第4章　テキストエディタでWebページを作る (2/2)

前章に引き続き，テキストエディタでHTMLを書き，Webページを作っていきます。Webページが通常の文書と異なる点は，クリック一つで他のWebページへ異動することが出来るリンク (Hyper link) が使用できることにあります。本性では実際にリンクを張ったWebページを作成した後，前章で作成した自己紹介をテーブル機能を使って装飾してみます。

4.1　リンクがHypertextの命

リンク (Hyper link) はHTMLの要の部分です。では，

1. `http://www.kanto-gakuin.ac.jp/` … 関東学院大学
2. `http://www.kantei.go.jp/` … 首相官邸
3. `http://www.asahi.com/` … 朝日新聞

へのリンクを張ったHTMLを書いてみましょう (図4.1)。ここでは数字付き箇条書きを使っています。

リンクは現Webページから他方への一方向への移動を可能にするものです[1]。また12行目を図4.2のように修正すると，このリンクをクリックすることで宛先が自動的に記入されてメールソフト[2]が起動されます。

4.2　テーブルを作ろう

前章で作った自己紹介のページを，HTMLのテーブル (Table) 機能を使って装飾してみます (図4.4)。

テーブルタグは

[1] 戻る時にはブラウザの「戻る」ボタンを使用する。
[2] メーラ (mailer)，もしくはMUA(Mail User Agent) とも呼ばれる。


```
<HTML>
<HEAD><TITLE>Link</TITLE></HEAD>
<BODY>
<H1>リンクの練習</H1>
<HR>
<OL>
<LI><A HREF="http://www.kanto-gakuikn.ac.jp/">
関東学院大学</A>
<LI><A HREF="http://www.kantei.go.jp/">首 相 官
邸</A>
<LI><A HREF="http://www.asahi.com/">朝　日　新
聞</A>
</OL>
<HR>
<ADDRESS>Tomonori Kouya : &lt;tkouya@cs.sist.a
c.jp&gt;</ADDRESS>
</BODY>
</HTML>
```

図 4.1: リンクの練習

```
<ADDRESS>Tomonori Kouya : &lt;<A HREF="tkouya@cs.sist.ac.jp">tkouya@cs.sis
t.ac.jp</A>&gt;</ADDRESS>
```

図 4.2: リンクの練習

```
<table>
    <tr>
        ...
        <td>セルの内容</td>
        ...
    </tr>
    <tr>
        ...
    </tr>
    ...
</table>
```

という構造になります。<tr> タグで囲まれた部分は行 (row, 横方向) の内容を，<td> タグで囲まれた部分は行を列 (column, 縦方向) に細分化したセルの内容を記述します。行，列，セルの関係を図示したのが図 4.3 です。

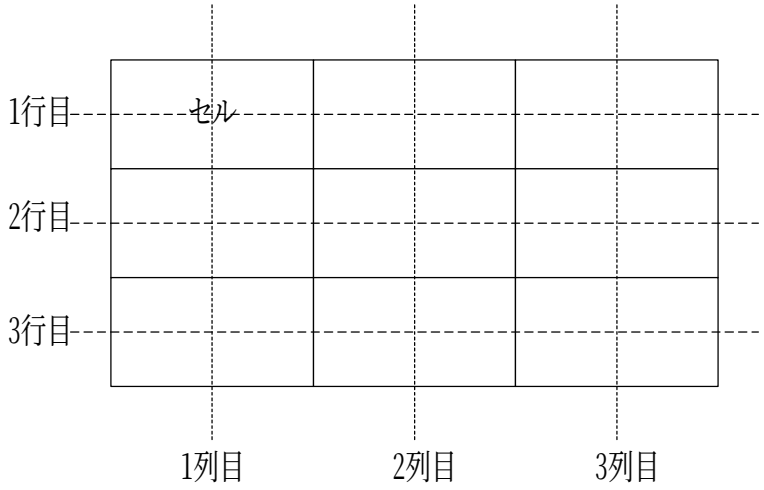

図 4.3: テーブルの行，列，セル

4.3 テーブルの中にテーブルを作ろう

更に先ほどの例の最後に「その他」の列を付加し，その内容の部分にテーブルを入れ子にしてみましょう。

練習問題

1. 図 4.5 の趣味・資格の次に「自己アピール」の項目を作り，自己アピールの文章を挿入せよ。


```
<html>
<title>最初の Web ページ</title>
<body>
<h1>自己紹介</h1>
<hr>
<table border>
<tr>
<th>氏名</th>
<td></font><dd><font size=5 color="blue"><s
trong>幸谷　智紀</strong></font></td>
<tr>
<tr>
<th>学科</th>
<td><a href="http://www.kanto-gakuin.ac.jp">
関東学院大学</a>工学部機械工学科</td>
</tr>
<tr>
<th>学籍番号</th>
<td>9999999999</td>
</tr>
<tr>
<th>メールアドレス</th>
<td>tkouya@mbox.kanto-gakuin.ac.jp</td>
</tr>
</table>
</body>
</html>
```

