第 3 回：テキストデータと文書構造の表現

1. テキストデータと文書の構造

- テキストデータの特性 (前回の復習)
  - 文字情報以外のデータを含まない。
  - 原則として単一の文字エンコード方式しか使えない。
  - 文字の大きさやスタイルなど書式情報は一切保存されない。
  - タブ tab (ASCII コード 09, Unicode 0009)，スペース space (ASCII コード 20, Unicode 0020), 改行などもコードで表現される。このうち改行記号は基本ソフト (OS) によって表現方法が異なる。

- 同じテキスト形式のファイルでも，拡張子でファイルの種類を明示することができる。
  - 内容は単なるテキストデータであっても，用途によって拡張子のつけ方が異なる。
  - Windows では拡張子により起動するアプリケーションを関連付けている。

ファイルの種類	拡張子	大学 PC で関連付けられたソフトウエア
テキスト	txt	EmEditor
CSV ファイル	csv	Excel
HTML 文書	html, htm	Internet Explorer
Perl スクリプト	pl	Perl
XML 文書	xml	Internet Explorer

- 文書の構造 structure とは？
  - データの区切りを表すには？
  - データとデータの関連を表すには？
  - これらの情報を明示するには？

  テキストデータで文書の構造を表現するには，構造を表示する「しるし」を**テキストとして**加える→**通常のテキストと，文書構造を表すテキストとの区別を明確にする必要がある。**

- 構造の表示方法
  - データを規則的に入力していく：固定長テキスト，タブ区切りテキスト, CSV, その他
  - データの構造を表示するための「メタデータ」metadata を使う：HTML, SGML, XML

- テキストデータの文書構造とソフトウエア
  - テキストデータはソフトウエアを選ばないので，エディタなどを使って自由に編集できる。
  - 文書構造によっては専用のエディタや編集ソフトが存在する。また，表計算ソフトが対応する形式もある (§2 参照)。

2. 規則的なデータの入力：固定長テキスト，タブ区切りテキスト，CSV

テキストデータで文書構造を表す方法として最も典型的なのは, 特定の記号でデータの区切りを表すことにより，テキストデータを「簡易な」データベースとして利用するものである。

- データベースの定義：
  - データをたくさん集め，規則に従って並べたもの
  - データベースソフトを使っているか，検索システムがあるか，は必ずしも重要でない
  - 典型的なデータベースの構造：Excel のワークシート

- データベースの基本用語
  - データベースとしての「表」：**リスト** list
  - データの基本単位「行」：**レコード** record
  - 各レコードの項目「列」：**フィールド** field
  - 「見出し」 (多くは最初のレコードとして記載)：**フィールド名**

2.1. **固定長テキスト** (fixed-length value)

- 行がレコード
- 各レコードは，行頭からの文字数をそろえることによってフィールドの区切りとする

2.2. **タブ区切りテキスト** (tab separated value = TSV)

- 行がレコード
- フィールドの区切りはタブ tab

2.3. **カンマ区切り** (comma separated value = CSV *) *CSV **は頻出用語なので覚えるとよい**

- 行がレコード
- フィールドの区切りはカンマ comma (,)
- フィールドの内容としてカンマが含まれる場合には，フィールド全体を 2 重引用符 (") で囲む。ソフトウエアによっては，スペースが入る場合などにも引用符を自動的に挿入する場合がある。
- フィールドの内容として 2 重引用符が含まれる場合には，2 重引用符を 2 回入力する (2 重引用符 1 つでは，フィールドの内容を囲む記号と誤って判断されるので注意)。

2.4. **その他の簡易データベース**

- 行の区切り，フィールドの区切りを特定の文字に定め，オリジナルの形式でリストを作ることもできる。
- 規則的にデータが入力されたテキスト形式のファイルを読み込める一般的なソフトウエアで，どの程度オリジナルの形式のリストが利用できるかは，ソフトによって異なる。

	固定長	タブ区切り	CSV
レコード	行	行	行
フィールド区切り	行頭からの文字数	タブ	カンマ
利点	見やすい	エディタなどで処理しやすい	多くのソフトが対応する一般的な形式
欠点	コンピュータが処理にしくい	タブでインデントが行われるが，フィールドが揃わない場合は多少見づらい	見づらい カンマのほかに引用符が使われ，テキスト処理の観点からは多少複雑

表 1: さまざまな簡易データベース構造のまとめ

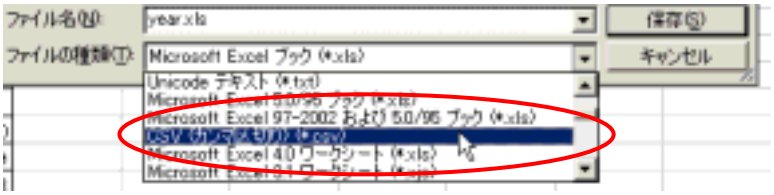
担当：千葉 庄寿 (CHIBA Shoju) e-mail: schiba@reitaku-u.ac.jp

