ありうべき日本語処理とは

by 中 博俊

我々が普段利用している日本語。

無意識のうちに日本語処理を
行っていませんか？

日本人が、日本語を使う上で、
自然体に利用できる情報処理
が求められています。

残念ながら私は日本語学者ではなくDeveloperです。

記述の中には日本語の歴史などで誤りがあるかもしれませんが、その節はご容赦ください。

キーワードについて

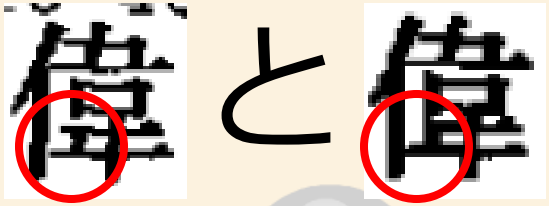
- 言語処理についてのキーワードはいろいろあります。
- 1つ1つおさらいしましょう。

キーワードについて

- 文字セット(Character Sets)
  - 字体を定義した文字の集合
  - ASCII, JIS, Unicode
    - JIS(n面m区o点) 区点コードなどとも
    - Unicodeなど(U+0000などと表現)
- エンコード(Encode)
  - ある文字セットなどに番号を振り、実際に取り扱う形式のこと
  - ShiftJIS, EUC, UTF-8, UTF-16
    - ShiftJISにもJISにない漢字が含まれている。文字セットでもある。

キーワードについて

- 字体(Character)
  - 概念的なもの。

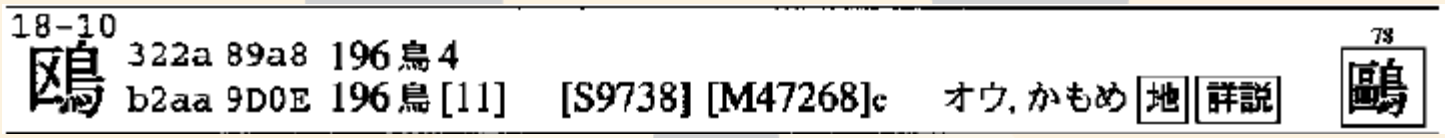
－  などを区別しない。

- 字形(Glyph)
  - 文字通り字の形
  - 前述の違いを区別する
- 書体(Style)

－薔薇(メイリオ)薔薇(MSP明朝)

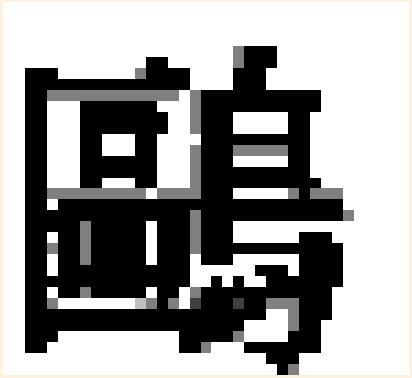
キーワードについて

- 包摂
  - 偉と偉は違う字だけれど、見る人はその字の違いに有意差を見出さない関係。
    - JISでも包摂関係の設定はそこそこある。



両者は包摂関係にある。

両者の字体は同一。
コード化した場合も同一(JISコード)



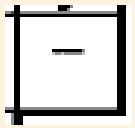
文字に関するJIS標準

- ASCII(1963)
- JISX0201(1969)
  - ラテン文字と半角カタカナ
- JISX0208(1978(旧JIS), 1983(新JIS))
  - 第1水準, 第2水準
- JISX0212(1990)
  - 補助漢字
- JISX0213(2000)
  - 第3水準, 第4水準

ASCIIとJISの非互換

- ASCIIは文字集合です。
- JISも文字集合です。
- 両者は同一、または包含関係のように思われますが、文字集合としての互換性はありません。

YEN SIGN



OVER LINE

	0	@	P	`	p
!	1	A	Q	a	q
"	2	B	R	b	r
#	3	C	S	c	s
$	4	D	T	d	t
%	5	E	U	e	u
&	6	F	V	f	v
'	7	G	W	g	w
(	8	H	X	h	x
)	9	I	Y	i	y
*	:	J	Z	j	z
+	;	K	[	k	{
,	<	L	¥	l	|
-	=	M	]	m	}
.	>	N	^	n	‾
/	?	O	_	o	

文字に関するJIS標準

- ASCII(1963)
- JISX0201(1969)

- 初版制定年度から考えても、日本語がカタカナだけとはいえ出るだけで大きな前進
- 通貨記号がないと実質的に利用できないため、￥と＼の違いはどうでもよかったと思われる。

