



# Web教科書 入門編

ネットワークの基礎からWeb制作まで

中村文隆 著

CQ出版社

# Web 教科書・入門編

中村文隆 著

はじめに	11
0.1 本書の目的と構成	11
0.1.1 WWWについて	11
0.1.2 本書の目的	12
0.1.3 本書の構成	14

---

## 第 1 部 基礎知識

---

第 1 章 Web の基礎知識	16
1.1 Web ページ作りに必要なもの	16
1.1.1 コンピュータとソフト	16
1.1.2 インターネットへの接続	17
1.1.3 Web サーバ	17
1.1.4 コンテンツと情熱	19
1.2 HTML の雰囲気	19
1.2.1 Web サーバのイメージ	20
1.2.2 HTML とブラウザ	21
1.2.3 いろいろなブラウザ	23
1.2.4 手間と結果のトレード・オフ	24
1.2.5 Web 演劇の「ト書き」 - タグ	26

第 2 章 コンピュータの基礎知識	30
2.1 コンピュータでの整理整頓 ファイル・フォルダ・ディレクトリ	30
2.1.1 コンピュータ上のファイル整理	30
2.1.2 パスの書き表し方 基本編	32
2.1.3 パスの書き表し方 応用編	33
ディレクトリツリー /	
カレントディレクトリと親ディレクトリの省略記法 /	
絶対パスと相対パス, ルートディレクトリ	

---

2.1.4	拡張子というもの	39
	拡張子の意義 / よく使われる拡張子の種類 /	
	拡張子の文字数についてのしきたり / Windows での拡張子の表示	
2.1.5	ファイルのプロパティ	45
	ファイルのサイズ サイズの一般論 /	
	ファイルサイズ 文字で考えてみる / 記録メディアの容量 /	
	テキストとバイナリ / 圧縮 / 圧縮法の雰囲気	
2.2	舞台裏 - コンピュータの基礎	53
2.2.1	コンピュータは電気で動く	54
	電気で表す 0 と 1 / コンピュータの動作原理 /	
	コンピュータの掛け声? クロック / 余談 - CPU クロックとバスクロック /	
	画面表示に関する補足 / コンピュータの動作手順書 - プログラム	
2.2.2	記憶装置 - データを保管する方法	62
	メモリ - RAM と ROM /	
	磁気記録装置 - ハードディスク ・ フロッピーなど /	
	光を利用した記録装置 - CD-ROM , MO など / まとめ	
<b>第 3 章 ネットワークの基礎知識</b>		<b>70</b>
3.1	インターネットとは	70
3.1.1	インターネットの仕組み	70
	インターネット / IP アドレス / IP アドレスによる通信 / ドメイン名 /	
	ドメイン名の規則 / DNS	
3.2	World Wide Web ( WWW )	82
3.2.1	URL について	82
	URL の例	
3.2.2	WWW	84
	Web のイメージ	

---

---

## 第 2 部 Web の作成と公開

---

<b>第 4 章 初めての Web ページ</b> .....	86
4.1  まずは文字だけで .....	86
4.1.1  最初のごく簡単なページ 87	
使用するテキストエディタ / 開始前の準備 / まずは挨拶ページから /	
ファイルとして保存する / ローカルで表示する /	
ファイルが見つからないとき	
4.1.2  実習の基本手順 98	
4.2  最初のページを理解する .....	98
4.2.1  タグについて 98	
まず観察する / HTML の骨組み / <P> は段落 /   は改行 /	
<HR> / ミスタイプ時の結果 / <H1> ~ </H1> ... <H6> ~ </H6> /	
4.2.2  HTML のバージョンと作法 107	
融通のきく HTML 書法 / 最大公約数的 HTML	
4.3  よく使われるタグ .....	117
4.3.1  文字の書式・装飾 117	
FONT , BASEFONT / I , EM , U , B , STRONG , S , STRIKE /	
SUB , SUP	
4.3.2  レイアウト 119	
CENTER / DIV / NOBR , WBR / PRE	
4.3.3  リスト 123	
OL , LI / UL , LI / DL , DT , DD	
4.3.4  その他 125	
META / !DocTYPE	
4.4  特殊文字 .....	129
4.5  文字コード 130	
文字コードが違おうと... / 文字コードとは / Web ページでの文字コード	
4.6  まとめ .....	133

---

<b>第 5 章 画像を使ってみる</b> .....	134
5.1 静止画とアニメーション	134
5.1.1 背景画像の前に背景色	134
背景の指定 - ベタ塗り / 16 進数の色指定 / 既存の色指定からの微調整 / 本文文字色の指定	
5.1.2 背景画像	140
画像ファイルをコピーする / 背景画像の指定 / 色に関する注意 / 画像のフォーマット / 背景画像の入手	
5.1.3 文書中の画像	145
IMG タグの基本 / テキストの位置揃えと回り込み / 小さい画像の使い方 / より快適なページのための画像使用法 / WIDTH と HEIGHT の注意	
5.1.4 GIF の透過設定とアニメーション GIF	156
透過 GIF / アニメーション GIF	
5.2 画像について	158
5.2.1 画像ファイルと各種画像フォーマット	158
5.2.2 画像ファイルの作成と編集	159
<b>第 6 章 Web の公開 !</b> .....	162
6.1 サーバに送り込む	162
6.1.1 サーバに「送る」ということ	162
Web を公開する , ということ / 何を送るのか / どこに送ればよいのか / どこから送るか / どうやって送るか	
6.1.2 ブラウザを使った FTP 転送	168
ブラウザ FTP の制限事項 / FTP の開始 / ファイルを送る / 転送した結果をチェックする	
6.1.3 コマンドラインからの FTP 転送	176
コマンドライン FTP について / 開始準備 / カレントディレクトリの確認と移動 / ディレクトリを作成する /	

---

画像ファイルを転送する / 削除したい場合 , 整理したい場合 /  
ページを表示させて確認 / ファイル名がindex.htmlである理由

<b>第 7 章 クリックしてね !</b> .....	188
7.1 リンクを使う .....	188
7.1.1 別のファイルにリンクする 188	
ハイパーリンク / ファイルを二つ準備する / リンクを張る / リンクスイッチの色指定 / ボタン画像を押してリンク先に飛ぶ / リンクスイッチ画像の縁取り / サーバへの転送とチェック	
7.1.2 リンクと URL 197	
相対 URL と絶対 URL / 外部サイトへのリンク / リンク切れへの注意	
7.1.3 リンクあれこれ 199	
新規にウィンドウを開く / メーラーを起動する / 同じ文書内にリンクする	
7.1.4 イメージマップ 202	
イメージマップについて / 使用例 / マップ定義 / 座標値の取り方 / イメージマップのまとめ	

---

## 第 3 部 レイアウトの基礎

---

<b>第 8 章 フレームによるページの区画分け</b> .....	210
8.1 フレームを使ったページ .....	210
8.1.1 フレームによる画面分割 211	
用意するファイル / FRAMESET タグの書き方 / FRAME タグの書き方 / NAME と TARGET / 基本のまとめ / FRAMESET のオプション	
8.1.2 フレーム対応のよしあし 219	

