Jeux de caractères, codes et encodages

Souvent des confusions terminologiques entre des notions distinctes

Jeux de caractères (*charset*) (ou répertoire de) ensemble de caractères distincts. Spécifie le nom et une forme visuelle de référence des caractères

Code de caractère (*character code* ou *code point*) valeur numérique entière positive unique associée à chaque caractère d'un répertoire (peut s'exprimer en binaire, décimal, hexadécimal...)

Page de codes (*code page* – CP) : ensemble de caractères et de valeurs numériques associées

Encodage (*encoding*) Méthode (algorithme) pour représenter les caractères dans une forme électronique en assignant au code du caractère une séquence de bits (généralement cadrés en octets)

`http://www.cs.tut.fi/~jkorpela/chars.html`

En 1968 première standardisation de l'ASCII

American Standard Code for Information Interchange

- L'une des premières propositions de répertoire, code et encodage
- Caractères alphanumériques anglais (min/MAJ), quelques symboles mathématiques et de contrôle
- 128 codes
- Codé sur 1 octet (sur les 7 premiers bits de poids faibles)

Inadapté pour représenter d'autres langues telle que le français (pas de caractères accentués, cédille, ...)

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

De 1968 à 87, une multitude de pages de codes

Motivés par des enjeux économiques ou politiques des nations, des constructeurs de machines ou des développeurs de systèmes d'exploitation ou logiciels définissent une ou plusieurs pages de codes qui leur sont propres suivant leur internationalisation

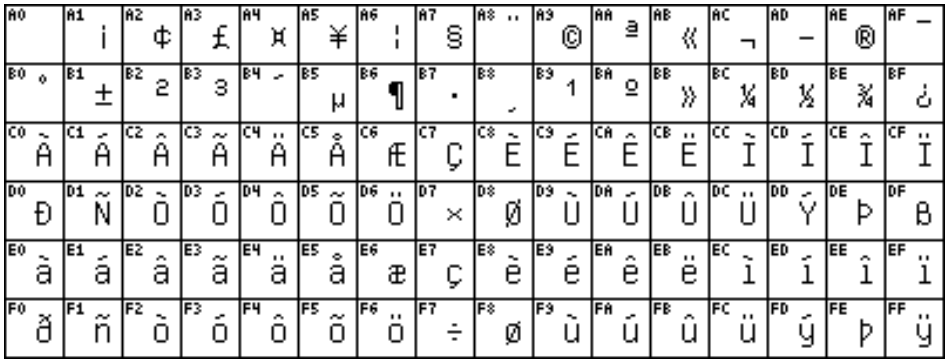
- En Europe et aux Etats-unis : les machines *mainframes* d'IBM utilisent le jeu EBCDIC ; les Macintosh d'Apple, le MacRoman, les systèmes Unix le Multinational Character Set ou MCS produit par Digital Equipment Corporation (DEC) ; repris dans les versions de MS-DOS produites par Microsoft
- L'Union soviétique, isolée par la guerre froide, définit la norme KOI8-R pour l'écriture cyrillique en russe, l'impose en Bulgarie et en ex-Yougoslavie pour l'écriture cyrillique du serbo-croate, crée une variante pour l'ukrainien
- Les pays asiatiques orientaux développent des pages de code multiples

Insuffisantes pour couvrir convenablement les langues proches voire certaines langues ; Des filiations et des recoupements mais aucune interopérabilité générale

Depuis 1987, des normes ISO 8859

- contient le répertoire ASCII, utilise les mêmes codes et un encodage similaire : chaque code est représenté sur 1 octet
- utilise le 8ième bit pour coder 128 caractères supplémentaires spécifiques à une langue donnée
- ISO 8859-1 (ou latin-1) pour europe de l'ouest, ISO 8859-5 pour l'alphabet cyrillique des langues slaves, ISO 8859-6 pour l'arabe...

une meilleur interopérabilité mais toujours des insuffisances et des spécificités locales (l'Inde, le Japon, la Chine...) et impossibilité de gérer des textes multilingues (i.e. différentes écritures reposant sur un encodage distinct sur une même page)



Vers un codage universel : ISO/IEC 10646 et Unicode

- Le standard ISO/IEC 10646 définit un répertoire universel de caractères (Universal Character Set UCS) avec un code pour chaque caractère.
  Rapidement le Consortium Unicode qui poursuivait le même projet fusionne son répertoire avec ce dernier
  Par facilité on désigne ce répertoire fusionné par le nom d'*Unicode*
- Les 256 premiers caractères codés sont ceux d'ISO 8859-1
- Fin 2009 environ 100 000 caractères (dont plus de la moitié pour les seuls sinogrammes), chacun possédant un unique code ; en perpetuel extension
- L'Unicode propose différents encodages

L'Unicode propose différents encodages

- **UTF-32** encodage d'un caractère sur 1 mot de 32 bits ; algorithme simple mais rarement utilisé car utilisation non efficace de la mémoire (surtout si on utilise que les caractères latins pour lesquels on sait qu'1 octet suffit)
- **UTF-16** encodage d'un caractère sur 1 ou 2 mots 16 bits ; Unicode fut originellement conçu pour ne representer que 65665 valeurs, désormais utilise certains codes réservés du premier mot pour coder sur 4 octets certains caractères ; encodage très simple pour les 65665 premiers caractères codés
- **UTF-8** les codes $<$ à 128 (répertoire ASCII) utilisent un octet
  Les autres codes utilisent 2 à 4 octets chacun d'eux avec des valeurs de 128 à 255 (i.e. débutant tous par 1bin)

Caractère	ASCII	ISO-8859-1	UTF-8
e	01100101	01100101	01100101
é		11101001	11000011 10101001
€			11100010 10000010 10101100