

第2章 Webの仕組み

前章でも触れましたが、いわゆる「ホームページ」を閲覧するための機構全体は、本来 WWW(World Wide Web) と呼ぶべき物です。しかし、“WWW”と発音するのは難しいことと、World Wide であることは自明なこととして、“Web”と簡略化した呼称も多く用いられるようになってきました。以下、本書でもこの“Web”という呼び方を使うことにします。

この章では、Internet で提供されているサービスの一つであるこの Web の全体像を紹介していきます。

2.1 Webの仕組み

Internet 上で提供されるサービスの大部分は、24 時間、Internet のどこからでも使用可能です。これはサービスを提供する側が、故障でもしない限り動き続けるコンピュータだからです。このように、サービスを提供する側のコンピュータをサーバ (Server) と呼びます。反対に、サービスを要求し受ける側をクライアント (Client) と呼びます。クライアントもコンピュータですが、これはユーザが使いたいときにだけ動作させるようになっているのが普通です。使わないときには電気代節約のために電源さえ切られてしまいます。

それならば、サーバはクライアントの要求に応じて起動すればよいようですが、Internet は ISP と契約した不特定多数のユーザが世界中で使用していることを忘れてはいけません。少なくとも全世界のユーザに対して使用可能にしようとするれば、サーバは 24 時間動作し続けている方が合理的なのです。

これは Web でも全く同様です。Web の場合、サービスを提供するサーバ側を Web サーバ (Web Server)、クライアント側で Web サービスを要求するプログラムのことを (Web) ブラウザ (Browser) と呼びます。図 2.1 は、ブラウザが Web サーバ (www.hogehoge.jp) に対し、「ルートディレクトリにある、あらかじめ指定されたファイル (この場合は“index.html”という名のファイル) をよこせ」という要求をした時の動作を示しています。

ブラウザは Web サーバから、閲覧したい Web ページの元になるファイルを Internet を介して取得し、それをユーザが使用しているクライアント側のホストの画面に表示する役割を果たしています。これに対し、Web サーバは要求に応じて自分のディスクに保存されているファイルを送り出しているだけです¹。但し、Web サーバはこの作業を複数のユーザからの要求に同時に答える必要があります。負担が少ない程良い訳です。

¹それだけではないケースもある (第 17, 第 18 を参照)。

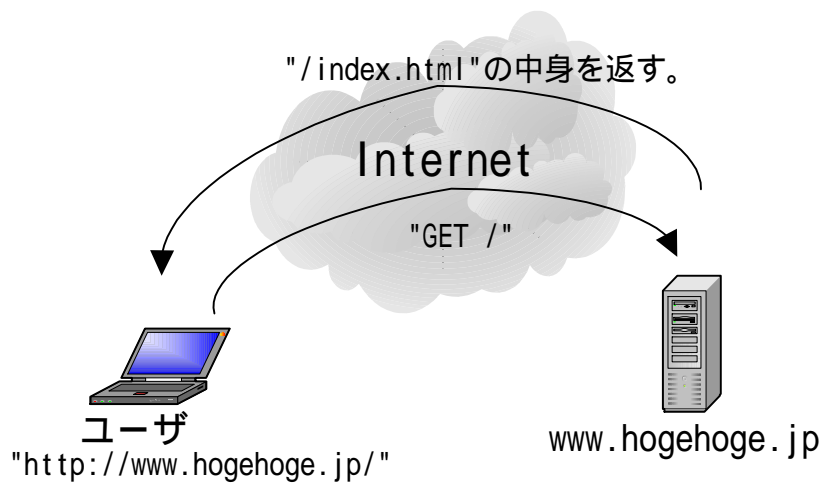


図 2.1: Web の仕組み (概念図)

2.2 絶対パス指定と相対パス指定

Web とは直接関係があるように感じられないかも知れませんが、Web サーバへ Web ページを upload する際に必要となる知識として (第 6 章を参照)、ここではディレクトリ構造について解説します。

ブラウザで Web ページを表示するためのデータが記入されたファイルは Web サーバのディスク (Disk, 大容量記憶装置) に保管されています。ファイルにはそれぞれ名前があり、これをファイル名と言います。データの量が多い時は、複数のファイルに分割しておいた方が扱いやすくなるものです。従って、データの量が増えるほど、ファイルの数も増えて行きます。

