

日本語とコンピューターの半世紀

～当事者として紆余曲折を振り返る～

JEPA セミナー

2022年3月2日(水)

加治佐俊一

本日の内容

2022年3月2日 加治佐俊一氏：日本語とコンピューターの半世紀
～当事者として紆余曲折を振り返る～

2022.02.08

半世紀を振り返るとコンピューターは飛躍的な進化と普及を遂げ、日本語処理も飛躍的に進歩しました。講師は1970年に、コンピューターによる日本語処理に初めて触れて興奮しました。就職後の40年間は、主にソフトウェアに従事しました。中でもリコーでの7年間とマイクロソフトでの27年間は、様々な日本語とのかかわりがありました。本セミナーでは、当事者としての想いや紆余曲折を振り返るとともに、残された課題について話をします。技術的な話も含まれますが、多くの方に理解していただけるように、分かり易く説明します。

JISの通称と規格名

通称	JIS規格名	文字数
JIS78	JIS C 6226-1978 (JIS X 0208:1978)	第一水準: 2965, 第二水準: 3384 非漢字: 453, 計: 6802
JIS83	JIS C 6226-1983 (JIS X 0208:1983)	第一水準: 2965, 第二水準: 3388 非漢字: 524, 計: 6877
JIS90	JIS X 0208:1990	第一水準: 2965, 第二水準: 3390 非漢字: 524, 計: 6879
JIS2004	JIS X 0213:2004	第一水準: 2965, 第二水準: 3390 第三水準: 1259, 第四水準: 2436 非漢字: 1183, 計: 11233

日本語の文字とのかかわりを
過去から振り返る

ここからはじまる

小学五年生 1970年 @大阪万博



ソ連館

チェコスロバキア館

[PyCharm](#) by JetBrains
本社:チェコ共和国 首都プラハ

IBM館

アメリカ館

ロシア系ベンダーJetBrainsがウクライナ侵攻を非難

「IntelliJ」「PyCharm」など人気IDE開発

ITMedia News: <https://www.itmedia.co.jp/news/articles/2202/25/news091.html>

ロシア出身のエンジニアらが設立し、統合開発環境「IntelliJ IDEA」や「PyCharm」などを開発するチェコのJetBrainsは2月24日（現地時間）、Twitterで「当社はロシアのウクライナ侵攻を非難します」との立場を示した。



JetBrainsはPython、Java、PHP、C言語、Rubyなどの開発言語に対応したIDEを公開しているソフトウェアメーカー。IntelliJ IDEAやPyCharmはそれぞれJava、PythonのIDEとしてシェアトップともいわれる。米Googleが提供するAndroidのIDE「Android Studio」もIntelliJ IDEAがベース。他にもプログラミング言語「Kotlin」をはじめプログラミング周辺技術を多数手掛けている。

JetBrainsはPython、Java、PHP、C言語、Rubyなどの開発言語に対応したIDEを公開しているソフトウェアメーカー。IntelliJ IDEAやPyCharmはそれぞれJava、PythonのIDEとしてシェアトップともいわれる。米Googleが提供するAndroidのIDE「Android Studio」もIntelliJ IDEAがベース。他にもプログラミング言語「Kotlin」をはじめプログラミング周辺技術を多数手掛けている。

同社はロシアにも3拠点を構えているが、今回のウクライナ侵攻については反対の立場を取っている。

「JetBrainsとして、今回の攻撃を非難します。私たちの心は、私たちの同僚と家族を含むウクライナの人々とともにあります」（JetBrains）

Copyright © ITmedia, Inc. All Rights Reserved.

