

HTML 文書作成の手引き

科学書刊株式会社:電子版の原稿

「橋梁&都市 PROJECT: 2011」 (ISSN 1344 - 7084)

島 田 静 雄

2011年5月

あらまし(summary)

HTMLの書式に従って文書を作成するには、ワードプロセッサを始めとして、幾つかのソフトウェアを使いこなす技能が要求されます。これらのマニュアルには、コンピュータ本体の専門用語に加えて、従来からの印刷・出版関係の特殊な専門用語が出てきます。文書を作成するとなると、これらの専門用語について、意味と機能とを理解しておく必要があります。まずHTMLの用語ですが、これはHyper Text Markup Languageの頭字語(acronym)です。この後半のMLは、日本語で言えば編集校正記号に当たり、原稿の文字並びに補助的に書き込む、組み版を指示する言語です。HTML文書の作成者は、文章原稿と共に、この組み版言語を適切に挿入したテキストファイルを作成し、拡張子を(.htm)または(.html)に直します。コンピュータがHTMLファイルをモニタに表示するとき、この組み版言語を解釈して、決められた書式(レイアウト)で表示してくれます。表示に使うソフトウェアを**閲覧ソフト**と言い、マイクロソフト社の**インターネットエクスプローラ**が主に使われます。一般の人がインターネットを利用するときは、HTMLファイルを受信して、受動的に閲覧します。自分側から能動的に文書情報を発信したいとなると、ファイル作成に注意深い丁寧な作業が要求されます。この作業の手引きを、この報文にまとめました。

この文書は、表紙を含めて全**20**ページあります

目次

あらまし(summary)

0. はじめに(introduction)

- 0.1 文書は読み捨てと保存利用とがあること
- 0.2 小部数印刷の需要が多いこと
- 0.3 閲覧利用と印刷利用との二本立て
- 0.4 初期の目的は HELP ファイルの作成にあった
- 0.5 文書形式を無くすわけにはいかないこと
- 0.6 文書作成の三要素
- 0.7 小部数印刷の専門家がタイピストであったこと
- 0.8 ワープロソフトの大衆化
- 0.9 多くのファイル数を必要とする作業を意識すること

1. 文章原稿の作成

- 1.1 作成作業の道具と技能
- 1.2 実践的な作文技能は個人の責任で覚える
- 1.3 Word の使い方を覚える
- 1.4 図と表の挿入は特別な技術であること
- 1.5 筆者の個人的な作業方法
- 1.6 文章の校正は他人にも見てもらう

2. HTML 文書への簡易変換

- 2.1 Word のファイルから HTML 文書に変換する
- 2.2 変換された HTML ファイルの特徴
- 2.3 リバースエンジニアリングで勉強する

3. HTML ファイルの構造

- 3.1 文書としての論理的構造を理解しておく
- 3.2 複数の HTML ファイルに構成する
- 3.3 一単位の HTML ファイルの物理的構造を理解する

4. HTML タグの意味と機能

- 4.1 編集作業の書き込みとの関連で理解する
- 4.2 書式全体が準拠する仕様はファイルの最初に置かれる
- 4.3 標準化された文字フォントを使うのがよい
- 4.4 寸法も相対表示で指定するのがよい
- 4.5 タグは文字単位の書式の表し方から覚える
- 4.6 行単位のスタイル指定を理解する

5. 文字並びの特別な表示方法

- 5.1 スペースを含む書式で表示する方法
- 5.2 全体を字下げで表示したいブロック
- 5.3 箇条書きにしたいとき
- 5.4 表に構成したいとき

6. 文章スタイルの設定と表示

- 6.1 標準として提案する印刷スタイル
- 6.2 モニタ上での表示スタイル
- 6.3 文字並びは何通りかのスタイル指定方法がある
- 6.4 空きの設定
- 6.5 余白の設定
- 6.6 大寸法のグラフィックスを使わない

7. 表の作成と応用

- 7.1 文書の表示は表の作成が応用されている
- 7.2 表の作成にはテンプレートを使う
- 7.3 テンプレートの HTML ファイルを利用する
- 7.4 テンプレート用ファイルの説明

8. 目次と索引の作り方

- 8.1 ページ番号を使わないこと
- 8.2 目次と索引に宛てる印刷ページの設計が必要
- 8.3 エクセルを編集作業で利用すること

あとがき

用語索引

英字		ゴシック体	4.3	同人誌	0.2
ALIGN	6.4	込め物	6.4	道具	1.1
CSS	4.1			な	
DTD	4.2			日本語ワープロ専用機	0.8
DTP	0.1	さ			
GUI	1.4	作業領域	6.5	は	
HTML Help	0.4	システムフォント	4.3	パイカ	4.3
HTML タグ	4.1	ショートカットキー	0.8	パディング	6.4
IE	0.4	字下げ	5.2	パラグラフ	1.2
ML	4.1	自叙伝	0.2	左詰め	6.4
MS ゴシック	4.3	自費出版	0.2	ファンクションキー	0.8
OS	0.4	実用文書	0.1	フォーム	6.5
Windows HELP	0.4	実用文書のまとめ方	1.3	フォント	4.3
Word	1.1	終了タグ	4.1	太字	4.3
WYSIWIG	0.6	初期値	6.3	文章	0.6
		書式	0.6	ヘッダー	1.2
		書式	1.5		
あ		スタイル	0.6	ま	
アスキーファイル	0.8	スタイルシート	4.1	マイコン	0.7
インタネットエクスプローラ	0.4	全	1.3	右詰め	6.4
		全	2.2	見出し	1.2
インデント	5.2	隙間	6.4	明朝体	4.3
インライン	6.3	数式エディタ	1.4	メニューバー	6.5
ウインドウ	6.5	センタリング	5.2	メモ帳	1.1
エリート	4.3	属性	4.1	や	
閲覧ソフト	0.4			余白	4.6
		た		全	6.5
か		タイトルバー	6.5	ら	
カーボンコピー	0.7	段組み	4.6	リスト	5.3
箇条書き	5.3	段落	1.2	リバーエンジニアリング	2.3
画素数	6.6	中央揃え	6.4	ルビ	1.6
解像度	6.2	ツールバー	6.5	レイアウト	0.6
開始タグ	4.1	綴り方	1.2	わ	
活字	4.3	テキスト	0.6	ワードプロセッサ	1.1
キーボードショートカット	0.8	テキストエディタ	0.7		
		テキストファイル	0.8		
起承転結	1.2	デジタルカメラ	6.6		
技能	1.1	デスクトップ	6.5		
技法	1.1	デフォルト	6.3		
行	4.6	テンプレート	1.2		
継承	4.1	体裁	0.6		
組み版言語	4.1	ドットマトリックスプリンタ	0.8		
罫線	1.4		4.3		
懸崖	6.3	等幅	4.3		

0. はじめに(introduction)

0.1 文書は読み捨てと保存利用とがあること

インターネット時代に入って、従来は紙の印刷物として利用してきた文書が、電子化文書としてパソコンの画面でも閲覧できるようになりました。これには、著作権や出版権とからんで、新しい形のビジネスモデルの模索が始まっています。文書の利用者側（ユーザ）から見れば、私的に閲覧する新聞・雑誌、さらに手紙などは、一度読めば済む一過性の情報であることが多いので、特に必要が無ければ手元に紙の形で残す必要がありません。技術レポート・ビジネス文書に代表される実用文書は、紙の形での利用が主です。その文書作成過程にパソコンの利用が標準になりました。平行してモニタ画面上での閲覧との二本立て利用もできるようになりました。商業出版物は、大部数を販売することでビジネスとして成り立ちます。これに対して、実用文書は、ユーザが個人的な DTP(desktop publishing) 作業で、見栄えのよい原稿を作成することが普通になりました。或る程度まとまった部数で、体裁のよい印刷・製本が必要であれば、専門業者に頼まなければなりません。それに拘（こだわ）らなければ、パソコン付属のプリンタを使い、ゼロックスでコピーを取り、簡単なホッチキス綴じで済ますことができます。保存については別の課題がありますが、この報文では触れません。

0.2 小部数印刷の需要が多いこと

部数にして 100 冊前後の小部数の印刷物を作成する需要の多い環境の代表は、小中高を始めとした教育現場です。パソコンやコピー機械が普及していなかった時代、学校の先生方は、仕事の大部分が ガリ版切りと謄写版印刷でした。現代では、原稿作成がワープロになり、少し部数を多く必要とするときは、謄写版をやや高級化した軽印刷機を使います。製本も手が掛かります。自分で作業をするのは大変ですので、予算があれば、街中（まちなか）の印刷屋さん頼みます。作業が面倒で費用のかかる印刷物にしろなくても、パソコンの画面を利用することに代えることも考え付くでしょう。しかし、教育利用には何度も読み返して書き込みのできる、消耗品的にも使う印刷物の形が必要です。商業印刷物としての体裁を持つ教科書・参考書・教材などの発売は、教育の公平化と教師側の負担を減らす目的意識があるのですが、一面で、検定またはそれに近い官僚的な統制に加担して、自由な発想が制限されることとの隣合わせの危険もあります。個人ベースで小部数の印刷物を作って配布したい希望を持つ人は、少なくありません。いわゆる自費出版がそうです。代表的なものが自叙伝の類（たぐい）です。年配の方は、自作の短歌や俳句をまとめることを見ます。自分が書いた小説などを発表したい人も多く居て、その一つの現れが同人誌です。これらがインターネット利用の大衆化と共に、印刷利用を目的としない、閲覧利用を主目的とした多様な形態を持つようになってきました。特に、目立ちたがり屋が自己発信に使う格好のツールとして、大衆利用を受け付けるホームページに HTML 文書を寄稿することが増えています。

