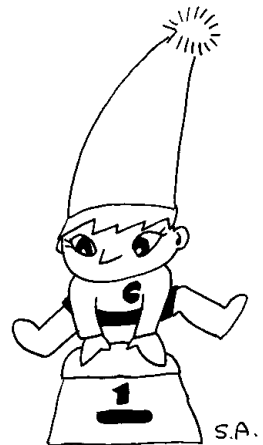


第1章

タグがたくさん

まずはHTMLから始めよう。この本の主題はウェブ（Web）用の「プログラミング」（プログラムの作成法）だが、これにはHTMLの知識が欠かせない。必要なところを手早く見ていくことにする。文章中心のウェブページなら、この章を読めば十分実用的なものを作ることができるようになるはずだ。次章以降でも、作成するプログラムとの関連で必要なHTMLの要素を少しずつ追加していく。また、すでにHTMLを知っている人も、プログラミングの予行演習という観点から、この章にざっと目を通して欲しい。



タグが肝心

何はともあれ、例を見てみよう。

プログラム例1-1 簡単なHTMLの例1 amu.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>拙者はハムスターでござる</title>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
</head>

<body>
<h1>ヒマワリは体にワリい</h1>
<p>
拙者はハムスターでござる。もう名前はあって、アムという。
なぜアムと言うかという、主人がアムなったらとかいう歌手が好きで、
そこからこの名前をとったようである。
自分の名前を自分で決めることはできないのが世の定めであるから、
好き嫌いとはかく、拙者もこの名前を受け入れて使っている。
</p>
<p>
この場を借りて、ひとつ、みなさんに訴えたいことがある。
拙者の仲間に関して「大好きなのはXXXXの種〜」
などという歌があるようじゃが、あれはコレステロールのもとじゃ。
長生きするためには控えるに限る。
嘘だと思ふなら、動物病院に行って訊いてみたまえ。
異口同音に「あの歌は困ったものです」と言っておる。粗食が一番じゃ。
粗食のアムスター、ではなくハムスターは長生きをしとるぞ。
たぶん、人間も同じじゃろう。
</p>
</body>
</html>

```

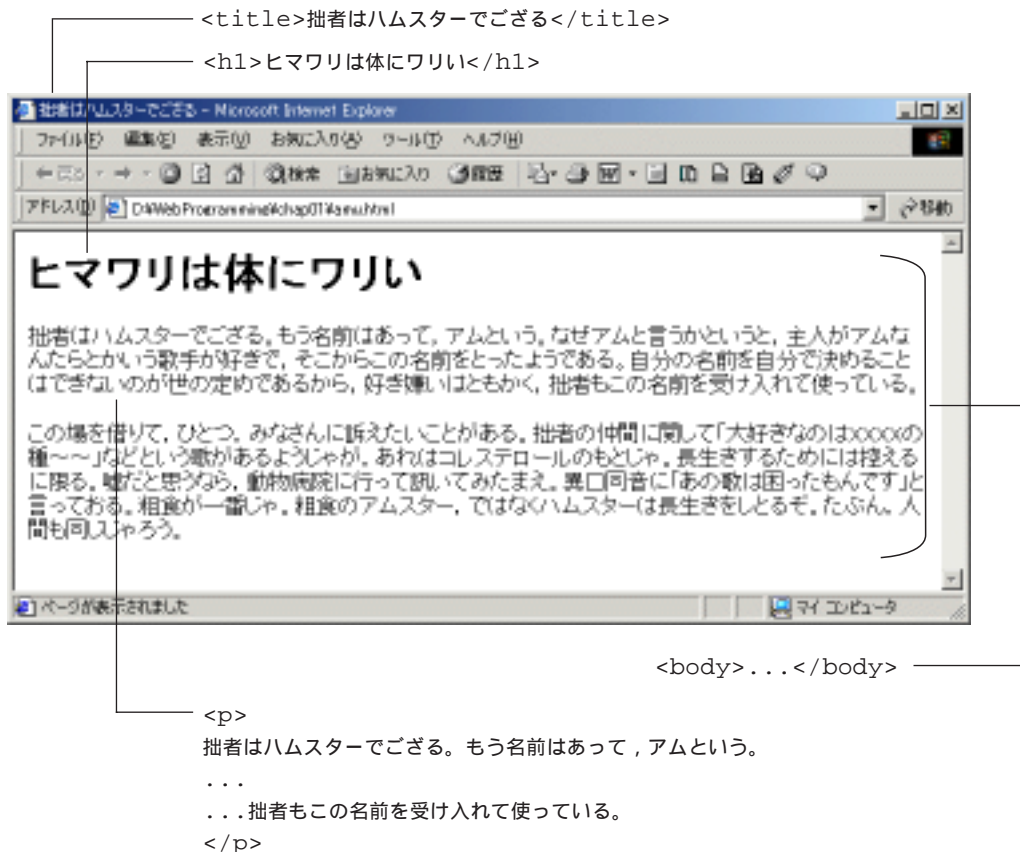
普段使っているワープロやエディタを起動して、この「HTML文書」を入力し、たとえば、amu.html といった名前で「テキスト形式」でファイルに保存してブラウザで表示すると〔脚注〕、図1-1のように表示される。これは、ウィンドウズ Windows用のInternet Explorerで表示したところだが、別種のコンピュータの別のブラウザ、たとえばMac OS XのiCabで表示しても、ほぼ同じように表示される(図1-2) マックオーエステン アイキャブ。

〔脚注〕本書のホームページにもおいてある。http://www.musha.com/からリンクをたどって欲しい。

メモ

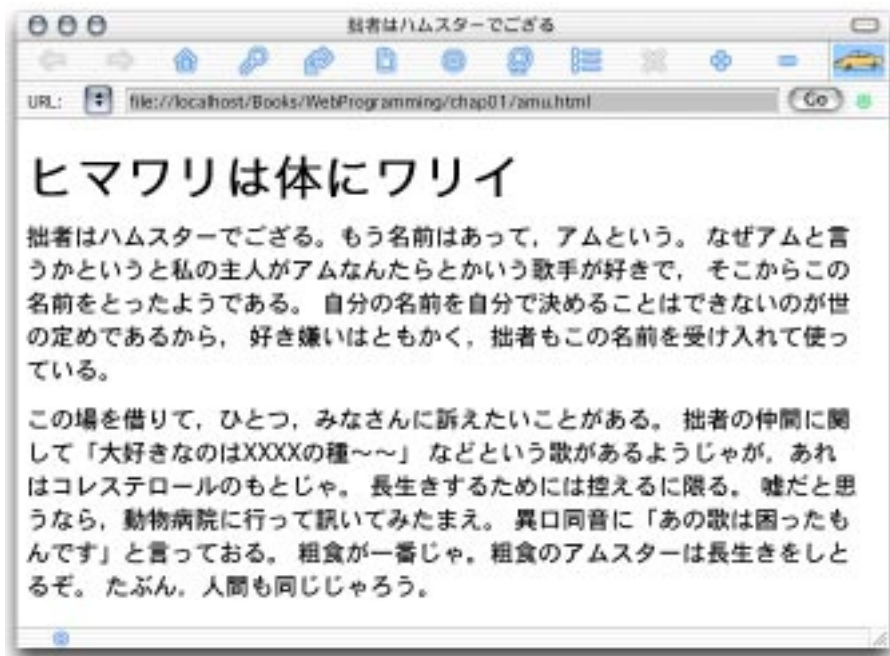
自分で作ったファイルをどうすればブラウザで表示できるかわからない人は、筆者のサイト <http://www.musha.com/> にある本書のホームページで解説しているので参照して欲しい。これ以降も、ほとんどの読者が知っていきそうな事柄や、時間の経過とともに大きく変わってしまいそうな事柄については、本文中では詳しい説明はせずウェブページで説明する。**[WebPage]** のマークを付けておくので、必要に応じてホームページを参照して欲しい。そうはいても、いきなり「ウェブページを参照」では、くじけてしまう人がいると思うので、この**[WebPage]**だけは付録Aに収録しておいた。情報などが古くなってしまっている可能性があるため、付録の説明でうまくいかなかったらウェブページを参照して欲しい。

図1-1 WindowsのInternet Explorerによる表示



HTMLの文書では「タグ (tag)」が重要な働きをする。どんなブラウザでも同じように表示されるための、中心的な役割をしている。タグは、その名前 (英語名) を「<」と「>」で囲んだ形をしており、このタグを見てブラウザが文字や画像などをウィンドウ内にどう表示するかを決める。上の

図1-2 Max OS XのiCabによる表示



例は短いですが、それでも<html>、<head>、<title>、<meta>、<body>など、たくさんのタグが使われている。

トレーナーやTシャツなど首のところに「タグ」が付いている。タグを見ると大きさ(S, M, L, XL, XXL, ……)やメーカー、ときには洗濯の仕方などがわかる。同じように、文字や単語、文などにタグを付けて、これはタイトル、これは大きく、これは斜体で、などと示すわけだ。

多くのHTMLタグには「開始タグ」と「終了タグ」がある。終了タグは開始タグの名前の頭に「/」が付いたもので、開始タグと終了タグに囲まれた部分はそのタグが適用される対象となる。たとえば、開始タグ<title>と終了タグ</title>に囲まれた「拙者はハムスターでござる」という文字列が、タイトル(表題)というわけだ。例に出てくる、ほかのタグについても簡単に説明しておこう。

