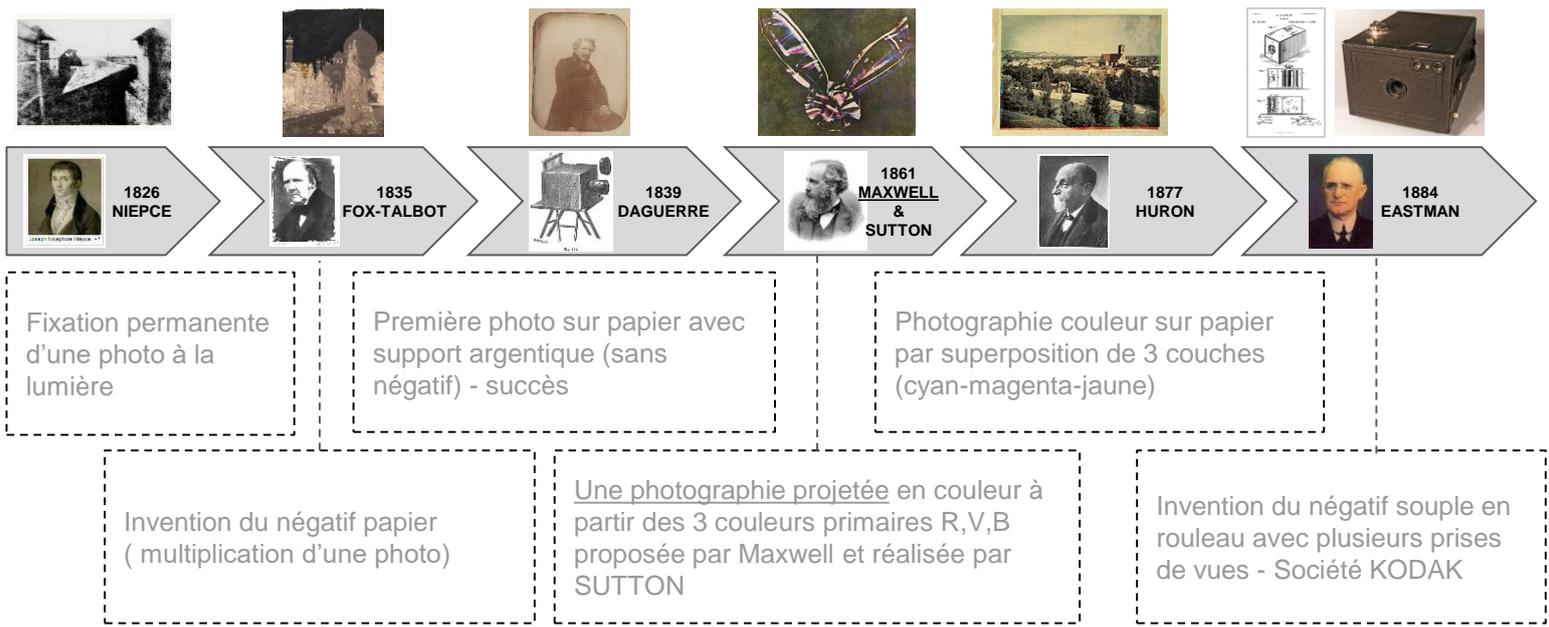


Thème: Photographie numérique

1- Les pionniers de la photographie "argentique" - Quelques repères datés sur les principales inventions



Le négatif d'une photo était obligatoire pour l'obtention d'une photo sur papier dans le procédé de la photographie "argentique" .

Le développement photographique a été très à la mode vers la fin des années "seventies" avec l'achat d'agrandisseurs (voir photo), de matériels pour révéler et fixer la photo ...

