

## 文書構造を意識したホームページ演習での ASP (Active Server Pages) の利用

末 木 俊 之

Use of ASP in the Homepage Exercise Which is  
Conscious of Document Structure

Toshiyuki SUEKI

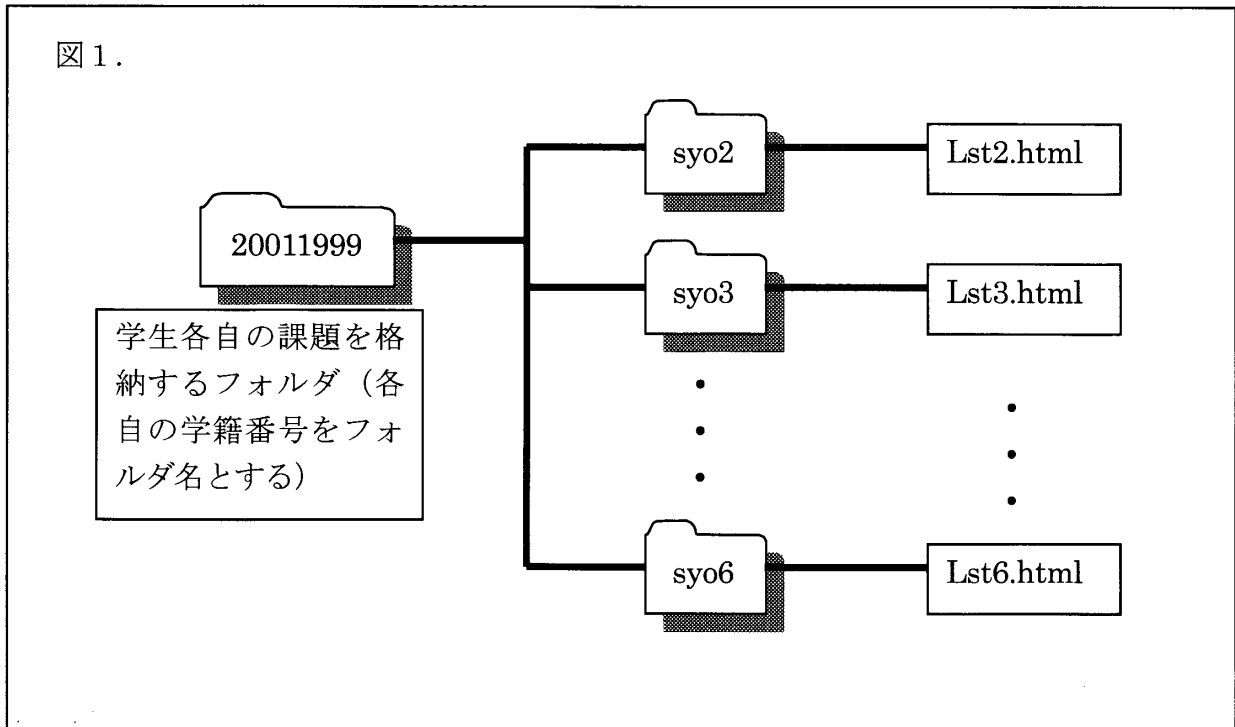
### 1. 文書情報処理としてのホームページ作成演習

コンピュータリテラシー教育での演習内容として、大学1年生向けには、ワープロ・表計算ソフトの基本技能を中心とした演習を実施している。そしてそれに続き、文書処理演習として、ワープロ検定を目標としたビジネス文章作成技術を教えるワープロ演習と、ホームページ作成演習を実施していた。ホームページ作成演習では、ホームページ作成ソフトは使用せずに、“メモ帳”、“ワードパッド”などのシンプルなテキストエディタを使い、HTML言語を直接記述し、HTMLテキストファイルを作成するやり方で演習してもらっている。教室のIIS (Internet Information Server)の動作しているサーバ内ディレクトリに学生各自のディレクトリを用意し、そこに学生各自の自己紹介をするような内容で写真、絵入りの2、3の簡単なホームページを作成してもらった。クラス毎に用意したメインのホームページから、学生各自のメインホームページに展開し、さらにそのページから2、3のページにリンクし、またリンク先ページからメインページに戻れるようにリンク付けしてもらった。またフレーム分割されたページの作成なども演習に入れてみた。

このように、文章処理演習という名目でワープロ・ホームページ演習を実施していたが、特にワープロ演習とホームページ演習には関連性があるものでは無かった。またホームページ演習では、基本的ないくつかのタグを紹介し、演習手順を説明するプリント資料を学生に配布するのみで、漠然としたホームページ作成演習の感が否めなかった。授業終了後のアンケートでは、ホームページ演習に教科書を使って欲しいという希望を持つ学生もいた。漠然とした演習で、文章処理としてのホームページの体系的な説明が乏しいという印象を与えているということだろう。

ホームページは単に漠然と情報を表示するというだけの機能を有するのではなく、データベースと連携し、データの発生・変更・変換・移動・削除などの操作ポイントとしての役割が大きくなってきている。その場合には単に情報が漠然と表示されるだけではだめで、ホームページ内にある情報が、データとして操作可能なものになっていなければならない。データを表現するためにはXML (eXtensible Markup Language)が使用されてきている。このマークアップ言語により情報のデータ構造を厳密に表現して、ホームページに格納することができた。こ

図 1.



のによりホームページ上での情報操作が容易になった。ホームページ上での情報の表示スタイル（レイアウト）は、ホームページとは別にスタイルシートを作成し、スタイル付けを行うことができる。同様にデータをホームページ内に格納すること（XMLなどを使って情報のデータ構造を厳密に定義しているホームページ）を意図しないホームページであっても文書構造を厳密に定義して記述し、表示スタイル（レイアウト）はホームページテキスト本体とは別にスタイルシートにて定義する方法が推奨されている。また文書構造の曖昧な記述を許すHTML言語を、XMLのように厳密に意味・文書構造を表現する言語として定義しなおされた言語としてXHTMLが登場し、今後はこの言語がホームページ記述言語として浸透していく様子である。HTML言語と大きな食い違いは無く、マークアップ言語として厳密に記述するということが強調されているようだ。ホームページテキストの文章構造を厳密に記述することにより、ホームページ内にある要素をJava-

scriptなどの言語にて操ることも容易になる。またスタイルシート使って特定の構造に特定の表示スタイルを付与することもホームページテキストが厳密・正確な文章構造で記述されていなければ難しい。

このようにホームページにおいては、情報（データ）のデータ構造を厳密に記述すること、また文章構造を厳密に記述すること2つの観点から体系的に学ぶ必要があるようだ。

XMLによりデータ構造を厳密に記述するというのは、データベースも絡み工学的で専門的であるので、まずは文章構造の厳密な定義という観点からの学習ということだろう。そして厳密な文書構造を反映したホームページが作成できるようになったら、次のステップとしてホームページデザイン（スタイルシートによる）の学習に進むという流れが考えられる。

個人情報公開するホームページを手早く作成するという事なら、文書構造を厳密に意識してホームページを作成する必要もないだろうが、膨大な情報を公開するケースでは、ホーム