今回のVista問題は一体何の問題？

- 字形が変わる
- 字が増える
- 結合文字の正式対応

いままでも散々変わっています。

字形が変わる(JISX0208の変遷)

78　1978 年 2 月 1 日刊行第 1 次規格に用いられた字形 (第 2 次規格で改変)。

78 誤　第 1 次規格の正誤票で誤とされた字形で，誤字形であったもの。当該区点位
当する区点位置は，51-48(嗤)，54-82(幣)，73-28(蘂)，80-19(睚) の
については，附属書 7 “区点位置詳説” に示す。

78/1　第 1 次規格第 1 刷に用いられ，1978 年 11 月発行の正誤票で置換えられた
該区点位置に包摂される。

-78/4　第 1 次規格第 4 刷より前の規格票に用いられ，第 4 刷附属の正誤票で置換
はなく，当該区点位置に包摂される。

-78/4X　第 1 次規格第 4 刷より前の規格票の字形索引に用いられ，第 4 刷附属の正
誤字形ではなく，当該区点位置に包摂される。

78/4 正　第 1 次規格第 4 刷附属の正誤票で置換えが指
く，当該区点位置に包摂される。

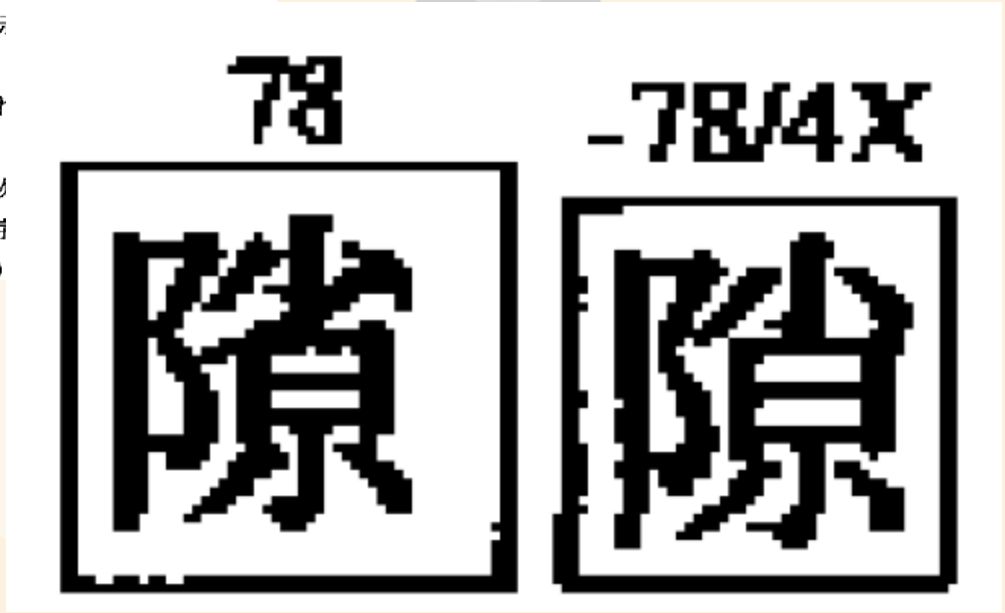
78/4-　第 1 次規格第 4 刷以降の第 1 次規格で用いら

78/5　第 1 次規格第 5 刷だけに見える字形。

83　1983 年の第 2 次規格に用いられた字形 (第 3 次

78-83　第 1 次規格・第 2 次規格を通じて用いられた字

なお，この規格の例示字体の字形は，第 3 次規格 (1990



字形が変わる

今回の変更は2000年の国語審議会の審議がベースになっています。

この2000年の国語審議会の答申のポイントは3つ

- 表外漢字字体表
  - 今回の範囲
- 国際社会に対応する日本語の在り方
  - Hirotoshi, Nakaと書くかNAKA Hirotoshiと書くか等
- 現在社会における敬意表現
  - 敬語についてなど。この後文化審議会 国語分科会(国語審議会の現在の継承機関) 敬語小委員会で、5種類に分けるなどが最近の話題。

