■ CSS3 transforma e transições

Em CSS, Figura 2-11 esta aparência:

Transição: todos os metros cúbicos 2000ms linhas vectoriais assim(0,280, -0,315, 0.685, 1.390);

A utilização de tais valores cria animações com um "primavera" ou "saltar" para eles, também conhecido como *push-pull* animações. I vai explorar a utilização de tais animações no Capítulo 4.

■ **Dica**Ceaser (<http://matthewlein.com/ceaser/>) e (<http://cssglue.com/cubic>) Cúbicos são excelentes ferramentas para gerar o código de CSS facilitando graficamente manipulados metros cúbicos linhas vectoriais assim curvas. Ambos incluem um serviço de teste para permitir que você veja o resultado visual de alterações. Pedro Beverloo do recurso (<http://peter.sh/experiments/css3-transition-timing-functions/>) também é útil principalmente para a visualização de função degrau.

Animação em Passos

Também é possível para animar um elemento em etapas, em vez de como uma transição suave. (pense a súbita, movimento incremental da segunda mão em um relógio). Estas etapas são criados através da função e variantes. Aqui eu vou economizar o código mostrando apenas o código CSS3 para propriedades padrão, Firefox e Webkit.

Digamos que você tenha um h1 você deseja animar no mouseover (consulte a listagem 2-20).

**Listando 2-20.**Seqüência de transição em etapas para uma rubrica

<style> h1 {

Font-family: Futura; "Arial Black", Arial, sans-serif; alinhar texto: centro;

}

H1:foco {

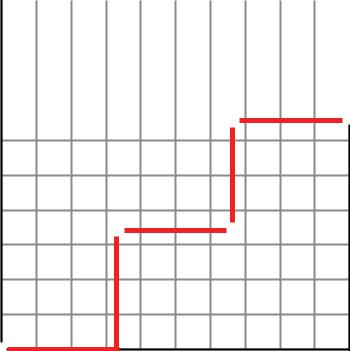
-moz-transição: 4s todos os passos(3, final); -webkit-transição: 4s todos os passos(3, final); transição: 4s todos os passos(3, final); -moz-transformar: translateX(400px); -webkit-transformar: translateX(400px); transformar: translateX(400px);

}

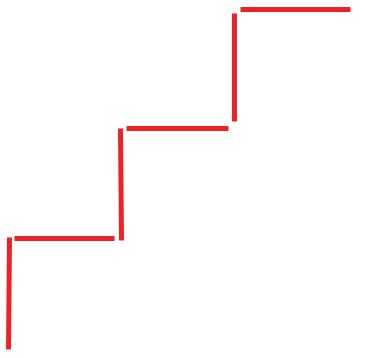
</Estilo>

O código na listagem 2-20 vai animar todos os elementos h1 em uma série de três "salta" durante quatro segundos após um atraso de dois segundos, com visível nenhum movimento entre cada etapa. Outras variações são possíveis, conforme mostrado na Tabela 3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Função** | **Gráfico** | | |  | **Descrição** |  |
|  |  |  |  |  |  |  |
| Passos(3) |  |  |  |  | Animação sobre x número de passos, (passos(3) |  |
|  |  |  |  |  |
|  |  |  |  |  | É ilustrado no gráfico). Pausa na partida. |  |
|  |  |  |  |  |
|  |  |  |  |  | Equivalente a passos(x, Final). |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Passos(3), |  |  |  |  |  |  |  |  |  |  |  |  |  |  | O elemento ainda estiver no início, em pausa no fim. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Passos(x), iniciar |  |  |  |  |  |  |  |  |  |  |  |  |  | Instant começar a animação, elemento está em pausa |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | No final. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Adicionando suporte para dispositivos móveis em transições CSS3

Até agora você só ativado transforma em :planar. Que é de longe o mais comum pseudo-classe com que para iniciar uma transição, mas não é o único, como você verá no Capítulo 3.

:o mouse pode colocar duas questões para navegadores instalados nos dispositivos móveis:

 O usuário do dedo podem obscurecer animações, particularmente em telas menores.

 Alguns dispositivos não suportam :o mouse (que faz sentido como, estritamente falando, todas as actuais plataformas móveis dependem de contacto directo). Em vez disso, eles cobrem a interação do usuário com simples

:foco.

Capítulo 2 ■ CSS3 transforma e transições

Se você optar por usar :pairar, você deve abranger a possibilidade de :foco apenas as plataformas móveis através de um selector agrupados como mostrado no código a seguir:

Img.incline:foco, img.incline:foco {

-moz-transformar: gire(7.5Deg); -o-transformar: gire(7.5Deg); -ms-transformar: gire(7.5Deg); -webkit-transformar: gire(7.5Deg); transformar: gire(7.5Deg);

}

Resumo

Neste capítulo você aprendeu a sintaxe para CSS3 transforma: escala, rodar, inclinar e traduzir, incluindo como inverter as imagens e transformações da colheitadeira. Eu também abrangeu o código para a forma mais simples de animação, CSS3 transições, mostrando a você como criar transições, como modificar a sua distribuição e atrasos e duas formas comuns de iniciá-las.

A circulação e distribuição de transições é comumente controladas através de linhas vectoriais assim curvas, embora também é possível utilizar a função de passos e atalhos de palavra-chave.

No próximo capítulo, vamos explorar como aplicar essas técnicas de animação para elementos de imagem.

**Capítulo 3**

**Transições CSS3 para imagens**

Os usos mais comuns para CSS3 transições em páginas da web são, primeiro, gerando efeitos visuais para interface de utilizador (UI) elementos (discutido no próximo capítulo) e, segundo, criando breve efeitos animados para imagens. Neste capítulo, você vai usar a sintaxe do módulo de transições para melhorar as imagens e legendas de sua animando-los. Estas técnicas demonstram fácil métodos para melhorar suas páginas da web visualmente, tornando o conteúdo da imagem e informações associadas mais interativo minimizando a tela "imóveis": uma consideração importante na idade de desenvolvimento da Web móvel.

Imagem de simples efeito de mixagem

A primeira transição você vai trabalhar irá demonstrar muitos dos conceitos fundamentais utilizados em exercícios posteriores: posicionamento de imagens que são exactamente o mesmo tamanho na parte superior de cada um dos outros e iniciar eventos no :o mouse (ver Figura 3-1).



**Figura 3-1.***Um efeito de cross-fade usando uma transição opacidade*

Existem algumas formas possíveis para alcançar o efeito mostrado na Figura 3-1:

 *Opção 1*: Especifique a primeira imagem como o CSS de fundo de um elemento de recipiente com a segunda imagem dentro do próprio elemento.

 *Opção 2*: Criar um recipiente o elemento com posição: parente que detém tanto imagens, com a segunda imagem tendo em posição: absoluta.

 *Opção 3*: Especificar ambas as imagens como plano de fundo e a transição entre eles.

Todas as três abordagens têm as suas vantagens e desvantagens. O primeiro e o terceiro métodos são potencialmente mais rápido para criar e mais responsiva, mas são menos acessíveis. Usando o primeiro método também significa que qualquer

31

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

Muda para as imagens terão de ser feitos em lugares diferentes, como uma imagem existirá apenas no CSS e o outro como um elemento HTML. A segunda opção pode envolver um pouco mais código, mas tem a vantagem de manter ambas as imagens como <img> elementos e assim permanece mais acessível. A terceira opção é tecnicamente fora da spec em que este artigo foi escrito, mas tem talvez a maior facilidade de uso.

Vou demonstrar todos os três métodos usando fotografias fornecidas pelo Ton Rulkens ([www.flickr.com/photos/47108884@N07/4595559479/](http://www.flickr.com/photos/47108884@N07/4595559479/)) e Peter Shanks ([www.flickr.com/photos/botheredbybees/1954163161/](http://www.flickr.com/photos/botheredbybees/1954163161/)), usado com permissão.

Ambas as imagens tem de ser exactamente o mesmo tamanho. Existem várias maneiras de alcançar esse objectivo:

 A cultura das imagens para a mesma dimensão em um aplicativo como o Adobe PhotoShop (esta é a solução mais óbvia).

 Você pode modificar a largura e altura das imagens através do CSS ou atributos HTML, embora isso muitas vezes leva à distorção visual.

 Você pode definir uma largura e altura sobre a div e usar oculto para guarnição de estouro:desligar porções das imagens que caiam fora desta área.

 Se ambas as imagens são representados como elementos do código, você pode usar o mesmo CSS clip valores para cada.

Opção 1: Primeira imagem como fundo de uma CSS

O HTML para esta opção é muito simples, como mostrado na lista 3-1.

**Lista 3-1.**HTML a opção 1 para criar duas imagens em camadas

<div class=mixagem>

<img src=jatropha híbrida.jpg alt="Jatropha hybrid folha">

</div>

■ **Nota**HTML5 amostras de código mostrado neste livro estão em "minified" sintaxe para poupar espaço. Os elementos são fechadas somente quando necessário e valores de atributo são citada somente se contiverem espaços.

O CSS mostrado na lista 3-2 também é muito simples.

**Lista 3-2.**CSS para criar um efeito de cross-fade para opção de HTML 1

Div.mixagem { background: url(leaf-veias.jpg); fundo-size: tampa; } div.mixagem div.mixagem img { Largura: 418px; altura: 500px; } div.mixagem img { transição: 3s opacidade facilidade; }

Div.mixagem img:foco { opacidade: 0; }

Opção 2: ambas as imagens como elementos HTML

Alternativamente, você pode empilhar ambas as imagens dentro de um único recipiente como imagens individuais; o HTML é mostrado na lista 3-3

**Lista 3-3.**Opção de HTML 2 para criar imagens em camadas

<div class=mixagem>

<img src=leaf-veias.jpg alt="Red de pasta salpicada folha"> <img src=jatropha.jpg alt="Jatropha hybrid folha">

</div>

Em vez de complicar o seu HTML potencialmente acoplando uma classe para a segunda imagem, você vai usar um enésimo-filho pseudo-selector para alterá-lo, como mostrado na lista 3-4.

**Lista 3-4.**CSS para criar um efeito de cross-fade para opção de HTML 2

Div.mixagem { posição: relativos; }

Div.mixagem div.mixagem img { Largura: 418px; altura: 500px; } div.mixagem img:nth-Child(2) { posição:absoluta; left:0; top:0; transição: 3s opacidade facilidade; }

Div.mixagem img:nth-Child(2):foco { opacidade: 0; }

Opção 3: ambas as imagens como plano de fundo

Enquanto a maneira mais fácil de código, esta opção é também a mais ousada: é fora da especificação atual (e, quando este artigo foi escrito, suportado apenas em crómio). Neste caso, a contendo div é totalmente desprovida de conteúdo e tudo o que é conseguido através do CSS, como mostrado na lista 3-5.

**Lista 3-5.**CSS para criar uma imagem de efeito cross-fade (Opção 3)

Div.mixagem { Largura: 418px; altura: 500px; transição: 3s de imagem de fundo a facilidade; imagem de fundo: url(leaf-veias.jpg); }

Div.mixagem:foco { background-image: url(jatropha.jpg); }

Se estavam dispostos a empurrar ainda mais seu código, uma abordagem alternativa seria usar a mesma div vazio com o CSS4 propriedade cross-fade, mostrado na lista 3-6.

**Lista 3-6.**CSS4 cross-fade usado para criar uma transição da imagem

Div.mixagem { Largura: 418px; altura: 500px;

Imagem de fundo: -webkit-cross-fade(url(jatropha.jpg), url(leaf-veias.jpg),0); transição: fundo 2s-imagem linear; }

Div.mixagem:foco { background-image: -webkit-cross-fade(url(jatropha.jpg), url(leaf-veias.jpg),100); }

Tal como o seu nome indica, cross-fade é uma abordagem mais eficiente, mas com aplicação prática limitada no tempo; em vez de fingir um efeito de dissolver com a transição de opacidade, cross-fade processa as imagens com o algoritmo apropriado.

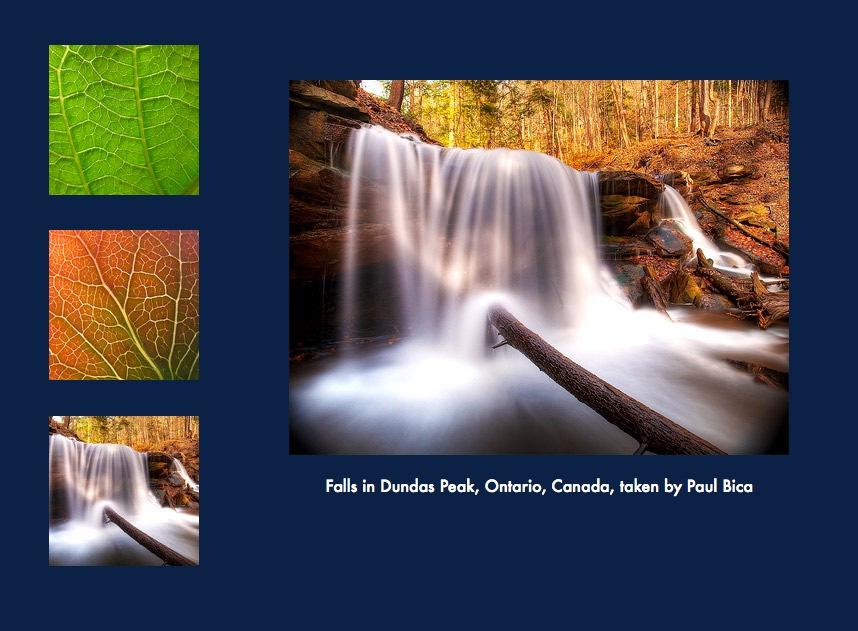
**CSS4? O "'Tás sobre WILLIS?**

Como CSS3 está se tornando popular em navegadores, a atenção do W3C foi movida para a próxima fase do desenvolvimento do CSS, que inclui a nova aparência seletores e regras para imagens e gradientes. Suporte do navegador (como deste escrito) é limitada e experimental, mas crescente. Mais relevantes para esta seção é o CSS4 Valores de imagem e Substituído o módulo de conteúdo, do qual você pode ler uma visão geral em <http://dev.w3.org/csswg/css4-images/>. Há também um módulo para fundos e bordas (<http://dev.w3.org/csswg/css4-background/>) bem como texto (<http://dev.w3.org/csswg/css4-text/>).

É quase inevitável que em futuro próximo o termo "CSS4" vai se tornar tão amplamente incompreendidos e utilizada abusivamente como "CSS3" é hoje, como discutido no Capítulo 1.

Uma simples Galeria de Imagens melhoradas com CSS3

Para o segundo exemplo, você vai criar uma imagem miniatura galeria com HTML e melhorar a apresentação das imagens grandes na galeria com transições CSS3 (consulte a Figura 3-2).



**Figura 3-2.***Uma simples galeria de imagens*

Você precisará de pelo menos três pares de imagens. Cada par será composto por uma imagem em miniatura e uma versão de tamanho integral da mesma imagem. O grande versão podem ser independentemente do tamanho que desejar, desde que é razoável; eu sugiro fazer as miniaturas cerca de 150 x 150 pixels em tamanho. Para manter a sua organização do ficheiro claro, seguir uma convenção de nomenclatura. Por exemplo, se a imagem em tamanho completo é x,jpg, nome a miniatura emparelhado x\_polegar.jpg, ambos armazenados em uma pasta de imagens.

A marcação de HTML

Seu objetivo é manter o HTML usado na galeria como limpo e simples possível. Para que fim, você usar uma lista de definição como a base para seu markup. A lista de definição é feito para conter pares de elementos: um (a definição prazo) para sua miniatura e o outro (a definição de declaração) para a imagem grande que ela corresponda. Montar o seu conteúdo no markup em relação às suas pastas, o HTML para a página terá

Algo como Lista 3-7 (usando imagens em destaque no exercício anterior, com um adicional de fotografia por Paulo Bica ([www.flickr.com/photos/dexxus/4137841698/](http://www.flickr.com/photos/dexxus/4137841698/)).

34

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Lista 3-7.**HTML5 para uma simples Galeria de Imagens

<!DOCTYPE HTML> <html lang=pt> <cabeça>

<meta Charset=utf-8> <título> Galeria CSS3</title> <style>

Corpo { background: #234; }

</Estilo>

</cabeça>

<body>

<dl id=gallery>

<dt><img src=jatropha\_polegar.jpg alt="Jatropha Leaf miniatura"> <dd><img src=jatropha.jpg alt="Jatropha Leaf Grande">

<dt><img src=leaf-veias\_polegar.jpg alt="Nervuras miniatura"> <dd><img src=leaf-veias.jpg alt="Nervuras">

<dt><img src=cascada\_polegar.jpg alt="Cascada miniatura"> <dd><img src=cascada.jpg alt="Cascada Grande">

</dl>

</body>

</html>

O CSS Inicial

O CSS utilizado aqui deve ser bastante auto-explicativa: você está na lista de definições de posicionamento em relação a fim de que o grande, absolutamente posicionado as imagens são organizadas em relação à lista, não o corpo do documento. Posicionando a imagens grandes absolutamente também significa que você pode empilhar-los exatamente no mesmo ponto e que o resto do documento - incluindo as miniaturas-irá agir como se as imagens de tamanho completo não estão lá. Por último, você ocultar a grandes imagens com visibilidade: oculto. e revele-os novamente usando um :passe o seletor de dt elementos com um combinator adjacentes, como mostrado na lista 3-8.

**Lista 3-8.**CSS para um simples Galeria de Imagens

Dl#gallery { posição: relativos; }

Dl#gallery dt img { Largura: 150px; altura: 150px; margin: 2.2em; }

Dl#gallery dd { posição: absoluta; esquerda: 200px; superior: 2.2em; visibilidade: hidden; } dl#gallery dt:foco + dd { visibilidade: visível; }

Movendo o mouse sobre a miniatura da imagem faz com que as imagens grandes é emparelhado com na definição lista para aparecer imediatamente. No entanto, há um problema de interface que você precisa resolver antes de você pode trazer a animação para a Gallery.

Melhorar a galeria

O que você fez até agora funciona, mas é um pouco barulhentos: você notará que colocando o mouse para a direita da imagem em miniatura torna o associado imagem grande aparecem imediatamente. Você vai fixar a ambas estas questões com mais algumas linhas de CSS, mudando de ocultar as imagens grandes com visibilidade (que não pode ser animado)

A opacidade que pode) e ao mesmo tempo compartilhar o tamanho da definição de termos com as imagens em miniatura, como mostrado na lista 3-9.

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Lista 3-9.**CSS para uma Galeria de imagem aprimorada com transições gerenciadas

Dl#gallery dd { posição: absoluta; esquerda: 200px; superior: 2.2em; opacidade: 0; transição: .85s Opacidade linear; }

Dl#gallery dt:foco + dd { opacidade: 1; }

Agora as imagens grandes fade in suavemente à medida que você move o mouse sobre as imagens em miniatura.

Adição de legendas

É muito útil para os usuários se eles podem ler legendas de imagem. Que não requer a adição de quaisquer tags extras neste caso, apenas a legenda conteúdo e CSS. Eu vou modificar um dos <dt ><dd> pares como um exemplo, mostrado na lista 3-10.

**Listando 3-10.**Exemplo de um HTML simples imagem de Closed Caption

<dt><img src=jatropha\_polegar.jpg alt="Jatropha Leaf miniatura">

<dd><img src=jatropha.jpg alt="Jatropha Leaf Grande"> um close-up Foto de um híbrido de Jatropha

O CSS também deve mudar em resposta (consulte a listagem 3-11).

**Listando 3-11.**CSS para uma simples imagem de Closed Caption

Dl#gallery dd { posição: absoluta; esquerda: 200px; superior: 2.2em; opacidade: 0; alinhar texto: centro; font-family: Futura, Arial, sans-serif; color: branco; transição: .85s Opacidade linear; }

Dl#gallery dd img { Display: bloco; margin: auto; padding-bottom: 1.2em; }

Neste caso o fade in legendas com as imagens. Também é comum para animar as legendas separadamente a partir de imagens que você vai fazer no próximo exercício.

Alterar o início do evento

Enquanto :O mouse funciona para iniciar o esmaecimento da grande imagem na galeria, pode parecer mais natural para os usuários a *clicar* na miniatura. Lá, você tem algo de um problema: não há equivalente direto para o JavaScript onclick manipulador de eventos em CSS. No entanto, existem poucas alternativas neste caso.

:ativo

Enquanto ele é mais fortemente associada com links, é possível usar o :ativo pseudo-classe para iniciar a transição, como mostrado na listagem 3-12.

**Listando 3-12.**CSS para um efeito sobre o mouse para baixo

Dl#gallery dt:activo + dd { opacidade: 1; }

Você verá a principal desvantagem desta abordagem imediatamente: a imagem grande aparece somente enquanto o botão do mouse pressionado sobre a miniatura.

:alvo

O uso de :alvo pode ser a forma mais eficaz de pseudo-selector na presente instância, embora exija algum além de seu markup. :target deriva da utilização da tradicional "âncoras" valores de id, detecção de cliques em elementos ligados a esses ids.

O seu HTML alterações ao mostrado na listagem 3-13.

**Listando 3-13.**HTML para uma Galeria de Imagens com :alvo

<dl id=gallery>

<dt><a href=#jatropha><img src=jatropha\_polegar.jpg alt="Jatropha Leaf miniatura"></a> <dd id=jatropha><img src=jatropha.jpg alt="Jatropha Leaf Grande">

Um close-up Foto de um híbrido de Jatropha leaf

<dt><a href=#veias><img src=leaf-veias\_polegar.jpg alt="Nervuras miniatura"></a> <dd id=veias><img src=leaf-veias.jpg alt="Nervuras">

Close-up photgraph de nervuras

<dt><a href=#cascada><img src=cascada\_polegar.jpg alt="Cascada miniatura"></a> <dd id=cascada><img src=cascada.jpg alt="Cascada Grande">

Cai em Dundas Pico, Ontario, Canadá </dl>

Em circunstâncias normais um link para uma âncora obrigaria a página para ir para a localização do id com um visual saltar; que é evitada neste caso porque o dd elementos são absolutamente posicionado perto do topo da página.

Apenas uma linha de CSS requer a modificação. O seu cursor muda como declaração :mostrado na listagem 3-14.

**Listando 3-14.**CSS para uma Galeria de Imagens com :alvo

Dd#jatropha:alvo, dd#veias:alvo, dd#cascada:alvo { opacidade: 1; }

Esta abordagem tem igualmente a vantagem de modificar a URL, o que significa que os usuários podem ser levados a certas imagens através do uso de links com apenso ids. Por exemplo, [www.yourdomain.com/gallery.html#cascada](http://www.yourdomain.com/gallery.html#cascada) poderia trazer até a última imagem na galeria automaticamente.

Pop-up simples legendas de imagem

É possível também exibir a legenda de uma imagem em mouseover, quer em uma galeria ou como parte de uma interface do utilizador. O ideal para esta é uma marcação de uma lista de definição, como no exemplo anterior, ou uma <figura> e <figcaption>, que são os elementos que você vai usar para este próximo exemplo, como mostrado na Figura 3-3 e listando 3-15. (ESTOU usando fotografias por Bradley Davis ([www.flickr.com/photos/backpackphotography/244716694/](http://www.flickr.com/photos/backpackphotography/244716694/)) e Wolfgang Staudt ([www.flickr.com/photos/wolfgangstaudt/](http://www.flickr.com/photos/wolfgangstaudt/)). Se você decidir usar seu próprio imagens, certifique-se de que eles são de dimensão semelhante.)



**Figura 3-3.***Legendas de imagem animada*

Neste caso, as legendas são descrições da imagem, mas essa técnica também pode ser usada para a navegação gráfica para um local (ver capítulo 4).

**Listando 3-15.**HTML5 Figura e Código Figcaption

<figura>

<img src=demónios-torre.jpg alt="uma foto da torre do Diabo, inWyoming, EUA"> <figcaption> Torre do Diabo, Wyoming, EUA</figcaption>

</figura>

<figura style=left:550px>

<img src=sunrise-ponto.jpg alt="uma foto do nascer do ponto, o Parque Nacional Bryce, Utah, EUA">

<figcaption> o Parque Nacional Bryce, Utah, EUA</figcaption> </figura>

Você vai fazer algumas coisas no CSS. Em primeiro lugar, as imagens e o elemento figura circundantes deve ser definido com o mesmo tamanho e flutuavam lado-a-lado. Você também as legendas de estilo antes de ocultá-los com mais CSS.

**ReM: não a banda**

Tradicionalmente, fontes em páginas web têm sido médias em pixels, percentagens, ou ems. Os dois últimos têm sido geralmente a abordagem preferida porque eles são inerentemente expansíveis. Desde o 1em é literalmente a largura do  caractere M, tornando o texto do parágrafo 20% maior foi tão fácil como informando que o font-size para o  seletor de p foi 1.2em. Usando em também facilita a tamanho proporcionalmente o espaço entre elementos como tamanhos de fonte aumentar e diminuir. Por exemplo, você definir a medianiz entre uma imagem e o texto que o rodeiam medindo a margem da imagem em  unidades, criando uma relação visual entre a cópia do corpo e ilustrações.

Um problema com o uso de  um sistema de medida em como em páginas da web é que ela compostos, devido ao fato de que a unidade está sempre definido em relação ao tamanho da fonte do elemento pai. Definição li elementos a 1.2em em tamanho é fino até que você aninhar uma lista dentro dele: o conteúdo do interior do li será proferida no 1.4em em tamanho.

Rem (para raiz em) recebe em torno deste problema através da medição em si em relação ao elemento raiz - que é o elemento html. Isso significa que você pode declarar um único tamanho do tipo de letra no seletor de html e escala tudo em relação ao que, como mostrado na listagem 3-16.

**Listando 3-16.**Rem Font Dimensionamento para um documento

Html { font-size: 62,5% } body { font-size: 1.4Rem; } h1 { font-size: 2.4Rem; }

Isto também se traduz bem para pixels: no stylesheet acima, corpo de texto na página será equivalente a 14px em tamanho e elementos de h1 prestados a 24 pixels.

O suporte do navegador para rem é surpreendentemente bom: todos os recentes navegadores que suportam CSS transforma (Safari 5+, Chrome Firefox 3.6+, IE9+, e Opera 11,6+) também apoiar a unidade de rem).

Vejamos agora a base CSS para uma figura e legenda (consulte a listagem 3-17).

**Listando 3-17.**Base CSS para uma figura e legenda

Figura { flutuação: esquerda; }

Figura, figura img { Largura: 500px; altura: 333px; }

Figcaption {

Font-family: Baskerville, "Baskerville Old Face", Garamond, "Times New Roman", serif; font-style: italic; fundo: rgba(0,0,0,0.4);

Font-size: 2rem; padding: 0.8Rem; color: #fff;

}

Nesta fase, a página será algo como Figura 3-4.



**Figura 3-4.***Imagens com legendas posicionado, antes de estarem ocultos*

Agora que eles estão decorados, você ocultar legendas utilizando transbordamento: escondida na figura elemento. Ao mesmo tempo que você precisa para posicionar a legenda. Para as imagens neste exemplo, trazendo as legendas para baixo a partir do topo será provavelmente olhar o melhor. Existem alguns métodos possíveis para a localização de legendas. Vou usar o posicionamento relativo e um valor para a parte inferior que é ligeiramente maior do que a altura da imagem mais a altura da legenda (consulte a listagem 3-18).

■ **dica**é uma boa ideia verificar que a deslocalização de um elemento funciona antes de adicionar nas transições.

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Listando 3-18.**CSS para ocultar e a posição de uma legenda

Figura { flutuação: esquerda; }

Figura, figura img { Largura: 500px; altura: 333px; transbordamento: hidden figcaption; } {

Font-family: Baskerville, Garamond, "Times New Roman", serif; font-style: italic; fundo: rgba(0,0,0,0.4); font-size: 2rem; padding: 0.8Rem; color: #fff; POSIÇÃO: relativos; parte inferior: 400px;

}

Figura:foco figcaption { parte inferior: 340px; }

Por último, você vai adicionar na transição da legenda, como mostrado na listagem 3-19.

**Listando 3-19.**CSS para a transição de uma legenda

Figcaption {

Font-family: Baskerville, Garamond, "Times New Roman", serif; font-style: italic; fundo: rgba(0,0,0,0.4);

Font-size: 2rem; padding: 0.8Rem; color: #fff; POSIÇÃO: relativos; parte inferior: 400px; transição: 2s todos; }

■ **Nota**guardo nossas imagens acessíveis usando nomes descritivos e os valores de atributo alt. Isto é muito importante: sempre lembrar que nem todos serão capazes de ver ou interagir com seus projetos.

Pilha do cartão de imagem e ventilador revelar

Como páginas da web se tornam mais complexos, eles se tornam mais difícil resumir e ilustrar. Se você tiver criado uma extensa página da Gallery, pode ser muito difícil escolher apenas uma imagem para o interesse dos visitantes faísca. Uma solução possível é a utilização de várias imagens exibidas em um *controle deslizante keyframed* gallery (como mostrado no Capítulo 5) ou visor interativo. Neste caso, vou visivelmente pilha várias fotografias na parte superior de cada um dos outros, revelando-los no mouseover para criar um maior interesse e compreensão do conteúdo vinculado (veja a Figura 5).

**Figura 3-5.***Uma animada efeito do ventilador da placa*

Novamente, você vai usar imagens do mesmo tamanho para produzir os melhores resultados, como mostrado na listagem 3-20.

**Listando 3-20.**HTML para uma imagem animada Efeito do ventilador da placa

<div id=cardfan>

<img src=bike.jpg alt="uma fotografia de uma bicicleta estacionada em italiano rua"> <img src=florence.jpg alt="uma fotografia da ponte em Florença, Itália">

<img src=roma.jpg alt="uma fotografia de um aqueduto ruínas fora de Roma">

</div>

Você enrolado as imagens em um recipiente elemento para tornar mais fácil a referência e o controlo através de CSS. O div e as imagens que ele contém são do mesmo tamanho. Você também centro-alinhe o recipiente para a página durante a modelagem e empilhamento as imagens no interior (consulte a listagem 3-21).

**Listando 3-21.**Base CSS para um efeito do ventilador da placa

#cardfan, #cardfan img { Largura: 640px; altura: 480px; } #cardfan { margin: 0 auto; }

#cardfan img { border: 32px solid #ffe; caixa-sombra: 12px 12px 10px rgba(0, 0, 0, 0,2); posição: absoluta; }

Você deseja que as imagens Fan out quando o usuário passa o mouse sobre eles; você pode alcançar esta girando o *primeiro* e o *último* imagens na pilha de 12 graus através do uso do primeiro filho e ::última criança pseudo-classes, como mostrado na listagem 3**-**22.

**Listando 3-22.**Rotação do primeiro e do último imagens em um cartão de efeito de pilha

#cardfan:foco img:primeira criança { transformar: gire(12graus);

}

#cardfan:foco img:última criança { transformar: gire(-12graus);

}

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

O código na listagem 3**-**22 irá produzir a imagem mostrada na Figura 3**-**6.



**Figura 3-6.***Imagens empilhadas girado com CSS transforma em torno de seus respectivos centros*

Como você pode ver, a imagem gira em torno de seu centro. Neste caso, você deseja que as imagens a serem ligeiramente, assim você desmembrados necessidade de mover o eixo de transformação *abaixo* as imagens (consulte a listagem 3**-**23).

**Listando 3-23.**Compensando a transformar a origem para imagens

#cardfan img { border: 32px solid #ffe; caixa-sombra: 12px 12px 10px rgba(0, 0, 0, 0,2); posição: absoluta;

Transformar o centro de origem: 600px;

}

O código na listagem 3**-**23 irá produzir a saída mostrada na Figura 3**-**7.

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens



**Figura 3-7.***Imagens empilhadas girado ao redor de um desvio origem, realçado com mira*

Agora você pode realmente as placas do ventilador, aplicando uma transição ao mesmo tempo, como mostrado na listagem 3**-**24. (Nota que eu já alterou alguns valores para aumentar o efeito do ventilador).

**Listando 3-24.**Full CSS para uma animada Galeria de ventilador da placa

#cardfan, #cardfan img { Largura: 640px; altura: 480px; transição: .6s transformar facilidade;

}

#cardfan { margin: 0 auto; }

#cardfan img { border: 32px solid #ffe; caixa-sombra: 12px 12px 10px rgba(0, 0, 0, 0,2); posição: absoluta;

Transformar o centro de origem: 1200px;

}

Div#cardfan:foco img:primeira criança { transformar: gire(24 deg);

}

Div#cardfan:foco img:última criança { transformar: gire(-24graus);

}

Você também pode associar a galeria de imagens para a página, como mostrado na listagem 3**-**25.

43

N

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Listando 3-25.**Um associado do ventilador da placa Gallery

<div id=cardfan>

<a href=gallery.html>

<img src=bike.jpg alt="uma fotografia de uma bicicleta estacionada em italiano rua"> <img src=florence.jpg alt="uma fotografia da ponte em Florença, Itália">

<img src=roma.jpg alt="uma fotografia de um aqueduto ruínas fora de Roma">

</a>

</div>

Se você deseja girar a *primeira e segunda* imagens e deixar a fotografia na parte superior inalterado, seletor de alterar o seguinte:

Div#cardfan:foco img:Last-filho

A

Div#cardfan:foco img:nth-criança(2n)

Você também pode girar a posição padrão das imagens levemente para permitir que o usuário algum entendimento do que está para vir, como mostrado na listagem 3**-**26.

**Listando 3-26.**Um ventilador de placa de indefinido visualmente Gallery

#cardfan img:primeira criança { transformar: gire(6 graus);

}

#cardfan img:nth-criança(2n) { transformar: gire(-6ºC);

}

Você também pode associar as fotografias individualmente para a Gallery exemplo no início deste capítulo (consulte a listagem 3**-**27).

**Listando 3-27.**Um ventilador de placa de galeria com imagens vinculadas individualmente

<div id=cardfan>

<a href="gallery.html#bike">

<img src=bike.jpg alt="uma fotografia de uma bicicleta estacionada em italiano rua">

</a>

<a href="gallery.html#Florence">

<img src=florence.jpg alt=="uma fotografia da ponte em Florença, Itália">

</a>

<a href="gallery.html#aqueduto">

<img src=roma.jpg alt="uma fotografia de um aqueduto ruínas fora de Roma">

</a>

</div>

No entanto, se você associar as imagens individualmente você precisará alterar o CSS como a organização de seu markup mudou. Em vez de referenciar eles como filhos, vais referência as imagens através do valor do seu atributo src (consulte a listagem 3**-**28).

44

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Listando 3-28.**Uma galeria do ventilador da placa com as imagens referenciadas por seu atributo src

#cardfan img[src="bike.jpg"] { transformar: gire(6 graus);

}

#cardfan img[src="roma.jpg"] { transformar: gire(-6ºC);

}

#cardfan:foco img[src="bike.jpg"] { transformar: gire(24 deg);

}

#cardfan:foco img[src="roma.jpg"] { transformar: gire(-24graus);

}

Nota que a técnica na cotação de 3**-**26 é menos flexível do que criar e fazendo referência a valores de id para suas imagens: se você alterar os nomes por qualquer motivo, você terá que alterar o seu CSS em resposta.

Você também pode levantar as imagens individuais para o primeiro plano quando pairou sobre (consulte a listagem 3**-**29).

**Listando 3-29.**Uma galeria de um ventilador de placa de imagem elevada para o primeiro plano

Img[src="bike.jpg"]:foco, img[src="florence.jpg"]:foco { z-index: 2; }

Você poderia ter escrito menos markup (mas ligeiramente mais CSS) utilizando ::antes e ::depois de pseudo-seletores de elemento para gerar as outras imagens. (Observe que você não pode usar o conteúdo gerado em <img> tags, como eles são

Uma forma de elemento substituído). Também é importante observar que esta abordagem praticamente mata a acessibilidade, como a maioria dos leitores de tela não actualmente conteúdo gerado pelo acesso, nem pode JavaScript obter acesso a ele. A técnica mostrada aqui é mencionado como uma possibilidade interessante, não como um método de produção de conteúdo em um site central.

■ **Nota**Troca elementos são elementos HTML que têm um intrínseco largura e altura, sem beneficiar do CSS; que é, qualquer etiqueta que produz um espaço reservado que tem então o seu conteúdo substituído por um recurso externo. Por exemplo, quando você usar o <tipo de entrada=texto>, é exibida uma caixa de texto em um tamanho que é adequado para entrada de texto de linha única. Isso não significa que você não pode aplicar o CSS para redimensionar, apenas que o navegador substitui a instâncias da entrada> tag <com elementos de um tamanho predeterminado por padrão. O mesmo acontece com <img>; sem CSS, as imagens são carregadas para a página em sua largura e altura natural.

<br>, <img>, <video>, <iframe>e <object> são todos substituídos elementos, como são <Entrada>, <button>e <textrarea>.

Mais importante, no contexto deste exercício e como uma regra geral, conteúdo gerado não pode ser aplicada a troca elementos. Que é que você não pode usar ::antes ou ::depois em <img> ou as outras etiquetas listadas acima. Além disso, você pode transformar elementos inline a menos que eles também são substituídos elementos.

Tendo esta rota, seu código HTML e CSS simplifica para o que você vê na cotação de 3**-**30.

**Listando 3-30.**Código HTML e CSS para um efeito do ventilador da placa com conteúdo gerado

<div id=cardfan>

<img src=florence.jpg alt="uma fotografia da ponte em Florença, Itália">

</div>

#cardfan { posição: relativos; margin: 0 auto; }

#cardfan, #cardfan img, #cardfan img:antes de #cardfan img:após { Largura: 640px; altura: 480px;

}

#cardfan img, #cardfan:antes de #cardfan:após { fronteira: 32px solid #ffe;

Caixa-sombra: 12px 12px 10px rgba(0, 0, 0, 0,2); posição: absoluta;

Transformar o centro de origem: 1200px;

}

Div#cardfan::antes { conteúdo: url(bike.jpg); } #cardfan::após { conteúdo: url(roma.jpg); }

#cardfan::antes, div#cardfan:após { posição: absoluta; esquerda: 0; superior: 0; } #cardfan::antes { transformar: gire(6 graus); }

#cardfan::após { transformar: gire(-6ºC); }

O código na listagem 3**-**30 cria o mesmo efeito do ventilador da placa, com discutivelmente mais adaptabilidade; para alterar as fotografias de ambos os lados da imagem central, você só precisa modificar o seu CSS e não de seu markup. Esta abordagem também tem desvantagens significativas, como observado anteriormente.

Você pode animar estas imagens geradas pelo encadeamento de seletores de pseudo-, como mostrado na listagem 3**-**31.

**Listando 3-31.**O código CSS para animar o Conteúdo Gerado

Div#cardfan:Foco::antes { transformar: gire(24 deg);

}

Div#cardfan:Foco::após { transformar: gire(-24graus);

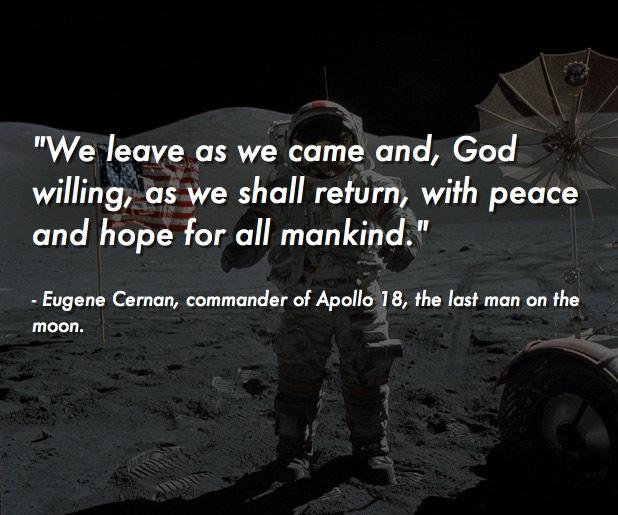
}

■ **Nota**Por que razão o pontos duplo no ::antes e ::depois? Uma das poucas alterações formais para selectores existente entre CSS2 e 3 foi o da W3C além de outra vírgula em selectores de conteúdo gerado, para distinguir a sua natureza diferente. Browsers modernos irá reconhecer e apoiar :: ou : antepostas seletores de conteúdo gerado. Neste livro, utilizei a nova versão formal; para uma maior compatibilidade com versões mais antigas de navegadores, talvez você queira usar apenas uma vírgula.

Claquete legendas de imagem com diferentes efeitos de entrada e de saída

Enquanto trazendo nas legendas de imagem a partir da parte superior ou inferior funciona para peças curtas de texto, as transições podem ser um pouco grandes e pesados quando há uma grande quantidade de texto envolvidos. Se o texto da legenda é muito longa, pode ser melhor "swing" no planar.

Você vai usar muito a mesma abordagem que você usou anteriormente: uma imagem de dimensões conhecidas (fornecido cortesia da NASA), usando o transbordamento: oculto para ocultar a legenda até que você revelar sobre o mouse (ver Figura 3**-**8).



**Figura 3-8.***Uma claquete animação legenda*

O HTML e CSS para a claquete legenda mostrada na Figura 3**-**8 é mostrado na listagem 3**-**32.

**Listando 3-32.**O código para criar uma legenda de claquete

<figura class=claquete>

<img src=apollo-17.jpg alt="Fotografia de astronauta na Lua"> <figcaption>

<p>Nós deixar como viemos e, se Deus quiser, como havemos de voltar, com paz e esperança...</p>

- Eugene Cerna, comandante do Apollo 18, o último homem na lua. </figcaption>

</figura>

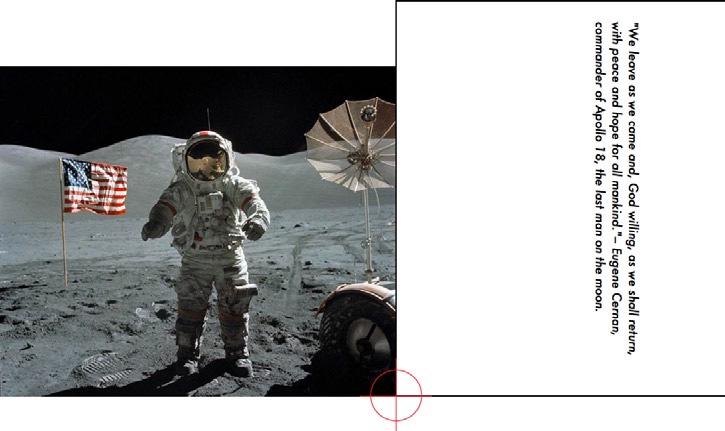
Figura.claquete { posição: relativos; }

Figura.claquete figcaption { posição: absoluta; superior: 0; esquerda: 0; padding: 2rem; } figura.claquete, figura.claquete figcaption { width:618px;altura:515px; } figura.claquete figcaption q { font-size: 2rem; exibição: bloco; margin-bottom: 2rem; }

Você ainda não oculta a legenda , o que torna mais fácil a posição: você precisará girar o elemento figcaption 90 graus, de modo que o seu canto inferior direito corresponde ao canto da imagem (ver Figura 3**-**9).

47

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens



**Figura 3-9.***Figura e girado Legenda com relocados transformar origem realçado*

O código para a posição da legenda na Figura 3**-**9 pode ser visto na listagem 3**-**33. (Note que usei o CSS3 de dimensionamento de caixa de propriedade para garantir que o figcaption permanece o direito a largura e a altura após o enchimento é adicionada).

**Listando 3-33.**O código CSS para posicionar uma claquete Legenda

Figura.claquete figcaption { font-family: Futura, Arial, sans-serif; font-weight: 100; font-style: italic; COLOR: black;

Caixa de calibragem: caixa de fronteira;

Font-size: 1.2Rem; padding: 2rem; padding-top: 8rem; fronteira: 2px preto sólido;

Transformar a origem: inferior direita; transformar: gire(90graus);

}

Agora que a legenda está no local correto, você pode desativar a fronteira, defina o transbordamento para oculto, e inverter as cores (consulte a listagem 3**-**34).