```
<!DOCTYPE ...>
```

これは実はHTMLのタグではなく、DOCTYPE宣言と呼ばれるもので、この文書がどのような規格に基づいている文書であるかを示している。記述しなくてもブラウザは表示してくれるが、この宣言がある方が「正式」なので、将来を考えると書いておく方がよからう。

```
<html>...</html>
```

ファイルがHTMLで書かれた文書であることを示すタグ。HTMLファイルは、通常、その先

頭のDOCTYPE宣言に続き<html>タグが来て、最後は</html>で終わる。

<head>...</head>

「ヘッダ」部分。...の部分には、本文として表示されるのではないが、この文書に関する事柄を記述する。たとえば、ページのタイトル（表題）などだ。

<title>...</title>

...の部分に、この文書のタイトルを書く。通常はブラウザウィンドウのタイトル部分に表示される（図1-1, 1-2を参照）。上に書いたように、<title>...</title>は「ヘッダ」に入るので、<head>...<title>...</title>...</head>と並ぶことになる^[脚注]。ここに指定した文字列は、「お気に入り（ブックマーク）」に入れたときに表示されるし、また検索エンジンはこれを見てページの内容を判断することが多い。したがって、この部分は簡潔に、それでいて内容がよくわかる文字列を書くようにすること。

<body>...</body>

...の部分にブラウザ画面に表示される文字や画像など本体（コンテンツ）を書く。

<h1>...</h1>

見出し（ヘッダ）。見出しのレベルには<h1>...</h1>、<h2>...</h2>、.....、<h6>...</h6>と6種類あって、<h1>が一番の大見出しになる。この例では、一番上のレベルの見出しだけを使っている。

<p>...</p>

普通の段落（パラグラフ）。...の部分に書かれた内容が、ひとつの段落としてまとまって表示される。終了タグ</p>は書かなくても次の開始タグ<p>で段落の開始はわかるが、今後の新しい規格への対応を考えると</p>も省略しない方がよい。

上の例のように、見出しと普通の段落からなる文章を書きたい場合は、<body>と</body>のあいだに、見出しタグ（<h1>から<h6>まで）と<p>タグを使って次々と段落を追加していけばよいわけだ。

属性と属性値 <タグ名 属性1="属性値1" 属性2="属性値2" ...>

さて、ひとつだけ説明をとばしたタグがある。<meta>タグだ。このタグには終了タグはない。このタグ単独で（対象の文字列なしで）役目を果たす。そのかわりに「属性」が指定されている。<meta>タグにはhttp-equiv, contentなどの属性があり、それぞれの属性に「値（属性値）」を指定することによって、この文書の性質をブラウザに知らせて、ブラウザによる処理を助

[脚注] このように、タグの内側にさらにタグが現れることを「タグが入れ子になっている（ネストされている）」という。この「入れ子」という概念は、これから何度も登場する

ける役目をする。次のタグによって、このファイルの内容 (content属性) はテキストで ("text/html;"), 日本語の「シフトJIS」コードで書かれている (charset=Shift_JIS") ことを指定しているのだ (UNIXなどを使ってEUCコードでファイルを保存する場合は「Shift_JIS」のかわりに「euc-jp」を指定する。「_」と「-」に注意すること)。

