

## POUR VÉRIFIER SES ACQUIS



### 1 Vrai ou faux ?

Identifier les propositions exactes.

- a. Les trois couleurs de base pour la photographie numérique sont : rouge, jaune, bleu.
- b. Les couleurs sont codées habituellement avec des nombres compris entre 0 et 1 000.
- c. La qualité d'une image numérique dépend en général du nombre de pixels dont elle est constituée.
- d. Parmi les deux images suivantes, celle de droite a la plus petite résolution.



- e. Une image compressée prend plus de place qu'une image non compressée.
- f. L'appareil photo numérique applique de nombreuses modifications lors de la capture avant l'enregistrement.

### 2 QCM

Pour chaque proposition, identifier la bonne réponse.

- a. Le code couleur pour le vert est :  
☐ (255,0,255)    ☐ (0,255,0)    ☐ (0,0,255)
- b. La définition d'une image numérique est exprimée en :  
☐ cm<sup>2</sup>    ☐ octets    ☐ pixels
- c. Pour enregistrer une image dans un format qui prend le moins de place possible, il faut choisir l'extension :  
☐ Tiff    ☐ Raw    ☐ Jpg
- d. Les filtres des photosites sont répartis en carré de quatre :  
☐ deux verts, un rouge, un bleu  
☐ un vert, deux rouges, un bleu  
☐ un vert, un rouge, deux bleus

### 3 Qui suis-je ?

Recopier et compléter les phrases.

- a. Les petits carrés monochromes formant une image numérique sont les ...
- b. Les données comme la date, la géolocalisation ou les réglages de l'appareil sont enregistrées dans un fichier au format ...

→ Vérifier vos réponses p. 188

## POUR S'ENTRAÎNER

### 4 Qualité des images imprimées

**CAPACITÉ :** Distinguer les photosites du capteur et les pixels de l'image.

À l'impression, on estime qu'une image est :  
 – de bonne qualité si sa résolution est supérieure à 12 000 pixels par cm<sup>2</sup> ;  
 – de qualité moyenne si la résolution est entre 4 000 et 12 000 pixels par cm<sup>2</sup> ;  
 – de qualité mauvaise en dessous de 4 000 pixels par cm<sup>2</sup>.

- Recopier et compléter le tableau en indiquant la qualité de la photo (bonne, moyenne ou mauvaise) selon sa définition et ses dimensions.

| Dimension  | 11 × 15<br>(en cm) | 15 × 20<br>(en cm) | 20 × 27<br>(en cm) |
|--|--------------------|--------------------|--------------------|
| Définition<br>(mégapixels ou millions de pixels) |                    |                    |                    |
| 10 mégapixels                                    |                    |                    |                    |
| 5 mégapixels                                     |                    |                    |                    |
| 3 mégapixels                                     |                    |                    |                    |
| 800 000 pixels                                   |                    |                    |                    |



## 5 Choisir le bon capteur CAPACITÉ : Distinguer les photosites du capteur et les pixels de l'image.

1. Calculer la définition de chaque capteur (arrondir au million de pixels). Quelle est la différence entre le capteur B et le capteur C ?

2. Sachant que plus un photosite est petit, moins il est sensible à la lumière, quel est le capteur permettant le meilleur rendu parmi les trois ?

3. Léa souhaite acheter un appareil qui lui permette de réaliser des photos qu'elle imprimera en grand format pour en faire des affiches. Tom souhaite pouvoir faire de jolies photos par temps couvert, lorsque la luminosité est faible. Quels sont les appareils les plus adaptés aux besoins respectifs de Léa et de Tom ?



| Capteur              | A             | B             | C             |
|----------------------|---------------|---------------|---------------|
| Taille (mm × mm)     | 5,76 × 4,29   | 6,16 × 4,6    | 5 × 4,51      |
| Nombre de photosites | 5 908 × 4 400 | 5 500 × 4 100 | 5 000 × 4 510 |

