標準化教育プログラム［個別技術分野編－電気電子分野］

第１２章　文字コード標準
（日本語文字の符号化）

本資料は，経済産業省委託事業である「平成１８年度基準認証研究開発事業（標準化に関する研修・教育プログラムの開発）」の成果である。

2006年10月20日
富士通株式会社ソフトウェア事業本部
関口正裕

（標準講義時間 90分）

© JSA 2007

学習のねらい ・・・・・ 第12章 文字コード標準

- 文字コードの標準化に関して，次のことを理解し，考えるきっかけとする。
  - 文字コードを標準化する意義，目的。
  - 関係者間での利害の対立をどのように調整・解決してきたか，課題は何なのか。
  - 情報技術の標準化と，社会の様々な活動との関係。

目　次 ・・・・・ 第12章 文字コード標準

p. 3

◆ 解　説

本資料は，大きくわけて次のような構成になっている。

文字コードとは(シート5)

文字コードに関するごく簡単な技術的背景知識を説明。

歴史 (シート6～12)

文字コード標準の初期から現在までの歴史を説明し，現在主流 (となりつつある) UCS/Unicodeの登場までの課程をたどる。

いくつかの論点(シート13～17)

主にUCS/Unicodeに関して，標準ができあがるまでに行われた議論について，その論点と結論を説明する。

·UCSとUnicode (13～14)
類似の課題に対する異なる解として登場したUCSとUnicodeの，それぞれの原型にあった違いをどのように一本化したのかを述べる。

·漢字の統合 (15～16)
UCS/Unicodeの標準化にあたって，日本で最も熱心に議論された「漢字の統合 (Unification)」について，議論の要点を述べる。

·なぜ漢字の文字コードを国際標準にするのか? (17)
これは，UCS/Unicodeの標準化に直接関係者の間では (当然のこととして) 議論にならなかったが，当時も今も，ときどき出る疑問。

現状と課題 (シート18～21)

現在熱心に議論されている点，及び，課題として認識されつつも議論の糸口の無いものも含め，文字コードの標準化活動が今後解決していかなければならない課題をいくつか示す。

トピックス: 文字化け (シート22～24)

残念ながら，パソコンを使っていると (特にインターネットのように不特定の相手との通信を行うと) 「文字化け」に出会うことが珍しくない。ここではトピックスとして，文字化けが起きる仕組みについて簡単に説明し，その原因の一部に文字コードの標準化との関連がある点について論じる。

おことわり

- 用語について
  - 本資料中では，一貫して「文字コード」の語を用いているが，標準化分野での正式な用語は「符号化文字集合」という。「文字コード」という語は，情報技術分野・通信技術分野ではごく一般的に使う言葉なので，こちらに合わせた。
  - 他の文献等を読む際には注意。

p. 4

◆ 解　説

このシートは単なる余談ですが，本資料の用語法についての背景は次の通り。

スライドにも書いた通り，標準化分野での正式な用語は「符号化文字集合」である。おそらくこれは，英語の標準化分野の用語である“coded character set”を訳したものと想像する。しかし，この「符号化文字集合」の語は，標準化活動以外の場面ではめったに使われることはないし，そもそも標準化活動の場面でもあまり使わない。情報技術分野・通信分野の技術者は，普通「文字コード」を使う。（公式な文書，例えば，JIS等では，必ず「符号化文字集合」と書くが。）

実は上に書いた“coded character set”の語も，英語の技術文書等では一般的でない，標準分野特有の言葉なのであって，普通は“character encoding”，“code set”（“codeset”と一語で綴ることもある），“charset”などの語を使う。

言うまでもなく，用語の統一は標準化活動の基礎だが，残念ながら「文字コード」を意味する用語については，日本でも米国でも，標準化に失敗していると言わざるを得ない。

本資料では，符号化文字集合という用語を用いることも検討したが，少なくとも情報技術分野・通信分野の基礎知識を持つ人にとっては，「文字コード」はなじみのある言葉だが「符号化文字集合」では意味がわからず，教材としての遡及力に欠けると判断し，あえて「文字コード」を用いることにした。

文字コードとは

- コンピュータなどデジタル機器の内部では，すべてが数値。
- 文章などの文字データも数値化する必要がある。そこで，それぞれの文字に数値（番号）を割当てることで，文章を数値の集まりとして扱う。
- この，文字に対する数値の割当て規則を文字コードと言う。
- ASCII，シフトJIS，ISO-2022-JP，Unicode，などが文字コードの例。
- 文字コードの数値は，普通，16進数で書く。

H	e	l	l	o	,		世	界	さ	ん

↓

0048	0065	006C	006C	006F	0027	0020	4E16	754C	3055	3093

p. 5

◆ 解　説

このスライドは，情報技術分野の背景知識をもたない人のための解説を意図した。さすがに，この程度を知らないと，以降の議論を理解するのは辛い。逆に，学習者に情報技術の基礎知識が期待できるなら，このシートは冗長。

なお，JIS等では「符号化文字集合」を次のように定義している。

文字集合及びその集合の文字と符号化表現との間の関係を定めるあいまいさのない規則の集合。(JIS X 0221)

文字集合を定め，かつその集合内の文字とビット組合せとを1対1に関係付ける，あいまいでない規則の集合。(JIS X 0208)

こられの定義で，「符号化表現」とか「ビット組合せ」とか呼んでいる概念を，このシートでは大雑把に「数値」としている。技術的には，少々厳密さに欠けるが。

このシートで文字コードの例として挙げたものの読み方は次の通り: ASCII (アスキー)，シフトJIS (シフトジス)，ISO-2022-JP (アイソ・にいまるにいにい・ジェイピー)，Unicode (ゆにこーど)。なお，ISO-2022-JPについては，本資料では詳しく扱わないが，これは日本国内の電子メールで利用されている文字コードの名前で，ISO/IEC 2022という国際規格の考え方に基づいているのでこの名があるが，国際規格が定めたものはない。いわゆるデファクト標準である。

図は，文字コードのイメージを示す。上段のようなテクストを，文字コードによって数値で表現すると，下段のようになる，という意図。なお，ここで示している数値は，スライドにも書いている通り16進数で表しており，実際にそれぞれの文字のUnicodeのコード値である。(単なる例であり，説明不要。)