図 4.4: テーブルを使った自己紹介のページ


```
<html>
<title>最初の Web ページ</title>
<body>
<h1>自己紹介</h1>
<hr>
<table border>
<tr>
<th bgcolor=green>氏名</th>
<td></font><dd><font size=5 color="blue"><s
trong>幸谷　智紀</strong></font></td>
<tr>
<tr>
<th bgcolor=red>学科</th>
<td><a href="http://www.kanto-gakuin.ac.jp">
関東学院大学</a>工学部機械工学科</td>
</tr>
<tr>
<th bgcolor=yellow>学籍番号</th>
<td>9999999999</td>
</tr>
<tr>
<th bgcolor=black>メールアドレス</th>
<td>tkouya@mbox.kanto-gakuin.ac.jp</td>
</tr>
<tr>
<th>その他</th>
<td>
<table>
<tr>
<th>趣味</th>
<td>読書</td>
</tr>
<tr>
<th>資格</th>
<td>普通免許</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

図 4.5: テーブルの中のテーブル

2. 枠無テーブルを利用して，図 4.6 のような自己アピールのページを作れ。

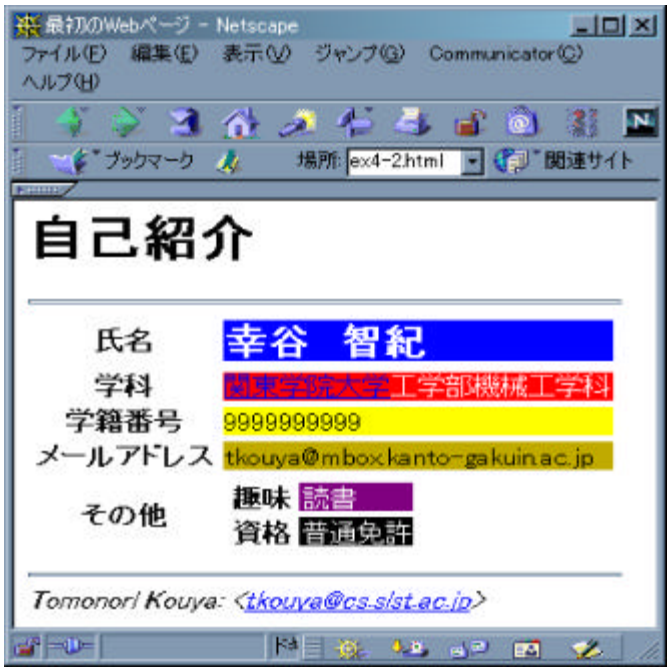

図 4.6: 枠無テーブル版の自己アピール

3. (自由課題) 日本政府の省庁の親子関係が一目でわかるような Web ページを，テーブルを利用して作れ。セルに背景色をつけるなどして工夫すると読みやすいものを作ることが出来る。

コラム★Webサーバとブラウザと通信の実際

> WWW のしくみは拍子抜けするほど簡単だ。単純に TCP/IP のソケットを WWW サーバに対して開いて “GET パス名 HTTP/1.0<CR><CR>” という文字列を送れば，WWW サーバの方は何も考えずに，たとえ相手が親の仇であろうが企業スパイであろうが，そのパス名のファイルを MIME 形式にして返してくる。
>
> 志村 拓・榊 隆 (「インターネットを 256 倍使うための本 Vol.1」より)

殆どの人は，Web ページを見る時，サーバとブラウザ（クライアント）でどんなやり取りが行われているのか，ご存知ないでしょう。また，そのやり取りを盗聴できるなんて考えたこともないでしょう。

別段，そんなことを知らなくても Web ページを見ることも作ることも出来ます。しかし，車のエンジンの構造を知っておけば，アクセルを踏み込むとエンジン音が大きくなる理屈がわかります。エンジンが掛からない時は，セルモータがいかれているか，それを動かすバッテリが上がっている可能性を考え付くことも出来ます。こと，IT 関係の技術は原理原則を知っておくと，困った時に大変役に立ちます。

ここでは実際に Web サーバとブラウザが通信している内容をご覧に入れます。

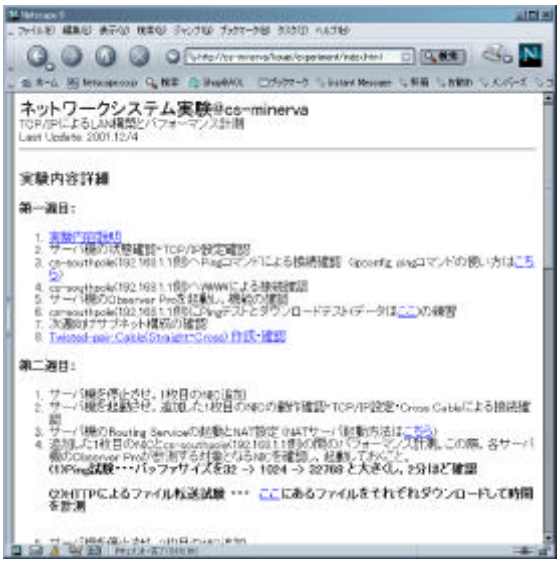

図 4.7: サンプルに使った Web ページ画面

データは小包 (パケット) になってやってくる

第 1 章でも述べていますが，Internet のアイディアはパケット通信 (packet communication) から発生しました。これは全てのデータをパケットという単位に分割し，受信先で再構成するというものです。現在の Internet でもこの精神は全くそのまま生かされており，大きいデータは複数のパケットに分割されて届けられます。Web でもそれは全く変わりません。大き目の HTML ファイルやそこに張り込まれている画像はパケット単位でバラバラにされて送られているわけです。

ここで取り上げる例 (図 4.7) は，画像の含まれていない HTML ファイルですが，１パケットに収まるサイズではないので，最終的には 4 つのパケットに分割されて送信されました。まずこのことを頭に入れておいて下さい。

通信の例

この例では図 4.8 にある通り，クライアント側 (IP アドレス: 192.168.1.11) で Netscape 6 を起動し，Web サーバ (133.133.133.133) にある”/kougi/experiment/”をアクセスしている例です。この通信で飛び交ったパケットは全部で 14 個 (図 4.8 内の (1)〜(14)) あるのですが，そのうち実際に Web ページ転送に必要なデータを運んでいるのは 5 個のパケットです。それ以外のパケットは，送られてきたパケットに対する確認だったり ((5), (7), (10))，Web ページ転送のための準備 ((1)〜(3))・終了 ((11)〜(14)) のために使われています。

ではやり取りされている内容を具体的に見ていくことにしましょう。(1)〜(3) までのパケットのやり取りが行われて[3]，ようやくブラウザからアクセスしたいページの位置を知らせるパケット (4) が Web サーバへ届けられます。その内容は以下のようになっています (一部省略してあります)。

