# 第 5 回：テキストファイルでの多言語テキスト編集 Unicode 編（1）

**本日のポイント**：

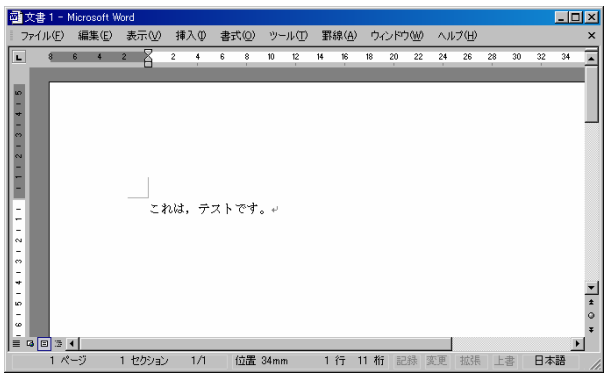
- テキストデータの保存形式
  - バイナリーファイルとテキストファイル
- Unicode とは
  - 「文字集合の国際化」という発想
  - アプリケーションの Unicode 対応
  - Unicode の変換方式：UTF-16 と UTF-8
- Word と EmEditor を使った外国語テキストファイルの編集

## 1. 文字データの保存形式

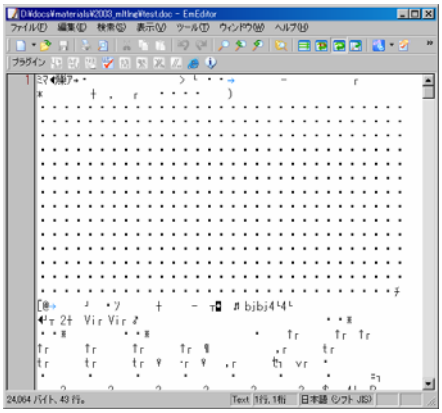
文字データは，多くのソフトウエアで利用できるが，文字データをファイルとして保存する場合，ファイルの保存形式をどうするかが問題になる。「Word 文書」や「テキスト文書」など，文字データを保存できるファイルの保存形式は数多い。各形式の特性をよく理解し，どの形式でデータを保存するかを決めることになる。

### 1.1. バイナリーファイルとテキストファイル

- 文字データを保存するファイル形式には，大きく分けて 2 種類ある：
  - **バイナリーファイル**：文字データとその他のデータが混在
  - **テキストファイル**：文字データのみ
- バイナリーファイルには，文字データとそれ以外のデータ(レイアウト情報，書式情報，文書情報，画像データなど)が一つのファイル中に混在している。テキストの編集など，ファイル内のテキストを処理するためには，データの内部構造を解釈できるソフトウエアが必要である。したがって，**汎用性**が低くなる。
- テキストを含むバイナリーファイルは，大雑把に言って以下のような種類に分けることができる。
  - －特定のソフトウエアに特化した形式：例) Word 文書
  - －複数のソフトウエアで扱える形式：例) RTF (Rich Text Format)←「参考 1」参照
  - －特定のソフトウエア用の形式だが，一定のツールで閲覧可能：例）PDF (Portable Document Format) ← 作成には専用ツールが必要。「参考 2」参照




簡単な Word 文書



左の Word 文書を「EmEditor」で開いたところ


担当：千葉庄寿（schiba@reitaku-u.ac.jp）

- 一方，テキストファイルには，文字データのみが含まれる。したがって，データの構造は単純で，テキストを扱えるあらゆるソフトウエアで利用することができ，**汎用性**は高い（EmEditor のようなテキストファイル専用のエディタはもちろん，Word や Excel といったソフトウエアでもテキストファイルを扱うことができる)。

  テキストファイルには，**書式やレイアウト，書式・フォント情報などを保存することはできない**。テキストファイルで文字以外に利用できるのは，**改行，タブ，スペースのみ**。(改行，タブ，スペースには，文字と同じようにコードが割り当てられている。)

