

Introduction à l'informatique

pour les mathématiques, la physique et les sciences
computationnelles

Yann Thorimbert



**UNIVERSITÉ
DE GENÈVE**

CENTRE UNIVERSITAIRE
D'INFORMATIQUE

Chapitre 2 - Partie 3

Codage des medias

Yann Thorimbert



**UNIVERSITÉ
DE GENÈVE**

CENTRE UNIVERSITAIRE
D'INFORMATIQUE



Chapitres du cours

1. Origines des ordinateurs et des réseaux informatiques
2. Codage des nombres
3. **Codage des médias** ←
4. Circuits logiques
5. Architecture des ordinateurs
6. Conception et exécution de programmes
7. Algorithmique, programmation et structures de données

Rappel

- Idée fondamentale : on peut tout coder sous forme de nombres ou de séquence de nombres.
- Exemple de codage symboles-nombres (**correspondance**) :

Symbole	Codage proposé en base 10
	1
	2
	3

Codage des caractères | **Convention ASCII**

- Idée fondamentale : on peut tout coder sous forme de nombres ou de séquence de nombres.
- Exemple de codage symboles-nombres (**correspondance**) :

Symbole	Codage proposé en base 10
A	65
B	66
...	...

- Apparition dans les années soixantes pour pouvoir échanger dans les pays occidentaux (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)



Codage des caractères | **Convention ASCII**

- Un octet pour coder chaque caractère.
- Historiquement, 1 bit réservé au contrôle des erreurs (bit de parité).
⇒ 7 bits pour le codage, $2^7 = 128$ caractères différents.
- Contrôle d'erreur, exemple :
 - On fixe que tout octet "sain" doit comporter un nombre pair de bits valant 1.
 - Expéditeur choisit le bit de parité afin de respecter la parité.
On veut envoyer la séquence 0110111. Le bit de parité doit donc valoir 1.
 - Le destinataire vérifie que la parité est respectée.
On reçoit l'octet 00110111. Il y a eu une erreur.

Exemple : table ASCII des caractères datant de 1972

Bits					0	0	0	0	1	1	1	1
					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b ₄	b ₃	b ₂	b ₁	Column	0	1	2	3	4	5	6	7
↓	↓	↓	↓	Row ↓								
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Exemple : table ASCII des caractères datant de 1972

Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b ₄	b ₃	b ₂	b ₁	Column	0	1	2	3	4	5	6	7
				Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	-	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

$\text{cod}_{\text{ASCII}}("M") = 1001100$
 $= 0x4D$

(on ignore le bit de parité)



Codage des caractères | **Limitations d'ASCII**

- Problème d'ASCII ?



Codage des caractères | Limitations d'ASCII

- Problème d'ASCII ?
 - Variantes de caractères latins ?
 - Caractères d'autres langues ?
 - Caractères d'autres domaines (p.ex. mathématiques) ?
- Norme Latin-1 remplace le bit de parité par un bit supplémentaire pour coder les caractères. Le nombre de caractères codables (256) demeure faible.



Codage des caractères | **Convention ASCII**

- Influence historique.
- Impact encore :
 - URLs.
 - Adresses e-mail traditionnelles.
 - Noms de fichiers traditionnels.
 - Valeurs des caractères de l'alphabet latin dans le standard UTF.



Codage des caractères | **Standard Unicode**

- Années 1990.
- Établit un **standard** pour environ 150'000 caractères, ainsi que des règles d'une autre nature (exemple : sens de lecture).
- Nombre de bits alloués à chaque caractère : dépend du **type de codage** d'Unicode !
- Certains codage d'Unicode ne permettent pas de représenter tous les caractères.

Standard Unicode | **Points de code**

- Désigne les "numéros" correspondant aux caractères.
- La façon dont les caractères sont représentés en mémoire dépend du type de codage utilisé et n'est pas une constante du standard lui-même.
Exemple : ASCII code un sous-ensemble d'Unicode.
- Les points de code sont communément écrits en hexadécimal et préfixés par "U+".

Exemple : U+0041 → A , ou encore U+2207 → ∇ et U+1F643 → 🙄.



Comme en ASCII



Standard Unicode | **Codage en taille fixe**

- Simplicité et efficacité de l'implémentation.
- Temps d'accès et de traitement d'un caractère est constant.
- Exemple d'UCS-2 : 2 octets par caractère, de U+0000 à U+FFFF.
En l'occurrence : points de code et codage coïncident.
- Pourquoi utiliser une taille variable ?



