

Activité 3 : Codage informatique des couleurs

Objectif de l'activité :

Durée : 01H30

- **Comprendre** l'encodage numérique d'une image noir et blanc, couleur et nuances de gris.

A savoir :

Toute l'informatique est basée sur la manipulation de bits. « Bits » est la contraction des mots « Binary Digit » ; **un bit peut prendre la valeur « 0 » ou la valeur « 1 »** ; on parle d'état logique.

Soit « a » un bit. On peut donc avoir $a = 0$ ou $a = 1$ et avec cela, on fait beaucoup de choses...

PARTIE A

Codage des images en noir et blanc

Q1 – Très simplement, pour une image en noir et blanc affichée sur un écran (de smartphone ou d'ordinateur), la couleur d'un pixel est soit _____ soit _____.

Q2 – Pour coder la couleur d'un pixel, il faut donc _____ bit(s).

Dans la suite, le noir est associé à l'état logique « 0 » et le blanc est associé à l'état logique « 1 ». Ceci est arbitraire, on aurait pu convenir du contraire.

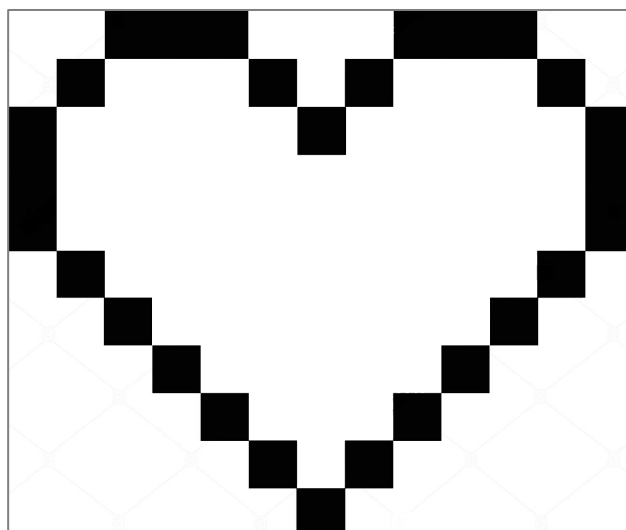
0 => NOIR

1 => BLANC

Q3 – Combien de pixels l'image ci-contre contient-elle en largeur ? _____ px

Q4 – Combien de pixels l'image ci-contre contient-elle en hauteur ? _____ px

Q5 – Calculer le nombre total de pixels de l'image : $N =$ _____



Q6 – Le tableau ci-dessous correspond à de la mémoire informatique. **La remplir** avec le code numérique (0 ou 1) de la couleur de chaque pixel de la photo ci-dessus.

→ La photo sera balayée pixel par pixel, de gauche à droite et de haut en bas.

Q7 – **Colorier** la place occupée dans la mémoire par le **troisième octet** uniquement.

Q8 – **Calculer** en octets le poids total que devrait avoir cette image : _____

PARTIE B

Codage des images en couleur

A savoir : Les trois couleurs fondamentales sont souvent notées « R » pour Red, « G » pour Green et « B » pour Blue. « RGB » donc, ou « RVB » en français.

La couleur d'un pixel résulte du mélange d'un peu de R, d'un peu de G et d'un peu de B.

Pour un pixel, chacune de ses trois couleurs fondamentales est codée sur **8 bits**.

Un paquet de **8 bits** s'appelle **un octet** ; on utilise très souvent le ko, le Mo ou encore le Go comme unités pour exprimer le poids d'un fichier numérique ou la capacité d'une clé USB, d'un disque dur.