---

<b>第 9 章 テーブルを使ったレイアウト</b> .....	220
9.1 ページの見た目を整える .....	220
9.1.1 テーブルタグ 220	
テーブルタグの基本 / 複数の行または列にまたがるセル /	
セルの幅と高さの指定 / セル内の配置指定 / そのほかのオプション /	
表を入れ子にする	
9.1.2 テーブルタグでレイアウトする 233	
使用上の注意 / レイアウトの青写真 /	
メイン部分の作成 - 以前のページを貼り付ける /	
メイン部分の作成 - 表の中で表を使ってみる	
9.2 レイアウトについて .....	242
9.2.1 見た目と論理構造 242	
9.2.2 ブラウザによる違い 243	
9.2.3 HTML でのレイアウト 244	

---

## 第 4 部  少し進んだ話題

---

<b>第 10 章 アクセスカウンタ</b> .....	246
10.1   アクセスカウンタをつける .....	246
10.1.1   アクセスカウンタ 246	
アクセスカウンタとは / プログラムについて /	
プロバイダのカウンタを利用する場合 / アクセスカウンタの仕組み	
10.2   CGI と SSI .....	252
10.2.1   CGI 253	
10.2.2   SSI 254	
SSI とは / 日付や時刻 / ファイルの最終更新時刻 /	
そのほかの SSI コマンド	

---

<b>第 11 章 お話しようよ</b> .....	258
11.1    フォーム	258
11.1.1    フォームでアンケート	258
フォームを使ったアンケートページ ( 1 ) / FORM タグ /	
INPUT タグ - 送信とリセット /	
INPUT タグ - ラジオボタンとチェックボックス /	
INPUT タグ , TEXTAREA タグ - テキスト入力 /	
メールで送られてくるフォームの内容 /	
SELECT タグ - リストの中から選ぶ	
<b>第 12 章 スクリプトを使ってみる</b> .....	270
12.1    小技として使うスクリプト	270
12.1.1    ちょっとしたスクリプト	271
JavaScript の使い方 / JavaScript を書く作法 / 関数定義 /	
関数のパラメータ /	
関数内での処理の記述 / 関数 toNext ( ) の働き / 関数を呼び出す	
12.1.2    イベントハンドラとプロパティ・メソッド	281
イベントハンドラ / プロパティ・メソッド / 次に表示されるページの	
URL / プロパティの使用例 - 画像の上きたらメッセージを出す /	
メソッドの使用例 - 新しいウィンドウを開く /	
うまくうごかないとき	
12.1.3    小技あれこれ	288
メニューから選択してリンクに飛ぶ / 選んだら即ジャンプ /	
マウスが乗ると画像が変化する / 複数画像を切り替える - 配列を使う	
12.2    JavaScript 補足	296
12.2.1    セキュリティの問題	296

---

<b>第13章 まとめとこれから</b> .....	298
13.1 もっといろいろなこと .....	298
13.1.1 CSSによるレイアウトの強化 298 例題 / 利点 / 別の書き方 / 補足	
13.1.2 iモードへの対応 303 iモード / そのほかの携帯端末	
13.2 まとめ .....	305
13.2.1 XMLとXHTML 305	
13.2.2 おわりに 306	
 参考文献&URL .....	 308
索引 .....	310
付属CD-ROMについて .....	315





図2 図1のHTMLソース

```

<H1>KDC Information</H1>

Welcome to the KDC WWW server. This server is still in the process of
being set up. If you have question on this KDC information page, send
e-mail to montal@sk.jp.
<DL>

<DT><A HREF=ListMode/QuickGuide.html>Help</A>
<DC>On this program,
or the<A HREF=http://info.cern.ch/hypertext/WWW/TheProject.html> World-Wide Web</A> .

<DT><A HREF=http://info.cern.ch/Experiment/DataCoeries/bySubject/Physics/HEP.html>HEP</A>
<DC>World Wide Web services provided by other High-Energy Physics institutes.

<DT><A HREF=hw.html>CWC</A>
<DC>EEC Integrated Workstation environment initiative.

<DT><A HREF=Foot.html>Foot</A>
<DC>WG Manager Support (Foot) [DOC]

<DT>See also:
<DC>
<A HREF=http://info.cern.ch/hypertext/DataSources/Basics.html>Types of server</A>, and
<A HREF=http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html>OTHER SUBJECTS</A>
</DL>

```

日本最初のWebページから8年ほど経った現在，世界のWebサーバの数は爆発的に増加しました．現在どのくらいのWebサーバが世界中で稼働しているのか，正確なところはおそらく誰にもわからないでしょう．

これだけ普及したWWWあるいはWebの中でも，ホームページは言葉としてもだいぶ一般化し，新しい情報発信の手段として注目を集めてきています．初めは研究や技術の世界にかぎられていたWebページも，インターネットの普及とともに日常のありとあらゆる話題を網羅する時代です．それに伴い，一般の人がWebページを作成し，公開する数も飛躍的に増加しています．

とはいえ，コンピュータやネットワークについての経験や知識があまりない人にとってはWebページを作成することはまだまだそれほど簡単なことではないようです．自力でWebページを立ち上げるためには，多くの場合，コンピュータに関する参考書，ネットワークに関する

図3 1992年9月当時の世界のWebサーバ(CERN92-07 21December1992より)



参考書，Webページの基礎的な参考書，少し進んだWebページの参考書，等々と，いくつかの書籍を揃えて勉強する必要があります。

### 0.1.2 本書の目的

本書では，現在よく見かけるレベルのWebページを一人で作り，運営していくために必要となる基礎知識と技術を，できるだけコンパクトにまとめることを試みました。パソコンを持っていて，Webページを見ることやメールのやりとりはなんとかしたことがある(あるいはこれからしたい)けれども，Webページを作るとなるとどうしていいかわからないという人が読んでも，どうにかWebページが作成できるようになる，ということを目指しています。もちろん，コンピュータについてある程度予備知識はあるけれどWebページを作ったことがないという人，Webページはどうか作っているのだけれど，いまひとつ実感がわからないという人にも役に立つように考えました。

巷では「すぐできる」「簡単にできる」といったWebページ作成の参考書も見かけます。そうした本に比べると，本書は最初，難しく感じられるかもしれません。実際には，時間をかけてじっくり馴染んでいくことができる環境が身近にあれば，Webページの作成というのは内容としてそれほど難しいということはありません。難しいと感じるのは主に，馴染みのない用語にとまどうことと，コンピュータやネットワークという実体の見えにくい環境で作業することによります。言われた通り，書かれている通りに操作すればともかく何か結果は出るけれども，自分が一体何をしたのか，その意味がわからないとなかなか覚えられませんし，環境が変わると応用も効かないものです。