## 6 Données EXIF CAPACITÉ : Retrouver les métadonnées d'une photographie.

• Associer les images à leur fichier de métadonnées.

|               | Fichier 1              | Fichier 2              | Fichier 3              |
|---------------|------------------------|------------------------|------------------------|
| Date et heure | 2010:03:11<br>13:45:32 | 2018:01:01<br>00:04:15 | 2016:12:24<br>19:15:05 |
| Compression   | Jpeg                   | Jpeg                   | Png                    |
| Flash         | Non                    | Oui                    | Oui                    |



## 7 Étapes de la prise de vue CAPACITÉ : Identifier les étapes de la construction de l'image finale.

• Classer dans un ordre cohérent les principales étapes indiquées ci-dessous permettant la construction d'une image numérique finale.

- a. Enregistrement des données EXIF.
- b. Mise au point, stabilisation.
- c. Correction de la saturation.
- d. Capture des valeurs R, V, B.
- e. Enregistrement au format Raw.
- f. Enregistrement au format Jpeg.

## 8 Profondeur de couleur CAPACITÉ : Distinguer les photosites du capteur et les pixels de l'image.

La profondeur de couleur est la mémoire nécessaire pour enregistrer la valeur de la couleur d'un pixel. Elle est exprimée en bits par pixel (bpp) et est comprise entre 1 et 32 bits. Les profondeurs usuelles de couleurs sont : *color* (couleur) en 8 bpp, *highcolor* (haute couleur) en 16 bpp et *truecolor* (vraie couleur) en 24 bpp.

1. Quelle est la mémoire requise pour enregistrer une image dont la définition est de 2,4 Mpx en résolution *color* (8 bpp) ? Par combien faut-il multiplier ce résultat pour obtenir la mémoire nécessaire en *highcolor* ? en *truecolor* ?

2. Combien d'images de 2,4 Mpx avec une profondeur de couleur de 8 bits peut-on stocker dans une carte mémoire de 16 milliards de bits (16 gigabits) ?



## POUR S'ENTRAÎNER À PROGRAMMER

### 9 Tests de luminosité CAPACITÉ : Traiter par programme une image pour la transformer en agissant sur ses pixels.

1. Qu'affiche la fonction `testLuminosite` pour les valeurs suivantes ?

- a.  $R = 50, V = 20$  et  $B = 10$       b.  $R = 80, V = 30$  et  $B = 120$

#### • Je comprends un programme

```
1 def testLuminosite(R,V,B):
2     if R+V+B<150:
3         print("c'est un pixel foncé")
4     else:
5         print("ce n'est pas un pixel foncé")
```



Besoin d'aide pour comprendre  
une boucle non bornée ?  
Revois la Méthode p. 179

2. On souhaite repérer les pixels clairs, c'est-à-dire ceux dont la somme des valeurs est supérieure à 500. Remplacer les lignes 4 et 5 par une instruction conditionnelle faisant afficher si le pixel est clair. On indiquera si un pixel n'est ni clair ni foncé.

### 10 Calcul du nombre de bits nécessaires

CAPACITÉ : Distinguer les photosites du capteur et les pixels de l'image.

Un codage d'une couleur sur  $n$  bits correspond à  $2^n$  couleurs. L'objectif du programme suivant est de donner le nombre de bits nécessaires pour coder une certaine quantité de couleur.



#### Coup de pouce Python

- Ligne 3 : l'opérateur `**` est celui de la puissance. L'instruction `2**nbBit` calcule  $2^{nbBit}$ .