ページも本（ブック）と構造が似てくるようである。表紙があり、目次があり、章があり、章の頭には見出しがあり、節があり、段落があり、必要ならば章・節の内容を短くまとめた概要のブロックがあり、本文の流れからちょっと離れた内容の記述を格納するにはコラムのブロックを作り、最後には、著者・出版社・出版日などの情報を入れるブロックを入れる。

そのようなホームページでは、XHTML 言語にて厳密な文書構造を表現したホームページを作成することが望ましいようである。複数人にて作業する場合には、章ごとに別ホームページとして分担し作成することになるだろう。その場合には全員がコンセンサスを取り、全て同じ文書構造を持ったホームページとして仕上げる。そしてそれら章ごとに分けて作成されたホームページを同じ表示スタイル（レイアウト）にて表示させるように、1つのスタイルシートを用意し、全てのホームページから参照させることにする。複数のホームページを同じ文章構造を持ったホームページとして作成してあるならば、表示スタイル（レイアウト）が統一のとれたホームページ群となる。最後に目次と表紙に相当するホームページなどを作成すれば、全体として1つの本（ブック）と類似の体裁をもったホームページが出来上がることになる。

2001年度のホームページ演習では、教科書として『ユニバーサル HTML/XHTML』神崎正英著 MYCOM 毎日コミュニケーションズ社刊を使用して演習を実施した。この書籍は、ホームページを本（ブック）の文書構造と対比させ、体系的に HTML 言語を教授する内容を持っているように見受けられる。

## 2. 文章構造を厳密に記述したホームページの作成

### 2.1 章単位のホームページ作成

採用した教科書2章から6章までをそれぞれ、独立したホームページとして作成してもらった。また画像入りホームページなどもあるため、章ごとにフォルダを作成し、そのフォルダ内に、該当する章のホームページ、必要な画像ファイルなどをまとめて格納してもらった。（図1. 参照）例えば第2章に該当するホームページなら、“syo2”という名称のフォルダ内に、“Lst2.html”という名前で作成してもらう。

各章のホームページに共通な決め事としては、以下の①～④程度である。

①章のタイトルを<TITLE>タグ内に記述すること。

②章の先頭では、<h1>タグで章タイトル文字列を記述すること。

さらに細かい節があるときは、必要に応じて<h2>、<h3>、<h4>、<h5>、<h6>タグを使用する。

（図2の例だと、2. 1節タイトルは、<h2>タグで、2. 1. 4節タイトルは、<h3>タグで記述している。）

③段落タグ<p>をきちんと使用すること。段落分けを意識して、タグの終わり</p>もきちんと記述すること。

④その他タグも開始・終了、タグの入れ子関係を正しく記述する。

完全に教科書の内容をホームページに再現するつもりはなく、演習の対象となる例題とその章・節の見出しをきちんとホームページに記述してもらった。

使用した教科書に従って、各種タグの演習を、例題をホームページ上に記述してもらうことによって1通り演習してもらった。

（図2.）は第2章のホームページの一部である。

```

<HTML>
<HEAD>
  <TITLE>第2章：情報のかたちと文章の要素-ユニバーサルHTML/XHTML</TITLE>
</HEAD>
<BODY>
  <h1>第2章：情報のかたちと文章の要素</h1>
  <h2>2.1 情報共有の準備</h2>
  <h3>2.1.4 要素のマークアップ：構成要素を記号で印付けする</h3>
  <p>◇例2. 2 <BR>
    ここが<strong>大切な内容</strong>なのです！！
  </p>

  <h3>2.1.5 要素の属性：構成要素の性質を示す</h3>
  <p>◇例2. 3 <BR>
    <element attr="vall" attr="val2">ここに何か書きます！</element>
  </p>

  <h2>2.2 HTMLと情報のかたち</h2>
  <h3>2.2.4 パラグラフ要素でトピックを明確に示す</h3>
  <p>◇例2. 2 </p>
  <p class="abstract">
    「パラグラフ」（段落）とは単なる改行ではなく、
    「ある1つの考えを述べるための文の
    集まり」という文章の基本単位です。
  </p>
  <p>パラグラフの表示方法も、ブラウザや環境によって異なります。
    多くのブラウザは、パラグラフの間に1行分の空白を置くようです。
    しかしそれは初期のブラウザの表現に倣っているだけで、そのよう
    な決まりはありません。
  </p>

```

図2.

### 3. 表示スタイルの付与（スタイルシート）

#### 3.1 スタイルシートの学習

1通りのタグについて学び、いくつかのホームページを作成した次のステップは、スタイルシートの学習である。使用した教科書（『ユニバーサルHTML/XHTML』神崎正英著MYCOM毎日コミュニケーションズ社刊）では第6章の内容となっている。スタイルシート言語は、カスケーディングスタイルシート（Cas-

ading Style Sheet=CSS）である。

第6章の例題を、ホームページ上に記述してもらい、スタイルシートを“default.css”という名称で第6章のフォルダに作成してもらった。フォルダ“syo6”内に、ホームページファイル“Lst6.html”とスタイルシートファイル“default.css”が作成されることになる。

このスタイルシートに関しては、ワープロ機能と対比するとおもしろい。

図 3.

| ワープロ機能としての分類                     | ワープロ機能詳細項目  | 対応するスタイルシートのプロパティ |
|----------------------------------|---|-------------------|
| 文字書式 (文字に関する文字設定)                | サイズ指定   | font-size         |
|                                  | 文字の太さ指定 (太文字)   | font-weight       |
|                                  | フォント指定  | font-family       |
|                                  | 文字色指定   | color             |
|                                  | 斜め文字  | font-style        |
|                                  | 文字飾り (下線、上線、取り消し線、点滅)   | text-decoration   |
|                                  | 文字の網掛け (文字背景の色)   | background-color  |
|                                  | 上付き文字、下付き文字   | vertical-align    |
|                                  | 文字間隔の指定   | letter-spacing    |
| 罫線 (ホームページではテーブル)・テキストボックスに関する書式 | セル (テーブルのセル要素) 内での文字配置 (上詰め、中央揃え、下詰め)                           | vertical-align    |
|                                  | 罫線のスタイル設定   | border-style      |
|                                  | 罫線の太さ設定   | border-width      |
|                                  | 罫線の色設定  | border-color      |
|                                  | マージン (ボックス要素と外の要素との間隔)  | margin            |
|                                  | 罫の高さの指定   | height            |
| テキストボックスに関する書式                   | 罫の幅の指定  | width             |
|                                  | パディング (外罫と内部の文字との間隔) (Internet Explorer V5.5では、テーブル要素では働かなかった。) | padding           |
| 段落書式 (段落に関する設定)                  | 文字の水平位置 (左寄せ、センタリング、右寄せ)  | text-align        |
|                                  | 均等割付  | text-justify      |
|                                  | インデント (上記マージン : marginでも同様の効果をつけられると思われる。)                      | text-indent       |
|                                  | 行間隔の指定  | line-height       |
|                                  | 段落の前後の間隔の指定   | margin            |

