



HTML5 版

HTML 演習

中島省吾 著

Webアプリケーション構築に必要な
HTML、CSS、JavaScriptの基礎を学ぶ

SCC



HTML5 版

HTML演習

中島省吾 著

Webアプリケーション構築に必要な
HTML、CSS、JavaScriptの基礎を学ぶ

SCC



HTML5 版

HTML 演習

中島省吾 著

Webアプリケーション構築に必要な
HTML、CSS、JavaScriptの基礎を学ぶ

SCC

- 本書に記載されたURL等は執筆時点でのものであり、予告なく変更される場合があります。
- 本書の使用（本書のとおりに操作を行う場合を含む）により、万一、直接的・間接的に損害が発生しても、出版社および著者は一切の責任を負いかねますので、あらかじめご了承ください。

-
- Microsoft、Windowsは米国Microsoft Corporationの米国およびその他の国における登録商標です。
 - その他、本書に記載されている会社名、製品名などは、一般に各社の商号、登録商標または商標です。
 - 本書では™および®の記載は省略しました。

はじめに

つい最近まで、個人がインターネットを利用するには、必ずパソコンが必要でした。そのため、ネットショッピングなどのサイトを訪問しようと思えば、会社や自宅に戻るかインターネットカフェのような場所を利用するしか方法はありませんでした。

しかし、ノートパソコンが普及してくると公衆無線 LAN を設置するカフェや施設が増え、ノートパソコンを持ち込んでインターネットを利用できるようになりました。その後、携帯電話からもインターネットメールやホームページを表示できるようになり、通信料金も定額制が導入されて、通信費用を気にせずインターネットへアクセスができるようになりました。

そして現在では、アクセス端末も携帯電話からスマートフォンへと移行が進み、パソコンと遜色ないインターネットサービスを、いつでもどこでも利用できるようになってきました。

このように、インターネットへの接続と利用が特別なユーザーのものではなくになると、それに合わせて新たなサービスも普及してきました。有名なものには、Twitter や Facebook といったソーシャルネットワーキングサービス(以降 SNS)やネットゲーム、GPS を利用した地図サービスなどがあります。これらは、常時ネット接続されていることを前提としたサービスです。さらに、SNS と位置情報などを組み合わせてクーポンを配信するなど、数多くの新しいサービスが登場しています。

ところで、このようなインターネットサービスを、自分でも提供してみたいと考えたことはないでしょうか。もちろん、ホームページやブログなどは、簡単に作成できるツールや Web サイトが存在するので、専門知識がなくても作成することは可能です。

しかし、これまでにない Web サービスを実現したり、オリジナルの Web サイトを構築したりしようと思えば、サーバサイドスクリプトや Web アプリケーションの知識がどうしても不可欠になります。

それでは、Web アプリケーションにおいて最も基本的な技術は何かといえば、それはやはり「HTML」でしょう。Web の歴史は HTML から始まりました。しかし、現在の Web ページは、HTML だけで構成されているわけではありません。実際には、見た目を決める「スタイルシート(CSS)」や動きを与える「JavaScript」など多くの技術の上に成り立っています。

そこで、本書では、Web ページの基礎技術である HTML の最新バージョン「HTML5」、CSS の最新バージョン「CSS3」、そして「JavaScript」について、基礎から解説することにしました。

※なお、本書では、サーバサイドスクリプトや Web アプリケーションの作成そのもの、あるいはサーバの作成については扱っていません。

これら、最新の技術を理解する早道は、実際にコードをエディタで入力し、ブラウザに表示してみることです。本書でも、説明の中で数多くのソースコードが登場します。これらのコードを実際に入力して表示させながら説明を読むことで、より理解が深まることでしょう。

本書の直接の目標は、「HTML5 のソースコードを見て、記述されている内容が理解できる」ことです。HTML5 が理解できるようになれば、最新の Web ページを応用して自前のページを作成することも夢ではなくなります。

著 者

サポート

本書に記載された項目に関する最新情報などを公開しています。

URL <http://www.scc-kk.co.jp/scc-books/support/B-400/support.html>

標準学習時間

内容	学習時間
	(1 時限=90 分)
第 1 章 Web ページの基本	2 時限
第 2 章 表示	3 時限
第 3 章 リストとテーブル	3 時限
第 4 章 フォーム	4 時限
第 5 章 C S S 3	4 時限
第 6 章 JavaScript	4 時限
	合計 20 時限

※上記は集合教育で使用する場合の目安です。

HTML のソースコード

本書ではソースコードがどこにあるかがひと目でわかるように、ソースコードの背景に網掛けをしています。また、ソースコードで使用する文字で紛らわしいものに関しては、以下のように区別しやすい形にしています。

表記意味

- 1 数字の 1 (イチ)
 - l 英小文字の l (エル)
 - I 英大文字の I (アイ)
 - 0 数字の 0 (ゼロ)
 - o 英小文字の o (オー)
 - O 英大文字の O (オー)
-

目次

はじめに

第1章 Web ページの基本	1
1.1 Web サイトの仕組み	2
1.1.1 ワールド・ワイド・ウェブ	2
1.1.2 動的な Web ページ	5
1.1.3 Web アプリケーション	7
【確認問題】	8
1.2 Web ページの技術	9
1.2.1 HTML	9
1.2.2 カスケーディング・スタイル・シート	14
1.2.3 クライアントサイドスクリプト	17
【確認問題】	18
1.3 HTML ファイルの基本	20
1.3.1 HTML を勉強するための準備	20
1.3.2 DOCTYPE 宣言	22
1.3.3 HTML のタグ	23
1.3.4 HTML の属性	25
【確認問題】	27
1.4 コンテンツ・モデル	28
1.4.1 コンテンツ・モデルとは	28
1.4.2 コンテンツ・モデルのカテゴリ	29
1.4.3 セクション	32
1.4.4 Web ページの構造化	34
【確認問題】	36
演習問題	37
第2章 表示	39
2.1 テキストの表示	40
2.1.1 重要性、強調などを表す要素	40
2.1.2 時間や注釈、ルビなどを表す要素	42
2.1.3 リンクの指定	45
【確認問題】	52
2.2 画像の表示	53
2.2.1 画像ファイル	53
2.2.2 img 要素	53

2.2.3 画像にリンクを指定する	56
【確認問題】	58
2.3 音楽や映像を表示する	59
2.3.1 サウンドファイルの種類	59
2.3.2 サウンドの再生	60
2.3.3 動画ファイルの種類	62
2.3.4 動画の再生	63
【確認問題】	66
2.4 図形の表示	67
2.4.1 四角形の描画	67
2.4.2 円を描く	70
【確認問題】	73
演習問題	74
第3章 リストとテーブル	75
3.1 リスト	76
3.1.1 リストとは	76
3.1.2 並列順序型リスト	77
3.1.3 順序付きリスト	79
3.1.4 定義型リスト	81
3.1.5 マーカーや数字の変更	83
【確認問題】	86
3.2 テーブル	87
3.2.1 テーブルとは	87
3.2.2 テーブルの基本	88
3.2.3 タイトル、見出しの付いたテーブル	90
3.2.4 ヘッダー、フッターの付いたテーブル	92
3.2.5 テーブルのセルを結合する	94
【確認問題】	97
演習問題	98
第4章 フォーム	99
4.1 フォームとは	100
4.1.1 フォームの役割	100
4.1.2 フォームのコントロール	101
4.1.3 form の基本構文	102
【確認問題】	106
4.2 フォームを記述する	107
4.2.1 様々なテキストフィールド	107