文字コード標準の初期の歴史

- 初期の代表的な標準
  - 1963年: ASCII誕生。(現在の形のASCIIは1968年の改正による。)
    - + 世界で最初の標準文字コードと言われる。
    - + 名前はAmerican Standard Code for Information Interchangeの略で，アメリカの国内規格として標準化された。
  - 1967年: ISO 646。
    - + ASCIIをもとに国際規格として標準化。
  - 1969年: JIS C 6220 (後に JIS X 0201 と名前が変わる)
    - + ISO 646をもとに，日本用に一部の文字をアレンジ(“¥”など)。
    - + カタカナ(現在，ふつう半角カタカナと呼ばれているものに相当)を利用する仕組みを導入。
- このころの文字コードは，いわゆる「1バイトコード」。
  - 印刷装置の活字の制約から，文字数は100字程度が上限であった。
  - 通信分野で，1文字を4～6ビット程度で表す技術が使われていた。

p. 6

◆ 解　説

このスライドでは省略しているが，この時期にヨーロッパでは次のようなことが起きていた。

·同じアルファベットを使う言語でも，フランス語やドイツ語などのアクセント類や変音記号を必要とする言語圏では，その種のアクセントを伴った文字を固有に追加した文字コードが作られていた。

·北欧の言語では，英語のアルファベットに加えて，アクセント類だけでなく，特有の文字をいくつか必要とする(“æ”，“þ”，“ð”など)が，これらの追加も行われた。その結果，ASCII (ISO 646) の94文字の領域では不足する地域が現れた。

·ヨーロッパ横断的な情報交換のニーズに応えるための文字コードが求められたが，このためには思い切った整理が必要で，アクセント類はアクセントだけを1文字と考え，「重ね打ち」によって表現する仕組みが考えられた。

このスライドの最後に，いささか唐突に「印刷装置の活字」が登場するが，これは次のような背景による:

·当時のコンピュータの出力は，プリンタが普通だった。これは，出力専用の装置がプリンタであるというだけでなく，いわゆる対話型の端末も，ロール氏に印字されるタイプのもの (いわゆるテレタイプ) が普通であった。

·当時のプリンタは，文字を活字で持つものが多く，文字種の増加は，機械的な形状を大きくし，動作を遅くする結果となった。このため，文字数を小さく抑えることが重要であった。現在主流の「ドットマトリックス方式」のプリンタが登場するのは，1970年代以降である。

ASCIIのコード表（参考）

- 1文字を7ビットで表現する1バイトコード。
- 横軸が上3ビット，縦軸が下4ビットを表す。
  - 例えば，“A”は，横軸が“100”，縦軸が“0001”なので，“1000001”という7ビットの二進数になり，これは16進数で書けば41である。
- アミカケ部分は“制御コード”または“制御文字”と呼ばれる領域。

	000	001	010	011	100	101	110	111
0000				0	@	P	`	p
0001			!	1	A	Q	a	q
0010			"	2	B	R	b	r
0011			#	3	C	S	c	s
0100			$	4	D	T	d	t
0101			%	5	E	U	e	u
0110			&	6	F	V	f	v
0111			'	7	G	W	g	w
1000			(	8	H	X	h	x
1001			)	9	I	Y	i	y
1010			*	:	J	Z	j	z
1011			+	;	K	[	k	{
1100			,	<	L	\	l	|
1101			-	=	M	]	m	}
1110			.	>	N	^	n	~
1111			/	?	O	_	o	

p. 7

◆ 解　説

このスライドは，ASCIIのコード表を示す。情報技術のバックグランドがあれば，ASCIIについては知識があると考えられるので省略してよい。

ここに示すのは，かなり現代的なASCIIの解釈であって，厳密なことを言うと，前のスライドで述べている「初期の歴史」に登場するASCIIとは若干異なる。主な違いは次の通り

·制御コード領域をアミカケにしている点。
現代的には，制御コード領域は場所だけ確保するというイメージで，ASCIIの一部とはみなさないのが普通。しかし，当時のASCIIは，制御コード領域も含めて全体をASCIIと規定していた。

·SPACE（“!”の上の，何も描かれていない場所）を，制御コードではなく，空白文字として扱っている点。
当時は，SPACEは「何も印刷せずに1文字分飛ばす」という制御機能と考えるのが一般的だった。

·全ての文字を固定的に示している点。
当時のASCIIは，一部の文字について“オプション”を認めていた。

なお，日本の技術書では，“ASCII”と題して，逆斜線（“L”と“l”の間，101/1100の場所）の位置に円記号“¥”を示すものが多いが，本来のASCIIは（当時も今も）“¥”を含まない。この位置を“¥”に変更したのは，前のスライドのJIS C 6220 であり，これが日本において“ASCII相当”として広く普及したため。

漢字の文字コードの登場

- 1970年代から日本語情報処理が登場。
- 漢字を扱う上での当時の技術的課題。
  - 文字数が多く入力が困難。表示・印刷も困難。
  - 1文字あたりの字画が複雑で，微細な表示印刷能力が必要。
  - アメリカで発達した文字処理技術の単純な延長では，漢字を扱うことができなかった。
- 当初は，新聞の編集システムのような特殊な分野で使われた。
  - コンピュータの利用で編集・印刷にかかる時間を短縮するニーズ。（欧米の事例など。）
  - 商業印刷物としては，印刷品質に対する要求が高くない。
  - 全国紙を発行する新聞社にとっては，速報性は非常に重要であり，多額の投資が可能であった。

