第 6 回：テキストファイルでの多言語テキスト編集 (2) さまざまなエンコード方式編

本日のポイント：

- デジタル情報としての文字：デジタルデータの基本概念
  - ビット bit
  - コード code
  - バイト byte
  - 文字コード code とエンコード方式
- Word と EmEditor を使った外国語テキストファイルの編集

コンピュータで文字を扱う場合，「どの文字をどのコードで表すか」をあらかじめ取り決めておく必要があり，その取り決めは言語や地域によってさまざまな種類があることは，既に述べた (第 5 回資料§2.1)。主要な言語・地域で標準的に用いられるエンコード方式を以下に挙げる。

言語(地域)	エンコード方式の名称
英語 (アメリカ)	us-ascii
日本語	Shift JIS
西ヨーロッパ言語	ISO-8859-1
中国語 (中国)	GB2312
中国語 (台湾)	BIG5
韓国語	EUC-KR
タイ語	TIS-620

エンコード方式が違えば，どの文字が収録されているか，またどの文字がどのコードで収録されているかが異なってくるので，エンコード方式の選択は重大である。

では，「Shift JIS **では文字** t **は** 16 **進数で** 74 **というコードをもつ**」ということは，具体的にどういう意味を表しているのだろうか？今回は，まずコンピュータにおけるデータの表記法を学び，各言語・地域で用いられているエンコード方式を使ったテキストの編集方法をマスターすることにしよう。

1. デジタル情報としての文字

1.1. コンピュータと数：デジタルデータの原理

コンピュータは，あらゆるデータを数値として処理する。我々が数を扱う場合，通常一桁が 0 から 9 までの 10 進数 decimal scale という数え方で数を扱うのに対し，コンピュータでは，数は 0 と 1 の 2 **進数** binary scale という単位で処理される。この 2 進数の数値を**ビット bit** と呼ぶ。

実習１：下記の進数対応表の 2 進数の空欄を埋めなさい。

10 進数	0	1	2	3	4	5	6	7	8	9
2 進数	0	1	10	11	100	101	110	111	1000	1001
10 進数	10	11	12	13	14	15	16	17	18	19
2 進数	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011
10 進数	20	21	22	23	24	25	26	27	28	29
2 進数	10100									

担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

2 進数は，コンピュータでは処理しやすいが，我々にとっては桁数が増える上，0 と 1 の羅列で読み間違いやすい。また， 2 進数と 10 進数の切り替えには，いちいち計算が必要なので，10 進数はコンピュータにとってかなり扱いにくい。そこで，2 進数と相性のよい **16 進数** hexadecimal でデータを表記するのが一般的である。

16 進数	0	1	2	3	4	5	6	7
10 進数	0	1	2	3	4	5	6	7
2 進数	0	1	10	11	100	101	110	111
16 進数	8	9	A	B	C	D	E	F
10 進数	8	9	10	11	12	13	14	15
2 進数	1000	1001	1010	1011	1100	1101	1110	1111
16 進数	10	11	12	13	14	15	16	17
10 進数	16	17	18	19	20	21	22	23
2 進数	10000	10001	10010	10011	10100			
16 進数	18	19	1A	1B	1C	1D	1E	1F
10 進数	24	25	26	27	28	29	30	31
2 進数								

実習２：上記の 2 進数の空欄を埋め，2 進数が実際に 10 進数・16 進数とどう対応しているかを調べなさい。

16 = 2^4 なので，2 進数 4 桁でちょうど 16 進数の 1 桁分に対応する。この対応関係はどんなに桁数が増えても一定なので，桁が増えても 2 進数を 4 桁ごとにわければ，対応する 16 進数を簡単につくることができる。

ヒント：実際に進数計算をする必要がある場合には，Windows XP に標準で付属する「電卓」を使うと便利である ([スタート]→[プログラム]→[アクセサリ]から起動)。この「電卓」には，進数の変換ができるオプションがついている。[表示]から[関数電卓]を選ぶと，16 進数までの進数選択メニューや追加の数値入力ボタンが表示される。

