

---

# HTML4からHTML5への変更点

---

1. [HTML4との違い](#)
2. [構文](#)
3. [言語](#)
4. [API](#)

---

## HTML4との違い

2009/8/26

- [HTML 5 differences from HTML 4](#)

HTML5は、WWWのコア言語の5回目の改定になります。2009年8月現在、まだ草案段階で、仕様は確定されていません。

---

## 後方互換性

HTML5は、後方互換性がありますが、他の要素やCSSで代替するほうがよいとされる要素・属性はHTML5で廃止されました。制作者は、font要素、center要素、isindex要素、plaintext要素などの廃止要素は使用できませんが、ユーザーエージェントは、これらの要素が以前同様に機能するようサポートする必要があります。

## 開発モデル

---

HTML5の仕様は、HTMLとXMLという2つの仕様がデザイナーと開発者で実装可能に使えるようになることがゴールです。

## Webアーキテクチャへの影響

---

HTML5で定義された以下の領域と機能が、Webアーキテクチャに影響を与えられています。

- 言語を定めるための基礎としてのDOMの使用
- ブラウジングコンテキストの概念
- ユーザーエージェント条件とオーサリング条件の区別
- 実装においてブラックボックス同等の要件がある抽象的な定義よりはむしろ命令的な定義の使用
- HTML4のブロックおよびインラインの概念を置換する新しいコンテンツモデルの概念
- alt要素のようなアドオンの代わりに、irrelevant要素、progress要素などの新機能のビルトイン概念としてアクセシビリティに焦点をあてる
- 詳細にセマンティックを定義することに焦点をあてた（HTMLの曖昧なセマンティックを置換するアウトライン・アルゴリズム）
- サーバー送信イベント機能（event-source要素）
- datagrid要素
- menu要素とcommand要素
- 起源概念
- オフラインWebアプリケーションキャッシュ
- ブラウジングコンテキストアルゴリズムの定義と関連するセッションの履歴横断アルゴリズム
- コンテンツタイプと文字エンコーディングの特定
- パーサーの明示的な定義
- 2つの構造化されたストレージ機能
- コンテンツ編集機能とUndoManager機能
- ドラッグ&ドロップ、コピー、貼り付けアーキテクチャ
- クロスドキュメントメッセージング機能（postMessage API）
- インラインフレームのための新たなサンドボックス機能

## 構文

---

HTML5では、Web上で公開されているHTML4とXHTML1と互換性のあるカスタムHTML構文を採用しています。しかし、この構文はHTML4における<em/content/>といった、難解なSGML機能とは互換性がありません。カスタムHTML構文を使用する文書は「text/html」 MIMEタイプで提供する必要があります。

HTML5は、ポピュラーな実装と互換性のある構文に対し、詳細な構文解析規定（エラー処理含む）を定義します。ユーザーエージェントは「text/html」 MIMEタイプを持っているリソースに対して、その規定に従います。

