PROGRAMAÇÃO FRONT END I

Karine Birnfeld

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Evolução histórica do HTML

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Relatar a história do HTML.
- Reconhecer as diferenças entre as versões HTML.
- Identificar as mudanças do HTML5.

Introdução

HTML, ou Hyper-Text Markup Language, é o que cria a estrutura básica da World Wide Web. HTML consiste em marcadores codificados chamados *tags*, que cercam e diferenciam trechos de texto, indicando a função e a finalidade do que está marcado. Algumas *tags* especificam como as informações são exibidas no documento, enquanto outras criam *links* entre documentos e, ainda, outras, conteúdo na Internet. HTML5 é a última revisão do padrão HTML desenvolvida pelo World Wide Web Consortium.

Neste capítulo, você conhecerá a história da linguagem de marcação HTML, desde seu surgimento à evolução dos padrões HTML, e explorará as alterações no novo padrão HTML5 e as implicações que isso tem para os profissionais da tecnologia da informação.

História do HTML

Em 1980, nasceram os princípios fundamentais do HTML a partir de um primitivo modelo de hipertexto conhecido como Enquire, escrito em linguagem Pascal, no CERN (Conseil Européen pour la Recherche Nucléaire — Organização Europeia para a Pesquisa Nuclear), por meio das pesquisas de Tim Berners-Lee, que, na época, trabalhava na divisão de computação da instituição (Figura 1).

Com o auxílio de Robert Cailliau, em 1990, Tim Berners-Lee construiu o primeiro navegador/editor, chamado de WorldWideWeb, e criou o protocolo HTTP (HyperText Transference Protocol — Protocolo de Transferência de Hipertexto) para distribuir conteúdo na rede. O HTTP era alimentado com uma nova linguagem de marcação, o HTML — baseado no SGML (Standard Generalized Markup Language), uma linguagem amplamente aceita para a estruturação de documentos e da qual o HTML herdou as *tags* de título, cabeçalho e parágrafo. A grande novidade foi a marcação `a` com o elemento `href`, que permitia a ligação (*link*) entre vários documentos.

Figura 1. Esta máquina NeXT foi usada por Tim Berners-Lee, em 1990, para executar o primeiro servidor Web.

Fonte: History... ([2018], documento *on-line*).

O programador Marc Andreessen, que logo fundaria Netscape, inicia o projeto de seu próprio navegador: o Mosaic. Em dezembro de 1992, Andreessen, participando de uma lista de discussão mundial (WWW-talk) para difundir as propostas de Lee sobre o HTML, propôs a implementação de uma *tag* para imagens: a *tag* `img` (FLATSCHART, 2011).

Um documento chamado “Hypertext Markup Language” foi publicado em 1993 pela IETF (Internet Engineering Task Force). Nesse mesmo ano, o navegador Mosaic foi lançado, permitindo a exibição de imagens, listas e formulários (FLATSCHART, 2011).

Em 1994, realizou-se, em Genebra, a primeira conferência mundial sobre Web, a World Wide Web Conference, da qual surgiu a especificação HTML 2.0. Marc Andreessen e Jim Clark fundaram a Netscape Communications, apontando para o nascimento do primeiro navegador de alcance global. No final de 1994, foi criado o W3C (World Wide Web Consortium — Consórcio World Wide Web) para coordenar o desenvolvimento de padrões abertos para a Web (FLATSCHART, 2011).

Com a criação do W3C, o desenvolvimento do HTML mudou de local novamente. Uma primeira tentativa abortada de estender o HTML em 1995, conhecida como HTML 3.0, abriu caminho para uma abordagem mais pragmática, conhecida como HTML 3.2, que foi concluída em 1997. O HTML 4.01 se seguiu no final de 1999 (RAGGETT, 1997; RAGGETT; LE HORS; JACOBS, 1999).

No ano seguinte, a associação ao W3C decidiu parar de desenvolver HTML e, em vez disso, começou a trabalhar em um equivalente baseado em XML, chamado XHTML. Esse esforço começou com uma reformulação do HTML 4.01 em XML, conhecido como XHTML 1.0, que não adicionou novos recursos – exceto a nova serialização –, e foi concluído em 2000 (PEMBERTON *et al*., 2000). Após o XHTML 1.0, o foco do W3C passou a facilitar para outros grupos de trabalho estenderem o XHTML, sob a bandeira da Modularização XHTML. Paralelamente, o W3C também trabalhou em uma nova linguagem que não era compatível com as linguagens HTML e XHTML anteriores, denominada XHTML 2.0 (BIRBECK *et al*., 2010).