たとえ話になりますが，子供が料理を覚える過程を考えてみましょう。子供というのは，一度も自分では料理したことがない，という意味です。皮を剥いたり茹でたり炒めたりということも全く初めて，という状態から料理を覚えるまでを想像してみてください。料理には，皮を剥く，切る，炒める，茹でるなど，色々な作業が必要です。最初はたとえばサラダを作るのに野菜を切ったり，粉吹きイモを作るのに皮を剥いて茹でて，としたり，簡単なメニューから始めます。やっていることは簡単ですが，いくつかのレシピをこなしていくうちに，野菜を切るのは食べやすい大きさにしたりドレッシングを馴染ませたりするためであること，皮を剥くのは，皮が固くて食べられなかったり食べにくいものは皮を剥くのだ，ということ，茹でたり炒めたりするのは食べやすく，消化をよくするためだということ，言葉ではよくわからなくとも経験として身につけていくことになります。そしていずれは自分でアレンジした料理を作ったり，新しい食材でも料理ができた，という応用ができるようになっていくわけですが，もしレシピをレシピどおりに実行するだけで，こうした料理の作業の意味を全く理解しないままだったとしたら，何を作るにもレシピどおりでなければできない，ということになってしまいます。たとえば野菜炒めなら，何か適当な野菜があれば切って炒めれば済むだけのことで，「切る」「炒める」ということの意味を理解していなければ，あり合わせの野菜を使って野菜炒めを作る，ということすらできないわけです。

この例はあまりに日常的すぎるので，「あり合わせの野菜で野菜炒めを作ることすらできない」などというのは突飛すぎると思われるかもしれません。しかし，小説や映画などに出てきそうな，料理は食べるだけで，本当に一度も作ったことがない人がもしいたら，野菜を切るということすらどうやっていいか途方に暮れてしまうだろう，ということは想像していただけるのではないのでしょうか。

Webページ作りに関しても，今のたとえ話と似たところがあります。Webページ作りでも料理でも，経験のある人にとっては簡単と思えることを一つ一つ自分で実際に経験しながら理解

して身につけていかななくてはなりません．とりあえずできる簡単レシピ，というのは，作ることの楽しさを体験するという意味で重要な役割をもっています．ただ，そこから抜け出して自分なりの料理やWebページを作るには，ある程度時間をかけてじっくりと内容を理解するという経験を積んでおくことは絶対に必要なのです．

そういった意味から，本書では，通常「やっているうちにわかる」として簡単に書かれていることが多いことについてもかなりの程度，詳しい解説を試みています．もちろん，ややこしい勉強ばかりで結果がなかなか出ないのではやる気が失せてきますから，例題としてはできるだけ実用に近いものをとりあげてみました．項目によっては，細部は思い切って切り捨て，主要な考え方に的を絞った解説になっているものもありますが，将来，自分でいろいろ調べたときに，もっと進んだ解説を理解するための基礎的な力をつく内容になっていると思います．

本書を読まれて，たんなるネットサーファーではなく，Webページの海を自力で航海できる航海者になる方が増えることを願っています．

### 0.1.3 本書の構成

本書は5部構成になっています．

**第1部**はコンピュータとネットワーク，それにWebに関する基礎知識をまとめました．コンピュータやネットワークについては，Web作成に必要な知識に的を絞った解説になっています．

**第2部**は，基礎的なWebページの作成と公開までを扱っています．綺麗なレイアウトや高度な機能はできませんが，第2部を理解することがWebページ作成の重要な基礎がためになります．レイアウトにあまり凝らないのであれば，第2部までで基本的なWebページの作成に必要な知識はまかなうことができます．

**第3部**はWebページの見た目を整えるための方法についてまとめています．本書中でも繰り返し触れていきますが，最初の頃のWebページは，レイアウトということについて，必ずしも強いとはいえませんでした．このレイアウト面を補うため，ここ数年様々な方法が提案され，使われるようになってきています．第3部はそうしたレイアウトの手法について，基礎的な部分に的を絞って解説しています．

**第4部**では，Webページとプログラムとを連携させる方法のさわりについて紹介しています．プログラムに関係したことを本格的に扱うには紙数が限られているため，この第4部は実用的な例題を中心として，雰囲気を感じてもらえる方向になっています．いずれ機会があれば，稿を改めて基礎的な部分からの解説を試みたいと思っています．

最後の**第5部**はまとめです．ここでは実的な話はなく，Webにまつわるいくつかの話題をお話的にとりあげました．

読み方の例として，コンピュータやネットワークについてある程度予備知識のある方は，第1部は適当に飛ばして第2部へ進まれてもかまいません．また，予備知識のない方でも，理論的な話が得意でない方は第1部を適当に流し読みされて先へ進み，必要なときに必要なところに戻ってくる，という読み方でもかまわないと思います．第2部までを理解すれば，最低限，自分のWebページを運用することができるようになるでしょう．第3部まで進めば，ある程度見栄えもするWebページを作成できる段階になります．第4部以降は少し高度な内容に入っていきますから，試してみたくなったときに必要なところを読む，という読み方もできます．



# 第1部 基礎知識

第1章	Webの基礎知識	16
第2章	コンピュータの基礎知識	30
第3章	ネットワークの基礎知識	70

# Webの基礎知識

この章では、Webのデザインを始める前にWebやインターネット、コンピュータについて、最低限必要となる知識をざっとみていくことにしましょう。いわば、旅行前の下調べ、というところでしょうか。旅の場合も、ほんとうの知識や経験というのは現地に行ってみて初めて身につくものですが、Webの場合も同じです。下調べのときによくわからなかったことも、実際に経験してみればなんとなくわかってくるものですから、この章に書いてあることがあまりよく理解できなくても心配することはありません。なにかそういうことを読んだな、という程度に読み流すつもりで進んでみてください。

## 1.1

### Webページ作りに必要なもの

本書の目的はWebページの作成です。そこでまず、Webページを作成するのに何が何を必要かをざっとみておきましょう。

#### 1.1.1 コンピュータとソフト

まず、コンピュータはどうしても必要です。しかし、それほど高機能なものは必要ありません。執筆時点では10万円前後のコンピュータを店頭でもよくみかけるようになってきていますが、そのクラスのコンピュータで十分にWebページの作成は行うことができます。

新しく買う場合、今のところ市場ではDOS/V(Windows)とMacintosh(MacOS)が主流ですが、どちらでも大丈夫です。DOS/VでもMacintoshでも、BSDやLinuxなどUNIX系のOSが動作します。Linuxを含めてほかのUNIX系の機種とOSでももちろんかまいません。要するに、ネットワークに接続ができてWebページが閲覧できるものであればよく、それほどの高性能である必要はない、ということです。WindowsやMacintoshの価格帯でいうなら、10万円台のものでも十分に活用できます。

主に使用するソフトウェアについてですが、Webページを閲覧することのできるブラウザはもちろん必要です。最近の機種であればこれは最初から入っている(使えるようになっている)ものがほとんどですから心配ありません。Webページの作成には、テキストエディタと呼ばれる、編集用のソフトウェアを使います。これも、たいいていの機種では簡単なものが初めから備わっています。当面はこのテキストエディタで十分間に合います。なお、「テキストエディタ」という名前のソフトがあるわけではありません(絶対ないとはいいいませんが、普通は別の名前です)。テキストエディタというのはソフトの種別のことで、書式のない文字情報を編集するためのソフトウェアをまとめてこのように呼んでいます。具体的になんというソフトウェアを

