

スマートフォン/タブレット向け技術としての HTML5 の現在

iPhone の登場以来、HTML5 対応を謳うサイトが増えている。実際、iPhone 専用サイトには、従来の Web には無い操作性や機能を実現したものが少なくない。iPad や Android 端末も登場し、デスクトップには IE9 がもうすぐ公開される。今後、HTML5 の利用はますます広がることだろう。しかし、その一方で、HTML5 の仕様はまだ策定中で、最終草案にもなっていない段階だ。このような状況で、HTML5 にはどんなスタンスで取り組むのが良いのだろうか。W3C HTML5 Japanese Interest Group の議長をつとめる矢倉氏に、HTML5 について、スマートフォン/タブレット向けの技術という観点から解説をお願いした。

インタビューした人：矢倉 眞隆氏

株式会社ミツエーリンクス R&D 本部 Web 標準チームチームに所属。Web 標準 Blog (<http://standards.mitsue.co.jp/> (www.adobe.com/http://standards.mitsue.co.jp/)) を担当し、標準化作業の紹介、標準のあるべき姿、仕様書の翻訳など、Web 標準関連の情報を発信する。また、Web 標準準拠サービスの提供を行い、Web 標準に準拠したサイト構築、コンテンツの有効活用や品質管理の支援を行う。Web が進化する技術なので、その時代に合わせて Web 標準がどうあるべきかを考えていくのが仕事。

Dreamweaver CS5

体験版 (www.adobe.com/go/trydreamweaver_jp) 今すぐ購入 (www.adobe.com/go/buydreamweaver)

"ブラウザ" だけで出来る事のすべてが HTML5

「狭い定義としては、HTML5 は W3C 仕様書 ([HTML5 \(www.adobe.com/http://www.w3.org/TR/html5/\)](http://www.adobe.com/http://www.w3.org/TR/html5/)) として公開されているものになります。でも、『これだけが?』と言われると全く違う。Web Storage のようなアプリが使う API 仕様も、もともと HTML5 仕様書の中にあっただけという事もあり、HTML5 と言われます。ちょっと大げさですけど、HTML5 は次世代 Web 技術の総称と言えるかと。仕様が複数に分割されたのは、HTML5 が 1 つの大きな仕様になって策定に時間がかかるのを避けるため、他にも、モバイルからデザインまで様々な仕様が個別に議論されています。どこまでが HTML5 かと言うのは難しいですが、Web ブラウザだけで出来る事が HTML5 と一般的には考えてよいかもしれませんね」

そもそも HTML5 とは? という問いに対し、矢倉氏から頂いたのが上の答えだ。同じ HTML でも、HTML5 は HTML4 よりもずいぶん指す範囲が広がったという印象を受ける。実際に W3C の HTML5 と HTML4 の違いについてのドキュメント ([HTML5 differences from HTML4 \(www.adobe.com/http://standards.mitsue.co.jp/resources/w3c/TR/html5-diff/\)](http://www.adobe.com/http://standards.mitsue.co.jp/resources/w3c/TR/html5-diff/)) には、主要な変更点として、文書のマークアップの改善の他に、以下の 2 点が含まれている。

- Web アプリケーションのような新しい表現に対し、マークアップと API を提供する
- 詳細な処理モデルを定義し、相互運用可能な仕様の実装を手助けする

1 つ目の項目に、やはり、Web ブラウザだけで出来ることをより広げることが HTML5 の主目的の 1 つである、と明確に述べられている。canvas タグや video タグの追加も、この流れの一環として考えてよさそうだ。

2 つ目の項目は、より興味深い。これは、HTML と DOM がどう関わるかが明確に定義されていなかったことに大きな原因があるそうだ。DOM を操作した時のブラウザの挙動について、不明確な部分は各ブラウザベンダーが個別に仕様を決めざるを得ず、非互換を生み出す原因となっていた。JavaScript の使用が広がるにつれ、この問題は急速に顕著になった。そこで、"詳細な処理モデル" も W3C 仕様書に含められることとなったというわけだ。

HTML5 では、各ブラウザベンダーが独自に実装していた多くの事が、共通仕様を元に実装される。いままでは、実装が先行して、開発者が実装間の差を埋めるという作業が必要になってしまっていた。「いろんな事が安心してできる。地ならしをしている事に対してもっと評価して欲しいですね」は矢倉氏の弁だ。

デバイス向けに役立つ HTML5 の機能

「スマートフォンのブラウザは HTML5 対応が進んでますし、HTML5 にはモバイルを意識した機能があるので」、ということで、矢倉氏より紹介された、スマートフォン/タブレット向けに有効と思われる HTML5 の機能を幾つか具体的に見ていこう。