Standard Unicode | **Taille variable**

- Pourquoi utiliser une taille variable ?
- Un texte écrit avec les caractères utiles à l'anglais (comme le code HTML d'une page web) n'a besoin que d'un octet par caractère.
⇒ Grand gaspillage de ressources (mémoire, réseau).
- Idée du codage à taille variable : les caractères les + fréquents sont codés sur moins de bits.
- La façon même de coder le caractère devient une fonction de ce dernier.

Standard Unicode | **Taille variable**

- Les caractères fréquents sont codés sur moins de bits.
- La façon même de coder le caractère devient une fonction de ce dernier.
Le code contient l'information sur la longueur du codage.
- Complexité accrue de l'algorithme de décodage : il faut parcourir les bits pour savoir où commencent et finissent les caractères !





Standard Unicode | **Exemple de l'UTF-8**

- Compose la grande majorité des pages Web.
- Caractères codés sur 1, 2 ou 3 octets.
- Environ 1 million de caractères définis.
- Coïncide avec Latin-1.



Standard Unicode | **Compatibilité de codage**

Exemple en Python d'incohérence entre codage et décodage :

```
original_string = "Ce codage n'est pas réussi !"
# Encodage de la chaîne en utf-8
encoded_string = original_string.encode('utf-8')
# Décodage de la chaîne encodée
decoded_string = encoded_string.decode('latin-1')
print(decoded_string)
```

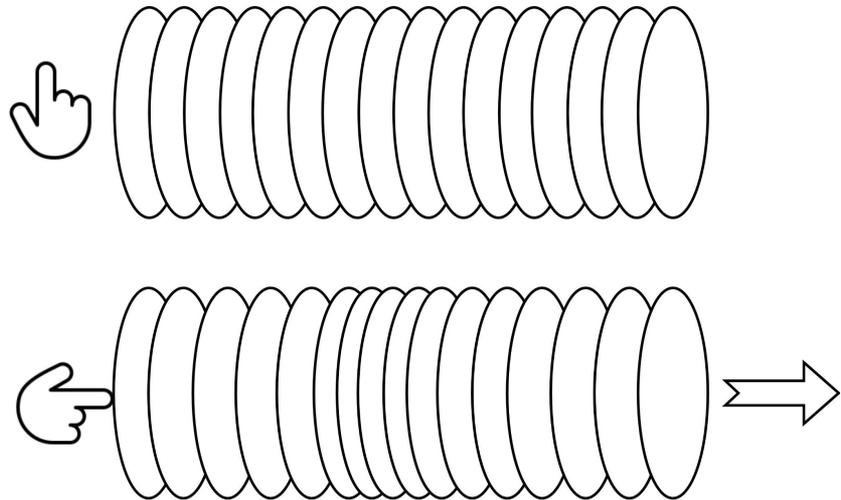
Note

- Le codage des caractères établit une correspondance entre une signification (un concept, comme la lettre "A") et une séquence de bits.
- Il ne dit rien sur la façon de coder la couleur des pixels qui affichent le caractère, ou la façon de coder l'agencement de l'encre sur une feuille de papier. Cela reste à voir.



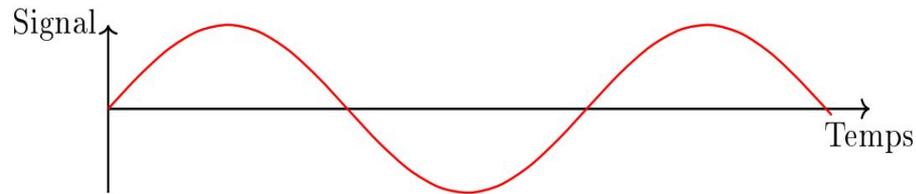
Codage des sons | Qu'est-ce qu'un son ?

- Onde longitudinale d'un milieu, compressions locales de la matière.
- Variation de pression peut être mesurée par des appareils comme un microphone ou l'oreille humaine.
- **Fréquence** [Hz] : taux de variation.
Correspond à la "note".
- **Intensité** [W/m^2] : Amplitude de la variation (souvent exprimée en dB).
Correspond au volume sonore



Codage des sons | **Mesure du signal**

- Appareil de mesure possible : convertir une **valeur de pression** acoustique en une valeur de tension **électrique**.
- En pratique : nous sommes contraints à **échantillonner** la réalité physique :



$P(t)$



$U(t_i)$



Codage des sons | **Mesure du signal**

- L'échantillonnage donne lieu à un signal digital, **discrétisé** aussi bien dans le temps que dans les niveaux de fréquence ou de volume.
- Heureusement, l'oreille humaine n'est pas parfaite et on peut se donner l'illusion que le signal est comme celui d'origine. Equivalent des dégradés de couleur.
- Typiquement : taux d'échantillonnage de 40 kHz.
⇒ Fichiers potentiellement volumineux !



