

Bases du codage en informatique

Balliot - décembre 2023

Table des matières

I – Le binaire s’impose.....	1
Élément binaire - Bit.....	1
II - L’octet.....	4
III - L’écran.....	6
IV - Le codage des couleurs.....	6
V – Le codage des images numériques.....	8
ANNEXES.....	9
Annexe 1 – Taille d’un fichier Ko, Mo, Go, To.....	9
Annexe 2 – Écriture d’un octet en hexadécimal.....	9
Annexe 3 – Modes de compression d’image.....	9
Annexe 4 - Résolution d’une image.....	10
Annexe 5 – Synthèse des couleurs : additive ou soustractive.....	11
Synthèse additive.....	12
Synthèse soustractive.....	13
Additif ou Soustractif.....	13
Quadrichromie.....	14
Annexe 6 : Exercices de conversion d’un système de couleur à l’autre.....	15
Le lien entre les deux systèmes RVB et CMJ.....	16
Annexe 7 – Image vectorielle ou image bitmap ?.....	18
Annexe 8 – Transparence.....	20
Annexe 9 - Calques.....	21
Annexe 10 – Exercice – Conversion de bases numériques.....	21
Annexe 11 – Programmer une conversion décimal / hexadécimal.....	22
Annexe 12 – Quelques liens.....	22

I – Le binaire s’impose

Pour compter nous utilisons un système décimal, parce que l’on a commencé par compter sur nos doigts et que nous avons dix doigts (sans utiliser les pieds). On a donc inventé dix chiffres et on utilise leur position dans l’écriture des nombres. Ainsi 235 signifie 5 unités plus 3 dizaines plus 2 dizaines de dizaines.

D’autres systèmes ont existé (base douze, base vingt..)

En informatique tout est codé suivant un système binaire. Dans un ordinateur on a un ensemble d’éléments qui ont deux états possibles (tension 0 volt ou tension 5 volts, courant passe ou pas, polarité positive ou négative, etc..).

Le défi : comment représenter une information (nombre, texte, image, etc) en n’utilisant que des signes binaires ?

Élément binaire - Bit

Binaire : objet n’ayant que deux états possibles s’excluant

oui/non, blanc/noir, 0/1, allumé/éteint, absent/présent, ouvert/fermé..

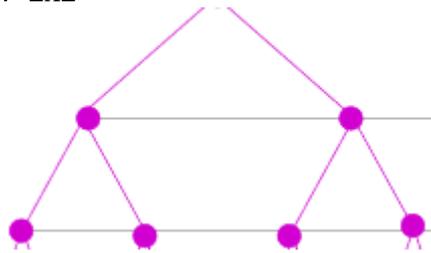
On peut se le représenter sous forme d'une ampoule (allumée ou non), d'un interrupteur. En informatique on a l'habitude, pour simplifier, d'utiliser les deux symboles 0 et 1. Il n'y a pas de 0 ni de 1 dans un ordinateur mais c'est une notation pratique pour se représenter les éléments binaires, que l'on appelle des **bits** (bit = abréviation de binary digit).

Avec un bit on ne peut donc représenter deux informations. En associant plusieurs on augmente le nombre d'informations.

Avec 2 bits : on a les éventualités suivantes :

- le premier est à 0 et le deuxième est à 0
- le premier est à 0 et le deuxième est à 1
- le premier est à 1 et le deuxième est à 0
- le premier est à 1 et le deuxième est à 1

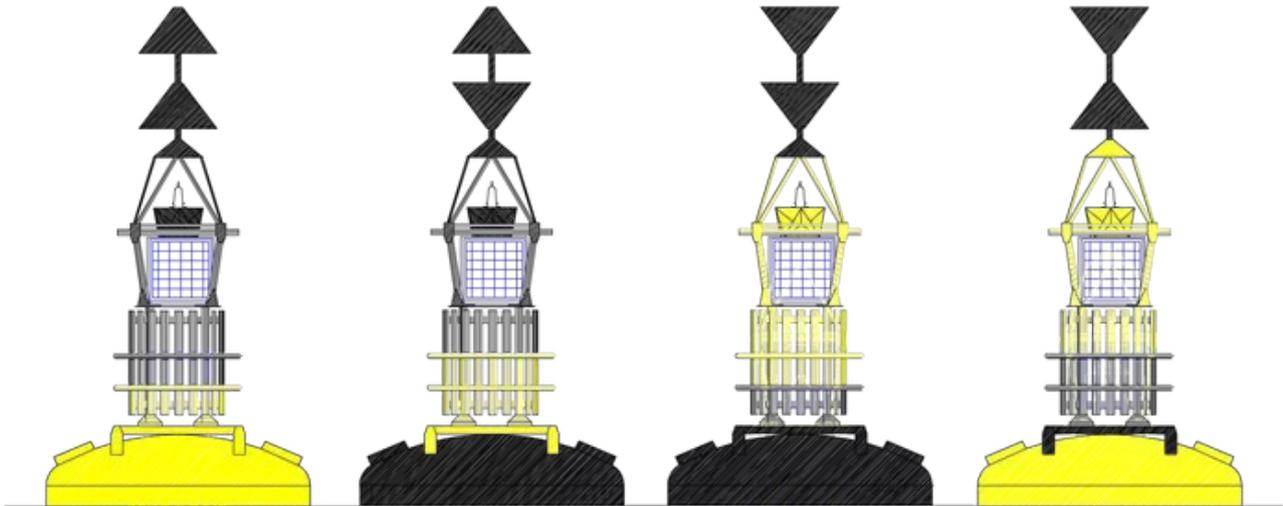
Donc 4 informations différentes $4=2 \times 2$