使うのかは第4章以降で説明しています。

以上をまとめると、廉価版クラスのコンピュータ1台と、標準で付属しているソフトウェアがあれば本書を読んでWebページを作ることは十分可能、ということになります。

### 1.1.2 インターネットへの接続

つぎはインターネットへの接続です。これは、本書の範囲ではとにかくインターネットに接続できる環境があれば形態は問いません。必要なときだけプロバイダにADSLやISDNで接続する、あるいはアナログの電話回線で接続するダイヤルアップ接続でも、ケーブル会社などの光ケーブルを使った接続でもかまいませんし、常時接続といわれる24時間つながり放しの形でもかまいません。もちろん、大学や企業などでインターネットにつながっている端末があればそこからでもできます。

実際どのように接続するかという具体的なノウハウまでは本書では扱いませんが、とりあえずWebページが閲覧できるような環境であれば大丈夫です。

ただ、一点注意しておきますと、Webページを作成する過程では頻繁にサーバにアクセスすることが考えられます。常時接続や大学・企業のインターネット環境であれば、接続時間が長くなっても料金が上がることはありませんが、一般のダイヤルアップ接続の場合には通信料金に気をつけなければなりません。本書の執筆途中、特定プロバイダへのISDN接続に限ってどれだけ使っても1ヵ月ごとに定額、というサービスをNTTが開始しました(商品名・フレッツISDN)。これに応じて、プロバイダ側でも定額で使い放題、というサービスを始めるところが増えています(注:現在はフレッツADSLがお勧めです。いずれ家庭に光ファイバが入るFiber To The Home=FTTHの時代がくるでしょう)

インターネットへの接続料金は、電話代+プロバイダの料金、です。これまでも定額で使い放題、というプロバイダはありましたが、電話代のほうは深夜・早朝に限った定額サービスしかありませんでした。深夜・早朝(23時~7時)の定額サービスはNTTの商品名で「テレホーダイ」という名称なので、俗に「てれほタイム」などと呼ばれ、23時以降にネットワークが混雑する、という現象があるくらいです。料金はテレほ一だも、フレッツ系のサービスも同程度なので、作業する時間帯によってどちらを選択するかは違いますが、家庭でWebページを作るならばこの種の割引サービスを使わない手はありません。CATVケーブルなどの料金定額系のサービスも、利用できるならば考えてみる価値があります。

いずれにしても、Webページのこまめなメンテナンス(維持管理)のためには、できるだけ料金を気にせずに接続ができるインターネット環境を整えることが大切です。

### 1.1.3 Webサーバ

Webページの公開にはサーバが必要になります。サーバというのはコンピュータの役割を表していて、サーバという機種のコンピュータがあるわけではありません。サーバに対応する言葉にクライアントというものがあります。ごく簡単にいってしまうと、サーバはあるサービスを提供する側で、クライアントはサービスを受ける側になります。あるサービスというのは、ここではWebページを表示するサービス、と考えておくことにしましょう。このほかにも電子メールのサービスやファイル共有のサービスなど、さまざまなサービスがあって、1台のコンピュータがいくつものサーバ機能を兼ねることや、あるサービスに関してはサーバの役割をしつつ別のサービスではクライアント側である、ということもあります。

原理的にはどんなコンピュータでもサーバになることができますが、一般的には、より高性能で耐久性の高いコンピュータがサーバの役割を担うほうが望ましいといえます。また、Web

サーバでは、常時インターネットに接続して24時間電源が入っている、ということが多くの場合、望めます。家庭のPCではこの条件を満たすことが難しいため、通常はどこかで運用しているWebサーバに間借りすることが多いでしょう。

今、WebページのサーバということでWebサーバと書きましたが、サーバとしてはWorld Wide Webのサーバということで、WWWサーバという呼び方もあります。環境によっては自分がWebページを作成するコンピュータがサーバを兼ねることもありますが、ダイヤルアップ接続など常時接続しているわけではない場合、プロバイダか、会社・学校などのWebサーバを使用してWebページを公開することになります。別の言葉でいうと、Webページを作成するために使用するコンピュータとは別に、Webページを公開するためのコンピュータ(サーバ)が必要になります。

通常、Webサーバには常にインターネットに接続されているコンピュータを使い、24時間電源を入れて連続運転するのが基本です。24時間稼働、つねにインターネットに接続というのは、言い換えればいつでもどこからでもWebページにアクセスしてもらえる環境ということになります。

社内や学内など、外部に公開したくないコンテンツ(内容)のWebサーバは逆にインターネットに接続しないようにする場合もあります(その場合でも社内や学内のネットワーク = LAN には常に接続されているのが普通です)、特定の時間帯しか稼働しないサーバというのもあり得ます。こうしたケースで使われるコンピュータもWebサーバには違いないのですが、ここではそういったケースは除いて考えることにしましょう。

ところで、つねにインターネットに接続されているという条件ですが、これは光ケーブルのような環境で常時接続というだけでは不十分で、世界中から、サーバとして使用されているコンピュータをつねに参照できるようにしていなければなりません。契約するプロバイダにもよりますが、現状ではケーブルTV会社の個人ユーザー向け接続などでは接続先(要するに家庭など)にサーバを置くことを認めていないところも多いようです。もちろん、ダイヤルアップで接続するような環境では、サーバを置く条件をそもそも満たすことができません。

そんなわけで、Webページとは別に、作成したWebページを公開するためのWebサーバの確保、というのもWebページ作りに必要な条件の一つになります。会社や学校で、個人のページを置くことを認めているサーバがあればそれを使うのがもっとも安上がりな方法です。もちろん、仕事や授業でWebページを作らなければならない場合は、はじめからサーバが与えられているわけですからそれを使えばよいことになります。

そういったサーバがない場合は、プロバイダが提供するサーバを使うことになります。サーバ提供の形はプロバイダによって様々で、インターネットへの接続契約をすれば自動的にWebページを置く環境もセットで提供しているところもありますし、申し込みば無料で使わせてくれるところもあります。場合によってははじめから有料のところや、Webページとしてサーバに置くデータ量が多い場合に追加料金が発生するケースもあります。

こうしたプロバイダとは別に、無料ないし有料でWebサーバのレンタルをしてもらえるところもあります。無料のもので有名などころには、たとえばジオシティーズのサービスがあります。これはYAHOO JAPANが提供しているサービスで、作成したWebページには強制的に広告が入りますが、その代わりに利用料は無料です。有料のサービスは、置くことのできるデータ量や、利用できる付加サービスによって価格はまちまちで、年間1万円を切るようなところから個人ではちょっと払えないような額のサービスまであります。

なお、どの場合でも、サーバに置くことのできるデータ量や、サーバ上で実現できる機能には通常、制限があります。また、ドメイン名やURLという、自分のWebページを表すのに関係