何百、何千のファイルが一箇所に固まっていると見通しが悪くなりますし、必要なファイルがどこにあるのか探すことも面倒になります。そこで、ファイルを入れる入れ物を作ることになりました。これをディレクトリ (Directory) と呼びます。Windows で言う所のフォルダ (Folder) と同じ物と思って下さい²。ディレクトリ (フォルダ) もファイルと同様名前が付きます。これをディレクトリ名 (フォルダ名) と呼びます。

ディレクトリにはファイルとディレクトリも入れることが出来ます。従って、ディレクトリの中のディレクトリの中のディレクトリの中の... という具合に、ディレクトリの階層構造を成したものが出来てきます。図 2.2 にこの例を示します。

一番左にある “/” をルートディレクトリ (Root Directory) と呼び、これが全てのディレクトリの元になります。ファイルは全てこのルートディレクトリ以下のディレクトリ置かれることとなります。この場合は、ルートディレクトリの下に “home” というディレクトリがあり、“home” の下に “public.html” と “tmp” という 2 つのディレクトリがあります。そして、“public.html” の下には “cgi-bin” と “bitmap” というディレクトリが存在しています。

今、“public.html”ディレクトリの下に (中に) “index.html” というファイルがあるとしましょう。こ

²本書では区別のため、Windows の場合はフォルダ、Web サーバの場合はディレクトリという呼び方を使います。

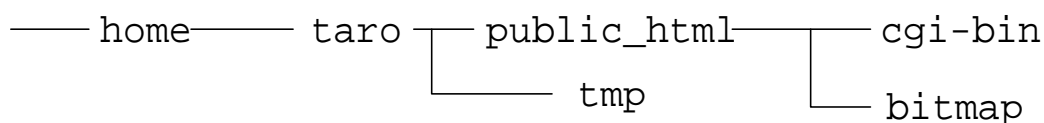


図 2.2: ディレクトリツリーの例

のファイルの位置を示す方法として、絶対パス指定と相対パス指定の 2 つがあります。前者はルートディレクトリを基準とした位置指定方法、後者は現在のディレクトリ (カレントディレクトリ) からの相対的な位置を示すものです。

従って、この “index.html” を絶対パス指定で表示すると

```
/home/taro/public_html/index.html
```

となります。ディレクトリの区切りをスラッシュ “/” で表わしています³。

これに対し、相対パス指定の場合、例えばカレントディレクトリが同じ “public_html” であれば

```
./index.html
```

又は

```
index.html
```

となります。もしカレントディレクトリが “taro” であれば

```
./public_html/index.html
```

又は

```
public_html/index.html
```

となります。そしてカレントディレクトリが “bitmap” であれば

```
../index.html
```

となります。

Web サーバにファイルを要求する際、ファイル位置があらかじめ分かっている時は直接絶対パス指定を使うことが出来ます。しかし、Web サーバに置いておくファイルの中で、同じルートディレクトリ以下の別のファイルの位置を指定する時には、相対パス指定の方が後々都合が良いことがあります。

どちらも場合に応じて使いこなすことが出来るようになります。

³Windows では **ドット**マークで区切るのが普通。欧米ではスラッシュが逆に傾いた「バックスラッシュ」になる。

2.3 ブラウザと HyperText Markup Language(HTML)

ブラウザで Web サーバから取得してきたファイルに書き込まれているデータには、ブラウザ画面に表示させたい形式(フォーマット)が記述されています。この時、表示されたものを Web ページ(Web Page)、その形式を記述する言語を HTML(HyperText Markup Language)と呼んでいます。一昔前でしたら、「ブラウザはパソコン上で動作するプログラム」と言うことも可能でしたが、今では携帯電話でもブラウザが動いていますね。しかし、今の所、携帯電話上のブラウザは PC のそれに比べてハード面での制約が多いので、本書では PC 上のブラウザのみを対象とします。

前述した通り、Web の登場後すぐに Microsoft 社と Netscape 社との間でブラウザ戦争と呼ばれるシェア争いが始まりました。前者のブラウザは Internet Explorer、後者のブラウザは Netscape Navigator です(図 2.3)。