## 2. Unicode とは

### 2.1. コンピュータと文字データ：「文字集合」と「文字エンコード方式」

コンピュータでは，全てのデータはデジタル情報として扱われる。コンピュータが文字を正確に，かつ高速に認識するためには，「どの文字をどのコード code であらわすか」をあらかじめ取り決めておく必要がある。具体的には，

(1) どの文字を扱うか（**「文字集合」** character set）を決め，
(2) 各文字に識別可能なコード(背番号のようなもの)を割り当てた**「文字エンコード方式」**(character encoding scheme，「文字コード体系」ともいう）を取り決める必要がある。さらに，
(3) 複数のコンピュータが文字データを共有するには，(1)と(2)を共通にする必要がある。

どの文字が使われるかは言語によって異なるので，文字集合はその言語が話されている国や地域によって違い，結果として，文字エンコード方式も国や地域によって異なってくる。外国語のテキストファイルを編集するためには，**各言語・地域の標準的な文字エンコード方式を知り，それを使ってテキストを作成する必要がある**。(各言語・地域で利用されている文字エンコード方式の種類と利用方法については，第 7 回以降の授業で詳しく扱う。)

### 2.2. Unicode という発想

これまでコンピュータ上で扱われてきた文字データの多くは，各言語・地域で使われている文字の種類（文字集合）にあわせ，国や地域ごとに策定された規格であるため，文字とコードの対応はまちまちであり，多言語テキストの作成は困難である。

もし，あらゆる国や地域で使われている文字を収録した文字集合があれば，全ての文字を統一したコードで指定する文字エンコード方式ができ，多言語混在処理の問題は解消される。この「文字集合の国際化」という大胆な発想の転換のもと，新たな標準として設計・開発されたのが Unicode である。Unicode はコンピュータの業界標準規格として，Windows XP をはじめ, 新しいソフトウエア（OS, アプリケーション）に実装がすすんでいる。

Unicode の最も大きな特徴は，全ての文字を 2 **バイト（「全角」と同じデータ長** = 16 **ビット**）で表すことにある（バイト，ビットについては次回授業で解説)。西ヨーロッパ言語 ISO 8859-1 では 1 バイト，日本語 Shift JIS では 1 バイト長（「半角」 = 8 ビット）と 2 バイト長（「全角」 = 16 ビット）の 2 種類の文字が混在して使われている。このように文字あたりのビット数を 2 バイトに統一することで，処理を単純にすることができるとともに，収録できる文字数が飛躍的に増大する（単純計算では $2^{16} = 65{,}536$ 個の文字が表現可能)。

