

Objectifs

- Comprendre les codages binaires, décimaux, et hexadécimaux
- Appréhender le codage des caractères (ASCII et ASCII étendu)

1 Codage décimal (base 10)

- Que signifie " base 10 " ?
- La numération romaine est-elle en base 10 ?

2 Codage binaire (base 2)

Ici chaque élément peut coder 2 valeurs (0 ou 1) là où le système décimal encode 10 valeurs différentes dans chaque élément (0 à 9). Pour la justification de l'utilisation du binaire, allez voir la partie *Why Binary?* dans la partie Approfondissement en fin de TD.

- Que signifie numération binaire positionnelle ?

2.1 Du binaire au décimal

- Partez de cette définition pour compléter le tableau ci-dessous et convertir en base décimale les nombres exprimés en base 2 :

Binaire	Analyse	Décimal
0		
101	$1 * 2^2 + 0 * 2^1 + 1 * 2^0$	5
1		
110010		
111101		

2.2 Du décimal au binaire

- En utilisant la méthodologie proposée ci-dessous, convertissez en binaire la série de nombres en base 10 qui suit :

Méthodologie :

Binaire	0	0	0	0	0	1	0	0	1
Décomposition	$1 * 2^9$	$1 * 2^8$	$1 * 2^7$	$1 * 2^4$	$1 * 2^5$	$1 * 2^4$	$1 * 2^2$	$1 * 2^1$	$1 * 2^0$
Valeur décimale	256	126	64	32	16	8	4	2	1

Lecture : Si je veux représenter 9 je dois trouver la somme correspondante dans la ligne des valeurs décimales. En l'occurrence, c'est $8 + 1$. Je peux donc remonter à la ligne Binaire pour récupérer les "1" dont j'ai besoin, le reste ce sont des "0" : $8_{10} = 000001001_{12}$

Tableau à compléter :

Décimal	Analyse	Binaire
0		
1		
2		
3		
4		
5		
6		
7		
8		
9	$9 = 8 + 1 = 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$	1001
32		
66		
128		
255		

- Quel est l'entier maximal que l'on puisse coder sur 16 bits ?

3 Addition en Binaire

3.1 Additions en binaire

Règles d'additions en binaire:

- $0 + 0 = 0$, on pose 0
- $0 + 1 = 1$, on pose 1
- $1 + 1 = 10$: on pose 0 et on retient 1
- $1 + 1 + retenue = 11$: on pose 1 et on retient 1

Explication pour $1010_{(2)} + 0011_{(2)}$ (on lit de droite à gauche):

	1	0	1	0
+	0	0	1	1
Opérations	on pose 1	0 + la retenue , on pose 1	on pose 0 et on retient 1	on pose 1
Résultat	1	1	0	1

On a bien le résultat correspondant en système décimal : $10_{(10)} + 3_{(10)} = 13_{(10)}$

- A vous pour : $1111_{(2)} + 1111_{(2)}$

3.2 Conversion et addition

- Convertissez ces nombres en binaire, et additionnez les : $19_{(10)} + 71_{(10)}$

Table 1: Table d'addition hexadécimale

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

4 Codage hexadécimal (base 16)

- Quels sont les 16 chiffres du système hexadécimal ?
- Convertissez $4D5_{(16)}$, et $F48A_{(16)}$ en décimal.
- Addition en hexadécimal. En vous appuyant sur le tableau 1, combien vaut $D + C$ en base 16 ? Exprimez ensuite ce calcul en base 10.

5 Codage des caractères (Codage ASCII)

- Combien de codes différents existe-t-il dans l'ASCII standard (cf. Tableau 1 page 4)?
- Ce qui signifie qu'ils sont codés sur combien de bits ?
- Observez la table ci-après. Quel problème peut survenir lorsqu'on écrit en français?
- Quelle solution a été adoptée par l'ISO pour résoudre ce problème ? Quelle conséquence ?

Figure 1: Table ASCII (source : <http://www.asciitable.com/>)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@	96	60	140	##96;	`
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A	97	61	141	##97;	a
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B	98	62	142	##98;	b
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C	99	63	143	##99;	c
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D	100	64	144	##100;	d
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EM (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Source: www.LookupTables.com

Approfondissement

- Pourquoi choisir le binaire pour les ordinateurs (*Why Binary ?*) ? <https://www.youtube.com/watch?v=thrx3SBEPtL8>
- Un convertisseur Binaire-Décimal-Hexadécimal :
<http://sebastienguillon.com/test/javascript/convertisseur.html>
- Un exemple de problème d'encodage (ou *encoding*) :

	Encoding:Utf-8	Encoding:ISO-8859-1 (latin1)
Interprété comme:Utf-8	Étonnant que ça ait été si délicat à faire.	Étonnant que a ait t si délicat faire.
Interprété comme:Latin-1	Ã<89>tonnant que Ãa ait Ã©tÃ© si dÃ©licat Ã faire.	Étonnant que ça ait été si délicat à faire.