する名前のようなものも、一般には自由に使えるわけではありません。いずれも無料のサービスやプロバイダの付加サービスではある程度、サービス提供側に決められた範囲で利用することになります。有料サービスの場合は、データ量や機能面では無料のものより融通がききます（もちろん料金は変わってきますが）。また、ドメイン名やURLについても一定のルール内で自分が選ぶことができる場合が多いようです。いずれにせよ、作成したWebページを世界に向けて発信するにはWebサーバを確保しておかねばなりません。参考までに、ジオシティーズのURLを下記に記載しておきます。

<http://www.geocities.co.jp/>

会社や大学、プロバイダのサーバを利用する場合は、それぞれ利用に必要な情報(後の章でできます)は、管理者に問い合わせるか、管理者の用意したマニュアルを参照しなくてはなりません。本書でそうした情報を理解するための知識を習得してください。

#### 1.1.4 コンテンツと情熱

さて、ここまで述べてきたものを揃えれば、外枠としては準備が整ったこととなります。しかし、Webページの作成にはもっと大切なものがあります。それはコンテンツと情熱です。

コンテンツというのはWebページに載せる内容のことと思えばよいのですが、要するにWebページを作って何を発信するのか、ということです。わたしの周りにも、コンピュータはあまり得意ではないけれど、なんとかWebページを作ってしまった、という人が何人もいますが、そういう人たちは例外なく、「こういう内容のページを作りたい!」という情熱をもっていました。実際にWebページを作るまでにはいろいろ苦労することもちろんありますが、最終的にはなんとかできてしまいますし、情熱のある人が作ったWebページというのは楽しいものが多いのです。HTMLなどは参考書を見ればすぐに理解できてしまう、というような人でも、情熱がなければWebページができてくることはないわけです。Webページを作るのに本当に必要なのは、「何かを発信したいという熱意」と筆者は思っています。発信したいコンテンツと情熱さえあれば、Webページは必ず作れます。

コンテンツと情熱と必要な環境が整ったら、いよいよWebページ作成の旅の始まりです。

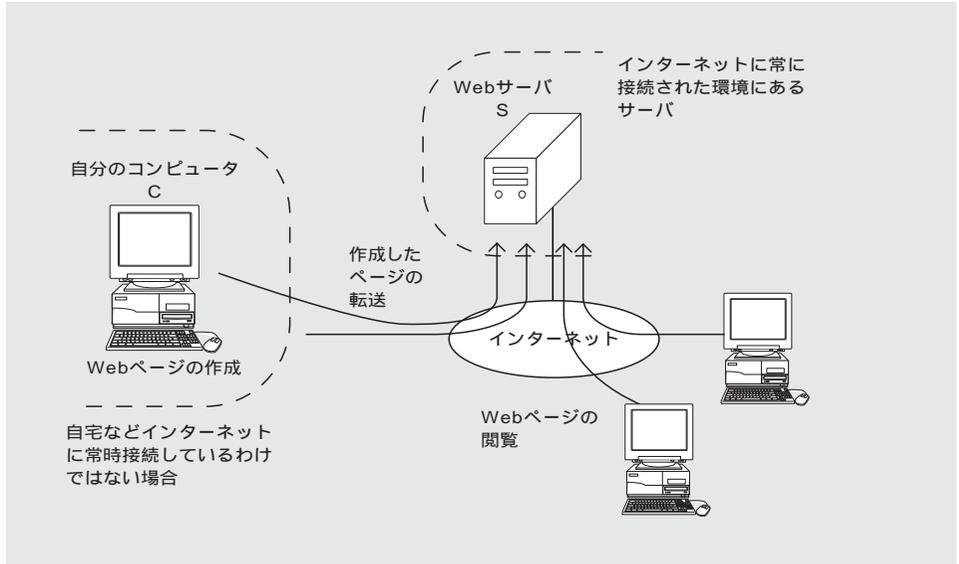
## 1.2

### HTMLの雰囲気

さて、本章の残りの部分では、Webページを作成していくためのイメージ作りとしてWebサーバを中心とした自分のWebページの運用形態を確認し、ブラウザやHTMLについてざっとみていくことにします。これから先学んでいくことについて、何のために、どういうことをしていくのか、というおおまかなイメージをつかんでください。

なお、ここまでは意識して「ホームページ」という言い方を避け、Webページとしてきました。「ホームページ」は本来、個人や組織などのWebページの中で最初に表示されるページのことを指すものです。ほかに「トップページ」などということもありますが、一般的にはホームページとWebページはほとんど同じ意味で使われることも多いようです。本書ではとくに必要があるとき以外はすべてWebページで通していきます。

図1.1 レンタルWebサーバのサーバとクライアント接続形態



### 1.2.1 Webサーバのイメージ

まず、Webサーバをレンタルで確保する場合を図1.1で見えます。これにあてはまるのは、プロバイダ経由でインターネットに接続し、レンタルサーバでWebページを公開するケースや、企業・学校などでサーバは人任せというケースです。企業・学校のサーバの場合はレンタルという言葉は少し違いますが、誰かが設定してくれたサーバを使わせてもらう、という意味でこの分類に含めます。本書では主にこの形態でのWebページ作成を念頭において記述していきます。ただし、Webサーバの持ち方が主に関係してくるのは第6章、第10章、第11章のみで、他の章を読む際には気にする必要はあまりありません。

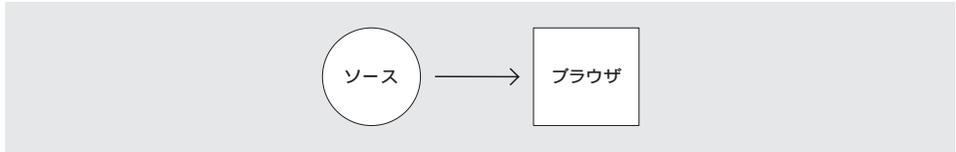
プロバイダ経由で接続する場合についていうと、図1.1の形態では、Webページを作成するために使うコンピュータCは自宅にあります。ページの作成時にはインターネットに接続されている必要はありません(接続されていても構いませんが、ページの作成とは関係ありません)。作成に必要な知識は本書全体で学んでいくことになります。

一方、Webページを公開するためのサーバSは、プロバイダやレンタル業者など、どこか別の場所にあるコンピュータです。自宅のコンピュータで作成したWebページは、インターネットを通じてWebサーバに転送することになります。転送するときはもちろんインターネットに接続している必要があります。転送に関する方法は、本書の第6章で扱います。作成し、サーバに転送したWebページは、原則としてインターネット上にいるすべてのユーザーが閲覧できる状態になります。この形態での作業手順を仮にまとめると、

- (1)自宅のコンピュータでWebページを作成，チェックする
- (2)Webサーバに転送する
- (3)(自分も含めて)インターネット上で閲覧する

となります。Webページの更新があるたびに(1)~(3)の手順を繰り返すことになります。企業や大学のサーバを企業内や学内から利用する場合、自宅ではなく自分の机の上や教室の机の上にあるコンピュータが、コンピュータCになります。この場合はプロバイダの場合に比べる