Na época em que a evolução do HTML foi interrompida, em 1998, partes da API para HTML desenvolvidas pelos fornecedores de navegadores foram especificadas e publicadas sob o nome DOM Nível 1 (em 1998) e DOM Nível 2 Principal e DOM Nível 2 HTML (a partir de 2000 e culminando em 2003). Esses esforços acabaram com algumas especificações do Nível 3 do DOM publicadas em 2004, mas o grupo de trabalho foi fechado antes que todos os rascunhos do Nível 3 fossem concluídos (WOOD *et al*., 1998; LE HORS *et al*., 2000; LE HORS *et al*., 2004).

Em 2003, a publicação do XForms, uma tecnologia que foi posicionada como a próxima geração de formulários da Web, despertou um interesse renovado na evolução do próprio HTML, em vez de encontrar substitutos para ele. Esse interesse nasceu da constatação de que a implantação do XML como tecnologia da Web limitava-se a tecnologias totalmente novas (como RSS e Atom posterior), em vez de substituir as tecnologias implantadas existentes (como HTML) (DUBINKO *et al*., 2000).

Uma prova de conceito para mostrar que era possível estender os formulários do HTML 4.01 para fornecer muitos dos recursos introduzidos pelo XForms 1.0, sem exigir que os navegadores implementassem mecanismos de renderização incompatíveis com as páginas da Web em HTML existentes, foi o primeiro resultado disso. O interesse foi renovado. Nessa fase inicial, enquanto o rascunho já estava disponível ao público, e eram solicitadas sugestões de todas as fontes, a especificação estava sob os direitos autorais da Opera Software (HICKSON, 2005).

A ideia de que a evolução do HTML deveria ser reaberta foi testada em um *workshop* do W3C, em 2004, onde foram apresentados alguns dos princípios subjacentes ao trabalho em HTML, bem como a proposta preliminar, cobrindo apenas recursos relacionados a formulários. A proposta foi rejeitada com o argumento de que ela conflitava com a direção escolhida anteriormente para a evolução da Web. A equipe e os membros do W3C votaram para continuar desenvolvendo substituições baseadas em XML (JACOBS; WALSH, 2004).

Pouco tempo depois, Apple, Mozilla e Opera anunciaram conjuntamente sua intenção de continuar trabalhando no esforço sob o guarda-chuva de um novo local chamado WHATWG (Web Hypertext Application Technology Working Group — Grupo de Trabalho de Tecnologia de Aplicações de Hipertexto para Web). Uma lista de discussão pública foi criada, e o rascunho foi movido para o *site* WHATWG. Os direitos autorais foram posteriormente alterados para pertencerem em conjunto aos três fornecedores e permitirem a reutilização da especificação (WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP, 2019).

O WHATWG foi baseado em vários princípios fundamentais, em particular que as tecnologias precisam ser compatíveis com versões anteriores, que especificações e implementações precisam corresponder, mesmo que isso signifique alterar a especificação em vez das implementações, e que as especificações precisam ser detalhadas o suficiente para que as implementações possam alcançar interoperabilidade completa sem engenharia reversa (WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP, 2019).

O último requisito, em particular, exigia que o escopo da especificação HTML incluísse o que havia sido especificado anteriormente em três documentos separados: HTML 4.01, XHTML 1.1 e HTML nível 2 do DOM. Também significava incluir significativamente mais detalhes do que anteriormente havia sido considerado a norma (WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP, 2019).

Em 2006, o W3C manifestou interesse em participar do desenvolvimento do HTML 5.0 e, em 2007, formou um grupo de trabalho encarregado de trabalhar com o WHATWG no desenvolvimento da especificação HTML. Apple, Mozilla e Opera permitiram ao W3C publicar a especificação sob os direitos autorais do W3C, mantendo uma versão com uma licença menos restritiva no *site* WHATWG (WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP, 2019).

Por vários anos, os dois grupos trabalharam juntos sob o mesmo editor: Ian Hickson. Em 2011, os grupos chegaram à conclusão de que tinham objetivos diferentes: o W3C queria traçar uma linha para os recursos de uma Recomendação HTML 5.0, enquanto o WHATWG desejava continuar trabalhando em um padrão para HTML, mantendo continuamente a especificação e adição de novos recursos. Em meados de 2012, uma nova equipe de edição foi introduzida ao W3C para cuidar da criação de uma Recomendação HTML 5.0 e preparar um Rascunho de Trabalho para a próxima versão HTML (HICKSON *et al*., 2014).

Em 28 de outubro de 2014, o W3C publicou o HTML5 como uma recomendação. O HTML passou a ser um ambiente para a execução de aplicativos complexos, em vez de uma linguagem de marcação simples como costumava ser. Os desenvolvedores começaram a aproveitar esses recursos em vez de *plug-ins*, como o Adobe Flash ou o Silverlight, da Microsoft, menos conhecido, o que levou ao abandono gradual do suporte a *plug-ins* pelos fabricantes de navegadores (HICKSON *et al*., 2014).

