

HTML5 スタンダード・デザインガイド

Webサイト制作者のためのビジュアル・リファレンス&
セマンティクスによるコンテンツデザインガイド

エ・ビスコム・テック・ラボ [著]

もくじ

Chapter 1

HTML5 とインフォメーションデザイン

誰のための Web ページか

013

これまでの HTML.....	014
HTML5 の登場とセマンティクス.....	015
構造化データの活用.....	018
ブラウザによる記事の抽出.....	018
SNS によるページ情報の取得.....	019
検索エンジンによるリッチスニペットの表示.....	019
エージェントの登場.....	020
セマンティック・マークアップの実践.....	023
マークアップを見るときのポイント.....	023
Apple.....	024
メインコンテンツのマークアップとアウトライン.....	026
アウトラインにおけるルート階層の見出し.....	027
ナビゲーションのマークアップ.....	029
補足・関連情報のマークアップ.....	029

Mozilla.....	030
メインコンテンツのマークアップとアウトライン.....	034
ダウンロードボタンのマークアップ.....	035
ナビゲーションのマークアップ.....	036
セマンティック・マークアップを行うときに注意したいこと.....	037

HTML5 のコンテンツ・モデル.....	038
特別なカテゴリ.....	039
HTML5 の文法とセマンティクス.....	041
HTML5 で新しくできるようになったこと.....	042
HTML5 の自由度とセマンティクス.....	044

Chapter 2

Web ページの基本

Web ページを構成する最も基本的な要素

045

DOCTYPE 宣言.....	046
ルート要素.....	047
html.....	
HTML の名前空間.....	048
メタデータの設定.....	050
head.....	
コンテンツ.....	051
body.....	

スタイルシート.....	052
style.....	
スタイルシートの設定をコンテンツの中に記述.....	053
メディアクエリの指定.....	054
代替スタイルシートの指定.....	056
スクリプト.....	057
script / noscript.....	
スクリプトの読み込みと実行に関する設定.....	058
スクリプトについての説明を記述する場合.....	059
スクリプトが動作しない環境用のコンテンツを記述する場合.....	059

セクションとアウトライン	062	見出し	080
セクションの明示	062	h1 / h2 / h3 / h4 / h5 / h6	
アウトラインの作成	064	セクション内に見出しを1つだけ記述した場合	080
独立したコンテンツ	066	セクション内に複数の見出しを記述した場合	082
article		見出しに適用されるデフォルトスタイルシート	084
セクション	071	見出しのグループ化	086
section		hgroup	
ナビゲーション	074	ヘッダーとフッター	087
nav		header / footer	
補足・関連情報	076	汎用要素	089
aside		div / span	

ページタイトル	092	http-equiv 属性で指定できるメタデータ	106
title		言語の種類	106
リンクに関する設定	093	エンコードの種類	107
base		優先代替スタイルシート	107
ベース URL を指定する	093	リフレッシュ	108
ブラウジング・コンテキストを指定する	094	クッキー	108
関連 URL	095	コンテンツの内容をより詳細に明示する	109
link		利用できる規格	109
代替文書の URL	096	バリデータ	110
フィードの URL	096	microdata (マイクロデータ)	111
外部スタイルシート	097	microdata とは	111
アイコン	099	microdata で利用できるボキャブラリー	111
先読みキャッシュ	100	microdata を利用した情報の付加	112
検索	100	アイテムタイプ	113
メタデータ	102	プロパティ名	114
meta		Expected Type (予期される形式)	115
エンコードの種類	102	RDFa	121
name 属性で指定できるメタデータ	102	RDFa とは	121
Web アプリケーション名	103	RDFa で利用できるボキャブラリー	121
ページの著者名	103	RDFa を利用した情報の付加	122
ページの概要	103	microformats (マイクロフォーマット)	126
ページを生成したソフトウェア	103	microformats とは	126
ページのキーワード	103	microformats で利用できるボキャブラリー	126
		microformats を利用した情報の付加	126

Chapter 5

外部コンテンツ

HTML5 で扱うことができる画像・動画・各種コンテンツ

129

画像.....	130	テキストトラック.....	149
img		track	
インラインフレーム.....	134	イメージマップ.....	152
iframe		map / area	
サンドボックス化.....	135	選択できないエリアを構成する.....	153
srodoc 属性でコンテンツを記述する.....	137		
シームレス表示.....	138	Canvas.....	154
外部コンテンツ.....	140	canvas	
object / param		SVG.....	156
プラグインのパラメータ.....	142	svg	
下位互換用のコンテンツ.....	142	MathML.....	157
外部コンテンツの組み込みポイント.....	143	math	
embed			
ビデオ / オーディオ.....	144		
video / audio / source			
複数のコンテンツの指定.....	145		
指定できる属性.....	147		
ポスターフレーム画像の指定.....	147		
プリロードの方法を指定.....	148		
メディアグループの設定.....	148		

Chapter 6

段落書式

明確に意味が定義された段落書式

159

段落.....	160	リスト.....	169
p		ol / ul / li	
段落レベルの区切り.....	161	割り振られる番号.....	170
hr		関連リスト.....	172
連絡先・問い合わせ先.....	162	dl / dt / dd	
address		語句の定義を記述する場合.....	173
引用.....	163	図版とキャプション.....	174
blockquote / q		figure / figcaption	
引用元の情報.....	164	参照している図版を示す場合.....	175
整形済みテキスト.....	166	グループキャプションをつける場合.....	176
pre			
コンピュータ・コードの記述.....	167		
コンピュータの入出力内容の記述.....	167		

リンク	178	重要	198
a		strong	
表示中のページへのリンクを無効にする.....	179	強調・強勢	199
ページ内リンクを設定する.....	179	em	
ブラウジング・コンテキストを指定する.....	180	変更・更新した情報	200
リンク先に関する補助的な情報を記述する.....	181	s	
リンクタイプを明示する.....	181	語句の区別	201
フィールドまたは代替文書へのリンク.....	182	i / b / u	
著者情報へのリンク.....	182	学名や慣用語などの区別.....	201
コンテンツのパーマリンク.....	182	注目してほしい語句の区別.....	202
ヘルプ情報へのリンク.....	183	不明瞭な語句の区別.....	202
ライセンス情報へのリンク.....	184	ハイライト	204
前後のページへのリンク.....	184	mark	
推奨していないリンク.....	184	参照先の注目箇所を示す.....	204
リファラー情報の送信を禁止したリンク.....	185	引用文内の注目箇所を示す.....	204
先読みキャッシュを指定したリンク.....	185	コンピュータ・コード内の注目箇所を示す.....	205
検索関連の情報へのリンク.....	185	ユーザーが実行した処理に関連する箇所を示す.....	205
ページに付加したタグのリンク.....	185	ルビ (ふりがな)	206
略語	187	ruby / rt / rp	
abbr		説明用のルビ.....	208
定義	188	2つのルビ.....	209
dfn		未対応ブラウザ用の括弧の記述.....	210
作品のタイトル	189	双方向アルゴリズムの変更	212
cite		bdo	
日時	190	双方向アルゴリズムの隔離	213
time		bdi	
日時の記述形式.....	191	追加・削除	214
コンピュータ関連	193	ins / del	
code / var / samp / kbd		改行	216
コンピュータ・コード.....	193	br	
変数.....	194	改行を認める箇所	217
コンピュータからの出力内容.....	194	wbr	
コンピュータへの入力内容.....	195		
同時に入力するキー.....	195	上付き・下付き	196
選択するメニューの項目.....	195	sup / sub	
コマンドラインの表示.....	195	但し書き・注意	197
上付き・下付き	196	small	
sup / sub			
但し書き・注意	197		
small			

Chapter 8

テーブル 表組コンテンツの表示

219

テーブル	220	テーブルのキャプション	226
table / tr / th / td		caption	
セルの結合	222	列のグループ	229
複数の行にまたがったセル	222	colgroup / col	
複数の列にまたがったセル	222	行のグループ	231
代替の見出し	223	thead / tbody / tfoot	
見出しとデータの関係を示す	224		
見出し側で示す場合	224		
データ側で示す場合	224		

Chapter 9

フォーム 強化されたコントロール

235

フォーム	236	パスワード	252
form		日時	253
送信されるデータ	237	年月日の入力	253
オートコンプリートを無効にする	238	年月の入力	254
入力データの検証を無効にする	239	週の入力	254
<form> の外にフォームコントロールを用意する	239	時刻の入力	254
		日時の入力 (タイムゾーンの情報なし)	255
		日時の入力 (タイムゾーンの情報あり)	255
		日時関連のコントロールで指定できる属性	256
ラベル	241	数値	257
label		大まかな数値	257
フォームコントロールのグループ	242	カラーピッカー	258
fieldset / legend		チェックボックス	258
グループを無効にする	243	初期状態を「オン」にする	259
データの入力・送信	244	ラジオボタン	260
input		ファイルの選択とアップロード	260
コントロールの無効化	245	ファイルの種類を指定する	262
初期値の指定	245	複数のファイルを選択する	263
オートフォーカス	245	隠し情報	265
テキスト	246	エンコードの種類を送信する	265
書字方向の情報を送信	246	送信ボタン	266
入力できる文字数の指定	247	フォームの設定を変更する	266
データ形式の指定	247	画像ボタン	267
ブレースホルダの指定	248	リセットボタン	268
表示専用にする	248	ボタン	269
入力を必須にする	248		
表示する文字数の指定	249	ボタン	270
検索	250	button	
URL	250	指定できる属性	271
電話番号	251		
メールアドレス	251		
複数のメールアドレスを入力する	252		