メディアクエリー

まずはメディアクエリーから。media 属性は、既に表示用／印刷用など専用の CSS を指定する目的で提供されていた。メディアクエリーでは、これが拡張されて、画面の幅、高さ、解像度などの条件により、適用する CSS を指定できるようになっている。iPhone の画面サイズを利用すれば、iPhone 専用の CSS を適用することも可能だ。

```
<link rel="stylesheet"
  media="only screen and (max-device-width: 480px)"
  href="iphone.css"
/>
```

(参考までに、メディアクエリーの仕様は矢倉氏が翻訳したものが公開されている：メディアクエリー (www.adobe.com/http://standards.mitsue.co.jp/resources/w3c/TR/css3-mediaqueries/))

メディアクエリーを使うと、表示領域の幅に応じて CSS を切り替えることが可能だ。そのため、1 つの HTML ソースを異なる画面サイズに対応させるといった使い方もできる。例えば、デスクトップ上では 3 カラム、タブレット上では 2 カラム、スマートフォン上では 1 カラム用の CSS を選択するという使い方だ。

Dreamweaver CS5 の機能拡張として Adobe Labs に公開されている HTML5 Pack for Dreamweaver CS5 は、メディアクエリーをオーサリング環境に持ち込んだ最初の製品だ。3 種類の画面サイズで、それぞれ異なるスタイルが適用された状態を、同時に表示する機能を持つ。CSS ごと個別に確認する必要が無いため、メディアクエリー使用時の作業効率の向上が期待できる。



マルチスクリーンプレビューパネル。3 種類の CSS をあてた状態がひとめで確認できる。

メディアクエリーについて、「最初からマルチデバイスを想定して作る場合であれば、制作コストは抑えられるかもしれない。だが、CSS だけではレイアウト変更が難しい場合、例えばいま既にあるサイトを前提に進めないといけないとき、メディアクエリーは使いづらいだろう」というのが矢倉氏の見方だ。「それでも、多少の調整には有効。例えば、デスクトップ用の表示状態から、タブレットだけタッチ用にリンクの高さを広げるとか」

また、矢倉氏は、SVG とメディアクエリーを組み合わせた例も紹介してくれた。下の例では、ベクター画像の線の太さを調整することで、アイコンが小さくなくても、アイコンとして認識しやすい形状になるよう調整されている。



SVG とメディアクエリーを組み合わせたアイコンの例 (SVG logo with media queries (www.adobe.com/http://people.opera.com/andreasb/demos/demos_svgopen2009/svgscalelogo.html) - Opera のみで動作)。左側のアイコンは、メディアクエリーを使うことで、縮小されても外側の線が細くならないよう設定されている。

他にも、グラフの目盛の大きさや位置が表示サイズに応じて変わる例 (SVG chart with media queries (www.adobe.com/http://people.opera.com/andreasb/demos/demos_svgopen2009/svgscalegold.html) - これも Opera のみ) など、HTML のレイアウト以外にも様々な可能性がありそうだ。「マルチデバイスが本格的になった時代には、もっといろんな使われ方をしているでしょう」

ローカルにデータを保管

「スマートフォンだと難しくてもタブレットだと実用的なアプリがあります。例えばスプレッドシートやワープロ。そうすると Web アプリが有効な範囲も一段広がる。でも、通信への接続が必ずしも保証される訳ではないので、切れたときにも引き続き作業できないと実用的でない」。これは、モバイルでバイス共通の問題点だろう。HTML5 では、アプリケーションキャッシュや Web ストレージといった機能を使って、デバイスの記憶域を利用できる。

アプリケーションキャッシュは、html 要素の manifest 属性を使って、保管したいファイルを記述したマニフェストファイルの URI を指定する。

```
<!DOCTYPE HTML>
<html manifest="cache-manifest">
...
```

必要なファイルをローカルにキャッシュしておけば、ネットワーク未接続の状態でも Web アプリを確実に利用できる。また、キャッシュすることで通信量が減らせるという利点もある。ただし、キャッシュされたファイルは読み込まれた状態のままになるため、データ保管庫としての利用には適していない。

Web ストレージ (Web Storage (www.adobe.com/http://www.w3.org/TR/webstorage/)) はクッキーを拡張したような規格で、5 MB (から 10 MB) 程度の大きさの領域を使用できる。こちらはデータ保管向きの機能だ。データは、名前と値の組み合わせからなる連想配列形式で保存される。

```
localStorage.setItem("title", "タイトル");
localStorage.setItem("body", "本文");
```

この他にも、SQL を利用する規格 (Web SQL Database) があるが、「ベンダー間の合意がとれなかったため、現在は特定の実装に依存しない規格 (Indexed Database) の策定を進めている。Web SQL Database は WebKit と Opera で実装され、Gmail のスマートフォン版で利用されているが、その他のブラウザでは使えない。Indexed Database の標準化や実装にもまだ時間がかかる」。クロスブラウザなデータの保管は、Web ストレージを使うしかないようだ。