表外漢字字体表についてのポイント

印刷標準字体

字体の中で標準とする字形は常用漢字を除き康熙字典に原点を見出すこと。

３部首許容

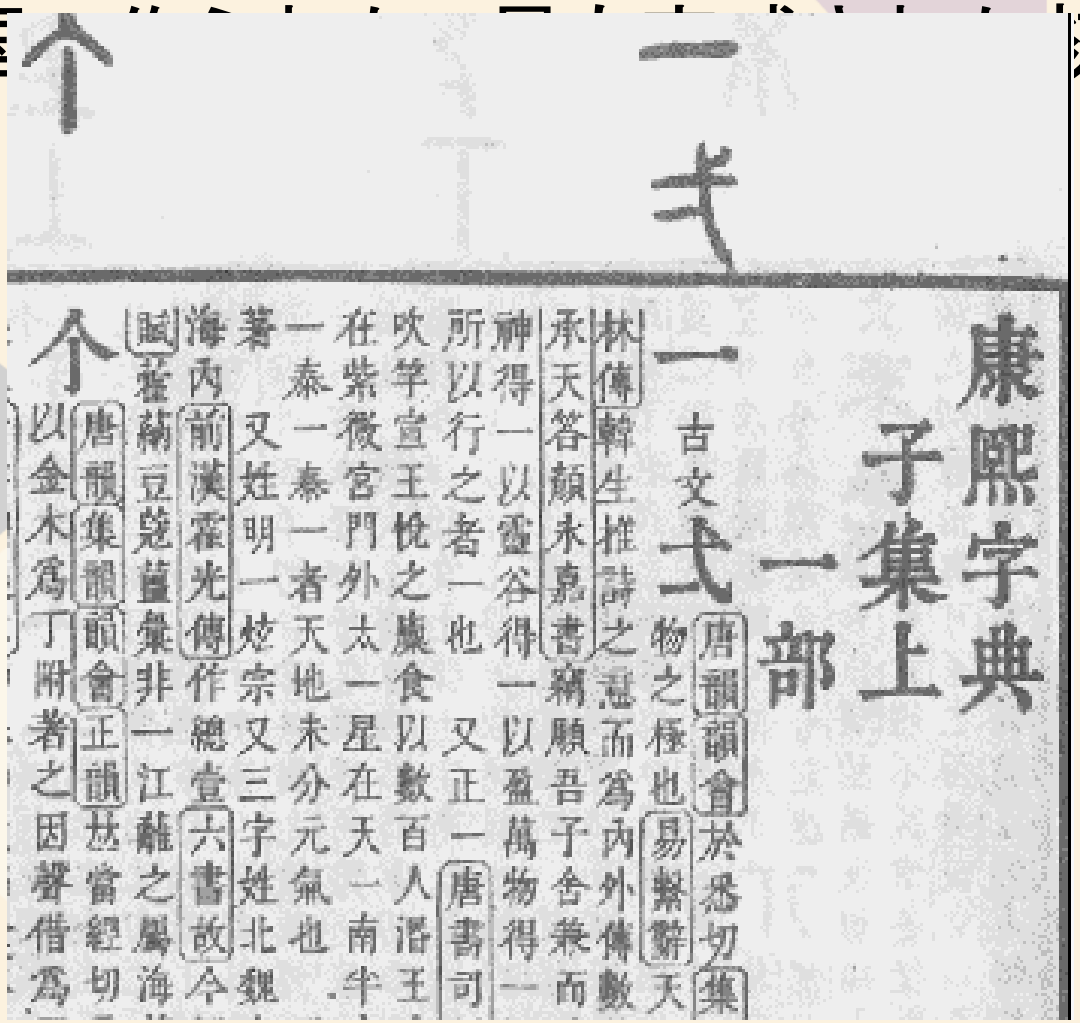
しんにゅう、しめすへん、しょくへんは昔より下の形を印刷で使ってきたから、特別に許す。

そもそも康熙(熙)字典ってなによ？

- 1716年に中国で作られた、最も完成された初めての漢字の事典

そもそも康熙(熙)字典ってなによ？

- 1716年に中国[illegible]切めての漢字の



常用(当用)漢字vs表外漢字

- 現在の常用漢字は1946(昭和二十一年)/1/16に吉田茂首相の名前で出されたものが最初。
- 1949(昭和二十四年)/4/28に字体の変更などが大掛かりに行われた。
- 澁谷區 → 渋谷区と変更されたのは戦後の漢字行政の決定による。
- 區は区とされたにもかかわらず、森鴎外の鴎の字は鷗に戻る。

常用漢字

- 常用漢字自体は今後定期的に改定しようとしている。
- 新常用漢字表を平成十九年度の文化審議会で策定するような方向
- 常用漢字表に表外漢字から採用されると、代表字形も変わるかもしれない。

異体字をどうするの？

- 澑(U+6F80, JIS213:1-63-8)
- 澁(U+6F81, JIS213:1-63-7)
- 渋(U+6E0B,JIS213:1-29-34)
- 区(U+533A, JIS213:1-22-72)
- 區(U+5340, JIS213:1-50-31)