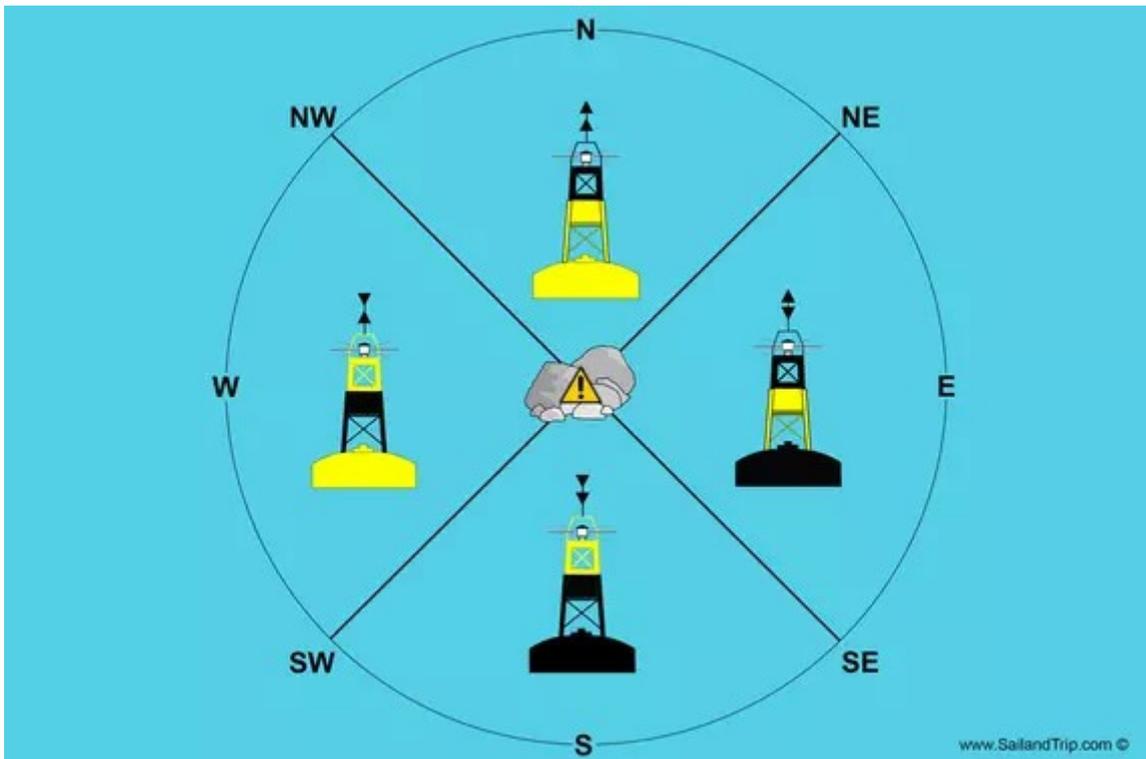
Visualisation avec un arbre binaire

Exemple d'utilisation de deux signaux binaires : le balisage cardinal en mer.

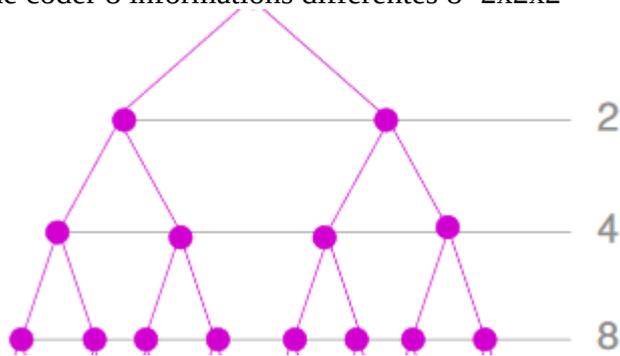
Les deux signes sont un triangle avec pointe vers le haut et un triangle avec pointe vers le bas. En associant deux triangles on désigne les quatre points cardinaux Nord, Est, Sud, Ouest.



Ces balises signalent la présence d'un danger et indiquent de quel côté passer pour l'éviter.



Avec 3 bits : pour chacune des éventualités précédentes on on a deux, donc on obtient la possibilité de coder 8 informations différentes $8=2 \times 2 \times 2$



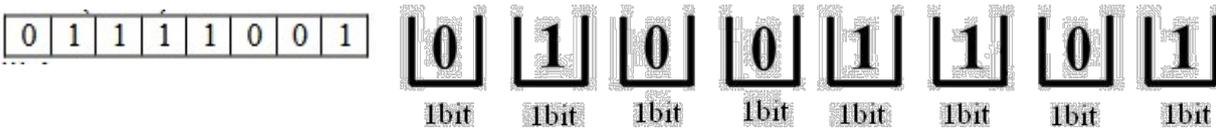
Et avec 4 bits combien d'informations ? Et avec 5, avec 6 ?

II - L'octet

Au minimum on a besoin de savoir coder les chiffres, les lettres minuscules et majuscules, les signes de ponctuation, etc.

On a fait le choix d'utiliser des séquences de 8 bits, appelées octets qui permettent de coder 256 informations différentes. Donc en informatique l'élément de base est l'**octet** (byte en anglais).
 $256 = 2 \times 2 = 2^8$

Représentation imagée d'un octet :



TOUT est codé avec des octets

Par exemple la lettre **A** majuscule est codée 01000001 soit 65 en décimal ou 41 en hexadécimal, tandis que la lettre **a** minuscule est codée 01100001 (code ASCII), un seul bit les différencie. Certains caractères spéciaux comme les lettres accentuées nécessitent 2 octets.

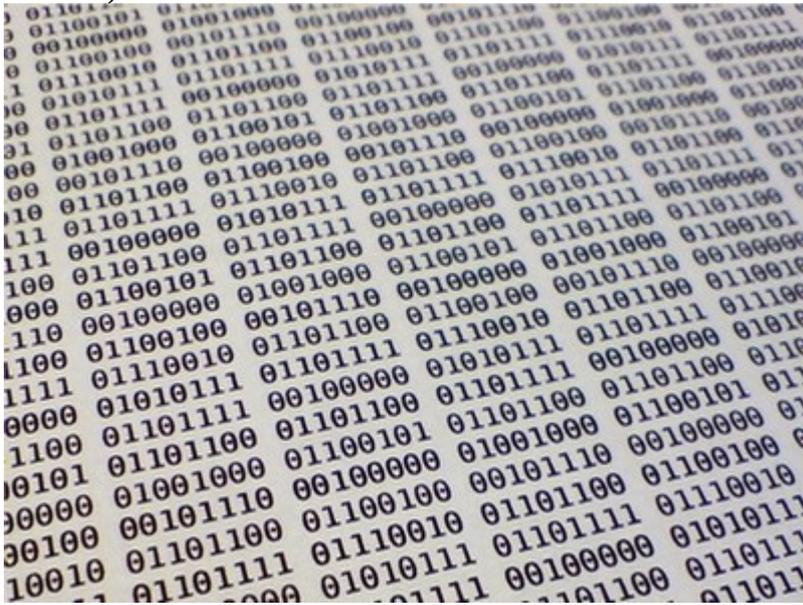
Ci-dessous les codes ASCII des caractères de base :

Dec	Hex	char		Dec	Hex	char	Dec	Hex	char	Dec	Hex	char
0	00	NUL	caract. null	32	20	espace	64	40	@	96	60	'
1	01	SOH		33	21	!	65	41	A	97	61	a
2	02	STX		34	22	"	66	42	B	98	62	b
3	03	ETX		35	23	#	67	43	C	99	63	c
4	04	EOT		36	24	\$	68	44	D	100	64	d
5	05	ENQ		37	25	%	69	45	E	101	65	e
6	06	ACK		38	26	&	70	46	F	102	66	f
7	07	BEL	bell	39	27	'	71	47	G	103	67	g
8	08	BS	backspace	40	28	(72	48	H	104	68	h
9	09	TAB	tabul. horiz	41	29)	73	49	I	105	69	i
10	0A	LF	line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	Tabul. vertic	43	2B	+	75	4B	K	107	6B	k
12	0C	FF		44	2C	,	76	4C	L	108	6C	l
13	0D	CR	retour ligne	45	2D	-	77	4D	M	109	6D	m
14	0E	SO		46	2E	.	78	4E	N	110	6E	n
15	0F	SI		47	2F	/	79	4F	O	111	6F	o
16	10	DLE		48	30	0	80	50	P	112	70	p
17	11	DC1		49	31	1	81	51	Q	113	71	q
18	12	DC2		50	32	2	82	52	R	114	72	r
19	13	DC3		51	33	3	83	53	S	115	73	s
20	14	DC4		52	34	4	84	54	T	116	74	t
21	15	NAK		53	35	5	85	55	U	117	75	u
22	16	SYN		54	36	6	86	56	V	118	76	v
23	17	ETB		55	37	7	87	57	W	119	77	w
24	18	CAN		56	38	8	88	58	X	120	78	x
25	19	EM		57	39	9	89	59	Y	121	79	y
26	1A	SUB		58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	escape	59	3B	;	91	5B	[123	7B	{
28	1C	FS		60	3C	<	92	5C	\	124	7C	
29	1D	GS		61	3D	=	93	5D]	125	7D	}
30	1E	RS		62	3E	>	94	5E	^	126	7E	~
31	1F	US		63	3F	?	95	5F	_	127	7F	DEL

*Voir en annexe 10 la conversion de bases pour l'écriture d'un nombre.