コンピュータリテラシー演習にて使用しているワープロソフトは、マイクロソフト社のワードであるが、ワープロ基本技能として、文字(文字書式)に関する機能、段落書式に関する機能、罫線、テキストボックスなどがまず必須の技能として学ぶべきものだと思われる。これらワープロ基本機能とスタイルシートの機能(プロパティ)は対応付けができる。(図3. 参照。『スタイルシート辞典』アंक著 株式会社翔泳社刊を参照して作成。)ワープロ文章上で表現できる表示スタイルのほとんどは同様にホームペー

ジ文章上でも表現できそうである。

実際既にワープロソフトの中には、ワープロ文章を作成することが、ホームページ文章を作成することと同義になっているソフトも存在しているようである。そのようなワープロソフトの場合では、ワープロで文章を作成することにより、文章構造を独特なXML文章として記述するテキストファイルと、そのXMLテキストファイルに表示スタイルを付与するスタイルシートファイルが作成されているようである。そしてそのXMLファイルをブラウザソフトで開

くとホームページとしても問題なく表示されるようになっていると思われる。ワープロとホームページがXMLとスタイルシートの技術によって結合されたということが言える。ワープロ文章とホームページ文章の区別が無くなる時代になるということだろう。

2001年度の演習では、(図3.)に記述したプロパティを1通り演習してもらっただけであるが、学生の理解を確かめるために、1年次にワープロ演習で実施した課題と同じものをスタイルシートを使ってホームページとして作成してもらうような演習を用意すると面白いかもしれない。ワープロ文章とほとんど同じレイアウト表示のホームページを作成できる筈である。

### 3.2 統一スタイルシート作成

使用した教科書に則って演習を進めることにより、章ごとに独立したホームページが作成されている。この段階では、各ホームページに表示スタイル付けがされていない。ブック(本)らしい体裁のスタイルを付与し、見栄えのするホームページに仕上げる必要がある。

教科書『ユニバーサルHTML/XHTML』P.85(例6.27)にブック(本)の体裁を付与したホームページの例題がある。この例題を1つのホームページ(ホームページは“Lst6-2.html”、スタイルシートは、“syo6.css”。フォルダ“syo6”内に作成)として作成してもらった。ホームページは、ほぼ教科書例題通りのものである。またスタイルシート(図5.参照)もほぼ教科書の例題通りであるが、ホームページに背景画像を付けるなど、多少アレンジしてみた。背景画像は、好みのフリー素材の画像を学生各自のフォルダにコピーしてもらって作業した。

完成した“Lst6-2.html”ホームページの様子は(図4.)に示した。

スタイルシートにより付与した表示スタイルの概要は、以下のア)～エ)である。

ア). 章タイトル文字列を大見出し(h1タグ)のブロックとし、文字に背景色を付ける。

イ). 節タイトル文字列(h2タグ)ブロックは、文字に下線を付ける。-3文字分の左インデントを付ける。(本文書き始め位置から、左に3文字分ずらす。)

ウ). 大見出し以外のホームページ全体を、1つのブロックとし(クラス属性“maintext”のブロック)、6文字分左インデントを付ける(段落書き始めの位置を右に6文字分下げする。)などのスタイルを付与する。

エ). コラムのブロックは(クラス属性“box-column”のブロック)、枠線で周囲を囲み、背景色をつけ、右寄せで表示させる。

以上で本(ブック)らしい体裁を実現するスタイルシート(“syo6.css”)が用意されたことになる。

このスタイルシートを第6章フォルダ(“syo6”)の上位フォルダに名前を変更してコピーする。

“book.css”という名称でコピーする。そして、これまで作成した章ごとに独立したホームページにこのスタイルシートファイル(“book.css”)を参照させる。これで各章のホームページを、(図4.)の表示スタイルで統一することができる。(関連図は、図13.参照)

## 4. 本(ブック)構築

### 4.1 目次付きフレームホームページ作成

統一した表示スタイルを持つ章ごとに独立したホームページを左右にフレーム分割されたホームページにまとめる作業を行った。各ホームページにおける見出しタグ(h1～h6)文字列にアンカーを付け他のホームページから直接各見出し位置にリンクできるようにする。そして