4.2.2	ラジオボタン、チェックボックス	111
4.2.3	セレクト、テキストエリア	114
	【確認問題】	117
4.3	Web サーバで入力を確認する	118
4.3.1	Web サーバの準備	118
4.3.2	フォームの動作	119
	【確認問題】	126
	演習問題	127
第5章	CSS3	129
5.1	CSS とは	130
5.1.1	最新の CSS である CSS3	130
5.1.2	CSS3 の基本	131
5.1.3	CSS の参照方法	135
5.1.4	CSS3 のセレクトタ	138
	【確認問題】	146
5.2	様々なプロパティ	147
5.2.1	テキストの変更	147
5.2.2	CSS3 の新プロパティ	153
	【確認問題】	158
5.3	CSS3 で Web ページをレイアウト	159
5.3.1	ボックスモデル	159
5.3.2	回り込み	164
5.3.3	絶対位置によるレイアウト	167
5.3.4	相対位置によるレイアウト	170
5.3.5	レスポンシブ web デザイン	174
5.3.6	CSS フレームワーク	179
	【確認問題】	183
	演習問題	185
第6章	JavaScript	187
6.1	JavaScript とは	188
6.1.1	オブジェクト指向プログラミング	188
6.1.2	JavaScript の基礎	189
	【確認問題】	195
6.2	制御文	196
6.2.1	if 文	196
6.2.2	for 文による繰り返し処理	201
	【確認問題】	206

目次

6.3 関数と JavaScript のオブジェクト	207
6.3.1 関数	207
6.3.2 配列	209
6.3.3 ブラウザオブジェクト	214
6.3.4 Document Object Model	217
6.4 イベント	221
6.4.1 クリックイベント	221
6.4.2 タイマー処理	223
6.4.3 フォームの入力チェック	224
6.5 JavaScript の応用	233
6.5.1 Ajax	233
6.5.2 jQuery	238
【確認問題】	244
演習問題	246
【確認問題解答・演習問題解答】ダウンロードのご案内	248
付録	249
A. HTML5 のコンテンツ・モデル	250
A.1 コンテンツ・モデルのカテゴリ	250
A.2 トランスペアレント	252
B. Web サーバのインストール	254
B.1 XAMPP のインストール	254
C. HTML5 の要素、属性、値の一覧	260
D. CSS3 の主なプロパティ、値の一覧	294
E. JavaScript の主なオブジェクト、メソッド、プロパティの一覧	314
F. Web カラー	324
F.1 Web セーフカラー	324
F.2 カラーネーム	326
索引	329

第1章

Web ページの基本

普段、何気なく利用しているWebサイトのサービス。では、どんな仕組みでWebサイトが動いているのかご存じでしょうか？

この章では、インターネットやWWWの仕組み、Webページの構造、HTMLの基本を学びます。

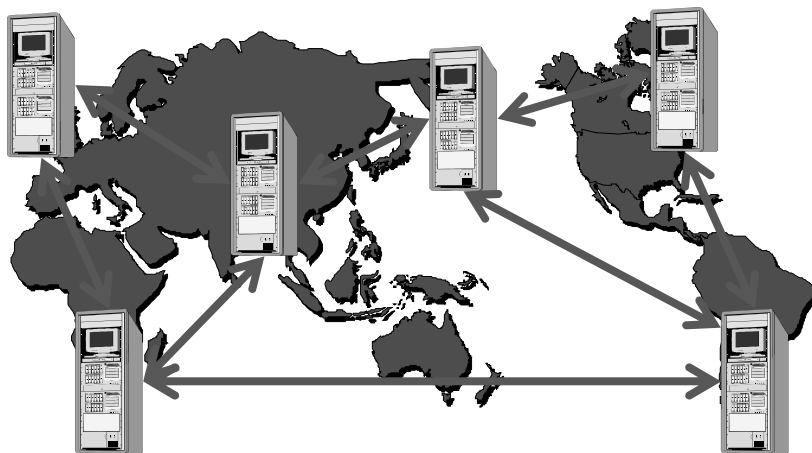
1.1 Web サイトの仕組み

インターネットには、検索サイトやショッピングサイトなど、数多くのサービスが存在します。しかし、多くのユーザーがその仕組みまで理解はしていないでしょう。ここでは、Web サイトの仕組みについて解説します。

1.1.1 ワールド・ワイド・ウェブ

現在、「インターネット」に関連する技術は生活に欠かせないものとなっています。このインターネットを、大きく普及させる原動力となったのが「ワールド・ワイド・ウェブ」と呼ばれる仕組みです。

ワールド・ワイド・ウェブは、コンピュータ同士が通信網を利用してデータを送受信します。この通信網に「インターネット」を利用することで、世界中のサーバと通信することが可能になります。その様子が、まるで「蜘蛛の巣(Web)」のように見えることから ワールド・ワイドWorld Wide ウェブWeb、通称「WWW」と呼ばれるようになりました。



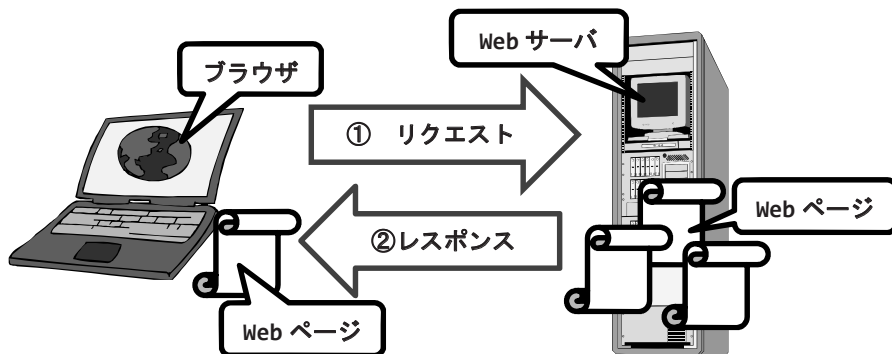
インターネットとは、世界中に張り巡らされたコンピュータの通信網

WWWでは、コンピュータ同士の通信手順に、ハイパーテキスト トランスファー プロトコル「HTTP(HyperText Transfer Protocol)」と呼ばれる方式を採用しています。

HTTP通信は、コンピュータがネットワーク上の別のコンピュータに対して、データを要求するところから始まります。この要求を「リクエスト」と呼びます。すると、要求されたコ

コンピュータはデータを返信します。この返信を「レスポンス」と呼びます。

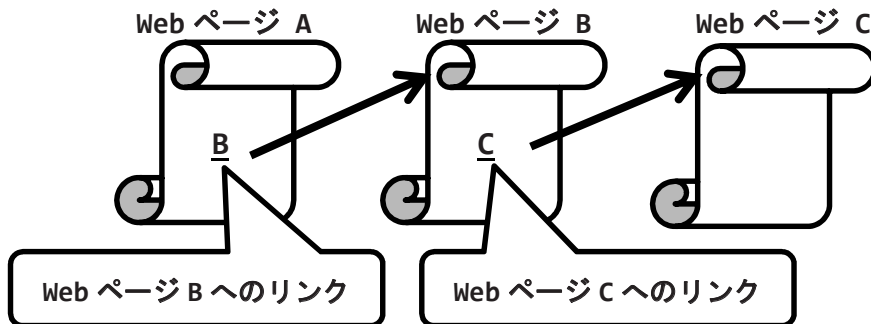
また、送受信されるデータのことをWWWでは「Webページ」と呼び、Webページをリクエストするソフトを「ブラウザ」、Webページをレスポンスするソフトを「Webサーバ」と呼びます。



