

001 HTML 5

HTML4.01 が施工されてから 11 年ぶりにバージョンが上がります。HTML4.01 が勧告され、最終バージョンという事で長期に渡り Web サイト作成に多く用いられてきました。

そこから、XML が Web サイトにも多く使用される事となり、2000 年以降 XHTML 1.0 として新しく Web 標準という言葉と共に今では、HTML の代わりとして XHTML が主流となっています。

HTML4.01 の限界

1993 年 Web という世界が幕開けし、現在 HTML のバージョンは 4.01 が最新のバージョンになります。W3C における HTML の開発は終了しました。それに伴い、W3C は今後 HTML から XML ベースへシフトしていく為に、Semantic Web（セマンティックウェブ）を理想とし、XHTML へシフトしていく事となります。

マークアップが XML ベースになる事で、厳格化した記述がマシンへの読み取りやすくなりました。

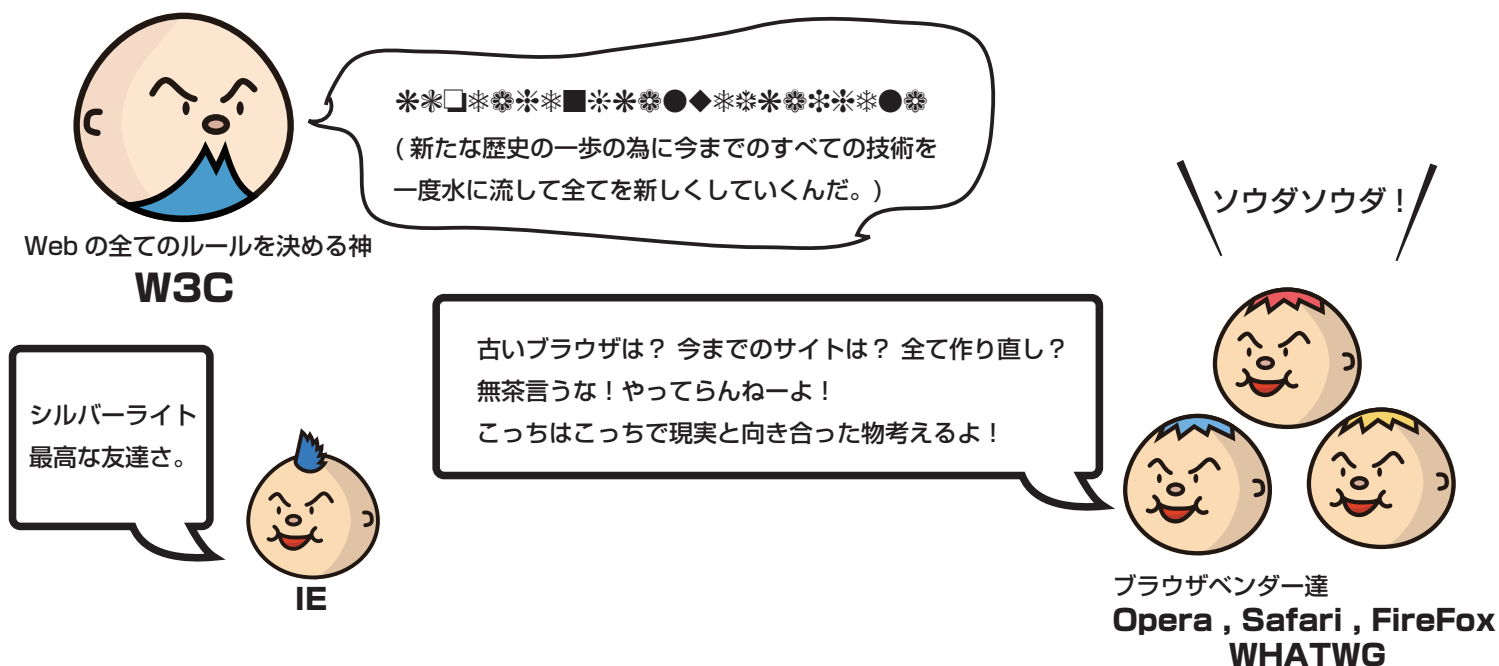
しかし、それと引き換えに人にとっては非常に難しいマークアップとならざるを得ず、XHTML 1.0 では HTML への後方互換性が考慮されていた為、現在でも広く普及していますが、XHTML 1.1 からはより XML の色が濃く表れるようになり、**XHTML 2.0 では、後方互換性が考慮されない** W3C の移行に各デベロッパー、ベンダーは次第についていけなくなってきました。