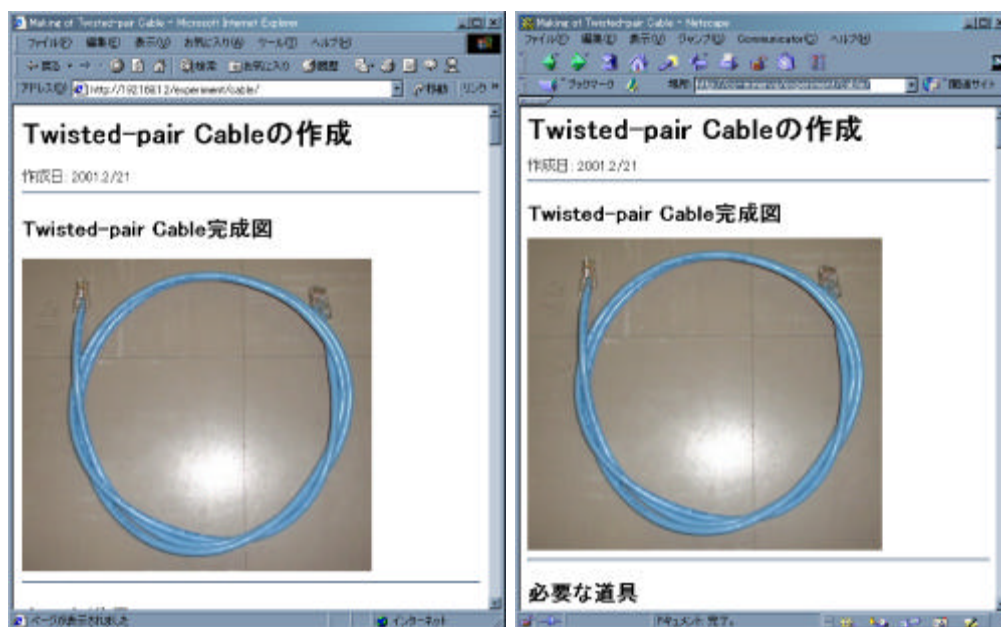


図 2.3: Internet Explorer(左) と Netscape Navigator(右)

現在では前者が圧倒的多数を占めているようですが、後者の愛好者も根強く⁴、今の所は双方で支障なく閲覧できる Web ページを作る必要があるでしょう。本書でもそのような方針で Web ページを作っていく予定です⁵。

