

HTML5

吉井 拓郎 宮地 正大
Takuro YOSHII Masahiro MIYAJI

1 はじめに

インターネットが始まってから現在に至るまでに莫大なデータ量の情報がインターネット上に知的財産として存在している。これらの情報を私たちは効率よく、ほしい時に的確な情報を得ることが必要不可欠である。この手段として 1990 年代に Hyper Transfer Markup Language (HTML) が公式な仕様として定義された。そしてかつては情報といっても文字のみであったが画像や音声、動画等の容量の大きい情報も存在するようになり、これらの情報を効率よく得るために HTML も進化し続け、現在では HTML5 の策定を Hypertext Application Technology Working Group (WHATWG) と World Wide Web Consortium (W3C) が進めている¹⁾。本稿では HTML5 へ改訂されてから何が出来るようになったのか、また必要とされなくなったものについて説明し、現状および今後の展望を述べる。

2 HTML5

HTML は修飾したい文字の前後にタグを用いることで、画像の埋め込みや文字の装飾等ウェブサイト上のドキュメントを記述するための言語である。HTML5 はこの HTML の 5 回目にあたる大幅な改訂である。多くのタグや API が追加され、プラグインを使わずに一部メディアを再生することも可能となった。HTML5 の本質は単なる枠であり、ブラウザを立ち上げると、どのようなメディアを見てもブラウザがサポートを行い、またネイティブアプリのように速くて安定した Web アプリを起動させることが理想となっている。

2.1 HTML の歴史

HTML は、Standard Generalized Markup Language (SGML) という文書の電子化のための書式規格を基に開発された。HTML は 1.0 から 4.01 まで進化を遂げた。しかし HTML は化学式や数式といった変わった表現を扱いにくく、また囲むべきタグを省略して記述可能であるためソースを見たときに分かりづらい問題が発生した。この問題を解決するために Extensible Hypertext Markup Language (XHTML) が誕生した。XHTML を用いる利点として、厳密な文書となるためにブラウザの動作が安定や、ウェブサイト装飾する Cascading Style Sheets (CSS) を使用する際の処理速度の向上が挙げられる。また、数式や化学式等の特殊な文字も扱えるようになった。しかし、XHTML は実際に使ってみると、HTML よりも分かりにくく、使いづらいため、期待されるほど普及しなかった。そこで W3C は XHTML2.0 の

策定を打ち切り、HTML5 の標準化に力を注ぐことを決定した。このように、Fig. 1 のような HTML の歴史があり、今後 HTML5 が策定される。

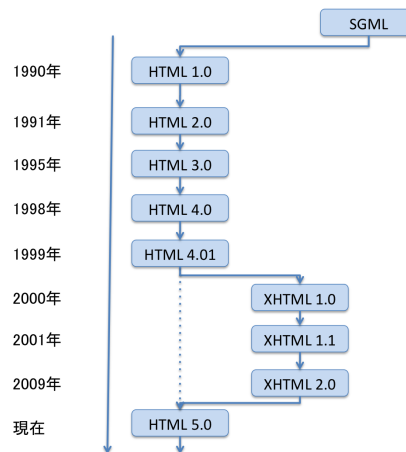


Fig.1 HTML の歴史

2.2 対応ブラウザ

2010 年 3 月現在、Firefox3.5 以降や Safari3.1 以降、OPERA10.5、GoogleChrome3.0 以降、Internet Explorer9 などの先進的なブラウザでは HTML5 の一部機能が段階的に対応している²⁾。

3 HTML5 への移行

HTML5 では

- 文章や数値に意味付けを行う
- 文章の構造を明確に識別
- プラグイン依存からの脱却

以上のようなことをコンセプトとしおり、検索精度の向上や、HTML の記述の簡略化や動作保証を目標としている³⁾。

3.1 廃止された要素

上記のコンセプトにより、HTML5 では Table 1 のタグが廃止された。しかし、後方互換性の問題があるため急に今まで使用していたタグを廃止としてしまうと、正しく表示されないサイトが多数発生するため、HTML5 対応のブラウザでは今回廃止されたタグもそのまま継続して使用可能にする予定である。

3.2 セマンティックウェブへ

現在、検索サイトの主な検索方式は、語句を入力した後、ウェブサイト上の語句を入力語句と一致させ、一致す

要素名	説明
acronym	頭文語
applet	Java アプレットの埋め込み
basefont	ページの基準フォントサイズ
big	フォントを大きく表示
center	要素の中央寄せ
dir	ディレクトリリスト
font	フォント装飾
frame	フレームの定義
frameset	フレームの構成
isindex	キーワード検索入力欄の埋め込み
noframes	フレーム未対応ユーザー向けの代替テキスト
s	テキストに取消線を引く
strike	テキストに取消線を引く
tt	テキストを等幅フォントにする
u	テキストに下線を引く

