

# Représentation des données non numériques

Il va très vite aussi être nécessaire de représenter d'autres données que des nombres.

On a donc inventer plusieurs codes *alphanumériques*. C'est à dire des codes qui permettent de représenter des chiffres, des lettres, des signes de ponctuation, etc.

Chaque code lie un symbole à un nombre. Ainsi on utilise des "tables de codage" pour savoir quel nombre correspond à tel caractère.

## EBCDIC (Extended Binary Coded Decimal Interchange Code)

Ce code est une extension du "BCD" (Binary Coded Decimal) mais sur 8 bits (4 bits de "zone" et 4 bits numériques)

Caractère	Bits de zone	Bits numériques	Hexa
0	1111	0000	F0
1	1111	0001	F1
2	1111	0010	F2
3	1111	0011	F3
4	1111	0100	F4
5	1111	0101	F5
6	1111	0110	F6
7	1111	0111	F7
8	1111	1000	F8
9	1111	1001	F9
A	1100	0001	C1

Caractère	Bits de zone	Bits numériques	Hexa
B	1100	0010	C2
C	1100	0011	C3
D	1100	0100	C4
E	1100	0101	C5
F	1100	0110	C6
G	1100	0111	C7
H	1100	1000	C8
I	1100	1001	C9
J	1101	0001	D1
K	1101	0010	D2
L	1101	0011	D3
M	1101	0100	D4
N	1101	0101	D5
O	1101	0110	D6
P	1101	0111	D7
Q	1101	1000	D8
R	1101	1001	D9

# ASCII (American Standard Code for Information Interchange)

Le code ASCII beaucoup plus connu et utilisé permet de stocker 128 caractères sur 7 bits.

## ASCII table

Ensuite le code ASCII fut stocké sur 8 bits car IBM a ajouté 127 codes supplémentaires pour d'autres caractères plus spécifiques (mais pas supportés par tous les ordinateurs).

Les codes 0 à 31 sont des caractères de contrôle pour la transmission des données ou l'affichage sur un terminal (tab, retour à la ligne, etc).

# Unicode et UTF-8

Le code ASCII étant d'origine américaine, beaucoup de caractères n'étaient pas supportés (par exemple, les caractères accentués).

L'organisation internationale de normalisation (ISO) a donc tenté de créer un standard pour l'échange de textes dans différentes langues : l'Unicode.

L'une des formes les plus utilisées (surtout sur internet) de ce standard est le code UTF-8 qui est extensible et compatible avec l'ASCII.

L'UTF-8 est à taille variable, les caractères peuvent être représentés sur 1, 2, 3 ou 4 octets. Si le caractère est représenté sur 1 octet, alors c'est un caractère ASCII.

Ainsi l'UTF-8 permet de représenter 1 114 112 caractères différents (et a la possibilité de coder au total 1 114 112 caractères différents).

## Représentation des données dans les langages de programmation

Chaque langage a sa propre manière de représenter ses données. Par exemple, en C une chaîne de caractères est stockée comme étant une succession de caractères séparés par des caractères spéciaux.

## En savoir plus

- [Wikipedia - Unicode](#)
- [Wikipedia - UTF-8](#) (explique aussi comment connaître le nombre d'octet d'un caractère)

---

Revision #1

Created 27 April 2023 06:15:03 by SnowCode

Updated 27 April 2023 06:27:26 by SnowCode