Novas versões do HTML passaram a ser recomendadas pelo W3C. Em 1 de novembro de 2016, o W3C publicou o HTML 5.1, como uma recomendação (FAULKNER *et al*., 2016a). Em 18 de agosto de 2016, o W3C publicou o Primeiro Projeto Público de Trabalho do HTML 5.2 (FAULKNER *et al*., 2016b). Em 14 de dezembro de 2017, o W3C publicou o HTML 5.2, como uma recomendação (FAULKNER *et al*., 2017). No mesmo dia, 14 de dezembro de 2017, o W3C também publicou o Primeiro Rascunho Público de Trabalho do HTML 5.3 (AAS *et al*., 2017).

Fique atento

Tecnicamente, o W3C considera oficialmente somente as versões HTML 2.0, HTML 3.2, HTML 4.0, HTML 4.1 e HTML5, 5.1, 5.2 e 5.3. As versões HTML e HTML+ são anteriores à criação do W3C, e a HTML 3.0 não chegou a ser lançada oficialmente, transformando-se na versão HTML 3.2.

Diferenças entre as versões HTML

As versões do HTML representam aprimoramentos padronizados na linguagem fundamental da World Wide Web. À medida que as novas tecnologias são desenvolvidas, assim como métodos mais eficientes para alcançar os resultados desejados das páginas da Web, desenvolvedores e administradores adotam padrões de linguagem aceitos e os designam usando números para trazer ordem e uniformidade à Web.

Os termos HTML, XHTML e HTML5 são frequentemente usados em *web design*. A maioria das pessoas iniciantes em *web design* geralmente acha esses termos confusos e não sabe qual linguagem dessas três aprender. A confusão é compreensível, porque todas elas são linguagens de marcação e essencialmente servem ao mesmo objetivo.

HTML *versus* HTML5

O HTML5 foi desenvolvido por um grupo conhecido como WHATWG, com o objetivo de aprimorar as versões HTML anteriores e resolver alguns problemas de compatibilidade entre navegadores. Usaremos 'HTML' para se referir às versões anteriores ao HTML5 (WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP, 2019). As principais diferenças entre HTML e HTML5 são as seguintes:

- O HTML5 oferece melhor suporte para várias formas de mídia, como áudio e vídeo, fornecendo *tags* para elas. O HTML não contém essas *tags* e conta com *plug-ins* de terceiros.
- O HTML não permite que o JavaScript seja executado no navegador. O HTML5 resolveu esse problema introduzindo a API JS Web Worker. Agora, devido ao suporte nativo ao JavaScript, as páginas da Web podem

ser mais bem-projetadas, usando *scripts* de *front-end* para aprimorar a experiência do usuário.
- O HTML5 inclui novos atributos de entrada, como *e-mail*, URLs, data e hora, pesquisa, etc.
- O HTML5 é independente do dispositivo e tem melhor compatibilidade com o navegador do que as versões HTML anteriores.
- O HTML5 apresenta melhores regras de análise (a análise é o processamento de texto) e maior flexibilidade dessa análise do que o HTML.
- O HTML5 também facilita a localização do local sem *plug-ins* de terceiros.
- O HTML5 também contém suporte nativo para gráficos vetoriais, portanto não há necessidade de *software* de terceiros, como o Adobe Flash.

Em resumo, o HTML5 é apenas uma versão melhor do HTML, com recursos e funcionalidades adicionais.

HTML *versus* XHTML

XHTML é uma família de tipos e módulos de documentos atuais e futuros, que reproduzem e estendem o HTML 4. Os tipos de documento da família XHTML são baseados em XML. XHTML 1.0 é o primeiro tipo de documento na família XHTML, uma reformulação dos três tipos de documentos HTML 4, como aplicativos do XML 1.0. Ele deve ser usado como uma linguagem para conteúdo compatível com XML e, se algumas diretrizes simples forem seguidas, operar em agentes de usuário compatíveis com HTML 4 (PEMBERTON *et al*., 2000).

XHTML foi desenvolvido como uma extensão para HTML. Não há muitas diferenças entre HTML4 e XHTML, sendo este último basicamente uma versão mais rígida do primeiro. As principais diferenças entre HTML e XHTML são as seguintes:

- O HTML4 permite que alguns elementos omitam a *tag* final. As *tags* finais são adicionadas ao fechar uma parte do texto, como um parágrafo. Eles geralmente são simbolizados com uma barra invertida (por exemplo, a *tag* de abertura de um parágrafo é `<p>`, enquanto a *tag* final é `</p>`). XHTML requer que todos os elementos incluam a *tag* final.
- O HTML4 permite a sobreposição de certos elementos. XHTML não permite que nenhum elemento se sobreponha.

- Os valores dos atributos (como tamanho da fonte) devem ser citados em XHTML, mesmo que sejam numéricos. O HTML não inclui valores entre aspas para atributos.
- Os atributos não podem ser minimizados em HTML.
- Há uma pequena diferença na maneira como os elementos vazios são manipulados.