このように、HTTP通信は非常に単純な手順なのですが、Webページにはある巧妙な仕掛けが施されています。それは、ドキュメントの中に別のドキュメントの場所「URL」を埋め込むことができるのです。

URLとは「ユニフォーム リソース ロケータUniform Resource Locator」の略で、インターネット上にあるWebサーバを特定するための「アドレス(住所)」のようなものです。WWWではこのURLを使ってWebサーバを特定し、Webページをリクエストします。

このようにドキュメントに埋め込まれたURLを「リンク」と呼び、リンクを埋め込むことができるドキュメントのことを「ハイパーテキスト」と呼んでいます。



第1章 Web ページの基本

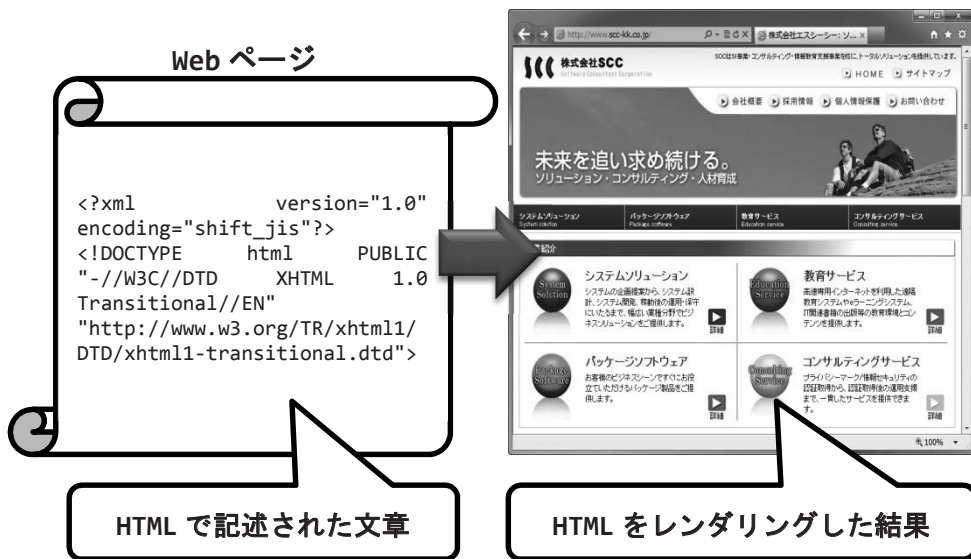
ハイパーテキストに埋め込むことができるのは、別のWebページのURLだけではありません。Webサーバ内の画像データなども文章の中に埋め込むことができます。また、文章の特定部分が「見出し」であることや、「表」であること、「文の意味や役割」といった情報も文章内に埋め込むことができます。

もちろん、このような情報を文章に埋め込むには、コンピュータが理解できるように、ルールに則ってWebページが記述されていなければいけません。このルールのことを

ハイパーテキスト マークアップ ランゲージ

「HyperText Markup Language」、通称「HTML」と呼んでいます。

そしてパソコンのブラウザは、レスポンスとして送信されてきたWebページのHTMLを理解して画面を合成します。この「HTMLを解釈してブラウザに画面を合成する作業」のことを「**レンダリング**」と呼んでいます。



One Point

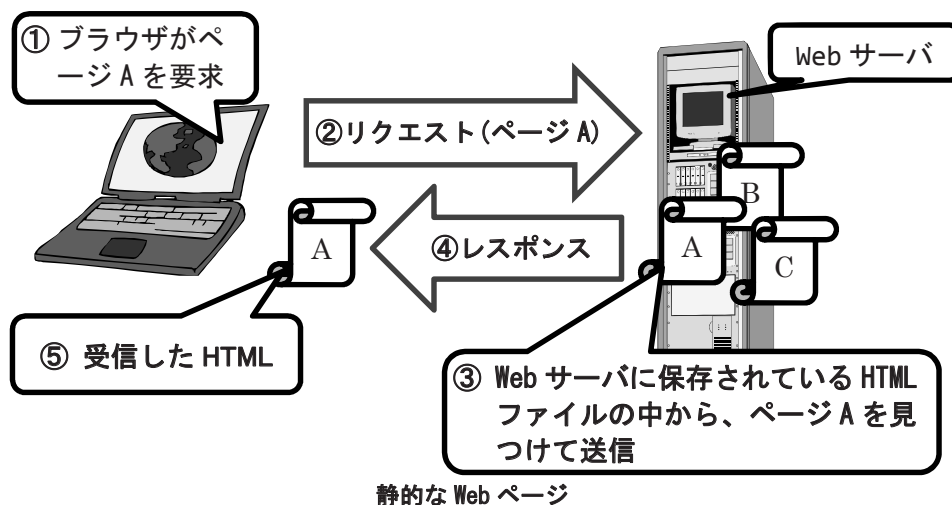
現在、パソコンで利用できるブラウザには、多くの種類があります。有名なものとしては、インターネット エクスプローラ「Internet Explorer」、サファリ「Safari」、グーグル クローム「Google Chrome」などがあります。これら主要なブラウザは、パソコンだけでなくスマートフォン用のものもあり、複数のパソコンやスマートフォン同士で、ブックマークなどのデータを共有できるものもあります。

1.1.2 動的な Web ページ

HTTP 通信では、ブラウザが Web サーバにリクエストすると、リクエストされた Web サーバは指定された Web ページをブラウザに送信します。

つまり Web サーバ側には、予めリクエスト時に送信するための Web ページが保存されている必要があります。Web ページは HTML で記述するため、Web サーバには「HTML ファイル」の形で保存されています。