漢字の文字コードの標準

- 1978年にJIS C 6226(いわゆるJIS漢字コード)が制定
  - 漢字の文字コードを標準化するための初期の研究は1970年には始まっているが，実際に規格として完成したのは1978年。
  - 初期の日本語情報処理システムは，ばらばらの文字コードによって漢字を処理していた。
  - 漢字は「これで全部」という一覧が存在せず，文字コード以前にまず漢字の一覧作りから始める必要があった。
- 中国
  - 1980年にGB 2312という中国国家標準の漢字コードが公布。JISの漢字コードと同一の構造。
  - 中国の国語政策に基づいた，簡化字(かんかじ)と呼ばれる中国式の簡略字体の漢字の文字コード。
- 韓国
  - 1986年に韓国標準のKS C 5601（後にKS X 1001と名前が変更された）という漢字コードが制定。
  - 日本から見ると「韓国の漢字コード」だが，韓国では「漢字も入った，ハングルの文字コード」の位置づけである。

p. 9

◆ 解　説

当時，日本には日常的に用いる漢字を示す政令として“当用漢字表”と“当用漢字字体表”があり，両者を併せて“当用漢字”と呼ばれていた。これは現在の“常用漢字”の前身で，もちろんJIS C 6226は，この当用漢字を網羅している。しかし，当時，ごく一般的な印刷出版用途においても“当用漢字表”に掲載された漢字だけでは不足するという現実があり，広く多方面で用いることを想定した漢字の標準文字コードは，当用漢字表に掲載されていない漢字を多数収容する必要が認識されていた。このスライドで“漢字の一覧作り”と言っているのは，この，当用漢字の他にとの漢字を採用するのか，という一覧作りのことである。

中国では，1950年代半ば以降，いわゆる“簡化字政策”と呼ばれる，漢字の略体化の政策が行われている。この流れの中で，1964年に“簡化字総表”，1965年に“印刷通用漢字字形表”が国語施策として公布されており，印刷出版で用いる漢字の範囲や，その漢字の字形が法令によって規定されていた。GB2312は，漢字に関しては“印刷通用漢字字形表”に掲載された文字に文字コードを振ったものと考えられる。

漢字のコード表の例（参考）

- **最新の漢字コードの規格である JIS X 0213:2004 のコード表の一部**
  - **「漢字の文字コード」と言っているが，実際には漢字だけでなく，かな文字や，英数字，記号類も含んでいる。**

						b7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						b6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						b5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
						b4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
						b3	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
						b2	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
						b1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
b7	b6	b5	b4	b3	b2	b1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	0	0	0	0	1	1	SP	、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨	：	￣	＿	ヽ	ヾ	ゝ	ゞ	〃
0	1	0	0	0	1	0	2	◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	＝	｜	‐	／	＼	～	∥	｜	…	‥
0	1	0	0	0	1	1	3	▷	▶	◁	◀	↗	↘	↖	↙	↔	⇔	⇐	⇑	⇓	↗	↘	°	′	″	￥	＄	￠	￡	％
0	1	0	0	1	0	0	4	あ	あ	い	い	う	う	え	え	お	お	か	が	き	ぎ	く	ぐ	け	げ	こ	ご	さ	ざ	し
0	1	0	0	1	0	1	5	ア	ア	イ	イ	ウ	ウ	エ	エ	オ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ
0	1	0	0	1	1	0	6	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ
0	1	0	0	1	1	1	7	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
0	1	0	1	0	0	0	8	─	│	┌	┐	┘	└	├	┬	┤	┴	┼	━	┃	┏	┓	┛	┗	┣	┳	┫	┻	┼
0	1	0	1	0	0	1	9	€	SP	¡	¤	：	©	ª	«	¬	®	‾	²	³	´	，	¹	º	»	¼	½	¾	¿	À
0	1	0	1	0	1	0	10	Ａ	˘	Ł	Ľ	Ś	Š	Ş	Ť	Ź	Ž	Ż	ą	˛	ˇ	ł	ľ	ś	˝	ŕ	ş	ť	ź	˚
0	1	0	1	0	1	1	11	?	ß	○	ｋ	ｂ	ｄ	ｆ	ｇ	Ｇ	œ	Œ	ｉ	ｕ	ｙ	ｅ	ｏ	３	ｅ	ｕ	ш	ｕ	Ｙ	ʌ
0	1	0	1	1	0	0	12	❶	❷	❸	❹	❺	❻	❼	❽	❾	❿	⓫	⓬	⓭	⓮	⓯	⓰	⓱	⓲	⓳	⓴	i	ii	iii
0	1	0	1	1	0	1	13	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱	⑲	⑳	I	II	III
0	1	0	1	1	1	0	14	倶	丈	乗	亙	丰	主	亨	仟	份	仿	伃	伋	你	佈	佐	佑	侈	佃	佗	伽	佛	俐	倆
0	1	0	1	1	1	1	15	咩	哆	哿	唄	唸	唵	啄	啖	啣	啻	喝	喜	喝	嗜	嗎	嘆	嘖	嘎	嘻	嘩	噛	噴	嚔
0	1	1	0	0	0	0	16	亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥	旭	葦	芦	鯵	梓	圧	斡	扱
0	1	1	0	0	0	1	17	院	陰	隠	韻	吋	右	宇	烏	羽	迂	雨	卯	鵜	窺	丑	碓	臼	渦	嘘	唄	欝	蔚	鰻
0	1	1	0	0	1	0	18	押	旺	横	欧	殴	王	翁	襖	鴛	鴎	黄	岡	沖	荻	億	屋	憶	臆	桶	牡	乙	俺	卸
0	1	1	0	0	1	1	19	魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾	外	咳	害	崖	慨	概	涯	碍
0	1	1	0	1	0	0	20	粥	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勘	巻	喚	堪	姦	完	官	寛	干	幹	患	感	慣
0	1	1	0	1	0	1	21	機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起	軌	輝	飢	騎	鬼	亀	偽	儀
0	1	1	0	1	1	0	22	供	挾	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強	彊	怯	恐	恭	挟	教	橋	況
0	1	1	0	1	1	1	23	掘	窟	沓	靴	轡	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君	薫	訓	群	軍	郡	卦	袈	祁

出典: JIS X 0213:2004（表1）

p. 10

◆ 解　説

このシートでは，単なる例として，JIS X 0213のコード表の一部を示している。

実際のコード表は，この右側・下側に，まだまだ延びている。（コード表全体の左上の部分を切り取って示している。）

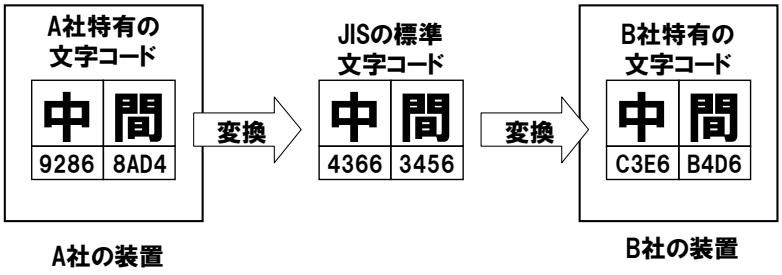
前に示したASCIIコードに比べて非常に多くの文字が含まれている点と，表の上・左に書かれたビット組合せのビット数が多いことに注目。