**Listando 3-34.**O código CSS para posicionar uma claquete Legenda

Figura.claquete, figura.claquete figcaption { width:618px;altura:515px;

Transbordamento: oculto;

}

Figura.claquete figcaption { font-family: Futura, Arial, sans-serif;

Font-weight: 100; font-style: italic; cor: branco; font-size: 1.2Rem; padding: 2rem; padding-top: 8rem; caixa de calibragem: caixa de fronteira;

Fundamento: rgba(0,0,0,0);

48

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

Transformar a origem: inferior direita; transformar: gire(90graus);

}

Figura.claquete:foco figcaption { transformar: gire(0);

}

Para tornar essa transição de uma animada, adicione o seguinte ao da figura.claquete figcaption declaração: transição: transformar 2s-cúbicos linhas vectoriais assim(.12,.49,.17,.87);.

(Observe que você está restringindo a transição para o rastreamento apenas a transformação utilizando a propriedade apropriada: este é tanto mais eficiente e torna muito mais fácil a via código quando você retornar a ele em seis meses . . . Embora evitando transições inesperado só porque você mudou algo entre o padrão e girado membros da legenda).

Enquanto isso funciona, ainda existem algumas questões. Por exemplo, o texto realmente não é claro o suficiente para ser lido contra o astronauta e o pano de fundo da lua, assim você deseja adicionar algum texto-sombra e o fundo para o figcaption.

**Listando 3-35.**Código melhorado para uma claquete Transição Legenda

Figura.claquete figcaption { font-family: Futura, Arial, sans-serif;

Font-weight: 100; font-style: italic; cor: branco; font-size: 1.2Rem; padding: 2rem; padding-top: 8rem; caixa de calibragem: caixa de fronteira;

Fundamento: rgba(0,0,0,0.6);

Texto-sombra: 3px 3px 1px rgba(0,0,0,0.6); transformar origem: inferior direita; transformar: gire(90graus);

Transição: transformar 2s-cúbicos linhas vectoriais assim(.12,.49,.17,.87);

}

Essa é uma melhoria significativa para a legibilidade do texto, mas a forma como a borda do fundo vem ainda parece um pouco estranho. Existem algumas soluções possíveis: um é para fazer a mais ampla, figcaption empurrando o texto à direita e a própria figcaption para a esquerda de modo a que a caixa vem para baixo em mais de um efeito visual "fatiar" (listando 3**-**36).

**Listando 3-36.**Código melhorado para uma claquete Transição Legenda

Figura.claquete figcaption { largura: 1236px; altura:515px;

Font-family: Futura, Arial, sans-serif; font-weight: 100; font-style: italic; cor: branco; font-size: 1.2Rem; padding:

Padding: 8 rem 2 rem 0 660px; caixa de calibragem: caixa de fronteira; fundo: rgba(0,0,0,0.6);

Texto-sombra: 3px 3px 1px rgba(0,0,0,0.6); transformar origem: inferior direita; transformar: gire(90graus);

Esquerda: -618px;

Transição: transformar 2s-cúbicos linhas vectoriais assim(.12,.49,.17,.87);

}

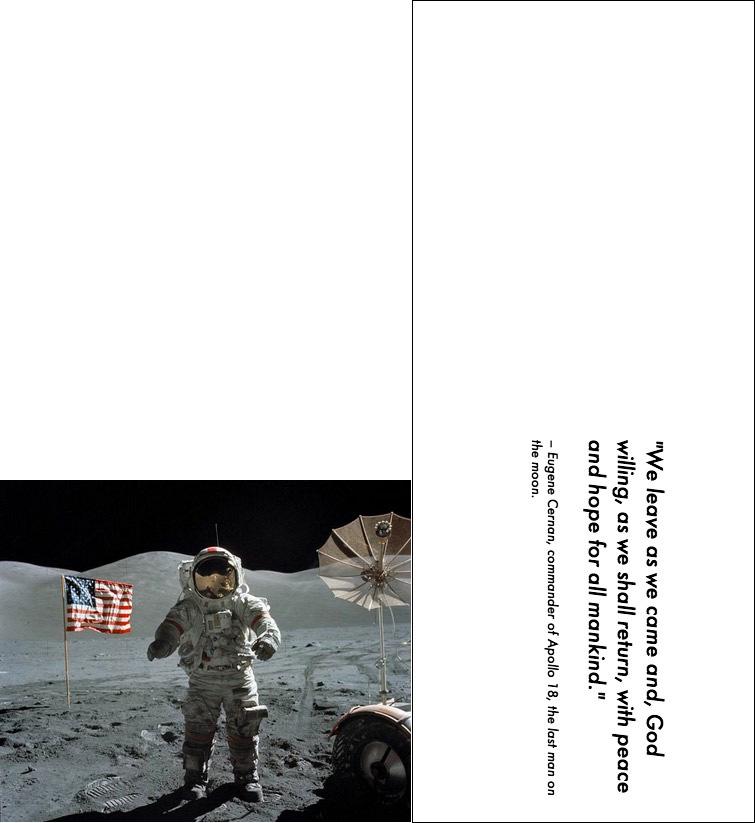
49

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

Neste caso, fiz o figcaption duas vezes tão grande como era antes e completadas o conteúdo em a partir do lado esquerdo de forma que ele chega a cobrir a mesma porção da imagem. Mudando a posição da caixa figcaption e a localização do elemento de transformação da origem você verá que é possível criar vários tipos diferentes de efeitos.

Criação de sequências específicas de transição

O outro método para a criação de uma legenda de transição está a desaparecer no fundo da legenda *após* o texto tenha sido girado no lugar. Isto requer a animação duas propriedades distintas e retardando um até depois de o primeiro ter concluído. A configuração do fundo da figcaption para completamente transparente por padrão pode iniciar este processo, seguido de separação a efeitos de transição com vírgulas e adicionar um atraso para um conjunto de valores. Consulte a Figura 3**-**10 para um exemplo.



**Figura 3-10.***Figcaption estendida para um melhor efeito de transição legenda delegar. (estouro: Hidden for desligado, as cores são invertidas e uma fronteira é adicionada para fins de ilustração.)*

50

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

Lista 3**-**37 apresenta todo o código CSS.

**Listando 3-37.**O código CSS para uma claquete Legenda

Figura.claquete { posição: relativos; transbordamento: hidden; }

Figura.claquete figcaption { posição: absoluta; superior: 0; esquerda: 0; padding: 2rem; } figura.claquete, figura.claquete figcaption { width:618px;altura:515px; } figura.claquete figcaption {

Font-family: Futura, Arial, sans-serif; font-weight: 100; font-style: italic; cor: branco; font-size: 1.2Rem; padding: 2rem; padding-top: 8rem; caixa de calibragem: caixa de fronteira; fundo: rgba(0,0,0,0);

Texto-sombra: 3px 3px 1px rgba(0,0,0,0.6); transformar origem: inferior direita; transformar: gire(90graus);

}

Figura.claquete:foco figcaption { transformar: gire(0);

Opacidade: 1;

Fundamento: rgba(0,0,0,0.6);

Transição: transformar 2s-cúbicos linhas vectoriais assim(.12,.49,.17,.87), fundo .9s linear 2.2s;

}

Figura.claquete figcaption q { font-size: 2rem; exibição: bloco; margin-bottom: 2rem; }

Você notará que o fundo agora desaparece em 200 milissegundos após o texto gira no lugar. O problema é que o restante legenda Desaparece instantaneamente quando o usuário move o mouse para fora da imagem, fazendo com que a mesma questão que você encontrou quando apresentei as transições no Capítulo 2. Mover o código de transição para o estado padrão para figcaption significará que a transição irá inverter a si no mouseout, que parecerá um pouco estranho, demasiado. O que você deseja, idealmente, é ter o figcaption desaparecem em uma maneira diferente do que apareceu.

Alterando a saída caso

Para obter esse efeito, você será animar três propriedades: opacidade, transformar e fundo. Você vai dividir a transição explicitamente em componentes separados para tornar mais fácil a via. A ordem em que você especificar propriedades deve ser o mesmo de cada vez para esse efeito ao trabalho.

Todavia, primeiro você irá adicionar a opacidade para a listagem de vários estados (3**-**38).

**Listando 3-38.**Melhorado o código CSS para uma claquete Legenda

Figura.claquete figcaption { opacidade: 0;

...

Figura.claquete:foco figcaption { opacidade: 1;

...

Agora isso não faz diferença, como a legenda é invisível no seu estado padrão devido ao transbordamento: escondida na figura elemento. Mas vai fazer a diferença mais tarde, como você vai ver em breve.

Em seguida, você vai quebrar a vários componentes de animação (listando 3**-**39).

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

**Listando 3-39.**O código CSS para uma claquete Legenda

Figura.claquete:foco figcaption { transformar: gire(0); opacidade: 1;

Propriedade de transição: Opacidade, Transformar, Fundo;

Transição de função de distribuição: Facilidade, Metros Cúbicos linhas vectoriais assim(.12,.49,.17,.87), linear;

Transição de duração: 0s, .9s, 1s;

Atraso de transição: 0s, 0s, 1s;

}

Ele ajuda a ler as alterações de transição como colunas, e é por isso que eu tenho adicionada a espaços desnecessários no código. Neste caso, opacidade não tem função de sincronização, sem duração e sem atraso (significando que produz efeito instantaneamente), considerando que o fundo tem uma função de distribuição linear, dura cerca de um segundo e tem um atraso de um segundo (o que significa que ele entra em vigor após o texto tiver girado no lugar).

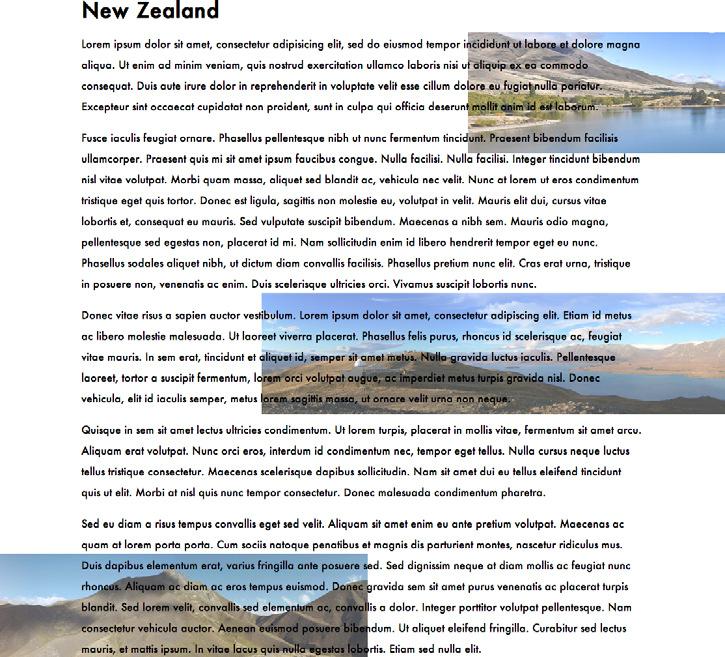
Agora para o estado padrão do figcaption (listando 3**-**40).

|  |  |  |  |
| --- | --- | --- | --- |
| **Listando 3-40.**O código CSS para uma claquete Legenda | |  |  |
| Figura.claquete figcaption { opacidade: 0; | |  |  |
| Propriedade de transição: | Opacidade, | Transformar, | Fundo; |
| Transição de função de distribuição; | Linear, | Facilidade, | Linear; |
| Transição de duração: | 2s, | 9999999s, | 0s; |
| Atraso de transição: | 0s, | 0s, | 0s; |
| ... |  |  |  |

Estes efeitos ocorrem durante a transição de *volta* para o estado padrão, definindo a opacidade da figcaption volta para 0 durante dois segundos. Você pode se perguntar por que a rotação de volta é para durar pouco menos de dez milhões de segundos. O ridiculamente elevado número é no lugar de modo que a transformação de volta para a orientação padrão para o figcaption (ou seja, pé sobre seu lado) efetivamente tem uma duração infinita. Visualmente, o que significa que não há rotação do texto tem lugar quando o usuário move o mouse para a distância a partir da imagem: a legenda simplesmente permanece ainda e se apaga.

Transições de imagem de fundo e a animação no Carregamento da Página

Para ilustrar as transições no carregamento da página, você vai fazer um guia de viagem online página para a Nova Zelândia (ver Figura 3**-**11).



**Figura 3-11.***A transição de imagens de fundo*

Quando a página for carregada, você quiser várias fotos para deslizar de um lado da página para o outro. Para este exemplo, vou usar fotografias por Chee Hong ([www.flickr.com/photos/chleong/6867222762/](http://www.flickr.com/photos/chleong/6867222762/)), Andreas Beeker ([www.flickr.com/photos/kiwiwings/2148854337/sizes/l/in/photostream/](http://www.flickr.com/photos/kiwiwings/2148854337/sizes/l/in/photostream/)), Gordon Wrigley ([www.flickr.com/photos/tolomea/4498923741/sizes/l/in/photostream/](http://www.flickr.com/photos/tolomea/4498923741/sizes/l/in/photostream/)) e David Pursehouse ([www.flickr.com/photos/mdid/2235443912/sizes/l/in/photostream/](http://www.flickr.com/photos/mdid/2235443912/sizes/l/in/photostream/)).

Primeiro, você vai configurar sua página de base, a margem de lucro para o qual permanecerá inalterada durante todo este exercício (consulte a listagem 3**-**41). Observe que as imagens são compensadas fora do lado esquerdo da página pela sua largura (1300 pixels) acrescido de uma quantia incremental, desfasando-as na horizontal).

**Listagem de 3-41.**Código HTML e CSS para um fundo de transição no Carregamento da Página

<div id=wrapper> <h1> Nova Zelândia</h1>

<p> Lorem ipsum dolor sit amet....

</div>

Corpo { font-family: Futura; margin: 0; altura: 200%; fundo:

53

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

Url(lago benmore-nova zelândia-panorama.jpg) -1300px 200px não repetir, url(lago tekapo-nova zelândia-panorama.jpg) -2000px 600px não repetir, url(Nova Zelândia-panorama.jpg) -3900px 1000px não repetir, url(lindis-pass-nova zelândia-panorama.jpg) -1300px 1300px não repita;

}

Div#wrapper { Largura: 600px; margin: 5em auto; padding: 3em; }

Infelizmente não existe ainda nenhum background-propriedade de opacidade em CSS; você deve desaparecer as imagens no PhotoShop antes de salvá-los, ou fazer outra coisa para manter a legibilidade do texto como as imagens passe por baixo dela. Para iniciar, adicionar um parcialmente transparente background-color para o div aplicando fundo: rgba(255, 255, 255, 0,8);.

Animar a imagens de fundo na Carga

É mais fácil para iniciar uma animação como este com CSS3 Keyframes (discutido no Capítulo 5), mas é possível conseguir um efeito semelhante usando somente transições, pela colocação de um pseudo-classe sobre o corpo. Para demonstrar esse conceito, você vai adicionar uma nova declaração para a sua lista de CSS (3**-**42).

**Listando 3-42.**O código CSS para um fundo de transição no Carregamento da Página

Corpo:hover { background:

Url(lago benmore-nova zelândia-panorama.jpg) calc(100%+1300px) 200px não repetir, url(lago tekapo-nova zelândia-panorama.jpg) 2400 px 600px não repetir, url(Nova Zelândia-panorama.jpg) 2400 px 1000px não repetir, url(lindis-pass-nova zelândia-panorama.jpg) 2400 px 1300px não repita; transição: 70s todos lineares;

}

Isso define as imagens para fora da tela para a extrema direita, partindo do princípio de que a janela de listagem é inferior a 2400 pixels de largura. (uma hipótese razoável para fazer agora, mas perigosas a longo prazo: uma solução melhor será para usar CSS3 variáveis ou calc uma vez que eles são totalmente suportados).

Existem três desvantagens dessa abordagem: o primeiro e o mais grave é que o momento em que o usuário do rato se move para fora da janela do navegador, as imagens será redefinido para sua posição padrão para fora da tela para a esquerda. Você poderia reduzir este problema ligeiramente movendo o código de transição para o  estado padrão do corpo, de modo a que a transição começará a ré quando o usuário do rato afastado.

A segunda questão é que todas as imagens de fundo devem ser animados juntos como um grupo; não há forma de transição imagens separadamente usando este método.

**O que é o caLc?**

Embora não seja tecnicamente necessário neste contexto, eu usei calc na declaração mostrado na listagem 3-42 como meio de demonstrar uma excitante nova propriedade. Calc vem desesperadamente perto da ideia de CSS3 variáveis (outro novo módulo CSS3), que nos permite especificar qualquer valor de comprimento como uma expressão aritmética. No caso da Listagem de 3-42, o cálculo simplesmente acrescenta a largura do elemento actual ao seu pai (o corpo) de modo que a imagem está assegurada a ser quando a página carregar off-screen.

Você pode ler mais sobre o calc em <https://hacks.mozilla.org/2010/06/css3-calc/>.

C

Capítulo 3 ■ TRANSIÇÕES CSS3 para imagens

A última questão não é assim tão grave é que as imagens de fundo fade subitamente quando se mover para trás a div. Existem duas soluções óbvias para este problema:

 Borrar as bordas esquerda e direita da div usando uma caixa de duplo-sombra na mesma cor do plano de fundo do div e uma grande quantidade de desfocagem, como mostrado na listagem 3**-**43.

**Listando 3-43.**O código CSS para desfocar a transição de imagens de fundo

Div#wrapper { Largura: 600px; margin: 5em auto; padding: 3em;

Fundamento: rgba(255, 255, 255, 0,8);

Caixa-sombra: 100px 0 100px rgba(255, 255, 255, 0,8), -100px 0 100px rgba(255, 255, 255, 0,8);

}

 Expanda a div para cobrir todo o corpo de forma a que tudo atrás dele serão desbotada. Defina a declaração de abertura como mostrado na listagem 3**-**44.

**Listando 3-44.**O código CSS para aplicar a opacidade de imagens de fundo em uma página

Html, corpo { min-altura: 100%; margin: 0; } html, corpo { altura: 100%; margin: 0; } div#wrapper { largura: 100%; min-altura: 100 %;

Estofamento: 3em;

Caixa de calibragem: caixa de fronteira; fundo: rgba(255, 255, 255, 0,8);

}

Criar e animar imagens de fundo falso

Se desejar um maior grau de controle sobre as imagens de fundo, você tem para falsos-las explorando a posição e o índice z propriedades CSS para forçar as imagens no fundo. Há dois principais métodos para fazê-lo. A primeira delas é a de criar imagens reais e vigor em segundo plano.

Em vez de desenho a imagens de fundo através do CSS esta técnica coloca-los como imagens no código, Acima, Abaixo ou mesmo dentro do div (listando 3**-**45).

**Listando 3-45.**Código HTML e CSS para criar imagens de fundo falso para uma página

<img src=lago benmore-nova zelândia-panorama.jpg alt="Lago Benmore, Nova Zelândia" id=benmore> <img src=lago tekapo-nova zelândia-panorama.jpg alt="Lago Tekapo, Nova Zelândia" id=tekapo> <img src=lindis-pass-nova zelândia-panorama.jpg alt="Lindis Pass, Nova Zelândia" id=lindis>

#benmore, #tekapo, #lindis { posição: absoluta; índice z: -1; } #benmore { topo: 300px; esquerda: 200px; }

#tekapo { topo: 600px; esquerda: 800px; } #benmore, #tekapo, #lindis {

Posição: absoluta; índice z: -1; opacidade: 0; transição: 4s linear todos os 2 s; } #lindis { topo: 900px; esquerda: 50px; }

Como imagens separadas, estes elementos podem ser animado individualmente, em todas as propriedades. Fornecer os elementos com um índice z negativo empurra-los no fundo profundo. Eles são animados na cotação de 3**-**46 Da mesma maneira que eles estavam no exercício anterior.

**Listando 3-46.**Animados imagens de fundo falso

Corpo:hover img#benmore, Corpo:hover img#tekapo, Corpo:hover img#lindis { opacidade: 0,6; }

Novamente, você normalmente usam animação keyframe se você queria a seqüência de loop independentemente da atividade do usuário. Como alternativa, você pode criar o mesmo efeito utilizando conteúdo gerado. Isto tem a vantagem de

Sem necessidade de marcação extra, mas impõe um limite de dois elementos adicionados (gerada usando :antes, um usando :depois). Como demonstrado anteriormente, estes elementos também podem também ser transformado, dado z-index e opacidade.

Resumo

Neste capítulo você aprendeu como cross-fade duas imagens usando CSS3 transições com opacidade e CSS4 cross-fade. Você também criou uma galeria de imagens simples e útil em muitos contextos local e aperfeiçoou progressivamente a galeria com as transições. Eu tenho mostrado a você como iniciar essas transições usando vários seletores de pseudo-(:alvo, :ativo e :mouse), cada uma das quais será conveniente para certas apresentações.

Você animado as legendas para as imagens e, dentro deste contexto, deslocadas o eixo de transformação para criar as rotações de "offset".

As transições módulo tem vários meios de retardar as sequências para criar efeitos de animação em camadas, desde a simples (transição de atraso) para hacks como fornecendo valores de tempo extremamente alta para manter os elementos de transição no lugar de modo eficaz para a eternidade. Você usar estas mesmas técnicas para criar diferentes e efeitos na transição de elementos que normalmente simplesmente inverta na mesma ordem de animação vieram.

Por último, você empurrou as transições em domínios em que a maioria dos desenvolvedores da bitola mal: a animação de conteúdo gerado pelo movimento e imagens de fundo no carregamento da página.

No próximo capítulo, eu vou mostrar a você como aplicar CSS3 transições para navegação no site e outros recursos da interface com o usuário.

**Capítulo 4**

**CSS3 transições para elementos de UI**

Outro papel óbvio para transições CSS3 é reforçar os elementos da interface de usuário dentro de páginas da web: criação de animação em navegação, formulários e botões, para tornar as informações no seu site mais clara e mais envolvente. Neste capítulo vou levar a princípios de animação e sintaxe de CSS que já experimentaste assim longe e expandir sobre eles nos ensinamentos que vai fazer seus designs mais interativo, interessante e divertido.

A navegação no site moderno Markup

Antes de adicionar transições para seu site navegação vale a pena tomar um momento e explorar a margem de lucro que você vai ser ligar o CSS regras. Para sites HTML5 a navegação principal deve estar localizado dentro de um elemento <nav>. Para garantir a compatibilidade com os leitores de tela, você vai adicionar um papel de navegação para o ARIA tag, como mostrado na lista 4-1.

**Lista 4-1.**O mínimo de estrutura de navegação do site HTML5

<Papel de navegação NAV =>

<a href=index.html>Home</a>

<a href=sobre.html>Sobre Nós</a>

<a href=produtos.html>Produtos</a> </nav>

■ **Observe**que você pode aprender mais sobre as funções de acessibilidade marco ARIA no

[www.w3.org/TR/wai-aria-practices/](http://www.w3.org/TR/wai-aria-practices/).

Como você pode ver, HTML5 permite que você coloque uma série simples de links do site dentro do elemento nav. Você pode encontrar, no entanto, que a adição de markup aumenta o valor semântico da navegação e permite que você modifique a sua aparência com maior flexibilidade. Na maioria dos casos, a navegação do site pode ser tratada como uma lista desordenada; alternativamente, você pode querer usar uma lista ordenada se o usuário será esperado para ler as páginas em uma ordem específica.

Seja qual for o método escolhido, atalhos do teclado também deve ser adicionado para articulações primárias através do atributo accesskey, como mostrado na lista 4-2. (Nota que o papel foi movido para o atributo lista desordenada.)

**Lista 4-2.**Um local acessível a estrutura do código de navegação

<nav>

<ul papel=navegação >

<li><a href=index.html accesskey=1>Home</a>

<li><a href=new-xyz-corp.html accesskey=2>Novidades</a> <li><a href=sobre-xyz-corp.html accesskey=a>Sobre Nós</a>

<li><a href=contactar-xyz-corp.html accesskey=6>Entre em Contato Conosco / Ajuda</a> </ul>

</nav>

Esta é a base de toda a interface de navegação principal você vai construir no presente capítulo. Para economizar espaço, você não sempre necessidade de incluir recursos de acessibilidade plena, mas é muito importante que você use esses recursos no seu código de produção final.

■ **Nota**sob algumas circunstâncias e com determinadas combinações de regras CSS é possível para pequenos desvios visual para aparecer entre os elos se estiverem dentro de <li> elementos organizados horizontalmente (como na cotação de 4-3). Enquanto se destinam a preservar o seu código não importa como você formatar como oposição ao HTML, que recolhe todas seqüencial de caracteres de espaço para baixo para um espaço único, com retornos de carro contando como um espaço, a menos que o conteúdo é envolto em <pré> tags-a causa para essas lacunas pode ser extremamente frustrante para a via para baixo.

O problema não reside com o CSS, mas com o HTML, sob a forma de retornos de carro entre cada item da lista. Enquanto CSS soluções para este problema existe (definição font-size: 0 no elemento pai, por exemplo, ou flutuante do item de lista elementos), a melhor opção é geralmente para remover o carro retorna, colocando todos os itens da lista em uma única linha, como mostrado na lista 4-3.

**Lista 4-3.**Uma estrutura de código de navegação local acessível sem espaços extras

< **** <ul papel=navegação><li><a href=index.html accesskey=1>Home</a><li><a href=new-xyz- **** corp.html accesskey=2>Novidades</a><li><a href=sobre-xyz-corp.html accesskey=a>Sobre Nós</a><li><a href=contactar-xyz-corp.html accesskey=6>Entre em Contato Conosco / Ajuda</a></ul>

</nav>

Por razões de clareza não vou estar usando esta solução em amostras de código para vir, mas você deve estar ciente do potencial problema e sua solução.

Independentemente de estarem ou não são escritos em uma única linha de código de navegação, estruturas construídas com listas ordenada ou desordenada exibirá cada link em uma linha separada. Para criar uma barra de navegação horizontal, você deve fazer um aditamento ao seu CSS.

Noções Básicas de Barra de Navegação Horizontal

Interfaces de navegação horizontal são usualmente empregados quando um site é relativamente pequena. (barras de menu pendente, que vou cobrir em breve, constituem uma excepção a esta regra.)

"O mágico número sete mais ou menos 2" é um efectivo Estado de polegar para determinar a estrutura de navegação: em média, os seres humanos podem recordar ou relacionar a até sete itens a qualquer momento. Mais de sete itens na sua navegação geralmente significa que você precisará para reconsiderar a UI (user interface)-*arrancamento* ou agrupamento de itens relacionados normalmente resolve o problema. Dependendo da largura da tela, você também pode aplicar sete itens de navegação geralmente horizontalmente em uma janela do navegador de navegação; muito janelas estreitas (como em um dispositivo móvel) ou mais de sete articulações primárias geralmente chamada para um formato de navegação orientada verticalmente.

Os links são automaticamente apresentados em linha, assim não há necessidade de adicionar o CSS apenas para parecerem organizados em uma barra de navegação horizontal. Se você envolveu o links em <li> tags, há pouco trabalho para ser feito: basta adicionar a declaração na cotação de 4-4 para seu stylesheet.

Os links são automaticamente apresentados em linha, assim não há necessidade de adicionar o CSS apenas para parecerem organizados em uma barra de navegação horizontal. Se você envolveu o links em <li> tags, há pouco trabalho para ser feito: basta adicionar a declaração na cotação de 4-4 para seu stylesheet.

**Lista 4-4.**CSS para criar uma barra de navegação horizontal a partir de uma estrutura de navegação HTML5

Nav li { Display: inline; }

Agora que você definiu a base markup para o local mais comum de formatos de navegação, você pode passar para o reforço com CSS3.

Uma barra de navegação simples com CSS3 Aprimorado

Vamos tomar o mais simples possível para navegação a partir da listagem de marcação 1-1 e coloque uma imagem de fundo atrás da barra de navegação. Você vai verificar visualmente formatar o texto de forma que ele ainda pode ser lido através da adição de um texto de sombra e um efeito hover, como na cotação de 4-5.

**Lista 4-5.**CSS para criar uma barra de navegação horizontal a partir de uma estrutura de navegação HTML5

Nav { background: url(images/nuvens.jpg) repita; padding: 1em 0; } nav { text-decoration: nenhum; color: #fff; padding: 1em; font-family: Futura, Arial, sans-serif;

Texto-sombra: 2px 2px 1px rgba(0, 0, 0, 0,6); } nav uma:foco { background: rgba(0, 0, 0, 0,7); }

Então você vai adicionar uma transição simples para graduar a imagem de fundo atrás de um link pairou ao longo do tempo, ajustando o nav um selector na cotação de 4 a 6.

**Lista 4-6.**Usando CSS3 para introduzir um efeito de transição para um link

Nav { text-decoration: nenhum; color: #fff; padding: 1em; font-family: Futura, Arial, sans-serif;

Texto-sombra: 2px 2px 1px rgba(0, 0, 0, 0,6); -webkit-fundo de transição: .85s facilidade em; -moz-fundo de transição: .85s facilidade em; -o fundo de transição: .85s facilidade em; fundo de transição: .85s facilidade em;

}

Finalmente, um pouco de uma verificação de segurança: sempre que você coloque uma imagem de fundo de um elemento você também deve definir um plano de fundo de cor que representa o principal matiz da imagem, apenas no caso de a imagem não

Carga. Modificando a declaração nav como mostrado na listagem 4-7 garante que o texto associado será legível sob todas as circunstâncias.

**Lista 4-7.**Criando um CSS Cor de Fundo como um fallback para uma imagem

Nav { background: url(images/nuvens.jpg) #007 não repita; padding: 1em 0; }

Você também pode aplicar uma aparência diferente para os links que o usuário tenha visitado anteriormente usando

Nav uma:visitado como um selector. Como mostrado na listagem 4-8, você pode até mesmo criar uma transição de cor diferente quando a pairar sobre tais links.

59

Capítulo 4 ■ CSS3 transições para elementos de UI

**Lista 4-8.**A criação de um efeito separado para links visitados

Nav uma:visitou:foco { background: #f00;

Fundamento: rgba(255, 0, 0, 0,7);

}

O efeito de transição ainda vai ser aplicado para links visitados, mas eles até se apagarem vermelho, não preto. Isto irá produzir um menu como mostrado na Figura 4-1.



**Figura 4-1.***Uma barra de navegação com um animado efeito hover*

Adicionando CSS3 desta forma é totalmente retro-compatível com os navegadores mais antigos. Se o navegador não apoiar as transições, os usuários simplesmente ver um parcialmente transparente de fundo atrás das actualmente pairou sobre o link.

■ **Dica**a maneira mais fácil de imaginar o comportamento de versões mais antigas de navegadores que suportam apenas CSS 2.1 é para ter em conta o facto de se impor uma duração de 0 em qualquer transição. Aliás, é a razão os valores padrão de transição de duração e de propriedade de transição são 0 e todos, respectivamente.

Se você quiser garantir ainda mais a compatibilidade do navegador, você pode adicionar um fallback preto plano (#000) de fundo para o pairou links, antes da seção da declaração rgba. Navegadores que compreender rgba irá seguir a última regra; aqueles que não irá seguir a cor hexadecimal. O fade-in ainda funcionará e aparecem corretamente em navegadores que suportam.

Destacando a página atual na navegação

Destacando a página atual do usuário na barra de navegação é problemático. O CSS não está ciente do estado interno de um aplicativo que não sabem *onde está a.* Você pode evitar esse problema através da criação de uma classe, atributo de dados ou o id para o auto-referencial link em cada página com outro no corpo tag, permitindo CSS para corresponder os selectores, como mostrado na Figura 4-2.



**Figura 4-2.***Uma barra de navegação com a localização atual realçado*

Começar com o código mostrado na cotação de 4-9.

**Lista 4-9.**Markup para um visual "Você está aqui" efeito sobre as barras de navegação do site

No index.html:

<body id=home>

<Papel de navegação NAV =>

<a href=index.html class=home>Home</a>

<a href=sobre.html class=sobre>Sobre Nós</a>

<a href=produtos.html class=produtos>Produtos</a> </nav>

Sobre.html:

<body id=sobre>

<Papel de navegação NAV =>

<a href=index.html class=home>Home</a>

<a href=sobre.html class=sobre>Sobre Nós</a>

<a href=produtos.html class=produtos>Produtos</a> </nav>

Visando o link que representou a página atual é então um simples processo de agrupar uma série de seletores descendentes, mostrar na cotação de 4-10.

**Listando 4-10.**CSS para destacar Current PageLinks

#home nav uma.home, #sobre nav.sobre um fundo {: rgba(0,0,212,0.6);}

Também é possível alcançar este efeito com uma folha de estilos incorporada em cada página que aborda cada link classe individualmente. Ambas as abordagens têm a mesma desvantagem: eles requerem personalização do markup para cada página. A melhor solução pode ser uma combinação de CSS e JavaScript (mostrado aqui sob a forma de JQuery).

O HTML retorna para o que se tinha no início (listando 4-1), e seu stylesheet toma simplesmente uma nova classe, mostrado na lista 4-11.

**Listando 4-11.**CSS para destacar "Você está aqui" Links através de JavaScript

Uma corrente { background:.rgba(0,255,0,0.7); }

Na cabeça de cada documento que você inclua o código mostrado na cotação de 4-12.

**Listando 4-12.**JavaScript para aplicar uma "Você está aqui" Class para a navegação no site

<script src=//Code.jquery.com/jquery.min.js></script> <script>

$(documento).pronto(função(){ $('ul[papel="navegação"]").Cada função(() {

Se (este.href === window.location.href){ $(esta). addClass('atual');} }); })

</script>

61

Capítulo 4 ■ CSS3 transições para elementos de UI

Você pode usar qualquer uma dessas abordagens para realçar a página atual na navegação mais avançada exemplos você vai criar na próxima seção.

Guia horizontal de navegação com transições CSS3

Como um exemplo mais avançado da navegação do site, você pode incluir a circulação no "guia" de navegação. Neste exemplo, você vai colocar os links em uma lista ordenada para fornecer mais estrutura como você fez na cotação de 4-2. Você vai criar a impressão visual de guias definindo os links lado a lado, adicionando uma fronteira-raio à sua parte superior direita e cantos superior esquerdo e colocando um gradiente no fundo de cada, por listagem 4-13. Isto irá produzir um menu como mostrado na Figura 4-3.



**Figura 4-3.***Um sistema de navegação guia de animação*

Neste exemplo, você vai usar um seletor de atributo para ter acesso à lista de itens e links.

**Listando 4-13.**Estilo CSS para links de navegação como guias

Ul papel=Navigation][li {

Visor: inline; font-family: Futura, Arial, sans-serif; texto-transformar: letras maiúsculas;

}

Ul[papel=Navigation] li um { text-decoration: nenhum; color: #fff; padding: 0.8Rem 1.2Rem 2rem 1.2rem;

Fronteira: 1px solid #777; border-radius: 5px 5px 0 0; fundo: linear-gradiente(para baixo, #dfc891, #776c51); caixa-sombra: 0 0 15px rgba(0,0,0,0.5);

Espaçamento de letra: 0,15rem; texto-sombra: 0 1px 0 #000;

}

Em seguida, você deseja sobrepor as abas levemente. Você pode fazer isso através do fornecimento de uma margem negativa de esquerda para cada item da lista. As guias estarão se movendo para cima no mouseover, assim você deliberadamente torná-los mais do que eles precisam estar em seu estado normal. Estouro de uso: ocultas na lista desordenada para ocultar o excesso na borda inferior, conforme mostrado na listagem 4-14.

**Listando 4-14.**Estilo CSS para links de navegação como guias

Ul[papel=Navigation] {

Fundamento: #000; padding-top: 3.2Rem; padding-bottom: 1rem; transbordamento: hidden; margin-top: 0; }

Ul papel=Navigation][li {

Visor: inline; font-family: Futura, Arial, sans-serif; texto-transformar: letras maiúsculas; margin-left: -.5rem; }

O link que representa a página atual terá uma classe de vanguarda que lhe é aplicada. Esta classe irá trazer o link apropriado na parte superior e ligeiramente mais alta do que todos os outros guia, usando posição: relativa, um valor negativo de cima e modificado z-Index (listando 4-15).

**Listando 4-15.**Colocando o Link para a página atual na parte superior dos outros

Ul[papel=Navigation] li a.forefront { -0.2rem; índice z: 2;}

Você pode aplicar essa classe usando qualquer das abordagens já discutidas: adicionando ao markup ou dinamicamente a sua aplicação com JavaScript. Por último, você vai levantar os links no hover elevando sua posição superior e animar a transição do estado padrão (listando 4-16).

**Listando 4-16.**Guia de animação dos links

Ul[papel=Navigation] li um { posição: relativos; superior: 0; transição: 0,2s todos lineares;

...

}

Ul[papel=Navigation] li um:foco, ul[papel=Navigation] li um:foco { topo: -0,6rem;

}

Animação de validação personalizados Erros para formulários HTML5

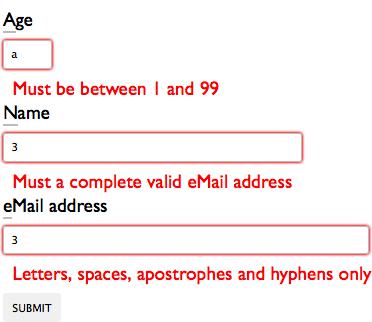
As informações inseridas pelos usuários em formulários geralmente é verificada pelo menos duas vezes: uma vez na extremidade dianteira (mais comumente com JavaScript) e uma vez no back-end (com PHP ou algum outro idioma do lado do servidor). Existem várias vantagens deste método:

 *Redundância*: Se o processo de validação do lado do cliente não funcionar ou se está bloqueada ou não suportados no navegador, o processo de back-end ainda vai olhar para os erros.

 *O imediatismo*: uma solução do lado do cliente normalmente fornece um feedback instantâneo como o visitante é inserindo informações em um campo ou depois de o campo perde o foco; curto de usando Ajax ou uma tecnologia semelhante, soluções do lado do servidor não pode fornecer uma resposta até que o botão Enviar é pressionado.

 *Segurança*: geral, uma linguagem do lado do servidor irá oferecer uma medida mais adequada e segura de obter informações de verificação de cartão de crédito VISA e Mastercard de usar JavaScript.

Houve muitos scripts de validação construído em JavaScript, mas em HTML5 a mesma capacidade de validar formulários é agora suportados nativamente no navegador padrão com o atributo e dois novos CSS pseudo-classes::válido e:inválido. A fim de demonstrar que você vai criar o formulário mostrado na Figura 4-4.



|  |  |  |  |
| --- | --- | --- | --- |
| **Figura 4-4.***Mensagens de Erro de forma animada com CSS3* |  |  |  |
| Você começar com a marcação de base para uma forma acessível, mostrado na lista 4-17. | |  |  |
| **Listando 4-17.**Um formulário HTML acessível5 |  |  |  |
| <form> |  |  |  |
| <label para=idade accesskey=a>idade</label> | **** |  |  |
| <tipo de entrada=número nome=idade id=idade size=3 maxlength = 2 min = 1 max=99 |  |  |
| Pattern="^([1-9]|[1-9][0-9]{1,2}$" necessária > |  |  |  |
| <label para=nome accesskey=n>Nome</label> |  |  |  |
| <tipo de entrada=Nome do texto=nome id=nome size=40 maxlength=38 **** |  |  |  |
| Pattern="^[a-zA-Z]'?[- "a-zA-Z]+$" placeholder="Seu nome completo" necessária > | |  |  |
| <label para e-mail=accesskey=e>endereço de e-mail</label> |  | **** |  |
| <tipo de entrada=nome de e-mail=email id=email size=50 maxlength=256 placeholder="Seu e-mail de contato" | |  |

Required>

<tipo de entrada=apresentar valor = Apresentar> </forma>

Como você pode ver, você está usando um campo de número com um valor mínimo e máximo para limitar a idade do usuário quando eles entram, um e-mail de entrada e uma entrada de texto padrão com uma expressão regular para obter o nome do usuário. Todas as entradas são necessários. (eu adicionei um padrão para o número de entrada embora é inválido em HTML5 a fim de obter a execução no Firefox 14 e anteriores, que reconhece padrão mas não o próprio valor de atributo número, assumindo o campo de entrada de texto padrão). Você pode usar algumas noções básicas de CSS para melhorar a exibição desses elementos, como mostrado na listagem 4-18.

**Listando 4-18.**CSS para uma típica forma acessível

Forma { font-size: 1.2Rem; font-family: "Gill Sans", Arial, sans-serif; } label { Display entrada,: bloco; }

Etiqueta:primeira carta { fronteira-BOTTOM: 2px solid #bbb; }

Fronteira de entrada {: 1px solid #bbb; padding: .4rem; border-radius: .2rem; margin: .5rem 0; } input[type="Enviar"] { border: nenhum; texto-transformar: letras maiúsculas; }

O único aspecto incomum de esta ligeiramente stylesheet é a terceira declaração, que sublinha a primeira letra de cada formulário rótulo para mostrar as accesskey para o campo associado. Para visualizar se a informação inserida na entrada é certo ou errado, você precisa adicionar o código na cotação de 4-19. (Nota a primeira declaração CSS, que desliga o padrão atual de Firefox UA estilos para entradas válidas e inválidas).

**Listando 4-19.**CSS para uma típica forma acessível

Entrada:válido, entrada:inválido {box-sombra: nenhum; } entrada:válido { border: 2px verde sólido; } entrada:inválido { border: 2px amarelo sólido; }

Enquanto você poderia certamente animar o que você tem até agora, existem algumas desvantagens para a forma como actualmente é apresentado: o navegador padrão para mostrando as entradas como inválido quando o usuário primeiro vê a página e não há indicação de exatamente *o que* o usuário digitou incorretamente.

Você não pode usar o pseudo-classes para gerar mensagens de erro após as entradas porque eles são substituídos elementos, mas você *pode* colocar as mensagens de erro do atributo título de enlaces *após* as entradas. Alguns exemplos são mostrados na cotação de 4-20.

**Listando 4-20.**Span elementos adicionados como mensagens de erro de validação após HTML5 Entradas de formulários

<span title="deve ser entre 1 e 99"></span>

<span title="deve concluir um endereço de correio electrónico válido"></span>

<span title="letras, espaços, apóstrofos e hífens apenas"></span>

Porque cada span apenas contém informações em seu atributo título e não tem o conteúdo visível, os navegadores mais antigos que não suporte totalmente CSS3 são desativados a partir mostrando qualquer mensagens confusas para o usuário. Remova a lista a partir de seu código e 4-20 substituí-la com a lista de 4-21.

**Listando 4-21.**A modelagem e a exibição de erros de validação

Entrada + span:após { conteúdo: attr(título); cor: vermelho; margin-left: 0.6Rem; opacidade: 0; } entrada:inválido + span:após { opacidade: 1; }

Há um problema imediato: o navegador continua a mostrar as entradas como errado fazendo o span Título atributo visível no momento em que a página é carregada. O motivo?

■ **Dica**uma entrada com um  atributo requerido é avaliado pelo seu padrão ou tipo de entrada antes de o usuário se concentra ou entra em qualquer informações no campo.

Removendo o atributo obrigatório a partir de entradas e adicionar uma transição para as mensagens de erro, como na cotação de 4-22, você alcançar o efeito que são depois. Note que eu adicionei um atraso para a transição: sem esta, a mensagem de erro seria mostrar imediatamente na inserção do primeiro caractere em um campo. Você deseja uma quantidade razoável de tempo para passar antes de começar dizendo ao usuário que sua informação está errada.

**Listando 4-22.**Exibindo uma mensagem de erro de validação do Formulário após um atraso

Entrada + span:após { conteúdo: attr(título); cor: vermelho; margin-left: 0.6Rem; opacidade: 0; propriedade de transição: opacidade;transição duração: 2s; atraso de transição: 2s;}

Você também pode exibir o status de uma entrada para a validade do mostrando um carrapato dentro do elemento de formulário. Voltando ao markup na cotação de 4-17 e o CSS na cotação de 4-18, adicionar o CSS na cotação de 4-23. (Você pode aproveitar mais esta rodando o carrapato.png em um sprite e manipulação de imagem para mostrar uma cruz quando a entrada é inválido).