```
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
```

この指定がないと、ブラウザの設定によっては日本語が正しく表示されないで、いわゆる「文字化け」を起こすことがある。日本語の文書では、この行はこのまま入れるようにしよう。属性値は必ず一組の引用符 (" "... " または ' '... ' ') で囲む。囲まなくても有効になる場合はあるが、正式には必ず囲むことになっているので、そうしておいた方が将来苦勞しなくてすむだろう。

メモ

タグや属性に大文字と小文字の区別はない。つまり、<h1>と書いても<H1>と書いても同じだ。ただし、XMLというこれから広まりそうな標準に則った書式で書く場合には、タグは小文字にする必要がある。このため、これから作るウェブページでは小文字のタグを使うことを勧める。以降、本書でも、タグや属性は原則として小文字で書く。

メモ

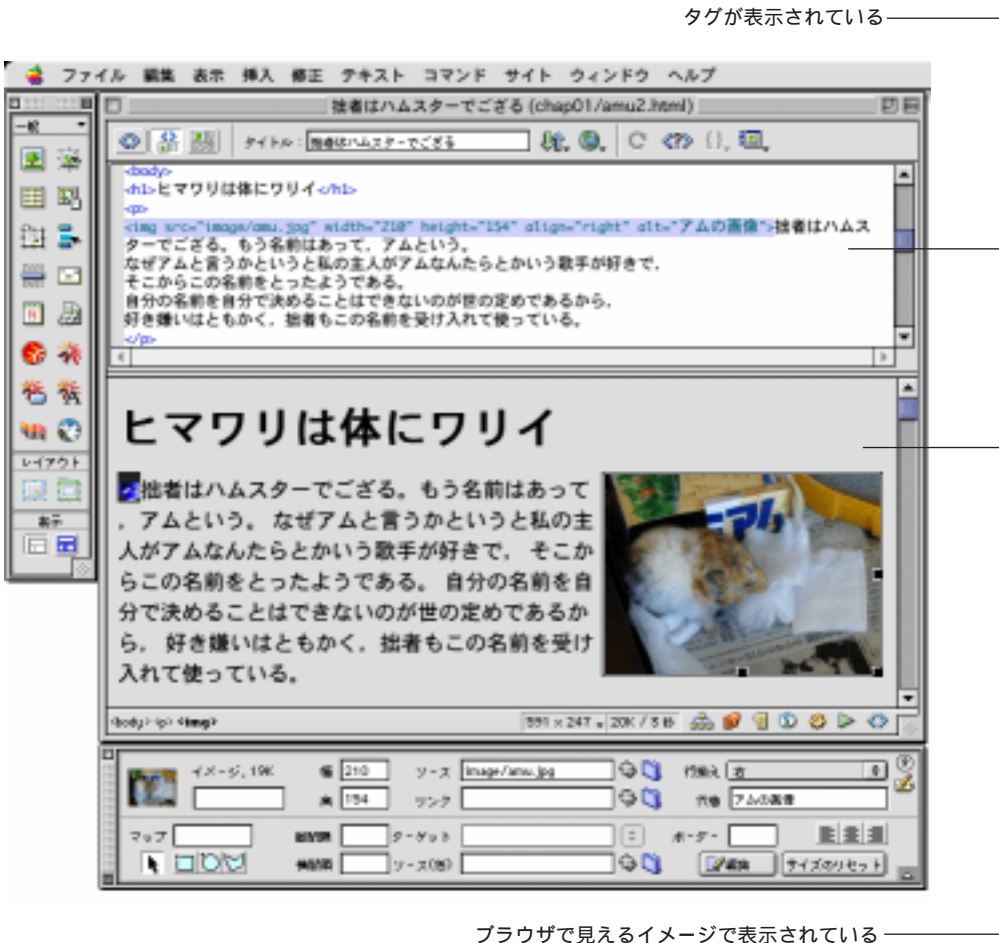
ブラウザごとに対応しているタグは異なる。ブラウザが対応していないタグは無視される。タグがなかったものとして扱われるわけである。したがって、無視されたタグに囲まれた文字は画面に表示はされるが、それがタグで意図したとおりに表示される保証はなくなる。ただし、本書で紹介するタグはここ数年流通しているほとんどすべてのブラウザでサポートされているので、この問題は起こらないはずだ。ほかで調べたタグを使うときには、そのタグにほとんどのブラウザが対応しているか意識して使って欲しい。

ウェブエディタの利用

ホームページが世の中に広まり始めた頃は、ほとんどの人が直接タグを使ってHTML文書を作成していたが、最近では「ウェブエディタ」あるいは「HTMLエディタ」と呼ばれるプログラムが使われるようになった。Adobe Golive、アドビ ゴーライブ Macromedia DreamWeaver、マクロメディア ドリームウィーバー Microsoft FrontPageなどなどたくさんの種類があるが、ウェブエディタを使うと、タグを使わずにワープロソフトと同じようにレイアウトや画像を扱える (図1-3)。

ウェブエディタだけで話がすむのなら、タグを覚える必要はないが、ここでタグの説明をしている

図1-3 ウェブエディタの画面 (DreamWeaver)。HTMLの「ソース」を見ながら、ワープロのように編集できる



のは、次のような理由があるからだ。

ウェブ関連のプログラミングをするには、タグを使ってウェブページを操作することがどうしても必要だ。

ウェブエディタを使っても、使い方を誤るととても読みにくいページができあがる。基本的なタグを理解していると、おかしな部分を修正できる。

画像の挿入や位置指定、表 (テーブル) の作成、細かなレイアウトの指定などを行うには、ウェブエディタがとても便利で効率的だ。そこでお勧めは、一応タグの働きを理解したうえで、ウェブエディタを利用するというものだ。そうすれば、両方の「イイトコ取り」ができる。

タグと表示の関係

では、HTML文書についてのまとめをかねて、別の角度からタグの特徴を整理してみよう。

HTML文書は、ブラウザのウィンドウに表示される「文字（テキスト）」と、文字が文書中でどのような意味を持つかを表す「タグ」から構成される。

タグが文書の構造を決める。タグの適用対象となる文字列が、「見出しの役目をする」とか、「普通の段落をなす」とかをタグによって指定する。

タグは対象となる文字列の文書中の役目を表すが、それがブラウザによってどのように表示されるかはタグ自体では決まらない。表示を決めるのはブラウザの役目である。たとえば「<h1>...</h1>」の部分は、太字の大きめの文字列にする」「<p>...</p>」で囲まれた段落はあいだを1行あけて表示する」といったことを決めているのはブラウザなのだ^[脚注]。

スペースやタブ、改行はどこにいくつ入れても表示結果は（ほぼ）同じになる。タグが文字の文書中の意味を決めるので、それに従ってブラウザがレイアウトをするからだ。このため、字下げ（インデント）や空行（何も文字が書いてない行）などを用いて、HTMLの文書を読みやすくすることができる。

落とし穴

ここでいう「スペース」とはいわゆる「半角」のスペース（仮名漢字変換をオフにした状態でスペースバーを押したときに入力される文字）だ。いわゆる「全角」のスペース（文字の幅が漢字と同じ空白文字）は漢字や平仮名などと同じように扱われ、その分のスペースが表示されてしまうので、ここでいう字下げの目的には使えない。半角のスペースと全角のスペースは文字コードが異なり、まったく別物として扱われる。

第2章以降でプログラムを入力する際にもスペースを使って字下げなどを行うことができるが、この場合も使えるのは半角のスペースだ。全角のスペースを使うとそれは空白としては扱われず、多くの場合エラーとなる。見ただけではわからないので、入力時に気を付ける必要がある。

では、ここまでに出てきたタグを使ってもう少し長い例を見よう。ちょっとしたエッセイを書いたら、こんな風にして、ホームページに公開することができる（図1-4）。ヘッダ部分（<head>...</head>）と本体部分（<body>...</body>）のあいだに空行を入れて、区切りを明示してみた。

[脚注]「スタイルシート」という仕組みを使うと、ブラウザに任せきりにしないで、ウェブページを書く人が細かくレイアウトを決められる。本書のサポートページに例をあげた^[1]。

プログラム例1-2 簡単なHTMLの例2 essay1.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>温泉</title>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
</head>

<body>
<h1>温泉</h1>
<p>朝から晩までコンピュータの前に坐っているため、肩や首、腰などが慢性的にこったり痛かったり。そんな私にうれしい味方が現れた。近くに温泉がオープンしたのである。近くの駅から無料送迎バスがあり、家から30分少しで温泉に浸ることができるようになった。</p>
<p>最初に行ったのは7月末の日曜日。友人が「いい、絶対いい」と勤めていたのを思い出し、「今日、温泉に行こうか」と家族みんなで繰り出した。10:00発の送迎バスに乗り、20分弱で到着。「カラスの行水コース」という1時間以内のコースを選び温泉へ。東京近郊の温泉はどこもだそうだが、お湯の色はコーラのような色だ。でも、とても気持ちがいい。</p>
<p>完全にはまってしまって、週に1度のペースで行っている。最近多いのが夜のコース。仕事と食事を終えて、午後7時あるいは8時のバスに乗り、カラスの行水コースで1時間、家を出てから2時間半ほどで戻り、お酒を飲んで「お休みなさい」というコースだ。</p>
<p>銭湯に比べるとちょっと高めだけど、温泉宿に1回泊まりで行くお金があれば、こちらには10回は行ける。そして、この温泉は体にとてもいいような気がする。お近くの方、おすすめです。</p>
</body>
</html>

```

リンク

簡単なページはできたが、上の例ではまだ「^{ウェブ}蜘蛛の巣」ページとは呼べない。ページ間を蜘蛛の巣のように張られた「リンク」を追加しよう。リンクを張るには、次のように<a>タグを使う。

近くに温泉がオープンしたのである。

<a>タグに^{ハイパーリンク}href (Hyperlink REFerence) という属性を指定し、値としてリンク先のページのアドレス (正式にはURL ^{ユニフォーム}Uniform Resource Locator ^{リソース} と呼ばれる) を書く。上の例では、http://www.onsen.com/ という (架空の) ページにリンクしている。このURLをはじめ、属性の値は必ず前後を「"」または「'」でくくる。そして、リンク対象の文字列の後ろに終了タグを置く。このようにURLをフルに記述するリンクのことを「絶対リンク」と呼ぶ (図1-5)。