```
GET http://cs-minerva/kougi/experiment/ HTTP/1.0
Accept: */*
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
 ...
```

第 1 行目で，アクセスしたいページの位置とそのアクセス手段 (GET)，Web ページ転送の手順方式 (HTTP 1.0) を知らせていることが分かります。以後のデータにも全て意味があり，例えば 5 行目はブラウザ自身の情報 (User-Agent) です。“Mozilla”とは Netscape Navigator のことを示しています。

こうしたブラウザからの要求を受けて，Web サーバはそのページを検索し，該当の HTML ファイルを探してブラウザに送り返します。以下は (6) で運ばれているデータの一部です。

[3]TCP 接続を行うための 3-way handshake(握手) と呼ばれている処理。

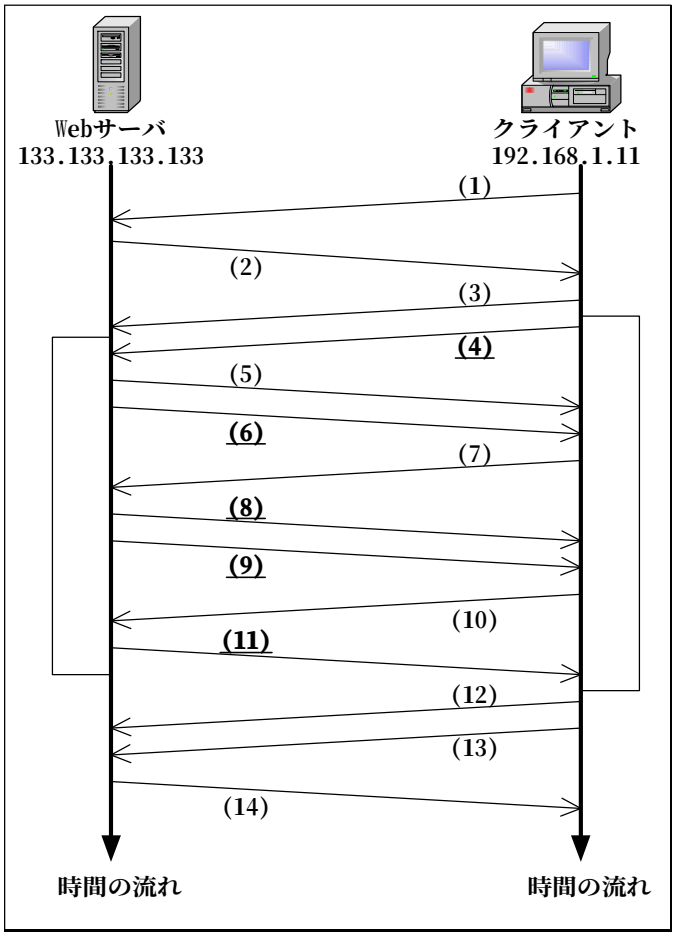

図 4.8: パケットのやり取り

