

HTML の書き方

牧之内研 M1 龍 浩志

2001 年 5 月 10 日

1 WWW

WWW(World Wide Web)とは、インターネット上に構築された「ハイパーテキスト環境」で、これを構成する主な要素がHTML(Hyper Text Markup Language)文書と呼ばれるマルチメディア文書です。

WWWでは、URL(Universal Resource Locator)によってインターネット上の資源の種類と場所を指定します。URLの構成は次のようになっています。

`scheme://server[port]/[path]`

schemeは、資源の種類(接続プロトコル名)を示すもので、HTML文書を要求・応答するのはHTTP(Hyper Text Transfer Protocol)です。

WWWを閲覧するためには、WWWブラウザと呼ばれるツールが必要です。代表的なものとしてはNetscape NavigatorやInternet Explorerなどがあります。

2 HTML

HTMLは、テキストファイルの間にタグを挿入して作ります(例1)。タグとは、`<と>`の間に命令(directive)を書いたもので、通常2つが対になっています(例: `<H1>見出し</H1>`)。/`のついたものは範囲の終りを示します。終りを示すタグがないものもあります(エンプティタグ)。`

例1

```
<HTML>
<HEAD>
  <TITLE>Basic of HTML</TITLE>
</HEAD>
<BODY bgcolor="#ffffff">
  <H1>HTMLの基礎</H1>
  HTMLの基本的な作り方。 <BR>
  <UL>
    <LI><A href ="www.html">WWW</A>
    <LI><A href ="tags.html">タグについて</A>
    <LI>以下略
  </UL>
</BODY>
</HTML>
```

タグの記述は大文字・小文字どちらでもかまいません。`<title>`も`<TITLE>`も`<TitLe>`でも同じ意味です。また、ブラウザによっては解釈されないタグもあります。解釈されなかったタグは無視されます。

以下では、一般的なHTML文書で用いられる(または便利だと思われる)主なタグについて説明します。

3 基本的なタグ

3.1 ドキュメントの構成

以下の3つのタグは、ファイルの内容がHTML文書であることを知らせるための構造タグです。実際の画面には表示されませんが、ドキュメントの構成を知らせる重要なタグです。

- **<HTML></HTML>**:HTML ファイルであることを明示するタグ
ファイルの始めと終わりにおかれ、その中に書かれたテキストは **HTML** のファイルであることを示します。
- **<HEAD></HEAD>**: ドキュメントの情報記載部分
タイトルやデフォルト URL, 制作者の情報, 日付などブラウザには現れない情報を記入します (タイトルは見えますが)。
- **<BODY></BODY>**: ファイルの内容
<HEAD> タグに続いて記述し, ブラウザに表示される内容を表します。
また, この中で **bgcolor** や **text** を指定することで, 背景色や文字の色を変えることもできます (この指定をアトリビュート (属性) といいます)。

3.2 タイトル

タイトルは **<TITLE></TITLE>** で表します。このエレメントは必ず **<HEAD>** タグ内におかれます。

3.3 見出し

文章の先頭に書かれるような見出しを指定するのは **<Hn></Hn>** (Header) タグです。n には数字が入ります。ヘッダには 7 つのレベルがあり, **<H1>** が最大, **<H7>** が最小です。但し, 見出しとして使えるのは 3 か 4 くらいまでです (後は小さすぎ)。

3.4 リスト

番号無しのリストを作るのは **** (Unnumbered List) です。リストアイテムは **** の後に記述します。終りを示すタグはありません。番号付リストは **** (Ordered List) です。

また, リストはいくらでもネスティング (入れ子にする) が可能です。つまり, リストアイテム自体を更にリストにすることができます。

他に, 定義型リスト (Definition List: **<DL></DL>**) というものもあります。

3.5 改行

文章を改行したいときには改行したい場所で **
** (BReak) を使います。

また, **<P></P>** (Paragraph) でその中身を 1 つの段落として扱うことができます。当然, **</P>** の後で改行が行われます。

3.6 コメント

<!-- --> の中に文字を書くと, それはコメントとして扱われ, ブラウザには表示されません。ソースを読みやすくする為などに使われます。

3.7 水平ライン

<HR> (Horizontal Rule) で水平なラインが引けます。レイアウト上結構便利です。エンプティタグです。

3.8 センタリング

<CENTER></CENTER> で囲んだ範囲をセンタリングすることができます。文字だけでなく, 表や図もセンタリングできます。

3.9 引用文

<BLOCKQUOTE></BLOCKQUOTE> で, 囲んだ部分を前後の文章から独立させ, 且つインデントを揃えることができます。また, 右側にもスペースができます。