**Listando 4-23.**Exibindo uma Imagem de Fundo para o formulário de validação de entrada

Entrada:focagem:válido { background-image: url(tick.png); fundo-repetir: não-repetição; fundo-posição: direito 6px; }

Você não pode desbotar em imagem de fundo diretamente (ainda não existe um controlo directo para opacidade em imagens de fundo), mas você *pode* manipular uma imagem, então se você queria para fade-no símbolo, você poderia colocar a imagem após a entrada e a transição de TI como você fez com os anteriores a validação de mensagens.

■ **Nota**Você também pode desativar o navegador incorporado na forma de mensagens de validação inteiramente através de JavaScript, como mostrado aqui, usando jQuery:

$(documento).pronto(função() { $('forma').bind('inválido', função(e){

$(e.target).attr('validade") }); });

Pressione o botão da interface de Transição

Com CSS, você também pode dar a impressão de um link ou o botão de abaixar ou afundar na página, como mostrado na Figura 5.



**Figura 4-5.***Um típico exemplo de botão*

Uma maneira simples de obter esse efeito é mostrado na lista 4-24.

**Listando 4-24.**CSS para fazer um link abaixado quando clicado

Um:ativa { posição: relativos; Superior: 1px; }

66

Capítulo 4 ■ CSS3 transições para elementos de UI

Naturalmente, também é possível obter mais detalhadamente o estilo de um link, como na cotação de 4-25.

**Listando 4-25.**HTML e CSS para criar um gigante Botão Ajuda

<a href=# class=bigbutton>Ajuda</a>

Um.bigbutton {

Font-family: "Auto-estrada Azul"; texto-transformar: letras maiúsculas; color: #fff;

Fundamento: gradiente radial(centro 50px, círculo mais distante de canto, #EF6634, #c63a17 43%#ba1a01 45%#bf6e4e 100%);

Visor: bloco inline; largura: 200px; altura: 200px; border-radius: 100%; font-size: 70px; text-decoration: nenhum; alinhar texto: centro; padding-top: 50px; caixa de calibragem: caixa de fronteira; font-weight: 900;

Caixa-sombra: 0 8px 0 rgb(183,0,0), 0 15px 20px rgba(0,0,0,.35); texto-sombra: 0 3px 1px rgba(122,17,8,.8);

Transição: .4s todas as facilidade em;

}

Aqui você cria a impressão de um botão 3D criando duas sombras caixa: um para criar a impressão de um Edge para o botão e outro abaixo dele como uma sombra geral. Você cria a impressão de o botão descida pela transição de quatro acções simultâneas:

1. Baixe o texto do botão.
2. Fisicamente Baixe toda a botão utilizando traduzir.
3. Reduzir a sombra que representa a borda do botão.
4. Reduzir e proteger a desfocagem de sombra sob o elemento inteiro.

Você vai fazer isso em uma declaração baseada no :ativo pseudo-seletor e como mostrado na listagem 4-26.

**Listando 4-26.**HTML e CSS para criar um gigante Botão Ajuda

.bigbutton:ativa { padding-top: 53px;

Transformar: traduzir(0, 4px);

Caixa-sombra: 0 4px 0 rgb(183,0,0), 0 8px 6px rgba(0,0,0,.45);

}

Também é possível criar um botão de efeito revelar, como você verá na próxima seção.

Botão de interface de revelar a transição

Botão de elementos HTML padrão também pode ser altamente personalizados com CSS que inclui transições. Neste caso, você deseja um botão para expandir em pairar, revelando uma mensagem promocional ou orientar dentro (veja a Figura 6).



**Figura 6.***Um botão em dois modos, migrou com CSS3*

O seu HTML será constituído por três elementos: um botão com duas span tags dentro dele. Você pode ver a margem de lucro na cotação de 4-27.

**Listando 4-27.**HTML para revelar um botão

Botão <>Inscreva-se

<span class=hidden>Gratuitamente</span> <span class=direita>&#9654;

</span>

</button>

■ **Dica**um caractere Unicode é muito mais adaptável e mais fácil de controlar do que uma imagem quando usado para o lado direito do seu elemento de UI. Utilizei uma entidade decimal personagem aqui para representar um preto à direita (um grande recurso para caracteres Unicode pode ser encontrada em copypastecharacter.com.)

Em seguida, você vai adicionar o CSS para configurar sua aparência de botões (listando 4-28).

**Listando 4-28.**Base CSS para um botão revelar

\* caixa de calibragem: {-fronteira-caixa;

Color: #333; font-family: Futura, Arial, sans-serif;

}

Botão {

Font-weight: 600;

Texto-sombra: 0px 1px 1px rgba(255, 255, 255, 0,3); border-radius: 34px; altura: 68px; largura: 180px;

Padding: 0 20px; font-size: 18px;

Justificativa: linear-gradiente(para baixo, #1e5799,#7db9e8); fronteira: 1px solid rgba(0,0,0,0.4);

Caixa-sombra: 0px 1px 1px rgba(255, 255, 255, 0,8) inset, 1px 1px 3px rgba(0, 0, 0, 0,2),

0px 0px 0px 4px rgba(188, 188, 188, 0,5); alinhar texto: esquerda;

}

Span.escondida, span.direito { color: #fff; } span.direito { PADDING-LEFT: 18px; }

Span.oculto {fundo: linear-gradiente(para baixo, #222, #000); exibição: bloco inline; largura: 0; margin-left: 22px;

Transbordamento: oculto;

Espaço branco: nowrap; padding: 22px 0; border-LEFT: 2px rgba(0,0,0,0.3 duplo); }

Porque o .oculto span é definido para o dimensionamento de caixa: fronteira-box, branco-espaço: nowrap, transbordo: ocultos e de largura: 0, você verá apenas a sua fronteira no lado esquerdo sob condições normais.

Pretender que vários eventos de transição para acontecer ao mesmo tempo e no mesmo movimento. Enquanto você poderia colocar as propriedades de transição dentro de declarações separadas, é muitas vezes mais eficiente para colocá-los no alto no CSS como possível.

Capítulo 4 ■ CSS3 transições para elementos de UI

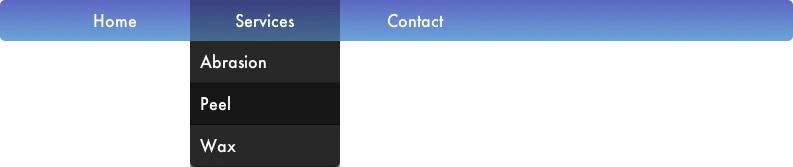
**Listando 4-29.**CSS para revelar um botão transição

Botão { transição: .6s todas as facilidade em; }

Botão:hover { Largura: 290px; color: #fff; texto-sombra: nenhum; } botão:hover span.oculto { Largura: 120px; padding: 22px; }

Acessível suspensa horizontal de navegação com transições CSS3

Menus de lista pendente são uma opção de navegação comum para sites complexos. Os usuários já estão familiarizados com o formato na UI em seus computadores, isso usando a mesma interface design em sítios web faz sentido. Consulte a Figura 7 para obter um exemplo.



**Figura 4-7.***Um menu pendente*

Menus de lista pendente em páginas HTML têm tradicionalmente sido criado usando o Flash ou JavaScript, mas essas ferramentas compartilhar uma desvantagem significativa: menus criados com eles pode ser difícil para a mobilidade para utilização com leitores de tela e/ou teclados.

Usando os recursos do ARIA em HTML5 markup com CSS3 transições, você obtém o melhor de todos os mundos: elementos da interface de usuário que são acessíveis, visualmente atraente e animado. A marcação de base é mostrado na lista 4-30.

■ **Dica**Você pode aprender muito mais sobre as funções do ARIA no [www.w3.org/TR/wai-aria/roles](http://www.w3.org/TR/wai-aria/roles).

**Listando 4-30.**HTML para um local acessível no menu pendente

<Papel de navegação NAV = etiqueta aria="Menu Principal"> <ul papel=menus>

<li papel=menuitem tabindex=0><a href=#>Home</a>

<li papel=menuitem aria-haspopup=true tabindex=0><a href=#>Serviços</a> <ul class=submenu Menu papel=-oculto aria=true>

<li papel=menuitem tabindex=-1><a href=#>Abrasão</a> <li papel=menuitem tabindex=-1><a href=#>Peel</a>

<li papel=menuitem tabindex=-1><a href=#>Cera</a> </ul>

<li papel=menuitem tabindex=0><a href=#>Contato</a> </ul>

</nav>

O código se assemelha o que você começou com na cotação de 4-1 (sem accesskeys, que foram omitidos neste exemplo para economizar espaço). A função de navegação foi movido de volta para o elemento nav, que também ganhou um rótulo descrevendo a sua finalidade, o equivalente de auxílio de um título dica.

Cada item da lista tem um papel de menuitem, indicando que ele é passível de ser, e um de 0 tabindex (indicando que ele pode ser saltou entre usando a tecla TAB) ou -1 (o que significa que outros controles, como as teclas de seta, deve ser usado para concentrar-los). Por último, o item da lista contendo o submenu tem um atributo de haspopup aria e o submenu em si tem um atributo oculto de aria para indicar que ele não está visível por padrão.

Para isso, você vai adicionar algumas noções básicas de CSS, mostrado na lista 4-31.

**Listando 4-31.**Para um código CSS3 no menu pendente

Corpo { font-family: Futura, Arial, sans-serif; } nav { Height: 41px;

Justificativa: linear-gradiente(para baixo, rgb(93,146,207) 0%, rgb(79,79,181) 100%); } nav ul { margin: 0; }

Nav, ul.submenu { background: #35f; border-radius: 5px; Padding: 0; }

Nav ul li { Display: bloco; largura: 150px; alinhar texto: centro; flutuação: margem esquerda;: 0; padding: 0; }

Nav li:foco { background: rgba(0,0,0,0.4); }

Nav { color: #fff; text-decoration: nenhum; exibição: bloco; padding: 10px; }

Nav ul.submenu { background: rgba(0,0,0,0.8); posição: relativos; border-radius: 0 0 5px 5px; altura: 0px; transbordamento: hidden; }

Submenu nav ul.li { flutuação: nenhum; alinhar texto: esquerda; border-BOTTOM: 1px rgba(0,0,0,0.3 sólido); }

O código na listagem 4-31 deveria ser bastante auto-explicativa: os principais itens da lista são lançadas como lado a lado com o submenu colocado abaixo e, neste caso, oculto, definindo a sua altura para 0.

A implementação do CSS3 na maioria dos navegadores atuais exige que você defina medidas explícitas quando a transição de um dimensões do elemento; nem altura: 100% nem a altura: auto funcionará neste caso. Além disso, deseja que o mesmo efeito de transição a ser adicionado a vários elementos; em vez de repetir, você adicionar a declaração no topo da Listagem de 4-32.

**Listando 4-32.**CSS para uma animada no menu pendente

Nav ul li { transição: .3s todos lineares; }

Nav ul li ul:foco.submenu { height: 126px; }

Esta marcação e CSS podem se tornar a base de muitos projetos de interface. Por exemplo, com alguns pequenos ajustes, verticalmente expandir menus "acordeão" também pode seguir este modelo.

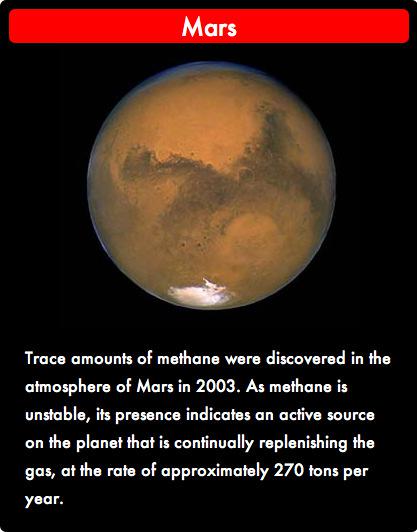
Início CSS3 Efeitos de transição após um clique de botão

Na maioria HTML padrão, não há forma de estado de gravação; isto é, você pode captar a acção do utilizador do rato sobre um elemento (:foco), pressionado para baixo (:activo), e em alguns casos, entrou dentro de um elemento (:foco), mas existem poucas imediatamente óbvio modos de dizer "se este elemento é clicado, isso; mas se estiver desligado, desfazer a ação".

Você tem algumas opções com CSS, no entanto. O estado de ambos os elementos de rádio e a caixa de seleção é legível pelo CSS através do uso do :verificado pseudo-classe. Ocultando a caixa mas mantendo a acessibilidade através da utilização de um rótulo associado elemento você pode usar CSS para "switch" a apresentação da página.

■ **Observe**enquanto as técnicas vou discutir as próximas são válidos em HTML5 (que permite que você coloque elementos de formulário em qualquer lugar em uma página), eles são semanticamente questionável. Existe uma escola de pensamento que alega que elementos do formulário deve ser utilizado apenas dentro de formas e de que estes tipos de comportamentos devem ser relegado exclusivamente para JavaScript. Você deve estar ciente desta tese antes de prosseguir com as seguintes técnicas.

Em seguida, irei demonstrar como alcançar a imagem mostrada na Figura 8.



**Figura 4-8.***Uma imagem e descrição de Marte mostrado com um elemento clicado (Imagem cortesia da NASA)*

Seu markup será composto de um recipiente div, uma caixa de seleção de entrada, e uma interior div contendo alguns conteúdos, por listagem 4-33.

**Listando 4-33.**Caixa de seleção Markup para abrir e fechar uma Associada div

<div id=mars>

<tipo de entrada de seleção id=marstoggle => <div class=alternar>

<img src=mars.jpg alt=Mars>

|  |  |  |
| --- | --- | --- |
| <p>vestígios de metano foram descobertos na atmosfera de Marte em 2003... **** | **** |  |
| Como o metano é instável, sua presença indica uma fonte ativa no planeta que é |  |
| Continuamente o reabastecimento de gás, à taxa de aproximadamente 270 toneladas por ano. **** |  |  |
| </div> |  |  |
| </div> |  |  |
| Você estilo exterior do div e seu conteúdo usando o código na cotação de 4-34. |  |  |
| **Listando 4-34.**Div para texto pop-up |  |  |
| Div#mars { background: #000; color: #fff; |  |  |
| Font-family: Futura, Arial, sans-serif; |  |  |
| Largura: 400px; padding: 1.6Rem; |  |  |
| Altura de linha: 175%; border-radius: 6px; } |  |  |
| Img { Display: bloco; margin: 0 auto; } |  |  |
| Div#mars p { margin: 1rem; } |  |  |

Você deseja ocultar a div interna usando um seletor de irmão para definir a sua altura para 0 e escondendo alguma coisa que cai fora dela. Ao mesmo tempo você também definir a transição usando os princípios de animação push-pull discutidos anteriormente. (Consulte a listagem 4-35).

**Listando 4-35.**A transição para a abertura e o fechamento de uma div

Entrada ~ div { transbordamento: oculto; altura: 0; transição: .6s todos os metros cúbicos linhas vectoriais assim(0,735, **** 0,145, 1,625); }

Para completar a funcionalidade básica do seu elemento interativo que pretende expandir o div interior se a caixa de verificação estiver activada. Como referi, altura precisa ser explicitamente configurado para elementos de transição corretamente em navegadores atuais (consulte a listagem 4-36).

**Listando 4-36.**Div para texto pop-up

Entrada:verificado ~ div { height: 480px; }

Isso funciona muito bem aqui, mas ocultar e mostrar o conteúdo da página com um botão de rádio será provavelmente não funciona bem para a maioria dos designs. Para contornar isso, você pode associar uma etiqueta com a entrada, "ligar" o texto da etiqueta para a caixa de seleção usando o atributo de Como resultado, clicando na etiqueta irá ligar o elemento de formulário e fazer com que o seu CSS para continuar a responder, mesmo se a caixa de verificação em si é oculta (consulte a listagem 4-37).

**Listando 4-37.**Para uma completa Markup Checkbox-Associated div

<div id=mars>

<label para=marstoggle>Mars</label>

<tipo de entrada de seleção=class=alternar id=marstoggle> <div>

<img src=mars.jpg alt=Mars>

<p>vestígios de metano foram descobertos na atmosfera de Marte em 2003...

</div>

</div>

Rótulo { Display: bloco; alinhar texto: centro; font-size: x-grande; fundo: vermelho; border-radius: 6px; padding: .2rem; } etiqueta:foco { background: amarelo; color: #000; cursor ponteiro: entrada; } { DISPLAY: none; }

Se você estiver disposta a ignorar algumas questões de semântica, existem muitas possibilidades para esta caixa de controle. Ele pode até ser usado em menus para os estados de retenção aberta (por exemplo, como uma série de guias com conteúdo de baixo ou como uma sanfona menu).

Animação de elementos de formulário com CSS3

É possível alargar as transições para modificar a aparência de elementos de formulário diretamente. Um exemplo simples seria um visual "wobble e fade" quando as informações inseridas em um elemento texto está incorreta, como mostrado na listagem 4-38 e Figura 4-9.

**Listando 4-38.**CSS para elementos de forma animada simples

Entrada { padding: 1rem; transição: .5s 2s todos os metros cúbicos linhas vectoriais assim(0,475, -0.600, 0,435, 1,650); }

Entrada:inválido { border: 3px vermelho sólido; transformar: translateX(10px); }



**Figura 4-9.***Uma animada, conjunto personalizado de botões de rádio*

Com alguns avisos, você pode estender esta para botões de rádio, usando uma técnica proposta pela primeira vez Simurai (<http://simurai.com/>). A marcação de base é muito simples: cada botão de rádio tem o mesmo nome, o que significa que cada botão desliga o outros desligado quando clicado. Um botão de rádio é automaticamente ativado por padrão, como mostrado na listagem 4-39.

**Listando 4-39.**HTML para Botões de rádio animados

<tipo de entrada="Rádio" name="radiobutton" verificado> <tipo de entrada="Rádio" name="radiobutton">

<tipo de entrada="Rádio" name="radiobutton"> <input type="Rádio" name="radiobutton"> <input type="Rádio" name="radiobutton">

Em seguida vem a CSS, mostrado na lista achatados 4-40.

**Listando achatados 4-40.**CSS animados de botões de rádio

Corpo de entrada { background-color: rgb(20%,20%,20%); } input {

Aparência: nenhum;

Margem: 10px; largura: 22px; altura: 22px; border-radius: 50%; cursor: ponteiro; alinhar vertical: médio;

Caixa-sombra: hsla(0,0%,100%,.15) 0 1px 1px, inset rgba(0,0,0,.5) 0 0 0 1px;

Background-color: rgb(20%,20%,20%); imagem de fundo: radial-gradiente( hsla(0,100%,90%,1) 0%,

Hsla(0,100%,70%,1) 15%, hsla(0,100%,60%,.3) 28%, hsla(0,100%,30%,0) 70%

);

Fundo - repetir: não-repetição;

Propriedade de transição: posição de fundo, transformar; transição duração: .15s, .25s;

Transição de função de distribuição: metros cúbicos linhas vectoriais assim(.8, 0, 1, 1);

}

Entrada:verificado {

Propriedade de transição: posição de fundo, transformar; transição duração: .2s, .25s;

Atraso de transição: .15s, 0s; transição de função de distribuição: metros cúbicos linhas vectoriais assim(0, 0, .2, 1);

}

Entrada:ativo { transformar: escala(1,5); transição: transformar .1s-cúbicos linhas vectoriais assim(0, 0, .2, 1); } entrada entrada,:ativa { background-posição: 22px 0; }

Entrada:verificado { background-posição: 0 0; }

Entrada:verificado ~ Entrada, Entrada:verificado ~ entrada:ativo { background-posição: grelhas -22º px 0; }

■ **Observe**o aparecimento propriedade está destinado a retirar todos os estilos padrão pré-configuradas a partir de um elemento ou elementos para permitir a assumir a aparência de outros elementos (por exemplo, ao estilo de um span como uma caixa de texto). Aparição foi proposto para CSS3, mas não inserir a especificação do Módulo de UI, embora não está sendo considerado para as iterações futuras da especificação. A propriedade tem o prefixo apoio no Webkit, e parcial no Firefox. Suporte prefixo Por esse motivo, no momento de escrita deste, o código mostrado na listagem achatados 4-40 funcionará melhor em cromado.

Uma vez que você chegar passado as complexidades do gradiente de cores hsla (excelente para criar diferentes cores "avisador" luzes) o resto do código é simples: os botões de rádio são feitas maiores em foco e sobre a liberação de fundo de imagem do gradiente radial se move para a posição do botão.

Resumo

Neste capítulo você já procurou em algumas das maneiras CSS3 transições podem ser usados para melhorar os elementos da interface de usuário, a partir de formas de navegação do site. Há um número infindável de possibilidades, que deixo à vossa imaginação e experimentação.

Para todos a sua utilidade, transições são limitados por dois factos: dependem de algum tipo de ação do usuário para iniciar a eles e eles sempre animar entre dois pontos ou estados. Movendo Objetos em uma curva é impossível com as transições, como está tendo eles loop ou executar por si. Todos esses recursos são da competência da CSS animações módulo (a que me refiro como "animações keyframe" a mais distingui-los de transições). Vou discutir esse módulo no próximo capítulo.

****

**Capítulo 5**

**CSS3 Keyframe animações**

Para criar animações simples de elementos entre dois estados, CSS as transições são fáceis de implementar e utilizar, mas essa simplicidade vem com várias limitações significativas. Para criar animações mais complexas com um maior grau de controle que você precisa o CSS animações módulo. Vou consultar a animações criados pelo módulo de animações CSS como *animações keyframe* para distingui-los de transições.

Se você estiver mais familiarizado com ferramentas de animação como Flash ou provir de um vídeo ou filme de fundo, com base no tempo, "sem caixilho" abordagem que CSS animações keyframe utilização pode parecer um pouco Confuso no primeiro. Para garantir que todos começa com a mesma compreensão, vou definir keyframes e de interpolação, depois olhar para o modo como são implementados pelo módulo de animações CSS.

Keyframes e de Interpolação

Animação moderna herdou a terminologia e processos dos tradicionais desenhados à mão, animação cel-shaded desenvolvido pela Disney e outros animadores no iníciodo século xx. Após estudos de carácter, esboços, e um script foram concluídas, desenvolvimento de animação foi algo como:

1. Os principais quadros de uma seqüência foram elaborados por uma supervisão animador. (No Disney Studios, este teria sido provavelmente um dos "nove homens velhos", master animadores que criou a criação de fotografias.) em um recurso como o *passeido Dumbo*, por exemplo, uma seqüência na qual passeido Dumbo tenta voar pelo marulhar seus ouvidos podem incluir dois *keyframes*: uma estrutura com orelhas do passeido Dumbo, os outros com os seus ouvidos para baixo.
2. Confinar a animação apenas para estes dois quadros, contudo, teria feito a seqüência um aspecto extremamente trêmulo (ou passeido Dumbo aparecem para voar como um hummingbird). Para criar uma sequência de animação mais suave, o keyframes foram entregues a um "inbetweener", um animador de nível inferior que gostaria de chamar os quadros intermediários necessários, utilizando o primeiro e o último keyframe como referência. Este processo se tornou conhecido como *interpolação.*
3. A sequência completa seria tintadas,, e alinhados. colorizado Reproduzidos em 24 fotogramas por segundo, a transição entre cada desenho parece fluido, criando uma animação contínua.

Hoje você é o mestre animador, com o navegador assumindo o papel de inbetweener. Criando um bom CSS3 animação é uma questão de fornecer o navegador com completa keyframes com informações suficientes para tween suavemente entre eles. O navegador é obrigado a fazer uma série de hipóteses quando de interpolação. Animações pobres são geralmente o resultado de não fornecer a folha de estilos com informações suficientes sobre os elementos ou fazer suposições que são contrárias a esses construídos em CSS3.

■ **Nota**reprodução suave de sequências de animação ocorre se um novo quadro é mostrado aproximadamente a cada

50 milissegundos, dependendo da velocidade de movimento no ecrã. Mantendo os princípios do CSS, as animações de Módulo não tentativa de definir a taxa de reprodução ou o número de fotogramas apresentados por segundo (FPS). Todos os CSS animação é definida pelos estados (para e a partir de um estado) ou quanto tempo uma sequência leva (tempo antes e depois de um tempo de estado). O resto é deixado para o navegador ou cliente. Você pode otimizar sua declarações CSS para reduzir a carga sobre o navegador e criar uma animação mais eficiente, mas você não criar as sequências "fotograma a fotograma" (com a possível excepção da etapa transições, discutido no Capítulo 3), e você não pode definir uma frequência de fotogramas.

CSS3 Animação Keyframe sintaxe

Uma animação keyframe começa sempre com um nome de animação que, como com um valor de id, deve ser exclusivo. Se duas sequências de keyframe têm o mesmo nome, apenas o último será reconhecido. A animação própria sequência podem ser especificadas de duas formas. A primeira é como um de ... a declaração, como mostrado na lista 5-1.

**Lista 5-1.**Keyframes para um simples da esquerda para a direita de animação

@keyframes deslize { do { esquerda: 0; } {esquerda: 100%; }

}

A animação também pode ser especificado como percentagem no tempo, mostrado na lista 5-2.

**Lista 5-2.**Vários fotogramas para um complexo de animação

@keyframes multislide { 0% { esquerda: 20px; }

20% { direita: 200px; }

80% { esquerda: 50px; }

100% { direito: 180px; }

}

Se você não definir explicitamente os estados de início ou de término (DE/0% ou a/100%) na sua declaração de animação, o navegador irá interpolar a partir de ou para os estados inicial ou final do elemento(s). Você também pode criar um híbrido de palavras-chave e valores na declaração, como mostrado na lista 5-3.

**Lista 5-3.**Uma seqüência de Keyframe com uma combinação de palavras-chave e valores percentuais

@keyframes multislide { 20% { direita: 200px; } 80% { esquerda: 50px; }

{ direito: 180px; }

}

Porque ele descreve somente dois estados, a animação CSS sintaxe mostrada na cotação de 5-1 produz um resultado que é essencialmente equivalente a uma transição, embora o método de animação ainda mantém várias vantagens sobre uma seqüência de transição, como você vai ver em breve. Praticamente, da keyframe as sequências podem ser escritos em qualquer lugar no seu CSS, mas recomendo que na maioria dos casos você mantê-los no topo da sua folha de estilos, juntamente com qualquer @font-face declarações, para fácil referência.

Você pode querer colocar sequências de muito longa na parte inferior da folha de estilos (para obter deles fora do caminho) ou mesmo mantê-los como um arquivo separado .css (trouxe em sua página através de @importação ou link < >), embora isso adiciona uma solicitação HTTP.

Você chamar a seqüência de animação keyframe CSS aplicando CSS separado propriedades para um elemento, mostrado na lista 5-4.

**Lista 5-4.**Como chamar uma seqüência de Animação Keyframe CSS

#redbox { background-color: vermelho;

Largura: 100px; altura: 150px; animação nome: deslize; animação duração: 5s; animação de função de distribuição: facilidade em;

}

Como você pode ver, estas propriedades (com excepção de animação-name), são muito semelhantes àqueles para transições introduzida no Capítulo 3, e eles têm a mesma função. Uma diferença é que a animação de duração pode ser definida para o infinito palavra-chave, em vez do tempo em segundos ou milissegundos. O módulo de animação também tem a propriedade de atraso de animação, e adiciona animação de contagem de iteração, animação direção animação-play-estado e animação-fill-modo.

A animação também pode ser chamado de animação em uma única propriedade de atalho, como mostrado na lista 5-5.

**Lista 5-5.**Como chamar uma seqüência de Animação Keyframe CSS com um atalho

#redbox { animação: deslize 5s facilidade no 2s; }

Os valores de animação pode ser declarada em qualquer ordem, com excepção da duração e valores de atraso, que deve ser indicado com duração primeiro e atraso na sequência.

Suporte para Animação Keyframe em browsers mais antigos

Os navegadores mais antigos exigem fornecedor prefixos, como já discutido para transições. Isto é complicado pelo fato de que o @keyframes declaração também precisa de prefixação, como mostrado na lista 5-6.

**Lista 5-6.**Como chamar uma seqüência de Animação Keyframe CSS para os mais velhos browsers webkit

@-webkit-keyframes multislide { 0% { esquerda: 20px; }

20% { direita: 200px; }

80% { esquerda: 50 %; }

100% { direito: 180px; }

}

#impusermos {

Largura: 100px; altura: 150px; -webkit-animação nome: multislide; -webkit-animação duração: 10,5s; -webkit-Animação de função de distribuição: facilidade em; -webkit-Animação de atraso: 200ms;

}

Capítulo 5 ■ CSS3 KEYFRAME animações

Esta repetição obviamente constrói como você escrever no suporte para outras versões mais antigas de navegadores; algumas das ferramentas discutidas no capítulo 10 pode facilitar e automatizar esse processo.

Controlar a reprodução de Animação Keyframe

Se você tentar reproduzir a animação que você criou, você verá que o elemento redefine para o seu estado inicial. Se você quiser que o elemento para parar no final da estrutura, você tem várias opções.

 Definir uma animação de longa duração para o elemento, deixando que assumem essencialmente uma eternidade para chegar à final da estrutura.

 Defina a direção de animação de modo que o elemento ventos no mesmo lugar.

 Definir o modo de preenchimento de animação para a frente.

A animação de modo de preenchimento é bastante estranhamente chamado, mas serve um propósito particular, mostrado na Tabela 5-1.

**Tabela 5-1.***Modo de preenchimento de animação valores de propriedade*

|  |  |
| --- | --- |
| **Modo de preenchimento** | **Descrição** |
|  |  |
| Para a frente | Posição do elemento termina em estrutura final. |
| Para trás | Retorna para a posição de elemento primeiro quadro. |
| Ambos os | Define o elemento para a posição do primeiro keyframe imediatamente no carregamento da página, independentemente |
|  | O local padrão do elemento em outras CSS. Aplicável apenas se a animação de atraso tem um |
|  | Valor superior a 0. |
| Nenhum | Elemento começa e termina na posição inicial padrão fornecido pelo CSS fora de keyframes. |
|  |  |

Você também poderá retornar seqüências de animação usando animação de direção, como mostrado na Tabela 5-2.

**Tabela 5-2.***Direção de animação valores de propriedade*

|  |  |
| --- | --- |
| **Direção de animação** | **Descrição** |
|  |  |
| Normal | Animação reproduz normalmente para a frente |
| Membro suplente | Animação reproduz para a frente depois inverte, voltando à sua posição inicial |
| Marcha à ré | Animação Reproduz para trás |
| Suplente-ré | Animação Reproduz para trás no primeiro abre várias, inverte para |
|  | A reprodução normal no segundo |
|  |  |

Mistura e encadeamento animações Keyframe

Mesclando várias animações keyframe é muito semelhante à fusão de transições, como mostrado na listagem 7.

**A lista de 5 a 7.**Combinando vários keyframe animações sobre um elemento

@keyframes lefttoright { 0% { esquerda: 0; }

100% { esquerda: 800px; }

}

@keyframes toptobottom { 0% { superior: 0; }

25% { topo: 100px; }

50% { superior: 0; }

75% { topo: 100px; }

100% { superior: 0; }

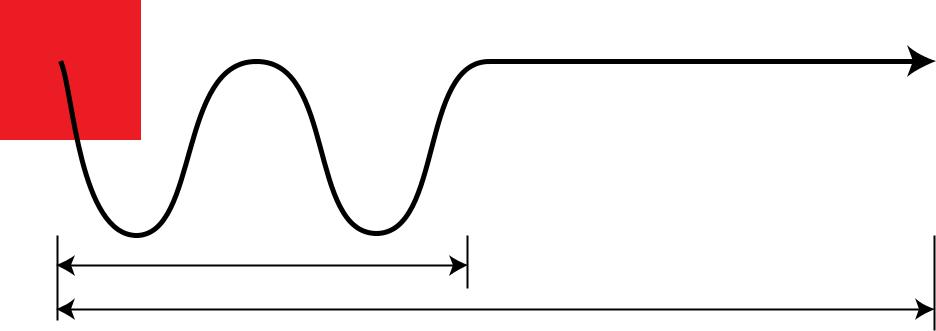
}

#Caixa { background: vermelho; largura: 100px; altura: 100px; POSIÇÃO: absoluta; animação nome: lefttoright, toptobottom;

Animação duração: 4s, 2s; animação de função de distribuição: facilidade de facilidade em;

}

O plotados resultado da combinação de estas duas sequências de animação é mostrado na Figura 5-1.



3 seg.

6 seg.

**Figura 5-1.***Uma entidade resultante da CSS3 animação*

Esta animação pode ser alterada mantendo a mesma keyframes mas alterando sua duração, como mostrado na lista 5-8.

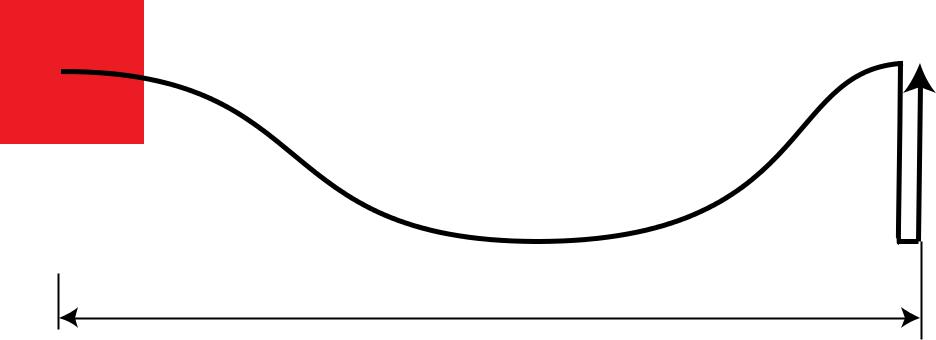
**Lista 5-8.**Uma entidade resultante da CSS3 Animação alteradas alterando as durações Keyframe

#Caixa { background: vermelho; largura: 100px; altura: 100px; POSIÇÃO: absoluta; animação nome: lefttoright, toptobottom;

Animação duração: 3s, 6s; animação de função de distribuição: facilidade de facilidade em; animação-fill-mode: ambos;

}

O resultado está esquematizada na figura 5-2.



6 seg.

**Figura 5-2.***Movimento de um elemento após a alteração dos valores de duração de animação*

Também é possível "corrente" sequências de animação keyframe através da introdução de um atraso entre eles, mostrado na lista 5-9 e Figura 5-3.

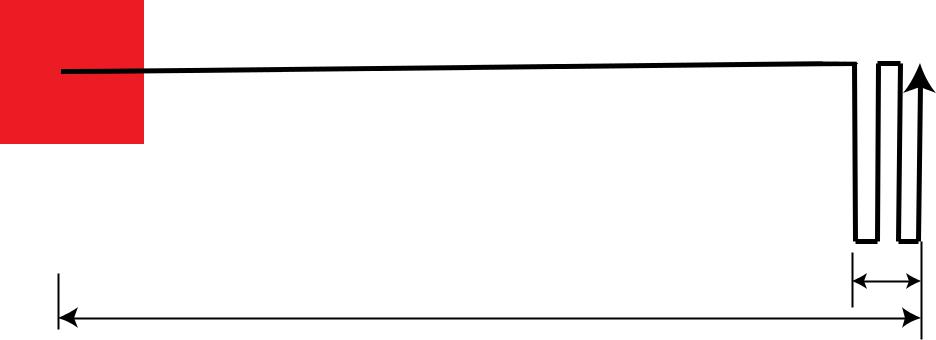
**Lista 5-9.**CSS3 encadeados Animação

#Caixa { background: vermelho; largura: 100px; altura: 100px; POSIÇÃO: absoluta; animação nome: lefttoright, toptobottom;

Animação duração: 3s, 6s; animação de função de distribuição: facilidade de facilidade em; animação de atraso: 0s, 3s;

Animação-fill-mode: ambos;

}



3 seg.

6 seg.

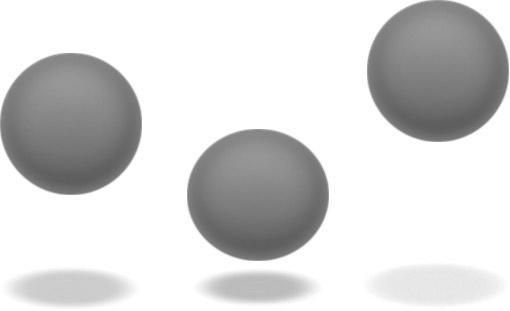
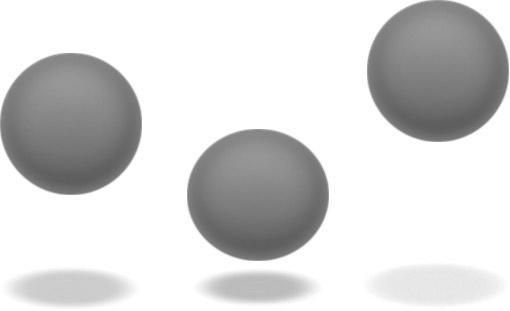
**Figura 5-3.***Movimento de um elemento depois de definir valores para a animação de atraso*

Você vai utilizar combinações de todos esses recursos no CSS3 Módulo de animação para criar animações complexas como os saltos bola descritas na próxima seção, e slideshows de imagem.

Repetir sequências de animação

Aumentar a contagem de iteração permite que seu animações para repetir várias vezes. Uma animação pode ser configurado para repetir vezes sem conta usando a palavra-chave infinita. Por exemplo, animação infinita poderia ser usado para criar uma interminável de saltar a esfera (ver Figura 5-4).

Para criar uma abordagem realista a esfera de borracha que você precisa para realizar vários efeitos visuais ao mesmo tempo: como a bola em contato com a superfície teórica sob ela, deve "esmagar" ligeiramente antes de recolhê (usando os princípios de squash e estique discutido no Capítulo 4) enquanto a sombra debaixo dele se torna mais difusa e se move mais longe do ponto de impacto como a bola sobe no ar. Para adicionar realismo extra, você pode inserir um desagravamento funcionar para determinados keyframes. (Esta animação foi derivado do trabalho original por CoDrops em <http://tympanus.net/codrops/2012/05/22/creating-an-animated-3d-bouncing-ball-with-css3/>,

Usado com permissão.) Consulte a listagem d

**Figura 5-4.***Capturas de um animado saltando seqüência de es*

**Listando 5-10.**CSS para um repetir indefinidamente balançar bola usando Squash e Stretch princípios de animação

@keyframes ballbounce {

@keyframes ballbounce {

0% { top: 0;

animation-timing-function: ease-in; }

50% { top: 140px; height: 140px;

animation-timing-function: ease-out;}

55% { top: 160px; height: 120px; border-radius: 50 % / 60px;

animation-timing-function: ease-in;}

65% { top: 120px; height: 140px; border-radius: 50 %;

animation-timing-function: ease-out;}

95% { top: 0; animation-timing-function: ease-in;}

100% { top: 0;animation-timing-function: ease-in;}

}

@keyframes shadowshrink {

0% { bottom: 0;

margin-left: -30px;width: 60px;height: 75px;

background: rgba(20, 20, 20, .1);

box-shadow: 0px 0 20px 35px rgba(20,20,20,.1);

border-radius: 30px / 40px;

animation-timing-function: ease-in;}Capítulo 5 ■ CSS3 KEYFRAME animações

50% {inferior: 30px;

Margem esquerda: -10px;largura: 20px;altura: 5px; fundo: rgba(20, 20, 20, .3);

Caixa-sombra: 0px 0 20px 35px rgba(20,20,20,.3); border-radius: 20px / 20px; animação de função de distribuição: facilidade;}

100% { parte inferior: 0; margin-left: -30px; largura: 60px; altura: 75px;

Fundamento: rgba(20, 20, 20, .1);

Caixa-sombra: 0px 0 20px 35px rgba(20,20,20,.1); border-radius: 30px / 40px; animação de função de distribuição: facilidade em;}

}

#bola {

Largura: 140px; altura: 140px; border-radius: 70px;

Fundamento: rgb(187,187,187); Fundo: linear-gradiente(para baixo

Rgba(187,187,187,1) 0%,rgba(119,119,119,1) 99 %); caixa-sombra: inserção 0 -5px 15px rgba(255,255,255,0.4),

Inserção -2px -1px 40px rgba(0,0,0,0.4), 0 0 1px #000; animação: ballbounce 1s infinito;

}

#sombra {topo: 280px; largura: 60px; altura: 75px;

Caixa-sombra: 0px 0 20px 35px rgba(20,20,20,.1); border-radius: 30px / 40px;

Transformar: scaleY(.3);

Animação: shadowshrink 1s infinito;} #bola, #sombra { posição: absoluta; superior: 0; } #sombra { esquerda: 65px; }

#wrapper { posição: relativos; largura: 200px; margin: 40px auto; }

<div id wrapper =>

<div id = sombra > </div> <div id = bola > </div> </div>

Como você pode ver, mesclando e sincronização de várias sequências de animação CSS você pode criar complexo e realista o movimento em elementos HTML.

Pausando Keyframe animações

Animações podem ser pausados alterando o valor da animação-play propriedade do Estado. O código da Listagem de 5-9 e 5-10, você pode adicionar o código na cotação de 5-11.

**Listando 5-11.**Pausando um CSS3 Animação

#wrapper:foco #bola, #wrapper:foco #sombra { animação-play-estado: pausado;

}

Porque o valor padrão de animação-play-estado é executado, a animação será retomada quando o rato se move para fora a partir de pairar sobre o wrapper div. Como você verá no Capítulo 6, é inteiramente possível para definir a trabalhar e estados de pausa através de outros meios.

Resumo

CSS3 Animações utilizando keyframes proporcionar maiores oportunidades para animar elementos HTML em páginas da web com muito mais variações do que aqueles criados através de transições. Enquanto construção de prefixados suporte para os navegadores mais antigos podem ser um pouco trabalhoso processo, a sintaxe essencial torna CSS3 animações muito mais eficaz e divertido. Enquanto um certo número de propriedades compartilham semelhanças com o módulo de transições, a animação de modo de preenchimento

É muito diferente: enquanto elementos afetados por transições CSS3 normalmente voltarà seu estado inicial, é incomum para animações de pleno direito a fazê-lo, exigindo cuidados com tanto sentido de animação e animação-fill-modo.

No próximo capítulo, você vai usar a sintaxe que você aprendeu aqui para criar animação avançada de conteúdo da web, incluindo apresentações de imagem.

Pausando Keyframe animações

Animações podem ser pausados alterando o valor da animação-play propriedade do Estado. O código da Listagem de 5-9 e 5-10, você pode adicionar o código na cotação de 5-11.

