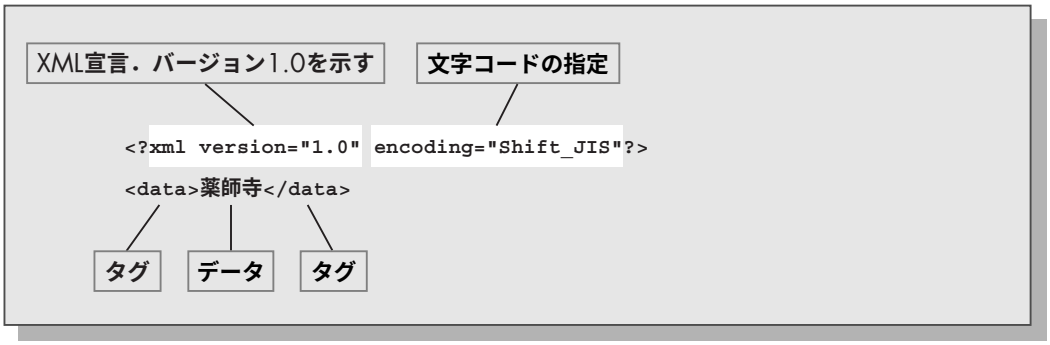
XML文書のルール

リストどおりに入力することができ，表示もできたところで，**リスト**1-1の文書を詳しく見ていきましょう．この文書は，XML文書のルールに従って書かれています．

リスト1-1　サンプルXML文書．(CD-ROM⇒1stDay/list1.xml)



❶ XML宣言

xml version="1.0"の部分は，XML宣言と呼びます．

XMLは，Webの仕様を策定している機関である，W3C(World Wide Web Consortium)の定めた仕様です．versionに指定した数字は，XMLの仕様のバージョンを表します．xml version="1.0"と書けば，この文書が，XML1.0の仕様に則った文書であることを表しています．必ずしも書かなければならない仕様ではないのですが，XML文書であることを宣言し，文字コードを明らかにする上でも書いておいたほうがよいでしょう．

❷ 文字コードの指定

先にも書いたように，encoding="Shift_JIS"の部分は，このXML文書が，何という文字コードで作成されているかを表しています．**リスト**1-1は，Shift_JIS(シフトJIS)コードで作成されていることがわかります．

では，他の文字コードの場合は，どう書けばよいでしょうか．

XMLで標準的に使われる文字コードは，Unicode(ユニコード)です．Unicodeの場合は，何も指定する必要はありません．つまり，

```
<?xml version="1.0"?>
<data>薬師寺</data>
```

のように書きます．

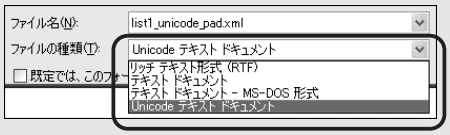
ただし，このファイルは，Unicodeで保存する必要があります．

例えばWindowsのメモ帳なら[Unicode](**図**1-14)を，ワードパッドなら[Unicodeテキストドキュメント](**図**1-15)をファイル保存時に指定します．

メモ帳

メモ帳で，文字コードを何も指定しないか encoding="utf-16"**と書いて，**[Unicode]**を指定して保存する．**

図1-14　メモ帳を使ってUnicodeで保存する場合．



ワードパッド

ワードパッドで，文字コードを何も指定しないか encoding="utf-16"**と書いて，**[Unicode**テキストドキュメント**]**を指定して保存する．**

図1-15　ワードパッドを使ってUnicodeで保存する場合．

UnicodeとXML

一口にUnicodeといっても，Unicodeには，UTF-16(16-bit UCS Transformation Format)**とUTF-8(8-bit UCS Transformation Format)という種類があります．**UTF-16**は16ビットの可変長マルチバイト，**UTF-8**は8ビットの可変長マルチバイトで文字を表現しています．メモ帳やワードパッド保存時に**[Unicode]**を指定すると**utf-16**で保存されますから注意しましょう．**

XML**を解釈して処理するソフトウェア・モジュールであるXML**プロセッサでは，**UFT-16**とUTF-8をサポートするよう仕様で定められています．**