Table1 廃止された要素

ればそのウェブサイトを表示する方式を取っている。セマンティックウェブは W3C のティム・バーナーズ＝リーによって提唱され、ウェブページの文字を扱うだけでなく、意味も扱うことによって利便性を向上させるプロジェクトである。これらのスタートとして、セマンティック要素が HTML5 で追加された。セマンティックウェブを用いると 2000 年 1 月 1 日の記事を調べる際、2000 年 1 月 1 日と同じ意味である「平成 12 年元旦」と検索しても、2000 年 1 月 1 日の記事がヒットするようになる。

3.2.1 ブロック・セマンティック要素

ブロック・セマンティックには、文章に意味付けを行い、ソースコードの構造の明確化する目的がある。HTML4 以前ではグループ化の要素にあたる div タグを利用し文章のまとまりごとにグループ化を行ってきたが、ソースコードの構造を明確にするために、グループ化を意味する header や footer 等のタグが追加された。実際にこれらのソースコードを Fig. 2 に示す。また、複数の人の発言が会話形式で記述されていることを明確にする dialog タグや、図と文章に意味付けする figure タグが追加された。これらを用いることで文章の意味付けも可能となる。

3.2.2 インライン・セマンティック要素

HTML5 ではホームページの本文中に登場する数値に対する意味を関連付けることが可能なインライン・セマンティック要素がある。Fig. 3 のように、検索エンジンが 20 という数字を見たとき、これがテストの点なのか、また日付であるのか判断することは難しい。しかし、インライン・セマンティック要素により Fig. 3 のような meter や、time といったタグを用いることにより、数字に意味を持たせることができる。

HTML4

HTML5

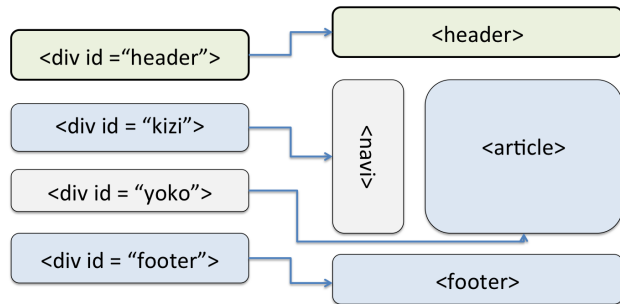


Fig.2 HTML4 と HTML5 のグループ比較 (4) より参照)

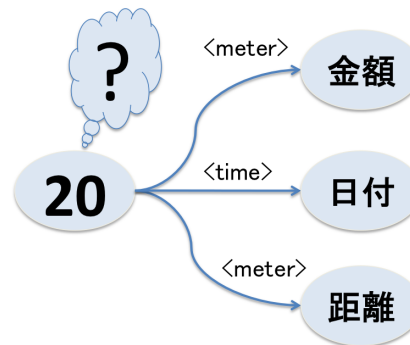


Fig.3 数字のセマンティック化

3.3 追加 API

HTML5 ではセマンティック要素のように、新たな API が追加された。代表的な追加 API として以下のようなものが挙げられる。

- オフライン Web アプリケーション
- 位置情報の取得
- ビデオやオーディオの再生
- 対話型要素
- グラフィック描写
- 簡易データベース (Web storage)
- マルチスレッド (Web Workers)
- 双方向通信 (Web Sockets)
- ドラッグ&ドロップ

HTML4.0 以前、アニメーションの実現や図をウェブ上に描くには Flash Player や Java アプレットのプラグインが使われ、また動画再生には Flash Player が用いられてきた。このように上記の追加された API は、プラグインを使用して実現していたが、HTML5 ではプラグインを使用せずに JavaScript で実現可能となった。プラグインはソフトウェアに機能を追加する小さなプログラムである。プラグインを用いる利点としては、プラグインとソフトウェアを分けることで、ソフトウェア本体をスリム化し、欲しい機能のみプラグインで追加できる利点

がある。一方で、プラグインを用いると相性の問題や予期せぬ動作により、クラッシュしてしまうこともある。よってHTML5では専用プラグインに頼らないというビジョンを推進している。

