文字情報技術促進協議会

JCITPC

行政事務標準文字を使いこなそう

2025年12月12日

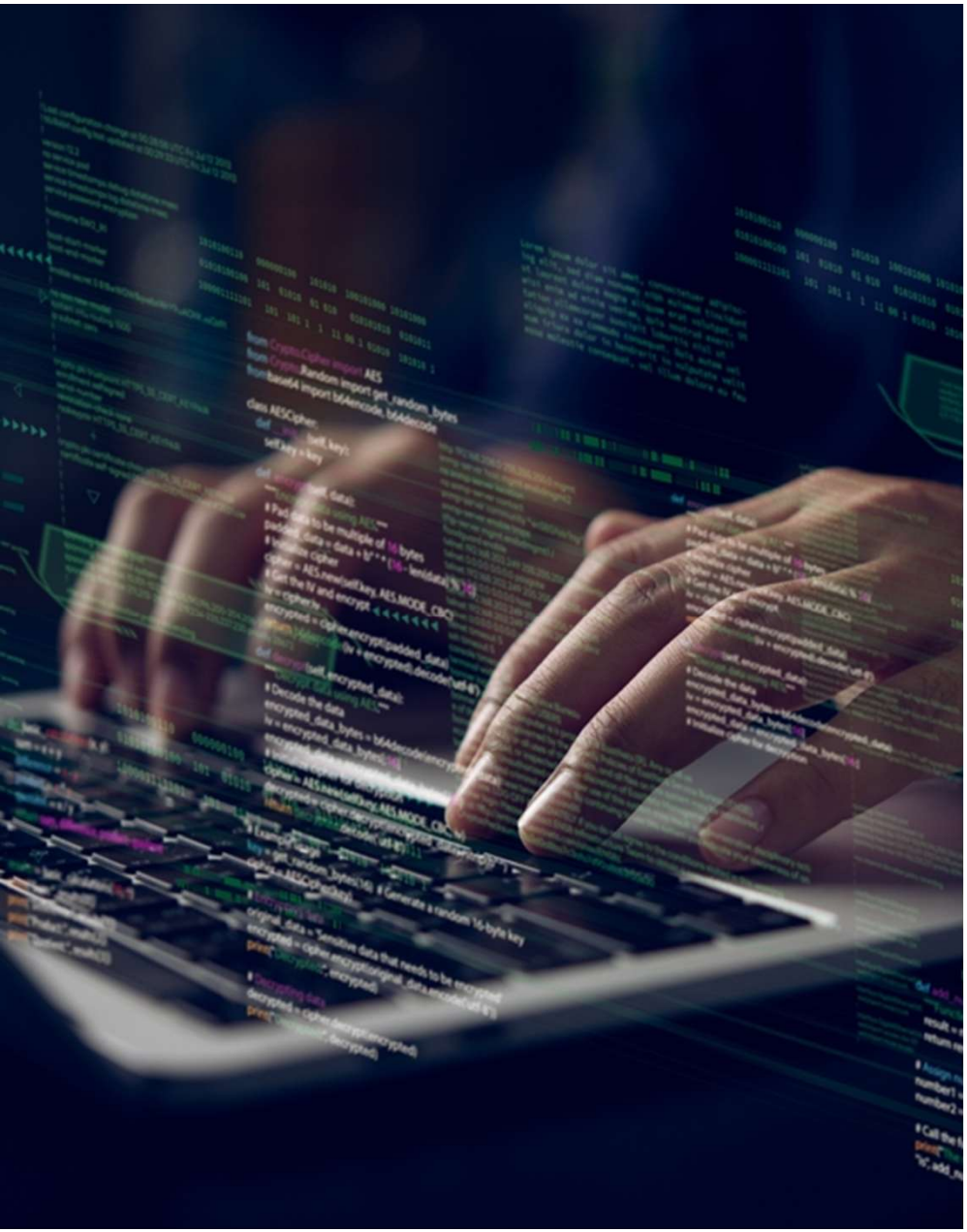
日本電気株式会社

袴田 博之

本資料の内容は講師個人の見解に基づくものであり、講師の所属する組織の公式見解ではありません

Agenda

- 「使いこなす」とは
- 基礎知識
  - 行政事務標準文字
  - 国際標準
  - 公的証明と本人確認
- 実践のポイント
- 今後の展望



「使いこなす」とは？

疑問

- 調達側
  - どういう背景/経緯のものなのか？
  - 今までと何が違うのか？外字から解放してくれる救世主なのか？
  - どこで使う/使われるのか？
  - どこで使わない/使われないのか？
- 実装側
  - 仕様の範囲は？
  - 設計上のポイントは？

本日のゴール

最初に申し上げておくと
スッキリした答えがあるわけではありません

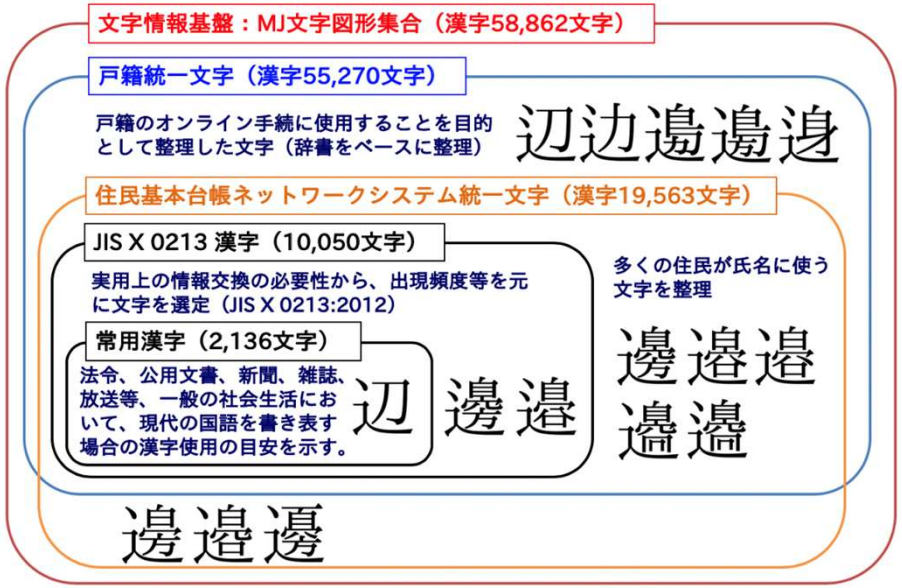
「使いこなす」状態に至るための判断材料を
少しでも多く持って帰っていただければ

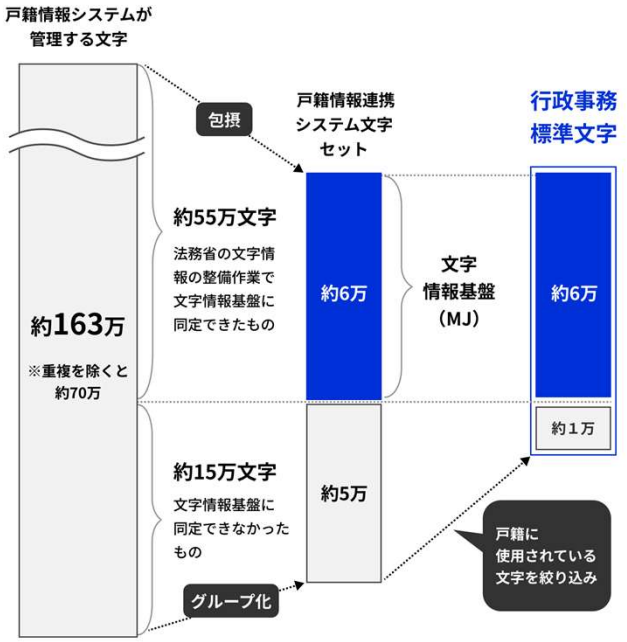
文字情報技術促進協議会
JCITPC

基礎知識(1)
行政事務標準文字

行政事務標準文字の構成

- 行政事務の標準はJIS X 0213、戸籍氏名文字等に行政事務標準文字
- 文字情報基盤約6万文字に、戸籍由来の文字字形 約1万文字を国(デジタル庁)が追加したもの
- ベース・レジストリ共通の文字集合
- 追加の約1万文字の国際標準化に取り組んでいる