1.2. 文字コード

文字をコンピュータで扱う場合にも，文字は 2 進数のビットの列で処理する必要がある。そこで，どの文字をどの数値で扱うかをとりきめ，文字をその数値，つまり**「コード」**(code, 「符号」ともよばれる)で代用する方法がとられる (コードは通常 16 進数で表記する)。

実習３：上記の 2 進数と 16 進数の対応関係を参照しながら，以下の 2 進数の値を 16 進数で表記しなさい。

16 進数			
2 進数	111 0100	1110 0100	110 0001
16 進数			
2 進数	1010 1001	11 1111	1011 1111

実習４：2 進数と 16 進数の対応関係を参照しながら，以下の文字のコード(16 進数表記)を 2 進数で表しなさい。

文字	t	T
16 進数	74	54
2 進数		

1.3. ビット数と収録文字数

コンピュータで表現できる文字の数は，**ビット数**，つまりコードを何桁の 2 進数で表すかによって決まる。表現できる文字数が多くなればなるほど，コードも長くなる。言語・地域によって文字数は異なるので，エンコード方式を策定する際には，コードのビット数を決めるのが大変重要である。

	2 進数表記	16 進数表記	文字の最大数
1 ビット (2 進数 1 桁)	0〜1	0〜1	2^1 = 2 文字
2 ビット (2 進数 2 桁)	00〜11	0〜4	2^2 = 4 文字
3 ビット	000〜111	0〜8	2^3 = 8 文字
4 ビット	0000〜1111	0〜F	2^4 = 16 文字
5 ビット	00000〜11111	0〜1F	2^5 = 32 文字
6 ビット	000000〜111111	00〜3F	2^6 = 64 文字
7 ビット	0000000〜1111111	00〜7F	2^7 = 128 文字
8 ビット	0000 0000〜1111 1111	00〜FF	2^8 = 256 文字
16 ビット	0000 0000 0000 0000 〜1111 1111 1111 1111	0000〜FFFF	2^{16} = 65,536 文字

表のように，コードを表記する場合には，**0 を補って桁数をそろえる**のが普通である。

- 最も初期のエンコード方式である ASCII は，印刷可能な 94 文字 [1] を収録した 7 ビットのエンコード方式である。1963 年という早い時期に策定され，広く普及した ASCII は, 多くの言語･地域の標準的なエンコード方式に取り入れられている。

http://czyborra.com/charsets/iso646.html
(現在閉鎖中)

- これに対し，ASCII 以後のエンコード方式の多くは，ASCII よりも多くの文字数を収録するため，ビット数を増やしている。
  - 西ヨーロッパ言語, およびタイ語は 8 ビットで, ASCII の 2 倍の文字 (2^8 = 256 文字) が収録できるように拡張されている。
  - 中国語 (中国，台湾)，韓国語，日本語では，さらに，8 ビットと 16 ビットが混在して使われる。膨大な漢字やひらがな，カタカナ，ハングルを 16 ビットで表すことで，収録文字数を増やしている。

[1] 印刷可能な 94 文字に加え，さらにスペース，タブ，改行などのコードが定義されている。

担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

言語(地域)	エンコード方式の名称	利用ビット数	収録文字数
ASCII	us-ascii	7bit	94
日本語	Shift JIS	8bit, 16bit	6802(うち漢字 6349)
西ヨーロッパ言語	ISO-8859-1	8bit	189
中国語 (中国)	GB2312 [2]	8bit, 16bit	7445(うち漢字 6763)
中国語 (台湾)	BIG5	8bit, 16bit	10353
韓国語	EUC-KR	8bit, 16bit	8224(うちハングル 2350, 漢字 4888)
タイ語	TIS-620	8bit	181