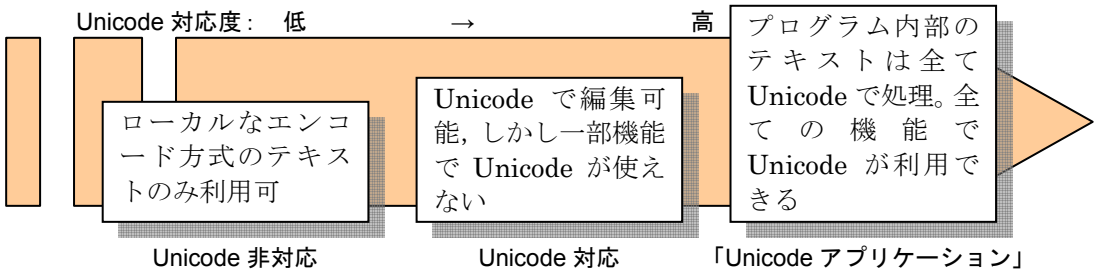
### 2.3. アプリケーションの Unicode 対応

Unicode で文字を入力・表示するためには，OS **のほか，「フォント」「入力システム」「ア**

**プリケーションソフトウエア」の全てが Unicode に対応している必要がある**。Windows XP は Unicode に対応しており，多くの国・地域の言語に対応した入力システムと Unicode フォントが標準で備わっている。

一方，アプリケーションソフトウエアの Unicode 対応状況はさまざまである。これまで授業で主に利用してきた Word2002 は，Unicode テキストを処理できるばかりでなく，プログラム自体が Unicode で書かれており，Unicode にほぼ完全に対応した**「Unicode アプリケーション」**である。また，EmEditor や Windows XP に付属する「メモ帳」，Internet Explorer なども Unicode アプリケーションである。そのほかにも表計算ソフト Excel2002 など，Unicode 対応がかなり進んだアプリケーションがある。

一方で，Unicode テキストを扱うことができるが，プログラム内部では Unicode を利用していないため，検索など，一部の機能で Unicode を利用できない，というアプリケーションも多い。これらは「Unicode 対応」だが，厳密な意味での「Unicode アプリケーション」ではない。さらに，Unicode にまったく対応していないローカルな文字エンコード方式専用のアプリケーションもある。




## 2.4. Unicode の変換方式

Unicode は「文字集合」であり，実際にテキストファイルの「文字エンコード方式」として利用する場合には，ウェブページや電子メールなど，状況に応じ，文字とコードとの対応をとりきめた Unicode の文字エンコード方式にあたる「変換形式」Transformation Format を選択してテキストを保存する。主に使われるのは UTF-16 と UTF-8 の 2 種類で，「Unicode アプリケーション」の殆どはこの 2 種類の変換形式を処理できる（他にも UTF-7 や UTF-32 などがあるが，現時点では一般的ではない)。

UTF-16：

Unicode 本来の 2 バイト（全角文字のデータ長）のコード化形式であり，コード体系が Unicode とだけ指示されている場合は，UTF-16 を指すと考えてよい。UTF は Unicode Transformation Format の略。

なお，UTF-16 は，コードの内部的な処理方法の違いにより，UTF-16LE（LE = Little-Endian）と UTF-16BE（BE = Big-Endian）の 2 種類に分かれる。2 つを区別するため，UTF-16 でテキストを保存する場合には，BOM（Byte Order Mark）をファイルの先頭に加えることになっている（次回詳しく解説する)。

UTF-8 “Unicode Transformation Format, 8bit form”：

Unicode 本来の変換形式である UTF-16 は全ての文字を 2 バイト単位で処理しなければならないので，旧来のソフトウエアでは文字が全く処理できなくなってしまう。そこで一定の計算式をもちいてデータを変換し，8 ビット単位でしか扱えないソフトウエアでも **ASCII の文字だけは表示できるよう工夫したもの**が UTF-8 である。変換の結果，文字は，その種類によって 1 バイトから 6 バイト（!）のコードで表される。


担当：千葉庄寿（schiba@reitaku-u.ac.jp）

変換の結果，UTF-8 の基本ラテン文字のコードは ASCII のコードとまったく同一となるので，ASCII を使う限り，データは UTF-8 も ASCII も同じになる。

| | t | e | s | t | て | す | と |
|---|---|---|---|---|---|---|---|
| ASCII | 74 | 65 | 73 | 74 | - | - | - |
| Shift JIS | 74 | 65 | 73 | 74 | 82C4 | 82B7 | 82C6 |
| UTF-16 | 0074 | 0065 | 0073 | 0074 | 3066 | 3059 | 2068 |
| UTF-8 | 74 | 65 | 73 | 74 | E381A6 | E38199 | E381A8 |

ASCII の文字だけは見える，という特性は，例えば Web ページなどの作成に非常に都合がよい（Unicode 非対応のアプリケーションで読み込んだ場合でも，ASCII の部分は正しく処理することができる)。

なお，日本語 Shift JIS の漢字が 2 バイトなのに対し，UTF-8 ではハングルやひらがな，カタカナ，漢字に 3 バイト必要になり，その分ファイルのサイズが大きくなる。