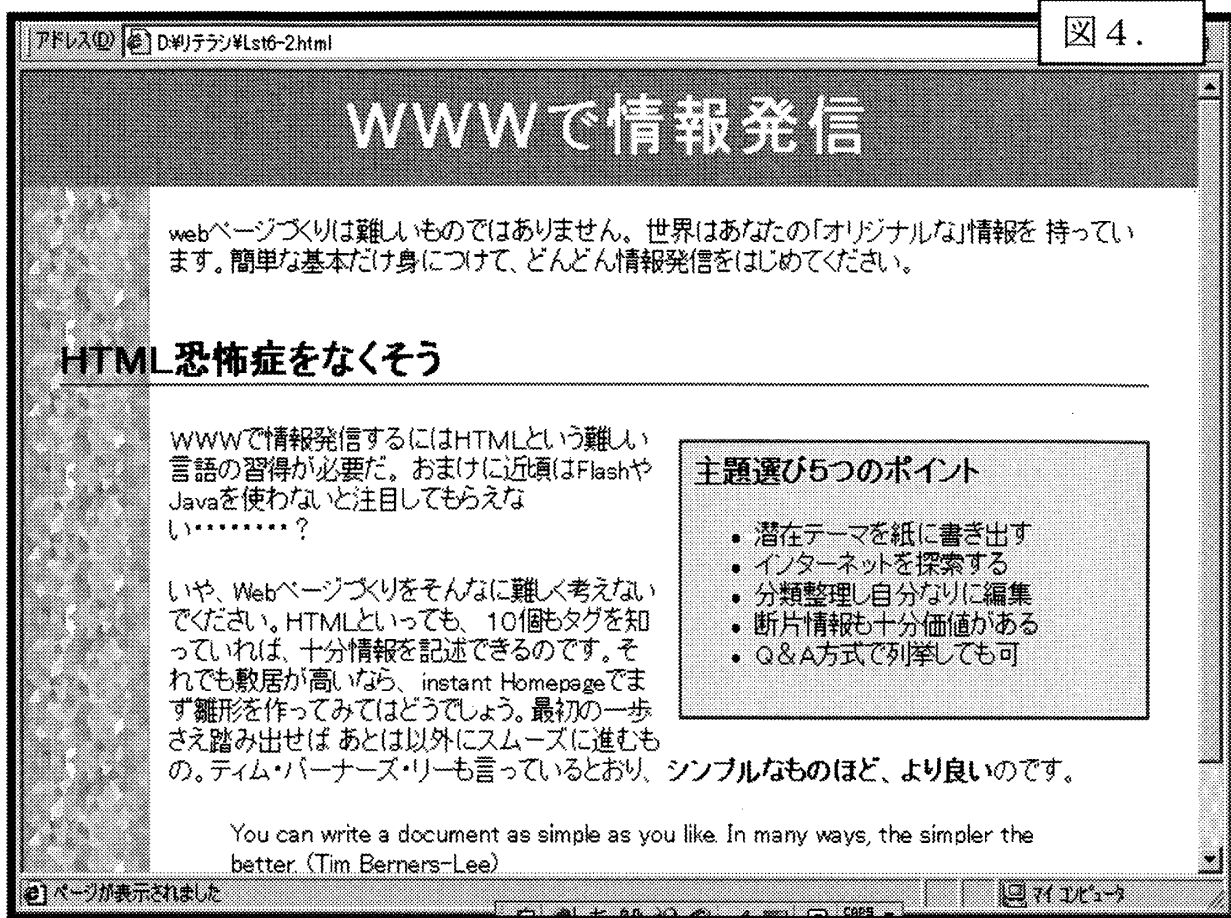


図 4.

```

body {margin : 0 ;
      background-image : url (backg118.gif) ;
      background-repeat : repeat-y ;
      background-position : left ;
      background-attachment : fixed}

div.maintext {margin : 1em 2em 1em 6em}

h1 {text-align : center ; font-size : 250% ; letter-spacing : 0.2em ;
    padding : 0.3em ; color : white ; background-color : lime ;}

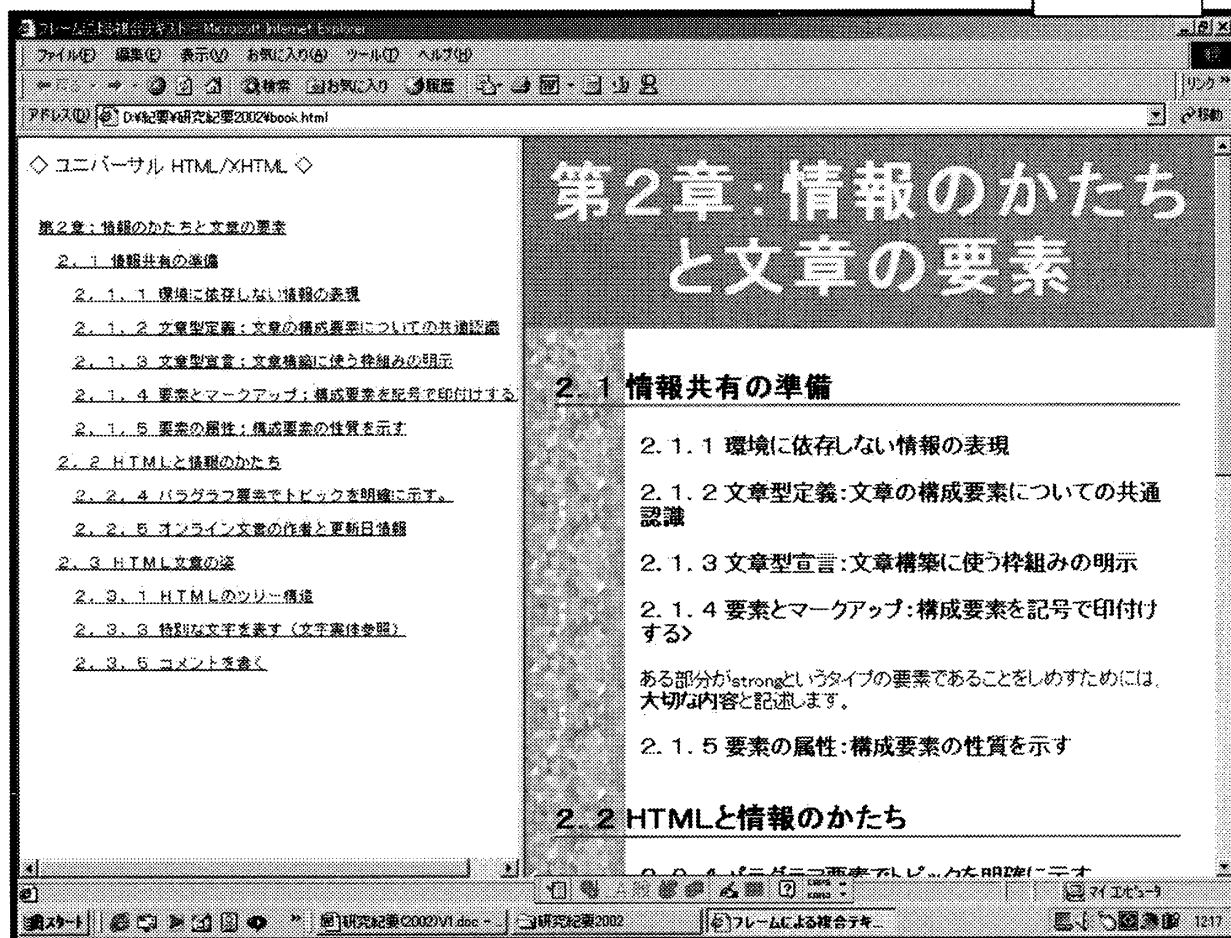
h2 {margin : 1.5em 0 1em -3em ;
    border-bottom : gray thin solid ;}

div.box-column {
  border : thin gray ridge ; margin : 0.5em 0 0.5em 0.5em ;
  padding : 0.5em ; float : right ; width : 40% ; background : #eee ;}

```

図 5.

図 6.



左フレームに見出しタグ文字列（章・節のタイトルに相当する文字列）が目次として表示され、そこをクリックすることにより、右フレーム表示が目次に対応するページに切り替わるようにする。

学生にはフレーム分割されたホームページ学習のため、まず1つのホームページ（第2章のホームページ）の目次のみが表示されるようなフレーム分割ホームページ（図6.）を作成してもらった。

まず第2章のホームページ“Lst2.html”には、見出しタグの箇所にアンカーを追加して、目次ページから見出し箇所に直接リンクが張れるように準備してもらった。

（図7. アンダーラインの引かれている箇所を追加）

そして、以下の2つのホームページを作成してもらった。

①フレーム分割指示を記述するメインホームページを、“book.html”という名前で作成。

（図8. 参照）

②左フレームに表示される、目次ホームページを、“mokuji.html”という名前で作成。

（図9. 参照）

以上により、“book.html”ページを起動することにより、（図6.）の目次フレーム付きページが表示されることになる。

3章以降の複数のページについても、適切なアンカーを追加し、追加したアンカーと見出し文字列を目次フレームページにコピーして目次



```

<HTML>
<HEAD>
<TITLE>第2章：情報のかたちと文章の要素</TITLE>
  <link rel="stylesheet" href="../syo6.css" type="text/css">
  <meta http-equiv="Content-Style-Type" content="text/css">
</HEAD>
<h1><a id="caption2" name="caption2">第2章：情報のかたちと文章の要素</a></h1>
<div class="maintext">
  <h2><a id="caption2-1" name="caption2-1">2. 1 情報共有の準備</a></h2>
  <h3><a id="caption2-1-1" name="caption2-1-1">
    2. 1. 1 環境に依存しない情報の表現</a></h3>
  <h3><a id="caption2-1-2" name="caption2-1-2">
    2. 1. 2 文章型定義：文章の構成要素についての共通認識</a></h3>
  <h3><a id="caption2-1-3" name="caption2-1-3">
    2. 1. 3 文章型宣言：文章構築に使う枠組みの明示</a></h3>

```

図7.

```

<HTML>
<HEAD>
<TITLE>フレームによる複合テキスト</TITLE>
</HEAD>
<FRAMESET COLS="30%, 70%">
  <FRAME NAME="mokuji" TARGET="child" SRC="mokuji.html">
  <FRAME NAME="child" SRC="syo2/Lst2.html">
</FRAMESET>
</HTML>

```

図8.