Embora as diferenças acima sejam as mais importantes, também existem outras muito sutis, mas são realmente raras. Você sempre pode verificar a documentação do XHTML para obter mais informações. O argumento é que o XHTML foi projetado para resolver alguns problemas em HTML, incorporando alguns recursos do XML.

HTML5 *versus* XHTML

Como XHTML e HTML são praticamente os mesmos, as diferenças entre XHTML e HTML5 são as mesmas que as existentes entre HTML4 e HTML5. No entanto, a seguir, estão algumas das diferenças mais sutis entre HTML5 e XHTML (PIETERS, 2014):

- Enquanto XHTML faz distinção entre maiúsculas e minúsculas, o HTML5 não (HTML também não faz distinção entre maiúsculas e minúsculas).
- O HTML5 tem um tipo de documento muito mais simples que o XHTML e o HTML (o `DOCTYPE` diz ao navegador como interpretar os dados).
- O HTML5 é compatível com todos os navegadores, enquanto o XHTML não.
- O HTML5 é mais brando, seguindo os passos do HTML4, do que o XHTML.
- O HTML5 é mais adequado para dispositivos móveis, como *tablets* e telefones, enquanto o XHTML é mais adequado para telas de computador.

Como conclusão, todas as três são linguagens de marcação com apenas alguns recursos presentes em cada um que não estão presentes nos outros. Embora o XHTML tenha sido projetado para ser uma versão melhor do HTML4, incorporando alguns recursos do XML, o HTML5 acabou sendo muito melhor do que os dois e é, de longe, a linguagem de marcação mais usada atualmente devido à adição de muitos recursos importantes. O principal argumento é que

todos eles são apenas versões diferentes do padrão HTML das linguagens de marcação, mas com estilos e recursos sintáticos diferentes. Se você é iniciante em *web design*, HTML4 e XHTML não são recomendados, e faria sentido aprender primeiro o HTML5 por causa de sua compatibilidade aprimorada e seu uso generalizado.

Link

Um maior detalhamento de todas as mudanças entre as versões HTML, XHTML e HTML5 pode ser encontrado na especificação da linguagem pelo *link* a seguir.

https://qrgo.page.link/zH5Uj

Mudanças do HTML5

O W3C projetou o padrão HTML5 com vários objetivos: substituir *plug-ins* de multimídia proprietários por padrões abertos, permitindo que os aplicativos da Web comportem-se mais como aplicativos nativos, adicionando recursos para serviços baseados em localização e fazendo alterações na sintaxe que separa o conteúdo da apresentação.

O HTML5 foi feito para ser simples, implicando em uma sintaxe extremamente mais simples e limpa. A simples declaração do tipo de documento (`DOCTYPE`) foi apenas mais uma das facilidades incluídas na nova versão. Agora, você precisa inserir apenas um `<!DOCTYPE html>` no início do seu documento, e tudo estará pronto. Além disso, a sintaxe da HTML5 é compatível também com a HTML4 e XHTML1.

A linguagem apresenta, também, um elemento novo, o `<canvas>`, responsável por substituir muitas das implementações antes feitas em Flash. Isso faz muitos desenvolvedores considerarem que esse já se encontra obsoleto e futuramente morto.

A extensão de *tags* a um tool de novos e interessantes recursos fez uma grande diferença na linguagem. *Tags* como: `<header>` e `<footer>`, que estendem a funcionalidade de tabelas agora para a página como um todo, `<section>`, `<article>` e `<nav>`, que permitem marcar áreas específicas dos *layouts*, `<video>` e `<audio>`, para uma inclusão melhorada de conteúdos

multimídia nas páginas, e <menu> e <figure>, para bem arranjar textos, imagens e menus, trazem todo um conjunto de implementações e funcionalidades bem-pertinentes para a Web de hoje.

Além disso tudo, a remoção de outros recursos, como as *tags* , <big>, <font>, etc., fazem com que a responsabilidade do CSS aliada à nova linguagem só aumente, otimizando o desenvolvimento *front-end* (SILVA, 2011).

Como em todas as atualizações anteriores do padrão HTML, os navegadores HTML5 são compatíveis com versões anteriores. As páginas mais antigas continuam sendo exibidas em novos navegadores, e não será necessário reescrever nenhum documento existente.

O novo padrão é vasto. Por essa razão, o W3C dividiu essas novidades em oito áreas tecnológicas. Isso ajuda na homologação do padrão e suporte dos fabricantes, que estão sendo feitos em porções.