現実では、HTML4.01 か XHTML 1.0 でほとんどすべてのウェブコンテンツは作られており、W3C の理想と現実とは相反するものとなってきます。

ブラウザベンダーが結束・誕生

一方このやり方に疑問を抱く各ブラウザベンダーですが、W3C のやり方に相反する様に後方互換性やウェブ制作者の移行を汲み取った新しい HTML 策定の為、Opera, Mozilla, Apple は WHATWG というワーキンググループを独自に設立しました。これが HTML5 の誕生になります。

ここまでの流れ



002 HTML 5 の設計原則

では、2007 年から新たに策定されることとなった HTML5 ですが、HTML の過去の過ちを犯さない様あらたな設計に取り組む訳ですが、その設計原則を記した文書「HTML Design Principles (HTML の設計原則)」を公開しました。これまでの HTML や XML の設計原則とは異なった特徴を備えています。

設計原則 4 つのポイント

もともと W3C の方向性に異を唱えた WHATWG ですから、これまでの W3C の方針とはやはり大きく異ってきます。さらに W3C が HTML5 の方針を認めた以上 WHATWG と調和する必要がありこれまでの HTML で得てきたノウハウと WHATWG から掲げる革新的な考え方を上手く融合する必要がありました。

HTML5 策定で重要視されるポイントは大きく分類すると

互換性

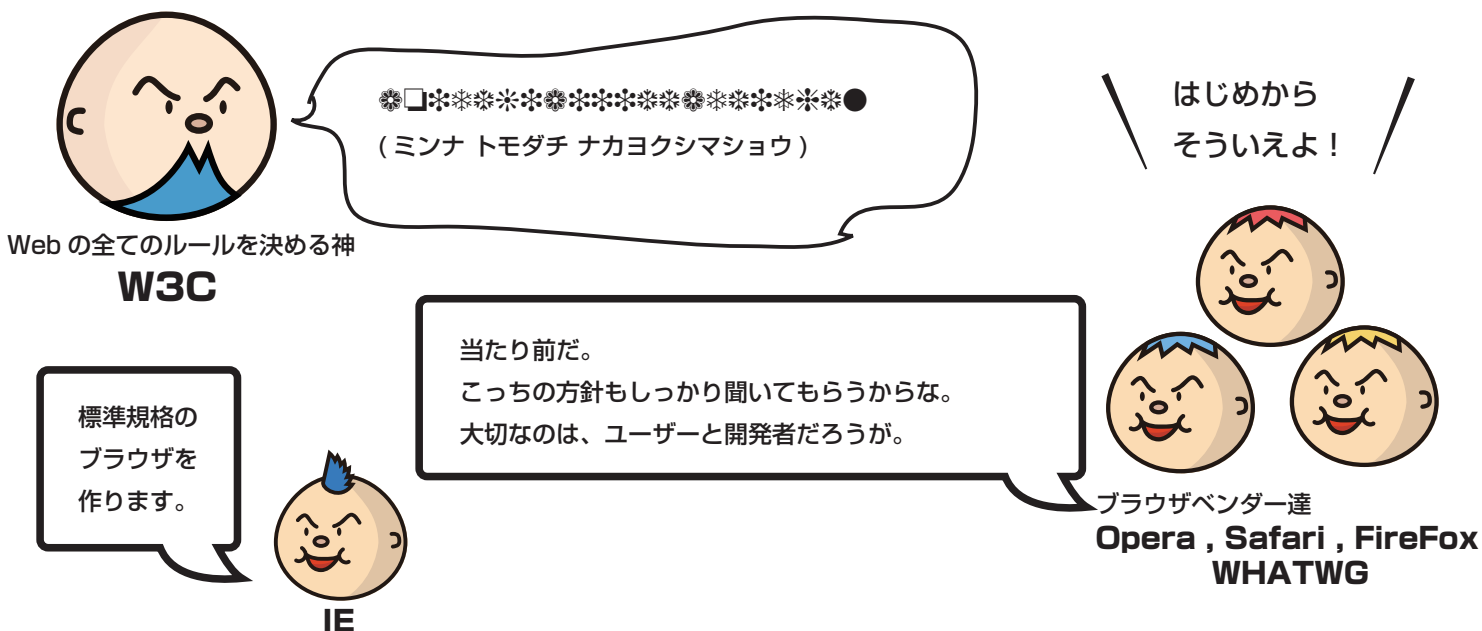
実用性

相互運用性

ユニバーサルアクセス

の4つになります。それぞれの原則や概要をまずは把握してみましょう。

ここまでの流れ



基本的に IE はこの時まだまだ独自路線。

やっとこさ 2008 年リリースされた IE 8 から Web 標準規格への移行が始まるのがこの頃。

まだまだ IE ユーザーが多かったこの頃、Firefox を筆頭にユーザーが取られ始めてやっと本格的な始動が始まっていく頃である。しかし、当時は IE も未来の自分の人気がここまで低迷しようとは思っていなかっただろう。

003 HTML 5 の設計原則 1. 互換性

① コンテンツの互換性

コンテンツの互換性です。

これまで作られてきたウェブサイトもこれまで通り、問題なく利用可能となることが考慮されます。※XHTML2.0 と異なる点この原則のおかげで、HTML5 にブラウザが準拠した場合でも過去のウェブコンテンツが利用不可になるわけではありません。

※しかし、**ありとあらゆる過去のコンテンツが保証されるわけではありません。**

② 古いブラウザの互換性

2つ目は古いブラウザに対する互換性です。

