

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

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

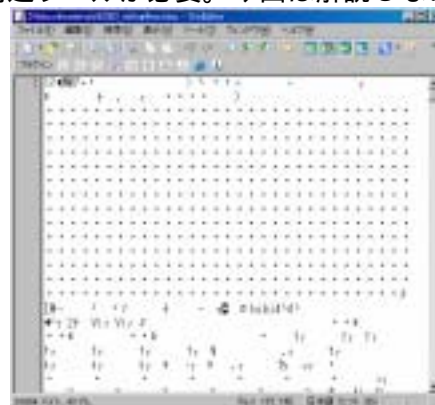
文字データは、多くのソフトウェアで利用できるが、文字データをファイルとして保存する場合、ファイルの保存形式をどうするかが問題になる。「Word 文書」や「テキスト文書」など、文字データを保存できるファイルの保存形式は数多い。各形式の特性をよく理解し、どの形式でデータを保存するかを決めることになる。

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ェアが必要である。したがって、汎用性が低くなる。
- 具体的には、バイナリーファイルは以下のような種類にわけることができる。
 - 特定のソフトウェアに特化した形式：Word 文書
 - 複数のソフトウェアで扱える形式：RTF (Rich Text Format, 「参考」参照)
 - 特定のソフトウェア用の形式だが、一定のツールで閲覧可能：PDF (Portable Document Format) ← 作成には、別途ツールが必要。今回は解説しない



簡単な Word 文書



左の Word 文書を「EmEditor」で開いたところ

- 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ェアで利用することができ、汎用性は高い (Word や Excel といったソフトウェ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一切できない。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、スペースはそれぞれ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る必要がある。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「**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、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**。(このような言語・地域ごとに利用されているエンコード方式の種類と利用方法については、次回授業で詳しく扱う。)

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、新しいソフトウェア (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 (単純計算でも $2^{16} = 65,536$ 個の文字が表現可能)。

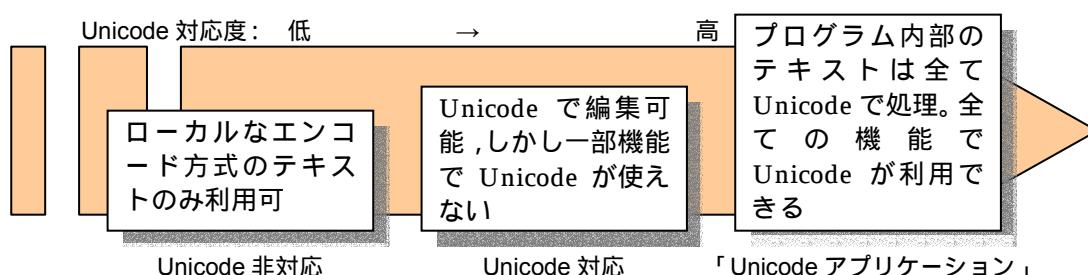
2.3. アプリケーションの Unicode 対応

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

アプリケーションソフトウェア」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ェアの 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「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エンコード方式 (Unicode の規格書では「変換形式」 Transformation Format と呼ばれる) を使用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 種類のエンコード方式を処理できる (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。

UTF-16 は、コードの内部的な処理方法の違いにより、さらに UTF-16LE (LE = Little-Endian) と UTF-16BE (BE = Big-Endian) の 2 種類があり、多くの Unicode 対応ソフトウェアはどちらも利用可能。2 つを区別するために、BOM (Byte Order Mark) をファイルの先頭に加えて保存するのが普通である (次回もう少し解説)。

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、旧来のソフトウェ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ェアでも 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。

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 (何らかの事情で文字化けしていても一部は正しく読めるわけだ)。