XMLerのエディタ

既存のXML文書を利用したり加工したいとき，テキスト・エディタで開くと文字が化けて見えることがあります．Unicodeで作成された文書をシフトJISのみ対応のテキスト・エディタで開くと，文字化けしてしまいますから，シフトJISだけでなくUnicodeも扱えるテキスト・エディタを使うべきです．

❸ 要素，タグ

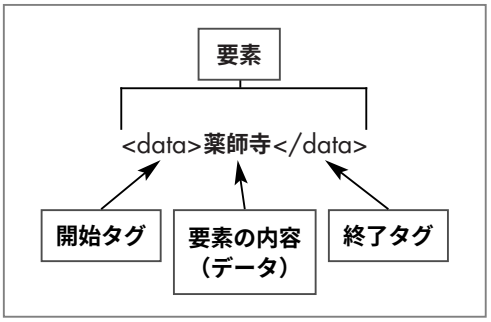


図1-16　XMLの要素とタグ．

リスト1-1のXML文書の中で，「薬師寺」の部分がデータだと書きました．では，<data>薬師寺</data>は，まとめて何と呼ぶのでしょうか?

XMLを扱う上では，<data>薬師寺</data>を「要素」と呼びます．これはよく使う言葉なので，必ず覚えてください．「薬師寺」の部分は，正確には「要素の内容」と呼びます．<data>や</data>は，タグと呼びます．<data>は開始タグ，</data>は終了タグです．dataを要素名（タグ名）といいます（**図**1-16）．

要素名として使えない文字

リスト1-1では，「data」という要素名（タグ名とも呼ぶ）を使っていますが，これは，必ずしも「data」でなければならないわけではありません．XMLでは，自由にタグ名を決めることができます．XMLでは，タグに日本語が使えます．

「data」というのは筆者が付けたタグ名ですから，他のタグ名でもかまいません．「名前」でも，「寺」でも，「address」でも「住所」でも何でもかまいません．

しかしXMLでは，要素名として使えない文字があります．今すぐ覚える必要はありませんが，XMLのタグ名には使用できない文字があるということだけ理解しておきましょう．

表1-2　XMLのタグ名に使えない文字．

一文字目に数字が入っているタグ	**［例］ <1号> <1月>など**
タグにスペースが含まれているもの	**［例］ <a b> <名 前>など**
一文字目にハイフンの入ったタグ	**［例］ <-ab> <-名前>など**
予約語であるxmlが使用されているタグ	**［例］ <xmldoc> <xmldata>など**
半角カナのタグ	**［例］ <ﾃﾞｰﾀ> <ﾙｰﾄ>など**

大文字と小文字の区別

XMLでは，大文字と小文字は厳密に区別されます．<data>というタグは，<Data>や<DATA>とは異なります．**リスト**1-1を，例えば右の**表**1-3ように書くのは誤りです．

表1-3　大文字と小文字の区別．

まちがったタグ	正しいタグ
<Data>**薬師寺**</data> →	<Data>**薬師寺**</Data>
<DATA>**薬師寺**</data> →	<DATA>**薬師寺**</DATA>
<data>**薬師寺**</Data> →	<data>**薬師寺**</data>
<data>**薬師寺**</DATA> →	<data>**薬師寺**</data>

<data>と</data>は，開始タグと終了タグで一セットです．開始タグと終了タグのタグ名が異なってはいけません．うっかりまちがえると，**図**1-10(p.15)のようなエラーになります．

❹ ルート要素

リスト1-1のXML文書には，「薬師寺」という一個だけのデータがありました．では，このXML文書にデータを増やしてみましょう．

例えば，**リスト**1-2のように，「山寺」というデータを追加してみます．

リスト1-2　「山寺」というデータを追加したXML文書．しかし，この文書は誤りである．
(CD-ROM⇒1stDay/list2._error.xml)

エラー

```
<?xml version="1.0" encoding="Shift_JIS"?>
<data>薬師寺</data>
<data>山寺</data>
```

一見，このXML文書は，開始タグと終了タグは一セットになっていて，正しいように見えます．しかし，XMLの決まりに従っていない，まちがった文書です．IE6で表示させると，**図**1-17のようなエラー・メッセージが表示されます．