2 桁の 16 進数で表されるコードは，ビット 8 桁分のデータ(2^8 = 256 通りの値)をあらわすことができる。現在のコンピュータでは，8 ビットを最も基本的なデータの単位とし，特に**「バイト」**byte と呼んでいる。

- バイトはたとえばファイルのサイズの記述に使われる (B と略される)。
- 日本語 Shift JIS のような 8 ビットと 16 ビットの文字が混在するエンコード方式の場合，8 ビットの文字は便宜的に**「半角文字」**，16 ビットの文字は**「全角文字」**と呼ぶが，バイトを使いそれぞれ**「1 バイト文字」**，**「2 バイト文字」**と呼ばれることもある。

閑話休題：現在ほとんどのコンピュータはこのように 8 ビットを 1 バイトとして処理するが，8 ビットのエンコード方式が広く利用されるようになる以前は，コンピュータはコードを 7 ビットごとに扱っていた (つまり 7 ビット= 1 バイトだった)。

1.4. Unicode **のエンコード方式（「変換方式」）とビット数** (**この節第 5 回§**2.4.**とほぼ同内容**)

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 対応ソフトウエ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)。

[2] 実際には，現在の中華人民共和国が策定しているエンコード方式の標準は，収録漢字を大幅に増やした GB18030 である (現在中国で販売されるソフトは GB18030 に対応することが義務づけられている)。GB2312 は GB18030 の一部として含まれ，GB2312 で作ったテキストはそのまま GB18030 テキストとして使用できるよう工夫されている。但し，電子メールなど多くのメディアで実際に利用されているエンコード方式の主流は現在でも GB2312 である。

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。UTF-8⇔UTF-16 の文字の対応は完全に規則的なので，自動的に処理することができる。

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，変換の結果，UTF-8 の文字は，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。上記の表からわかるように，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，その分ファイルサイズが大きくなる。

1.5. Windows コードページと標準的なエンコード方式 (少々高度な情報)

エンコード方式の正式な名称に加え，Windows では，表にみられる独自の呼び方をすることがあるので，覚えておくとよい (Internet Explorer や Word でのエンコードの選択欄，テキストエディタ EmEditor などで使われている)。

Windows では，各言語・地域のエンコード方式を**「コードページ」**Codepage と呼ばれる独自の番号で管理している。言語・地域によっては，Windows **のコードページと，対応する標準的なエンコード方式の収録する文字数が異なることがあるので注意が必要**。

言語・地域で標準的に使われるエンコード方式とコードページ

言語(地域)	標準的な文字エンコードの名称	Windows での名称	Windows コードページ Codepage
日本語	Shift JIS	日本語 (シフト JIS)	932
西ヨーロッパ言語	ISO-8859-1	西ヨーロッパ言語(ISO)	
		西ヨーロッパ言語(Windows)	1252
中国語 (中国)	GB2312	簡体字中国語 (GB2312)	936
中国語 (台湾)	BIG5	繁体字中国語 (Big5)	950
韓国語	EUC-KR	韓国語 (EUC)	
		韓国語	949
タイ語	TIS-620		
		タイ語 (Windows)	874

詳しいコードページの情報は，http://www.microsoft.com/globaldev/reference/WinCP.asp に詳しいので，参照するとよい。Windows の主なコードページは以下のとおり。

担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

- 1 バイト言語のコードページ
  - 1250 (中央ヨーロッパ言語, Central Europe)
  - 1251 (キリル言語, Cyrillic)
  - 1252 (西ヨーロッパ言語, Latin 1 *)
  - 1253 (ギリシャ語, Greek)
  - 1254 (トルコ語, Turkish)
  - 1255 (ヘブライ語, Hebrew)
  - 1256 (アラビア語, Arabic)
  - 1257 (バルト言語, Baltic)
  - 1258 (ベトナム語, Viet Nam)
  - 874 (タイ語, Thai)