2022年02月25日 11時30分 公開

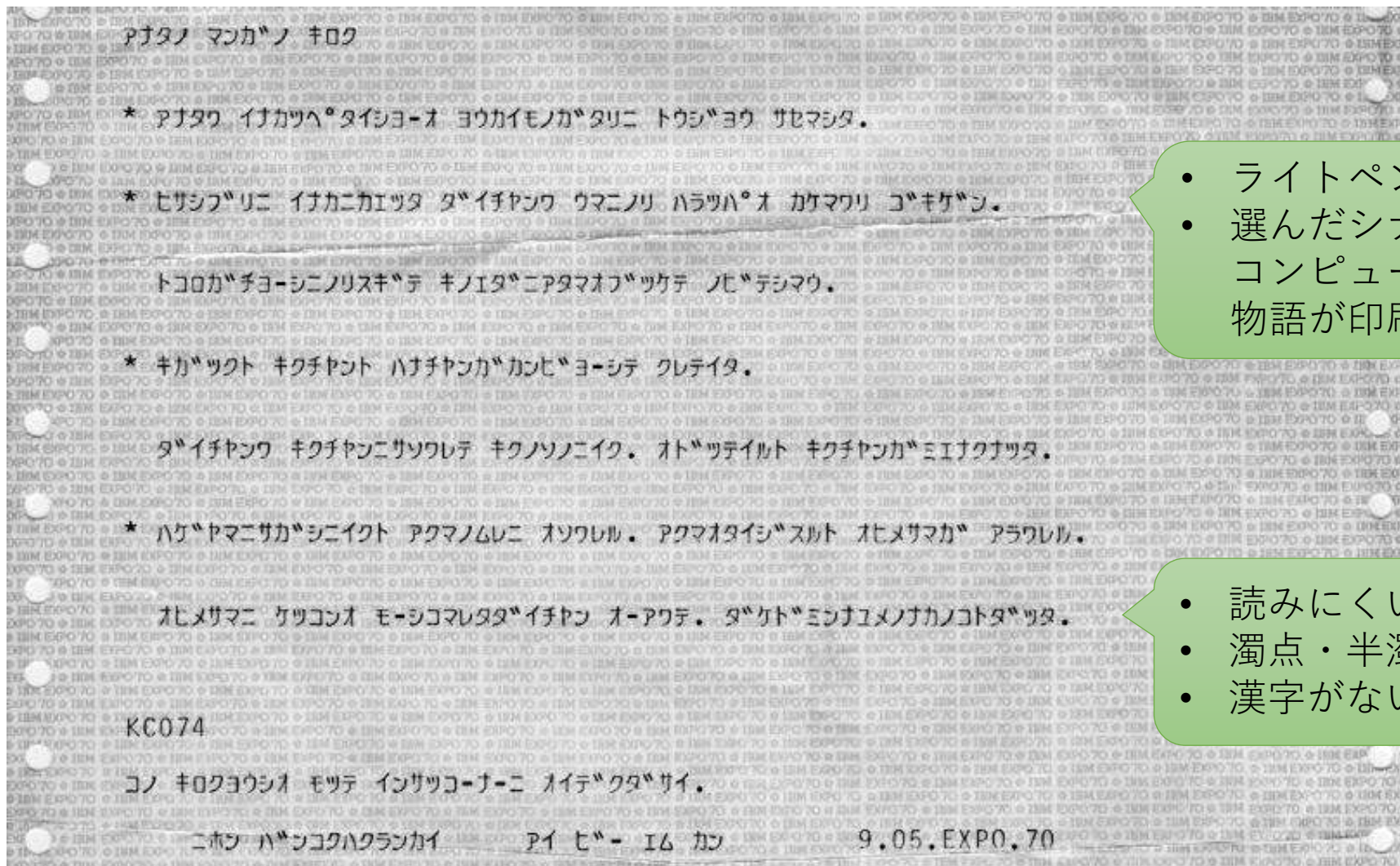
コンピューターとの出会い

1970年 @ 大阪万博



コンピューターによる文字との出会い

IBM館



- ライトペンでシナリオを選べる
- 選んだシナリオで、一連の処理をコンピューターが自動的に処理し、物語が印刷される

- 読みにくい
- 濁点・半濁点が独立
- 漢字がない

リコーでの日々

1982年4月 -

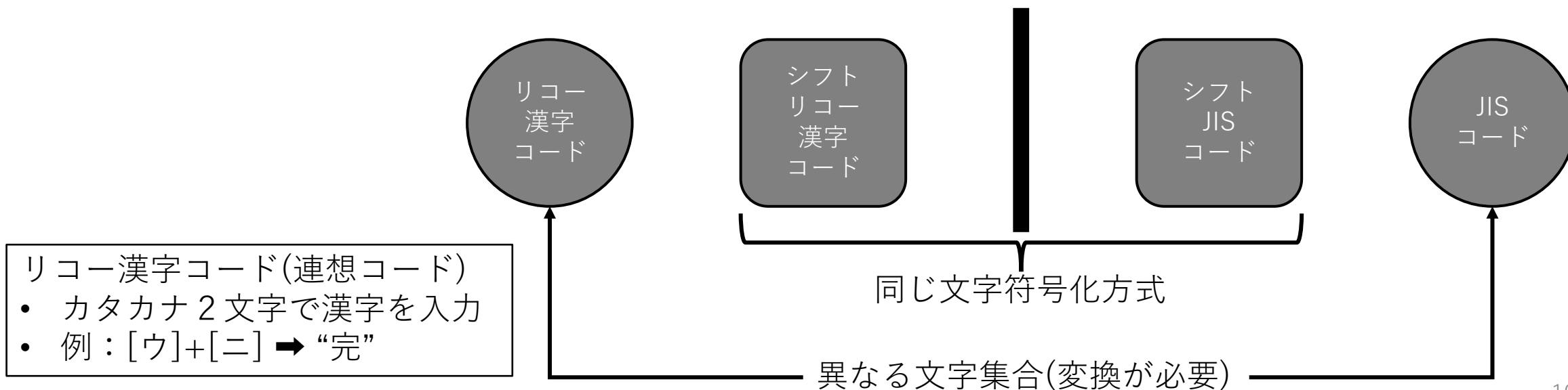
- RICOH(オフコン)の統合ソフトの開発
- IBM PC/AT 互換機のBIOS開発
- Unix Workstationの開発と個別ソリューションの開発

はじめての漢字コード

1982年暮れから1983年 @リコー

- MS-DOS 2.0発表
 - マイクロソフト漢字コード(シフトJIS)
- 米マイクロソフトに1か月半の出張して、表計算ソフトMultiplanをオフコンに移植
- リコー漢字コード ➡ シフトリコー漢字コード

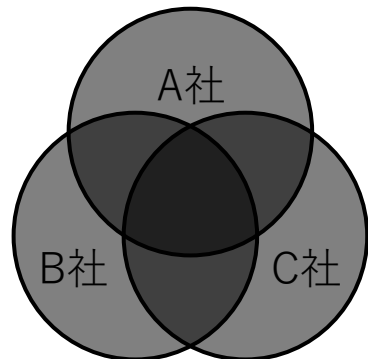
- 左右で異なり変換が必要
- 強い方の標準の集合と方式に巻き取られる



はじめての標準化活動 - Σ プロジェクト

1986年前後 @リコー

- Σ OS (Unix System V) と C 言語 (ANSI C) の標準化委員会に参加
- 符号化方式は、UJIS (Unixnized extended JIS) = “EUC-JP”
- リコーは AT&T と提携し、Unix System V の Workstation を販売
- 各社ハードウェアの違いを標準仕様に追加する
- 自社ハードウェアに「 Σ ワークステーション」の印をつけたい
- ばらばらで生産性が上がらない



- 和 (Union) は崩壊につながる

マイクロソフトでの日々

1989年4月 -

- 「いくらUnixで良いコード書いても、本流には戻せないよ。だったら、OS/2やった方がいいんじゃない。」
- OS/2, Windowsなど
- 技術政策など

一昔前までのWindows の日本語に関連した変更

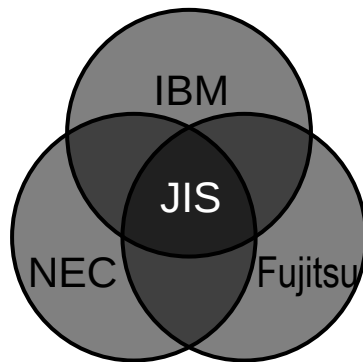
リリース年	プラットフォーム	文字コード・集合	フォント	スタイル
1983 -	MS-DOS2-, OS/2, Windows 2.x, 3.0	Shift-JIS + 各社カスタマイズ	各社カスタマイズ	各社カスタマイズ
1993 -	Windows 3.1, Windows NT 3.1, 3.5, 3.51	MS標準文字セット JIS90, Unicode(10646)	“MS 明朝” “MS ゴシック”	日本語スタイルガイド初版
1995 -	Windows 95			UI のカタカナは、半角に。 「ファイル」は、「ファイル」
1998 -	Windows 98, Windows NT 4.0 SP4, Windows 2000, XP	補助漢字をUCS拡張	“MS UI ゴシック” - カタカナは、全角の幅の85%	半角カタカナを全角カタカナに変更
2007 -	Windows Vista	JIS2004	“メイリオ”	
2009 -	Windows 7			「コンピュータ」から「コンピューター」へ。 音引きルールの変更