なお, 日本語 Shift JIS の漢字が 2 バイトなのに対し, UTF-8 ではハングルやかな, 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**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, **ひとつのエンコード方式**しか適用できない。
 - たとえば, 日本語のエンコード方式と中国語やドイツ語のエンコード方式を混在させることはできない。
 - そこで, 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, 文字が間違っただけで変換をされてしまうことがある (正しい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. Unicode を用いたテキストファイルの作成・保存と編集

3.1. Word 文書をテキストファイルとして保存

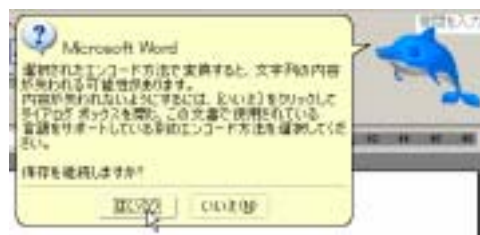
Word 2002 には, [ファイル]→[名前を付けて保存]を選択し, ファイルの種類として「書式なし(*.txt)」を選ぶことで, ファイルをテキストファイルの形式で保存する機能がある。



「書式なし」を選んで保存した外国語のテキストは, エンコード方式を指定して保存することができる。「エンコード方法」として「Windows (規定値)」(大学 PC の場合は日本語

Shift JIS), ではなく「その他」を選び, エンコード方式を選択する。

なお, 指定したエンコード方式が文字を正しく保存できない場合, 右図のような確認メッセージが表示される。



3.2. エンコードされたテキストファイルを Word に読み込む

Word2002 には, エンコードつきテキストファイルの読み込み機能もある。これを使って, 殆どの言語のテキストファイルを Word に読み込み編集することができる。

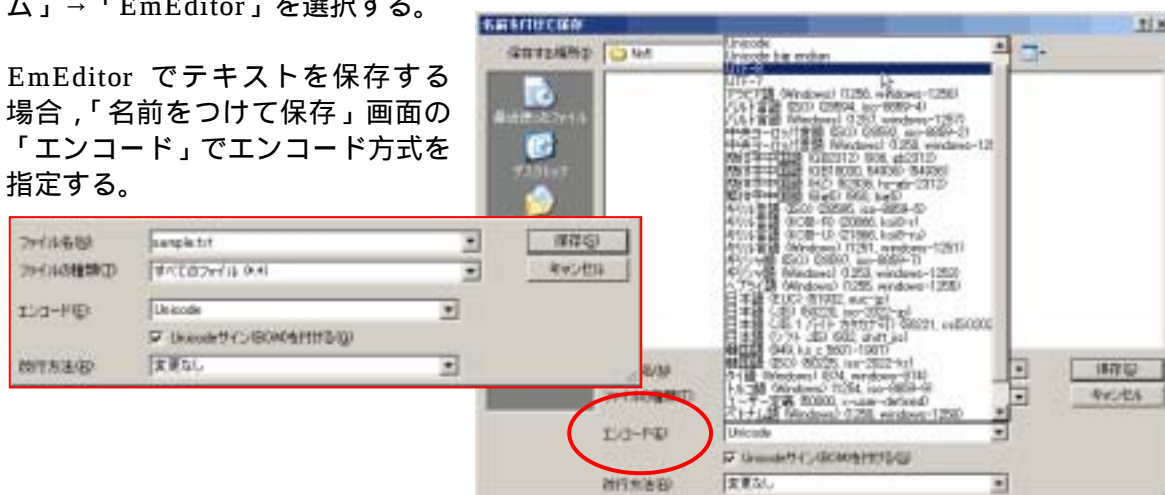
1. [ファイル]→[開く]を選択する
2. 読み込むファイルを選択する。[ファイルの種類]を「テキストファイル(*.txt)」ないし「全てのファイル(*.*)」に指定し, テキストファイルを選択する。
3. 「ファイルの変換」ダイアログでエンコード方法を選択する。「エンコード方法」を適切に指定し, 正しいエンコード方式を選ぶ (変換内容はプレビューで確かめることができる)。
4. 正しく表示できたことを確認したら, 「OK」ボタンをクリックして文書を開く。