「引用文」と書いてはいますが, 普通に文章のレイアウトを整える際にかなり便利だったりします。

3.10 文字飾り

Web サイトでは文字の飾りつけが結構重要な要素だったりします。 で太字 (Bold), <I></I> で斜体 (Italic), <U></U> で下線つき (Underlined), <S></S> で取消線つき (Strike-through) の文字が書けます。

また, タグで文字の色 (color) や文字の大きさ (size, 1 ~ 7) を変えることもできます。

4 ちょっと複雑なタグ

4.1 アトリビュート

アトリビュートとは, タグの機能の微調整をしたり, 条件を付け足したりするものです。アトリビュートの書式は, 一般に以下ようになります。

<TAG attribute = 値>

値に数値以外のものを取るときは, ” (ダブルクオーテーション) で囲みます。

4.2 イメージの貼付

HTML でイメージ (図) を表示させるには (IMaGe) を使います。必ず src アトリビュートを伴います。src はソース (SouRCe) を表し, 表示させるイメージのファイル名を指定します。

他に, 以下のようなアトリビュートが使用できます。

- alt="テキスト ": イメージの代わりにテキストを表示させる (ALternative)
イメージが表示できなかった場合に表示されるテキストを指定します。
- width,height: イメージの大きさを変化させる
width で幅,height で高さを調整できます。値にはピクセル数または%を取ります。%はウィンドウの大きさに対するパーセンテージです。
- align: イメージとテキストの位置関係 (ALIGNment)
イメージの左右のテキストがどのように表示されるかを指定します。値にtop, center(middle), bottomをとった場合, イメージ横のテキストは1行になります。複数行を表示させるときは, 値をrightまたはleftにします。
- border: 枠を作る
イメージの縁に枠を作ります。値にはピクセル数を取ります。

4.3 テーブルの作成

テーブルとは表組のことです。テーブル作成を示すタグは<TABLE></TABLE> です。テーブルの横方向の枠を作るのは<TD></TD>(Table Data), 縦方向の枠を作るのは<TR></TR>(Table Row) です。通常, 以下のような書式になります。

```
<TABLE>
<TR>
<TD>1 段目第 1 列</TD><TD>1 段目第 2 列</TD>
</TR>
<TR>
<TD>2 段目第 1 列</TD><TD>2 段目第 2 列</TD>
</TR>
</TABLE>
```

そのほか, テーブルでは以下のようなタグ・アトリビュートが使えます。

- <TH></TH>: ヘッダをつける (Table Header)
<TD>の代わりに使うことで, 行や列の見出しを作ることができます。中身はセンタリングされ, フォントはボールドになります。

- `<CAPTION></CAPTION>`: タイトルをつける
テーブルの上部にタイトルを入れます。通常、`<TABLE>` タグの直後にいれます。 `align = "bottom"` でテーブルの下にすることもできます。
- `border`: 枠の太さ
`<TABLE>` タグの中で使います。 枠組みの先の太さを変更できます (ピクセル数)。
- `align, valign`(Vertical ALIGNment): 文字位置のコントロール
`<TD>`, `<TH>` タグの中で使います。 セル内の文字位置を調整することができます。 `align` で水平位置 (left,center,right), `valign` で垂直位置 (top,center,bottom) が調整できます。
- `rowspan`: 複数行にわたるセルを作る
`<TD>`, `<TH>` タグの中で使います。 値の数の行に跨るセルを作ります。
- `colspan`: 複数列にわたるセルを作る (COLUMN SPAN)
`<TD>`, `<TH>` タグの中で使います。 上と殆んど一緒です。

4.4 リンク

他のページなどへのリンクを貼るときには、`<A>`(Anchor) タグ+`href`(Hypertext REFerence) を使います。通常、次のような書式になります。

```
<A href = "ファイル名">テキスト</A>
```

ブラウザでは、タグに囲まれたテキスト(ホットテキスト)を選択することでファイルの場所へと飛びます。

指定するファイルは HTML ファイルに限りません。 イメージファイルやテキストファイルを指定すればそれ自体を表示させることができます。 また、ファイルのダウンロードを促すときにもこれを使います。 違うサーバのページ(もしくはファイル)にリンクを貼るときにはそのページ(ファイル)の URL を指定します。 “mailto:メールアドレス” でメールアドレスの URL 指定もできます。

他に、ファイルの途中へリンクを貼るための `name` アトリビュートもあります。

4.5 メタ情報

`<META>` タグで、テキストの著者や更新日、キーワードなどを記述する事ができます。 このタグは、`<HEAD>` タグの間に書き、ブラウザには直接表示されません。 必ずしも書く必要のあるタグではありませんが、幾つか重要な機能があります。

