

Jalons de l'histoire des jeux de caractères

Thomas Soubiran
CERAPS (UMR 8026 CNRS – Université de Lille)

Séminaire CIS
GOUVERNANCE ET RÉGULATION D'INTERNET
Lille, 15 mars 2022

- ▶ Une multiplicité des jeux de caractères ont été conçus
ce qui ne veut pas nécessairement dire mis en œuvre
- ▶ depuis le début des années 1950 jusqu'aux années 1990
- ▶ L'objet de cette présentation est de passer en revue
- ▶ un ensemble de jeux de caractères
à défaut d'en faire la chronologie
- ▶ pour en faire ressortir les caractéristiques et les contraintes pesant sur leur conception
- ▶ ainsi que la multiplicité

- ▶ Il s'agit en quelque sorte de faire de la « plomberie » (MUSIANI 2012)
- ▶ en abordant le problème d'un point de vue
 - ▶ informatique
 - ▶ mais aussi linguistique
 - ▶ et des usages
- ▶ au travers des solutions qui y ont été très progressivement apportées

- i) Démarche et conventions
- ii) Codages ≤ 8 -bits —1950–1980—
- iii) Codages CJKV
- iv) Années 1980–1990

Démarche et conventions

- ▶ « plomberie » (MUSIANI 2012)

c-à-d en faisant de la conception des objets étudiés une partie intégrale de l'étude

- ▶ importance de caractériser la question techniquement

- ▶ non pas que les autres aspects ne soient pas importants
- ▶ mais pour faire ressortir l'ensemble des contraintes qui pèsent sur l'élaboration et la mise en œuvre
- ▶ à commencer par les contraintes matérielles au regard de l'usage prévu d'un jeu de caractères

- ▶ l'histoire de l'informatique est aussi une histoire d'abstraction vis-à-vis du matériel
- ▶ ajout de couches logicielles
- ▶ de plus en plus éloignées du fonctionnement de la machine

OS, langages de programmation de haut niveau, systèmes de fichiers, fenêtrage, bureau, ...

- ▶ mais l'éloignement ne le fait pas pour autant disparaître
- ▶ et si on l'oublie, le matériel finira toujours par se rappeler à notre bon souvenir plus ou moins douloureusement...

- ▶ le matériel a beaucoup pesé sur l'élaboration des jeux de caractères

la taille des jeux de caractères et la complexité algorithmique de leur implémentation est une fonction non-linéaire des ressources matérielles disponibles —cf. UNICODE

- ▶ avec ici une difficulté plus d'ordre historiographique

- ▶ il est plus facile de trouver des informations sur les JdC
- ▶ que sur leur utilisation

à commencer par des informations sur les machines ou les logiciels pour lesquels ils ont été conçu ou qui les ont utilisés

- ▶ toujours le problème des sources

► autres difficulté historiographique : lister les jeux de caractères utilisés

- il n'y a pas que des standards nationaux et internationaux
- mais aussi beaucoup de JdC « maison »

proposés par un vendeur ou un autre : machines, logiciels,...

► Or,

- il est plus facile de trouver des informations sur les standards que sur les autres JdC
- et il est plus facile de trouver des informations sur les JdC utilisés p. ex. par IBM, Microsoft ou Apple que Bull, Olivetti ou Siemens
- sans parler d'entreprises ayant une surface moindre ou ayant une existence moins longue

et donc ayant laissé d'autant moins de traces

- risque de biais de sélection rétrospectif qui surreprésente certains JdC
- difficulté à laquelle s'ajoute celle de dater les JdC

et donc d'en retracer la chronologie

- ▶ les jeux de caractères permettent de souligner « maison » l'importance de l'informatique hors-normes
- ▶ depuis le début, l'informatique se déploie souvent en dehors de tous standards
 - qui arrivent souvent après
- ▶ il y a aussi les standards de fait
 - comme l'architecture IBM-PC compatible
- ▶ les standards ne disent pas toute l'histoire
- ▶ ne pas oublier aussi que des standards ne sont que des textes
 - ▶ ce n'est pas parce qu'une loi est adoptée qu'elle est nécessairement suivie d'effet
 - ▶ ou des effets prévus

- ▶ plutôt que d'établir une chronologie
- ▶ la démarche adoptée vise à
 - ▶ retenir des jeux de caractères caractéristiques
 - ▶ c-à-d permettant de faire ressortir
 - ▶ les problèmes —et dilemmes—auxquels sont confrontés les faiseurs de table
 - ▶ et les solutions adoptées
 - ▶ et aussi de faire ressortir les relations entre jeux de caractères
- ▶ sans chercher à combler les trous et en essayant d'éviter d'imprimer la légende au début n'était pas ASCII

Jeux de caractères

- ▶ jeu de caractères désigne ici une table assignant un code numérique à chacun des éléments d'un ensemble de caractères
- ▶ un JdC associe généralement un nom à chaque caractère
- ▶ mais aussi, souvent, une définition

qui peut varier d'un standard à l'autre —cf. Jukka Korpela, *Soft hyphen (SHY) – a hard problem ?*, <https://jkorpele.fi/shy.html>

- ▶ les JdC comportent souvent des aspects explicitement linguistiques et sémantiques

comme on le verra à différentes reprises

- ▶ **Note :** le code peut être distinct de son encodage

c-à-d sa représentation en mémoire —cf. UNICODE et UTF-8 ou UTF-16

- ▶ standard sera ici réservé aux jeu de caractères publiés par une organisation de standardisation :
 - ▶ c-à-d d'organisations où sont représentées différentes parties intéressées
 - ▶ et dont le contenu des publications font l'objet de procédures délibératives
 - ▶ par opposition aux jeu de caractères développés par une entreprise pour son usage propre
- ▶ et norme sera réservé aux jeux de caractères rendus obligatoires par un État

- ▶ caractère désigne ici...ce qui est codé par un jeu de caractères
- ▶ difficile à définir de façon concise
- ▶ p. ex., la définition adoptée par UNICODE est celle de graphème,
 - ▶ soit la plus petite unité d'un langage écrit porteuse d'une valeur sémantique
 - ▶ par opposition aux différentes formes qu'elle peut prendre.

par opposition à des glyphes
- ▶ mais qui correspond à la façon adoptée par le consortium UNICODE de traiter la question
- ▶ autre définition (MACKENZIE 1980), p. 17 :

A character is a specific bit pattern and an assigned meaning

qui a moins le mérite de la simplicité

- ▶ Au cours des premières années de l'informatique, il n'existe pas d'industrie logicielle.

Les programmes sont écrits pour une machine particulière par ses utilisateurs

- ▶ Mais, au fil des années, l'offre logicielle va progressivement de plus en plus tirer la croissance de l'informatique

- ▶ particulièrement avec l'avènement de l'informatique personnelle
 - ▶ et de ses *Killer Apps* dont la disponibilité va longtemps peser sur le choix des ordinateurs achetés.

les logiciels sont encore rarement portables

- ▶ L'extension du marché s'est avant tout faite par la diversification de ses usages et de ses usagers,
- ▶ ces extensions privilégiant de plus en plus les usages textuels de l'informatique.

- ▶ Les ordinateurs sont de moins en moins des machines à calculer et de plus en plus des machines à lire et écrire (traiter du texte) :

- ▶ années 1950 : applications principalement numériques

mais aussi des applications administratives et commerciales

- ▶ années 1960 : terminaux, bases de données

- ▶ à partir des années 1970 :

- ▶ éditeurs de texte, PAO —ce qui a d'ailleurs motivé D. Knuth à concevoir T_EX—
- ▶ bureautique
- ▶ communication
- ▶ jeux
- ▶ . . .

- ▶ avec, dans les années 1970, un décalage de plus en plus marqué entre les nouveaux usages des jeux de caractères et ce pourquoi ils avaient été conçus au départ

la transition des terminaux vers des PC

- ▶ La multiplication des usages textuels de l'informatique rend toujours plus cruciale la question des jeux de caractères.
- ▶ Mais leurs limites intrinsèques
 - à commencer par leur multiplicité et leur hétérogénéité
- ▶ ainsi que leur inadéquation à de nouveaux usages
- ▶ vont constituer des freins au développement de l'industrie logicielle et donc à l'industrie informatique en général
- ▶ qui conduira à la constitution de projets de codes dit « universels » car visant à inclure à terme toutes les écritures connues.