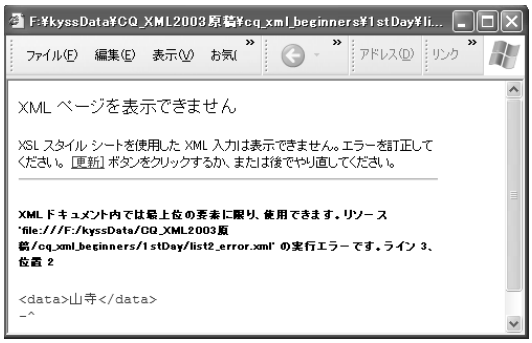


図1-17　リスト1-2**をIE6で表示したときのエラー・メッセージ．**

リスト1-2のどこがまちがっているのでしょうか？ XMLには，「ただ一つのルート要素がある」という決まりがあります．ルート要素とは，ルート(root＝根っこ)というように，XML文書のいちばん根っこにあたる要素です．**リスト**1-2には，その根っこがありません．**図**1-18のように，根っこがなければXMLの木は育ちません．

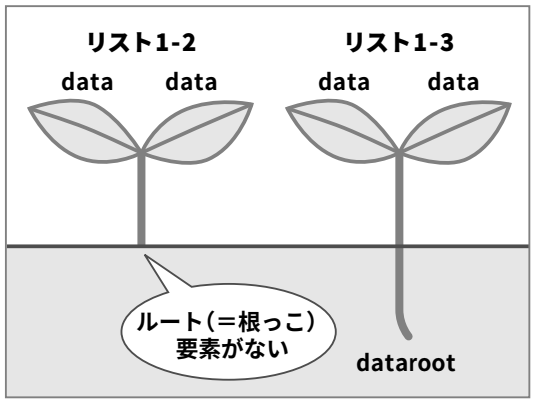


図1-18　リスト1-2**にはルート要素がない．**

リスト1-2に，ルート要素を追加してみましょう．ここでは仮に，ルート要素のタグ名を，「dataroot」としておきます．**リスト**1-3のように追加してください．

二個の「data」要素が，ルート要素「dataroot」から生えていることをわかりやすくするために，<data>薬師寺</data> と <data>山寺</data> は，［TAB］キーでインデント(字下げ)しておきましょう．

