



Fiche ressource et connaissances

- Le code binaire

Le code binaire

Le code binaire est le langage utilisé par le microprocesseur de l'ordinateur pour traiter les informations. Le microprocesseur, c'est le cerveau de l'ordinateur. Grâce aux milliards de petits interrupteurs qui le composent, il envoie (ou reçoit) des signaux électriques. Avec ces impulsions électriques, on écrit le langage binaire, langage de l'ordinateur.

0 → pas de courant électrique / 1 → courant électrique

Chaque 0 ou 1 d'un numéro binaire est appelé **BIT** (Binary digiT). C'est une information binaire.



Microprocesseur d'un ordinateur

Le comptage binaire

Le système de comptage utilisé en mathématiques est appelé **système décimal**. C'est un **système en base 10** car il est basé sur **10 chiffres (de 0 à 9)**.

En informatique, le système de comptage est un **système en base 2** car basé sur **2 chiffres (0 et 1)**. Il s'agit du **système binaire**. Tous les réseaux informatiques et appareils informatiques fonctionnent en code binaire.

→ **Je sais compter en mathématiques avec un système décimal, mais comment fait-on pour compter en informatique avec un système binaire ?**

Comptage binaire sur 7 bits



Pour compter en binaire on a besoin du tableau ci-contre. Les colonnes sont nommées par le poids du bit. La première a pour poids « 1 », et de droite vers la gauche le poids du bit est doublé à chaque colonne (1 ; 2 ; 4 ; 8 ; 16 ; 32 ; 64...)

Comptage décimal

0 →

1 →

2 →

0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0

⋮

65 →

1	0	0	0	0	0	1
---	---	---	---	---	---	---

⋮

127 →

1	1	1	1	1	1	1
---	---	---	---	---	---	---

On peut ensuite convertir un nombre binaire en nombre décimal, et inversement.

La valeur 0 ou 1 de chaque bit agit comme un coefficient multiplicateur du poids.

Exemple de comptage sur 7 bits

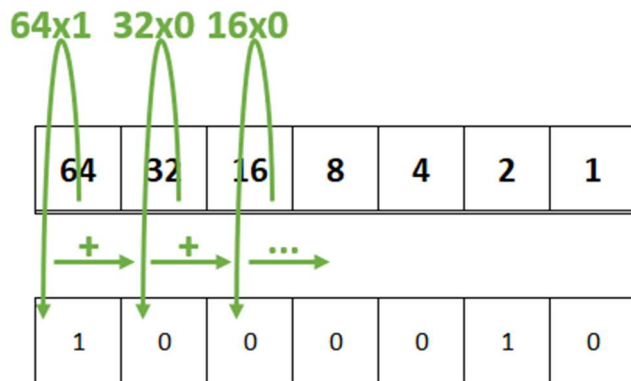


Pour le code binaire « 1000001 »

Du binaire vers le décimal

Si l'on regarde bien l'opération, on se rend compte que pour convertir un nombre binaire en décimal, il faut additionner les colonnes où le bit est à 1. Ici, il s'agit des colonnes 64 et 2.

Donc 100010 en binaire correspond à 66 en décimal.

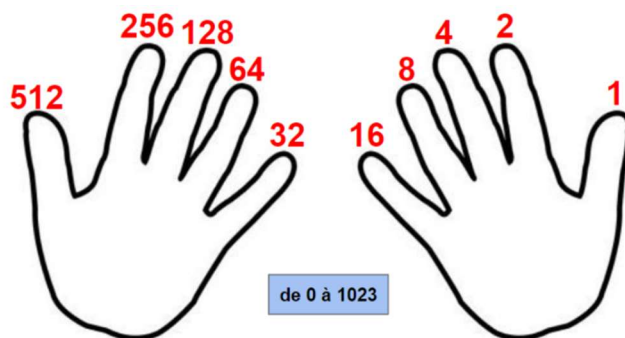
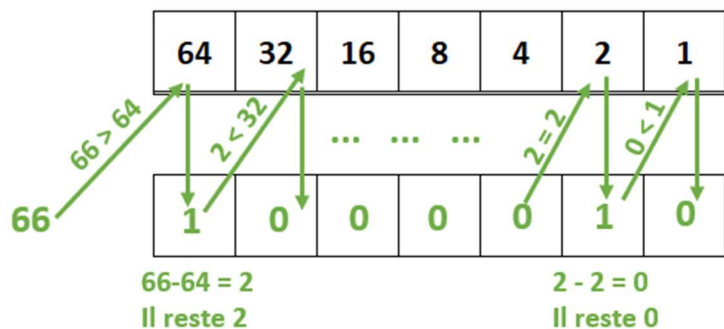


$$100010 = 64 \times 1 + 32 \times 0 + 16 \times 0 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 64 + 2 = 66$$