0.3 閲覧利用と印刷利用との二本立て

文学作品の小説は、読者が頭から順に読むことが前提です。推理小説などは、話しの展開を楽しみます。後ろの解決部分を先読みすることをしませんし、前のヒント部分を読み返すこともしません。楽しむのため二度以上読む場合であっても、謎解きの筋書きを知った上で、通して読みます。技術系の文書は、そういった利用をしません。一般的な読み物として編集したものを除き、ランダムに項目を探す利用が普通です。したがって、数ページ程度の短い文書は例外としても、目次と索引があるのが必須の条件です。もともと、インターネットの目的は、強力な検索機能を使って、膨大なデータの集合から、或るヒント（キーワード）を手掛かりにして参考事項を引き出すことにありました。つまりデータベースの利用です。最初の中は、表題・著者・発行機関・発行年の情報を得て、図書館で原典に当たりました。この情報をもう少し親切にしたものが、抄録部分も読めるようにする方法です。さらに欲張って、必要があれば全文が読めるようにしたデータベースが利用されるようになりました。学术论文のようなページ数の少ない文書がこの方法に移行して、従来からの学術誌の印刷出版を取り止めることが増えています。そうすると、読者側で、従来のような印刷形式のコピーが得られるようにする対応が必要です。これには、現在のところ PDF 形式のファイルでダウンロードサービスがされています。ここで注目したいことは、文書内容が、2種類以上の電子化データで利用されるようになったことです。一つは、パソコンの狭いモニター画面で閲覧するだけを目的としたもの、もう一つが印刷した方が使い勝手が良いものです。文書作成の手間から言えば、最初に印刷利用を目的とした文書を作成し、それを HTML 文書に落とすことが効率的です。この報文は、その作業手順を解説することを目的としています。

0.4 初期の目的は HELP ファイルの作成にあった

大型コンピュータが使われていた時代から、コンピュータを使いこなすための種々の参考書（レファレンス）類は重要な文書でした。これらを理解していなければ、仕事になりません。これらは、必要な事項がすべて書いてある書類の集合です。日常的な利用には、使い勝手を良くした手引書（マニュアル）を工夫したものを手許に置きました。コンピュータの OS（オペレーティングシステム）が単純な DOS の環境であったとき、コマンドレファレンスが無くても、コマンドを忘れるか、使い方が分からないと、キーボードを前にして、ユーザは何もできませんでした。この不便さを少なくする方法の一つが、プログラムとして動作する HELP の組み込みです。しかし、この使い方が分からないと同じ不便になりますので、或る共通した操作方法が提案されるようになりました。パソコンの OS に Windows が採用されるようになって、当初は Windows HELP が利用されました。西暦 2002 年以降はインターネットとの相性のよい HTML Help が標準になりました。この作成は、かなり専門的な作業です。HTML Help 利用の基本的な考え方は、印刷物（ハードコピー）としての形を持つレファレンスと並行して、モニタ上で閲覧する（ソフトコピー）に位置づけられていることです。レファレンスはページ数の多い文書です。その中の必要な項目を検索して表示する 閲覧ソフト (browser) に、マイクロソフト社の インターネットエクスプローラ（以降は IE で表記します）が主に使われます。関連のある HTML ファイルを一つのファイルにまとめた（コンパイルした）ものが、通称で言う HTML Help ファイルであって、拡張子は (.chm) です。Windows 系のパソコンで使います。このアイコンをクリックすると、IE が HTML Help の項目を表示します。自分のパソコンに (* .chm) ファイルが無くても、外部の WEB サイトにある、元の HTML ファイルにオンラインリンクして使う方法もできるようになりました。これらの方法が大衆化してきたものが、個人単位で情報発信をするホームページの開設です。こちらの方のファイル作成は、モニタの画面上でのデザインの方に凝り、一つ、または多くても数個のファイルを扱う程度であることが、HELP ファイルと異なります。

0.5 文書形式を無くすわけにはいかないこと

印刷した文書を作成するとき、実際に利用する量よりも多くの部数を作成するのが普通です。その分費用が嵩み、資源の無駄使いにも繋がります。印刷した時点では利用の頻度が多いのですが、しばらくするとその頻度が減り、廃棄が進み、探しても見つからなくなります。著作物を作成して保存しておくには相応の費用が掛かります。この費用を補う方法のビジネスモデルが、従来は著作権を盾にして、印刷・出版関係で機能していました。しかし、コンピュータ関係の出版物では、書店で発売されたときに購入しておかないと、後から見たいときの手掛かりが無くなります。参考文献として書いてあっても、一般図書館は、この種の印刷物を保存の対象とはしません。これを解決する方法が電子化図書です。必要なときにダウンロードして、必要な部数だけの印刷物を再現できるようにします。2010 年末から話題になった電子化図書の利用は、既に紙の形で出版されている書籍の閲覧だけを考えていて、紙に印刷する再現を問題にしています。小部数の実用文書の場合には、電子化文書を先に作成しておいて、必要な時にハードコピーに印刷する方法に移行しています。したがって、保守的ですが、印刷仕上がりの文書形式に作成する技術を棄てておくことが必要です。

0.6 文書作成の三要素

実用文書を作るときには三つの要素を考えます。文章・書式・体裁です。文書に作りたい目的は、それぞれです。いずれにしても、はじめに、或る出来上りの構想があります。それは、書式（または割り付け；1 ページ単位でのスタイル：*注参照）と体裁（紙質・寸法・綴じなど）についての漠然としたイメージです。普通は、何か見本があるか、大まかな執筆要綱が別にあります。より具体的には、印刷出版関係の専門知識を理解しなければなりません。例えば、A4 版・10.5 ポ・横書き・2 段組み・1 ページ 25 文字 50 行 2 段組み、などが意味する内容の理解です。一般の人が形式を整えた文書にしたいときは、コンピュータが利用できなければ、手書きの原稿を文書編集の専門家に依頼します。文章の量は、400 字詰めの原稿用紙で何枚、のように数えます。手書きに代えて、パソコンで文章原稿を作成するようになって、完成した書式（文書スタイル）をモニタで確認することもできるようになりました。これを WYSIWIG (What You see is What You Get) と言います。日本語で言えば、「モニタで見た原寸で印刷が得られる」ことです。これは執筆者にとって便利であると思いがちです。実は、かなり専門的な印刷出版関係の知識が必要になるため、一般ユーザは、パソコンを前にして悪戦苦闘を強いられることにもなります。そこで、最低限、原稿文書だけ（テキスト）の入力作業だけで済まし、残りの作業をワープロに詳しい人に頼むという方法も利用されます。面倒な原稿準備作業の例は、図・写真・数式などの製作と貼り込みです。こちらの方も、従来は、特殊な技能集団が印刷関係者と共に協力していました。

(* レイアウトが書式に、スタイルが体裁に当たりますが、HTML の用語ではスタイルを使います)

0.7 小部数印刷の専門家がタイピストであったこと

一般企業は、その企業の顔を持った、カタログなどのビジネス文書の作成が必須の業務です。これが典型的な小部数文書です。最も利用の多い文書は手紙の類です。通常は相手に渡す本体と控えのコピーを取ります。この作業に当たる専門家が、欧米ではタイピストとしての技能を持つ、多くは女性秘書です。機械的なタイプライタを使うとき、コピーはカーボン紙を挟みます。これをカーボンコピー(carbon copy)と言い、電子メールのソフトではCCとして表されている用語の原義です。機械式のタイプライタを使っていた時代は、ミスタイプをしないよう、神経を集中させることに大きな技能努力が必要でした。そのため、タイピング技能については自己流ではなく、正しい使い方の教育課程を踏まえます。手紙のような一枚もののタイピングでは、ミスタイピングによる打ち直しの手間をあまり気にしません。しかし、ページ数の多い文書をタイピングで作成するときには、校正による作業の手戻りは、時間的にも大きな損失です。この不便さを救済する革命が、電動タイプライタをマイクロコンピュータ(マイコン)で制御するテキストエディタの利用でした。

0.8 ワープロソフトの大衆化

テキストエディタの主目的は、コンピュータプログラミングのソースコードやマニュアルを作成することに用いられました。こちらの場合には、コンピュータが理解できる記憶媒体(テキストファイルなど)に文字データを記録することが利用の主目的でした。紙に印刷することの方には重点を置きませんでした。テキストファイルは、文字コードと、単純な制御コード(空白SP・改行LF・復帰CR・タブTAB)だけで構成されます。英字だけを使う場合は、アスキーファイルと言います。テキストファイルと言うときは、日本語の2バイトコードを使うことを含みますので、英語専用のパソコンでは読めないことに注意が必要です。一方、1980年頃までは、日本語の印刷に使う邦文タイプライタは、一般企業では必須の事務機械でした。英文タイプライタ並みの電動化が難しかったのですが、ドットマトリックスプリンタ・マイクロコンピュータ(マイコン)・CRTキャラクタディスプレイ三つを組み合わせた、日本語ワープロ専用機が1980年代に爆発的に普及しました。ビジネス文書として使う場合、タイピング作業が少し複雑です。それは、編集用の機能をコンピュータに知らせる方法です。その方法は、キーボードに特別な入力キー(ファンクションキー)を付けること、もう一つは、例えばCTRLキーを押しながら文字キーを押す方法です。後の操作をキーボードショートカット、そのときのキー組み合わせをショートカットキーと言います。ワープロソフトは、ショートカットキーがバイナリー形式のコードでテキスト文書に書き込まれていて、そのコード仕様がワープロソフトごとに異なります。そのため、或るワープロソフトで作成したデータを別のワープロソフトでは読めない、つまりデータの互換性がありません。データ読み変えの機能は、それぞれのワープロソフトごとに充実されてきて、ユーザが困ることは少なくなりました。それでも多くの問題が残ります。これを一般的に解決する方法として、組み版言語(markup language: ML)が提案されるようになりました。この一種がHTMLタグ(tag:荷札の意味)です。HTMLファイルは、テキストエディタで開いて読めますし、編集もできます。