Semântica

Trata-se de um grande desafio para os mecanismos de busca da atualidade classificar a importância e relevância das informações contidas nas páginas. Com esse intuito, novas *tags* foram criadas, dando maior importância semântica aos textos. Essas *tags* se comportam como a tradicional *tag* , que leva este nome por dividir o texto em blocos, até então anônimos. Agora, os mesmos podem ser classificados, seguindo as principais possibilidades (Figura 2):

- <section> — apresenta a sessão, o tema da qual os artigos têm em comum;
- <nav> — trata-se dos elementos para a navegação do visitante, basicamente o menu;
- <article> — contém o texto do artigo, o conteúdo propriamente dito;
- <aside> — informações relacionadas às do artigo, mas que precisam ser separadas do contexto, como notas de rodapé;
- <header> — contém o cabeçalho;
- <footer> — contém o rodapé do artigo;
- <time> — armazena a data e hora daquela informação;
- <mark> — serve para marcar parte do texto, colocando-o em destaque ou para uma referência.

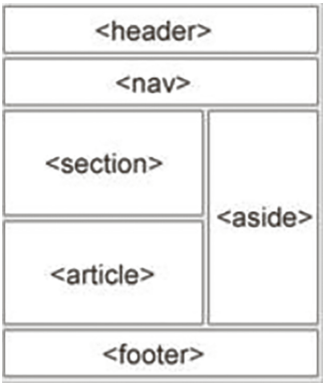

Figura 2. Exemplo de disposição das *tags*.
Fonte: Poyatos (2014, documento *on-line*).

Quando tratamos do uso de formulário, a *tag* `<input>` já é amplamente atualizada para caixas de texto, botões de seleção e de acionamento. Entretanto, quando fosse necessário validar uma informação em uma caixa de texto, longos trechos de codificação JavaScript faziam-se necessários.

Destacam-se, portanto, alguns tipos novos e interessantes para a *tag* `<input>`. Vários foram criados e já possuem validação nativa por parte do navegador Web (Figura 3).

```
01 <!DOCTYPE HTML>
02 <html lang="pt-br">
03  <head>
04   <meta charset="UTF-8">
05   <title>Exemplo de Formulário</title>
06  </head>
07  <body>
08    <form method="POST">
09       <label for="endemail">E-mail: </label><br />
10       <input type="email" name="endemail" placeholder="Digite seu E-mail"
11        required /><br />
12       <label for="idade">Idade: </label><br />
13       <input type="number" name="idade" min=0 max=120
14         placeholder="Informe sua Idade" /><br />
15       <label for="datNasc">Data de Nascimento: </label><br />
16       <input type="date" name="datNasc" />
17       <br />
18       <input type="submit" value="Enviar" />
19     </form>
20  </body>
21 </html>
```

Figura 3. Exemplo de codificação JavaScript.

O exemplo utiliza os tipos `email`, `number` e `date`, embora existam outros. Repare, também, nos novos atributos `placeholder` e `required`. O `placeholder` é responsável por exibir mensagens, como marca d'água na caixa de texto que se encontra vazia. Já o `required` (linha 11) quer dizer obrigatório, ou seja, exige o preenchimento da caixa de texto, impedindo o formulário de ser submetido enquanto a condição não for satisfeita.

O exemplo faz uso, também, dos novos atributos `min` e `max` (linha 13), úteis para se estabelecer intervalos de números para o tipo `number` (Figura 4).

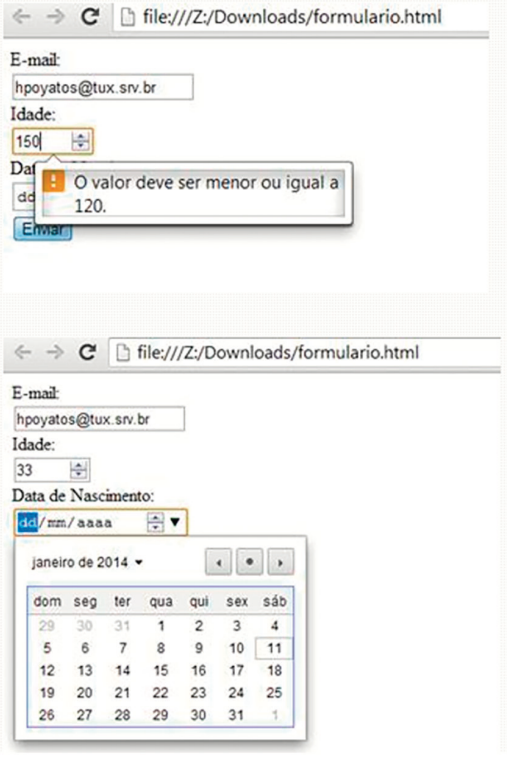

Figura 4. Exemplo anterior em execução. Novos tipos de caixa de texto validados nativamente.

Fonte: Poyatos (2014, documento *on-line*).