を作成する作業をすればよい。

各章のホームページ見出し箇所にアンカーを追加する作業は、手作業で実行してもらうことにしたが、目次フレームを作成する作業は、単純ではあるが面倒なので、教室のサーバにて動作している ASP (Active Server Pages) を利用し、自動的に目次フレーム付きホームページを作成する機能を持つホームページを用意し、学生に使ってもらうことにした。

#### 4.2 目次フレーム付きホームページ自動作成

指定した複数のホームページから、見出しタグ文字列とアンカーを取り出し、目次フレームを作成し、左フレームに目次、右フレームに本文を表示する本 (ブック) の体裁をもつホームページを自動的に構築するページを作成してみた。教室サーバ内に構築してある ASP (Active

Server Pages) プロセス動作環境 (1999年度から運用) にて動作するホームページである。

まず図10. のホームページ (“makebook.html”) を作成した。

画面下の、**BOOK 作成実行** ボタンをクリックすると、このホームページと同ディレクトリに存在する VBScript のみからなる “makeContents.asp” ホームページが動作する。(図14. にスクリプトを記載した。)

この “makeContents.asp” ホームページは、サーバ内でスクリプトが実行され、目次フレーム付きホームページを自動的に作成し、表示して終了する。

1つの本 (ブック) の体裁でまとめる対象となるホームページの名称は、あらかじめ “makebook.html” ホームページと同ディレクトリに、“component.txt” という名称のテキストファイルを用意して記述することにした。“makeContents.asp” ホームページのスクリプトは、同ディレクトリに存在する “component.txt” テキストファイルを読み込み、そこに記述されている名前のホームページを開く。

そしてホームページテキストを1行ずつ読み込み、<h1> ~<h6> までの見出しタグを検

図 9.

```

<HTML>
<HEAD>
<TITLE>目次</TITLE>
</HEAD>
<BODY>
◇ ユニバーサル HTML/XHTML ◇
<BASE target="child">
<PRE>
<FONT SIZE=2>
  <A HREF="syo2/Lst2.html#caption2">第 2 章：情報のかたちと文章の要素</A><BR>
  <A HREF="syo2/Lst2.html#caption2-1">2. 1 情報共有の準備</a><BR>
    <A HREF="syo2/Lst2.html#caption2-1-1">2. 1. 1 環境に依存しない情報の表現</a><BR>
    <A HREF="syo2/Lst2.html#caption2-1-2">2. 1. 2 文章型定義：文章の構成要素についての共通認識
</a><BR>
    <A HREF="syo2/Lst2.html#caption2-1-3">2. 1. 3 文章型宣言：文章構築に使う枠組みの明示</a><BR>
    <A HREF="syo2/Lst2.html#caption2-1-4">2. 1. 4 要素とマークアップ：構成要素を記号で印付けする
</a><BR>
    <A HREF="syo2/Lst2.html#caption2-1-5">2. 1. 5 要素の属性：構成要素の性質を示す</a><BR>
  <A HREF="syo2/Lst2.html#caption2-2">2. 2 HTMLと情報のかたち</a><BR>
  <A HREF="syo2/Lst2.html#caption2-2-4">2. 2. 4 パラグラフ要素でトピックを明確に示す。</a><BR>
  <A HREF="syo2/Lst2.html#caption2-2-5">2. 2. 5 オンライン文書の作者と更新日情報</a><BR>
  <A HREF="syo2/Lst2.html#caption2-3">2. 3 HTML文章の姿</a><BR>
  <A HREF="syo2/Lst2.html#caption2-3-1">2. 3. 1 HTMLのツリー構造</a><BR>
  <A HREF="syo2/Lst2.html#caption2-3-3">2. 3. 3 特別な文字を表す（文字実体参照）</a><BR>
  <A HREF="syo2/Lst2.html#caption2-3-5">2. 3. 5 コメントを書く</a><BR>
</FONT>
</PRE>
</BODY>
</HTML>

```

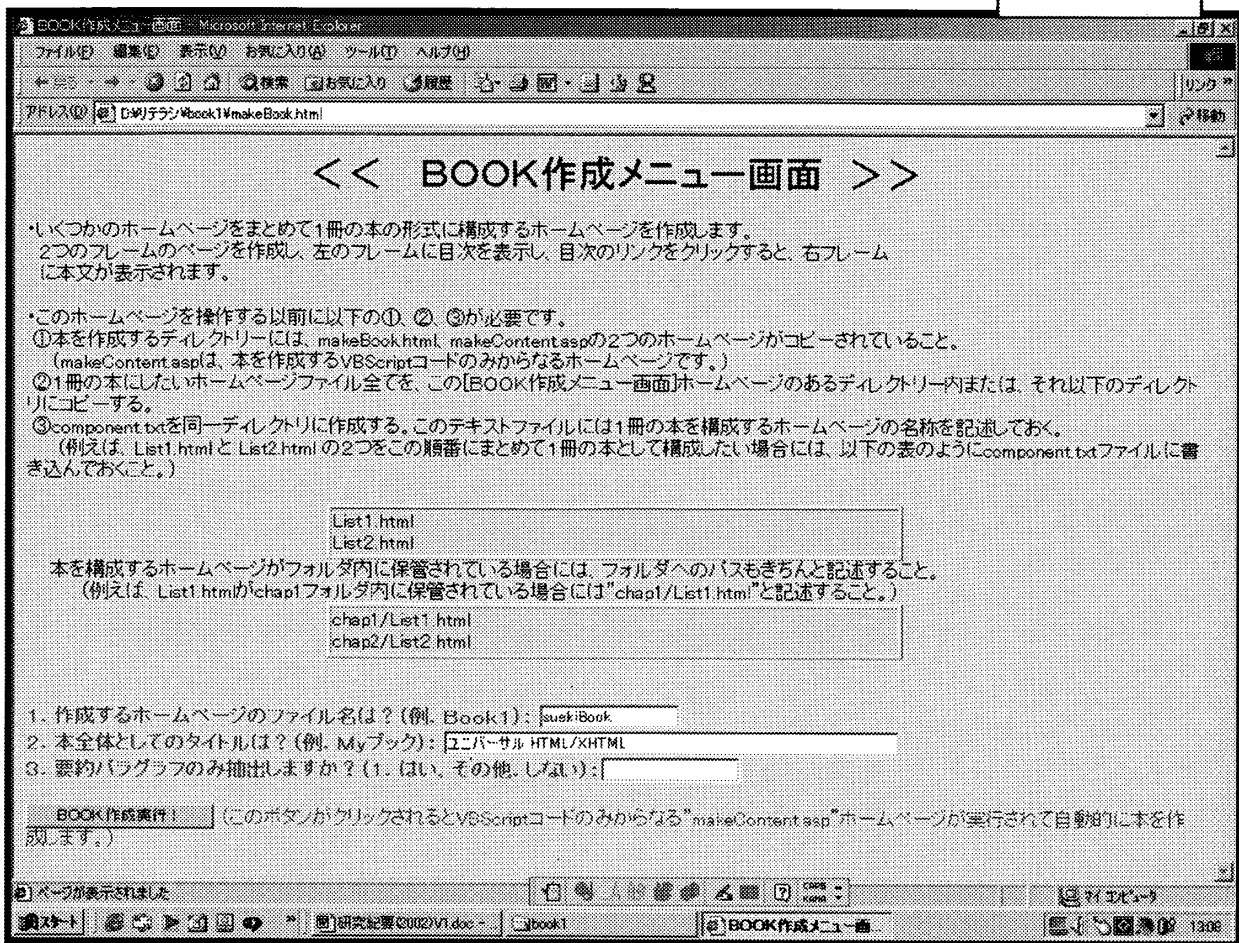
索し、見出し文字列と、アンカータグ <a> の id 属性値を取り出し目次フレームページに書き出す。(図14.)における“SearchLinkText”という名称のサブルーチンがその文字列検索を行っている箇所である。VBScript 言語に用意されている InStr 関数を使用して文字列を検索する単純なコーディングで作成した。DOM (Document Object Model) に準拠したパーサの関数などを使ってホームページ内要素を検索するような方法を取ればよりスマートだろうが、現在の教室の環境では使用できないので、泥臭い単純な文字列検索になっている。