ASCIIは上端が3ビット，左端が4ビットだったが，X 0213は上端も左端も共に7ビット。そこで1文字あたり2バイト使い，各バイトの下位7ビットとして埋め込む。（最上位のビットが余るが，ここは，他の都合によって0にしたり1にしたりする。）このため，この種の文字コードのことを「2バイトコード」と言ったりDBCS（Double byte code set）と呼んだりすることがある。これは，日中漢の漢字コードのコード表に共通の技術的特徴。

なお，2進数のビット組合せは，縦も横も“0100001”から始まっているが，その下や右に書かれた10進数は“1”から始まっている。これは，2進数のビット組合せは，実際の「コード」だが，10進数は「表の中の位置（座標）」を表す番号だから。この10進数で書かれた，表の中の位置を表す数値のことを「区点番号」と呼ぶ。（縦軸が「区番号」，横軸が「点番号」。）

中間の形式としてのJIS漢字コード

- 標準の制定後すぐに普及。
- 情報処理システムの内部処理には標準の文字コード(JIS漢字コード)をそのまま使わず，別の機種などと情報をやりとりする際にいったん経由する中間形式としての利用。
- 日本では「内部コード」と「外部コード(標準コード)」の併用が一般化。



p. 10

◆ 解　説

このシートでは，単なる例として，JIS X 0213のコード表の一部を示している。

実際のコード表は，この右側・下側に，まだまだ延びている。(コード表全体の左上の部分を切り取って示している。)

前に示したASCIIコードに比べて非常に多くの文字が含まれている点と，表の上・左に書かれたビット組合せのビット数が多いことに注目。

ASCIIは上端が3ビット，左端が4ビットだったが，X 0213は上端も左端も共に7ビット。そこで1文字あたり2バイト使い，各バイトの下位7ビットとして埋め込む。(最上位のビットが余るが，ここは，他の都合によって0にしたり1にしたりする。) このため，この種の文字コードのことを「2バイトコード」と言ったりDBCS (Double byte code set) と呼んだりすることがある。これは，日中漢の漢字コードのコード表に共通の技術的特徴。

なお，2進数のビット組合せは，縦も横も“0100001”から始まっているが，その下や右に書かれた10進数は“1”から始まっている。これは，2進数のビット組合せは，実際の「コード」だが，10進数は「表の中の位置 (座標)」を表す番号だから。この10進数で書かれた，表の中の位置を表す数値のことを「区点番号」と呼ぶ。(縦軸が「区番号」，横軸が「点番号」。)

ISO/IEC 10646 UCS（Unicode）

- UCS/Unicodeとは
  - 1990年代に登場した新しい文字コード。
  - 世界中で必要なすべての文字を含んだ共通文字コードを目指す。
- UCS/Unicode登場の背景: 1980年代後半の文字コード標準
  - 各国ごとに異なる文字コードが標準化されていた。
  - 各国の文字コードを整合させるための規格ISO/IEC 2022があったが，後発で各国の規格を整理するものであったため，複雑で難解な仕様となっていた。
  - ヨーロッパでは，国境・言語圏を越えた情報流通の一般化により，特定の国や言語によらない文字コードが求められていた。
- 二つの提案
  - このような背景の中，おおよそ同じことを目標にした異なる文字コードが，同時期に設計された。主にヨーロッパ主導で登場したUCS（10646）と，米国企業が主導したUnicodeである。

UCSとUnicodeの相違点

- ISO/IEC JTC1/SC2で多オクテット符号の検討開始（UCS）
  - いわゆるアルファベット類だけ整理し，その他は場所だけ用意して既存の規格をはめ込む方式。
    - 漢字も日中韓の場所を別々に用意して，それぞれ国内規格をはめ込む予定だった。
  - 各国が自由に使えるよう区画には十分な余裕を持たせ，4バイト（最大約20億符号）の符号としていた。
- Unicodeの提案
  - すべての文字を整理し，重複しないように配置。
    - 漢字も整理する（unification/統合）が，このとき「重要でない形の違いは無視」して文字数を削減する，とした。
    - アクセント類も組み合わせる方式（combining character）によって文字数を削減。
  - きちんと整理すれば2バイト（最大約6万5千符号）で十分とした。

p. 13

◆ 解　説

ラテン・ギリシア・キリル（ロシア）の，いわゆる三大アルファベットについては，ISO/IECもUnicodeも，当初から整理の上で一元的に配置することにしていた。ヨーロッパ諸国には，素朴な感覚として，自分たちが共通の文字を使って様々な言葉を表記している，という感覚があるようだ。おそらくは，これに加えて，一人で複数の言葉の読み書きができることがめずらしくないため，「何語で書くかによって文字コードを選択する」のは実用上も不便なのだろう。また，特に情報分野では，世界中どこに行っても基本的なラテンアルファベットは使われており，この範囲が共通になっていないと実用上問題が大きいという問題意識もあった。

UCSとUnicodeの一本化

- 経緯
  - 主にヨーロッパ諸国が一本化を強く要求。
  - 様々な妥協により，両者が歩み寄り。
    →「木に竹を接いだよう」との批判も。
  - 1992年にUnicode 1.0として，1993年にISO/IEC 10646として，同一内容で出版された。
- 一本化の内容
  - 符号の大きさは，4バイトの領域を用意し，当面（不足するまで）は2バイトの範囲だけを利用するという妥協になった。
  - Unicodeの組み合わせる方式（combining character）を採用したが，主要なものについては，あらかじめ組み合わせた形も用意しておくことになった。
  - 漢字は「統合」に決着したが，Unicodeが従来行っていた統合の方針は不適切として，統合をやり直すことになった。