図1.2 ソースとブラウザの抽象的關係図



と、サーバは近くにあるかもしれませんが、やはり別のコンピュータであり、自分で自由に設定することはできません。

ここでのポイントは、ページを作成するコンピュータとWebサーバという二つのコンピュータがあって、場所的に離れている、ということです。たんに場所的に離れているだけでなく、Webサーバは自分が自由に設定、管理できるものではない、ということも重要です。レンタルのサーバの場合、サーバに置くデータの量や機能面での制限があることが多く、ページを作成するに当たっては許可されていることとされていないにも注意を払うことが必要です。たとえば、第11章で出てくるCGIやSSIについてはサーバによって使えなかったり、使えても一定の制限があったりするケースがあります。

とはいえ、自分でサーバを管理しなくともよい、ということは逆に大きなメリットでもあります。はじめは誰かが設定してくれているサーバでWebページ作成の腕を磨き、ある程度マスターしたところで自前のサーバを考えてみるのも一つの手です。サーバを自前で立ち上げて、本書で覚えることがらはすべてそのまま通用します。以上が本書で主に想定するWeb作成の環境です。

### 1.2.2 HTMLとブラウザ

では続いて、Webの仕組みをざっと眺めてみます。図1.2をみてください。この図はとても単純にしたもので、四角と丸が線で結ばれています。丸には「ソース」、四角には「ブラウザ」と書かれています。

「ブラウザ」というのは、たとえばNetscapeやInternet Explorerのこととってください。この「ブラウザ」は、英語では"browser"で、「ざっと見る」という意味の"browse"(ブラウズする)から来ています。つまり、「ブラウズするもの」が、「ブラウザ」ということです。しかし、Webが普及したこともあるのでしょうか、最近の辞書には"browse"の意味として「Webページをみること」という意味も載るようになってきているようです。

要するに、「ブラウザ」というのは人間がWebページを見ることができるよう、Webページを「表示してくれる」ソフトウェア、ということができます。このWebページというのは、たとえば図1.3のようなものです。これはこの本の出版元であるCQ出版社のトップページをInternet Explorerで表示したのですが、Webを作成した経験のない人であれば、この画面がつまりWebページだ、と考えているかもしれません。別の言い方をすると、「画面に表示されるイメージ」=「Webページ」というイメージですが、これは半分当たりで半分は違っています。というのも、コンピュータ上(今の場合、CQ出版社のWebページが保管されているサーバ上)には、図1.3のような画像のイメージが保管されているわけではないからです。サーバに保管されているのは、図1.4のような内容のファイルです。このファイルが、図1.2にある「ソース」の中身にあたります。

図1.4が「ソース」、図1.3が「ブラウザ」によって表示された画面ということを踏まえて、もう一度図1.2に戻ってみます。この先、本書ではWebページを作成するための様々な手法を取り上げていきます。この際、私たちが作成していくのは図1.2にある「ソース」のほうとい



図1.5 おもなブラウザの一覧表

<ul style="list-style-type: none"> <li>• Netscape Communicator &amp; Netscape Navigator</li> <li>• Internet Explorer</li> </ul>	執筆時点ではこの二つが「二大ブラウザ」と呼ばれている。昔は Netscape 系は有料であったが、シェア戦争が激化するうちにいつしか無料となった。現在では Windows, Macintosh とともに、この二つのブラウザは「あって当たり前」といった雰囲気になってきている。
<ul style="list-style-type: none"> <li>• Mosaic</li> </ul>	HTML が普及し始めた頃は、ブラウザといえはまず Mosaic, と言っていいほどメジャーなブラウザであった。
<ul style="list-style-type: none"> <li>• HotJava</li> </ul>	Java が世に出始めた頃、実際に Java が動くブラウザとして Java 提唱元の Sun Microsystems が開発し、配布を始めたブラウザ。
<ul style="list-style-type: none"> <li>• Lynx</li> </ul>	文字のみの表示を行うブラウザ。主に UNIX 系で使われていると思われるが、動作も軽く、ブアな(表示能力の乏しい)端末しか使えない環境では意外に重宝する。
<ul style="list-style-type: none"> <li>• WebBoy</li> </ul>	MS-DOS でも使用できるブラウザ。IBM 製で、古いパソコンを再活用、というコンセプトで売り出されている。パソコン関係の部品はリサイクルしにくいので、せめて Web ブラウザとしてだけでも古い PC を活用できれば、というのはなかなかよいと思うのだが、あまり売れているような話は聞かない。
<ul style="list-style-type: none"> <li>• Arena</li> <li>• Amaya</li> </ul>	Arena は、HTML3.0 の、Amaya は HTML3.2 の検証に用いられていた / 開発されているブラウザ。W3C という標準化を進めようという組織があり、これらのブラウザが使われているのも主にその組織の内部ではないかと思われる。標準化を進めるにあたってどれか特定のブラウザに偏るのは好ましくない、という思想があるのかもしれない。

HTMLの書き方を具体的にお話するときにまた触れることにはなりますが、もともとHTMLというものは、「表示のレイアウトはブラウザまかせ」という発想をもっているものだ、ということをご記憶しておいてください。

### 1.2.3 いろいろなブラウザ

HTMLでは「表示はブラウザまかせ」ということですが、ここで少し、その実例をみてることにしましょう。

最近では、ブラウザといえばNetscapeかInternet Explorerというのが定番になっているといっいでいいでしょう。しかし、この2種類のブラウザ以外にも、Webページを閲覧することのできるソフトウェアは意外にたくさんあります。図1.5に様々なブラウザの一覧を掲げます。

図1.5の中で、少し極端な例になるかもしれませんが、文字しか表示できないLynxというブラウザを用いた例を図1.6に示します。表示させているページは図1.3と同じ、CQ出版社のページです。図1.3と図1.6を比べてみるとわかるとおり、文字として含んでいる情報はどちらも一応同じ、といえますが、見た目、つまりレイアウトはかなり違います。この違いはブラウザのもつ機能の違いからきているものです。Lynxでは画像を表示することがいっさいできません。画像表示が指定されている部分には、HTMLソース側でとくに指定されていなければ、[IMAGE]などのように「そこに画像があるよ」ということを示す文字列が入ります。また、WebページにはほかのWebページなどにジャンプするためのリンクという機能がありますが、リンクについては下線と色の変化(図1.6では白黒の濃淡となっています)で示されています。反転している文字の部分が現在注目しているリンクとなっており、矢印キーで注目リンクを移っていきます。リンク先にジャンプしたい場合にはEnterキーを押します。

このように、ブラウザが違っても同じHTMLソースであってもかなり違ったイメージで表示さ

図1.6 CQ出版社トップページ (Lynx)