HTML5 では、新たな機能が数多く盛り込まれています。しかし、古いブラウザで機能しないようでは利用がためられます。古いブラウザを利用し、サポートしていない場合は、一部の機能を落としてでもある程度の機能を提供出来るようにするか、変わるとなるコンテンツを表示する仕組みを導入する必要があります。(フォールバックコンテンツ)

③ 機能の再利用

3つ目は機能の再利用です。

もともと独自規格をとっていた各ブラウザベンダーですが、独自ゆえに該当するブラウザでのみ利用出来る事が普及の妨げになってきました。独自で開発された機能の中には非常に役に立つ機能が数多く含まれている為、これと同じ機能を実現する為に新たに開発する事は避け、そのまま仕様を標準として採用することとしています。

例えば、contenteditable 属性というものがあります。

これはもともと、IE5.5 から実装されている機能ですが HTML5 で正式に採用されました。

④ 利用方法の互換性

4つ目は利用方法の互換性です。

HTML5 のマークアップでは、XHTML のような XML ベースのマークアップではありません。あくまで HTML がベースになっているため、br 要素などの終了タグを表す、
 スラッシュは**不要になります**。

しかし、すでに XHTML が普及した現在で考えた場合、いままでのマークアップを否定するということは、過去の Web コンテンツをすべて変更する必要がある為、ウェブ制作の現場で大きな混乱が予想されます。その為、HTML5 の設計原則では、**過去の利用方法を出来る限り認められています**。

⑤ 革新より発展を優先

5つ目は革新よりも発展を優先させる点です。

あいまいな表現になりますが、理想の為に新たな技術を考えるのではなく、今あるものを改良していく、今あるものの延長線上に新しいものを加えていく事が原則になります。

「HTML5 になったからとはいえ、全く別の言語にはならない」という事です。

004 2. 実用性

① 実用性

実用性は、新しく仕様を開発していく点で非常に重要なことです。理想をただおいかけるのではなく、ウェブ制作の現場において必要とされていることを優先しなければなりません。さらにその優先度についても原則としてうたわれています。

【ユーザー】>【作成者】>【実装者】>【仕様書】>【理論上の純粋さ】

の順番で優先される事になります。

もちろんセキュリティについても十分考慮すべき点として原則では触れられています。

005 3. 相互運用性

① 相互運用性

HTML5では、ブラウザの挙動についても明確に定義されています。

これにより、制作者はブラウザの違いを意識せず HTML5に準拠したブラウザすべて同じ挙動になる事が保証されます。

006 4. ユニバーサルアクセス

① ユニバーサルアクセス

HTML5の開発原則では、あらゆるメディア（PC の他モバイル端末やテレビ、ゲーム機など）あらゆる言語、そしてあらゆる人（ハンディキャップをもった人も含む）でもアクセス出来るように機能を開発することをうたっています。