Q9 – **Calculer** les nombres de combinaisons possibles pour les nombres de bits suivants :

1 bit : _____ 2 bits : _____ 3 bits : _____ 4 bits : _____ 5 bits : _____ 6 bits : _____

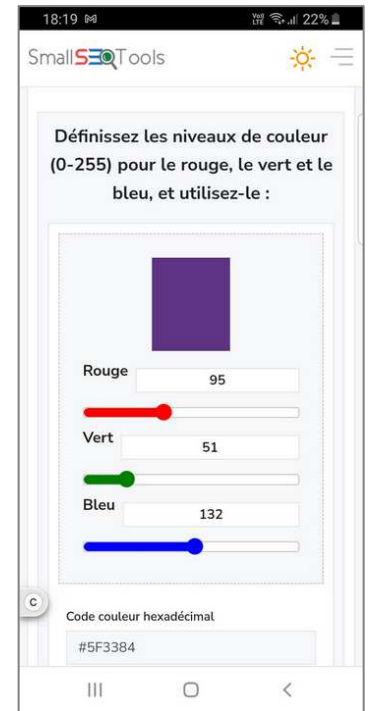
7 bits : _____ **8 bits : _____** 9 bits : _____ 10 bits : _____ 11 bits : _____ 12 bits : _____

Q10 – Pour définir la couleur d'un pixel, on dispose donc de :

- _____ nuances de rouge (couleur fondamentale « R » codée sur 8 bits).
- _____ nuances de vert (couleur fondamentale « V » codée sur 8 bits).
- _____ nuances de bleu (couleur fondamentale « B » codée sur 8 bits).

➤ A l'aide de votre smartphone ou d'un PC, se rendre à l'adresse <https://smallseotools.com/fr/rgb-to-hex/>

➤ **Manipuler** la palette de couleurs RVB et **constater** les limites à 255.



Q11 – Si les nuances R, V et B sont toutes mises à ...

0, alors la couleur obtenue est du _____

255, alors la couleur obtenue est du _____

Q12 – Donner le réglage RVB permettant d'avoir :

du vert pur : R = _____ V = _____ B = _____

du jaune pur : R = _____ V = _____ B = _____

Rappel : La couleur d'un pixel est obtenue par mélange des nuances de rouge, de vert et de bleu.

Q13 – Partant du rappel ci-dessus et de la réponse à la Q10, **calculer** le nombre total de couleurs que peut prendre un pixel :

PARTIE C

Codage des images en nuances de gris

➤ **Manipuler** à nouveau la **palette de couleurs** et **régler** les composantes RVB sur des valeurs égales (valeur de R = valeur de V = valeur de B).

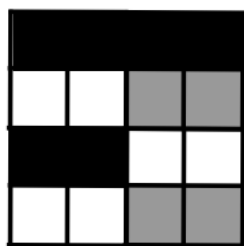
Q14 – Quel « type de couleur » obtient-on quand les nuances R, V et B sont égales ?

Q15 – Le blanc et le noir vérifient-ils la constatation de la Q14 ?

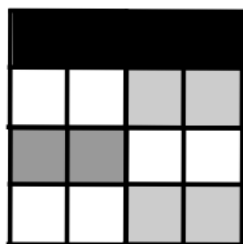
OUI

NON

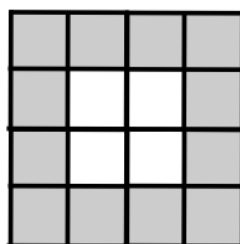
On donne ci-dessous quatre images A, B, C et D en nuances de gris ; on donne aussi quatre matrices correspondant au codage en nuances de gris de chacune d'elles.



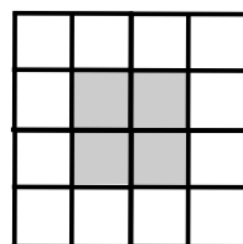
A



B



C



D

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

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

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

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

Q16 – Associer à l'aide de traits les images aux matrices d'encodage.

Q17 – Rappeler en octets le poids de l'image en noir et blanc de la partie A (voir la réponse Q8) :

Q18 – Calculer en octets le poids total que devrait avoir l'image de la partie A (143 px) si son encodage se fait en nuances de gris (1 x 8 bits par pixel) :

Q19 – Calculer en octets le poids total que devrait avoir l'image de la partie A (143 px) si son encodage se fait en RGB (3 x 8 bits par pixel) :

Q20 – Préciser les différences qu'on observerait visuellement selon que l'image est encodée en N/B, nuances de gris ou RGV et les rapporter aux différences de poids en octets.