Ex : 65 en décimal (base dix) s'écrit 01000001 en binaire (base deux) et 41 en hexadécimal (base seize).

Voilà comment on peut s’imaginer la mémoire d’un ordinateur (sans correspondance avec la réalité bien sûr) suite d’octets



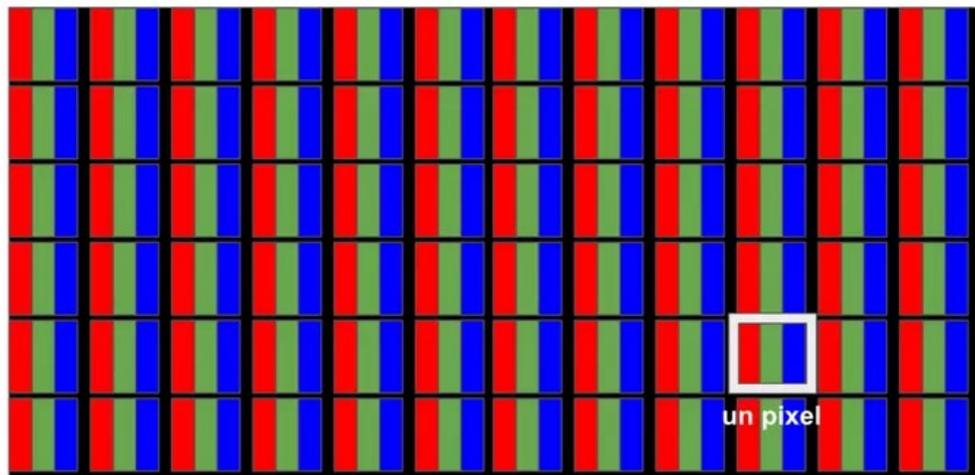
Toute l’information est stockée sous forme de suite d’octets que l’on appelle un **fichier**, et c’est ainsi qu’elle circule, jusqu’à un terminal, par exemple un écran, où elle est décodée.

III - L'écran

Un écran d'ordinateur, de tablette, de smartphone, est constitué d'un ensemble de petits rectangles lumineux nommés pixels.

La résolution d'écran est généralement exprimée en largeur x hauteur, par exemple 1920 x 1080, ce qui signifie qu'il y a 1920 pixels en largeur et 1080 pixels en hauteur.

Chaque pixel correspond à un rayon lumineux d'une couleur donnée, couleur obtenue par composition de trois rayons, un rouge, un vert, et un bleu.



IV - Le codage des couleurs

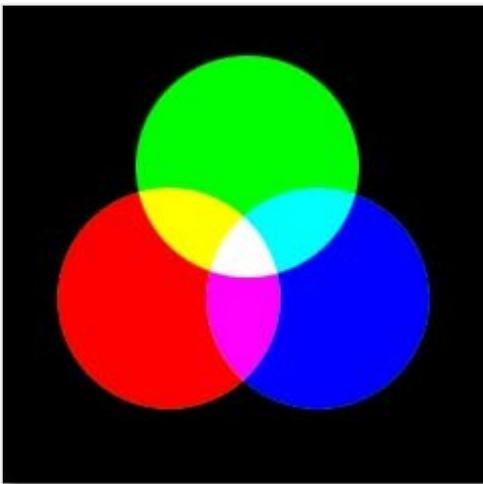
Les trois couleurs de base en informatique sont **RVB** – Rouge Vert Bleu – (RGB en anglais).

Modèle additif

Le postulat de la synthèse additive est tout simple : en utilisant seulement trois des couleurs de l'arc-en-ciel, il est possible de reconstituer la lumière blanche, en d'autres termes en additionnant la couleur des trois faisceaux rouge, vert, bleu, on obtient le blanc. Leur mélange en différentes proportions donnera toutes les autres couleurs. Le mélange deux par deux des primaires donnera une couleur deux fois plus claire qui est la complémentaire de la troisième. La synthèse additive concerne tous les mélanges de couleurs d'origine lumineuse tels que les spots, projecteurs trichromes, moniteurs CRT ou LCD, scanners, appareils photo numériques, etc.



Projecteur de scène



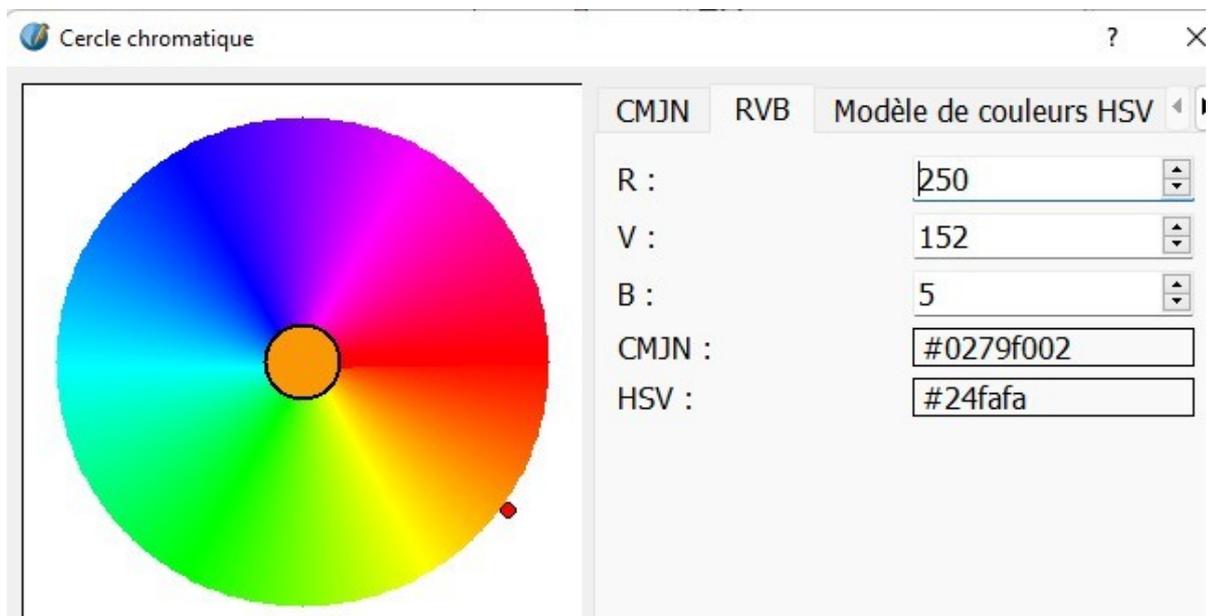
Synthèse additive des couleurs

Chacune des 3 couleurs de base RVB est codée par un octet, donc une couleur est codée par trois octets.

Pour chaque composante on a 256 niveaux de luminosité. Si on les note de 0 à 255 le rouge sera représenté par le triplet d'octets (255,0,0), le vert par (0,255,0), le bleu par (0,0,255), le blanc par (255,255,255), le noir par (0,0,0).