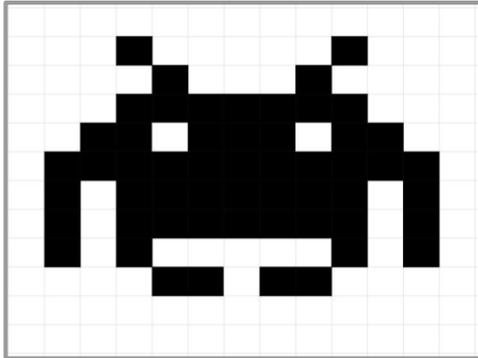


Avec l'arrivée du numérique, le négatif ne rentre plus dans le processus d'obtention d'une photo. Il apporte un effet artistique...

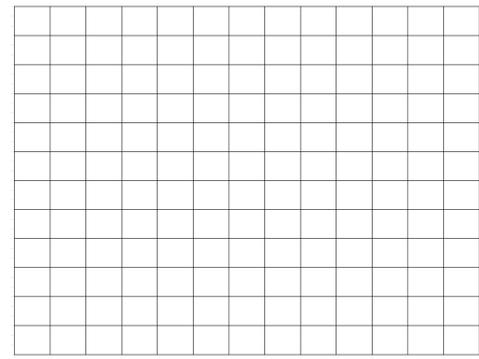
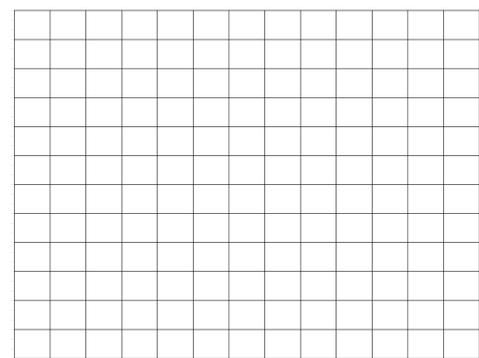
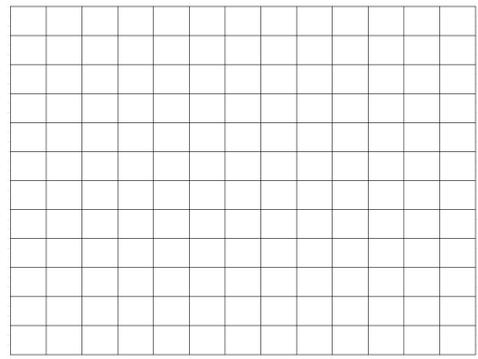


1 pixel de l'image correspond à un carré:

Dessiner le négatif de la photo du jeu "space invaders":



Numériser avec des 0 et des 1 la photo du jeu "space invaders":



N É G A T I F N & B N É G A T I F N & B

Un négatif N&B d'une photo numérique est:

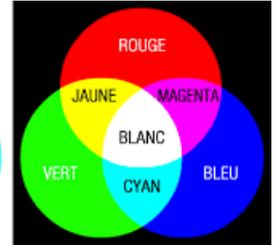
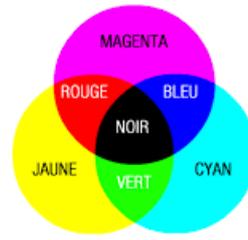
2.3 - image numérique en couleur

Chaque pixel de la photo, sur un écran de smartphone, utilise la synthèse additive (association de 3 couleurs primaires ROUGE-VERT-BLEU) pour définir une couleur à partir de 256 niveaux de rouge, de vert et de bleu.

L'impression sur papier d'une photographie part de la synthèse soustractive pour afficher une couleur par pixel imprimé

Synthèse soustractive

Synthèse additive



Si pour chaque couleur primaire nous avons une profondeur de 256, quel est le nombre total de couleurs?

Nbre couleurs: $256 \times 256 \times 256 = 16\,777\,216$ couleurs

Une image numérique comporte un nombre défini de couleurs contrairement à l'image que notre œil peut observer (signal analogique = quantité infini). Du nombre de couleurs affichées, dépendra la qualité de l'image.