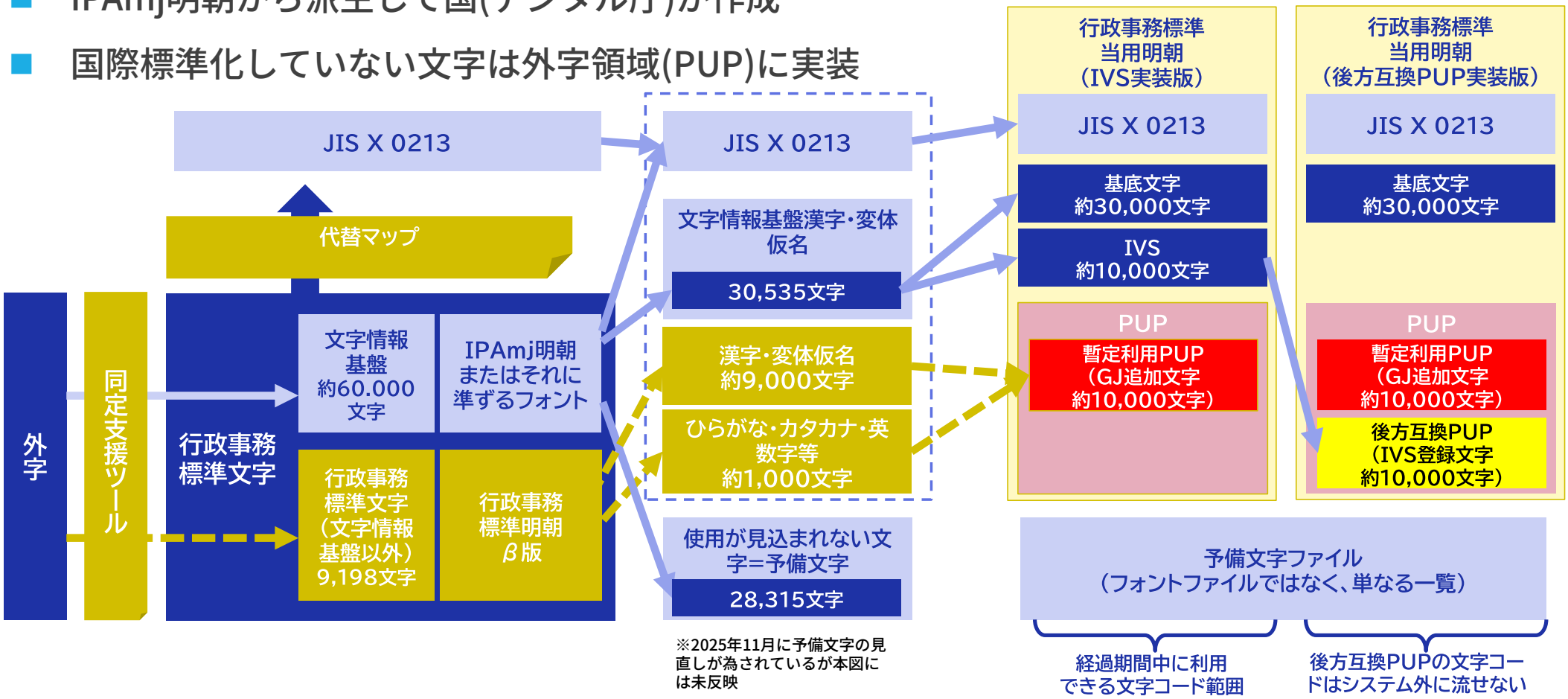




出典:https://www.digital.go.jp/policies/local_governments/character-specification

行政事務標準文字のフォント実装

- IPAmj明朝から派生して国(デジタル庁)が作成
- 国際標準化していない文字は外字領域(PUP)に実装



抽象度の違い

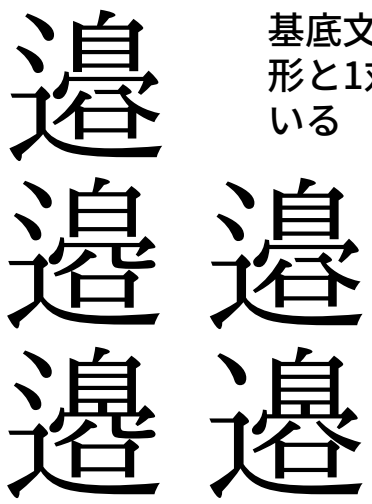
JIS X 0213
文字概念集合

字形の差異を
概念として包含

概念を例示しているだけで、この字形を「正しい」としているわけではない

文字情報基盤/行政事務標準文字
文字図形集合

辞書典拠のある文字概念と、その字形バリエーションという整理

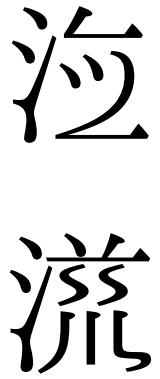


基底文字もMJ文字図形と1対1で紐付けている

IVSを利用して
字形バリエーション
を表現

行政事務標準文字追加分
固有名詞用字形集合とでも

辞書典拠どころか、どの文字の異体字かの判断すら難しい字形を含む

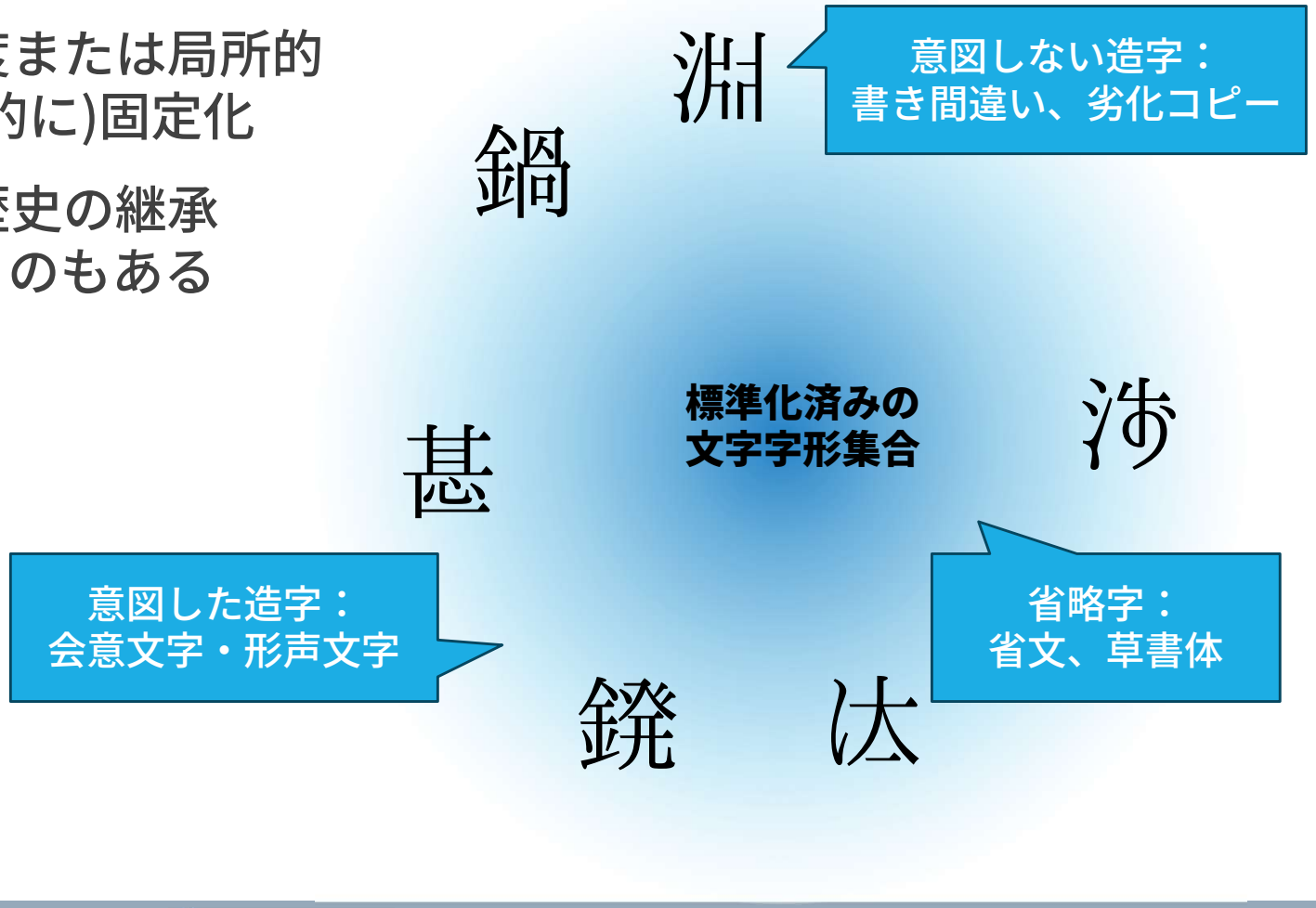


準拠フォントならどれでもOKとすべき
(デザイン差すら問題になる用途には利用すべきでない)