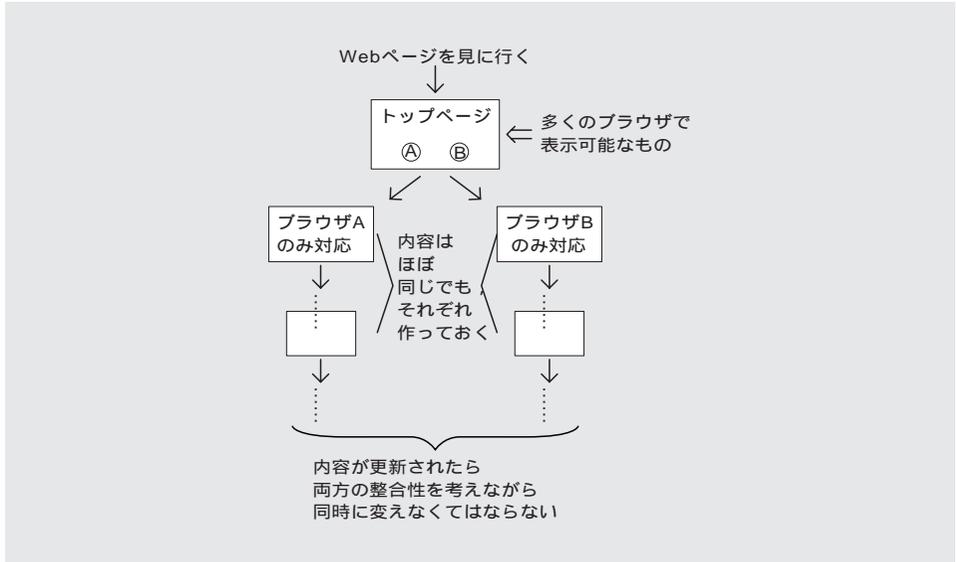
れます。こうした文字のみのブラウザというものは極端だ、と思われるかもしれませんが、しかし、たとえば最近流行しているiモードでも、簡易な画像は表示できますが文字を主体とした表示が基本です。実際にはiモードで表示可能なHTMLソースには使用できる機能に制限があり、Lynxとの直接的な比較には少し無理がありますが、たとえばiモード対応ページを通常のブラウザで閲覧することは可能です。この場合、ブラウザとしてiモード対応の携帯電話を用いていれば、電話番号のリンクをクリックすることで自動的に電話をかけることができますが、Netscapeなどのブラウザではこのようなことはできません。表示イメージだけではなく、ソースに埋め込まれたある種の指示に関してもブラウザによって対応できるかどうか異なってくるわけです。

この種のブラウザによる違い、というものはNetscapeとInternet Explorerという2大メジャーブラウザの間にも存在しています。HTMLではタグというものをを用いて、表示をはじめブラウザに対する各種の指示を書いていきますが、このタグに関してもNetscapeとInternet Explorerでは互換性のないものが存在します。タグだけでなく、ある種の拡張機能のようなものもそれぞれのブラウザによって違ったものをサポートしていることがままあります。さらには、同じNetscapeもしくはInternet Explorerであっても、バージョンが古いものでは新しいものに備わっている機能をもたないわけです。このため、Webページによっては「推奨ブラウザ」という形で、「このページはバージョンXXより新しいYYのブラウザで閲覧してください」という指示を掲げているところも見受けられます。

#### 1.2.4 手間と結果のトレードオフ

一般的に言えば、こうしたブラウザ限定のページはあまりよくないと思われませんが、一概にだめということもできません。たとえば、ページの作成者がどうしても実現したい機能があり、ページを作成する時点において、実現したい機能がある特定のブラウザでしかサポートされていない、というようなケースです。このようなケースでは、対象となるブラウザを限定してページを作るよりほかに仕方がないこともあるでしょう。ただ、この場合でもWebページのもつリンクという機能を生かし、特定のブラウザでしか閲覧できない部分と、多くのブラウザで閲覧できる部分とを切り分けておく、といった対処の仕方は考えられます。ページを作る以上、できれば多くの人に閲覧してもらいたいというのはページ作成者に共通した気持ちでしょう。

図1.7 ブラウザ振り分け対応ページの問題図



し、そのためにはできるだけ多様なブラウザで表示が可能なページであるにこしたことはないといえます。

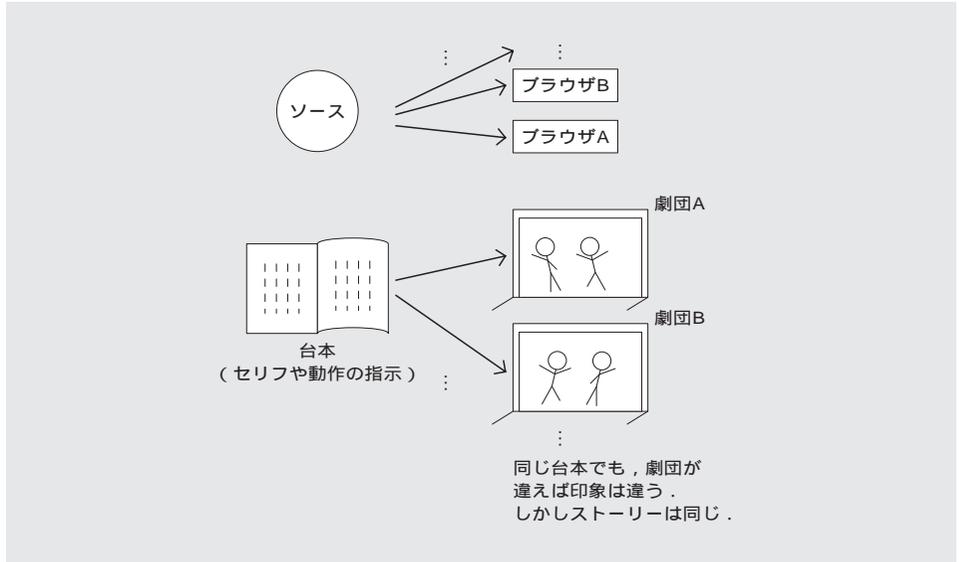
とはいうものの、様々なブラウザが存在し、個々のブラウザも頻繁なバージョンアップを繰り返している現状では、すべてのブラウザで閲覧できるページを作るということはあまり簡単ではありません。もちろん、ごく基本的な機能に限定すればこれは可能です。さきほど出てきたiモード対応のページなどは、クリック一つで電話をかけるといったiモード特有の機能を除けば、現存するほぼすべてのブラウザで閲覧可能なものの例といえるでしょう。しかし、Netscapeなどの、表示に関してより高機能なブラウザを利用できるのならば、それなりにレイアウトにも気を配ったページを作りたくなるのもまた人情です。実際、iモード専用のWebサイトというものもありますが、既存のWebサイトで、iモード対応と一般ブラウザ対応のページの両方を用意しているところもよく見かけます。さきほどの特定ブラウザ限定ページにしても、より親切であろうとするならば、一般のブラウザでも表示可能な(しかし機能的には少し劣る)ページと、特定のブラウザのみで表示可能な(その分高度な機能もサポートした)ページとの両方を用意する(図1.7)、ということも考えられます。

このようにブラウザの多様性に配慮したページを作成すれば、ページを閲覧する側にとってはより快適な環境が実現されるといえます。

しかし、逆にこれはページを作成する側にとっては負担が増えることも意味します。Webページは作りっぱなしでよいというものではなく、新しい情報が出てくるにつれて更新を行っていかねばなりません。多様なブラウザを意識して複数のページを作成するという方針をとれば、図1.7のように更新のたびにそれぞれのページをすべて更新しなければならず、場合によってはこのことが意外な負担になることも考えられます。このことは、ブラウザの機能の問題に限らず、たとえば日本語と英語の両方のページを用意する、といったケースでも同じように発生しうる問題です。要するに、ごく一般的にいえば、より多くの人に見てもらうためにはより多くの手間がかかる(かかりうる)ということになります。