Áudio e vídeo

O novo recurso mais comentado do HTML5 é o suporte nativo para reprodução de áudio e vídeo no navegador. Os navegadores atuais contam com *plug-ins*, como Adobe Flash, Apple QuickTime e Microsoft Silverlight, para reproduzir conteúdo multimídia. Para usuários de desktop, esses *plug-ins* funcionam bem e são relativamente transparentes de usar; para o crescente número de usuários que deseja assistir a vídeos em seus iPhones, iPads ou Blackberries, não há *plug-ins* disponíveis. O HTML5 resolveu o problema do *plug-in*, exigindo suporte interno para vídeo em navegadores e permitindo que os designers insiram as *tags* `<video>` e `<audio>` em suas páginas. Essas *tags* contêm informações que o navegador pode usar para interpretar e renderizar o conteúdo multimídia (SILVA, 2011).

Observe o exemplo a seguir, que demonstra o uso do elemento `audio`:

```
<audio src="som.mp3">
</audio>
```

Somente isso, simples e direto, sem necessidade de *plug-ins*, Flash, Quick Time, Windows Media Player e qualquer outro sistema a ser considerado quando se pretende incorporar som em uma página Web. Basta declarar o caminho para o arquivo de som no atributo `src` do elemento. O som é reproduzido com as funcionalidades nativas do navegador.

O elemento `audio` admite os seguintes atributos: atributos globais, `src`, `controls`, `crossorigin`, `preload`, `autoplay`, `mediagroup`, `loop` e `muted` (HICKSON *et al*.; SILVA, 2011).

O atributo `src` destina-se a indicar o caminho para o arquivo de som a ser incorporado na página.

O atributo `controls` destina-se a fazer com que o navegador renderize os controles nativos para o usuário interagir com som. Esse atributo é de uso obrigatório, salvo nos casos em que pretendemos criar controles personalizados.

O atributo `crossorigin` destina-se a indicar a origem do arquivo de som a ser incorporado na página. Os valores possíveis são `Anonymus` e `use-credentials`.

O atributo `preload` destina-se a indicar como o autor espera que seja o carregamento do som na página, tendo em vista otimizar a experiência do usuário.

Gráficos e efeitos em 3D

O elemento `canvas` é outro novo recurso do HTML5. Os desenvolvedores podem combinar gráficos vetoriais, imagens, áudio e vídeo em um espaço predefinido na página. Esses elementos podem ser programados via JavaScript, e os usuários poderão interagir com eles via *mouse* e teclado. A renderização será realizada ao lado do cliente, garantindo um processamento mais suave e menos gargalos na rede. O elemento `canvas` pode conter coisas tão simples, quanto formas ou linhas geométricas, e coisas tão complexas, quanto jogos e modelos em camadas que podem ser manipulados pelo usuário (SILVA, 2011).

É necessário que o desenvolvedor faça uma análise criteriosa para uso desse elemento. Não devemos usá-lo se houver um elemento mais apropriado. Por exemplo: para marcar o cabeçalho ou o topo de um *site*, ainda que ele tenha sido projetado com uso maciço de imagens, talvez seja mais apropriado o uso do elemento `h1` devidamente estilizado, e não o `canvas`.

Atributos HTML

Novos atributos foram introduzidos pela HTML5, vários existentes foram redefinidos e muitos elementos tiveram a lista de atributos suportados ampliada. A HTML4.01 prevê 118 atributos, e a HTML5 expandiu essa lista.

Existem atributos que são globais, ou seja, comuns a todos os elementos da HTML. Um deles é o `accesskey`, que se destina a definir uma tecla de atalho para ativar ou dar foco ao elemento quando o usuário navega com auxílio do teclado. O atributo `class` destina-se a definir uma classe para o elemento com a finalidade de servir de referência para estilização e *scripts*. Por fim, o atributo `contenteditable` foi inventado pela Microsoft e implementado em seus navegadores desde a versão 5, reconhecido e adotado pela HTML5. Admite os valores `true`, `false` e `inherit`, que definem se o conteúdo do elemento é editável ou não, ou se sua condição de edição é herdada. A edição se faz no próprio agente de usuário que renderiza o conteúdo (SILVA, 2011).

Geolocalização

Geolocalização é uma funcionalidade que, apesar de ter sido lançada no ano de 2008, ganhou destaque e começou a ser difundida na mesma época em que a HTML5 ganhou seu espaço e começou a ser estudada e aplicada pela

comunidade de desenvolvimento Web. Talvez essa coincidência tenha sido a responsável pela ideia de que a geolocalização tenha sido criada pelo HTML5 (SILVA, 2011).