当協議会が提供するフォント/デジタル庁が提供するフォント以外の互換フォント利用は注意が必要

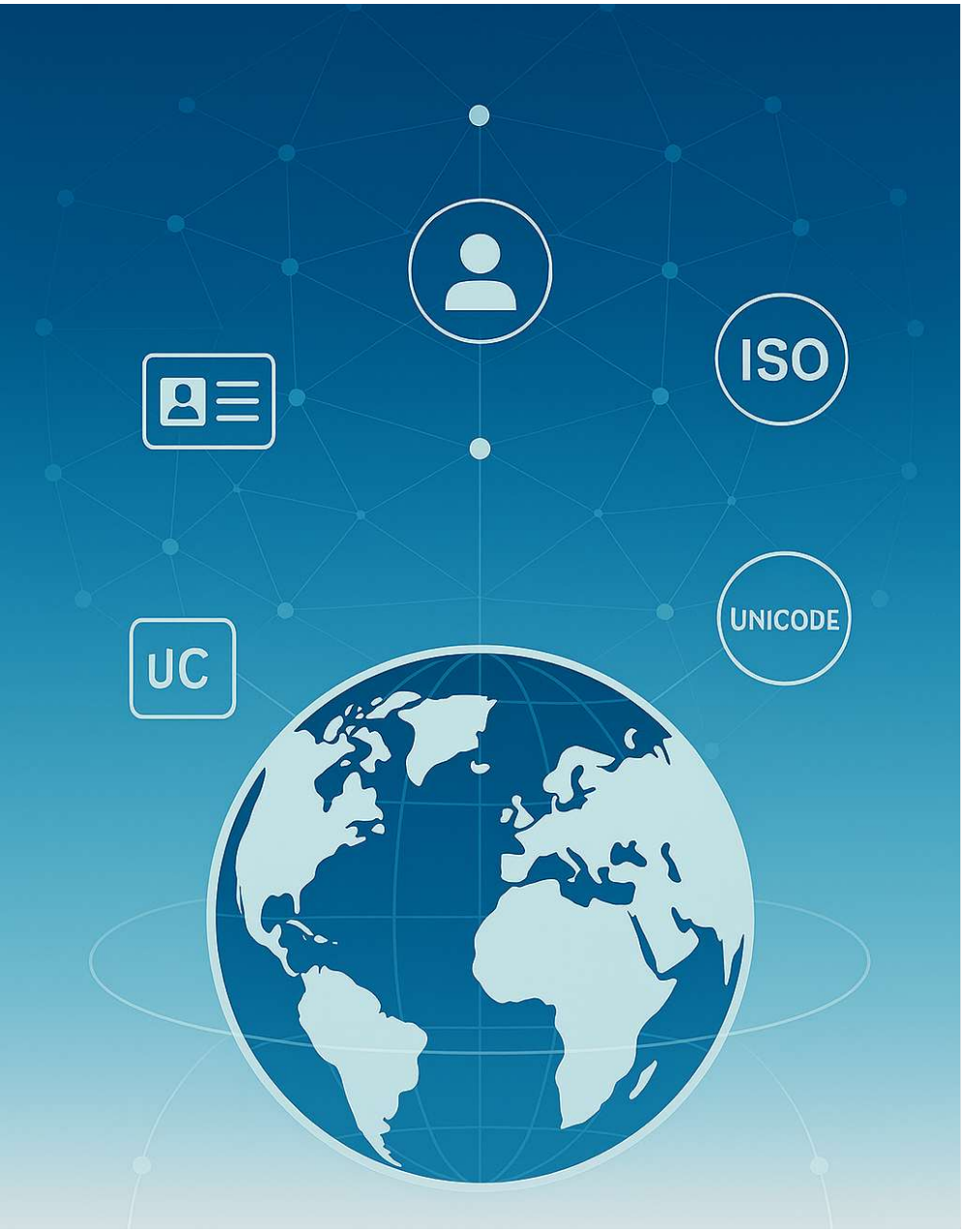
文字字形が増えてしまった構造

- 淘汰されるはずの、低頻度または局所的なバリエーションが(電子的に)固定化
- 造語能力を有する漢字＝歴史の継承
  →意図してその形、というのもある



ここまでのまとめ

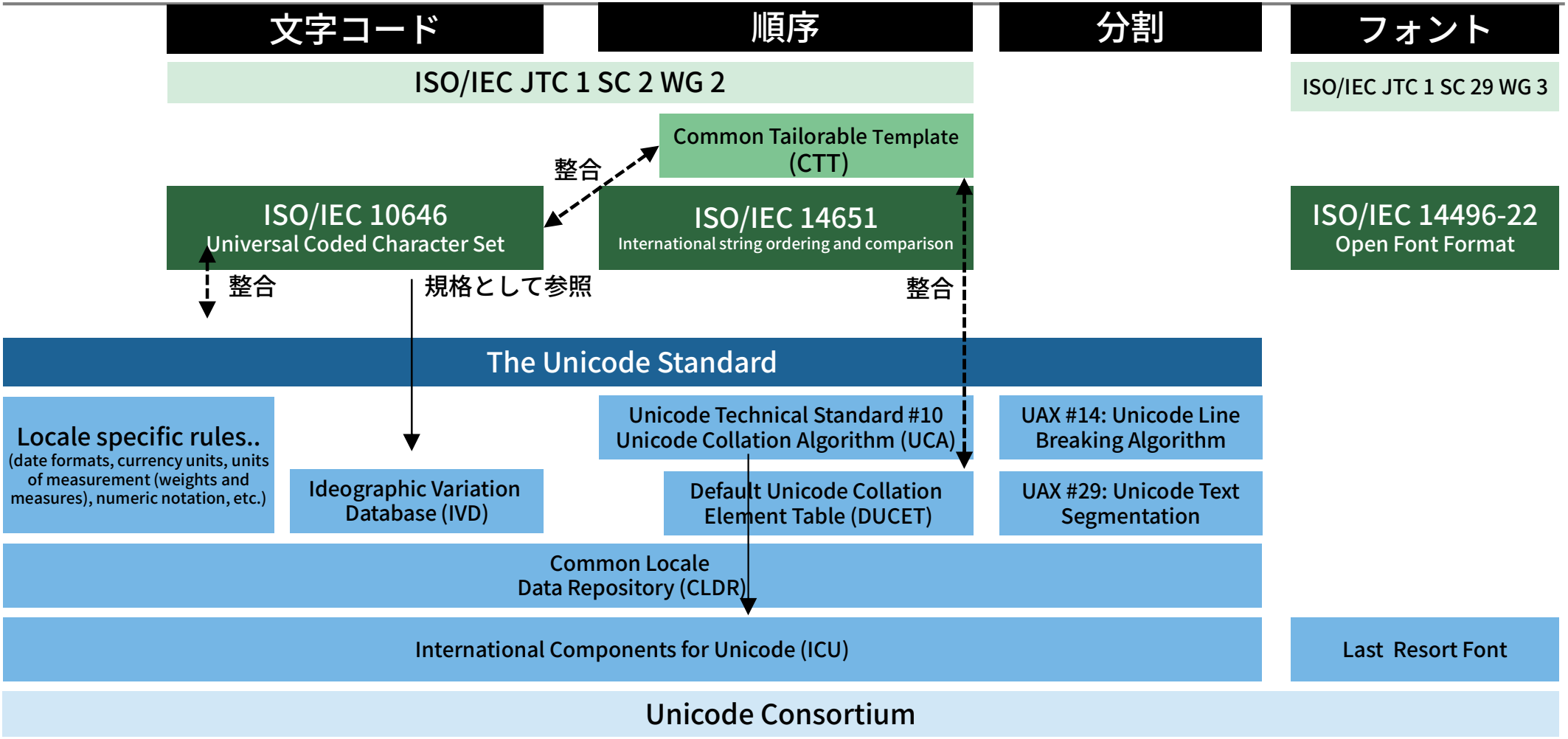
- 行政事務標準文字はJIS X 0213とともに行政事務とベース・レジストリで用いる
- 行政事務標準文字 追加分の国際標準化はこれから順次進む
- コードが同じでも、指し示す文字字形の抽象度が同じとは限らない
- 異体字＝書き間違いのバリエーション、ばかりではない