漢字の統合をめぐる議論

- 漢字の統合とは
  - 日中韓の「同じ漢字」には，共通の（唯一の）文字コードを与える。
  - 問題は，何を「同じ漢字」とみなすか。
  - 現在のUnicode/10646の統合は，中国が中心となり，関係各国の専門家とUnicode Consortiumの代表とが議論の上で作り上げた基準に基づく。
- 日本での議論
  - 1990年代半ばに活発な議論が起きる。
  - UCS/Unicodeの漢字の統合の是非の議論に加えて，ある種のナショナリズム的な「日本の漢字を他国に自由にさせるな」のような議論，文化的に中国語圏の波に飲み込まれる危機感に基づく議論，さらには，「人間の創造性がコンピュータによって制約される」というような議論など，様々な異なる観点の議論が渾然と行われ，混迷した。

p. 15

◆ 解　説

Unicodeの是非については，1990年代前半に世界中で議論になりました。論点は様々ですが，日本における議論の大部分は，漢字の扱い方，特に漢字の統合 (Unification) の妥当性に関するものでした。

スライドの後半に概要を書いていますが，当時の議論は非常に多岐にわたっており，要約が困難です。また，筆者自身が当時の論争の渦中にいたため，客観的な整理が困難という面もあります。

ここでは，当時の論点を追うことのできる文献をいくつか紹介するに留めたいと思います。(特定の立場からの論述は避け，多くの意見を収録したものを選んであります。)

·吉目木晴彦他; 電脳文化と漢字のゆくえ 岐路に立つ日本語, 平凡社, 1998.

·小林龍生他編; インターネット時代の文字コード, bit 2001年4月号別冊, 共立出版, 2001.

·千野栄一他; 特集文字 西夏文字からデジタルフォントまで, 月刊ユリイカ 詩と評論, 1998年5月号, 青土社, 1998.

漢字の統合・非統合の基準（参考）

区別しない（同じ文字として扱う）パターンの例

辶・辶・辶	礻・示・礻	且・且・皂	飠・食・飠	黄・黄	盈・盈	曷・曷
包・包	青・青	每・每	冊・冊	爭・争	謠・謠・謠	彔・彔
歩・歩	者・者	臭・臭	幷・并	骨・骨	呂・呂	直・直
県・県	吳・吳・吳	眞・眞・真	爲・為	単・単	曾・曾・曾	成・成
専・専	内・内	晉・晋	龜・龜	艹・艹		

区別する（別の文字として扱う）パターンの例

拡・擴	策・筴	𠂤・𡗜	圣・巫	僉・僉	区・區	夾・夾
単・單	雇・雇	戋・戔	賛・贊	襄・襄	韭・韮	間・閒
朶・朵	雋・隽	恒・恆	奐・奐	人・入	宗・宗	叉・叉

出典: JIS X 0221-1:2001 附属書S（上段はS.1.5より，下段はS.1.4.3より。）

p. 16

◆ 解　説

このスライドでは，現在のUnicodeおよびISO/IEC 10646で，共通に採用されている，具体的な漢字の統合の規則（の一部）を，JISの規格票からの引用で示します。

このスライドに示すのは，前のスライドで書いた「問題は，何を『同じ漢字』とみなすか」に呼応しており，現在の統合の規則が同じとみなすもの，みなさないものの具体例です。

統合の規則は，個々の漢字の形や構成に関する規則の積み上げで書かれていますが，ここに示したのは「部分字形」と称する，様々な漢字の一部に共通して比較的高い頻度で現れるようなモノであって，広い意味で「同じ字」と呼ばれるものが多いものについて，「同じ文字とみなして，一つの文字コードを与える」のか「異なる文字とみなして，異なる文字コードを与える」のかの判断基準となるものです。

この規則は，実際に各国の専門家が集まって，多数の漢字の実例を見ながら議論を行い，作り上げた結果です。

この教材（モジュール）では，漢字の統合の規則そのものを解説したり，理解を求めたりするものではありません。このスライドは，具体的に現在の文字コード標準が採用している統合の規則の一部を眺めることによって，雰囲気だけでも感じていただくことと，稀にみかける「Unicodeの漢字の統合は，漢字のことを全くわかっていないアメリカジンが，文字の表を見てテキトーに並べたデタラメである」のような誤解を解消することを目的としています。

なぜ漢字の文字コードを国際標準にするのか？

- 利用者の分布
  - 日本に住んでいると「（日本の）漢字は日本だけの事情」と考えがちだが，そうではない。
  - 全世界の人達が，誰でも漢字を実際に必要とするわけではない（ネジとはちがう）が，漢字の利用者が特定の国に集中しているわけでもない。
- 産業界の要求
  - 国際企業は，製品のコストダウンのため，できるだけ共通仕様で世界中に売りたい。漢字/日本語処理も，世界共通仕様の範囲内で対応したい。
  - このためには，漢字の文字コードも他人事ではなく，国ごとにばらばらでは困る。世界共通の国際標準が求められる。

p. 17

◆ 解　説

例えば，中国語を話し，情報機器で中国語を漢字を使うことを必要としている利用者は世界中にいる。海外で，日本語をパソコンで使う利用者も，留学生，企業の駐在者などを含め多い。

例えば，イギリスで，パソコンで漢字を使いたいユーザが非常に多ければ，黙っていてもイギリスのパソコンメーカーは漢字を扱えるようにするための工夫をするだろう。しかし，イギリスに漢字の利用者がいるとは言っても，全体から見れば少数。国際的に標準化された漢字の文字コードがあるから，漢字をサポートするコストが下がり，イギリスで販売されているパソコンでも，漢字が使えるという状況が生まれたとも考えられる。

実際，現在イギリスで販売されているパソコンは（WindowsでもMacintoshでも）設定さえすれば漢字が使える。これは20年前には考えられなかった。当時は日本語が使えるパソコンは日本でしか購入できなかったため，日本からパソコンを持っていき，電源等を改造して使う必要があった。