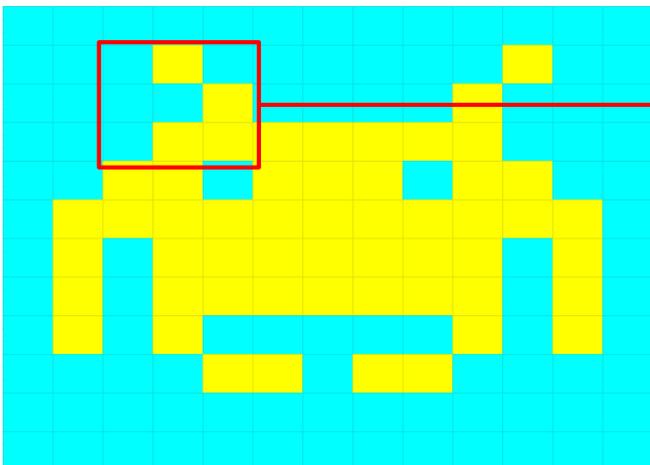
Compléter la première rangée afin de retrouver les chiffres de la deuxième rangée.

Puis pour la dernière colonne du tableau indiquer le chiffre décimal obtenu sur chaque ligne de 8 bits (1 octet):

					2^2			
			16	8	4	2	1	Nbre décimal
0	0	0	0	0	1	1	1	
0	0	1	0	1	0	0	1	
1	1	1	1	1	1	1	1	

Donner la valeur décimal d'un pixel ROUGE, BLEU, VERT, MAGENTA, JAUNE, BLANC en synthèse additive: Pixel rouge = (valeur rouge, valeur verte, valeur bleue) avec valeur = nombre décimal 0 ou 255

Compléter les valeurs en rouge, vert, bleu de chaque pixel repéré sur l'image "space-invader" en cyan et jaune



R: V: B:	R: V: B:	R: V: B:
R: V: B:	R: V: B:	R: V: B:
R: V: B:	R: V: B:	R: V: B:

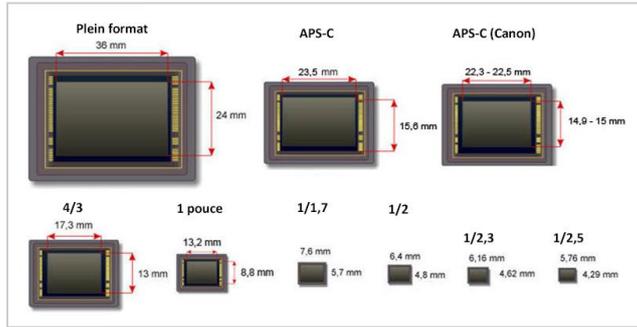
Compléter le tableau puis calculer le nombre de "données" nécessaire:

Bravo, vous avez compris sur quel principe repose une photo en couleur



3- Technologie du capteur optique d'un appareil photographique numérique (smartphone y compris)

3.1- A quoi ressemble un capteur optique numérique:



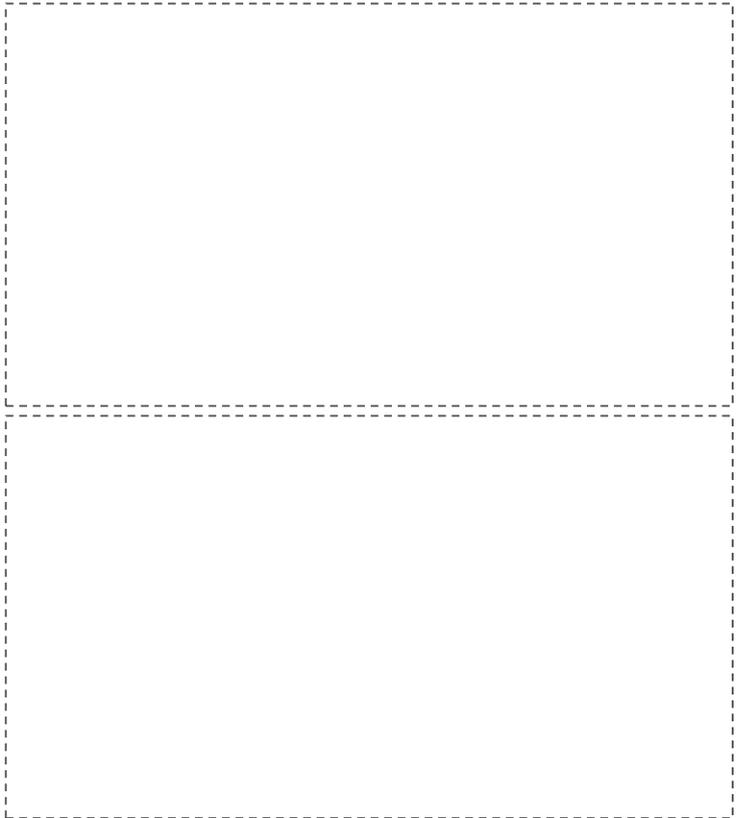
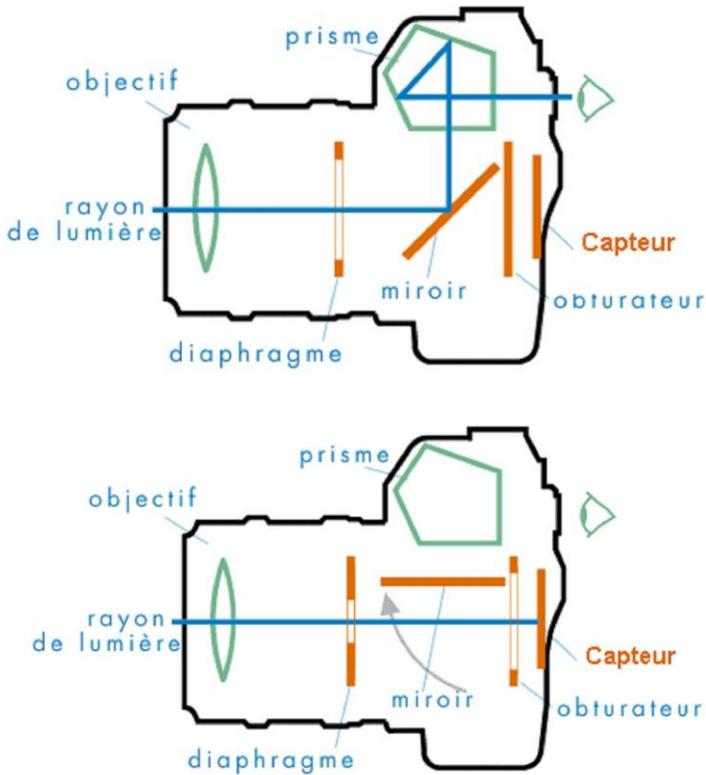
Entourer les capteurs dédiés aux smartphones

La qualité d'une photo dépend-elle des dimensions du capteur?

3.2- La différence entre appareil reflex argentique et numérique

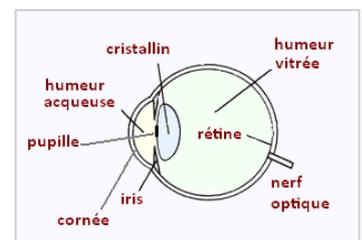


Fonctionnement d'un appareil photographique reflex:



Quelle est la particularité de l'image arrivant sur le capteur?

Peut-on comparer ce fonctionnement à celui de notre oeil?