図1-4 ここまでのタグを使って書いた少し長めのページ

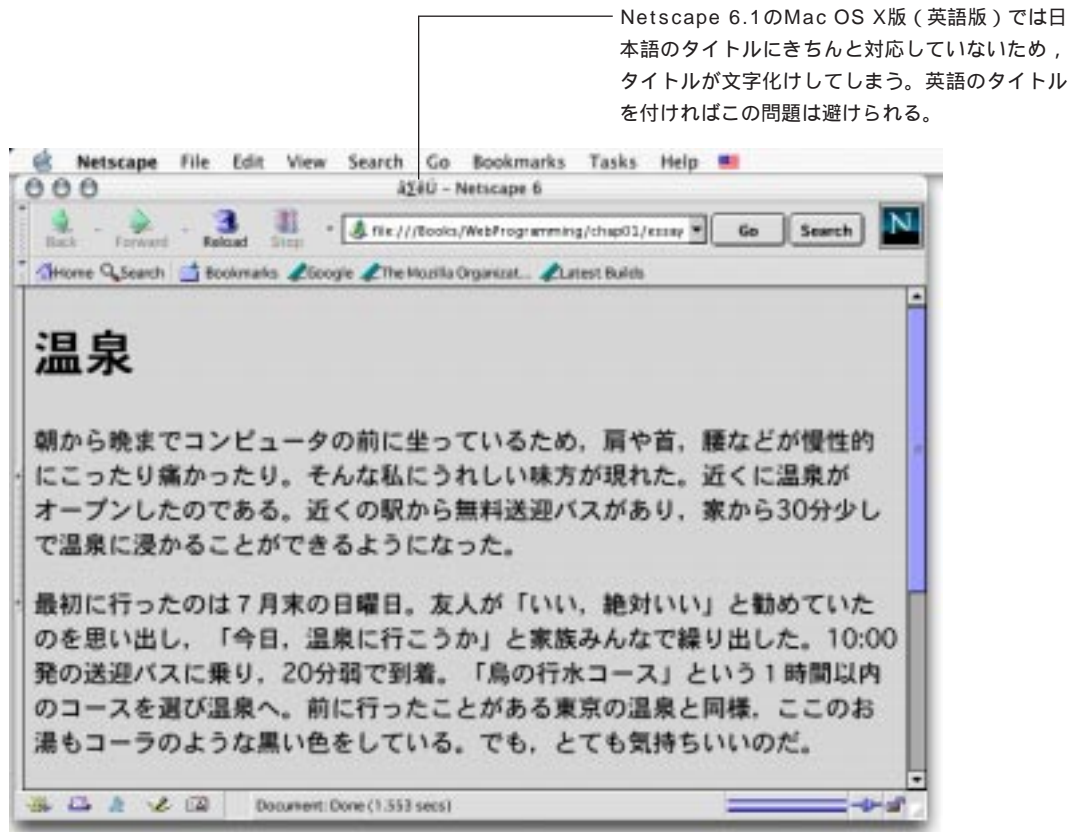
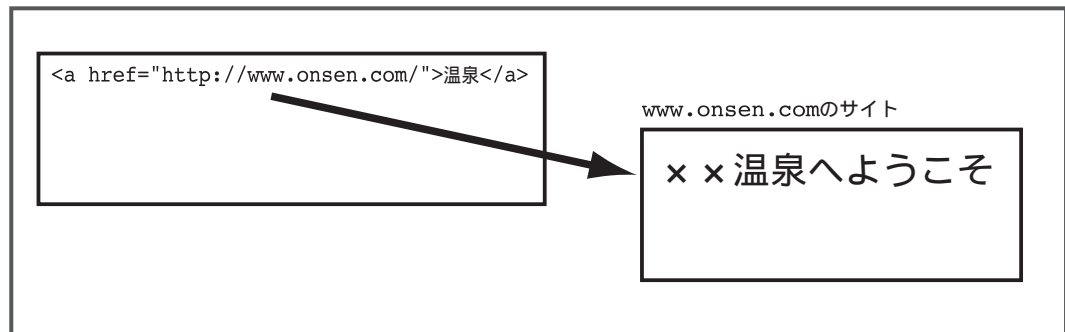


図1-5 リンク



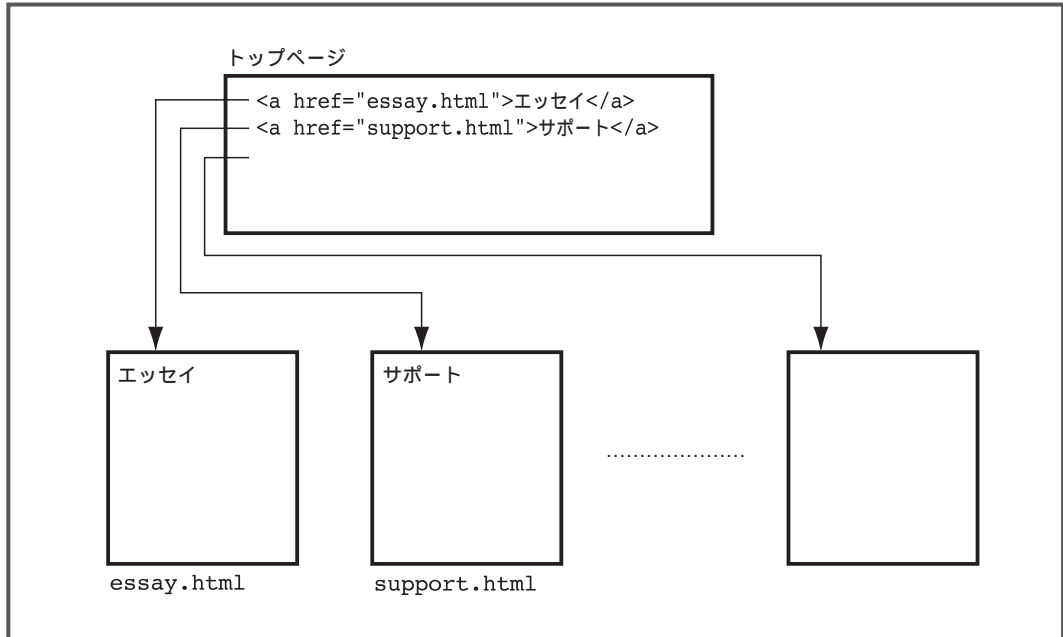
メモ

<a>タグのaはAnchor（錨の意味）なのだが、なぜanchorなのかは、以下で説明する。

サイト内のリンク 相対リンク

さて、「温泉」を含めてエッセイが3本ほどたまったので、まとめて「エッセイのページ」を作ることにした。このページのほかにも、いくつかページを作ることにしたので、玄関役をするトップページ^[脚注]も作り、「エッセイのページ」からトップページへのリンクも作ることにする。トップページの名前は、慣習に従ってindex.htmlという名前だ。図1-6のような構成にしよう。

図1-6 トップページとエッセイページの関係



次に「エッセイのページ」内の構成を考えよう。3本ただらだと並べると読みにくいので、最初に目次を付け、横線を引いて区切りを付けたり、目次に戻るようリンクを付けたりする。そして、玄関役のトップページへのリンクも付けよう。HTMLで書くと次のようになる。

[脚注] 「ホームページ」は、もともとインターネットで公開しているページの玄関役をする、一番上のページのことを指しており、普通のページを含めインターネット上に公開されているページのことを「ウェブページ」と呼んでいた。しかし、世の中に広まるにつれ、「ホームページ」が玄関だけでなくどのページについても使われるようになった。そこで、玄関役のページは「トップページ」と呼ばれることが多い。

プログラム例1-3 エッセイのページ `essay3.html`

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<!-- essay3.html -->
<html>
<head>
<title>Essay</title>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
</head>
<body bgcolor="#73BEC2">

<a name="top"></a><h1 align="center">エッセイ?</h1>
<h2>目次</h2>
<ul>
  <li><a href="#bus">バスと五円玉</a> (2001年10月12日)</li>
  <li><a href="#night">夜の十分間</a> (2001年10月2日)</li>
  <li><a href="#onsen">温泉</a> (2001年9月14日)</li>
</ul>
<hr>