例えば img 要素の alt 属性では、利用シーンに応じてどのようなコンテンツを入れるべきか規定されています。

もちろん新しい機能についても、アクセシビリティは重視されています。

video 要素や audio 要素を用いることで音楽や映像を挿入する事ができますが、そこには字幕を入れる仕組みも導入されています。

あくまでも原則

HTML5の開発原則について触れましたが、あくまでも原則になります。あらゆる機能の原則に従っている訳ではありません。しかし、この原則にしたがい、W3C、WHATWG で HTML5の策定議論が行われています。XHTML2.0のような運命にならない為にも非常に重要な原則です。



□□□□□□□□□□□□□□□□□□□□
(全てはより良い Web の為に！)

原則な！



007 HTML4 と HTML5 の違い

コンテンツモデル

HTML5 では、新たにコンテンツモデルという概念が導入されました。これは HTML4 におけるブロックとインラインという概念に置き換わるものです。

コンテンツモデルとは、**要素をグループ化するもので**、どの要素の中にどんなグループ要素を入れても良いかを規定しています。従来のブロック・インライン要素よりもこの点に関しては細かく分かれており、このコンテンツモデルの規定に従ったマークアップをしないと、規定準拠とならないので、マークアップの際には十分に気をつけなければいけない概念となります。

008 コンテンツモデル

コンテンツの種類

HTML の種類は0個以上のコンテンツカテゴリーに属します。HTML5 の仕様では大まかに次のカテゴリーが使われます。

メタデータ・コンテンツ (Metadata content)

フレーズング・コンテンツ (Phrasing content)

フロー・コンテンツ (Flow content)

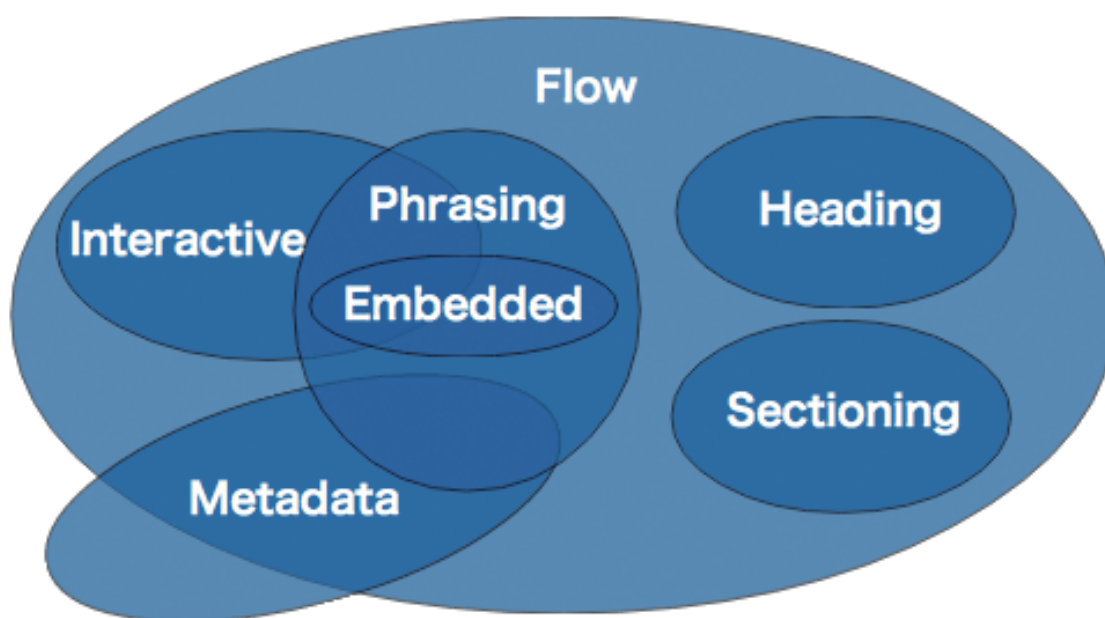
エンベディッド・コンテンツ (Embedded content)