注：最近のことは、現役のマイクロソフトの人にお任せします。

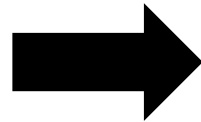
OEM製品だけではないWindowsへ

1992年春 @マイクロソフト

- それまでのマイクロソフトのOS(DOS, Windows, OS/2)はOEM製品
 - NEC PC-98 DOS, Fujitsu FMR, IBM, ...
 - 各社はシフトJISのコード体系の中にメーカー独自の漢字拡張
- アプリは共通化できても、データのやり取りには問題がある
- 日本で共通の漢字キャラクターセット(文字集合)が必要



DOS, Windows 3.0までの世界



マイクロソフト標準キャラクターセット

日本特有の状況

- 全ては救えない
- 非情と妥協

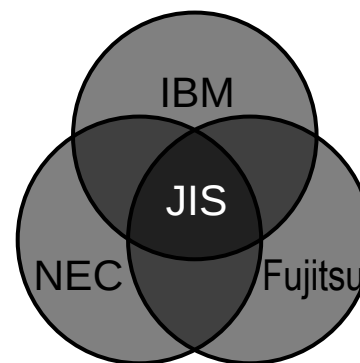
メーカー独自の漢字拡張

- 標準化プロセスを通さないで、各社製品に文字が追加されている
- IBM互換機と言え、元祖はIBMメインフレームのSystem/360
 - 機械語レベル(コンピューターが実行する命令)で互換だが、データは?
- 8ビットパソコンの時代に追加
- 16ビットパソコン普及前のワープロの全盛期の激しい競争下で追加
- 二つの可能性
 - 一旦特定メーカーの製品を使ったら、作成したデータは他のメーカーでは使いにくい(ベンダーロックイン)
 - 特別な意図はなく、必要だと思ったから都度追加した
- 各社の漢字拡張を含んだデータをできるだけ共通に扱えるようにするため
の集合が必要

MS漢字を定義する

1992年春 @マイクロソフト

- 当時最新のJISであるJIS90を採用する
- 新規開発のTrueType Fontで採用
 - 「MS明朝」と「MSゴシック」
- Plan A: 純正のJIS90
 - 観測気球を打ち上げたが、無理だと悟る
- Plan B: 以下を検討してJIS90を拡張
 - アプリ互換性
 - DOS互換ボックスはIBM DOS/V互換
 - Win16サブシステムはDOSと同じ文字セット
 - Win32, OS/2サブシステムは、Win16にならえ
 - 既存のデータはできるだけ使えるようにする
 - OEM各社の状況
 - パソコンシェアはNEC PC-98が圧倒的
 - OASYS(富士通)の拡張部はIBMやNECと重複
 - NECの拡張の一部もIBMと重複
 - 日立、沖電気、東芝、三菱、…
 - DECとCompaqは大丈夫
 - 日本語プリンターのデバイスフォントは…