0.9 多くのファイル数を必要とする作業を意識すること

著者は永らく技術系の教育環境に居ましたので、私的に作成する文書の大部分は、教材に使うことを目的とした、A4版横書きにして、多いときは100~200ページの分量です。これをWordで編集し、PDF版にまとめたものをダウンロードしてお使い頂くようにしてあります。同時に、WEBでページ単位の閲覧ができるようにしています。こちらは、小単位の、少なくとも同じページ数分のHTMLファイルの集合です。これらのWEBサイトは、差し当たり下記に置いてあります。

中日本建設コンサルタント株式会社：<http://www.nakanihon.co.jp/>

HTMLのファイル数が多いので、その中身を検索して必要個所が閲覧できるように、目次と索引さらにページ前後のリンクを設定して作成してあります。この全体構成にエラーが無いように検査するツールとして、Microsoft HTML Help Workshopを使っています。このバージョンは1996/1998以降、変更がありません。したがって、HTMLファイルの方の作成仕様をHTML4.0にしておく保守的な態度を持たせて作業をしています。この仕様の範囲で問題は起きません。多くのファイル数を扱いますので、個別のファイル寸法を小さくするように、またあまり凝った機能を使わないようにしています。この報文は、著者の作業の手の内を解説したものです。

1. 文章原稿の作成

1.1 作成作業の道具と技能

身体（からだ）、特に手を使う作業全体を技術と言います。技術には三つの要素があります。**道具・技法・技能**です。スポーツは、身体全体も道具として技術を磨きます。文書を作成する道具は文房具（stationery）と総称しますが、これに**パソコン装置一式**がハードウェア（道具）として必需品になりました。そして、パソコンを使いこなす技法と技能とが要求されます。技法は、ソフトウェア、平たく言えばプログラムと、その使い方です。作業は二段階で行われます。第一段階が、原稿作成です。**テキストエディタ**と**ワードプロセッサ**（ワープロ）を主に使います。Windows 系のパソコンでは、前者に**メモ帳**（NotePad）、後者に **Word** を使います。第二段階は、原稿ファイルから HTML ファイルに変換する作業です。このとき、ホームページ作成用に、市販のソフトウェア（例えば IBM 社のホームページビルダー）もありますが、無くても、メモ帳と Word で作業ができます。なお、ソフトウェアと言うときは、プログラムだけでなく、**マニュアル**などの参考書も、意義に含めます。HTML と **CSS** の**用語辞典**は必需品です。これも書物の形だけでなく、電子化文書の利用ができます。作成した HTML ファイル構成に文法エラーが無いように注意しますが、この全体検査に筆者は **HTML Help Workshop** を使っています。これは Microsoft 社からダウンロードできます（2011 年のところ英語版しかありません）。三つ目の要素が技能ですが、技能の大部分は、キーボードのタイピング作業です。技能は上手下手がありますので、自分の責任で覚えます。文字を扱う作業は、マウスを使う頻度よりもキー操作が多くなります。二つ上の項で説明した**キーボードショートカット**をマスターしておくことを勧めます。

1.2 実践的な作文技能は個人の責任で覚える

作文と言うと硬い用語です。昔の小学校では**綴り方**と使いました。良い見本を見て、練習を重ねて自分で覚えます。書店に行けば、「手紙の書き方」のような実用書があります。作文の良し悪しは、他人に見てもらって改善するのがよいのですが、適切な助言のできる人が身近に居るとは限りません。優しい、自然な言葉遣いで文章を書きます。若者同士や身内で使う言い方を避け、言わば、**お行儀のよい**文章にします。学校の先生は、生徒との会話に迎合するような、くだけた話し言葉を使うことも目に付きます。外部の人が読むには、みっともない文体になっていることがあります。形式的なまとめ方をする一般的な規則として、漢詩をまとめるときの作法からきた起承転結で、文章を分けることの説明を見ます。手紙の書き方では、拝啓に始まり、時候の挨拶、本文、そして敬具で締めるスタイルが一つの定番です。この前後に、宛名・日付・表題または用件、発信者名などが付きます。ワープロのソフトでは、この構想に基づいたテンプレート（雛型）が利用できるようになっています。しかし、一般的な文章作成では、起承転結単位は、**段落**（パラグラフ：paragraph）単位に分ける規則と考えるとよいでしょう。文章単位を段落単位で構成し、必要に応じて**見出し**（ヘッダー：header）を付けます。この考え方は欧米の作文技法としては一つの常識です。実は、HTML 文書のタグに<P>、<H1>～<H6>があります。これがパラグラフとヘッダーを意味した英字の略です。

1.3 Word の使い方を覚える

原稿文書の作成は、印刷して利用する形式を見ながら文字入力をする **Word** を使います。実を言うと、Word は、印刷形式を整える編集用ソフトウェア、つまりワードプロセッサです。Word の作業画面のメニューバーには、編集と印刷目的のための専門用語が並びます。機能が重複するのですが、アイコンを表示するツールバーも賑やかに並びます。年配の方は、専門用語の意味と機能とが分かり難いことと、操作法が面倒であることがあって、パソコンの利用を覚え損なった人も少なくありません。若い人は覚えるのが早いので、外見には熟練者に見えることもありますが、基本的な知識を埋めておくことが大切です。この参考には、下記の URL を見て下さい。

実用文書のまとめ方 http://www.nakanihon.co.jp/gi_jyutsu/Shimada/bunshyo/top.html

原稿の文章だけを扱うツールがテキストエディタである**メモ帳**です。こちらの方は、作業画面がずっと簡単です。メモ帳を介して実質的な文章データを扱った上で、そのデータを Word に取り込んで編集します。筆者は、これを**文書のお化粧**と説明しています。PDF ファイルは、Word 上のページレイアウトをそのまま再現します。Word ファイルから PDF ファイルへの変換は、ツールバーの**ファイル→名前をつけて保存**で作業が済みます。ただし、古い Word バージョンでは変換機能の追加が必要です。HTML ファイルは拡張子が(.htm)または(.html)です。**スタイルシート**の拡張子は(.CSS)ですが、どちらも実質がテキストファイルですので、メモ帳で開いて読め、編集に使い、元のファイルに上書き保存できます。

1.4 図と表の挿入は特別な技術であること

文書は、文章だけを扱うのが基本ですが、実用文書では図と表も必要です。これは、鉛の活字を使う編集・印刷の時代からも引き継いでいる特殊な技能が必要です。まず、図は、線図のようなイラストと写真の区別がありますが、どちらも Word 上は同じ扱いです。技術文書では数式があり、これも実質的に図の扱いをします。筆者の場合には、Word から間接的に使うマイクロソフトの数式エディタを使って作成しています。図は、作業データとしては、文字並びとは別のオブジェクト（物）の挿入扱いをします。Word の作業画面に図を挿入するときは、GUI (graphical user interface) の環境で拡大・縮小が自由に行えますので、全体のレイアウトに合わせるように寸法を調整することができます。しかし、図は左寄せ、または中央にしか置けなくて、右寄せにして左側に文字を並べるレイアウトができません。これを解決する一つの便宜的な方法が 2 段組みです。図の幅に制限を受けますが、ページの右側に図を置くことができます。もうひとつの解決法は、1 行 2 列の表を使って、左枠に文字並び、右の枠に図を入れます。表の枠線（罫線；けいせん）を見えないように指定すると、図を右寄せにレイアウトできます。なお、テキストエディタは、図の挿入ができません。これがワープロとの相違です。

1.5 筆者の個人的な作業方法

私的なことですが、ユーザが見ている PDF 版のレポートは、マイクロソフト社の Word で原稿を作成した、そのままのレイアウトで変換したものです。モニタ上での Word の作業画面を印刷レイアウトにしておいて、大体 600~800 字で一段落になるように文章単位をまとめます。一ページに 3~4 段落が入ります。普通の読み物に較べると、筆者の段落文字数は多い方です。段落単位の文字並びを本文と言い、行末で自動改行され、段落の区切りは改行 (Enter または Return) をキーインします。原稿は、思いついた構想で、ランダムに、文章単位で入力します。文字並びは、コピー・切り取り・貼りつけ・挿入・削除などで、適当な文字数で段落単位にまとめます。筆者の原稿は、内容の簡単な見出しを頭に付けますが、これは、筆者の個人的な原稿作文技法です。この見出しは、この段落の要旨です。段落単位は、作文の論理的な流れを判断して前後順を変更することもあります。文書は、通常、章・節・項の三段階に分けて、それぞれに見出しを付けます。段落単位に見出しを付け、これをそのまま項の見出しとするのが分かり易いのですが、複数の段落を集めて、これに項の見出しを付けることもしています。複数の項を集め、その全体に、節の見出しを付け、さらに節をまとめて章の見出しを付けます。この文字並びの約束が書式です。段落単位の文字並びが一ページに収まらず、途中から次のページに続くのは見えてくれが悪くなりますので (widow, orphan の用語があります)、ページ全体の文字数を見て、文章の追加や削除を加えています。また、節の始まりが、なるべくページの先頭になるように、項の集合を調整することもしています。見開きの書物の体裁にするときは、章の始めを奇数ページにするのが決まりです。そうすると、前の章が奇数ページで終わると空白のページを挿入することになります。小部数の実用文書では、両面印刷をしないことも多いので、この決まりは採用していません。