学生は、まず“メモ帳”等のテキストエディタソフトにて、各自のディレクトリ内に、“component.txt”という名称でテキストファイルを作成する必要がある。(図11.)のように第2章から第6章のホームページの名称を記述する。

そして各自のディレクトリに、2つのホームページ (“makebook.html”、“makeContents.asp”)をコピーする。そして、教室の IIS (Internet Information Server) サーバを經由して、“makebook.html” ページを開く (図10. 参照)。後は、ホームページの名前を入力するテキストボックスと、本 (ブック) のタイトル文を入力するテキストボックスに適切な入力を行ない、**BOOK 作成実行** ボタンをクリックするだけである。

例えば、ホームページの名前として、“htmlbook”、本全体としてのタイトルを“◇ユニバーサル HTML/XHTML ◇”と入力して、**BOOK 作成実行** ボタンをクリックするならば、(図12.) の様なホームページが自動作成される。フレームを定義するメインページは、“htmlbook.html” という名称で作成され、目

図10.



syo2/Lst2.html  
syo3/Lst3.html  
syo4/Lst4.html  
syo5/Lst5.html  
syo6/Lst6.html

図11.

次を表示する左フレームのページは、“content-shtmlbook.html”という名称で作成される。また左フレームのページの頭には、本のタイトルとして入力された“◇ ユニバーサル HTML/XHTML ◇”文字列が表示されることになる。以上の演習結果により学生各自のフォルダ内ファイル構成は、(図13.) のようになる。

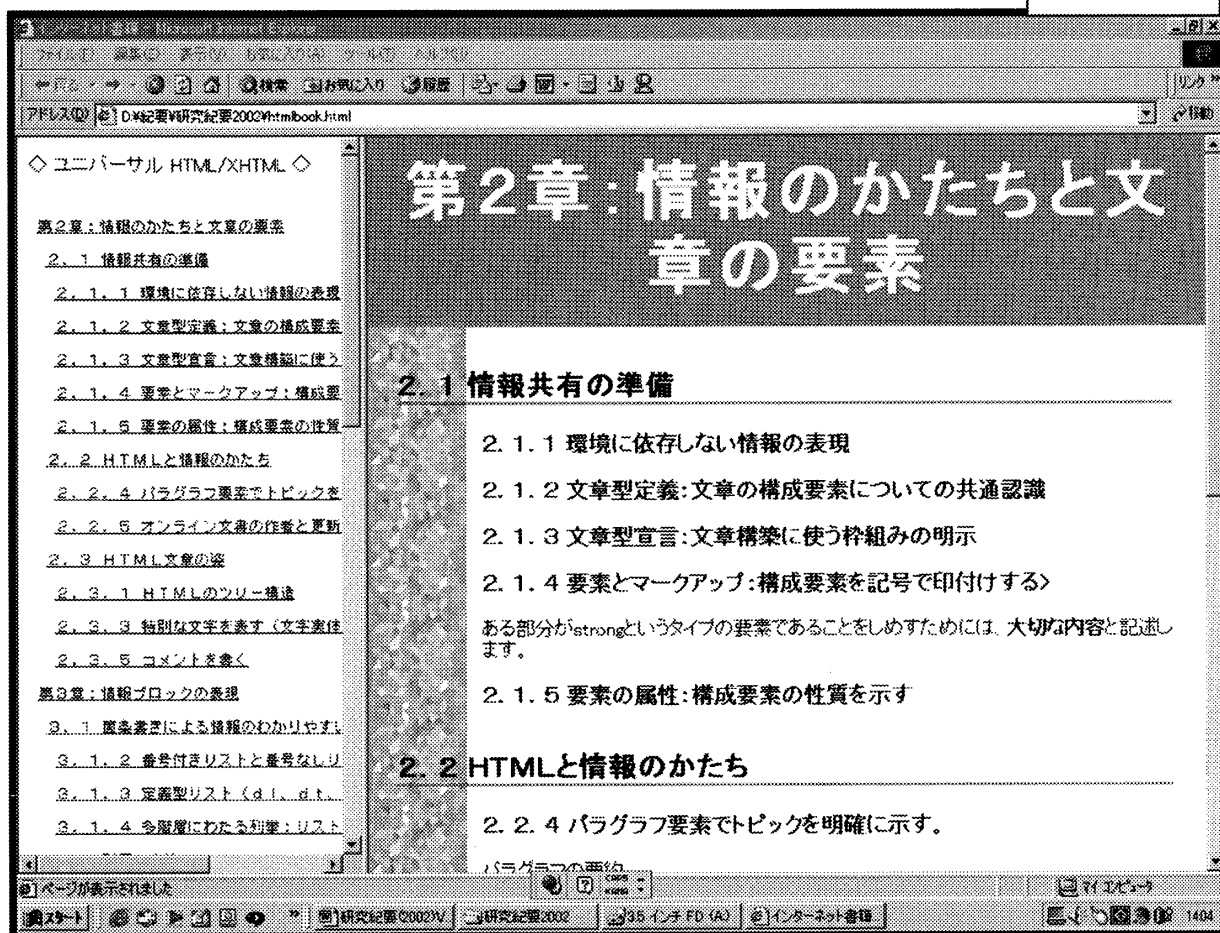
**5. 結び**

複数のホームページに分けて演習した

HTML 言語の演習結果は、目次フレーム付きホームページの形態にまとめられた。さらにこれらのホームページとは別に自由形式で学生各自の自己紹介するようなページを作成し、そこから目次フレーム付き本(ブック)形式ページにリンクしてもらった。これでホームページの演習を完結させた。

“メモ帳”などのテキストエディタソフトを使ってHTML・XHTML 言語テキストでホームページを記述する場合、特にスタイルシートを使おうとすると、きちんと(タグの入れ子関係を正しく、タグの開始・終了の対応を正確に記述するなど)ホームページテキスト、スタイルシートテキストを記述しないと正常に表示されない。スタイルシートテキストに全角文字(全角のスペースも不可) がちょっと混在するだけ

図 1 2 .