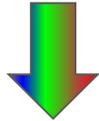
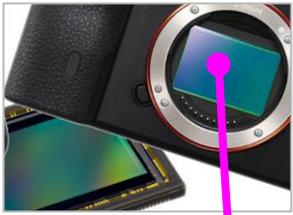


3.4- Les différentes parties constitutives d'un capteur optique et leurs fonctions



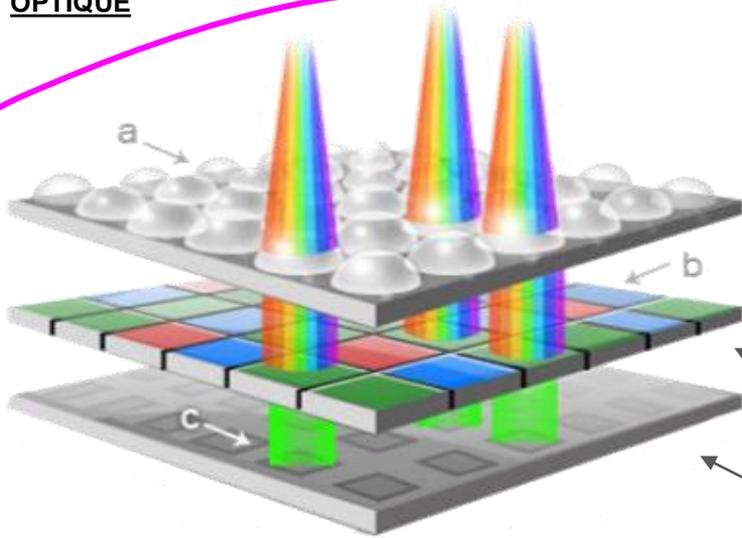
La lumière qui rentre dans l'objectif est projeté sur la surface du capteur photo. La surface doit recueillir en chacun de ses points, des informations d'intensité et de couleur (R,V,B) suivant le "sujet" pris en photo.

Les photosites disposés sur la surface du capteur photo sont sensibles à la lumière mais sont incapables de distinguer la couleur rouge, verte ou bleue.



LENTILLES DE L'OBJECTIF

CAPTEUR OPTIQUE



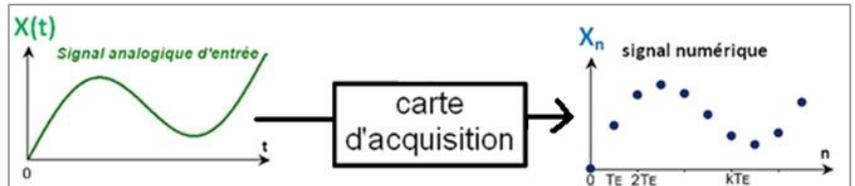
V	R	V	R	V	R
B	v	B	v	B	v
V	R	V	R	V	R
B	v	B	v	B	v
V	R	V	R	V	R
B	v	B	v	B	v



Signal de sortie du capteur et pré-traitement numérique:

Un photosite est l'équivalent d'un panneau solaire.

Convertisseur analogique/numérique CAN:



Signal numérique sur 14 bits après le CAN veut dire:

4- Fichier RAW et traitement

Un fichier RAW comporte, des informations concernant l'image, des données sur la luminosité, la balance des blancs,

C'est pourquoi lorsqu'on ouvre un fichier brut issu du capteur optique, l'image est souvent fade.



Logiciel "Darktable" permettant de visualiser des fichiers RAW comme avec l'extension ".DNG" (smartphone SAMSUNG)

Il y a un écart entre

5- Traitement post-opératoire ou "développement de la photo" prise avec le capteur optique

5.1 Apparition des pixels vrais de l'image avec l'opération de "dématriçage"

Un pixel vrai est

Méthode de dématriçage la plus simple :

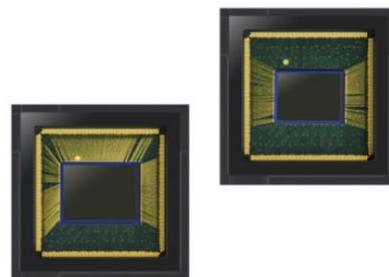
Dématriçage le plus simple

V	R	V	R	V	R
B	V	B	V	B	V
V	R	V	R	V	R
B	V	B	V	B	V
V	R	V	R	V	R
B	V	B	V	B	V

Combien de pixels vrais sont donc issus du capteur ci-contre:

Samsung annonce un capteur photo pour smartphones de 64 mégapixels:

Ces deux modèles utilisent la même taille de photosite de 0,8 micron du modèle Slim T32 qui plafonnait jusqu'à présent à 20 mégapixels. De plus, les deux nouveaux capteurs reprennent la technologie TETRACELL qui combine quatre photosites pour n'en faire qu'un seul. Cela fera par exemple chuter la définition du modèle 64 mégapixels à seulement 16 mégapixels, mais permettra d'obtenir des images beaucoup mieux détaillées en basses lumières.