On utilise plutôt le système hexadécimal pour définir une couleur. Ainsi le rouge sera noté #FF0000 le # indiquant qu'il s'agit d'une notation hexadécimale, l'hexadécimal FF étant le décimal 255 et l'hexadécimal 00 étant le décimal 0 . **Voir l'annexe 2.**

Lorsque l'on utilise la palette de choix des couleurs dans un logiciel de traitement d'image ou de dessin on voit apparaître d'autres systèmes de codage des couleurs



Pour le orange choisi (voir disque au centre) on obtient les codes couleurs ci-dessus

On a le système CMYK ou CMYK utilisé en imprimerie et par les imprimantes et également le système TSL (teinte, saturation, luminosité) ou TSV (teinte, saturation, valeur,, ce dernier étant davantage basé sur la perception des couleurs . Voir annexes.

V – Le codage des images numériques

L'image numérique

Pour transmettre une image ou une photo on utilise un fichier donnant pour chaque pixel le code de sa couleur. On appelle image numérique une image ainsi définie. A partir d'un fichier on reconstitue un rectangle de pixels.

Une image numérique est une matrice rectangulaire de pixels, donc de triplets d'octets.(bitmap).

Avec l'appareil photo numérique on définit pour chaque photo un ensemble de pixels. On peut obtenir des fichiers plus ou moins gros selon la résolution choisie, par exemple 1600x1200.

Exemple d'une photo classique 3264x1840 pixels, elle contient 6 millions de pixels environ donc sera codée par 18 millions d'octets (environ 18 Mo ou 18 Mb) ce qui est lourd pour un fichier.

Heureusement on a mis au point des systèmes de compression d'image qui permettent de les partager sans perte de précision ou avec perte raisonnable grâce à des fichiers moins lourds.

On reconnaît les différents formats d'image par leur suffixe.

Les formats principaux de fichiers d'images

.bmp ou **.tiff** → tous les pixels sont mémorisés, aucune compression.

.png → compression sans perte et **avec gestion de la transparence**

.jpg ou **.jpeg** → compression avec perte, images plus légères

.webp → idem à jpg pour le web

.gif → pour images animées

.svg → image vectorielle (autre système de codage très différent basé sur les courbes de Bézier)

Avec la compression JPEG on divise à peu près par 10 la taille du fichier

Remarque : bitmap ou vectoriel ?

Une image définie par l'ensemble de ses pixels est appelée une image **bitmap**. Si on l'agrandit beaucoup on voit les pixels flous (car en nombre limité) et elle perd en précision, devient floue.

Une image **vectorielle** est définie non par ses pixels mais par des formules mathématiques et peut donc être agrandie sans perte de précision, ce qui est un gros avantage.

Un logiciel de base simple et gratuit (logiciel libre) pour créer des images vectorielles : **Inkscape**. Parfait pour créer des logos.

ANNEXES

Annexe 1 – Taille d'un fichier Ko, Mo, Go, To

La taille d'un fichier ou d'une mémoire est égal à son nombre d'octets (ou bytes), elle est notée en KiloOctets Ko ou MegaOctets Mo ou GigaOctets ou Go ou TeraOctets ou To.

Mais en informatique un Kilo n'est pas 1000 mais 1024.

Le système décimal dans lequel nous écrivons les nombres, en utilisant 10 chiffres, n'est pas adapté à l'informatique où tout est binaire. 1000 n'est pas une puissance de 2 donc pas facile à utiliser en binaire, alors que $1024 = 2^{10}$

1Ko=1024 octets, 1Mo=1024x1024 octets, etc. Mais en pratique 1Mo c'est à peu près 1 million d'octets.

Annexe 2 – Écriture d'un octet en hexadécimal

Écrire un octet en utilisant le binaire n'est pas pratique (8 chiffres)

10001110 ou 01110101 ou 11111111 pour « le plus grand »

En l'écrivant en décimal ce n'est pas pratique non plus, 0 pour le plus petit ou 255 pour le plus grand, la conversion d'un système à l'autre étant fastidieuse.

Les informaticiens utilisent donc le système hexadécimal (à base 16) car 16 étant une puissance de 2 les conversions avec le binaire sont très simples.

Tout octet en hexadécimal s'écrit avec deux chiffres seulement au lieu de huit en binaire.

Les seize chiffres héxa sont 0 1 2 3 4 5 6 7 8 9 A B C D E F

Exemple 11111111 s'écrit FF. On remarquera que le premier chiffre hexa est obtenu avec les quatre premiers chiffres binaires et le second avec les quatre derniers, donc conversion très facile.

Exemple l'octet 01011011 s'écrira 5C en hexa car 0101 → 5 et 1011 → C

Annexe 3 – Modes de compression d'image

Compression sans perte (exemple PNG) : c'est juste une façon différente de coder. Si dans une image on a 2000 pixels blancs consécutifs, au lieu d'écrire 2000 fois le pixel blanc (soit 6000 octets puisqu'un pixel correspond à une couleur codée par 3 octets, on écrira 2000 blancs, c'est à dire 2 informations (la couleur et le nombre)

Compression avec perte (exemple JPG) : en plus on assimilera des couleurs voisines à une même couleur. Légère perte de précision plus ou moins poussée.

(en réalité c'est un peu plus compliqué et on applique des algorithmes mathématiques élaborés, mais le principe reste valable)

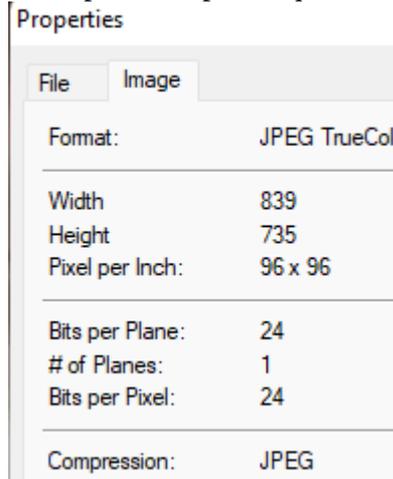
Souvent un fichier JPG pèse 6 à 10 fois moins que le fichier d'origine (BMP par exemple)

Annexe 4 - Résolution d'une image

Une image numérique contient un nombre défini de pixels. Si on l'agrandit on espace les pixels et on a une image plus floue. La résolution d'une image est le nombre de pixels par unité de longueur lors de l'affichage.