文字情報技術促進協議会
JCITPC

基礎知識(2)
国際標準

国際標準の関係性


文字コード
順序
分割
フォント
ISO/IEC JTC 1 SC 2 WG 2
ISO/IEC JTC 1 SC 29 WG 3
Common Tailorable Template (CTT)
整合
ISO/IEC 10646
Universal Coded Character Set
ISO/IEC 14651
International string ordering and comparison
ISO/IEC 14496-22
Open Font Format
整合
規格として参照
整合
The Unicode Standard
Locale specific rules..
(date formats, currency units, units of measurement (weights and measures), numeric notation, etc.)
Ideographic Variation Database (IVD)
Unicode Technical Standard #10
Unicode Collation Algorithm (UCA)
Default Unicode Collation Element Table (DUCET)
UAX #14: Unicode Line Breaking Algorithm
UAX #29: Unicode Text Segmentation
Common Locale Data Repository (CLDR)
International Components for Unicode (ICU)
Last Resort Font
Unicode Consortium

国際標準の関係性

文字コード	順序	分割	フォント

字形	符号化文字 (UCS)	IVS	文字符号化形式 UTF-8 (8bit単位)	UTF-16 (16bit単位)	UTF-32 (32bit単位)
亜	U+4E9C		E4 BA 9C	4E9C	00004E9C
丈	U+2000B		F0 A0 80 8B	D840 DC0B	0002000B
邊	U+908A		E9 82 8A	908A	0000908A
邊	U+908A	U+E0109	E9 82 8A F3 A0 84 89	908A DB40 DD09	0000908A 000E0109

は 306F
ば 3070
ぱ 3071

か 304B
が 304C
が゚ 304B 309A

が゚ 304B 309A

改行するときに、

か 304B
゛ 000D
゜ 000A
゚ 309A

となるのはダメ

か 304B
＋
゚ 309A
↓
が゚ 304B 309A

文字概念を
データとしてどう表現するか？

整列したときに
どちらが先に来るか？

分割して良い
位置はどこか？

合成した字形の
出し方は？

国際標準の構造

- 文字コード
  - 名前と箱
  - エンコーディング
  - 部分集合
- 順序
- 分割
- リファレンス実装
- フォント実装

文字コード - 名前と箱

- 名前：文字概念に対する名前付け
- 箱：コードポイント
  - 10646：文字概念に符号=文字コード(箱の中の位置)を付与＝基底文字
  - IVD：字形バリエーション：基底文字にぶら下がる

文字コード -エンコーディング

- 外部連携：UTF-8 (以外を用いるのはよほどの理由がない限り避けるのが妥当)
- 内部処理：UTF-16かUTF-8

符号化形式	符号化方式	単位幅(bit)	エンディアン	BOM	スカラー値1つのオクテット長(参考)	サロゲート表現	処理系
UTF-8	UTF-8	8	なし	原則不要 (付与可だが非推奨)	1〜4バイト	×	Web/HTML/HTTP/JSON/XMLの既定 Linux/Unix系ロケール・ファイル名 Rust/Go(文字列はUTF-8保証) 多くのツール・プロトコル
UTF-16	UTF-16BE	16	ビッグエンディアン	付与可	2または4バイト (1または2コードユニット)	○	一部ファイルフォーマットやプロトコルで明示的にBE指定(例：OpenTypeの一部テーブルなど)、ICU内部はUTF-16(環境依存でBE/LE)
UTF-16	UTF-16LE	16	リトルエンディアン	付与可	2または4バイト (1または2コードユニット)	○	Windows(ワイド文字API、NLS) .NET(System.StringはUTF-16) Java/JavaScriptはUTF-16コードユニットベースの文字列モデル
UTF-32	UTF-32BE	32	ビッグエンディアン	付与可	常に4バイト (1コードユニット)	×	一部の低レベルAPI／ファイルフォーマットで採用例あり(限定的)
UTF-32	UTF-32LE	32	リトルエンディアン	付与可	常に4バイト (1コードユニット)	×	POSIX/glibcのwchar_tは概ねUTF-32 一部ツール／処理系の内部表現で採用

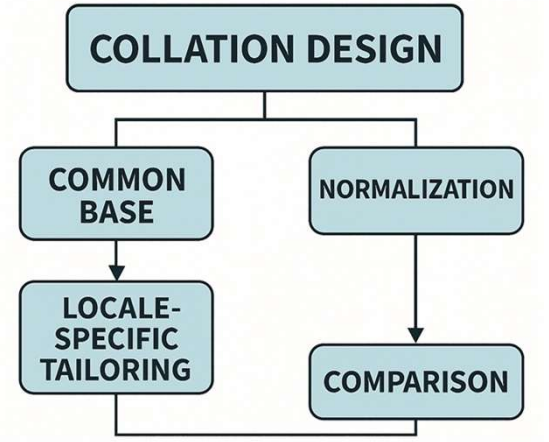
文字コード - 部分集合＝サブセット

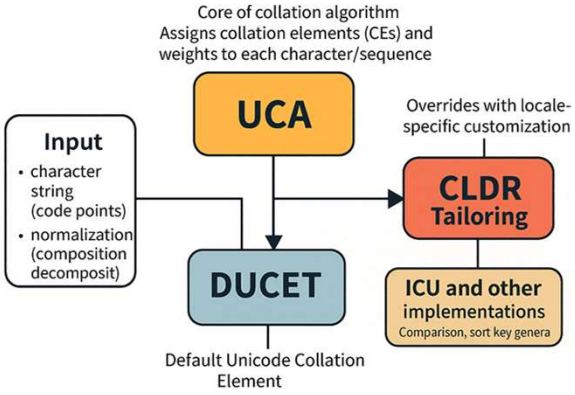
- ISO/IEC 10646(UCS)は巨大(15万文字以上)
  - 全部を対象にするのはほとんどの場合オーバースペック
- ISO/IEC 10646 Annex A で部分集合(サブセット)を定義している

元の集合	Annex A	
	#	Subset name
JIS X 0213	285	BASIC JAPANESE
	286	JAPANESE NON IDEOGRAPHICS EXTENSION
	371	JIS2004 IDEOGRAPHICS EXTENSION
文字情報基盤漢字・変体仮名	390	MOJI-JOHO-KIBAN IDEOGRAPHS-2016
	1042	KANA SUPPLEMENT の一部 (U+1B001～U+1B0FF)
	1115	KANA EXTENDED-A
行政事務標準文字 追加分		※これから