Du décimal vers le binaire

Pour convertir du décimal vers le binaire, c'est le système inverse. On compare notre numéro décimal au poids des bits par colonne en commençant par le plus grand poids.

Si le numéro décimal est supérieur au poids du bit, alors le bit est à 1. Sinon il est à 0.



On peut aussi compter en binaire avec ses doigts. Il faut pour cela donner un poids de bit à chaque doigt comme pour les colonnes. Ensuite si le bit est à 1 on lève le doigt sinon s'il est à 0, on laisse le doigt baissé.

La table ASCII

La table ASCII (prononcée ASKI) est une norme informatique de codage de caractère. Elle permet de définir 128 codes à 7 bits allant de 0 à 127. On s'en sert notamment pour les systèmes informatiques tels que les claviers d'ordinateurs.

La table ASCII

Nombre décimal	Nombre binaire	Touche de l'ordinateur			
0	0	Aucune touche	64	1000000	@
1	1	Début de rubrique	65	1000001	A
2	10	Début du texte	66	1000010	B
3	11	Fin du texte	67	1000011	C
4	100	Fin de rubrique	68	1000100	D
5	101	Demande, formulaire	69	1000101	E
6	110	Réponse à la demande	70	1000110	F
7	111	Signal sonore	71	1000111	G
8	1000	Retour arrière	72	1001000	H
9	1001	Tabulation horizontale	73	1001001	I
10	1010	Déplacement bas	74	1001010	J
11	1011	Nouvelle ligne de tabulation	75	1001011	K
12	1100	Saut de page	76	1001100	L
13	1101	Retour à la ligne	77	1001101	M
14	1110	Nouveau jeu de caractère	78	1001110	N
15	1111	Jeu de caractère précédent	79	1001111	O
16	10000	Données binaires	80	1010000	P
17	10001	USB 1	81	1010001	Q
18	10010	USB 2	82	1010010	R
19	10011	USB 3	83	1010011	S
20	10100	USB 4	84	1010100	T
21	10101	Réponse négative	85	1010101	U
22	10110	Synchronisation	86	1010110	V
23	10111	Fin de transmission	87	1010111	W

24	11000	Annulation	88	1011000	X
25	11001	Fin de bande	89	1011001	Y
26	11010	Erreur	90	1011010	Z
27	11011	Echap	91	1011011	[
28	11100	Tri fichiers	92	1011100	\
29	11101	Tri groupes	93	1011101]
30	11110	Tri enregistrement	94	1011110	^
31	11111	Tri des unités	95	1011111	_
32	100000	Espace	96	1100000	`
33	100001	!	97	1100001	a
34	100010	"	98	1100010	b
35	100011	#	99	1100011	c
36	100100	\$	100	1100100	d
37	100101	%	101	1100101	e
38	100110	&	102	1100110	f
39	100111	'	103	1100111	g
40	101000	(104	1101000	h
41	101001)	105	1101001	i
42	101010	*	106	1101010	j
43	101011	+	107	1101011	k
44	101100	,	108	1101100	l
45	101101	-	109	1101101	m
46	101110	.	110	1101110	n
47	101111	/	111	1101111	o
48	110000	0	112	1110000	p
49	110001	1	113	1110001	q
50	110010	2	114	1110010	r
51	110011	3	115	1110011	s
52	110100	4	116	1110100	t
53	110101	5	117	1110101	u
54	110110	6	118	1110110	v
55	110111	7	119	1110111	w
56	111000	8	120	1111000	x
57	111001	9	121	1111001	y
58	111010	:	122	1111010	z
59	111011	;	123	1111011	{
60	111100	<	124	1111100	
61	111101	=	125	1111101	}
62	111110	>	126	1111110	~
63	111111	?	127	1111111	effacer