**Listando 5-11.**Pausando um CSS3 Animação

#wrapper:foco #bola, #wrapper:foco #sombra { animação-play-estado: pausado;

}

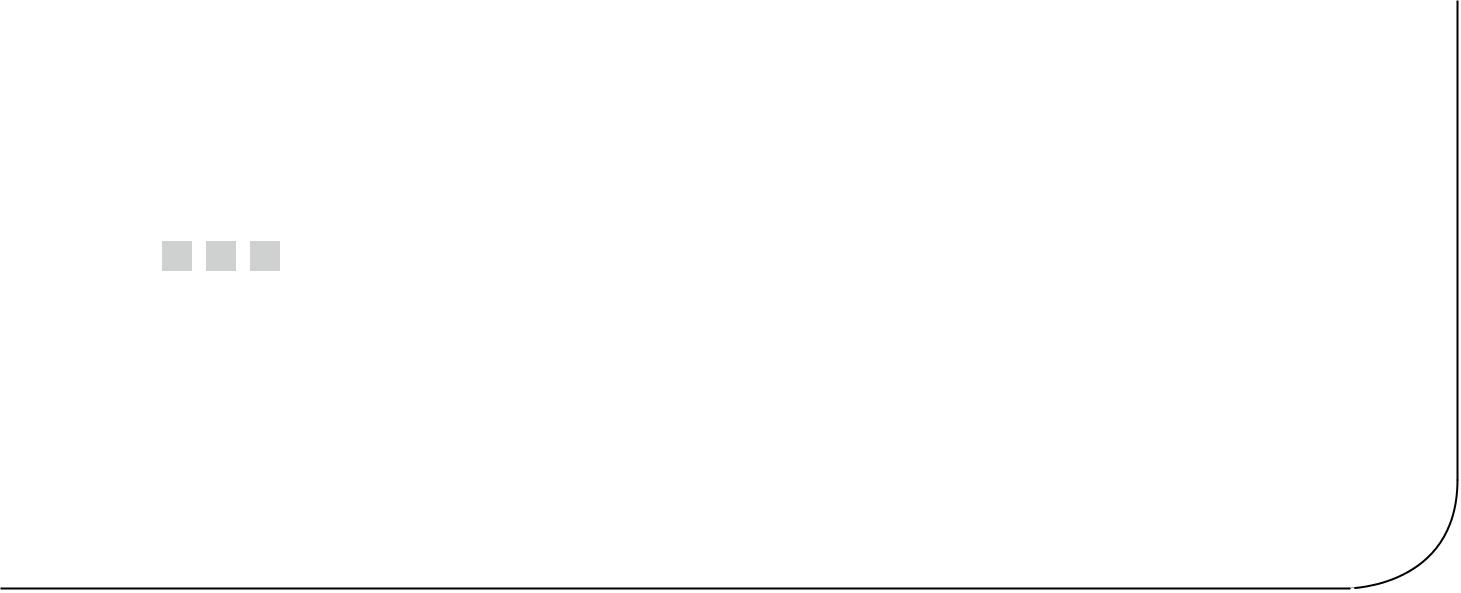
Porque o valor padrão de animação-play-estado é executado, a animação será retomada quando o rato se move para fora a partir de pairar sobre o wrapper div. Como você verá no Capítulo 6, é inteiramente possível para definir a trabalhar e estados de pausa através de outros meios.

Resumo

CSS3 Animações utilizando keyframes proporcionar maiores oportunidades para animar elementos HTML em páginas da web com muito mais variações do que aqueles criados através de transições. Enquanto construção de prefixados suporte para os navegadores mais antigos podem ser um pouco trabalhoso processo, a sintaxe essencial torna CSS3 animações muito mais eficaz e divertido. Enquanto um certo número de propriedades compartilham semelhanças com o módulo de transições, a animação de modo de preenchimento

É muito diferente: enquanto elementos afetados por transições CSS3 normalmente voltarà seu estado inicial, é incomum para animações de pleno direito a fazê-lo, exigindo cuidados com tanto sentido de animação e animação-fill-modo.

No próximo capítulo, você vai usar a sintaxe que você aprendeu aqui para criar animação avançada de conteúdo da web, incluindo apresentações de imagem.

**Capítulo 6**

**CSS3 Keyframe animações para**

**Conteúdo da Web**

Como você viu no capítulo anterior, a sintaxe do CSS módulo de animação é mais poderoso do que a sintaxe de transições, permitindo muito mais controle de conteúdo da web através do uso de keyframes. Este capítulo irá ilustrar que a energia através de vários exemplos: um andar de diapositivos, uma galeria de imagens que duplica a aparência do clássico Lightbox plug-in para JavaScript e um logotipo animação.

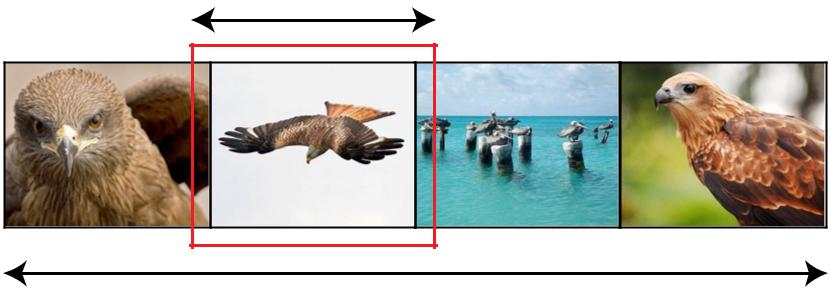
Um simples CSS3 Slides

Tradicionalmente, galeria de imagens efeitos de slides foram criados usando o Flash ou JavaScript, muitas vezes sob a forma de um quadro plug-in, como Nivo controle deslizante (<http://nivo.dev7studios.com>) e da câmara ([www.pixedelic.com/plugins/camera](http://www.pixedelic.com/plugins/camera)). Enquanto existem vantagens em usar um quadro/abordagem plug-in (principalmente na variedade de transições disponíveis), há uma série de vantagens em completar a sua apresentação de slides em CSS3 demasiado, como você verá em um momento.

O código HTML

Marcação das imagens para um CSS3 slides é um processo bastante simples: você colocar as imagens dentro de um contentor que se move em uma faixa através de outro elemento de "janela" que é do mesmo tamanho que cada imagem individual na faixa (ver Figura 6-1).

500px



2000px

**Figura 6-1.***Organização de um CSS slideshow (vermelho "janela" recipiente elaborado maior para ênfase)*

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

No código, este é criado usando a lista 6-1, para que mais markup será adicionado em mais exemplos,

**Lista 6-1.**Código HTML para uma base CSS3 Slides

<div id=diapositivos>

<figura id=imagestrip>

<img src=preto-kite.jpg alt="Fotografia de um preto kite"> <img src=red-kite.jpg alt="Perfil de um vermelho kite">

<img src=pelicanos.jpg alt="Pelicanos no moorings no mar"> <img src=pária-kite.jpg alt="Fotografia de pária kite">

</figura>

</div>

Neste exemplo, você tem quatro imagens em 400 pixels de altura por 500 pixels de largura, de modo que a largura total do seu recipiente interior deve ser 2000 pixels. (Observe que todas as imagens devem ser do mesmo tamanho; as imagens no exemplo são mostrados por Challiyil Eswaramangalath Pavithran Vipin, Ariful H Bhuiyan, Márcio Cabral de Moura, e Alan Saunders, licenciado sob Creative Commons e usadas com permissão).

O CSS Base, mostrado na lista 6-2 é igualmente simples:

**Lista 6-2.**Código CSS Base para um CSS3 Galeria de imagens da apresentação de diapositivos

Div#slideshow { posição: relativos; fundo: #000; transbordamento: hidden; } figura#imagestrip, div#slideshow { de dimensionamento de caixa: caixa de fronteira; }

Div#diapositivos, figura#imagestrip img { Largura: 500px; altura: 400px; bóia: esquerda; } figura#imagestrip { posição: absoluta; largura: 2000px; margin: 0; }

Para criar o controle deslizante de animação mais simples você deve mover o interior figura horizontalmente em incrementos de 500 pixels, com uma pausa após cada movimento para permitir que o visualizador de tempo para apreciar cada imagem. Como discutido no capítulo anterior, CSS3 animações não funcionam de maneira explícita e quadros. Você deve tratar a animação como uma divisão de tempo: para cada imagem que você designar uma quantidade igual de tempo durante o qual será ainda com o restante do tempo designado para movimento. Neste exemplo, cada imagem será ainda para 20% do tempo e a faixa como um todo será em movimento para 20% do tempo, dividida em quatro sequências. Assim cada movimento intersticial terá 5% do tempo global.

A parte importante para se lembrar quando escrever o keyframes é que as propriedades que você altere *deve* estar presente como um valor definido entre vários keyframes se você deseja mantê-los o mesmo; do contrário, o navegador irá reverter para uma interpolação usando os valores que você não deseja usar.

Sua primeira tentativa na declaração keyframe pode parecer algo como Lista 6-3.

**Lista 6-3.**Keyframes para um controle deslizante de imagem simples

@keyframes controle deslizante {

0% { transformar: translateX(0px); }

20% { transformar: translateX(0px); }

25% { transformar: translateX(-500px); }

45% { transformar: translateX(-500px); }

50% { transformar: translateX(-1000px); }

70% { transformar: translateX(-1000px); }

75% { transformar: translateX(-1500px); }

95% { transformar: translateX(-1500px); }

100% { transformar: translateX(-2000px); }

}

Então você chamar a seqüência de animação. Ao contrário de transições, animações keyframe não exigem um início do evento, o que significa que esta animação será executado no carregamento da página, como mostrado na lista 6-4.

**Lista 6-4.**Keyframes para um controle deslizante de imagem simples

Figura#imagestrip { animação: Cursor 10s infinito; }

Você verá que isso funciona, mas há apenas um problema: o último movimento no final revela uma janela vazia, como não há nenhuma imagem para além de 2000 pixels para a faixa de imagem para mostrar. Você corrigir este através da introdução de um pouco de fazer batota, colocando uma cópia da primeira imagem no final da faixa, alongar a faixa como um todo para 2500 pixels na nossa lista de CSS (6-5).

**Lista 6-5.**Modificado o código HTML para uma base CSS3 Slides

<div id=diapositivos>

<figura id=imagestrip>

<img src=preto-kite.jpg alt="Fotografia de um preto kite"> <img src=red-kite.jpg alt="Perfil de um vermelho kite">

<img src=pelicanos.jpg alt="Pelicanos no moorings no mar"> <img src=pária-kite.jpg alt="Fotografia de pária kite"> <img src=preto-kite.jpg alt="Fotografia de um preto kite">

</figura>

</div>

Como o último keyframe agora visualmente corresponde à primeira, isto cria uma seqüência de animação suave completa.

Uma variação com imagens de plano de fundo

Você pode criar um efeito equivalente usando apenas imagens de fundo - por exemplo, para criar uma apresentação de slides atrás de um elemento de banner ou logotipo. Isto irá simplificar o seu código para um único <figura>, mas vai complicar o seu CSS um pouco. (Observe que você vai repetir a primeira imagem no final da lista de planos de fundo apenas como você fez nos exemplos acima, e pela mesma razão. Consulte a listagem de 6-6).

**Lista 6-6.**Código HTML e CSS para um controle deslizante usando imagens de plano de fundo

<style>

Figura#imagestrip { Largura: 500px; altura: 400px;

Fundamento: #000; caixa de calibragem: caixa de fronteira; **** background-image: url(preto-kite.jpg), url(vermelho-kite.jpg), url(pelicanos.jpg),

Url(pária-kite.jpg), url(preto-kite.jpg); fundo-repetir: não-repetição;

Posição de fundo-x: 0, 500px, 1000px, 1500px, 2000px; animação: Cursor 20s infinito;

}

@keyframes controle deslizante {

0% { background-Posição-x: 0, 500px, 1000px, 1500px, 2000px; } 20% { background-Posição-x: 0, 500px, 1000px, 1500px, 2000px; }

25% { background-Posição-x: -500px, 0px, 500px, 1000px, 1500px, 2000px; } 45% { background-Posição-x: -500px, 0px, 500px, 1000px, 1500px, 2000px; } 50% { background-Posição-x: - Milhares px, -500px, 0px, 500px, 1000px, 1500px; } 70% { background-Posição-x: - Milhares px, -500px, 0px, 500px, 1000px, 1500px; } 75% { background-Posição-x: -1500px, - Milhares px, -500px, 0px, 500px, 1000px; }

95% { background-Posição-x: -1500px, - Milhares px, -500px, 0px, 500px, 1000px; } 100% { background-Posição-x: -2000px, -1500px, - Milhares px, -500px, 0px, 500px; }

}

</Estilo>

<figura id=imagestrip></figura>

Você também pode criar um único elemento de recipiente com várias imagens de fundo e mova-o visível através de uma "janela", ou fazer uma única imagem para o fundo que une todas as imagens. Esta abordagem permitirá o CSS mais fácil, mas também tornará substancialmente mais difícil de mudar a galeria mais tarde.

Pausar a apresentação de slides

É razoável para permitir que o usuário pause o controle deslizante para se concentrar em uma imagem. A maneira mais fácil de iniciar uma acção desse tipo é um passe o mouse sobre a barra deslizante em si; ao mesmo tempo, você deve colocar um identificador visual sobre a apresentação de slides para indicar claramente que ele está em um estado de pausa. Para este exemplo, vou fazê-lo por fading o elemento imagestrip e colocação de um texto substituto para um ícone de pausa na tela como um pseudo-elemento (Eu também poderia usar um arquivo de imagem). Todas as alterações são complementos para o CSS como mostrado na listagem 6-7.

**Lista 6-7.**O código CSS para pausar uma apresentação de slides no passe Galeria de Imagens

Div#slides:foco figura#imagestrip { animação-play-estado:pausado; opacidade: 0,5; } div#slides:foco:antes {

Conteúdo: "❚❚"; font-size: 200px; color: rgba(255,255,255, 0,7); posição: absoluta;

Esquerda: 160px; superior: 80px;

}

Observe que você pode facilitar no visual identificadores de estado pausado colocando uma transição adequada no interior do elemento figura, tais como transição: 1s Opacidade linear.

Alterando a transição entre as imagens

Afinal existem muitos tipos possíveis de desbotar e panos entre cada imagem em sua galeria deslizante, alguns dos quais são mostrados ao lado.

Fade-In-Out

Depois de uma simples horizontal ou vertical de crawl, a transição mais comum para um controle deslizante é ter cada imagem Fade to/ de preto em seqüência. Sem alterar o markup, você poderia desaparecer a faixa de imagem, faixa de mover a imagem "sob cobertura de escuridão" enquanto é preto e apagam a faixa na novamente como um todo, como mostrado na lista 6-8.

**Lista 6-8.**O código CSS para pausar uma apresentação de slides no passe Galeria de Imagens

@keyframes controle deslizante {

0% { transformar: translateX(0); }

20% { opacidade: 1; }

22% { opacidade: 0; transformar: translateX(0); }

23% { opacidade: 0; transformar: translateX(-500px); } 25%, 45% { opacidade: 1; }

47% { opacidade: 0; transformar: translateX(-500px); }

48% { opacidade: 0; transformar: translateX(-1000px); } 50%, 70% { opacidade: 1; }

72% { opacidade: 0; transformar: translateX(-1000px); }

73% { opacidade: 0; transformar: translateX(-1500px); } 75%, 95% { opacidade: 1; }

97% { opacidade: 0; transformar: translateX(-1500px); }

98% { opacidade: 0; transformar: translateX(-2000px); }

100% { opacidade: 1; transformar: translateX(-2000px); }

}

Note que estou agrupamento keyframes que têm as mesmas propriedades da mesma maneira que você faria grupo seletores de CSS ordinário. No entanto, normalmente você deseja fazer a chamada sobre a animação mais para evitar que apareçam apressado. Eu gostaria de sugerir o uso de animação: Cursor 20s infinita.

Porque as outras imagens na faixa não são visíveis durante o fade sequências, você pode descartar a imagem duplicada final na extremidade do controle deslizante div, executá-lo de volta para o início em completa escuridão antes de iniciar a animação novamente. Não vai ser possível no próximo exemplo.

Fade-In-Out durante o movimento

Você pode manter a impressão de um efeito deslizante com um fade por fading as imagens durante o seu movimento da esquerda para a direita. Enquanto ele é inteiramente possível para fazer isso em um único keyframe animação, é provavelmente mais fácil de criar o código como duas seqüências em execução simultaneamente.

Para isso, você reverter para o original com o controle deslizante seqüência keyframe, agora a trabalhar mais de trinta segundos e adicionar um novo fader keyframe declaração (ver lista 6-9).

**Lista 6-9.**O código CSS para pausar uma apresentação de slides no passe Galeria de Imagens

@keyframes controle deslizante {

0% { transformar: translateX(0px); }

20% { transformar: translateX(0px); }

25% { transformar: translateX(-500px); }

45% { transformar: translateX(-500px); }

50% { transformar: translateX(-1000px); }

70% { transformar: translateX(-1000px); }

75% { transformar: translateX(-1500px); }

95% { transformar: translateX(-1500px); }

100% { transformar: translateX(-2000px); }

}

@keyframes fader {

0% { opacidade: 1; }

70% { opacidade: 1; }

90% { opacidade: 0; facilidade; }

95% { opacidade: 0; }

100% { opacidade: 1; Facilidade em; }

}

Figura#imagestrip { largura: 2500px;

Animação: Cursor 30s infinita, fader 7.5s infinito; }

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

A filosofia por trás da distribuição é simples: com um comprimento total de animação de trinta segundos a cada imagem no slide irá aparecer totalmente no ecrã durante 6 segundos (20% do tempo de animação) e mova para a esquerda em 1,5 segundos (5% do total de tempo). Passando uma segunda animação que apaga o elemento imagestrip perto do fim mais de 7,5 segundos, você pode mesclar os dois para obter um resultado suave.

Mixagem

Para conseguir um efeito de mixagem você tem três opções, mas em todos os casos as imagens já não estão colocados como um "strip", mas empilhados um em cima do outro, com o superior imagem desbotada em volta.

A primeira e mais fácil opção é simplesmente definir cada imagem de plano de fundo como um keyframe para um caso contrário vazio (mas corretamente dimensionado) elemento (consulte a listagem 6-10).

**Listando 6-10.**O código CSS para um controle deslizante de imagem mixagem

@keyframes imageswap {

0% { background-image: url(preto-kite.jpg); }

20% { background-image: url(vermelho-kite.jpg); }

40% { background-image: url(pelicanos.jpg); }

80% { background-image: url(pária-kite.jpg); }

100% { background-image: url(preto-kite.jpg); }

}

Isto irá simplesmente e facilmente mixagem entre as imagens; no entanto, ele não pode dar a você os resultados ou grau de controle você depois. Uma abordagem alternativa é trazer as imagens para a página usando o filtro de mixagem, como mostrado na listagem 6-11.

**Listando 6-11.**Alternativa para um código CSS Mixagem deslizante de imagem

@keyframes controle deslizante {

0% { background-image: url(preto-kite.jpg); }

20% { background-image: cross-fade(url(preto-kite.jpg), url(vermelho-kite.jpg),0%); } 25% { background-image: cross-fade(url(preto-kite.jpg), url(vermelho-kite.jpg),100%); } 45% { background-image: cross-fade(url(vermelho-kite.jpg), url(pelicanos.jpg),0%); }

50% { background-image: cross-fade(url(vermelho-kite.jpg), url(pelicanos.jpg),100%); } 70% { background-image: cross-fade(url(pelicanos.jpg), url(pária-kite.jpg),0%); } 75% { background-image: cross-fade(url(pelicanos.jpg), url(pária-kite.jpg),100%); } 95% { background-image: cross-fade(url(pária-kite.jpg), url(preto-kite.jpg),0%); }

100% { background-image: cross-fade(url(pária-kite.jpg), url(preto-kite.jpg),100%); }

}

A terceira opção, usando imagens "real", é ligeiramente mais complexo: o nível superior da imagem deve ser acabada, atrasada antes de retornar para o primeiro plano. (penso de um baralho de cartas, com a transição entre a placa na parte superior e um baixo antes de o primeiro cartão é colocado em um monte de descarte para um período de tempo.) O CSS se torna o que é mostrado na lista 6-12.

**Listando 6-12.**Terceira Opção o código CSS para um controle deslizante de Imagem Mixagem

@keyframes controle deslizante {

0%, 25% { opacidade: 1; } 30%, 100% { opacidade: 0; }

}

Figura#imagestrip {

Largura: 500px; altura: 400px;

Fundamento: #000; caixa de calibragem: caixa de fronteira; transbordamento: oculto; POSIÇÃO: relativos;

}

Figura#imagestrip img { posição: absoluta; superior: 0; esquerda: 0; }

O HTML também muda; nota que efectuei as imagens na  ordem inversa no novo código (listando 6-13). Posicionado absolutamente, a última imagem na figura será no topo. (alternativamente, você poderia colocar um índice z inline a propriedade sobre cada).

**Listando 6-13.**Código HTML para terceira opção Mixagem deslizante de imagem

<figura id=imagestrip>

<img src=preto-kite.jpg alt="Fotografia de um preto kite">

<img src=pária-kite.jpg alt="Fotografia de um pária kite" style="animação: Cursor 10s 7.5s infinito;">

<img src=pelicanos.jpg alt="Pelicanos no moorings no mar "style="animação: Cursor 10s 5s infinito;">

<img src=red-kite.jpg alt="Fotografia de um vermelho kite" style="animação: Cursor 10s 2.5s infinito;">

<img src=preto-kite.jpg alt=" Fotografia de um preto kite" style="animação: Cursor 10.1s 0s infinito;">

</figura>

Como cada imagem na seqüência keyframe é "sólido" para um quarto do comprimento da animação e a animação como um todo é de dez segundos de duração, cada chamada subseqüente da keyframe é adiada por um período suplementar de um quarto do tempo total. O prazo suplementar concedido à primeira passagem (animação da black kite) é fornecida de modo que o retorno da fotografia não "passo" a desaparecer no seu suplente no início da figura.

Adição de legendas

Você pode adicionar legendas para a apresentação de slides em muito da mesma forma que você fez no exemplos anteriores com transições de imagem. A margem de lucro passa a ser mostrado na lista 6-14.

**Listando 6-14.**Código HTML para um controle deslizante de Imagem com legendas

<div id=diapositivos> <figura id=imagestrip>

<figura>

<img src=preto-kite.jpg alt="Black kite"> <figcaption>Preto kite</figcaption>

</figura>

<figura>

<img src=red-kite.jpg alt="Red kite"> <figcaption>Vermelho kite</figcaption>

</figura>

<figura>

<img src=pelicanos.jpg alt=Pelicanos> <figcaption>Pelicanos</figcaption>

</figura>

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

<figura>

<img src=pária-kite.jpg alt="Pária kite"> <figcaption>Pária kite</figcaption>

</figura>

</figura>

</div>

À COTAÇÃO 6-2 e 6-3 você adicionar o markup mostrado na listagem 6-15.

**Listando 6-15.**O código CSS para adicionar legendas a um controle deslizante de imagem

Figura#imagestrip figura figcaption {

Posição: absoluta; antecedentes: rgba(0,0,0,0.4); color: #fff; largura: 500px; padding: 8px; font-size: 18px; superior: -42px;

Transição: 1s topo linear;

}

Figura#imagestrip:foco { animação-play-estado:pausado; } figura#imagestrip figura:foco figcaption { superior: 0; }

Um método para Pausa On-Click

Você também pode usar o método de rótulo discutido anteriormente para criar um método alternativo para pausar a apresentação de slides animação adicionando o código mostrado na listagem 6-16 apenas após a abertura <div> juntamente com o associado CSS.

**Listando 6-16.**Código HTML e CSS para adicionar uma pausa para uma imagem On-Click Controle deslizante

<tipo de entrada de seleção=id=Pausar><etiqueta para=Pausar></label>

Rótulo {

Visor: bloco; índice z: 24; transição: for superior a todas as facilidade em;

}

Entrada:verificado ~ figura#imagestrip { animação-play-estado:pausado;

}

Pausa de entrada#:verificado ~ rótulo { background: rgba(0,0,0,0.4);

}

Pausa de entrada#:verificado ~ etiqueta:antes { conteúdo: "❚❚"; font-size: 200px; color: rgba(255,255,255, 0,5); posição: relativos;

Esquerda: 160px; superior: 80px;

}

Enquanto o acima é uma opção, mantenha em mente a acessibilidade questões discutidas anteriormente neste capítulo se você optar por empregar.

Criando um fallback para versões mais antigas do Internet Explorer

O transbordamento: hidden parte da declaração será lido e seguido pelo Internet Explorer (IE) 9 e versões anteriores, embora a animação keyframe não. Este irão obscurecer as outras imagens, o que significa que os utilizadores do IE antes da versão 10 não vê-los. Isto pode ser evitado com um comentário condicional que proporciona a visibilidade das imagens para os usuários como mostrado na listagem 6-17. (Observe que o IE 6, 7 e 8 exigirá um pouco mais de JavaScript e CSS para os navegadores para reconhecer elementos HTML5, tais como <figura>, discutido no Capítulo 9).

**Listando 6-17.**CSS condicional para tornar as imagens visíveis em versões anteriores do IE

<!--[if lte IE 9]>

Div#slideshow { transbordamento: visível; } <![endif]-->

Uma advertência contra usando Marquee a animação para Texto

O <Marquee> tag tem uma longa e bastante triste história, indo todo o caminho de volta para as primeiras versões do Internet Explorer. Nunca um elemento padronizado nem uma parte de qualquer especificação HTML, <Marquee> foi comumente usado para criar uma rolagem "ticker tape" efeitos para texto em páginas da web durante o final da década de noventa. Juntamente com GIFs animados e texto piscando, <Marquee>, excepto em contextos muito específicos, se tornou uma das marcas de bad web design.

Enquanto os ingénuos design podem ser tentados a utilizar estas técnicas como forma de animar um texto crawl, deveriam ser evitados. As tendências de design de lado, Marquee text tem um número de problemas de usabilidade:

 O sistema visual humano é atraído para circulação e Marquee text está em constante movimento; um retângulo recurso pode ser extremamente distraente em uma página.

 Pela mesma razão, Marquee text pode ser muito difícil de ler, especialmente para usuários com deficiência visual.

 Incluindo links em Marquee text faz uma má ideia pior: links pode ser muito difícil de seguir e clique. Como Marquee text se move em um loop, faltando um link significa que o utilizador tem de esperar até o próximo aspecto do link se perca a primeira vez, que pode ser extremamente frustrante. Por esse motivo, quaisquer links importantes que aparecem em um retângulo também deve aparecer em uma forma estática na página da web.

News Ticker/Animação de notificação

Em vez de usar marquee para animar texto, nesta seção você vai criar um News Ticker seqüência de alertas com CSS3 animação. Cada nova notificação será pilha no canto inferior direito da página, exibindo durante um determinado período de tempo antes de desaparecer. O usuário deve ser capaz de clicar em cada notificação para obter mais informações e o tempo restante para cada painel será mostrado em uma barra de progresso. (Consulte a Figura 6-2.)



**Figura 6-2.***News Ticker conduzido com CSS3*

A margem de lucro para este é bastante simples: as notificações são div elementos contidos em um maior div, com cada barra de progresso como uma div com um span dentro dele, como mostrado na listagem 6-18.

**Aquilo sobre o elemento de progresso?**

HTML5 tem markup para exibir a hora de um processo: o  elemento <progressos>. Enquanto você pode modelar o elemento de progresso da maneira que você deseja, você não pode usar o CSS para modificar o progresso demonstrado visualmente na tag (JavaScript). Uma tal tarefa está fora do âmbito deste capítulo, como estamos usando CSS3 para cada recurso sempre que possível, de modo que ele não é um elemento adequado para usar neste caso.

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

**Listando 6-18.**HTML para um CSS3-Sistema de Notificação temporizada Acionada

<div id=quebra-news>

<div class=Notificação><a href=#><span>☂</span>Chuva esperada</a> <div class=progressos><span></span></div>

</div>

<div class=Notificação><a href=#><span>**✈**</span>planos de viagem alterada</a> <div class=progressos><span></span></div>

</div>

<div class=Notificação><a href=#><span>❄</span>neve leve</a> <div class=progressos><span></span></div>

</div>

</div>

O estilo básico de CSS para as notificações é mostrado na lista 6-19.

**Listando 6-19.**CSS para uma seqüência de alerta de notificação

Div#quebra-news {

Posição: fixo; fundo: -20px; direita: 15%;

}

Div.Notificação {

Posição: relativos; largura: 275px; border-radius: 10px; fundo: linear-gradiente(rgb(215,215,215), rgb(165,164,169); padding: 60px 20px 40px; fronteira: 2px solid #999;

Margem superior: 10px;

Caixa-sombra: 3px 3px 6px rgba(0,0,0,0.1) inset, 0 0 6px 2px rgba(0,0,0,0.1); opacidade: 0,9;

}

Div.Notificação {

Cor: branco; texto-curso: 1px solid #000;

Text-decoration: none; font-family: Futura, sans-serif; font-size: 20px;

}

Div.notificação um span {

Font-size: 60px; PADDING-RIGHT: 20px; alinhar vertical: médio;

}

Div.progress {

Altura: 5px; border-radius: 2px; fronteira: 1px solid #999; margin-top: 32px; fundo: rgb(215,215,215);

}

Div.progress span {

Fundamento: #000; exibição: bloco; largura: 0; altura: 3px;

}

Existem três sequências de animação: popup, que aciona cada notificação para cima; progresso, para mostrar o tempo restante; e o fade, para tornar cada notificação desaparecer no final. Cada um destes são mostrados na lista 6-20.

95

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

**Listando 6-20.**CSS para uma seqüência de alerta de notificação

@keyframes popup {

0%, 30% { altura: 0; padding: 0 20px; DISPLAY: none; }

}

@keyframes fade {

100% { opacidade: 0; }

}

@keyframes progressos {

100% { largura: 100%; }

}

Observe que você está tomando uma abordagem ligeiramente diferente com estas sequências keyframe: porque os estados padrão para os elementos já foram definidos na cotação de 6-20, você está usando somente o keyframes para definir a *partir de* estado (no caso de pop-up) ou o *para* estado (no caso de progresso e fade). O navegador será automaticamente tween entre esses valores e o padrão incorporado, inline, e estilos ligados conforme apropriado. É apenas quando você incluir valores tanto para 0% e 100% em seu @keyframes declaração de que você controle toda a aparência do elemento (considerações de animação do modo de preenchimento de terras).

Todos os pop-ups de notificação compartilham a mesma animação, com excepção do atraso antes de a animação que é inicializada para cada pop-up. Trabalhar o seu CSS eficaz significa que você deve colocar como muito semelhantes o CSS como possível em uma única declaração, como mostrado na listagem 6-21.

**Listando 6-21.**Sequências de Keyframe chamado a partir de uma declaração única partilhada

Div.Notificação {

...

Nome de animação: popup, fade; animação duração: 2s, 1s;

A função de distribuição de animação: metros cúbicos, 0.730 linhas vectoriais assim(0,325, 0,695, 1,650); animação-fill-mode: para trás, para frente;

Animação de atraso: 2s, 14s;

}

Sem quaisquer declarações contraditórias, cada painel notificação herdará todas da lista de estilos no 6-21, mas atraso de animação é a única coisa que você tem que mudar para cada (consulte a listagem 6-22).

**Listando 6-22.**Definição de diferentes valores para a subsequente notificação Animation-Delay Painéis

Div.notificação:nth-Child(2) { Atraso de animação: 6s, 18s;

}

Div.notificação:nth-Child(3) { Atraso de animação: 12s, 24s;

}

Você adoptar uma abordagem semelhante para a barra de progresso para cada painel lista (6-23).

**Listagem de 6-23.**Definição de diferentes valores Animation-Delay para Barras de Progresso

Div.progress span {

Fundamento: #000; exibição: bloco; largura: 0; altura: 3px; animação: progressos 12s 4s frente linear;

}

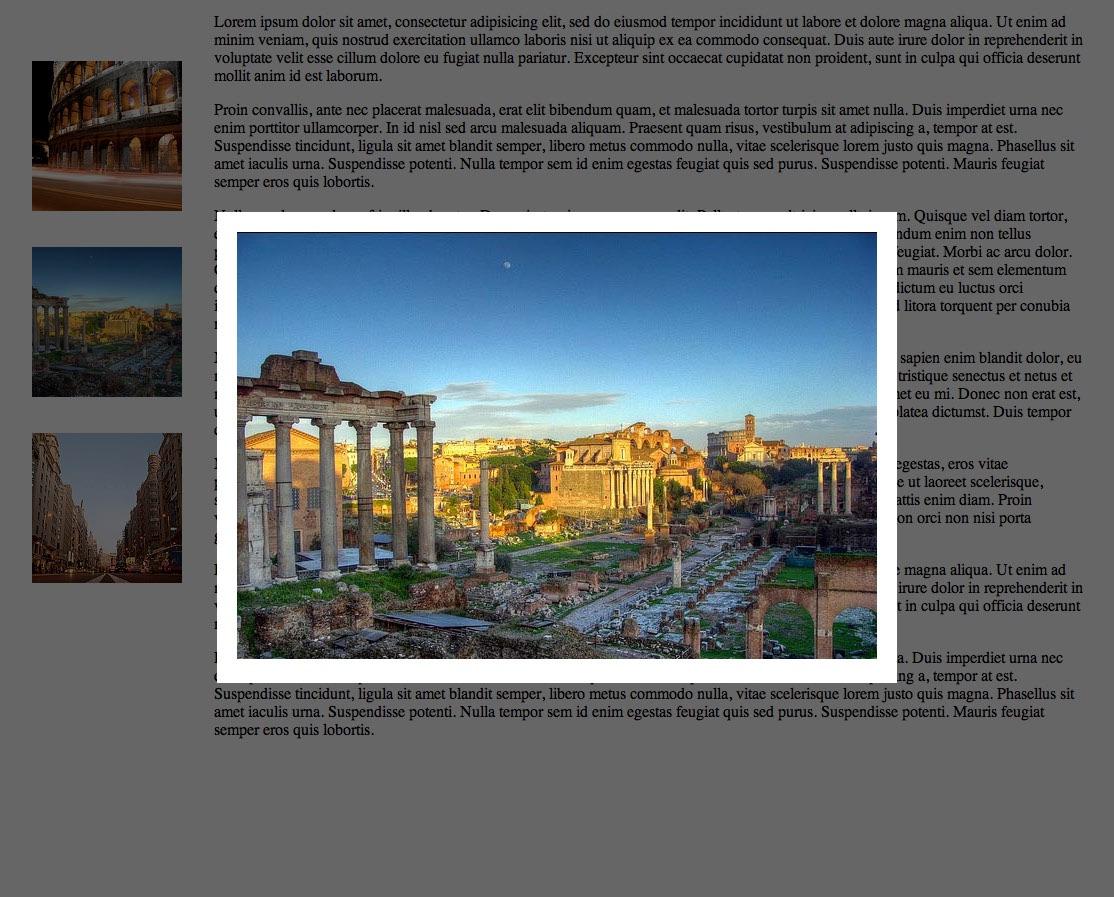
Div.notificação:nth-Child(2) div.progress span { Atraso de animação: 6s; } div.notificação:nth-Child(3) div.progress span { Atraso de animação: 12s; }

Naturalmente, no mundo real de mão moldando cada notificação em puro CSS3 levaria um grande esforço. Como você verá no Capítulo 9, você pode usar a base de trabalho aqui para integrar perfeitamente com os pontos fortes de JavaScript.

Um Lightbox Galeria de imagem equivalente em CSS3

*Lightbox*é um termo genérico para uma das primeiras técnicas de modal populares usado para exibir imagens da galeria: o clássico efeito Lightbox é um fade-out da página seguido de uma expansão e ao desbotamento da imagem no centro da página. A sua popularidade levou ao excesso na web, com muitos desenvolvedores simplesmente usando o padrão que veio com o código porque de desconhecimento ou preguiça. Por escrito o equivalente em CSS permite que o desenvolvedor para personalizar facilmente a aparência da galeria para seus requisitos.

Primeiro, você vai usar um muito semelhante markup (listando 6-24) ao que você usou para o primeiro exemplo de galeria no Capítulo 3. Neste momento você usará as imagens fornecidas por Robert Lowe, Jon Rawlinson e Camilo Rueda López, licenciado sob Creative Commons (Figura 6-3).



**Figura 6-3.***Grande imagem ampliada em um CSS3-Lightbox equivalente*

**Listando 6-24.**HTML para um Lightbox equivalente em CSS3

<body id=base> <dl id=gallery>

<dt><a href=#col1><img src=coliseu-noite-pequeno.jpeg alt="Coliseu à noite"></a> <dd id=col1><a href=#><img src=coliseu-noite.jpeg alt="Coliseu à noite" ></a> <dt><a href=#col2><img src=coliseu-forum-pequeno.jpeg alt="Coliseu e Fórum Romano"></a> <dd id=col2><a href=#><img src=coliseu-forum.jpeg alt="Coliseu e Fórum Romano" ></a>

<dt><a href=#col3><img src=grand-via-Madrid-pequeno.jpeg alt="Grand Via, Madrid, Portugal"></a> <dd id=col3><a href=#><img src=grand-via-Madrid.jpeg alt="Grand Via, Madrid, Portugal" ></a>

</dl>

Você precisa do dd a toda a altura e a largura da página *e* no centro o seu conteúdo. Para fazer isso, você vai colocar o CSS no elemento HTML em si, de modo a que o dd pode medir em si em relação ao que, e usar o módulo para o centro da flexbox da criança elementos do dd (listando 6-25).

**Listando 6-25.**Base CSS para um Lightbox equivalente em CSS3

Html { min-altura: 100%; POSIÇÃO: relativos; }

Corpo { margin: 0; altura: 100%; margin-right: 2em; } dl#gallery { flutuação: esquerda; }

Dl#gallery dt { Largura: 150px; } dl#gallery dd {

Margem esquerda: 0; fundo: rgba(0,0,0,0); posição: absoluta; superior: 0; fundo: 0;

Largura: 100%; altura: 100%;

Visor: caixa de caixa;-pack:centro; alinhar a caixa:centro; visibilidade: oculto;

}

Dd { background: #fff; exibição: bloco; transição: 4s todas as facilidade em; }

Observe que há uma desvantagem potencial para essa abordagem se a página prorroga significativamente passado na parte inferior da janela do navegador, como este CSS fará com que a imagem para sempre na vertical contra o próprio centro da altura do corpo de conteúdo.

Para expandir e mostrar o elemento dd, você vai animar a imagem com uma seqüência de keyframe e fade página pela transição do fundo da lista de dd (Aceleração Total).

**Aceleração total listagem.**Keyframe sequência para um efeito Lightbox

@keyframes blowup {

0% { largura: 0; Altura: 0; opacidade: 0; }

30% { Largura: 640px; altura: 0; opacidade: 0; }

60% { Largura: 640px; altura: 480px; opacidade: 0; margin: 20px; } 100% { Largura: 640px; altura: 480px; opacidade: 1; margin: 20px; }

}

Dd:alvo {

Visibilidade: visível; fundo: rgba(0,0,0,0.6); transição: 2s linear de fundo;

}

Dd:alvo uma caixa {-sombra: 0 0 8px 8px rgba(0,0,0,0.3); } dd:alvo uma img { animação: blowup 3s frente; }

Ao vincular o conteúdo do dd para o id para o corpo, um clique sobre a imagem será reorientação dos o navegador e desapertar a animação.

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

Adição de legendas

Existem algumas opções para adição de legendas para o Lightbox CSS3 código. O primeiro adiciona um montante mínimo para o código existente, como elementos span (listando 6-27). O efeito é mostrado na Figura 6-4.

**Listando 6-27.**Keyframe sequência para um efeito Lightbox

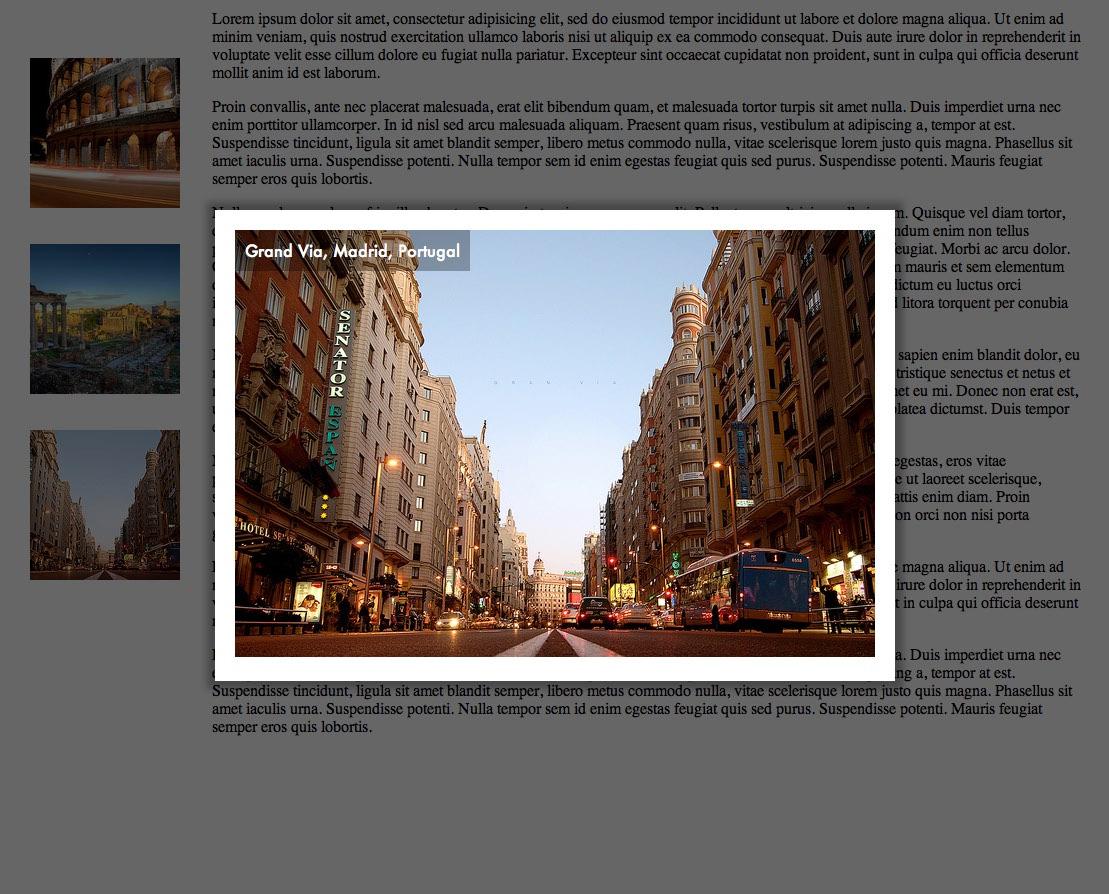
<dl id=gallery>

<dt><a href=#col1><img src=coliseu-noite-pequeno.jpeg alt="Coliseu à noite"></a> <dd id=col1><a href=#><img src=coliseu-noite.jpeg alt="Coliseu à noite"> <span>Coliseu à noite</span></a>

<dt><a href=#col2><img src=coliseu-forum-pequeno.jpeg alt="Coliseu e Fórum Romano"></a> <dd id=col2><a href=#><img src=coliseu-forum.jpeg alt="Coliseu e Fórum Romano"> <span>Coliseu e Fórum Romano</span></a>

<dt><a href=#col3><img src=grand-via-Madrid-pequeno.jpeg alt="Via Grand, Madrid, Espanha"></a> <dd id=col3><a href=#><img src=grand-via-Madrid.jpeg alt="Grand Via, Madrid, Portugal"> <span>Grand Via, Madrid, Portugal</span></a>

</dl>



**Figura 6-4.***Legenda no mouseover para uma grande imagem Lightbox CSS*

Adicionando CSS para o recipiente de dd, tornando posição: posicionamento relativo e o span dentro dela, é absoluta podem fazer a transição a legenda no mouse (listando 6-28).

**Listando 6-28.**CSS para uma legenda em um Lightbox Gallery

Dl#gallery {

Flutuação: esquerda; font-family: Futura, Arial, sans-serif; margin-bottom: 12em;

}

Dl#gallery dd {

Margem esquerda: 0; fundo: rgba(0,0,0,0); posição: absoluta; superior: 0; fundo: 0; largura: 100%; altura: 100%;

Visor: caixa de caixa;-pack:centro; alinhar a caixa:centro; visibilidade: oculto;

}

Dl#gallery dt { Largura: 150px; margin: 2em 2em 0 2em; }

Dd:alvo { visibilidade: visível; fundo: rgba(0,0,0,0.6); transição: 2s linear de fundo;

}

Dl#gallery dd { background: #fff; exibição: bloco; text-decoration: nenhum; } dl#gallery dd um span {

Visor: bloco; background-color: rgba(0,0,0,0.3); cor: branco; POSIÇÃO: absoluta; superior: 20px; esquerda: 20px;

Padding: 10px; opacidade: 0; transição: 1s opacidade facilidade em;

}

Dl#gallery dd um:foco span { opacidade: 1; }

Dl#gallery dd:alvo uma img { animação: blowup 3s frente; } dl#gallery dd:alvo um {

Caixa-sombra: 0 0 8px 8px rgba(0,0,0,0.3); transição: 4s todas as facilidade em; POSIÇÃO: relativos;

}

CSS3 também pode ser usado para animar um logotipo corporativo no carregamento da página (ver Figura 6-5). Vou explorar esse tópico mais quando chegarmos ao design ágil (Capítulo 9). É importante para reproduzir a animação apenas uma vez: um loop de animação irá causar uma grande distração do visualizador.