実習１：Kadai サーバの [schiba] → [kenkyuu2005] → [No3] フォルダを file_server の Home にある kenkyuu2005 フォルダにコピーしなさい。file_server の Home にコピーした No3 フォルダを開き，year.xls を Excel で開きなさい（「Excel ブック」形式の拡張子は xls)。

実習２：実習１で開いた year.xls **を §** 2.2 **～ §** 2.3. **(ファイルの種類をそれぞれ「テキスト (タブ区切り)」「**CSV **(カンマ区切り)」と選択) の形式で** No3 **フォルダに保存しなおしなさい。ファイル名は以下のようにすること (全て半角で入力すること)。**

- **テキスト (タブ区切り)** year.txt
- CSV **(カンマ区切り)** year.csv

ファイルの種類は [ファイル]→[名前をつけて保存] を指定した後，ダイアログの「ファイルの種類」で指定する。左下図は CSV **形式を指定するところ：**



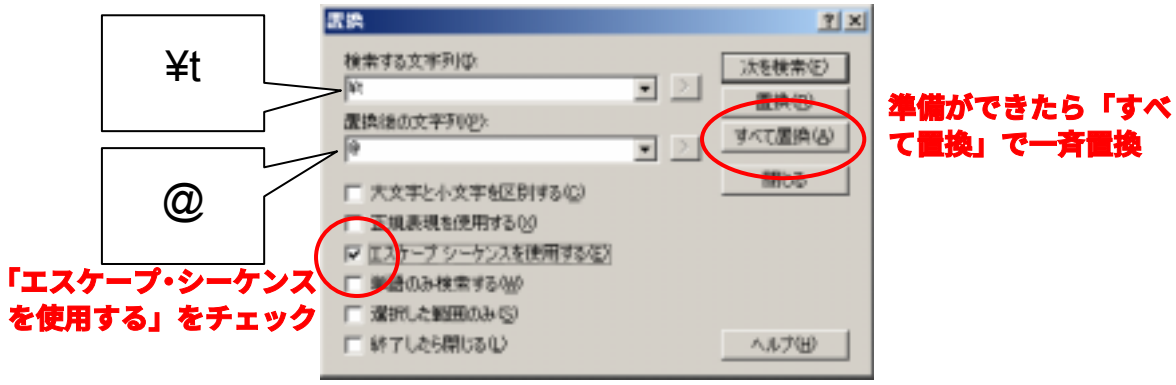
実習３：実習２で作成したテキスト形式のデータ year.txt **と** year.csv **をそれぞれ** EmEditor **で開き，変換されたデータの構造を確認しなさい。その際，前回資料 §** 4 **を参考に，タブなどが表示されるよう設定を変更してみなさい。なお，**CSV **形式のファイルは大学** PC **では** Excel **に関連付けられているので，**EmEditor **で開くには，**(1) Emeditor **を起動し [ファイル] → [開く] で「ファイルの種類」を「全てのファイル」としてファイルを指定する (2) No3 フォルダを「マイコンピュータ」で開き，**year.csv **アイコンを右クリックして [送る] → [EmEditor] を指定する (右上図)，のいずれかの方法を使って開く (自宅** PC **の場合，後者はアイコンを右クリックして現れるメニューの「**EmEditor**」を選択すればＯＫ)。**

実習４：以下のデータを固定長データとして EmEditor **で作成し，**html.txt **というファイル名で保存しなさい。各フィールドの間は半角スペースを使って埋め，フィールド幅をそろえなさい。**

要素名	意味	終了タグ
h1～h6	見出し	要
p	段落	省略可
em	強調	要
strong	強い強調	要
blockquote	引用文	要

実習５：実習５で作成した html.txt **を** Excel **で固定長データとして開き，フィールドを** Excel **のセルで正しく分割して表示しなさい。表示を確認し，**html.xls **というファイル名をつけて** Excel **ブック形式で保存しなさい。**

実習６【応用】：実習２で作成した year.txt **を** EmEditor **で開き，タブを半角の@マークに変更し** year2.txt **というファイル名で保存しなさい (これは §** 2.4.**の区切り文字を自由に指定した形式にあたる)。次に，**year2.txt **を** Excel **で正しく開きなさい (区切り文字として@マークを指定すること)。なお，タブを@マークに変更させるための置換ダイアログは次ページ図の通り (タブなど特殊記号の指定方法，および正規表現については次回以降の授業で詳しく解説する)。**