- nombre de pixels par cm ou par pouce (unité Pixels Per Inch)
- pour Web 72 ppi (ppi ou ppp ou dpi)
 - pour imprimante ≥ 300 ppi
 - pour travaux précis ou agrandissements 600, 1200 ppi

Voici par exemple ce que l'on peut lire dans les propriétés d'une image



Properties	
File	Image
Format:	JPEG TrueColor
Width	839
Height	735
Pixel per Inch:	96 x 96
Bits per Plane:	24
# of Planes:	1
Bits per Pixel:	24
Compression:	JPEG

- et pour une autre :

Dimensions	5184 x 3888
Largeur	5184 pixels
Hauteur	3888 pixels
Résolution horizontale	350 ppp
Résolution verticale	350 ppp
Profondeur de couleur	24
Compression	
Unité de résolution	2
Représentation des couleurs	sRGB

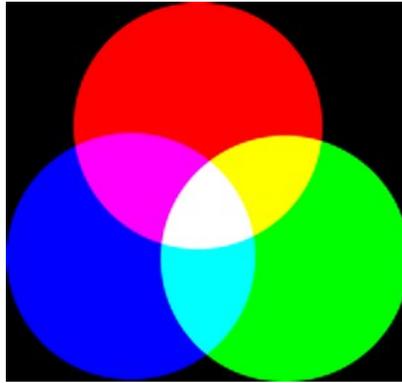
- la profondeur de couleur est le nombre de bits par couleur $24=8 \times 3$

Annexe 5 – Synthèse des couleurs : additive ou soustractive

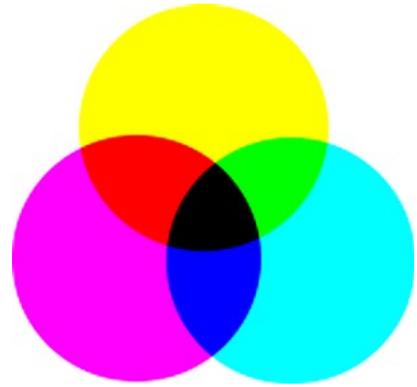
En informatique on utilise la synthèse additive des 3 couleurs de base RVB car il s'agit de composer des rayonnement lumineux.

En peinture ou en imprimerie on mélange des encres. Le mécanisme correspond à une synthèse soustractive. Il ne s'agit plus de rayonnements mais d'absorption de rayonnements. C'est le système CMJ (cyan, magenta, jaune) ou CMY en anglais.

Il existe un système de passage de l'un à l'autre, de RVB à CMJ.



Synthèse additive

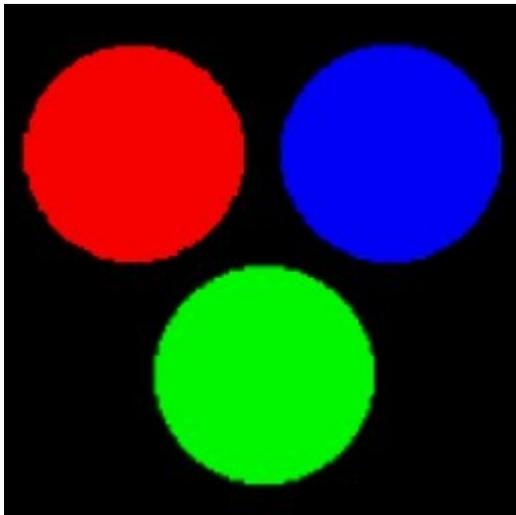


Synthèse soustractive

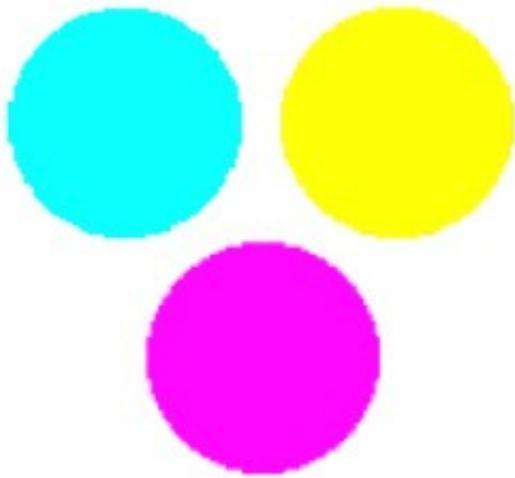
Un peu de théorie

Il faut distinguer les manières dont s'effectue la synthèse des couleurs.

Elle peut être **additive** : on part du noir et on ajoute des couleurs (cas d'un écran noir sur lequel on envoie des rayons lumineux).



Elle peut être **soustractive** : on part du blanc et on obscurcit au moyen de couleurs (encres)



(cas d'utilisation d'encre sur un papier).



Synthèse additive

C'est ce qui est utilisé en télévision et sur l'écran de votre ordinateur. Chaque pixel peut recevoir un mélange de 3 rayons lumineux. Les couleurs fondamentales sont **rouge, vert, bleu**, soit **RVB** en français ou **RGB** en anglais (red, green, blue).

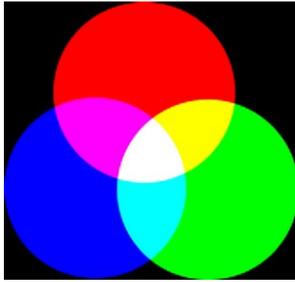
En informatique chaque composante étant codée sur un octet, une couleur sera caractérisée par 3 octets, ou un triplet de nombres entiers compris entre 0 et 255 représentant les intensités des 3 couleurs de base, ou entre 00 et FF en hexadécimal plutôt utilisé par les informaticiens.

Ce système permet donc de coder $2^8 \times 2^8 \times 2^8 = 2^{24}$ soit 16 777 216 couleurs différentes !

Exemples

Couleur	Code RVB décimal	Code RVB hexadécimal
Rouge	(255, 0, 0)	(FF, 00, 00) ou #ff0000
Vert	(0, 255, 0)	(00, FF, 00) ou #00ff00
Bleu	(0, 0, 255)	(00, 00, FF) ou #0000ff
Un orange	(170, 85, 0)	(AA, 55, 00) ou #aa5500
Blanc	(255, 255, 255)	(FF, FF, FF) ou #ffffff
Noir	(0, 0, 0)	(00, 00, 00) ou #000000
Jaune	(255, 255, 0)	(FF, FF, 00) ou #ffff00
Un gris	(123, 123, 123)	(7B, 7B, 7B) ou #7b7b7b
Fuchsia ou Magenta	(255, 0, 255)	(FF, 00, FF) ou #ff00ff

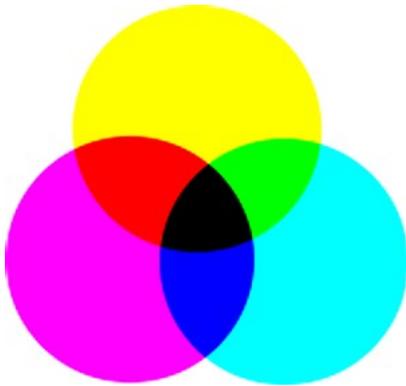
L'addition des couleurs :



Synthèse soustractive

C'est ce qui se passe quand on peint avec de l'aquarelle, ou que l'on imprime en utilisant un mélange d'encres.

Les couleurs fondamentales sont alors le cyan, le magenta, et le jaune, soit **CMJ** en français ou **CMY** en anglais (avec yellow pour jaune).



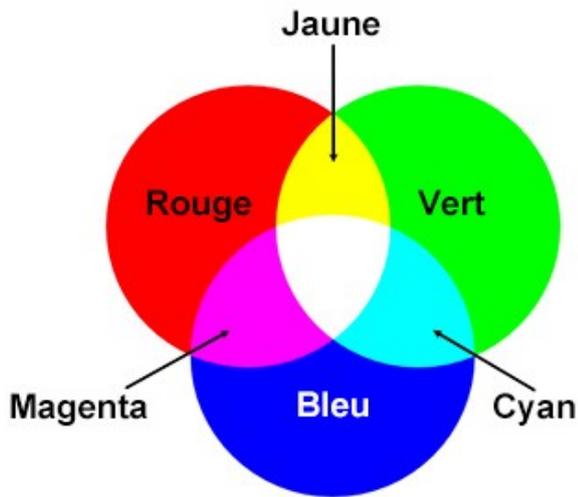
En superposant du jaune et du magenta on obtiendra du rouge (le jaune "mange" le bleu du magenta) ; de même du jaune et du cyan donneront du vert et la superposition du magenta et du cyan donnera du bleu. Enfin la superposition des 3 couleurs donnera du noir.