選択肢	272	暗号鍵	283
select / option / optgroup		keygen	
画面に表示するラベルの指定.....	273	計算結果	284
サーバーに送信する値の指定.....	273	output	
初期状態を選択済みにする.....	274	進捗状況	285
ブレースホルダの表示.....	274	progress	
複数の選択を可能にする.....	275	メーター	286
選択肢のグループを示す.....	276	meter	
入力候補	277	最適な範囲の明示.....	287
datalist			
スライダーで入力候補を示す.....	278		
色のコントロールで入力候補を示す.....	279		
下位互換用の記述.....	279		
複数行のテキスト	280		
textarea			
初期値を指定する.....	281		
折り返しの改行の送信.....	282		

Chapter 10

インタラクティブ

ユーザーアクションを求めるインターフェースの設定

289

付加的な情報	290	ダイアログボックス	295
details / summary		dialog	
ウィジェットを開いた状態で表示する.....	291		
コマンドリスト	292		
menu / command			

APPENDIX

HTML5 の構文規則	296	要素一覧&索引	
タグの記述形式.....	296	要素&ブラウザ対応一覧	306
属性の記述形式.....	297	逆引き索引	310
開始タグと終了タグの省略.....	297	HTML4.01/XHTML1.0 からの変更点.....	310
文字参照.....	297	HTML5.1/HTML Living Standard における追加・変更点.....	310
HTML5 の文法チェック	298	各種ツール/サービス.....	310
HTML5 のバリデータ	298	参照規格/ドキュメント.....	311
未対応ブラウザへの対応	299	索引	312
html5shiv の利用	299	要素.....	312
グローバル属性	300	属性.....	313
HTML5 のグローバル属性.....	300	属性値.....	314
		キーワード.....	315

本書の使い方

- ・本書では W3C の HTML5（勧告候補）に基づき、各要素の定義や機能、使い方について以下の形でまとめています。
- ・本書に掲載した内容は 2013 年 1 月 10 日現在のものです。
- ・HTML5 は策定中の規格であり、機能や使い方、ブラウザの対応状況などは変わる可能性がありますのでご了承ください。

HTML5で新しく追加された機能には **HTML5** というマークをつけて示しています。

ブラウザの対応状況。各アイコンについてはP.012を参照してください。



要素の機能。

要素の記述形式。

指定できる属性。

※「グローバル」はグローバル属性(P.300)を示しています。

ナビゲーション

`<nav>...</nav>`

属性
グローバル

要素に関する基本情報。

カテゴリ	要素が属するカテゴリ(参照: P.038)
記述可能な場所	要素を記述できる場所
コンテンツ・モデル	要素の中に記述できるコンテンツの種類
構成するボックス	要素が標準で構成するボックスの種類

```

<nav>
  <ul>
    <li><a href="#">HOME</a></li>
    <li><a href="#">COLUMN</a></li>
    <li><a href="#">GIFT</a></li>
    <li><a href="#">CONTACT</a></li>
  </ul>
</nav>
  
```

このスクリーンショットは、ナビゲーションメニューのHTML5実装を示しています。メニュー項目は「HOME」、「COLUMN」、「GIFT」、「CONTACT」です。コードは、`<nav>` タグで囲まれ、`` と `` タグで構成されています。`` というリンクが各項目に使用されています。

```

<nav>
  <ul>
    <li><a href="#">Earl Grey</a></li>
    <li><a href="#">Green Tea</a></li>
    <li><a href="#">Black Tea</a></li>
  </ul>
</nav>
  
```

このスクリーンショットは、「Short Tea Breaks」セクションのHTML5実装を示しています。メニュー項目は「Earl Grey」、「Green Tea」、「Black Tea」です。コードは、`<nav>` タグで囲まれ、`` と `` タグで構成されています。`` というリンクが各項目に使用されています。

サンプル:
要素の活用方法などを具体的に解説します。ソースは以下のように色分けして掲載しています。

- HTMLソース
- スタイルシート

NOTES:
注意したいポイントや各種TIPS、ベンダーサイトにおけるHTML5を利用したマークアップの実例、HTML5.1やHTML Living Standardに関する情報などを掲載しています。

NOTES

DIFFERENCES:
HTML4.01/XHTML1.0からの変更点をまとめています。

DIFFERENCES

HTML5 の規格について

HTML5 の規格は W3C と WHATWG で策定作業が進められており、現在のところ以下の規格がリリースされています。本書では W3C の HTML5 に基づいて各要素の解説を行っていますが、HTML5.1 や HTML Living Standard で採用された要素や属性がある場合は「NOTES」で紹介しています。

HTML5 HTML5

<http://www.w3.org/TR/html5/>

W3C がリリースしている規格です。W3C の規格策定の手順に従って標準化の作業が進められており、2012 年 12 月 17 日に勧告候補がリリースされました。これにより、新しい機能が追加されることはなくなり、安定した規格として実装と検証が進められていくこととなります。なお、最終的な勧告は 2014 年の第 4 四半期にリリースされる予定となっています。

※ HTML5 の勧告候補では一部の要素が「リスクのある機能」としてリストアップされており、最終的な勧告からは除外される可能性があります。詳しくは P.309 を参照してください。

HTML5.1 HTML5.1 HTML5.1 Nightly

<http://www.w3.org/TR/html51/>

<http://dev.w3.org/html5/spec/Overview.html>

HTML5 の次のバージョンとして W3C がリリースしている規格です。現在のところ、草案 (Working Draft) と Nightly (Editor's Draft) がリリースされています。

HTML Living Standard HTML living Standard

<http://whatwg.org/html>

WHATWG がリリースしている規格です。最新の機能や提案をよりスピーディに規格に反映していくことを目指しており、頻繁に更新が行われています。なお、WHATWG (Web Hypertext Application Technology Working Group) は、Apple や Mozilla、Opera によって設立されたグループです。

W3C の規格策定の流れ

Working Draft (草案)

Last Call Working Draft (最終草案)

Candidate Recommendation (勧告候補)

Proposed Recommendation (勧告案)

Recommendation (勧告)

W3C における各規格は「草案」として提案され、審議を繰り返して最終的な「勧告」となります。なお、草案の前段階として「Editor's Draft」が公開されている規格や、よりスピーディに更新していく「Living Standard」や「Living Specification」として公開されている規格もあります。

対応ブラウザについて

本書では、要素ごとに以下のブラウザにおける対応状況を示しています。
また、サンプルの画面キャプチャには基本的に Chrome を使用しています。



iOS5およびAndroid 3.x以前のバージョンの対応状況については、必要に応じて注釈の形で掲載しています。

Internet Explorer 9以前のバージョンの対応状況などについては、必要に応じて「IE NOTES」にまとめています。

IE NOTES

IE8以前のバージョンは <nav> に未対応です。

要素が構成するボックスについて

要素が標準で構成するボックスの種類については、HTML5 で提示されたデフォルトスタイルシートの設定に基づいて掲載していますので、見た目のデザインを設定するときの参考にしてください。構成するボックスの種類を変更したい場合は CSS の display プロパティを利用します。

なお、ボックスの種類については詳しくは、『CSS3 スタンダード・デザインガイド』（エ・ビスコム・テック・ラボ編著／マイナビ刊）の P.046 を参照してください。

サンプルについて

本書に掲載したサンプルデータは下記の URL からダウンロードすることができます。

<http://book.mynavi.jp/support/pc/4532/>

※サンプルは基本的に Internet Explorer 9 以上でレイアウトが崩れないように作成してあります。

※Internet Explorer 8 に対応する方法については P.299 を参照してください。



Chapter 1

HTML5とインフォメーションデザイン

これまでのHTML
HTML5の登場とセマンティクス
構造化データの活用
セマンティック・マークアップの実践
HTML5のコンテンツ・モデル

これまでの HTML

Web ページで情報を発信する WWW (World Wide Web) のシステムは、1989 年に CERN (欧州原子核研究機構) の Tim Berners-Lee によって提案されました。そして、「どのようなコンピュータからでも情報を閲覧し、共有できるようにする」という WWW の大命題を実現するため、Web ページの記述言語として採用されたのが HTML (HyperText Markup Language) です。

当時は、やっと Windows 3.0 や Macintosh Classic が登場した頃で、フルカラー環境は珍しく、テキストブラウザが現役で使用されているような時代でした。そのため、HTML によって「見出し」や「段落」といったページの論理構造を示し、閲覧環境の表示能力に合わせて Web ページを読みやすく表示して、アクセシビリティを確保したのです。