* Latin 1 は ISO-8859-1, ないし Windows コードページ 1252 の通称。

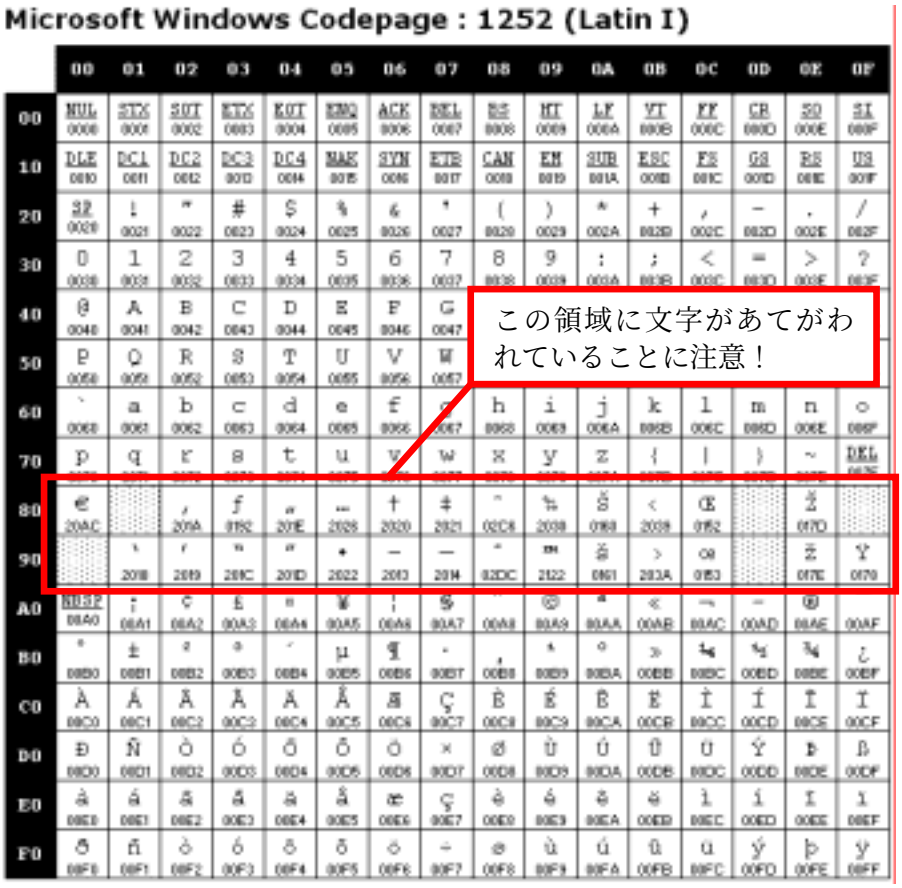
- 2 バイト言語のコードページ
  - 932 (日本語シフト JIS, Japanese Shift-JIS)
  - 936 (簡体字中国語, Simplified Chinese GBK)
  - 949 (韓国語, Korean)
  - 950 (繁体字中国語, Traditional Chinese Big5)

外国語テキスト文書を正しく使うためには，標準的なエンコード方式とともに，Windows 独自のエンコード方式に注意する必要がある。次ページの表は，コードページ 1252 (Latin 1)，西ヨーロッパ言語のコードページである。ISO-8859-1 のコード表と比べ，違いを調べよう。

コードページ 1252 (西ヨーロッパ言語) では，ユーロ記号や引用符のバリエーション，さらには若干のアクセント記号つきアルファベットなど，**標準的な** ISO-8859-1 **のコードでは表現できない文字が使えることに注意しよう**。これらは，Windows 上でのみ有効な文字コードで，Mac やその他の OS では文字化けしてしまう。逆にいうと，1987 年に制定されたエンコード方式 ISO-8859-1 の規格に従った場合，ユーロ記号は**使えない**。

ISO-8859-1 **に「正式に」収録されている文字**

	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ



(http://www.microsoft.com/globaldev/reference/sbcs/1252.htm)

