

HTML と ASP を使って XML ファイルを生成

総合情報基盤センター技術専門職員 畑 篤
hata@cns.toyama-u.ac.jp

1. はじめに

インターネットの急速な発展に伴い、コンピュータを介して、組織や個人を結びつけるようになってきました。これらの情報は主に HTML 形式の情報です。しかし、HTML 形式では複雑なデータ構造が記述できないなどといった、表現力に限界があります。その点、XML 形式の情報では複雑なデータ構造を記述することが出来ます。

XML 形式情報の特徴は下記のとおりです。

1. 文書データに意味づけが出来る
2. 記述規則が比較的簡単
3. レイアウト用の定義は含まない
4. プラットフォーム独立している

利用方法として、Web サービスや定義ファイルとして利用されています。

XML 文書はわかり易い反面、記述するのは、結構面倒です。ここでは XML 文書を簡単に記述するために、ASP と HTML を利用して XML 文書を作成することを紹介します。

2. XML について

2-1 XML (eXtensible Markup Language)

HTML のようなシンプルなフォーマットで文章構造を記述でき、独自にタグを定義することができるマークアップ言語です。

2-2 XML の記述の決まり

- ① XML 文章であることを宣言する
- ② 文字コードを指定する
- ③ すべての要素には、終了タグを書く
- ④ すべての要素は正しく入れ子にする
- ⑤ XML 文章には、1つのルート要素を書く
- ⑥ 指定した文字コードで保存する
- ⑦ 大文字、小文字を区別する
- ⑧ 属性値は、二重引用符で挟む

2-3 XML 文書例

下記に XML の記述例を記載します。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<全体>
  <情報>
    <業種>学術・開発研究機関</業種>
    <企業名>日本原子力研究所</企業名>
    <郵便番号>277-0842</郵便番号>
    <所在地県>千葉県</所在地県>
    <住所>柏市末広町 14-1</住所>
    <資本金></資本金>
    <従業員数></従業員数>
    <URL>http://www.jaeri.go.jp</URL>
    <備考></備考>
  </情報>
  <情報>
    <業種>学術・開発研究機関</業種>
    <企業名>(独法)宇宙航空研究開発機構</企業名>
    <郵便番号>182-8522</郵便番号>
    <所在地県>東京都</所在地県>
    <住所>調布市深大寺東町 7-44-1</住所>
    <資本金>51250000</資本金>
    <従業員数>1772</従業員数>
    <URL>http://www.jaxa.jp/about/employ/recruit</URL>
    <備考></備考>
  </情報>
  <情報>
    <業種>学術・開発研究機関</業種>
    <企業名>(独法)産業技術研究所</企業名>
    <郵便番号>305-8561</郵便番号>
    <所在地県>茨城県</所在地県>
    <住所>つくば市東 1-1-1 つくば中央第一事業所</住所>
    <資本金></資本金>
    <従業員数></従業員数>
    <URL>http://www.aist.go.jp</URL>
    <備考></備考>
  </情報>
</全体>
```

2-4 XML の Web 表示

XML 文書を Web 画面で表示した場合は図 2-1 のように表示されます。



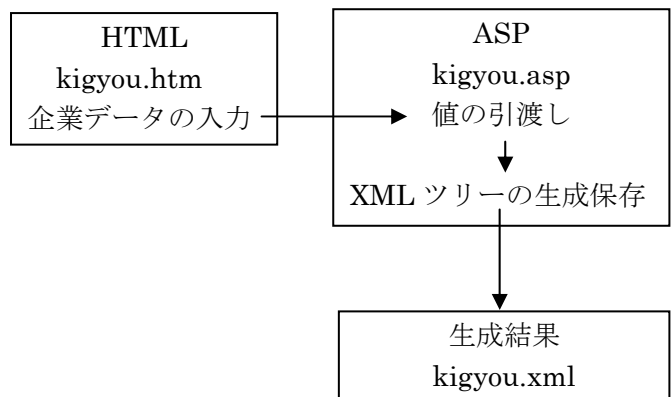
図 2-1 XML 文書を Web 表示例

3. XML ファイルの作成

3-1 XML ツリーの作成方法

XML 文書中の各ノードには、DOM (Document Object Model) を使ってアクセスすることが出来ます。DOM 機能を利用して、HTML から入力された任意データを XML ツリー形式として作成します。

生成された XML ツリーは、ASP(Active Server Pages) を使って保存します。



以下に HTML ファイルと、ASP ファイルを使って企業登録 XML ファイルの作成例を紹介します。言語として VBScript を利用しています。

3-2 HTML ファイルソース(kigyuu.htm)