**Figura 6-5.***É acrescentada uma legenda para um logotipo corporativo no carregamento da página*

100

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

Animação do logotipo no Carregamento da Página

Como a íris componentes não serão animados, apenas a íris como um todo, você normalmente criar o logotipo como uma imagem (possivelmente como um ficheiro vectorial SVG). Para fins de ilustração, vamos do puro CSS.

(Consulte a listagem 6-29).

**Listando 6-29.**Marcação de HTML para um logotipo corporativo

<div id=recipiente> <span id=iris1></span> <span id=iris2></span> <span id=iris3></span> <span id=iris4></span> <span id=iris5></span> <span id=iris6></span> <span id=iris7></span> <span id=iris8></span> <div id=iris>

</div>

</div>

<h1>ávido <span>Laboratórios</span></h1>

Para isso você pode adicionar CSS para criar a íris parte do logotipo e o estilo do texto (listando 6-30).

**Listando 6-30.**Base CSS para um logotipo corporativo

Div#container span {

Visor: bloco; largura: 100px; border: 7px branco sólido; altura: 32px;

Posição: absoluta; superior: 90px; esquerda: 90px; background-color: #333;

Transformar a origem: parte superior direita; índice z: -4;

}

Div#iris {

Border-radius: 50%; fronteira: 30px solid #fff; largura: 200px; altura: 200px; POSIÇÃO: relativos;

}

Div#container span#iris1 { transformar: traduzir(-35px, 55px) gire(0ºC); } div#container span#iris2 { transformar: traduzir( 70px, 90px) gire(45graus); } div#container span#iris3 { transformar: traduzir(-130px, 70px) gire(90graus); } div#container span#iris4 { transformar: traduzir(-160px, 20px) gire(135 graus); } div#container span#iris5 { transformar: traduzir(-145px, -35px) gire(180deg); } div#container span#iris6 { transformar: traduzir(-95px, -60px) gire(225 graus); } div#container span#iris7 { transformar: traduzir(-45px, -45px) gire(270deg); } div#container span#iris8 { transformar: traduzir( -20px, 5px) gire(315deg); } h1 {

Font-family: "Univers LT 55"; texto-transformar: letras maiúsculas; font-size: 80px; letra espaçamento: 10px; POSIÇÃO: absoluta; superior: 10px; esquerda: 150px; alinhar texto: esquerda; altura de linha: 50px;

}

101

Capítulo 6 ■ CSS3 ANIMAÇÕES KEYFRAME PARA CONTEÚDO DA WEB

H1 span {

Font-family: "Univers marcação 45 Luz"; font-size: 25px; exibição: bloco; letra espaçamento: 20px; texto-travessão: 30px;

}

Finalmente, você pode criar a lista de animação (6-31).

**Listando 6-31.**Keyframe para a sequência de animação logotipo corporativo

@keyframes spinner { 0% {

Transformar: traduzir(800px,-25px) gire(378deg); opacidade: 0; }

100% { transformar: traduzir(2px,-25px) gire(-56graus); opacidade: 1; }

}

Div#container {

Posição: relativos; largura: 200px; altura: 200px; animação nome: giram;

Animação duração: 2s; transformar: gire(-18 deg);

Animação: rotor 1.5s 2s tanto linear;

}

Note que você não estiver conectando a animação para qualquer estado específico, de modo que a animação keyframe será executado automaticamente no carregamento da página.

Resumo

Neste capítulo que já experimentaste usa de CSS3 Keyframes para criar animações para galerias de imagens e outros conteúdos de página. Sequências de Keyframe pode incluir qualquer forma de movimento complexo que precisa de ser chamado a qualquer hora: clique do mouse, carregamento da página, ou outra forma de interação com o usuário.

Uma questão que eu não abordadas na animação tão longe é a escalabilidade de exibição: quanto maior a sequências de elementos no seu tornar-se, mais difícil o navegador e a GPU terá de trabalhar para transformar, transição, e animá-los. Além disso, imagens bitmap são uma grande parte do total de tamanho de página, abrandar as velocidades de conexão, especialmente em dispositivos móveis. As imagens também não respondem bem ao dimensionamento: duplicação dos DPI da tela que seu trabalho baseado em bitmap é exibido no resultará na degradação da imagem.

No próximo capítulo, você vai olhar para uma potencial resposta a todas estas questões, sob a forma de gráficos vetoriais SVG integrado com CSS3.

**Capítulo 7**

**Integrando CSS3 animações com SVG e filtros**

Cada navegador que suporta o CSS transforma, transições e animações também suporta SVG (gráficos vetoriais escaláveis), um formato de imagem que tem sido desde há muito negligenciadas pela maioria dos desenvolvedores da web. Depois de ser ignorado pelo Internet Explorer (IE) para uma década, SVG está vivendo algo de um renascimento com o seu apoio no IE9 e todos os outros browsers modernos, tornando-o ideal para a implantação em desenvolvimento para plataformas móveis e outros casos de uso. Como você verá neste capítulo, CSS3 transições e keyframes pode integrar com SVG muito bem.

Filtros, discutida na segunda metade deste capítulo, são novos para o CSS mas padronizado em SVG. Na verdade, filtros CSS são diretamente derivadas do SVG padrão. Os filtros permitem em tempo real e interativo edição visual de conteúdo HTML, particularmente imagens, que foi anteriormente só possíveis no PhotoShop. Como uma propriedade CSS, os filtros podem ser facilmente animados, apenas como qualquer outra coisa.

Uma introdução à SVG

SVG é um formato baseado em XML. Este facto permite que dados SVG sejam criadas e alteradas em um editor de texto comum e em quase qualquer linguagem de desenvolvimento web. SVG suporta as suas próprias inclinações, interatividade, texto e camadas, mas o recurso mais importante para os nossos propósitos é o facto de o formato descreve  formas de vetor. Isto significa que as imagens SVG podem ser dimensionados para qualquer tamanho ou resolução ou transformado em qualquer forma sem qualquer perda de qualidade. O facto de o formato oferece uma descrição baseada em texto de informações sobre vetores também tende a tornar os arquivos SVG relativamente pequena: uma simples interface de usuário (UI) forma como um botão play, por exemplo, pode ser descrita em SVG como três pontos e uma cor de preenchimento, em vez de individualmente definidos pixels. Isto torna o formato naturalmente ágil e um encaixe perfeito para o design de móveis de alta DPI exibe e manipulação com CSS.

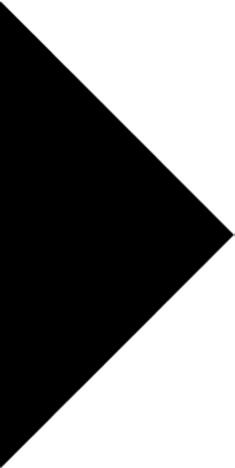
Considere, por exemplo, uma interface do botão Reproduzir o elemento. Reduzida a seu mínimo absoluto, o código SVG para descrever uma tal forma pode ser algo como a listagem 7-1.

**Lista 7-1.**Um simples arquivo SVG

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>"> <pontos poligonais="0,0 0,400 200,200 "/>

</svg>

Dados SVG podem ser visualizadas diretamente em um navegador. Salvando o código na lista 7-1 com o nome de arquivo reproduzir.svg e o carregamento em uma janela de navegador dá a você o resultado visual mostrado na Figura 7-1.



**Figura 7-1.***Um SVG botão play exibida em um navegador*

Note que o elemento polígono SVG usa pontos para descrever formas de uma forma semelhante ao a ligar mapas. Em última análise, SVG elementos devem ser prestados como pixels pelo seu visor e por padrão nos pontos especificados para o polígono mapa de pixels na tela. (o botão play será exibida no navegador como 400 pixels de altura e duzentos pixels de largura por padrão e será posicionado no extremo no canto superior esquerdo da tela.) Finalmente, a renderizado definitivo dimensões de um elemento SVG são arbitrários: o botão poderia ser tornada uma mile high (se sua tela foi grande o suficiente) ou dois centímetros de largura (por exemplo, em uma página impressa) e todos os outros factores de terras, a qualidade seria a mesma.

Novamente, como com a ligar mapas, geralmente não é muito eficiente para gravar todos os de seu código com a mão para SVG. Para muitas tarefas que você vai encontrar consideravelmente mais fácil de usar um aplicativo de desenho e exportar o resultado como um arquivo SVG. (I vai discutir essas ferramentas em breve.)

Colocação SVG em uma página da Web

Existem três principais métodos usados para colocar um ficheiro SVG em uma página HTML: referenciada como uma imagem em linha na sua página (também chamado *embedded* SVG), e referenciado como um objeto.

SVG como uma imagem em linha

O método para a colocação de um elemento SVG em uma página da web mais familiares aos desenvolvedores da web é usar a tag <img>. Antes de você fazer isso, sua SVG código deve conter um pouco mais de informação quanto à sua "natural" size (novamente, mantendo o código para um mínimo absoluto), como mostrado na lista 7-2.

**Lista 7-2.**Código SVG preparado para inserção em uma página da Web como uma imagem

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>" NEGATOSCÓPIO="0 0 400 400"> <pontos poligonais="0,0 0,400 200,200 " fill="rgba(90,70,80,0.5)" />

</svg>

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS

O NEGATOSCÓPIO atributo especifica uma "tela" de 400 por 400 pixels. Sem largura ou altura especificada para o elemento de imagem no CSS, o elemento SVG reserva um "espaço" de 400 × 400 pixels para si quando é colocado na sua página como uma imagem com a norma HTML, mostrado na lista 7-3.

**Lista 7-3.**Um elemento SVG inserido em uma página da Web como uma imagem

<img src=svg/reproduzir.svg alt=Reproduzir>

Você também pode referenciar SVG como uma imagem em qualquer lugar que você normalmente usaria uma imagem no seu CSS-por exemplo, como pano de fundo para um elemento (listando 7-4).

**Lista 7-4.**SVG aplicada como uma imagem de fundo em CSS

H1 { background: url(svg/onda.svg); }

Enquanto estiver fazendo referência a um arquivo SVG externo desta forma é o método mais simples e comum, não efectuar vários inconvenientes:

 O arquivo SVG é tratada unicamente como uma imagem; qualquer interactividade ou scripts escritos no código são ignorados.

 Você não pode "alcance o interior" o código SVG directamente para alterar a aparência dos elementos usando CSS com o mesmo grau de liberdade que você tem com SVG em linha.

Inline SVG

Se você incorporar SVG dados diretamente na sua página, você precisa incluir um pouco mais de informação para o código SVG. Você precisará fornecer a largura e altura do elemento quer como atributos (negatoscópio, largura ou altura atributos) ou como um estilo (consulte a listagem de 7-5).

**Lista 7-5.**SVG aplicado em linha com HTML

<!DOCTYPE HTML> <html lang=pt> <cabeça>

<meta Charset=utf-8>

<título > SVG Exemplo de arquivo incorporado</title></head>

<body>

<h1 > Conteúdo padrão</h1>

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>" style="Largura: 200px; altura: 400px"> <pontos poligonais="0,0 0,400 200,200" />

</svg>

</body>

</html>

Inline SVG salva um pedido HTTP extra, o que é particularmente importante para o mobile páginas onde lag é prevalente. Outras vantagens sobre simplesmente referenciando o arquivo como uma imagem são a capacidade de influenciar diretamente a aparência do SVG com CSS. Scripts interatividade com o SVG também é suportado.

A principal desvantagem para inline SVG é o simples facto de ele adiciona mais código para a sua página HTML.

SVG adicionado como um objeto ou iframe

Adicionando SVG para uma página da web como um objeto ou iframe é o método mais antigo, em que a interactividade é mantida com scripts dentro do elemento (listando 7-6).

**Lista 7-6.**SVG aplicado como um objeto ou iframe para uma página da Web

<object type="image/svg + dados xml"="icon.svg"> Aviso para os navegadores mais antigos ou conteúdo alternativo </objeto>

<iframe src="icon.svg">

Aviso para os navegadores mais antigos ou conteúdo alternativo </iframe>

No entanto, aplicado como um <object> ou <embed> tag, a capacidade de personalizar a aparência do SVG elementos é minimizado . O <object> ou <embed> tag também pode aparecer com as barras se o conteúdo SVG transborda o seu recipiente.

Manipulação de SVG com CSS

Como você pode colocar SVG em uma página da web de várias maneiras diferentes para que você possa manipular a aparência de um elemento SVG utilizando vários métodos diferentes. Esta é mais complexa pelo fato de que a SVG tem sua própria sintaxe nativo para alcançar alguns efeitos visuais.

No nível mais simples, você pode redimensionar um elemento SVG quando é colocado na sua página como uma imagem, como mostrado na lista 7-7.

**Lista 7-7.**Inline CSS usado para redimensionar um elemento SVG

<img src=jogar.svg alt=Play style="Largura: 50px; altura: 50px">

Você também pode alterar a aparência do elemento a partir de uma folha de estilo incorporado ou vinculado, da mesma forma que você escrever regras de apresentação para qualquer outro tipo de imagem. (Para isso, SVG elementos utilizados em uma página são frequentemente um atributo id).

Você pode alterar a cor de preenchimento de um formato SVG nativamente dentro do arquivo SVG, usando encher (veja lista de 7-8). (Nota que encha pode tomar qualquer valor de cor utilizados em CSS: palavra-chave, hexadecimal RGB, ou hsl).

**Lista 7-8.**Encha o atributo usado para cor um elemento polígono SVG

<pontos poligonais="0,0 0,400 200,200 " fill="red" />

Você também pode alterar a cor de preenchimento usando uma folha de estilos incorporada no arquivo SVG (listando 7-9).

**Lista 7-9.**Modificando a Aparência de um elemento SVG com uma folha de estilos incorporada

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>" NEGATOSCÓPIO="0 0 400 400"> <style type="text/css">

Polígono { encher: azul; } </Estilo>

<pontos poligonais="0,0 0,400 200,200" /> </svg> Por último, você também pode fazer isso através do seu CSS se o arquivo SVG está incorporado diretamente em sua página (listando 7-10).

**Listando 7-10.**Modificando a Aparência de um elemento SVG em linha com uma folha de estilos incorporada

<!DOCTYPE HTML> <html lang=pt> <cabeça>

<meta Charset=utf-8>

<título > SVG Exemplo de arquivo incorporado</title> <style>

Polígono { encher: vermelho; acidente vascular cerebral: preto; STROKE-Largura: 9px; } </Estilo>

</cabeça>

<body>

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>" style="Largura: 200px; altura: 400px"> <pontos poligonais="10,19 10,390 190,2" />

</svg>

</body>

</html>

Observe que isso não significa que você de alguma forma magicamente ter novos tempos ou encher as propriedades de qualquer elemento HTML, tanto quanto você pode desejar. Estas propriedades são muito especificamente aplicados ao conteúdo SVG apenas.

Você também pode detectar o mouse sobre elementos SVG e alterar a aparência do conteúdo SVG em resposta (listando 7-11).

**Listando 7-11.**Passe o mouse sobre uma detecção CSS elemento SVG

Polígono { encher: vermelho; acidente vascular cerebral: preto; STROKE-Largura: 9px; } polígono:foco { encher: preto; }

Nota uma pequena mas significativa vantagem de usar o botão para o SVG em linha: a área "hot spot" para o mouse é exactamente a forma do polígono. Em uma imagem bitmap padrão, a área é sempre rectangulares, não importa como o formato da imagem pode ser exibida.

Isso significa que você pode também esses efeitos de transição, como você pode com qualquer outro elemento influenciado por CSS. Para fazer o CSS mais claras e específicas, eu adicionei uma id para o elemento de polígono e abordadas de forma que na minha estilos, como mostrado na listagem 7-12. (Nota que eu também trouxe o polígono "no" um pouco das bordas, como acidente vascular cerebral é adicionada para o exterior da forma, da mesma forma que o CSS fronteira é para elementos HTML. Sem isso, as dicas do curso seria cortado pelas bordas do SVG "tela" como prorrogada para além deles).

**Listando 7-12.**CSS em um elemento de transição SVG

<!DOCTYPE HTML> <html lang=pt> <cabeça>

<meta Charset=utf-8>

<título > SVG Exemplo de arquivo incorporado</title> <style>

#play { encher: vermelho; acidente vascular cerebral: preto; STROKE-Largura: 9px; transição: 1s todos lineares; } #play:foco { encher: preto; acidente vascular cerebral: preto; STROKE-Largura: 9px; }

</Estilo>

</cabeça>

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS

<body>

<svg version="1.1" xmlns="<http://www.w3.org/2000/svg>" style="Largura: 200px; altura: 400px"> <polígono id="play" pontos="10,19 10,390 190,2" />

</svg>

</body>

</html>

Qualquer CSS3 transformar, transição, ou animação que pode ser aplicado a um elemento HTML pode também ser aplicada a SVG, tornando a combinação de formas de vetor e CSS extremamente poderoso.

Uma animada Imagemap SVG

Neste exercício você vai usar SVG para duplicar a funcionalidade de um HTML imagemap UI, mas você vai adicionar várias etapas evolutivas: secções da sua "imagemap SVG" irá mascarar o conteúdo da imagem que serão transferidos com CSS3 sobre a pairar sobre a áreas adequadas (ver Figura 7-2).



**Figura 7-2.***Um imagemap SVG*

Primeiro, você precisa de um desenho SVG. Felizmente, há uma grande quantidade de conteúdo SVG gratuito disponível on-line: neste caso, vou usar um mapa do Canadá fornecidos pela Wikimedia Commons. Para fins de ilustração, eu já editou o mapa para apenas as províncias de British Columbia, Alberta e Saskatchewan.

Cada província é demarked por um caminho, fornecido como uma seqüência de coordenadas. Vamos começar com o posicionamento do SVG em nossa página e a criação de um simples efeito hover em cada caminho. O resultado será algo como listando 7-13.

**Listando 7-13.**Realce um simples Imagemap SVG

<!DOCTYPE HTML> <html lang=pt> <cabeça>

<meta Charset=utf-8>

<título > SVG Exemplo de arquivo incorporado</title> <style>

Caminho { encher: #fdfdfd; } caminho:foco { encher: vermelho; } </Estilo>

</cabeça>

<body>

<svg version="1.1" Xmlns:svg=<http://www.w3.org/2000/svg>

Xmlns="<http://www.w3.org/2000/svg>" xmlns:xlink="<http://www.w3.org/1999/xlink>" x="0px" y="0px" width="1000px" height="600px" NEGATOSCÓPIO="0 0 1000 660">

<caminho id="alberta" d="M1654.393,678.219 . . .>

<caminho id="british-Columbia" d="M982.854,27.912l150.51,33.221c17. . .>

. . .

</svg>

Para alcançar o efeito de máscara, você está indo para incorporar cada "mostrar através da imagem após cada caminho", e depois rode o caminho para um clipPath. Você vai associar os elementos juntos referenciando a id do encaixe apropriado caminho para a imagem e agrupar os elementos juntamente com <g> .

Início código para a imagem SVG, incluindo o caminho para Alberta, será algo como listando 7-14.

**Listando 7-14.**Cortar caminho para SVG uma imagem

<g>

<clipPath id="ab-Clipper">

<caminho encher="#D3D3D3" d="M1654.393,678.219 . . .> </clipPath>

<image clip-path="Url(#ab-clipper)" height="100%" width="100%" x="800" y="50" xlink:href="lago-louise.jpeg" preserveAspectRatio="xMidYMin slice" />

</g>

ClipPath define o caminho dentro dela para agir como uma máscara para os elementos que fazem referência à ID de clipPath como o lago-loise imagem.JPEG. A imagem é o elemento de ligação cruzada para um bitmap imagem no mesmo local que o mapa. A imagem é definido para todo o seu "naturais" de largura e altura e movido sobre os eixos x e y até que esteja no mesmo local

Como o caminho. (é provável que você precise definir o link para o corte caminho como a última ação que você tomar com a imagem; caso contrário, poderás jogar um jogo de adivinhação quanto à posição da imagem enquanto ele é mascarado e provavelmente invisível).

Agora que você criou uma imagem bitmap mascarados. O CSS para revelar esta porção do mapa (listando 7-15) é simples.

**Listando 7-15.**CSS para mostrar uma imagem recortada sobre a pairar no um elemento SVG

Imagem svg { opacidade: 0; transição: 400ms opacidade facilidade; } svg imagem:hover { opacidade: 1; }

Há apenas um problema: o caminho de recorte é invisível, o que significa que existe uma folga de Alberta-médias em seu mapa. O efeito hover funciona, mas o usuário não verá onde para mover o mouse se você aplicar o mesmo efeito para a outra província caminhos. Uma vez que é transformada em um clipe, a preencher no caminho é inútil. Você pode alterar a cor durante todo o dia e nunca ver qualquer diferença.

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS

A solução é colocar uma cópia preenchida do caminho *entre* a cortar caminho e a imagem no código, como mostrado na listagem 7-16.

**Listando 7-16.**A área cortada se tornou visível em SVG com a adição de um caminho cheio copiado

<g>

<clipPath id="ab-Clipper">

<caminho encher="#D3D3D3" d="M1654.393,678.219 . . .> </clipPath>

<caminho encher="#FF0000" d="M1654.393,678.219. . . />

<image clip-path="Url(#ab-clipper)" height="100%" width="100%" x="800" y="50" xlink:href="trigo.jpeg" preserveAspectRatio="xMidYMin slice" />

</g>

A imagem de bitmap (por Kenny Louie, licenciado sob Creative Commons: [http://flickr.com/photos/](http://flickr.com/photos/kwl/3102355428) [kwl/3102355428](http://flickr.com/photos/kwl/3102355428)) é ainda invisível, mas é traduzida por "acima" do caminho, copiado ainda mostra quando o usuário passa o mouse sobre a área preenchida com o caminho copiado.

A etapa final é a ligação entre cada uma das áreas na imagem SVG. O link vai dentro de cada grupo no próprio arquivo SVG e requer o uso do namespace xlink (listando 7-17).

**Listando 7-17.**Um imagem recortada em SVG

<g>

<a xlink:href=" <http://www.hellobc.com/>"> <clipPath id="bc-Clipper">

<caminho d="982.854 27.912M,L,33.221150.51 c17. . .> </clipPath>

<caminho id="british-Columbia" d="M982.854,27.912l150.51,33.221c17. . .>

<image clip-path="Url(#bc-clipper)" height="100%" width="100%" x="80" y="50" xlink:href="false-creek.jpeg" preserveAspectRatio="xMidYMin slice" />

</a>

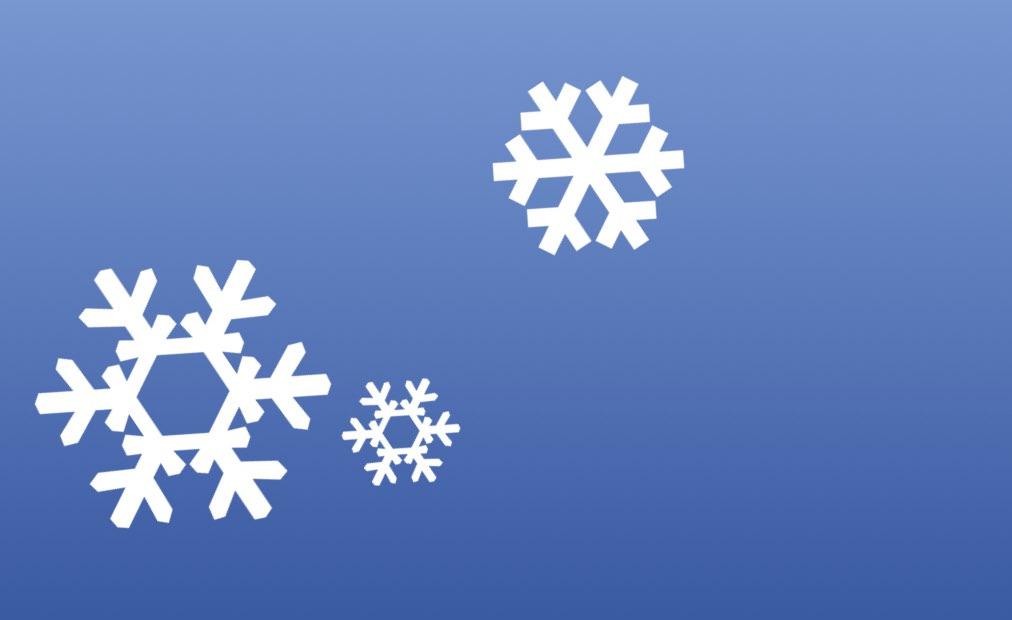
</g>

Observe que a área do link seguinte à beira do caminho, tal como a ligar mapas real e a transição acontece dentro da mesma área.

Floco de animação SVG

A escalabilidade de formas de vetor significa que você pode usar várias cópias de um elemento SVG em tamanhos diferentes sem se preocupar com a qualidade da imagem. Vou demonstrar esta animando chuviscos para um vetor de fundo sazonal cena. (Para isso, vou usar um ligeiramente modificado SVG floco baixado do Wikimedia Commons em <http://upload.wikimedia.org/wikipedia/commons/5/50/Snow_flake.svg>. Consulte a Figura 7-3.)

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS



**Figura 7-3.***Ainda a partir de uma queda de floco de animação SVG*

Primeiro você vai colocar o floco de neve na página como várias imagens. Os flocos serão diferentes tamanhos e terá início em locais diferentes, mas eles compartilharão a mesma animação características: eles cairão do céu, derrapar lateralmente no Breeze como fazê-lo. (Consulte a listagem 7-18).

**Listando 7-18.**SVG chuviscos colocado como imagens em uma página

<img src="floco.svg" alt="" class="flake" style="top: -50px" >

<img src="floco.svg" alt="" class="flake" style="esquerda: 200px; largura: 60px; altura: 60px; superior: -120px;" >

<img src="floco.svg" alt="" class="flake" style="esquerda: 640px; largura: 120px; altura: 120px; superior: -400px;" >

Em seguida, você vai aplicar o CSS para criar a impressão de um céu de inverno através de um gradiente e criar uma base para o tamanho chuviscos. Ao mesmo tempo, você chamada em duas sequências de keyframe: um para girar a chuviscos à medida que caem para o fundo (chamado de neve) e outro para o desvio chuviscos de lado a lado (desvio). (Consulte a listagem 7-19).

**Listando 7-19.**Base CSS para um floco de animação

Html { min-altura: 100%; }

Corpo { altura: 100%; fundo: linear-gradiente(#b5d3ff, #30509A); } img.flake { Largura: 150px; altura: 150px; POSIÇÃO: relativos; animação: neve 8s infinita frente, linear

Drift 12s facilidade em a frente infinito; }

As sequências de keyframe executado em diferentes comprimentos, e cada floco de neve começa com uma altura diferente (listando 7-20); a combinação cria a impressão de movimento alternado aleatória.

111

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS

**Listando 7-20.**Sequências de um floco de neve Keyframe Animação

@keyframes neve {

100% { topo: 700px; transformar: gire(2.2Turn); }

}

@keyframes drift {

0% { esquerda: -5px; }

25% { esquerda: 55px; }

55% { esquerda: -15px; }

100% { esquerda: 0px; }

}

Isso funciona, mas como você pode ver, existem duas questões. A primeira é que, para adicionar mais chuviscos, você precisa adicionar mais imagens para o markup, que rapidamente se torna cansativo. A segunda questão é o facto de os pequenos chuviscos será interpretado como sendo mais longe, e portanto deve demorar mais a cair na parte inferior do ecrã (que,

Para este exemplo, vamos assumir que é 700 pixels de altura). Você pode fazer uma chamada no chuviscos através do uso de classes com diferentes timings para o movimento, mostrado na lista 7-21.

**Listando 7-21.**Desacelerando Animação criando classes Secundário

Img.flake { Largura: 150px; altura: 150px; POSIÇÃO: relativos; animação: neve 8s infinita frente, linear

Drift 12s facilidade em a frente infinito;

}

Img.lento {

Animação: neve 16s infinita frente, desvio linear 24s facilidade em a frente infinito;

}

Classes separadas significa que um floco mais lenta poderia ser controlada por duas classes. A criação de mais variação significaria criar mais classes que retorna para o mesmo problema com a adição de mais chuviscos. Você vai abordar esta questão quando você começar a integrar JavaScript com CSS3 animações no Capítulo 9.

Ferramentas para SVG

Agora um dos maiores obstáculos à difusão do uso de SVG é a relativa escassez de ferramentas de design. Os mais populares incluem:

 Adobe Illustrator tem uma opção de exportação SVG mas, como acontece com muitas ferramentas WYSIWYG, o código cria não é terrivelmente eficiente: gerados arquivos SVG geralmente contêm muito mais código do que o necessário.

 Inkscape (<http://inkscape.org/>), a alternativa de código aberto, partilha a mesma questão como Adobe Illustrator. Inkscape, contudo, tem a vantagem de trabalhar com arquivos SVG nativo e suporta SVG filtros.

 Raphaël é uma pequena biblioteca JavaScript que permite a fácil criação e manipulação de SVG dentro de JavaScript.

112

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS

Introdução ao CSS3 Filtros

Filtros CSS a permitir o processamento de conteúdo da web antes de aparecer na página. Mais comumente (mas não exclusivamente), esses filtros são aplicados a imagens bitmap. Filtros de alterar significativamente o típico fluxo de trabalho de produção de imagens de desenvolvimento web: em vez de "em" permanentemente assar efeitos visuais em pixels de uma imagem, os designers podem otimizar e exportar uma imagem a partir do PhotoShop que continuará relativamente intocada, deixando alterações visuais para CSS.

Isto significa que as imagens podem ser alterados com a voar em CSS, em vez de ter de re-editar originais no PhotoShop, exportá-los, e depois voltar a carregá-las para um servidor. Isso significa também que esses efeitos podem ser animados.

Preto e Branco/cinza de Transição do Filtro

O efeito do filtro de cinza é ideal para carteiras de on-line ou galerias fotográficas. Em vez de usar soluções complexas JavaScript ou flash, um CSS filtro pode facilmente converter uma imagem a cores para preto e branco. Você pode desfazer esta conversão no passe e prenda uma transição para facilidade entre os dois estados, como com a imagem mostrada na Figura 7-4 (por Andrew Larsen em [www.flickr.com/photos/papalars/4013594219](http://www.flickr.com/photos/papalars/4013594219), licenciado sob Creative Commons). Consulte a listagem de 7-22.



**Figura 7-4.***Uma imagem a cores (direita) filtrado com CSS (esquerda)*

**Listando 7-22.**Aplicando um filtro cinza de Transição

Img { border: 20px solid #fff; caixa-sombra: 10px 10px 8px rgba(0,0,0,0.3); } img.bw { filtro: cinza(1); transição: 1s filtro linear; }

Img.PN:foco { filtro: cinza(0); }

<img src=lago-louise.jpg alt="Lago Louise, Alberta, Canadá" class=PN>

Tal como com os outros filtros vou descrever neste capítulo, o filtro cinza toma valores entre 0 (Nenhum efeito) e 1 (Full) com valores de ponto flutuante entre. Observe que não é possível uma transição suave para um estado de "none" ou nenhum filtro aplicado; o filtro deve ser dado um novo valor.

Transição do filtro sépia

A sépia filtro pode ser usado para criar um efeito de fotografia "envelhecido", conforme mostrado na Figura 7-5 (aplicado a uma fotografia por Robb Report Norte, [www.flickr.com/photos/robbn1/3650713106](http://www.flickr.com/photos/robbn1/3650713106)) .

Capítulo 7 ■ Integração CSS3 ANIMAÇÕES COM SVG E FILTROS



**Figura 7-5.***Uma imagem a cores (direita) filtrado com CSS (esquerda)*

O código para obter o efeito mostrado na Figura 7-5 é muito semelhante ao de cinza (consulte a listagem 7-23).

**Listando 7-23.**CSS para transição de um filtro Sépia em uma imagem

Img { border: 20px solid #fff; caixa-sombra: 10px 10px 8px rgba(0,0,0,0.3); } img.old { filtro: sépia(1); transição: 1s filtro linear; }

Img.old:foco { filtro: sépia(0); }

<img src=barn.jpg alt="antigo celeiro" class=anos>

Enquanto o celeiro fotografia na Figura 7-5 anteriormente tinha sido processado em um editor de imagem para adicionar "intempéries" e uma vinheta parcial aparência, você utilizado apenas CSS para fornecer um tom de sépia à imagem. Os outros efeitos também podem ser adicionadas com CSS.

Desenvolvimento de Polaroid Efeito de fotografia

É possível combinar tanto um filtro e uma caixa-sombra com um valor de inserção para re-criar a aparência de um desenvolvimento Polaroid fotografia por circundante de uma imagem com uma div e animar o estilo aplicado ao recipiente elemento. A imagem realizada dentro dele terá de ser "empurrado" usando o índice z de modo que a inserção de sombra do interior div sobreposições (consulte a listagem 7-24).

**Listando 7-24.**CSS para transição de uma imagem com um efeito Polaroid

Div.polaroid { flutuação: esquerda; fronteira: 25px solid #f3f4e3; fronteira-bottom-largura: 45px; caixa-sombra: 0 0 200px 200px rgba(29,25,4,1) inserção;

Filtro: sépia(.8); transição: 3s todas as facilidade em; } div.polaroid img { posição: relativos; índice z: -1; }

Div.polaroid:foco { filtro: sépia(.2); caixa-sombra: 0 0 50px 0 rgba(29,25,4,0.2) inserção }

<div class="polaroid" > <img src="barn.jpg" alt="Fotografia de um antigo celeiro" > </a>

Filtro de desfocagem de Transição

Blur filtros precisam de ser utilizado com cuidado; sua aplicação excessivamente pode fazer um site difícil de ler ou interagir com. Blur é um dos poucos filtros que não toma um valor entre 0 e 1. Em vez disso, ele usa uma medição de comprimento para definir a quantidade de desfocagem. (Consulte a Figura 7-6, que utiliza outra imagem Creative Commons por Louise Estivador, [www.flickr.com/photos/aussiegall/6311469113](http://www.flickr.com/photos/aussiegall/6311469113).)



**Figura 7-6.***Uma imagem a cores (direita) filtrado com CSS blur (esquerda)*

Note que o efeito de desfocagem se estende por todo o caminho através da imagem, incluindo a fronteira e sombra. É possível limitar a extensão da desfocagem fazendo a fronteira e sombra as propriedades de um segundo elemento, pais, tais como um div. (Consulte a listagem 7-25).

**Listando 7-25.**CSS para aplicar um filtro de desfocagem para uma imagem

Img { border: 20px solid #fff; caixa-sombra: 10px 10px 8px rgba(0,0,0,0.3); } img.old {: filtro de desfocagem(2px); transição: 1s filtro linear; }

Img.old:foco {: filtro de desfocagem(0px); }

Resumo

SVG e os filtros são dois dos mais poderosos recursos que você pode adicionar ao CSS animações. SVG permite resolução livres transições e animações, enquanto os filtros permitem rápido do lado do cliente o processamento de imagem visual.

SVG é uma dúzia de anos; filtros CSS são novos. Ambos são imediatamente limitada pelo facto de que aquilo que você criar com eles é unitária: suas criações não pode ser imediatamente repetido. A animação de um elemento em uma tela é fácil e você pode reutilizar essa animação para animar um outro elemento, mas cada novo elemento deve ser criada separadamente.

Você pode usar o JavaScript para simplificar, capacitar e variar o seu CSS3 animações e transições, facilmente replicar elementos animados e fazer novas animações com um poder que você vai explorar no próximo capítulo.

**Capítulo 8**

**Integração de animação com CSS3 Ágil Web Design e JavaScript**

Agora é hora de levar todos os elementos de que você já procurou na medida em conjunto em suas páginas da web. Para fazer isso você irá integrar CSS3 transforma, transições e animações com os princípios do desenvolvimento local atual, incluindo ágil Web Design (propulsão).

A um nível mais básico, uma resposta local será composto de um design fluido, com a maioria dos elementos medidos em percentagens, rem, em vh ou unidades vw, em vez de pixels, combinadas com uma série de pontos de *interrupção CSS* definida no CSS media queries. A posição desses pontos de interrupção normalmente é definido pelos valores em que o design para a página "pausas" quando a largura do navegador (mais correctamente referida como uma porta de *visualização*) alterações. Para maior conveniência, esses valores são geralmente medidos em pixels, embora existam bons argumentos para a medição em rems ou ems para quebrar tanto quanto possível a partir do conceito de "tamanhos de ecrã".

Projetando um site com resposta web design princípios permite que os visitantes têm uma suave e contínuo e experiência ininterrupta com cada dispositivo que eles usam para acessá-lo: navegador de desktop, tablet, telemóvel, dispositivos e ainda para vir.

É uma boa prática para designar pontos de interrupção em larguras onde seu projeto falhar, em vez de usar o conjunto de dimensões de apresentação da última iPhone ou iPad modelo. Dispositivos móveis mudar rapidamente, e há muito mais variedade no smartphone e tablet dimensões, particularmente no mercado Android, do que a maioria das pessoas perceber. É muito melhor ter o site responda a dimensões que são relevantes para o seu projeto, em vez de as resoluções do arbitrário deste ano da tecnologia.

Em cada ponto de interrupção, você especificar alterações de layout: elementos são redimensionadas ou reposicionada e aparecem ou desaparecem. Uma *primeira* filosofia móvel inverte o processo de desenvolvimento típico: desde seu início você a projetar um site para telas pequenas (com uma resolução horizontal de 320 pixels ou menos) e adaptar a página como a porta de visualização alarga, permitindo que o site mais espaço para "respirar". Mobile primeiro tem a vantagem de permitir que você para se concentrar no desenvolvimento de funcionalidades que o site absolutamente necessidades, dado o espaço severamente restringido e a largura de banda da maioria dos dispositivos móveis.

■ **nota**como uma regra, usuários móveis devem ter exactamente o mesmo acesso às ferramentas de navegação e os recursos de um site como o fariam ao usar o site com um navegador de desktop. Antes de liberar um recurso para usuários móveis em um site ágil, considerar se o componente é necessária em todos os.

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

Muitos desenvolvedores web pensar ágil web design meramente em termos de @media queries. Enquanto as consultas são uma componente muito importante e será o foco aqui, é importante entender que muitas soluções ágeis irá exigir contribuições adicionais a partir de JavaScript e soluções do lado do servidor como PHP (normalmente referido como *rima: design ágil + Componentes do Servidor*).

O que eu vou concentrar neste capítulo é a utilização de CSS animações para facilitar a transição entre os pontos de interrupção em uma página de resposta. Convém notar que os designers de *amor* este stuff -provavelmente você vai encontrar-te constantemente puxando o canto inferior direito da janela do browser para trás e para a frente para apreciar os efeitos que você está prestes a criar muitos de seus usuários nunca verá isto. A maioria dos visitantes virá para um web site com seu navegador em um conjunto de largura e deixar inalteradas para a duração da sua visita, principalmente usuários móveis que não têm uma opção de alterar tamanhos de visualização. Muitas das técnicas que iremos explorar neste capítulo devem por conseguinte ser considerados "bom ter" em vez de necessário.

Mais adiante neste capítulo, você vai ver como JavaScript pode ser integrado com CSS3 animações e transições para os tornar mais eficazes e eficientes.

Redimensionamento de elementos em resposta Web Design

Sem transições

Usando os princípios I discutido acima pode animar" o conteúdo de uma página da web na porta de visualização redimensionar sem usar transições ou animações keyframe.

Redimensionamento de Imagens e Vídeos dinamicamente

Primeiro, você vai redimensionar imagens e vídeos em resposta ao tamanho de visualização (consulte a Figura 8-1).



**Figura 8-1.***Dois membros de uma imagem dentro de uma página de resposta*

Por elementos de dimensionamento em relação aos respectivos contentores, você pode facilmente "animar" o tamanho dos elementos como a janela do navegador é redimensionada. Tecnicamente, esta não é animação. Nessa fase, você está simplesmente o dimensionamento de uma imagem dinamicamente pelo aumentá-lo em relação ao seu recipiente (consulte a listagem de 8-1).

**Lista 8-1.**Uma imagem de resposta

Html { font-size: 62,5%; } p { font-size: 1rem; }

.esquerda { max-largura: 100%; altura: auto; bóia: esquerda; margin-right: 2rem; margin-bottom: 1rem; } <Section>

<p><img src="pentacon-bike.jpg" alt="Pentacon Bike" class="left">Lorem ipsum dolor sit amet. . .

</section>

Como redimensionar o navegador os usuários têm a impressão de que a imagem (fornecido pelo filtran, [www.flickr.](http://www.flickr.com/photos/filtran/2978448269/) [com/photos/filtran/2978448269/](http://www.flickr.com/photos/filtran/2978448269/)) está sendo dinamicamente redimensionados. A foto será exibida em tamanho natural se o navegador está configurado suficientemente ampla, mas vai Redimensionar para ajustar as larguras de visualização que não pode conter a sua plena dimensão.

Existem duas possíveis desvantagens dessa abordagem: dependendo do tamanho inicial nativa da imagem em relação ao corpo do texto, a imagem pode aparecer fora de escala em larguras de tela grande ou pequena. Alternativamente, você pode definir a imagem como uma percentagem da largura do seu recipiente, de modo que as escalas para baixo através de toda a gama de visualização. Por exemplo, digamos que você tenha determinado que um acervo considerável tamanho total para os parágrafos é de 900 pixels de largura, com 90 caracteres por linha de medida (texto). Dada a regra de apresentação para o HTML e parágrafos, isso significa que a largura da seção elemento seria equivalente à 90rem. O tamanho natural da imagem é 425 pixels de largura. Para tornar a imagem completamente escalável, você traduzir todos esses valores percentuais:

425 / 900 = 0.4722

Isto significa que a imagem ocupa 47.22% da largura dos parágrafos e traduz para o CSS que você vê na lista 8-2.

**Lista 8-2.**Uma alternativa de resposta Imagem

Seção { max-largura: 90rem; }

.esquerda { largura: 47.22%; altura: auto; bóia: esquerda; margin-right: 2,5%; margin-bottom: 2%; }

Esta combinação de declarações CSS irá criar a aparência da página mostrada na Figura 8-2.



**Figura 8-2.***Uma alternativa de resposta imagem*

Um problema remanescente é o potencial de muito grandes ou muito pequenos imagens para ser criada no extremo de visualização tamanhos. Você pode querer proteger o projeto definindo tamanhos mínimo e máximo para a imagem (consulte a listagem de 8-3).

**Lista 8-3.**Uma imagem de resposta com um tamanho mínimo e máximo

.esquerda { largura: 47.22%; altura: auto; bóia: esquerda; margin-right: 2,5%; margin-bottom: 2%; max-Largura: 425px; min-Largura: 150px;

}

Você pode obter o mesmo efeito em <video> elementos usando a mesma técnica. (fazendo um vídeo de resposta quando ele é incorporado através de um serviço como o YouTube ou no vimeo é significativamente mais complicado; eu recomendo Dave Ruppert FitVids do plugin JQuery em <http://fitvidsjs.com/>alcançar que).