Windows コードページによる標準的なエンコード方式の拡張は，韓国語 (949) やタイ語 (874) でも行われている。

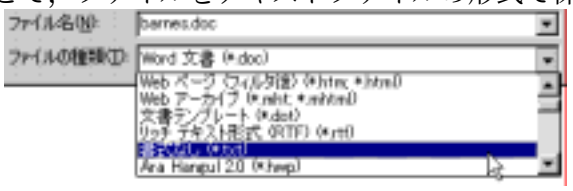
問題 1：あなたの選択している外国語の Windows コードページの名称と番号を確認しよう。
問題 2：ユーロ記号が入ったテキストを，例えば電子メールで ISO-8859-1 を使って送信した場合，相手によってはどのような問題が発生するだろうか。考えてみよう。
問題 3：ISO-8859-1 を使ってユーロを用いる必要のあるテキストを作成したい場合，どうしたらよいだろうか。解決策を考えよう。

2. Word および EmEditor での外国語テキストファイルの保存と編集

各言語・地域のエンコード方式によるテキストファイルの保存と保存の手順は，前回の Unicode での保存・編集の方法と同じである。ただし，エンコード方式により収録文字が大きく異なるので，エンコード方式の選択をよく確認して作業する必要がある。

2.1. Word 文書をテキストファイルとして保存

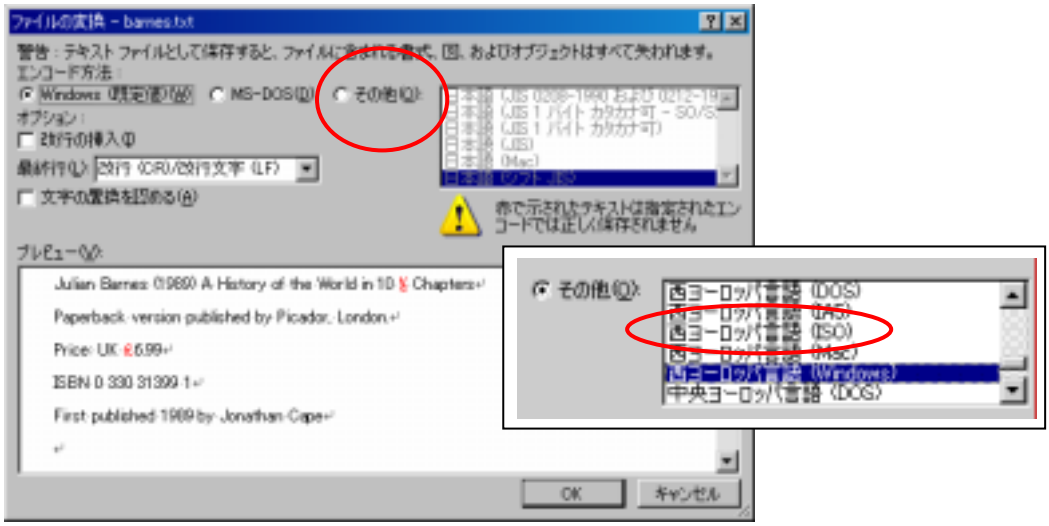
Word 2002 には，[ファイル]→[名前を付けて保存]を選択し，ファイルの種類として「書式なし(*.txt)」を選ぶことで，ファイルをテキストファイルの形式で保存する機能がある。



担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

「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，保存したい言語・地域のエンコード方式を選択する。

実習５：file_server の Kadai にある [schiba] フォルダから [2003fl] フォルダを開き，その中の[No6] フォルダを file_server の Home にコピーしなさい。コピーしたフォルダにある barnes_noeuro1.doc を Word で開き，ファイルの種類を「書式なし」とし，barnes1.txt というファイル名をつけてテキストファイルとして保存しなさい。エンコード方式として，「西ヨーロッパ言語 (ISO)」を指定しなさい。