3. より複雑なデータ構造の表現

- 以下の文書（中尾浩著『文科系のパソコン技術：ライティングシステム序説』中公新書より）はどのような構造をしているか考えてみよう。また，この文書をテキストデータとして表現する場合，文書構造をどのように表現したらよいか考えよう。

第２章　書く＝考えるためのパソコン

１　手書きの限界点

（前略）
「書く」とは頭脳行為である。しかし，肉体行為であることも忘れてはならない。ペンであろうとパソコンであろうと，執筆途中で思考の流れをできるだけ妨げない方法を選ぶことが大切なのである。ペンで書いているほうが思考の流れを妨げないのなら，パソコンなど使うべきではないが，パソコンも所詮は道具である。慣れればこちらのほうがはるかに手直しは楽である。日本人はどうも「原稿用紙に万年筆で文字を刻み込むように」といった，いわゆる根性論が好きだが，そんなところでは苦労しないでも，もっと時間と精力を費やす場面はあるはずだ。
梅棹忠夫氏は『知的生産の技術』の中で面白いことを言っている。

> さきに，タイプライターに対する美学的アプローチということをのべたが，もちろん，能率が問題にならないわけではない。手がきよりはやい，ということもたいせつだが，それとともに，タイプライターがきは，らくだ，ということがある。手がきというのは，じつは手で書いているのではなく，全身でかいているのである。身心ともにひじょうな緊張を必要とする。それにくらべると，タイプライターこそは，ほんとに指さきの労働だけで，はるかにらくなのだ。電動タイプライターとなると，いっそうつかれがすくなく，しかもうつくしい。（『知的生産の技術』岩波新書，125—126 ページ，圏点原著）

執筆という行為は確かに頭脳労働であるが，肉体労働的な面を軽視することは禁物である。梅棹氏が言っている「緊張」感は執筆行為における大敵なのだ。頭脳労働を円滑にするためにも肉体的な緊張は少ないほうがよい。
手書きは「手でかいているのではなく，全身でかいている」「身心ともにひじょうな緊張を必要とする」とはうまく言ったものである。梅棹氏が書いている「タイプライター」を「パソコン」や「ワープロ専用機」に置き換えれば，ほとんど私が言いたいことを尽くしている。

２　文書作成の六つの基本機能

六つの基本機能
単に文書を書くだけなら，実際問題としてワープロ専用機であってもパソコンでワープロソフトを使っていても，さほど違いはない。ワープロ専用機も文書作成に機能を絞った＜コンピュータ＞であることには違いはないからである。
ここで問題になるのは，パソコンであれワープロ専用機であれ，その機能をどこまで使いこなしているかである。たとえばあなたは以下の文書作成の六つの機能を使いこなしているだろうか。

1．削除（カット）
2．挿入
3．複写（コピー＆ペースト）
4．移動（カット＆ペースト）
5．他文書参照
6．単語登録
（後略）

担当：千葉 庄寿（しょうじゅ） (CHIBA Shoju)　　e-mail: schiba@reitaku-u.ac.jp

- 著作物をはじめとする「自然言語データ」は意外に複雑な構造をしている。そのため，文書の内容をできる限り厳密に表現しようとすると，テキストデータの文書構造は当然複雑になる。電子化の目的をはっきりさせれば，もちろん工夫次第で簡易データベース (リスト) 的な表現を用いることも可能である。

4. メタデータを利用した構造の表示

- 自然言語のように，簡易データベースのような単純な構造で収まらない複雑なデータの構造をテキストデータとして表現するためには，文書構造の表現方法を工夫する必要がある。ここではウェブページなどで使われる HTML と，より汎用的なデータ構造の記述に用いられる XML を紹介する。
- HTML，XML とも，非営利の国際研究組織 WWW **コンソーシアム** (World Wide Web Consortium，W3C) により策定されている規格である。HTML と XML の仕様書 (specification) [1] は W3C のホームページ (`http://www.w3.org`) で読むことができる。

4.1. HTML (Hypertext Markup Language)

- HTML は，コンピュータや利用環境に依存せずに情報を交換することを目的として考案された，一種のプログラミング言語である。テキストに「**タグ付け**」markup をおこなうことで文書の構造を表現する。
  - **hypertext** (ハイパーテキスト)：目的の情報へジャンプするリンク (**ハイパーリンク** hyperlink) を伴うテキスト。
  - **markup** (マークアップ)：必要な情報を「**タグ**」tag としてテキストに追加すること。
  - HTML = 「**ハイパーリンクを作成するためのタグ付けをする言語**」
- HTML では文書構造のタグづけの方法や内容が決まっている。
  - HTML のタグは角括弧 (< >) で囲んであらわす。
  - 余分な改行や半角スペース，タブは無視される (半角スペース 1 文字分と解釈される)。
  - タグの内側に書く内容には，**要素名 (=文書構造を表す名称)**，**属性名** attribute とその **値** value の 3 種類がある。要素名，属性名は予め決められたものしか利用できない (以下では属性名とその値については時間の関係で扱わない)。
    - ✓ タグは，**開始タグ**と**終了タグ**がある。
    - ✓ 開始タグから終了タグまでをひとまとめにして「**要素**」element，また開始タグと終了タグに囲まれた部分を (その要素の)「**内容**」content という。以下の図 1 を参照。

