

1 - Comment coder le monde en 0 et 1 ?

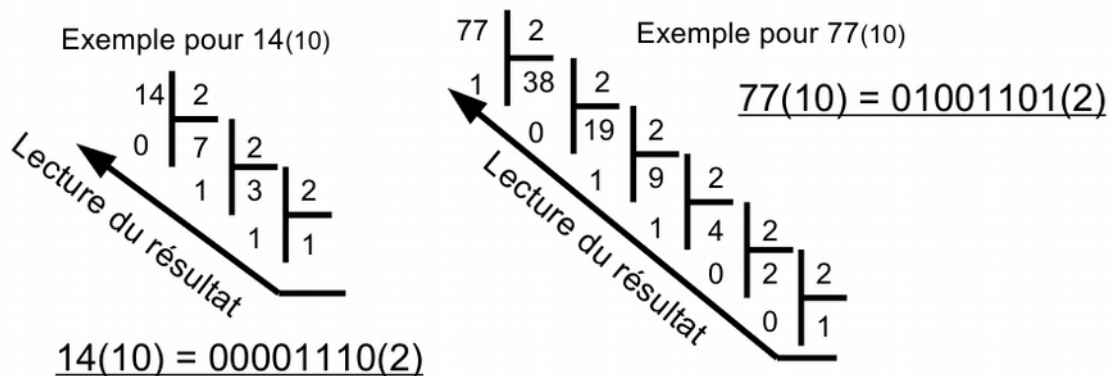
Pour coder des informations, que ce soient des nombres, du texte, des images, de la vidéo ou des sons, un ordinateur utilise uniquement des 0 et des 1.

1 - Représentation d'un nombre

Possibilité 1 : par conversion décimal vers binaire

1

Rappel :



$$5 (10) = 101 (2)$$

$$51 (10) = 110011 (2)$$

$$42 (10) = 101010 (2)$$

Avec un octet :

le nombre maximum en binaire est donc : **1111 1111(2)**

le nombre maximum en décimal est donc : **255(10)**

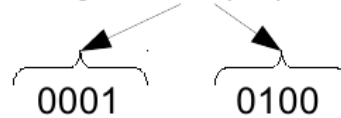
Possibilité 2 : par décimal codé en binaire (DCB)

2

Chiffre : Binaire :

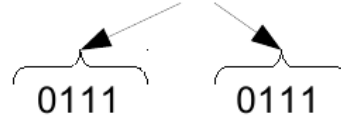
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Codage de 14(10)



$$\underline{14(10) = 00010100(\text{DCB})}$$

Codage de 77(10)



$$\underline{77(10) = 01110111(\text{DCB})}$$

$$5(10) = 0000101(\text{DCB})$$

$$51(10) = 01010001(\text{DCB})$$

$$42(10) = 01000010(\text{DCB})$$

Avec un octet :

le nombre maximum en DCB est donc : **1001 1001(DCB)**

le nombre maximum en décimal est donc : **99(10)**

2 - Représentation d'un texte

Pour écrire, nous avons besoin de lettres majuscules, minuscules, de chiffres et de signes.

La norme ASCII établit une correspondance entre une représentation binaire et le texte.

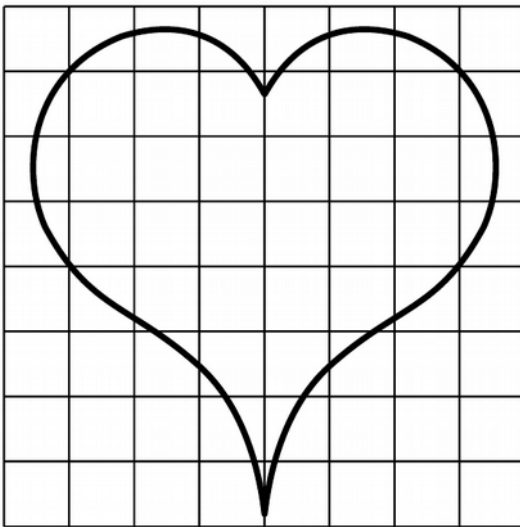
00101110	.		
00101111	/		
00110000	0		
00110001	1		
00110010	2		
00110011	3		
00110100	4		
00110101	5		
00110110	6		
00110111	7		
00111000	8		
00111001	9		
00111010	:		
00111011	;	/ →	0010 1111(2)
00111100	<		
00111101	=	B →	0100 0010(2)
00111110	>		
00111111	?	O →	0100 1111(2)
01000000	@		
01000001	A		
01000010	B	N →	0100 1110(2)
01000011	C		
01000100	D	J →	0100 1010(2)
01000101	E		
01000110	F		
01000111	G	O →	0100 1111(2)
01001000	H		
01001001	I	U →	0101 0101(2)
01001010	J		
01001011	K		
01001100	L	R →	0101 0010(2)
01001101	M		
01001110	N	.	→ 0010 1110(2)
01001111	O		
01010000	P		
01010001	Q		
01010010	R		
01010011	S		
01010100	T		
01010101	U		

Avec un octet, il existe **256** codes possibles.

3 - Représentation d'une image

La représentation d'une image repose sur le découpage de l'image en pixels. En noir et blanc, le pixel est à 1 s'il est noir, à 0 s'il est blanc.

Codage des 64 pixels :



0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0

Pour cette image, nous avons besoin de **8 octets**

Codage de l'image : **0111 1110(2)**
 1001 1001(2)
 1000 0001(2)
 1000 0001(2)
 0110 0110(2)
 0011 1100(2)
 0001 1000(2)
 0001 1000(2)