ISO/CEI 10646–UNICODE

Codages ≤ 8 -bits

Quelques codes précurseurs

- ▶ **années 1840** : code Morse pour le télégraphe
- ▶ **années 1870** : code Baudot pour téléscripteurs modifié par Murray en 1901
- ▶ **années 1890** : code Hollerith pour les cartes perforées d'abord conçu pour réaliser les tabulations du recensement étasunien de 1890

- ▶ conception des codes Morse et Baudot de façon à faciliter l'implémentation matérielle
- ▶ et déjà, ce qu'on appelait pas encore l'internationalisation

- ▶ avec des variations nationales et linguistiques :

- ▶ **code Morse** : code wabun —和文モールス符号— *wabun mōrusu fugō*, texte japonais en code morse—
- ▶ **code Baudot** : variantes britanniques, étasuniennes et cyrilliques du code Baudot

- ▶ et standardisations au niveau national
- ▶ et international par le CCITT —Comité Consultatif International Télégraphique et Téléphonique—avec l'Alphabet international n° 2 pour le code Baudot

- ▶ particularité du code Baudot :

- ▶ code modal qui utilise un embrayage —shift—
- ▶ qui permet de doubler la capacité du code $2 \times (2^5 - 1)$

- ▶ H. Hollerith est le fondateur d'une société qui sera plus tard connue sous le nom d'IBM

qui conduira à la pérennité de son code jusqu'aux années 1980 et au de-là

Code Baudot

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	A	É	E	I	O	U	Y	B	C	D	f	G	H	J	FIG
1x	*	K	L	M	N	P	Q	R	S	T	V	W	X	Z	t	

(a) Lettres

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	1	&	2	°	5	4	3	8	9	0	F	7	h	6	
1x	*	(=)	No	%	/	-	;	!	'	?	,	:	.	LET

(b) Figures

FIGURE 1 – Code Baudot

Pour commencer :

- ▶ années 1950–1980
- ▶ codages ≤ 8 -bits
- ▶ écritures principalement alphabétiques

- ▶ Les premiers ordinateurs s'apparentent plutôt à des machines à calculer programmables
- ▶ pour des applications avant tout numériques
 - et notamment militaires dans le contexte de la Guerre froide
- ▶ Toutefois, le support de caractères alphanumériques se répand rapidement.
- ▶ Une des motivations de l'inclusion des caractères alphanumériques fut le développement d'ordinateurs
- ▶ à destination des entreprises et les administrations.
 - comme l'Univac I conçu par J. Presper Eckert and John Mauchly qui fut utilisé dès 1951 par le *Census Bureau* notamment pour les tabulations du recensement en remplacement des appareils électro-mécaniques —cf. Hollerith
- ▶ Exemple d'usage commercial : la réservation de billets dès la fin des années 1950 au Japon
 - conçu par Hitachi pour la compagnie nationale de chemin de fer du Japon (日本国有鉄道, *Nippon Kokuyū Tetsudō*)
- ▶ Un des moteurs de leur inclusion fut aussi le développement des premiers langages de programmation de haut niveau tels FORTRAN ou ALGOL.

- ▶ Dès les années 1950, on assiste à une multiplication des codes.
 - ▶ Un inventaire réalisé en 1960 dans le cadre des travaux du sous-comité X3.2
 - ▶ en liste pas moins d'une cinquantaine rien qu'aux EU (BEMER 1960)
- ▶ Cette prolifération touche même certains fabricants en interne comme IBM
 - ▶ qui va commencer par normaliser ses codes avec, BCDIC
 - ▶ qui est un code 6-bits descendant direct du code d'Hollerith
 - ▶ aussi utilisé notamment par d'autres fabricants
- ▶ Dans la seconde moitié de la décennie, l'armée des EU lance un projet pour élaborer un premier standard : FIELDATA.

- ▶ dans les années 1960, la prolifération continue
- ▶ mais avec un premier effort de standardisation :

- ▶ 1963, 1965, 1967 : ASCII (American Standard Code for Information Interchange) conçu à l'initiative de l'American Standards Association (ASA) et d'autres organisations de normalisation
- ▶ 1964 : EBCDIC (Extended Binary Coded Decimal Interchange)

JdCpropre à IBM

- ▶ 1965 : ECMA-6 développé en relation avec ASCII et ISO/CEI 646
- ▶ 1967 : ISO/CEI 646 développé en relation avec ASCII et ECMA-6
- ▶ 1969 : JIS X 0201 variante et extension JIS (*Japanese Industrial Standards*) de ISO/CEI 646 —ce standard sera décrit plus loin dans la sous-partie 4, p.65—

Ainsi que : transcode —IBM— , DEC Six Bits —variante 6-bits d'ASCII—, DEC RADIX 50, CDC display code, ASCII bang-bang , . . .

- ▶ À la fin des années cinquante, la multiplication des codages de caractères apparaît déjà comme un problème
- ▶ Plusieurs organisations de standardisation

CCITT, ISO, ASA, BSI, ECMA

- ▶ vont donc former des groupes de travail pour leur standardisation
- ▶ qui vont travailler en étroite collaboration.
- ▶ La première rencontre eut lieu en janvier 1961 à Paris sous l'égide du TC97 de l'ISO.
- ▶ Leur coopération va aboutir à la publication de différents standards entre 1963 et 1967
- ▶ dont le contenu est similaire :

- ▶ ASCII—1963–1967—
- ▶ ECMA-6 —1965—
- ▶ ISO/CEI 646-IRV—1967—

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
x0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
x1	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
x2	SP	!	”	#	⌘	%	&	,	()	*	+	,	-	.	/
x3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
x4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
x5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	-
x6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
x7	p	q	r	s	t	u	v	w	x	y	z	{		}	—	DEL

FIGURE 2 – Standard ISO/CEI 646-IRV

- Dans leurs versions finales, ASCII – ECMA-6 – ISO/CEI 646 sont des encodages 7-bits de $2^7 = 128$ caractères comprenant :

- les 26 lettres de l'alphabet latin majuscules et minuscules

mais sans aucun signe diacritique

- les chiffres décimaux
- différents symboles

! " # \$ % & ' () * + , - . / ...

- ainsi que des caractères de contrôle

- ▶ Les caractères sont regroupés par type.
- ▶ Les premiers 32 emplacements —0x00–0x1F— sont ainsi réservés aux caractères de contrôle C0 :

- ▶ **communication entre appareils :** NUL , (N)ACK —(Negative) Acknowledgement—, EOT —End of Transmission—, (SE)TX —Start|End of Text—, DC{1,2,3,4} —Device Control—, ...
- ▶ **gestion des erreurs :**
- ▶ **séquences d'échappement :** ESC —Escape—, SI —Shift in—, SO —Shift out—
- ▶ **formatage du texte :** CR —retour de chariot \r—, LF —saut à la ligne \n—, HT —tabulation horizontale \t—, VT —tabulation verticale \v—, ...

- ▶ Ces caractères se trouvent au début de la table du fait que le caractère NUL devait avoir tous ces bits à zéro
- ▶ et ne pouvait donc se situer qu'en première position.

- ▶ Les emplacements suivants reviennent aux caractères imprimables

sauf le dernier qui revient au caractère de contrôle C1

DEL

- ▶ soit un total de $0x7E - 0x20 = 94$

c'est important pour la suite. . .

- ▶ eux-mêmes groupés par

- ▶ symboles et marques de ponctuation
- ▶ chiffres
- ▶ lettres majuscules et majuscules.

- ▶ Les caractères imprimables sont positionnés dans la table en fonction leur ordre de tri.