1.6 文章の校正は他人にも見ってもらう

作文した文書は、作者の責任で注意深く検査するのが原則です。原稿は、時間をおいて読み返すと修正したい箇所が見つかります。しかし、ワープロ作業での文字入力ミスや、仮名漢字変換で起こす誤字・当て字は、第三者に見つけられるまで、当の作者では発見できないことも多いものです。筆者の文章は「です・ます」にしていますが、これには幾つかの理由があります。第一は、英文との対応を考えています。「…です」は be 動詞に当たり、「…ます」は一般動詞の文末に使います。第二は、読み上げて使うときの不自然さを避けるためです。これは、必ずしも眼の不自由な人を対象とするものではありません。このとき、「である」調は、黙読では不自然ではありませんが、声に出すと、固い言い方になります。「我が輩は猫である」は、be 動詞を使う「I am a cat」に当たります。である調の場合には、一般動詞の終わりを終止形で使うことにはなりますが、読み上げるとぶっきらぼうに聞こえます。ファイル化した文書を音声で伝える方法も研究されるようになりました。英語の場合は、かなり古くから実用されていました。日本語文章をコンピュータが読んで、間違った読み方にならないような文章の書き方に工夫が必要になってきたことです。英語が生で入るのも困りますし、カタカナ語は元の言語を知らなければ理解できません。漢字熟語は同音意義で誤解されることも多いことと、音と訓の読み分けが必要です。そのため、第三に、なるべく和語での言い方を採用しています。これには、耳に優しく響く「ですます調」が向きます。漢字の読み誤りを避けるため、ルビに当たる入力方法の研究も必要になっています。筆者の場合には、括弧を使って読みを補う方法を採用しています。また、コンマ（読点）の適切な使い方と同時に、息継ぎの箇所にも半角スペースを挿入することも試しています。

2. HTML 文書への簡易変換

2.1 Word のファイルから HTML 文書に変換する

書店に行けば、ホームページ作成用の参考書が多く見られます。その多くは、マニュアルの解説を教科書のように並べた編集になっています。首尾の揃った HTML ファイルに構成するときの出発は、原稿文書のテキストファイルをテキストエディタ（メモ帳）で編集することから始めるのが定石になっています。このテキストファイルの拡張子を.txt から.htm または.html に直して、改めてこのアイコンをクリックすれば、IE が**オフライン**(off-line)で起動して閲覧ができます。しかし、初心者がここまでの作業を済ませ、意図したレイアウトで画面が得られるようになるまでには、かなり遠回りの勉強をしなければなりません。最も手っ取り早く WEB 用ファイルを作成するには、Word の機能を利用します。Word 上で印刷レイアウトの表示にしておいて、写真の貼り込みや表を組んだ原稿を完成しておきます。この状態で、表示を **WEB レイアウト**にすると、HTML 文書に直して IE の画面に表示される文書スタイルを見ることができます。HTML ファイルに変換するには、「ファイル→名前を付けて保存→WEB ページ (*.htm, *.html)」と選択して保存します。そうすると、Word(*.doc)と同名のファイル名が付いた HTML ファイルが同じフォルダに作成されます。手作業でテキストファイルを作成するとき、例えば、タグであると区別する半角文字の < と > は < と > に書き換えなければなりません。この作業は鬱陶しいことと、ミスタイプも起き易いのです。Word 原稿から自動で HTML ファイルに変換すると、元の印刷レイアウトとはかなりスタイルが変わりますが、変換全体でエラーがありませんので、そのまま使うこともできます。しかし、テキストの文字並びが複雑ですので、これを短く、簡単にします。この作業が HTML 文書作成の勉強になります。

2.2 変換された HTML ファイルの特徴

HTML のモニタ上の表示スタイルは、文書内容の本質は同じですが、元の Word の印刷表示スタイルとは変わります。元の表示が二段組み、三段組みであっても、WEB スタイルでは一段組みであって、本文の文字列はウインドウ画面横幅に合わせた自動改行になります。ページ区切りも無くなります。変換された HTML ファイルは、実質がテキストファイルです。この中身を見るときはメモ帳で開きます。これには、白紙のメモ帳を開いておいて、HTML ファイルのアイコンをこの画面にドラッグアンドドロップ (drag and drop)する操作を覚えておくと便利です。この文字並びは、かなり複雑です。その理由は、一つの HTML ファイルですべてのスタイル機能を組み込むからです。機能として使うことがないスタイル情報も、大きな行数と文字数で含まれています。使わないデータを削除して、見易いコンパクトな形に直す方法を覚えることを勧めます。一般的な方法の一つがスタイルシート (*.CSS)の利用です。筆者の場合には、さらに、基本機能のテンプレート相当の HTML ファイルを使って、必要箇所を追加するか変更して、多くの HTML ファイルの作成作業を単純化しています。後の項で例題のリストを示します。

2.3 リバースエンジニアリングで勉強する

一般的に技術を覚えるときは、見本を見て真似をすることから始めます。学ぶと言うのは、**まねぶ**の転じた言葉と言われています。リバースエンジニアリング (reverse engineering) とは、工業製品やソフトウェア製品を分解するなどして中身を調査する分析方法を言います。これを行うこと自体は、原則的には合法行為とされています。しかし、調査した結果を真似て、利用するときには、著作権や特許について十分な配慮が必要です。HTML ファイルの作成でも、参考書の例題を真似して勉強することで覚えます。HTML ファイルでは、IE のメニューから、「ソースの表示」をクリックすると、元のテキストを見ることができます。ただし、スタイルシートは WEB サイト側でリンクされるファイルですので、閲覧ができません。この部分については、参考書などで勉強し、自分で開発しなければなりません。リバースエンジニアリングでは、良い見本を見ることが大切です。どれが良くて、どれが良くないかの判断も難しいのですが、一般論として、小中学生が喜ぶ派手な作品を避けます。意味の薄いマンガ風のイラストを実用文書に使っていると、作者の知識レベルが軽蔑されます。実用文書では、ゼロックスコピーの利用も多くなりますので、モノクロでコピーすると区別ができなくなるカラーの使用を避けます。カラーでないと表現できない図 (例えば地質図など) は、工業製図ではハッチングなどを使った表現方法を工夫しています。

3. HTML ファイルの構造

3.1 文書としての論理的構造を理解しておく

一つの文書としてまとめた内容を持たせるときは、下の表のような分類を踏まえます。これは、文書作成のときに言う「起承転結」の構成のうち、「起と承」を具体的に示したものと理解して下さい。

表1 文書の論理的構成

分類	内容	備考	対応する英語
頭の項目	表題		title
	抄録	要旨です。簡潔に内容を紹介をします。	abstract
	目次	ページ数が多いときにはリンク情報を付けます。	table of contents
	用語説明	キーワードですが、普通は本文中に説明します。	glossary
	まえがき	文書作成の背景や管理上の説明を当てます。	preface
文書本体	序論	いきなり主題に入るのではなく、事前情報です。	introduction
	主文	見出しを付けて章・節・項に分けます。	body
	参照文献	外部ファイルとのリンクを作ります。	
	索引	文書内の項目とリンクさせるように構成します。	
終の項目	謝辞など	付録などは別ファイルにすることがあります。	

(参考)に実用文書のまとめ方を見て下さい。pdf版での、p. 12、p. 27の表をもとにまとめました)

3.2 複数のHTMLファイルに構成する

手紙のような一枚もの実用文書は、一つのHTMLファイルにまとめることができ、モニタ1画面を使うようにデザインされます。ページ数が多くなると、複数のHTMLファイルに分割して、インターネットでの送信単位を小さくします。オフラインの環境で使うHTML Helpは、複数のHTMLファイルのセットで構成し、ランダムに項目にアクセスできるようにします。この構成方法をWEB用に応用します。最初にアクセスする先頭のファイル名を決めます。筆者の場合には top.html にしています。ファイル数が多いので、文書の表題名を意味するような名前を付けたフォルダに、関連のあるファイルをすべて収めます。top.htmlには、表1の頭の項目にある表題・抄録・まえがきを含めます。目次と索引は、ファイル名として、index.html, keyword.htmlとして別にまとめます。他のファイルと同じようにリンクを設定して呼び出して表示させるようにします。HELPの画面デザインでは左側のフレームに目次を表示して本文を右の枠に並べますが、表示が複雑になりますので、筆者はこの方法を採用していません。表1の文書本体は、章・節・項に分けて作文しますので、適当な単位で複数のファイルに分けて構成します。項目の見出しをファイル名にする代わりに、筆者は、chapterxxyyzz.htmlの形を採用しています。ここに、xx, yy, zzは、章・節・項に使う、または、その順を示す二桁の番号数字です。