リスト1-3　リスト1-2に，ルート要素<dataroot>を追加したXML文書．(CD-ROM⇒1stDay/list3.xml)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<dataroot>
	<data>薬師寺</data>
	<data>山寺</data>
</dataroot>
```

見やすくするためにタブを使ってインデント(字下げ)している．XML文書の機能に影響はない．

リスト1-3をIE6で表示すると，**図**1-19のように，きちんと表示されました．これは正しいXML文書です．

IE6で表示した状態で，＜dataroot＞の横の－(マイナス記号)をクリックしてみてください．**図**1-20のように，一度閉じられ，再びクリックすると，ルート要素から生えている，二個の<data>要素が現れます．

XML文書には，一個だけのルート要素が必要だということを，覚えておきましょう．

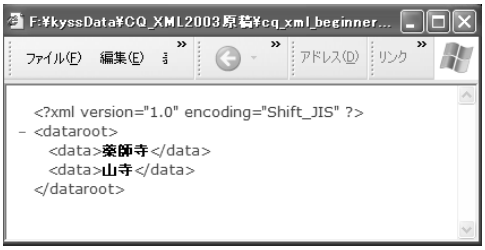


図1-19　リスト1-3をIE6で表示．正しく表示されている．

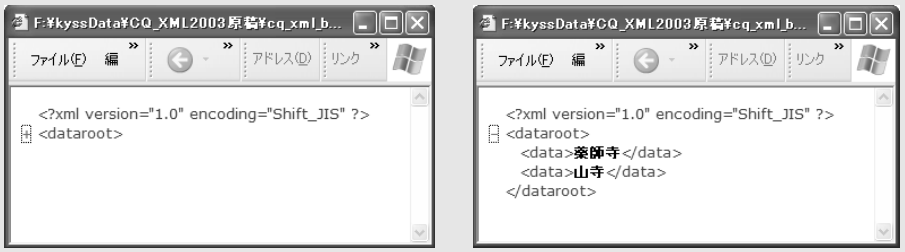


図1-20　リスト1-3をIE6で表示．＜dataroot＞の横の－をクリックしたところ(左図)，再度＋をクリックしたところ(右図)．

図1-20を見ると，エクスプローラなどでフォルダやファイルを表示したときの，ツリー表示の画面に似ていますよね．そうなんです，XML文書は，ツリー構造(木構造)になっているのです．ツリーだから，根っこがなければ枯れてしまうんですよね．ただし，根っこは必ず一つだけです．

❺ 親子関係

XMLツリーには，親子関係があります．「dataroot」から「data」が生えているのですから，「dataroot」から見れば「data」は子供です．逆に「data」から「dataroot」を見れば親にあたります．

二個の「data」は，同じ「dataroot」という親から生えているので，兄弟です．先に生まれた（先にこの世に出現した）「data」が兄で，後から生まれた（後に出現した）「data」が弟です．**リスト**1-3では，<data>薬師寺</data>という要素が「兄」で，<data>山寺</data>が「弟」になります．兄から弟を見れば「後の兄弟」です．逆に，弟から兄を見れば「先の兄弟」です．この「後」や「先」という順序は，XML文書を処理するうえでは非常に重要になってきます．

このような親子兄弟の関係は，基本を学ぶ段階で，しっかりと理解しておきたいものです．**リスト**1-3を図式化すると，**図**1-21のようになります．XMLには，階層構造があることがわかります．階層構造は，XMLの大きな特長の一つです．

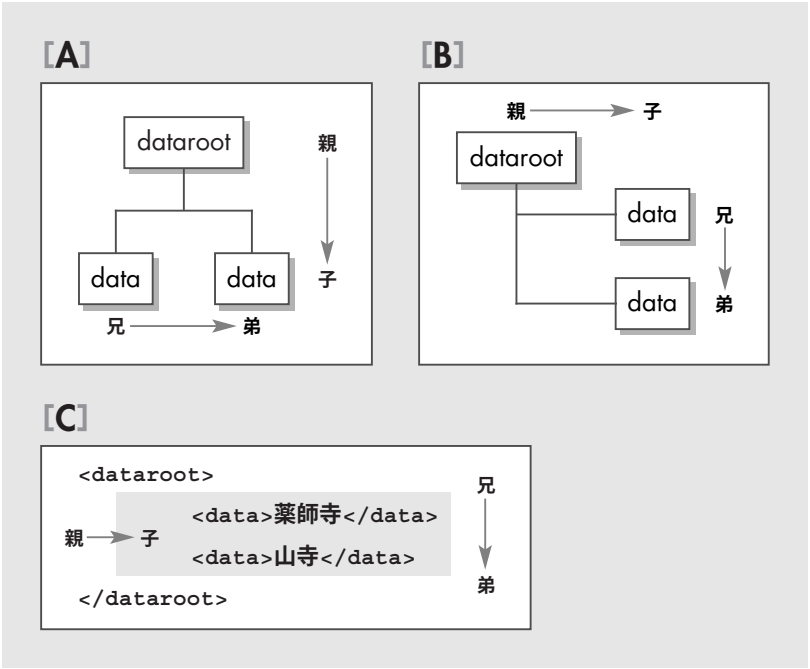


図1-21　リスト1-3**の構造を異なった表現で図式化したもの．**

［A］も［B］も［C］も，図の表現方法が異なるだけで，同じ意味を表しています．

XMLの構造を説明するために，よく使われるのは，［A］のような表しかたです（繰り返される要素は表現しないこともある）．［B］も比較的よく使われます．筆者も，書籍では［A］の表現で説明することもありますが，いますぐ実務でXMLを知る必要がある場合は，最初から［C］の方法で考えていくことをお勧めします．

［A］［B］は，XMLのタグ名や階層構造を決めてから，それに合ったデータを作成する場合には，わかりやすい方法です．しかし，既存のデータが存在し，これをXML化していくと

いう場合には，データを入れ込みながら考えると複雑化してしまいます．［A］［B］は「データありき」の打ち合わせには，使いにくいのです．また，［A］［B］で考えた構造で，いざXML文書のソースを書こうとすると手の止まる人がいます．最初から，［C］のように，階層構造と実データを絡めて理解するクセをつけておくことが，実務，特にXML文書を扱うプログラミングでは必要です．

［C］の方法で理解しておけば，決めなければならないタグ名が二十個程度で，構造も祖父母親子兄弟といった三世代程度のシンプルなものであれば，パソコンでソースや一部のデータを入力しながら，どのようなタグ名や構造にするかを話し合うことができますし，決定したソースにデータを追加していき，そのまま実務に利用できるというメリットがあります．

XML文書では，親子兄弟関係に気を配り，要素は正しく入れ子になっていなければなりません．もし，**リスト**1-3を，**リスト**1-4のように書くと，**図**1-22のように，エラーになります．

リスト1-4　リスト1-3**で要素が正しく入れ子になっていない**XML**文書．**
(CD-ROM⇒1stDay/list4_error.xml)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<dataroot>
      <data>薬師寺</data></data>
      <data>山寺
</dataroot>
```