Une autre méthode de base est l'algorithme d'interpolation bilinéaire (autant de pixels créés que de photosites)

L'algorithme de dématriçage est gardé top secret par les fabricants. En effet c'est lui qui donne "une couleur marketing" aux photos de chaque marque. Avec le numérique, le rôle de la "chambre noire" en argentique permettant de "révéler" l'image après la prise de la photo est une fonction désormais incluse dans l'appareil.

La "chambre noire" est toujours présente dans le logiciel de traitement numérique "Darktable" (à partir d'un fichier RAW). Il offre la possibilité de "développer sa photo" (en fait améliorer le rendu de celle-ci) et s'adresse à des amateurs avertis et professionnels

5.2- Le fichier JPEG, un fichier incontournable pour la sauvegarde des photos

En observant la taille d'un fichier RAW (.DNG) et celle d'un fichier JPEG, il est aisé de comprendre lequel va être préféré pour la sauvegarde de nos photos!



Fleur_courgette.dng

Type d'élément : Fichier DNG
Prise de vue : 27/06/2019 08:01
Dimensions : 496 x 280
Taille : 18,0 Mo



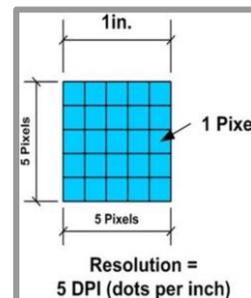
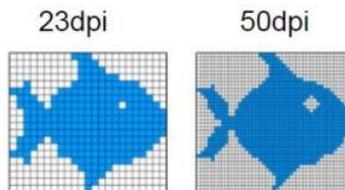
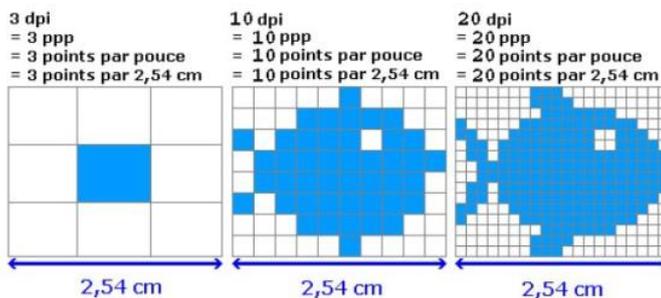
Fleur_courgette.jpg

Type d'élément : Fichier JPG
Prise de vue : 27/06/2019 08:01
Dimensions : 4032 x 2268
Taille : 3,40 Mo

Il y a deux notions importantes pour la photo :

La définition de l'image :

La résolution de l'image :



La résolution est sélectionnée sur l'appareil photo par l'utilisateur. Ces résolutions d'images dépendent de l'algorithme sélectionné en fonction du format de l'image souhaitée et non directement du capteur photographique

La taille d'un fichier non compressé se calcul avec la formule suivante:



T: taille en octets
D: définition photo en pixels
P: profondeur en octets

Exemple pour la photo "Fleur courgette" de dimensions 496*280

Définition =

Profondeur = 1 octet par couleur R,V,B = 3 octets

Taille =



Fleur_courgette.tif

Type d'élément : Fichier TIF
Dimensions : 496 x 280
Taille : 407 Ko

On retrouve presque le poids initial avec le format TIFF

Pour autant la photo n'en est pas meilleure. La dégradation JPEG est permanente.