プラグイン不要のアニメーション

Canvas を使うとアニメーションが実現できる。Canvas は 2 次元グラフィック API を使って描画する。プログラマー向きの技術と言えるだろう。オブジェクトではなく線を描く API のため、アニメーションはパラパラ漫画のように 1 枚 1 枚 JavaScript で描くことになる。「単純な API のため比較的高速という利点はあるが、複雑な描画は、グループやレイヤーと

言った機能を持つフレームワークが出ないと辛いかも」

CSS 側で動的な効果を出すための仕組みも CSS3 で追加されるようだ。「CSS トランジションという仕様では例えば、1秒間で背景を赤から青に変えるという表現を 1 行で記述できる。コードが簡単で、コピペして使いやすい。デザイナーにも取っ付き易いと思います」。複雑なアニメーションには CSS アニメーションという仕様が別にあるが、対応ブラウザが、Safari と Chrome のみというのが、まだ使いづらいところか。

解像度の違いへの対応

デスクトップ環境では解像度は 96 dpi で一定だが、デバイスでは様々だ。例えば、iPhone3 の解像度は 163 dpi、iPhone4 の解像度は 326 dpi。一般にデバイスの方が解像度が高いため、デスクトップ上で作成したビットマップ画像は、デバイス上で表示すると小さくなる。「ビットマップを拡大/縮小すると、画像が汚くなってしまう。そのため、様々な解像度のデバイスが存在する環境では、ビットマップ画像は使いにくい。CSS や SVG のように、ビットマップ画像に頼らない方法が主流になるのでは」というのが矢倉氏の見方だ。

また、レイアウトに関しては、「ピクセルに依存するデザイン、つまみ幅を固定にしまうと、iPhone では幅 320 ピクセルでできれいに見えても、Android 端末の幅480 ピクセルでは間延びして見えるということが起きる。レイアウトは、相対単位 % を組み合わせた柔軟な指定が必要になってくるでしょう。コンテンツに関しては、どうでもいいと思うんですけど。」

まだレイアウトに関しては CSS は未熟とのこと。CSS3 では Flex Box と呼ばれる新しいレイアウト機能が追加されるが、実装はまだ限定的だそう。

どこで情報を収集したらよいか

「まだ情報が分散している状態。また不確かな情報もある。実装も動いてれば、使う方の考え方も変えないといけない時期なので、答えるのが難しいですが」、と言いながらも、幾つかのサイトを教えて頂いた。

まず、HTML5 の仕様を学ぶのであれば：

- <http://www.whatwg.org/html5> (www.adobe.com<http://www.whatwg.org/html5>)
- <http://www.html5rocks.com/> (www.adobe.com<http://www.html5rocks.com/>)
- <http://html5doctor.com/> (www.adobe.com<http://html5doctor.com/>)
- <http://html5.jp> (www.adobe.com<http://html5.jp/>)

それから、ブラウザの実装状況は、各ブラウザベンダーのサイトをチェック。一番充実しているのは Mozilla とか。逆に Android は殆ど情報が無いとも。

- <https://developer.mozilla.org/ja/HTML/HTML5> (www.adobe.com<https://developer.mozilla.org/ja/HTML/HTML5>)
- <http://developer.apple.com/safari/> (www.adobe.com<http://developer.apple.com/safari/>)
- <http://www.opera.com/docs/specs/> (www.adobe.com<http://www.opera.com/docs/specs/>)
- <http://ietestdrive.com/> (www.adobe.com<http://ietestdrive.com/>)

その他に、ブラウザの差異をまとめているサイトもいくつか：

- <http://www.findmebyip.com/litmus/> (www.adobe.com<http://www.findmebyip.com/litmus/>)
- <http://quirksmode.org/> (www.adobe.com<http://quirksmode.org/>)

最後に、フレームワーク関連は：

- <http://www.sencha.com/products/touch/> (www.adobe.com<http://www.sencha.com/products/touch/>)
- <http://raphaeljs.com/> (www.adobe.com<http://raphaeljs.com/>)

「フレームワークは、ただ使うだけでなく、HTML5 タグを理解した上で、実際にフレームワークが何を埋めてくれているのか、その価値を理解することで HTML5 を学ぶ良い手段としても利用できる」、というアドバイスもご参考まで。

上条 晃宏氏： アドビにて Flash プラットフォーム上のユーザー体験やアーキテクチャデザインのコンサルティングを担当、数多くの Flash / Flex プロジェクトに携わる。

Copyright © 2011 Adobe Systems Incorporated. All Rights Reserved* (www.adobe.com/go/gftray_foot_all_rights_reserved_jp) .

* 英文のみ