2.4 HTML の例

Web ページの例を図 2.4 に、この Web ページを記述した HTML を図 2.5 に示します。

同じ Web ページを表示させると、Internet Explorer でも Netscape Navigator でも殆ど同じよ

⁴筆者もその一人。

⁵一部例外がある(第 16 章参照)。

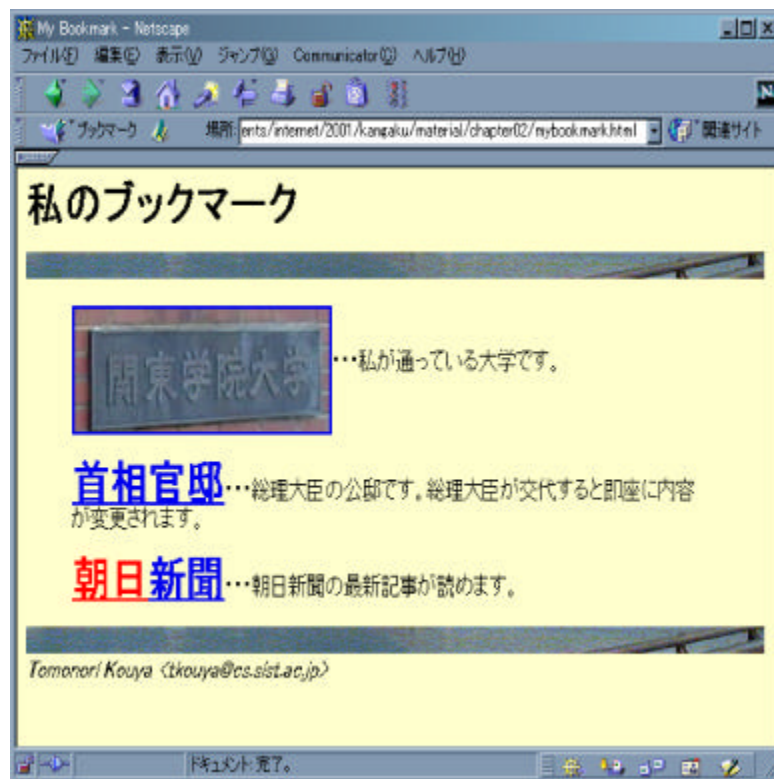


図 2.4: Web ページの例: Netscape Navigator による表示

```

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
  <meta name="Author" content="Tomonori Kouya">
  <meta name="GENERATOR" content="Mozilla/4.75 [ja] (Windows NT 5.0; U) [
Netscape]">
  <title>My Bookmark</title>
</head>
<body text="#000000" bgcolor="#FFFFCC" link="#0000EE" vlink="#551A8B" alin
k="#FF0000">

<h1>
私のブックマーク</h1>
<img SRC="line.gif" height=18 width=640>
<blockquote><a href="http://www.kanto-gakuin.ac.jp/"><img SRC="kangaku.jpg
" ALT="関東学院大学" height=82 width=221 align=CENTER></a> ... 私が通って
いる大学です。
<p><b><font size=+3><a href="http://www.kantei.go.jp/">首相官邸</a></font>
</b> ... 総理大臣の公邸です。総理大臣が交代すると即座に内容が変更されます。
<p><b><font size=+3><a href="http://www.asahi.com/"><font color="#FF0000">
朝日</font>新聞</a></font></b> ... 朝日新聞の最新記事が読めます。</blockq
uote>
<img SRC="line.gif" height=18 width=640>
<address>
Tomonori Kouya &lt;tkouya@cs.sist.ac.jp></address>

<br>&nbsp;
</body>
</html>

```

図 2.5: HTML による Web ページの記述

```
<HTML>
<HEAD>
...
<TITLE>Web ページのタイトル</TITLE>
...
</HEAD>
<BODY>
...
Web ページ本文
...
</BODY>
</HTML>
```

図 2.6: HTML ファイルの構造

うな結果を得るものの微妙な違いがあることも分かります。ブラウザは同じものでも、PC の画面の解像度が異なっていたり、表示可能な色数が少なかったり、ブラウザの **Version** が古かったりするとこの違いはさらに顕著になります。環境によって **Web ページ** の表示のされ方が違うことを認識しておいて下さい。

図 2.5 が **HTML** の一例です。はじめて見る人には、何やら難しげな文字が並んでいますが、よく見ると規則性が分かってきます。

まず、`<単語>…</単語>` という部分が目に付きます。このような `<単語>` を **タグ (tag)** と呼びます。タグは文書の構造を表わすもので、例えば 12 行目から 13 行目の「`<h1> 私のブックマーク </h1>`」は“私のブックマーク”という文字列が大見出しであることを指定しています。タグの種類は他にも多数あり、後の章でも触れますが、「タグを覚えること＝良い **Web ページ** を作ることが出来る」ではありませんので、本書では必要に応じて、使用しなければならないものだけを **pickup** していくことにします。

図 2.5 を単純化して **HTML** の構造の大枠を抜き出すと、図 2.6 のようになります。詳細については次章以降で触れていくことにしましょう。

練習問題

1. 図 2.2 の構成で、“index.html”が“public.html”にあるとする。カレントディレクトリが以下の場合、この“index.html”が存在するディレクトリを示す相対パス指定はどのようなになるかを考えよ。

- /(ルートディレクトリ)
- /home
- /home/taro/public.html/cgi-bin

2. 『「タグを覚えること＝良い Web ページを作ることが出来る」ではありません』という主張の根拠は何か？ この主張に対する反論があれば述べよ。

コラム★ Web ページを作る目的と作成手順

ぼくが多くの人を引きつける魅力あるホームページづくりができないとは思わなかった。それは、「できるまでやめなければ、できる」と考えられた。いままでの仕事だって、いつもそうだったのだ。できないというところでやめなければ、できるのだ。

糸井 重里 (「ほぼ日刊イトイ新聞の本」より)

Web ページの目的

後の章で説明している通り、Web ページはワープロ文書と同様に、Web ページ作成のためのソフトウェア (第 5 章を参照) で簡単に作成することが出来るようになりました。ワープロソフトの中には、ワープロ文書を Web ページに変換する機能を持つものも現れています。そのため、ワープロ文書を作るのと同じ感覚で Web ページを作ってしまう人が多いようです。