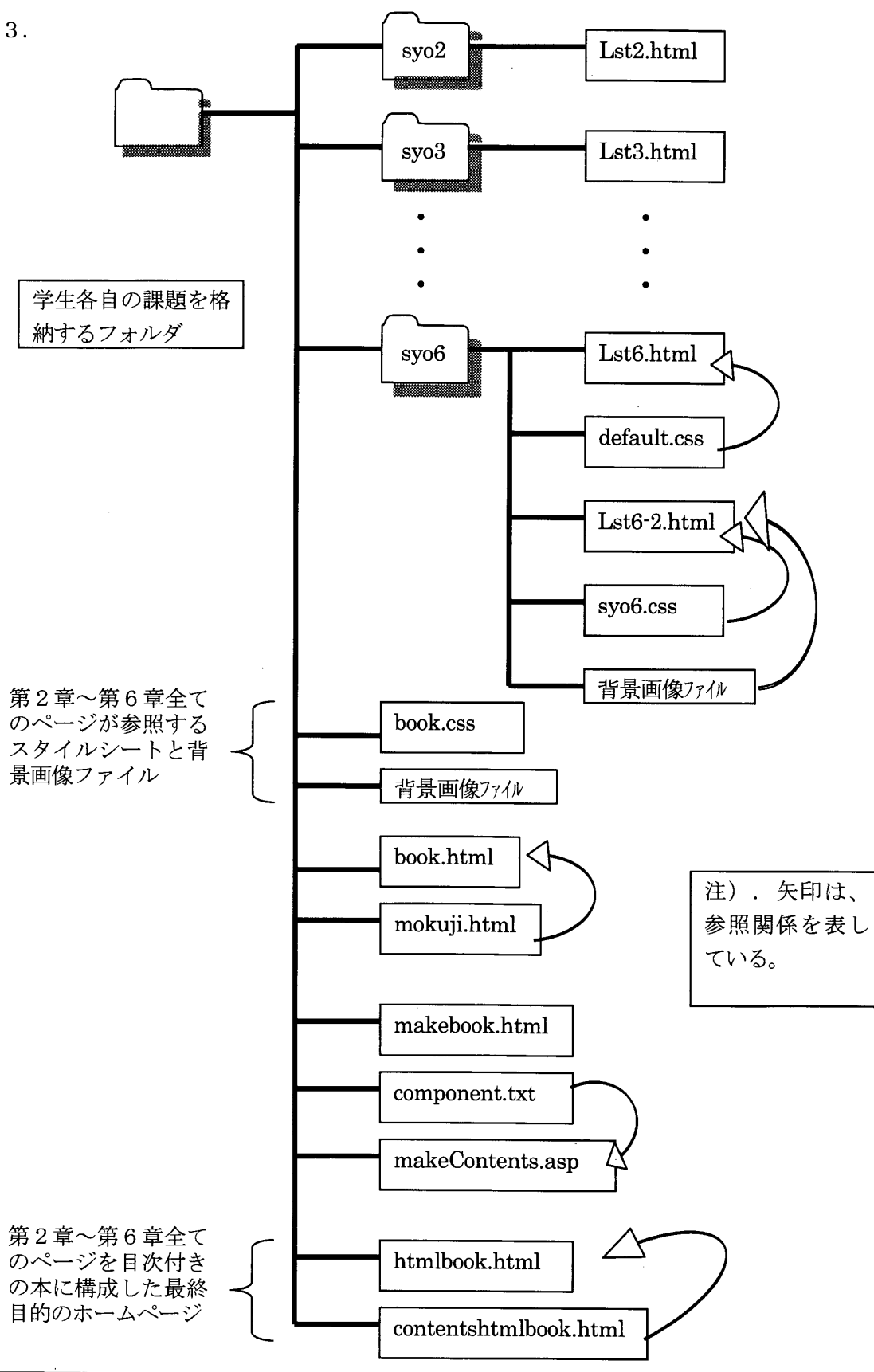
でも正常に表示されなくなるなど大変である。異常表示となる原因を特定するのもなかなか困難である。テキストエディタできちんとしたホームページを作成するのは一種のプログラミングと同じようなものである。学生にとって最初は難しく感じると思われる。

ホームページテキストは、文書構造をきちんと記述することに専念し、表示スタイルは別のスタイルシートテキストを用意することにより、同じ文書構造を持った複数のホームページに統一された表示スタイルを付与することが簡単にできるという点を学生に理解してもらうということを意図した。演習を通して複数のホームページが同一表示スタイルで統一され、1つの本(ブック)形式でまとめ上げられる工程を一通り実習してもらったことになる。

1通りスタイルシートの基本的なプロパティを使ってもらい、1つの表示スタイル(本のレイアウト)を体験してもらう演習であったが、時間的余裕があれば、もっと多様なスタイルシートを用意して、1つのホームページがスタイルシート次第でいろいろな表示スタイルに変化する様子を体験してもらいたい。さらにはスタイルシートを使いこなして学生各自独自の表示スタイルのホームページを作成するような演習に進行させ、応用力をつけてもらうような演習に進めるのも1つの方向である。ホームページデザイン演習とでも呼ぶ方向であろうか。

またもう1つの方向として、ホームページの文章構造、データ構造に関係する事項に力を入れる演習の方向もある。XHTML、XML文書、ホームページ内要素を Javascript などのスク

図13.



```

<%@ LANGUAGE="VBScript" %>

<%
  dim tagName(6)
  dim tsoContentsHp
  dim firstHp
  dim noElement
  dim bookName
  dim bookTitle
  dim bookSummary
  dim contentsName

  noElement= 5
  tagName(0)="h1"
  tagName(1)="h2"
  tagName(2)="h3"
  tagName(3)="h4"
  tagName(4)="h5"
  tagName(5)="h6"

  Response.Buffer=True
  FirstHp=""

  bookName= Request.QueryString("bookName")
  bookTitle= Request.QueryString("bookTitle")
  bookSummary= Request.QueryString("bookSummary")

  if bookName= "" Then
    Response.Redirect "makeBook.html"
  Else
  End if

  contentsName= "contents" & bookName & ".html"
  MainRoutine()
  Response.Redirect bookName & ".html"
%>

<%
Sub mainRoutine()

  Set ObjSync= Server.CreateObject("UniSoft.UniSync")
  ObjSync.OpenMutex "1", True
  ObjSync.LockMutex

  Set fso = Server.CreateObject("Scripting.FileSystemObject")
  fp1=Server.MapPath("component.txt")
  ERR.Clear
  if fso.FileExists(fp1)=0 Then
    set tso=fso.CreateTextFile(fp1, True, False)
  End if
  Set tso=fso.OpenTextFile(fp1, 1)

  If Err.Number <> 0 Then
    htmlName=""