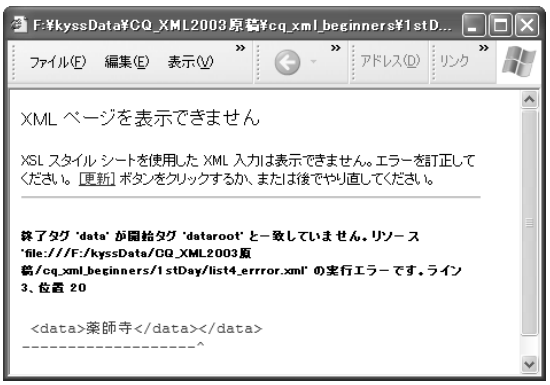


図1-22　リスト1-4**の表示結果．正しく入れ子になっていないので，エラーになる．**

❻ 属性

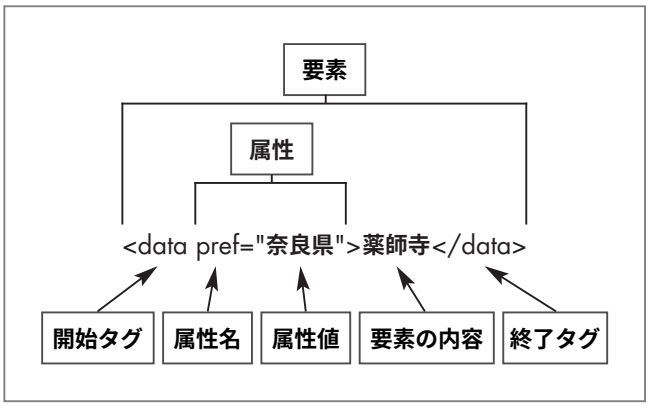
では，**リスト**1-3のデータに，場所を表すデータを追加したい場合は，どうすればよいでしょうか?「薬師寺」は「奈良県」，「山寺」は「香川県」という情報を持っているとします．

その場合は，**リスト**1-5のように，属性を追加する方法があります．これで，「薬師寺」というデータに「奈良県」，「山寺」というデータに「香川県」という付帯情報があることが一目でわかるXML文書ができました．皆さんも，テキスト・エディタで入力して，IE6で表示してみてください．

リスト1-5　リスト1-3**に，「**pref**」属性を追加したXML文書．**(CD-ROM⇒1stDay/list5.xml)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<dataroot>                                    属性(属性名＝"属性値")
      <data pref="奈良県">薬師寺</data>
      <data pref="香川県">山寺</data>
</dataroot>
```

このpref=""を属性(厳密には「要素に対する属性指定」)，""(ダブル・クォート)ではさまれた中のデータを属性値，prefを属性名と呼びます．



XMLには，属性値は，引用符(")で挟むという決まりがあります．

HTMLソースを書いたことのある人なら，リンクを表すといったコードでは，「a要素のhref属性」といった表現を聞き慣れているかもしれません．XMLでも同じ考えかたをします．ただし，HTMLの場合は，ネットサーフィンして行き当たったWebページのソースを片端から表示すればわかりますが，というように，属性値を引用符で挟んでいないコードがかなりあります．ブラウザでは問題なく表示されていますが，XMLの場合は，必ず引用符で挟まなければなりません．HTMLコードを書き慣れている人は，**図**1-23のようなエラーとなるXML文書を作成してしまわないように気をつけてください．**リスト**1-5はデータが二件だけのXML文書ですが，何百件以上のデータを扱う場合，一箇所引用符で挟み忘れると，そのまちがいを探して修正するのが手間になります．