### 2.5. まとめ： テキストファイルで多言語テキストを利用する際の注意点

(1) 文字情報以外は保存されない
- レイアウト情報，書式情報は保存されない。
- データの重要な部分を太字(ボールド)などの文字飾りやフォントの種類，大きさで区別することはできない。

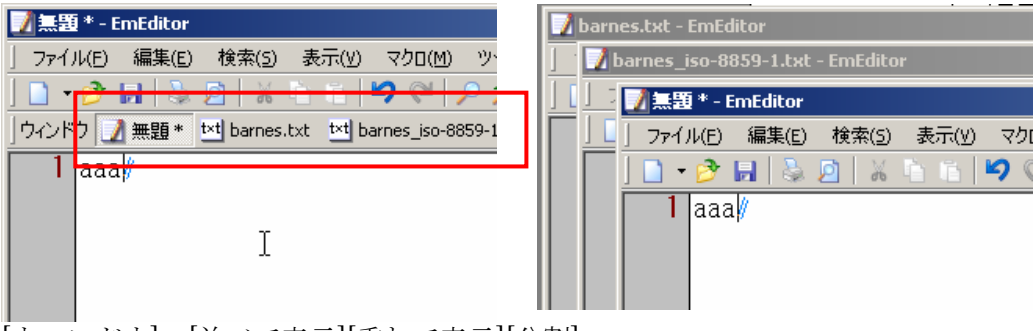
(2) 日本語のテキストファイルと，他の言語のテキストファイルは**文字エンコード方式**が異なる。
- どの文字が収録されているか（文字集合)，および
- どの文字をどのコードであらわすか（文字とコードの対応)，が異なる。

(3) ひとつのテキストファイルには通常**ひとつの文字エンコード方式**しか適用できない。
- たとえば，日本語の文字エンコード方式と中国語やドイツ語の文字エンコード方式を混在させることは通常できない。
- あらゆる文字を収録した文字集合である Unicode を使うことで，多言語混在テキストが作成可能になる。
- Unicode には，目的によって複数の文字エンコード方式（変換方式）が存在する。
- 保存時・読み込み時に間違った文字エンコード方式を選ぶと，正しく入力されているはずの文字が化けたり，文字が間違った変換をされてしまうことがある（正しい文字エンコード方式を選択していない状態で保存すると，せっかく作成したテキストが使い物にならなくなる)。

## 3. EmEditor の基本機能

- 行の折り返し表示の調整（右図)：どのようにテキストを表示したいかを自分で選ぶことができる 
- フォントの分類と設定
  - [表示]→[フォント分類]：言語ごとにフォントを設定（エディタでは通常 1 種類しかフォントを指定できないので，それぞれの言語に合ったフォントを設定してやる。各言語・地域のフォントについては第 4 回資料を参照。）
  - [表示]→[フォントの設定]：文字の大きさやフォント名，スタイルを表示・印刷を別々に指定する（設定は言語ごと。また，変更した内容は随時［表示］メニューに追加表示されていくので，元に戻すのは簡単である。）

- 設定の選択：[ツール]→[設定の選択]で編集モードを変更
  - ファイルの編集内容に合わせてテキストを色分けするほか，リンク機能を付加する。標準的な編集モードは Text。
- ウィンドウ
  - [ウィンドウ] → [タブを有効にする]：複数のテキストを一つのウィンドウ内に表示するか（左下図），ばらばらのウィンドウで表示するかを指定（右下図）