## HTML構文

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>文書タイトル</title>
  </head>
  <body>
    <p>段落</p>
  </body>
</html>
```

HTML5に使用可能なもう1つの構文はXMLです。この構文は、XHTML1文書の実装と互換性があります。この構文を使用する文書は、XML MIMEタイプが提供される必要があります、要素はXML仕様にしたがって、「http://www.w3.org/1999/xhtml」 名前空間におかれる必要があります。

下記の例のようにXML文書には「application/xhtml+xml」または「application/xml」といったXML MIMEタイプを指定します。

## XML構文

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

## 文字エンコーディング

---

HTML5における文書の文字エンコーディングの指定方法は3つあります。

1. 例えば、HTTP Content-Typeヘッダなど、転送レベルで指定。
2. ファイルの先頭に記述するUnicode文字にBOMを指定することでエンコーディングを特定させる。
3. head要素の最初の子要素としてmeta要素のcharset属性に指定。  
    <meta charset="utf-8"> とすると、UTF-8エンコーディングが適用されます。  
    これは、<meta http-equiv="Content-Type" content="text/html"; charset=utf-8"> の代わりです。

XML構文の場合は、XML仕様に基づいて文字エンコーディングを指定する必要があります

## DOCTYPE宣言

---

HTML5のHTML構文の場合、ページを標準モードで表示させるにはDOCTYPE宣言を必ず記述する必要があります。XML MIMEタイプの文書は、常に標準モードとして扱われるため、DOCTYPE宣言の記述は省略してもかまいません。

旧バージョンのDOCTYPE宣言は、HTML言語がSGMベースだったためDTDの参照が必要でしたが、HTML5では不要になりました。

HTML5のDOCTYPE宣言は、<!DOCTYPE html> とシンプルに記述できます（大文字小文字の区別なし）。

## その他

---

その他、いくつかの構文に変更があります。

- 現在、HTMLはIRIsのネイティブサポートがあります。HTML5では単にエラー処理として扱われていましたが、文書の文字エンコーディングがUTF-8またはUTF-16の時のみ、IRIを利用することができるようになりました。
- lang属性は、XMLのxml:lang同様、有効な言語識別子だけでなく、空文字列も指定できるようになりました。

## 言語

---

2009/8/26

## 新要素

要素名	説明
section要素	一般的な文書またはアプリケーションのセクション。文書構造を表すのにh1～h6要素と併用する。
article要素	ブログのエントリーや新聞記事など、文書内の独立した一片の内容を表す。
aside要素	ページの主題と関連の薄いコンテンツ部分を表す。
header要素	セクションのヘッダーを表す。
footer要素	セクションのフッターを表す。フッターには作者情報、著作権情報などを含むことができる。
nav要素	ナビゲーションとなる文書のセクションを表す。
dialog要素	下記のような会話文をマークアップするために使用する。  <pre>&lt;dialog&gt; &lt;dt&gt;Aさん&lt;/dt&gt; &lt;dd&gt;Bさん、元気？&lt;/dd&gt; &lt;dt&gt;Bさん&lt;/dt&gt; &lt;dd&gt;元気よ！Aさんは？&lt;/dd&gt; &lt;dt&gt;Aさん&lt;/dt&gt; &lt;dd&gt;相変わらずよ。&lt;/dd&gt; &lt;/dialog&gt;</pre>
figure要素	グラフィックやビデオなどのコンテンツの埋め込みに、表題を付けるのに使用する。  <pre>&lt;figure&gt; &lt;video src="movie"&gt;&lt;/video&gt; &lt;legend&gt;ビデオの説明です。&lt;/legend&gt; &lt;/figure&gt;</pre>
audio要素 video要素	マルチメディアコンテンツを表す要素。いずれの要素も、制作者がスクリプトを利用して独自のユーザーインターフェースを作成可能なAPIを提供する。しかし、ユーザーエージェントが自動的にインターフェースを追加するような仕組みも導入される。異なるタイプの多様なストリームが利用可能な場合、source要素をこれらの要素と併用する。
embed要素	プラグインを利用するコンテンツに使用する。
mark要素	マークが付けられたテキストを表す。
progress要素	ダウンロードや負荷の高い処理など、実行中のタスクを表します。
meter要素	ディスク容量などの測定値を表す。

time要素	日付や時刻、またはその両方表す。
ruby要素 rt要素 rp要素	ルビ（振り仮名）をマークアップする。
canvas要素	グラフやゲームなど、動的にレンダリングされるビットマップグラフィックスに使用する。
command要素	ユーザーが呼び出し可能なコマンドを表す。
datagrid要素	インタラクティブなツリーリスト、または表データを表す。
details要素	ユーザーが要求に応じて取得可能な追加情報やコントロールを表す。
datalist要素	<p>input要素に新たに追加されたlist属性と併用することで、コンボボックスを作成可能。</p> <pre> &lt;input list="browsers"&gt; &lt;datalist id="browsers"&gt; &lt;option value="Firefox"&gt; &lt;option value="Interenet Explorer"&gt; &lt;option value="Opera"&gt; &lt;option value="Safari"&gt; &lt;/datalist&gt; </pre>