Additif ou Soustractif

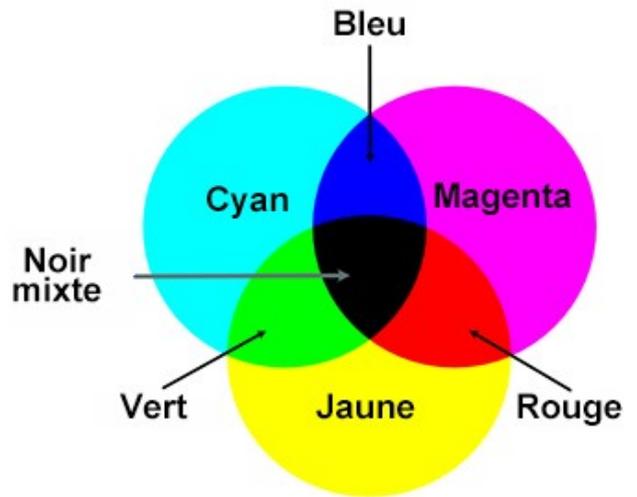
On voit une tomate rouge sur un écran : ce sont des rayons lumineux rouges qui sont **émis** (lumière positive).

On voit une tomate rouge sur une affiche : l'encre rouge a absorbé tous les rayons lumineux **sauf** la couleur rouge. Le rouge du papier nous renvoie le rouge qui n'a pas été absorbé (système négatif).

	Synthèse additive	Synthèse soustractive
Obtention	Rayons lumineux	Encres, pigments
Couleurs primaires	RVB rouge, vert, bleu	CMJ cyan, magenta, jaune
Les 3 couleurs de base ensemble	Lumière blanche	Noir obtenu par mélange des 3
Aucune	Noir, pas de lumière	Blanc, absence d'encre
Aspect rouge	Rayon rouge seul	Tache absorbant tout sauf le rouge
Aspect jaune	Rayon rouge + rayon vert	Encre jaune (absorbe les rayons bleus)



Couleurs additives



Couleurs soustractives

On remarque que les couleurs secondaires de chaque système sont les couleurs primaires de l'autre.

Quadrichromie

En fait les imprimeurs utilisent la **quadrichromie**. Il est plus facile, plus efficace, et moins coûteux d'utiliser du noir plutôt que de chercher à l'obtenir en mélangeant les couleurs de base.

Pour obtenir la teinte voulue on ajoute donc aux 3 couleurs de base, une 4^{ème}, le noir. On a donc du **CMJN** en français ou **CMYK** en anglais (K pour black).

A noter que la plupart des imprimantes utilisent 4 cartouches d'encre CMJN et le logiciel de l'imprimante convertit la couleur RVB en une couleur CMJN afin de répartir les jets d'encre.

Annexe 6 : Exercices de conversion d'un système de couleur à l'autre

Dans la suite on définira une couleur soit par 3 entiers R, V, B, soit par un vecteur RVB de 3 entiers.

Question Q1 :

Un gris (100,100,100) deviendra plus clair en multipliant ses valeurs par 1,5 devenant (150,150,150). On ne peut bien sûr pas les multiplier par 3.

On veut augmenter l'intensité d'une couleur en utilisant le principe suivant : on multiplie par un même coefficient les trois composantes, sans dépasser le maximum 255 pour la plus intense ce qui limite la valeur maximale du coefficient multiplicateur.

- Écrire une fonction **PlusIntense** prenant comme paramètre une couleur RVB et le coefficient proposé et qui renvoie la nouvelle couleur plus intense.

Question Q2 :

- Écrire une fonction **OCTETdecversHEX** qui prend comme paramètre un entier compris entre 0 et 255 et renvoie l'écriture hexadécimale sur 2 caractères de cet entier.

La notation classique en hexadécimal est une chaîne de 7 caractères commençant par # (dite aussi notation HTML car utilisée sur le WEB). Donc (7B,83,A2) s'écrit **#7B83A2**

exemple de tableau de quelques couleurs classiques

Valeur	Couleur	Valeur	Couleur	Valeur	Couleur	Valeur	Couleur
#00FFFF	aqua / cyan (bleu-vert eau)	#008000	green (vert)	#000080	navy (bleu marine)	#C0C0C0	silver (argent)
#000000	black (noir)	#808080	gray (gris)	#808000	olive (jaune olive)	#008080	teal (sarcelle)
#0000FF	blue (bleu)	#00FF00	lime (vert citron)	#800080	purple (violet)	#FFFFFF	white (blanc)
#FF00FF	fuchsia / magenta (fuchsia)	#800000	maroon (bordeaux)	#FF0000	red (rouge)	#FFFF00	yellow (jaune)

- Écrire une fonction **RVBversHEXA** qui transforme les 3 composantes d'une couleur écrites en décimal en écriture hexadécimale de cette couleur.

(255,0,0) → #FF0000

Question Q4 :

- Inversement écrire une fonction **HEXAversRVB** qui, à partir de l'écriture hexadécimale d'une couleur, sous forme de triplet hexadécimal, renvoie son écriture en décimal.

(FF,00,00) → (255,0,0)

Question Q5 :

Reprendre la question précédente mais cette fois-ci en saisissant sur la console les valeurs hexadécimales de la couleur utilisée en entrée et en vérifiant la validité de l'entrée. On accepte indifféremment les lettres en majuscule ou en minuscule.

Le lien entre les deux systèmes RVB et CMJ

En utilisant la notation entre 0 et 255 pour chaque composante on a

$R + C = 255$, soit $C = 255 - R$ (cyan complément du rouge)

$V + M = 255$, soit $M = 255 - V$ (magenta complément du vert)

$B + Y = 255$, soit $Y = 255 - B$ (jaune complément du bleu)

La couleur RVB = (100, 150, 200) est donc identique à la couleur CMJ = (155, 105, 55).

CMYK to RGB conversion formula

The R, G, B values are given in the range of 0..255.

The red (R) color is calculated from the cyan (C) and black (K) colors:

$$R = 255 \times (1-C) \times (1-K)$$

The green color (G) is calculated from the magenta (M) and black (K) colors:

$$G = 255 \times (1-M) \times (1-K)$$

The blue color (B) is calculated from the yellow (Y) and black (K) colors:

$$B = 255 \times (1-Y) \times (1-K)$$

Question Q6 :

- Écrire une fonction **RVBversCMJ** qui transforme une couleur RVB en une couleur CMJ (en utilisant les notations décimales des octets)

Question Q7 :

En fait les composantes CMJ sont souvent indiquées avec une échelle de 0 à 100 et non en valeurs d'octets.

Un orange RVB = (170,85,0) composé de 2/3 de rouge et 1/3 de vert sera noté en CMJ
CMJ= (85,170,255) en échelle 0-255 ou plutôt CMJ = (33, 67, 100) en échelle 0-100.

- Écrire une fonction **CMJ255versCMJ100** qui transforme une couleur CMJ en valeurs entières de 0 à 255 en une couleur CMJ en valeurs entières correspondantes en échelle 0-100.