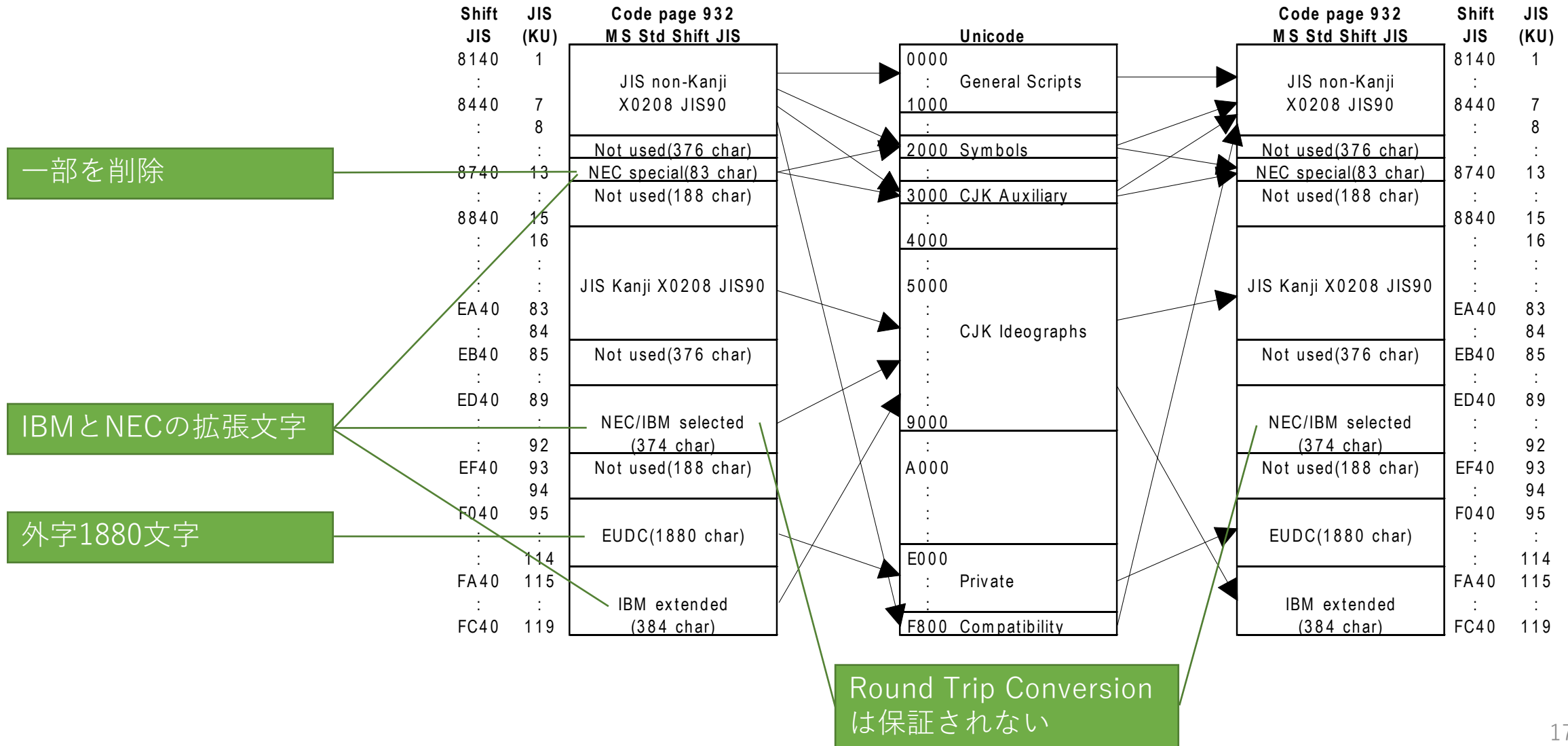


DOS, Windows 3.0まで



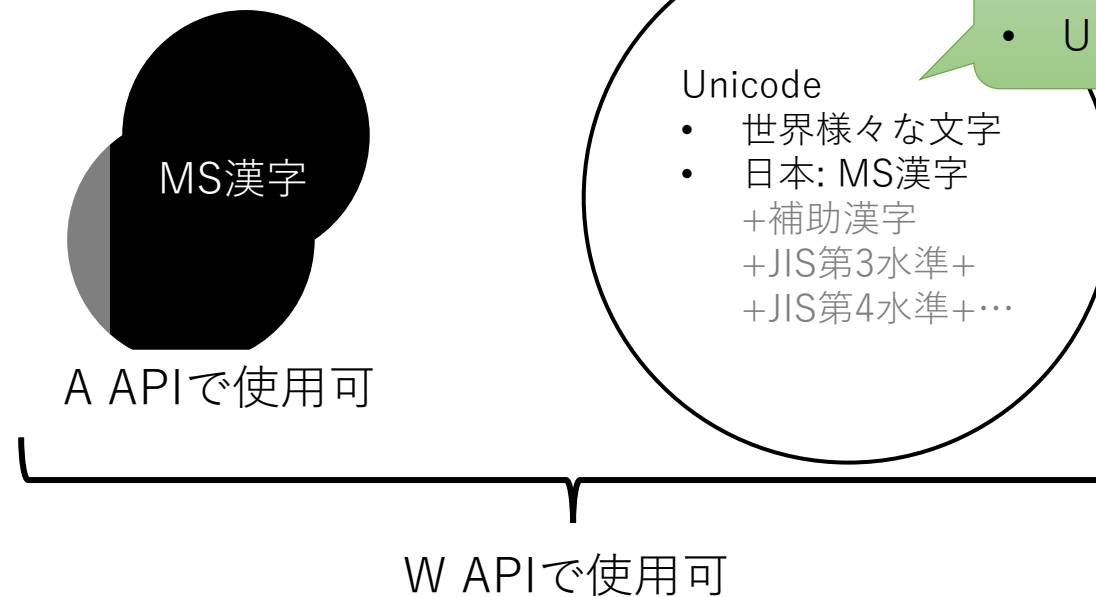
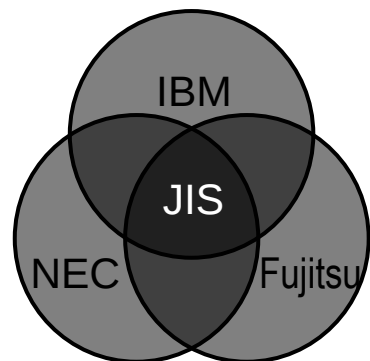
MS漢字を新しく定義

MS漢字とUnicodeとの変換



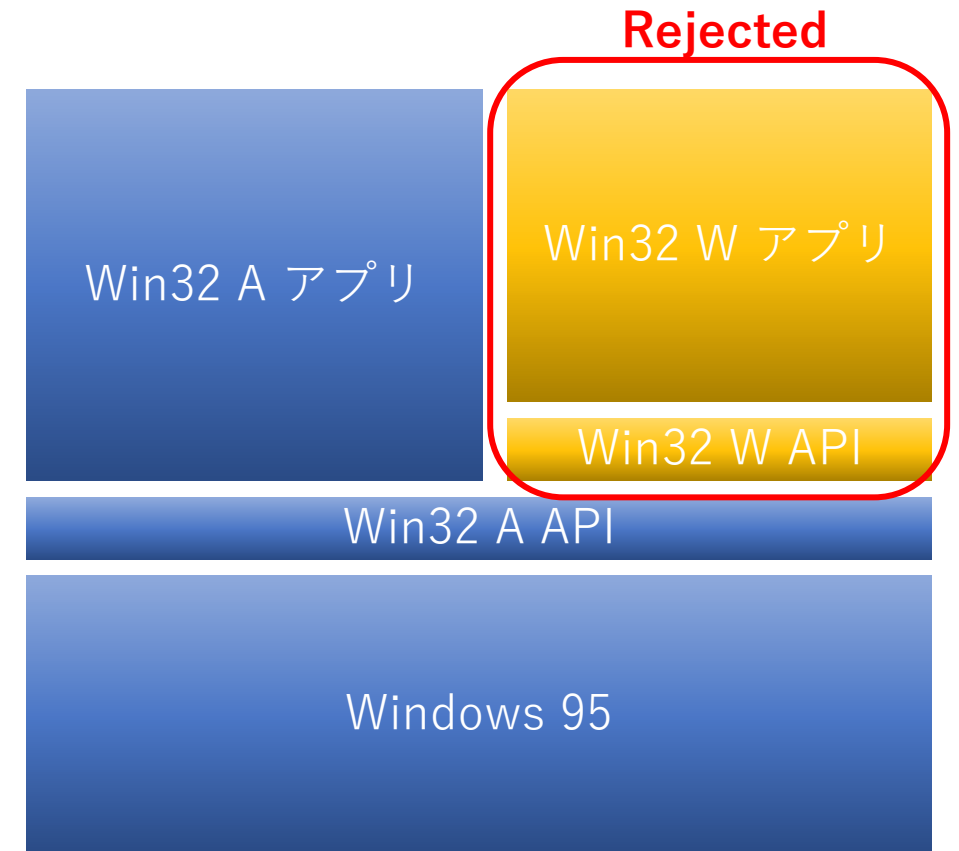
Unicodeと整合させて実装する

- Windows NTは NTカーネルの上にWindowsを32bitで構築
- OS内部の文字処理はすべてUnicodeで処理
- APIはUnicodeの [W API]と従来のANSI(日本語ではシフトJIS)系の[A API]
- MS漢字の将来拡張は無し。文字の追加はUnicodeのみ