keygen要素	キーのペアを生成するためのコントロールを表す。
output要素	スクリプトによる計算結果など、あらゆるタイプの出力結果を表す。
inputのtype属性に新しく追加された値	<p>下記の属性値は、ユーザーエージェントがインターフェースを提供し、決められたフォーマットでサーバーにデータを送信できるように定義されました。カレンダーの日付選択インターフェースや、ユーザーのアドレスブックと統合した機能に役立ちます。ユーザーがサーバーにデータを送信する前にフォームの内容をチェックできるため、フィードバックを待つ時間が短縮され、ユーザーエクスペリエンスが向上します。</p> <ul style="list-style-type: none"> <li>• tel</li> <li>• search</li> <li>• url</li> <li>• email</li> <li>• datetime</li> <li>• date</li> <li>• montd</li> <li>• week</li> <li>• time</li> <li>• datetime-local</li> <li>• number</li> <li>• range</li> </ul>

- color

これらの新しいタイプのアイデアは、ユーザーエージェントが、例えばカレンダーの日付ピッカーやユーザーのアドレス帳を統合するといったユーザーインタフェースを提供することが可能となり、定義済みのフォーマットをサーバーに送信できるというもの。

## 新属性

対象要素	属性名	説明
a要素 area要素	media属性	link要素との一貫性を保持するための付加的なもの。
a要素 area要素	ping属性	属性値はスペース区切りのURLで、リンクをたどる時にpingされなければならないリソースを指定する。現在、ユーザートラッキングはほとんどダイレクト処理を通じて行われている。この属性は、ユーザーエージェントがどのURLがpingされるかをユーザーに通知することができます。また、プライバシーを意識しているユーザーに対し、その機能を無効にする方法をを提供することも可能。
area要素	hreflang属性 rel属性	a要素とlink要素との一貫性を保持するため。
base要素	target属性	a要素との一貫性を保持するため。iframeなどWebアプリケーションにおいて役立つため、a要素とarea要素のtarget属性は非推奨ではなくなりました。
li要素	value属性	非推奨でなくなりました。
ol要素	start属性	非推奨でなくなりました。
meta要素	charset属性	文書の文字エンコーディングを指定する。
input要素（type="hidden"除く） select要素	autofocus属性	ページを読み込んでいる間にフォームコントロールにフォーカスを当てる。

textarea要素 button要素		
input要素 textarea要素	placeholder属性	ページを読み込んでいる間に入力のヒントとなる入力支援テキストを指定する。
input要素 output要素 select要素 textarea要素 button要素 button要素 fieldset要素	form属性	ページがロードされた時に特定のフォームコントロールにフォーカスを与える仕組みを提供します。ユーザーが自動フォーカス機能をオフにすることも含め、この属性はユーザーエクスペリエンスを向上させる。
input要素 output要素 select要素 textarea要素 button要素 fieldset要素	form属性	コントロールが関連するフォームを指定する。これにより、これらの要素をform要素の中に書かなくてもよく、ページ内の好きなところにこれらの要素を配置可能。
input要素 textarea要素	required属性	ユーザーはどのフォームに記入することで、フォームを送信することができます。ただし、type属性にhidden、image、submitなどのボタントypeが指定されているinput要素のには指定不可。
fieldset要素	disabled属性	指定するとすべてのコンテンツを無効化する。
input要素	autocomplete属性 min属性 max属性 multiple属性 pattern属性 step属性	制約を指定する為の属性。datalist要素やselect要素と併用するlist属性がある、
form要素	novalidate属性	フォーム確認を無効化するのに使用。フォームを常に送信可能にすることも可能。
input要素 button要素	formaction属性 formenctype属性 formmethod属性 formnovalidate属性 formtarget属性	これらの属性を指定すると、form要素のaction、enctype、method、novalidate、target属性は上書きされる。
menu要素	type属性 label属性	グローバル属性contextmenuを併用してコンテキストメニューの提供す



		るといった、要素を典型的なユーザーインターフェースで見られるようなメニューに変形させることが可能。
style要素	scoped属性	スコープしたスタイルシートの指定が可能。このようなstyle要素のスタイル規則は、ローカルツリーにのみ適用されます。
script要素	async属性	スクリプトのローディングおよび実行に影響与える。
html要素	manifest属性	オフラインWebアプリケーションのためのAPIと連携して使われるアプリケーションキャッシュを指定。
link要素	sizes属性	参照するアイコンのサイズを示すために、アイコンの関係（rel属性によって設定）と共に使用可能。
ol要素	reversed属性	リストが降順であることを示す。
iframe要素	seamless属性 sandbox属性	ブログのコメントなど、コンテンツをサンドボックス化するのに使用。

## グローバル属性

HTML4で定義されていたいくつかの属性はすべての要素に適用されるようになりました。。これらはグローバル属性と呼ばれ、class、dir、id、lang、style、tabindex、title属性が該当します。

