文書処理

平成 13 年 6 月 13 日

1 XML

XML(Extensible Markup Language) はマークアップ言語の枠組みを決める言語です。

マークアップ言語は，文章の内容とは別に，特定の文字列に意味を持たせてコンピュータに処理させようとする言語です．例えば HTML のタグは，表示されずにブラウザに指示を与えています．

XML は HTML とはレベルの違う言語で，処理の内容ではなく，マークアップの付け方についての約束です．具体的なマークアップは，使う人が自由に決められます．例えば，名簿を書く人は次のようなマークアップを使うかもしれません．

```
<名簿>
<氏名>
<姓>藤沢</姓>
<名>太郎</名>
</氏名>
<属性>
 <年齢>18 才</年齢>
 <職業>学生</職業>
 <趣味>読書</趣味>
</属性>
</名簿>

<名簿>
<氏名>
<姓>慶應</姓>
<名>花子</名>
</氏名>
<属性>
 <年齢>19 才</年齢>
 <職業>学生</職業>
 <趣味>テニス</趣味>
</属性>
</名簿>
```

ここに出てくる＜名簿＞や＜氏名＞というタグがどういう意味を持つかは，この文書を書く人と読む人の間で取り決めておかなくてはなりません．

XML は HTML とは違い，必ずしも表示を目的としたのものではありません．例えば，あるソフトから別のソフトへデータを転送するときに，XML 形式のデータを使うことがあります．

XML のマークアップの付け方に従い，具体的なタグの名前と意味を決めたもの，およびそれを処理するためのソフトウェアをひっくるめて，XML アプリケーションといいます．例えば，文書処理用の XML アプリケーションとしては，次のようなものがあります．

XHTML HTML のマークアップを XML に従うように改造したもの.

SmartDoc HTML と LaTeX の両方を生成できるようなマークアップ言語.

今週は SmartDoc を使った文書処理について学習します.

2 マークアップ言語と WYSIWYG

MicrosoftWord などのワードプロセッサの多くは WYSIWYG(What You See Is What You Get) 方式と呼ばれ，文書の見た目が表示された状態で編集を行い，文書を作成します．それに対して SmartDoc などのマークアップ言語は，文章の中にさまざまなマークアップを埋め込んだ文書ファイルを作り，それに対してコンパイルという作業を行うことで実際の文書を作成します．手間がかかるように見えますが，同じ文書ファイルからウェブ用，印刷用など用途に応じた文書を作ることができます．ただし，そのためには見た目を指定するマークアップではなく，文章の構造や意味を表すマークアップを付けておかなければなりません．そうしておけば，表示デバイスの能力も考慮して，最適な表示イメージを作ることができます.

SmartDoc 文書をコンパイルすると，HTML4, HTML3, LaTeX などの文書ファイルが生成できます．LaTeX もマークアップ言語の一種で，論文や市販の書籍の作成に使われており，高品質の印刷 (特に数式) が可能です.

3 SmartDoc を使ってみる

SmartDoc で文章を書き始める前に，どの文字コードを使うかを決めなければいけません．XML では，どんなコンピュータでも文字化けせずに正しく処理できるように，UTF-8, UTF-16 以外の文字コードを使う時は，必ず先頭で宣言する必要があります．また，生成するファイルの文字コードも決めておかなければなりません．ここでは，すべて iso-2022-jp を使うことにします.

それでは，`first.sdoc` というファイルを作り，以下の内容を書き込んで ください.

リスト 1: first.sdoc

```
<?xml version='1.0' encoding="iso-2022-jp"?>

<doc xml:lang="ja">
<head>
</head>
<body>
```

XML のマークアップの付け方

HTML と同様に開始タグと終了タグで要素を表します.
HTML と違い，開始タグだけの要素はありませんし，終了タグを省略することもできません．
内容が空っぽの要素は，次のどちらかの書き方をします.

- <xxx></xxx>
- <xxx />

これがはじめての SmartDoc で作成した文章です.
</body>
</doc>

sdoc ファイルから，実際に表示に使えるファイルを生成するためには，sdoc コマンドを使用します．その前に，あらかじめ設定ファイルを作っておきます.

- ~/.defaults.properties というファイルを作り，下のように書いておきます.
- 設定を文書によって変えたいときは，SmartDoc.properties というファイルに同様の設定を書き，sdoc コマンドを実行するときのカレントディレクトリ (通常は sdoc ファイルがあるディレクトリ) に置いておきます.