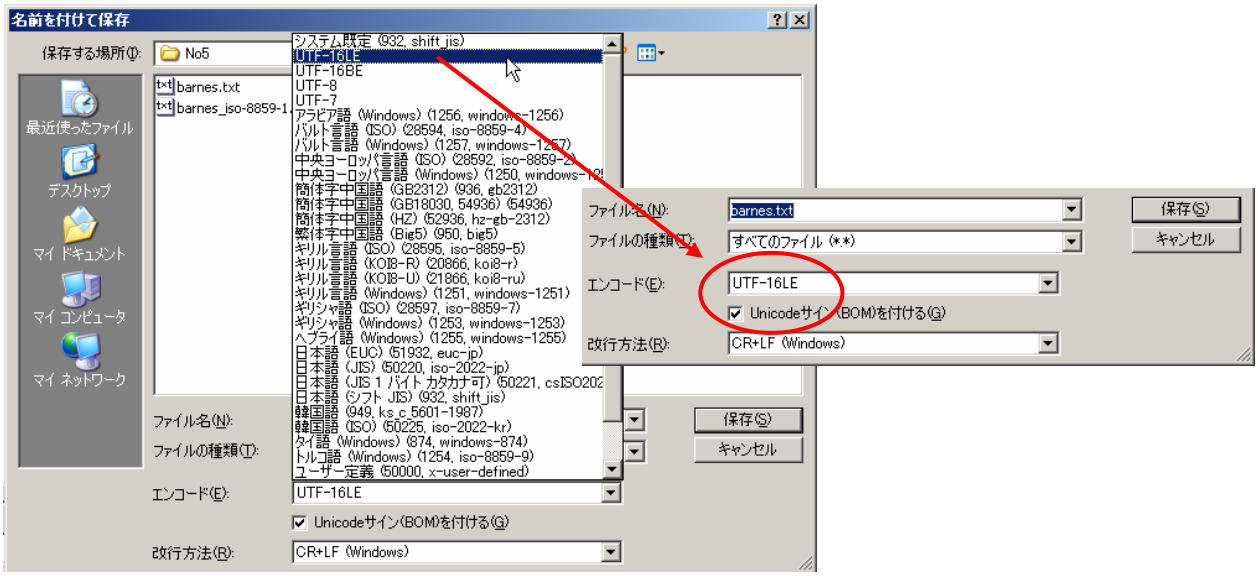



  - [ウィンドウ]→[並べて表示][重ねて表示][分割]

## 4. Unicode を用いたテキストファイルの作成・保存と編集 (1) EmEditor の場合

EmEditor Professional [1] は，大学の学生用 PC にインストールされている，多言語の編集に適した Unicode 対応テキストエディタである。起動は［スタート］ボタンから「プログラム」→「EmEditor」を選択する。

EmEditor でテキストを保存する場合，「名前をつけて保存」画面の「エンコード」で文字エンコード方式を指定する（下図）。

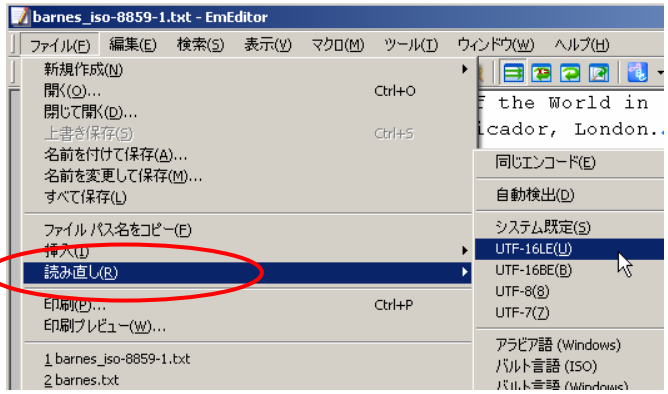



[1] Emurasoft が開発・販売するテキストエディタで，Windows に付属する「メモ帳」よりも高機能である（例えば，Unicode はもちろん多くの国・地域のエンコード方式に対応しているほか，HTML やプログラミング言語など，編集内容にあわせてテキストを色分け表示してくれる）。最新版のバージョンは 5。下記 URL よりソフトをダウンロードしてインストールできる：

