

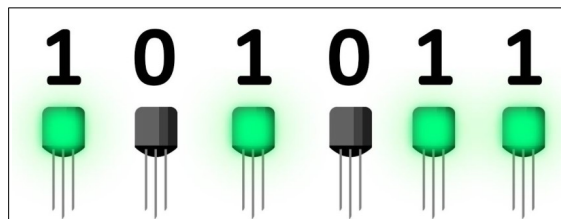
2^e – Sciences Numériques et Technologie

Chapitre 1 : La photographie numérique

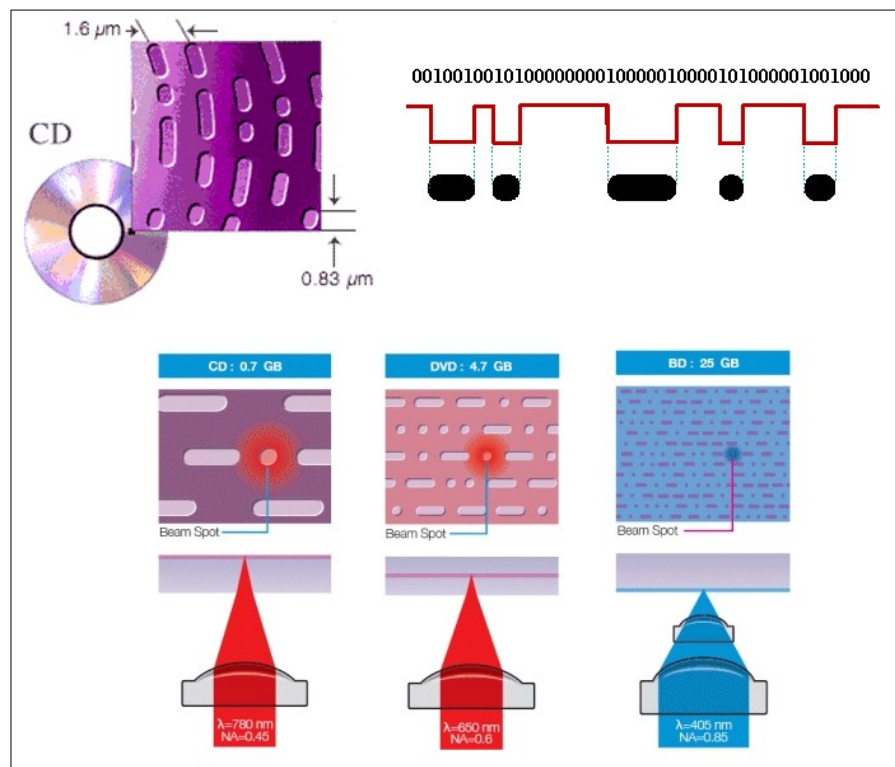
I. Préambule : bits, octets et poids d'un fichier informatique

La plus petite unité d'information manipulable par une machine numérique est le **bit** « *binary digit* ». Un bit peut avoir deux états auxquels on attribue les symboles 0 ou 1. Cela permet de représenter et stocker physiquement cette information par :

un signal électrique (le courant passe ou ne passe pas) stocké dans un « transistor »,



des aspérités (trous et bosses) gravées sur une surface (l'écriture en braille ou les CD/DVD),



Une information codée sur 1 bit donne 2 états différents : **0** ou **1**

Une information codée sur 2 bits donne 4 états différents : **00**, **01**, **10** ou **11**

Avec 3 bits, on peut obtenir 8 états différents : **000**, **001**, **010**, **011**, **100**, **101**, **110** ou **111**

etc...

Les octets : Un ensemble de 8 bits est appelé un **octet** (en anglais **byte**).

Un octet peut donc prendre 256 valeurs différentes : **00000000** , **00000001** , **00000010** , **00000011** , **00000100** , , **11111110** ou **11111111**

Les multiples :

En informatique, et contrairement à sa signification dans la vie courante, 1 Kilo vaut $2^{10}=1024$ et non pas 1000. On le note avec une majuscule.

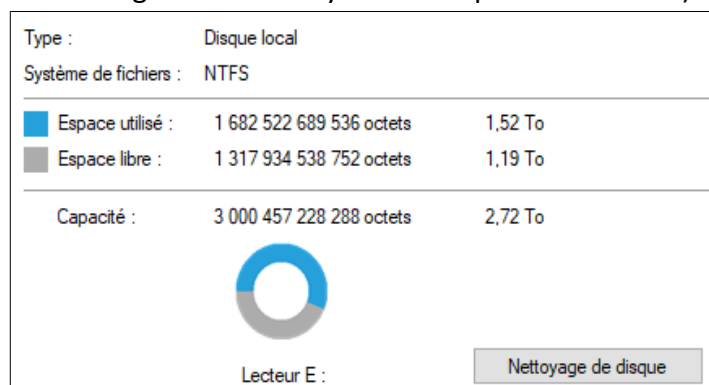
Nous avons vu qu'une unité de mesure de la quantité de stockage informatique est l'octet (appelé aussi byte).

Donc 1 Kilo octet, noté 1 Ko (ou 1 Kb) vaut $2^{10}=1024$ octets.

1 Mega octet, noté 1 Mo (ou 1 Mb) vaut donc $2^{10}=1024$ Ko, soit $2^{10} \times 2^{10}=2^{20}=1048576$ octets (donc différent de 1 million d'octets !)

De même, 1 Go (Giga octet) vaut 1024 Mo , 1 To (Tera octet) vaut 1024 Go, etc... (les préfixes suivants sont Peta, Exa, Zetta, Yotta).

- a) Expliquer ces différences observées entre les valeurs en Octets et les valeurs en To sur l'image ci-dessous représentant l'utilisation de la mémoire d'un disque dur (ce disque dur était annoncé en magasin comme ayant une capacité de 3 To !!):



- b) Une photographie numérique a un poids de 6,3Mo, et mesure 3840 pixels (largeur) sur 2160 pixels hauteur). On réduit ses dimensions (largeur et hauteur) d'un coefficient de réduction égal à $\frac{1}{5}$.

a) Calculer ses nouvelles dimensions en pixels.

b) Quel est le coefficient de réduction de sa surface ?

c) En déduire son nouveau poids en Ko.

- c) Combien d'image numériques de 6,3Mo peut contenir une clé USB de 1Go ?

Le système de numération binaire :

En « écriture décimale », le nombre 453 vaut 453 car $453=4 \times 10^2+5 \times 10^1+3 \times 10^0$ (on rappelle que $10^0=1$!!). On peut noter 453_{10} pour indiquer qu'il est en base 10.

La « base binaire » ou « base 2 » fonctionne de la même façon mais au lieu d'avoir des puissances de 10 avec 10 chiffres, on a des puissances de 2 avec 2 chiffres : 0 et 1

Le nombre 10011010 en binaire vaut :

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	1	0	1	0
128	64	32	16	8	4	2	1

$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 128 + 16 + 8 + 2 = 154$ en écriture décimale.

Ce nombre se note 10011010_2 pour indiquer qu'il est en base 2.

Donc $10011010_2 = 154_{10}$

Pour transformer un nombre décimal en écriture binaire :

Exemple : 174

Dans 174 on peut mettre 128 une fois, donc

Puis : $174 - 128 = 46$

Dans 46 on ne peut pas mettre 64, donc :

Dans 46 on peut mettre 32 donc :

Puis $46 - 32 = 14$.

Dans 14 on ne peut pas mettre 16 donc :

Dans 14 on peut mettre 8 donc :

Puis $14 - 8 = 6$

Dans 6 on peut mettre 4 donc :

Puis $6 - 4 = 2$

Et dans 2 on peut mettre 2 donc :

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1							
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0						
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1					
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0				
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1			
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1		
128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1	1	0
128	64	32	16	8	4	2	1