標準化が一般ユーザにもたらした恩恵と言えるだろう。

現在の状況

- 文字コード標準
  - ISO/IEC 10646 UCS（Unicode）が普及しつつある。
  - 従来型の，国ごと地域ごとの文字コードはだんだん減り，今後はUnicodeに収束すると予想されている。（しかし，おそらく数年オーダーではない。）
  - Unicode自体の拡張も続いている。
- 漢字の統合
  - 統合そのものの是非に関する議論はあまり行われなくなった。
  - 技術的には，統合を前提に，統合されている形を使い分ける仕組みが検討されている。
  - 国語審議会が公表した“表外漢字印刷標準字体表”との関連で，「正しい漢字」との関係での議論が増えている。
- 実装の問題
  - 現在の文字コードは，標準が先行しており，実際の情報システムの対応が追いついていない。標準だけあっても利用できない，絵に描いた餅の面がある。

p. 18

◆ 解　説

本稿執筆時点（2006年）の，UCS/Unicodeをめぐる状況を概観。

情報技術分野では，

2000年の国語審議会の答申は，あくまでも国語施策の観点から述べられているが，情報機器との関係についても言及されている。

なお，文字コードの「標準化が先行し，実際の情報システムが追いつかないため，規格があっても実際には使えない」という状態を指して「絵に描いたもじ」というダジャレがあり，業界で流行っている。

課題（技術）

- バリエーションの増加
  - UCS/Unicodeは，世界中に多数ある文字コードを一本化することを目指したが，実用につれてUCS/Unicode自体にもバリエーションが増えている。
    - + 多数の変換形式: UTF-8，UTF-16，UTF-32，…。
    - + バイト順による差と解決方法の差: UTF-16BE，UTF-16LE，BOM，…。
    - + 正規化の問題，特にcombining character の扱いに関するもの。
  - 現時点では簡易な解決策はなく，今後の進展が待たれる。
- 既存文字コードとの変換規則
  - UCS/Unicodeへの移行には，従来の既存文字コードとの間の対応・変換が必要。
  - 変換方式にも差があり，この差に起因するトラブルが増えている。
  - 変換規則の標準が一応存在するのだが，使われていない。

p. 19

◆ 解　説

バリエーションの問題は，近年技術的に問題になっている。応用と結びつく形で，例えばインターネットのディレクトリサービスではUTF-8に統一するなど，分野ごとの標準として特定のバリエーションを規定するような動きがあるが，なかなか一本化にはなっていない。

ただし，幸いなことに，このエンコーディングのバリエーションの問題は，ソフト・ハードを開発する技術者には問題だが，情報システムを利用する一般利用者にとってはほとんど問題にならない。このバリエーションが原因で利用者が不便を被ることは，あまりない。（皆無ではないのだが…。）

逆に，変換規則の差については，一般利用者に見える形のトラブルの元となっている。

課題（社会: 国内）

- 文字種
  - そもそもどれだけの文字があれば十分なのかが不明確。これは30年前から変わっていない。
  - 技術革新がない限り，網羅的な一覧がないと，文字コードは設計できない。
  - 日本で使う漢字の網羅的な一覧を作るのは，社会的に見て，誰の仕事なのだろうか?
- 字体・字形
  - 30年前は，情報機器で漢字を扱うことができる，というだけで画期的だった。
  - 現在は，ただ使えるだけではだめで，正しい形，美しい形が要求される。
  - 正しさには規範が必要だが，それは国語施策の問題で，情報技術の領域ではなくなってしまう。
  - 特に，初等教育（学校での漢字指導）との関係が重要。

課題（社会: 国際）

- 国際的に見ても，主要な文字はおおよそ標準化が完了し，新たな標準化の対象は利用者の少ない文字になっている。
  - 少数民族に固有の文字（UNESCOの識字教育施策との関係）
  - 識字教育との関係古代の文字，死滅した文字
  - 特定分野に固有な記号類
- 課題
  - 文化/福祉/学術上の有用性と産業上の有用性
  - 費用負担の問題
  - 専門家の不足

p. 21

◆ 解 説

UCS/Unicodeはすでに10万字以上を含み（因みに，このうち約7万字が漢字），世界の大多数の人の日常的な情報機器の利用に足りる状態になっていると考えられる。

少数民族に固有の文字が，まだ相当数残っているが，これは従来情報機器で利用できず，その文字の利用者自身が情報機器による利用を期待していないものが大半と言える。また，日本で暮らしていると実感がわかないかもしれないが，全世界平均の識字率（文字が一通り読み書きできる人の比率）は75%程度と言われており，2割以上の人は満足に文字を使えない。UNESCOは，識字率の向上に情報技術を活用しようとしているが，その前提となるのが，情報機器で民族固有の文字が扱えることである。

古代文字に対する要求は主として学術的なもの。

情報分野の標準化活動は，産業での有用性が評価の尺度として重要だが，国際的な文字コード標準は産業上の有用性から離れる方向に向かいつつあるようにも見える。費用負担の問題を含め，今後の大きな課題と考えられる。

また，利用者が少ない文字は，文字コード開発を担当できる専門家の少なさにも繋がる。筆者自身も含め，見たことも聞いたこともない文字に関する文字コードの原案を見せられても，妥当性を評価することができない。

UNESCOの識字教育施策についての参考情報: http://portal.unesco.org/education/en/ev.php-URL_ID=40338&URL_DO=DO_TOPIC&URL_SECTION=201.html

トピックス: 文字化け

- 文字化けとは
  - –文字データの一部(または全部)が，誤りや異常な文字となって表示されること。
- 文字化けの原因

  1.データそのものの誤り・異常
  → 通信エラー，ソフトウェアの誤り(バグ)，ハードウェアの不良，…

  2.文字コード(コード表)の非互換
  → いわゆる「機種依存文字」問題。

  3.使用する文字コードの「打合せ」の失敗
  → いわゆる「自動判定」の失敗や，打合せ方法に問題。

  –パソコン通信時代(1990年代前半まで)の文字化けは，1.と2.が原因だった。

  –現在，WWWで起きる文字化けの大部分は3.。

  –2.は，文字コードの標準化によって防げる(はず)。

p. 22

◆ 解　説

世間で一般に言う文字化けには，技術的には三種類のことなる現象がある。このスライドでは，1，2，3と番号を付けている。

1は，文字コードデータそのものが正しくない値に変わってしまうことが原因。ソフトウェア・ハードウェアの不良や，通信の際に生じる雑音によって起きる。1980年代半ばのパソコン通信時代には，アナログモデムを用いた冗長性のない通信方式が主流であり，雑音に弱かった。このため，ある程度の文字化けは普通であった。この種の文字化けは，正常な文字の途中に1文字から数文字，異常な文字が混ざるという現れ方をするのが特徴。最近は，通信方式やハードウェアが改良されたことに加えて，データの誤りの有無を確認しながら通信し，誤りが見つかった場合には再送信する冗長性の高い通信プロトコルの利用が一般化(インターネットのTCP/IPという通信プロトコルは，この種の技術を多重に組み合わせている)したため，ほとんどみかけなくなった。