- ▶ À l'instar du code Baudot, ASCII-ISO/CEI 646 mélange donc caractères imprimables et caractères de contrôle.
- ▶ qui occupent $\frac{1}{4}$ d'une table voulue très compacte
- ▶ L'inclusion de ces derniers provient du fait que, comme le nom *American Standard Code for Information Interchange* l'indique
- ▶ ces standards ont été avant tout conçus pour le transfert d'information
- ▶ Dans les années 1960, les communications passent encore par des flux.
- ▶ Il n'existe pas encore de distinctions entre couches introduites par les modèles comme TCP ou OSI.

pas de distinction entre, p. ex., la couche transport et la couche application

- ▶ Une de ses premières machine a utiliser ASCII fut d'ailleurs un téléscripteur en remplacement du code ITA2
- ▶ C'est aussi l'époque où les ordinateurs commencent à être équipés de terminaux
- ▶ permettant, par exemple, de transférer des données vers l'ordinateur
- ▶ et d'afficher le résultat des commandes via une imprimante ou un écran.
- ▶ Là aussi, les caractères de contrôle sont utilisés pour formater les données et les sorties.
- ▶ D'où la nécessité de limiter la taille des codes, pour limiter les volumes de données transférés

avec tous les inconvénients que cela ne manqua pas de susciter très rapidement

- ▶ la question de la longueur du code fit l'objet de long débat (MACKENZIE 1980)
- ▶ le rejet d'un code modal faisait d'ISO/CEI 646 un code d'au moins 7-bits
au regard du nombre des codes retenus comme nécessaires
- ▶ un code 8-bits fut envisagé
et présentait différents avantages d'un strict point de vue numérique
- ▶ l'arbitrage fut soumis au vote
- ▶ et une longueur de 7-bits l'emporta
- ▶ au regard des coûts de communication mais aussi de la fiabilité des matériels
- ▶ mais aussi d'autres considérations
p. ex. l'utilisation d'un bit de parité lors des opérations lecture-écriture qui ne pouvaient à l'époque qu'enregistrer 8-bits comme les bandes perforées et qui ne laissaient donc que 7 bits disponibles

- ▶ après négociations notamment du fait

- ▶ de l'implication de parties prenantes provenant d'horizons très divers

télécommunication, constructeurs informatique, programmeurs, armée, organismes de standardisation nationaux et internationaux. . .

- ▶ et d'un nombre de positions limitées

- ▶ contribution des concepteurs des langages de programmation

Certains caractères ont été inclus pour des langages COBOL, ALGOL, PL/I

- ▶ L'introduction de certains caractères fut aussi motivée pour former des lettres diacritiques

- ▶ comme ,  et  pour les accents

- ▶ reprenant ainsi une technique datant des machines à écrire

- ▶ L'ISO/CEI 646 diffère toutefois de ASCII en ce qu'il normalise des variantes nationales.

dont ASCII fait parti

- ▶ Les 32 premiers caractères de contrôle sont invariants.
- ▶ Parmi les 94 caractères imprimables,

- ▶ 82 sont invariants
- ▶ et 12 peuvent changer d'un code à l'autre.

norme AFNOR Z62010 pour la France

- ▶ Pour la distinguer des variantes introduites par la suite, la version commune du standard est appelée ISO/CEI 646-IRV —IRV pour *International Reference Version*.

- ▶ EBCDIC est un codage 8-bits utilisé à partir 1964 par IBM avec l'introduction de la série System/360.
- ▶ Il s'agit d'une extension de codes 6-bits BCD(IC).
 - ▶ Comme ASCII, il comporte des caractères imprimables et caractères de contrôle mais en nombre plus important (64)
 - ▶ Il sont aussi disposés en début de table. L'organisation des codes est toutefois très différente
 - du fait de la rétrocompatibilité avec BCD(IC)

- ▶ EBCDIC est complètement incompatible avec ISO/CEI 646
- ▶ IBM a participé aux travaux du sous-comité X3.2.
- ▶ Toutefois, EBCDIC a été préféré à ASCII par souci de compatibilité avec les cartes perforées utilisant BCDIC.

Ce qui est parfois interprété comme une manière de rendre les clients d'IBM captifs

- ▶ EBCDIC fut un JdC très utilisé jusqu'au années 1980

du fait des parts de marché d'IBM

- ▶ internationalisation avec de nombreuses variantes.

y compris pour le japonais —kanji—ou le chinois —hànzì—

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	ST	HT	SSA	DEL	EPA	RI	▣	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	OSC	NEL	BS	ESA	CAN	EM	PU	▣	IS	IS	IS	IS
2x	PAD	HOP	BPH	NBH	IND	LF	ETB	ESC	HTS	HTJ	VTS	PLD	PLU	ENQ	ACK	BEL
3x	DCS	PU	SYN	STS	CCH	MW	SPA	EOT	SOS	SGC	SCI	CSI	DC	NAK	PM	SUB
4x											\$.	<	(+	
5x	&										!	£	*)	;	~
6x	-	/									!	,	%	_	>	?
7x										`	:	#	@	'	=	"
8x		a	b	c	d	e	f	g	h	i						
9x		j	k	l	m	n	o	p	q	r						
Ax		~	s	t	u	v	w	x	y	z						
Bx																
Cx	{	A	B	C	D	E	F	G	H	I						
Dx	}	J	K	L	M	N	O	P	Q	R						
Ex	\		S	T	U	V	W	X	Y	Z						
Fx	0	1	2	3	4	5	6	7	8	9						APC

FIGURE 3 – EBCDIC-US

- ▶ Le développement de l'informatique personnelle à partir de la seconde moitié des années 1970

par de nombreux fabricants

- ▶ conduit à la multiplication des encodages maison pour la plus part codés sur 8-bits :

- ▶ Comodore : PETSCII (*PET Standard Code of Information Interchange*),
- ▶ Atari : Atari ST character set, ATASCII (*ATARI Standard Code for Information Interchange*)
- ▶ Sinclair Research : ZX80 code 8-bits sans aucun lien avec ISO/CEI 646
- ▶ Amstrad : Amstrad CPC character set, Amstrad CP/M Plus character set
- ▶ Digital Research : GEM *character set*
- ▶ DEC : *Multinational Character Set*, *National Replacement Character Set*
- ▶ Apple : Mac OS Roman, . . .
- ▶ NeXT : NeXT character set, basé sur Postscript Character Set
- ▶ IBM : Code page 437 et quelques centaines d'autres pour de nombreuses écritures. . .
- ▶ Microsoft, . . .

- ▶ et cela jusqu'à la fin des années 1990
- ▶ c'est aussi à cette période qu'apparaissent les pages de code pour gérer cette profusion —voir plus loin, p. 94—

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	Ç	û	é	â	ä	à	ç	ê	ë	è	ï	î	ï	Ä	Å	
9x	É	æ	Æ	ó	ô	õ	û	ü	ÿ	Ö	Ü	€	£	¥	₧	f
Ax	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	½	¼	¡	«	»	
Bx	■	■	■													
Cx	L	L	T	T	-	+	T	T	L	F	L	T	T	=	T	L
Dx	L	T	T	L	L	F	F	T	T	J	F	■	■	■	■	■
Ex	α	β	Γ	π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	φ	ε	η
Fx	=	±	≥	≤	∫	∫	÷	≈	°	·	·	√	≈	≈	■	■

(a) cp437 en mode texte

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	◊	◊	▼	♦	♣	♣	•	□	◊	■	♠	♠	♠	♠	◊
1x	▶	◀	:	!!	¶	§	—	↑	↓	→	←	L	↔	▲	▼	
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	Ç	û	é	â	ä	à	ç	ê	ë	è	ï	î	ï	Ä	Å	
9x	É	æ	Æ	ó	ô	õ	û	ü	ÿ	Ö	Ü	€	£	¥	₧	f
Ax	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	½	¼	¡	«	»	
Bx	■	■	■													
Cx	L	L	T	T	-	+	T	T	L	F	L	T	T	=	T	L
Dx	L	T	T	L	L	F	F	T	T	J	F	■	■	■	■	■
Ex	α	β	Γ	π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	φ	ε	η
Fx	=	±	≥	≤	∫	∫	÷	≈	°	·	·	√	≈	≈	■	■

(b) cp437 avec caractères graphiques

FIGURE 4 – cp437

- ▶ Ces JdC entretiennent des liens plus ou moins ténu avec ASCII–ISO/CEI 646.
- ▶ comme le montre le cp437 utilisé par la série IBM-PC pour les États-Unis

- ▶ Ils ont très souvent en commun les caractères imprimables

qui sont identiques à ISO/CEI 646–ASCII

- ▶ les caractères de contrôle étant eux souvent remplacés par des symboles
- ▶ D'autre part, l'octet s'étant imposé entre temps dans les architectures informatiques,
- ▶ ces JdC utilisent aussi le bit de poids fort laissé vacant pour coder la partie haute de la table pour augmenter le nombre de caractères disponibles.