<h3><a name="bus">バスと五円玉</a></h3>
<p> 土曜の午後。家にランドセルをおくと、一キロ弱離れたバス停まで田舎道を歩いていく。小学校の三年か四年だった私は、軽い受け口を直す矯正治療に、毎週バスで歯医者に通っていた。一時間に一本しか来ないので、乗り遅れないよう十分前にはバス停に着く。いつもの車掌さんに軽くあいさつしてバスに乗り、行き先を告げて子供料金二十五円を渡し、切符を切ってもらふ。歯医者のある市の中心街へは二十分ほどでつく。</p>
<p>治療を終えてバス停に戻る。次のバスまで時間があるときには、停留所を二つ分、そう子供の足で十五分ほど歩く。そこから乗ると、五円安くなるのだ。二回歩けば自動販売機のジュース一杯分。ほかの医者に通っていた友達から教わった「必殺技」だった。</p>
<p>比較的すぐにバスが来たのだろう、五円の俟約をせずに歯医者の近くのバス停から乗った日のことだった。同じ停留所から、メガネをかけた、いかにも街育ちといった雰囲気少年が乗った。何となく気になって様子を見ていたのだが、出口付近に立ったまま、あいている椅子にすわろうとしない。</p>
<p>その子は、なんと二つめの停留所、つまり私が俟約のため歩いてくる停留所で降りてしまった。この行動は私の理解を超えていた。いったい全体、何でこんな短い区間をバスに乗るのだ!</p>
<p>五円玉がとてとても貴重だった、そんな大昔の思い出である。</p>

<hr>
<a href="#top">このページの最初に戻る</a> <a href="index.html">トップページに戻る</a>
<hr>

<h3><a name="night"></a>夜の十分間</h3>
<p>玄関の戸を開け外に出る。真っ暗闇の中、トイレを目指す。あと少しだ。手探りで裸電球のスイッチをパチッといれ、あたりが明るくなってホッと一息。シャーといい気持ち。</p>
<p>おかしい。何やら暖かいものが股間を走っている。しまった。またやってしまった。</p>

```

<p>私が小さい頃住んでいた家は、明治時代に建てられた古いもので、トイレが家の外にあった。暗闇に白い雪が舞うような寒い夜でも、ぬくぬくとした布団からやっとの思いで抜け出し、用を足しに行かなければならなかった。時間にすれば十分もかからないのだが、なかなか布団を出る決心がつかず最後の最後まで我慢して、やっとのことで起き出していくのが常だった。</p>

<p>意識があって起き出せばよいのだが、時として夢の中で布団から抜け出して、トイレに駆けつけるときがあった。ホッとしたのもつかの間、下半身に広がる冷たい感触で我に返り、ごそごそ起き出して始末をしなければならなかった。</p>

<p>小学生の時、使いを頼まれ、叔母の家に一人で泊まったことがあった。いつものように夜中に用を足したくなり起き出したのだが、寝ぼけていた私にはトイレの場所がわからない。家の中を、あっちへ行きこっちへ行きしているうちに、とうとう我慢ができなくなり、畳の上を水浸しにしてしまった。</p>

<p>それでも眠かった私は、そのまま再び布団にもぐりこみ、知らん顔をして眠り込んでしまった。半分眠りながら、叔父が起き出してきてあたりを見回していたような記憶が残っている。翌朝、何もなかったように朝飯をいただき、家に戻ったのだが、あの夜のことは実は今でも鮮明に覚えている。</p>

<p>今の私、夢の中でトイレに行くことはあっても、そこで用まで足すことはなくなった。だが、ふと気がつくと足がトイレに向いていることが多い。あの夜の十分間が夢だったら、もっと平和な日々が送れているのかもしれないのだが。</p>

<hr>

このページの最初に戻る トップページに戻る
<hr>

<h3>温泉</h3>

<p>朝から晩までコンピュータの前に坐っているため、肩や首、腰などが慢性的にこったり痛かったり。そんな私にうれしい味方が現れた。近くに温泉がオープンしたのである。近くの駅から無料送迎バスがあり、家から30分少しで温泉に浸ることができるようになった。</p>

<p>最初に行ったのは7月末の日曜日。友人が「いい、絶対いい」と勧めていたのを思い出し、「今日、温泉に行こうか」と家族みんなで繰り出した。10:00発の送迎バスに乗り、20分弱で到着。「カラスの行水コース」という1時間以内のコースを選び温泉へ。前に行ったことがある別の東京の温泉と同様、ここのお湯もコーラのような黒い色をしている。でも、とても気持ちいいのだ。</p>

<p>完全にはまってしまって、1週間に1度は行っている。最近多いのが夜のコース。仕事と食事を終えて、午後7時あるいは8時のバスに乗り、カラスの行水コースで1時間、少し送迎バスで待つけれど2時間半ほどで家に戻って、あとはビールでも飲んで「お休みなさい」というコースだ。</p>

<p>銭湯に比べるとちょっと高めだけど、温泉宿に1回泊まりで行くお金があれば、こちらには10回は行ける。そして、この温泉は体にとてもいいような気がする。お近くの方、おすすめですよ。</p>

<hr>

このページの最初に戻る トップページに戻る
<hr>

</body>

</html>

このHTMLコードをブラウザで表示すると図1.7と図1.8のようになる。

図1-7 エッセイのページ(最初の部分)

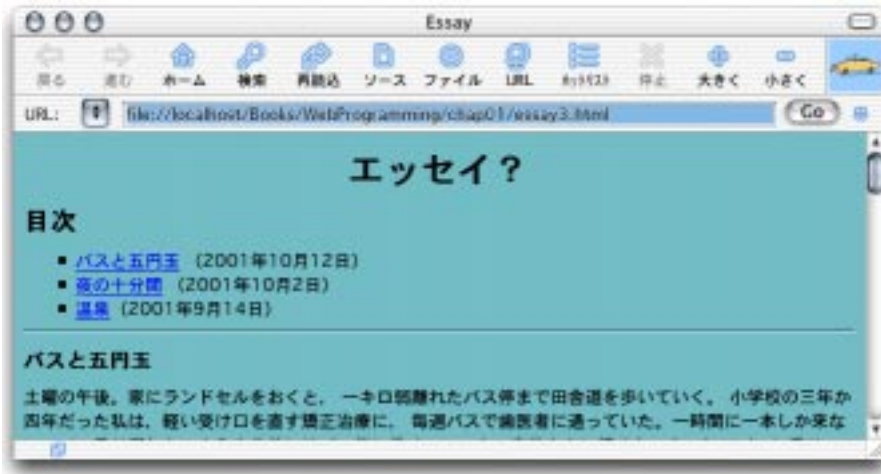
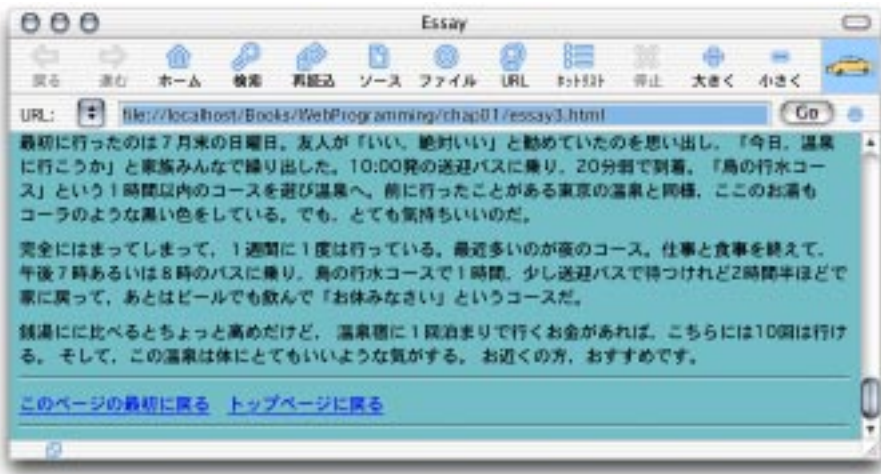


図1-8 エッセイのページ(最後の部分)



以下で、このHTMLコードに新しく出てきたタグなどを見よう。

コメント <!-- ... -->

<!-- ... -->で、...のところに「コメント」を書ける。たとえば、このファイルは essay3.html という名前で保存することにしたので、その名前をコメントとして書いた。そのほか、次のようなメモを書いたりしてもよい。