もちろん、さきほど述べたように、機能のある程度限定すれば手間を減らすことは可能です。かわりにこの場合にはレイアウトや表現ということに関して多少我慢しなければなりません。

図1.8 同じ台本(ソース)でも劇団(ブラウザ)が異なれば印象が違ってくる



また、Webに用いられる技術に精通し、各種ブラウザの対応状況を把握していれば、互換性の隙間を縫うようにして、高機能でありながら広範囲のブラウザに対応したページを作成する、ということも可能です。精通や把握といえば大変そうですが、必要とされる技術は、時間と手間さえかけられるなら習得するのもそれほど難しいことではありません。とはいえ、もともとこの種のことに近い位置にいる人でなければ、習得にかかる時間と手間は無視できないものになるでしょう。

結局のところ、万能の解答というものは今のところ存在しません。一般的には、高機能、美しい表示結果、一般性、というプラスの方向性と、メンテナンスの煩雑さ、習得の手間、特殊性、といったマイナスの方向性とのバランスで、自分のWebページでどこまでやるか、ということが決まってきます。このトレードオフのバランス点がどこになるかは人それぞれ、というほかはありませんが、つねにこのことを意識しておくのは大切なことだと思います。

### 1.2.5 Web演劇の「ト書き」 - タグ

さきほど、HTMLというもので書かれた「ソース」が、「ブラウザ」によって表示されるということがWebの一番基本的な仕組みである、と書きました。例え話になりますが、この「ソース」と「ブラウザ」の関係は、図1.8のように、「台本」と「劇団」の関係になぞらえてみることもできます。

演劇の台本には、台詞のほかに「ト書き」といわれるものが書かれています。図1.9のように、登場人物の台詞の間に、登場人物がどのような動きをするのか、台詞はどのように読まれるべきなのか、という指示が書かれているわけです。舞台上で実際に演劇の公演が行われるとき、ト書きは私たちに直接意識されることはありません。そのかわりに、俳優の動きでト書きの指示は実現されていることとなります。

しかし、同じ台本をもとにした演劇であっても、公演を行う劇団が違えば見た目は随分違ってることがあります。同じ劇団であっても演出家が違えばこれもまた随分違うことがあります。極端な話、シェークスピアなどの古典劇を現代風にアレンジしたような舞台では、見た目だけでいえば全く別の舞台といえなくもありません。また、同じような舞台構成、演出であ

図1.9 ト書き入り架空の台本

```

サトル，舞台中央付近へ．

サトル「割れんばかりの大音声で）神さま，食べても食べても太らない体をください！」

カヲル，上手へ．

カヲル「(ささやくような声で)そんなものはありません」

```

っても劇団や俳優が違えば細かいところは様々に違いがあります。それでも、同じ台本をもとにした劇は、見ている人にとって「ああ、これはマクベスだ」「これはロミオとジュリエットだ」ということがはっきりわかるものです。このことは、さきほどのブラウザによる表示の違いがあっても、含まれている情報は共通している、ということに通じています。

Webの台本であるHTMLにも、ちょうど演劇のト書きに相当するものが存在しています。それが「タグ」と言われているもので、図1.10がその例です。図1.10は、台本の例として挙げた図1.9を少し意識しています。ここで示されているタグは"<CENTER>"、"<FONT>"の2種類です。図1.10をNetscapeで表示させたものが図1.11になります(なお、図1.10は説明のために必要最低限の部分に絞ってあります。「正しい」とされているHTMLとして書くためにはもう少しほかのタグを書き加えてやる必要がありますが、ここではそのことは忘れておきます)。

演劇台本のト書きにも書き方のルールがあるようですが、台本の場合には読み手が人間であることもあり、多少ルールから外れた書き方をしても、(こいつ、台本の書き方も知らないのか、と言われるかも知れないにせよ)ト書きとわかる範囲の逸脱であればどうにか書き手の意図は汲み取ってもらえるでしょう。しかし、古くから言われているようにコンピュータというものは融通の利かない存在です(融通を利かせるようにしようという研究は現在活発に行われていますが、現状ではまだまだと言わざるをえません)。Webにおけるト書きであるタグの書き方にもきっちりとしたルールがあり、これを守らないと意図した結果が得られないことがよくあります。

タグはブラウザに対する指示を行うもので、指示の種類はいろいろありますが、書き方のルールはシンプルです。それは、

- (1) タグは、"<"と">"で挟んで書く
- (2) <CENTER>に対して、</CENTER>のように、<XXX>というタグを書いたら、必ず、</XXX>という対応するタグを書いて閉じる。言いかえると、<XXX>と</XXX>は必ず

図1.10 タグを使ったソースの例

```

サトル

<CENTER>

「<FONT SIZE="+3">神さま、食べても食べても太らない体をください！</FONT>」

</CENTER>

カヲル「<FONT SIZE="-3">そんなものはありません</FONT>」

```



図1.12 図1.10の表示イメージ (Lynx)



別々のタグにすると煩雑になってくることは想像できると思います。大きさだけにしても、たとえば6種類のサイズを指定できるようにするとしたらそれぞれのサイズごとに違ったタグを決めてやらなくてはなりません。色は最近のコンピュータでは1677万色指定できるものが多くなってきていますから、仮りに色をすべて別々のタグで指定するというを行うとしたら1677万個ものタグが必要になりますが、それはあまり現実的ではありません。

こういった事態を避けるために、タグにはパラメータという、追加の指示を与えてやることのできるようになっていきます。パラメータというのは、`<FONT"SIZE = +6">`でいえば、「`SIZE = +6`」の部分になります。このあたりは台本のト書きとは違い、コンピュータならではの書き方のスタイルとを考えてください。

少々説明が長くなりましたが、ルール(1)とルール(2)を踏まえて、もう一度、図1.10と図1.11を見てみましょう。二つのルールを踏まえて図1.10を見ると、どの部分がト書き = タグであるか、ということが見えてくると思います。図1.10と図1.11を比較すると、表示された結果である図1.11にはト書き = タグの部分が全く現れていないことがわかります。

もう一つ参考のために、図1.10のHTMLソースを先ほどのLynxで表示した画面を図1.12に示します。図1.12を見ると、真ん中に寄せる、という`<CENTER> ~ </CENTER>`の効果は得られていますが、文字の大きさを指定する`<FONT> ~ </FONT>`の効果は現れていません。このように、ブラウザ側で対応不可能なタグの指示については無視されてしまう、ということも覚えておいてください。このことは大したことはないように思えるかもしれませんが。しかしたとえば、`<FONT"SIZE = +6"> ~ </FONT>`に続けて、「上記で大きな文字で表示されている部分については...」というような解説文を書いたりすると、文字サイズの変わらないブラウザで見ている人にとってちんぷんかんぷんな内容になってしまう、というようなことが考えられます。レイアウト指定はあくまでも補助的なもの、ということを頭の片隅に置いておいてください。