標準規格としては、1995 年に HTML2.0 がリリースされ、その後、HTML3.2、HTML4.0、HTML4.01、XHTML1.0 とバージョンアップしていきます。この間、「論理構造の記述と視覚的なデザインの記述を分類する」という考え方が明確化され、HTML と CSS によるページ制作が浸透しましたが、HTML の基本的な機能は HTML2.0 の時代から大きく変化してきませんでした。

しかし、HTML はもともと論文のようなシンプルな構造の文書をマークアップするために作られたため、Web ページで扱う情報が多様化するに従い、機能が不十分だと言われるようになっていきます。

そこで、新しく登場したのが HTML5 です。

1989	WWWのシステム提案
1990	最初のWebサイトが稼働
1993	HTMLの草案リリース
1995	HTML2.0 リリース
1996	CSS1.0 リリース
1997	HTML3.2 リリース HTML4.0 リリース
1998	CSS2.0 リリース
1999	HTML4.01 リリース
2000	XHTML1.0 リリース
2002	XHTML1.0 (Second Edition) リリース
2005	CSS3 関連の草案リリース
2008	HTML5 草案リリース
2011	CSS2.1 勧告リリース
2012	HTML5 勧告候補リリース

HTML5の登場とセマンティクス

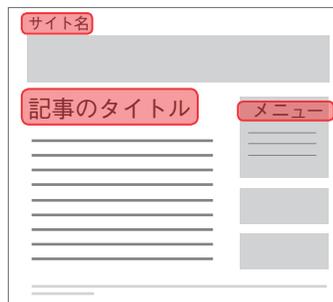
HTML5はこれまでのHTMLに置き換わる、新たに策定された規格です。ただし、拡張によって進化してきた従来のHTMLとは異なり、下位互換性を意識しつつ、Webを取り巻く環境の現在と将来を踏まえて再構築を行ったものです。そして、この再構築の根幹となっているのが「セマンティクス (semantics)」という概念です。

「セマンティクス」は、辞書で引くと「意味」や「意味論」といった訳が出てきて日本語で明確に表すのは難しい言葉なのですが、HTML5においてはマークアップした中身が何であるかを示すことを指し、「意味づけ」とも呼ばれます。そして、セマンティクスの大きな役割は、意味づけによってページの中身を人間以外のものが理解できるようにすることにありません。

これまで、HTMLでは論理構造の記述によってページの中身を人間以外のものに伝えようとしてきました。しかし、たとえば「見出し」という構造を示しても、その中身がサイト名なのか、記事のタイトルなのか、サイドバーのメニューの見出しなのかといった判別はできません。そのため、検索エンジンなどは論理構造から重要と思われる部分を類推し、抜き出して利用してきました。

製作者サイドでは他に手段がないことから、論理構造の記述を利用して、「これが重要なコンテンツです」ということを示そうと努力してきました。これが、いわゆるSEO対策（検索エンジン最適化）と呼ばれるものです。しかし、現実的にはそれだけでページの中身を類推するのは難しく、外からの被リンク数など、さまざまな情報を加えて判別材料としていました。

従来のHTMLによるマークアップ



```
<body>
<h1>サイト名</h1>
.....
<h2>記事のタイトル</h2>
.....
<h3>メニューの見出し</h3>
.....
.....
</body>
```

見出しであることを示した箇所。

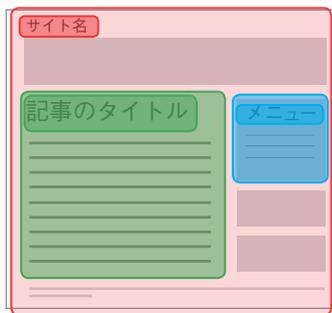
<h1>~<h3>を利用して見出しであることを示していますが、検索エンジンではどれが何の見出しか判別できないことから、前後のコンテンツに含まれるキーワードや文章量などから、重要と思われる部分の類推が行われます。

セマンティクスを導入した場合、製作者はどこが記事のタイトルで、どこがメインの本文なのか、といったページの中身を正確に伝えることができるようになります。これにより、検索エンジンも論理構造から中身を類推する必要がなくなり、必要な情報を的確に取り出すことができるようになります。

また、製作者はページの中身をどこまで詳細に明示するかを選択することが可能です。たとえば、HTML タグのみで大まかに示すこともできますし、microdata (P.111) などのメタデータを活用することで明確に示すこともできます。さらに、HTML5 ではマークアップから文書の構造（アウトライン）をどのように抽出するかが定義されていますので、アウトラインも含めて重視してほしい部分を類推させるという考え方もできるでしょう。

なお、このように HTML によってページの中身を示すことは「セマンティック・マークアップ」とも呼ばれ、それによって人間以外のものが中身を理解できるようになったデータは「構造化データ (structured data)」と呼ばれます。そして、Web ページの情報を構造化データとして扱えるようにし、そのデータを活用していこうという考え方が「セマンティック Web」と呼ばれています。

HTML5 によるマークアップ (HTML タグでページの中身を大まかに示したもの)



```

<body>
<h1>サイト名</h1>
.....
<article>
<h2>記事のタイトル</h2>
.....
</article>
<nav>
<h3>メニューの見出し</h3>
.....
</nav>
.....
</body>

```

ページ全体のコンテンツであることを明示した箇所。

ページ全体のタイトルであることを明示した箇所。

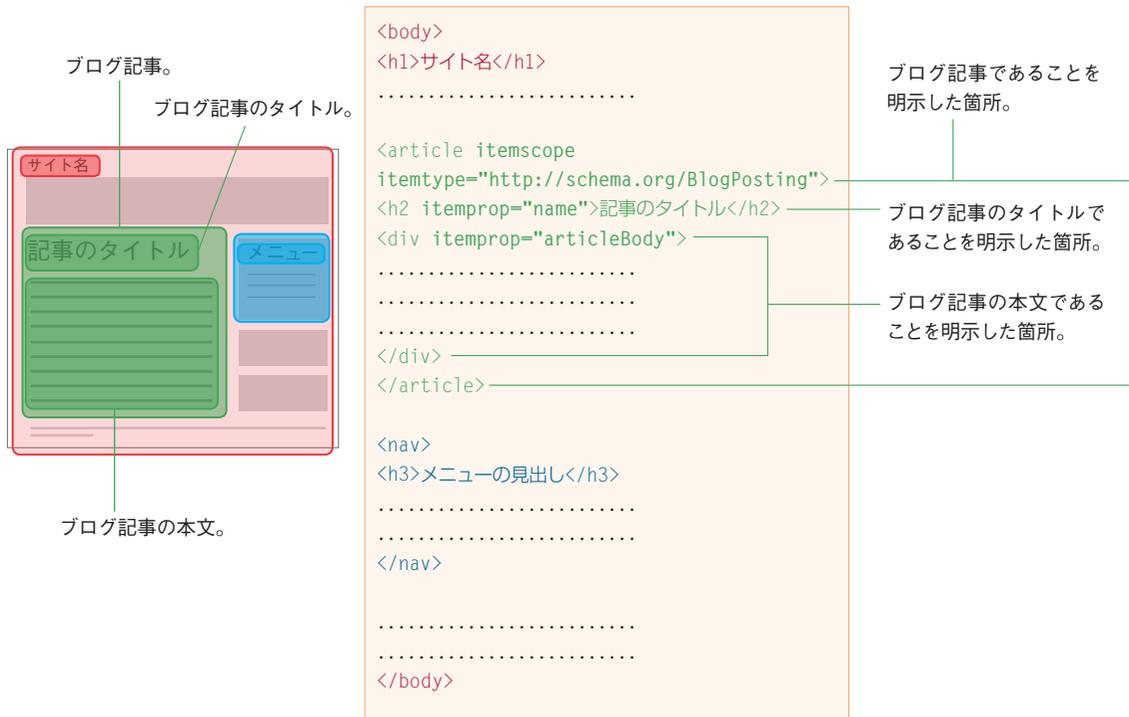
独立したコンテンツのタイトルであることを明示した箇所。

独立したコンテンツであることを明示した箇所。

ナビゲーションのタイトルであることを明示した箇所。

主要なナビゲーションであることを明示した箇所。

HTML5によるマークアップ (microdata でページの中身を明確に示したもの)



NOTES セマンティック Web の提唱

「セマンティックWeb」は、WWWを考案・開発したティム・バーナーズ・リーによって提唱されたもので、1998年に個人的なNotesとして書かれたドキュメント「Semantic Web Road map」はオンラインで読むことができます。

その後、1999年に出版された「Weaving the Web」や、2001年5月の「Scientific American」に掲載された「The Semantic Web」などにおいて、セマンティックWebについての著述を読むことが可能です。

The Semantic Web
Tim Berners-Lee, Ian Houghton, James Hendler, John Handley, Daniel G. Beaulieu
Status: An attempt to give a high-level idea of the architecture of the Semantic WWW. Editing status: Draft. Comments welcome
[Go to English version](#)

Semantic Web Road map

A road map for the Web, an architectural plan intended by working except thought experiments.
This was written as part of a requested road map for future Web design, from a level of 20/2008. It was spun off from an Anticipatory overview for an area which required more discussion than that overview could afford.
Nevertheless, from 20/000 feet, larger visions seem to get a small section, to be architecture, then, in the sense of how things hopefully will fit together. So we should recognize that while it might be slowly changing, this is also a living document.
This document is a plan for exploring a set of connected applications for data on the Web in such a way as to form a consistent logical web of data (semantic web).
Introduction
The Web was designed as an information space, with the goal that it should be useful not only for human-human communication, but also that machines would be able to participate and help. One of the major obstacles to this has been the fact that most information on the Web is designed for human consumption, and even if it is not driven from a database with well-defined strategies (in at least some terms) for its collection, that the structure of the data is not suited to a robot traversing the web. Laying aside the artificial intelligence problems of trying machines to behave like people, the Semantic Web approach instead develops languages for expressing information in a machine-processable form.
This document gives a road map — in response for the incremental introduction of technology to take us, step by step, from the Web of today to a Web in which machine reasoning will be ubiquitous and devastatingly powerful.
It follows the note on the [ontology](#) of the Web, which defines existing design decisions and principles for what has been accomplished to date.