Besoin d'aide pour comprendre une boucle ?  
Revois la Méthode p. 179

#### • Je comprends un programme

```
1 nombreCouleursSouhaitees=int(input("Nombre de couleurs souhaitées="))
2 nbBit=0
3 while nombreCouleursSouhaitees>2**nbBit:
4     nbBit=nbBit+1
5 print("Il faut un nombre de bits égal à", nbBit)
```

1. Quel est le type de la variable `nombreCouleursSouhaitees` ?
2. À quelle condition la boucle `while` de la ligne 3 s'arrête-t-elle ?
3. Quelle est l'utilité de l'instruction de la ligne 4 ?
4. L'utilisateur saisit la valeur 2.
  - a. La boucle `while` se termine-t-elle ?
  - b. Qu'affiche le programme ?
  - c. Reprendre les questions a. et b. si l'utilisateur saisit la valeur 40.



### 11 Modification de la luminosité CAPACITÉ : Traiter par programme une image pour la transformer en agissant sur ses pixels.

Dans un logiciel de traitement d'image, il est possible de modifier la luminosité d'une image. Pour cela les valeurs de rouge, vert et bleu de chaque pixel sont augmentées (plus clair) ou diminuées (plus sombre). On rappelle que chaque couleur est codée par un entier compris entre 0 et 255.



1. Le programme suivant a pour but de changer la luminosité d'un pixel en ajoutant un nombre entier compris entre - 255 et 255 à chacune des valeurs rouge, vert et bleu. Compléter les lignes 5, 6 et 7.

#### • Je complète un programme

```
1 R=int(input("Rouge="))
2 V=int(input("Vert="))
3 B=int(input("Bleu="))
4 lum=int(input("Valeur à ajouter (entre - 255 et 255)="))
5 R=.....
6 V=.....
7 B=.....
8 print("Nouvelles valeurs",R,V,B)
```

2. Les valeurs des variables  $R$ ,  $V$ ,  $B$  ne peuvent pas être négatives ou dépasser 255. Les valeurs trop grandes doivent donc être ramenées à 255 et les valeurs négatives à 0.

Ajouter au programme précédent, avant la ligne 8, des instructions conditionnelles permettant cette correction.



#### Coup de pouce Python

- Les lignes 1, 2 et 3 créent les variables  $R$  (rouge),  $V$  (vert) et  $B$  (bleue) du pixel.
- La ligne 4 permet la saisie par l'utilisateur du nombre entier à ajouter.
- Les lignes 5, 6 et 7 calculent les nouvelles valeurs de  $R$ ,  $V$ ,  $B$  après modification de la luminosité.
- La ligne 8 affiche les nouvelles valeurs de  $R$ ,  $V$ ,  $B$ .



Besoin d'aide pour écrire un programme ? Revois la Méthode p. 175



Besoin d'aide pour écrire des instructions conditionnelles ? Revois la Méthode p. 177

## 12 Définition d'une image

**CAPACITÉ :** Distinguer les photosites du capteur et les pixels de l'image.

1. Compléter la fonction suivante afin de calculer la définition d'une image selon sa résolution en points par pixel (ppp) et ses dimensions en pouces.

#### • Je complète un programme

```
1 def calculDefinition(resolution, largeur, longueur) :
2     definitionLargeur=.....
3     definitionLongueur=.....
4     return definitionLargeur,definitionLongueur
```



Besoin d'aide pour écrire une fonction ? Revois la Méthode p. 181

2. Écrire un programme qui demande à l'utilisateur la saisie de la résolution et des dimensions de l'image et qui appelle la fonction précédente.

3. Donner la définition d'une image de 5 pouces sur 7 en 200 ppp.

## 13 Filtre de repérage des rouges

**CAPACITÉ :** Traiter par programme une image pour la transformer en agissant sur ses pixels.

Traduire en programme Python l'algorithme suivant qui permet de mettre en valeur les pixels à dominante de rouge (le rouge est la valeur la plus importante dans le triplet  $R$ ,  $V$ ,  $B$ ).

#### • Je traduis un algorithme en programme

```
1 demander les valeurs de R, V et B à l'utilisateur
2 si R > V et R > B alors
3     R ← 255
4     V ← 0
5     B ← 0
6 afficher R, V, B
```