2.も，パソコン通信時代にはよく見られたが，最近は減っている。これについては次葉で述べる。

3.は，比較的最近(1990年代半ば以降)になって起き始めた新しいタイプの文字化け。これについては次々葉で述べる。

トピックス: 文字化け: コード表の非互換

- だいたい同じだが一部違う文字コードの存在
  - 標準が決めていない部分(空き領域・未定義領域)に，独自の文字を追加する場合がある。
  - これは，特定のメーカーが特定の機器に固有に行うことが多い。
  - 本来の標準文字コードからはずれた部分なので，同じ機種同士でないと正常に文字が再現できない。(文字が消えたり，別の文字に変わったりする。)
- パソコン通信時代は，「標準文字コード+α」が一般的
  - 複数のパソコンメーカーが，他社と互換のないパソコンを販売しており，「+α」の内容が，メーカーごとに千差万別であった。
- → 当時，この「+α」の部分を「機種依存文字」などと呼んだ。
- これは，文字コードの標準化が不十分だったために起きた不具合とも言える。

p. 23

◆ 解 説

1980年代の日本のパソコン市場は，NECの「PC8800シリーズ」「PC9800シリーズ」がトップシェアで，それを富士通，シャープ，日立，IBMなど，他の何社かが特定の得意分野で追う，という構造であった。当時のパソコンは，大抵「MS-DOS」を採用し，「シフトJIS」という文字コードを使っていたが，この「シフトJIS」は正式な標準になっていたなかったこともあり，基本的な部分は各社共通だったが，細部が様々に異なっていた。また，本来は未定義のコードに，各社が工夫を凝らした様々な文字を追加していた。

パソコン通信では，様々なメーカーのパソコンを使う人たちが集まってメッセージのやりとりをするため，メーカーによって文字の割当てが異なる部分は，自分のパソコンと相手のパソコンとで，別の文字として表示されてしまう。これが当時問題となり「機種依存文字」などという言葉を生んだ。

この状況は，1993年以降大きく変わった。マイクロソフト社が1993年に「日本語Windows 3.1」を販売したが，この際に，パソコンメーカーとの契約によって，マイクロソフト社が決めた文字コードの採用を義務づけ，メーカーが個別に独自の文字の追加・変更を行うことを禁止した。パソコンメーカー各社からの反発は大きかったようだが，結果として日本のWindowsを採用するパソコンは，メーカーや機種によらず完全に同じ文字コードが使われるようになり，「機種依存文字」による文字化けはほとんど姿を消した。(Windows以外のOS，unix系やMacOSなどは，相変わらずWindowsとは異なる文字コードを使うものも多かったため，この原因による文字化けが皆無になったわけではないが。)

トピックス: 文字化け: 打合せに失敗

- 文字コードの打合せ(negotiation)とは何か?
  - 文字コードは多数あり，相互に同じ文字コードを使っていないと，意味のある情報交換が行えない。
  - 昔のテクノロジでは，場面ごとに使用する文字コードを一種類に統一するのが常識だった。
  - 現在，特にインターネット周辺技術では，使用する文字コードを互いに確認し合うのが一般的。これを「打合せ」と言う。
- WWWでの文字コードの打合せ
  - 通常は，WWWサーバーからウェブブラウザに，一方的に使用する文字コードを通知するだけ。
  - WWWでは，文字コードの打合せは義務ではない。実施しないことも多い。(サーバーの設定による。)
  - 打合せを省略した場合は，ウェブブラウザが推定する。(自動判定と呼ぶ。)
- 打合せと文字化け
  - 打合せを省略している場合，ブラウザが推定を間違うと文字化けが起きる。
- 今後の方向性
  - 打合せが必要なのは，様々な文字コードが使われるため。今後は，文字コード自体がUnicodeに一本化し，打合せをやめる方向性も。

p. 24

◆ 解 説

このスライドの記述は，技術的にはかなり不正確である。WWWでの，実際の文字コードの打合せの仕組みは，相当に入り組んでいる。実際の文字コードの打合せの仕組みについては，例えば，HTML 4.01仕様の，5.2 Character Encoding に記載がある。これは，WWWでの打合せの全貌ではなく，サーバがHTML文書をブラウザに向けて送信する際の打合せの仕組みを述べているに過ぎない。実際のWWWでは，この種の仕組みがいくつか組み合わせて使われる。

打合せが省略された場合，ウェブブラウザは文字コードを推定しようとするが，具体的な推定のアルゴリズムは，ウェブブラウザが様々に工夫しており，何か標準があるわけではない。コードの値の出現頻度の分布を見たり，特定のコード値の組合せの出現の有無をみたりする手法がいくつか知られている。確実な推定のためには十分に長いサンプルデータが必要だが，実はブラウザの一画面に収まる程度のデータでは，大抵の推定アルゴリズムには不足。このため，特に短いページだと，間違った推定を行ってしまうことが起きる。推定を間違った場合，別の文字コードだと思ってデータを解釈するので，めちゃめちゃな文字が表示されることになる。(ただし，現在WWWで日本語用に使われている文字コードは，どれも，ASCII文字はASCIIコードで表すように設計されているため，文字コードを誤解しても，ASCII文字は正しく解釈できる。このため，「日本語の部分だけ」文字が化ける結果になる。

なお，スライドでは明記していないが，打合せを実施するためには，サーバー側で実際の文字コードについての情報を設定する必要がある。この設定が間違っているために，実際の文字コードとは異なる文字コードを通知してくるサーバが，ときどきある。そこで一部のブラウザは，打合せを行った場合でもサーバから通知された文字コードを全面的に信用せず，それを参考にしながら文字コードの推定アルゴリズムを実行するものがある。実は，サーバーの管理者が確認に使うブラウザに，この機能が備わっているため，設定の誤りに気付かないという状況があるらしい。親切な機能も一長一短である。