Os recursos de geolocalização do HTML5 são de particular interesse de desenvolvedores e usuários de dispositivos móveis. A localização geográfica foi definida como "a arte de descobrir onde você está no mundo e (opcionalmente) compartilhar essas informações com as pessoas em quem você confia". Ao incluir ferramentas para localização geográfica no navegador, o HTML5 facilita a criação de aplicativos que sabem onde os usuários estão, o que está por perto e como eles podem chegar aonde querem ir. Os aplicativos podem incluir mapas e orientações, compras e entretenimento e ferramentas de redes sociais. Embora existam preocupações legítimas de privacidade com esses recursos, deve-se observar que a especificação de design atual exige que os usuários escolham explicitamente permitir que os aplicativos de localização geográfica saibam onde estão (SILVA, 2011).

Armazenamento de dados

O HTML5 também tem suporte para bancos de dados SQL ao lado do cliente e cache *offline*. Isso é particularmente importante para usuários móveis, pois a maioria tem conectividade e largura de banda limitadas em seus dispositivos. Um exemplo desse conceito de "aplicativo em cache local" seria um conjunto de calendários que permite que os usuários baixem seus dados atuais, façam ajustes em seus dispositivos móveis e, posteriormente, carreguem na próxima vez que houver uma conexão disponível. Os aplicativos podem armazenar gráficos e códigos no dispositivo local, garantindo tempos de carregamento rápidos e tráfego de rede mínimo (SILVA, 2011).

APIs para comunicação

Outro novo recurso do padrão HTML5 é a capacidade de criar aplicativos da Web que se comportam mais como locais. Alguns recursos, como gerenciamento de arquivos de arrastar e soltar e armazenamento local, permitirão que os desenvolvedores criem aplicativos baseados na Web, que se integram perfeitamente ao computador do usuário. Com HTML5, os usuários podem fazer *upload* e *download* de arquivos simplesmente arrastando-os para a janela do navegador (SILVA, 2011).

Integração

Um dos grandes problemas no envio de arquivos pela Web (*upload*) foi finalmente resolvido com o chamado `xmlHttpRequest` ao se enviar arquivos muito grandes, não se tinha uma ideia de progresso desse envio (VAN KESTEREN, 2012).

O mérito é de um novo evento incluído na API, `onprogress`, especificamente no atributo `upload`. O mesmo recebe um objeto que possui o atributo `lengthComputable`, que traz informações de progresso que podem ser exibidas em uma barra de progresso (novidade também a *tag* `<progress>`, específica para esse fim) (SILVA, 2011).

A integração homem-máquina também recebeu novidades. O recurso de *drag 'n' drop* (arrastar-e-soltar), tão utilizado em interfaces gráficas dos sistemas operacionais, agora pode ser utilizado em páginas e aplicações Web. Um elemento pode ser considerado *dragable* ("arrastável"), podendo, até mesmo, transportar informações contidas nele para outros elementos (que chamaremos aqui de "alvo").

Muitos são os recursos trazidos pela tecnologia HTML5 e pela nova geração de navegadores Web. Portanto, muitas são as oportunidades de seu uso, que podem ir desde experiências mais ricas de navegação e interatividade em um *website* até avançadíssimos jogos eletrônicos baseados em Web. Animações e *streaming* de áudio e vídeo, antes possíveis apenas pelo *plug-in* proprietário Adobe Flash, agora são realidade de forma nativa.

Tratando em particular dos jogos eletrônicos, WebSockets ou SSE trazem uma troca de informações ágil, essencial para partidas com múltiplos jogadores. Animações realizadas usando Canvas, webGL ou recursos do CSS3, acompanhados por uma envolvente trilha sonora reproduzida pelo novo suporte de *streaming* de áudio, utilizando-se dos eventos de toque em tela, acelerômetro e recurso de *drag 'n' drop*, farão os jogos eletrônicos baseados em Web alcançarem patamares possíveis outrora apenas por jogos instaláveis em sistemas operacionais.

Link

Você pode verificar todas as mudanças que ocorreram da HTML4.01 para HTML5 no *link* a seguir.

https://qrgo.page.link/SiQtS

Fallbacks e condicionais, verificando a existência da API (como em muitos exemplos citados neste capítulo), farão com que até mesmo visitantes oriundos de navegadores mais antigos possam acessar sua página ou aplicação, embora não tenham a mesma experiência. Frameworks como o Modernizr (SILVA, 2011) auxiliam nos *fallbacks* e podem providenciar *polyfills*, que são a possibilidade de preencher a falta de suporte a alguns recursos da HTML5 por outros mais antigos ou *plug-ins*.

Os novos recursos podem ser muito úteis em *layout* responsivo – trata-se de criar versões do *website* planejadas para vários tamanhos de tela e processadores, ou seja, em um único código, criar páginas que se adaptem às telas que vão desde pequenos *smartphones* até televisores de muitas polegadas.

Referências

AAS, P. *et al. HTML 5.3*: W3C First Public Working Draft. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 14 Dec. 2017. Disponível em: https://www.w3.org/TR/2017/WD-html53-20171214/single-page.html. Acesso em: 26 set. 2019.

