

Les données informatiques



Introduction : Les systèmes informatiques manipulent toutes sortes d'informations. Mais sous quelle forme circulent-elles dans les circuits électroniques ? Comment un ordinateur peut coder un texte, un nombre, une image ou coder une touche tapée au clavier ?

Bits, octets...**Comment sont codées les données informatiques ?**

Qu'il s'agisse d'un texte, d'une image, d'un son ou de n'importe quel programme informatique, ceux-ci sont stockés en mémoire de l'ordinateur ou sur des supports de stockages sous forme d'octets.

Un **octet** est un nombre informatique composé d'une série de huit 0 ou 1 (d'où le nom OCTet). Chacun de ces 0 et 1 constitue ce que l'on appelle **1 bit**.

Grâce à un système de numération appelé **système binaire**, ces octets sont capables de coder les valeurs entre **0 et 255**. Un octet ne peut donc coder plus de 256 valeurs, pour des nombres plus grand il est nécessaire de juxtaposer plusieurs octets les uns aux autres.

ex : 1 0 1 0 1 1 0 1 est un nombre binaire, il est composé de 8 bits, c'est donc un octet.

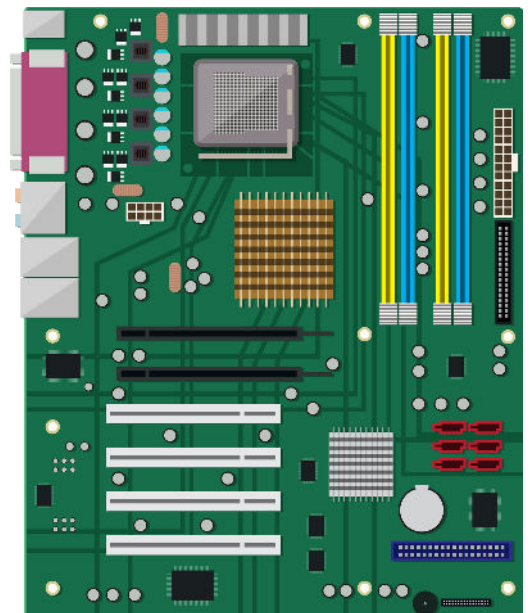
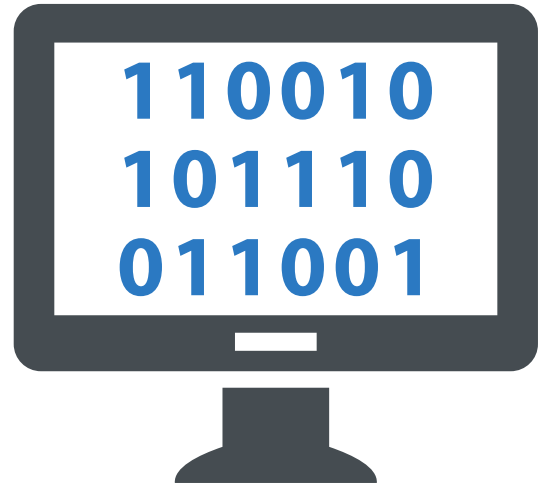
Pour information : un octet est l'espace nécessaire pour stocker une lettre ou caractère ou bien un chiffre.

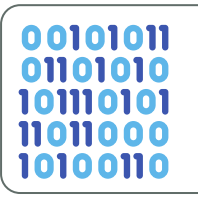
Des multiples...**Kilo-octet, Méga-octet, Giga-octet...**

Pour exprimer la capacité de stockage d'une carte mémoire, d'un disque dur ou même de la mémoire d'un ordinateur il a vite été nécessaire d'utiliser des multiples de l'octet. Au fil du temps les capacités de stockage ont explosées et de nouveau multiples sont apparus dans le langage informatique

- 1 Kilo-octets = 1 000 octets = 1 Ko
- 1 Méga-octet = 1 000 000 octets = 1 000 Ko = 1 Mo
- 1 Giga-octets = 1 000 000 000 octets = 1 000 Mo = 1 Go
- 1 Téra-octets = 1 000 000 000 000 octets = 1 000 Go = 1 To
- 1 Péta-octets = 1 000 000 000 000 000 octets = 1 000 To = 1 Po

Aujourd'hui les disques durs les plus volumineux disposent d'une capacité de mémoire qui s'expriment en Téra-octets.