リスト 2: ~/.defaults.properties または SmartDoc.properties

```
html4.encoding=iso-2022-jp
html3.encoding=iso-2022-jp
latex2e.encoding=iso-2022-jp
latex2e.option=a4j
latex2e.image=DVI2PSLaTeX2eImageHandler
latex2e.box=AscmacLaTeX2eBoxHandler
```

3.1 ブラウザで見る

HTML4 形式のファイルを作るには次のようにします.

```
% sdoc -format:html4 first.sdoc
```

すると，first.html というファイルが作成されます．なお，これはブラウザによっては正しく表示できないので，そのようなブラウザも考慮する場合は HTML3 を使います.

Emacs での文字コードの変更

emacs で現在編集中のファイルの文字コードを変更するためのコマンドは，C-x RET f ([Mule] → [Set Coding System] → [Buffer File]) です．コマンド実行後Coding system for visted file(default, nil):と表示されたら，文字コード名を入力します．モードラインの左端を見て，

```
[あ]--J:
```

J となっていれば iso-2022-jp です．E の場合は euc-jp，S が sjis です.

```
% sdoc -format:html3 first.sdoc
```

作成されるファイルは，HTML4 の場合と同じ`first.html` です.

3.2 印刷する

印刷するときには LaTeX ファイルを生成します.

```
% sdoc -format:latex2e first.sdoc
```

すると，`first.tex` というファイルが作成されます．LaTeX は，また別のマークアップ言語ですので，印刷するためにはさらに platex コマンドで処理が必要となります.

```
% platex first.tex
```

すると，`first.dvi` というファイルが作成されます．これをプリンタに出力するには，

```
% dvi2ps first.dvi | lpr -P プリンタ名
```

とします．プリンタ名は，次の表を見てください.
1 年間それぞれ 1 人あたりプリンタの制限枚数は 500 枚となっています．500 枚をこすと 1 枚 5 円取られます．無駄な印刷を行わないよう心掛けましょう.

3.3 練習問題

実際にプリンタで印刷してみましょう．一度に全員が出そうとすると混乱するので，先生の指示にしたがってください.

4 基本的な機能

4.1 ヘッダ

今度はいろいろな機能を使って書いてみましょう．head 要素には，このファイルについての情報を書きます.

表 1: プリンターの位置

プリンター名	部屋
nps0	o 2 階コピー室
nps1	o 17
nps2	ι 18
nps3	ε 17
nps4	κ 18
nps5	λ 18
nps6	ε 2 階コピー室
nps7	ε 2 階コピー室
nps8	o 2 階コピー室
nps10	λ 11
nps11	メディアセンター 1 階
nps12	新オープンエリア
nps13	メディアセンター 1 階
nps14	メディアセンター 1 階

プリンターに出力せずにどのように印刷されるかをチェックする

platex コマンドを実行した際に作成される dvi ファイルを xdvi コマンドを使用して表示することができます.

```
% xdvi first.dvi
```

また pdf 形式のファイルを作成して見ることもできます. pdf 形式は Adobe 社が規定したデータ形式の一種で，UNIX, Windows, Macintosh 用に表示ソフトが無料で配布されています. dvi2ps コマンドの結果をファイルに保存し，ps2pdf コマンドを使用して pdf ファイルを作成します. acroread コマンドを使用して pdf ファイルを表示します.

```
% dvi2ps first.dvi >first.ps
% ps2pdf first.ps
% acroread frist.pdf
```

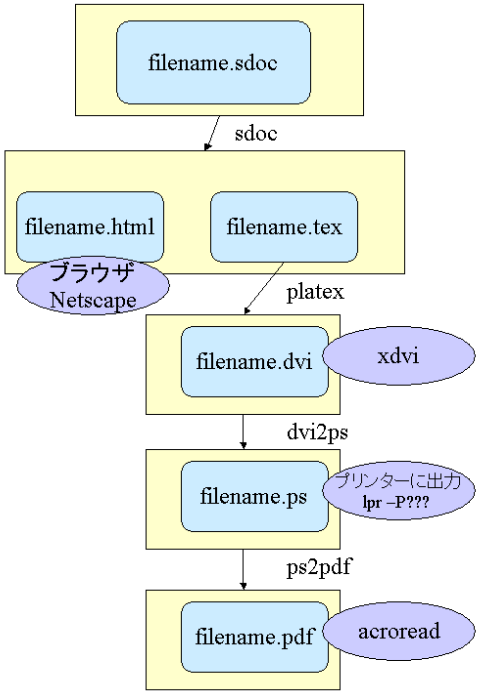

図 1: 処理の流れ

title 文書のタイトル

author 著者

date 文書の作成日

hp 著者のホームページ URI

email 著者の メールアドレス

リスト 3: 文書の例

```
<?xml version='1.0' encoding="iso-2022-jp"?>

<doc xml:lang="ja">
<head>
<title>SmartDoc の文章</title>
  <author>藤沢　太郎</author>
  <hp>http://www.sfc.keio.ac.jp/</hp>
  <date>平成 13 年 7 月 7 日</date>
</head>
<body>
これは SmartDoc で作成した文章です.
</body>
</doc>
```