3.3 一単位のHTMLファイルの物理的構造を理解する

一つのHTMLファイルも、表1の文書の論理的構成を踏まえた、物理的な構成方法になっています。

表2 HTMLファイル文の物理的構成

分類	内容	タグ	対応する英語
頭の項目	ファイル管理	<!DOCTYPE HTML ...	おまじないの情報です
	ファイル種別	<HTML>~	ファイル末で</HTML>で締めます
	頭の項目	<HEAD>~</HEAD>	下のブロック項目を挟みます
	表題	<TITLE>~</TITLE>	
	機能の追加	<META ...> <LINK ...>	追加の情報です。複数あります。これも定型のおまじないがあります。
文書本体	書式情報を含めた本文	<BODY>~	ファイル末までがブロックです
		<DIV>~</DIV>	複数の文書単位を集めるときのブロック指示
		<H1>, <P>など	文章文字並びのデータが入ります。この部分の解説が通常の参考書の主題です。
終の項目		</BODY> </HTML>	

4. HTML タグの意味と機能

4.1 編集作業の書き込みとの関連で理解する

一般の人が印刷用の原稿を作成するときの方法は、400 字詰め原稿用紙に文章を書きます。これを編集者のところに持ち込むと、原稿の行間の余白に赤鉛筆で組み版の指示を書き込みます。これを組み版作業に送り、版下印刷(ゲラ刷り)を作成してもらいます。これが返ってきたところで校正しますが、この場合の修整指示も赤で記入します。組み版をコンピュータ化し、コンピュータに指示する用語が組み版言語、ML(markup language)です。HTML 文書の場合には HTML タグ(tag)と言います。タグは、単純化した英字(要素文字並び、element)を<>で囲い、例えば**開始タグ**<P>と、**終了タグ**</P>との間に文字並びを挟みます。開始タグは、より細かな指示をする文字並びを追加記入することができて、これをタグの**属性**(attribute)と言います。属性を含めた開始タグの書き方の文法も理解して使うのですが、文字並びが長くなりますので、利用頻度の高いタグの書き込み作業は手が掛かります。そこで、タグと属性仕様を別ファイルにまとめて定義して参照させる方法が便利です。このファイルを**スタイルシート**と言ひ、英語では**CSS**(cascading style sheet)と言ひます。ホームページを作成するユーザは、モニタの画面上での表示のスタイルを、便利で見栄えが良くなるような希望を寄せますので、HTML 本体の仕様の改訂が進んでいます。XHTML 版、さらには HTML4 から HTML5 への移行などがその現れです。この場合であっても、大部分の基本タグは**継承**(inherit を訳した硬い用語)されます。そうではあるのですが、筆者としては、なるべく HTML4 の仕様の基本機能だけを使う保守的な態度を取っています。タグ要素と属性の種類は多いのですが、以下で紹介するのは、その基本機能に絞ったものです。

4.2 書式全体が準拠する仕様はファイルの最初に置かれる

ファイルが「HTML 形式である」とコンピュータに知らせる記述は、ファイルの拡張子(.htm、または.html)と、ファイルの先頭にある<HTML>タグです(表2参照)。この行の前に、<!DOCTYPE …で始まる宣言文で、どのバージョンの HTML に準拠するかの情報を記述します。これを **DTD**(document type definition)と言ひます。HTML 4 の場合は、

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

HTML 4 の仕様で HTML 文書を作成するとき、この情報が無くてもよいのですが、Word 文書から HTML 文書に変換すると、この文が自動的に入ります。ただし、Word の新しいバージョンでは、<? …で始まる文や、開始タグ<HTML>中に属性として書くようになりました。一般のユーザは、これらの文は、**おまじない**として、そのままコピーして使ひます。DTD 違いは、モニタ画面の表示に影響することがあります。特殊な表示方法に凝らなければ、問題は起こりません。<HEAD>と</HEAD>とで囲われた文の集合(ブロック)は、管理情報です。ここでの重要な文は、文字コードの宣言です。日本語の環境は、下のように入ります。

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

この文も無くても問題を起こしませんが、通信回線でデータが送受信されますので、稀にコード違いで表示画面の文字化けが起こることがあります。これも、**おまじない**として、そのまま使ひます。

4.3 標準化された文字フォントを使うのがよい

印刷文書の書式で重要なものが、**活字**(typeface)の、デザイン図形としての書体と寸法です。これを**フォント**(FONT)と言う概念で括ります。日本語の環境では、大きな分類は、**明朝体**と**ゴシック体**(gothic)の二種です。出版社や新聞社では、自社固有のデザインを使っています。機械式の英文タイプライタでは、活字寸法として横幅 1 インチに 10 字入る**パイカ**(pica:12 ポイント)が標準であって、やや小さめに 12 文字が入るものを**エリート**言ひました。行詰めはどちらも 1 インチ高さ 6 行詰めを標準としていました。日本語のタイプライタもこれと寸法を合わせました。日本語の活字幅を全角と言ひ、英字はその半分幅(半角)で使うのがコンピュータでは標準です。モニタにシステムが表示するフォントを**システムフォント**と言ひ、**等幅**のゴシック体を使ひます。Windows では**MS ゴシック**です。レーザプリンタが使えるようになって、プロポーショナルフォントでも印刷ができるようになりました。筆者は、元になる Word 文書には、眼に優しい等幅の明朝体フォントで作成しています。強調したい文字は、ゴシック体の代わりに**太字**(bold)にしています。しかし、HTML 文書は、モニタ上で見ることが目的ですので、デフォルトのシステムフォントで表示するようにして、テキストファイルから書体情報の方を消去しています。

4.4 寸法も相対表示で指定するのがよい

インターネットを IE でモニタに表示すると、モニタの解像度に合わせて、タイトルバー、メニューバーの配置はモニタの横幅一杯で表示されるように尺度がデザインされています。そこに表示される文字は、システムフォントが使われ、標準寸法で書かれます。フォームの横幅寸法を狭く変えても、適度な折り返し表示になりますが、文字寸法やアイコンの尺度は一定です。一方、フォーム内の表示領域は、全体尺度を拡大・縮小させることができます。マウスのアイコンを表示画面上に置いて、「Ctrl, +」「Ctrl, -」のキーボードショートカットで変えることができます。全体画面が小さくなると左右に余白が入り、大きいと左右が欠け、水平スクロールバーが現れますので、移動させて見るすることができます。この機能がありますので、HTML 文書の中での文字寸法をポイント数で絶対指定をするよりも、基準のデフォルト寸法に対しての相対的な大小指定 (%指定) を採用します。

4.5 タグは文字単位の書式の表し方から覚える

文書の編集は、最小の図形単位としての活字図形を並べて、図と共に、決められた用紙寸法を埋める作業です。活字の、図としてのスタイルを指示するタグの中、基本的なものを表3に示します。タグの英字は、初期のHTML4では大文字を使いましたが、最近では小文字を使う方向に進んでいます。表示ソフトは大文字と小文字を区別しませんが、ここでは大文字で説明します。

表3 文字または文字並びのスタイル指定

タグ	日本語での意味	元になる英語
~	字体、寸法、色などを属性で指定します	font
~	太字	bold
<I>~</I>	イタリック	italic
[~]	上付き	superscript
_~	下付き	subscript
<>U>~</U>	下線 (アンダーライン)	underline

4.6 行単位のスタイル指定を理解する

表3の、文字単位のスタイル指示された文字並びを、行の中で配置する方法を設計します。ページの横幅を一単位とした行(line)の使い方です。ウインドウの表示領域(フォーム)を用紙領域として、上下左右の余白(margin)を指定することができます。一行分の横幅を複数の領域に分ける方法を使うと、見栄えのよい画面表示になります。用紙に印刷するときは、**段組み**(段落と間違えなように)と言います。横書きの文書ではカラム(column)です。HTML文書では種々の方法を使うことができます。簡単な方法は、表枠を使う方法です。行単位に文字列を書き込むとして、文字並びの種類が二つあります。見出し(ヘッダー: header)と本文段落(パラグラフ: paragraph)です。本文に使う活字寸法はデフォルトで決めます。ヘッダーは活字寸法を指定することもできますが、デフォルトでは本文の活字寸法を基準として或る高さ比の寸法系列が指定されます。行の高さはこの高さで決まります。HTML文書では、枠単位の背景色と文字色を属性で指定することができます。

表4 行単位の文字並びのスタイル指定

タグ	日本語での意味	元になる英語
<BODY>~</BODY>	用紙領域の使い方を指示します。属性として重要なものが余白(margin)の定義です。	body of report
<DIV>~</DIV>	スタイルシートで文字集合単位を区別したいときに、主に使います。	division
<H?>~</H?>	見出し文字並び。ただし?は1,2,...6です。	header
 	改行指示。終了タグ</>はありません	line break
<HR>	水平線を挿入します。終了タグはありません。	horizontal rule
<CENTER>~</CENTER>	中央揃え、センタリング 右揃えなどはALIGN属性で指示します。	center ; アメリカ英語であることに注意します。

5. 文字並びの特別な表示方法

5.1 スペースを含む書式で表示する方法

HTML 文書は、英語の環境で開発されましたので、単語の分ち書きをし、その間に半角のスペース（空白）を入れます。HTML 文書のソーステキストは、読み易さのために「空白・タブ・改行・復帰」を適度に使います。しかし、表示のときは、これらはすべて空白扱いをし、単語の切れ目に使う一つの空白を残し、他はすべて無視されます。この表示規則は、プログラムのソースコードを表示したいととして、適度なインデントを使うリストを引用表示するとき、文字並びが連続表示になりますので困ります。もとの表示を生かすようにするには、そのリストの前後に<PRE>のタグを使います（表 5）。