Windows 95でUnicodeを部分実装したかった 1993年夏

- 目標
 - W APIをWindows 95でサポート
 - Windows 95とWindows NTで共通の W API 動作のWindowsアプリの普及
 - Windowsの32bit アプリは極力Unicodeベースにしたい
- 実現しなかった
 - 市場でのUnicode対応アプリの普及は遅れた
- 考え方に対する理解はある程度得られた
- 開発スケジュールに対するリスク
- W APIからA APIへの変換に、テーブルとコードに400KBのメモリが必要(当時のPCの搭載メモリは2から4MBが主流)



「歩く」と「けんけんぱ」どちらが速くて楽？

なぜUnicode? 文字処理を「けんけんぱ」に例える

- 歩く
 - 前を見て前に進む
 - 楽で速い
- けんけんぱ
 - 下を見て、次を判断しながら前に進む
 - 判断が入るので遅くて楽ではない

(注)ここでの「けんけんぱ」の「けんけん」は、片足だけを使うのではなく、歩くように両足を交互に使うとする



「歩く」と「けんけんぱ」

「英語」と「日本語」のシフトJISでの処理

- 英語 - アルファベットや数字は1枠で表現

I	T		a	n	d		A	I	
									

- 日本語 - 漢字は2枠(2Byte)を使って表現

I	T	と	A	I	は	進	化	し	て
									
									

```
if (IsDBCSLeadByte(c)) {  
    // 漢字の処理  
}
```



Unicode

「日本語」も普通に歩けるようにしよう

- 日本語 by Unicode - 枠を大きくする

I	T	と	A	I	は	進	化	し	て
									



- Unicode
 - 枠が大きいので、世界のたくさんの文字を入れられる
 - 1枠は一文字 (現在のUnicodeは少し複雑になっている…)
 - 処理も速くて楽

(仮説)「¥」の悲劇

```
C:\Users¥Shunichi>dir /W D:¥tmp
ドライブ D のボリューム ラベルは Data です
ボリューム シリアル番号は 2096-8073 です

D:¥tmp のディレクトリ

[.]                [..]                scratch.txt
1 個のファイル                20 バイト
2 個のディレクトリ 3,056,497,283,072 バイトの空き領域

C:\Users¥Shunichi>
```

- シフトJISの誕生(諸説あり)
 - BASICで漢字を使えるようにする
 - 英字で横80文字縦25行の画面
 - 画面のメモリは1文字に1バイト
 - 漢字の表示には英字を2文字分
 - 漢字は2バイト(英字2文字)に収めたい
 - 漢字は2バイトになるように変換
 - パス(Path)の区切り文字の決定(諸説あり)
 - MS-DOS 2.0でファイル階層構造をサポート
 - Unixでの区切り文字は"/"
 - "DIR /W"で既に"/"が使われている
 - 英語版ではバックスラッシュ(0x5c)
 - 日本語では、JISの0x5cは“¥”
 - ちなみに韓国では₩
- 
- それぞれの設計自体は悪くないが、意図せず組み合わせられて困難な状況が生まれた
 - シフトJISがBASIC以外のところでも使われた。ファイルシステムからネットワークまで
 - シフトJISの2バイト目に区切り文字(0x5c)が含まれる漢字が存在
 - 例：「ソ」「表」「予」「構」…

「¥」の悲劇

日本語の文字列を英語として処理すると問題が起きる

- 英語として処理すると

予		定		表	
97	☞	92	☞	95	☞
☞	5C	☞	E8	☞	5C
¥				¥	



- 海外のソフトをそのまま使うと問題が起きる
- 文字化けが起きる
- 変な名前のファイルができる
- 特にファイル名やマシン名などは、他とのやり取りで問題が起きやすい

- 日本語として処理すると

予	定	表
☞	☞	☞
97	92	95
5C	E8	5C
☞	☞	☞



```
if (IsDBCSLeadByte(c)) {  
    // 漢字の処理  
}
```