3.3. EmEditor (メモ帳) でテキストファイルを保存・編集

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」でエンコード方式を指定する。



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)。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(大学の PC にはインストール済み)。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 (シェアウェアと呼ばれる方式) 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, 学生の場合には在学中無料で利用できるアカデミックライセンスに申し込むことができる (詳細は上記ホームページを参照)。

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、エンコード方式を変更する (WWW ブラウザで「エンコード」を変更するのと同じように、正しいエンコード方式で保存された文書であれば、エンコードを変更して読み直すことで正しく表示することができる。ただし、間違ったエンコード方式のまま文書を編集し、「上書き保存」するとデータが全く異なるエンコード方式で保存されて、データが台無しになってしまうので注意！)。

Windows XP に付属する「メモ帳」でも Unicode (UTF-16) や UTF-8 でテキストを編集保存することができる²。

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。

	Word2002	EmEditor	メモ帳 Notepad
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	Unicode	Unicode
UTF-16BE	Unicode (Big-Endian)	Unicode big endian	Unicode big endian
UTF-8	Unicode (UTF-8)	UTF-8	UTF-8

3.4. 実習

実習の準備：

file_server の Kadai にある [schiba] → [2004fl] フォルダを開き、[No5] フォルダを file_server の Home にコピーしなさい。

実習 1：

- (1) [No5] フォルダにある Word 文書 barnes_euro.doc を Word で開き、テキストファイル (「書式なし (*.*)」) として保存する。まず、「エンコード方法」として「Windows (規定値)」を選び、強制的に Shift JIS で保存しなさい。ファイル名は barnes1.txt にしなさい。
- (2) Word を閉じ、barnes1.txt を「EmEditor」でファイルを開いて、Shift JIS に含まれない文字がどのように表示されるかを確認しなさい。
- (3) barnes_euro.doc を再び Word で開き、「エンコード方法」を Unicode (UTF-8) に設定して再度「書式なし」テキストとして保存しなさい。ファイル名は barnes2.txt としなさい。
- (4) EmEditor で (3) で作成したファイルを開きなさい。[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認しなさい。また、その理由を考えなさい (フォントについては前回第 4 回の資料を参照)。

実習 2：

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。なお、Windows XP の「メモ帳」で外国語のエンコード方式を利用したテキストを編集するためには、「システムロケール」(第 2 回資料参照) を変更することが必要である (大学の PC では変更不可)。

- (1) [No5] フォルダにある Word 文書 multilingual.doc を Word で開きなさい。最後の 2 行に自分の名前とメールアドレスを加え、「上書き保存」しなさい。
- (2) 再度テキストファイル（「書式なし (*.*)」）として multilingual.txt という名前をつけて保存しなさい。「エンコード方法」（エンコード方式のこと）は Unicode (UTF-16 のこと) としなさい。
- (3) Word を閉じ、できあがったファイルを「メモ帳」で開きなさい（[スタート] → [プログラム] → [アクセサリ]）。Windows XP のメモ帳は Unicode に対応しており、Unicode (UTF-16, UTF-8) で編集された多言語のテキストファイルを開くことができる（6 ページ注を参照）。
- (4) [表示]→[フォント]で「Arial Unicode MS」を選び、「メモ帳」のフォントを変更しなさい。（Arial Unicode MS については前回第 4 回の資料を参照。）
- (5) 同様に、(1) で作成したファイルを EmEditor で開きなさい。[表示] → [フォントの設定] で表示フォントを (3) と同じ Arial Unicode MS に変更しなさい。

「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 の場合、欧文フォントを利用した場合に、日本語など、そのフォントに収録されていない文字を表示可能なフォントで代用して表示してくれる。

実習 3：以下のテキスト文書を作成し、指定された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、エンコード方式を「Unicode」（= UTF-16LE）にして [No5] フォルダに保存しなさい。「Unicode サイン (BOM) をつける」オプションをチェックすること（UTF-16 を利用するときには BOM は必須。UTF-16LE か UTF-16BE かをソフトウェアが BOM で判断する）。