5.2 全体を字下げで表示したいブロック

パラグラフ単位を指定する文字列集合<P>…</P>単位は、行集合の前後に一行分の空白が入ります。しかし、引用文の集合全体をタブで字下げしたいことがあります。一行で納まる文字並びの場合には、半角のスペース文字 が定義されていますので、それを必要個数だけ並べるか、センタリング<center>で代用することができます。複数行全体を字下げ（インデント:indent）したいときは、<BLOCKQUOTE>のタグを使います（表 5）。

表 5 行単位の並びのスタイル指定

タグ	日本語での意味	元になる英語
<PRE>…</PRE>	スペースを含むそのままの表示	paragraph の render
<BLOCKQUOTE>…</BLOCKQUOTE>	複数行をまとめて字下げする	block quotation

5.3 簡条書きにしたいとき

簡条書き（リスト）で行の文字並びを表示する方法として三つの方法があります。一つは、順に頭に番号を付けるもの、二つには、番号の代わりに行頭に黒丸などの記号（●;bullet）を付けるものです。簡条書き文は、本文から字下げで表示されます。このリストは、入れ子で使うことができます。例えば、目次を章単位で番号表示のリストに作り、その章の下位の節番号を付けたリストを子リストに構成することができます。三つ目に、行頭記号を使わないリスト形式があります。用語説明の文に応用します。見出し文と説明文とを一種のリストの形体を持たせた構造に組み立てることができます。

表 6 リストに構成するスタイル

タグ	子タグ	日本語での意味	元になる英語
…	…	番号付きリスト	ordered list
…	…	黒丸などを頭に付けた番号無しリスト	bulleted list
<DL>…</DL>	<DT>…</DT>	用語説明の見出し語の方	definition list
	<DD>…</DD>	説明文の方	definition term

5.4 表に構成したいとき

表の形式を使うと、非常に応用の広い表示方法が得られます。表の枠線を描かないようにすると、見掛けを段組みで表示したような表示画面が得られます。これを、0 からテキストエディタで編集する作業は面倒です。Word を使って GUI の環境で完成イメージの表を作成して HTML テキスト形式に変換するのが便利です。

表 7 表に組むときのスタイル指定

主タグ	子タグ	孫タグ	日本語での意味	元になる英語
<TABLE>	<CAPTION>…</CAPTION>		表の見出し	table caption
		<TR>…</TR>	<TH>…</TH>	行方向セルの見出し要素
</TABLE>	<TR>…</TR>	<TD>…</TD>	行方向のセル要素	header

6. 文章スタイルの設定と表示

6.1 標準として提案する印刷スタイル

筆者がこの報文で言う実用文書の印刷スタイルは、A4版の用紙を縦に使う横書きレポート書式を基本に考えています。このときの標準スタイルの数値は表8です。

表8 A4用紙を縦位置にして横書きにするときのスタイル

	寸法 (ミリ)	寸法 (インチ)	ポイント数	用紙横幅との比
用紙の横幅	210 ミリ	8.3 インチ	6959	100 %
用紙の高さ	297 ミリ	11.7 インチ	8427	141 %
左右の余白	20 ミリ	0.8 インチ		5 %
上の余白	20 ミリ	0.8 インチ		
下の余白	25 ミリ	1 インチ		
有効横幅	170	6.7 インチ	4819	90 %
標準活字寸法	10.5 ポ	(1/13)		
標準字詰め (全角)	45 字			
標準行高さ	12 ポ	(1/6) インチ		
標準行詰め	49 行			

6.2 モニタ上での表示スタイル

モニタを文字表示 (character display) で利用するとき、半角英字が一行 80 字並べて読めるようにすることで、最初の解像度 (resolution) に 640×480 ピクセルが採用されました。この横縦の画面寸法比は、旧のテレビ画面寸法比の 4:3 です。線図を主としたグラフィックスに應用すると、やや解像度が不足しますので、横方向のピクセル数が 640→800→1024→1280 のように解像度の高いモニタが開発されてきました。大きな画面でピクセル数が大きいと、横幅に入る文字数を増やすことができます。読み易さからは、適度な文字寸法と文字列の幅があります。インターネットのプロバイダのホームページは、モニタ全画面を使う画面スタイルに、横方向を幾つかの小単位の枠に分けています。HTML 文書を、モニタ全画面を使って、システムフォントの標準文字寸法で表示すると、段落単位の文字数が多くなります。元の A4 版の原寸スタイル幅で見たいときは、IE 表示のウインドウの横幅寸法を手動で狭くします。または、表示領域を拡大することと併用します。HTML 文書は、表示領域の寸法が変わっても、文書の文字列が自動改行され、右端が欠けることはありません。ただしグラフィックスがあると そうはなりません。この説明は後の 6.6 節を見て下さい。

6.3 文字並びは何通りかのスタイル指定方法がある

スタイルを指定する基本的な方法は、或る範囲の文字列の始めと終りとを開始タグと終了タグとで挟みます。HTML ファイル全体に適用するスタイルの例として、本文に使う文字フォントの書体と寸法があります。何も指定しなければ、システムフォントが初期値 (デフォルト) として使われます。それ以外のスタイルを指定したいときは、HEAD の領域に <BASEFONT …> を入れます。しかし、局所的な文字並びに、共通スタイル以外の指定をしたいことがあります。このときは、文字列の前後を ~ で囲みます。これを インライン のスタイル指定と言います。文字並びは、普通、ヘッダータグ <H?> ~ </H?>、または段落単位のタグ <P> ~ </P> で区切ります。この開始タグの中に、その文字並びでだけ使う属性を含めることができます。このときは、属性を書き込んで、例えば <P …> の形です。段落文字並びの途中で、部分的に ~ で区切って別のスタイルを指定することができます。しかし、 が現れた時点で、<P …> の仕様に戻ります。これを <P…> の中での属性の 継承 です。しかし、別の段落範囲には継承されません。そうすると、同じ属性指定を繰り返して書かなければなりませんので、テキストファイルの文字数も増え、また読み難くなります。そこで、定型的な追加の属性指定に名前を付けて、これを CLASS="name" の形で参照する方法を使うことができます。HTML ファイル単体で利用するとき、HEAD の領域にスタイル情報をまとめます。これを別ファイルのスタイルシートにまとめ、このファイル名をリンクする方法が推奨されています。これが、拡張子 (.CSS) のついたテキスト形式のファイルです。CSS は、cascading style sheet の頭字語です。動名詞の cascading が使われていますが、日本語に訳し難い語です。トランプのカードをずらして重ねて部分的に見える形がそうです。菊の花の栽培に懸崖があります。これが cascade に当たります。

6.4 空きの設定

文字並びは、行単位を矩形枠とした中に納めます。表枠も幅を区切った行として扱います。行の使い方が文字並びのスタイルの、一段階上のスタイル指示です。行の高さは、読み易さを助けるため、文字の高さ寸法よりも大きくします(表8参照)。文字並びを、行の矩形枠内での配置を言う属性用語がALIGNです。単にALIGNと使うときは、文字並びの、左右の空きの使い方を言います。左詰め(LEFT)が標準です。中央揃え(CENTER)・右詰め(EIGHT)も指定できます。文字並びの高さ方向の配置は、属性用語ではVALIGNです。標準は上詰め(TOP)です。中央(MIDDLE)、下揃え(BOTTOM)を指定できます。細かいことを言うと、枠線を見えるように描いたとき、文字と枠線とが接すると見難い表示になりますので、幾らか隙間をいれることができます。鉛の活字を使った組み版では、文字の間隔または文字並びの行間、表の場合は上下または左右の枠の隙間、などを設定するときは、込め物(space または pad)を挟みました。HTMLの用語では、動名詞のpadding (パディング) が隙間の寸法指定に使われます。この設定作業は非常に専門的ですので、経験のある植字工が作業をしていました。したがって、HTMLタグでこれらを細かく設定して、見栄えのよい表示にしたいとなると、かなりの経験が必要です。筆者の場合には、パディングの設定はすべて削除しています。表枠に記入する文字並びの始まりが、左の枠線に接するように表示されます。これは、実用的には大目に見てもらいます。

6.5 余白の設定

紙に印刷するときは、用紙の上下左右に余白を取ります。綴じると情報が見難くなることを防ぐ意義もあります。このスタイル設計は、行の並べ方を決めることに当たります。パソコンのモニタ全画面をデスクトップと言います。通称で言うウインドウは、文字や図を含む情報単位の表示枠ですが、プログラミングの用語ではフォームです。標準のデザインは、上段にタイトルバー、メニューバーがつき、必要に応じて、ツールバーや、右端と下縁にも追加の枠がつきます。残りが作業領域です。余白の設定は、プリンタの方に指示する項目ですので、WYSIWIGが保証されていない環境では、モニタ上のスタイルと、印刷したスタイルとは異なります。Wordの場合には、表示を印刷レイアウトにすると、印刷用紙上の印刷仕上がりが見られます。このときの余白の設定は、一例として表8に示しました。横幅一杯に文字列が長く並ぶと読み難くなりますので、一般の図書は段組みをして一行の文字数を抑えます。IEの文字表示は原則として段組みができません。HTML文書を読むときは、ウインドウの横幅を狭くするか、全体の画面を大きくして、文字並びの自動改行をするのがよいでしょう。モニタ上で文書を見るだけの場合であっても、ウインドウの表示領域の左右と上下に幾らかの余白(MARGIN)を設けると、目に優しく、読み易くなります。この寸法は、<BODY>タグの属性で設定できます。この属性スタイルの指定は、共通項目としてスタイルシートを使います。幾つかのスタイルを決めて、場面に応じて使い分けたいとき、個別に名前を付けておき、その名前を、CLASS="name"として参照するように使うことができます。スタイルシートに"name"が見つからなくても、エラーにはなりませんので、大らかに使うことができます。

