

図 1 (ファイル構成)

style.css	全体のデザインを設定する
index.html	トップページ
a.html	XML とは
xml-1.html	XML 文書の構成
xml-sengen.html	XML 宣言
xml-ver.html	バージョン宣言
xml-en.html	エンコーディング宣言
xml-stand.html	スタンドアロン宣言
dtd.html	DTD
dtd-kousei.html	DTD の構成
dtd-toi1.html	" 解答ページ(不正解)
dtd-toi2.html	" 解答ページ(正解)
dtd-kousei-2.html	要素型宣言
dtd-kousei-2-1.html	内容モデル
dtd-kousei-2-2.html	子要素の出現順序
dtd-kousei-2-3.html	子要素の出現回数
dtd-kousei-2-4.html	要素と文字データの混合
dtd-kousei-2-5.html	要素の優先順位
dtd-toi3.html	" 解答ページ(正解)
dtd-toi4.html	" 解答ページ(不正解)
dtd-kousei-3.html	属性リスト宣言
dtd-kousei-4.html	二つの実体宣言
xml-ins.html	XML インスタンス
xml-ins-1.html	要素
ins-toi1-1.html	" 解答ページ(正解)
ins-toi1-2.html	" 解答ページ(不正解)
xml-ins-2.html	属性
ins-toi2-1.html	" 解答ページ(不正解)
ins-toi2-2.html	" 解答ページ(正解)
xml-ins-3.html	名前の付け方
ins-toi3-1.html	" 解答ページ(正解)
ins-toi3-2.html	" 解答ページ(不正解)
xml-ins-4.html	要素の階層構造
ins-toi4-1.html	" 解答ページ(不正解)

ins-toi4-2.html	" 解答ページ(正解)
ins-toi4-3.html	" 解答ページ(不正解)
xml-ins-5.html	空要素
ins-toi5-1.html	" 解答ページ(正解)
ins-toi5-2.html	" 解答ページ(不正解)
xml-ins-6.html	文字の置き換え
xml-ins-7.html	CDATA セクション
xml-ins-8.html	コメント
xml-ins-9.html	確認問題
ins-toi9.html	" 解答ページ
b.html	XSLT とは
01.html	XML 文書の例(gakusei.xml)
gakusei.xml	XML 文書
xslt-kousei.html	XSLT の構成
xslt-kousei-1.html	XML 宣言
xslt-kousei-2.html	スタイルシート宣言
xslt-kousei-3.html	出力宣言
xslt-ten.html	XSLT の構成 2
html-1.html	HTML 形式に変換
html-toi.html	" 確認問題
html-toi2.html	" 解答ページ
text-1.html	TEXT 形式に変換
text-toi.html	" 確認問題
text-toi2.html	" 解答ページ
msxsl.exe	XSLT プロセッサ

トップページ(index.html)

The screenshot shows a Microsoft Internet Explorer browser window with the title "XMLとXSLTの初心者ユーザー向けオンライン教材". The address bar shows "アドレス http://2-1024/index.html". The main content area has the following structure:

XMLとXSLTの初心者ユーザー向けオンライン教材

XML編

卒論テーマと学習対象者

XSLT編

XMLとXSLTの初心者である人文情報学科生を対象に、XMLを使用する上で必要最低限の基本的な知識と技術を学習出来る教材を作成する。XMLの基本的な知識を得るために、XMLとXSLTをどのように学習していくと良いのか。また、オンライン教材を作成することで、これらの材料の学習にどのように役立つのかを考えることをテーマとする。