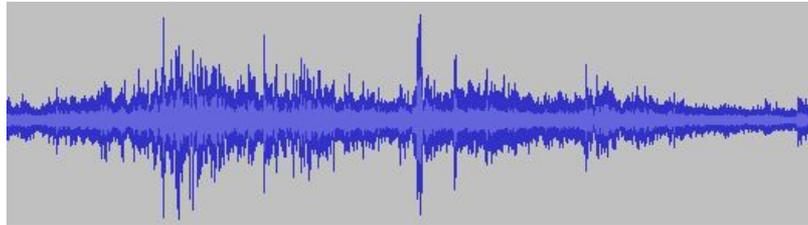
Codage des sons | **Types de codage**

- Coder la séquence $P(t)$ telle qu'**échantillonnée**.
Avantage : enregistrements de sons **quelconques**, non structurés.
Problème : **compression** de l'information souvent nécessaire.

- Coder une séquence **synthétique** de triplets (fréquence , volume , durée).
Avantages :
 - Codage économe de sons **structurés** comme la musique.
 - Caractère générique ; peut être **interprété** de différentes façons (par exemple avec timbres d'instruments distincts)
Désavantage : **spécifique** à certaines structures de sons.

Codage des sons | Types de codage

- Coder la séquence $P(t)$ telle qu'échantillonnée.



- Coder une séquence **synthétique**.





Exemple de format de synthèse : MIDI

- **M**usical **I**nstrument **D**igital **I**nterface.
- Conçu dans les années 1980 pour coder de l'information musicale. (C'est également une technologie audio et un protocole).
- Code la durée des notes, le volume, les notes elles-même, etc.
- Notes de musiques (gamme chromatique) codées sur 7 bits.

Exemple de format : MIDI

votamatic.unige.ch : HZTX

Notes de musiques codées sur 7 bits.
Combien d'octaves peut-on couvrir, à
raison de 12 "degrés" par gamme ?



Exemple de format : MIDI

votamatic.unige.ch : HZTX

Réponse : 7 bits permettent de coder
 $2^7 = 128$ états.

$128 / 12 \approx$ entre 10 et 11 octaves.





Exemple de format : MIDI | Table de codage des notes

Octave	Note numbers											
	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

Image : Davide Baccherini

(Diapositive hors champ)



Exemple de format : MIDI | Code Python

```
from midiutil import MIDIFile
mf = MIDIFile(1) # Une seule piste
track = 0
mf.addTrackName(track, time=0, trackName="Piste 1")
mf.addTempo(track, time=0, tempo=120)
liste_notes_midi = [60, 60, 60, 62, 64, -1, 62, -1, 60, 64, 62, 62, 60]
for i in range(len(liste_notes_midi)):
    note = liste_notes_midi[i]
    if note >= 0:
        mf.addNote(track,
                    channel=0,
                    pitch=note,
                    time=i,
                    duration=1,
                    volume=100)
filename = "ma_musique.mid"
with open(filename, 'wb') as outf:
    mf.writeFile(outf) # Ecriture du fichier sur le disque
```

(Diapositive hors champ)



Codage des images | **Types de codage**

- À l'instar de ce que l'on a vu pour les sons, deux approches possibles : échantillon du monde physique VS synthèse d'éléments.
- Images de type **raster** (ou **bitmap**) : on code la couleur de chaque pixel.
- Images de type **vectériel** : on code des éléments géométriques au sein de l'image.