異体字をどうするの？

- .NET System.String
  - PS C:¥Users¥localnaka> "渋" -eq "澁"
  - False
  - PS C:¥Users¥localnaka> "渋" -eq "澁"
  - False
- .NET System.Data.SqlTypes.SqlString
  - $a = New-Object Data.SqlTypes.SqlString "渋"
  - $b = New-Object Data.SqlTypes.SqlString "澁"
  - $a -eq $b
- SQL Server 2005
  - declare @a table(col nvarchar(100))
  - insert into @a values('渋')
  - insert into @a values('澁')
  - insert into @a values('澁')
  - select * from @a where [col] collate Japanese_90_CI_AI = '渋'

異体字をどうするの？

- .NET System.String
  - PS C:¥Users¥localnaka> "渋" -eq "澁"
  - False
  - PS C:¥Users¥localnaka> "渋" -eq "澁"
  - False
- .NET Syst[illegible] Data.[illegible]
  - $a = N[illegible]
  - $b = Ne[illegible]
  - $a -eq $b
- SQL Server 2005
  - declare @a table(col nvarchar(100))
  - insert into @a values('渋')
  - insert into @a values('澁')
  - insert into @a values('澁')
  - select * from @a where [col] collate Japanese_90_CI_AI = '渋'

全部だめ

今回のVista問題は一体何の問題？

- 字形が変わる
- 字が増える
- 結合文字の正式対応

今回のVista問題は一体何の問題？

Unicode対応してないの？

今回のVista問題は一体何の問題？

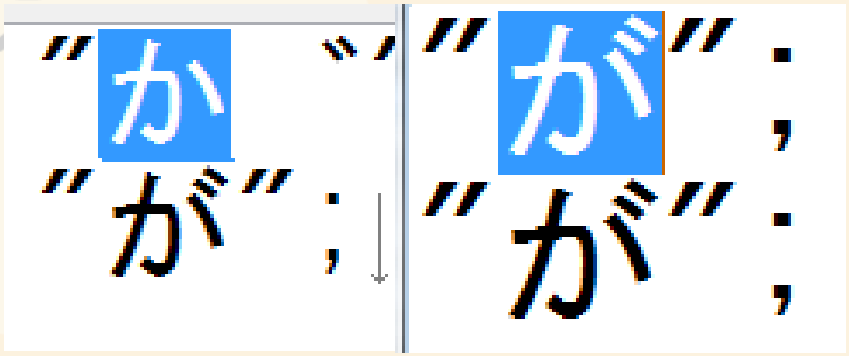
- 字形が変わる
- 字が増える
- 結合文字の正式対応

Unicode結合文字をどうするの

- サロゲートペアのことではありません。

304B か HIRAGANA LETTER KA
304C が HIRAGANA LETTER GA
≡ 304B か 3099 ゙
304D き HIRAGANA LETTER KI

- “が”という文字はU+304Cで定義しているが、U+304B, U+3099も同様とみなすという意味。
- Windows Vistaから正式に対応

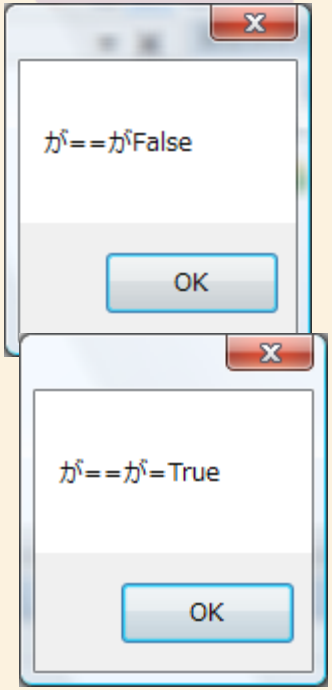


Unicode結合文字をどうするの

- .NET System.String1(ただのEqual)
  - string a = "が;
  - string b = "が";
  - MessageBox.Show((a + "==" + b +"=" + (a == b).ToString()).ToString());
- .NET System.String1(カルチャ依存)
  - MessageBox.Show((a + "==" + b + "=" + (string.Equals(a, b, StringComparison.CurrentCulture)).ToString()).ToString());
- .NET System.Data.SqlTypes.SqlString
  - System.Data.SqlTypes.SqlString a = new System.Data.SqlTypes.SqlString("が);
  - System.Data.SqlTypes.SqlString b = new System.Data.SqlTypes.SqlString("が");
  - MessageBox.Show((a + "==" + b +"=" + (a == b).ToString()).ToString());
- SQL Server 2005
  - declare @a table(col nvarchar(100))
  - insert into @a values(nchar(12363) + nchar(12441))
  - insert into @a values('が')
  - select * from @a where [col] = 'が'