BIRBECK, M. *et al. XHTML™ 2.0*: W3C Working Group Note. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 16 Dec. 2010. Disponível em: https://www.w3.org/TR/xhtml2/. Acesso em: 26 set. 2019.

DUBINKO, M. *et al. Forms 1.0*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 14 Oct. 2013. Disponível em: https://www.w3.org/TR/2003/REC-xforms-20031014/. Acesso em: 26 set. 2019.

FAQ - WHATWG. *Web Hypertext Application Technology Working Group*, [*S. l.*], 2018. Disponível em: https://whatwg.org/faq. Acesso em: 26 set. 2019.

FAULKNER, S. *et al. HTML 5.2*: W3C First Public Working Draft. Cambridge; Sophia--Antipolis; Minato; Beijing: World Wide Web Consortium, 18 Aug. 2016a. Disponível em: https://www.w3.org/TR/2016/WD-html52-20160818/single-page.html. Acesso em: 26 set. 2019.

FAULKNER, S. *et al. HTML 5.1*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 1 Nov. 2016b. Disponível em: https://www.w3.org/TR/2016/REC-html51-20161101/. Acesso em: 26 set. 2019.

FAULKNER, S. *et al. HTML 5.2*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 14 Dec. 2017. Disponível em: https://www.w3.org/TR/html52/. Acesso em: 26 set. 2019.

FLATSCHART, F. *HTML 5*: embarque imediato. Rio de Janeiro: Brasport, 2011. 256 p.

HICKSON, I. *Web Forms 2.0*: W3C Member Submission. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 11 Apr. 2005. Disponível em: https://www.w3.org/Submission/web-forms2/. Acesso em: 26 set. 2019.

HICKSON, I. *et al. HTML5: A vocabulary and associated APIs for HTML and XHTML*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 28 Oct. 2014. Disponível em: https://www.w3.org/TR/html50/. Acesso em: 26 set. 2019.

HISTORY of HTML. *Web Hypertext Application Technology Working Group*, [*S. l.*], [2018]. Disponível em: https://www.htmlwasher.com/history-of-html/. Acesso em: 26 set. 2019.

JACOBS, I.; WALSH, N. *Architecture of the World Wide Web, Volume One*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 15 Dec. 2004. Disponível em: https://www.w3.org/TR/2004/REC-webarch-20041215/. Acesso em: 26 set. 2019.

LE HORS, A. *et al. Document Object Model (DOM) Level 2 Core Specification: Version 1.0*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 13 Nov. 2000. Disponível em: https://www.w3.org/TR/DOM-Level-2-Core/. Acesso em: 26 set. 2019.

LE HORS, A. *et al. Document Object Model (DOM) Level 3 Core Specification: Version 1.0*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 7 Apr. 2004. Disponível em: https://www.w3.org/TR/DOM-Level-3-Core/. Acesso em: 26 set. 2019.

PEMBERTON, S. *et al. XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition): A Reformulation of HTML 4 in XML 1.0*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 26 Jan. 2000. Disponível em: https://www.w3.org/TR/xhtml1/. Acesso em: 26 set. 2019.

PIETERS, S. *HTML5 Differences from HTML4*: W3C Working Group Note. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 9 Dec. 2014. Disponível em: https://www.w3.org/TR/html5-diff/. Acesso em: 26 set. 2019.

POYATOS, H. Programando em HTML5. *DevMedia*, Rio de Janeiro, 2014. Disponível em: https://www.devmedia.com.br/programando-em-html5/31040. Acesso em: 26 set. 2019.

RAGGETT, D. *HTML 3.2 Reference Specification*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 14 Jan. 1997. Disponível em: https://www.w3.org/TR/REC-html32. Acesso em: 26 set. 2019.

RAGGETT, D.; LE HORS, A.; JACOBS, I. *HTML 4.01 Specification*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 24 Dec. 1999. Disponível em: https://www.w3.org/TR/html401/. Acesso em: 26 set. 2019.

SILVA, M. S. *HTML 5*: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011. 320 p.

VAN KESTEREN, K. *XMLHttpRequest Level 2*: W3C Working Draft. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 17 Jan. 2012. Disponível em: https://www.w3.org/TR/2012/WD-XMLHttpRequest-20120117/. Acesso em: 26 set. 2019.

WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP. HTML: Living Standard. *Web Hypertext Application Technology Working Group*, [*S. l.*], 2019. Disponível em: https://html.spec.whatwg.org/. Acesso em: 26 set. 2019.

WOOD, L. *et al*. *Document Object Model (DOM) Level 1 Specification: Version 1.0*: W3C Recommendation. Cambridge; Sophia-Antipolis; Minato; Beijing: World Wide Web Consortium, 1 Oct. 1998. Disponível em: https://www.w3.org/TR/REC-DOM-Level-1/. Acesso em: 26 set. 2019.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS