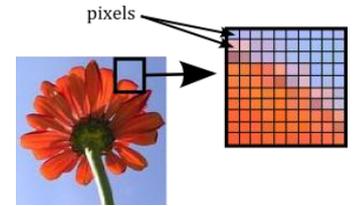


# 1. L'image numérique

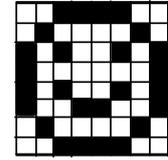
## 1. Définitions

Une image numérique est un ensemble de points appelés **PIXELS** (contraction de **PICTure** **E**lements). Chaque pixel possède une couleur.



### 1.1. Dimensions et taille d'une image

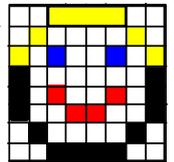
1. Quelles sont les dimensions de l'image ci-contre en pixels ?
2. Combien de pixels contient-elle au total ?



### 1.2. Codage d'une image (appelé aussi profondeur de couleur)

Pour faire simple, le codage d'une image correspond au nombre maximum de couleurs différentes qu'elle peut contenir.

3. Combien de couleurs différentes contient l'image précédente ?
4. En s'appuyant sur le tableau ci-dessous, compléter la phrase suivante : « Cette image est codée sur ..... bit »
5. Sur la nouvelle image ci-contre, colorier les cheveux en jaune, les yeux en bleu et la bouche en rouge.
6. Combien de couleurs différentes contient cette image ?
7. Avec un codage sur 2 bits, combien de couleurs pourrait contenir l'image ?
8. Compléter la phrase : « Cette nouvelle image est codée sur ..... bits ».



Codage sur...	Un bit	2 bits	3 bits	4 bits	...	$n$ bits
Nombre maximal de couleurs	2	4	8	16		$2^n$

9. Quel est le nombre maximal de couleurs que peut contenir une image codée sur 8 bits ?

### 1.3. EXERCICE

Voici les caractéristiques d'une photo numérique:

10. Quelles sont les dimensions en pixel de l'image ?
11. Calculer le nombre de pixels que contient l'image.

Un mégapixel correspond à 1 millions de pixels c'est à dire  $10^6$  pixels

12. Combien de Megapixels contient l'image ?
13. La photo est codée sur combien de bits ?
14. Combien de couleurs différents peut-elle contenir ?

Attributs de l'image

Dimensions de l'image : 5184 x 3456

Profondeur de bit : 24-bit couleur

Pages : 1 page(s)

Pixels par pouce : 72 PPP

Dimen. image (en pouces) : 72.00 x 48.00

Profil couleur intégré : sRGB IEC61966-2.1

Taille : 5,8 Mo

Créé : 22/08/2019 11:09:36

Modifié : 08/09/2019 19:20:44

Accédé : 08/09/2019 19:20:43



Le poids d'une photo est le nombre d'octets qu'elle va occuper sur une carte mémoire ou une clé USB (certains logiciels l'appellent « taille »)

15. Repérer, dans les propriétés de l'image, le poids de la photo.
16. Convertir le poids de la photo en octets.
17. La mémoire d'un téléphone permet de stocker 8 Go (giga-octets) de données. Combien de photos peut-on stocker sur ce téléphone ?

#### Vu en physique :

1 Mega =  $10^6$  (1 million)  
1 Giga =  $10^9$  (1 milliard)

## 2. Tableau associé à une image

Une image numérique est au départ un tableau de nombres. Ces nombres indiquent la couleur que doit prendre chaque pixel. Des logiciels transforment ensuite ces nombres en couleur visibles sur l'écran donc en image.

### 2.1. Exemple de transformation d'un tableau de nombres en image (image codée sur 1 bit)

**CODAGE sur UN bit :**  
**1** : BLANC  
**0** : NOIR

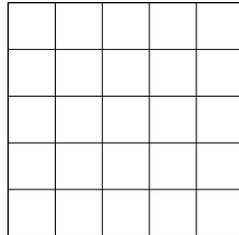
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$


Tableau sauve sur carte SD

Image affichée

**CARACTÉRISTIQUES DE L'IMAGE**

Dimension: ... x ... pixels

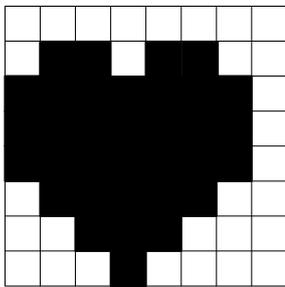
Nombre total de pixels : ..... pixels

Nombre de couleurs : .....

Type de codage : ..... bit

18. Dessiner, ci-dessus, l'image associée au tableau stocké sur la carte SD.

### 2.2. Exemple de transformation d'une image en tableau: (image codée sur 1 bit)



**CARACTÉRISTIQUES DE L'IMAGE**

Dimension: ... x ... pixels

Nombre total de pixels : ..... pixels

Nombre de couleurs : .....

Type de codage : ..... bit

19. Ecrire, ci-dessus, le tableau associé à l'image codée sur 1 bit.

20. Combien de bits (c'est-à-dire de 0 ou 1) faut-il écrire sur la carte SD ?

21. En déduire le poids en octets du fichier image correspondant au tableau.

**Un octet = 8 bits**

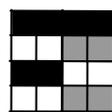
## 3. Image en niveaux de gris codée sur 8 bits

### 3.1. Tableau associé à une image 8 bits niveaux de gris

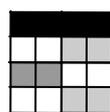
22. Compléter le cadre de droite

**Codage 8 bits niveaux de gris :**  
 de ..... Nuances de gris  
 à ..... : (NOIR) à ..... : (BLANC)  
 valeur du gris « moyen » : 127  
 Plus un nombre est petit, plus le pixel est sombre

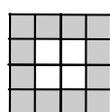
23. Associer à chaque image ci-dessous le tableau de nombres qui lui correspond



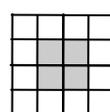
A



B



C



D

1

$$\begin{pmatrix} 255 & 255 & 255 & 255 \\ 255 & 127 & 127 & 255 \\ 255 & 127 & 127 & 255 \\ 255 & 255 & 255 & 255 \end{pmatrix}$$

2

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 255 & 255 & 63 & 63 \\ 0 & 0 & 255 & 255 \\ 255 & 255 & 63 & 63 \end{pmatrix}$$

3

$$\begin{pmatrix} 127 & 127 & 127 & 127 \\ 127 & 255 & 255 & 127 \\ 127 & 255 & 255 & 127 \\ 127 & 127 & 127 & 127 \end{pmatrix}$$

4

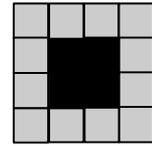
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 255 & 255 & 127 & 127 \\ 63 & 63 & 255 & 255 \\ 255 & 255 & 127 & 127 \end{pmatrix}$$

**3.2. Eclaircissement ou assombrissement d'une image 8 bits niveaux de gris**

Pour **assombrir** une image, il faut **diminuer** d'un même nombre la valeur de ses pixels  
 Pour **éclaircir** une image, il faut **augmenter** d'un même nombre la valeur de ses pixels

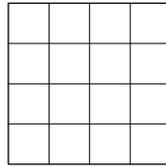
**Attention :** Dans tous les cas, la valeur d'un pixel ne peut être inférieure à 0 (noir) ou supérieure à 255 (blanc)

24. Voici une image et son tableau de nombres associé :



$$\begin{pmatrix} 127 & 127 & 127 & 127 \\ 127 & 0 & 0 & 127 \\ 127 & 0 & 0 & 127 \\ 127 & 127 & 127 & 127 \end{pmatrix}$$

Ecrire ci-dessous le nouveau tableau si on éclaircit l'image d'une valeur égale à 128. Dessiner ensuite l'allure de la nouvelle image.