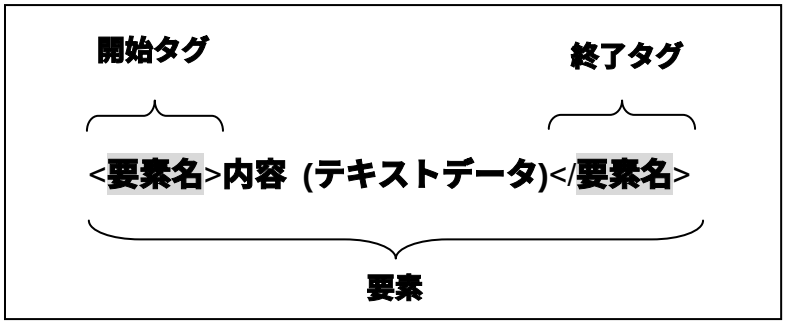


図 1：HTML のマークアップの基本構造

[1] **仕様書** specification とは，機械やプログラムなどについて，その内容や性能を記述した技術文書のこと。HTML の仕様書は HTML を作成する場合の原典であり，HTML 文書の記述方法が詳細に述べられている。XML の仕様書は，HTML に比べて技術的で難しい。XML in 10 points などの解説から入るとよい。

- ✓ 開始タグと終了タグは，一部の例外を除きペアで使われる。一部終了タグを両略できるものもある。
- ✓ タグは入れ子関係にできる (例えば，strong 要素を p 要素の中に書くことができる)。

例： この段落には強調が含まれています。

4.1. XML (Extensible Markup Language)

- HTML と同じ W3C により策定され，1997 年 12 月に 1.0 が正式に勧告された，非常に新しい文書構造の記述規格。
- 文書の交換を目的に 1970 年前後から開発が進められ， 1986 年に国際規格 ISO8879 となった SGML (Standard Generalized Markup Language) の改良版。
  - HTML は，SGML の文書構造の記述方法の応用例の 1 つ。
  - コンピュータ処理が難しい SGML の欠点を改良し，処理効率を向上させるための仕様の簡素化が行われている。
    - ✓ 終了タグの省略の禁止
    - ✓ 最上位要素の義務的配置　など
- XML によるマークアップの主な決まり
  - タグの内側に書く内容は，要素名，属性名とその値である (HTML と同じ)。ただし，**要素名，属性名は自由につけることも，特定の構造記述に合わせることもできる**。
  - 情報のまとまりをあらわす要素 element は開始タグ, 終了タグのペアで内容を囲んで表わすのは HTML と同じ。HTML (および HTML の元になっている SGML) では，一部タグは終了タグの省略が可能だったが，XML では省略は禁止。

    <title>フィンランド語文法のまとめ**</title>**

    - ✓ 終了タグがないときは,「空要素」であることを明示するタグ形式を使う。

    <xsl:apply-template />

  - 要素は入れ子にできるが，交差はできない (HTML と同じ)。
  - 最上位の要素はひとつだけ (SGML では制限がなかった)。

HTML, XML 文書内で使用できない文字：< > &, と 2 種類の引用符（' と "）は特殊。タグとして用いる場合以外は，それぞれ以下のように置き換えて使うことが必要になる。

< > & ' "

実習７： No3 **フォルダに入っている** nakao.txt (**テキスト文書**) **を開き，以下のようなタグを使い** HTML **のタグづけを行ってみよう。§**3 **のテキストを参照し，文書のどの部分をどうタグ付けするかは各自決めてよい。**

要素名	**意味**	**終了タグ**
h1〜h6	**大見出し〜小見出し**	**要**
p	**段落**	**省略可**
em	**強調**	**要**
blockquote	**引用文**	**要**

タグ付けが終わったら，[**ファイル**] → [**名前をつけて保存**] **でファイルを** nakao.html **というファイル名** (HTML **文書の拡張子は** html) **で保存しなさい。ファイルをブラウザで表示させて印刷し，名前と学籍番号を記入して提出すること。提出は第** 3 **回授業か第** 4 **回授業開始時とする。**

担当：千葉庄寿 (CHIBA Shoju) e-mail: schiba@reitaku-u.ac.jp