Par contre, ce format est utilisé pour l'impression à condition de passer directement de RAW à TIFF

6- Programmation Edupython : traitement d'images

6.1 Afficher la photo à l'aide de python :

Placer l'image "pomme_base.jpg" dans le même dossier que celui où vous allez sauvegarder sous pommes_"vos_initiales".py le programme ci-contre.



Ouvre la bibliothèque PIL pour l'affichage d'une image

Montre l'image stockée dans la variable appelée img

Le fichier de l'image à ouvrir est stocké dans la variable img

```
from PIL.Image import *
img=open("pomme_base.jpg")
img.show()
```

ASTUCE: utiliser les touches CTRL + S pour sauvegarder puis cliquer sur la touche F5 pour lancer le programme

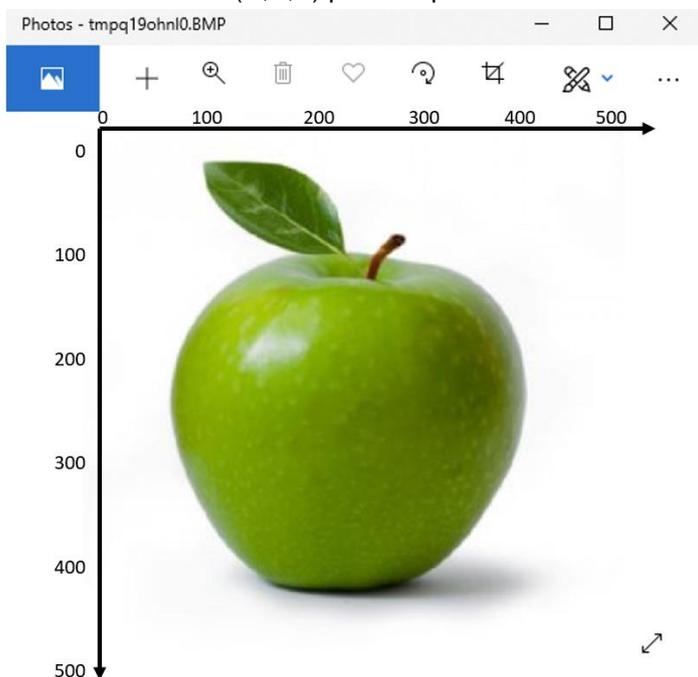
Le programme ci-dessus a bien ouvert l'image pomme_base.jpg comme souhaité.

Mais la programmation Python ne se limite pas à l'affichage d'une image!

Nous allons pouvoir analyser les couleurs de la photo et ainsi modifier les couleurs de celle-ci pixel par pixel...

6.2 Afficher la couleur R,V,B d'un pixel :

En recopiant le programme ci-dessous, Python va extraire les valeurs en couleur (R,V,B) pour les points recherchés :



```
from PIL.Image import *
img = open("pomme_base.jpg")
print("1:",img.getpixel((0,0)))
print("2:",img.getpixel((250,250)))
print("3:",img.getpixel((499,499)))
img.show()
```

On va afficher la valeur (R,V,B) du pixel 499,499

On obtient dans la console:

```
>>>
1: (255, 255, 255)
2: (106, 140, 2)
3: (255, 255, 255)
```

6.3 Afficher et sauvegarder le négatif de la pomme:

Comme on l'a vu précédemment, pour créer un négatif, il suffit de récupérer les valeurs (R,V,B) de chaque pixels et de calculer (255-R, 255-V, 255-B) et de les remplacer par cette valeur.

ASTUCE: pour se souvenir ce que fait chaque ligne du programme, on peut mettre des commentaires après un #

Le programme ci-dessus affiche bien le négatif sur la ligne y=250

```
from PIL.Image import *
img = open("pomme_base.jpg")
print("1:",img.getpixel((0,0)))
print("2:",img.getpixel((250,250)))
print("3:",img.getpixel((499,499)))
for x in range(500): #on va faire le négatif sur une ligne a y=250
    (R,V,B)=img.getpixel((x,250))
    img.putpixel((x,250),(255-R,255-V,255-B))
img.show()
```

On va faire 500 fois les deux lignes en-dessous avec x qui va varier entre 0 et 499.