4 Flash と HTML5

HTML5 の策定により、上記のような新要素が追加され、ブラウザのみで行えることが増加した。そして、HTML5 では、Fig. 4 のように Flash のプラグインを用いて行っていた動画の再生や音楽の再生も、Flash Player を用いずに実現可能となった。このように、HTML5 は Flash にしか出来なかった範囲をサポートし始めた。ここでは、動画再生の観点から Flash と HTML5 の比較を行う。

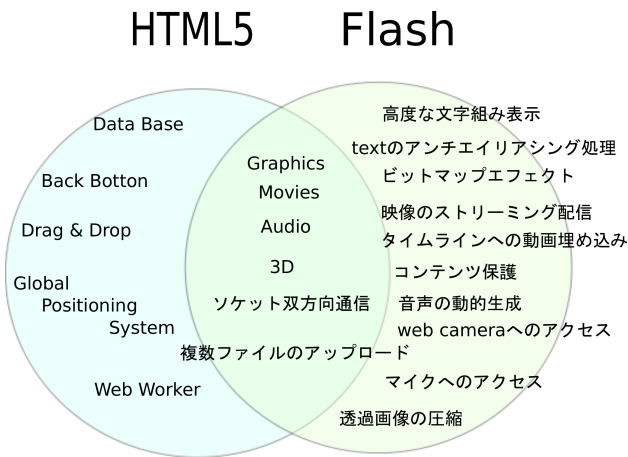


Fig.4 Flash vs HTML5⁽⁵⁾ より参照)

4.1 動画再生に関する技術的な違い

現在、ストリーミング動画再生サイトで Flash Player を用いて動画再生を行うと、クライアントマシンの GPU の有無に関わらず CPU のみで動画の再生を行っている。しかし HTML5 の video タグを利用することで、CPU 以外にも GPU を利用しながら動画再生が可能となった。これにより、CPU の力不足により処理の遅延があった高画質な動画でも、CPU と同時に GPU を利用することで動画再生をスムーズに行える。

5 メリット

現在 iPhone や米国で発売された iPad には Flash Player が搭載されていない環境であるが、アプリケーションを通じてウェブサイト上の動画を変換することにより現在では視聴が可能となっている。このようにアプリケーションを通じてストリーミング動画再生サイトを見る必要があり、ブラウザでは見ることが不可能である。これらの機器に Flash が未対応の理由として、デバイスのパフォーマンスの悪化やプラグインによるバグが多いからだといわれている。HTML5 での video タグの実装により、モバイルデバイスでもブラウザが対応すれば動画の視聴が可能となる。更にはスロー再生や早送り等の

動画の制御も可能となり、プラグインを使わないので動作の安定にも繋がる。また、GPU 動画再生支援により、動画再生を CPU と GPU を用いて再生できる。このような多くのメリット挙げられる。

6 今後の展望

私の見解では、Fig. 1 を見ると、HTML は平均的に 4 から 10 年に一度新たな策定が行われているので、HTML6 が 2020 年には策定される可能性がある。HTML5 ではセマンティックウェブへの移行や、多くの API が追加された。もし HTML5 以後に HTML6 というものが策定されるのであれば、パワーポイントやエクセル、ワードといったパソコンに必要不可欠なアプリケーションを、ブラウザ上で扱えるようになるかもしれない。今後全てのアプリケーションやメディアをブラウザ一つで動作させることが可能になると予想される。

7 まとめ

HTML5 の策定により、これまでも XHTML2.0 や HTML4.0 と比べるとセマンティックウェブへの移行により、検索エンジンとの親和性が高まり、検索精度が向上する。またプラグインを必要としていた動画再生や音楽再生も、プラグインなしで実現できるようになった。このように、プラグイン依存からの脱却も進んでいる。HTML5 で追加された機能を最大限に利用し、応用したウェブサイトを多く作るにより、今後 HTML5 はより広がっていき、まだ現在未対応となっているブラウザは対応せざるをえない状況になっていくだろう。

参考文献

- 1) Web 標準の基礎 (1) ～XHTML の基礎～。
<http://mikilab.doshisha.ac.jp/dia/research/report/2007/0911/003/report20>
- 2) Css3, html5 のブラウザ対応状況一覧。
<http://blog.creamu.com/mt/2009/10/css3html5.html>
- 3) HTML5 とは？
<http://www.html5-guide.com/>
- 4) HTML5 を使ったブログのデザイン。
<http://www.html5.jp/html5doctor/designing-a-blog-with-html5.html>
- 5) Flash だどこまでできる! HTML5 と Flash の機能比較。
<http://clockmaker.jp/blog/2010/02/flash-vs-html5/>