順序＝照合＝Collation

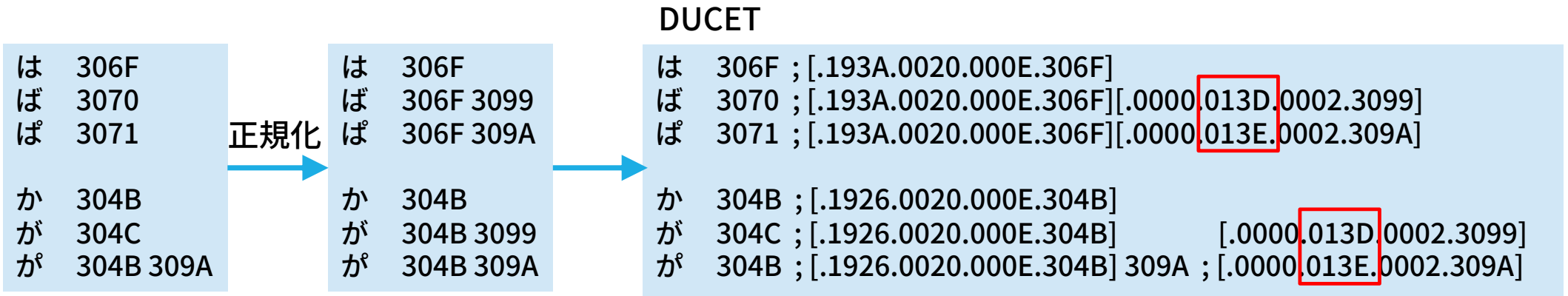
- 文字/文字列を比較して、どちらが「大きいか」を決定する
- 要件
  - 画面の「文字コード順」は、人の感覚的な辞書順と一致しない
  - 表記ゆれ(合成/分解、濁点・アクセント違いなど)を同じと見なしたい場面が多い
  - 並びや検索の慣習は言語・地域で異なるため、共通基盤＋ロケール調整が必要
- 仕組み
  - 文字や文字列に4レベルの「重み付け」を与え、その重みで比較・並べ替え
  - DUCET: 言語に依存しない既定重み(標準の初期値)
  - CLDR: 言語・地域ごとの慣習ルール(アクセントの扱い、ケース、句読点、数値ソート等)
  - 実装(例: ICU): DUCET＋CLDRを適用して、アプリやDBが使える比較・ソートキーを提供





照合時の動作

- 文字列を正規化＝比較の前提を揃える
- 4レベルの重み付けに従って大小を決定



4レベルの重み付け

レベル	分類
L1	基底文字
L2	アクセント
L3	大文字小文字/異体字
L4	句読点
Ln	同一

※IVS:無視するのが既定。明示的に処理しないと並び順は不定となる

照合規則の実装

- 後方互換との兼ね合いで実装パターンは様々。 CLDRベースで独自実装というパターンも
- バージョン間で差異がある場合もあり

カテゴリ	基盤	ICU互換度指数	代表的ギャップ	代表環境
ICUそのもの／ ICU直ラッパー	ICU	7–8	一部環境でテーラリング非対応／API露出が限定 (例：JavaScript Intl、.NET 5+)	Android PostgreSQL(ICU) SQLite+ICU JavaScript Intl(ブラウザ/Node) PHP intl Apple Foundation Windows(System ICU 直接利用) .NET 5+(既定ICU)
CLDR準拠の同等実装	CLDR	4–6	ロケール拡張(-u-co-)／テーラリングが限定的	Go collate OpenJDK java.text
Windows NLS ベース(版付きWindowsコレーション含む)	NLS(UCA近似＋Microsoftテーラリング)	6-8	BCP47の-u-co等による動的指定不可、任意テーラリング不可、numeric等のオプション露出が限定、UCA/ICUと細部差	Win32/WinRT API .NET Framework ≤4.x SQL Server(Windowsコレーション：_SC、*_UTF8、版付き_140/_150 など)
Microsoft SQL レガシーコレーション(SQL_系)	独自(非UCA/NLSの旧来仕様)	2–4	UCA非整合、コードページ依存、補助文字の扱いが不完全、テーラリング不可、UTF-8非対応	SQL Server(SQLコレーション：互換用)
独自UCAベース	UCA (独自/定義済)	3–5	動的co切替不可、テーラリング不可	Oracle Database MySQL 8.0
POSIX/glibc ロケール	POSIX/glibc	2–4	BCP47拡張不可、機能露出が限定	Python locale C/C++標準 locale

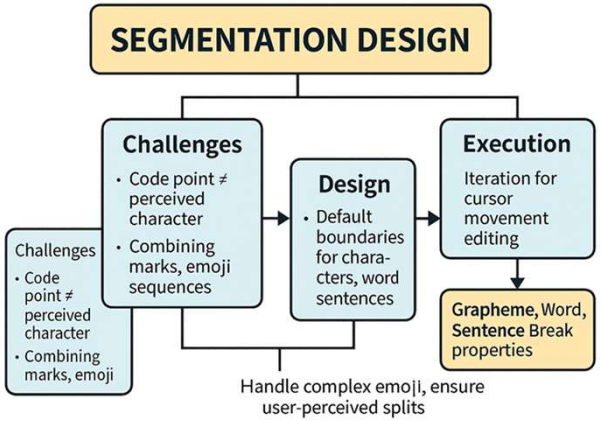
照合の実装 - DBMS

- 照合はDBMSの基本機能→旧バージョン互換性とICU互換性の両立の仕方は様々
- 照合の既定ではIVSは区別しない→区別する機能はOracle 23ai以降とSQL Server 2017以降

項目／DBMS	Primary	Secondary	Tertiary	Quaternary	Identical
区別する差異	基本文字のみ(アクセント・大文字小文字・幅・カナ差は無視)	アクセントを区別、大小は無視	大文字小文字・幅・カナ差なども区別	句読点・空白など可変要素(alternate=shifted時)	上位で同一の場合にコードポイントで最終タイブレーク
Oracle Database	例：NLS_SORT=BINARY_AI／BINARY_CI(近似)	言語別リンギスティック＋大小無視で近似	言語別リンギスティック＋大小区別で近似	明示指定不可(実装依存)	例：NLS_SORT=BINARY(コードポイント／バイト順)
SQL Server	例：Latin1_General_100_CI_AI(CI＋AI)	例：*_AS_CI	例：_AS_CS／_AI_CS	ユーザー指定不可(コレーション定義に依存)	例：_BIN2(コードポイント順。_BINは旧式)
MySQL／MariaDB	例：utf8mb4_0900_ai_ci(AI＋CI)	例：utf8mb4_0900_as_ci(AS＋CI)	例：utf8mb4_0900_as_cs／utf8mb4_0900_ai_cs	明示指定不可(定義に依存)	例：utf8mb4_bin／utf8mb4_0900_bin(コードポイント順)
PostgreSQL(ICU)	例：und-u-ks-level1	例：und-u-ks-level2	例：und-u-ks-level3(必要に応じて kc=true)	例：und-u-ka-shifted(alternate=shifted)	例：und-u-ks-identical
SQLite	ICU拡張：強度=PRIMARY／内蔵NOCASEはASCIIのみ近似	ICU拡張：強度=SECONDARY	ICU拡張：強度=TERTIARY	ICU拡張：alternate=shifted指定可	内蔵BINARY(コードポイント／バイト順)
IBM DB2	UCA強度=PRIMARY(UCA系コレーション)	UCA強度=SECONDARY	UCA強度=TERTIARY	UCAのALTERNATE=SHIFTED 指定可	UCA強度=IDENTICAL(またはバイナリ順)

分割

- 文字列をどこで分割して良いかの規定
- 必要な背景
  - コードポイントと「見た目の文字」は一致しない(結合文字、合成、絵文字のZWJシーケンスなど)
  - カーソル移動・選択・削除、語単位の操作や検索に一貫した境界が必要
  - 言語ごとに語・文の境界が異なるため、言語非依存な既定ルールが求められる
- 設計目標
  - 文字(拡張書記素集群)・語・文の「既定の境界」を提供し、ロケールでテーラリング可能
  - 正規等価や複雑な絵文字シーケンスを崩さず、ユーザー知覚に近い分割を保証
  - 高速な反復(イテレーション)で実用的な編集操作を支える
- 設計構造
  - 文字プロパティ駆動の規則セット(Grapheme_Cluster_Break／Word_Break／Sentence_Break、Extended_Pictographic、ZWJなど)
  - 拡張書記素集群(Extended Grapheme Cluster)を最小の「見た目の文字」単位として定義
  - CR+LFは一体として扱うなどの「非分割」優先ルール→分割許可ルールの順で適用
  - 語・文分割はデフォルト規則を提供し、辞書ベース分割(タイ語など)は別途テーラリングで補完