Semantic Web Road map

<http://www.w3.org/DesignIssues/Semantic.html>

NOTES HTML5 と従来の HTML との違い

HTML5と従来のHTMLとの違いについては、W3Cの「HTML5 differences from HTML4」という草案にまとめられています。

HTML5 differences from HTML4

<http://www.w3.org/TR/html5-diff/>

構造化データの活用

HTML5はまだまだ使われ始めたばかりの規格ですので、セマンティック・マークアップを施し、構造化したWebページのデータがどのように活用されることになるかは未知数です。しかし、少しずつではありますが、実際に活用されている場面や、活用が予測される場面も見えてくるようになっていきます。

ブラウザによる記事の抽出

iOS や OS X の Safari に用意されたリーダー機能を利用すると、ページ内の記事のみを抽出して読むことができます。どの部分が記事であるかは文章量などからブラウザが類推し、記事と判別した場合にはリーダーボタンを表示する仕組みです。そのため、文章量が少ない場合などには判別ができず、リーダーボタンは表示されなくなります。

しかし、記事を <article> でマークアップして独立したコンテンツであることを明示しておくこと、文章量が少なくてもブラウザはそこが記事であると認識し、リーダーボタンを表示できるようにになります。また、製作者の意図しない範囲が抽出されるのを防ぐことも可能です。



記事。<article>でマークアップ。

「リーダー」ボタンをクリック。



記事部分のみが抽出して表示され、文字サイズなどを自由に調節して読むことができます。

NOTES

ページ内で複数の<article>を使用した場合には記事として判別されなくなります。

NOTES

Android版のFirefoxにもリーダー機能が搭載され、記事のみを抽出できるようになっています。ただし、HTML5によるマークアップは判別には影響せず、文章量での判別が行われます。

SNSによるページ情報の取得

Facebook や Google+ などの SNS で Web ページの情報を共有すると、タイトルや URL、概要文やサムネイル画像といったページに関する情報が表示されます。こうした情報は SNS がページから自動的に抽出して表示する仕組みになっていますが、ページ内のメインコンテンツとは関係のない、広告などのデータが抽出されてしまうケースがあります。

しかし、RDFa や microdata などのメタデータを利用してページに関する情報を明示しておくことで、製作者が意図した情報をきちんと表示することができます。



FacebookでWebページの情報を共有したときの表示。
RDFaで示したデータが抽出されます。

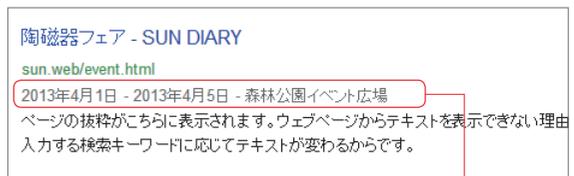
```
<meta property="og:type" content="article">
<meta property="og:title" content="Short Tea Breaks - ちょっとしたお茶の時間">
<meta property="og:url" content="http://sun.web/tea.html">
<meta property="og:image" content="http://sun.web/img/tea-thumb.jpg">
<meta property="og:site_name" content="SUN DIARY">
<meta property="og:description" content="緑茶に紅茶、ウーロン茶など、さまざまなお茶のおいしさと楽しみ方について">
```

RDFaを利用してWebページに関する情報を明示したものを。

語彙にはFacebookが採用したOGP (The Open Graph protocol) を使用しています。

検索エンジンによるリッチスニペットの表示

特定の情報に限られますが、Microdataなどで情報を明示しておく、Googleの検索エンジンによって抽出され、検索結果に「リッチスニペット」として表示されるようになっています。リッチスニペットにはページ制作者のサムネイル画像やイベント情報、商品の価格情報などがあります。なお、リッチスニペットとして表示されないデータも収集されており、Google ウェブマスターツールの「構造化データテストツール」や「構造化データダッシュボード」で確認することができます。



Googleの構造化データテストツールによる
検索結果のプレビュー。

microdataで示したイベントの開催日時と開催場所
がリッチスニペットとして表示されています。

```
<section itemscope itemtype="http://schema.org/Event">
<h1 itemprop="name">陶磁器フェア</h1>
<dl>
<dt>場所:</dt><dd itemprop="location">森林公園イベント広場</dd>
<dt>日時:</dt><dd><time itemprop="startDate" datetimes="2013-04-01">2013年4月1日</time>
~ <time itemprop="endDate" datetimes="2013-04-05">4月5日</time></dd>
</dl>
</section>
```

エージェントの登場

近年、構造化データを利用するものとして iOS の Siri や Android の Google Search など、「エージェント」と呼ばれるものが登場しています。エージェントはユーザーの問いに対して的確な答えを返してくれるもので、従来の検索エンジンとは明らかに異なるものとなっています。

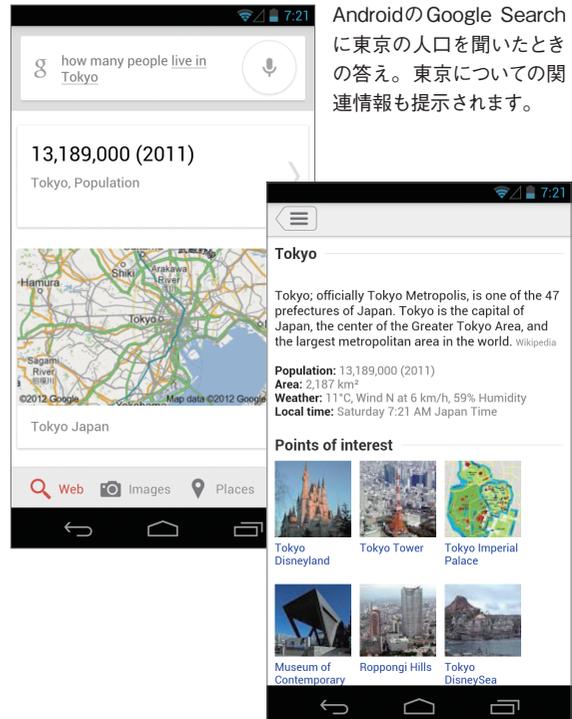
従来の検索エンジンの場合、ページに含まれるキーワードを抽出し、被リンク数などの情報を補完して答えと思われるものをリストアップします。このとき、データの中身を理解しているわけではありませんので、探しているものとは異なる答えが提示されることも多々あります。

エージェントの場合、バックグラウンドに構造化されたデータベースを持っており、データの中身のある程度理解した上で処理を行っています。そのため、探しているものの答えを的確に返すことができるというわけです。ただし、データベースにない情報については的確に答えることができません。本格的に利用できるのが英語環境となっているのも、日本語のデータベースが拡充されていないためです。

現在、エージェントが使用しているデータベースは人間の手によって構築されていますが、Web ページ上には多様で、鮮度の高い情報が膨大に存在します。そのため、Web ページ上の情報が構造化データとして扱えるようになると、こうしたデータベースをより効率よく構築することが可能になるとともに、エージェントが Web ページの中身を直接理解できるようにすることも期待されます。



iOSのSiriに東京の人口を聞いたときの答え。



AndroidのGoogle Searchに東京の人口を聞いたときの答え。東京についての関連情報も提示されます。

NOTES Siri や Google Search のバックグラウンドにあるもの

■ Wolfram Alpha

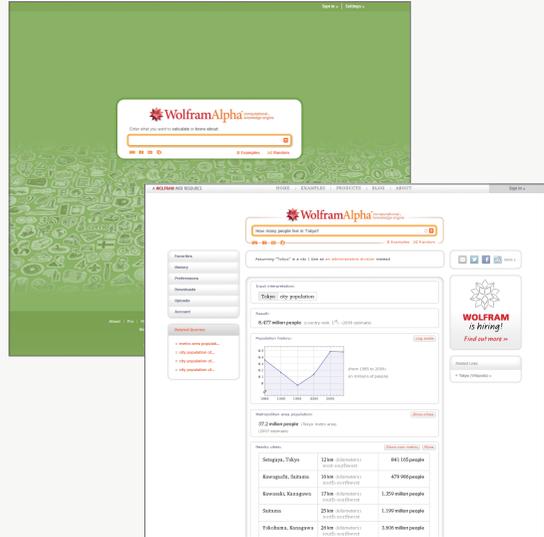
<http://www.wolframalpha.com/>

Siriがバックグラウンドで使用しているシステムです。数式処理システム「Mathematica」の開発者であるStephan Wolframが中心となって開発したもので、独自のデータベースに加え、Best Buyの商品データやFAAのフライトデータなども活用するようになっています。