6.6 大寸法のグラフィックスを使わない

A4の用紙寸法で見るときの写真の寸法は、キャビネ判(11.5×16.5cm)か、その半分寸法が扱いやすい単位です。デジタルカメラの解像度が上がって、大寸法でも高精度デジタル写真が得られるようになりました。モニタ画面で見える範囲であれば、最大で1280×960ピクセル(pixel;ピクセル)あれば十分です。この縦横のピクセル数を掛けたものをデジタルカメラの画素数と言い、上の数値は約120万画素です。寸法の大きな画像は、ファイルのバイト寸法も大きくなりますし、通信回線を介してデータを送る際の応答が遅くなります。また、画面に納まらないことも起こります。そこで、筆者は、Wordを印刷レイアウトにして、任意の寸法の画像データを挿入し、その画像枠の隅をマウスでドラッグし、ページ幅に納まるように変更しています。横幅の大きな画像は、ピクセル数にして640の90%程度、縦長の画像は480の90%程度を目安にしています。見掛け上二段組みにしたいとき、画像の横幅をページ幅の約1/2にしたものも扱いやすい寸法です。このように調整したWordの原稿をHTMLファイルに変換すると、寸法調整をした画像データと、元の画像データとが付属のファイルに記録されます。したがって、ファイル寸法を小さくし寸法調整後のファイルを使って、HTMLファイルを書き直します。Word原稿に載せた方の図も寸法調整後の画像ファイルに置き換えると、用紙に印刷する図の解像度は幾らか下がりますが、Wordのファイル寸法、さらにはPDFに変換したファイル寸法も節約できます。

7. 表の作成と応用

7.1 文書の表示は表の作成が応用されている

印刷物は、指定された用紙の領域を、種々の図形単位で埋めていく作業です。気が付かないことですが、標準的な文字並びのスタイルは、物理的には1列×複数行の表に構成されています。表の枠線（罫線）を表示していないだけです。このことを積極的に利用すると、弾力的なレイアウトで全体スタイルをデザインすることができます。筆者のHTML文書は、文書全体の大枠を、枠線を使わない<TABLE>～</TABLE>で囲っています。行単位は<TR>であって、それを分割する単位が<TD>です。一行を二つ以上の領域に分けません。こうすることで、一行単位で背景色と文字色とを指定できますので、見出しの部分をタイトルバーのようにデザインすることができます。行の領域に書く文字並びは、パラグラフのタグ<P>を使うまでもないので、省いてあります。文字数が多くて、一行の横幅に入りきれないと、自動改行が効いて、複数文字並びが納まるように行の高さ、つまり表としての枠高さが高くなります。<TABLE>タグは、入れ子式に、<TD>枠内に使うことができます。こちらは枠線を書き、部分的に左右上下の複数セルを繋ぐこともしています。この文書で使っている表がそうです。

7.2 表の作成にはテンプレートを使う

一般的な表を組み上げるようにソーステキストを作成するのはかなり面倒です。主要な注意は、見栄え良く作成するための列幅の寸法定義です。実践的には、Wordの画面でレイアウトを作成しておいて、HTMLファイルに落としたテキストファイルの原稿を修正することです。このとき、表のテキスト並びを制御するタグの組み立てが、下のようになっていることを理解しておきます。これは、列幅指定をピクセル数で指定した場合と%指定をしたときの比較です。このどちらも、IEの表示画面で、ウインドウの横幅を変化させると、列幅も相対的に変化し、文字幅に対してセル幅が狭くなると、文字列はセル内で自動改行されます。

<pre><table border> <caption>表…幅指定がピクセル</caption> <tr> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> </tr> <tr> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> </tr> <tr> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> <td width=160>&nbsp;</td> </tr> </table></pre>	<pre><table border width=90%> <caption>表…幅指定が%</caption> <tr> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> </tr> <tr> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> </tr> <tr> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> <td width=20%>&nbsp;</td> </tr> </table></pre>
---	--

7.3 テンプレートの HTML ファイルを利用する

Word の印刷レイアウトの作業画面から WEB 用に変換した HTML ファイルを作ると、同名のファイル名で、拡張子(.htm)または(.html)のついたファイルと、同じ名前に(.files)の付いたフォルダが追加されることがあって、グラフィックスデータが入ります。ただし、この変換操作では、スタイルシートのファイルを作成しませんので、HTML ファイルのテキスト構造が非常に複雑になっています。これは、属性を、個別の文字列単位ごとに定義するからです。まず、この HTML ファイル単体で共通に使うスタイル名は、<HEAD>~</HEAD>の中で<STYLE>~</STYLE>の中にまとめます。さらに、個別の<P>、<H?>の開始タグ、さらには文字列中でもを使ってインラインでスタイルが指定されます。したがって、重複した記述も多くなります。このテキスト並びを軽快で見易くすることを勧めます。この作業は、2.3 節で説明したリバーズエンジニアリングです。しかし、HTML 文書ではファイル数が多くなりますので、筆者は、テンプレートのファイルを作成しておいて、書き込み追加か、部分的な変更をするようにして作業の効率化を図っています。見本として、下の図 1 と、次ページの図 2 とを見て下さい。

目次ページ 前ページ 次ページ

1. 引張材と圧縮材

1.3 偏心载荷を考える簡単な組み合わせ構造

1.3.1 偏心して作用する力を受ける構造

図1.7 空間直線の定義を示す見取り図

単独に考える真っ直ぐな引張材・圧縮材の力学モデルは、細い棒です。実際の部材は或る断面積がありますが、断面の重心を通る軸を直線材とする力学モデルで図示します。或る断面形状をした部材は、細い棒状の部材を束ねたと考えます。この扱いは、第4章で説明します。ここでは、数個のバネの集合で構成するモデルを解説します。まず、水平な台に等間隔に並べたバネで、曲げ剛性の大きな梁を支え、その梁の上から荷重を載せる力学モデルを説明します。このモデルは、バネ支承の本数が多くなっても簡単に解析できて、種々の場面で考えられています。バネが等分布している場合の例は、弾性的な地盤の上に建物の剛な基礎構造が載っているモデルとして扱います。矩形断面の圧縮材に軸力が偏心して作用するとき、最初に平面であった断面形が圧縮力を受けて変形しても、断面が平面を保つ平面保持の仮定を使います。橋梁工学では、並列した何本かの主桁を支間中央の分配横桁で繋ぐ格子構造を解析するとき、主桁の作用が横桁に対してバネ支承になる構造を考えるのもそうです。