実習６ (実習５の応用問題，§2.5. を参照)： [No6] フォルダにある barnes_euro.doc を Word で開き，ファイルの種類を「書式なし」とし，barnes2.txt というファイル名をつけてテキストファイルとして保存しなさい。まず，エンコード方式として「西ヨーロッパ言語 (ISO)」を指定し，どの文字が変換の際問題をおこすか確認しなさい。その上で，テキストを「西ヨーロッパ言語(Windows)」(Windows コードページにより拡張された文字を含む西ヨーロッパ言語用のエンコード方式) として保存しなさい。

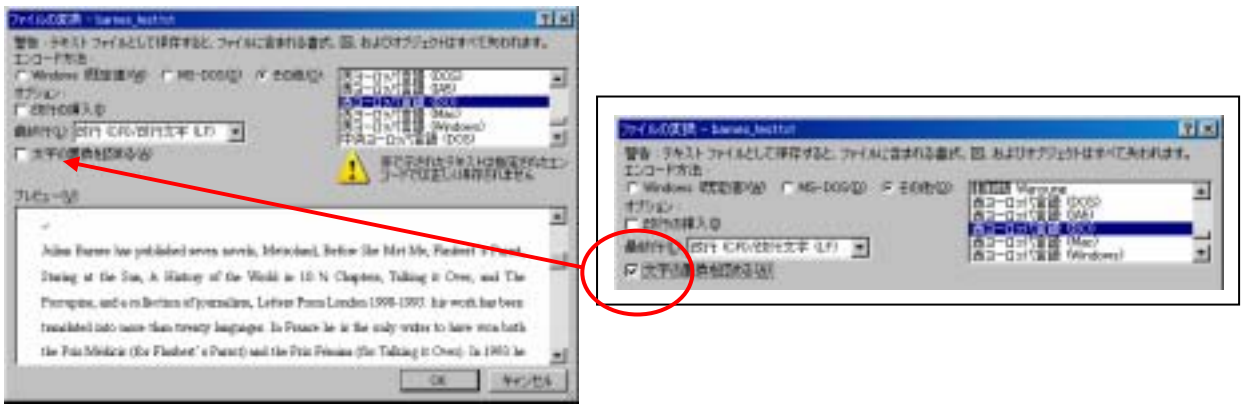
Word 2002 による書き出しでは，テキストを希望するエンコード方式に書き出した場合におこる変換漏れを Word が検証し，指摘してくれる利点がある (例えば，テキストに日本語の全角文字が入ったテキストを欧文として書き出そうとすると注意がでる。このような場合，必要ならばもとのテキストに戻り，修正を加えてテキストを的確かつ安全に書き出すことができる)。変換できない文字が含まれている場合，テキスト文書ではその文字が「?」や他の文字に置き換えられてしまう [3] ので注意すること。

※ Word を使ってテキストファイルを保存する場合のヒント

正しいエンコード方式を指定したにも関わらず，「正しく保存されない」というメッセージが出る場合がある。赤で示された箇所が記号の場合，Word のオートコレクト機能によりその記号の文字種が微妙に変更されたために起こっていることが多い (引用符のオートコレクトについては第 4 回資料§3 を参照)。この場合，「文字の置換を認める」というオプションをチェックする

[3] 変換メニューの「文字の置換を認める」オプションをチェックすると，Word は自動的に変換できない文字を「それらしい」文字に可能な限り置換したうえで変換をおこなう。

と, これらの記号を保存したいエンコード方式にある文字種の記号に自動的に変更した上でテキストファイルを保存することができる (この処理をしないと，該当する記号が「?」に置き換えられて保存されてしまうので注意)。



実習７ (実習５の応用問題，オートコレクトを含む文書の変換と保存)：[No6] フォルダにある barnes_noeuro2.doc を Word で開き，ファイルの種類を「書式なし」とし，barnes1b.txt というファイル名をつけてテキストファイルとして保存しなさい。エンコード方式として「西ヨーロッパ言語 (ISO)」を指定し，そのまま保存した場合にどの文字が問題をおこすか確認しなさい。その上で，「文字の置換を認める」のチェックを入れ，該当する記号が正しく保存されることを確認して保存しなさい。