このように HTML ファイルとして保存されている Web ページのことを「**静的な Web ページ**」と呼びます。



これに対して、検索サイトやショッピングサイトなどでは、ブラウザから入力した検索キーワードやユーザー名によって、レスポンスされてくる Web ページは変化します。

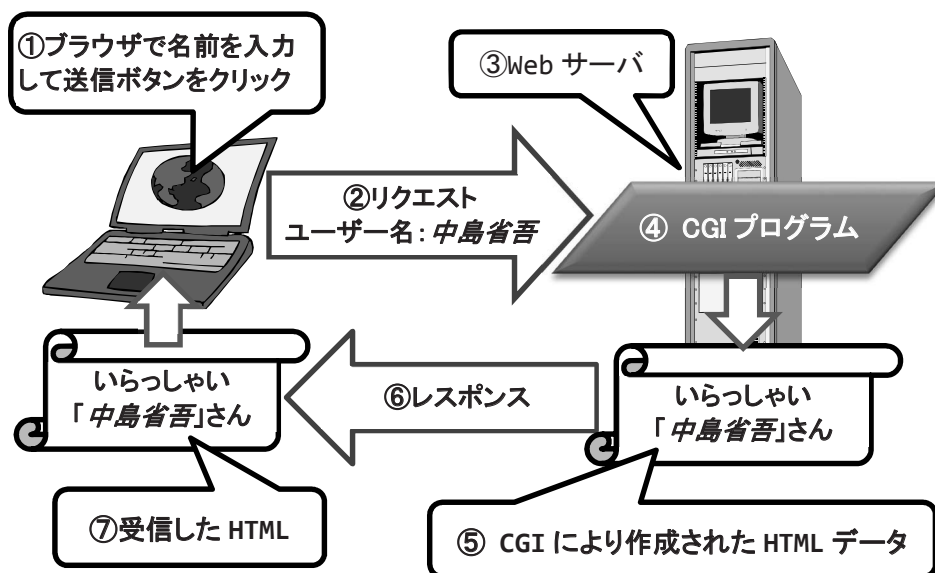
このように、どのようなデータがブラウザから送信されてくるのか予測できないような場合、予め Web ページを作成して保存しておくことはできません。

そこで、ブラウザから Web サーバ側にリクエストが届いた時点で、リクエストに添付されたデータを元に「プログラム」が Web ページを作成する仕組みが考え出されました。

このような仕組みを「コモン ゲートウェイ インターフェイス Common Gateway Interface」、通称「CGI」と呼んでいます。

CGI では、リクエストがあると Web サーバと連携してプログラムが起動します。この Web サーバ側で動作するプログラムを「**サーバサイドプログラム**」と呼びます。そしてリクエストに添付されたデータを使い、プログラムはその場で Web ページを作成します。

CGI では、ブラウザから届いたデータを利用して自動的に(能動的)に HTML データを作成するため、このような Web ページを「動的な Web ページ」と呼んでいます。



動的な Web ページ

One Point

動的な Web ページを作成するためには、Web サーバ側で Web サーバと連動する CGI プログラムが必要です。このような CGI プログラムを作成できるプログラミング言語には「Perl」「PHP」「Python」「Ruby」などがあります。

また、Web サーバと連動するプログラムを作成できるプログラミング言語のことを、「サーバサイドスクリプト言語」と呼ぶ場合もあります。

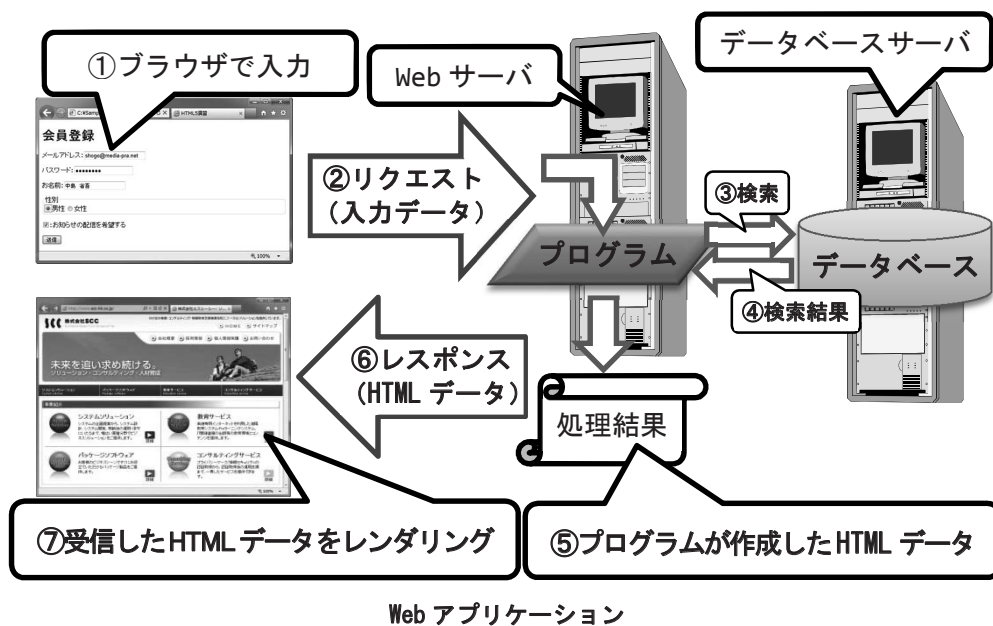
近年、Web サイトは大規模化しており、複数台の Web サーバを束ねて同時運用することも珍しくありません。このような場合には、Oracle社の「Java」やMicrosoft社の「.NET(ドットネット)」といった、分散処理に対応した技術を用いる場合もあります。

1.1.3 Web アプリケーション

CGI のようなサーバサイドプログラムを利用すれば、ブラウザからの入力をきっかけにして様々なプログラムを走らせ、その処理結果をブラウザに返信することができます。つまり、動的な Web ページを使えばワードプロセッサや計算表ソフトといった「アプリケーション」も作成できるということです。

このように、ブラウザ画面を出力先とするアプリケーションのことを「Web アプリケーション」と呼びます。

また、サーバサイドプログラムは、他のサーバと連携することもできます。たとえば、検索サイトやショッピングサイトなど大規模な Web アプリケーションでは、データベースサーバと連携することで膨大なデータを処理しています。



One Point

データベースサーバにも、多くの種類があります。Web アプリケーションで利用されるデータベースサーバには、Oracle 社の Oracle、Microsoft 社の SQL Server、IBM 社の DB2、オープンソースの MySQL (マイエスキューエル)、PostgreSQL (ポストグレスキューエル) などがあります。

確認問題

■確認問題 1.1 (1)

HTTP による通信で、ブラウザから Web サーバに対して Web ページを要求することを何と呼びますか。次の中から番号で答えなさい。

- ① ウェブ
- ② リクエスト
- ③ レスポンス
- ④ プロトコル

■確認問題 1.1 (2)

Web サーバから送信されてきた HTML を解釈して画面を合成することを何と呼びますか。次の中から番号で答えなさい。

- ① ハイパーテキスト
- ② リンク
- ③ レンダリング
- ④ URL

■確認問題 1.1 (3)

Web アプリケーションでは、ブラウザから入力されたデータを利用してサーバサイドプログラムが HTML データを生成します。このような Web ページを何と呼びますか。次の中から番号で答えなさい。

- ① ワードプロセッサ
- ② 表計算ソフト
- ③ 静的 Web ページ
- ④ 動的 Web ページ

1.2 Web ページの技術

Web ページの基本は、Web サーバから送信されてきた Web ページをブラウザがレンダリングして表示することですが、そのためには HTML 以外にも様々な技術が使われます。

ここでは、Web ページに関連する技術の中から HTML、CSS、JavaScriptの概要を解説します。

1.2.1 HTML

WWW では、ブラウザが Web サーバに Web ページをリクエストすると、該当する HTML ファイルのデータがブラウザに返信されます。このような「静的な Web ページ」の場合、Web サーバがなくても HTML ファイルを直接ブラウザで開いて確認することができます。

たとえば、次のようなテキストファイルを用意したとします。

注意: ブラウザに表示するファイルは、文字エンコーディングに注意してください。以下のファイルは、Windows 上で表示するため、文字コードを CP932 (Shift_JIS を Microsoft が独自に拡張した文字コード) で保存しています。