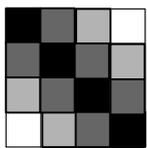
$$\left( \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right)$$


**Pour les plus rapides**

25. Jouons le rôle de PYTHON en exécutant un algorithme :

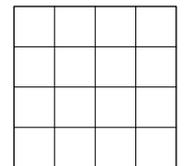
Pour chaque pixel de l'image :  
 La valeur du pixel vaut elle-même moins 127  
 Si le nombre obtenu  $\leq 0$  :  
 La valeur du pixel vaut 0

a. Exécuter cet algorithme pour l'image suivante (son tableau est donné) :



$$\begin{pmatrix} 0 & 50 & 127 & 255 \\ 50 & 0 & 50 & 127 \\ 127 & 50 & 0 & 50 \\ 255 & 127 & 50 & 0 \end{pmatrix}$$

Exécution de  
 l'algorithme  $\rightarrow$

$$\left( \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right)$$


- b. Quel est le rôle de cet algorithme ?
- c. L'image finale ressemble-t-elle à l'image initiale en plus sombre ?

**4. Image en couleur (codage RVB 8 bits par couche)**

Dans une image couleur codée en RVB, chaque pixel ne possède pas une valeur mais trois.

Dans l'ordre, un pixel possède une valeur de ....., une valeur de ..... et une valeur de .....  
 Chaque valeur de chaque couleur est comprise entre 0 et 255.

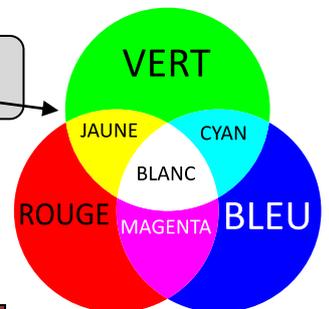
**4.1. Comment obtenir 16 millions de couleurs à partir de 3 couleurs primaires ?**

Pour chaque pixel, le logiciel (chargé d'afficher l'image) fait une **synthèse additive** des 3 valeurs de rouge, vert et bleu.

Par exemple :

Si le pixel possède une grande valeur de Rouge, une grande valeur de vert et pas de bleu, il est :

.....



26. Compléter le tableau suivant en complétant les pourcentages :

Couleur du pixel	% de Rouge	% de Vert	% de Bleu
Rouge	100	0	0
Bleu			
Vert			
Blanc			
Noir			
Jaune			
Cyan			

**4.2. Nombre de couleurs possibles dans une image RVB**

Comme chaque couleur est codée sur 8 bits, on peut obtenir 256 nuances de chaque.

En résumé, pour chaque pixel :

Nombre de nuances de rouge possible : .....  
 Nombre de nuances de vert possible : .....  
 Nombre de nuances de bleu possible : .....  
 } donc, nombre de couleurs possibles pour un pixel :

**4.3. Valeurs associées à un pixel**

100% de rouge (par exemple) signifie que la valeur « rouge » du pixel vaut 255.

27. Ecrire les 3 valeurs de RVB d'un pixel pour les couleurs suivantes :

Couleur du pixel	Rouge	Bleu	Vert	Blanc	Jaune	Noir	Gris « moyen » (c'est un blanc « peu lumineux »)
Valeurs (R,V,B)	(255,0,0)						

**4.4. Construction d'une image RVB à partir de son tableau**

Dans le tableau associé à une image couleur RVB, il y a 3 valeurs associées chaque pixel.

28. Dessiner l'image associée au tableau suivant :

un pixel

(0,255,0)	(0,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(0,255,0)	(0,255,0)
(0,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(0,255,0)
(255,255,0)	(255,255,0)	(0,0,255)	(255,255,0)	(255,255,0)	(0,0,255)	(255,255,0)	(255,255,0)
(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)
(255,255,0)	(0,0,0)	(255,255,0)	(255,255,0)	(255,255,0)	(255,255,0)	(0,0,0)	(255,255,0)
(255,255,0)	(255,255,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(255,255,0)	(255,255,0)
(0,255,0)	(255,255,0)	(255,255,0)	(255,0,0)	(255,0,0)	(255,255,0)	(255,255,0)	(0,255,0)
(0,255,0)	(0,255,0)	(255,255,0)	(255,0,0)	(255,0,0)	(255,255,0)	(0,255,0)	(0,255,0)

→

**Pour les plus rapides**

29. Ecrire le tableau associé à l'image suivante :

vert	jaune	jaune	vert
bleu			bleu
bleu	rouge	rouge	bleu
noir	cyan	cyan	noir

→