2.2. 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，さまざまな言語のテキストファイルを Word に読み込み処理することができる。

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。

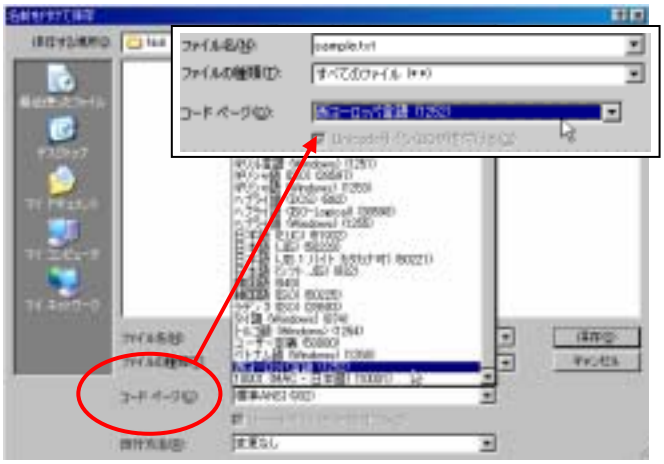
実習７： 実習５で作成した barnes1.txt をテキストファイルとして Word に読み込みなさい。エンコード方式を「西ヨーロッパ言語 (ISO)」に指定し，文字がきちんと表示されることを確認して開くこと。

2.3. EmEditor でテキストファイルを保存・編集

EmEditor は，Unicode だけでなく，さまざまな言語や地域のエンコード方式もサポートしている。(Windows に付属する「メモ帳」はその PC のシステムロケールで設定されたエンコード方式と Unicode (UTF-16, UTF-8) のみに対応している。)

担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

EmEditor でテキストを外国語のエンコード方式を指定して保存する場合は，「名前をつけて保存」画面の「コードページ」で正しいエンコード方式を指定する(右図)。



指定したエンコード方式に含まれない文字が存在する場合には，下図のような警告メッセージが出る。「いいえ」をクリックして元に戻ると，保存できない文字が赤字で表示され，問題箇所を確かめることができる。(そのまま保存すると保存したファイルでは該当箇所が「?」となってしまう。)

EmEditor で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。

実習８： 先週の課題として[No5] フォルダに作成したファイル intro.txt を EmEditor で開きなさい。外国語と日本語で作成した自己紹介のテキストを，新規に作成した EmEditor ないし Word の文書にコピーし，外国語と日本語に分けて 2 つのファイルに別々に保存しなさい (ファイルの保存場所は [No5] フォルダにしなさい)。日本語の自己紹介テキストの保存には日本語 Shift JIS を，外国語の自己紹介の保存にはその外国語で標準的に用いられるエンコード方式を用いなさい。ファイル名は，保存する言語に合わせ以下のようにすること(intro の後にくる _ は「アンダーバー」)。

英語	**intro_en.txt**
韓国語	**intro_ko.txt**
スペイン語	**intro_es.txt**
タイ語	**intro_th.txt**
中国語(台湾)	**intro_zh-TW.txt**
中国語(中国)	**intro_zh-CN.txt**
ドイツ語	**intro_de.txt**
日本語	**intro_ja.txt**
フランス語	**intro_fr.txt**

intro_ …の後にくる言語名の表記は，RFC1766 という国際規格に従っている (RFC1766 はウェブページなどで用いられる。日本語は jp でなく，ja であることに注意)。

次回授業について：

次回は大学のメールシステム Active!Mail を用いて外国語での電子メールの送受信を行う。各自，別配布した情報システムセンターの配布資料「電子メールの利用 -Active!Mail- 」を参照し，Active!Mail でのメールの送受信の方法を確認しておくとよい。なお，次回実習で利用するので，外国語で作成した自己紹介のテキスト (実習 8) は必ず準備しておくこと。