科学書刊株式会社:電子版「橋梁&都市 PROJECT: 2011」

前ページ 次ページ

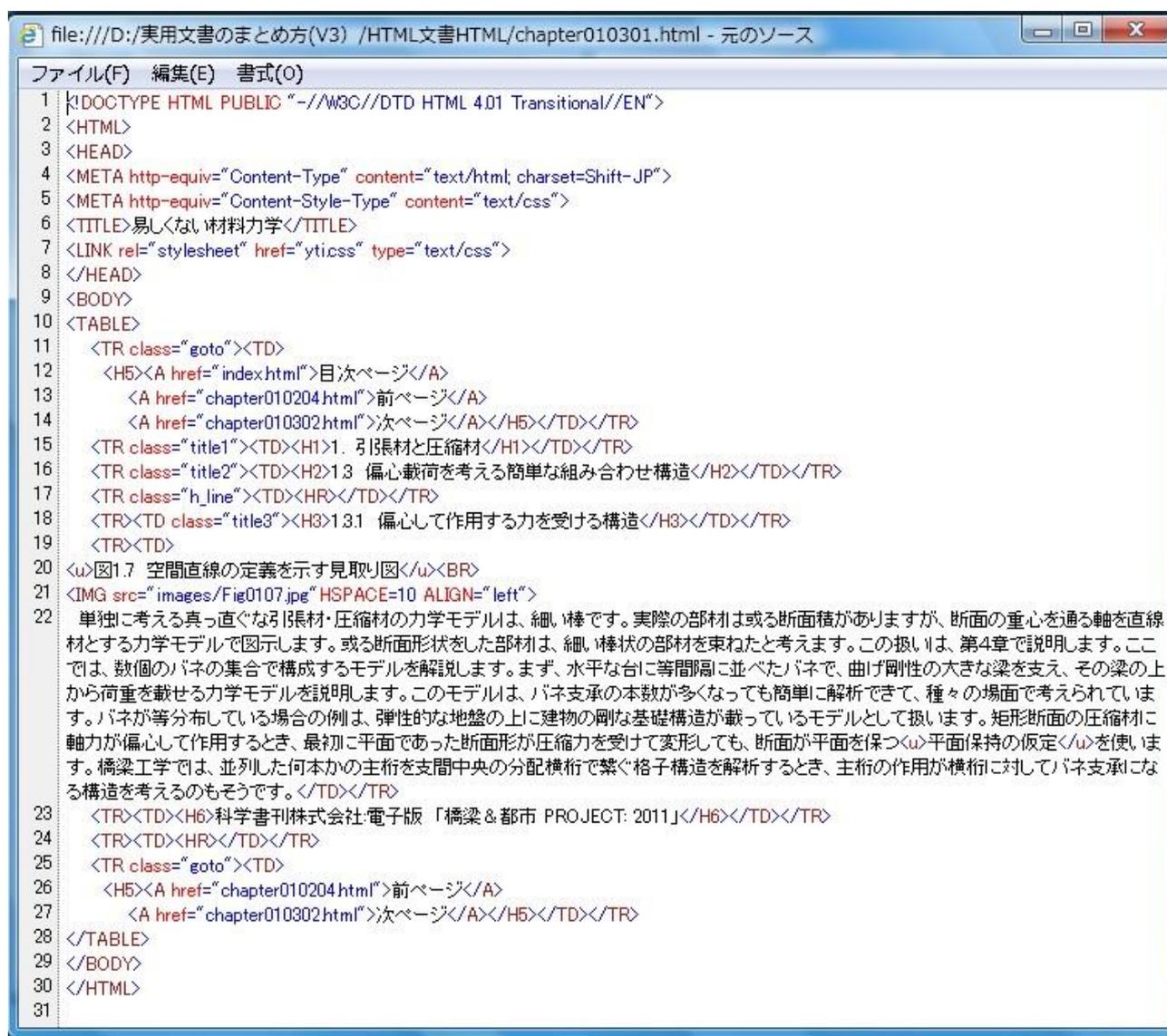
図 1 IE で表示した HTML 文書の例：テンプレートしても使うスタイル

7.4 テンプレート用ファイルの説明

下の図 2 は、前ページの IE 表示画面図 1 から、ソースの表示で出力したテキストです。

- 行番号 1～10 は、すべてのファイルで共通して使います。ただし
- 行番号 6 はタイトル（表題）の文字並びです。これは、文書の種類で書き換えます。
- 行番号 13～27 が、ファイルごとの書き換えをする部分です。
- 行番号 13, 14 は、ページの前後へのリンク情報です。同じ書式が行番号 26, 27 です。
- 行番号 15, 16, 18 が、小・節・項の見出しの文字列です。
- 行番号 22 は、複数行の表示が並びますが、この文字並びが一つの段落（パラグラフ）です。
- パラグラフ単位を示すタグ<P>を使うまでもないので、ここでは省いてあります。
- ヘッダーのタグは、リンクしたスタイルシートに属性が追加されています。
- 文字の背景の色指定、文字の色指定などの属性は、スタイルシートの方に書いてあります。
- 行番号 28～30 が、ファイルの終わりの定型です。

この HTML ファイルでは図が挿入されています。図は原則として左詰めです。図の表題（キャプション）は、印刷物では図の下側に入れるのですが、ここでは図の上へ書き、本文を図の右に回り込むように表示させてあります。



```
file:///D:/実用文書のまとめ方(V3)/HTML文書HTML/chapter010301.html - 元のソース
ファイル(F) 編集(E) 書式(O)
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <HTML>
3 <HEAD>
4 <META http-equiv="Content-Type" content="text/html; charset=Shift-JP">
5 <META http-equiv="Content-Style-Type" content="text/css">
6 <TITLE>易しくない材料力学</TITLE>
7 <LINK rel="stylesheet" href="yti.css" type="text/css">
8 </HEAD>
9 <BODY>
10 <TABLE>
11 <TR class="goto"><TD>
12 <H5><A href="index.html">目次ページ</A>
13 <A href="chapter010204.html">前ページ</A>
14 <A href="chapter010302.html">次ページ</A></H5></TD></TR>
15 <TR class="title1"><TD><H1>1. 引張材と圧縮材</H1></TD></TR>
16 <TR class="title2"><TD><H2>1.3 偏心載荷を考える簡単な組み合わせ構造</H2></TD></TR>
17 <TR class="h_line"><TD><HR></TD></TR>
18 <TR><TD class="title3"><H3>1.3.1 偏心して作用する力を受ける構造</H3></TD></TR>
19 <TR><TD>
20 <u>図1.7 空間直線の定義を示す見取り図</u><BR>
21 <IMG src="images/Fig0107.jpg" HSPACE=10 ALIGN="left">
22 単独に考える真っ直ぐな引張材・圧縮材の力学モデルは、細い棒です。実際の部材は或る断面積がありますが、断面の重心を通る軸を直線材とする力学モデルで図示します。或る断面形状をした部材は、細い棒状の部材を束ねたと考えます。この扱いは、第4章で説明します。ここでは、数個のバネの集合で構成するモデルを解説します。まず、水平な台に等間隔に並べたバネで、曲げ剛性の大きな梁を支え、その梁の上から荷重を載せる力学モデルを説明します。このモデルは、バネ支承の本数が多くなっても簡単に解析できて、種々の場面で考えられています。バネが等分布している場合の例は、弾性的な地盤の上に建物の剛な基礎構造が載っているモデルとして扱います。矩形断面の圧縮材に軸力が偏心して作用するとき、最初に平面であった断面形が圧縮力を受けて変形しても、断面が平面を保つ<u>平面保持の仮定</u>を使います。橋梁工学では、並列した何本かの主桁を支間中央の分配横桁で繋ぐ格子構造を解析するとき、主桁の作用が横桁に対してバネ支承による構造を考えるのもそうです。</TD></TR>
23 <TR><TD><H6>科学書刊株式会社電子版「橋梁&都市 PROJECT: 2011」</H6></TD></TR>
24 <TR><TD><HR></TD></TR>
25 <TR class="goto"><TD>
26 <H5><A href="chapter010204.html">前ページ</A>
27 <A href="chapter010302.html">次ページ</A></H5></TD></TR>
28 </TABLE>
29 </BODY>
30 </HTML>
31
```

図 2 テンプレートに使うことができる HTML ファイルのソースコード（図 1 参照）

8. 目次と索引の作り方

8.1 ページ番号を使わないこと

目次と索引とは、文書本体が完成した後での作業です。そのため、紙に印刷する文書は、本文のページ番号を必要としますので、目次と索引は、本文とは別のページ番号を切ることになります。雑誌の場合は、表紙から通しのページ番号を付けます。ページ数の少ないレポート形式の文書も通してページ番号を付けます。目次も索引も付けないこともありますが、目次を付ける場合であっても、番号付きの章・節・項のリストで済みます。この場合の目次は、抄録としての目的があります。ややページ数が多くなる A4 版の実用文書では、頭の項目と文書本体とに別のページ立てをすることもありますが、実践的にはコピーを依頼するときのページ抜けや重複の確認と、全体の文書量が何ページになるかが情報として必要です。実際の作業でも、目次や索引にページ番号を付けることは手間が掛かります。電子化文書の場合には、ページ番号を目次や索引に使うことができません。それに代わる数字に、筆者は章・節・項に番号を付けておいて、その番号を使います。HTML 文書は、モニタをスクロールしなくて読める分量の文字数で項立てにし、項の単位でファイル化し、これをリンクして呼び出す単位にします。

8.2 目次と索引に宛てる印刷ページ的设计が必要

目次は、それを通覧するとき、文書の内容を知る情報として読みます。読み物として編集される文書は、頭から通読しますので、目次は簡単に済ませることもします。技術系の文書は、ある目的意識を持って内容に当たる読み方もしますので、適度な詳しさの目次があると役に立ちます。ただし、適度の基準は難しいのですが、章・節・項の三分類の見出しを使う場合には、目次を節単位で止めて、それを補うように索引を使い分けます。目次の分量は、印刷ページ数にして 2 ページ前後に抑えるのが手頃です。索引は、項目数が多くなりますので、印刷ページでは段組みをして余白を減らし、ページ数の増加を抑えます。しかし、HTML 文書では段組みにする方法が使えません。筆者の方法は、用語は章・節・項の番号と対にして、それをセミコロン (;) の区切り符号を付けて連続文字並びにしています。段落区切りを「あ、い、う…」の見出しにしています。見本は WEB 版を見て下さい。

8.3 エクセルを編集作業で利用すること

目次と索引は、同一文書群へのリンクの集合です。元になる Word の文書をコピーして、目次と索引語を残すようにして本文を削除したものを作業用ファイルとします。これを、目次の集合ファイルと、章・節・項の番号を追加データとした索引語のファイルとに分けます。索引語を「あいうえお順」に並べ直す方法として、筆者は、索引語の並びをテキストファイルに落とし、それをエクセルで読み込んで作業をしています。漢字熟語は、訓読みがありますので、JIS のコード順では位置が狂います。そのため、読みの見出し語一文字をつけた列を仮に追加しておいて、並べ替えの補助にします。これをタブ区切りのテキストファイルに落とし、これを再度 Word に読み込んで、三段組みの索引に編集し直します。これが、PDF ファイルにしてある索引のページです。元にしたエクセルのファイルを元に、CSV 形式に落とし、テキストからカンマ記号をすべて削除することで、索引用の HTML ファイルに変換しています。HTML ファイルでは索引語の三段組みができませんので、行数を増やさないための方法が上で説明した、あいうえお順の段落構成です。

あとがき

HTML 文書の作成の参考書は、主にホームページの作成を主題として書店に行けば多くを見られます。著作の大部分は、マニュアルの目次構造順に編集されています。しかし、マニュアルとは、全体の知識が有った上で、部分的な文法や規則などの記憶の確認用に要点を編集したものです。記憶の確認と、普段使わない機能の説明を見るため、ランダムに内容を探します。参考書の著者は、全体についての総合知識がありますので、マニュアルを解説するような順で精一杯 易しく書いたつもりでしょうが、そこには大きな思い込みがあります。初心者は、部分的に丁寧な説明であっても、それが応用される全体との関係が分からないのです。ここにまとめた「HTML 文書作成の手引き」は、全体が見本となるように、首尾を整えた文書です。本文は WEB 版から、現在動作中のソースを見ることができるようにして、紙に印刷しただけの参考書とは、一味違った、実務者向けの編集にしました。

2011-5 月版