parmi lesquels on peut noter diacritiques et ligatures

- ▶ Les ordinateurs qui utilisent ces JdC se caractérisent par —au début des années 1980 du moins— :
 - ▶ des architectures 8-bits
 - MOS Technology 7501, MOS Technology 6510/8500, Zilog Z80(A), Intel 8088. . .
 - ▶ et une mémoire de ~ 16 Kb
 - avec des possibilités d'extensions plus moins larges
- ▶ les ressources sont ici limitées moins pour des raisons techniques
 - les processeurs 16-bits sont commercialisés depuis le milieu des années 1970
- ▶ que commerciales, la compétition —féroce— se faisant aussi sur les prix

- ▶ Parallèlement à ces mappes maison, d'autres JdC sont publiés par les organismes de standardisation nationaux et internationaux.
- ▶ avec l'ajout de variantes nationales de ISO/CEI 646
- ▶ et les standards :
 - ▶ ISO/CEI 2022
 - ▶ ISO/CEI 8859

- ▶ ISO/CEI 2022 est un standard ISO qui n'est pas à proprement parler un JdC
- ▶ mais plutôt un ensemble de règles permettant d'alterner entre différents encodages de JdC
- ▶ au moyen de séquences d'échappement qui identifient chaque encodage de façon unique.
- ▶ pour l'échange d'information
- ▶ Il a été publié la première fois en 1973 et sa dernière révision date de 1994.

- ▶ ISO/CEI 2022 est basé sur ISO/CEI 646 dont il généralise les règles de variations pour l'internationalisation
- ▶ Même si ISO/CEI 2022 est un encodage 8-bits, les encodages inclus doivent être compatibles avec des canaux 7-bits.
- ▶ Il repose sur la propriété de ISO/CEI 646 qu'un encodage 7-bits permet de définir :
 - ▶ 94 caractères imprimables
 - ▶ et 32 caractères de contrôle
- ▶ Pour assurer la compatibilité avec des codes 8-bits, les caractères sont répartis dans des tables de 256 caractères.
- ▶ Ces tables sont ensuite divisées en quatre zones : G0, G1, G2, et G3
 - informellement désignées C0, GL, C1 et GR pour souligner leur relation avec ISO/CEI 646.
- ▶ C0 et GL ont en effet les mêmes limites que ISO/CEI 646.
- ▶ GL et GR sont attribués aux caractères graphiques.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x																
3x																
4x																
5x																
6x																
7x																
8x																
9x																
Ax																
Bx																
Cx																
Dx																
Ex																
Fx																

FIGURE 5 – Groupes du standard ISO/CEI 2022

- ▶ Les codes des caractères doivent donc tomber dans les plages 0x21–0x7E
soit 94 caractères comme pour ISO/CEI 646, ou 0x20–0x7F —96 caractères—
- ▶ ISO/CEI 2022 n'est toutefois pas limité aux encodages 8-bits car il permet l'utilisation de codes multi-octets :
 - ▶ Comme on le verra plus loin dans la partie consacrée aux caractères CJK, p.65
 - ▶ en utilisant deux octets et 94 emplacements
 - ▶ on peut en effet encoder $94 \times 94 = 8\,836$ caractères, avec trois octets, $94^3 = 830\,584$ caractères, . . .
- ▶ La seule contrainte est que G0 soit de type 94ⁿ.

- ▶ Exemples de séquences d'échappement : voir ISO/CEI 8859, JIS X 0201
- ▶ La séquence d'échappement n'annonce pas seulement quel encodage est utilisé mais aussi le nombre de bytes utilisés.
- ▶ De nombreux JdC se conforment à ISO/CEI 2022.
 - plus de 200 ont été enregistrés jusqu'au début des années 2000
- ▶ La procédure d'enregistrement d'un encodage dans ISO/CEI 2022 par l'ISO est définie par ISO/IEC 2375.

- ▶ À partir du milieu des années 1980, l'ISO commença à publier une série de standards regroupés sous l'appellation ISO/CEI 8859.
- ▶ Ce standard finira par comprendre un ensemble de 15 codes 8-bits
- ▶ pour les langues parlées en Europe
- ▶ ainsi que l'arabe (6), l'hébreu (8), le turc (9) ou encore le thaï (11).
- ▶ Les différentes parties de ISO/CEI 8859 reprennent d'autres standards comme ECMA-94 —parties 1 à 4— ou ELOT 928 pour le grec (7).

- ▶ ISO/CEI 8859 est conforme à ISO/CEI 2022 et adopte la répartition des codes en quatre zones :
 - ▶ Les trois premiers groupes sont communs à tous les standards ISO/CEI 8859-n.
 - ▶ Le groupe GL est identique à ISO/CEI 646.
 - ▶ Les groupes C0 et C1 ne sont pas définis par ISO/CEI 8859-n mais par ISO 6429.
 - ▶ Enfin, le groupe GR est spécifique à chaque codage.
- ▶ ISO/CEI 8859 a pour séquences d'échappement : ESC - F, ESC . F, ESC / F.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x																
3x																
4x																
5x																
6x																
7x																
8x																
9x																
Ax																
Bx																
Cx																
Dx																
Ex																
Fx																

(a) Groupes

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	:	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI		
9x	DCS	PU	PU	STC	CCH	MW	SPA	EPA	SOS	SGC	SCI	CSI	ST	OSC	PM	APC
Ax		¡	¢	£	¤	¥	¦	§	-	©	ª	«	¬	-	®	-
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

(b) ISO/CEI 8859-1 combiné avec ISO 6429

FIGURE 6 – Standard ISO/CEI 8859

- ▶ ISO/CEI 8859-1

ou Latin-1

- ▶ est destiné à l'Europe de l'Ouest ajoute à ISO/CEI 646-IRV un ensemble de symboles i ¢ £ § ¨ © « ¶ · , ¹ º » ¼ —parmi lesquels des caractères typographiques— ainsi que de lettres diacritiques ß.

- ▶ Il a été conçu par l'ECMA qui le publia sous l'intitulé ECMA-94

ce JdC s'inspirant à son tour du *Multinational Character Set* créée par DEC

- ▶ ISO/CEI 8859-1

- ▶ ne compte pas l'intégralité des diacritiques utilisées et laisse de côté certains caractères rares.


- ▶ Il comporte toutefois quelques surprises comme l'omissions de la ligature œ

Œ et œ figuraient dans une première version mais ont été supprimés dans des circonstances décrites dans (ANDRÉ 1996).

- ▶ alors que ÿ est, lui, inclus —mais pas sa majuscule—.

- ▶ Le vide laissé en C1 sera notamment « comblé » par MS avec la page de code WINDOWS 1252
- ▶ qui procédera de même pour d'autres variantes de ISO/CEI 8859 pour ajouter des caractères.

Le code page WINDOWS 1251 pour l'alphabet cyrillique modifie ainsi ISO/CEI 8859-5, WINDOWS 1253 pour l'alphabet grec reprend, lui, ISO/CEI 8859-7, . . .

- ▶ D'autre part, l'ISO publiera à la fin des années 1990 une version modifiée de ISO/CEI 8859-1, ISO/CEI 8859-15
- ▶ qui supprime huit caractères dont les fractions ainsi que 
- ▶ pour les remplacer par différentes lettres

comme notamment ,  et  ainsi par le symbole 

- ▶ UNICODE est une extension directe d'ISO/CEI 8859-1.
- ▶ Ce dernier occupe en effet les 256 premiers emplacements de la table des codes
- ▶ et correspond au bloc Basic Latin.

- ▶ En parallèle, certaines applications cessent de déléguer la gestion de l'encodage à l'OS
- ▶ alors que de nouveaux usages apparaissent comme la composition de documents
- ▶ qui se développe dans les années 1970

d'abord pour la rédaction de documentation —troff—

- ▶ C'est, p. ex., le cas de T_EX :

- ▶ T_EX est un logiciel de composition pour documents scientifiques
- ▶ conçu par Donald Knuth à partir de 1977–1978 lors d'un congé sabbatique.
avec l'aide de nombreuses personnes, étudiants ou collègues
- ▶ T_EX fonctionne de façon indépendante de l'OS
- ▶ et utilise ses propres encodages, polices et format d'affichage —DVI—.