本教材の使い方

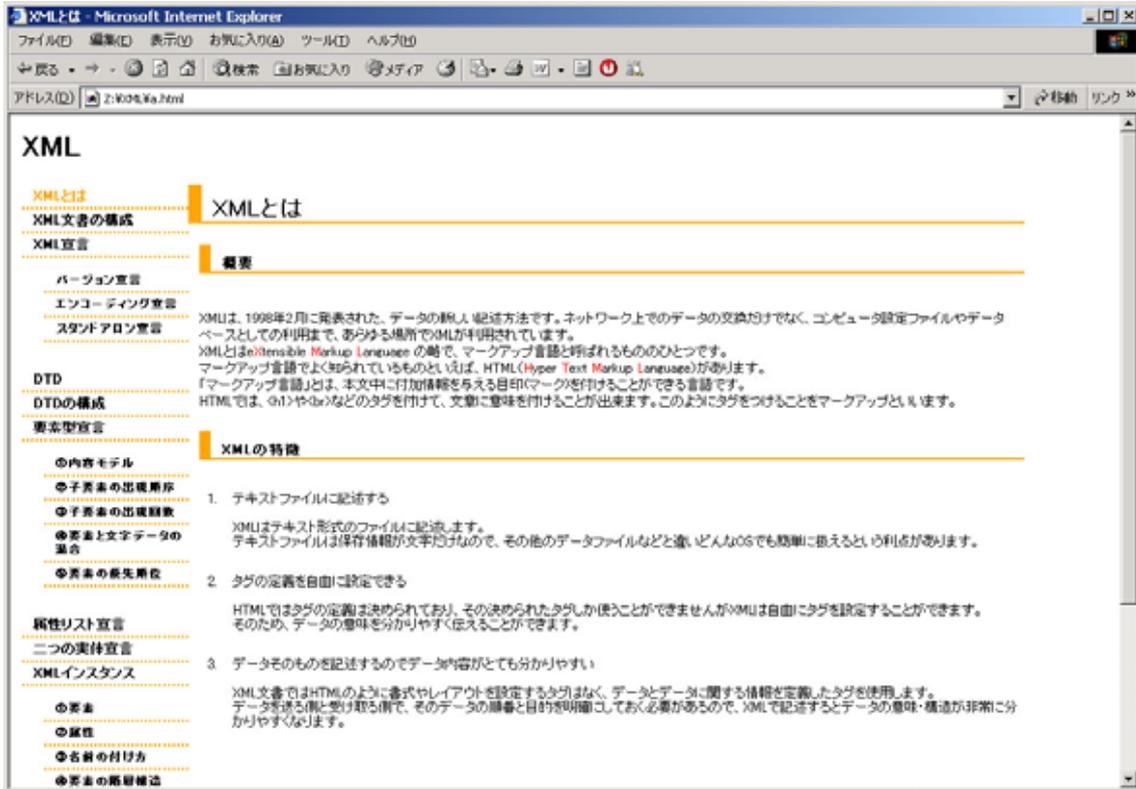
まずは、サクラエディタやメモ帳などのテキストエディタをインストールしてください。ここでは、サクラエディタを使用して教材を進めていきます。

XML編とXSLT編がありますが、まずはXML編から始めてください。

XML編	XSLT編
1. XMLとは	1. XSLTとは
2. XML文書の構成	2. XSLTの構成 <ul style="list-style-type: none">XML宣言スタイルシート宣言出力宣言
3. XML宣言 <ul style="list-style-type: none">バージョン宣言エンコーディング宣言スタンドアロン宣言	3. XSLTの構成2
4. DTD	4. HTML形式に変換
5. DTDの構成	5. TEXT形式に変換
6. 要素型宣言 <ul style="list-style-type: none">内容モデル空要素の出現頻度	

At the bottom of the browser window, a status bar shows "ページが表示されました" and "インターネット".

XML 編



The screenshot shows a Microsoft Internet Explorer browser window displaying a page titled "XMLとは". The page content is as follows:

XMLとは

XML文書の構成

XML宣言

- バージョン宣言
- エンコーディング宣言
- スタンダードアロケーション宣言

DTD

DTDの構成

- 要素型宣言
- 内容モデル
- 子要素の出現順序
- 子要素の出現回数
- 要素と文字データの混合
- 要素の優先順位

属性リスト宣言

二つの実体宣言

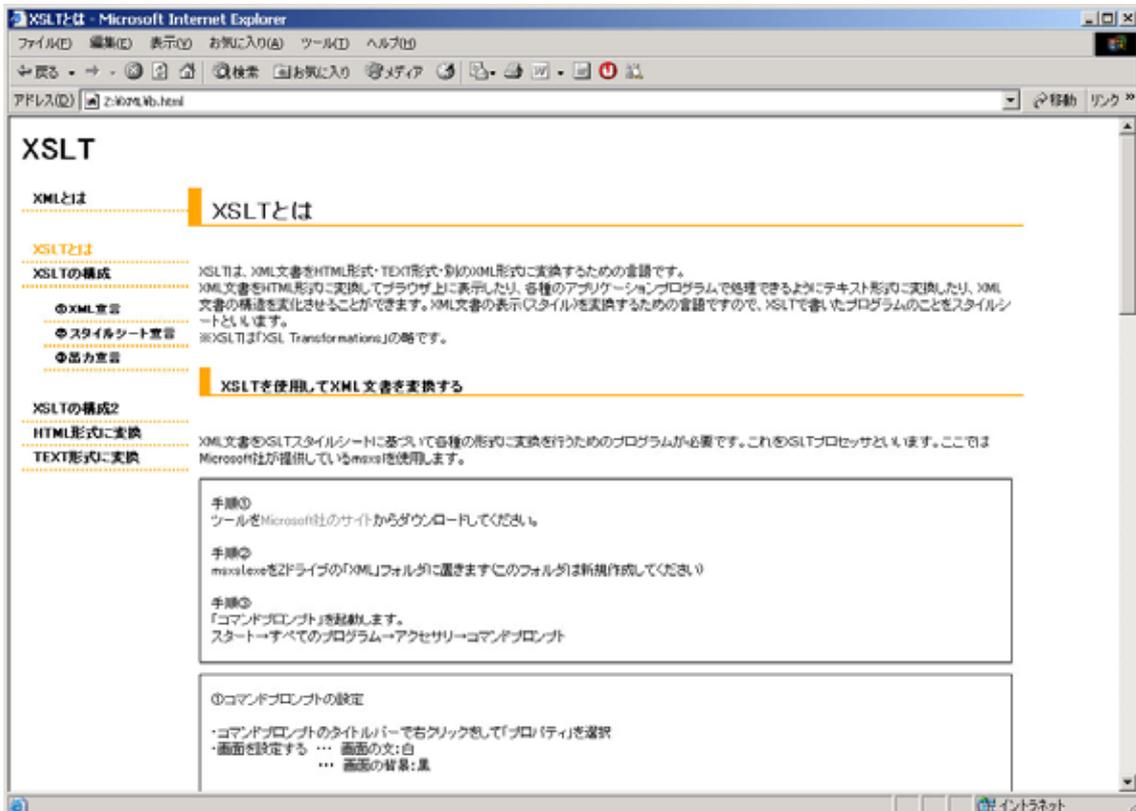
XMLインスタンス

- 宣言
- 属性
- 名前付け方
- 宣言の範囲指定

XMLの特徴

- テキストファイルに記述する
XMLはテキスト形式のファイルに記述します。テキストファイルは保存情報が文字だけなので、その他のデータファイルなどと違い、どんなOSでも簡単に扱えるという利点があります。
- タグの定義を自由に設定できる
HTMLではタグの定義は決められており、その決められたタグしか使うことができませんがXMLは自由にタグを設定することができます。そのため、データの意味を分かりやすく伝えることができます。
- データそのものを記述するのでデータ内容がとても分かりやすい
XML文書ではHTMLのように書式やレイアウトを設定するタグがなく、データとデータに関する情報を定義したタグを使用します。データを流す側と受け取る側で、そのデータの順番と目的を明確にしておく必要があるため、XMLで記述するとデータの意味・構造が非常に分かりやすくなります。

XSLT 編



The screenshot shows a Microsoft Internet Explorer browser window displaying a page titled "XSLTとは". The page content is as follows:

XSLTとは

XSLTとは

XSLTの構成

XSLTは、XML文書をHTML形式・TEXT形式・別のXML形式に変換するための言語です。XML文書をHTML形式に変換してブラウザ上に表示したり、各種のアプリケーションプログラムで処理できるようにテキスト形式に変換したり、XML文書の構造を美化させることができます。XML文書の表示スタイルを実装するための言語ですので、XSLTで書いたプログラムのことをスタイルシートともいいます。
※XSLTは「XSL Transformations」の略です。

XSLTを使用してXML文書を変換する

XSLTの構成2

HTML形式に変換

TEXT形式に変換

XML文書をXSLスタイルシートに基づいて各種の形式に変換を行うためのプログラムが必要です。これをXSLTプロセッサといいます。ここではMicrosoft社が提供しているmsxslを使用します。

手順①
ツールをMicrosoft社のサイトからダウンロードしてください。

手順②
msxsl.exeをDドライブの「XML」フォルダに置きますこのフォルダは新規作成してください。

手順③
「コマンドプロンプト」を起動します。
スタート→すべてのプログラム→アクセサリ→コマンドプロンプト

④コマンドプロンプトの設定

・コマンドプロンプトのタイトルバーで右クリックして「プロパティ」を選択
・画面を設定する ... 画面の文:白
... 画面の背景:黒

選択形式の問題

確認問題

▼外部DTDを使うXML宣言でどちらか正しいものを選びなさい。

こちらを参考にしながら解答して下さい。

- ①<?xml version="1.0" encoding="Shift_JIS" standalone="yes" ?>
- ②<?xml version="1.0" encoding="Shift_JIS" standalone="no" ?>