```
<!-- 後で、ここに追加する。候補は次のとおり。
<h2>韓国のバス</h2>
<h2>韓国の携帯電話</h2>
-->
```

また、後で説明する画像（イメージ）などを使って、複雑なナビゲーション（移動）用のリンクを作った場合には、どういう仕組みになっているか、コメントで解説をしておくともよいだろう。自分が後で読んだとき、他人が読んだときに、わかりやすくしておくことはとても大切な。

色の指定 <body text="#RRGGBB" bgcolor="#RRGGBB">

<body>タグに、text属性やbgcolor属性を指定することによって、文字（テキスト）や背景の色を指定できる。プログラム例1-3では次のようにして、背景色を青緑色にしている。

```
<body bgcolor="#73BEC2">
```

text="#00FF00"といったようにすると、文字の色も指定できる。ここで#73BEC2や#00FF00は、第0章で紹介したRGBという方式で色を指定したものだ。HTMLでは、#の後に16進数2桁ずつで赤、緑、青の成分の値を表す。使いたい色に対応する数値を調べるには、ウェブエディタを使うのが一番簡単だ。本書のホームページにも主なものを載せたので参考にして欲しい^[1]。

<body>タグのtext属性による指定ではすべての文字の色が変わってしまうが、個別の文字や単語の色を指定することも可能だ。ただし、色を乱用するとかえって読みにくくなるので、控えめに使うのがよからう。

位置指定 align="left" | "center" | "right"

`<h1>`タグはすでに説明したが、今度の`<h1>`には^{アライン}`align`属性が付いている。`align`は揃えるという意味で、右揃え、左揃え、中央揃え（センタリング）を指定できる。それぞれ、`align`の値として`"right"`、`"left"`、`"center"`を指定する^[脚注]。`align`属性を省略すると左揃えされる。

```
<h1 align="center"><a name="top"></a>エッセイ?</h1>
```

`align`属性は、`<h1>`から`<h6>`までや`<p>`などのほか、これから登場する`<hr>`など多くのタグでも指定可能だ。とりあえず、右揃えやセンタリングができそうな要素に対しては、`align="right"`などと指定して試してみてもよからう。

アンカー

`<a>`（アンカー）タグは前にも出てきたが、ここにはリンク先を示す`href`属性は指定されていない。

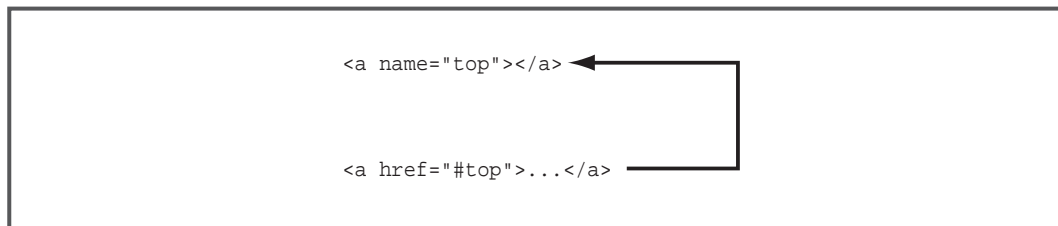
```
<a name="top"></a>
```

実はこちらの方が「アンカー（錨）」としての使い方である。このように、`name`属性で値を指定しておく、文書中の飛び先（リンクされる方）として使える。ここは飛び先を指定するだけで、開始タグ`<a>`のすぐ後ろに終了タグ``を書いてかまわない。リンク元では次のように指定する。

```
<a href="#top">このページの最初に戻る</a> <a href="index.html">トップページに戻る</a>
```

これは、各エッセイの下にあるナビゲーション用のリンクだが、「このページの最初に戻る」をクリックすると「同じページにある」`name="top"`と指定してある`<a>`タグの位置に移動する（図1-9）。

図1-9 <a>タグのname属性とhref属性の関係



[脚注] 節見出しにある「a | b | c」のように書いて、「aあるいはbあるいはcのいずれか」であることを表す。この例の場合、`align="left"`、`align="center"`、`align="right"`のいずれかの形式が使えることを意味する。コンピュータの世界で広く用いられている表記法だ。

「トップページに戻る」の方をクリックすると、こちらは（「#」が付いていないので）同じページではなく、このファイル（essay3.html）のあるフォルダ（ディレクトリ）内にあるindex.htmlというページが表示されることになる。

この2種類のリンクはいずれも同じサイトへのリンクを表すもので、http://で始まるURLではなく、リンクに対してそのページからの相対的な位置を指定するので「相対リンク」と呼ばれる。

リスト , ,

次に、目次の部分を見よう。

```
<ul>
  <li><a href="#bus">バスと五円玉</a> (2001年10月12日)</li>
  <li><a href="#night">夜の十分間</a> (2001年10月2日)</li>
  <li><a href="#onsen">温泉</a> (2001年9月14日)</li>
</ul>
```

この部分は、リスト（列挙）を表す。ulはunordered list（順序が付けられていないリスト、順番がとくに意味を持たないリスト）の略だ。番号が付かずに表示される。下の例のように、の代わりに（ordered list）を使うと、図1-10のように番号が付く。

```
<ol>
  <li><a href="#bus">バスと五円玉</a> (2001年10月12日)</li>
  <li><a href="#night">夜の十分間</a> (2001年10月2日)</li>
  <li><a href="#onsen">温泉</a> (2001年9月14日)</li>
</ol>
```

プログラム例1-3では番号を付ける必要はないのでを使った。

図1-10 番号付きのリスト。を使う。各行の先頭に番号が付いている

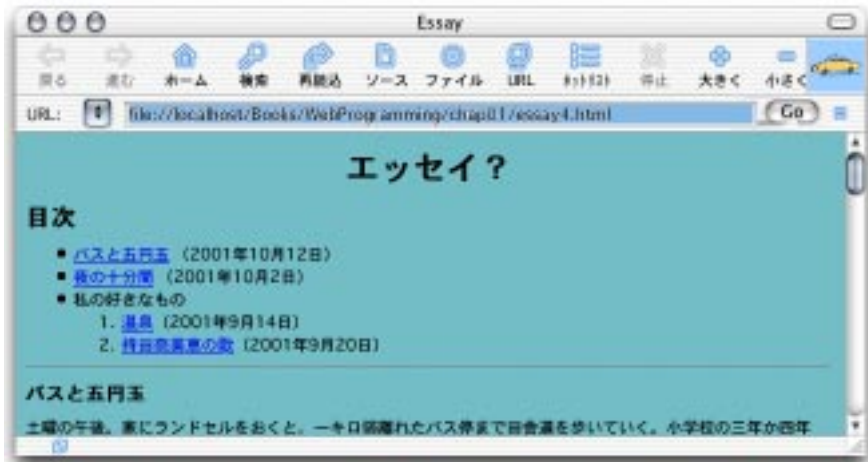


メモ

プログラム例1-3では、列挙される項目を空白文字を使って字下げ（インデント）して、構造をわかりやすくしている。下のようリストの中にリストを入れること（リストの「入れ子」）もできるが、このような場合はとくに字下げがないと、全体の構造がよくわからなくなる。