ファイル名 : sample121_1.html

初めてのHTML

ようこそ、HTML5の世界へ

ファイル拡張子を「.html」にしているのは、Windows では HTML ファイルとして関連付けが行われ、ダブルクリックでブラウザにファイルを表示させることができるためです。

このファイルをブラウザで開くと、以下のようにファイルの内容を表示することができます。



Internet Explorer 11 で表示した例

第1章 Web ページの基本

こうして、ファイル内のテキストをブラウザに表示することはできましたが、この「sample121_1.html」は、拡張子こそ「html」ですが正式な HTML ファイルとしては不十分です。

なぜなら、HTML には「タイトルは<title>タグで括る」、「本文の始まりから終わりを<body>タグで括る」といった多くのルールが決まっていて、そのルールに従って文章の役割を明確にしなくてはならないからです。

では、HTML のルールを当てはめた、HTML ファイルを作成しましょう。

ファイル名 : sample121_2.html

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>初めてのHTML</title>
  </head>
  <body>
    ようこそ、HTML5の世界へ
  </body>
</html>
```

注意：ここでは、HTML ファイルの文字コードを「UTF-8」で保存しています。

このように、HTML では「<タグ名> ~ </タグ名>」で文章を囲い、文章に意味や役割を設定します。たとえば<title> ~ </title>で囲まれた「初めての HTML」部分は、ページタイトルとしてブラウザが解釈します。そのため「sample121_2.html」を IE11 で表示すると、ウィンドウにではなく、タブにページタイトルとして表示されます。



Internet Explorer 11 で表示した例

また、タグ自体は文章に意味や役割を与えるものなので、ブラウザには表示されません。

このような「タグにはどんな種類があり、意味は何か」といったルール作りを「標準化」と呼び、HTMLは「ワールド ワイド ウェブ コンソーシアムWorld Wide Web Consortium」、通称「W3C」と呼ばれる団体がHTMLの標準化を進めています。

本書執筆時点（注：2016年）におけるHTMLの最新バージョンは、W3Cが2014年に「勧告（バージョンが確定すること）」したHTML5です。

ちなみに、HTML5以前のバージョンは、1999年に勧告されたHTML 4.01です。HTML5の勧告まで、実に15年もの歳月を要していますが、これには深い訳があります。

本来、W3Cが考えていたシナリオでは、HTML4.01の次は、エクステンシブル ハイパーテキスト マークアップ ランゲージExtensible HyperText Markup Language」通称「XHTML」へ移行する予定でした。

ところが、XHTMLはWeb開発者の賛同を得ることができず、しびれを切らしたブラウザメーカーや開発者が、新たなコミュニティを発足し「ウェブ アプリケーションズWeb Applications 1.0」や「ウェブ フォームズWeb Forms 2.0」といった新機能を提唱するようになりました。その結果、XHTMLの標準化作業は打ち切られ、「Web Applications 1.0」「Web Forms 2.0」を吸収した形で「HTML5」の標準化作業へ移行しました。

このような経緯があったため、HTML5の勧告には時間がかかってしまったのです。

このように紆余曲折あって誕生したHTML5ですが、その最大の特徴は「CSS3」や「JavaScript」といった「本来はHTMLとは別の技術」を、HTML5の仕様の中に含めてしまった点にあります。

これにより、これまでブラウザ毎に仕様がバラバラだったり、「フラッシュFlash」のようなブラウザとは別のプログラム（プラグイン）が必要だったりした以下の機能について、ブラウザ単体で、かつ統一した仕様で処理が実行できるようになりました。

- マルチメディア再生、グラフィックス処理
- フォーム機能の強化
- オフラインでの動作や、ローカルファイルへのアクセス
- ドラッグアンドドロップの実現
- 同じページのままWebサーバとの通信
- バックグラウンドでの処理
- クッキーCookieよりも大きなデータをブラウザ側に保持

One Point

「W3C」が行っている「HTML の標準化」とは、HTML に新たに追加するタグや機能を、複数の人や企業が検討する作業を言います。そしてある程度決まった状態を「ドラフト」と呼び、実際に試すことができるようになります。次に、ドラフトで発生した問題を修正した「勧告」の状態になります。この「勧告」をもって、バージョンが確定します。

参考

HTML のように Web で利用するファイルを作成するときは、Windows の「メモ帳」は使用しない方がよいでしょう。

その理由は、メモ帳でファイルの文字コードを UTF-8 で保存すると、ファイルの先頭にバイトオーダーマーク (byte order mark)、通称「**BOM**」が、必ず追加されるためです。

そこで、HTML ファイルを保存するときは、文字コードに UTF-8 (BOM なし) もしくは、UTF-8N が選択できるテキストエディタを使用してください。

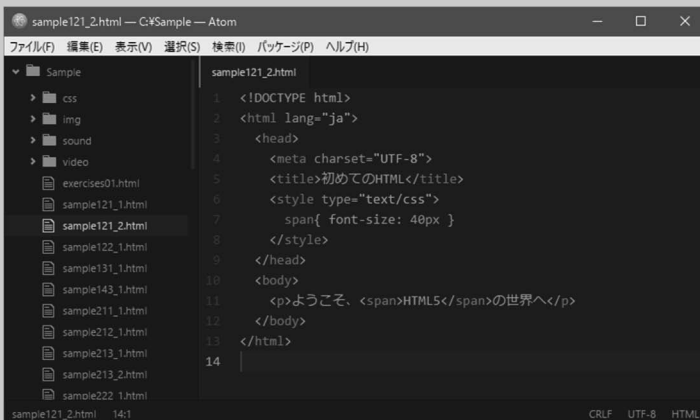
たとえば、筆者のお勧めのテキストエディタは、「ATOM」です。

ATOM は、オープンソースのテキストエディタで、Windows 用、Mac OS 用などがあり、デフォルトの保存形式が「UTF-8 (BOM なし)」なので、とても便利です。

以下のサイトからダウンロードできます。是非、使ってみてください。

<https://atom.io/>

ATOM



The screenshot shows the Atom text editor interface. The title bar reads "sample121_2.html - C:\Sample - Atom". The menu bar includes "ファイル(F)", "編集(E)", "表示(V)", "選択(S)", "検索(O)", "パッケージ(P)", and "ヘルプ(H)". The left sidebar shows a file tree with folders "css", "img", "sound", "video" and several HTML files, with "sample121_2.html" selected. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4 <meta charset="UTF-8">
5 <title>初めてのHTML</title>
6 <style type="text/css">
7   span{ font-size: 40px }
8 </style>
9 </head>
10 <body>
11 <p>ようこそ、<span>HTML5</span>の世界へ</p>
12 </body>
13 </html>
14
```