その機能はSiriだけでなく、Wolfram Alphaのサイトでも試すことが可能です。また、Androidでは以下のアプリがWolfram Alphaを利用した検索機能を備えています。

Androidのアプリ「iris. (alpha)」:

<https://play.google.com/store/apps/details?id=com.dexetra.iris>



Wolfram Alphaのサイトで東京の人口を聞いたときの答え。

■ Knowledge Graph

<http://www.google.com/insidesearch/features/search/knowledge.html>

Google Searchがバックグラウンドで使用しているシステムです。独自のデータベースに加え、WikipediaやCIA World FactbookといったWebサイトのデータも活用するようになっています。Googleによると、2012年の時点で5億件を超える項目と、35億件以上の属性や相互関係などのデータの蓄積があるとされています。

その機能はAndroidのGoogle Searchだけでなく、GoogleのWebサイトで試すこともできます。また、iOS版の「Google検索」で試すことも可能です。

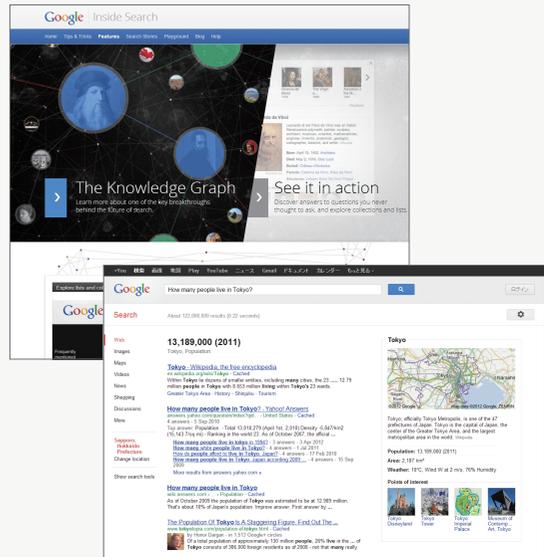
Googleの英語版サイト:

<http://www.google.com/webhp?hl=en>

iOS版の「Google検索」:

<https://itunes.apple.com/jp/app/google-japanese/id284815942>

※iOS版の「Google検索」でKnowledge Graphを活用した検索を行うためには、iOSの[設定] > 一般 > 言語環境 > 言語を「English」に設定する必要があります。



Knowledge Graphのページ(上)と、Googleの英語版サイトで東京の人口を聞いたときの答え(下)。

※ Knowledge Graphを利用した検索は2012年12月にGoogleの日本語版サイトにも導入されました。「東京の人口」と聞いても英語版のような形で答えは返ってきますが、「東京」といった単語で検索すると、ページの右側に東京に関する情報が表示されます。

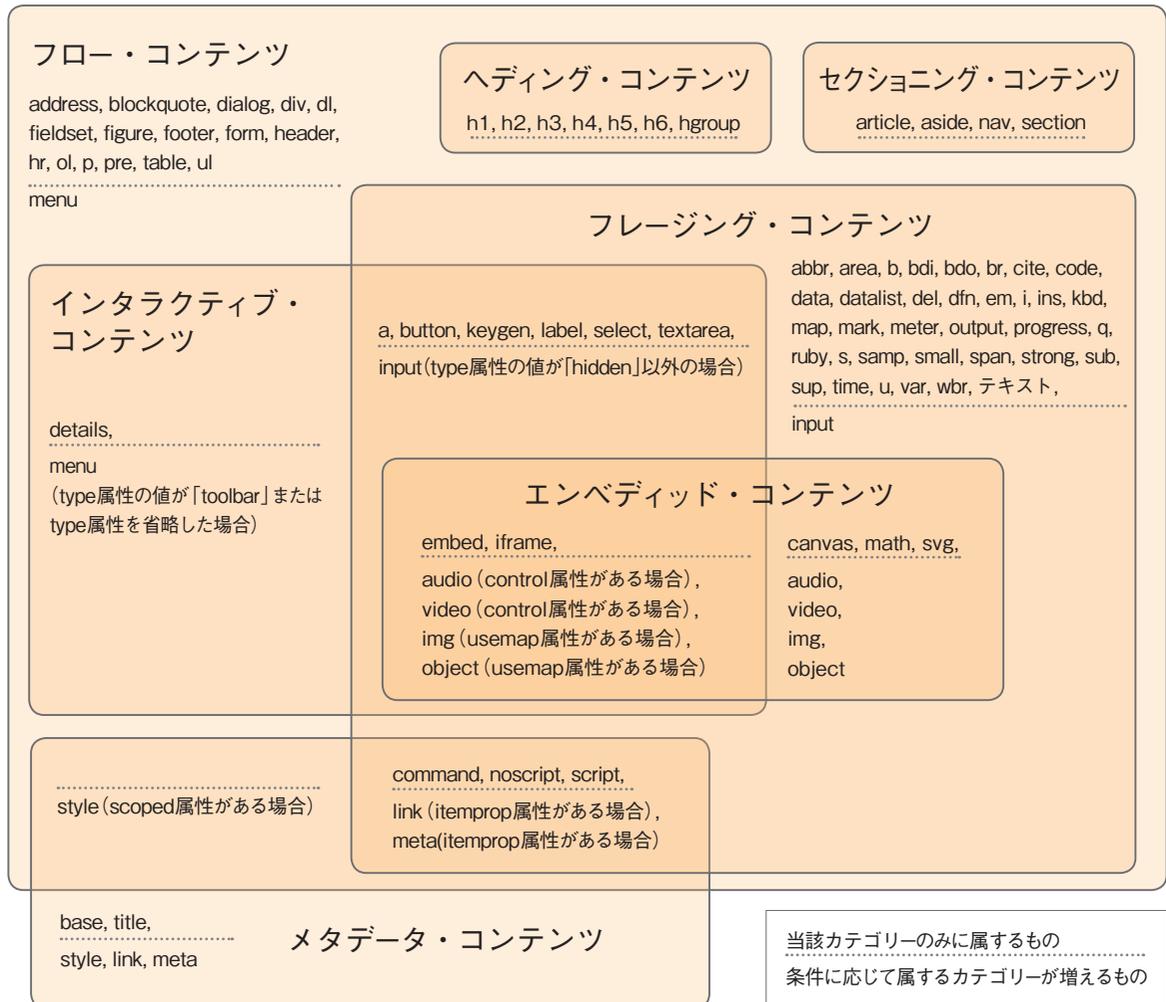
HTML5 のコンテンツ・モデル

HTML5 では、セマンティクスの導入に加えて、まったく新たな「コンテンツ・モデル (Content Models)」が導入されていることも重要なポイントです。

コンテンツ・モデルは要素の相互関係のルールを定義したもので、要素ごとに子要素として内包できるコンテンツの種類が決められています。コンテンツの種類は以下の7つのカテゴリで示されており、カテゴリごとに特徴の共通した要素が分類されています。

たとえば、<p> は「フロー・コンテンツに属し、フレージング・コンテンツを内包できる」と定義されていますので、このルールに従ってマークアップを行う必要があります。

一見、複雑なルールに思えるかもしれませんが、こうしたルールの形成に影響を与えているのがセマンティクスで、さまざまなコンテンツを柔軟にマークアップできる仕組みとなっています。



■ フロー・コンテンツ (Flow content)

「フロー・コンテンツ」には <body> 内でコンテンツのマークアップに使用できる要素が属しています。

■ ヘディング・コンテンツ (Heading content)

「ヘディング・コンテンツ」には、見出しを示す要素が属しています。

■ 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」には、アウトライン (コンテンツの階層構造) を構成する役割を持つ要素が属しています。

■ フレージング・コンテンツ (Phrasing content)

「フレージング・コンテンツ」には短いフレーズのセマンティクスを示す要素が属しています。

■ 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」には画像や動画など、外部リソースを利用する要素が属しています。

■ 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」には、ユーザーの操作を必要とするリンクやフォームなどを構成する要素が属しています。

■ メタデータ・コンテンツ (Metadata content)

「メタデータ・コンテンツ」には、付加情報 (メタデータ) を示すための要素が属しています。基本的に <head> 内に記述しますが、条件によっては <body> 内に記述できるものもあります。

特別なカテゴリー

HTML5 には 7 つのカテゴリーとは異なる分類も用意されており、基本的な分類のみでは規定するのが難しい相互関係や記述ルールが定義されています。各要素は必要に応じてこうしたカテゴリーにも分類され、空の記述を防止したり、リンクを自由に設定したりすることができるようになっていきます。