属性名	説明
contenteditable属性	要素が編集可能領域であることを示します。ユーザーは内容を変更し、マークアップを操作可能。
contextmenu属性	制作者によって提供されるコンテキストメニューを示すために使用可。
draggable属性	新しく定義されたドラッグ&ドロップAPIと組み合わせて使用可。
irrelevant属性	要素がまだないか存在、関連しないことを示す。
ref属性 registrationmark属性 template属性	これらのグローバル属性はデータ・テンプレート機能を支援する。

data-*属性	制作者が定義できる属性。将来のHTMLのバージョンとの衝突を避けるために、属性にdata-というプレフィックスをつければ、それ以外は好きに属性を定義できる。これらの属性の唯一の要件は、それらがユーザーエージェントの拡張に使用されないということです。
----------	--

下記は反復モデルのための属性です。これらはグローバル属性であり、すべてのHTML要素上や、「http://www.w3.org/1999/xhtml」名前空間にある属性と共に他のいかなる名前空間におけるどんな要素上でもそのように使われるでしょう。

- repeat
- repeat-start
- repeat-min
- repeaet-max

HTML5は、HTML4からのonevent-nameという形式のすべてのイベントハンドラ属性をグローバル属性に変更しました。そして、新しいevent-source要素やクロスドキュメントメッセージAPIと共に用いるonmessage属性といった、それが定義する新しいイベントのために、いくつかの新しいイベントハンドラ属性を追加しました。

## 変更された要素

これらの要素は、Web上でもっと便利に使えるように、すこし意味合いが変更されました。

要素名	説明
href属性のないa要素	プレースホルダーリンクに相当。a要素には、a要素内にブロック要素も含むことが可能。
address要素	セクショニングする新しい概念によってスコープされた。
b要素	テキストに重要性をつけることなしに、通常のテキストとはスタイル上異なる文書の要約、レビューにおける製品名など、典型的な印刷上の体裁が太字にされるテキストに使用。
hr要素	段落レベルにおける主題の変化を表す。
i要素	通常のテキストとは異なる声やムードを表す。例えば、部類名称、専門用語、別の言語からの慣用句、考え、船の名前など、典型的な印刷上の体裁がイタリック体で印刷されるテキストに使用。使用法は、言語によって異なります。
label要素	ブラウザがラベルからフォーカスを移動させるべきでない要素。基本的なプラットフォームユーザーインターフェースで標準となっている振る舞いはこの限りではない。

menu要素	ツ実際のメニューに役立つように再定義された。
small要素	注釈などの小さい文字部分を表す。
strong要素	強い強調ではなく重要性を表す。
q要素	q要素のための引用符は、ユーザーエージェントではなく制作者によって提供される。

## 廃止要素

ユーザーエージェントは下記の要素をサポートする必要がありますが、制作者はそれらを使用することができませんのでCSSで代替してください。 isindex要素は、パーサーによって既にサポートされています。

- basefont
- big
- center
- font
- s
- strike
- tt
- u

下記の要素は、エンドユーザーに対するユーザービリティやアクセシビリティによくない影響を及ぼすため、HTML5で廃止されました。

- frame
- frameset
- noframes

下記の要素は、あまり使用されておらず、また混乱を起こすため、他の要素で代替可能なことから、HTML5で廃止されました。

要素名	説明
acronym要素	多くの混乱を起こしたため削除されました。略語にはabbr要素を使用すること。
applet要素	object要素が支持されたため廃止されました。
isindex要素	フォームコントロールで代替可能。
dir要素	ul要素が支持されたため廃止されました。

noscript要素は、単にHTML構文に従っているだけです。その使用がHTMLパーサーに依存するため、XML構文には含まれていません。

## 廃止属性

HTML4にあったいくつかの属性はHTML5で廃止されました。それらが互換性の理由で、ユーザーエージェントに何らかの影響を及ぼすなら、それらがどのようにそれらのシナリオで動作しなければならないか定義されます。

対象要素	廃止された属性
caption要素	align属性
a要素	accesskey属性 rev属性 charset属性 shape属性 coords属性 name属性（id属性で代替する）
area要素	accesskey属性 nofref属性
button要素	accesskey属性
input要素	accesskey属性 align属性（CSSで代替）
label要素	accesskey属性
legend要素	accesskey属性
textarea要素	accesskey属性
link要素	accesskey属性 rev属性 charset属性
img要素	longdesc属性 name属性（id属性で代替する） align属性（CSSで代替） border属性（CSSで代替） hspace属性（CSSで代替） vspace属性（CSSで代替）
iframe要素	longdesc属性 caption属性

	frameborder属性（CSSで代替） marginheight属性（CSSで代替） marginwidth属性（CSSで代替） scrolling属性（CSSで代替）
link要素	target属性
head要素	profile属性
html要素	version属性
body要素	alink属性（CSSで代替） link属性（CSSで代替） text属性（CSSで代替） vlink属性（CSSで代替） background属性（CSSで代替） bgcolor属性（CSSで代替）
meta要素	scheme属性
object要素	archive属性 classid属性 codebase属性 codetype属性 declare属性 standby属性 align属性（CSSで代替） border属性（CSSで代替）