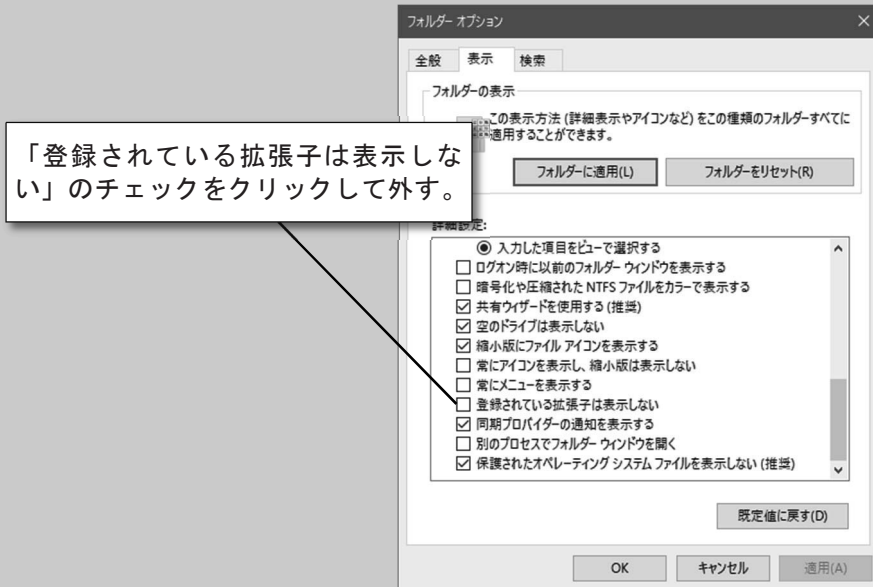
At the bottom right of the editor, the status bar shows "CRLF UTF-8 HTML".

参考

Windows は、よく使う拡張子がデフォルトで非表示になっています。その場合「フォルダーオプション」を使い、拡張子を表示するように変更しておきましょう。

Windows 10 の「フォルダーオプション」を使う手順は、次のようになります。

エクスプローラーを起動して、[表示]タブのリボンにある[オプション]ボタンをクリックし「フォルダーオプション」ダイアログを表示します。[表示]タブをクリックして、「登録されている拡張子は表示しない」のチェックをクリックして外します。[適用]ボタンをクリックして適用後、[×]ボタンでダイアログを閉じます。



1.2.2 カスケーディング・スタイル・シート

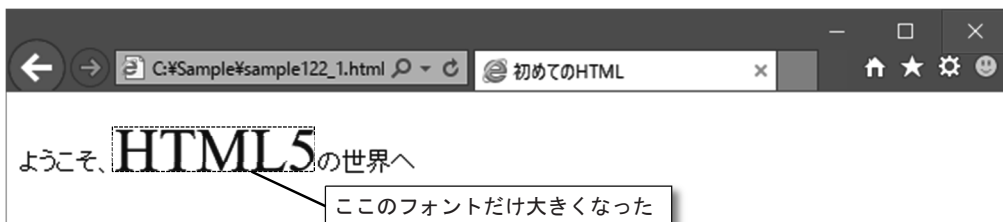
カスケーディング スタイル シート
「Cascading Style Sheets」、通称「CSS」とは、Web ページをどのように修飾(デザイン)するかを指示する仕様です。

たとえば、文章の一部分のフォントサイズを 40 ピクセルで表示したい場合、次の様に、`<style type="text/css">` ~ `</style>` タグで CSS を設定します。

ファイル名 : sample122_1.html

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>初めてのHTML</title>
    <style type="text/css">
      span{ font-size: 40px }
    </style>
  </head>
  <body>
    <p>ようこそ、<span>HTML5</span>の世界へ</p>
  </body>
</html>
```

上記の CSS を使うと、文章の特定の部分のフォントサイズを変更できます。



Internet Explorer 11 で表示した例

HTML4.01 までは、タグ自体に見た目を変更できる機能が多く用意されていました。先ほどのフォントも、CSS を使わずに変更することができました。しかし HTML5 では、様々な問題から、見た目を変更する機能はほとんど CSS へ移行することにして、見た目をタグで変更することは禁止されました。

これには、次のような理由があったからです。

たとえば、数十ページ以上ある Web サイトを作成していたとき、途中で見出しのデザインを少しだけ変更したいと思ったとします。HTML だけで Web ページを記述していると、すべてのページの見出しを丹念に検索して変更していく必要があります。

しかし、文書の構造と見た目を分離する CSS を使った場合は、どうでしょうか。

CSS で見出しのデザインを定義しておけば、CSS の見出しを変更するだけで、すべての文書の見出しが変わります。

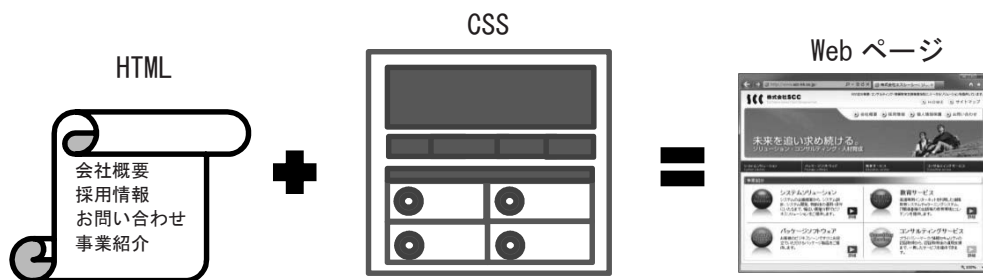
また、CSS の部分は、別のファイルに分離することもできるので、このファイルを予めブラウザが読み込んでおけば、利用しているページのレンダリングが速くなるといったメリットもあります。

実は、Web ページのレイアウトについても、HTML5 では大きな変更があります。

HTML4.01 以前の Web ページでは、table タグを使って Web ページのレイアウトを整えるものが多くありますが、本来 table タグは「表」を表現するためのタグであり、レイアウトのためのタグではありません。

もし、目の不自由なユーザーが音声読み上げソフトを使って Web ページを閲覧している場合でも、入り組んだ table タグでは、正しい順番で再生することができません。

このようなユーザービリティの観点からも、HTML5 では文章と見た目を完全に分離して、見た目は CSS に記述するようになったのです。



CSS にも HTML 同様にバージョンがあります。最新のバージョンは CSS3（一部、公開草案として CSS4 と呼ばれるバージョンもある）ですが、2011 年 6 月に勧告された CSS2.1 以降のバージョンについては、「CSS Color Module Level 3」「CSS Namespaces Module Level 3」等モジュール単位で機能が追加され、それぞれ個別に勧告されるようになりました。

第1章 Web ページの基本

このように、複数存在する CSS のバージョンの中で、HTML5 が利用する CSS は、CSS3 以降になります。

CSS3 には、ブロックの角を丸くデザインしたり、文字に陰をつけたり、グラデーションを表現できるなど、多くの新機能が追加されています。

本書では、この CSS3 を 5 章から学習します。

1.2.3 クライアントサイドスクリプト

動的な Web ページを作るには、Web サーバ側で HTML を生成するプログラムが必要になります。このサーバ側で動作するプログラムを作成する言語を、「サーバサイドスクリプト」と呼ぶことはすでに説明しました。

この「サーバサイドスクリプト」とは別に、HTML と一緒に記述して Web ページに動きや処理を追加する「クライアントサイドスクリプト」と呼ばれる言語があります。

「クライアント」とは、サーバに対して命令を与える側のコンピュータやソフトウェアのことで、WWW では「ブラウザ」がクライアントソフトの仲間になります。

つまり「クライアントサイド」とは、クライアントである「ブラウザ側」で動作するプログラミング言語という意味になります。

クライアントサイドスクリプトには、いくつかの種類がありますが、HTML5 と組み合わせるのは、「JavaScript」と呼ばれる言語です。

JavaScript は、「オブジェクト指向」という考え方を採用しているため、何か命令を実行するときは、用意されている「オブジェクトの機能(メソッド)を呼び出す」必要があります。

この JavaScript の仕様を決定しているのは、「ヨーロッパ European Computer Manufacturer Association」通称「ECMA」と呼ばれる標準化団体です。そして、現在多くのブラウザが採用している JavaScript は、「ECMAScript 6th エディション」と呼ばれる、2015 年に作成された言語仕様に基づいています。

また、実際に JavaScript で利用できるオブジェクトや機能の集合を、JavaScript の「アプリケーション プログラミング インターフェイス Application Program Interface」、通称「API」と呼んでいます。HTML5 では、専用の API が多数提供されています。

本書では、この JavaScript を 6 章から学習します。



検索ボタンを押す前に、入力途中から検索結果を表示する。



マウスの動作を検知して、画像を切り換えたり移動させたりする。



タイマーやインターバルを利用して自動的に処理を呼び出す。

JavaScript でできること

確認問題

■確認問題 1.2 (1)

HTML は、何によって文章に意味や役割といった情報を埋め込みますか。次の中から番号で答えなさい。

- ① タグ
- ② 内容
- ③ CSS
- ④ JavaScript

■確認問題 1.2 (2)

HTML5 の特徴として、正しいものをすべて次の中から番号で答えなさい。

- ① 「CSS3」や「JavaScript の API」を、仕様から分離することに成功した。
- ② 同じページのまま Web サーバと通信できる。
- ③ バックグラウンド処理ができない。
- ④ マルチメディア再生ができる。

■確認問題 1.2 (3)

CSS の特徴として、正しいものをすべて次の中から番号で答えなさい。

- ① CSS は、Web ページの文章とデザインを分離する。
- ② CSS は、HTML ファイル内に記述することはできない。
- ③ CSS は、別のファイルに分離することはできない。
- ④ CSS では、`tablet` タグを使って Web ページのレイアウトを整える。

■確認問題 1.2 (4)

ブラウザ側で動作するプログラミング言語を何と呼びますか。正しいものを番号で答えなさい。

- ① Java 言語
- ② API
- ③ サーバサイドスクリプト言語
- ④ クライアントサイドスクリプト言語

1.3 HTML ファイルの基本

Web サイトや Web ページの仕組みを理解するには、最初に HTML の構造を理解するのが早道です。ここでは、HTML の基本構造を学習します。

1.3.1 HTML を勉強するための準備

HTML を勉強するのに特別なツールは必要ありません。用意するのは「テキストエディタ」と「ブラウザ」です。

テキストエディタは、文字コードに「UTF-8(BOMなし)」または「UTF-8N」が選択できるテキストエディタを用意してください。

ブラウザについては、「HTML5 対応ブラウザ」を用意します。インターネット エクスプローラ Internet Explorer(バージョン 11 以上)や、グーグル クローム Google Chrome、サファリ Safariなど、最新バージョンのブラウザであれば、HTML5 の仕様のほとんどに対応しています。

テキストエディタの準備ができたなら、以下のように入力してみましょう。

ファイル名 : sample131_1.html

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>HTML5演習</title>
  </head>
  <body>
    <h1>初めてのHTML5</h1>
    <p>HTML5は、最新のHTMLです。<br>
    今後、Webページの主流になることは間違いないでしょう。</p>
  </body>
</html>
```