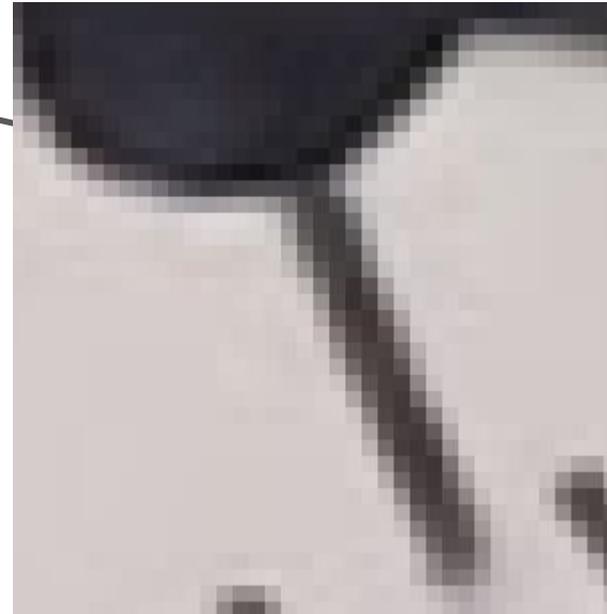


Codage des images | Types de codage

- À l'instar de ce que l'on a vu pour les sons, deux approches possible : échantillon du monde physique VS synthèse d'éléments.
- Images de type **raster** (ou **bitmap**) : on code la couleur de chaque pixel.
 - Exemples de formats : BMP, PNG, JPG, ...
 - Adapté pour la photographie.
- Images de type **vectoriel** : on code des éléments au sein de l'image.
 - Exemples de formats : EPS, SVG, ...
 - Adapté pour le dessin.
 - Adapté pour l'application de transformations (zoom, rotation, ...)

Codage des images | Types de codage

- Images de type **raster** (ou **bitmap**) : on code la couleur de chaque pixel.



Codage des images | Types de codage

- Images de type **raster** (ou **bitmap**) : on code la couleur de chaque pixel.

bleu	bleu	rouge	rouge
blanc	blanc	blanc	blanc
blanc	vert	bleu	rouge
rouge	rouge	rouge	orange

Codage des images | Bitmaps

- Images de type **raster** (ou **bitmap**) : on code la couleur de chaque pixel.
- Certains formats permettent une **compression**, d'autres pas.
- Exemple ci-contre de compression **sans perte** pour l'image bitmap précédente.

bleu x2	bleu	rouge x2	rouge
blanc x5	blanc	blanc	blanc
blanc	vert	bleu	rouge x4
rouge	rouge	rouge	orange



Images et sons | Compression de l'information

Format image	Format audio	Compression	Perte d'information
PPM, BMP	WAV	Non	Non
PNG, GIF	FLAC	Oui	Non
JPEG	MP3, AAC, OGG	Oui	Oui
WEBP	WMA	Oui	Mixte

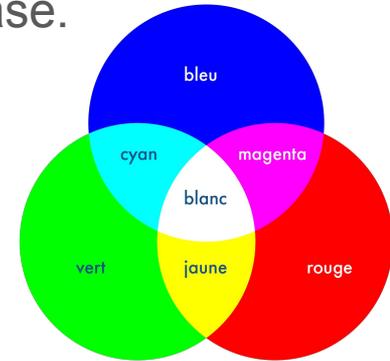


Codage des couleurs

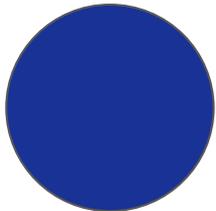
- Au sein de l'oeil humain : trois types de récepteurs sensibles à la longueur d'onde de la lumière (cônes).
⇒ Trois "composantes" de base que nous nommerons couleurs **primaires**.
Ces couleurs sont primaires **entre elles** : aucune ne peut être exprimée comme un mélange des autres.
- Un oeil sain distingue environ 10 millions de couleurs.
- D'un point de vue strictement physique, il n'y a pas de couleurs primaires, chaque couleur correspondant simplement à une longueur d'onde d'une onde électromagnétique.

Codage des couleurs | Synthèse additive

- En synthèse additive à 3 couleurs primaires, une couleur est décrite comme un mélange (combinaison linéaire) des 3 couleurs de base.

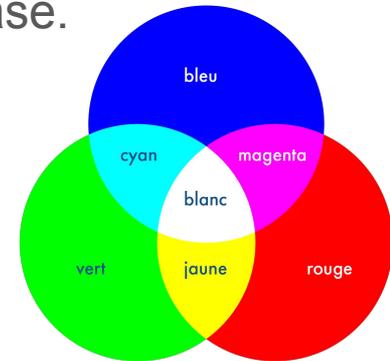


- Exemple :

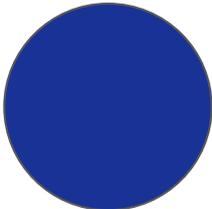


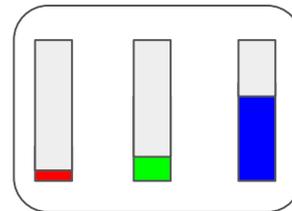
Codage des couleurs | Synthèse additive

- En synthèse additive à 3 couleurs primaires, une couleur est décrite comme un mélange (combinaison linéaire) des 3 couleurs de base.