- ▶ T_EX a été conçu alors que les caractères disponibles sont en nombre très limités
- ▶ et n'incluent donc pas, par exemples, les symboles mathématiques.
- ▶ Les caractères ne pouvant être entrés au moyen du clavier sont identifiés au moyen de commandes spécifiques

```
\alpha pour  $\alpha$ , \int pour  $\int$ , . . .
```

- ▶ Il en va de même pour les lettres diacritiques :

- ▶ `P\'olya Gy\"orgy` pour écrire le nom du mathématicien hongrois Pólya György
- ▶ ou encore `Erd\H{o}s P\'al` pour écrire le nom de son compatriote et confrère Erdős Pál

- ▶ T_EX utilise une technique de composition datant des machines à écrire
- ▶ reprise pour les terminaux et qui perdure jusqu'à aujourd'hui

- ▶ Au départ, T_EX nécessitait que le code source du document soit écrit en ASCII
- ▶ mais utilise son propre encodage en interne.
- ▶ D. Knuth a ainsi conçu un ensemble JdC 7-bits pour T_EX

OT1 —T_EX text similaire à ASCII—, OML —T_EX math italic—, OMS —T_EX math symbol—, . . .

- ▶ ainsi que des polices

l'absence de polices de caractères adaptées allant avec l'absence de JdC

- ▶ avec un logiciel conçu en parallèle à T_EX : METAFONT





	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	-	.	x	*	÷	◇	±	≠	⊕	⊖	⊗	⊙	⊛	⊜	∘	⋅
1x	≠	≡	⊆	⊇	≤	≥	≪	≫	~	≈	⊂	⊃	⋈	⋉	⋊	⋌
2x	←	→	↑	↓	↔	↗	↘	≈	⇐	⇒	⇑	⇓	⇔	↖	↙	∝
3x	'	∞	∈	∋	△	▽	♢	◊	∇	∃	¬	∅	℔	ℑ	ℒ	⊥
4x	ℵ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ
5x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ
6x	⊢	⊣	⊤	⊥	⊦	⊧	{	}	⟨	⟩			↑	↓	\	!
7x	√	⊔	⊓	⊒	⊑	⊒	⊓	§	†	‡	¶	♣	◇	♥	♠	♠

FIGURE 7 – TeX OMS

- ▶ Différentes extensions ont ensuite été proposées par des utilisateurs
- ▶ notamment pour ajouter des symboles diacritiques : Cork, LY1, . . .
- ▶ différents compilateurs ont aussi été créés spécifiquement pour le japonais : jT_EX, pT_EX
- ▶ Le support d'Unicode dans T_EX demeure toutefois très partiel au travers de packages
- ▶ les sources de T_EX ayant été gelées par D. Knuth en 1989
- ▶ ce qui a conduit à la création d'autres compilateurs pour le langage T_EX utilisant directement UNICODE

Ω, X₃T_EX, LuaT_EX

- ▶ les JdC sont à l'époque encore très pauvres typographiquement

- ▶ p. ex., seul le tiret court  est présent et n'est pas distingué du symbole moins , manquent les tirets cadratin  ou (d|s)emi-cadratin ,...

les chose commençant à changer dans les années 1980 dans ce cas avec le cp WIN-DOWS 1252 dans la portion C1 de ISO/CEI8859-1 ou MacRoman

- ▶ de mêmes pour les différents type d'espaces typographiques : insécable, sécable fine, insécable fine, cadratin, ½ cadratin. . .
- ▶ guillemets
- ▶ . . .

- ▶ car, comme le note D. Knuth (KNUTH 1984),

« the standard ASCII code for computers was not invented with book publishing in mind. However, your terminal probably does have two flavors of single-quote marks, namely `'` and `'` »

les JdC conçus jusque là le sont pour des application en mode texte, et non pour composer des textes

► Autres exemples de logiciels utilisant des JdC propres :

- Adobe : *PostScript Standard Encoding*
- Ventura : *Ventura International* pour Ventura Publisher dérivé de GEM
- WordPerfect : *Iconic Symbols*
- Lotus : *Lotus International Character Set* dérivé de DEC *Multinational Character Set*, *Lotus Multi-Byte Character Set*

► et leurs propres polices de caractères :

- comme Zapf Dingbat conçues par Hermann Zapf
- et notamment popularisée par Adobe
 - de par leur inclusion dans les polices PostScript
- qui en fit don au consortium UNICODE et dont les codes se situent dans le bloc Dingbats
- de même que certains des caractères *Iconic Symbols* de WordPerfect furent inclus dans le standard

- ▶ à partir des années soixante, plusieurs standards furent publiés en URSS :

- ▶ GOST10859 qui est un code à longueur variable de 4-7 bits avec une structure hiérarchique

- ▶ 4-bits : chiffres
- ▶ 5-bits : chiffres puis symboles
- ▶ 6-bits : chiffres puis symboles puis caractères cyrilliques majuscules avec une variante où les caractères cyrilliques sont remplacés par des caractères latins majuscules
- ▶ 7-bits : chiffres puis symboles puis caractères cyrilliques en capitales puis lettres latines en capitales et une série d'autres symboles

- ▶ série KOI —Код обмена информацией (Code pour l'échange d'informations)— :

- ▶ KOI-7
- ▶ KOI-8 (KOI-8 GOST 19768-74) : extension 8-bits de ISO/CEI 646
- ▶ dont les variantes

KOI-8R, KOI-8U, KOI-8T,...

- ▶ seront reprises par différentes pages de code et enregistrés dans la base de l'IANA

- ▶ en République fédérative socialiste de Yougoslavie, « YUSCII », nom informel de différents codes 7-bits :

SLOSCII —slovène—, CROSCII —croate—, SRPSCII —serbe—, MAKSCII —macédonien—

Standard KOI-8

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x																
9x																
Ax																
Bx																
Cx	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
Dx	п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
Ex	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
Fx	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	

FIGURE 8 – Standard KOI-8

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	Ž	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	Š	Đ	Ć	Č	_
6x	ž	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	š	đ	ć	č	DEL

(a) JUSI.B1.002

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	Ж	А	Б	Ц	Д	Е	Ф	Г	Х	И	Ј	К	Л	М	Н	О
5x	П	Љ	Р	С	Т	У	В	Њ	Ѕ	З	Ш	Ћ	Ќ	Ч	_	
6x	ж	а	б	ц	д	е	ф	г	х	и	ј	к	л	м	н	о
7x	п	љ	р	с	т	у	в	њ	ѕ	з	ш	ћ	ќ	ч		DEL

(b) JUSI.B1.003

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	Ж	А	Б	Ц	Д	Е	Ф	Г	Х	И	Ј	К	Л	М	Н	О
5x	П	Љ	Р	С	Т	У	В	Њ	Ѕ	З	Ш	Ћ	Ќ	Ч	_	
6x	ж	а	б	ц	д	е	ф	г	х	и	ј	к	л	м	н	о
7x	п	љ	р	с	т	у	в	њ	ѕ	з	ш	ћ	ќ	ч		DEL

(c) JUSI.B1.004

FIGURE 9 – Standards «YUSCII»

CJKV

- ▶ les JdC évoqués précédemment concernent essentiellement des écritures alphabétiques
- ▶ qui est loin d'être le seul système utilisé
- ▶ cas particulier de l'Asie de l'Est

- ▶ nombreuses particularités de l'écrit dans différents pays de l'Asie de l'Est
- ▶ notamment, l'utilisation conjointe de plusieurs écritures
- ▶ mais avec les caractères chinois en commun
- ▶ Les sinogrammes sont composés de différents types d'unités logographiques :
 - ▶ les pictogrammes qui représentent un objet
 - ▶ les idéogrammes qui représentent un objet
 - ▶ les idéophonogrammes qui résultent de compositions sémantico-phonétiques à partir des pictogrammes et idéogrammes
 - ▶ Les idéophonogrammes sont composés d'au moins deux composants : un composant sémantique qui suggère le sens du mot et un composant phonétique qui suggère sa prononciation