CMJ (85,170,255) → CMJ (33, 67, 100)

Quadrichromie

En fait les imprimeurs utilisent la **quadrichromie**. Il est plus facile, plus efficace, et moins coûteux d'utiliser du noir plutôt que de chercher à l'obtenir en mélangeant les couleurs de base.

Pour obtenir la teinte voulue on ajoute donc aux 3 couleurs de base, une 4^{ème}, le noir. On a donc du **CMJN** en français ou **CMYK** en anglais (K pour black).

A noter que la plupart des imprimantes utilisent 4 cartouches d'encre CMJN et le logiciel de l'imprimante convertit la couleur RVB en une couleur CMJN afin de répartir les jets d'encre.

Exemple :

Les valeurs des composantes couleurs sont données là en échelle 0-100.

Soit la couleur CMJ=(61, 80, 22). 22 est la plus petite valeur.

On utilisera du noir d'intensité 22 et on diminuera les autres couleurs suivant une proportion calculée en fonction de la dose de noir employée.

La formule utilisée est :

$$N = \min (C, M , J) ;$$

$$C \text{ nouveau} = 100(C-N)/(100-N)$$

$$M \text{ nouveau} = 100(M-N)/(100-N)$$

$$J \text{ nouveau} = 100(J-N)/(100-N)$$

On utilisera ainsi la couleur CMJN = (50,74,0,22) en quadrichromie.

Question Q8 :

- Écrire une fonction **CMJversCMJN** qui transforme une couleur CMJ en couleur CMJN en utilisant les formules données ci-dessus.

Conversion RVB en CMJN

En réalité les 4 valeurs d'une couleur CMJN sont presque toujours écrites sur une échelle de 0 à 100, par exemple 56 pour 56 %, et la conversion se fait directement à partir de la couleur RVB. On va convertir une couleur RVB utilisant l'échelle 0-255 en son correspondant CMJN utilisant l'échelle 0-100.

Les formules utilisées sont :

On convertit d'abord R, V et B en rp, vp, bp d'échelle 0-100 (on reste avec des entiers)

$$rp = 100R/255 ; vp = 100V/255 ; bp = 100B/255 ;$$

On détermine le plus grand des trois : maxip = max(rp, vp, bp) ;

La composante Noir : N=100 – maxip // complément à 100 du plus grand des trois.

$$\text{Puis } C = 100(\text{maxip}-rp)/\text{maxip} ; M = 100(\text{maxip}-vp)/\text{maxip} ; J = 100(\text{maxip}-bp)/\text{maxip} ;$$

on obtient par exemple RVB(100,50,200) ---> CMJN(50,75,0,22)

Question Q9 :

- Écrire une fonction **RVBversCMJN** qui prend en entrée une couleur en mode RVB échelle 0-255 donc valeur décimale des octets et la convertit en mode CMJN (valeurs entières de 0 à 100).

Quelques valeurs pour vérifier et compléter

Couleur	RVB	RVB hexa (html)	CMJN
Noir	0,0,0	#000000	0,0,0,100
Blanc	255,255,255	#FFFFFF	0,0,0,0
Rouge	255,0,0	#FF0000	0,100,100,0
Vert	0,255,0	#00FF00	100,0,100,0
Bleu	0,0,255	#0000FF	100,100,0,0
Jaune	255,255,0	#FFFF00	0,0,100,0
Cyan	0,255,255	#00FFFF	100,0,0,0
Magenta	255,0,255	#FF00FF	0,100,0,0

Jaune d'or		#E4A700	0,27,100,6
	100,50,200		50,74,0,22
Marron	128,0,0	#800000	
Pourpre	128,0,128	#800080	
	109,81,184		41,56,0,28
Olive		#808000	
Gris		#808080	
Gris argent		#C0C0C0	
Bleu Navy		#000080	

En utilisant un logiciel de dessin (Paint ou autre) voir la palette des couleurs, en choisir une et voir son codage proposé par le logiciel.

Question Q11 Bonus

- Écrire une fonction **CMJNversRVB** qui, à partir du code CMJN d'une couleur calculera son code RVB.

Annexe 7 – Image vectorielle ou image bitmap ?

Une image bitmap est un ensemble fini de pixels, chaque pixel ayant une valeur de couleur. Si on zoome une telle image on voit apparaître les pixels et l'image devient floue.

Une image vectorielle est définie par des formules mathématiques basées sur les courbes de Bézier.

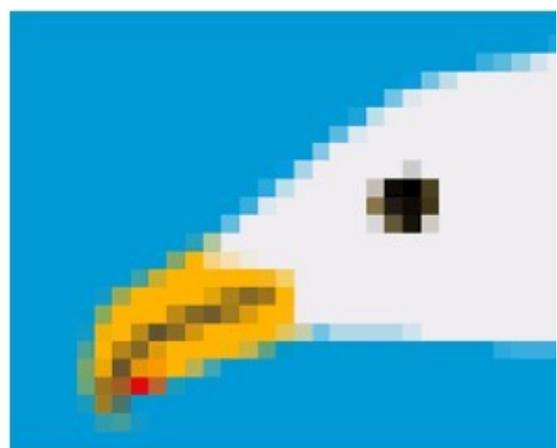
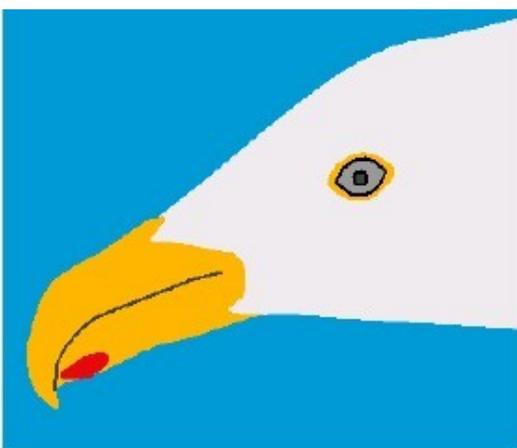
Pierre Bézier est le mathématicien ayant mis au point ce système.

Le dessin vectoriel a l'avantage de ne pas souffrir du phénomène de la pixelisation lorsque l'on cherche à faire un zoom sur une partie ou la totalité de l'image. En effet, dans ce cas, automatiquement les courbes composant le dessin sont recalculées pour être mises à l'échelle.

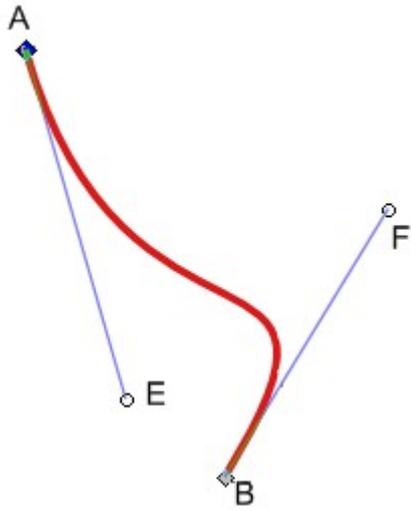
Inkscape est un logiciel libre simple de dessin vectoriel. Idéal pour créer des logos..

[Dessin vectoriel avec Inkscape, tuto vidéo](#)

[Courbe de Bézier avec Inkscape tuto vidéo](#)



Zoom sur la tête d'un goéland : à gauche à partir d'une image vectorielle, à droite à partir d'une image bitmap.