セクショニング・コンテンツ (Sectioning content)

インタラクティブ・コンテンツ (Interactive content)

ヘディング・コンテンツ (Heading content)

各コンテンツの関係図



※コンテンツモデルは HTML5.JP 引用



メタデータ・コンテンツ (Metadata content)

メタデータ・コンテンツとは、以降のコンテンツの表示や動作をセットしたり、自身のドキュメントと他のドキュメントとの関連性をセットしたり、他の " 帯域外 " の情報を伝達するコンテンツです。

要素

base command link meta noscript script style title



フロー・コンテンツ (Flow content)

ドキュメントやアプリケーションの本体に使われるほとんどの要素はフロー・コンテンツに分類されます。

要素

a abbr address area (map 要素の子孫の場合) article aside audio b bdi bdo blockquote br button canvas cite code command datalist del details dfn div dl em embed fieldset figure footer form h1 h2 h3 h4 h5 h6 header hgroup hr i iframe img input ins kbd keygen label map mark math menu meter nav noscript object ol output p pre progress q ruby s samp script section select small span strong style (scoped 属性が存在している場合) sub sup svg table textarea time u ul var video wbr テキスト



セクショニング・コンテンツ (Sectioning content)

セクショニング・コンテンツは、見出しやフッターの範囲を定義するコンテンツです。

要素

article aside nav section



ヘディング・コンテンツ (Heading content)

ヘディング・コンテンツは、セクションのヘッダーを定義します (明示的にセクショニング・コンテンツ要素を使ってマークアップしようが、ヘディング・コンテンツ自身の存在によって示そうが、どちらでも構いません。)。

要素

h1 h2 h3 h4 h5 h6 hgroup



インタラクティブ・コンテンツ (Interactive content)

インタラクティブ・コンテンツは、ユーザー・インタラクションに特化したコンテンツのことです。

要素

a audio (controls 属性が存在している場合) button details embed iframe img (usemap 属性が存在している場合) input (type 属性が hidden 状態でない場合) keygen label menu (type 属性が toolbar 状態の場合) object (usemap 属性が存在している場合) select textarea video (controls 属性が存在している場合)



フレージング・コンテンツ (Phrasing content)

フレージング・コンテンツはドキュメントのテキストのことです。段落内のレベルでテキストをマークアップする要素も同様です。段落からフレージング・コンテンツが続きます。

要素

a (フレージング・コンテンツのみが入れられている場合) abbr area (map 要素の子孫の場合) audio
b bdi bdo br button canvas cite code
command datalist del (フレージング・コンテンツのみが入れられている場合) dfn em embed
i iframe img input ins (フレージング・コンテンツのみが入れられている場合) kbd keygen label
map (フレージング・コンテンツのみが入れられている場合) mark math meter noscript object
output progress q ruby s samp script select small span strong sub sup svg textarea
time u var video wbr テキスト



エンベディッド・コンテンツ (Embedded content)

エンベディッド・コンテンツとは、ドキュメントに他のリソースを組み込むコンテンツ、もしくは、ドキュメントに挿入される他の語彙のコンテンツのことです。

要素

audio canvas embed iframe img math object svg video

new 後から追加



パルパブル・コンテンツ (Palpable content)

はじめの策定時にはおらず、後に追加されたコンテンツモデル。直訳すると明確な内容のある要素。
この要素に分類されたコンテンツには内容がある要素にしなければならない。

要素

a abbr address article aside audio (controls 属性が存在する場合) b bdi bdo blockquote
button canvas cite code data dfn div dl (要素の子が少なくともひとつの name-value グループを含む場合) em embed fieldset figure footer form h1 h2 h3 h4 h5 h6 header i iframe img
input (type 属性が hidden 状態にない場合) ins kbd keygen label main map mark math meter
nav object ol (要素の子が少なくともひとつの li 要素を含む場合) output p pre progress q ruby s
samp section select small span strong sub sup svg table textarea time u ul (要素の子が少なくともひとつの li 要素を含む場合) var video 要素間ホワイトスペースでないテキスト

引用：

HTML5 JP コンテンツモデル

<http://www.html5.jp/tag/models/index.html>

参考：

W3C HTML5 仕様書

<https://www.w3.org/TR/html5/dom.html>