分割の規定

- 濁点・半濁点は前置した文字と分離しない、と規定
- 前置する文字に関する制約規定はない

が 304B 309A

か 304B
~~000D~~
~~000A~~
゛ 309A

UNICODE LINE BREAKING ALGORITHM
https://www.unicode.org/reports/tr14/#Properties

Table 1. Line Breaking Classes

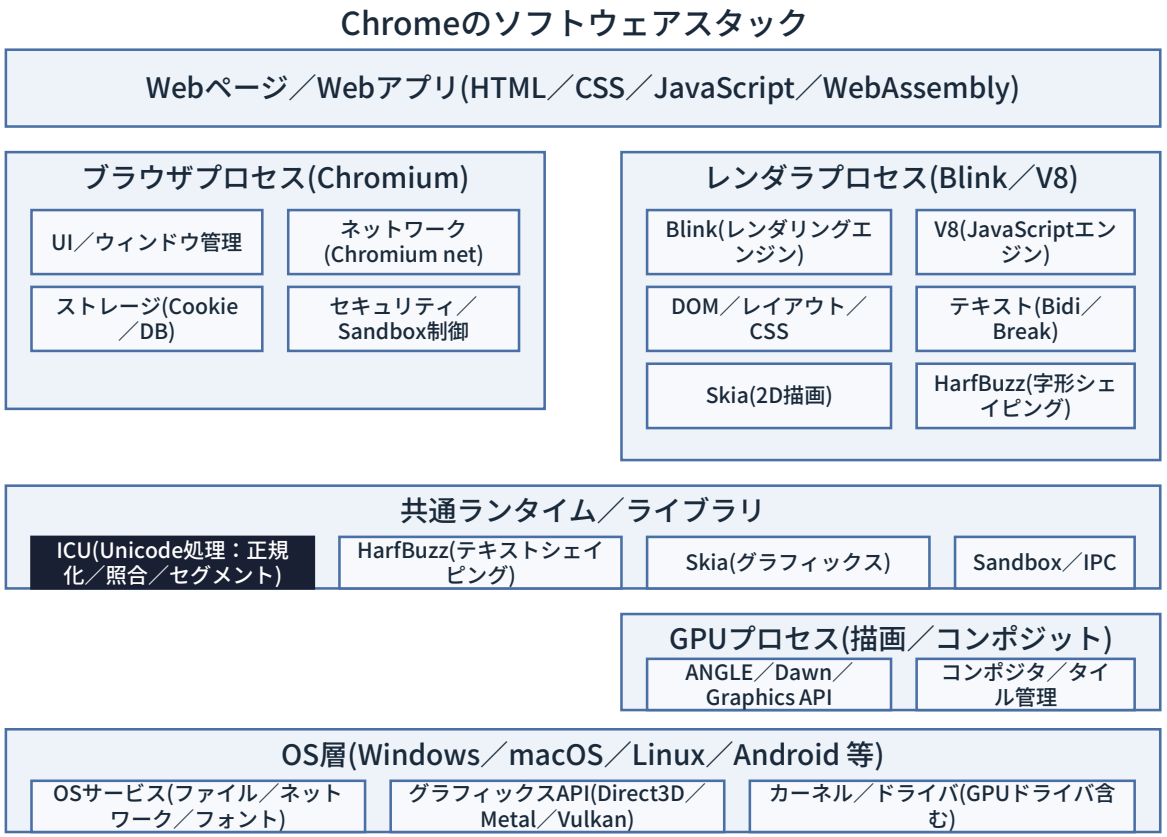
Class	Descriptive Name	Examples	Behavior
		Non-tailorable Line Breaking Classes	
BK	*Mandatory Break*	NL, PARAGRAPH SEPARATOR	Cause a line break (after)
CR	*Carriage Return*	CR	Cause a line break (after), except between CR and LF
LF	*Line Feed*	LF	Cause a line break (after)
CM	*Combining Mark*	Combining marks, control codes	Prohibit a line break between the character and the preceding character
NL	*Next Line*	NEL	Cause a line break (after)
SG	*Surrogate*	Surrogates	Do not occur in well-formed text
WJ	*Word Joiner*	WJ	Prohibit line breaks before and after
ZW	*Zero Width Space*	ZWSP	Provide a break opportunity
GL	*Non-breaking ("Glue")*	CGJ, NBSP, ZWNBSP	Prohibit line breaks before and after
SP	*Space*	SPACE	Enable indirect line breaks
ZWJ	*Zero Width Joiner*	Zero Width Joiner	Prohibit line breaks within joiner sequences

https://www.unicode.org/Public/UCD/latest/ucd/LineBreak.txt

```
3099..309A    ; CM # Mn    [2] COMBINING KATAKANA-HIRAGANA VOICED SOUND MARK..
COMBINING KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK
```

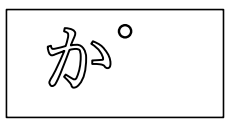
リファレンス実装＝ICU

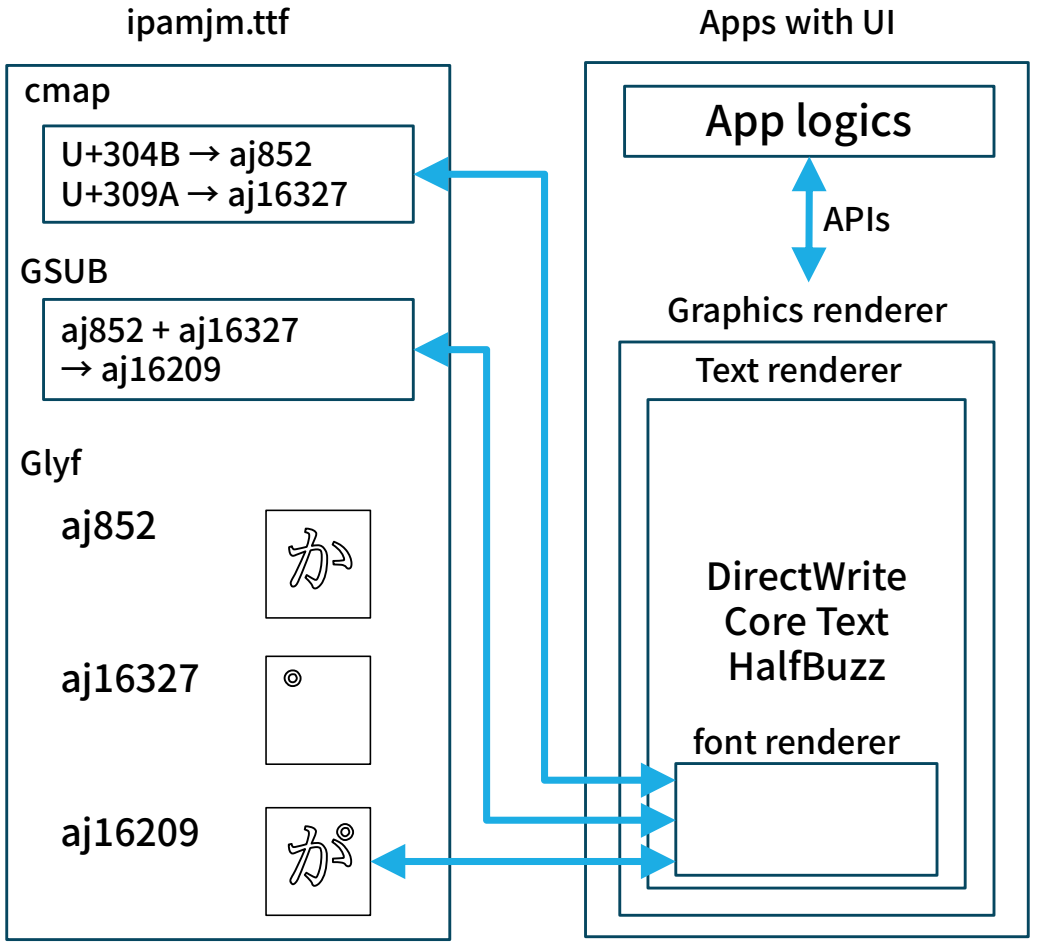
- 文字を含む地域性は非常に複雑
  →「仕様書」だけでは定義しきれない
  →リファレンス実装が存在
  →ICU: International Component for Unicode
- C/Java/Rustで実装
- ソフトウェアスタックの奥底に位置している
  →アプリ側から意識する必要はない