文書が作成された時間 (sdoc を使ってこの HTML ファイルが作成された時間) は

```
<time />
```

で表示されます．date 要素で使用すると便利です．

4.2 段落

p 要素を使う方法と，段落と段落の間に空白行を入れる方法があります．

リスト 4: 段落の例

```
<p>１段落目です．p タグを使って段落を囲みます．とりあえず少し長く書いて行が変わっている方が見やすいでしょう．そろそろ行が変わったかな？</p>
<p>２段落目です．新しい段落となります</p>

３段落目です．p タグを使わず次の段落との間に１行間を空けます．空けることで段落変えとして表示されます．どうでしょうか？

これが４段落目となります．
```

１段落目です．p タグを使って段落を囲みます．とりあえず少し長く書いて行が変わっている方が見やすいでしょう．そろそろ行が変わったかな？
２段落目です．新しい段落となります．
３段落目です．p タグを使わず次の段落との間に１行間を空けます．空けることで段落変えとして表示されます．どうでしょうか？
これが４段落目となります．

4.3 章立て

長い文章を書くときには章や節を作り，それぞれタイトルをつけます．SmartDoc では次の 5 種類があります．

- part
- chapter
- section
- subsection
- subsubsection

上から順に大きなまとまりを表しています．使用方法は，このテキストを例にとると

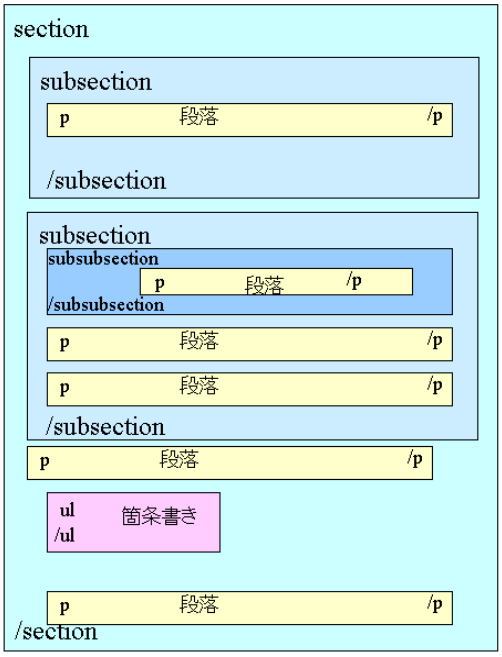

図 2: 入れ子構造

```
<section title="XML">
XML(Extensible Markup Language) は....
</section>
....
<section title="SmartDoc を使ってみる">
SmartDoc で文章を書き始める前に, .....
<subsection title="ブラウザで見る">
HTML4 形式の....
</subsection>
.
.
.
</section>
```

このように<section>から</section>までがひとまとまりとなり，title 属性でタイトルを記入します．大きなまとまり (section) の中で小さなまとまり (subsection) がはじまり終了する構造を入れ子構造と呼びます．SmartDoc で一番多い失敗は，この入れ子構造が間違っていることですから，気をつけてください．

4.4 箇条書き

HTML と同様に ul,ol,dl 要素を使います．

4.5 表の書き方

table 要素が表全体を表し，thead 要素は見出しの行を，tbody 要素は表の本体を表します．
項目は，HTML と同様に th,tr,td 要素を使う方法と，コンマと改行で区切る方法があります．
以下がその実行例と書き方になります．

行1	行2
1	2
3	4

リスト 5: 表の書き方 (tr, td 要素を使う方法)

```
<table>
<thead>
<tr><td>行1</td><td>行2</td></tr>
</thead>
<tbody>
<tr><td>1</td><td>2</td></tr>
<tr><td>3</td><td>4</td></tr>
</tbody>
</table>
```

リスト 6: 表の書き方 (コンマで項目を区切る方法)

```
<table>
<thead>
行1, 行2
</thead>
<tbody>
1,2
3,4
</tbody>
</table>
```

コンマでデータを区切る方式を，CSV 形式と呼びます．SmartDoc では CSV 形式のファイルを読み込むこともできます．

リスト 7: CSV ファイルの例 measure.csv(身長と体重)

```
160,55
180,78
168,60
174,80
```

リスト 8: CSV ファイルの読み込み

図 3: 四角形

```
<table>
<thead>
身長, 体重
</thead>
<tbody src="measure.csv" />

</table>
```

measure.csv というファイルの内容を読み込み, 表を作成しています.