- 日本語を識別して正しい処理を行う
 - 手間をかけてプログラムの変更が必要

➡ Unicodeに移行することが賢明

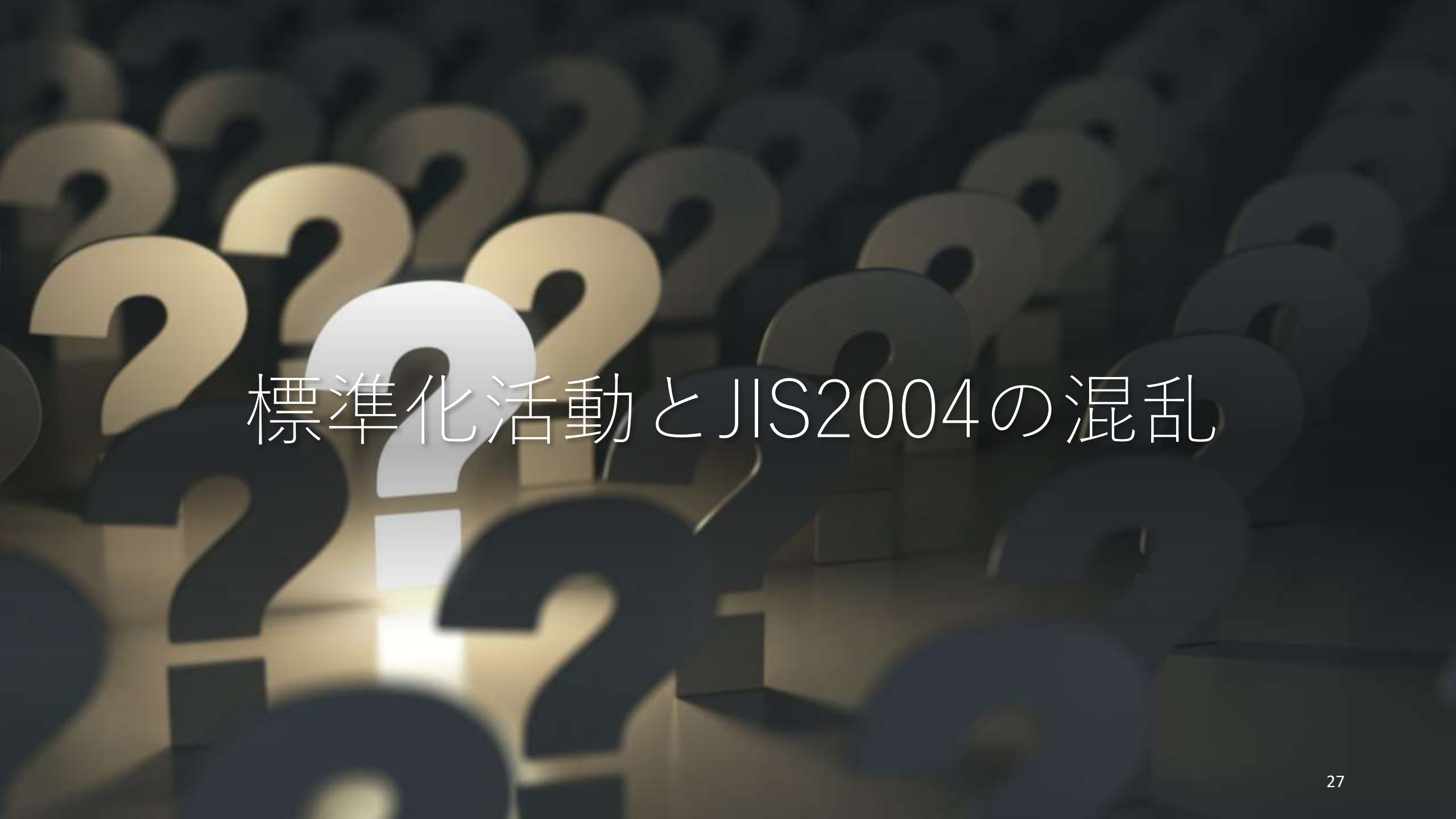
多くのUnicode対応のWindowsが世に出た

- Windows NT 3.1
- Windows NT 3.5
- Windows NT 3.51
- Windows NT 4.0
- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Windows 11

一昔前までのWindows の日本語に関連した変更

リリース年	プラットフォーム	文字コード・集合	フォント	スタイル
1983 -	MS-DOS2-, OS/2, Windows 2.x, 3.0	Shift-JIS + 各社カスタマイズ	各社カスタマイズ	各社カスタマイズ
1993 -	Windows 3.1, Windows NT 3.1, 3.5, 3.51	MS標準文字セット JIS90, Unicode(10646)	“MS 明朝” “MS ゴシック”	日本語スタイルガイド初版
1995 -	Windows 95			UI のカタカナは、半角に。 「ファイル」は、「ﾌｧｲﾙ」
1998 -	Windows 98, Windows NT 4.0 SP4, Windows 2000, XP	補助漢字をUCS拡張	“MS UI ゴシック” - カタカナは、全角の幅の85%	半角カタカナを全角カタカナに変更
2007 -	Windows Vista	JIS2004	“メイリオ”	
2009 -	Windows 7			「コンピュータ」から「コンピューター」へ。 音引きルールの変更

注：最近のことは、現役のマイクロソフトの人にお任せします。



標準化活動とJIS2004の混乱

文字の標準化活動への参加

1994年頃-

- [情報規格調査会](#)の[SC2 符号化文字集合]から参加のお誘い
 - Windowsチームは、業界団体から呼び出されて説明することはあった
 - 委員会に参加して日本の文字を正しいプロセスで標準化する
 - Unicodeと連携したプロセス
 - 後日、情報規格調査会の役員会社にもなった
- 汎用電子情報交換環境整備プログラム
 - 平成14年度～平成20年度
 - 電子政府推進の為に文字情報基盤を構築
 - 経済産業省、内閣官房、総務省、法務省、文化庁が5府省連携
 - 戸籍文字、住民基本台帳文字の同定作業