```
<ul>
  <li><a href="#bus">バスと五円玉</a>（2001年10月12日）</li>
  <li><a href="#night">夜の十分間</a>（2001年10月2日）</li>
  <li>私の好きなもの
    <ol>
      <li><a href="#onsen">温泉</a>（2001年9月14日）</li>
      <li><a href="#namie">持田奈美恵の歌</a>（2001年9月20日）</li>
    </ol>
  </li>
</ul>
```

図1-11 リストの入れ子



横線 <hr>

ホリゾンタル ルール

目次の下には横線（Horizontal Rule）が引かれている。

```
<hr>
```

横線は、視覚的に区切りを入れるのに便利だ。width="50%"のように幅を指定したり、size="4"のように太さの指定もできる。

メモ

HTMLにはHTML 2.0, HTML 3.2, HTML 4.0, XHTMLなどいろいろな規格があるが、この本では現在使われているほとんどすべてのブラウザで意図したとおりに表示できるページを作ることを念頭において説明する。たとえば、最新のHTMLの規格に厳密に従うと、対応する終了タグがないものについては<hr />のように閉じる前に「/」を書いて、この旨を表すことになっている。しかし、このような表記法を用いるとまだ使われているブラウザ、たとえばNetscape 4.7などで意図したとおりに表示できなくなってしまう。

これでプログラム例1-3に登場したタグはすべて見おわった。

画像（イメージ）

上の例ではリンクも出てきてだいぶウェブページらしくなったが、もうひとつ、ウェブページに欠かせない要素がある。画像（イメージ）だ。これは次のようにタグを使って指定する。

```
<p>

```

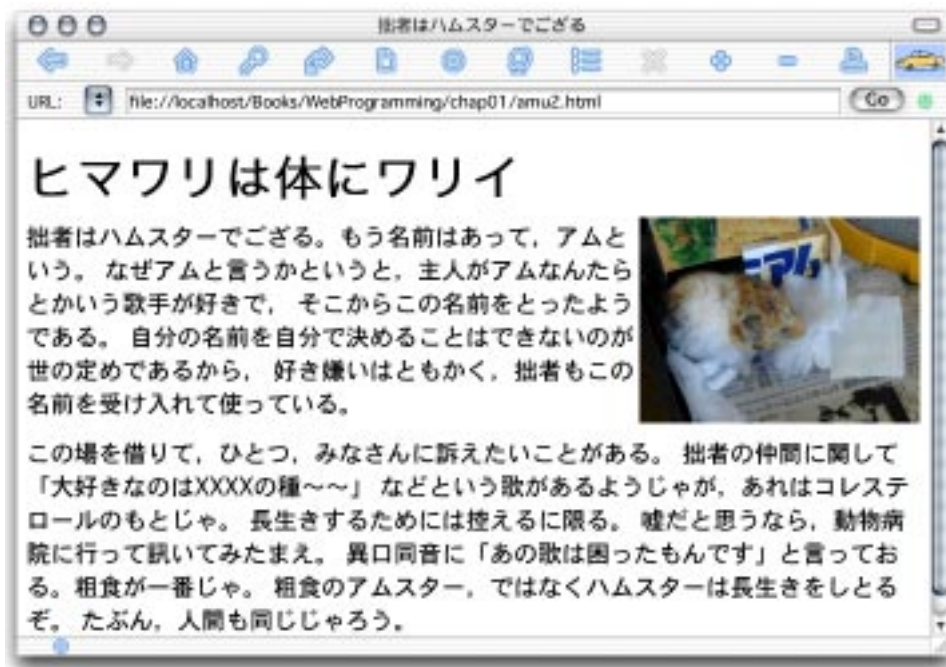
拙者はハムスターでござる。もう名前はあって、アムという。
なぜアムと言うかという私の主人がアムなんたらとかいう歌手が好きで、
そこからこの名前をとったようである。
自分の名前を自分で決めることは...

これをブラウザで読み込むと、図1-12のようになる。

タグにはいくつか属性が指定できるが、必須のものはsrc属性だ。ここではamu.jpgという画像ファイルを指定している。後の属性はすべて指定しなくてもよいもの（オプション）だ。

画像の大きさはwidthで横幅、heightで高さをいずれもピクセル単位で指定する。この大きさは元

図1-12 画像の指定



メモ

画像の指定にも相対指定と絶対指定がある。上のように指定した場合は、HTMLファイルを基準にしてひとつ下にあるimageというフォルダ(ディレクトリ)にあるamu.jpgというファイルを参照する(`src="image/amu.jpg"`)。ここにURLを指定することもでき(絶対指定)、その場合はそのURLにある画像が持ってこられて表示される。

この方法を使うと、いろいろなページ(たとえば好きな歌手のホームページ)にある画像をいかにも自分が作ったかのごとく自分のページで表示できてしまう。しかし、他人の画像や文書を自分のページで使うときには、了承を得て行わなくては行けない。著作権を侵害することになってしまい、罰せられる可能性もある。

の画像の大きさとは無関係に指定できる。逆にいうと、きちんと比率を保って指定しないと、ひしゃげたような画像になるわけだ。ひとつの画像を、いろいろな場所で大きさを変えて利用する場合などにwidthやheightを指定すると、異なる大きさの画像をいくつも用意する必要がなくなる。

widthやheightをまったく指定しないと、元の大きさのまま表示される。画像の大きさを指定すると、ブラウザにとっては画像を読み込む前にレイアウトが決定できるため、全体の表示を速くできる。したがって、大きさを指定する方がよい。先に触れたウェブエディタを使えば、通常、画像を

指定するだけで自動的にwidthやheightをタグの属性として付加してくれるので便利だ。

alt="..."で「代替文字列」を指定する。この文字列は、画像が読み込まれているあいだや、利用者が画像の表示をオフにしていたりする場合に表示される（最近はほとんど使われないが、画像表示機能のないブラウザでは、この代替文字列だけが表示される）。alt属性は省略可能だ。

画像をうまく使うとページの「かっこよさ」がだいぶアップする。上手に利用したい。

メモ

目の見えない人がホームページの内容を読み上げてもらうソフトを使うような場合を考えると、この代替文字列があることにより、どんな画像が置かれているかがわかる。できるだけ代替文字列は指定したい。

メモ

タグの属性には必須の属性と省略可能な属性がある。たとえば、タグのsrc属性（画像のファイル名やURLを指定）は必須属性だ。これに対してalign属性やwidth, heightなどの属性は省略可能だ。

まとめ

超特急でHTMLの主要なタグを見てきたが、いかがだろうか？ よくわからないところがあったら、実際に例題を表示し、自分なりにいろいろ変えてみて欲しい。次章からのプログラミングについても同様だが、自分で試してみないとなかなか身に付かないものだ。

このほかウェブページでよく使われる要素として「フレーム」がある。これについては、本文とは関係ないので[HTML4.01](#)に載せておくので、参考にされたい。

HTML文書の「型紙」

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>タイトル</title>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
</head>
<body>
  【本文】
</body>
</html>
```

HTMLのタグとDOCTYPE宣言

<code><!DOCTYPE ...></code>	常にHTMLファイルの先頭に入れる決まり文句
<code><html>...</html></code>	HTMLのコードはこのタグに囲まれる
<code><head>...</head></code>	ヘッダ部分
<code><title>...</title></code>	ヘッダ部分に書かれるタイトル
<code><meta ...></code>	文字コードなどの指定
<code><body>...</body></code>	HTMLの本体部分（ブラウザに表示される部分）。 <code>text="#RRGGBB"</code> <code>bgcolor="#RRGGBB"</code> で、文字色、背景色の指定も可能
<code><!-- --></code>	コメント