- ▶ La liste de ces caractères est longue
 - plusieurs dizaines de milliers
- ▶ et sujette à discussion
- ▶ de nouveaux sinogrammes continuent d'apparaître
- ▶ et d'autres tombent en désuétude
- ▶ Il existe de plus des variantes entre langues
 - dans la liste des caractères utilisés mais aussi dans leur calligraphie
- ▶ mais aussi pour une même langue
 - chinois traditionnel utilisé à Taïwan, Hong-Kong ou Macao et chinois simplifié promu par la RPC
- ▶ et aussi plusieurs façons de les trier

- ▶ les caractères han sont composés de traits

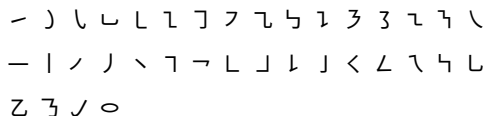





FIGURE 10 – Bloc UNICODE CJK Strokes

- ▶ au moins jusqu'à 84 comme le taito U+3106C 
- ▶ qui combine  (nuageux) et  (dragon en vol)
- ▶ et signifie ?
- ▶ ce qui nécessite une résolution suffisante pour leur compréhension

écrans, imprimantes

- ▶ du fait de leur grand nombre, les caractères han nécessitent aussi des méthodes particulières de saisie

- ▶ Les caractères chinois ont été adoptés par d'autres langues
- ▶ mais, pour cela, ont dû être adaptés à ces langues

- ▶ La langue chinoise se caractérise en effet par une quasi absence de flexions
les mots n'ont généralement qu'une forme grammaticale

- ▶ et l'utilisation de tons

- ▶ À l'inverse du chinois, le japonais est une langue

- ▶ atonale
- ▶ agglutinante
- ▶ utilisant des suffixes comme marqueurs grammaticaux

- ▶ C'est pourquoi l'écriture de la langue japonaise combine les hànzi —漢字, appelés kanji en japonais—avec des hiraganas —平仮名 ou ひらがな—

- ▶ qui est une des deux écritures syllabiques —仮名, kana—utilisée au Japon

à côté des katakanas —片仮名 ou カタカナ—, plutôt réservés aux mots d'origine étrangères : エンコーディング (« enkōdingu », *encoding*)

- ▶ Aux caractères han et kanas s'ajoutent

- ▶ les rōmaji

caractères latins —chasse pleine : r o m a j i—

- ▶ soit quatre écritures auxquelles s'ajoutent différentes méthodes de translittération

► l'écriture de la langue coréenne combine aussi

- ▶ sinogrammes —hanja—
- ▶ et caractères syllabiques appelés hangeul (한글)
- ▶ composés de lettres —jamos (자모)—organisées en blocs

한 :    , 글 :   

- ▶ les hanja restent en utilisation pour, p. ex., distinguer les homonymes

le koréen n'est pas, lui-aussi, une langue tonale

- ▶ ou pour les textes classiques ou légaux

- Au Viêt Nam, les caractères idéographiques —chữ hán et chữ nôm—
- ont été progressivement remplacés au XX^e siècle par un système d'écriture dérivé de l'alphabet latin —quốc ngữ—





Les caractères han sont aujourd'hui réservés à des usages historiques ou religieux.

- ▶ **1969** : JIS X 0201 —7 ビット及び 8 ビットの情報交換用符号化文字集合 – *7 and 8 coded character sets for information interchange*—
- ▶ **1971** : IBM Kanji System
- ▶ **1974** : KS C 5601-1974
- ▶ **1978** : JIS X 0208 —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 – *7 and 8 byte coded KANJI sets for information interchange*—
- ▶ **1980** : CCCII —中文資訊交換碼 – *Chinese Character Code for Information Interchange*—
- ▶ **1980** : GB/T 2312-1980 *Guobiao Standards*
- ▶ **1984** : Big5
- ▶ **1984** : Shift JIS
- ▶ **1986** : KS X 1001 —*Code for Information Interchange (Hangul and Hanja)*— 정보교
환용부호계 (한글및한자)—

- ▶ JIS X 0201 est un code 8-bits contenant, dans l'ordre :

- ▶ des caractères de contrôle (C0)
- ▶ les caractères latins —rōmaji—
- ▶ d'autres caractères de contrôle (C1)
- ▶ et les katakana.

- ▶ Les 94 premiers caractères imprimables sont quasiment identiques à ceux de ISO/CEI 646

le symbole du Yen  se substitue à la barre oblique inversée  et le trait suscrit  à la tilde 

- ▶ JIS X 0201 peut être encodé avec 7-bits utilisant un shift ou 8-bits : ESC) I, ESC) F

- ▶ En mode 7-bits, l'encodage utilisant les premiers 96 caractères, soit les 96 suivants.
- ▶ Le caractère de contrôle 0x0E sert à embrayer en mode katakana et le caractère de contrôle 0x0F embraye en mode latin.

Standard JIS X 0201

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC	DC	DC	DC	NAK	SYN	ETB	CAN	EM	SUB	ESC	IS	IS	IS	IS
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI		
9x	DCS	PU	PU	STS	CCH	MW	SPA	EPA	SOS	SGC	SCI	CSI	ST	OSC	PM	APC
Ax		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ヱ	ォ	ャ	ュ	ヨ	ッ
Bx	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
Cx	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
Dx	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	°	°
Ex																
Fx																

FIGURE 11 – Standard JIS X 0201

- ▶ JIS X 0201 a été repris par le cp 952 d'IBM et dans le bloc Halfwidth and Fullwidth Forms du standard UNICODE.
- ▶ La même disposition a été reprise par le standard koréen KSC 5601-1974
- ▶ pour les caractères hangeules
 - et à son tour repris repris comme CP par IBM —cp 1040—ainsi que par le standard UNICODE dans le bloc Halfwidth and Fullwidth Forms
- ▶ un standard comme JIS X 0201 permet une écriture *acceptable* du japonais
- ▶ mais qui est loin d'être *satisfaisante*
- ▶ un codage 8-bits étant trop étroit, les mappes suivantes vont donc recourir à un code multi-octets
- ▶ et là les choses se compliquent un peu...

- ▶ D'abord appelé JIS C 622, JIS X 0208 est un encodage 2×8 -bits compatible avec ISO/CEI 2022.
- ▶ Pour cela, l'espace des codes est divisé en 94 blocs d'une longueur de 94 caractères
 - organisés en lignes de 94 cellules
- ▶ Chaque ligne regroupe des caractères de même type :
 - ▶ les huit premières lignes rassemblent ponctuation, symboles, caractères alphanumériques, hiragana, katakana, grec, cyrillique, boîtes
 - ▶ les lignes 0x30 à 0x74 rassemblent 69 lignes de kanji répartis en deux niveaux de compatibilité (L1 : 0x30-0x4F et L2 : 0x50-0x7F)

- ▶ Le code utilise un paire de deux octets allant de 1 à 94 appelées *kuten*
 - ▶ mot formé à partir de 区 —ligne— et de 点 —cellule—.
 - ▶ Le premier octet indexe une ligne. Il est obtenu en soustrayant 32 à la position dans la table.
 - ▶ Le second octet indexe, lui, la cellule.
 - ▶ Par exemple, la ligne n° 3 rassemble les rōmaji —chasse pleine— dans une disposition identique à celle de ISO/CEI 646-IRV et la ligne n° 5 rassemble, elle, les katakana.
- ▶ matrice à 2 dimensions indexée par 区 —*i*— et de 点 —*j*— pouvant contenir $94^2 = 8\,836$ caractères

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x		1	2	3	4	5	6	7	8							
3x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4x	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
5x	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
6x	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
7x	80	81	82	83	84											

1–2 : caractères
 spéciaux
 3 : romaji
 4 : hiragana
 5 : katakana
 6 : grec
 7 : cyrillique
 8 : boîtes
 16–84 : kanji