入力が完了したら、ファイル名を「sample131_1.html」にして保存します。

保存する場所(ディレクトリやフォルダ)は、自分が分かりやすい場所であれば何処でもかまいません。本書では Windows 10 を使用しているので、C ドライブのルートに Sample フォルダを作り保存しています。保存の際に注意する点としては、文字コードを「UTF-8」に

すること、拡張子を「.html」にすることです。

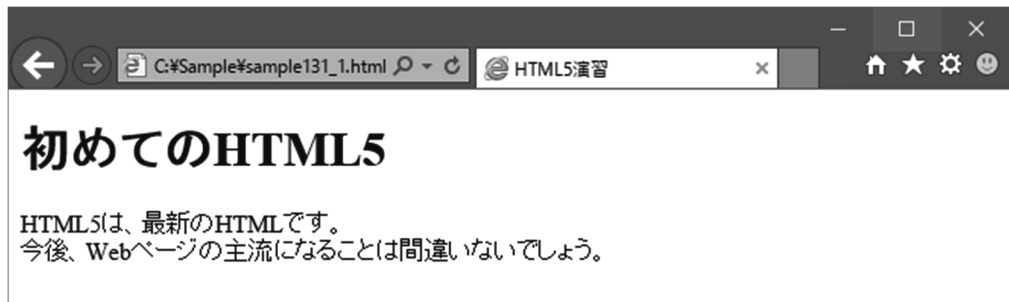
One Point

文字コードとは、コンピュータが文字を識別するために各文字に割り当てられる番号やその対応関係のことです。

日本の文字コードには、ISO-2022-JP、EUC-JP、Shift_JIS、UTF-8 などがあります。

たとえば、ブラウザが Shift_JIS だと判断して表示した Web ページの文字コードが、実は UTF-8 だった場合、文字とその番号が異なるため「文字化け」の状態になります。

保存できたら、ブラウザで表示します。ブラウザを起動したらアドレス欄に、保存した場所とファイル名(パス)を入力します。Windows ならエクスプローラーなどで、ファイルを直接ダブルクリックして開くこともできます。



Internet Explorer 11 で表示した例

上の図のように表示できれば、HTML を勉強する準備は完了です。今後、本書で新たな機能を紹介したときは、上記の手順で入力してブラウザで確認することができます。

1.3.2 DOCTYPE 宣言

「HTML を勉強するための準備」で入力した HTML ファイルを見ると、先頭行に次のような記述があります。

```
<!DOCTYPE html>
```

これを「^{ドクタイプ}DOCTYPE宣言」と呼び、この文書が HTML5 で作成されていることを表します。HTML5 以前の HTML、たとえば HTML 4.01 の DOCTYPE 宣言は、以下のような記述でした。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

上記の宣言には、HTML のバージョンやタグ定義が記述された文書を参照するための URL といった情報が記載されていますが、HTML5 の DOCTYPE 宣言には必要ありません。

通常、DOCTYPE 宣言は記述しなくても Web ページは表示されます。しかし、宣言がない場合、ブラウザが「互換モード」と呼ばれるモードで HTML を解釈して、CSS や JavaScript の表示や挙動が異なる場合があります。ですから、DOCTYPE 宣言は必ず付けたほうがよいといえます。

One Point

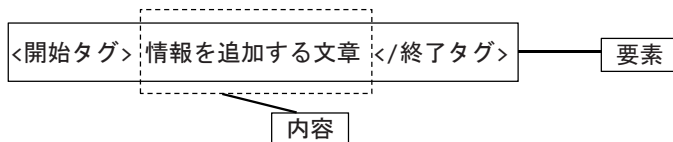
ブラウザが Web ページの HTML をレンダリングするとき、「標準モード」と「互換モード」のどちらかを使います。

「標準モード」では公式の CSS 標準に準拠した形で HTML と CSS が解釈されますが、「互換モード」ではそのブラウザが過去に定めていた独自のルールで HTML を解釈します。