```

<script type="text/VBScript">
<!--
sub window_onload()
  productData=""
  set xmldoc=createObject("MSXML2.DOMdocument")
  xmldoc.async=false
  fileValue=xmldoc.load("gyousyu.xml")
  if fileValue=true then
    set productNode=xmldoc.getElementsByTagName("業種")
    if productNode.length>0 then
      for i=0 to productNode.length-1
        productData=productData & "<option value='" & productNode(i).text & "'>" &
productNode(i).text & "</option>"
        next
      optionData="<option value='dummy'>***業種を選択/***/</option>" & productData
      selectData="<select name='Select0' onChange='dataCheck00'>" & optionData &
"</select>"
  
```

```

        document.all("selectBox").innerHTML=selectData
    else
        exit sub
    end if
else
    msgbox "業種ファイルがありません！"
    exit sub
end if
end sub

sub dataCheck0()
    kigyoy.Data0.value=kigyoy.Select0.options.value
    kigyoy.okButton.disabled=false
end sub

sub dataCheck()
    if kigyoy.Data0.value="dummy" then
        msgbox "業種を選択してください"
        exit sub
    end if

    if len(kigyoy.Data1.value)<=0 then
        msgbox "企業名を入力して下さい。"
        window.event.returnValue=false
        exit sub
    end if

    if len(kigyoy.Data2.value)<=0 then
        msgbox "郵便番号を入力して下さい。"
        window.event.returnValue=false
        exit sub
    end if

    . . . . .
    .
    kigyoy.action="kigyoy.asp"
end sub

-->

```



図 3-1



図 3-2 HTML での入力画面

```

</script>
</head>
<body>
<div>
<h1>求人登録</h1>
<form name="kigyoy" method="post" onsubmit="dataCheck()" >

```

```

<table border="1">
<tr><th>業種</th><td><span id="selectBox"></span><input style="display:none;" type="text"
size="20" name="Data0"></td></tr>
<tr><th>企業名</th><td><input type="text" name="Data1" size="40"
style="ime-mode:active;"></td></tr>
<tr><th>郵便番号</th><td><input type="text" name="Data2" size="10"
style="ime-mode:inactive;"></td></tr>
. . . . .
</table>
<div>
<input disabled="true" name="okButton" style="cursor:hand;" type="submit" value="OK">
</div>
</form>
</div>
</body>

```

3-3 HTML ファイルの各ルーチン説明

window_onload ルーチン

- XMLDOM オブジェクトへのアクセスを可能にする。
- gyosyu.xml (業種データファイル) を読み込み、業種ノードリスト (<業種>) を HTML 要素の「option」の値として格納する。
- gyosyu.xml が存在しない場合は、メッセージを表示する

dataCheck0 ルーチン

- 業種リストで選択した内容を kigyuu.Data0 に格納する。
- 「OK」ボタンが使用可能状態にする。

dataCheck ルーチン

- データ入力確認ルーチン。
- 必要項目が入力されている場合、kigyuu.asp を起動させる。
- 未入力の場合は、メッセージを表示する。

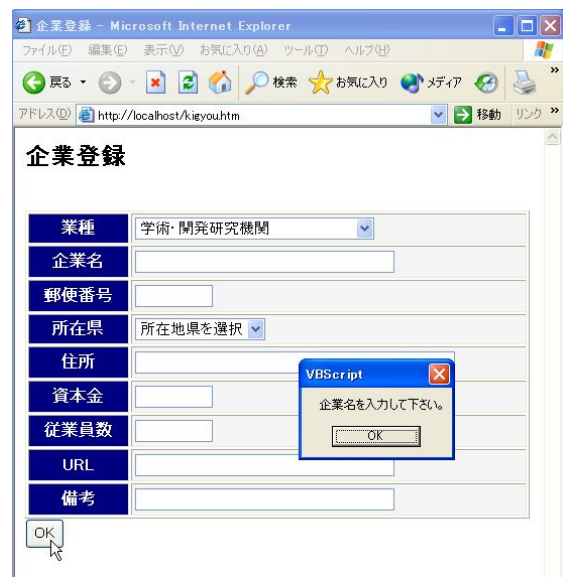


図 3-3 必要項目が入力されていない場合の表示

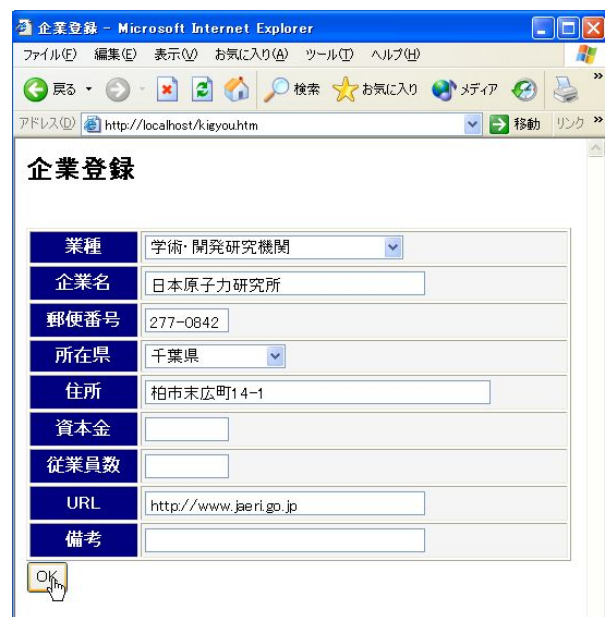


図 3-4 「OK」ボタンをクリックすると”kigyuu.asp”が起動

3-3 ASP ファイルソース(kyujin.asp)