解答

問題のプログラム(Javascript 部分)

```
<script type="text/javascript" language="JavaScript">
<!--
function Check() {
    var tolink = document.url
    for(i=0 ; i<=tolink.list.length -1 ; i++){
        if (tolink.list[i].checked) {
            var n = tolink.list[i].value;
            document.location.href=n;
        }
    }
}
-->
</script>
```

解答ページ(dtd-toi1.html)

DTDの構成 - Microsoft Internet Explorer

ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 検索 お気に入り メディア

アドレス(D) Z:\XML\dtd-toi1.html

XML

- DTDの構成
- DTDの構成**
- 確認問題解答

不正解!

```
<?xml version="1.0" encoding="Shift_JIS" standalone="yes" ?>
```

※スタンドアロン宣言を「yes」にすると、内部DTDのみ参照することになります。

[戻る](#)

解答ページ 2(dtd-toi2.html)

DTDの構成 - Microsoft Internet Explorer

ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 検索 お気に入り メディア

アドレス(D) Z:\XML\dtd-toi2.html

XML

- DTDの構成
- DTDの構成**
- 確認問題解答

正解!

```
<?xml version="1.0" encoding="Shift_JIS" standalone="no" ?>
```

※スタンドアロン宣言を「no」に指定すると、外部DTDのみ参照することになります。

[戻る](#)

穴埋め形式の問題(html-toi.html)

```
<?xml version='1.0' encoding='Shift_JIS' ?>
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:output method= encoding='Shift_JIS' />  

<xsl:template match='/>
<html>
<head>
</head>
  < />  
</html>
< >  
<body>
  <table border='1'>
    < />  
  </table>
</body>
</xsl:template>
< >  
  <tr>
    <td>< ></td>  
    <td><xsl:value-of select='番号' /></td>
    <td><xsl:value-of select='名前' /></td>
    <td>< ></td>  
  </tr>
</xsl:template>
</xsl:stylesheet>
```

問題のプログラム(Javascript 部分)

```
<script type="text/javascript">
<!--
function check()
{
  if(document.toi.toi1.value == "xsl:apply-templates select='学生マスタ'")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}
```

```
function check2()
{
  if(document.toi2.toi3.value == "/xsl:template")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check3()
{
  if(document.toi4.toi5.value == "xsl:template match='学生マスタ'")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check4()
{
  if(document.toi6.toi7.value == "xsl:apply-templates select='学生'")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check5()
{
  if(document.toi8.toi9.value == "xsl:template match='学生'")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check6()
```

```
{
  if(document.toi10.toi11.value == "xsl:value-of select='@学科' /")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check7()
{
  if(document.toi12.toi13.value == "xsl:value-of select='自己紹介/アピール' /")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}

function check8()
{
  if(document.toi14.toi15.value == "html")
    {alert("正解です。");}
  else
    {alert("不正解です。もう一度入力して下さい。");}
}
// -->
</script>
```

アラート表示(正解の場合)

```
<?xml version='1.0' encoding='Shift_JIS' ?>
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:output method='html' encoding='Shift_JIS' />
  <xsl:template match='/'>
    <html>
      <head>
      </head>
      <xsl:apply-templates select='学生マスタ' />
    </html>
  </xsl:template>
  <xsl:template match='学生マスタ'>
    <body>
      <table border='1'>
        <tr>
          <td><xsl:value-of select='番号' /></td>
          <td><xsl:value-of select='名前' /></td>
        </tr>
      </table>
    </body>
  </xsl:template>
</xsl:stylesheet>
```

Microsoft Internet Explorer

正解です。

OK

アラート表示 2(誤答の場合)

```
<?xml version='1.0' encoding='Shift_JIS' ?>
<xslstylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsloutput method='html' encoding='Shift_JIS' />
<xsltemplate match='/'>
<html>
<head>
</head>
<xsl:apply-templates select='学生マスタ' />
</html>
</xsl:template >
<xsl:template match='ああああ'>
<body>
<table border='1'>
<tr>
<td><xsl:value-of select='番号' /></td>
<td><xsl:value-of select='名前' /></td>
<td></td>
</tr>
</body>
</xsl:template>
</xsl:stylesheet>
```

Microsoft Internet Explorer

不正解です。もう一度入力して下さい。

OK