■ **Dica**Também é possível redimensionar texto dinamicamente sem usar keyframes, transições, ou media queries. A  unidade vw mede a largura da porta de visualização: 1VW é centésimo da largura da janela do navegador. Se a porta de visualização é 400 pixels de largura, 1VW seria equivalente a 4px. Você pode utilizar esta unidade para a escala de seu texto como o navegador redimensiona:

H1 { font-size: 4VW; }

Com este CSS, h1 elementos sobre a página da web será redimensionar como o navegador se expande e se contrai. Você também pode usar o vh e a VW sobre outros elementos bem.

Resposta imagens de plano de fundo

Você pode facilmente redimensionar imagens de fundo dinamicamente em resposta a alterações de visualização usando a propriedade Tamanho de fundo (aqui utilizando uma imagem por Vinoth Chandar em [www.flickr.com/photos/](http://www.flickr.com/photos/vinothchandar/6168933212/) [vinothchandar/6168933212/](http://www.flickr.com/photos/vinothchandar/6168933212/)) como na cotação de 8-4.

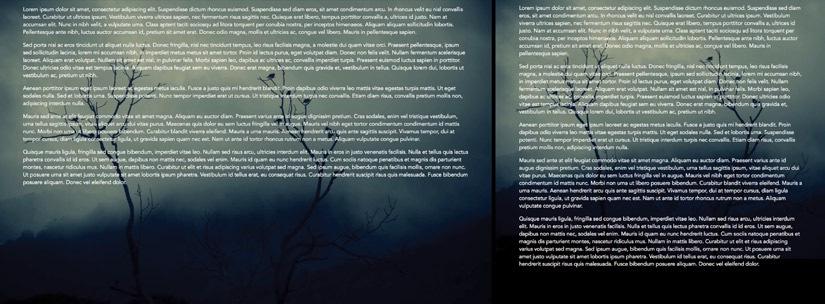
Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

**Lista 8-4.**Uma imagem de fundo de resposta

Html, corpo { min-altura: 100%; font-size: 62,5%; }

Corpo { background-image: url(nevoeiro.jpg); fundo-size: tampa; }

Combinado com alguns o corpo do texto, isto dá o efeito mostrado na Figura 8-3.



**Figura 8-3.***Uma imagem de fundo de resposta*

Para outras formas de "transição" imagens de fundo, consulte o Capítulo 3.

Redimensionamento de elementos em resposta Web Design com as transições

É perfeitamente possível a transição de Elementos entre @meios de consulta de interrupção. A partir de uma perspectiva de design, o principal problema a ser consciente do potencial do elemento "saltar" entre Estados sobre visualização redimensionar, que pode ser surpreendente para o usuário.

Vamos criar um projeto que tem um grande h1 elemento no meio da página acima de uma listagem de fotografia (8-5).

**Lista 8-5.**HTML para obter uma imagem de fundo ágil e transferidos Texto

Corpo { background-image: url(nevoeiro.jpg); fundo-size: tampa;

Fundo - repetir: não-repetição; color: #fff; font-family: Avenir, Arial, sans-serif;

}

H1 { font-family: 'Calluna Sans", Arial, sans-serif; alinhar texto: centro; font-size: 10rem; margin: 8rem auto;

}

Como a porta de visualização restringe a rubrica texto será naturalmente quebrar em espaços, como mostrado na Figura 8-4.



**Figura 8-4.***Um elemento de rubrica sem resposta quebrando no espaços sob um estreito de visualização*

Se em vez disso quisemos manter o título em uma única linha, poderíamos reduzir o tamanho do texto a interrupção e a transição entre eles como mostrado na listagem 8-6.

**Lista 8-6.**HTML para obter uma imagem de fundo ágil e transferidos Texto

H1 { font-family: 'Calluna Sans", Arial, sans-serif;

Alinhar texto: centro; font-size: 10rem; margin: 8rem auto; transição: 1s font-size linear;

}

@tela mídia e (max-largura: 1100px) { H1 { font-size: 8rem; }

}

@tela mídia e (max-Largura: 900px) { H1 { font-size: 7rem; }

}

@tela mídia e (max-Largura: 800px) { H1 { font-size: 6rem; }

}

O código mostrado na listagem 8-6 fornece o resultado mostrado na Figura 8-5.



**Figura 8-5.***Um elemento de cabeçalho de resposta*

Indicando o tamanho de visualização com CSS3 Media Queries e Transições

Usando o Media Queries, você pode acionar um elemento para "metamorfose" na aparência como o navegador é estreitada-por exemplo, para mostrar os diferentes modos de visualização em uma página de resposta, para que os usuários compreendam que eles não estão olhando para uma simples interpretação de um site de desktop em uma plataforma móvel, mas em que uma página seja personalizando a si mesmo ao tamanho da janela do navegador (consulte a Figura 8-6).



**Figura 8-6.***Uma resposta a transição símbolo em três estados diferentes, representando três diferentes tamanhos de visualização*

Primeiro de tudo, você vai configurar uma série de pontos de interrupção:

 120em ampla e acima será considerado "widescreen".

 A 80em largo você pode assumir o utilizador está a visualizar o site com um tablet.

 A 40em e abaixo você deve assumir o usuário está usando um smartphone.

Em seguida você cria os elementos que serão usados para mostrar estes diferentes estados do puro CSS. Haverá um total de três elementos: um contendo div com uma posição absoluta que irá manter todos os elementos no canto superior direito do ecrã; um span representando o visor e outro span representando a base ou o botão, como mostrado na lista 8-7.

**Lista 8-7.**Código HTML para um modo de exibição em uma transição de página de resposta

<div id="viewingmode">

<span id="display"></span> <span id="buttonbase"></span>

</div>

Em seguida, você vai criar a base CSS para os elementos na cotação de 8-8. Isto irá incluir a transição configurações.

**Lista 8-8.**O código CSS para criar o modo de visualização de símbolos em um Design ágil

\* caixa de calibragem: {-fronteira-box; }

Corpo { font-family: Avenir, sans-serif; margin: 100px 0; } #viewingmode {

Largura: 150px; altura: 150px; fundo: rgba(0,0,0,0.2); posição: absoluta; superior: 0; Direita: 0; alinhar texto: centro;

}

Tela /\*, em widescreen padrão apresentação \*/

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

#exibir {

Largura: 80%; altura: 50%; fronteira: 12px solid #585858; border-radius: 5px; margin-top: 20px; background-color: #EEE;

}

/\* base, no suporte de monitor padrão - Configuração do teclado \*/

#buttonbase {

Largura: 90px; fronteira: 12px solid #585858; border-radius: 5px;

Posição: absoluta; superior: 100px; esquerda: 30px; transformar origem: 25px 5px;

}

#visor, #buttonbase { Display: bloco inline; transição: .5s todos lineares; }

Por último, você vai criar o media queries para alterar a aparência da #visor e #buttonbase elementos (listando 8-6). Observe que essas alterações em cascata: menor tamanhos de ecrã irá herdar as mudanças feitas no media queries para telas maiores.

**Lista 8-6.**O código CSS para criar o modo de visualização de símbolos em um Design ágil

@tela mídia e (max-largura: 80em) {

#exibir { largura: 50%; altura: 60%; fronteira: 10px solid #585858; } #buttonbase { Largura: 8px; altura: 8px; fronteira: nenhum;

Border-radius: 50%; fundo: #fff; topo: 101px; esquerda: 70px; }

|  |  |  |  |
| --- | --- | --- | --- |
| } |  |  |  |
| @tela mídia e | (max-largura: 40em) { | | |
| #exibir { border-largura: 23px | | | 5px; largura: 45%; altura: 73%; } |
| #buttonbase { | Topo: 110px; | } |  |
| } |  |  |  |

Conforme observado, a maioria dos usuários - principalmente usuários móveis não ver estas animações porque os dispositivos móveis têm larguras de tela fixa e nada para ajustar a. Uma abordagem alternativa poderia ser a de mostrar se a tela está sendo realizada no modo de retrato ou paisagem, como mostrado na lista 8-7.

**Lista 8-7.**Media Queries para mostrar o modo de retrato e de paisagem em um dispositivo

@tela mídia e (orientação: paisagem) e (max-largura: 80em) {

#exibir { transformar: gire(90graus); }

}

Isso cria a mudando telas você ver ilustrada na Figura 8-6, com cada dispositivo "Transmutação" no próximo como a porta de visualização é redimensionada de acordo com a orientação do dispositivo. Os mesmos princípios podem ser aplicados para animar muitos aspectos diferentes de páginas da web com design ágil.

Otimizando o CSS transições e animações para dispositivos móveis

É inevitável que os processadores em dispositivos móveis vai crescer mais rápido e mais potente ao longo do tempo, mas eles sempre ficar para trás as habilidades de modelos de desktop completo. Também é inevitável que os desenvolvedores tendem a código para a plataforma na frente dos seus narizes, do que aquilo que seu público pode estar usando (como vimos durante o navegador guerras e no prefixo fornecedor vieses). Os desenvolvedores a escrever código para a tela na frente deles e o resultado nem sempre pode diminuir a escala para as habilidades mais restrito de dispositivos móveis.

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

Existem várias formas possíveis de melhorar os efeitos CSS que lento em dispositivos móveis:

 *Tente a transições de funil e otimizações através da GPU do dispositivo*: Devido à sua natureza mais complexos e exigentes, muitos navegadores irá tentar suavizar 3D transforma, transições e animações (coberto no Capítulo 9) por pipeline-los através da Unidade de processamento de gráficos especializada do dispositivo sobre o qual o CSS animação é trabalhar. Você pode levante uma viagem sobre este fluxo otimizado por iniciar uma declaração CSS bloco com um "vazio" manipulação 3D que não irá alterar o elemento visual, mas vai permitir o acesso à GPU para o 2D transforma que siga a mesma declaração:

Transformar: traduzir3d(0,0,0);

■ **Nota**Remy Sharp tem um muito eficaz vídeo mostrando a vantagem de transições de afunilamento e otimizações

Por meio de um dispositivo da GPU no YouTube em [www.youtube.com/watch?v=IKl78ZgJzm4](http://www.youtube.com/watch?v=IKl78ZgJzm4).

 *Limitar ou substituto animações com media queries*: Você pode configurar versões mais limitada de animações para dispositivos móveis através da criação de diferentes versões sob um @media consulta.

 *Assegurar que as páginas que contenham elementos animados a escala móvel completamente para baixo para tamanhos de ecrã*: animações que são cortados nas extremidades não será obviamente uma boa aparência ou executar eficazmente em um dispositivo móvel.

Integrando CSS3 Media Queries com SVG

Você também pode usar o media queries (entre outros recursos CSS3) ao alvo elementos em SVG.

Apenas como o Adobe PhotoShop e outros aplicativos gráficos, SVG inclui o conceito de múltiplas camadas. Isso significa que você pode mesclar vários desenhos em um arquivo SVG e interruptor a visibilidade de cada usando CSS.

Vamos voltar para a ideia de colocar ícones no canto superior direito da janela do navegador, mas torná-los SVG em vez; neste caso, uma série de desenhos de SVG agrupados diferentes tipos de carroçaria, incluindo o *mesomorph* *ectomorph e*tipos. Por razões de espaço, código apenas para o primeiro tipo de carroçaria é mostrado na lista simplificada 8-8.

**Lista 8-8.**Arquivo SVG de múltiplos níveis de desenho

<svg version="1.1" xmlns="Http://www.w3.org/2000/svg" width="142px" height="340px" NEGATOSCÓPIO="0 0 142 340">

<style>

G { visibilidade: oculto; }

G:alvo { visibilidade: visível; } </Estilo>

<g id="ectomorph">

<caminho d="M11.356,682.57c5.297,6.354 10.084 10.253,,,17.781 14.844,C,694.04318.907,15.905,690.475,11.356,682.57z"/>

. . . .

</g>

<g id="mesomorph">

<caminho d="M9.981,679.538c0,0-8.719,7.188-8.719,17.125 . . ." />

. . . .

</g>

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

Cada uma das luzes agrupadas SVG desenhos é na camada superior de um outro, como mostrado na Figura 8-7. As camadas são depois escondidos com CSS. A próxima linha na folha incorporado liga a visibilidade de um grupo se um URL objectivos. Você pode usar CSS para colocar o arquivo SVG como uma imagem de fundo em uma div, como mostrado na lista 8-9.



**Figura 8-7.***Diferentes camadas nomeadas de um desenho vectorial SVG sobrepor uma à outra*

**Lista 8-9.**Arquivo SVG de múltiplos níveis de desenho

Div#formas { background-image: Url('bodyshapes.svg#mesomorph'); posição: absoluta; superior: 0; Direita: 0; largura: 145px; altura: 355px; }

Usando a mesma técnica de identificação, você pode trocar a visibilidade das camadas no arquivo SVG usado na imagem de fundo do div (listando 8-10).

**Listando 8-10.**Arquivo SVG de múltiplos níveis de desenho

@tela mídia e (max-largura: 1000px) {

Div#formas { background-image: Url('bodyshapes.svg#ectomorph); }

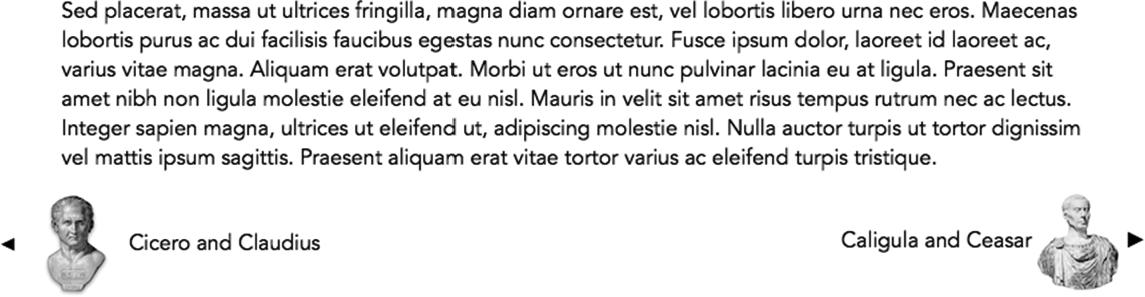
}

Por último, é de notar que os elementos SVG pode ser animado no e de si mesmos com a <animar /> elemento, a visibilidade de cada camada pode ser animado com as transições se a SVG foi inserido diretamente na página, em vez de ter sido usada como uma imagem de fundo.

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

Disparo transições CSS3 com JavaScript

JavaScript pode ser utilizado para disparar transições e animações em CSS3. Por exemplo, você poderia ter elementos fade in quando os usuários atingir a parte inferior de uma página para aumentar o seu interesse no conteúdo relacionado (ver Figura 8-8). CSS não pode detectar o estado da barra de rolagem - você precisa usar JavaScript para fazer isso e seguidamente animar os elementos que aparecem usando CSS3.



**Figura 8-8.***Elementos de rodapé animado com transições CSS desencadeada por JQuery*

■ **Observe**que há um argumento de que se você estiver iniciando uma animação com JavaScript, você bem que poderia continuar a criar a animação em que a mesma língua. No entanto, conforme discutido no Capítulo 1, transições CSS3 será mais suave e eficiente, e irá atingir uma taxa de quadros mais alta do que o que você pode conseguir em JavaScript.

Existe uma razão bibliotecas como JQuery trânsito (<http://ricostacruz.com/jquery.transit/>) e mover. js (<http://visionmedia.github.com/move.js>) estão sendo cada vez mais usados para gancho JavaScript em CSS3 métodos de transição e animação. Esta separação de funções reflecte uma separação entre conteúdo e apresentação e comportamento: neste exemplo, JavaScript é usado para detectar eventos DOM e CSS é usada para apresentar alterações na aparência para que o conteúdo.

Primeiro, vamos supor que você tenha mais do que suficiente para preencher o conteúdo de visualização: vou mostrar um cabeçalho e um fragmento de *Lorem ipsum* texto de enchimento na cotação de 8-11 para indicar o início do corpo do texto. Na parte inferior da página você terá dois links dentro de um elemento de rodapé. O primeiro link levará o usuário para o conteúdo que é logicamente "anterior" para a página que estão atualmente no; o segundo link, à direita, irá direcioná-los para a "próxima" página após o atual. Eu usei o Unicode preto à direita e à esquerda do triângulo apontando formas dentro dos links para poupar espaço: você deve usar as entidades HTML (&#9664; e &#9654; respectivamente).

**Listando 8-11.**Código HTML de uma página com conteúdo em Extra-Long um rodapé engatado com JavaScript e CSS3 transições

. . .

<script src="//Ajax.googleapis.com/ajax/libs/jquery/1.8/jquery.min.js"></script> </cabeça>

<body>

<artigo>

<plataforma><h1>Uma História do Império Romano</h1></plataforma> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. . .

<rodapé id="articlefooter">

<a href=# id=**◀ prevpage>**<img src=cicero.png alt=Cícero>Cícero e Cláudio</a> <a href=# id=nextpage>Caligula e Ceasar <img src=César.png alt=**▶ Caeasar>**</a>

</rodapé>

</artigo>

</body>

O rodapé deve ser empurrada abaixo da borda inferior da janela de visualização pelo conteúdo do artigo. Você está indo para a página Estilo, recuo os links no rodapé levemente e tornando-os invisíveis baixando sua listagem de opacidade (8-12).

**Listando 8-12.**Código CSS Base para uma página com navegação Rodapé

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Corpo { font-family: Avenir, sans-serif; | | | | Margem: 100px 0; } |  |  |
| Artigo { Largura: 768px; margin: 0 | | | Auto; | } |  |  |
| Rodapé#articlefooter | { | Padding: 0 | 25px; | } |  |  |
| Rodapé#articlefooter | Um | { |  |  |  |  |
| Text-decoration: nenhum; color: #000; | | | | Opacidade: 0; POSIÇÃO: relativos; |  |  |
| } |  |  |  |  |  |  |
| Rodapé#articlefooter um | | Img { largura: 77px; altura: 77ps; alinhar vertical: médio; } | | | **** |  |
| A#prevpage { PADDING-LEFT: 70px; bóia: esquerda; transição: 1s 1s Opacidade linear, 1s 1s translateX | | | | |  |

|  |  |  |
| --- | --- | --- |
| Linear; } | **** |  |
| A#nextpage { PADDING-RIGHT: 70px; bóia: direito; transição: 1s .5s Opacidade linear, 1s 1s |  |
| TranslateX linear; } |  |  |

Você já tenha associado o CSS3 código de transição com os links: se eles são alimentadas ao mesmo tempo, .linkmoveright, associado com a #nextarticle elemento, irá mover primeiro. Após um pequeno atraso, será seguido pelas

.linkmoveleft classe, associado com #prevarticle.

Note que você fez a transição mais eficiente por declarar as propriedades que estão mudando: porque são várias propriedades, você usar uma repetição separados por uma vírgula.

Em vez de associar as alterações para os elementos que será iniciada por suas transições com um mouse ou :foco :pseudo-selector, você está indo para definir como uma nova classe (consulte a listagem 8-13).

**Listando 8-13.**Transforma para rodapé elementos de navegação

.linkmoveleft { transformar: translateX(-70px); opacidade: 1; }

.linkmoveright { transformar: translateX(70px); Opacidade: 1; }

Por último, você vai adicionar um script na parte inferior da página que irá examinar algumas variáveis e juiz quando adicionar essas classes para os elementos.

Como usado na cotação de 8-14, articleheight variável determina a altura total do corpo, incluindo todo o seu conteúdo. scrollTop mede quantos pixels da página estão *acima* da borda superior da janela do navegador: este será 0 quando a página carregar, com o valor crescente como o usuário rola para baixo. Dividindo articleheight por 2 e comparar o resultado para scrollTop, você pode determinar quando o usuário tiver rolado através de meia página e depois aplicar a lista de classes (8-14).

**Listando 8-14.**JQuery código para colocar as aulas sobre elementos de Rodapé de navegação

<script> $(função() {

Var articleheight = $("corpo").altura(); $(janela).Role(função() {

Se ($(esta).scrollTop() > (articleheight / 2)) { $("#prevpage").toggleClass("linkmoveleft"); $("#nextpage").toggleClass("linkmoveright");

});

});

</script>

Transições CSS3 será fogo em resposta a *qualquer* alteração no estado dos elementos que estão associados com o , se essas alterações são impostas pelo CSS, JavaScript, ou qualquer outra coisa. Neste caso, a colocação de novas classes que contêm alterações aos elementos" opacidade e posição com JQuery é suficiente para definir desligado

As transições.

Enquanto isso funciona, olhando para os resultados de forma crítica revela que a abordagem tem várias possíveis inconvenientes:

 Artigos particularmente longo pode apresentam o corpo do texto que é mais de duas vezes a altura da janela do navegador. A comparação no script na cotação de 8-14 significa que para tais artigos as transições de maio de incêndio antes que o usuário tenha atingido a parte inferior da página (ou seja, após a leitura de mais de metade do artigo que ainda não podem ver o rodapé).

 A função JQuery toggleClass significa que irá tentar desfazer a aplicação das classes quando o usuário rola para cima e fogo-las novamente quando o usuário retorna para a parte inferior da página. Tais transições repetido poderia revelar importuno.

 Por último, estamos supondo que o rodapé será sempre ser congruente com a parte inferior da página. Isto não é necessariamente verdade: podem ser observações sob o rodapé, que iria alargar a altura total do artigo significativamente, causando o JavaScript para aplicar as classes mais cedo.

■ **nota**sob a especificação HTML5, artigo elementos aninhados dentro de um outro artigo são assumidos para conter comentários sobre o pai.

Como uma abordagem alternativa, aplicar as transições apenas uma vez quando o rodapé é claramente na página (listando 8-15).

**Listando 8-15.**Código JQuery melhorado para colocar as aulas sobre elementos de Rodapé de navegação

$(função() {

Var footerBottom = $("#articlefooter").Offset().top + $("#articlefooter").altura(); $(janela).Role(função() {

Se ($(esta).scrollTop() > (footerBottom - $(janela).altura())) { $("#prevpage").addClass("linkmoveleft"); $("#nextpage").addClass("linkmoveright");

}

});

});

Se você queria para suporte ao Internet Explorer 6 através de 8, você poderia usar Modernizr (<http://modernizr.com/>) para detectar o suporte do navegador de transições CSS. Se o navegador não dispunha de apoio, JQuery poderia cair de volta a animar os elementos em si (consulte a listagem 8-16).

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

**Listando 8-16.**Código JQuery melhorado para colocar as aulas sobre elementos de Rodapé de navegação

<script src=//Ajax.googleapis.com/ajax/libs/jquery/1.8/jquery.min.js></script> <script src=scripts/modernizr.js></script>

</cabeça>

<body>

<artigo>

. . .

</artigo>

<script> $(função() {

Var footerBottom = $("#articlefooter").Offset().top + $("#articlefooter").altura(); $(janela).Role(função() {

Se ($(esta).scrollTop() > (footerBottom - $(janela).altura())) { se (Modernizr.csstransitions) {

$("#prevpage").addClass("linkmoveleft"); $("#nextpage").addClass("linkmoveright");

} else {

$("#prevpage").animar({ opacidade: 1, esquerda: '-=70'}, 1000, função() {

});

$("#nextpage").animar({ opacidade: 1, esquerda: '+=70'}, 1000, função() {

});

}

}

});

});

</script>

Se você queria para criar um efeito mais complexo, poderia escrever animações keyframe e aplicá-las como classes, como na cotação de 8-16, ou injectar a chamada para as animações diretamente usando JQuery. Para um exemplo final, você usará a última abordagem, chamando animações keyframe que duplicam o efeito que você acabou de criar com as transições (listando 8-17).

**Listando 8-17.**JQuery código usado para aplicar CSS3 animações

@keyframes leftmove {

100% { transformar: translateX(-70px); Opacidade: 1; }

}

@keyframes rightmove {

100% { transformar: translateX(70px); Opacidade: 1; }

}

<script> $(função() {

Var footerBottom = $("#articlefooter").Offset().top + $("#articlefooter").altura(); $(janela).Role(função() {

Se ($(esta).scrollTop() > (footerBottom - $(janela).altura())) {

$("#prevpage").css('Animação', 'leftmove 1s 2s encaminha'); $("#nextpage").css('Animação', 'rightmove 1s 1s encaminha');

});

});

</script>

■ **Nota**técnica mostrado na listagem 8-17 pode ser usado para evitar o semanticamente discutível interatividade demonstrada no Capítulo 4 por meio do formulário e os  botões de seleção de rótulo. Em vez de saltar através de aros de marcação para atingir os resultados desejados em CSS, você pode simplesmente empregam JavaScript para detectar eventos DOM sobre qualquer elemento e iniciar transformações CSS3, transições e animações sobre outros, como mostrado na listagem 8-18.

**Listando 8-18.**JQuery código usado para aplicar um CSS3 Animação sobre clique

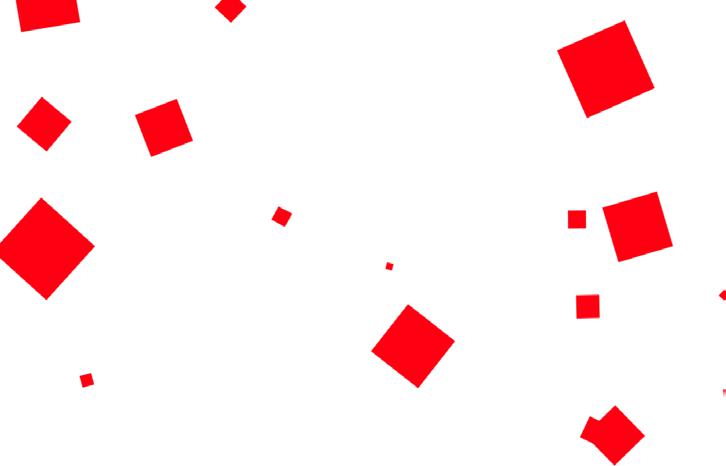
$(função() {

$("#a").Clique em função(() { $(".box").toggleClass("oscilação"); });

});

Personalizando as transições CSS3 com JavaScript

No capítulo 6 olhou para animar vários elementos SVG com CSS3. Ao fazer isso você encontrou uma grande questão: elementos para animar você tinha de os criar como elementos individuais na página, tornando difícil para criar a aparência de aleatoriedade (consulte a Figura 8 e 9).



**Figura 8-9.***Elementos randomizados animada com CSS3*

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

Em um esforço para reduzir a quantidade de codificação de mão o que você tem a fazer em qualquer página, você vai usar o JavaScript para resolver estas duas questões. Para manter as coisas simples você usar vermelho div elementos como a objetos animados. Na parte inferior de uma página que você vai adicionar o script mostrado na listagem 8-19.

**Listando 8-19.**Código JavaScript para criar elementos dispersos aleatória

<script>

Var html = [];

Para i = 0; i < 30; i++) {

Var randomX = Math.random() \* (100 - 1) + 1; var randomY = Math.random() \* (1200 - 1) + 1;

Var randomZ = Math.random() \* (100 - 1) + 1; **** html.push ('<div style="esquerda:"+randomX+'%;topo:"+randomY+'px;largura:"+randomZ+'px;altura: '+randomZ+'>

</div>');

}

$("corpo").append( html.junte(') ); </script>

Você usar um JavaScript loop para criar quantas cópias do elemento de div como desejar. Para cada elemento, você vai usar três variáveis para determinar a sua posição horizontal, offset a partir da borda superior da porta de visualização, juntamente com o tamanho horizontal e vertical.

No entanto muitos elementos você terá todos os siga as mesmas regras de animação keyframe mostrado na listagem 8-20.

**Listando 8-20.**Código JavaScript para criar elementos dispersos aleatória

Html { Height: 100%; }

Corpo { min-altura: 100%; margin: 0; } @keyframes floco {

100% { transformar: translateY(1800px) gire(1200deg); }

}

Div { background: vermelho; POSIÇÃO: absoluta; animação: floco 40s linear infinita; }

Embora isso funciona, há várias questões: todas as formas cair na mesma taxa e gire de forma síncrona. Um elemento que é pequeno parece estar mais longe e deve cair mais lentamente, enquanto todos os elementos deve começar em um delineamento de rotação. A fim de atingir este objectivo, você vai criar várias classes que será chamada a animação keyframe com temporizações diferentes e aplicar as classes com JavaScript, baseado no tamanho do elemento (listando 8-21).

**Listando 8-21.**Código JavaScript e CSS para criar elementos dispersos randomizado

<style>

@keyframes floco {

100% { transformar: translateY(1800px) gire(1200deg); }

}

Div { background: vermelho; POSIÇÃO: absoluta; animação nome: o floco de neve; animação de contagem de iteração: infinito; animação de função de distribuição linear:;

}

.pequenas { animação duração: 40s; }

.médio { animação duração: 20s; }

.grandes { animação duração: 10s; } </Estilo>

Capítulo 8 ■ Integração CSS3 ANIMAÇÃO COM RESPOSTA WEB DESIGN E JAVASCRIPT

<body>

. . .

</body>

|  |  |  |  |
| --- | --- | --- | --- |
| <script> |  |  |  |
| Var html = []; |  |  |  |
| Para i = 0; i < | 30; i++) { |  |  |
| Var randomX | = Math.random() \* (100 | - 1) | + 1; |
| Var randomY | = Math.random() \* (1200 - 1) + 1; | | |
| Var randomZ | = Math.random() \* (100 | - 1) | + 1; |
| Var randomR | = Math.random() \* (360 | - 1) | + 1; |
| Tamanhos var = | ['small','medium','grandes"]; | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Var dim = tamanhos[Math.round(randomX/50)]; | **** |  |  |
| Html.push ('<div style="esquerda:"+randomX+'%;topo:- | **** |  |
| '+randomY+'px;largura:"+randomZ+'px;altura:"+randomZ+'px;transformar:translateY(0px) | |  |

Gire('+randomR+'deg)" class="+dim+"></div>');

}

$("corpo").append( html.junte(') ); </script>

Este código JavaScript é básica e pode ser levada muito mais longe, mas o aspecto importante a ter em mente é que você está fazendo uso do núcleo pontos fortes de cada uma dessas tecnologias: você usar CSS para proporcionar regras da aparência e do JavaScript para fazer mudanças rápidas para o DOM.

Resumo

Este capítulo forneceu uma introdução para a integração de CSS3 animações, transforma e transições com resposta web design, JavaScript e SVG. Você pode criar a impressão de "animado" elementos em resposta ao redimensionamento de visualização pela redução de conteúdo usando unidades de percentagem e a VW e usando @media queries para animar os elementos de interrupção.

Você também pode usar o JavaScript para fornecer pontos de gatilho para animações que o CSS em si não pode detectar e usar a linguagem de scripting para fazer vários estudos randomizados "clones" de elementos para sequências de animação. A integração destas tecnologias pode ser levada muito mais longe: um bom exemplo é a solução para Markbåge Sebastian a dificuldade técnica de ter um elemento siga um caminho complexo a uma taxa constante de movimento usando SVG caminho dados para gerar a animação keyframe declarações (<http://csspathanimation.calyptus.eu/>).

Neste livro tão longe, você está usando CSS3 para mover os elementos em toda a superfície de plano da página. No próximo capítulo, você vai manipular elementos HTML no espaço 3D.

**Capítulo 9**

**CSS3 3D transforma, transições,**

**E animações**

Até agora você mudámo imagens, elementos de UI e outros conteúdos da web no plano da porta de visualização espaço planar. Mas usando a escala transformar não implica necessariamente que um elemento é mais longe a partir do visualizador, só que é maior ou menor; usando o padrão css falhar para conferir um verdadeiro sentido de perspectiva ou profundidade.

O 3D propriedades do CSS transforma tudo o que alterar o módulo. 3D transforma permitem que você manipule o conteúdo em uma projeção de espaço 3D. Esses elementos 3D-projetadas podem então ser animados usando a transição ou sintaxe keyframe que já experimentaste nos capítulos anteriores.

3D transforma são um aspecto do CSS3 que tendem a *não* degradar bem no navegador: se a 3D transformar declarações não são entendidas pelo cliente, elementos serão geralmente aparecem empilhados uns sobre os outros. Você deve ponderar cuidadosamente este antes de elementos de processamento em browsers mais antigos

Apesar de poderem ser utilizadas como tal, CSS 3D transforma não são destinados para tornar toda a sites da web com um "ambiente" 3D em que a janela do navegador gira entre uma página e outra. Em vez disso, as novas propriedades são normalmente utilizadas tanto como o resto da CSS é utilizado: para fazer escolhas de apresentação de alguns elementos em uma página da web. Criando e interagindo com toda a "mundos" de conteúdo 3D é melhor para a esquerda para o contexto do elemento de lona, através de tecnologias como WebGL.

Devido às suas demandas computacionais mais pesado, 3D transforma são pipelined diretamente para o cliente da GPU, fornecendo uma elevada taxa de imagens durante a animação.

Perspectivas financeiras

A chave para transformar elementos HTML no espaço 3D é a perspectiva, que pode ser especificado como um CSS propriedade:

#ngc-1763 { perspectiva: 600px; }

Ele também pode ser especificado como um valor em transformar:

#ngc-1763 { transformar: perspectivas financeiras(600px); }

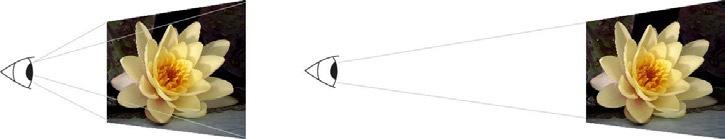
Ambos os métodos são exatamente equivalentes e produzem o mesmo resultado. Naturalmente, ambos os métodos requerem prefixos de fornecedor para obter suporte em versões mais antigas de navegadores.

Note que você não verá qualquer diferença após a aplicação da perspectiva sozinho. Definição de perspectivas financeiras é um precursor necessário para a manipulação de 3D, mas não alterar a aparência de um elemento em si.

Perspectiva pode ter qualquer valor de comprimento CSS positivo e define a *distância do ponto de vista do elemento de 3D*. Este pode ser o aspecto mais difícil da 3D transforma a compreender, e é por isso que esta propriedade merece maior atenção do que alguns outros. A perspectiva de propriedade está ilustrada na Figura 9-1.

Capítulo 9 ■ CSS3 3D transforma, transições e animações

(a imagem mostrada aqui é de Ola Wiberg, por [www.flickr.com/photos/powi](http://www.flickr.com/photos/powi), usados pela permissão, como são todas as fotografias utilizadas neste capítulo).



**Figura 9-1.***A perspectiva Cbanges em 3D transforma cria diferentes resultados visuais*

Na imagem à esquerda, o ponto de vista é relativamente perto os elementos afetados; a perspectiva adquirida pela definição para 200px é semelhante à visualização da cena através de uma lente macro da câmera. 3D alterações são susceptíveis de aparecer dramática e distorcida. Na imagem à direita, o ponto de vista (definida para 2000px) está distante de os elementos afetados, renderizar o conteúdo relativamente "flat", como se visualizado através de uma lente teleobjectiva. O visual se transforma no espaço 3D são mais susceptíveis de ser subtil e discreto.

Enquanto elementos individuais podem ter propriedades distintas perspectivas financeiras aplicadas a elas, isso implica que cada elemento tem o seu próprio visual origem e será exibido não relacionados com outros elementos no espaço 3D. Transformação dos elementos significa que cada um irá mostrar um outro ponto de fuga enquanto potencialmente tributar o navegador mais como ele calcula as orientações independentes de cada elemento. A menos que você estão deliberadamente a apontar para um *início-*como efeito, é recomendado que você aplique perspectiva apenas uma vez em um  elemento principal que contém o conteúdo que você deseja manipular no espaço 3D. A contendo div ou possivelmente o elemento de carroçaria em si tanto pode funcionar bem. Com a perspectiva de declaração no lugar sobre o elemento certo, 3D transforma de sua criança elementos irão criar uma aparência visual consistente, como mostrado na imagem do lado direito da Figura 9-2.



**Figura 9-2.***A perspectiva Cbanges em 3D transforma cria diferentes resultados visuais*

Apenas como 2D transforma têm uma origem de transformar, 3D transforma a ter uma perspectiva de origem de que os padrões para o centro do elemento. Enquanto não é errado colocar a perspectiva de origem noutras (usando qualquer comprimento CSS ou palavra-chave adequadas, horizontal seguido de localização vertical relativamente ao elemento), você deve ser

Capítulo 9 ■ CSS3 3D transforma, transições e animações

Ciente de que usando valores altos quando isso pode produzir distorção visual extremas, tais como efeitos isométrica. Um elemento individual com a sua própria perspectiva tem a sua perspectiva de origem "sepultados" dentro de si, cancelando o efeito da maioria dos 3D manipulações. Colocação da perspectiva e pela associação, a perspectiva de origem em elemento pai normalmente fornecerão melhor resultados visuais.

Vamos dar uma olhada em estes efeitos visualmente. Ambas as capturas de tela na Figura 9-2 compartilham o mesmo código básico mostrado na lista 9-1.

**Lista 9-1.**HTML e CSS compartilhada para um simples 3D-manipulado Galeria de Imagens

<style>

Html { Height: 100%; }

Corpo { min-altura: 100%; fundo: #333; margin: 60px; } img { Largura: 300px; altura: 300px; margin: 30px; }

</Estilo>

<body>

<img src=nenúfar.jpg alt="Gul Näckros Nenúfar"> <img src=red-tulip.jpg alt="Red Tulip">

<img src=dahlia.jpg alt=Dahlia> <img src=tulipas.jpg alt=Tulipas>

<img src=tarda-tulip.jpg alt="Tarda Tulip"> <img src=applebloom.jpg alt="applebloom Doce"> <img src=amarelo-tulip.jpg alt="Yellow tulip"> <img src=orchid.jpg alt=Orchid>

<img src=papoila.jpg alt="Blue Himalyan poppy"> </body>

Na Figura 9-2, a captura de esquerda tem perspectivas financeiras e a perspectiva de origem aplicadas para as imagens individuais, enquanto a direita screenshot mostra as mesmas propriedades aplicada ao corpo. Ambos os exemplos têm a mesma rotação aplicada às imagens.

Na imagem à esquerda, as imagens serão giradas *internamente* a si mesmos, mostrando sem mudança de perspectiva. À esquerda, as imagens serão giradas *em relação ao* corpo, transformada a partir de uma perspectiva coerente Vanishing Point.

É interessante notar que no segundo exemplo, o efeito 3D é ágil: porque a perspectiva de origem está no lado direito do corpo, ajuste a janela do navegador muda a posição de origem e assim altera o ponto de fuga para todos os elementos 3D. Alargamento e estreitamento o navegador fará com que os elementos de imagem para girar para que permaneçam em alinhamento com a perspectiva de mudança de origem.

Quando a colocação de perspectiva origem extra-consciente dos comportamentos predefinidos em CSS para elementos: recipiente elementos com elementos de criança não tendo flutuado altura, por exemplo, ou o corpo que está sendo a altura de seu conteúdo por padrão. Estes aspectos podem ter influências significativas quando colocar a origem e manipulação de conteúdo em um espaço 3D implícita.

Para demonstrar esse, tomar o código HTML na cotação de 9-1, mas substitui o CSS na cotação de 9-2.

**Lista 9-2.**CSS mostrando limitações imprevistos 3D Perspectiva CSS

Corpo { perspectiva: 1200px; perspectiva de origem: centro; margin: 2rem; }

Img { Largura: 500px; altura: 375px; bóia: esquerda; margin-right: 2rem; margin-bottom: 2rem; } img { transformar: rotateY(45graus); }

O código CSS define a perspectiva de origem para o elemento de carroçaria em seu valor padrão de centro para ambos os eixos X e Y , mas como você pode ver na Figura 9-3, o efeito nas imagens mostra que a origem não está no centro do documento, mas no topo.

Capítulo 9 ■ CSS3 3D transforma, transições e animações



**Figura 9-3.***Orientação incorreta de imagens devido à perspectiva de origem desalinhado*

Para obter a perspectiva sobre as imagens corretas, você deve fazer o elemento de carroçaria pelo menos tão elevado como o conteúdo que ele contém. Neste caso, você usará o padrão min-altura: 100% sobre o html e altura: 100% sobre a abordagem do corpo , que você pode ver na listagem 9-3. Uma vez que o valor CSS3 é suportado em navegadores, você será capaz de usar min-Altura: contêm-flutua sobre o elemento de carroçaria sozinho.

**Lista 9-3.**CSS Perspectiva 3D corrigida para a altura do corpo

Html { Height: 100%; }

Corpo { perspectiva: 1200px; perspectiva de origem: centro; margin: 2rem; margin-top: 8rem; min-altura: 100%; }

Img { Largura: 500px; altura: 375px; bóia: esquerda; margin-right: 2rem; margin-bottom: 2rem; } img { transformar: rotateY(45graus); }

A mudança para o CSS fornece a correta transforma a imagens, como você pode ver na Figura 9-4.

138

apítulo 9 ■ CSS3 3D transforma, transições e animações

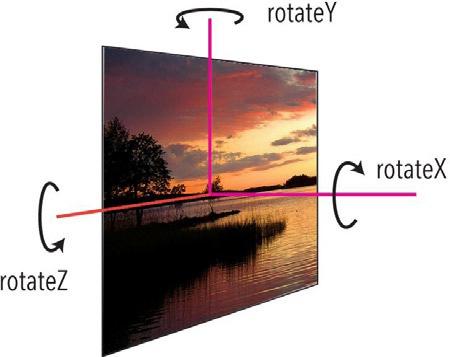


**Figura 9-4.***Orientação correta das imagens com altura corporal corrigido*

Como você pode ver, a colocação adequada de ambas as perspectivas financeiras e a perspectiva de origem, tanto no código e o espaço cartesiano de sua página da web, podem fazer uma diferença significativa para a forma como o seu conteúdo é exibida.

Rotação

Agora que você fez um pouco de rotação em 3D, você deve tomar um momento para explorar completamente. 2D transforma tem apenas uma rotação; não surpreendentemente, 3D tem três. O que pode ser confuso é a orientação e o efeito desses três eixos, mostrado na Figura 9-5.



**Figura 9-5.***Opções para a rotação no espaço 3D*

Como você pode ver, rotateX é efectivamente a mesma rotação CSS 2D, e se move *em torno* do eixo como um moinho de vento. rotateY oscila o elemento de lado a lado como uma porta e rotateZ se move em torno de um eixo horizontal, como um gato porta. Naturalmente, o efeito visual dessas transformações irão variar conforme a perspectiva de origem

É alterada.

Se o elemento for girado em torno de mais de um eixo de cada vez, geralmente é mais eficaz utilizar o gire3d atalho:

Img { transformar: gire3d( 0, 1, 0, 45 graus); }

Você pode pensar de girar3d como uma matriz multiplicador para cada um dos eixos (x,y,z): se qualquer um dos três primeiros slots tem qualquer tipo de valor numérico, o elemento será girado pelo último valor ao redor os eixos apropriados. Por exemplo, a declaração acima irá girar uma imagem por 45 graus em torno do eixo y. Você pode combinar várias rotações rodando em bits para vários eixos:

Img { transformar: gire3d( 1, 1, 0, .15volta); }

Traduzir

3D elementos transformou também pode ser movido ao longo dos eixos X, Y e Z. Note que isso irá mover o elemento para ou mais longe do seu ponto de fuga se o elemento é girado ao redor dos eixos X ou Y. ou movido ao longo do eixo Z. Por exemplo, o código na lista 9-4 pode criar o efeito visual de uma abertura de crawl para um filme com créditos a desaparecer no espaço infinito.

**Lista 9-4.**Código, Conteúdo e CSS para uma abertura de crédito Crawl

<style>

Corpo { background: #000; perspectiva: 700px; }

Div#crawl { largura: 80%; alinhar texto: centro; color: #fff; font-family: 'Franklin Gothic Medium", sans-serif; font-size: 4rem; margin: 0 auto;

Capítulo 9 ■ CSS3 3D transforma, transições e animações

Transformar: rotateX(30graus) translateY(-200px); } </Estilo>

<body>

<div id="crawl"> <h1>Colheita azul</h1>

<p>As ondas de grãos movido como a superfície do oceano sob a luz da lua, instigados fracamente por um susurrus de vento do oriente. Caleb içada o desgaste do eixo de madeira de uma foice sobre o seu ombro e deu uma olhada para fora na escuridão, sujeira-exibisse face miniatura riscar sob a borda do seu chapéu de palha, logo acima do sol-terminador grelhado de pele.</p>

</div>

Naturalmente, isto também poderia facilmente ser animado (listando 9-5).