それがいけないというわけではありませんが、紙に印刷されることを前提としているワープロ文書と、ディスプレイで表示され、他の Web ページへ自由に移動することの出来る機能を備えた Web ページとは、区別して考えるべきものです。特に Web ページは、広く考えれば全世界の人が理解できることが望ましいのです。少なくとも自分の Web ページの読者対象としている人々が理解できる言葉 (日本語、英語、フランス語 …) で、わかりやすく記述されていなければなりません。

そうすると、このテキストで縷縷説明している「Web ページ作成のテクニック」は大して重要なことではありません。それよりも「分かりやすい文章を書く能力」や「読者に理解しやすい Web ページの構成能力」、そして「余すことなく情報を詰め込み、なおかつ魅力的な Web ページをデザインする能力」を持つことの方がずっと大切なことであり、そして難しいことなのです。

単なるテクニックよりも、構成能力。構成する以上に、魅力的なコンテンツを発見し、人に説明する能力を磨くこと。本書で述べている初歩のテクニックはそのための最初の踏み台に過ぎないことを忘れないようにして下さい。

Web ページ作成の手順

万人に読んでもらうことを Web ページの一番の目的とするならば、Web ページは書店に並ぶ本と同じ工程で作成されなければなりません。違いがあるとすれば、本は印刷・製本され、流通ルートに乗って初めて出版されたこととなりますが、Web ページは作成者が Web サーバに転送し URI を公表すれば誰でも読むことができるようになるという点だけです。自分以外の第三者に読んで貰わなければ意味がありませんから、公開するまでには何重にもチェック (推敲) する必要があります。

す。本だって、最低、著者と編集者のチェックを経る訳ですし、慎重な著者は周囲の人にも読んで貰い、その反応を確認しているのです。

Web ページは Web サーバに転送されて、他の人が読むことができるようになるものですから、自分の手元の PC でチェックを行うのは勿論、Web サーバに転送したものに対しても同様のチェックを行うことが最低でも必要です。この過程を図 2.7 に示します。

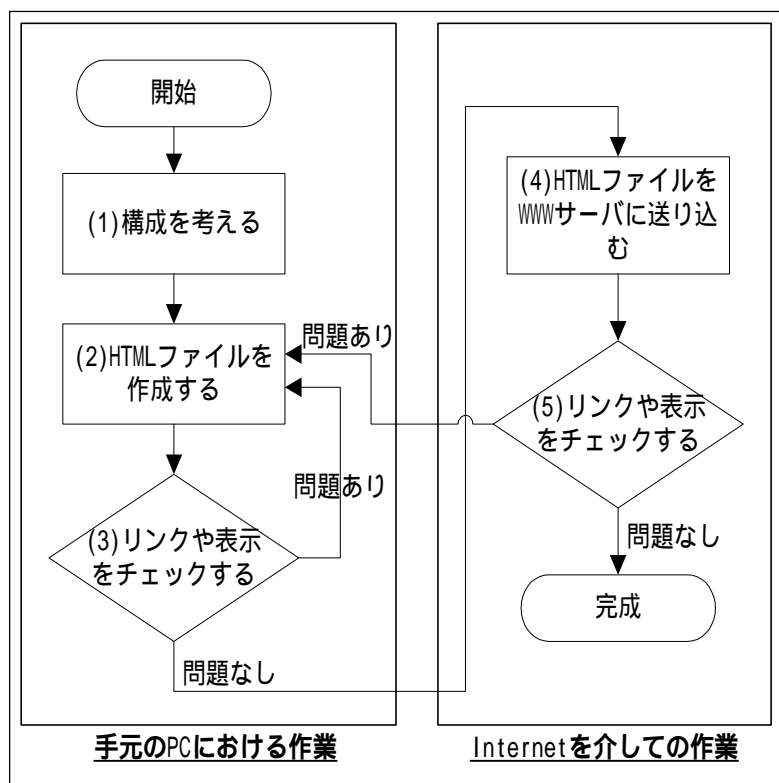


図 2.7: Web ページ作成手順のフローチャート

このチェックの際には、本と同様、自分以外の人に Web ページを見て貰うようにしましょう。自己満足が第一の趣味の Web ページなら兎も角、自分の会社や仕事に関連する情報を提供する Web ページが独り善がりでは人に読んで貰えないばかりか、業績に響く恐れが出てきます。