URL: http://www.emeditor.com/jp/

気に入った人は自宅 PC にインストールして利用してみるとよい。ソフトは 30 日の試用期間が設けられており，継続して利用したい場合にはユーザ登録してライセンスを購入する（**シェアウエア**と呼ばれる方式）か，ソフト販売店で製品版「EmEditor Professional 2004」（標準税込価格 6,825 円）を購入する。なお，機能は限定されているが無料で利用できるフリー版も公開されている（詳細は上記ホームページを参照）。

文字エンコード方式を指定してテキストファイルを読み込む場合には，「ファイル」→「開く」で「コードページ」から指定する文字エンコード方式を選ぶか，ファイルを開いた後，「ファイル」→「読み直し」を選択し，文字エンコード方式を変更する（右図）。




- 正しい文字エンコード方式で保存された文書であれば，「読み直し」で文字エンコード方式を変更すれば，正しく表示することができる。

  ※ このとき，間違った文字エンコード方式のままで文書を編集し，「上書き保存」すると，データが間違った文字エンコード方式に強制的に変換されてしまう！文字の一部だけが文字化けするヨーロッパ言語などを編集する際には，特に注意する必要がある（詳細は第 7 回以降取り上げる）。

- 「読み直し」は保存済みのファイルでなければ使用できない。

なお，Windows XP に付属する「メモ帳」でも UTF-16 や UTF-8 でテキストを編集保存することができる（「名前をつけて保存」画面の「文字コード」という項目で設定する）[2]。

文字エンコード方式の表記法は，アプリケーションによって多少異なる。

| | Word2003 | EmEditor | 「メモ帳」 |
|---|---|---|---|
| Shift JIS (日本語) | 日本語（シフト JIS) | 日本語（シフト JIS) | ANSI (日本語環境) |
| UTF-16LE | Unicode | UTF-16LE | Unicode |
| UTF-16BE | Unicode (Big-Endian) | UTF-16LE | Unicode big endian |
| UTF-8 | Unicode (UTF-8) | UTF-8 | UTF-8 |

## 5. 実習

**実習の準備：**

file_server の Kadai にある [schiba] → [2006fl] フォルダを開き，[No5] フォルダを file_server の[user-id] の [2006fl] フォルダにコピーしなさい。

**実習 1：**

(1) file_server の[user-id] の [2006fl] フォルダを開き，[No5] フォルダにあるテキスト文書 barnes.txt を EmEditor で開きなさい。

(2) 行の折り返し表示を調整し，指定文字数で折り返して行を表示させなさい（§3 参照)。

(3) ウィンドウ右下のステータスバーで，現在の文字エンコード方式を調べ，次ページの表に記入しなさい。

[2] 日本語版 Windows95, 98, ME の「メモ帳」は Shift JIS しか扱えないので注意。

| 文字エンコード方式 | Shift JIS ・ UTF-16LE ・ UTF-16BE ・ UTF-8 |
|---|---|
| BOM | あり ・ なし |

(4) [表示] → [フォント分類] でフォントの種類を「日本語」と「西ヨーロッパ言語」に変更し，ユーロ記号が正しく表示されるのはどちらかを確かめなさい。

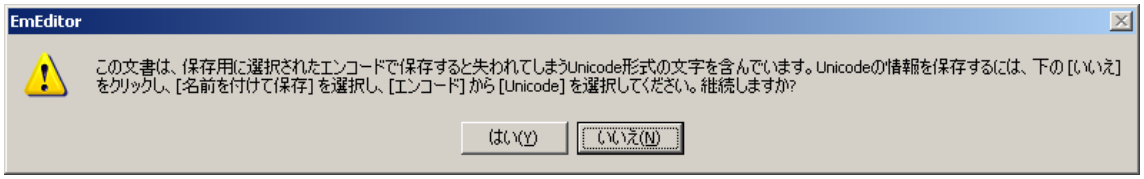
※ 「EmEditor」「メモ帳」とも，テキストの表示に使えるフォントは基本的に 1 種類である。ただし，EmEditor では，選択しているフォントで表示できない文字を「標準」フォント (大学では「ＭＳ　ゴシック」) で自動的に置き換えて表示してくれる。