**Lista 9-5.**CSS para um crédito de abertura animado Crawl

@keyframes crawl {

100% { transformar: rotateX(40graus) translateY(-2000px); }

}

Div#crawl { transformar: rotateX(40graus) translateY(1000px); animação: crawl 30s linear; }

Isto cria o resultado que você vê na Figura 9-6.



**Figura 9-6.***Ainda a partir de um crédito animados crawl criados com CSS3*

Você pode também empregar traduzir3d da mesma forma que o gire3d propriedade foi usado anteriormente.

141

Capítulo 9 ■ CSS3 3D transforma, transições e animações

■ **Observe**visualmente existe pouca diferença entre translateZ e scaleZ: ambos atingir o mesmo resultado, partindo do princípio de que nenhum outro transforma são aplicados. Caso contrário, a escala funciona da mesma forma que as outras propriedades, incluindo a escala3d.

Legenda de cartão Flip

Você pode usar uma combinação dessas 3D transforma para criar vários veriations de legendas de imagem no lado do "ré" imagens da galeria, revelado com o cursor do mouse (Figura 9-7).



**Figura 9-7.***3D imagem invertida galeria de legenda*

O primeiro exemplo que você vai criar irá usar transições com imagens em uma lista padrão. Todas as imagens são exactamente o mesmo tamanho, com legendas codificadas como span elementos depois de cada um, como mostrado na lista 9-6.

**Lista 9-6.**Código para um simples galeria de 3D

<ul id=gallery>

<li><img src=tulipas.jpg alt="Tulipas"> <span>Tulipas</span>

<li><img src=hepatica-nobilis.jpg alt=Nepatica-nobilis> <span style=padding-left:4rem>Nepatica nobilis</span> <li><img src=ekebyhosparken.jpg alt=Ekebyhosparken> <span style=padding-left:4rem>Ekebyhosparken</span>

</ul>

Neste exemplo, cada imagem flip horizontalmente no mouseover para mostrar a sua volta ou reversão face, como se você estivesse olhando através de uma transparência da imagem a partir do outro lado. Após uma pausa, o span conteúdo fade in sobre a parte superior desta imagem invertida (inline padding é usado simplesmente para mover o texto em torno dentro do espaço span). Mover o mouse para fora da imagem irá inverter esta sequência. (Consulte a listagem de 9-7).

**Lista 9-7.**CSS para um simples galeria de 3D

Corpo { background: #fff; font-family: Avenir, Arial, sans-serif; font-size: 1.5Rem; texto-sombra: 3px 3px 2px rgba(0,0,0,0.6); }

Ul#gallery { margin-top: 400px; perspectiva: 1000px; } ul#gallery li { Display: bloco inline; margin: 20px; } ul#gallery li img { Largura: 320px; altura: 244px;

Caixa-sombra: 0px 0px 16px rgba(0,0,0,0.3); transição: 1s todos lineares;

}

Ul#gallery li:foco img { transformar: rotateY(180deg); }

Ul#gallery li span { posição: absoluta; largura: 320px; altura: 244px;

Margem esquerda: -320 px; color: #fff; opacidade: 0; fundo: rgba(0,0,0,0.8); exibição: bloco em linha;

Caixa de calibragem: caixa de fronteira; padding-top: 6rem; PADDING-LEFT: 8rem; } ul#gallery li:hover span { opacidade: 1; transição: 1s 1s Opacidade linear; }

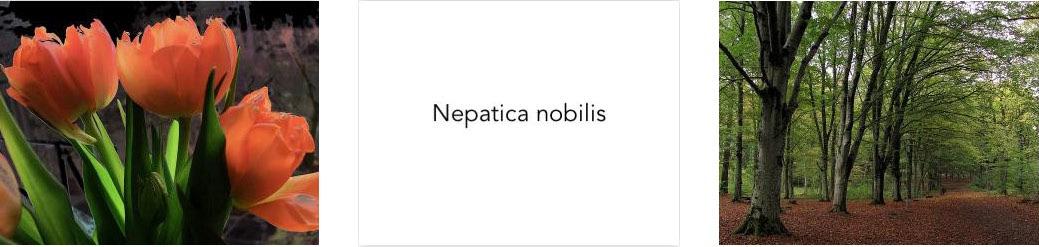
O elemento span usa o valor de duração da transição para vir em imediatamente após o spin foi concluída. Cada spin é em torno do centro da própria imagem; se você queria introduzir um pouco mais variabilidade, você poderá alterar a transição de origem para cada elemento.

Como um experimento, deixe o span elementos visíveis e untransitioned no seu CSS; você verá que a fiação *através de imagens, gire* o texto da legenda, um efeito que você terá a oportunidade de usar mais tarde. Um pouco como o índice z, untransformed conteúdo é assumida a existir em uma camada de base de "" de apresentação que o 3D-projetadas conteúdo pode auxiliar através, consoante a sua posição e perspectiva.

Também é interessante notar que a caixa da imagem-sombra *não é* mostrado como sendo jogado sobre a página em branco do seu corpo conteúdo; em vez disso, a sombra gira como parte da imagem, ao redor. Embora seja possível em teoria para manipular uma "sombra" div para fornecer a impressão de que é projectado, isto demora considerável esforço extra.

Como um todo, a galeria funciona bem, mas os visitantes a ver o "outro lado" da imagem podem ser confundidos.

(Consulte a Figura 9-8.) Fornecendo a impressão de que as suas imagens são cartões real com um simples verso que detém uma legenda com todos estes elementos se movendo juntos, requer um pouco mais de trabalho.



**Figura 9-8.***3D imagem invertida galeria de legenda com sólidos reverso*

Primeiro, você vai alterar o markup para fornecer uma página variação com mais valor semântico, mostrado na lista 9-8.

**Lista 9-8.**Para um código mais complexo Galeria de 3D

<figura>

<figcaption>Tulipas</figcaption> <img src=tulipas.jpg alt=Tulipas>

</figura>

<figura>

<figcaption>Nepatica nobilis</figcaption>

<img src=hepatica-nobilis.jpg alt="Nepatica nobilis"> </figura>

<figura>

<figcaption>Ekebyhosparken</figcaption>

<img src=ekebyhosparken.jpg alt=Ekebyhosparken> </figura>

Nota a ordem de origem: Cada legenda da imagem vem *antes de* a imagem referências. Isso é perfeitamente válido em HTML5, e será útil para a sua animação CSS. Você poderia colocar as legendas após as imagens, mas que seria necessário alterar o valor do índice z de legendas. O código CSS começa de uma forma semelhante ao último exemplo (consulte a listagem de 9-9).

**Lista 9-9.**Código CSS Base para uma galeria de 3D mais complexas

Corpo { background: hsl(100,0%,100%); font-family: Avenir, sans-serif; font-size: 1.5Rem;

|  |  |  |
| --- | --- | --- |
| Margem superior: 400px; | Perspectivas financeiras: 1000px; | } |
| Figura { margin: 20px; | Transição: 1s todos lineares; | |
| Caixa-sombra: 0px 0px 16px rgba(0,0,0,0.3); bóia: esquerda; } | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Figura, figura | | Img, figcaption | { Largura: 320px; altura: 244px; caixa de calibragem: caixa de fronteira; } | |
| Figura img, figcaption { posição: absoluta; } | | | |  |
| Figcaption { | Fundamento: #fff; | | Alinhar texto: centro; padding-top: 6rem; | } |
| Figura:Foco | { | Transformar: rotateY(180deg); } | |  |

A legenda da imagem pares são ambos exactamente o mesmo tamanho e posicionada absolutamente. Porque as fotografias vêm *depois* a legenda, eles são exibidos "acima" do texto. No entanto, girando cada figura como um todo nesta fase mostra somente o lado reverso da imagem e não o texto.

Rejeição de fundo de face Visibilidade

Os elementos de rotação em torno dos eixos Y e Z traz uma questão interessante: o que acontece quando você gira um elemento mais de 180 graus? O que está no outro lado de uma imagem que tinha, até este momento, uma efectiva "espessura" da 0?

Aparência neste estado é determinada pela rejeição de fundo de face de visibilidade a propriedade que tem um valor padrão de visíveis. Isto significa que quando um elemento vira passado 180, o que é mostrado o outro lado é o *inverso* do que é visto na orientação original do elemento: imagens se tornem efectivamente transparências, igualmente visível em ambos os lados, e o texto é processado espelho refeita. Rodando Rejeição de fundo de face de Visibilidade para Oculto provoca o renderizador para ignorar o outro lado do elemento quando ele gira 180 graus; na maioria dos casos isso significa menos trabalho para o renderizador e, como resultado, animações suaves e limpas.

Transformar o estilo

Por padrão o navegador irá supor que as crianças de um elemento transformou 3D tomar no 3D transformações de seus pais, mas são projectados no mesmo plano: não há "antes" ou "atrás" sentido para tais elementos filhos. Formalmente, este seria declarado como transformar-style: televisão aplicada ao elemento pai.

Se desejar os elementos filhos de um 3D-elemento manipulado para mover no mesmo espaço 3D como seus pais mas ser projectados no seu próprio avião, um valor de preservar-3d *deve ser aplicada ao elemento pai*. Isso permite que você coloque um elemento filho atrás do outro, para que o cartão pode ter duas visíveis "lados".

Neste caso não é mais um complemento para tornar o código-juntos essas propriedades não será, por si só, ser suficiente para fornecer a impressão de frente e verso de um cartão. Para a legenda para ser apresentado corretamente no outro lado da imagem, ele deve ser movido horizontalmente *antes de* qualquer animação tem lugar. (pense duas cartas de jogar sendo colocados de volta para trás de modo a que os seus rostos são visíveis em ambos os lados).

Seu código muda para o que você pode ver na cotação de 9-10.

**Listando 9-10.**Complete o código CSS para uma galeria de 3D mais complexas

Corpo { background: hsl(100,0%,100%); font-family: Avenir, sans-serif; font-size: 1.5Rem; margin-top: 400px; perspectiva: 1000px; }

Capítulo 9 ■ CSS3 3D transforma, transições e animações

Figura { margin: 20px; transição: 1s todos lineares; caixa-sombra: 0px 0px 16px rgba(0,0,0,0.3); bóia: esquerda; transformar-style: preservar-3d; }

Figura, figura img, figcaption { Largura: 320px; altura: 244px; caixa de calibragem: caixa de fronteira; } figura img, figcaption { posição: absoluta; rejeição de fundo de face de visibilidade: oculto; }

Figcaption { background: #fff; alinhar texto: centro; padding-top: 6rem; transformar: rotateY(180deg); }

Figura:foco { transformar: rotateY(180deg); }

Há uma variação mais a considerar: aquilo que se deseja que o outro lado da fotografia a ser fracamente visível através da função closed caption? Desligar a rejeição de fundo de face de visibilidade: ocultos não serão o suficiente para criar esse efeito por si própria. Você precisará posicionar cada elemento no espaço 3D com um pouco mais de precisão (consulte a listagem 9-11).

**Listando 9-11.**Completa o código CSS alternativo para uma galeria de 3D mais complexas

Corpo { background: hsl(100,0%,100%); font-family: Avenir, sans-serif; font-size: 1.5Rem; margin-top: 400px; perspectiva: 1000px; }

Figura { margin: 20px; transição: 1s todos lineares;

Transformar-style: preservar-3d; caixa-sombra: 0px 0px 16px rgba(0,0,0,0.3); bóia: esquerda; }

|  |  |  |  |
| --- | --- | --- | --- |
| Figura, figura img, figcaption | { Largura: 320px; altura: 244px; caixa de calibragem: caixa de fronteira; | | } |
| Figura img, figcaption { posição: absoluta; | | } |  |
| Figcaption { background: #fff; | Alinhar texto: centro; padding-top: 6rem; | |  |
| Transformar: rotateY(180 graus) | TranslateZ(1px); opacidade: 0,9; | |  |
| } |  |  |  |
| Figura:foco { transformar: rotateY(180deg); | | } |  |

A tradução muito ligeira da legenda ao longo do eixo z é o suficiente para colocar a legenda "atrás" a imagem de tal forma que a inversão do aspecto da fotografia abaixo pode ser visto como mostrado na Figura 9-9.



**Figura 9-9.***3D imagem invertida galeria de legenda com parcialmente transparente reverso*

Uma Circular Galeria de 3D

Você pode tomar todas as fotografias utilizadas até agora e os transformam em uma galeria de 3D totalmente circular que gira sobre o mouse (Figura 9). O markup, mostrado na lista 9-12, é muito simples.



**Figura 9.***Uma circular Animado galeria de 3D*

**Listando 9-12.**Código HTML para uma circular Galeria de 3D

<div>

<figura>

<img src="färentuna-igreja.jpeg" alt="Färentuna igreja Färingsö, Suécia"> <img src="árvores de inverno.jpeg" alt="árvores no inverno, Suécia">

<img src="visualizar-de-rastaholm.jpeg" alt="Vista do Rastaholm, Suécia">

<img src="queda-park.jpeg" alt="Park em queda"> <img src="sunset2.jpeg" alt="Sunset"> <img src="tulipas2.jpeg" alt="Tulipas">

<img src="hepatica-nobilis.jpg" alt="Nepatica-nobilis"> <img src="sunset.jpg" alt="Sunset">

<img src="ekebyhosparken.jpg" alt="Ekebyhosparken"> </figura>

</div>

Você vai empurrar essas imagens em 3D em um círculo posicionando-os absolutamente, de modo a que todos eles pilha na parte superior de cada um dos outros, localizando sua comum transformar origem-z "voltar" no espaço 3D e rotação de cada independentemente em torno do eixo Y. Como existem oito imagens, cada um irá ser girado em incrementos de 45 graus para criar um círculo uniformemente distribuída de fotografias (listando 9-13).

**Listando 9-13.**Código CSS Base para uma circular Galeria de 3D

Div { perspectiva: 1000px; margin-top: 400px; largura: 1400px; }

Figura { transformar-style: preservar-3d; altura: 244px; transformar origem-x: 660px; } img { Largura: 320px; altura: 244px; POSIÇÃO: absoluta; esquerda: 500px; }

Figura, img { transformar origem-z: -500px; }

Div img:nth-criança(1) { transformar: rotateY(0ºC); } div img:nth-Child(2) { transformar: rotateY(-45graus); } div img:nth-Child(3) { transformar: rotateY(-90 deg); } div img:nth-criança(4) { transformar: rotateY(-135 graus); } div img:nth-Child(5) { transformar: rotateY(-180 graus); } div img:nth-Child(6) { transformar: rotateY(-225 graus); } div img:nth-filho(7) { transformar: rotateY(-270deg); } div img:nth-criança(8) { transformar: rotateY(-315deg); }

Note que alterar a altura do elemento figura Altera a coordenada Y da galeria origem, pichações o círculo de imagens para cima e para baixo no espaço 3D. Na metade da altura completa das imagens dentro dele, o círculo de fotografias parece ser em nível de "olho" com o visualizador.

146

Capítulo 9 ■ CSS3 3D transforma, transições e animações

Observe também que você poderia usar JavaScript para automatizar a distribuição de imagens que você vai fazer na próxima seção.

Existem dois caminhos que você pode tomar para girar o gallery no planar. Um é para aumentar o valor de cada indivíduo rotateY imagem pelo mesmo montante: uma abordagem excessivamente complexa sem recorrer a scripts. Como você pode ver, já definidas as imagens e a figura que rodeia a terem a mesma origem z, de modo a segunda opção é para animar a figura em si (listando 9-14).

**Listando 9-14.**Animação CSS para um código 3D Circular Gallery

@keyframes spin { 100% { transformar: rotateY(360 graus); } } figura:foco { animação: spin 12s linear infinita; }

Enquanto o 3D e os efeitos de animação funciona bem, a Gallery é falta um outro aspecto do realismo: na vida real, objectos distantes não são apenas menores, eles também são reóstato. Ocultando a backfaces das imagens não funciona: que irá tornar as imagens desaparecem, Não murchará e adicionando abrange por detrás de cada imagem, por a solução para a primeira galeria, é demasiado extra markup. Nem o filtro de desfocagem de trabalho sobre os valores dos pixels 3d-transformado imagens, pelo menos não como este por escrito.

Existem algumas formas possíveis para atingir o efeito: a quarta e a sexta de imagens que podem ser vistos em "Back" do círculo galeria poderia ser dinamicamente com opacidade e fez sólido como eles vêm para a frente. Alternativamente, uma caixa-sombra aplicados igualmente a todos os lados de cada imagem com poderia ser aplicada sem desfocagem

(listando 9-15).

**Listando 9-15.**CSS efeito de sombreamento para uma circular Galeria de 3D

Img { box-sombra: 0px 0px 0px 48px rgba(0,0,0,0.8); }

A limitação dessa abordagem é que o fundo de cor da figura contendo div ou o próprio corpo deve ser preto sólido; nada mais leve irá mostrar as bordas das sombras. (Consulte a Figura 9-11).



**Figura 9-11.***Uma Circular Galeria 3D com efeitos de sombra*

Como cada imagem permanece efectivamente independentes como seu próprio elemento, você pode combinar a legenda técnicas exploradas no Capítulo 3 com as fotografias na galeria circular, embora com algumas alterações para o markup (consulte a listagem 9-16).

Capítulo 9 ■ CSS3 3D transforma, transições e animações

**Listando 9-16.**Para um código 3D Circular Galeria de imagens com legendas de pop-up

<style> @keyframes spin {

100% { transformar: rotateY(360 graus); }

}

Corpo { background: preto; font-family: Geórgia, serif; font-style: italic; font-size: 1.2Rem; } div#gallery { perspectiva: 1000px; margin-top: 400px; largura: 1400px; }

Div#interior{ transformar-style: preservar-3d;

Transformar a origem-x: 660px; transformar origem-z: -500px; altura: 244px; } figura, img { Largura: 320px; altura: 244px; }

Figura { esquerda: 500px; POSIÇÃO: absoluta; transbordamento: oculto; transformar origem-z: figcaption -500px; } { background: hsla(0,0%,0%,0,5); color: #fff; POSIÇÃO: relativos; superior: 0; padding: 8px;

Transição: 0.6s topo linear; } figura:foco figcaption { topo: -38px; }

Div#gallery:foco div#interior { animação: spin 24s linear infinita; } div#gallery figura:nth-criança(1) { transformar: rotateY(0ºC); } div#gallery figura:nth-Child(2) { transformar: rotateY(-45graus); } div#gallery figura:nth-Child(3) { transformar: rotateY(-90 deg); } div#gallery figura:nth-criança(4) { transformar: rotateY(-135 graus); } div#gallery figura:nth-Child(5) { transformar: rotateY(-180 graus); } div#gallery figura:nth-Child(6) { transformar: rotateY(-225 graus); } div#gallery figura:nth-filho(7) { transformar: rotateY(-270deg); } div#gallery figura:nth-criança(8) { transformar: rotateY(-315deg); } </Estilo>

<body>

<div id="Gallery"> <div id="interno"> <figura>

<img src="färentuna-igreja.jpeg" alt="Färentuna igreja Färingsö, Suécia"> <figcaption>Färentuna Igreja, Färingsö, Suécia</figcaption>

</figura>

<figura>

<img src="árvores de inverno.jpeg" alt="árvores no inverno, Suécia"> <figcaption>árvores no inverno, Suécia</figcaption>

</figura>

<figura>

<img src="visualizar-de-rastaholm.jpeg" alt="Vista do Rastaholm, Suécia"> <figcaption>Vista de Rastaholm, Suécia</figcaption>

</figura>

<figura>

<img src="queda-park.jpeg" alt="Park em queda"> <figcaption>Sueco park no Outono</figcaption>

</figura>

<figura>

<img src="sunset2.jpeg" alt="Sunset"> <figcaption>Sunset</figcaption>

</figura>

148

<figura>

<img src="tulipas2.jpeg" alt="Tulipas"> <figcaption>Tulipas</figcaption>

</figura>

<figura>

<img src="hepatica-nobilis.jpg" alt="Nepatica-nobilis"> <figcaption>Nepatica nobilis</figcaption>

</figura>

<figura>

<img src="sunset.jpg" alt="Sunset"> <figcaption>Sunset</figcaption>

</figura>

<figura>

<img src="ekebyhosparken.jpg" alt="Ekebyhosparken"> <figcaption>Ekebyhosparken</figcaption>

</figura>

</div>

</div>

</body>

Você pode ver um ainda do resultado na Figura 9-12.



**Figura 9-12.***Uma circular 3D galeria com legendas de pop-up*

Reforço da galeria com seletores de Nível 4 e JavaScript

Existem muitas maneiras de aprimorar a galeria que você criou com JavaScript. O óbvio e forma primária é cuidar da rotação inicial das imagens com scripts em vez de fazer isso manualmente. (Por razões de simplicidade, vou assumir que você voltou para a marcação de base da galeria mostrado na listagem 9-17).

**Listando 9-17.**Código JQuery para distribuir automaticamente as imagens em um CSS3 Galeria Circular

<script>

Var numberOfImgs = $("figura img").O comprimento; var degreeSep = 360 / (numberOfImgs - 1); var ângulo = 0;

Para i = 1; i < numberOfImgs; i++) {

14

$("figura img:nth-filho(" + (i) + ")").css('transformar','rotateY('+ ângulo +'deg)'); ângulo ângulo = + degreeSep;

}

</script>

Isso eliminaria oito linhas repetitivas de CSS e produzir um resultado muito mais flexível: se as imagens foram adicionados ou removidos de dentro da figura, sua distribuição será ajustado automaticamente. Uma versão mais avançada do script pode usar trigonometria para determinar que a base do tamanho e do número de imagens e a origem-z distância foram o suficiente para fornecer o espaçamento aceitável entre fotografias; se não, a largura e altura da imagem e/ou origem-z distância poderia ser modificado por JavaScript.

Segundo, você pode permitir que o usuário para o spin a galeria manualmente por detectar a posição do mouse dentro do elemento figura e use seu deslocamento horizontal do centro para determinar a quantidade de rotação para a galeria (listando 9-18).

**Listando 9-18.**Código JQuery para rodar uma CSS3 Galeria circular com base na posição do Mouse

<script> $("figura").mousemove(função(e) {

Var relativeX = e.pageX - (esta.offsetLeft + 660); $(esta).css('transformar','rotateY('+ relativeX +'deg)');

});

</script>

Isso significaria retirar o CSS3 animação e substituindo em uma transição, como mostrado na listagem 9-19.

**Listando 9-19.**O código CSS para JQuery Mouse-Position Rotação de uma Galeria Circular

Figura { transformar-style: preservar-3d; transformar origem-x: 660px; transformar origem-z: -500px; altura: 244px; transição: 2s transformar linear; }

Figura:foco { cursor: ew redimensionar; }

■ **Observe que**esta poderia também ser realizado para algum grau usando CSS puro com dois elementos, cada a altura da figura mas metade da largura, colocado à esquerda e direita sobre a figura, com um mouse pseudo-selector em cada uma condução em sentido horário e anti-horário animação respectivamente. Mas há duas desvantagens significativas: não pode haver interatividade direta com a galeria abaixo (ou seja, sem legendas de pop-up), como os dois elementos sobrepostos que capturar todos os eventos do mouse e 2) usando CSS puro limitaria a animação para um ajuste a velocidade da rotação em qualquer direção, em oposição ao mais longe = rotação mais rápida da solução de JavaScript examinámos anteriormente.

Versões mais avançadas do presente JavaScript e as versões mais compatíveis com as convenções de interface móvel-pode incluir um script para medir um mouse e/ou fingertip arraste movimento na tela, girando a galeria circular um número proporcional de graus. Você também pode adicionar botões esquerda-direita para girar a galeria em incrementos.

Adicionando um Nível de CSS Seletor-4

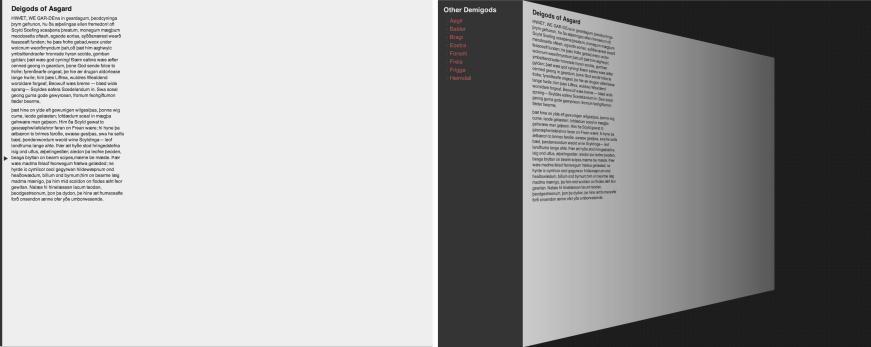
Uma vez que é suportado pelos navegadores, você poderia fazer a galeria mais exploratório usando o seletor de pai:

$figura img:nth-Child(2):foco { transformar: rotateY(-45graus); }

Isso iria girar a galeria circular como um todo para trazer a actual imagem pairava na frente e no centro.

3D CSS transforma e transições para elementos de UI

Como regra geral, o requisito para a navegação clara significa que elementos de UI deve empregar como poucos visual "truques" quanto possível. A técnica a seguir, derivados do trabalho iniciado pelo Hakim El Hattab (<http://hakim.se>, usadas com permissão), usa a abordagem oposta: quando a navegação é focada, o resto da página é a transição em 3D. Uma variação do código é mostrado na listagem 9-20. Você pode ver os resultados na Figura 9-13.



**Figura 9-13.***O conteúdo do corpo transformada com CSS3 com side-puxe a navegação*

**Listando 9-20.**Código HTML para navegação Side-Pull

<!DOCTYPE HTML> <html lang="pt"> <cabeça>

<meta charset="utf-8">

<título>3D dobrável menu</title>

<link rel="stylesheet" href="cunha.css"> </cabeça>

<body>

<div class="meny"> <h2>Outros Semideuses</h2> <ul>

<li><a href="#">Aegir</a> <li><a href="#">nórdico Balder</a> <li><a href="#">Bragi</a> <li><a href="#">Eostra</a>

</ul>

</div>

<div class="meny-seta"> </div> <div class="meny-conteúdo">

<div class="tampa"></div> <artigo>

<h1>Deigods de Asgard</h1>

<p>HWAET, GAR-DEna em geardagum, þeodcyninga þrym gefrunon. . . . .

</artigo>

Capítulo 9 ■ CSS3 3D transforma, transições e animações

</div>

</body>

<script src="meny.js"></script> </html>

Folha anexa é mostrado na lista 9-21.

**Listando 9-21.**Folha de estilos CSS para navegação Side-Pull

\*{ margin: 0; padding: 0; }

Html, corpo { altura: 100%; transbordamento: hidden; }

Corpo { background-color: #222; font-family: Lato, Helvetica, sans-serif; font-size: 16px; color: #222; }

.meny-wrapper {perspectiva: 800; perspectiva de origem: 0% 50%; }

.meny, .meny-conteúdo { de dimensionamento de caixa: caixa de fronteira; transição: transformar .5s facilidade; transformar-origem: 0% 50%; }

.meny { DISPLAY: none; POSIÇÃO: fixo; altura: 100%; largura: 300px; índice z: 1; margin: 0px; padding: 20px;

Transformar: rotateY( -30graus ) translateX( -97% ); }

.meny-ready .meny { Display: bloco; }

.meny-activa .meny { transformar: rotateY( 0deg ); }

.meny-conteúdo { background: #eee; padding: 20px 40px; largura: 100%; altura: 100%; transbordamento y: auto; }

.meny-activa .meny-conteúdo { transformar: translateX( 300px ) rotateY( 15graus ); }

.meny-conteúdo .cobrir { Display: nenhum; POSIÇÃO: absoluta;

Largura: 100%; altura: 100%; superior: 0; esquerda: 0; visibilidade: oculto; índice z: 1000;

Opacidade: 0; fundo: linear-gradiente(esquerda, rgba(0,0,0,0.15) 0%,rgba(0,0,0,0.65) 100%);

Transição: todos os arquivos .5s facilidade; }

.meny-ready .meny-conteúdo .cobrir { Display: bloco; }

.meny-activa .meny-conteúdo .cobrir { visibilidade: visível; opacidade: 1; }

.meny-seta { posição: absoluta; superior: 45%; esquerda: 12px; índice z: 10; font-family: sans-serif; font-size: 20px; color: #333;

Transição: esquerda 0.8s metros cúbicos linhas vectoriais assim(0,680, -0.550, 0,265, 1.515); }

.meny-activa .meny-seta { esquerda: -40px; opacidade: 0; }

.meny-dobra .meny, .meny-dobra .meny-conteúdo {transição: transformar 0.6s facilidade; }

.meny-dobra .meny-conteúdo {: posição fixa; índice z: 3; }

.meny-dobra .meny {transformar-origem: 50% 50%; }

.meny-dobra .meny.tampa direita .{ posição: absoluta;

Largura: 100%; altura: 100%; esquerda: 0; superior: 0; opacidade: 1;

Justificativa: linear-gradiente(para a direita, Rgba(0,0,0,1) 0%,rgba(0,0,0,0) 90%); }

.meny-ready .meny-dobra .meny.tampa direita .{ transição: opacidade 0.6s facilidade; }

.meny-activa .meny-dobra .meny.tampa direita .{ opacidade: 0; }

.meny-dobra .meny.esquerda { clip: rect( 0px, 150px, 10000px, 0px );

Transformar: traduzir3d( -150px, 0, -300px ) rotateY( 90 deg ) escala(1,25); }

.meny-activa .meny-dobra .meny.esquerda { clip: inicial; }

.meny-dobra .meny.direito { clip: rect( 0px, 300px, 10000px, 150px );

Transformar: traduzir3d( -150px, 0, -300px ) rotateY( -90graus ) escala(1,25); }

.meny-activa .meny-dobra .meny.esquerda, .meny-activa .meny-dobra .meny.direito { transformar: rotateY( 0deg ); }

Um { color: #c2575b; text-decoration: nenhum; transição: 0,15s cor facilidade; } Uma:foco { color: #f76f76; }

152

Capítulo 9 ■ CSS3 3D transforma, transições e animações

H1 { font-size: 24px; }

.meny { background: #333; color: #eee; }

.meny ul { margin-top: 10px; }

.meny ul li { lista-style: nenhum; font-size: 20px; padding: 3px 10px; }

.meny ul li:antes { conteúdo: '-'; margin-right: 5px; color: rgba( 255, 255, 255, 0,2 ); }

.meny-conteúdo>artigo { max-Largura: 400px; }

.meny-conteúdo p { margin: 10px 0 0 10px; font-size: 16px; altura: 1,32; }

Por último, o JavaScript,que cobre o cursor, toque e deslize eventos é mostrado na lista de 9-22.

**Listagem de 9-22.**JavaScript para navegação Side-Pull

(Função(){

Var meny = document.querySelector( '.meny' ); se (!meny || !meny.parentNode ) { } var menyWrapper retorno; = meny.parentNode; menyWrapper.className += ' meny-wrapper";

Var indentX = menyWrapper.offsetLeft, activateX = 40,

DeactivateX = meny.offsetWidth || 300, touchStartX = nulo,

TouchMoveX = null, banco estiveractivo = false, isMouseDown = falso;

Document.addEventListener( 'mousedown", onMouseDown, false ); document.addEventListener( 'mouseup", onMouseUp, false ); document.addEventListener( 'mousemove", onMouseMove, false ); document.addEventListener( "touchstart", onTouchStart, false ); document.addEventListener( "touchend", onTouchEnd, false ); window.addEventListener( "hashchange", onHashChange, false ); onHashChange();

Document.documentElement.className += ' meny-pronto"; função evento onMouseDown( ) { isMouseDown = true; } função onMouseMove( caso ) {

Se( !isMouseDown ) { var x = event.clientX - indentX;

Se (deactivateX ) { Desativar(); }else if( x < activateX ) {activar(); } }

}

Evento onMouseUp função( ) { isMouseDown = false; }

Função onTouchStart( ) { touchStartX evento = Evento.toca[0].clientX - indentX; touchMoveX = nulo;

Se (banco estiveractivo || touchStartX < activateX ) { document.addEventListener( "touchmove", onTouchMove, false ); }

}

Função onTouchMove( caso ) {

TouchMoveX = Evento.toca[0].clientX - indentX;

Se ( banco estiveractivo && touchMoveX < touchStartX - activateX ) { Desativar(); event.preventDefault();

}o resto se ( < activateX touchStartX && touchMoveX > touchStartX + activateX ) { activar();

Event.preventDefault(); } } função onTouchEnd( caso ) {

153

Capítulo 9 ■ CSS3 3D transforma, transições e animações

Document.addEventListener( "touchmove", onTouchMove, false ); se (touchMoveX === null ) {

Se (touchStartX > deactivateX ) { Desativar(); }

Else If (touchStartX < activateX \* 2) { activar(); }}}

Função onHashChange( caso ) {

Se( window.location.hash.match( "dobrar" ) && !document.BODY.className.match( 'meny-dobra" ) ) { addClass( Documento.body, 'meny-dobra" );

Var clone = document.createElement( "IV" ); clone.className = "eny direito";

Clone.innerHTML = meny.innerHTML + '<div class="tampa"></div>'; document.BODY.appendChild( clone );

AddClass( meny, "esquerda" ) ; } else {

RemoveClass( Documento.body, 'meny-dobra" );

Var clone = document.querySelector( '.meny.direito" ); se (clone ) {

Clone.parentNode.removeChild( clone ) ; } } }

Ativação de função() { se (banco estiveractivo === FALSE ) { banco estiveractivo = true;

AddClass( document.documentElement, 'meny-activa " ) ; }}

Desativação de Função() {

Se( banco estiveractivo === TRUE ) {

Banco Estiveractivo = falso;

RemoveClass( document.documentElement, 'meny-activa " ) ; }

}

Função addClass( elemento, nome ) {

Elemento.className = elemento.className.replace( /\s+$/gi, '' ) + ' ' + nome; }

Função removeClass( elemento, nome ) {

Elemento.className = elemento.className.replace( nome, '" );

}

})();

Reuniu, HTML, JavaScript e CSS criar o efeito que você vê na Figura 9-10 para navegadores em desktops e dispositivos móveis.

Resumo

3D transforma, transições e animações podem ser utilizados para adicionar profundidade e perspectiva ao seu conteúdo na web. Enquanto pode ser difícil visualizar o eixo z no contexto de uma televisão ecrã 2D, utilização de princípios discutidos aqui e nos capítulos anteriores - incluindo a manipulação da origem e transformação em torno de um ponto de vista, utilização do trabalho forçado, Rejeição de fundo de face de visibilidade, e transformar o estilo pode ajudar muito.

Até agora, você foi produzir todos os seu código manualmente. Enquanto que a melhor maneira de aprender, CSS geração de código pode ser feita de forma muito mais rápida e mais eficiente com ferramentas bem configurada para que você encontrará recomendações no próximo capítulo. Você também dar uma olhada no qual o próximo inovações em CSS, juntamente com tecnologias relacionadas, são susceptíveis de vir a partir.

**Capítulo 10**

**Ferramentas, tecnologias e**

**O futuro da animação CSS**

Como um conjunto de módulos de W3C que estão apenas começando a se mover a partir de Rascunho de trabalho a recomendação de estado, o candidato CSS transforma, transições e animações especificações não ter ainda a indústria de ferramentas robustas que os mais velhos, finalizado specs possuem. Desenvolvedores de software não pode ser responsabilizada pela relativa escassez de ferramentas: a especificação tem sido um alvo em movimento, tornando o desenvolvimento de aplicativos desafiador. Ao mesmo tempo, o W3C está em movimento de avanço sobre novas tecnologias, tornando ainda mais difícil o desenvolvimento de ferramenta, enquanto o CSS em si mesma se torna muito mais potente.

Apesar do rápido ritmo de mudança, há um certo número de soluções disponíveis para acelerar e simplificar o fluxo de trabalho e desenvolvimento em CSS3 animações, tanto para compatibilidade retroativa em browsers mais antigos e de futuro para o desenvolvimento de aplicações agora e no futuro. Antes de se chegar ao que, no entanto, você deve tomar um momento para apreciar como usar de maneira eficiente as habilidades que você aprendeu até agora.

Escrito eficaz CSS3 Animações e Transições:

Evitando os reembolsos

Existem duas acções em CSS que são computacionalmente caro para um navegador para manipular: reconstruções e os reembolsos. Uma página *reflow* é iniciado quando o layout do conteúdo de página for alterada: pensar a aparência de um fluido local conforme a mudança da janela do navegador narrows. Um *repinte* ocorre quando um elemento muda a sua visibilidade, mas também em formas

Que não afetam o layout dos seus vizinhos: quando um elemento de visibilidade, opacidade ou cor de fundo muda, por exemplo.

Dos dois, os reembolsos são geralmente mais prejudicial ao desempenho, como um reflow irá "ripple" através da árvore DOM do elemento afetado para baixo através de todos os nós filho e aqueles após, forçando vários elementos para alterar a sua posição. Em muitos casos, os reembolsos podem essencialmente redesenhar a página inteira. Reflow ações podem não ser significativos para o desempenho em máquinas de desktop moderno, mas pode ser um sucesso em dispositivos alimentados com menor, como smartphones. Mais relevantes para os nossos interesses, uso de qualquer pseudoelement, como :o mouse ou um script manipulando o dom, iniciará um reflow.

■ **Nota**Se você estiver a mais de uma pessoa orientada visualmente, você pode ver um muito abrandou no processo de refluxo no Firefox como ele define a  página inicial do google.co.jp pela primeira vez no YouTube em [www.youtube.com/watch?v=nJtBUHyNBxs](http://www.youtube.com/watch?v=nJtBUHyNBxs).

Navegador enquanto os reembolsos são inevitáveis, seu impacto pode ser minimizado e localizado por seguir algumas regras:

 *Tente a afectar o elemento que deseja estilo diretamente, em vez de seu pai*. Que é tentar limitar o âmbito de qualquer reflow: em vez de alterar a classe de um elemento de recipiente para influenciar as suas crianças, tente a afectar as próprias crianças diretamente. Isso não evitar todos os reembolsos de alterar a altura de um elemento pode afetar as dimensões de sua mãe, por exemplo, causando um reflow para ondulação-mas é uma boa regra geral.

 *Evite definir estilos inline*. Mais especialmente, evitar a criação de vários estilos inline. Em vez disso, externalizar as CSS em uma classe e então alterar a classe. Um processo de trabalho que enfatiza o oposto de tentar manipular um estilo inline, ou fusão de estilos entre incorporado, ligados e estilos inline, por exemplo - provavelmente causará vários reembolsos como cada um está ajustada.

 *Aplicar animações para elementos usando transformar*,*ou a elementos que estão posicionados absolutamente ou fixo*. Elementos com estas características não afecta a posição dos outros, o que significa que somente o elemento afetados serão pintadas, sem o custo de um reflow.

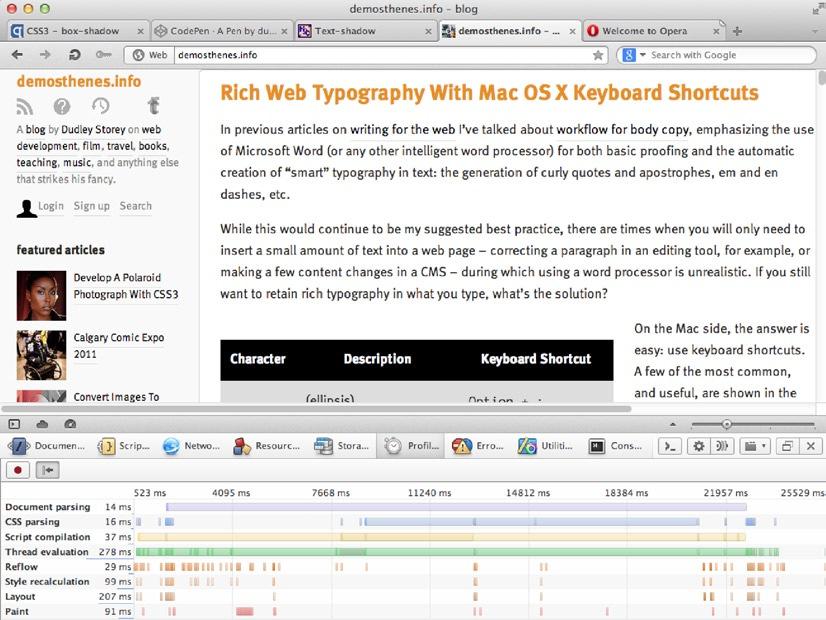
 *Criar animações que são "internas" para um elemento com uma altura fixa e largura e/ou que tenha escondido aplicada*. estouro: Animações dentro de tais elementos não irá afetar outros, evitando oportunidades para os reembolsos.

 *Especifique definir as dimensões da imagem*. Se o navegador sabe a largura e a altura da imagem elementos, ele não terá para enviar conteúdo ao redor como a imagem é colocada na página. Obviamente esta é complicada por tendências actuais em matéria de design ágil.

 *Evitar longos seletores descendente*. Tempo descendente com muitas tags seletores tendem a ser computacionalmente caro.

 *Evitar o uso do seletor universal*. O curinga (\*) O seletor é o mais pesado de todos computacionalmente seletor.

 *Tente especificar a propriedade que você deseja transição, em vez de usar "todas"*. Embora não directamente relacionados com os reembolsos, usando a opção "todos" em uma transição é um desperdício, como o navegador é então forçado a via *cada* possível mudança para o elemento. A especificação de um único imóvel que deseja via e alterar, tais como opacidade, é muito mais eficiente.

 *Use um perfil de estilo*. (Consulte a Figura 10-1.)

**Figura 10-1.***Uma ferramenta de perfil, tais como a Escala de ferramenta de desenvolvedor no Chrome ou o estilo Profiler no Opera Dragonfly, pode ajudar a destacar as áreas em que o seu CSS é particularmente ineficientes ou lenta*

Prefixo de ferramentas automático: criando

A fim de obter suporte em versões mais antigas de navegadores, CSS3 transforma, transições e animações devem ser escritos como declarações separadas com o navegador correto prefixos de fornecedor, conforme discutido no Capítulo 1. Enquanto a manutenção de cinco diferentes linhas de código para uma única transição é certamente possível, mantendo a coisa mais complexa pode ser um pesadelo. Enquanto ferramentas como Prefixr (<http://prefixr.com>) pode adicionar prefixos para o código existente, eles não são opções realistas para o código que está aberto para alterar: quaisquer alterações ao original CSS significa que você terá de passar novamente pelo mesmo processo para adicionar prefixos. Existem várias soluções possíveis para este problema, um do lado do servidor e o outro do lado do cliente, cada uma com suas vantagens e desvantagens.

-Livre de prefixo

Desenvolvedor Lea Verou grega (<http://lea.verou.me>) tem escrito um popular, leve e eficaz script que caiu em uma página, irá personalizar qualquer prefixo código CSS conforme necessário para o navegador que está exibindo a página (<http://leaverou.github.com/prefixfree/>).

Esta é a minha solução preferida para trabalho leve (tais como a disponibilização de uma CSS3 animação no cabeçalho de um post de blog); ela permite-me a escrever uma única linha de código que siga a especificação final esperado, minimizando o tamanho do arquivo e permite que o script pega compatibilidade retroativa. Há algumas questões a ser consciente de, no entanto:

 Existe um forte argumento de que este tipo de declaração CSS não é o papel de transformação de um script do lado do cliente, mas um do lado do servidor, como discutido abaixo.