互換モードがブラウザに用意されている理由は、CSS が標準化される以前の Web ページが、現在でも利用されている場合があるためです。

1.3.3 HTML のタグ

HTML のタグには「**開始タグ**」と「**終了タグ**」があり、意味や役割といった情報を追加したい文章を、この2つのタグで挟みます。この「開始タグ」から「終了タグ」までの部分を「**要素**」、タグに挟まれている部分を「**内容**」と呼びます。



開始タグの書き方は、まず半角記号「<」を書き、タグの名前(タグ名)を書きます。タグ名も半角で記述します。文字は大文字小文字どちらでもかまいませんが、通常小文字で記述します。開始タグは「>」で終了します。終了タグは「</」を記述してタグ名を書き「>」で終了します。

それでは、もう一度、サンプルの「sample131_1.html」を見てみましょう。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>HTML5演習</title>
  </head>
  <body>
    <h1>初めてのHTML5</h1>
    <p>HTML5は、最新のHTMLです。<br>
    今後、Webページの主流になることは間違いないでしょう。</p>
  </body>
</html>
```

このHTMLに登場する要素の意味は、次のようになります。

- **html 要素**

html 要素は、このドキュメントがHTML文書であることを表します。<html> ~ </html> は文書の「ルート」になります。「ルート」とは、「大元」とか「起源」「出身」といった意味で、ここからHTML文書の内容が始まります。html 要素の内容には、head 要素と body 要素が各1つ必要です。

- **head 要素**

^{ヘッド}
head要素は、ヘッダー情報(このページの基本情報)を表します。ヘッダー情報には、「meta要素」「title要素」などを含みます。

- **meta 要素**

^{メタ}
meta要素は、そのページがどんな性質でどのような値かという情報を設定します。文章に何か意味や役割を与えるものではないので、要素はタグだけを記述します。

このように開始タグ、終了タグの区別のない要素を「**空要素**」と呼び、「>」の前に/(スラッシュ)を記述します。ただし、HTML5では / を記述しなくても間違いではないため、本書では空要素に / は記述していません。

- **title 要素**

^{タイトル}
title要素は、文章のタイトルを表します。通常、title要素の内容は、ブラウザのタブやタイトルバーに表示されます。

- **body 要素**

^{ボディ}
body要素は、ブラウザのウィンドウ内に表示されるコンテンツを表します。body要素の内容には、見出し、段落、表、フォームなど、コンテンツを形作る要素を記述します。

- **h1 要素**

見出しを付ける要素です。要素は、レベルによって<h1>から<h6>まであります。h1要素は、一番上位のレベルになります。

- **p 要素**

p要素は、「段落」を表すために使います。したがって、段落以外の表現に使用すべきではありません。たとえば、p要素の終了タグの後ろにはスペースが生じますが、改行したい場合は、
タグを使うべきです。また、文章が段落ではなく「セクション(1.4 コンテンツ・モデルで説明します)」として区切られると考える場合は、セクション関連のタグを使います。

- **br 要素**

br 要素は、改行を表します。したがって、レイアウトなど改行以外の表現に使用すべきではありません。もし **P** 要素により段落の終端を表したのであれば、**br** 要素を入れなくても改行します。レイアウトには、**CSS** を使います。

One Point

要素同士には、「親子関係」や「兄弟関係」があります。

たとえば、ある要素の内容として別の要素が含まれる場合、それぞれの要素は親子関係にあると言います。

`sample131_1.html` を例にすると、**html** 要素の内容をとして **head** 要素、**body** 要素があることがわかります。この **head** 要素や **body** 要素から見ると、**html** 要素は「親要素」であり、**html** 要素から見た **head** 要素と **body** 要素は「子要素」です。そして、両者の関係を「親子関係」と呼びます。

また、**head** 要素と **body** 要素は、同じレベルの **html** 要素の子要素であることから「兄弟関係にある」と言います。

1.3.4 HTML の属性

HTML のタグには、「属性」のあるものがあります。属性とは、タグに追加する補足情報のことです。属性は「属性名」と「属性値」で構成されます。

属性の書き方は、属性名を記述し、「= (イコール)」と属性値を記述します。属性値は「(ダブルクォーテーション)」で囲みます。属性値以外は、すべて半角で記述します。

```
属性名="属性値"
```

属性をタグに指定するには、タグ名の後ろに半角スペースを記述して、属性を追加します。

```
<タグ名 属性="属性値"> 情報を追加する文章 </タグ名>
```

属性を複数指定する場合は、半角のスペースを入れ次の属性を記述します。

```
<タグ名 属性 1="属性値 1" 属性 2="属性値 2" . . . > 情報を追加する文章 </タグ名>
```

ここですべての属性を紹介することはできないので、「sample131_1.html」で登場した属性を紹介します。

- lang 属性

例：<html lang="ja">

ラング
lang属性は、文章がどの言語で記述されているかを指定します。「ja」は日本語です。その他に「en」英語、「zh」中国語、「ko」韓国語、「de」ドイツ語など多数あります。

- charset 属性

例：<meta charset="UTF-8">

キャラセット
charset属性は、このページの文字コード情報を表します。指定できる文字コードには「UTF-8」以外にも、「Shift_JIS」や「EUC-JP」などがありますが、HTML5 では文字コードに「UTF-8」を強く推奨しています。

注意する点としては、charset 属性で「UTF-8」を指定したのであれば、エディタでそのファイルを保存するときに、文字コードを「UTF-8」にしなければいけません。

その他の属性については、その都度必要に応じて紹介していきます。すぐにどのような属性があるのか知りたいときは、付録を参照してください。

One Point

以前の HTML では、表示位置を制御するために align 属性が使用されてきましたが、HTML5 では廃止されています。要素の表示位置は CSS で指定します。

One Point

HTML の属性には、ほとんどのタグに用意された共通の属性があります。この属性のことを「グローバル属性」と呼んでいます。

グローバル属性には、要素に対して識別子となる ID を設定する「id 属性」や、CSS で利用する class 属性などがあります。

確認問題**■確認問題 1.3 (1)**

以下のツールの中で、HTML ファイルを作成するのに最も適したツールはどれですか。番号で答えなさい。

- ① ブラウザ
- ② メールソフト
- ③ テキストエディタ
- ④ 紙と鉛筆

■確認問題 1.3 (2)

次の文章の空欄部分に当てはまる言葉を語句の中から選び、記号で答えなさい。

HTML のタグは、最初に「 ① 」、「 ② 」、「 ③ 」の順で記述します。この「 ① 」から「 ③ 」までの部分を「 ④ 」と呼びます。

語句

- ア. 要素
- イ. 開始タグ
- ウ. 終了タグ
- エ. 情報を追加する文章

■確認問題 1.3 (3)

次の文章の空欄部分に当てはまる言葉を語句の中から選び、記号で答えなさい。

HTML のタグには、「属性」のあるものがあります。属性とは、タグに追加する補足情報です。属性は「 ① 」と「 ② 」で構成されます。属性の書き方は、① を記述し、「=」と ② を記述します。② は「 ③ 」で囲みます。

語句

- ア. 半角スペース
- イ. 属性名
- ウ. ダブルクォーテーション
- エ. 属性値