■ パルパブル・コンテンツ (Palpable content)

「パルパブル・コンテンツ」にはコンテンツを構成する要素が属しています。

「<p></p>」のようにコンテンツのない、空の記述を防ぐため、フロー・コンテンツまたはフレージング・コンテンツを内包できる要素は、hidden 属性を持たない少なくとも 1 つのパルパブル・コンテンツを内包することが推奨されています。パルパブル・コンテンツには「テキスト」も含まれていますので、「<p> こんにちは </p>」のように記述すれば要件を満たしたことになります。なお、スクリプトなどで後からコンテンツを挿入するために空の状態で記述しておくことは認められています。

■ トランスペアレント (Transparent)

内包できるコンテンツの種類が「Transparent」と定義された要素では、臨機応変にリンクを設定したり、コンテンツの編集結果を示すことができるようになっていきます。詳しくは P.042 を参照してください。

パルパブル・コンテンツに属している要素

a, abbr, address, article, aside, audio (controls属性がある場合), b, bdi, bdo, blockquote, button, canvas, cite, code, data, details, dfn, div, dl (子要素に <dt><dd> を含む場合), em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hgroup, i, iframe, img, input (type属性がhidden以外の場合), ins, kbd, keygen, label, map, mark, math, menu (type属性がtoolbarまたはtype属性を省略した場合), meter, nav, object, ol (少なくとも1つの子要素を持つ場合), output, p, pre, progress, q, ruby, s, samp, section, select, small, span, strong, sub, sup, svg, table, textarea, time, u, ul (少なくとも1つの子要素を持つ場合), var, video, テキスト

パルパブル・コンテンツに属していない要素

area, br, datalist, del, dialog, hr, wbr, 半角スペース, メタデータ・コンテンツに属するすべての要素 (meta など)

コンテンツ・モデルが「トランスペアレント」な要素

a, ins, del, map, object

■ セクショニング・ルート (Sectioning root)

「セクショニング・ルート」にはアウトラインのルート階層を構成する要素が属しています。セクショニング・ルートの中に記述されたコンテンツはそれぞれ個別にアウトラインを構成し、他のセクショニング・ルートが構成したアウトラインには影響を与えないようになっています。

たとえば、Apple のページ (P.024) の場合、<figure> でマークアップされたスライダー内の見出しは <body> の構成するアウトラインには出力されていないことがわかります。

■ フォーム関連

フォーム関連の要素はフォームの記述ルールを詳細に定義するため、次のように細かなカテゴリーに分類されています。

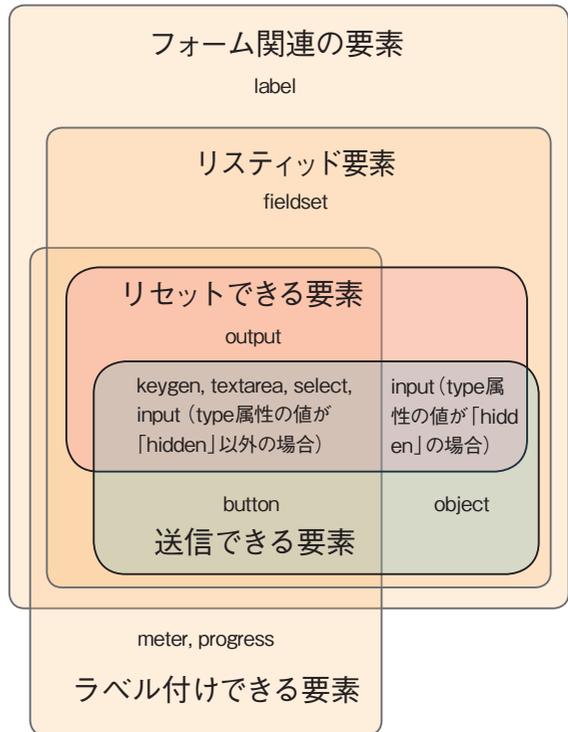
- ・ **フォーム関連の要素 (Form-associated elements)**
<form> と関連付けて利用することができる要素です。
- ・ **リストアップ要素 (Listed elements)**
form.elements と fieldset.elements でリストされる要素です。
- ・ **送信できる要素 (Submittable elements)**
入力・設定内容が送信される要素です。
- ・ **リセットできる要素 (Resettable elements)**
フォームをリセットしたときに影響を受ける要素です。
- ・ **ラベル付けできる要素 (Labelable elements)**
<label> によってラベル付けできる要素です。

■ カテゴリーに属していない要素 (None)

どのカテゴリーにも属していない要素もあります。こうした要素の多くは、<table> や といった特定の要素内で使用するものとなっています。また、Web ページの構成に必要な <html> や <head> もカテゴリーに属していない要素となっています。

セクショニング・ルートに属している要素

blockquote, body, details, dialog, fieldset, figure, td



カテゴリーに属していない主要要素

caption, colgroup, col, dt, dd, figcaption, head, html, li, tbody, thead, tfoot, tr, th など

HTML5の文法とセマンティクス

HTML4.01やXHTML1.0では、各要素はインライン要素とブロック要素の2種類に分類され、ルールも単純でわかりやすいものでした。HTML5ではこれまでの基本的な構文規則は保たれており、HTML5より前のルールで記述しても文法違反とはならないように設計されています。ただし、文法チェックの結果が同じであっても、HTML5とHTML5より前ではその理由が異なる場合がありますので注意が必要です。

たとえば、<div>と<p>を利用した右のサンプルの場合、HTML5とHTML5より前のどちらにおいても、上は文法準拠(○)となり、下は文法違反(×)となります。

HTML5より前でこのように○×となるのは、<div>と<p>がブロック要素で、<div>が内包できるのはブロック要素、<p>が内包できるのはインライン要素と定義されているためです。

HTML5の場合、<div>が内包できるのは「フロー・コンテンツ」、<p>が内包できるのは「フレーズ・コンテンツ」となっていますので、同じように○×となります。

同時にセマンティクスの観点から見た場合、<div>には「連続した要素」、<p>には「段落」というセマンティクスが与えられています。「段落」は「連続したフレーズ・コンテンツ」で構成される、特定のトピックスについて記述された文章」とされていますので、<div>で<p>をマークアップするのは問題ありませんが、<p>で<div>をマークアップするのはやはり問題があることがわかります。



```
<div>
  <p>Short Tea Breaks</p>
  <p>ちょっとしたお茶の時間</p>
</div>
```

<div>の中には<p>を記述することができます。



```
<p>
  <div>Short Tea Breaks</div>
  <div>ちょっとしたお茶の時間</div>
</p>
```

<p>の中には<div>を記述することは認められていません。

NOTES 大きな変更が加えられた要素

HTML5では各要素に細かなセマンティクスが与えられていますが、基本的な役割はHTML5より前と変わらないようにしてあります。たとえば、<p>はHTML5以前から「段落」と定義されていましたが、HTML5では「段落」がどのようなものであるかの説明が追加されているといった具合です。ただし、以下の要素については大きな変更が加えられており、従来のルールで使用するとコンテンツの中身について誤った情報を伝えることになってしまいますので、注意が必要です。

address, b, cite, dl, hr, i, menu, s, small, strong, u

NOTES 廃止された要素

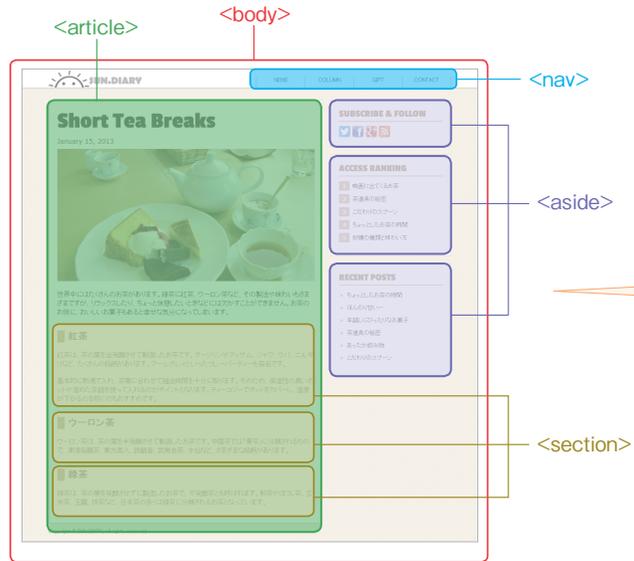
これまで使用できた要素のうち、HTML5では以下の要素が廃止されています。

acronym, applet, base, big, center, dir, font, frame, frameset, isindex, noframes, strike

セクションとアウトライン



セクションの明示



「セクショニング・コンテンツ」カテゴリーに属する要素を利用すると、セマンティクスの上でのコンテンツのグループ（セクション）を示すことができます。このうち、`<section>`のみですべてのセクションを示すこともできますが、より詳細に明示したい場合には `<article>` などを利用することが可能です。

