

### 1) Le principe des encodages

Dans la mémoire d'un ordinateur, toutes les données sont stockées sous forme binaire (0 ou 1)

#### Définitions

Un **bit** (*binary digit*) est un chiffre binaire, qui ne peut prendre que deux valeurs (0 ou 1).

Un **octet** est une suite de 8 bits.

L'association entre chaque caractère et chaque contenu binaire est ARBITRAIRE.

Depuis les années 1960, la norme la plus utilisée est **ASCII** :

American Standard Code for Information Interchange

Chaque caractère standard est représenté sur 7 bits,

avec un code binaire qui correspond à un nombre compris entre 0 et 127.

### 2) Diversité des encodages

ATTENTION : un même caractère peut être encodé par des contenus binaires variés.

Un même contenu binaire peut correspondre à des caractères variés.

Cela s'explique par :

des raisons matérielles : Windows / Linux / Apple

des raisons linguistiques : différents besoins selon les zones linguistiques (alphabets différents)

des raisons historiques : besoins qui évoluent dans le temps

Exemple du caractère euro : €

<b>encodages sur 1 octet</b>	hexadécimal	binaire	décimal
latin1 = ISO / 8859-1	80	1000 0000	128
latin10 = ISO / 8859-16	A4	1010 0100	164
x-mac-roman	DB	1101 1011	219

### 3) Unicode

la norme ISO/CEI 10646 décrit une table de 137 929 caractères !

Chaque caractère reçoit un identifiant unique **indépendant de son encodage en mémoire** !

Exemple : le symbole € occupe le rang 8364. Il est noté (en hexadécimal) U+20AC

encodage du caractère €	hexadécimal	
UTF-16BE	20AC	sur 2 octets
UTF-16LE	AC20	sur 2 octets
UTF-8 (1 à 4 octets)	E282AC	sur 3 octets

La référence unique « 8364 points de code » permet la traduction entre différents encodages.

Application : Déterminer le nombre de caractères envoyés pour 1 SMS (160 bits) lorsque :

a) chaque caractère est encodé sur 7 bits :  $160 // 7 = 22$  caractères

b) chaque caractère est encodé sur 1 octet :  $160 // 8 = 20$  caractères

c) chaque caractère est encodé sur 2 octets :  $160 // 16 = 10$  caractères