

# Analyse d'un fichier image

Ouvrez le fichier 1.bmp avec Hexed. Vous devriez voir à peu près cela :

File Information		
File Name	1.bmp	
File Size	58 bytes	

Data Inspector (Little-endian)		
Type	Unsigned (+)	Signed (±)
8-bit Integer	66	66
Binary	<input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/>

1.bmp x	
00000000	42 4D 3A 00 00 00 00 00
00001000	00 00 36 00 00 00 28 00
00010000	00 00 01 00 00 00 01 00
00011000	00 00 01 00 18 00 00 00
00100000	00 00 04 00 00 00 27 00
00101000	00 00 27 00 00 00 00 00
00110000	00 00 00 00 00 00 00 00
00111000	FF 00

⚠ Un octet est un ensemble de 2 lettres (notation hexadécimale) comme entourée en bleu sur le screen. Ici l'octet noté 42 vaut 66 en valeur décimale (cf screen).

☛ Il est aussi possible d'utiliser la console python pour faire rapidement la conversion. Il suffit de faire précéder le code hexadécimal de 0x :

```
>>> 0x42
```

```
66
```

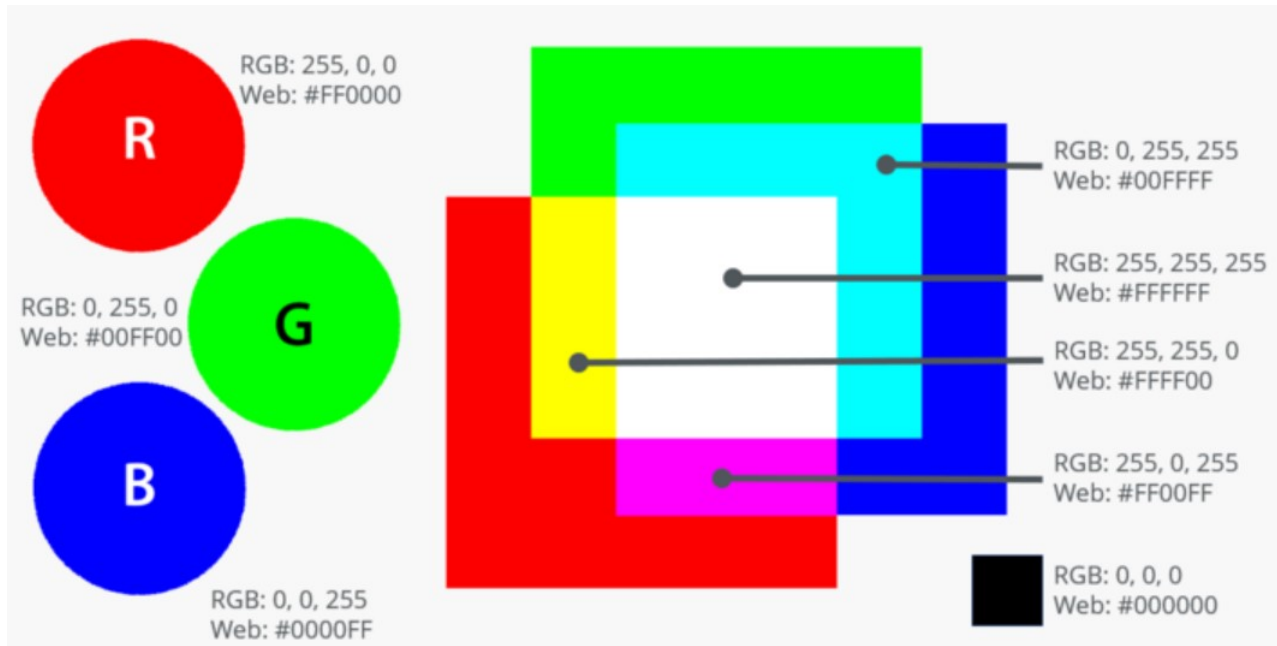
# Analyse de l'en tête du fichier image

- Les deux premiers octets sont .... et .... : cela correspond aux lettres .... et ....., ce qui signifie que le fichier est un fichier .....
- Les quatre octets suivants sont .....,.....,.... et .... : cela correspond à la taille du fichier en octets : Le fichier fait .....octets.
- Les octets 19 à 22 donnent la largeur de l'image : .....
- Les octets 23 à 26 donnent la hauteur de l'image : .....
- Les octets 29 à 32 le nombre de bits pour coder chaque pixel : on trouve ici 24 bits (3 octets), c'est à dire 8 bits (1 octets) pour chacune des couleurs Bleu, vert et Rouge (ds cet ordre!).

# Système RGB

Chaque pixel est colorié en fonction d'un système RGB, sur 3 octets, à partir de 3 couleurs principales : le Vert, le Bleu et le Rouge.

Voici quelques couleurs :



☛ Avec ce système, on peut définir  $256^3$  couleurs différentes !

# Analyse de la couleur du pixel contenu dans le fichier image

- A partir de l'octet 55, on voit les informations de l'image en elle-même!
- On commence par le pixel en bas à gauche. Chaque pixel est codé sur 3 octets (pour les 3 couleurs Bleu, Vert et Rouge). On lit tous les pixels de la ligne, puis on remonte d'un cran, toujours en lisant de gauche à droite...
- Les octets 55,56 et 57 sont ....., ..... et .... , ce qui correspond à un pixel de couleur ....

# Savoir lire un fichier .bmp et découvrir les 8 couleurs cachées !

On donne le fichier suivant :

00000000	42	4D	4E	00	00	00	00	00	BMN.....
00001000	00	00	36	00	00	00	28	00	..6... (.
00010000	00	00	08	00	00	00	01	00	.....
00011000	00	00	01	00	18	00	00	00	.....
00100000	00	00	18	00	00	00	27	00	.....'
00101000	00	00	27	00	00	00	00	00	..'.....
00110000	00	00	00	00	00	00	00	00	.....
00111000	FF	00	FF	00	FF	00	00	00	. . . . .
01000000	00	00	FF	FF	FF	FF	FF	00	.. . . .
01001000	FF	00	FF	00	FF	FF	+		. . . . .

1. Déterminer la nature du fichier
2. Déterminer la taille du fichier
3. Déterminer La largeur et la hauteur du fichier
4. Déterminer les différentes couleurs (dans l'ordre, de gauche à droite)

☛ Vous trouverez sur le site un programme va donner un carré de couleur rouge !

python qui permet de découvrir la couleur en fonction des paramètres r (Rouge),g (Vert) et b (Bleu). Par exemple,

```
>>> pixel(255,0,0)
```