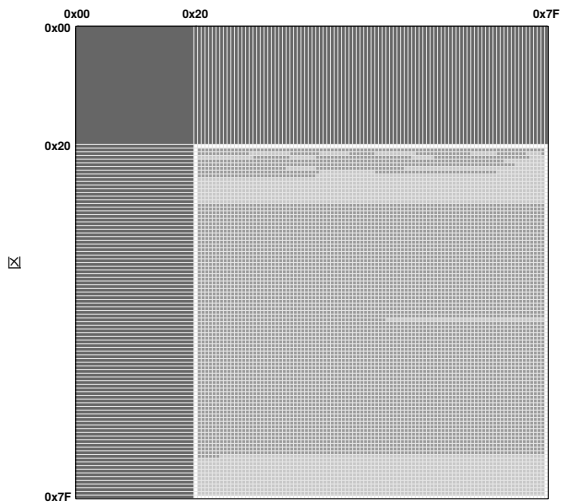
(a) Premier octet (indice des lignes)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x																
3x	0	1	2	3	4	5	6	7	8	9						
4x		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z					
6x		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	P	q	r	s	t	u	v	w	x	y	z					

(b) Ligne n° 3 : rōmaji

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x		ア	ア	イ	イ	ウ	ウ	エ	エ	オ	オ	カ	ガ	キ	ギ	ク
3x	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
4x	ダ	チ	ヂ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	
5x	バ	バ	ヒ	ビ	ビ	フ	ブ	フ	ヘ	ベ	ベ	ホ	ボ	ボ	マ	ミ
6x	ム	メ	モ	ヤ	ヤ	ユ	ユ	ヨ	ヨ	ラ	リ	ル	レ	ロ	ワ	ワ
7x	ヱ	ヱ	ヲ	ン	ヅ	カ	ケ									

(c) Ligne n° 5 : katakana



点

FIGURE 13 – Standard JIS X 0208

- ▶ La première version comporte 6 802 caractères en tout puis 6 879.
- ▶ JIS X 0208 peut être encodé de différentes façons.
 - ▶ Une première utilise les kuten directement dans l'intervalle 1–94, la valeur de la ligne pouvant être retrouvée en y soustrayant 32.
 - ▶ Une autre appelée EUC-JP utilise l'intervalle 0xA1 (161) à 0xFE (254) en mettant le 7e bit à 1.
- ▶ JIS X 0208 a été révisé à plusieurs reprises et incorporé dans d'autres JdC
 - comme Shift JIS
- ▶ et source d'inspiration pour d'autres JdC —voir plus loin—

- ▶ JIS X0208 fut notamment utilisé à partir de 1983 par la série PC-9800 de NEC.
- ▶ Les premiers modèles de cette —longue—série se caractérisent par
 - ▶ un processeur Intel 8086 16-bits
 - ▶ 128 Kb de mémoire (extensible jusqu'à 640 Kb)
 - ▶ toutefois, afin de dégager des ressources mémoire
 - ▶ les caractères kanji et katakanas étaient gravés en dur dans une ROM dédiée
 - ▶ comme pour le NEC PC-8001mkII SR sorti en 1985
 - ▶ le PC-8001 sorti en 1979 avait une ROM comportant les katakanas seulement
 - ▶ un processeur 8-bits
- ▶ une résolution couleur de 640×400 pixels

- ▶ les ordinateurs de la série des IBM-PC n'étant pas assez puissants pour gérer les kanji

processeur Intel 8088 8-bits, 14 Kb de mémoire

- ▶ IBM produisit une série destinée au marché japonais : IBM 5550

- ▶ avec un processeur Intel 8086 16-bits
- ▶ 128 Kb de mémoire
- ▶ une résolution de 1024×768 pixel

- ▶ les IBM 5550 utilisent le JdC IBM Kanji System développé dans les années 1970 en parallèle à JIS X 0208
- ▶ IBM a aussi proposé une extension EBCDIC pour la langue japonaise
- ▶ ainsi que d'autres constructeurs japonais

Fujitsu —JEF (*Japanese processing Extended Facility*)—, NEC —JIPS (*Japanese Information Processing System*)—, Hitachi —KEIS (*Kanji processing Extended Information System*)—

- ▶ ce qui précède illustre le fait que le support des kanji nécessite de pouvoir disposer

- ▶ d'une puissance de calcul
- ▶ d'une quantité de mémoire
- ▶ et d'une résolution

affichage, impression —commercialisation des premières imprimantes laser dans la seconde moitié des années 1970—

- ▶ suffisantes
- ▶ comme l'illustre aussi UNICODE

du fait de l'utilisation de tables

- ▶ ainsi qu'une méthode de saisie des caractères

combinaisons de katakanas pour le japonais

- ▶ Standard national de la RPC pour les sinogrammes simplifiés publié en 1980
obligatoire avant la publication du standard GB 18030 en
- ▶ Organisation similaire à JIS X 0208 :
 - ▶ Les caractères sont codés sur deux octets
appelés 区位 qūwèi
 - ▶ organisés sur une grille 94×94 découpées par lignes
 - ▶ la troisième ligne correspond par exemple à ISO/CEI 646-CN
qui est la version de la RPC de ISO/CEI 646
- ▶ Inclut, en plus des caractères latins, grecs, cyrilliques et japonais, les caractères zhuyin et pinyin hors ASCII.
- ▶ Pour encoder les points de code, il faut ajouter 160 au n° de ligne pour obtenir le premier octet et 160 au n° de colonne pour le second.
- ▶ Extensions :
 - ▶ GBK étend GB 2312 avec notamment des caractères ajoutés dans la version 1.1 d'UNICODE
 - ▶ GBK puis GB 18030 développé conjointement avec UNICODE.
Les caractères CJK présents dans les deux standards sont presque les mêmes, à l'exception de quelques caractères —*round-trip mappings*—

- ▶ CCCII est un codage conçu à Taïwan pour les sinogrammes traditionnels.
- ▶ Il reprend les principes constitutifs employés par JIS X 0208 et GB 2312
- ▶ mais utilise 3 octets pour une capacité maximale de $94^3 = 830\,584$ caractères.

deux octets est suffisant pour les hánzì les plus courants mais pas pour leur intégralité

- ▶ L'espace des caractères est divisé en 16 couches composées elles-mêmes de 6 plans consécutifs de taille 94×94 .

CCCI dispose donc les caractères dans une matrice à trois dimensions.

Couche	Plan	Description
1	1–6	Non-hanzi, hanzi
2	7–12	Hanzi simplifié
3–12	13–72	Variantes hanzi
13	73–78	Kana, kanji
14	79–84	Jamo, hangeul, hanja
15	85–90	Réservé
16	91–94	Autres caractères

- ▶ 53 940 positions effectivement attribuées après la dernière révision de 1987.
- ▶ les caractères sont ordonnés par clef de sinnogramme

Années 1980-1990

- ▶ dans les trois premières décennies, l'élaboration des JdC concerne essentiellement trois aires :

Amérique du Nord–Europe de l'Ouest, bloc de l'Est, pays de l'Asie de l'Est

- ▶ Dans les années 1980-1990, l'extension des JdC 8-bits se poursuit par le support d'autres écritures
- ▶ avec, notamment :

- ▶ 1986, 1990 : TIS-620 (Thai Industrial Standard 620-2533)
- ▶ 1988, 1991 : ISCII (Indian Script Code for Information Interchange), IS 13194
- ▶ 1993 : VSCII (Vietnamese Standard Code for Information Interchange), TCVN 5712 ISO-IR-180, à ne pas confondre avec VISCII qui désigne la translittération informelle du vietnamien n'utilisant que les 26 lettres de l'alphabet latin
- ▶ arabe, hébreu, . . .

- ▶ ISCII est un JdC pour différentes langues dérivées du brāhmī officiellement reconnues en Inde

bengali, le dévanâgarî, le goudjarâtî, le gourmoukhî, le kannada, le malayalam, l'odia, le tamoul et le télougou

- ▶ L'alphabet persan utilisé par le urdu, le sindhi et le kashmiri faisant l'objet d'un JdC distinct

le PASCII (*Perso-Arabic Script Code for Information Interchange*)

- ▶ ISCII est un codage 8-bits conforme à ISO/CEI 2022.
- ▶ qui s'appuie sur la propriété que si les alphabets de ces langues varient dans leur forme,
- ▶ ils partagent une structure phonétique commune.

- ▶ Les différents alphabets sont ainsi disposés de façon phonétiquement identique.

Note : ISO 15919 est un standard de translittération des écritures brahmiques

- ▶ Les glyphs sont obtenues par la combinaison de différents caractères.