- Exemple :


$$= 0.098 \mathbf{R} + 0.196 \mathbf{G} + 0.588 \mathbf{B}$$



Exemple de technologie | Écran LCD

- Écran à cristaux liquides.
- Chaque pixel contient trois "lampes", une par couleur primaire.

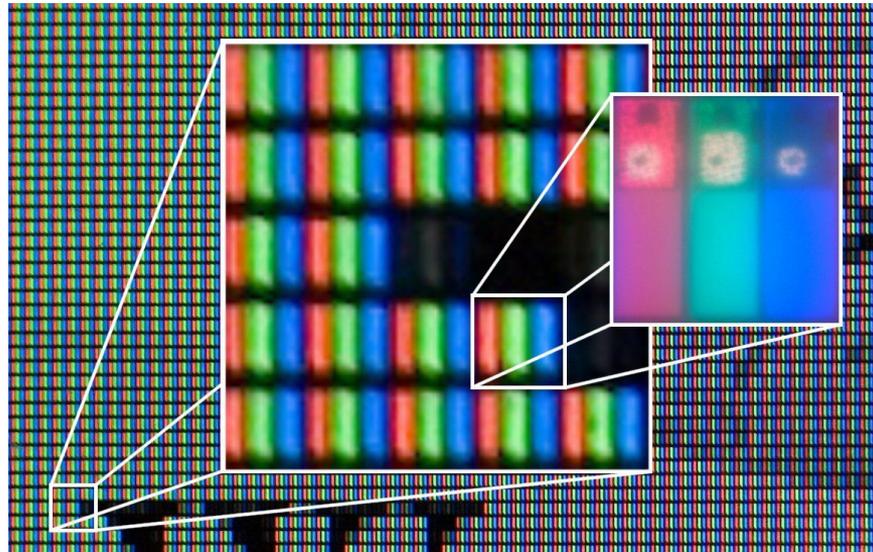


Image - Ravedave - CC BY 2.5



Codage RGB

- Une couleur est représentée par un triplet (R, G, B).
- Chaque composante est représentée par un nombre.



Codage des composantes RGB

Si on utilise un seul bit pour encoder le rouge, un seul pour le bleu et un seul pour le vert, combien de couleurs différentes peut-on exprimer ?

Réponse : 8 couleurs possibles car $2^3 = 8$.

Les couleurs sont alors codées par : 000, 100, 010, 001, 110, 011, 101, 111

Code	000	100	010	001	110	011	101	111
Couleur	Noir	Rouge	Vert	Bleu	Jaune	Cyan	Magenta	Blanc



Codage des composantes RGB

- Combien de bits faut-il par couleur primaire pour représenter plus de 10 millions de combinaisons ?



Codage des composantes RGB

- Combien de bits faut-il par couleur primaire pour représenter plus de 10 millions de combinaisons ?

$2^n 2^n 2^n = 2^{3n} > 10^7 \Leftrightarrow n = \log_2(10^7) / 3 \text{ approx } 7.75 \text{ bits}$
 \Rightarrow Un octet par couleur primaire semble adapté !

- Codage **RGB** standard : un octet par composante de couleur. Cela implique $2^8 = 256$ nuances par couleur primaire (de 0 à 255). On parle de **profondeur** de couleur.
- Codage **RGBA** : un octet de plus pour l'opacité.



Codage RGB hors du contexte informatique

- Exemple du "color picker" dans différents logiciels.
- Écriture décimale ou hexadécimale :
 - Décimale : triplet RGB comm (255, 127, 0) par exemple.
 - Hexadécimale : triplet #RGB comme #FF7F00 par exemple.



Codage RGB pour la manipulation d'images

Cf. démonstration en direct

(Diapositive hors champ)



Aparté : synthèse soustractive

- Exemple de CMJN utilisé en imprimerie.
- L'encre "ôte" de la lumière au papier blanc.
- Cyan, Magenta et Jaune absorbent chacune une partie du spectre.
- K (noir) ajouté pour des raisons techniques (coût, pureté du noir).

(Diapositive hors champ)