- テキストの更新日
キャッシュの更新に役立つ?
`<META http-equiv="Expires" content="Tue,18 Apr 2000 13:00:00 GMT">`
- URL 転送
指定秒数後に指定 URL へ画面更新します。 URL 変更などでよく見掛けます。
`<META http-equiv="refresh" content="5,http://www.db.is.kyushu-u.ac.jp/~hideo/index-j.html">`
- サーチエンジン
サーチエンジンのためのキーワード指定ができます。 最も一般的な用途といえます。
`<META name="keywords" content="MAPPLE,C++,JAVA">`
- デフォルト情報
デフォルトのスクリプト言語、スタイルシート言語、文書内の文字エンコーディング方法などを特定します。
`<META http-equiv="Content-Type" content="text/html; charset=EUC-JP">`

その他、様々ありますので調べてみてください。

4.6 その他

他にもいろいろありますが、例によって研究室にもたくさん参考書がありますのでそちらの方を読んでみてください。 もちろん Web 上にもその手の話はたくさんありますので探してみてください。

5 アップロード

作成した HTML ファイルは, そのまま手元に置いていても殆んど意味がありませんので, 通常は ftp など WWW サーバにアップロードします. こうすることで初めて万人がそのページを見れるわけです.

しかし, 研究室でサイトをつくった場合には, ftp を使う必要はありません. まず,

```
% mkdir ~/public_html
```

で, 自分のホームの下にディレクトリを作ります. そして, その中に index.html という名前で HTML ファイルを作ります. これでオッケーです. URL は

```
http://www.db.is.kyushu-u.ac.jp/~xxx/(xxx は自分のアカウント)
```

になります.

ちなみに, 研究室のページ (<http://www.db.is.kyushu-u.ac.jp/index-j.html>) からは index-j.html がリンクされるので, このファイルも用意した方が無難です.

6 スタイルシートと JavaScript

スタイルシートとは, フォント, テキスト, テーブルなどそのスタイルの指定をパターン化し, まとまった形でドキュメントに適応させることができるという機能です. これを用いれば HTML ページの体裁, テキストのフォントやサイズ, 色や背景, 配置などを整えることができます. 更に, スタイルシートはドキュメント全体や, ドキュメントのコレクション全体の表示特性を均一に指定する方法を提供します.

JavaScript は, ブラウザがユーザに対してどのように振る舞うかをコントロールするための機能とコマンドを持ったオブジェクト指向プログラミング言語です. ドキュメントとそれを表示しているブラウザをオブジェクトの集合としてとらえ, そのプロパティに対して計算をしたり変更を行ったりしています.

7 CGI と SSI

CGI(Common Gateway Interface) とは, 外部のアプリケーション (CGI スクリプト) で生成した情報を Web サーバ (httpd) 経由でブラウザに送るためのインターフェースの総称です. 通常, Web サーバでは静的な情報 (イメージやテキスト) をブラウザに送信しますが, CGI を用いれば静的な情報だけでなく CGI スクリプトで生成した動的な情報をブラウザに送ることが可能になります.

CGI はブラウザで入力した情報 (フォームやタグなど) を CGI スクリプトに渡すことができます. CGI スクリプトは渡された情報を加工したり, 他の情報を元にして新しいデータを生成したりすることができます. ここで生成されたデータは, HTML ファイルやイメージファイルと同じようにブラウザに表示されます. つまり, CGI を使うことでユーザのアクションによって動的に変化するドキュメントが作れるのです.

SSI(Server Side Include) とは, HTML ファイルの中に SSI コマンドを書いておくことで, サーバ側でそのコマンドが実行され, 結果がブラウザに表示されるという仕組みです. この SSI を用いれば, 現在の時刻やどのドメインからアクセスしているのかが判ります.

8 XML

XML(eXtensible Markup Language) とは, 主にインターネット上で文書やデータを交換したり配布したりするためのマークアップ言語です. HTML では, 決められたタグを使って文書を表現しますが, XML では, タグによってデータを表現し自由にそのタグを定義することができます. XML はデータの構造を記述したもので, レアウトの情報は持っていないので XML を表示させるにはスタイル指定言語が必要になります. ブラウザで XML 文書を表示するには, スタイル指定言語を用いて HTML 形式に変換することによって表示可能となります. Internet explorer5.0 以上のブラウザではこれらの変換を内部的に行っており, XML を表示することができます.

9 課題

今日の課題は「自分のページをつくる」です.

どんなものでも構いませんが, 少なくともトップページ, 自己紹介, その他の 3 ページ程度作ってみてください. また, できるだけいろんなタグを使ってみてください.

作ったものは, 先に述べた方法で自分のホームの下に置いて確認できるようにしてください. 研究室の本や, 研究室のページ (ブラウザメニューの「表示」 → 「ソース」でそのページのソースを見ることができます) など参考にしてみてください.