➡標準化と製品実装は車の両輪

但し、大変なこともある

JIS2004の「なぞ」

2007年 Windows Vista で JIS2004 をいち早く採用

謎	1978: JIS78
謎	1983: JIS83,90
?	2004: JIS2004

字体の議論となった漢字の例

～ 「しんにゅう」と「しょくへん」～

遡	遡	ソ / さかのぼる
遜	遜	ソン
謎	謎	なぞ
餌	餌	ジ / え・えさ
餅	餅	ヘイ / もち

標準 vs. 市場

標準原理主義

- 常用漢字表内の一貫性
➡ 字体は変更

謎 1978: JIS78

辻さんが葛飾で餅を食べた。

市場原理主義

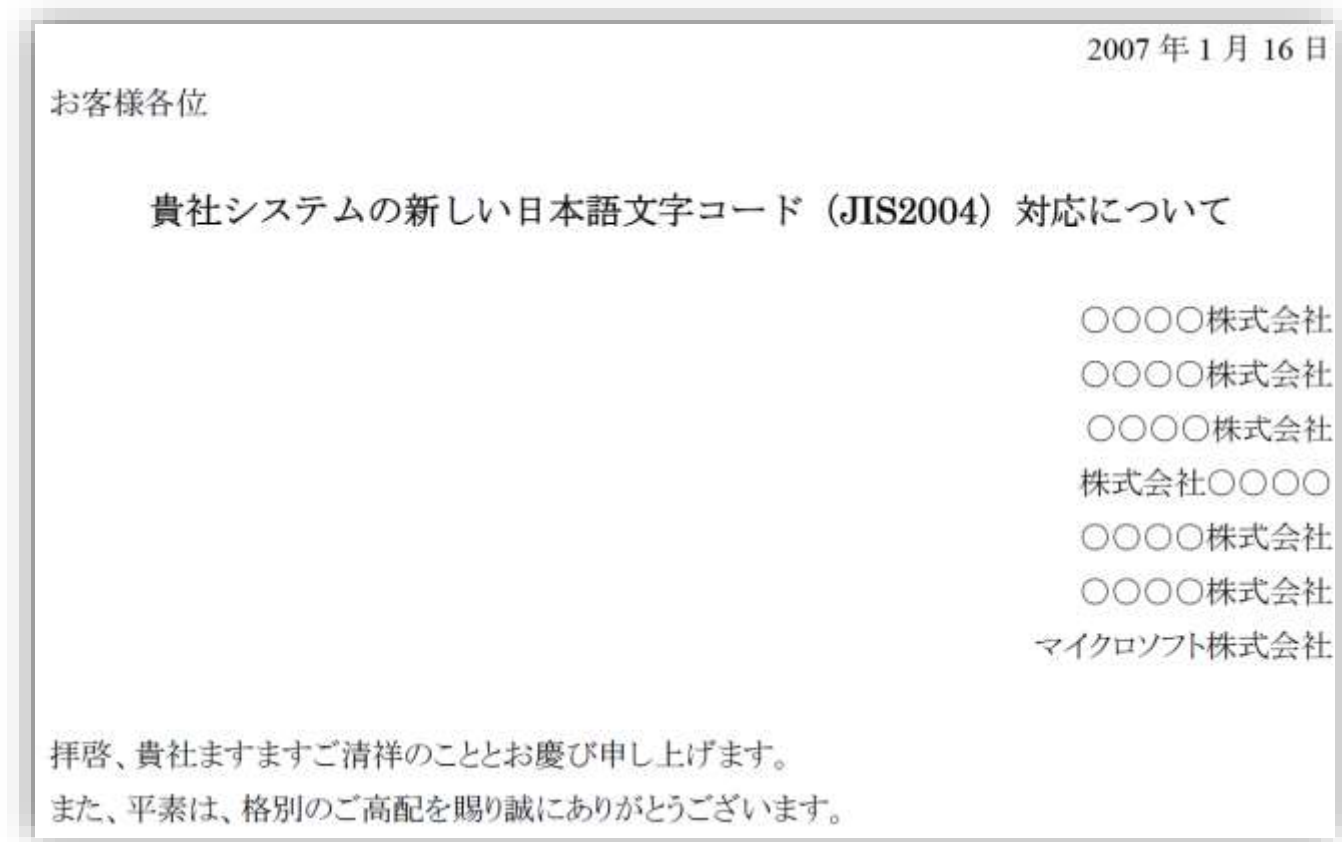
- 文字の安定
➡ 字体は維持

謎 1983: JIS83,90

辻さんが葛飾で餅を食べた。

字体の変更による混乱

2007年1月16日付お客様向け文書



時期を前後し、政府関係、業界団体、各企業から同様な顧客向けの文書や、膨大な技術情報が作成され、配布された。

字体の変更による混乱

2007年1月16日付お客様向け文書

JIS2004 (JIS X 0213:2004) は、2000 年12 月に発表された国語審議会答申の趣旨に従って、2004 年2 月に経済産業省により改正されたJIS 漢字コード規格です。JIS 2004 に対応するコンピュータ システムでは、従来のJIS90 (JIS X 0208) で定められた第1 水準漢字 (2,965 文字) と第2 水準漢字 (3,390 文字) に加え、第3 水準漢字 (1,259 文字) と第4 水準漢字 (2,436 文字) を利用することが可能となります。コンピュータ システムのJIS 2004 対応は、日本の国語施策への対応や人名用漢字の利用、将来的なシステム運用コストの低減、さらに、利用者には「正しい（標準）文字が書ける」や「より多くの文字を扱える」等の提供できるメリットがあると考えられています。

JIS2004 では、JIS90 で実装されている利用頻度の高い第1 水準漢字や第2 水準漢字を用いた場合、一部の字体変更（百数十文字）を除き互換性に関する問題は発生しません。

解決方法は？

- 時が解決する
- IVS技術の普及
 - IVS(Ideographic Variation Sequence): 異体字処理の仕組み
 - [一般社団法人文字情報技術促進協議会 \(moji.or.jp\)](http://moji.or.jp)



過去

現在

未来

- 時は待ってくれない
 - このような一つひとつの混乱が、日本国内のDX化の足を引っ張る。(本当に申し訳ありません)

理系と文系、専門家と一般人で Computer の使い方が違う？

長音表記の話

外来語をカタカナ表記する際、語尾にのばす音が含まれていた場合、長音符号「ー」を付与するかしないかについて、指針となるルールが複数存在

長音表記の二つのルール

学術・JIS 記述ルール

- JIS Z 8301 - 規格票の様式及び作成方法
- 学術用語に規定されていない用語について、「2音の用語は長音符号を付け、3音以上の用語の場合は（長音符号を）省くことを“原則”とする」主旨の規定
- 自然科学/工学系、および学術論文の分野で主に利用