Unicode結合文字をどうするの

- .NET System.String1(ただのEqual)
  - string a = "が;
  - string b = "が";
  - MessageBox.Show((a + "==" + b +"=" + (a == b).ToString()).ToString());
- .NET System.String1(カルチャ依存)
  - MessageBox.Show((a + "==" + b + "=" + (string.Equals(a, b, StringComparison.CurrentCulture)).ToString()).ToString());
- .NET System.Data.SqlTypes.SqlString
  - System.Data.SqlTypes.SqlString a = new System.Data.SqlTypes.SqlString("が);
  - System.Data.SqlTypes.SqlString b = new System.Data.SqlTypes.SqlString("が");
  - MessageBox.Show((a + "==" + b +"=" + (a == b).ToString()).ToString());
- SQL Server 2005
  - declare @a table(col nvarchar(100))
  - insert into @a values(nchar(12363) + nchar(12441))
  - insert into @a values('が')
  - select * from @a where [col] = 'が'



そんなことより

繰り返し文字をどうするの

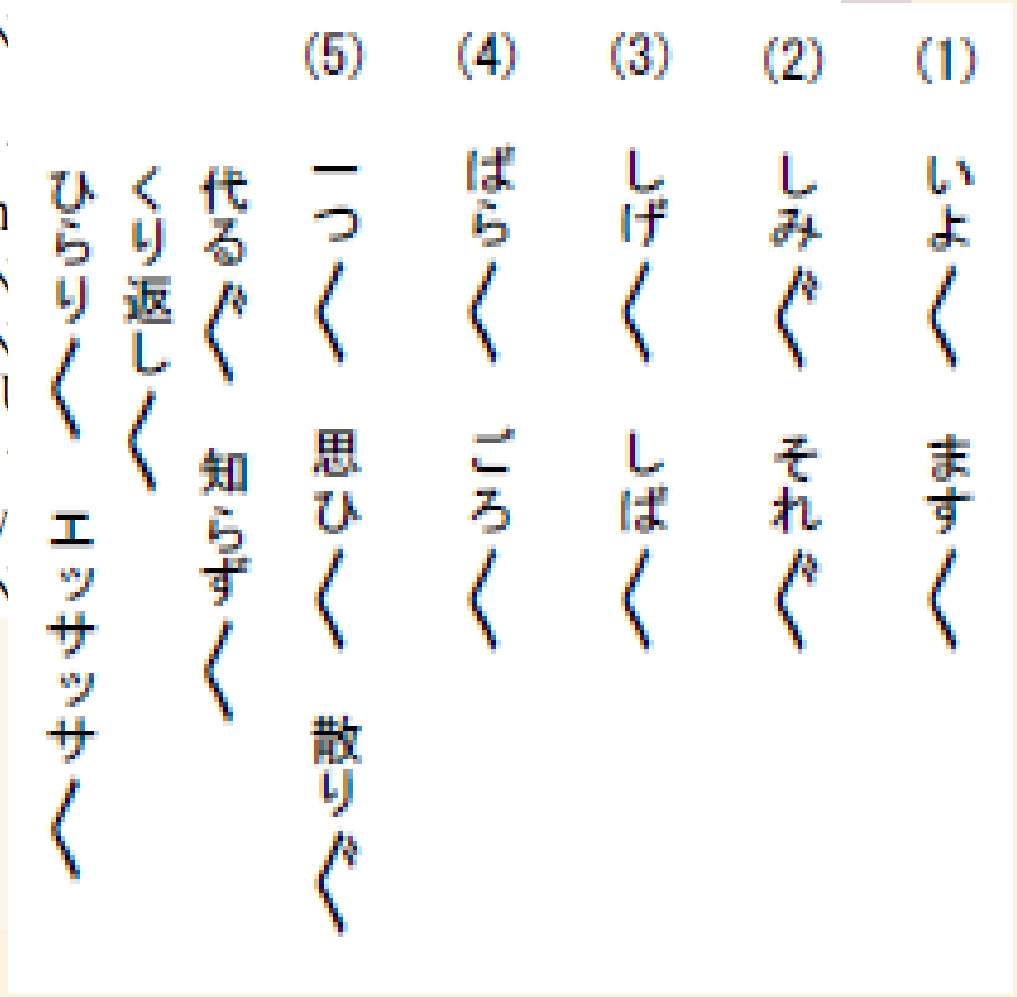
3031 〳 VERTICAL KANA REPEAT MARK

3032 〴 VERTICAL KAN
SOUND MARK
• the preceding
preferred to th

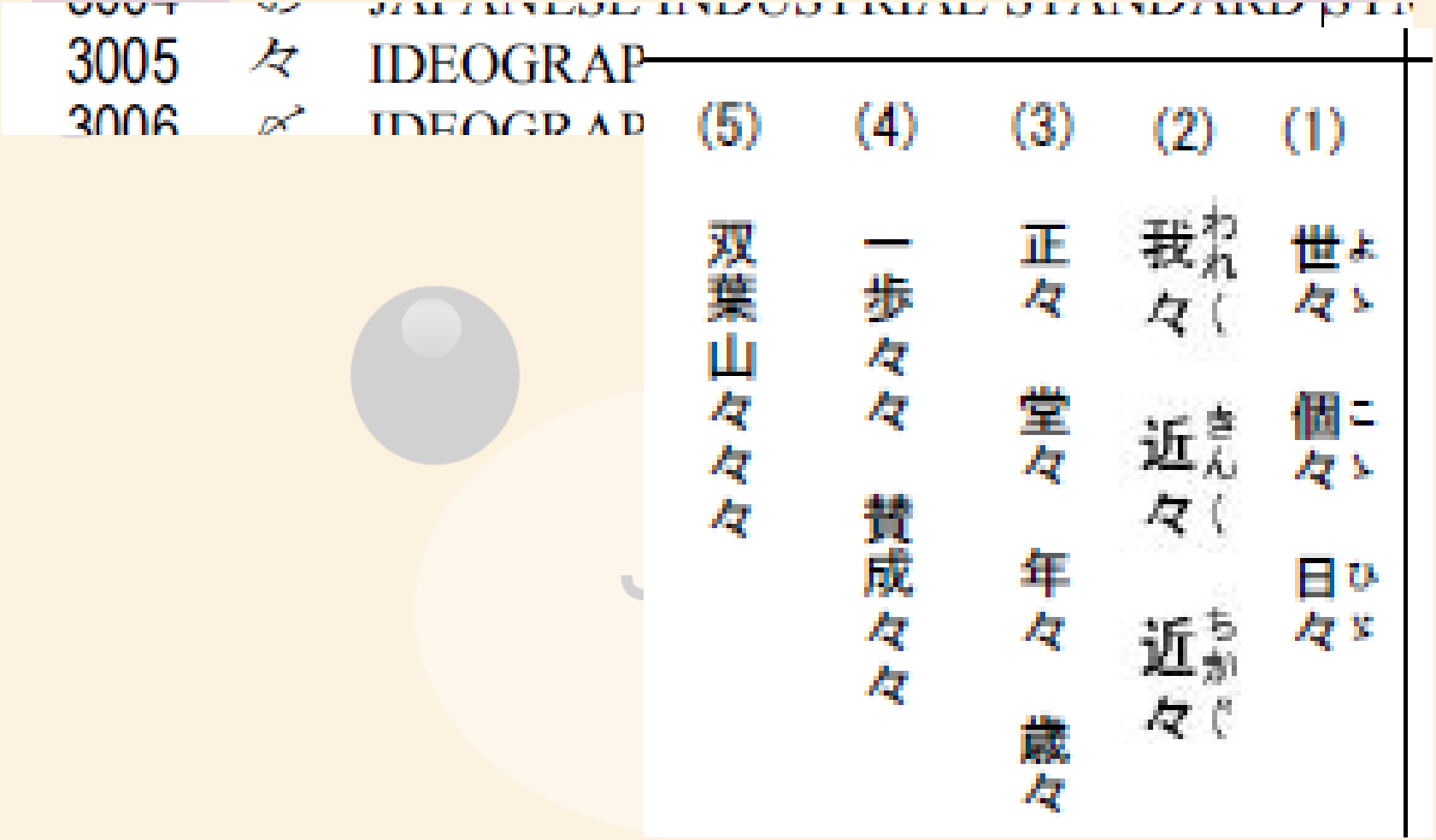
3033 / VERTICAL KAN

3034 /゛ VERTICAL KAN
SOUND MARK
• the preceding
conjunction w

3035 \ VERTICAL KAN



繰り返し文字をどうするの



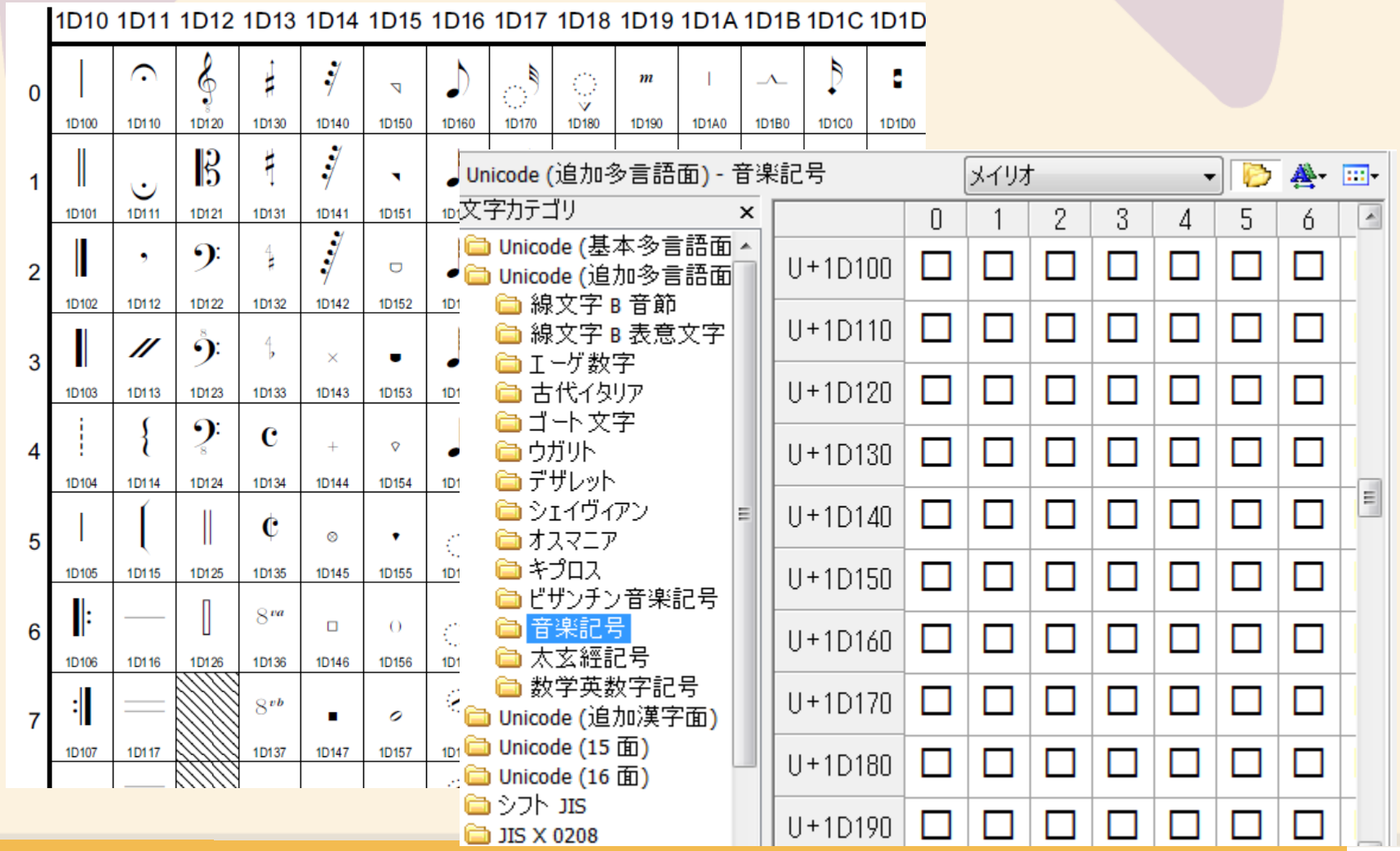
かなをどうするの

- 正假名 vs 現代仮名
  - –言う=言ふ
- 文語体 vs 口語体
  - –てふてふ=ちょうちょう
- 送りがなのゆれ
  - –味わう=味う

かなをどうするの

- 半角 vs 全角
  - –1=1
- 英文スペルの同一字形
  - –D(U+13A0チェロキー)
  - –A(U+0410キリル)

完全なユニバーサルフォントないし・・・



登録されていない漢字は？

- 今昔文字鏡
  - 過去に一度でも出現した文字を分けて登録する方針
  - 字形主義

あたらしい漢字政策が取られたら？

- 日本、韓国、北朝鮮、中国(香港)、台湾、ベトナムあたりが現在の漢字ユーザ
- 国の施策で漢字の省略を奨励したらどうするの？
- 字体は同じだけど、字形が大きく変わる
- その字形は別の国で使っている。
- →変更できない。
- 国別主義で解決できるのか？
  - 同一字形の別コードはフィッシングを生む

日本語は生きている。今後も入れ替え、変更は発生する。

固定化して考えてはならない。

Microsoft

ready
for a new day

Microsoft Office

Windows Vista

Microsoft Exchange Server 2007

参考文献など

Michel Caplan(International Fundamentals team)

http://blogs.msdn.com/michkap

JIS X 0213:2004 対応と新日本語フォント「メイリオ」について

http://www.microsoft.com/japan/windows/products/windowsvista/jp_font/default.mspx