以下は`<body>...</body>`の中に書かれるもの

<code><h1>...</h1></code> ,, <code><h6>...</h6></code>	見出し
<code><p>...</p></code>	パラグラフ（段落）
<code>...</code>	リンク
<code>...</code>	番号の付かないリスト
<code>...</code>	番号付きのリスト
<code>...</code>	リストの項目
<code><hr></code>	横線（区切り線）
<code></code>	画像

いろいろなタグで使える属性

<code>align="left" ("center", "right")</code>	左寄せ（センタリング, 右寄せ）
---	------------------

日本語とブラウザ

本文中に「（HTML文書では）改行はどこにいくつ入れても表示結果は（ほぼ）同じになる」と書いていたが、日本語の場合、改行の位置によって表示が異なることがあり、ちょっとした問題をはらんでいる。

英語の場合ならば、次の2つの（とても短い）段落はまったく同じように表示される。

```
<p>I am a hamster.</p>
```

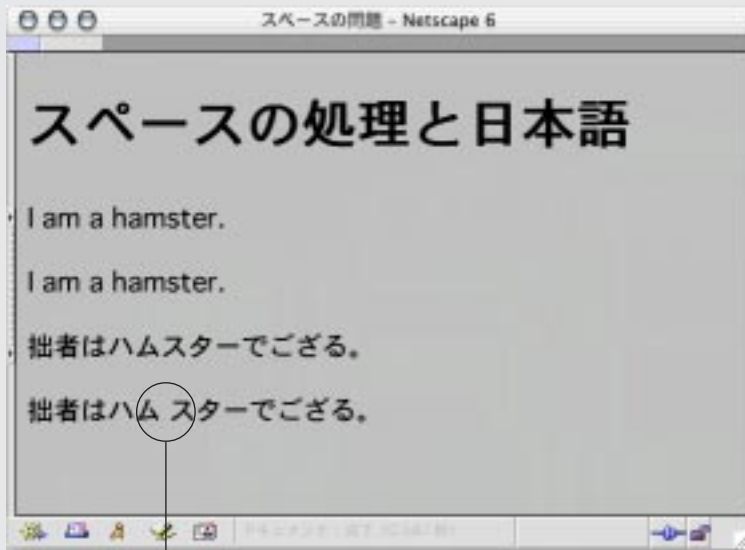
```
<p>I am  
a hamster.</p>
```

だが、次の2つの段落はほとんどのブラウザでは、表示が微妙に異なる（図1-13）。

```
<p>拙者はハムスターでござる。</p>
```

```
<p>拙者はハム  
スターでござる。</p>
```

図1-13 日本語とスペースの問題



「ハム」と「スター」のあいだの空白が気になる

（続く）

英語の場合、単語間は空白で区切られるが、日本語では普通分かち書きはしない。ところが、図1-13の一番最後の行では、「ハム」と「スター」のあいだに余分な空白が入ってしまっている。ブラウザが「HTMLの改行は、空白に置き換えて次の単語を表示する」という、英語用の規則を日本語にも適用してしまったわけだ。この現象は、ほとんどのブラウザで見られるもので、執筆時点で筆者の手元にある新旧併せて10近くのブラウザのうちでWindow版のInternet Explorerだけが、バージョン6になってようやく正しく、つまり空白をおかずに、処理をするようになった。

この問題を回避することはできる。改行の場所に気を付ければよい。最初の「拙者はハムスターである」のHTMLファイル（プログラム例1-1）を見て欲しい。改行を入れているのは、「`、`」や「`。`」など、その後に空白が入っても気にならないところだけにしているのだ。こうしておけば、変な空白は避けられる。

このほか、ワープロに入力するときのように、改行など入れずにひとつの段落を続けて入力するという回避方法もある。「温泉」（プログラム例1-2）の方はそうしている。こうすれば、そもそも改行がないのだから、これが余分な空白として残ることはなくなる。ただ、この方法にも問題は残る。ブラウザの[表示]メニューなどから[ソース]などの項目を選ぶと、そのページのHTMLソース（表示される文字だけでなく、HTMLのタグを含むファイルの中身全部）を見ることができるのだが、ワープロ式に改行なしで入力したソースは、多くのブラウザで見ると耐えない表示をされる。もともとの改行箇所ではしか改行してくれないので、長い段落を1行に書いてあると、はるか右までスクロールしなければ読めないのだ（図1-14）。

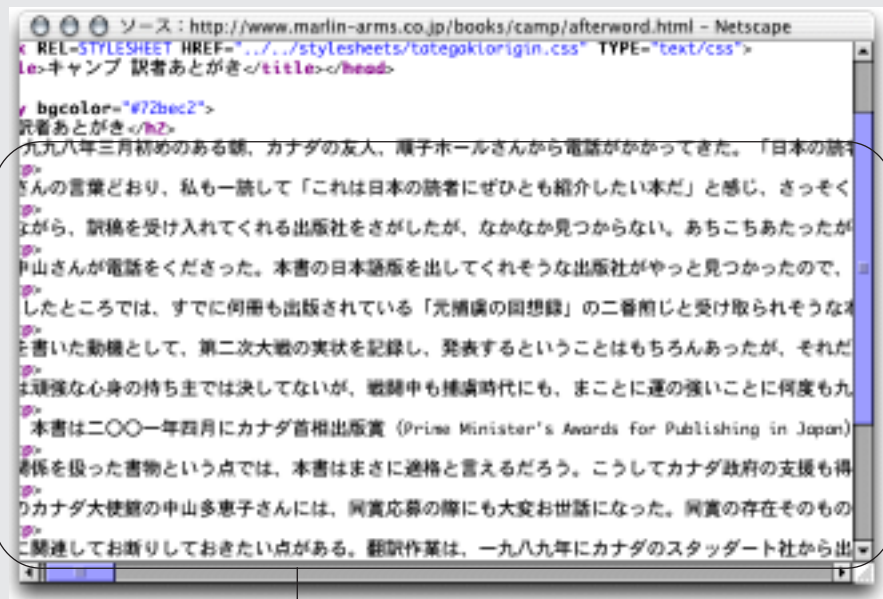
HTMLのソースを見る人はそれほど多くはないし、ファイルに保存してワープロやエディタで表示すれば、変な表示ではなくなるので、一般の目的には2番目の回避方法の方が手間がかからず、お勧めだろう。〔脚注〕

さて、なぜブラウザはこのような動作をするのであろうか。日本語など、分かち書きをしない言語のことをよく知らない人がブラウザを作っているからだ。そもそもウェブページというのは右回りに90度回転すれば、東洋で古来から使われていた巻物にとてもよく似ている。1枚1枚「ページ」を繰るのではなく、どんなに長い文書でもただただ左側に「スクロール」していけば読める。だから、ブラウザはとて東洋人向けのソフトウェアだと思う。しかし、日本語（や中国語など）を縦にも表示できる、巻物式のウェブブラウザで広く使われているものはまだ登場していない。

なぜ、巻物式縦書き対応ブラウザが出てこないのだろうか？ 日本人のソフトウェア開発のパワーが、足りないのだと思う。ごく一部の例外を除いて、多くのソフトウェアは欧米とくにアメリカからの輸入品だ。日本発のソフトウェアとしては英日・日英の機械翻訳プログラムや日本語文字認識など

〔脚注〕「空白があろうが、なかろうが気にならない」という人は、このような議論は無視してもよいが、あなたが作ったページを見た人が不快感をおぼえるのだから、回避するに越したことはない。

図1-14 ワープロ式でHTMLを入力したウェブページのソースを表示



スクロールしないと段落のすべてのソースを表示できない。

があるが、市場からみて外国との競争はほとんどない。世界中で流通しているような、日本発のソフトウェアはとも少ないのだ。どうしたらそのようなソフトウェアがどんどん出てくるようになるかは、大問題なのでまた別の機会にということにしよう。いずれにしても、本書を読んだ人の中から、そのようなパワーを持った人が生まれてくれれば、とてもうれしい。