```
HTTP/1.1 200 OK
Date: Wed, 16 Jan 2002 08:03:44 GMT
Server: Apache/1.3.9 (Unix)
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
 ....
```

その後，この Web ページの内容が分割されてブラウザに送信される訳です ((6), (8), (9), (11))。全てのデータを送り終わった時点で，Web サーバが通信の終了を宣言し ((11))，それを受けてブラウザも通信を切断します ((12)〜(14))。

リンクをクリックするごとに，このような通信が Web サーバとブラウザとで繰り返されているわけです。

おまけ：パケットキャプチャについて

以上の内容は，パケットキャプチャと呼ばれるソフトウェアを使って，通信内容を解読して作成しました。一般に，企業や大学構内で使用されている Ethernet と呼ばれる LAN(Local Area Network) でやり取りされる通信内容は全くガードされておらず，ほぼ丸見えといっていい状態です。最近ではスイッチングハブが使用されることが多いので，自分の通信内容が全て他人に覗かれる危険は少なくなりましたが，パケットそのものにはセキュリティ(安全性) という概念がないということは覚えておくべきでしょう。

以下に，パケットキャプチャの実際の内容をご覧に入れます。何が書いてあるのか，普通の人には全く皆目見当もつかないものですが，コンピュータ通信を勉強するとだんだん理解できるようになります。近くに知識のある人がいるようでしたら，この内容を解説して貰ってみて下さい。これは (8) のパケットの内容の一部です。

```
Packet 8:  00:50:DA:93:13:3C -> 00:50:DA:93:C3:07
    Network:  Ethernet
    Frame type:  802.3,  Frame size:  1518
    Time:  17h:22m 19.703 510s,  Diff. time: 0.000973
    Date:  Wed Jan 16 2002
IP,    133.133.133.133 -> 192.168.1.11
    Source IP:  133.133.133.133,  Destination IP:  192.168.1.11
    Version: 04,    IP header length: 05 (32 bit words)
    Service type:  0:  Precedence: 0, Delay: Norm, Throug: Norm, Reliab: Norm
    Total IP length: 1500
    ID: 3675h
```

```
    Fragment flags: [10] - don't fragment - last fragment
    Fragment offset: 0
    Time to live: 254
    PROTOCOL: [6] TCP
    Header checksum: 8199 (GOOD)
TCP  ACK,   [85] -> [1035]
    Source port: [85]    Destination port: [1035]
    Sequence number: 3877611471,   Acknowledgement: 4093031165
    TCP header length: 05 (32 bit words),   Window: 8760
    TCP data length: 1460,   Checksum: BC0Bh (GOOD)
    Sequence number + TCP data length: 3877612931
Data
    0000  CC 4E 49 43 82 CC 93 AE  8D EC 8A 6D 94 46 81 45     NIC の動作確認・
    0010  54 43 50 2F 49 50 90 DD  92 E8 81 45 43 72 6F 73    TCP/IP 設定・Cros
    0020  73 20 43 61 62 6C 65 82  C9 82 E6 82 E9 90 DA 91    s Cable による接
    0030  B1 8A 6D 94 46 3C 2F 6C  69 3E 0D 0A 0D 0A 3C 6C     確認</li>....<l
    0040  69 3E 0D 0A 83 54 81 5B  83 6F 8B 40 82 CC 52 6F    i>.. サーバ機の Ro
```