フォント実装

- 文字コードと字形をまとめたもの＝フォント
  だけではない
  - 文字列として描画する際の付帯情報を様々に付与
  - 「ぱ」：IPAmj明朝では「合成後の字形に置換する」定義を追加
- フォントデータの定義とレンダリングエンジンが協調して動作
  - レガシーのグラフィックス処理系では対応していない場合もあり





フォントフォールバック

- フォントは共有リソース
  →使いたい文字の範囲はアプリによって異なる
  →「・」「□」がなるべく出ないことが望ましい
- 複数のフォントを組み合わせてできるだけ文字が出るような仕組
  ＝フォントフォールバック
- つまり、「表示される」と「使って良い」は別

こんにちはD 😀

最優先フォント
（例: 游ゴシック）
こんにちは ✔ D ✕ 😀 ✕
→ローマ数字500の”D”と 絵文字が欠落

フォールバック1
（例: MS Gothic）
こんにちは ✔ D ✔ 😀 ✕
→ ローマ数字はあるが絵文字はなし

フォールバック2
（例: Emoji）
こんにちは ✕ D ✕ 😀 ✔
→ 絵文字のみ対応

「こんにちは」→ 最優先フォント (游ゴシック)
「 D 」→ フォールバック1 (MSゴシック)
「😀」→ フォールバック2（Emoji）
またはシステムデフォルトフォント

ここまでのまとめ

- 文字の標準化は、箱の中の位置を決めることだけではない
- ワールドワイドに対応したデータベースとアルゴリズムに扱い方を定義することも必要
- フォントも仕様を整合させたものが必要になる
- 表示される文字＝使っていい文字、ではない
- 実装のレベルは処理系/基盤によって異なるので、処理のルートで結果が異なることも起こりうる

文字情報技術促進協議会
JCITPC

公的証明と本人確認

公的な証明書類

- 戸籍：戸籍謄本、抄本
- 住民記録：住民票、戸籍附票の写し、転出証明書、印鑑登録証
- 税：納税通知書、納付書
- 保険：(国保/介護保険/後期高齢者医療)被保険者証
- 福祉：児童手当、各種給付・助成決定通知
- 教育：就学通知書

20業務標準化によって、
紙の書面は行政事務標準文字に集約していく

マイナンバーカード等の文字セット

- 行政事務標準文字の文字コードをデジタルに保持するものは現時点ではない

種類		文字セット	補足説明
マイナンバーカード	券面	住基統一文字 ＋外字 (→行政事務標準文字に移行)	住民票の表記と同じ (市町村外転居時は再発行となるが、必ずしも外字が廃されるわけではない)
	電子証明書	JIS X 0208 ＋JIS X 0212	券面表記文字のうち、左記文字セットに含まれない文字は、総務省作成の”代替文字確認ガイド”に基づき、自治体側で置換
在留カード 特別永住者証明書	券面	JIS X 0208 ＋JIS X 0212 ＋入管外字(176文字)	左記文字セットに含まれない文字は、法務省 “在留カード等に係る漢字氏名の表記等に関する告示”別表に基づき、出入国在留管理庁側で置換
運転免許証	券面	JIS C 6226 (JIS X 0208:1978) ＋外字	
	ICチップ内	JIS C 6226 (JIS X 0208:1978) ＋外字	外字イメージがICチップ内に格納されている

※JIS X 0213は、JIS X 0208を包含するが、JIS X 0212は包含しない
※住基統一文字、JIS X 0212は、文字情報基盤に包含されている
※マイナンバーカード 電子証明書の項は、公的個人認証サービス プロファイル仕様書 3.1.1版を参照

書類の対面確認が必要なケース

- 紙での受付＝目視確認と再入力、の局面は減少
- 表示は？

対面要否×突合有無	主な局面(例)	本人確認手段(代表)	マイナンバーカード普及の影響
対面不要(オンライン完結可)	● オンライン口座開設(銀行・証券) ● 暗号資産・送金アプリ有効化 ● 携帯回線のオンライン契約(eSIM等) ● 税・給付金のオンライン申請(e-Tax/マイナポータル)	マイナンバーカードIC読取＋JPKI、運転免許＋顔認証(セルフィー)	eKYCが標準化・即時化。紙・手書きの削減、非対面完結が広範に可能に。
対面必須＋突合あり(窓口で手書き氏名・住所とデジタル情報を照合)	● パスポート申請・交付 ● 運転免許の更新・再交付 ● 在留関連手続 ● 印鑑登録 ● 公証役場の嘱託・定款認証 ● (店頭限定の)銀行口座開設	顔写真付き公的身分証(マイナンバーカード／運転免許／旅券 等)、必要に応じ戸籍・住民票等	窓口でのカード提示により確認が迅速化。事前オンライン申請が広がる一方、交付・受領時の対面と突合は引き続き必要。
対面必須＋突合限定(提示中心・記帳はするが外部照合義務は弱い)	● 宿泊チェックイン(名簿記載) ● 店頭の年齢確認(酒・たばこ等) ● 国内線搭乗時の本人確認 ● 宅配の年齢確認商品受取 ● 郵便の本人限定受取	顔写真付き公的身分証提示(マイナンバーカード等)	カード提示の受容が進み手続が簡素化。事前Web登録やスキャン導入で手書き負担が軽減。

ここまでのまとめ

- 紙の書面は行政事務標準文字に集約していく
- マイナンバーカード 公的個人認証の基本4情報はJIS X 0208+JIS X 0212
- 手書き突合→再入力は減少する一方で、表示/印刷は必要

実践のポイント

調達側

- ユースケース
  - デジタル化に伴うワークフローの変化(対面前提→非対面・ネットで完結)の有無
  - 文字集合のバリエーションのどれが適切かを判断
  - 入力が本当に必要なのかを判断
- 調達
  - 要件書に記載する事項