(5) [表示] → [フォントの設定] で「日本語」「西ヨーロッパ言語」それぞれについて，「表示フォント」のフォント名とサイズの設定を確認しなさい。

| 日本語 | (サイズ　　pt) |
|---|---|
| 西ヨーロッパ言語 | (サイズ　　pt) |

(6) フォントの種類を「日本語」ないし「西ヨーロッパ言語」に変更した場合，一方でユーロ記号が正しく表示されない理由を出席カードに記入しなさい (フォントについては第 4 回の資料で解説済み)。

(7) [ファイル] → [名前をつけて保存] を選択し，「エンコード」を「日本語 (シフト JIS)」(Windows の日本語テキストの標準的な文字エンコード方式) に直し，ファイル名を barnes2.txt として「保存」を試みなさい。すると，「日本語 (シフト JIS) では保存できない Unicode 文字が含まれる」旨の警告 (下図) が出るので，一旦「いいえ」を押して保存をキャンセルし，どのような文字が日本語シフト JIS に含まれていないのかを確認してページ下の空欄に書き出しなさい (保存不可能な文字が強調表示される)。




次に，ファイルを barnes2.txt というファイル名で再度保存し，今度は強制的にエンコードを「日本語 (シフト JIS)」で保存しなさい (上図の警告で「はい」を選択する)。

(8) 一旦 EmEditor を閉じ，再度 barnes2.txt を開いて，先ほどチェックした「日本語 (シフト JIS)」に含まれない文字がどのように保存されるか確認しなさい ([表示] → [フォント分類] でフォントの種類を変更し，文字の表示を確認すること)。

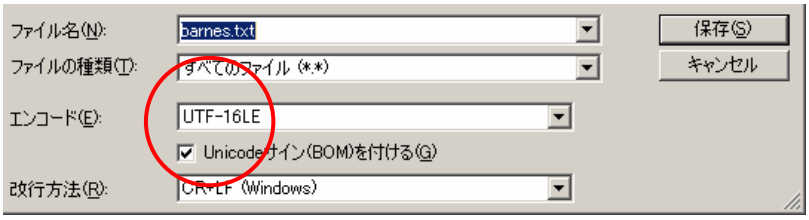
| 正しく保存できない文字 | € | |
|---|---|---|
| 強制保存した結果 | ? | |


担当：千葉庄寿（schiba@reitaku-u.ac.jp）

実習２：

以下のテキスト文書を EmEditor (ないし「メモ帳」) で作成し，指定された文字エンコード方式とファイル名で保存しておきなさい。(課題は次回授業で印刷して提出してもらうほか，後日授業で利用する。)

(1) EmEditor を使い，あなたの選択する外国語の基本的な挨拶表現と，その日本語訳を 10 通り入力し，greetings.txt というファイル名をつけ，文字エンコード方式を「UTF-16LE」(「メモ帳」で作成する場合には「Unicode」) にして [No5] フォルダに保存しなさい。EmEditor で保存する際には，下図のように「Unicode サイン (BOM) をつける」オプションをチェックしておくこと (§2.4.で述べたように，UTF-16 を利用するときには，ソフトウエアが UTF-16LE か UTF-16BE かを判断するのに BOM が必要となる)。




(2) 簡単な自己紹介の文をあなたの選択する外国語と日本語でそれぞれ作文し，[No5] フォルダに intro.txt というファイル名で保存しなさい。文字エンコード方式は UTF-8 にし，「Unicode サイン (BOM) をつける」オプションのチェックは外しておきなさい (Word, および「メモ帳」で UTF-16, UTF-8 などを指定してテキスト文書を Unicode で保存する場合には BOM がつくので注意。BOM が必須なのは UTF-16 の場合のみであり，UTF-8 を利用するときには BOM をつける必要はない)。


(以上)