- ▶ Ce principe de codage a notamment été motivé par l'élaboration de claviers utilisables partout en Inde.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x																
9x																
Ax		m̐	m̐	h	a	ā	i	ī	u	ū	ṛ	e	ē	ai	ē	o
Bx	ō	au	ō	k	kh	g	gh	ṇ	c	ch	j	jh	ṇ	ṭ	ṭh	ḍ
Cx	gh	ṇ	t	th	d	dh	n	ṇ	p	ph	b	bh	m	y	ṽ	r
Dx	ṛ	l	ṭ	ṭ	v	ś	ṣ	s	h	INV	ā	i	ī	u	ū	ṛ
Ex	e	ē	ai	ē	o	ō	au	ō								
Fx	0	1	2	3	4	5	6	7	8	9						

(a) ISO 15919

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x																
9x																
Ax		ँ	ं	ः	अ	आ	इ	ई	उ	ऊ	ऋ	ॠ	ऐ	ए	ॐ	औ
Bx	ओ	औ	ऑ	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ	ड
Cx	ढ	ण	त	थ	द	ध	न	न	प	फ	ब	भ	म	य	र	
Dx	ॠ	ॡ	ॢ	ॣ	।	॥	०	१	२	३	४	५	६	७	८	९
Ex	०	१	२	३	४	५	६	७	८	९						
Fx																

(b) dévanāgarī

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x																
9x																
Ax			ँ		अ	आ	इ	ई	उ	ऊ			ॠ	ॡ		
Bx	ॢ	ॣ		क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ	ड
Cx	ढ	ण	त	थ	द	ध	न		प	फ	ब	भ	म	य	र	
Dx		०			१		२	३	४	५	६	७	८	९		
Ex		०	१	२	३	४	५	६	७	८	९					
Fx		०	१	२	३	४	५	६	७	८	९					

(c) gourmoukhî

FIGURE 14 – Standard ISCII

- ▶ Jusqu'aux années 1980, la majorité des JdC sont encore codés sur 7–8 bits
- ▶ et ne couvrent qu'une ou, plus rarement, deux écritures et plus.
- ▶ On a toutefois pu voir qu'à la fin des années 1970
- ▶ commencent à apparaître des JdC codant conjointement un nombre plus étendu d'écritures.
- ▶ Cette tendance se poursuit dans les années 1980 :
 - ▶ 1980 : XCCS (Xerox Character Code Standard)
 - ▶ 1984 : T.51-ISO/IEC 6937
 - ▶ 1984 : TRON
 - ▶ ISO/CEI 10646 et UNICODE

- ▶ XCCS est un code 16-bits utilisé par Xerox à partir de 1980.
- ▶ qui fonctionne de façon similaire à JIS X 0208
 - ▶ le premier octet indexe un bloc, le second, le caractère dans le bloc
 - ▶ la taille des blocs n'est pas limitée à 94
- ▶ Élaboré à partir de 1978 et révisé en 1990,
- ▶ il inclut les écritures latines, l'arabe, l'hébreu, le grec, le cyrillique ainsi que les kanji et des symboles techniques.

► XCCS fut conçu

- non pas pour le transfert d'information et l'affichage en mode texte
- mais en relation avec le langage de balisage Interscript et le langage de description de pages Interpress.
- et pour la station de travail Xerox Star
- qui fut le premier ordinateur commercial à proposer une interface graphique
ainsi qu'une connexion Ethernet et un serveur d'impression entre autres caractéristiques aujourd'hui communes

► Il s'agit du précurseur direct d'UNICODE

- Il fut en effet principalement élaboré par Joe Becker
- qui rédigea ensuite la première mouture du standard UNICODE publiée en 1988.

- ▶ En 1983, l'ISO publia le standard T.51-ISO/IEC 6937.
- ▶ Il s'agit d'un encodage 8-bits pour les écritures latines dont les 128 premiers emplacements sont identiques à ISO/CEI 646.
- ▶ Sa particularité est d'encoder les lettres diacritiques latines sur deux octets.
 - ▶ Le premier indique le signe diacritique et le second, la lettre.
 - ▶ Il s'agit donc d'un encodage à longueur variable comportant 357 caractères —et pas seulement 256—.
- ▶ Cette façon d'encoder les caractères diacritique en combinant des graphèmes se retrouve dans le standard `UNICODE`,
 - ▶ la différence étant que `UNICODE` place les diacritiques après la lettre
 - ▶ et autorise un nombre illimité —mais contraint— de combinaisons

Certaines combinaisons sont en effet proscrites par le standard

- ▶ alors que T.51-ISO/IEC 6937 limite le nombre de diacritique à un.

- ▶ Le standard est limité aux écritures latines mais inclut une partie des diacritiques utilisés en français, allemand, espagnol ainsi que par les langues slaves ou baltes.
- ▶ Son adoption semble avoir été assez limitée
 - à la différence d'ISO/CEI 8859
- ▶ car incompatible avec les implémentations logicielles d'alors
 - la relation entre caractères et glyphes n'étant plus bijective
- ▶ UNICODE rencontrera un problème similaire qui se résoudra par l'introduction de caractères de compatibilité dans le standard

- ▶ une page de code est une identifiant unique attribué à un JdC
- ▶ qui permet d'utiliser différents JdC dans un même environnement
 - en changeant la page de code
- ▶ Le terme fut d'abord utilisé par IBM pour distinguer les différentes variantes de ses codages
 - puis repris par d'autres comme MS
- ▶ les pages de code ont ensuite intégré des JdC ayant d'autres provenance
 - standards, concurrence, . . .
- ▶ elles permettent un certain support multi—écritures
 - en permettant p. ex. d'utiliser un OS ou un logiciel avec le JdC voulu
- ▶ mais qui est limité par les JdC eux—mêmes
 - qui sont généralement limités à une une ou deux écritures, une même langue peuvent aussi avoir plusieurs variantes plus ou moins compatibles —fr_FR et fr_CA—
- ▶ encore très présents dans l'arrière—boutique. . .

Pour ne
pas conclure

Pour ne pas conclure

- ▶ difficulté à tirer des conclusions du fait des lacunes dans les sources
- ▶ quelques remarques pour terminer :
 - ▶ importance de prendre en compte les conditions littéralement matérielles des JdC
 - ▶ poids de la *legacy* 8-bits
 - ▶ de plus, l'élaboration des JdC s'est d'abord faite de façon
 - ▶ morcelée
 - ▶ mais en partie coordonnée
 - ▶ pour et dans des aires à la fois linguistiques, économiques et politiques distinctes

- ▶ enfin, lorsqu'on aborde la question des JdC, il ne faut pas sous-estimer :

- ▶ la complexité du problème

pour toute écriture, certaines apportant un surcroît de complexité et difficulté supplémentaire pour les jeux multi-scripturaux

- ▶ la multiplicité des JdC élaborés
 - ▶ l'importance des efforts consentis pour traiter la question du support multi-écriture
 - ▶ et ça, dès les années 1960
 - ▶ aussi partielles ou insatisfaisantes que les réponses apportées puissent paraître
- ▶ longue marche dont il reste encore à déterminer
- ▶ si ISO/CEI 10646–UNICODE en constitue l'aboutissement
- ▶ ou seulement une étape

Bibliographie

- ANDRÉ, Jaques (1996), « Caractères, codage et normalisation –de Chappe à Unicode », *Document numérique*, 3–4, 6, p. 13-49.
- BEMER, R. W. (1960), « Survey of Coded Character Representation », *Commun. ACM*, 12, 3, p. 639-642.
- FISCHER, Eric (2000), *The Evolution of Character Codes, 1874-1968*, URL : <https://archive.org/details/enf-ascii>.
- HARALAMBOUS, Yannis (2007), *Fonts & Encodings*, O'Reilly.
- KNUTH, Donald E. (1984), *The T_EXbook : a complete user's guide to computer typesetting with T_EX*, Addison-Wesley.
- LUNDE, Ken (2008), *CJKV Information Processing*, 2nd, O'Reilly Media, Inc.
- MACKENZIE, Charles E. (1980), *Coded-Character Sets : History and Development*, USA : Addison-Wesley Longman Publishing Co., Inc.
- MUSIANI, Francesca (2012), « Caring About the Plumbing : On the Importance of Architectures in Social Studies of (Peer-to-Peer) Technology », *Journal of Peer Production*, online, 1, 8 p.

**Merci pour
votre attention**