文字集合のバリエーション

- ユースケースに応じて選択する必要がある

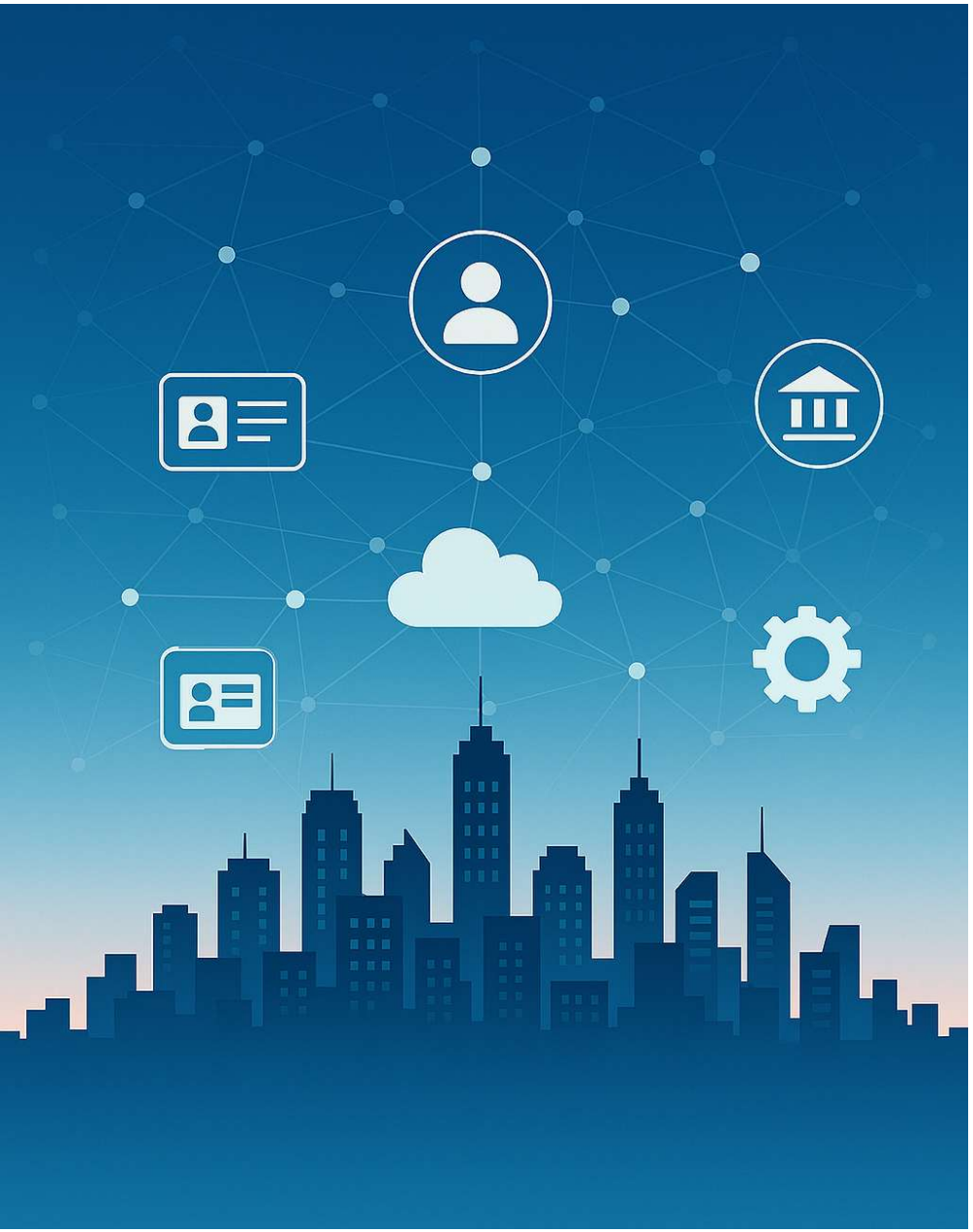
文字集合	メリット	デメリット
昔ながらのシフトJIS (JIS X 0208＋α)	● ミニマムセットだが、もはやない (レガシーの組込環境ぐらいしかない)	● カバー範囲が狭い ● 入力に強い制約を掛ける必要が生じる
JIS X 0208＋JIS X 0212	● ほとんどのプラットフォームで利用可能 ● マイナンバーカード内の基本4情報を表示できる	● 標準として古い
JIS X 0213	● 政府が行政処理の標準と定めている	● マイナンバーカード内の基本4情報の表示には不十分
文字情報基盤	● 追加の1万文字以外は行政事務標準文字と同じ ● 全て国際標準化済みなので、処理系でも正しく扱われることを期待できる	● 字形差異が小さい漢字が多数あるので、使う側のスキルが問われる ● 有償フォントがほぼ存在しない
行政事務標準文字	● 戸籍氏名文字を正確に再現できる	● 同上 ● 一部が外字扱い
外国語も表現可能な文字集合	● 入管正字以外の外国語の文字を使える	● 使いこなすのが更に困難 ● 処理系によって対応の程度が更にばらける

要件書に記載する事項

- 文字仕様
  - 文字集合：Annex Aのsubset #
  - エンコーディング：内部/外部を分ける
  - 前提とするフォント：明示する (フォントを示す＝文字集合を示すわけではない)
- 入力
  - IMEがカバーしない範囲の入力要否、必要な場合の手段
  - 仕様外の文字入力時の振る舞い
- データ処理
  - 移行：新旧の文字集合の突合要件＝他の文字に丸めるのかどうか
  - 照合：処理系(実装言語/DBMS)で照合規則が整合することの確認
- 出力
  - 改行、文字数制限のある局面での境界値(合成文字・IVS)テスト

実装側

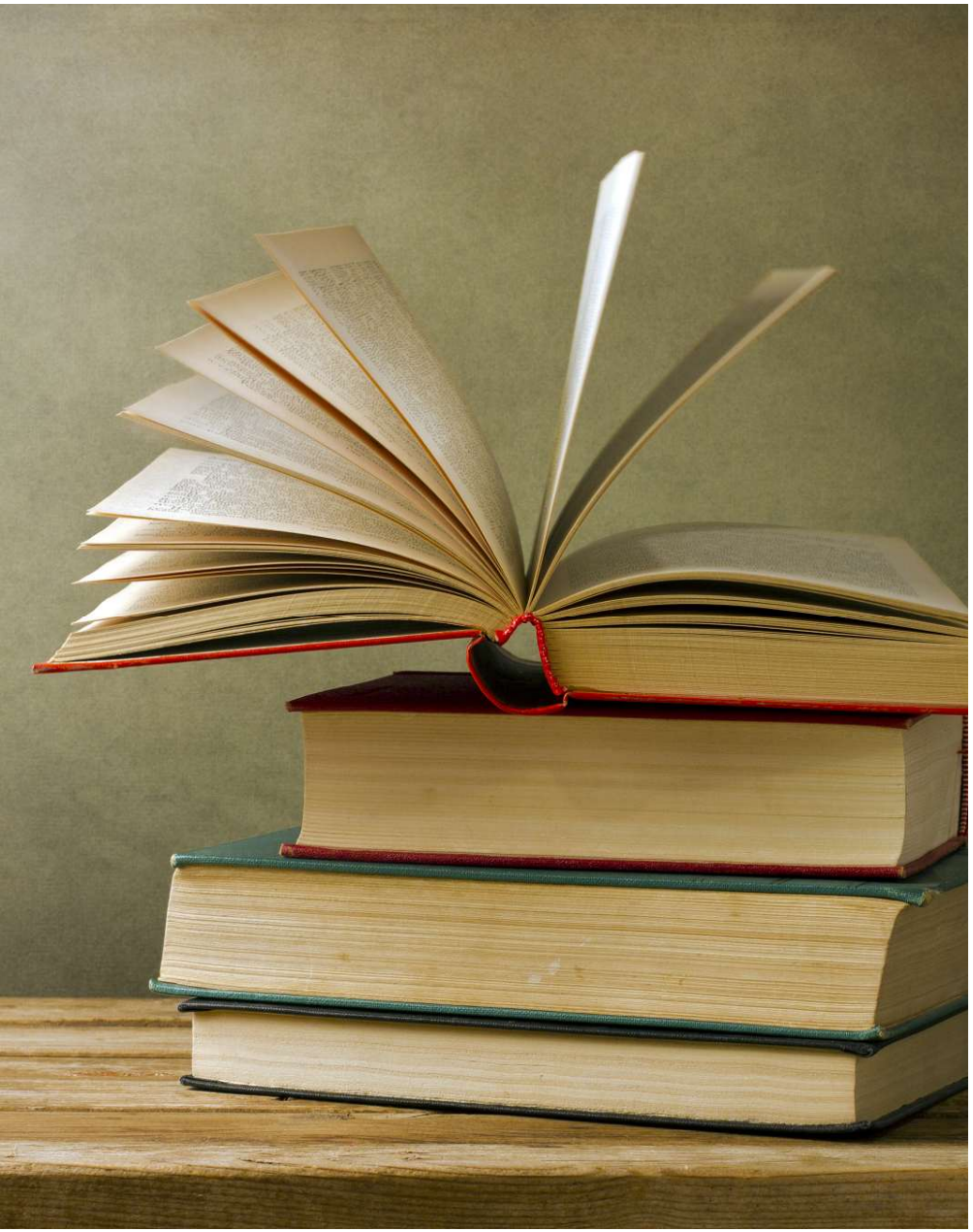
- 内部設計
  - モダンな処理系であれば差異はあっても概ね扱える
    - IVSの比較が必要ならば、DBMSの選択肢は限られる
  - テストセットを固める
    - 文字集合の境界値
    - IVS・合成文字×改行・文字数制限処理
- 外部設計
  - インタフェースのエンコーディング・文字集合：レガシーシステム向けの縮退変換機能の要否



今後の展望

行政事務標準文字の今後

- 文字字形集合
  - 商業法人・不動産登記の使用文字のうち、突合できない文字字形が約4,000
    (2025年4月25日 デジタル庁ベース・レジストリ推進有識者会合（第2回）資料6に基づく)
    (https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/c30d3c8e-17bf-4107-a1ab-b00c2f4c74ef/60b43962/20250425_base-registry-advisory-board_outline_06.pdf)
- 国際標準化
  - 途半ば
  - レアケースの扱いであり、明快なカテゴライズができるわけではない
  - 多角的な材料に基づいて、各論を丁寧に進めていく必要がある



最後に

行政事務標準文字を使いこなすには、
まだまだ整理しなければならないことが多い

協議会に入会して、
一緒に議論していきましょう


JCITPC
文字情報技術促進協議会