param要素	valuetype属性 type属性
script要素	language属性
legend要素	align属性（CSSで代替）
hr要素	align属性（CSSで代替） noshade属性（CSSで代替） size属性（CSSで代替） width属性（CSSで代替）
h1～h6要素	align属性（CSSで代替）
div要素	align属性（CSSで代替）
p要素	align属性（CSSで代替）
table要素	frame属性（CSSで代替） summary属性 align属性（CSSで代替） bgcolor属性（CSSで代替） border属性（CSSで代替）

	cellpadding属性（CSSで代替） cellspacing属性（CSSで代替） rules属性（CSSで代替） width属性（CSSで代替）
tr要素	align属性（CSSで代替） bgcolor属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） valign属性（CSSで代替） width属性（CSSで代替）
th要素	axis属性 abbr属性 align属性（CSSで代替） bgcolor属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） height属性（CSSで代替） nowrap属性（CSSで代替） valign属性（CSSで代替） width属性（CSSで代替）
td要素	axis属性 abbr属性 scope属性 align属性（CSSで代替） bgcolor属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） height属性（CSSで代替） nowrap属性（CSSで代替） valign属性（CSSで代替） width属性（CSSで代替）
tbody要素	align属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） valign属性（CSSで代替）
tfoot要素	align属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） valign属性（CSSで代替）
col要素	align属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） valign属性（CSSで代替） width属性（CSSで代替）

colgroup要素	align属性（CSSで代替） char属性（CSSで代替） charoff属性（CSSで代替） valign属性（CSSで代替） width属性（CSSで代替）
br要素	clear属性（CSSで代替）
dl要素	compact属性（CSSで代替）
menu要素	compact属性（CSSで代替）
ol要素	compact属性（CSSで代替） type属性（CSSで代替）
ul要素	compact属性（CSSで代替） type属性（CSSで代替）
li要素	type属性（CSSで代替）
pre要素	width属性（CSSで代替）

## API

---

2009/8/26

HTML5では、Webアプリケーションの作成を支援する多くのAPIが導入されました。これらのAPIは、アプリケーションのために導入された新しい要素と共に使用することができます。

- canvas要素と共に使用可能な2D描画API
- video要素とaudio要素で利用可能なビデオやオーディオを再生するAPI
- 永続的なストレージ、キー/値およびSQLデータベースの両方をサポート
- オフラインWebアプリケーションを有効化するAPI
- Webアプリケーションが特定のプロトコルやMIMEタイプのために、それ自身を登録可能にするAPI
- draggable属性と組み合わせるドラッグ&ドロップAPI
- ネットワークAPI
- 「戻る」ボタンを壊すのを防ぐために、履歴を開いてページがそれに加えるのを許可するAPI
- クロスドキュメントメッセージング
- 新しいevent-source要素と組み合わせるサーバー送信イベント
- メールの新着メールメッセージのような、新着情報をユーザーに警告するAPI

## HTMLDocumentに対する拡張

HTML5では、DOM Level 2 HTMLのHTMLDocumentインターフェースを拡張しました。そのインターフェースはDocumentインターフェースをインプリメントするすべてのオブジェクト上でインプリメントされます。それは複合文書前においても意味があります。さらにそれは、新しいメンバーを持っています。

getElementsByClassName()	class名から要素を選択します。このメソッドは、class属性を持ついかなる要素、SVGやMatdMLなどのDocumentオブジェクトで動作するように定義されました。
innerHTML	HTMLやXML文書をパースして解析することを容易にする属性。以前、この属性は標準的なものではなく、ブラウザのHTMLElementで利用可能だけでした。
activeElement hasFocus	どの要素に現在のフォーカスが当たっているか、Documentがそれぞれフォーカスを行っているかを決定する。
getSelection()	現在の選択を表すオブジェクトを返す。
designMode execCommand()	主に文書を編集するために使用。

## HTMLElementに対する拡張

HTMLElementインタフェースは、HTML5でさらに拡張しました。

getElementsByClassName()	基本的にHTMLDocumentにあるものよりよりスコープされたバージョン。
innerHTML	ブラウザだけでなく、XML文書内で使用される場合に、XMLコンテキストでも動作するように定義されています。
classList	classNameに対する便利なアクセサー。このアクセサーが返すオブジェクトは、has()、add()、remove()、toggle()といった要素のクラスを操作するメソッドを表します。
relList	a要素、area要素、link要素が持つrel属性に対して同じ機能を提供。