Une courbe de Bézier (chemin) comporte des nœuds (point d'ancrage) et des poignées (tangentes) permettant de régler les courbures.

Scribus le logiciel libre de PAO, présentation et mise en page, contient également une option de dessin vectoriel.

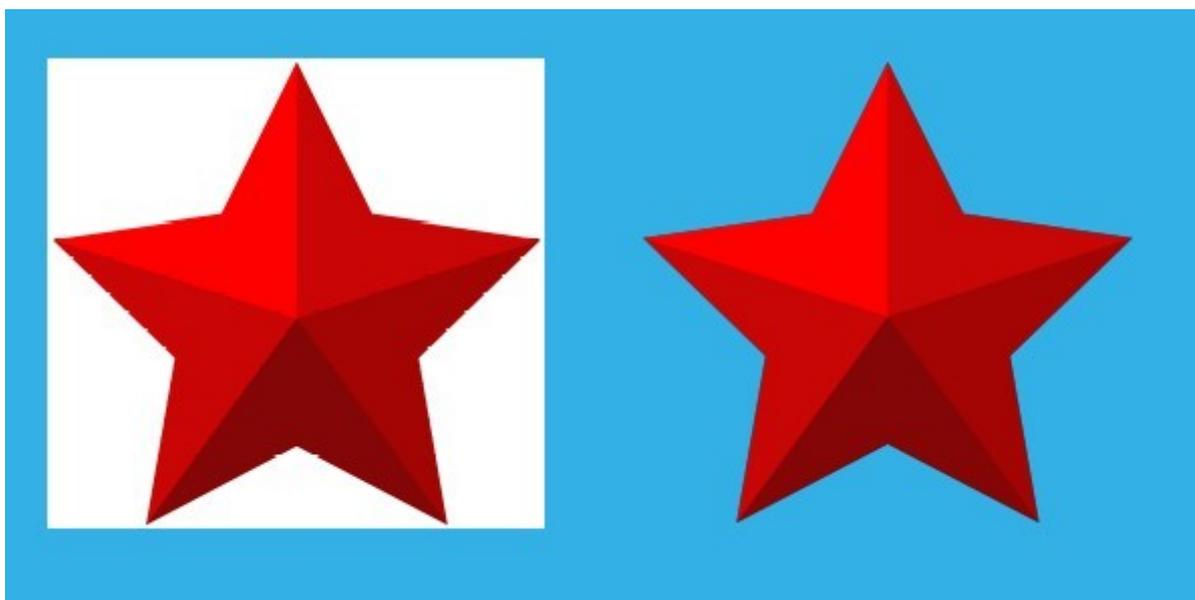
En mathématiques niveau lycée le logiciel **Géogébra** gère également les courbes de Bézier.

Annexe 8 – Transparence

Certains formats d'image, comme le png, permettent de gérer la transparence. Une image jpg sera affichée dans un rectangle quelque soit le fond, alors qu'une image png ayant une transparence sur une partie du cadre rectangulaire permettra de n'afficher sur le fond choisi que la partie non transparente de l'image.



la partie transparente est signalée par un pavage gris et blanc



à gauche l'image JPG, à droite l'image PNG, sur un même fond

Le format JPG ne gère pas la transparence, l'image est toujours affichée dans un cadre rectangulaire.

Annexe 9 - Calques

La plupart des logiciels de traitement d'image offrent la possibilité de travailler sur des calques. On utilise ainsi plusieurs couches superposables offrant des possibilités étendues.

Photofiltre : un petit logiciel libre gérant bien les calques.

Annexe 10 – Exercice – Conversion de bases numériques

Ex : 65 en décimal (base dix) s'écrit 01000001 en binaire (base deux) et 41 en hexadécimal (base seize).

Un nombre qui s'écrit $c_4c_3c_2c_1c_0$ dans une base b a pour valeur $c_0+c_1xb^1+c_2xb^2+c_3xb^3+c_4xb^4$
 ex : 41 en base seize c'est 1 unité + 4 x 16 donc 1 +64=65 en décimal

Écrire un même nombre dans des bases différentes

Préparer une boîte à 4 ou 5 cases pour recevoir les chiffres, 26 bâtonnets, ou allumettes, quelques élastiques.



Prenons 17 bâtonnets. Comment écrire ce nombre.

Dans notre système décimal on groupe les objets par paquets de dix*, le reste est compris entre 0 et 9 donc pour le représenter on a besoin de dix chiffres notés 0,1,2,3,4,5,6,7,8,9.

On a un paquet de dix et il reste 7 unités d'où l'écriture **17**

* pourquoi 10, sans doute parce que l'on a dix doigts et que l'on a commencé à compter sur nos doigts

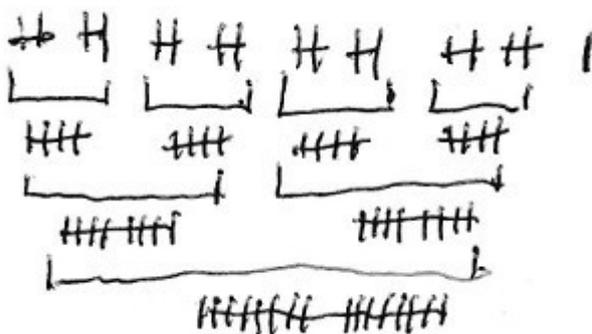
Dans la base 5

On groupe les objets par paquets de cinq, le reste est compris entre 0 et 4 donc pour le représenter on a besoin de cinq chiffres notés 0,1,2,3,4.

On a trois paquets de cinq et il reste 2 unités d'où l'écriture **32**

Dans la base 2 (système binaire)

On groupe les objets par paires puis par paires identiques, le reste est 0 ou 1 donc pour le représenter on a besoin de deux chiffres notés 0,1.



On obtient

un paquet de 2x2x2x2

aucun de 2x2x2

aucun de 2x2

aucun de 2

un reste de 1

.

D'où l'écriture **10001**

Dans la base 16 (système hexadécimal)

On groupe les objets par paquets de seize, le reste est compris entre 0 et 15 donc pour le représenter on a besoin de seize chiffres notés 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

On trouve un paquet de seize et il reste un

D'où l'écriture **11**

Recommencer tout avec 26 objets

On trouve

base 5 → 101

base 2 → 11010

base 16 → 1A

Annexe 11 – Programmer une conversion décimal / hexadécimal

En langage Python écrire une procédure – fonction – de passage de l'écriture d'un octet du décimal à l'hexadécimal puis de l'hexadécimal au décimal

Annexe 12 – Quelques liens

Couleurs

[De toutes les couleurs](#)

[Quelques codes de couleurs](#)

[Lumière et couleurs, vidéo](#)

[Couleurs primaires RVB ou CMJ ? Vidéo](#)

[RVB en informatique, CMJN en imprimerie, vidéo](#)

Scribus, logiciel libre de PAO pour affiches ou tracts

[Mon tutoriel Scribus](#)

[Scribus, les fiches du prof Tim](#)

Inkscape, logiciel libre pour le dessin vectoriel

[Dessin vectoriel avec Inkscape, tuto vidéo](#)

[Courbe de Bézier avec Inkscape, tuto vidéo](#)

[Inkscape, les fiches du prof Tim](#)