上の例では <tbody> タグを閉じる </tbody> がありません. その際には <tbody> の最後に" /" をつけます.

身長	体重
160	55
180	78
168	60
174	80

4.6 画像

```
<figure src="img/square.gif,img/square.ps" title="四角形" style="width:2cm"/>
```

画像は別のファイルに用意しておき, figure 要素で読み込みます.

画像ファイル名を src 属性で指定します. 画像のタイトルを title 属性で指定します. HTML ファイルに出力する場合は gif と jpeg, LaTeX に貼り込む場合は eps ファイルが使えます.

gif, jpeg(HTML でのみ使用可能) と eps ファイル (LaTeX でのみ使用可能) の両方が用意されている場合には, ファイル名をコンマで区切って複数指定します.

LaTeX に図を貼り込む場合は, `eclepsf.sty` ファイルが必要です. sdoc コマンドを実行するコンピュータにこのファイルがインストールされていないと, エラーになりますから,

http://www.sfc.keio.ac.jp/~hattori/ipl/info-2001-4/text-a/10/eclepsf.sty

からコピーして, sdoc ファイルと同じディレクトリに置いてください.

4.7 リンクの貼り方

a 要素を使用し，href 属性でリンク先の URI を記述します.

```
<a href="http://www.sfc.keio.ac.jp/~hattori/ipl/info-2001-4/">情報処理のページ</a>
```

情報処理のページ[1]

4.8 参考文献

bibliography 要素は参考文献を記述するためのものです．それぞれの文献を book 要素で表します．その中の author, title, publisher, year 要素は著者、タイトル、出版社、出版年を表します．book 要素の id 属性は，引用する際に a 要素で参照することができます．LaTeX では自動的に参考文献に番号が振られ，引用個所にはこの番号が表示されます．(現在，HTML で引用はうまく動作しません)

```
<a href="#sfcguide">SFC ガイド</a>からの引用です.

<bibliography>
<book id="cnsguide">
  <author>SFC メディアセンター</author>
  <title>SFC-CNS ガイド</title>
  <publisher></publisher>
  <year>2001</year>
  </book>
<book id="sfcguide">
  <author></author>
  <title>SFC ガイド</title>
  <publisher></publisher>
  <year>2001</year>
  </book>
  </bibliography>
```

[1, SFC ガイド] からの引用です.

5 エラーの読み方

sdoc コマンドの実行に失敗した場合，エラーメッセージが表示されます．このエラーメッセージを読むことで間違いの個所を探すことができます.

リスト 9: エラーメッセージの例 1

```
10.sdoc:269:10 [Fatal Error] The element type "subsection" must be terminated by the matching end-tag "</sub
Error: Stopping after fatal error: The element type "subsection" must be terminated by the matching end-tag
```

[1] http://www.sfc.keio.ac.jp/~hattori/ipl/info-2001-4/

エラーメッセージの 1 行目の 10.sdoc は sdoc コマンドが実行されたファイル名を表示しています．この次に来る数字 269 が間違いが存在する行の番号になります．その後には，subsection 要素は </subsection> というタグで閉じる必要があると書かれています．/subsection が存在するか，入れ子構造になっているかがこの間違いを修正するためにチェックするべきポイントとなります．

リスト 10: エラーメッセージの例 2

```
unknown:0: [Fatal Error] File "file:/a/fs0502a/t01000tf/10.sdoc" not found.
Error: Stopping after fatal error: File "file:/a/fs0502a/t01000tf/10.sdoc" not found.
```

エラーメッセージの最初に unknown とあります．上記のエラーの場合 /home/t01000tf/ に 10.sdoc ファイルが存在していないようです．ファイルが sdoc コマンドを実行したディレクトリに存在しているかどうかチェックしましょう．

5.1 練習問題

http://www.sfc.keio.ac.jp/~hattori/ipl/info-2001-4/text-a/10/error.sdoc ファイルは間違いを含む sdoc ファイルです．sdoc コマンドを実行して，間違いを発見し修正してみましょう．

参考文献

[1] ．*SFC* ガイド．2001.

[2] SFC メディアセンター．*SFC-CNS* ガイド．2001.