 Os usuários que navegar em um site que usa o prefixo -livres script mas que tenham JavaScript desligado-quer directamente quer através de uma extensão do navegador como NoScript ([http://noscript.net)](http://noscript.net)—and)-*e* está usando um navegador que continua dependente do fornecedor prefixos não verá qualquer transforma, transições ou animações. No entanto, esta Audiência especial é pequeno e especialmente no que diz respeito ao número de navegadores movendo para apoiar os prefixados CSS3-diminuir rapidamente em tamanho. Além disso, se você seguiu os princípios de melhoramento progressivo e degradação harmoniosa no Capítulo 2, a falta de CSS3 não deverá afectar a capacidade do usuário para desfrutar ou usar o seu site.

 A entrega da norma o prefixo CSS3 do servidor pode ser ligeiramente mais rápido em alguns casos, devido ao facto de - livre de prefixo deve processar o CSS do lado do cliente antes que ele possa ser usado. Na prática, isso geralmente é contrabalançada pela do lado do servidor de transformação requerida em outras soluções ou o maior tamanho do arquivo associado com o prefixo do código CSS.

 Unprefixed propriedades que aparecem em estilos inline não ser transformado pelo prefixo -livres para o Firefox 3.6 ou inferior (um caso muito incomum, pelo menos para a maioria dos sites e seus visitantes).

SASS, menos, bússola e Codekit

SASS (<http://sass-lang.com>) e menos (<http://lesscss.org>) são talvez o melhor descrito como "meta" para CSS, quadros permitindo que recursos tais como as variáveis, funções, loops, validação automática, otimização e minification de código, regras aninhados e (mais relevantes para os nossos interesses aqui) automático a prefixação da CSS3 através de "mixins". Ambos estruturas promovem a si próprios como *extensões de CSS*, embora esse não deve

Ser considerada um endosso do W3C. (dito isto, muitas das inovações quadros CSS promoveram, tais como variáveis, estão sendo tomadas em novos módulos de CSS).

Menos trabalha através de JavaScript, "tradução" um menos infuso stylesheet (estilos.menos) em navegador-pronto CSS em tempo de execução: como resultado, ele compartilha muitas das vantagens e desvantagens do (-livre de prefixo. SASS leva a abordagem de folhas precompiling escritos em SASS (estilos.scss) em um arquivo .css completa que pode ser utilizado por todos os browser.

Bússola (<http://compass-style.org/>) bundles muitos dos melhores adições para SASS juntos. Gerenciamento de diferentes ferramentas e extensões SASS é feita um pouco difícil pelo fato de que eles têm de ser controlados através da linha de comando como Ruby gems. CodeKit (<http://incident57.com/codekit>) é um quadro manager para Mac OS X que puxa-los todos juntos (incluindo menos, stylus, e Bússola) em uma interface gráfica de usuário e alguns outros recursos de Nice como a atualização automática da página em um navegador aberto quando o CSS mudou.

Por último, incluindo um "mixin" ajuda atalho ainda mais seu código; uma coleção da Biblioteca de scripts como Bourbon (<http://thoughtbot.com/bourbon/>) ou bússola irá permitir que você digite uma transição simples em SASS desta forma. (O código na lista 10-1 é mostrado na sintaxe da Bússola).

**Listando 10-1.**Usando um SASS Mixin para gerar código Vendor-Prefixed

#elemento {

@Incluir transição de propriedade(largura); @Incluir transição duração(2s);

@Incluir transição de função de distribuição(facilidade em); }

Elemento #:foco { largura: 180%

}

Capítulo 10 ■ FERRAMENTAS, TECNOLOGIAS E O FUTURO DA ANIMAÇÃO CSS

O código na listagem 10-4 será automaticamente alargada a fim de abranger todos os prefixos de fornecedor do navegador quando o código é compilado.

Prefixo de ferramentas automáticas: Inclusões no Servidor

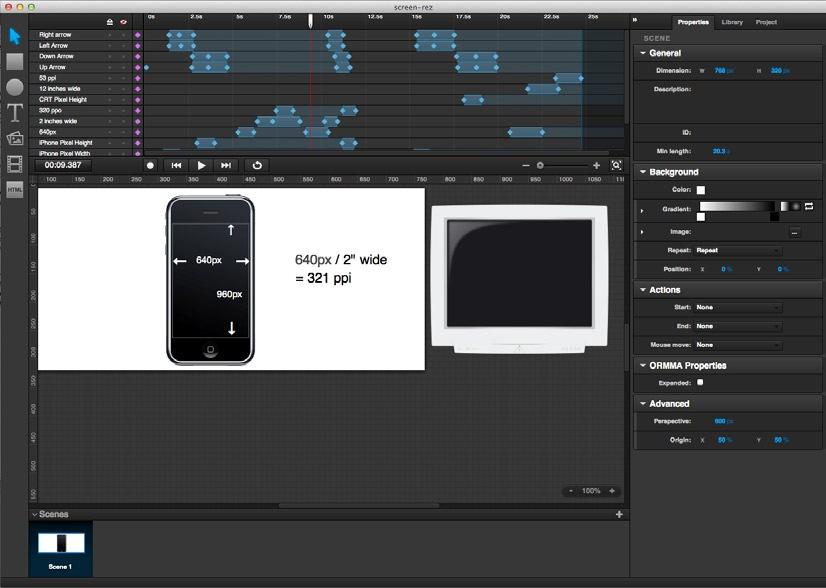
CSS Prefixer (<http://cssprefixer.appspot.com>) tem uma abordagem do lado do servidor: operando como um script Python, converte nonprefixed CSS em declarações de fornecedor quando um arquivo .css é servido. Enquanto ele é mais confiável do que uma solução do lado do cliente como -prefixo livres ou menos, não tente qualquer tipo de detecção de cliente: o CSS produzido é o prefixo para cada possível navegador, expandindo significativamente codebase . e o tamanho do arquivo.

Com base em GUI CSS3 Ferramentas de animação

Os editores de texto começar a se tornar contestadas como CSS3 animações crescer mais ambicioso e complexo: enquanto eles são perfeitamente adequados para pequenos projectos, ferramentas visuais têm vantagens significativas quando se trata de animar vários elementos separadamente. O campo está crescendo rapidamente e a seguir é apenas uma seleção.

Sencha Animator

Uma das primeiras ferramentas populares sobre o bloco, Sencha Animator ([www.sencha.com/products/animator](http://www.sencha.com/products/animator)) usa a conhecida interface cronograma juntamente com um conjunto completo de transformar e keyframing ferramentas para criar animações CSS (ver figura 10-2).

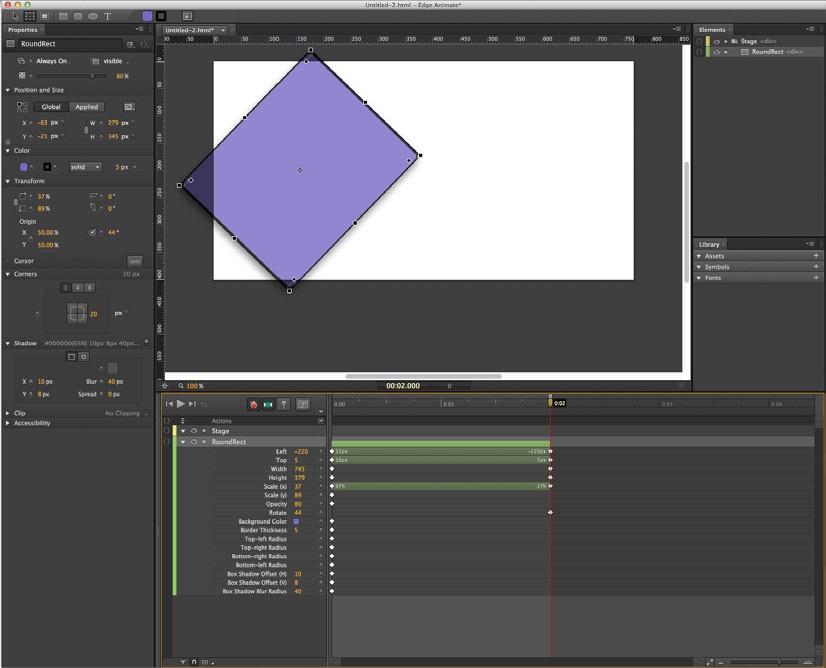


Capítulo 10 ■ FERRAMENTAS, TECNOLOGIAS E O FUTURO DA ANIMAÇÃO CSS

A saída do aplicativo é um pouco ofuscada, contudo, tornando o resultado difícil trabalhar com qualquer outra aplicação; seu padrão de meios de apresentar o trabalho (através de JavaScript e usando somente o código de prefixo de fornecedor para o Firefox e Webkit) significa que o resultado deve ser fortemente editadas para integral de compatibilidade do navegador.

Adobe Edge animar

A mais recente adição à gama de ferramentas de animação CSS (e apenas fora da beta), Adobe Edge animar (<http://html.adobe.com/edge/animate>) é uma aplicação muito promissor que se baseia no padrão Adobe UI (e o padrão cinzento escuro tema do Adobe Creative Suite 6, como mostrado na Figura 10-3) mas que adiciona uma série de melhorias significativas. Ele também utiliza o JavaScript como um quadro de apoio CSS3 animações, mas o faz de uma forma que suporta todos os browsers modernos e versões mais antigas com o fornecedor prefixos.



**Figura 10-3.***Captura de tela do Adobe Edge animar*

Animatable

Depois de fazer um splash criando um muito popular CSS3 de animação para créditos de abertura de televisão AMC *loucos homens*(consulte <http://stuffandnonsense.co.uk/content/demo/madmanimation/>e figura 10-4)*,*desenvolvimento de Animatable (<http://animatable.com>) parece ter parado, ou pelo menos ido tranquila; é ainda vale a pena manter um olho sobre o produto para a quebra de mudanças, no entanto.

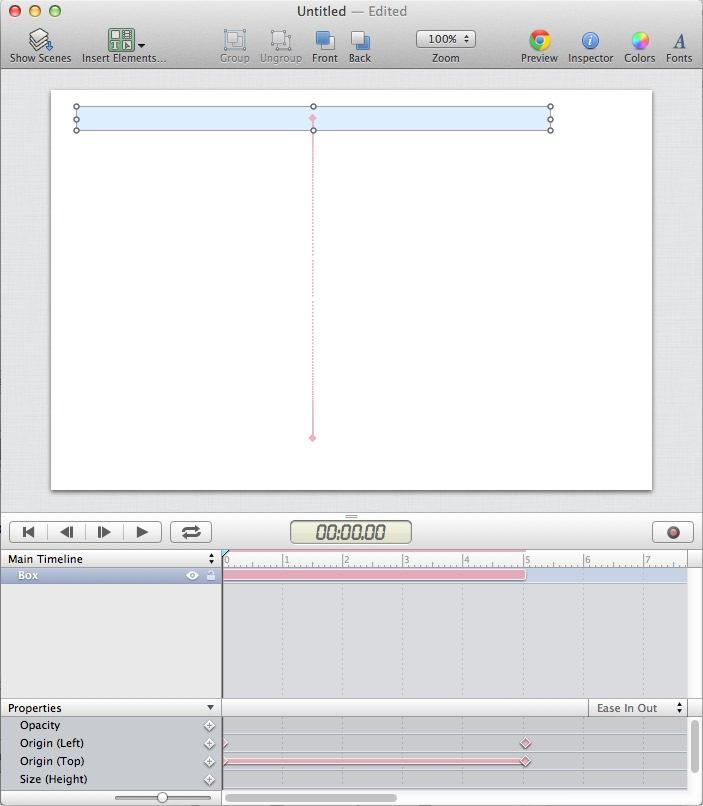
160

apítulo 10 ■ FERRAMENTAS, TECNOLOGIAS E O FUTURO DA ANIMAÇÃO CSS



**Figura 10-4.***Ainda a partir da seqüência de crédito Homens loucos CSS animação criada pela Animatable*

Tumulto Hype

Outra animação Web GUI, de todas as ferramentas descritas aqui tumulto Hype é o mais fortemente investidos em JavaScript: a aplicação descreve a si próprio como um "HTML5 ferramenta de animação" em vez de um programa de desenvolvimento CSS3. Esta é uma pena, pois sua interface é provavelmente a mais intuitiva para usar, como mostrado na Figura 10-

**Figura 10-5.***Captura de tela do tumulto Hype animação UI*

Tendências futuras: CSS filtros personalizados

Uma tecnologia Adobe anteriormente conhecido como Sombreadores de Web, o W3C adoptou esta nova tecnologia como uma proposta denominada CSS filtros personalizados.

Os filtros padrão explorado no Capítulo 9 pode ser melhor descrita como conversores de imagem simples: matiz-gire, Blur e sépia mudança de forma consistente a todos os pixels da imagem afetado, mas nada mais. Os filtros padrão CSS não pode afectar os pixeis individuais: você não pode usar um filtro na metade de uma imagem, nem pode usar um filtro para distorcer a imagem. Filtros personalizados explicitamente divide este processo em duas funções distintas: *Sombreadores de fragmento programáveis* podem

Ser utilizado para ajustar a cor de pixels, animar os toalhetes e criar transições personalizados; *sombreadores vertex* tratar a área de cada elemento de dom como uma malha visual*,* permitindo que o designer para ondulação, dobrar, torcer e distorcer a superfície de imagens bem como todos os outros elementos.

Uma chamada para um filtro personalizado, como proposto na especificação atual, lista de aparência 10-2

**Listando 10-2.**Uma transição utilizando um CSS Filtro Personalizado

#sombreado-elemento {

Filtro: custom.Url('wobble.vs") 40 40,

Amplitude 60, montante 0,0);

)

Filtro de transição: facilidade em 2s;

}

O número par 40 40 trabalha para definir a densidade da malha virtual que o elemento está dividido em: mais divisões irá criar um mais suave e fluido, e detalhado efeito. amplitude é a força do efeito e a quantidade é o grau em que o elemento é afectado.

O wobble.vs (Vertex Shader) ficheiro referido na primeira linha está escrito na linguagem de sombreamento OpenGL ES, utilizando a mesma sintaxe usada em WebGL para criar browser nativo 3D em páginas da web, como mostrado na listagem 10-3.

**Listando 10-3.**Um OpenGL ES Shader

Precision mediump flutuação; atributo vec3 uma posição\_; atributo vec2 um\_texCoord; tapete uniforme4 u\_projectionMatrix; amplitude de flutuação uniforme;

Montante de flutuação uniforme; variando o vec2 v\_texCoord; const float gire = 20,0; const float PI = 3.1415926; mat4 rotateX(float um) {. . .} mat4 rotateY(float um) {. . .} mat4 rotateZ(float um) {. . .} void main() {

V\_texCoord = um\_texCoord.xy;

O vec4 pos = vec4(a\_posição, 1.0);

Flutuação r = 1,0 - Abs((quantidade - 0,5) / 0,5); flutuar = r \* gire \* PI / 180,0;

Mat4 rotX = rotateX(a);

Mat4 rotY = rotateY(a / 4.0); mat4 rotZ = rotateZ(a / 8,0);

Flutuação dx = 0,01 \* cos(3.0 \* PI \* (pos.x + montante)) \* r; bóia dy = 0,01 \* cos(3.0 \* PI \* (pos.y + montante)) \* r;

Flutuação dz = 0,1 \* cos(3.0 \* PI \* (pos.x + pos.y + montante)) \* r; pos.x += dx;

Pos.y += dy; pos.z += dz;

Gl\_Position = u\_projectionMatrix rotZ \* \* \* rotX rotY \* pos;

}

Como você pode ver, este é *muito* diferente do CSS você está familiarizado com: é um idioma totalmente novo. No entanto aplicar todas esta como uma transição para o #sombreado-elemento (mostrado na listagem 10-4) muito retorna para os princípios você exploradas no Capítulo 2.

**Listando 10-4.**Uma transição utilizando um CSS Filtro Personalizado

#elemento sombreado:foco {

Filtro: custom.Url('wobble.vs") 40 40,

Amplitude 60, montante 1,0);

}

Todos os controles do filtro permanecem as mesmas, com excepção do montante; a transição no estado padrão original irá garantir que o elemento muda suavemente para o novo estado o mouse.

Filtros personalizados promessa um grau extremo de controle visual sobre o conteúdo em HTML que é inatingível com padrão CSS e que tem o potencial para revolucionar a animação e a interação na web. No entanto, esse processo irá demorar algum tempo: a especificação é muito novo e, quando este artigo foi escrito, é suportado apenas em compilações muito recente de cromo. Finalizando a especificação, ganhando o suporte de browser cruzado e lidar com questões importantes como a segurança é susceptível de levar vários anos.

Tendências futuras: Mistura e composição

Adobe é também avança CSS na área de mistura e composição, tendo os controles que você pode estar familiarizado com no Adobe PhotoShop e Illustrator - como multiplicar, escurecer e clarear, e avançados de recorte e traduzi-los para CSS.

Enquanto este trabalho é apenas na fase de proposta, promete um muito mais Visualmente vibrante web se implementado, especialmente se as propriedades podem ser animados. É concebível que a longo prazo, bitmap e vetor ilustração ferramentas irá tornar cada vez mais *os criadores de conteúdo* nas primeiras fases do processo de produção*,* enquanto a maioria dos trabalhos de edição será feito viver no navegador usando CSS.

Tendências futuras: Conciliar CSS3 e SVG

Para duas tecnologias da web que podem trabalhar tão bem juntos e ter influenciado uns aos outros de forma poderosamente (especialmente na área de transforma e filtros), CSS e SVG continuam a ter um notável número de conflitos.

Enquanto o CSS pode ser usado para animar um número de atributos de apresentação SVG bem como SVG próprias imagens (como você viu no Capítulo 9), e enquanto SVG pode animar a própria usando a <animar> tag, parte de um padrão conhecido como SMIL (Synchronized Multimedia Integration Language) de fundir as duas atualmente é extremamente difícil. CSS Animação Keyframe regras substituem as transições CSS e animações SMIL (SVG), mas são contradição quando se trata de transições CSS e animações SMIL. A questão da compatibilidade é mais complicada pelo Internet Explorer 10's falta de suporte para SMIL.

Estas questões são abordadas pelo W3C CSS-Efeitos SVG Task Force ([www.w3.org/Graphics/fx](http://www.w3.org/Graphics/fx)), mas a evolução e as mudanças que pode sair desse processo são desconhecidos neste momento.

Resumo

Tecnologias de animação Web vêm um caminho muito longo desde os dias quando a memória Flash e JavaScript foram as únicas formas de obter movimentos em uma página da web e eles têm certeza de ir ainda mais longe, com as ferramentas e sintaxes que analisámos aqui-e com aquelas que ainda estão para vir.

Este livro tem oferecido uma introdução e um mergulho profundo na baseada em padrões modernos de animação CSS3. Ao longo do caminho, você já experimentaste CSS transforma, transições, animações keyframe e manipulação de 3D, e juntamente com outras tecnologias e técnicas tais como SVG, JavaScript e

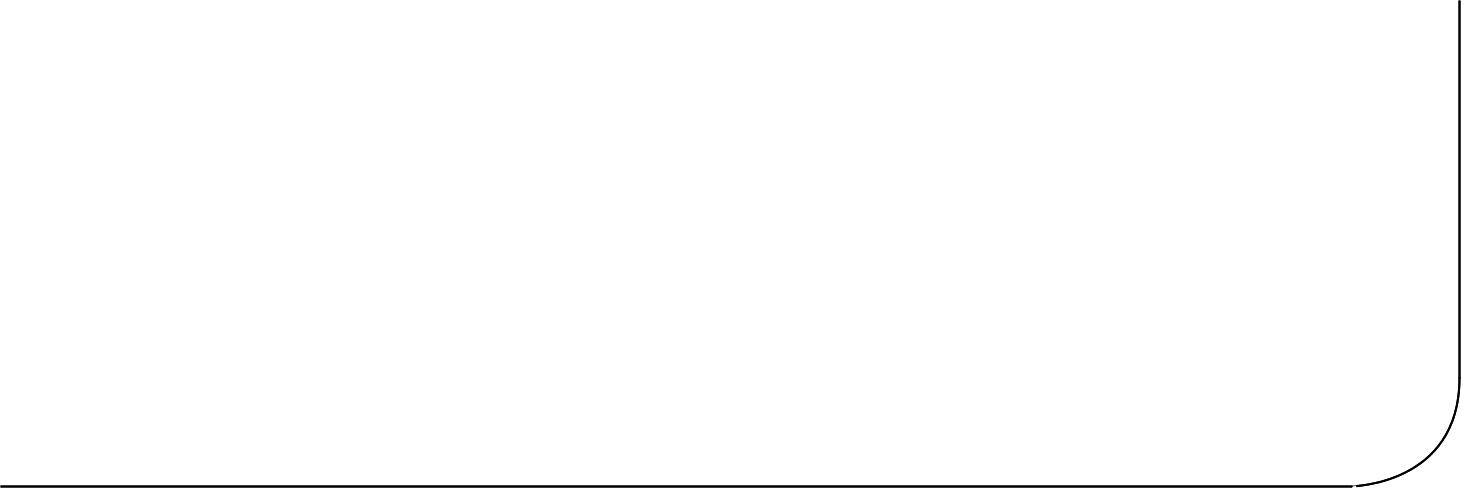
Design ágil. Ferramentas para o desenvolvimento solidário com esta mélange de idiomas estão próximos ao longo, mas o mais confiável - e a única que permitirá a você a liberdade para permanecer na vanguarda, você deve escolher para a prosseguir é o humilde editor de texto.

O período prolongado de escrever longo cross-compatibilidade do navegador declarações CSS com prefixos de fornecedor está a chegar ao fim. As versões mais recentes de navegadores estão caindo-los totalmente para as propriedades que você já procurou em neste livro.

O novo ambiente que esta rápida adoção de tecnologias está formando-a "Próxima Web"-promete ser mais abertos e mais ricos em possibilidade, e mais criativa do que qualquer coisa que vem tornar-se. Estou animado para ver quais novas perspectivas pode descobrir, como espero que são demasiado.

Estou ansioso para ver o seu trabalho e obter o seu feedback e estar inspirado no turno por sua explorações e inovações em CSS3 animação.

165

**Índice**

**um**

Animação no carregamento da página de fundo, [54](#page58)

Imagem de fundo falso, [55](#page59)

**B**

Transições de imagem de fundo, [52](#page56) linhas vectoriais assim curvas

Animação, em etapas, [27](#page32)

Metros Cúbicos comum linhas vectoriais assim as funções de temporização, [24](#page29) facilidade na animação de modelar, [26](#page31)

Facilitando a função de sincronização, [24](#page29) animação linear gráfico, [23](#page28) dispositivos móveis apoio, [28](#page33) modelar animação push-pull, [26](#page31) funções de distribuição, [23](#page28)

**c**

Folhas de estilo em cascata (CSS), [1](#page6) CSS3

Animação limitações, [5](#page10) suporte do navegador, [4](#page9) HTML5, [7](#page12)

*Vs*. JavaScript e Flash, [7](#page12)CSSWG, [1](#page6)

Descrição, [1](#page6) princípios de design, [6](#page11) desenvolvimento, [1](#page6) 3D transforma, [8](#page13) fornecedor preixes, [2–4](#page7)

Excepções, [3](#page8) problemas, [4](#page9)

Gradientes lineares, [2](#page7)

Navegador único preixes, [2](#page7) W3C, [1](#page6)

Imagem claquete o código CSS, [48](#page52) sair do evento, [51](#page55)

Sequências de transição, [50](#page54) transferida para transformar, [48](#page52)

CSS3 animações e transições, [155](#page156) ferramentas preixing automática

Do lado do cliente, [157](#page158)

Quadros, [158](#page159) preix meta-livre, o [157](#page158) lado do servidor [159](#page160)

Mistura e composição, [164](#page165) suporte de browser, [4](#page9)

Custom eeds, [162](#page163) 3D transforma

Galeria de 3D, [145](#page146) Rejeição de fundo de face de visibilidade, [144](#page145) cartão caption, [142](#page143) imagem gallary lábio, [137](#page138) Início - como modelar, [136](#page137) JavaScript, [149](#page150) perspectiva, [135](#page136) rotação, [139](#page140)

Estilo, [144](#page145) traduzir, [140](#page141) elementos de UI, [151](#page152)

Modelar em dispositivos móveis, [125](#page127) eeds, 113-115

Blur iltragem escolhido transição, [115](#page118) Cinza iltragem escolhido transição, [113](#page116) polaroid foto defeituoso, [114](#page117) sépia iltragem escolhido transição, [113](#page116)

Sombreadores de fragmento, [162](#page163) GUI, 159-161

Adobe edge animação, [160](#page161) sencha animação, [159](#page160) tumulto Hype [161](#page162)

HTML5, [7](#page12)

Animações e Transições CSS3 (*cont.*)

JavaScript [118](#page120) e Flash, [7](#page12) JQuery, [127](#page129)

Elementos dispersos, [131](#page133) desencadeando aleatória, [127](#page129)

Limitações, [5](#page10) manipulando SVG, [106](#page109)

Imagemap, [108](#page111) snowlake, [110](#page113) otimização de, [124](#page126)

Página relow, [155](#page156) colocação SVG, 104-106

Adicionando objeto/iframe, [106](#page109) inconvenientes, [105](#page108) elementos, [105](#page108)

Imagem em linha, [104](#page107) SVG em linha, [105](#page108)

Relow, regras para [156](#page157) pintar, [155](#page156)

Ripple, [155](#page156)

SVG

Dados, [103](#page106) Ile, [125,](#page127)[164](#page165) ferramentas [112](#page115)

Transforma (*ver* transformação, transições CSS) (*ver* transições, CSS) Tamanho de visualização , [123](#page125)

W3C, [155](#page156)

CSS3 animações keyframe, [75](#page79) alterando a imagem

Adicionar legendas , [91](#page94) criando fallback, [93](#page96) Mixagem [90](#page93) Fade-In, [88](#page91) Fade-In-out, movimento, [89](#page92) On-Click método, [92](#page95)

Direção de animação valores de propriedade, [78](#page82) animação de modo mal valores de propriedade, [78](#page82) mistura e encadeamento

Animação de atraso, [80](#page84) animação duração, [80](#page84) mesclando transições, [79](#page83)

CSS3 slideshow imagem de fundo, [87](#page90) código HTML, [85](#page88)

Lightbox gallery, [97](#page100) adição de legendas, [99](#page102) mouseover caption, [99](#page102)

Logotipo no carregamento da página, [101](#page104) marquee para texto, [93](#page96)

Noticiários/notiication, [93](#page96) suporte a navegadores mais antigos, [77](#page81) pausar [83](#page87)

Apresentação de slides em pausa, [88](#page91)

Controlo de reprodução, [78](#page82) repetir seqüências, [80](#page84) sintaxe, [76](#page80)

O processo de interpolação, [75](#page79) CSS3 transições, [57](#page61)

Clique no botão defeitos, 70-72 Caixa de marcação [71](#page75)

Seleção de marcação, para associados div, [72](#page76) Mars imagem, [71](#page75)

Abrindo e fechando o div, [72](#page76) texto pop-up, [72](#page76)

Página atual realçar, [60](#page64)

Erros de validação personalizados, para formulários HTML5, 63-65

Forma acessível, [64](#page68)

Mensagens de erro de forma animada, [64](#page68) imediatismo, [63](#page67)

Redundância, [63](#page67) segurança [63](#page67)

A modelagem e a exibição de erros de validação de [65](#page69) elementos de formulário animação, [73](#page77)

Navegação suspensa horizontal, [69](#page73) interfaces de navegação horizontal, [58](#page62) guia horizontal de navegação, [62](#page66)

O sistema de navegação guia de animação, [62](#page66) estilo CSS para links de navegação, [62](#page66)

Estrutura de navegação do site HTML5, [57](#page61) imagem mixagem defeituoso

Fundo CSS, [32](#page36) elementos HTML, [32](#page36) imagem como fundo, [33](#page37)

A navegação no site moderno markup, [57](#page61) barra de navegação aprimorada, [59](#page63) migrou de opacidade, [31](#page35)

Pressione o botão da interface de transição, [66](#page70) botão UI revelam transição, [67](#page71)

Grupo de Trabalho (CSSWG CSS), [1](#page6)

**D**

Galeria de 3D, [145](#page146) circular, [146](#page147)

Seletores de nível 4, [149](#page150) legenda de pop-up, [148](#page149) sombreamento defeituoso, [147](#page148)

DOM (Document Object Model), [11](#page16)

**e**

Facilitando a função de sincronização, [24](#page29)

Efeitos de entrada e saída, [46](#page50)

**F, G**

Sombreadores de fragmento, [162](#page163)

**h**

Navegação suspensa horizontal, [69](#page73) interfaces de navegação horizontal, [58](#page62)

**I, J, K, L, M, N, S**

Pilha do cartão de imagem, [40](#page44) Css transforma, [42](#page46) deset origem, [43](#page47) transformar a origem, [42](#page46)

Ventilador de imagem revelam, [40](#page44) galeria de ventilador da placa, [44](#page48) primeiro plano [45](#page49) sugerido do ventilador da placa, [44](#page48)

Código HTML e CSS, [46](#page50) atributo src, [45](#page49)

Galeria de imagens

Adicionar legendas, [36](#page40) marcação de HTML, [34](#page38) melhorar gallery, [35](#page39) inicial CSS, [35](#page39) início do evento

Ativo, [36](#page40) alvo, [37](#page41)

**P, Q**

Página relow, [155](#page156) imagem pop-up, [37](#page41)

Imagem animada, [38](#page42)

CSS

Ocultar, [40](#page44) transição, [40](#page44)

Rem font size, [39](#page43)

**r, S**

Resposta Web Design (propulsão), [117](#page119)

Sem transições, [118,](#page120)[120](#page122)

■ ÍNDICE

Imagem de fundo [120](#page122)

Imagem no mínimo e o máximo, [120](#page122) Redimensionar imagem e vídeo [118](#page120)

Com as transições, [121](#page123) ripple, [155](#page156)

**t**

Transformação, CSS, [9](#page14) descrição, [9](#page14) DOM, [11](#page16)

Fusão, [17](#page22)

Transformação de rotação, 10-11 descrição, [10](#page15)

Loating uma imagem, [11](#page16) possível os sistemas da unidade, [11](#page16)

Escala transformar, [14](#page19) inclinação, [16](#page21)

Traduzir modiier, [16](#page21)

Webkit CSS3 transformar aliasing questão, [13](#page18) transições, CSS, [18](#page23)

Atrasar e combinando defeitos, 21-23 descrição, [21](#page26)

Facilitando a funções, [22](#page27) propriedades, [21](#page26)

Funções de distribuição de transição, [23](#page28) descrição [18](#page23)

Do lado da distribuição, [19](#page24) processo de interpolação, [75](#page79)

**U, V**

Pressione o botão da interface de transição, [66](#page70)

Botão de UI revelam transição, [67](#page71)

**W, X, Y, Z**

World Wide Web Consortium (W3C), [1](#page6)

Pro CSS3 Animação

Dudley Andare

Seu trabalho está sujeito a direitos de autor. Todos os direitos são reservados pela editora, se a totalidade ou parte do material é em causa, speciically os direitos de tradução, reimpressão, reutilização de ilustrações, recitação, radiodifusão, reprodução em microilms ou em qualquer outra forma física e de transmissão ou de armazenamento e recuperação de informações electrónicas, adaptação, um software de computador ou por semelhante ou metodologia desiguais agora conhecido ou doravante desenvolvido. Isentos desta reserva legal são breves trechos em conexão com clientes ou análise científica ou material fornecido speciically para efeitos de introdução e executado em um sistema de computador para uso exclusivo pelo adquirente do trabalho. A duplicação desta publicação ou de suas partes só é permitida nos termos das disposições da Lei de Direitos Autorais do local do Publisher, na sua versão actual, e permissão

Para utilizar sempre deve ser obtido a partir de Springer. Permissões de uso podem ser obtidos através de RightsLink no Centro de apuramento de direitos autorais. As violações são puníveis ao abrigo das respectivas leis de direitos autorais.

ISBN-13 (PBK): 978-1-4302-4722-7

ISBN-13 (eletrônico): 978-1-4302-4723-4

Os nomes de marcas registradas e logotipos e imagens podem aparecer neste livro. Em vez de usar um símbolo de marca registrada com todas as ocorrências de um nome de marca registrada, logotipo ou imagem que usam os nomes, logotipos e imagens somente em um editorial de moda e a beneit do proprietário da marca, sem intenção de violação da marca registrada.

Ele utilizar nesta publicação dos nomes comerciais, marcas comerciais e marcas de serviço e termos semelhantes, mesmo se eles não estiverem identiied como tal, não é para ser tomado como uma expressão de opinião quanto à possibilidade de estarem ou não sujeitos a direitos de propriedade.

Enquanto os conselhos e as informações neste livro são acredita ser verdadeiras e precisas na data de publicação, nem os autores nem os editores nem o publisher pode aceitar qualquer responsabilidade por quaisquer erros ou omissões que possam ser feitas. Ele o Publisher não faz qualquer garantia expressa ou implícita com respeito ao material contido neste documento.

Presidente e Publisher: Paul Manning Editor de chumbo: Louise Corrigan revisor técnico: Sylvain Galineau

Conselho Editorial: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jef Olson, Jefrey Pimenta, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Espetando, Matt Wade, Tom Welsh

Editor de coordenação: Kevin Shea Copy Editor: Elizabeth Berry "concertador": SPi indexador Global: SPi Global

Artista: SPi Global

Cobrir Designer: Anna Ishchenko

Distribuído para o livro do comércio mundial pela Springer Ciência+Business Media Nova Iorque, 233 Spring Street, 6º piso, Nova York, NY 10013. Telefone para 1-800-SPRINGER, fax (201) 348-4505, de e-mail [orders-ny@springer-sbm.](file:///C:\Users\Caixa%2006\AppData\Local\Temp\orders-ny@springer-sbm.com) [com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\orders-ny@springer-sbm.com), ou visite [www.springeronline.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.springeronline.com).

Para obter informações sobre as traduções, envie um e-mail [rights@apress.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\rights@apress.com), ou visite [www.apress.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.apress.com).

Aprima e amigos de ED livros podem ser adquiridos a granel para fins académicos, corporativa ou uso promocional. versões eBook e licenças também estão disponíveis para a maioria dos títulos. Para obter mais informações, referência a nossa especial a venda a granel de licenciamento de eBook na página da web em [www.apress.com/bulk-sales](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.apress.com\bulk-sales).

Qualquer código fonte ou outros materiais complementares referenciado pelo autor neste texto está disponível aos leitores em [www.apress.com](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.apress.com). Para obter informações detalhadas sobre como localizar o código-fonte do livro, vá para [www.apress.com/source-code](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.apress.com\source-code).

**Conteúdo**

Sobre o Autor [xiii](#page178)

Introdução [xv](#page4)

■ Capítulo 1: Fundamentos de CSS3 [1](#page6)

Desenvolvimento de CSS [1](#page6)

Prefixos de fornecedor de CSS [2](#page7)

Fornecedor do prefixo de questões [4](#page9)

CSS3 Suporte do navegador [4](#page9)

Limitações do CSS3 Animação [5](#page10)

Princípios de projeto: melhoramento progressivo e degradação harmoniosa [6](#page11)

Por que razão CSS3 em vez de JavaScript ou Flash? [6](#page11)

Outras tecnologias [7](#page12)

Resumo [8](#page13)

■ Capítulo 2: CSS3 transforma e Transições [9](#page14)

O CSS transforma [9](#page14)

Gire [10](#page15)

O Webkit CSS3 Transformar Aliasing questão [13](#page18)

Escala [14](#page19)

Traduzir [16](#page21)

Inclinação [16](#page21)

Combinando Transformações [17](#page22)

Transições CSS [18](#page23)

Vii

■ CONTEÚDO

Atrasar e combinando os efeitos de transição [21](#page26)

Apresentando as funções de facilitar [22](#page27)

Funções de distribuição de transição e linhas vectoriais assim curvas [23](#page28)

Resumo [29](#page34)

■ Capítulo 3: transições CSS3 para imagens [31](#page35)

Imagem de simples efeito de mixagem [31](#page35)

Opção 1: Primeira imagem como fundo de uma CSS [32](#page36)

Opção 2: ambas as imagens como elementos HTML [32](#page36)

Opção 3: ambas as imagens como plano de fundo [33](#page37)

Uma simples Galeria de Imagens melhoradas com CSS3 [34](#page38)

A marcação de HTML [34](#page38)

O CSS Inicial [35](#page39)

Melhorar a galeria [35](#page39)

Adição de legendas [36](#page40)

Alterar o início do evento [36](#page40)

Pop-up simples legendas de imagem [37](#page41)

Pilha do cartão de imagem e ventilador revelar [40](#page44)

Claquete legendas de imagem com diferentes efeitos de entrada e de saída [46](#page50)

Criação de sequências específicas de transição [50](#page54)

Alterando a saída caso [51](#page55)

Transições de imagem de fundo e a animação no Carregamento da Página [52](#page56)

Animar a imagens de fundo na Carga [54](#page58)

Criar e animar imagens de fundo falso [55](#page59)

Resumo [56](#page60)

■ Capítulo 4: CSS3 transições para elementos de UI [57](#page61)

A navegação no site moderno Markup [57](#page61)

Noções Básicas de Barra de Navegação Horizontal [58](#page62)

Uma barra de navegação simples com CSS3 Aprimorado [59](#page63)

Destacando a página atual na navegação [60](#page64)

Guia horizontal de navegação com transições CSS3 [62](#page66)

Viii

■ CONTEÚDO

Animação de validação personalizados Erros para formulários HTML5 [63](#page67)

Pressione o botão da interface de Transição [66](#page70)

Botão de interface de revelar a transição [67](#page71)

Acessível suspensa horizontal de navegação com transições CSS3 [69](#page73)

Início CSS3 Efeitos de transição após um clique de botão [70](#page74)

Animação de elementos de formulário com CSS3 [73](#page77)

Resumo [74](#page78)

■ Capítulo 5: CSS3 Keyframe animações [75](#page79)

Keyframes e de Interpolação [75](#page79)

CSS3 Animação Keyframe sintaxe [76](#page80)

Suporte para Animação Keyframe em browsers mais antigos [77](#page81)

Controlar a reprodução de Animação Keyframe [78](#page82)

Mistura e encadeamento animações Keyframe [79](#page83)

Repetir sequências de animação [80](#page84)

Pausando Keyframe animações [83](#page87)

Resumo [83](#page87)

■ Capítulo 6: CSS3 Animações Keyframe para conteúdo da Web [85](#page88)

Um simples CSS3 Slides [85](#page88)

O código HTML [85](#page88)

Uma variação com imagens de plano de fundo [87](#page90)

Pausar a apresentação de slides [88](#page91)

Alterando a transição entre as imagens [88](#page91)

Fade-In-Out [88](#page91)

Fade-In-Out durante o movimento [89](#page92)

Mixagem [90](#page93)

Adição de legendas [91](#page94)

Um método para Pausa On-Click [92](#page95)

Criando um fallback para versões mais antigas do Internet Explorer [93](#page96)

Uma advertência contra usando Marquee a animação para Texto [93](#page96)

Ix

■ CONTEÚDO

News Ticker/Animação de notificação [93](#page96)

Um Lightbox Galeria de imagem equivalente em CSS3 [97](#page100)

Adição de legendas [99](#page102)

Animação do logotipo no Carregamento da Página [101](#page104)

Resumo [102](#page105)

■ Capítulo 7: Integrando CSS3 animações com SVG e filtros [103](#page106)

Uma introdução à SVG [103](#page106)

Colocação SVG em uma página da Web [104](#page107)

SVG como uma imagem em linha [104](#page107)

Inline SVG [105](#page108)

SVG adicionado como um objeto ou iframe [106](#page109)

Manipulação de SVG com CSS [106](#page109)

Uma animada Imagemap SVG [108](#page111)

Floco de animação SVG [110](#page113)

Ferramentas para SVG [112](#page115)

Introdução ao CSS3 Filtros [113](#page116)

Resumo [115](#page118)

■ Capítulo 8: Integração CSS3 animação com resposta Web

Design e JavaScript [117](#page119)

Redimensionamento de elementos em resposta Web Design sem transições [118](#page120)

Redimensionamento de Imagens e Vídeos dinamicamente [118](#page120)

Resposta imagens de plano de fundo [120](#page122)

Redimensionamento de elementos em resposta Web Design com as transições [121](#page123)

Indicando o tamanho de visualização com CSS3 Media Queries e Transições [123](#page125)

Otimizando o CSS transições e animações para dispositivos móveis [124](#page126)

Integrando CSS3 Media Queries com SVG [125](#page127)

Disparo transições CSS3 com JavaScript [127](#page129)

Personalizando as transições CSS3 com JavaScript [131](#page133)

Resumo [133](#page135)

■ CONTEÚDO

■ Capítulo 9: CSS3 3D transforma, transições e animações [135](#page136)

Perspectivas financeiras [135](#page136)

Rotação [139](#page140)

Traduzir [140](#page141)

Legenda de cartão Flip [142](#page143)

Rejeição de fundo de face Visibilidade [144](#page145)

Transformar o estilo [144](#page145)

Uma Circular Galeria de 3D [145](#page146)

Reforço da galeria com seletores de Nível 4 e JavaScript [149](#page150)

Adicionando um Nível de CSS Seletor-4 [150](#page151)

3D CSS transforma e transições para elementos de UI [151](#page152)

Resumo [154](#page155)

■ Capítulo 10: Ferramentas, tecnologias e o futuro da animação CSS .[155](#page156)

Escrito eficaz CSS3 Animações e Transições: evitando os reembolsos [155](#page156)

Prefixo de ferramentas automático: criando [157](#page158)

-Livre de prefixo [157](#page158)

SASS, menos, bússola e Codekit [158](#page159)

Prefixo de ferramentas automáticas: Inclusões no Servidor [159](#page160)

Com base em GUI CSS3 Ferramentas de animação [159](#page160)

Sencha Animator [159](#page160)

Adobe Edge animar [160](#page161)

Animatable [160](#page161)

Tumulto Hype [161](#page162)

Tendências futuras: CSS filtros personalizados [162](#page163)

Tendências futuras: Mistura e composição [164](#page165)

Tendências futuras: Conciliar CSS3 e SVG [164](#page165)

Resumo [164](#page165)

Índice [167](#page167)

Xi

Fotografia por Damain romba ([www.damianblunt.ca](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.damianblunt.ca))

**Sobre o Autor**



**Dudley andares**tem trabalhado em design digital a partir do momento em que seus pais lhe comprou seu computador irst, o Commodore 64 com 16 cores e 64K de memória. Nascido na Nova Zelândia, ele formou a partir da Universidade de Auckland com graus em História da Arte e da Psicologia. Continuando a trabalhar em mais computadores poderosos, ele criou visualizações digital programável de simulações, ferramentas de aprendizagem, e animações para a indústria transformadora, publicidade e arquitetos. Com a idade de 28 ele se mudou para o Canadá e

Começou a ensinar em multimédia de uma série de instituições, ganhando uma crescente especialização em tecnologias da web. Continuando a freelance, passou a última década como um instrutor no campo dos novos meios de produção e curso de Design no SAIT Polytechnic no sul de Alberta.

Há quatro anos, Dudley iniciou um blog de desenvolvimento web [www.demosthenes.info](file:///C:\Users\Caixa%2006\AppData\Local\Temp\www.demosthenes.info). Originalmente concebido como um auxílio de aprendizagem para seus alunos, o site rapidamente ganhou maior atenção. Encorajada por perguntas em suas aulas, o blog se tornou um centro de experimentação e de discussão. CSS

Um enthustiastic contribuinte para a comunidade para o desenvolvimento da web, Dudley tem escrito para a revista do esmagamento, um Guia do Estudante de Web Design e outros sites.

Dudley mantém uma presença activa no Twitter (@dudleystorey) e

Participa em web design e desenvolvimento das conferências episcopais de todo o mundo, incluindo a participação no W3C da Web Educação grupo comunitário.

Quando não os sermões, codificação, ou por escrito, Dudley goza de andar (especialmente de longa distância auto-suportados Bike Tours), leitura e ilm.