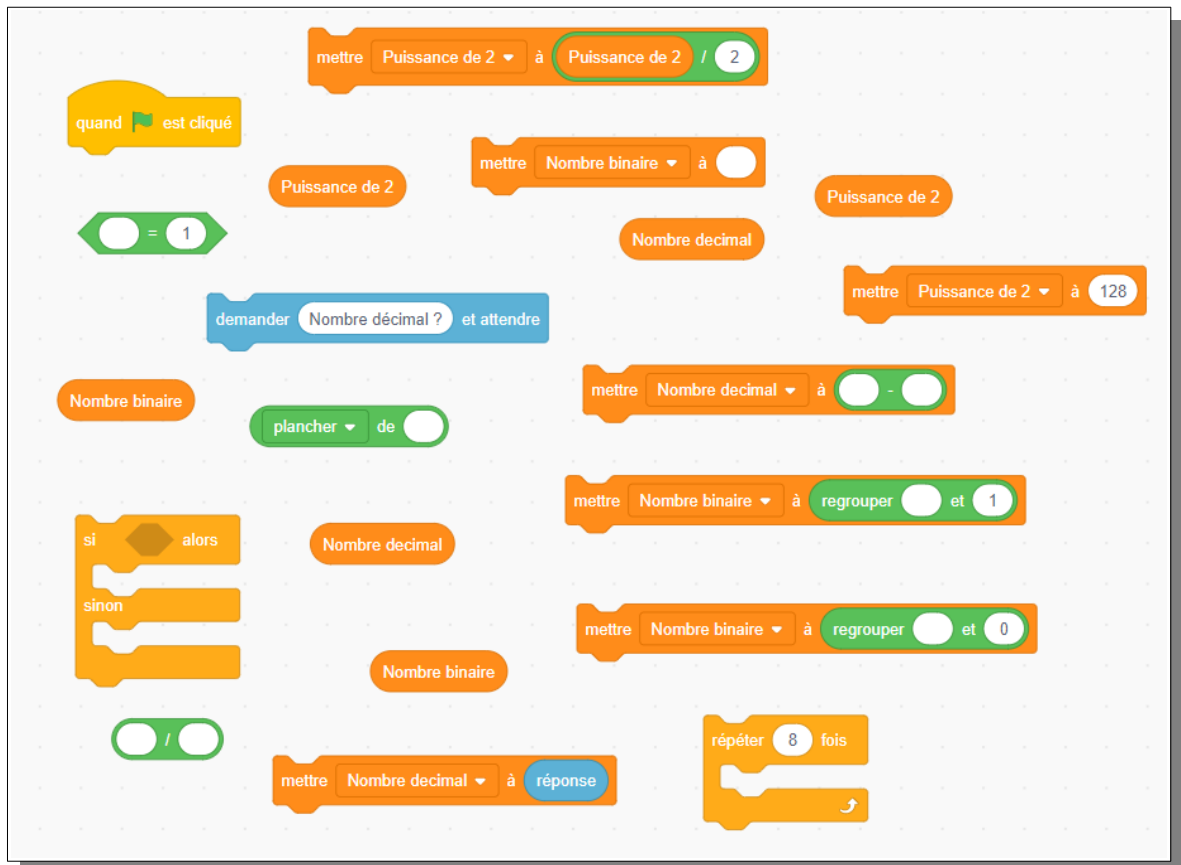
Donc l'écriture binaire de 174_{10} est 10101110_2

a) Combien vaut le nombre binaire 10101110_2 en base 10 ?

b) Écrire en binaire le nombre 197_{10} puis le nombre 2386_{10}

c) Combien vaut le nombre 00000000_2 ? Et le nombre 11111111_2 ?

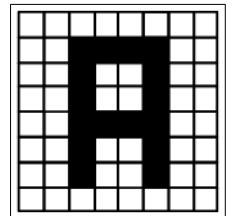
d) En utilisant vos connaissances d'algorithmique de collège et les blocs proposés ci-dessous (ce n'est qu'une proposition, il y a plusieurs solutions possibles) construire sur Scratch un algorithme qui permet de transformer un nombre décimal (choisi entre 0 et 255) en écriture binaire (on pourra revoir l'exemple donné ci-dessus avec le nombre 174...). **Une fois résolu le problème, ouvrir le document appelé « 2nde_SNT_01_La_Photographie_Numerique_1_Exercice » (celui avec une icône jaune, c'est un nouveau type de document).**



II. Et les images numériques dans tout cela ?

En Noir et Blanc pour commencer...

Le codage informatique de la lettre A dessinée ci-contre, si l'on décide qu'un carré blanc est codé par 0 et un carré noir par 1, serait le suivant :



On a regroupé les 0 et 1 par paquets de 8, qui constituent des octets et qui correspondent aux lignes. (Remarque : cette représentation sous forme de tableau de nombres s'appelle, en mathématiques, une Matrice ; elles sont étudiées en Spécialité maths de terminale).

On voit donc comment un ordinateur peut stocker (et afficher) une image aussi simple grâce à une suite de 0 et de 1. Ici, l'ordinateur a besoin de 8 octets, car il y a 8 lignes et chacune est un octet.

Pour une image couleur, c'est sensiblement différent car il existe plus de deux couleurs...

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