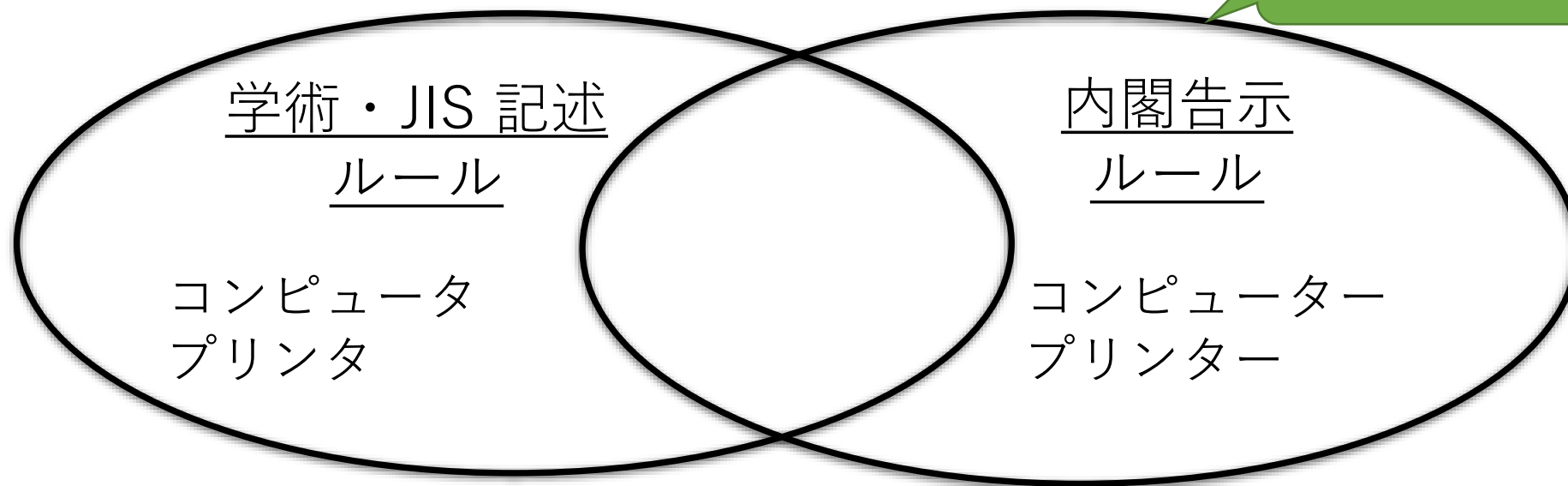
内閣告示ルール

- 国語審議会の報告を基に告示された1991年6月28日の内閣告示第二号をベースにしたルール
- 英語由来のカタカナ用語において、言語の末尾が-er、-or、-arなどで終わる場合に長音表記を付けることを推奨
- 新聞や放送はこの『外来語の表記』に準拠。長音符号を付けることを原則とする。ただし、慣用により省略する例外も認められている

長音表記の論点

2009年 Windows 7で変更

- Computer: コンピュータ または コンピューター
- Printer: プリンタ または プリンター

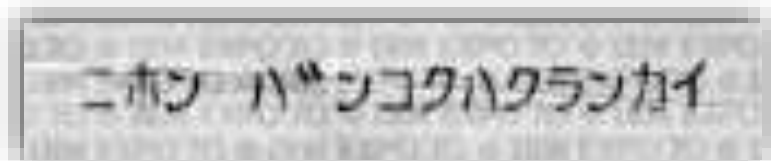


Windows 7で内閣告示ルールに変更

使われ続ける全角と半角

半角・全角って何ですか？

いまだに残る古い慣習、日本文化遺産？



(全角)
使用できる漢字はJIS第1水準及び第2水準までとします。

(半角)
氏名の読み方をカタカナで入力してください。

基本情報 / Applicant Information

氏名 (漢字) / Full Name (kanji)	姓 / Family Name	加治佐	名 / First Name	俊一	ミドル / Middle
	Name	K	(全角)		
	使用できる漢字はJIS第1水準及び第2水準までとします。エラーになる場合、または漢字が無い場合は代替の文字 (カタカナを含む) を入力してください。原則として、入学後の氏名は出願時に届け出た氏名とします。 / Please enter your name using the Roman alphabet if Chinese characters are not used for your name.				
	ミドルネームは頭文字のみ入力してください。 / Please enter only the first initial of your middle name.				

氏名 (カナ) / Full Name (katakana)	セイ / Family Name	カザサ	メイ / First Name	シュイチ	ミドル / Middle
	Name	K	(半角)		
	氏名の読みをカタカナで入力してください。 / Please enter your name using the Roman alphabet if you do not use katakana in your name.				
	ミドルネームは頭文字のみ入力してください。 / Please enter only the first initial of your middle name.				

今でも使われる文字たち

- 全角記号 # \$ %...
- 全角数字 1 2 3...
- 全角アルファベット A B C...
- 半角カタカナ ア イ ウ...
- 意味も無く入力させるのはやめよう
- データの整形に余計な手間がかかる、
データ処理のミスも増える

	FF0	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FFA	FFB	FFC	FFD	FFE
0		0	@	P	`	p	)	ー	タ	ミ	HW HF	ぢ			¢
1	!	1	A	Q	a	q	。	ア	チ	ム	ㇿ	ロ			£
2	"	2	B	R	b	r	「	イ	ツ	メ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
3	#	3	C	S	c	s	」	ウ	テ	モ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
4	\$	4	D	T	d	t	、	エ	ト	ヤ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
5	%	5	E	U	e	u	・	オ	ナ	ユ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
6	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
7	'	7	G	W	g	w	ア	キ	ヌ	ラ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
8	(	8	H	X	h	x	イ	ク	ネ	リ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
9	)	9	I	Y	i	y	ウ	ケ	ノ	ル	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
A	*	:	J	Z	j	z	エ	コ	ハ	レ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
B	+	;	K	[	k	{	オ	サ	ヒ	ロ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
C	,	<	L	\	l		ヤ	シ	フ	ワ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
D	—	=	M	]	m	}	ユ	ス	ハ	ソ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
E	.	>	N	^	n	~	ヨ	セ	ホ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ
F	/	?	O	_	o	《	ツ	ソ	マ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ	ㇿ

まとめ

- いろいろあるけど、コンピューターで日本語を扱える環境は整備されてきた。
- 課題もあるが、解決策も多いので、積極的に取り入れましょう。
- データは重要です。数値情報だけでなく、文字情報も容易にデータ交換ができ、活用できるように意識しましょう。

A large, white, three-dimensional question mark is mounted on a bright orange wall. The question mark is positioned in the center-left of the frame. The wall has a textured, slightly mottled appearance. Below the question mark, a white baseboard runs along the wall. The floor is made of light-colored wooden planks. A large, white, curved line separates the orange wall from a dark gray background on the right side of the image.

Q&A

感謝

コンピューターを使っている方々

標準化活動にかかわる方々

コンピューターやデバイスのメーカーの方々

プログラマーや技術サポートエンジニアの方々

元および現マイクロソフトの方々

取り上げなかった話題

- 縦書き処理
- Font の話
- Internet Browser の話
- 日本語Keyboardの話
- かな漢字変換の話
- ここ12年程の話題
- 絵文字統一の試み
- TRONとの出会い
- オフィスより徳島に近い

文字施策と文字コード標準の歴史

～ 文字施策、標準 (JIS と ISO/IEC)、製品での実装の関係 ～