```

```

Else
  createContentsHp()
  Do Until tso.AtEndOfStream
    htmlName=tso.ReadLine
    if firstHp="" Then
      firstHp= htmlName
    End If
    searchHtmlFile(htmlName)
  Loop
  tso.Close
  closeContentsHp
End If
createFrameHp

ObjSync.UnlockMutex

End Sub

Function SearchHtmlFile(htmlName)
  Set fso = Server.CreateObject("Scripting.FileSystemObject")
  fp=Server.MapPath(htmlName)
  ERR.Clear

  Set tso=fso.OpenTextFile(fp, 1)

  Do Until tso.AtEndOfStream
    txt=tso.ReadLine
    call SearchLinkText(txt, SearchText, SearchId, indent)
    if indent > 0 Then
      WriteText= ""
      For i= 1 to indent
        WriteText= WriteText & " "
      Next
      WriteText= WriteText & "<A HREF="" & htmlName
      if SearchId <> "" Then
        WriteText= WriteText & "#" & SearchId
      End if
      WriteText= WriteText & "">"
      WriteText= WriteText & SearchText & "</A><BR>"
      tsoContentsHp.WriteLine WriteText
    End If
  Loop
  tso.Close
End Function

Function SearchLinkText(txt, SearchString, idString, indent)
  SearchString= ""
  indent=0

  For i=0 to noElement
    '===== search element tag start [<element ]
    string1="" & tagName(i)
    pt1=InStr(1, txt, string1, 1)

```

```

if pt1 > 0 Then
    pt2= InStr(pt1, txt, ">", 1)
    if pt2 > 0 Then
        pt2= pt2 + 1
        '===== search end of element tag [</element>]
        string2= "</" & tagName(i)
        pt4= InStr(pt2, txt, string2, 1)
        if pt4 > 0 Then
            pt5=Instr(pt2, txt, ">", 1)
            if pt5 > 0 then
                pt5=pt5+1
                pt6=Instr(pt5, txt, "<", 1)
                if pt6 > pt5 then
                    SearchString= Mid(txt, pt5, pt6-pt5)
                    indent= i + 1
                End if
            End if
        End if
    Else
    End if

    '===== search [id] property
    ptld= InStr(pt1, txt, "id=", 1)
    if ptld > 0 Then
        ptld2= InStr(ptld + 1, txt, "''''", 1)
        if ptld2 > 0 Then
            ptld3= InStr(ptld2 + 1, txt, "''''", 1)
            if ptld3 > 0 Then
                idString= Mid(txt, ptld2+1, ptld3 - ptld2-1)
            End if
        End if
    End if

    Else
    End if
    i= noElement
End if
Next
End Function

Sub createContentsHp()
    Set fso = Server.CreateObject("Scripting.FileSystemObject")
    fp=Server.MapPath(contentsName)
    ERR.Clear
    if fso.FileExists(fp)=0 Then
        set tsoContentsHp=fso.CreateTextFile(fp, True, False)
    End if

    set tsoContentsHp=fso.CreateTextFile(fp, True, False)

    tsoContentsHp.WriteLine "<HTML>"
    tsoContentsHp.WriteLine "<HEAD>"
    tsoContentsHp.WriteLine "<TITLE>目次</TITLE>"

```



```

tsoContentsHp.WriteLine "</HEAD>"
tsoContentsHp.WriteLine "<BODY>"
tsoContentsHp.WriteLine "◇ " & bookTitle & " ◇"
tsoContentsHp.WriteLine "<BASE target=""child2"">"
tsoContentsHp.WriteLine "<PRE>"
tsoContentsHp.WriteLine "<FONT SIZE=2>"

End sub

Sub closeContentsHp ()
tsoContentsHp.WriteLine "</FONT>"
tsoContentsHp.WriteLine "</PRE>"
tsoContentsHp.WriteLine "</BODY>"
tsoContentsHp.WriteLine "</HTML>"
tsoContentsHp.Close
End sub

Sub createFrameHp ()
Set fso = Server.CreateObject("Scripting.FileSystemObject")
fp=Server.MapPath(bookName & ".html")
ERR.Clear
' if fso.FileExists(fp)=0 Then
'     set tso=fso.CreateTextFile(fp, True, False)
' End if

set tso=fso.CreateTextFile(fp, True, False)

tso.WriteLine "<HTML>"
tso.WriteLine "<HEAD>"
tso.WriteLine "<TITLE>インターネット書籍</TITLE>"
tso.WriteLine "</HEAD>"
tso.WriteLine "<FRAMESET COLS=""20%, 50%"">"
tso.WriteLine "<FRAME NAME=""child1"" TARGET=""child2"" SRC="" & contentsName & """">"
tso.WriteLine "<FRAME NAME=""child2"" SRC="" & firstHp & """">"

tso.WriteLine "</FRAMESET>"
tso.WriteLine "</HTML>"
tso.Close
End sub

```

リプトで操ることなどホームページ内の文章構造・データ構造に関連した各種技術がどんどん進化している。それらすべてを学生の演習で扱おうとしないまでも、それらの技術を体系的に理解しておかねばと思っている。

今回の目次フレーム付きホームページを自動作成するホームページでは、単純な文字列検索関数に頼ったコーディングをしているが、今後

は DOM (Document Object Model) に準拠したパーサなどを使用する、よりスマートなやり方も探してみたい。またクライアント側でも同様なパーサが使える環境が整う状況になるなら Javascript などのスクリプトを追加して、動的に変化する目次フレームなども作成できるのかもしれない。

## [参考文献]

- 1) 生形洋一著『ASP 実践プログラミング』  
1999年 技術評論社
- 2) Alex Fedorov 他著、有限会社トップスタ  
ジオ訳、河端善博 監修 『ASP2.0標準講  
座』1999年 翔泳社
- 3) Alex Homer、David Sussman 著、株式会  
社桐原ユニ 監修 『ASP Windows DNA  
プログラミング』1999年 ソフトバンク
- 4) Keith Brophy / Timothy Koets 著、中屋  
宏之 訳 『VBScript 入門 インタラクテ  
ィブ・ホームページの作成』1997年プレ  
ティスホール出版
- 5) アンク著 『スタイルシート辞典』1999年  
株式会社 翔泳社
- 6) 神崎正英著 『ユニバーサル HTML /  
XHTML』 2000年 株式会社 毎日コミ  
ュニケーションズ
- 7) Frank Boumphrey、Cassandra Greer、  
Dave Raggett、Jenny Raggett、Sebastian  
Chnitzenbaumer、Ted Wugofsk 著、株式  
会社サン・フレア訳 『ビギニング  
XHTML』2001年 株式会社インプレス