◆ 参考資料

WWW コンソーシアム (W3C) 勧告 HTML 4.01

http://www.w3.org/TR/html401/

のうち，5.2 Character Encoding の部分

http://www.w3.org/TR/html401/charset.html#h-5.2

ま と め ・・・・・ 第12章 文字コード標準

- 文字コード標準化の目的について
  - 文字コードの標準化によって，異なる機種間などでも情報交換が可能になる。
  - 日本では，標準の漢字コードは，主に交換の場面でだけ，外部コードとして使われた。
  - 日本語用の文字であっても国際標準の一部となっており，そのことには合理性がある。
- 利害対立について
  - UnicodeとUCSとは，当初対立したが，後に歩み寄り一本化された。
  - 漢字の統合をめぐる議論については，関係各国の専門家によって検討が行われた。
- 社会の活動との関係
  - 文字コードは情報技術であるが，文化・教育等との関係が深い。
  - 少数民族の文化，識字などの問題とも関連しており，単純な産業上の有用性だけで評価できない面を持っている。

演習問題A ・・・・・ 第12章 文字コード標準

1.文字コードが標準化されていることが役に立っている場面は何があるか。できるだけ身近な例を考えよ。

2.日本語の文字(特に漢字)のための文字コード標準の確立にとって問題になったことは何だったか。

3.普段使っている情報機器またはパソコンのソフトウェアは，どういう文字コードを使っているか。いくつか調べてみよ。

4.本稿で触れなかった地域を選び，1990年代以前の文字コードの状況と経緯について調べてみよ。それが，1990年代後半以降，どう変わったかも調べてみよ。

5.何らかの意味で漢字に関するニュースが年に何回か流れる。最近のものをいくつか選んで，文字コード標準との関係について考えよ。

演習問題B ・・・・・ 第12章 文字コード標準

1.1980年代後半の日米の情報システムを比較した場合，次のような状況があった。米国では内部処理にも外部とのやりとりにも，標準の文字コードASCIIを用いるものが多かったが，日本ではJISの漢字コードを内部処理に用いるものはほとんどなく，内部処理は装置に固有の文字コードで行い外部とのやりとりの際に文字コードの変換を行うことが一般的であった。当時の，日米のこの状況について，得失を検討せよ。文字コードの標準化と，この状況との関係についても考えよ。

2.利用者の少ない文字に対する文字コードの費用負担の問題について検討せよ。国際的に問題となっている少数民族に固有の文字に関する例と，日本で議論のある非常に稀な人名（主に姓）や古い地名に使われるめずらしい漢字の例とでは，共通点は何で相違点は何だろうか。

3.ある文字に関して，文字コードの標準ができることと，その文字が実際の情報機器で使えることとの間には差がある。標準があっても実際には使えない例，標準がなくても使える例をいくつか探し，なぜそういう状況なのかを考察せよ。標準化の観点から，その状況をどう評価するかも考えてみよ。

参考資料（国際規格）・・・・・ 第12章 文字コード標準

- ISO/IEC 646:1991, Information technology -- ISO 7-bit coded character set for information interchange
- ISO/IEC 2022:1994, Information technology -- Character code structure and extension techniques
- ISO/IEC 8859-1:1998, Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1
- ISO/IEC 10646-1:1993, Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane
- ISO/IEC 10646-1:2000, Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane
- ISO/IEC 10646-2:2001, Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 2: Supplementary Planes
- ISO/IEC 10646:2003, Information technology -- Universal Multiple-Octet Coded Character Set (UCS)

p. 28

◆ 解 説

このスライドと次のスライドは，文字コード関連の規格を網羅したものではなく，この教材（モジュール）を作成するにあたって直接参照したものだけを列挙しています。特に国際規格では，文字コードの規格は他にももっとたくさんあります。

参考資料（国内規格）・・・・・ 第12章 文字コード標準

- JIS X 0201（C 6220）-1976, 情報交換用符号
- JIS X 0201:1997, 7ビット及び8ビットの情報交換用符号化文字集合
- JIS C 6226-1978, 情報交換用漢字符号系
- JIS X 0208（C 6226）-1983, 情報交換用漢字符号系
- JIS X 0208:1990, 情報交換用漢字符号
- JIS X 0208:1997, 7ビット及び8ビットの2バイト情報交換用符号化漢字集合
- JIS X 0212:1990, 情報交換用漢字符号 ― 補助漢字
- JIS X 0213:2000, 7ビット及び8ビットの2バイト情報交換用符号化拡張漢字集合
- JIS X 0213:2004, 7ビット及び8ビットの2バイト情報交換用符号化拡張漢字集合
- JIS X 0221-1:1995, 国際符号化文字集合（UCS）― 第1部:体系及び基本多言語面
- JIS X 0221-1:2001, 国際符号化文字集合（UCS）― 第1部:体系及び基本多言語面

p. 29

◆ 解　説

JISは規格番号の変更があったため，改正の経緯がわかりにくい。

JIS X 0201 は，もともとJIS C 6220 という番号だったものが，完全に同一内容で番号だけ変更になったあと，1997年に内容の改正が行われた。

JIS X 0208 は，もともとJIS C 6226という名前で，1983年に一回改正されたのち，同一内容のまま番号だけ変更になったあと，さらに内容の変更を伴う改正が2回行われて現在に至っている。

参考資料（雑誌・書籍）・・・・・ 第12章 文字コード標準

- 小林龍生他編; インターネット時代の文字コード, bit 2001年4月号別冊, 共立出版, 2001.
- 安岡孝一, 安岡素子; 文字コードの世界, pp 11-27, 東京電機大学出版局, 1999.

p. 30

◆ 解　説

「インターネット時代の文字コード」については，この教材（モジュール）の執筆にあたって参照したのは，スライドに記載したbit別冊ですが，これは雑誌なので現在は入手困難です。2002年に，同名の書籍として共立出版からほぼ同一内容の書籍が刊行されていますので，そちらが入手しやすいでしょう。