国語審議会

http://www.mext.go.jp/b_menu/shingi/12/kokugo/index.htm

青空文庫(当用漢字表など)

http://aozora.gr.jp/kanji_table/

言葉言葉言葉

http://members.jcom.home.ne.jp/w3c/

Unicode 表

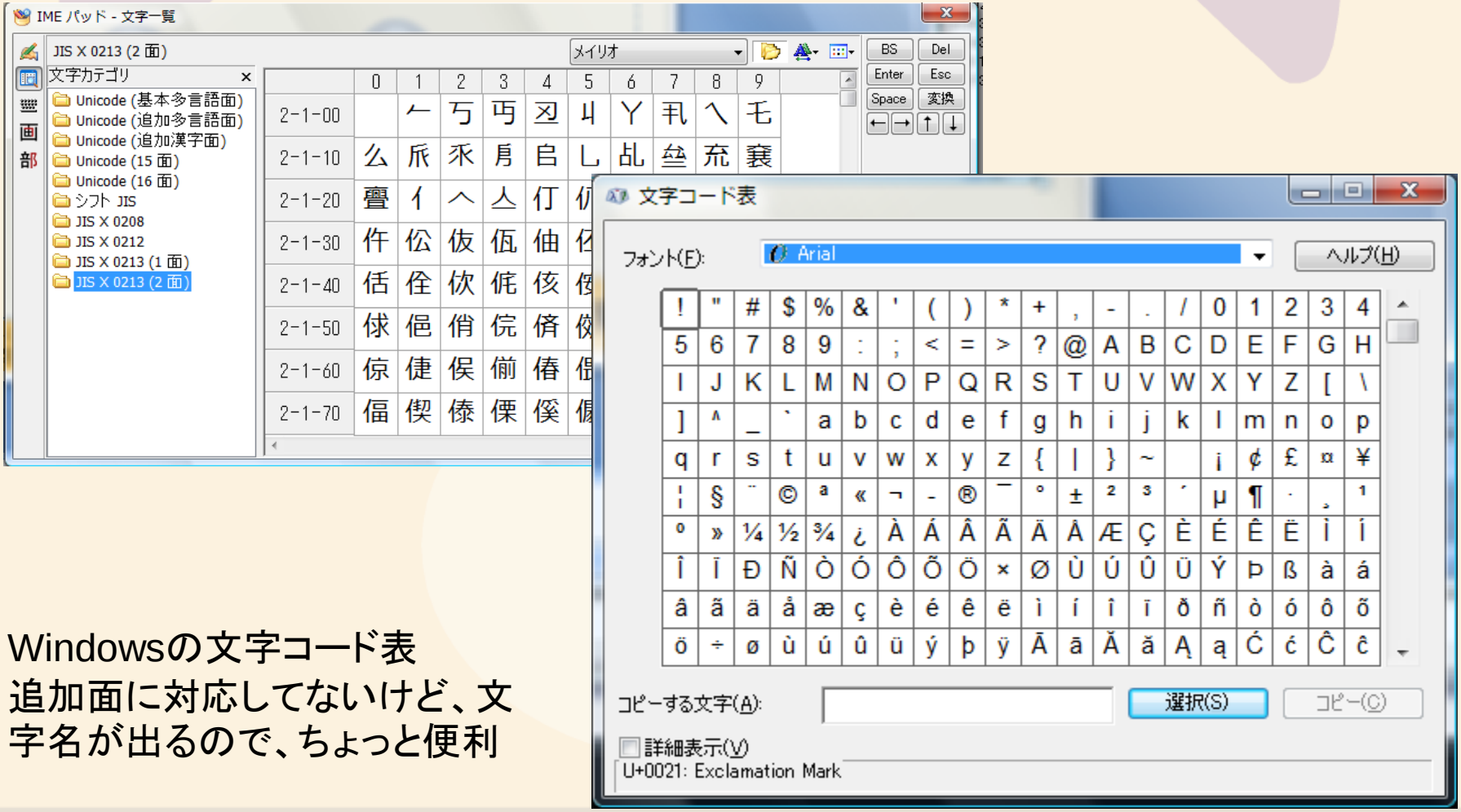
http://www.unicode.org/charts/

国語表記の基準

http://www.bunka.go.jp/kokugo/frame.asp?tm=20070409103237

文字コード表に親しもう

IME2007の文字コード表は秀逸



Windowsの文字コード表
追加面に対応してないけど、文字名が出るので、ちょっと便利