ATTENTION : il ne faut pas oublier le ":" à la fin de la ligne for

Et les indentations (une tabulation) pour chaque instruction de la boucle.

Modifier les lignes 6 à 8 pour faire le négatif sur une colonne, x=250



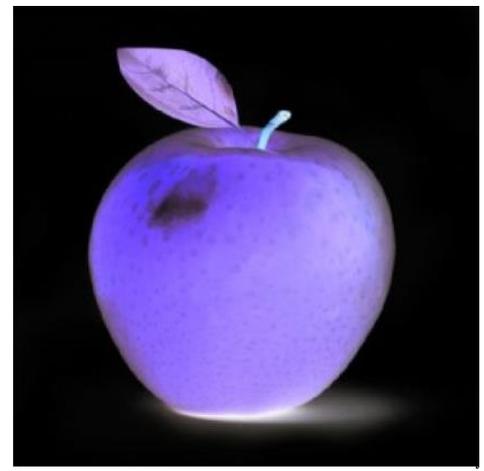
6.3 Afficher et sauvegarder le négatif de la pomme (suite):

Maintenant que vous savez faire sur une ligne et une colonne, comment faire sur toute l'image?

Complétez le code ci-dessous et testez, vous devez obtenir ceci:

```
from PIL.Image import *
img = open("pomme_base.jpg")
for x in range (500): #négatif colonne par colonne
    for y in range (500): #négatif du pixel de la colonne x et ligne y
        (R,V,B)=img.getpixel((_____,_____))
        img.putpixel((_____,_____), (255-R,255-V,255-B))
img.save("pomme_negatif.jpg", "JPEG") #on sauvegarde l'image
img.show()
```

Bravo, vous venez de créer manuellement le négatif d'une photo numérique.



Pour aller plus loin: Comment remplacer le fond noir par du vert (20,237,57)?

Il suffit de tester si le pixels à une teinte rouge <50 , une teinte verte <50 et une teinte bleue <50.

Pour cela on va utiliser la boucle condition if: (le et s'écrit and en python)

Rajouter cette partie juste avant la ligne img.show(), sauvegarder et tester le code suivant:

```
for x in range (500):
    for y in range (500): #pixel de la colonne x et ligne y
        (R,V,B)=img.getpixel((x,y))
        if R<50 and V<50 and B<50: #si c'est vrai on change la couleur du pixel
            img.putpixel((x,y), (20,237,57))
img.save("pomme_negatif_fond_vert.jpg", "JPEG") #on sauvegarde l'image
```

ATTENTION : il ne faut pas oublier le ":" à la fin de la ligne if
Et l'indentation (une tabulation) pour chaque instruction de la condition.



Pour celles et ceux qui adorent cela:

- 1) A partir du programme précédent, à la place du vert, faire un dégradé bleu jusqu'à la ligne 400 puis un fond brun jusqu'au bas de l'image et sauver l'image sous pomme_ciel.jpg.
- 2) A partir de l'image d'origine, faire un programme qui fait la pomme en orange (milieu ci-dessous).(on testera si le pixel a une teinte bleue <200 et l'on le remplacera par (255,V-50,70)
- 3) Pour rajouter le texte, il faut importer deux librairies en plus :
 - from PIL import ImageFont
 - from PIL import ImageDraw

Dans le programme après le traitement :

Il faut définir la font "font1 = ImageFont.truetype("ARLRDBD.TTF", 100)" : le fichier ARLRDBD.TTF doit être dans le même répertoire que le programme et le 100 représente la taille des caractères.

Puis créer l'objet draw qui contiendra le text : "draw = ImageDraw.Draw(img)"

Puis rajouter le texte à la position souhaitée ici (140,220) en pixels , le texte en "", la couleur en (R,V,B) et la font définit précédemment : "draw.text((140,220), "SNT", (0,255,255), font=font1)"