- (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。エンコード方式は UTF-8 にしなさい。「Unicode サイン (BOM) をつける」オプションのチェックはつけてもつけなくともよい（BOM が必須なのは UTF-16 の場合であり、UTF-8 を利用するときには BOM をつける必要はない。Word、および「メモ帳」で UTF-16, UTF-8 でテキスト文書を保存する場合には、UTF-8 にも必ず BOM がつく。なお、Unicode 対応ソフトウェアの多くは、ファイルを開く際に自動的にエンコード方式を UTF-8 と判断してくれる。）。



参考：保存形式の選択：Word 文書と「リッチ テキスト形式」の文書

「Word 文書」形式は Word という特定のソフトウェアに特化したファイル形式であり、Word 以外のソフトウェアで利用したり、他人に渡して使ってもらう際には問題が発生することがある。相手が Word2002 を使っている場合は Word 文書をそのまま渡せるが、以下のような場合には、Word 文書を別のファイル形式で保存しなおす³ 必要がある：

- Word を使っていない相手に文書ファイルを渡す場合。
- 異なるバージョン、または異なる OS、さらには異なる言語バージョンの Word を使っている相手に文書を渡す場合。同じ Word 文書なのに正しく読み込めないという**互換性**の問題を回避することがある。

Word 文書をやりとりする場合のヒント

- ✓ 相手が使っている文書編集ソフトの種類、バージョン、OSを確認する。
- ✓ Word 文書を相手に配布する際に、自分の使っている Word のバージョン（大学の PC の場合は「日本語版 Word2002 SP-2」）をはっきり示す（[ヘルプ]→[バージョン情報]で確認しよう）。
- ✓ ソフトウェアに違いがある場合には、ファイルとともに印刷した文書を渡し、相手に文書が正しく読めているかチェックしてもらう。

レイアウト付き文書の形式のうち、Word 文書よりも汎用性の高い形式として、「**リッチ テキスト形式**」(Rich Text Format, 拡張子は .rtf, 以下 RTF) がある。RTF は Microsoft の提供している汎用性の高い文書形式で、Word で簡単に作成することができるほか、Word 以外の多くのワープロソフトで開き、編集することができる。

1. [ファイル]→[名前を付けて保存]を選択
2. 「ファイルの種類」を「リッチ テキスト形式」にする。
3. 「保存」ボタンを押す。



注意 1：Word から RTF を作成した場合、同じような内容のファイルが Word 形式と RTF の 2 つできる。この状態で、RTF 文書上で編集を行ってしまうと、**どちらがオリジナルかわからなくなってしまう**ので、混乱を避けるためにも、RTF への変換は、元の Word 文書でのテキスト編集が完了してからおこなう。

注意 2：RTF は、Word 文書よりもファイルサイズが大きくなるので注意。Word 文書が大きければ大きいほど、RTF 文書のファイルサイズも雪だるま式に増す（300KB 程度の Word 文書を RTF にすると、4MB 近くになる！）。

注意 3：RTF にもバージョンがある。新しいバージョンの RTF で作成された文書を古い RTF をサポートするソフトウェアで読むと、意図したとおりに表示されない場合がある。

注意 4：Word2000 以降、Word は全ての言語のバージョンで同じ基本プログラムを使っている。従って、2000 以降（Word2000, 2002, 2003）の Windows 版の Word を利用する場合は、英語版、ドイツ語版、中国語版など他の言語のバージョンでも日本語版 Word で作った Word 文書をそのまま開ける可能性が高い。

備考：Word では、RTF のほかに、他のワープロソフトの文書形式を選んでファイルを保存することができる。また、Word の古いバージョンの形式、また他の言語バージョンの Word の文書形式を選ぶこともできる。

³ この作業をファイル形式の「変換」convert という。