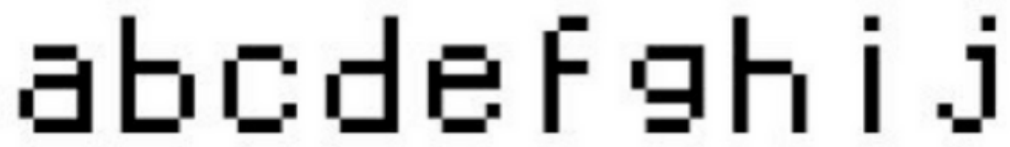
Activité 1 : le codage des images en binaires

Une image en noir et blanc

Rien de plus simple de coder une image en noir et blanc en langage binaire :

- Les 0 donnent des points blancs
- Les 1 donnent des points noirs

C'est ainsi que sont dessinées les lettres sur notre écran :



Exercice :

Quel est le symbole contenu dans cette succession de chiffres binaires... Sur une feuille à carreaux, trace un rectangle de 13 colonnes sur 10 lignes. Colorie les cases qui reçoivent un bit à 1. Commencer en haut à gauche et descendre de la même manière ligne par ligne.

```

00000000000000001000001000000010001000000
0111111100000110111011000111111111110010
1111111010010100000101000001101100000000
000000000000

```

Activité 2 : le codage du texte

Le code ASCII

Un ordinateur ne sachant traiter que des données sous formes de chiffres binaires. Comment peut-il alors mémoriser et afficher des textes ?

En **1961** l'informaticien américain **Robert Bemer** à mis au point un système de codage nommé **ASCII** (American Standard Code for Information Interchange « Code américain normalisé pour l'échange d'information »). L'idée est de faire correspondre à chaque lettre et chiffre un nombre précis.

Plus tard ce code évoluera pour intégrer les caractères avec accents mais aussi tous les symboles nécessaires pour toutes les langues à travers le monde (dont le chinois et l'arabe !)

Le savais-tu ?



Un **pixel** est le plus petit élément qui constitue une image, son nom vient de "Picture Element" en Anglais. Il est constitué d'une simple couleur. au début de l'informatique, les écrans et circuits électroniques généraient des pixels bien plus gros.



Robert Bemer dans les années 60

Les données informatiques

EXERCICE : En utilisant le tableau du code ASCII, décode le message secret ci-dessous...
Montre discrètement ton résultat au professeur lorsque tu as finis.

78 111 117 115 32 115 111 109 109 101 115 32 101 110 32 99 111 117 114 115 32 100 101
32 84 101 99 104 110 111 108 111 103 105 101 32 33

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	00 0000 NUL	01 0000 SOH	02 0000 STX	03 0000 ETX	04 0000 EOT	05 0000 ENQ	06 0000 ACK	07 0000 BEL	08 0000 BS	09 0000 HT	0A 0000 LF	0B 0000 VT	0C 0000 FF	0D 0000 CR	0E 0000 SO	0F 0000 SI	
1	16 0001 DLE	17 0001 DC1	18 0001 DC2	19 0001 DC3	20 0001 DC4	21 0001 NAK	22 0001 SYN	23 0001 ETB	24 0001 CAN	25 0001 EM	26 0001 SUB	27 0001 ESC	28 0001 FS	29 0001 GS	30 0001 RS	31 0001 US	
2	32 0010 SP	33 0010 !	34 0010 "	35 0010 #	36 0010 \$	37 0010 %	38 0010 &	39 0010 '	40 0010 (41 0010)	42 0010 *	43 0010 +	44 0010 ,	45 0010 -	46 0010 .	47 0010 /	A
3	48 0011 0	49 0011 1	50 0011 2	51 0011 3	52 0011 4	53 0011 5	54 0011 6	55 0011 7	56 0011 8	57 0011 9	58 0011 :	59 0011 ;	60 0011 <	61 0011 =	62 0011 >	63 0011 ?	B
4	64 0100 @	65 0100 A	66 0100 B	67 0100 C	68 0100 D	69 0100 E	70 0100 F	71 0100 G	72 0100 H	73 0100 I	74 0100 J	75 0100 K	76 0100 L	77 0100 M	78 0100 N	79 0100 O	C
5	80 0101 P	81 0101 Q	82 0101 R	83 0101 S	84 0101 T	85 0101 U	86 0101 V	87 0101 W	88 0101 X	89 0101 Y	90 0101 Z	91 0101 [92 0101 \	93 0101]	94 0101 ^	95 0101 _	D
6	96 0110 '	97 0110 a	98 0110 b	99 0110 c	100 0110 d	101 0110 e	102 0110 f	103 0110 g	104 0110 h	105 0110 i	106 0110 j	107 0110 k	108 0110 l	109 0110 m	110 0110 n	111 0110 o	E
7	112 0111 p	113 0111 q	114 0111 r	115 0111 s	116 0111 t	117 0111 u	118 0111 v	119 0111 w	120 0111 x	121 0111 y	122 0111 z	123 0111 {	124 0111 	125 0111 }	126 0111 ~	127 0111 DEL	F