たとえば、本章で作成するサンプルでは、ナビゲーションとして明示したい部分を `<nav>` で、メインコンテンツとして明示したい部分を `<article>` で、関連情報であることを明示したい部分を `<aside>` でマークアップしています。このとき、セクション内に記述した見出しは「セクションの見出し」として扱われます。

また、コンテンツ全体をマークアップした `<body>` は「セクショニング・ルート」という特別なカテゴリーに属しており、最上位階層（ルート）のセクションを示します。そのため、コンテンツ全体を `<section>` などでマークアップする必要はありません。

「セクショニング・コンテンツ」カテゴリーに属する要素	要素が示すセクションの内容	参照
<code><section></code>	汎用セクション	P.071
<code><article></code>	独立したコンテンツ	P.066
<code><nav></code>	ナビゲーション	P.074
<code><aside></code>	補足・関連情報	P.076

NOTES

セクション内に複数の見出しがある場合、1つ目の見出しが「セクションの見出し」として扱われ、残りの見出しは暗黙のセクションを構成します。詳しくはP.082を参照してください。

NOTES

「セクション内には見出しを書かなければならない」と言われることがありますが、HTML5においては「一般的に見出しを含むケースが多い」とされているだけで、見出しを含めずに記述しても問題はありません。セマンティクスの上でも、見出しを必要としないコンテンツが存在することは十分に考えられます。

```

<body>

<header>
<div id="header_inner">
<h1></h1>
<nav>
<ul>
<li><a href="news.html">NEWS</a></li>
<li><a href="column.html">COLUMN</a></li>
<li><a href="gift.html">GIFT</a></li>
<li><a href="contact.html">CONTACT</a></li>
</ul>
</nav>
</div>
</header>

<div id="main">

<article>
<header>
<h1>Short Tea Breaks</h1>
<time datetime="2013-01-15">January 15, 2013</time>
</header>



<p>世界中にはたくさんのお茶があります。緑茶に紅茶、ウーロン茶など、その製法や味わいもさまざまですが、リラックスしたり、ちょっと休憩したいときなどには欠かすことができません。お茶のお供に、おいしいお菓子もあると幸せな気分になってしまいます。</p>

<section>
<h1>紅茶</h1>
<p>紅茶は、茶の葉を全発酵させて製造したお茶です。ダージリンやアッサム、ジャワ、ウバ、ニルギリなど、たくさんの銘柄があります。…といったフレーバーティーも有名です。</p>

<p>基本的に熱湯で入れ、茶葉に合わせて抽出時間を十分に取ります。…温度が下がるのを防ぐのもおすすめです。</p>
</section>

<section>
<h1>ウーロン茶</h1>
<p>ウーロン茶は、茶の葉を半発酵させて製造したお茶です。中国茶では「青茶」に分類されるもので、凍頂烏龍茶、東方美人、鉄観音、武夷岩茶、水仙など、さまざまな銘柄があります。</p>
</section>

<section>
<h1>緑茶</h1>
<p>緑茶は、茶の葉を発酵させずに製造したお茶で、不発酵茶と
も呼ばれます。…緑茶に分類されるお茶となっています。</p>
</section>

</article>

<div id="sidebar">

<aside>
<h1>SUBSCRIBE & FOLLOW</h1>
<ul id="iconmenu">
<li><a href="url"></a></li>
…
<li><a href="url"></a></li>
</ul>
</aside>

<aside>
<h1>ACCESS RANKING</h1>
<ol>
<li><a href="post.html">映画に出てくるお茶</a></li>
…
<li><a href="post.html">砂糖の種類と味わい方</a></li>
</ol>
</aside>

<aside>
<h1>RECENT POSTS</h1>
<ul>
<li><a href="post.html">ちょっとしたお茶の時間</a></li>
…
<li><a href="post.html">こだわりのスプーン</a></li>
</ul>
</aside>

</div>

</div>

<footer>
<small>Copyright &copy; SUN DIARY, all rights reserved.</small>
</footer>

</body>

```

アウトラインの作成

セクションの相互関係や親子関係により、ソフトウェアはページの階層構造（アウトライン）を把握できるようになります。アウトラインを作成するアルゴリズムは右のようになっており、サンプルでは以下のようにアウトラインが作成されます。

なお、アウトラインでは各階層を示すのにセクションの見出しが使用されます。セクションの見出しがない場合には、セクションをマークアップした要素の種類に応じて、「ナビゲーション」といった見出しを自動的に付加してアウトラインを作成することが求められています。

アウトラインを作成するアルゴリズム

- 1 <body>のセクションでルート階層を構成します。
- 2 HTMLソースの上での記述順に従い、セクションごとに以下の処理を実行します。
 - ▶ 前のセクションと同じ階層の場合：
アウトラインの同じ階層に追加
 - ▶ 前のセクションの子階層の場合：
アウトラインの子階層に追加
 - ▶ 前のセクションより上位階層の場合：
アウトラインの親階層に追加

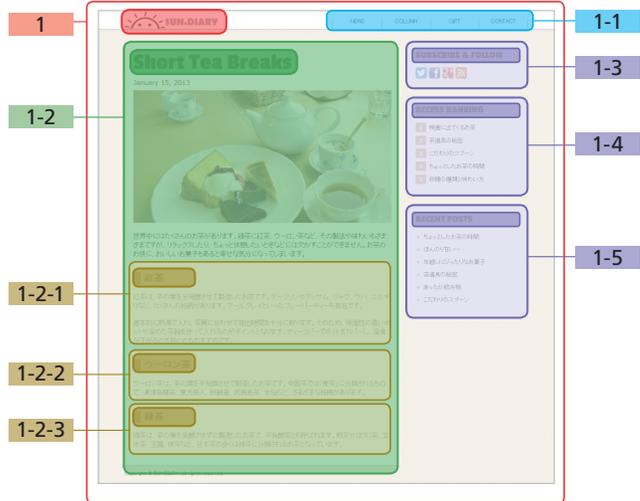
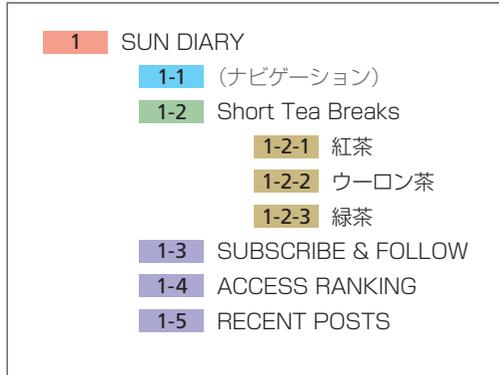
※「セクショニング・ルート」カテゴリに属する<body>以外の要素は、<body>の構成するアウトラインには影響を与えません。

※1つのセクション内に複数の見出しがある場合は暗黙のセクションが構成され、アウトラインに影響します。詳しくはP.082を参照してください。

NOTES

アウトラインはHTML5で定義されたセマンティクスの一部であり、他の要素と複合的に利用されることになります。

サンプルで作成されるアウトライン



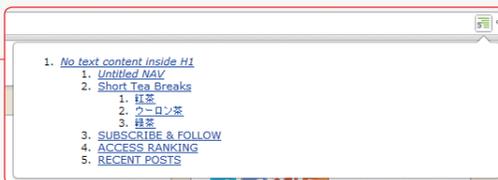
- … <body>で示したセクション。
- … <body>の見出し。
- … <nav>で示したセクション。
- … <section>で示したセクション。
- … <section>の見出し。
- … <article>で示したセクション。
- … <article>の見出し。
- … <aside>で示したセクション。
- … <aside>の見出し。

NOTES アウトラインの確認

以下のツールを利用すると、アウトラインを手軽に確認することができます。

HTML5 Outliner

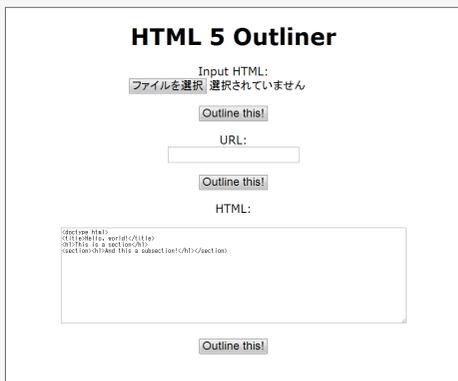
<https://chrome.google.com/webstore/detail/html5-outliner/af0ibpobokebhgfnknfndkgemglgomo>



Chromeの拡張機能で、アドレスバーのアイコンをクリックすると表示中のページのアウトラインを確認することができます。ただし、セクションの見出しが画像の場合、「No text content inside H1」と表示されます。また、セクションの見出しがない場合、「Untitled NAV」といった形で要素の種類が表示されます。

HTML 5 Outliner

<http://gsnedders.html5.org/outliner/>



オンラインで提供されているサービスです。HTMLファイルのアップロードやURLの指定、HTMLソースの入力によってアウトラインを作成することができます。ただし、日本語を含むページの場合、HTMLソースの入力では文字化けすることがありますので、ファイルのアップロードまたはURLの指定を利用します。