```

<%@Language="VBScript"%>
<%
    filePath=Server.MapPath("./")
    for i = 0 to 8
        textData(i)=Request.Form("Data" & i)
    next
    table(0)= "業種"
    . . . . .
    table(8)= "備考"
    set xmlDoc=Server.createObject("MSXML2.DOMDocument")
    xmlDoc.async=false
    fileValue=xmlDoc.load(filePath & "/kigyou.xml")
    if fileValue=false then
        set mainElement=xmlDoc.createProcessingInstruction("xml","version='1.0'
encoding='Shift_JIS'")
        set rootElement=xmlDoc.createElement("全体")
        xmlDoc.appendChild(mainElement)
        xmlDoc.appendChild(rootElement)
        call makeXML()
    else
        call makeXML()
    end if
    xmlDoc.save(filePath & "/kigyou.xml")
    Response.Redirect("kigyou.htm")

sub makeXML()
    set infoElement=xmlDoc.createElement("情報")
    for i = 0 to 8
        set dataElement(i)=xmlDoc.createElement(table(i))
        set dataText(i)=xmlDoc.createTextNode(textData(i))
        dataElement(i).appendChild(dataText(i))
        infoElement.appendChild(dataElement(i))
    next
    if fileValue=false then
        rootElement.appendChild(infoElement)
    else
        xmlDoc.documentElement.appendChild(infoElement)
    end if
end sub
%>

```

3-4 ASP ファイルの各ルーチン説明

メインルーチン

- XMLDOM オブジェクトへのアクセスを可能にする。
- XML ファイルの有無を確認し、XML ファイルがあれば、makeXML サブルーチンを実行する。
- ファイルが無い場合は、XML の宣言、ルート要素の作成を行い、makeXML サブルーチンを実行する。

makeXML サブルーチン

- XML ファイルを保存する。
- HTML (入力画面) へ戻る。
- 要素「情報」の作成
- 生成する要素分のデータ項目とデータ内容を受け取り、要素名と内容を生成する。
- 各要素に子要素として各内容を付加する。
- 情報要素に各要素を子要素として付加させる。(階層構造の作成)
- 新規で作成する場合、全体要素に子要素として、情報要素を付加させる。

```
<?xml version="1.0" encoding="Shift_JIS"?>↓
<全体><情報><業種>学術・開発研究機関</業種><企業名>日本原子力研究所</企業名><郵便番号>277-0842</郵便番号>
[EOF]
```

図 3-5 作成された XML 文書



図 3-6 作成した XML 文書を Web 画面表示



図 3-7 Web 画面表示例

4. まとめ

HTML と ASP を利用することにより、Web 画面で入力した内容が、XML 形式のデータとして保存することが出来ます。また、保存した XML ファイルを使い、Web でのデータ検索等を行うことも出来ます。XML データの抽出や表示については XSL を利用します。

このように、HTML, XML, XSL, ASP を利用することで簡易な Web データベースを作成することが出来ます。

ASP を利用するにあたっては、IIS(Internet Information Server)機能が必要になります。

5. 参考図書

XML+XSLT 実用スーパーサンプル集

CQ 出版社