セクションの見出しが画像の場合、alt属性の代替テキストの値が表示されます。また、セクションの見出しがない場合、要素の種類に関わらず「Untitled Section」と表示されます。

独立したコンテンツ



```
<article>...</article>
```

HTML5

属性

グローバル

カテゴリ

フロー・コンテンツ / セクショニング・コンテンツ /
バルバブル・コンテンツ

記述可能な場所

フロー・コンテンツが記述できる場所

コンテンツ・モデル

フロー・コンテンツ

構成するボックス

block



<article>でマークアップした部分。

<article> を利用すると、単体で完結した、独立したコンテンツであることを示すことができます。たとえば、ニュースやブログの記事、ユーザーからのコメントや掲示板の投稿、インタラクティブなウィジェットやガジェットなど、その部分だけを取り出してもコンテンツとして完結していることを明示したい場合に利用することができます。

サンプルでは、メイン記事の部分が独立したコンテンツであることを明示するため、<article> でマークアップしています。

IE NOTES

IE8以前のバージョンは<article>に未対応です。

```
<article>
```

```
<header>
```

```
<h1>Short Tea Breaks</h1>
```

```
<time datetime="2013-01-15">January 15, 2013</time>
```

```
</header>
```

```

```

```
<p>世界中にはたくさんのお茶があります。緑茶に紅茶、ウーロン茶など、その製法や味わいもさまざまですが、…お茶のお供に、おいしいお菓子もあると幸せな気分になってしまいます。</p>
```

```
...
```

```
<section>
```

```
<h1>緑茶</h1>
```

```
<p>緑茶は、茶の葉を発酵させずに製造したお茶で、不発酵茶とも呼ばれます。…緑茶に分類されるお茶となっています。</p>
```

```
</section>
```

```
</article>
```

NOTES

iOSやSafariのリーダー機能を利用すると、<article>でマークアップしたコンテンツを抽出して閲覧することができます。詳しくはP.018を参照してください。



「リーダー」ボタンをクリック。

世界中にはたくさんのお茶があります。緑茶に紅茶、ウーロン茶など、その製法や味わいもさまざまですが、リラックスした



こんにちは。先日のお茶会でいただいたダーズリンティーがとても気に入りました。アイスティーにするのもオススメとのことでしたので、今度試してみたいと思います。

Posted by: 匿名希望 2013年1月16日 20:12

次のお茶会も楽しみにしています。

Posted by: にゃん 2013年1月20日 18:35

<article>でマークアップしたコメント。

<article>でマークアップした記事。

<article> の中で <article> を利用すると、独立したコンテンツであり、なおかつ親要素の <article> に関連したものであることを示すことができます。

たとえば、ブログ記事に対して投稿された個々のコメントを独立したコンテンツとして明示したい場合、右のように <article> でマークアップします。このとき、<article> は <article> の中に記述していますので、各コメントが記事に関連したものであることも明示することができます。

```
<article>
<header>
<h1>Short Tea Breaks</h1>
<time datetime="2013-01-15">January 15, 2013</time>
</header>
```

```

```

```
<p>世界中にはたくさんのお茶があります。緑茶に紅茶、ウーロン茶など、その製法や味わいもさまざまですが、…お茶のお供に、おいしいお菓子もあると幸せな気分になってしまいます。</p>
```

```
<section>
<h1>緑茶</h1>
<p>緑茶は、茶の葉を発酵させずに製造したお茶で、不発酵茶とも呼ばれます。…緑茶に分類されるお茶となっています。</p>
</section>
```

```
<section id="comments">
<h1>Comments</h1>
```

```
<article id="comment-01">
<p>こんにちは。先日のお茶会でいただいたダーズリンティーがとても気に入りました。アイスティーにするのもオススメとのことでしたので、今度試してみたいと思います。</p>
<footer>
Posted by: 匿名希望 <time datetime="2013-01-16T20:12">2013年1月16日 20:12</time>
</footer>
</article>
```

```
<article id="comment-02">
<p>次のお茶会も楽しみにしています。</p>
<footer>
Posted by: にゃん <time datetime="2013-01-20T18:35">2013年1月20日 18:35</time>
</footer>
</article>
```

```
<form action="query.php" method="post">
...
</form>
</section>
```

```
</article>
```

セクション
3
セマンティックスによる文書構造の明示

画像



```
<img>
```

属性

alt / src / crossorigin **HTML5** / usemap / ismap / width / height / グローバル

カテゴリ

フロー・コンテンツ / フレージング・コンテンツ / エンベディッド・コンテンツ / インタラクティブ・コンテンツ (usemap属性を指定した場合) / バルバブル・コンテンツ

記述可能な場所

エンベディッド・コンテンツが記述できる場所

コンテンツ・モデル

空

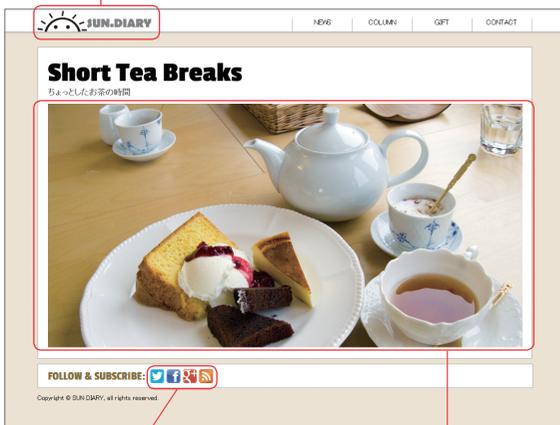
構成するボックス

- (置換要素として扱われることが期待される)

※置換要素の構成するボックスはインラインブロックボックスと同じように扱われます。



ロゴ画像: logo.png



お茶とケーキの写真: tea.jpg

アイコン画像:



twitter.png



facebook.png



googleplus.png



rss.png

`` は画像を示す要素です。src 属性で画像ファイルの URL を、alt 属性で代替テキストを示します。代替テキストは画像の内容を語句や文章で表したもので、画像から直接内容を抽出することのできないソフトウェアや、画像表示をオフにした閲覧環境などで使用されます。そのため、画像を代替テキストに置き換えてもページの内容が変わらないようにすることが求められています。

たとえば、サンプルではサイト名をロゴ画像で、メインコンテンツをお茶とケーキの写真で、Twitter などへのリンクをアイコン画像で表示し、alt 属性で代替テキストを示しています。

```
<header>
<div id="header_inner">
<h1></h1>
...
</div>
</header>

<div id="main">

<article>
<hgroup>
<h1>Short Tea Breaks</h1>
<h2>ちょっとしたお茶の時間</h2>
</hgroup>


</article>

<aside>
FOLLOW & SUBSCRIBE:
<ul id="iconmenu">
<li><a href="url"></a></li>
<li><a href="url"></a></li>
<li><a href="url"></a></li>
<li><a href="url"></a></li>
</ul>
</aside>

</div>
```



装飾的な目的で本文中に表示した写真：tea.jpg

代替テキストは必ず用意することが求められていますが、画像が補足的なもので、前後に関連した内容を記述している場合や、画像が装飾的なものであることを示したい場合には、alt 属性を空の値にすることができます。

たとえば、上のサンプルではサイト名をロゴ画像と文字の組み合わせで表示しています。この場合、ロゴ画像と文字の両方が「SUN DIARY」というサイト名を表していますので、ロゴ画像は補足的なものとして alt 属性の値を空にしています。また、本文中の写真は装飾目的で表示したもののため、こちらも alt 属性の値を空にしています。

この他、 では次のような属性を利用することが可能です。

の属性	機能
alt	代替テキストを指定。
src	画像のURLを指定。
crossorigin	CORSに関する設定を指定。
usemap	イメージマップの設定を指定（参照：P.152）。
ismap	サーバーサイドイメージマップとして利用。
width	画像の横幅を指定（単位：ピクセル）。
height	画像の高さを指定（単位：ピクセル）。

```
<header>
<div id="header_inner">
<h1>SUN DIARY</h1>
...
</div>
</header>
```

```
<div id="main">
```

```
<article>
<header>
<h1>Short Tea Breaks</h1>
<time datetime="2013-01-15">January 15, 2013</time>
</header>
```

```

```

```
<p>世界中にはたくさんのお茶があります。緑茶に紅茶、ウーロン茶など、その製法や味わいもさまざまですが、リラックスしたり、ちょっと休憩したいときなどには欠かすことができません。お茶のお供に、おいしいお菓子もあると幸せな気分になってしまいます。</p>
...
```

```
body > header h1
{font-family: 'Passion One', Verdana, Helvetica, sans-serif;
font-size: 30px;
color: #888888;}
```

サイト名の文字部分のデザインはスタイルシートで設定しています。

DIFFERENCES

HTML 4.01 や XHTML 1.0 で定義されていた longdesc、name、align、hspace、vspace 属性は HTML5 では廃止されています。

longdesc 属性は画像について詳しく説明したページの URL を示すものでしたが、<a> によるリンクに置き換えることが推奨されています。name 属性は id 属性に置き換えることが可能です。また、その他の属性についてはスタイルシートの設定に置き換えるようにします。